
Model-Agnostic Interpretable Machine Learning

Christoph Molnar

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

Vorgelegt von
Christoph Molnar
aus München

Eingereicht am 15.02.2022

Model-Agnostic Interpretable Machine Learning

Christoph Molnar

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

Vorgelegt von
Christoph Molnar
aus München

Eingereicht am 15.02.2022

Erster Berichterstatter: Prof. Dr. Bernd Bischl
Zweiter Berichterstatter: Prof. Dr. Giles Hooker
Dritter Berichterstatter: PD Dr. Fabian Scheipl

Tag der Einreichung: 15.02.2022
Tag der mündlichen Prüfung: 06.07.2022

Acknowledgments

This thesis would not have been possible without the help, support, guidance, and advice of many people! In particular, I would like to express my sincere gratitude to ...

- ... my supervisor Prof. Dr. Bernd Bischl for the freedom, trust, inspiring exchanges, support and advice in these years.*
- ... Prof. Dr. Giles Hooker and PD Dr. Fabian Scheipl for their willingness to act as the second and third reviewer for my Ph.D. thesis.*
- ... Prof. Dr. Christian Heumann and Prof. Dr. Helmut Küchenhoff for their availability to be part of the examination panel at my Ph.D. defense.*
- ... the Centre Digitisation.Bavaria (ZD.B) and the Bavarian Research Institute for Digital Transformation (bidt) for financial support during my work on these projects and their wonderful Ph.D. program.*
- ... all my coauthors for the fruitful collaborations.*
- ... all members of my working group: Thank you for the great collaboration, inspiring discussions and unforgettable time.*
- ... all remaining former and current colleagues at the Department of Statistics for the excellent general atmosphere.*
- ... my parents, my brothers and my wife who are always there for me.*

Zusammenfassung

Das maschinelle Lernen (ML) wird mit zunehmender Häufigkeit in Produkten und Prozessen eingesetzt. Standardmäßig funktionieren die meisten ML-Modelle als sogenannte "Black Boxes", was es schwierig macht, Erkenntnisse zu gewinnen, Vertrauen zu schaffen, die Modelle zu debuggen und individuelle Vorhersagen zu erklären. Das Gebiet des interpretierbaren maschinellen Lernens (IML) adressiert diese Probleme und hat zum Ziel, das Verhalten von ML-Modellen und deren Vorhersagen zu erklären.

Diese kumulative Dissertation besteht aus 10 Beiträgen, die sich alle mit modellagnostischen IML-Methoden beschäftigen. Modellagnostische IML-Methoden arbeiten so, dass es keine Rolle spielt, ob das zu interpretierende Modell ein neuronales Netz oder ein Entscheidungsbaum ist. Diese Arbeit trägt insbesondere in zweierlei Hinsicht zur IML-Forschung bei: Konsolidierung modellagnostischer IML-Methoden und Verbesserung etablierter modellagnostischer Interpretationsmethoden, insbesondere der Permutation Feature Importance (PFI) und des Partial Dependence Plot (PDP).

Als Beitrag zur Konsolidierung präsentiert diese Arbeit eine kurze Geschichte von IML, den aktuellen Stand der Technik und zukünftige Herausforderungen. Diese Herausforderungen sind oft mit allgemeinen Fallstricken verbunden, mit denen Anwender bei der Verwendung von IML-Methoden zur Interpretation von Modellen konfrontiert werden, wobei ein häufiger Fallstrick abhängige Features sind. In dieser Arbeit werden viele dieser allgemeinen Fallstricke identifiziert und mögliche Lösungen beschrieben. Außerdem wird SIPA vorgestellt, was für Sampling, Intervention, Prediction und Aggregation steht - ein allgemeines Schema, nach dem die meisten modellagnostischen Methoden funktionieren. Auf der Grundlage des SIPA-Schemas wurde *iml*, ein R-Softwarepaket für die modellagnostische Interpretation von maschinellem Lernen, implementiert.

PDP und PFI sind etablierte IML-Methoden, die zur Beschreibung von Featureeffekten und -wichtigkeit verwendet werden. Viele Limitationen und Verbesserungsmöglichkeiten dieser Methoden sind jedoch bisher nicht ausreichend erforscht - eine Lücke, die diese Arbeit füllt. Sowohl PFI als auch PDP können irreführende Erklärungen liefern, wenn es zur Extrapolation in unwahrscheinliche Regionen des Featuresraumes aufgrund von abhängigen Features kommt. Daher wird in dieser Arbeit vorgeschlagen, PDP und PFI in Untergruppen der Daten zu berechnen. Der Untergruppenansatz reduziert das Problem der Extrapolation erheblich und ermöglicht eine differenziertere Interpretation von Featureeffekten und -wichtigkeit. Eine weitere Möglichkeit, das Problem der abhängigen Features für PFI zu lösen, ist die Verwendung von conditional PFI, bei dem die Permutation des Features den Zusammenhang mit den anderen Features berücksichtigt. In dieser Arbeit wird die Relative Feature Importance eingeführt, die die conditional PFI verallgemeinert, indem sie die Konditionierung auf beliebige Features erlaubt und somit die Analyse des indirekten Einflusses von Features ermöglicht. Darüber hinaus werden IML-Methoden häufig verwendet, um Schlussfolgerungen über die reale Welt zu ziehen. Dies wirft die Frage auf, unter welchen Bedingungen die Modellinterpretation auf die reale Welt ausgedehnt werden darf und wie mit verschiedenen Arten von Unsicherheit umgegangen werden muss. In dieser Arbeit werden die Bedingungen untersucht, unter denen statistische Inferenz mit PDP und PFI möglich ist. Darüber hinaus werden Verbindungen zwischen PDP und PFI untersucht, und auf Grundlage dieser Gemeinsamkeiten werden neue Visualisierungen für PFI vorgeschlagen.

Darüber hinaus schlägt diese Arbeit mehrere modellagnostische Metriken für die Modellkomplexität vor, die auf einer funktionalen Dekomposition mit Accumulated Local Effects basieren. Mit diesen Metriken kann ein Modell nicht nur auf seine Leistung, sondern auch auf seine Interpretierbarkeit hin optimiert werden.

Neben PDP und PFI wird in dieser Arbeit eine neue Methode für Counterfactual Explanations vorgestellt, die zur Erklärung einzelner Modellvorhersagen verwendet werden können. Die vorgeschlagene Suchprozedur für Counterfactual Explanations wird als multikriterielles Optimierungsproblem formuliert, welche es dem Benutzer ermöglicht, den richtigen Kompromiss zwischen verschiedenen Metriken für die Counterfactual Explanations zu wählen.

Summary

Machine learning (ML) is increasingly making its way into products and processes. By default, most ML models operate as black boxes, making it difficult to derive insights, gain trust, debug the models and explain individual predictions. The field of interpretable machine learning (IML) addresses these shortcomings and aims to explain the average behavior of ML models and individual predictions.

This cumulative dissertation consists of 10 contributing articles, all dealing with model-agnostic IML methods. Model-agnostic IML methods work in such a way that it does not matter whether the model being interpreted is a neural network or a decision tree. In particular, this thesis contributes to IML research in two ways: consolidating model-agnostic IML methods and improving established model-agnostic interpretation methods, especially Permutation Feature Importance (PFI) and the Partial Dependence Plot (PDP).

As a contribution towards consolidation, this thesis presents a brief history of IML, the current state-of-the-art and future challenges. These future challenges are often associated with general pitfalls faced by practitioners in using IML methods to interpret models, with a common pitfall being dependent features. Furthermore, this thesis identifies many of these general pitfalls and describes possible remedies. Further, SIPA is introduced, which stands for sampling, intervention, prediction and aggregation – a shared framework by which most model-agnostic methods operate. Based on this shared framework, *iml*, an R software package for model-agnostic machine learning interpretation, was implemented.

PDP and PFI are well-established methods in IML that are used to describe feature effects and feature importance. However, many limitations and potential improvements have not been adequately explored – a gap that this thesis fills. Both PFI and PDP can be misleading if the features are dependent due to extrapolation in unlikely regions of the feature space. Therefore, this thesis proposes to compute PDP and PFI in subgroups of the data. The subgroup approach greatly reduces the problem of extrapolation and allows for a more nuanced interpretation of feature effects and importance. Another way to address the dependent feature problem for PFI is to use conditional PFI, where the permutation of the feature is conditional on all other features. This thesis introduces relative feature importance which generalizes the conditional PFI by allowing to condition on arbitrary feature subsets, and allowing indirect influence of features to be studied. Moreover, IML is often used to draw conclusions about the real world. This raises the question of the conditions under which the model interpretation may be extended to the real world and how to deal with various types of uncertainty. This thesis examines the conditions under which statistical inference with PDP and PFI might be possible. Further, connections between PDP and PFI are studied, and, based on commonalities, new visualizations for PFI are proposed.

In addition, this thesis proposes several model-agnostic metrics for model complexity based on functional decomposition with accumulated local effects. With these metrics, a model can be optimized not only for performance, but also for interpretability.

Besides PDP and PFI, this thesis introduces a new method for counterfactual explanations that can be used to explain individual model predictions. The proposed counterfactual search is formulated as multi-objective optimization problem, which enables the user to choose the right trade-off between different objectives for the counterfactual explanation.

Contents

I. Introduction and Background	1
1. Introduction	3
1.1. Motivation and Scope	3
1.2. Outline	5
2. Methodological and General Background	7
2.1. Supervised Machine Learning	7
2.2. Interpretability	8
2.2.1. Stakeholders	8
2.2.2. Definition of Interpretability	9
2.2.3. Intrinsically Interpretable Models	10
2.3. Model-agnostic Interpretable Machine Learning	11
2.3.1. Definition	11
2.3.2. Methods Overview	12
2.3.3. Analyzing Model Components vs. Behavior	13
2.3.4. SIPA Framework	14
2.4. Model-agnostic Interpretation Methods	15
2.4.1. Partial Dependence Plots	15
2.4.2. Accumulated Local Effect Plots	16
2.4.3. Feature Interactions	20
2.4.4. Permutation Feature Importance	20
2.4.5. Counterfactual Explanations	21
2.5. Dependent Features and Extrapolation	22
3. Concluding Remarks and Future Work	27
3.1. Dependent Features – A More Fundamental Problem?	27
3.2. Do We Really Have to Define Interpretability?	28
3.3. Can IML Reveal Insights About the Real World?	29
3.4. Which Uncertainty?	29
II. Contributions	31
Contributing Publications	33
4. Interpretable Machine Learning—A Brief History, State-of-the-Art and Challenges	35
5. General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models	51
6. iml: an R package for Interpretable Machine Learning	83
7. Sampling, Intervention, Prediction, Aggregation: A Generalized Framework for Model-Agnostic Interpretations	87

8. Model-agnostic Feature Importance and Effects with Dependent Features - A Conditional Subgroup Approach	101
9. Relating the Partial Dependence Plot and Permutation Feature Importance to the Data Generating Process	143
10. Relative Feature Importance	175
11. Visualizing the Feature Importance for Black Box Models	185
12. Quantifying Model Complexity via Functional Decomposition for Better Post-Hoc Interpretability	203
13. Multi-objective counterfactual explanations	217
Further References	241

Part I.

Introduction and Background

1. Introduction

It is one thing to have a model with which one can make accurate predictions; it is another to have a model that makes accurate predictions for the right reasons.

– Simulation and Similarity, Michael Weisberg (2012)

1.1. Motivation and Scope

Machine learning (ML) is increasingly used for automating tasks and making decisions. Also, many scientific applications rely on ML, for example in ecology (Bair et al., 2013; Esselman et al., 2015; Obringer and Nateghi, 2018), medicine (Boulesteix et al., 2020; Stiglic et al., 2020; Pintelas et al., 2020), the social sciences (Stachl et al., 2020; Zhao et al., 2020), and many more fields. For example, Obringer and Nateghi (2018) used ML to predict water reservoir levels for cities, based on population, soil moisture, water use, precipitation and so on.

In supervised ML, the focus is on optimizing a loss function on unseen data. Flexible, but well tuned models often outperform simpler models in performance on test data, and ensembles of different models often further outperform individual models. Before ML became more popular, the classical statistical modeling approach was (and still is) used in applications that focus on generating insights. The statistical approach prioritizes considerations of the data-generating process over predictive performance on unseen test data. These considerations of the data-generating process is what makes the statistical modeling approach often more interpretable: model parameters are usually connected to a concept of the data-generating process.

The focus of ML on test performance pushes interpretability of the model into the backseat. For structurally more restricted models, such as decision rules or linear models, individual components of the model can be interpreted in isolation and be related to understandable concepts. For example, a coefficient of a linear regression model can be mapped to an individual feature and be interpreted, more or less, in isolation. The terminal node of a decision tree gives us a list of binary decisions that lead to a certain prediction. Structurally less restricted models that are well tuned, such as neural networks and gradient boosted trees, often perform well but don't have a direct mapping of model parameter to feature effects or concepts, making interpretation more difficult.

As more machine learning is used, the need for tools to understand their decision processes grows. Regulated industries such as banks or health care require auditability of their products and processes, for which it's necessary to look inside how the model operates. ML models can encode social biases such as gender bias (Prates et al., 2019) and learn harmful, non-causal relationships (Caruana et al., 2015; Ribeiro et al., 2016b; Lapuschkin et al., 2019). In all these cases, the model performance on test data might not reveal the problem. ML is also increasingly used in science, an endeavor that requires interpretability for different reasons (Roscher et al., 2020): Explaining the world lies at the very heart of science. Using an opaque model to describe a phenomenon therefore seems like a step in the wrong

direction. However, in many situations, using the ML model approach might produce complex models that outperform state-of-the-art mechanistic models or classical statistical models. The opaque model might generalize better, but as the relationships of interest are encoded in the model, they are hidden from the researchers. All these different needs (auditability, fairness, knowledge generation, ...) can be addressed by making the models more interpretable.

Interpretable machine learning (IML) is a research field concerned with extracting knowledge from machine learning models and explaining individual predictions. The IML field covers (1) inherently interpretable models, (2) modifications of more complex models to make them more interpretable, and (3) post-hoc interpretation methods. This thesis focuses on model-agnostic interpretation methods, which are post-hoc methods that are applied on a trained model and have no influence over how the model is trained. Model-agnostic means that the IML methods can be applied to any ML model, as they do not rely on access to, for example, the model parameters, but only need access to the prediction function (Ribeiro et al., 2016a). For model-agnostic methods it does not matter whether the underlying model is a neural network, a random forest or a linear model. This allows a certain modularity, as the same explanation method can be used even when the underlying model is exchanged for a better performing model. Furthermore, model-agnostic methods allow comparison of models, which is not always possible for different interpretable model classes: We cannot compare the coefficients of a logistic regression with the splits made by a decision tree. But we can compare the PFI of both models, and even with other, more complex models.

Explaining “black box” ML models for high stakes decisions has been criticized (Rudin, 2019; Rudin et al., 2021) in favor of ML models that are interpretable by design. As we argue in Molnar et. al (2020), interpretable models should always be included in benchmarks, and a decision for a “black box” model has to be justified with a relevant increase in performance. The critique by Rudin (2019) centers on choice of model, but is not an argument against the use of model-agnostic interpretation method per se. Using an interpretable model does not prohibit the application of model-agnostic interpretation methods. On the contrary, model-agnostic interpretation methods can create additional insights into the model. While, for example, a decision rule list might be interpretable to some degree, the interpretation of the model might not tell us the average effect of a feature. Model-agnostic feature effect plots, such as the partial dependence plot, can be applied to answer this question.

While the field of ML interpretability has roots in statistical modeling and rule-based ML (Molnar et. al, 2020), many of the model-agnostic methods are relatively new and consolidation is necessary. In Scholbeck et. al (2020) we showed that many model-agnostic interpretation methods work under the SIPA framework of sampling, intervention, prediction and aggregation. The R package ‘iml’, was designed under the SIPA principle (Molnar et al., 2018) and implements many methods in one package: partial dependence plots (PDP), accumulated local effect (ALE) plots, individual conditional expectation (ICE) curves, permutation feature importance (PFI), the H-statistic, Shapley values, LIME and tree surrogate models. There are further commonalities between model-agnostic methods: PFI can be split into importances for individual data points and there are many parallels between PFI and PDP (Casalicchio et. al, 2019). A conditional version of PFI can also be extended to include features that were not used by the model (König et. al, 2021), which allows to study indirect influence of features and biases/fairness.

As many model-agnostic interpretation methods manipulate features individually, they show unexpected and possibly unwanted behavior when features are dependent. When features are correlated, many model-agnostic methods create new data that have a different distribution than the training data, and might even lie outside the convex hull of the data and represent physically impossible entities. As part

1.2 Outline

of this thesis, I look at the behavior of PDP and PFI when feature are dependent and suggest using regression trees to compute the PFI in subgroups (Molnar et. al, 2020). Misleading interpretation due to feature dependence is one of many pitfalls that one can run into when interpreting a model. Other pitfalls include a wrong causal interpretation and failing to account for interactions (Molnar et. al, 2020).

IML can be seen as “descriptive statistics” of models, and, as such, often lack uncertainty quantification. In the contribution Molnar et. al (2021), we proposed how PDP and PFI, which are external model descriptors, can be turned into an inferential statistical tool. We propose to use these same external descriptors (PDP/PFI) directly on the data-generating process, at least in theory and simulation, to define a ground truth that the model aims to recover. This allows us to conduct inference, but needs the additional, strong assumption of model unbiasedness.

Interpretation methods can also be used to quantify the interpretability of a model itself. In Molnar et. al (2020) we used ALE plots to quantify the interaction strength, the average complexity of the main effects of the features and the number of features. We constructed these measures to be model-agnostic, meaning they can be compared across different model types. We demonstrated that those measures can be used as additional objectives (next to predictive performance) for model selection.

While PDP and PFI are used to describe the average behavior of models, methods such as counterfactual explanations explain individual predictions. Counterfactual explanations describe minimal changes to the features of a data instance so that the prediction is substantially changed. We translated the search for counterfactuals into a multi-objective optimization problem (Dandl et. al, 2020), which allows to generate multiple counterfactuals, with different trade-offs. The user can then choose counterfactuals with the application-specific best trade-off.

1.2. Outline

Part I describes the methodological background of this thesis and Part II presents the individual paper contributions.

Section 2.1 in Part I introduces the concept of supervised ML. Section 2.2 discusses the need for interpretability by various stakeholders (Section 2.2.1) and why interpretability is so difficult to define (Section 2.2.2). An option for making ML more interpretable is to work with intrinsically interpretable models. This approach has some drawbacks, as described in Section 2.2.3.

This thesis focuses on model-agnostic interpretation methods that can be applied to any ML model in a post-hoc fashion, that is, after the model was trained. Section 2.3 provides a definition of the model-agnostic interpretation approach (Section 2.3.1), followed by an overview of available methods (Section 2.3.2). Section 2.3.3 compares model-agnostic interpretation with model-specific interpretation based on inspecting the learned parameters and structures. Most model-agnostic interpretation methods follow the same framework of sampling, intervention, prediction and aggregation (Section 2.3.4), the principle after which the R package ‘iml’ was designed.

The model-agnostic interpretation methods that were central to this thesis are presented in Section 2.4: The partial dependence plot (Section 2.4.1), accumulated local effect plots (Section 2.4.2), permutation feature importance (Section 2.4.4) and counterfactual explanations (Section 2.4.5).

Section 2.5 explains why dependent features cause problems for many interpretation methods. Section 3 concludes Part I with thoughts on the future of model-agnostic interpretation methods.

2. Methodological and General Background

This chapter provides the methodological background for this thesis on machine learning, interpretability and model-agnostic interpretation methods.

2.1. Supervised Machine Learning

Supervised ML is a form of inductive learning: The goal is to learn general rules from specific data examples.

An unknown data-generating process produces data that follows a distribution \mathbb{P}_{XY} . A data point from the data-generating process \mathbb{P}_{XY} consists of a p -dimensional feature vector $\mathbf{x}^{(i)} \in \mathcal{X}$ and the target $y^{(i)} \in \mathcal{Y}$. To refer to a subset of features, $\mathbf{x}_S^{(i)} \in \mathcal{X}_S$ with $S \subseteq \{1, \dots, p\}$ is used, \mathcal{X}_S being a $|S|$ -dimensional feature subspace. Complimentary, \mathcal{X}_C denotes the remaining features so that $S \cup C = \{1, \dots, p\}$ and $S \cap C = \emptyset$. The goal in supervised learning is to approximate the unknown function $f : \mathcal{X} \mapsto \mathbb{R}^g$ that maps from p features X to a prediction in \mathbb{R}^g , e.g. $g = 1$ for regression, and g is the number of classes for classification. We call the approximation of f by the ML model \hat{f} . This function \hat{f} is induced by training the model on a dataset drawn from \mathbb{P}_{XY} . Multiple draws comprise a dataset $\mathcal{D}_n = \left((\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(n)}, y^{(n)}) \right)$. Here, n denotes the number of draws, with each data point coming from the joint distribution $\mathcal{D}_i \sim \mathbb{P}_{XY}$, $i \in \{1, \dots, n\}$. This dataset \mathcal{D}_n is used to induce the model \hat{f} . The induction is done by an inducer algorithm: $I : \mathcal{D} \times \Lambda \rightarrow \mathcal{H}$, and the inducer maps from the hyperparameter space Λ and set of all datasets \mathcal{D} to the function hypothesis space \mathcal{H} that is defined by the ML model class the inducer can produce. For example, if the inducer only produces ML models that are linear in the feature space, only functions of the form $\hat{f}(\mathbf{x}^{(i)}) = \beta_0 + \beta^T \mathbf{x}^{(i)}$, $\beta \in \mathbb{R}^p$ are in the hypothesis space \mathcal{H} . The induction process is an optimization process in which a risk is minimized: $\mathcal{R}(\hat{f}) = \mathbb{E}_{XY}[L(Y, \hat{f}(X))] = \int L(Y, \hat{f}(X)) d\mathbb{P}_{XY}$. The risk requires a loss function $L : \mathcal{Y} \times \mathbb{R}^g \rightarrow \mathbb{R}_0^+$.

After model induction, the model is evaluated on a separate dataset to avoid an over-confident empirical risk estimation. For separation of training and testing, the dataset \mathcal{D}_n is split into a dataset \mathcal{D}_{n_1} for induction and a dataset \mathcal{D}_{n_2} for testing, so that $n_1 + n_2 = n$. The empirical risk on the test data is defined as:

$$\mathcal{R}_{\text{emp}}(\hat{f}_{\mathcal{D}_{n_2}, \lambda}) := \frac{1}{n_2} \sum_{i=1}^{n_2} L(y_i, \hat{f}_{\mathcal{D}_{n_2}, \lambda}(\mathbf{x}_i))$$

The resulting performance estimate is subject to variance. To get more stable results, resampling strategies such as bootstrapping or cross-validation can be employed. I denote sets of indices for training data with B_d in the d -th split repetition and for the evaluation data B_{-d} , so that $B_d \cup B_{-d} = \{1, \dots, n\}$. The training data might further be (repeatedly) split into training and validation datasets. The purpose

of the validation dataset is to find a good configuration of model parameters that are not optimized in the training phase itself (for example the architecture of a neural network).

2.2. Interpretability

This section discusses who needs interpretability for ML and what interpretability is. Furthermore, it gives an overview of IML methods, examines IML's roots in statistics and rule-based ML and discusses the differences between interpretable models vs. post-hoc interpretation.

2.2.1. Stakeholders

As more complex ML models are used to enhance products, to automate processes and even to conduct scientific research, the demand for interpretation increases. To understand the demand for IML, we have to consider all the different stakeholders that are involved (Tomsett et al., 2018): creators, operators, executors, decision subjects and examiners.

Carefully controlling the model performance by the model creators is critical, but there are many problems that can occur despite good test performance. For example, the prediction might rely on non-causal artifacts, for example asthma as predictor for improved pneumonia outcomes (Caruana et al., 2015), presence of snow to decide between wolves and dogs (Ribeiro et al., 2016b) and presence of watermarks for image classification (Lapuschkin et al., 2019). These artifacts might not show up in performance evaluation, but when using IML methods. IML methods enable model creators to debug the model and compare, for example, feature importance values with expert knowledge.

Operators are the people who work with the model output and executors are the ones that act on that information. For a sepsis warning system in a clinic, the nurses were the operators (receiving the patients sepsis scores on a tablet) and the doctors acted on the scores (by paying closer attention to that patient and giving antibiotics) (Sendak et al., 2020). The lack of model interpretability made the communication between nurses and doctors challenging and nurses actually tried to fill the lack of interpretation by providing their own, which was not always correct and could have had unintended consequences (Elish and Watkins, 2020).

The people who are affected by the decisions are called decision subjects. This could be the person that got their loan application rejected or who was diagnosed by a data-driven algorithm with a disease. These decisions can range from having negligible to major impacts on the life of people. Interpretability can enable the person to understand the decision and to take educated further steps. Interpretability might also be required to challenge decisions. ML models can encode social biases, such as gender bias (Prates et al., 2019) and learn harmful, non-causal relationships. Here, especially local explanation methods can help, which can justify individual predictions.

Examiners are external stakeholders that test, audit or otherwise investigate the model. Especially regulated industries such as finance and health care require auditing of algorithms. Important tools in the auditor's toolbox are ML interpretation methods (Johner et al., 2021).

ML is also increasingly used in scientific discovery (Roscher et al., 2020). However, replacing mechanistic or classical statistical models with ML leads to a partial loss of understanding the studied phenomenon. Interpretable machine learning allows the scientist not only to model the phenomenon

2.2 Interpretability

at hand, but to also learn more about the relationships between features and target, and thus about the data-generating process.

2.2.2. Definition of Interpretability

A big criticism of IML is that interpretability is not well defined (Lipton, 2018; Doshi-Velez and Kim, 2017). Miller (2019) tries to define interpretability in the following way: “Interpretability is the degree to which a human can understand the cause of a decision”. But this only moves the burden to defining “human understanding”, and therefore still remains vague as it is unclear when a human has “understood” the decision. A simulatability-based definition says: “Interpretability is the degree to which a human can consistently predict the model’s result” (Kim et al., 2016). This definition allows a way to quantify interpretability (correlation between human and model prediction), but is rather narrow. For example, measures of feature importance do not directly allow to predict the model’s results, but still offer valuable insights into the workings of the model.

How can research on interpretable machine learning be conducted when the defining element of the field, “interpretability” is not accurately defined? To answer this question and justify the field, I want to distinguish between the use of “interpretability” as a keyword and as a measurable quantity.

“Interpretable machine learning” and “explainable artificial intelligence” are useful keywords to bundle approaches with the shared goal of making machine learning models more transparent. The keywords draw together research areas such as statistics, rule-based ML, sensitivity analysis, social science and more. In the interpretability-as-keyword case, I would argue that we do not require a mathematical definition of interpretability. Similarly, the field of “deep learning” does not have a formal definition of when a neural network is deep.

The more severe criticism applies to the use of interpretability as measurable quantity. In ML research, new approaches can be benchmarked against the state-of-the-art in terms of predictive performance. But for IML, there is no ground truth to which an explanation can be compared to. This complicates research on interpretability. How can researchers “prove” that their approach is more interpretable than another? The answer to this question is far reaching, as the scientific evaluation of methods is tied to the definition of interpretability.

There is no conclusive, mathematical definition that can say when an ML model is explainable or interpretable. However, as Rudin (2019), Molnar et. al (2020) and Askira-Gelman (1998) argue, the answer might be that we don’t need one single definition of interpretability, but rather multiple aspects of interpretability. We can divide these measurable aspects of interpretability into: (1) human-based evaluations on a proxy task or in the real application and (2) function level or mathematical evaluations (Doshi-Velez and Kim, 2017).

Human-based evaluations in scientific studies are often based on proxy tasks, as an evaluation in the real application is more difficult to implement. Measurements include, for example, how much an interpretation improves a human’s task performance (Dhurandhar et al., 2017; Zhou et al., 2018; Plumb et al., 2019), the user’s ability to predict the outcome (Zhou et al., 2018), the user’s ability to reproduce a model’s output for a given input (Friedler et al., 2019; Poursabzi-Sangdeh et al., 2018), the user’s ability to predict how the prediction will change given a change in features (Friedler et al., 2019), how closely users follow the prediction of a model (Poursabzi-Sangdeh et al., 2018), how well users detect model errors (Poursabzi-Sangdeh et al., 2018), the user’s response times (Huysmans et al., 2011) and answer confidence (Huysmans et al., 2011). This selection shows how diverse the evaluation can be.

And even behind single measures listed here, for example the first criterion “how much interpretation helps with task” is task-dependent and can take on very diverse shapes. Human-based evaluations can also have very different outcomes based on prior knowledge of the target audience. For example, humans who are educated in linear regression models will work more successfully with interpreting a coefficient table than people who see such a table for the first time. A look into the social sciences (Miller, 2019) tells us that many properties that constitute a good explanation are in conflict with each other. This is another argument that even for humans no single best-performing explanation exists, but interpretability is more of a multi-objective problem.

Function level evaluations are derived analytically from the model itself. Examples are sparsity, linearity, and monotonicity in feature effects. Another example is model size (e.g. number of non-zero coefficients or length of decision rules) which is a model-class-dependent criterion for interpretability (Huysmans et al., 2011; Rüping et al., 2006; Askira-Gelman, 1998; Yang et al., 2017; Schielzeth, 2010; Lakkaraju et al., 2017; Fürnkranz et al., 2012; Ustun and Rudin, 2016). Model complexity cannot tell us whether a decision tree is more interpretable than a linear model, but we can compare the maximum tree depth of two decision trees. Other measures include fidelity, i.e., how well an explanation predicts the model outcome (Plumb et al., 2019). Measurable dimensions of interpretability even allow to optimize a model not only for predictive performance, but also for interpretability. Examples of these dimensions are sparsity in the features that were used, monotonicity in the feature effects, sparsity in the explanations and so on. In Molnar et. al (2020), we proposed quantifiable measures of model complexity, that can be computed in a model-agnostic way and which can be used in multi-objective optimization, as we show in an application.

I will use the word “interpretability” throughout the thesis for referring to extracting knowledge about a model and “explanation” for explaining individual predictions.

2.2.3. Intrinsically Interpretable Models

The field of interpretable modeling is old: linear regression model have been around for over 200 years and rule-based ML methods were developed in the mid of the 20th century. In this chapter, we briefly revisit basic “intrinsically interpretable models” as they constitute building blocks for many model-agnostic interpretation methods. Two pillars of IML are statistical regression models and rule-based ML models. Both model families are collections of supervised ML approaches, as defined in Section 2.1, and are deemed intrinsically interpretable, due to their structural design. Statistical regression models, such as the generalized additive models, are often applied with a different modeling mindset (Breiman et al., 2001), but we focus on these models from an ML perspective.

The linear regression model restricts the relationship between the target variable Y and the features X to a weighted sum: $\hat{f}(\mathbf{x}^{(i)}) = \beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_p x_p^{(i)}$. This strong model restriction allows an isolated interpretation of the coefficients $\beta_j, j \in \{1, \dots, p\}$ as effects of the individual features. The interpretation is additive, which means we can isolate the individual features and do not have to think about interactions. There are many adaptations that allow the linear regression model to capture more complex relationships, such as the generalized additive model (Hastie and Tibshirani, 2017).

Decision trees represent the relationship with partitions: $y^{(i)} = \sum_{m=1}^M c_m I\{\mathbf{x}^{(i)} \in R_m\}$, where R_m defines a data partition that can be defined by a set of conjunctions based on the feature space and c_m is often the loss-optimal constant model (e.g. the mean for L2-loss) and at the same time the prediction

2.3 Model-agnostic Interpretable Machine Learning

in this partition. Decision trees are special cases of decision rules, since trees are hierarchically organized with disjunct partitions.

A turning point for ML, especially for deep neural networks, was 2012, when a convolutional neural network won the ImageNet challenge (Deng et al., 2009). These complex models, especially deep learning, but also boosted tree ensembles and random forests gained a lot of popularity, due to their superior predictive performance in many prediction tasks. Compared to their intrinsically interpretable counterparts, the more complex, but well regularized/tuned ML models do not allow for straightforward interpretation. Interpretable models such as decision trees and linear regression models have structural restrictions that make them arguably more interpretable. These restrictions pose constraints on the hypothesis space, and when the best solutions are outside of that space, the models lack the flexibility to achieve a high performance. A purely linear regression model fails when interactions are present; decision trees have a hard time reconstructing linear relationships. The approach with, for example, deep neural networks is to start with a rather flexible model class and approach a good solution by careful model tuning and regularization. However, it is recommended to start a project with interpretable models and add complexity as needed (Molnar et. al, 2020). This allows to study the trade-off between model complexity and model performance.

Interpretable models remain building blocks for many other model-agnostic methods. For example, surrogate models such as LIME (Ribeiro et al., 2016b) make use of interpretable models that are fitted locally to a prediction.

2.3. Model-agnostic Interpretable Machine Learning

This chapter defines the term “model-agnostic interpretation method”, introduces the shared framework by which many model-agnostic methods work and illustrates the difference in interpretation of intrinsically interpretable models and of post-hoc model-agnostic approaches.

2.3.1. Definition

Model-agnostic interpretation methods describe the relationship between input features and predictions by systematically probing the prediction function. This probing requires access to the prediction function and the data. Model-agnostic methods therefore treat ML models as black boxes: Model-agnostic methods do not rely on “internal information” of the model, such as the estimated weights in a linear model or the learned structure in decision trees.

We can describe model-agnostic interpretation as a function $I : (F, \mathcal{X}, \mathcal{Y}) \mapsto E$, that maps from F , the space of prediction functions, the feature space \mathcal{X} and the target space \mathcal{Y} to the space of explanations E . The space of explanations depends on the interpretation method. For feature effect methods such as PDP or ALE plots $E = F^1$ is the space of 1-dimensional functions, or respectively $E = F^p$ for p-dimensional variants. For PFI method the explanation space for a single feature is $E = \mathbb{R}$. We call a method post-hoc, when it is applied after the model was trained. This applies to all the model-agnostic interpretation methods in this thesis.

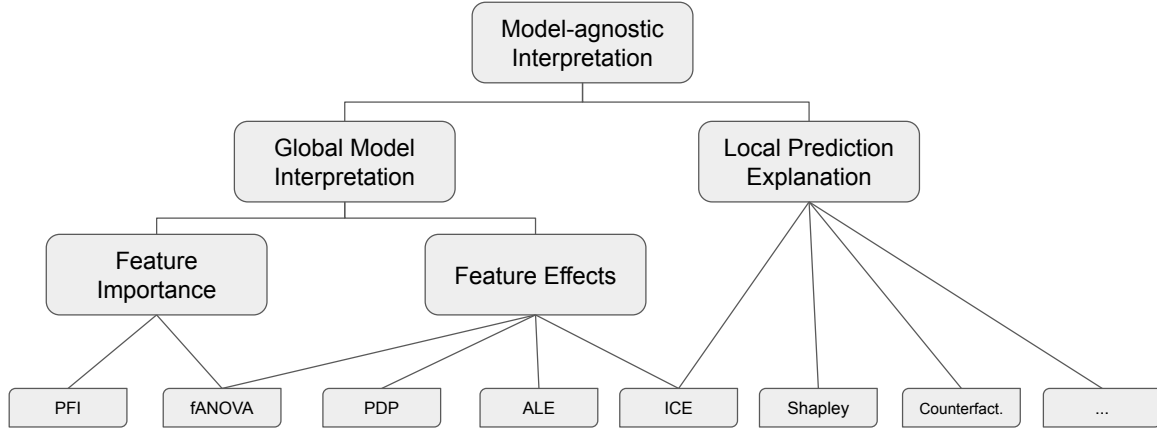


Figure 2.1.: Taxonomy of Methods

2.3.2. Methods Overview

Interpretation methods for ML can be categorized based on their scope, model type to which they can be applied and targeted data type. The targeted data type has the strongest influence on how the produced explanations look like: When images are input to the model, the explanations can usually be visualized as heatmaps that are laid over the original image. For text inputs, the explanations are usually based on highlighting text passages, or emphasis on words. This thesis focuses on tabular data, where a column usually represents one feature or one category (dependent on the encoding of the categorical features).

Figure 2.1 shows a short taxonomy of the model-agnostic interpretation methods for tabular data. The first differentiator is whether an interpretation method quantifies an average model behavior (global), or explains an individual prediction (local). In the first case, we speak of a global model interpretation, as the object of the interpretation is an average behavior. We can further distinguish the global interpretations into feature effects and feature importance measures. Feature effects describe how changes in features change the prediction on average. Feature effects usually constitute a projection of the high-dimensional prediction function \hat{f} to a lower-dimensional function $\hat{f}_S : \mathcal{X}_S \mapsto \mathcal{Y}$, with typically $|S| = 1$ or $|S| = 2$. In that sense, the purpose of a feature effect method is to reduce the dimensionality of the prediction function \hat{f} , which allows to isolate main effects and interaction effects of the prediction function. Examples of feature effect methods are the partial dependence plot (Friedman, 1991), individual conditional expectation curves (Goldstein et al., 2015), accumulated local effect plots (Apley and Zhu, 2020) and the functional ANOVA (Hooker, 2004, 2007).

Feature importance methods assign a relevance value to each feature. Many different proposals exist (Wei et al., 2015) ranging from model-specific approaches for linear models, over difference-based approaches from sensitivity analysis to hypothesis testing-based approaches. A popular, model-agnostic importance approach is PFI, as described in Section 2.4.4. For loss-based methods, importance is defined in terms of loss reduction that can be attributed to features: The higher the increase in loss when “destroying” the feature information (e.g. by permutation), the more important the feature is according to the PFI measure. Feature importance measures can also be derived from the variance of the respective feature effects based on PDP Greenwell et al. (2018) or functional ANOVA (Hooker, 2007, 2004).

2.3 Model-agnostic Interpretable Machine Learning

Methods that work with feature interactions fall either into the feature effect category or feature importance. When 2D-PDPs are used, for example, the goal is to visualize the combined feature effect. When the H-Statistic (Friedman et al., 2008) is used to quantify the strength of interaction, the resulting measure can be seen as a quantification of importance of the interaction.

Local explanation methods explain individual predictions of ML models. The methods differ in how they attribute the prediction to the features: LIME (Ribeiro et al., 2016b) fits a local surrogate model such as a linear model, Shapley values (Štrumbelj and Kononenko, 2014; Lundberg and Lee, 2017; Lundberg et al., 2018) average marginal contributions of features; Counterfactual explanations follow yet another approach: Counterfactuals are copies of the original data instance with minimal changes in the features, but with a relevant difference in the prediction, see also Section 2.4.5. A single ICE curve is also a local explanation that only highlights the influence of a single feature on the prediction of a data instance. For local methods, effect and importance merge, more or less, since the local importance of a feature is often a simple transformation of the feature effect, for example the absolute values of the Shapley values, or the t-statistic for the coefficients in a linear model.

Some local methods can be aggregated to global methods. Shapely values can be aggregated over the data to provide global model interpretations about feature importance, effects and interactions (Lundberg et al., 2018). ICE curves, when plotted for the entire dataset, allows for both a local and a global model interpretation at the same time. Averaging the ICE curves over all the data produces the PDP, a global interpretation of the feature effect.

2.3.3. Analyzing Model Components vs. Behavior

The goal of model-agnostic model interpretation methods differs from the goal of the interpretation of intrinsically interpretable models. A model is usually called intrinsically interpretable when a human can relate an individual component to the real world. A coefficient in a linear model can be interpreted as the linear, isolated effect a feature has on the prediction, when all other features remain unchanged. In a decision tree, the decision path defined by IF-clauses to a terminal leaf can be interpreted as the explanation for the prediction.

But components of complex models, that many would not label intrinsically interpretable, can be analyzed to some degree. Activation maps for convolutional neural networks visualize the images that activate neurons (Nguyen et al., 2016, 2017; Olah et al., 2017). For random forests one can use the minimal depth distribution of the features as a measure of importance (Paluszynska et al., 2020; Ishwaran et al., 2010). These examples of interpreting components of more complex models blur the lines between “intrinsically interpretable” models and model-agnostic post-hoc interpretation of “black box” models.

However, without an interpretable design, the strategy of interpreting the parts (weights, structure, etc.) becomes less viable. Fortunately, model-agnostic interpretation methods are available. In contrast to the *internal view* of analyzing model components, model-agnostic approaches take on an *external view* of the model function by analyzing its behavior. Model-agnostic methods ignore the inner structure of the model, and describe the model by how it behaves when the input features are changed. Model-agnostic methods can even address shortcomings of “intrinsically interpretable” models: For example, inspecting a decision rule list does not report the importance of individual features, but PFI can be reported along with the decision rule lists.

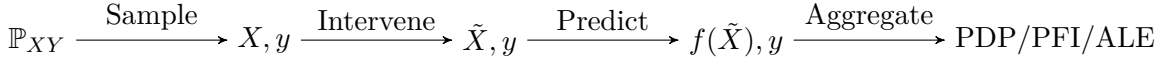


Figure 2.2.: The SIPA framework

In statistics, statistical models are constructed based on assumptions about the relationships and distributions of the variables in the real world. When these assumptions hold, then we cannot only interpret the coefficients, but we can also make inferential claims about their distributions. And if we additionally assume that the model structure is coherent with the true data-generating mechanism, and the model is trained on a representative sample, the inference can be extended to the entire population. However, this approach no longer works for model-agnostic interpretation of complex ML models. To link this external model interpretation to the data-generating process, we would need a ground truth against which we can compare the estimate, and assumptions about the distributions of the output of these interpretation methods. In Molnar et. al (2021) we defined ground truth versions of PDP and PFI, by applying these interpretation methods to the unknown true prediction function f as defined by the data-generating process. This allows us to study biases and sources of uncertainty when we compare, e.g. the PFI applied to the model and applied to the data-generating process.

2.3.4. SIPA Framework

The model-agnostic methods have in common that they treat a model as a black box and mostly work by “probing” the model with input data, and observing the output. This recipe can be summarized in a common framework, the SIPA framework (Scholbeck et. al, 2020). The SIPA framework is comprised of sampling data, applying an intervention on the data, getting model predictions for the new data and aggregating the results. The **sampling** step requires sampling data from the data-generating process. In the **intervention** step, the data instances are intervened upon, for example by permuting a feature column (for PFI). The **prediction** step takes this “design matrix” that comes out of the intervention step and adds the model prediction to the data. In the **aggregation** step, the explanation is produced from data and predictions. Figure 2.2 visualizes the SIPA steps.

The implementation of the R package `iml` (Molnar et al., 2018) was inspired by the SIPA framework. The SIPA framework allows methods to be shared between different methods. This was realized using an object-oriented programming approach, where the `InterpretationMethod` superclass was implemented, from which all other interpretation methods inherit. The `InterpretationMethod` class enforces that the intervention and aggregation steps are provided by the child classes and allows the individual and reusable implementation of intervention and aggregation steps. The sampling step is delegated to the user who has to choose the data based on which the interpretation should be computed. To handle the data, a `Data` class was implemented. The `Predictor` class is another basic but central class to `iml`: it holds the prediction model and serves as abstraction layer to the prediction function. The `Predictor` class works with many models by providing implementation for the popular ML libraries `mlr`, `mlr3` and `caret`.

2.4 Model-agnostic Interpretation Methods

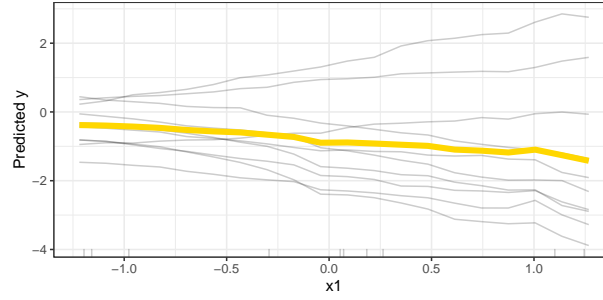


Figure 2.3.: ICE curves and PDP

2.4. Model-agnostic Interpretation Methods

This section explains the model-agnostic IML methods that were the most relevant to this thesis: the partial dependence plot, accumulated local effect plots, interaction effects, permutation feature importance and counterfactual explanations.

2.4.1. Partial Dependence Plots

The PDP (Friedman, 1991) describes the average change in the predicted outcome of an ML model when one or more of the features are changed while the remaining features are not. The partial dependence function for a feature set X_S is a marginalized version of the prediction function \hat{f} , where the features X_C (with $S \cup C = \{1, \dots, p\}$ and $S \cap C = \emptyset$) are integrated over, and therefore “removed”.

$$PD_S(x) = \int_{X_C} \hat{f}(x, X_C) d\mathbb{P}_{X_C}(X_C) \quad (2.1)$$

Since the distribution \mathbb{P}_X is usually unknown, the estimation relies on Monte Carlo integration:

$$\widehat{PD}_S(x) = \frac{1}{n} \sum_{i=1}^n \hat{f}(x, \mathbf{x}_c^{(i)}) \quad (2.2)$$

To visualize the partial dependence function, one has to define grid points along X_S , at which the partial dependence function $PD_S(x)$ is evaluated and plotted.

For categorical data, the principle is the same. But instead of defining a grid, each category is a “grid point”.

The PDP has two big shortcomings. One is feature interactions, and the other is feature dependence. When the prediction is not just the sum of individual feature effects, then the model encodes feature interactions. When two features interact, then the effect of one feature on the prediction function depends on the value of the other feature. As the PDP marginalizes over features X_C , any interactions between features X_S and X_C are averaged over and are invisible in the plot. This “hiding” of feature interactions is by design, but can result in incorrect interpretation. For example, a flat PD might invite to (wrongly) conclude that a feature has no effect on the prediction, even when the feature might be

quite influential for the prediction, but mostly through interactions with others. A solution to reveal potential interactions is to enhance the PDP with individual conditional expectation (ICE) curves. An ICE curve (Goldstein et al., 2015) for an instance $\{(x_S^{(i)}, x_C^{(i)})\}^{(i)}$ is defined as the curve $\hat{f}_S^{(i)}$ plotted along with $x_S^{(i)}$. The features values $x_C^{(i)}$ are kept fixed. Plotting the ICE curves can reveal interactions that are not visible in the PDP. Figure 2.3 shows examples for both PDP and ICE curves.

Feature dependence means that features X_S are not statistically independent from features in X_C , with linear correlation being a special case of dependence. Dependence can be a severe problem for the PDP and can result in misleading interpretations (Hooker and Mentch, 2019; Molnar et al., 2020). Feature dependence might even have the consequence that for the computation of the PDP impossible data points are created and used. ALE plots (Apley and Zhu, 2020), functional ANOVA (Hooker, 2004, 2007) and subgroup PDPs (Molnar et al., 2020) have been proposed as possible solutions. However, the problem remains fundamental, as the effects of dependent features might just not be fully separable.

2.4.2. Accumulated Local Effect Plots

Accumulated local effect (ALE) plots (Apley and Zhu, 2020) were developed as an alternative to the PDP, especially for the case of dependent features. ALE offers a decomposition of the prediction function into an intercept, main effects and interaction effects of increasing order.

$$\hat{f}(\mathbf{x}^{(i)}) = f_0 + \sum_{j=1}^p f_{ALE,j}(x_j^{(i)}) + \sum_{j \neq k}^p f_{ALE,jk}(x_j^{(i)}, x_k^{(i)}) + \dots + f_{ALE,1,\dots,p}(x_1^{(i)}, \dots, x_p^{(i)}),$$

where each $f_{ALE,S}$ is an ALE component with an according ALE plot visualization (at least for $|S| \in \{1, 2\}$). This decomposition is unique under an orthogonality-like property further described in Apley and Zhu (2020). The ALE first order effects $f_{ALE,j}$ of a single feature $x_j, j \in \{1, \dots, p\}$ for model \hat{f} is defined as

$$f_{ALE,j}(x_j) = \int_{z_{0,j}}^{x_j} \mathbb{E} \left[\frac{\partial \hat{f}(x_1, \dots, x_p)}{\partial z_j} \middle| X_j = z_j \right] dz_j - c_j \quad (2.3)$$

$$= \int_{z_{0,j}}^{x_j} \int_{X_C} \frac{\partial \hat{f}(x_1, \dots, x_p)}{\partial z_j} \mathbb{P}(X_C | z_j) dX_C dz_j - c_j \quad (2.4)$$

Here $z_{0,j}$ is the lower bound for X_j , which it makes sense to choose the minimum of x_j .

The expectation \mathbb{E} is computed with respect to the marginal distribution of the other features X_C ($C \cup j = \{1, \dots, p\}$) and conditional on the feature value of x_j . The constant c_j is defined so that the average of $f_{ALE,j}(x_j)$ is zero with respect to the marginal distribution of X_j . This ensures that the ALE components sum up to the full prediction function and deliver a full decomposition. The idea behind ALE is to isolate the effect of feature X_j from all other features X_C . This is done by integrating the expectation of the derivative of f with respect to X_j .

2.4 Model-agnostic Interpretation Methods

Estimation

For the estimation, the integral is exchanged for finite difference. In practice, this means that access to the gradient of the prediction function is not required. First, we have to estimate the uncentered ALE for feature X_j (with X_C representing the remaining features):

$$\hat{f}_{ALE,j}(x) = \sum_{k=1}^{k_j(x)} \frac{1}{n_j(k)} \sum_{i: x_j^{(i)} \in N_j(k)} [\hat{f}(z_{k,j}, \mathbf{x}_C^{(i)}) - \hat{f}(z_{k-1,j}, \mathbf{x}_C^{(i)})] \quad (2.5)$$

Equation 2.5 can also shed some light on why the method is called “Accumulated Local Effects”. In the inner sum, differences in predictions are computed, where the feature of interest is replaced with grid values $z_k, k \in \{1, \dots, k_j\}$, where k_j is the number of grid values that was set for feature X_j , and $k_j(x)$ indicates the interval number in which x falls into. This difference can be interpreted as the **local** effect that a feature has for a specific data instances in this interval. These differences are averaged across the instances that have their $x_j^{(i)}$ value in this interval, which is captured in $N_j(k)$. This average of differences represents the **effect** the feature has, locally. The local, average effects per interval are **accumulated** along the domain of X_j .

To finalize the computation of the ALE curve, the effect computed in Equation 2.5 has to be centered:

$$\hat{f}_{ALE,j}(x) = \hat{f}_{ALE,j}(x) - \frac{1}{n} \sum_{i=1}^n \hat{f}_{ALE,j}(x_j^{(i)}) \quad (2.6)$$

Figure 2.4 (an adaptation from Apley and Zhu (2020) and Molnar (2019)) visualizes the accumulation of intervals. It is possible to compute ALE curves also for higher-order effects, for example 2-way interactions, but I refer to Apley and Zhu (2020) for details. First and second-order ALE curves are implemented in the *iml* R package (Molnar et al., 2018), and in the ALEPlot package (Apley, 2018).

To make ALE work with categorical features, a “trick” has to be applied. Since ALE, by definition, requires an ordering, it only works for categorical features once they are ordered. For a given order, the categorical ALE can be interpreted as the difference in prediction when changing from one category to another.

ALE plots have also been criticized, because in some cases, ALE does not yield the same mathematical structures that were defined in the data-generating process (Groemping, 2020), which can be counter-intuitive for the interpretation of ALE.

ALE as Functional Decomposition

ALE can also be understood as a decomposition of the prediction function \hat{f} . The idea behind functional decomposition is that we can split a high-dimensional function $\hat{f} : X \mapsto Y$ into a sum of components with increasing dimensionality:

$$\hat{f}(\mathbf{x}^{(i)}) = f_0 + f_1(x_1^{(i)}) + \dots + f_p(x_p^{(i)}) + f_{12}(x_{12}^{(i)}) + \dots + f_{1,\dots,p}(x_{1,\dots,p}^{(i)})$$

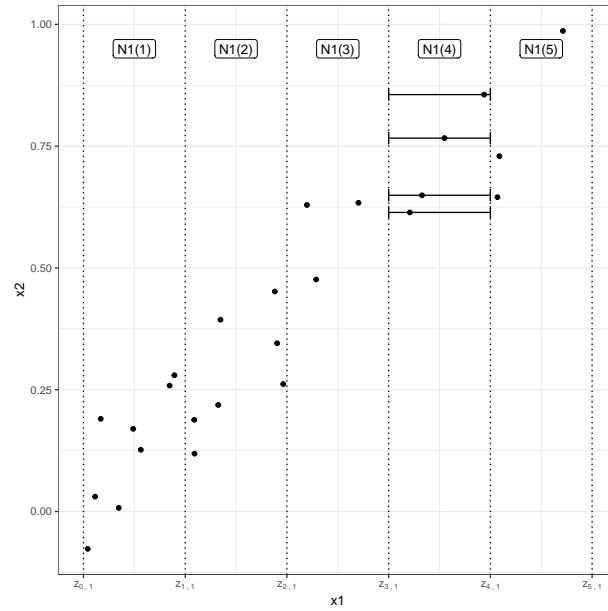


Figure 2.4.: ALE construction

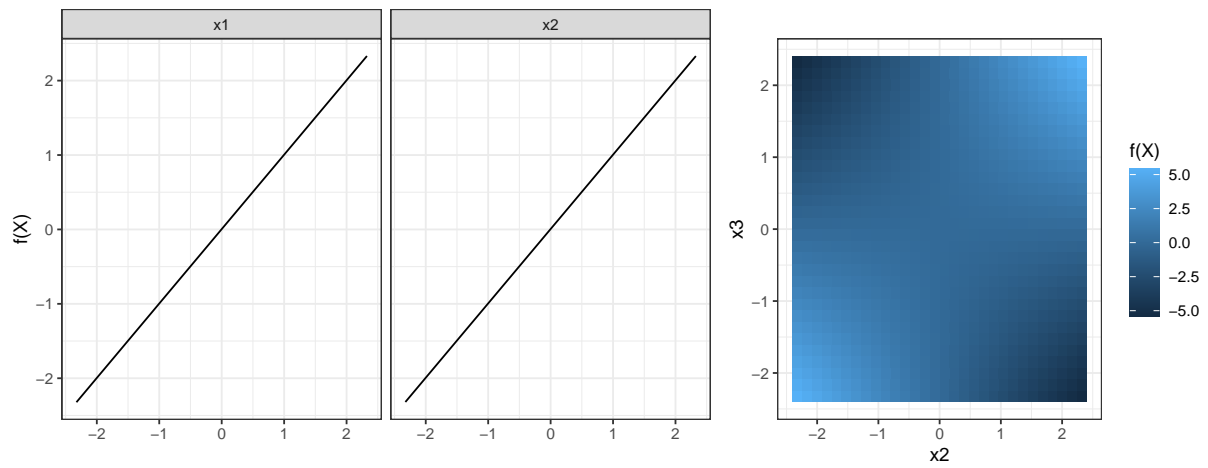


Figure 2.5.: Decomposition of function $f(X) = 2 + x_1 + x_2 + x_2 \cdot x_3$ using functional decomposition. Since all feature are independent ($X_1, X_2, X_3 \sim N(0, 1)$), the decompositions of PDP, functional ANOVA and ALE are the same, up to a constant.

2.4 Model-agnostic Interpretation Methods

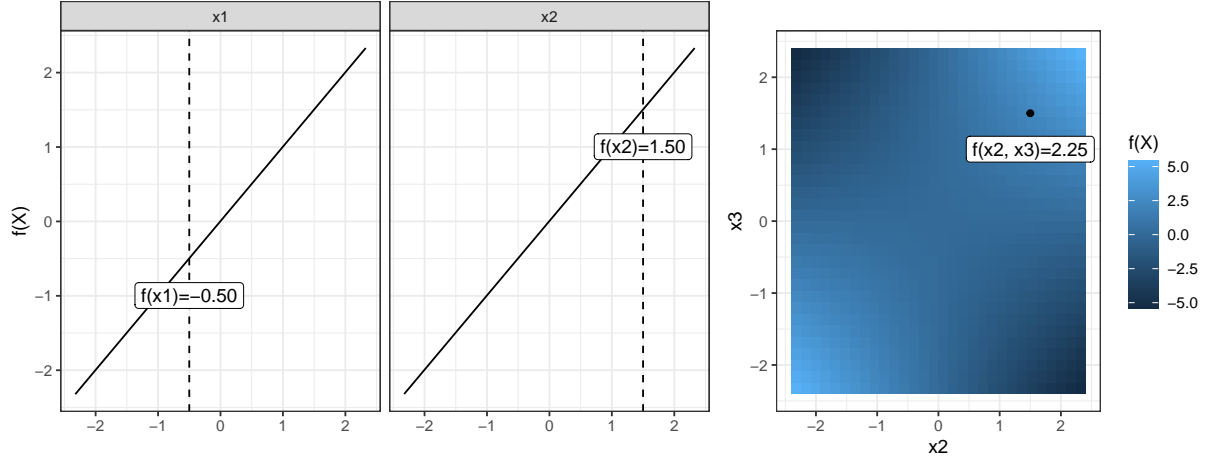


Figure 2.6.: Decomposition of function $f(X) = 2 + x_1 + x_2 + x_1 \cdot x_2$ using functional decomposition. Since all feature are independent $x_1, x_2, x_3 \sim N(0, 1)$, the decompositions of (centered) PDP, functional ANOVA and ALE are combined.

The term f_0 can be interpreted as an intercept and the f_j , $j \in \{1, \dots, p\}$ as first order or main effects. Beyond that, the terms capture interaction effects between features, for example, the term f_{12} captures an interaction between features X_1 and X_2 . Each of these lower effects map from a set of features to the target space and can be interpreted as the feature (interaction) effect.

With only the assumption that the components add up to the full function \hat{f} without further assumptions about the components and the relations between them, the components do not have a unique solution. Various proposals have been made how to uniquely define the functional decomposition. One such decomposition is provided by ALE. The components as computed by ALE fulfill a pseudo-orthogonal property: The pseudo-orthogonality says that the ALE operator, which maps a function to the S -th ALE component, applied sequentially to \hat{f} yields again the same ALE component. But more importantly it says that when first applying the ALE operator $H_S : \mathbb{R}^p \mapsto \mathbb{R}^{|S|}$ for component S on \hat{f} (yielding $f_{ALE,S}$) and then applying a different ALE operator $H_J : \mathbb{R}^p \mapsto \mathbb{R}^{|J|}$ for J with $J \neq S$ will result in a flat ALE curve that equals 0.

An example of a full functional decomposition with ALE plots is visualized in Figure 2.5. The main effects can be visualized with a curve, and second-order interactions using tile plots (heatmaps). For any input x , the prediction can be decomposed into the individual components. For example, the prediction for data instance $x_1 = -0.5, x_2 = 1.5$ decomposes into: $f(x) = 2.00 - 0.50 + 1.50 + 0 + 0 + 2.25 + 0$. This is also visualized in Figure 2.6.

We used the ALE composition in the paper Molnar et. al (2020), in which we defined measures of model complexity based on functional decomposition. ALE is computationally fast and allows to compute components independently of each other. Another functional decomposition is given with the functional ANOVA proposed by Hooker (2004), which was later generalized for dependent features (Hooker, 2007).

2.4.3. Feature Interactions

Features interact when “the prediction cannot be expressed as the sum of the feature effects, because the effect of one feature depends on the value of the other feature” (Molnar, 2019). In functional decomposition, every component that depends on more than one feature describes an interaction effect. The order of an interaction is the number of features that are involved in the term. This means that the following terms do not represent interactions: f_0, f_1, \dots, f_p . Second-order interactions are terms that include two features: $f_{1,2}, \dots, f_{1,p}, \dots, f_{p,1}, \dots, f_{p,p}$. For higher order interactions, equivalently more terms are used. We can distinguish between methods that extract the components of interaction terms, i.e., that try to retrieve something related to the decomposition terms and methods that quantify the strength of an interaction. Two-dimensional PDP and ALE plots visualize second-order interaction effects directly. Measures like the H-Statistic quantify the strength of interaction in terms of variance based on the PDP (Friedman et al., 2008). We used a variance based method to compute the share of ALE functional decomposition that is based on interactions to describe the complexity of a model in Molnar et. al (2020). Functional ANOVA expresses interaction strength in terms of the variance of the respective component Hooker (2004, 2007).

Another measure of strength of pairwise interactions uses repeated dichotomization of variable and constructs the interaction predictor by comparing the means of the resulting quadrants (Lou et al., 2013; Caruana et al., 2015). Identified interactions are then ranked and incrementally to a generalized additive model as 2D-tensors.

2.4.4. Permutation Feature Importance

Permutation feature importance (PFI) is one of many approaches to quantify the global importance of features in an ML model. PFI was first introduced for random forests (Breiman, 2001). Since then, PFI has been studied in detail, and many adaptations have been proposed (Ishwaran and Lu, 2019; Archer and Kimes, 2008; Janitzka et al., 2018; Strobl et al., 2008; Boulesteix et al., 2012; Strobl et al., 2007). Finally, a model-agnostic version was proposed by Fisher et al. (2019).

Model-agnostic PFI is defined as the increase in loss when a feature is permuted. Mathematically, PFI is defined as:

$$PFI_S = \mathbb{E} \left[L(\hat{f}(\tilde{X}_S, X_C), Y) - L(\hat{f}(X), Y) \right] \quad (2.7)$$

Here, L is a loss function, and \tilde{X}_S are the data where features X_S were perturbed. If this perturbation is a simple permutation, the result is the marginal PFI. This would be the case for $\tilde{X}_S \sim X_S$ and $\tilde{X}_S \perp X_C$. However, the marginal PFI suffers when features are dependent (e.g. correlated), as further described in Section 2.5. When \tilde{X}_S is sampled conditional on the features X_C , we call the approach the conditional feature importance (Candès et al., 2018; Molnar et al., 2020). Conditional sampling means that we sample $\tilde{X}_S \sim X_S | X_C$. Conditional PFI can also be used when features are dependent, but offer a different interpretation.

The PFI is defined as integral over an (unknown) distribution. As such, we can estimate it with Monte Carlo integration:

2.4 Model-agnostic Interpretation Methods

$$\widehat{PFI}_S = \frac{1}{n} \sum_{i=1}^n \left(\left(\frac{1}{m} \sum_{k=1}^m L(y^{(i)}, \hat{f}(\tilde{x}_S^{(i,k)}, \mathbf{x}_C^{(i)})) \right) - L(y^{(i)}, \hat{f}(\mathbf{x}^{(i)})) \right) \quad (2.8)$$

Here m is the number of times the permutation or sampling is repeated for more stable results. The estimation approach is illustrated in Table 2.1.

Table 2.1.: Illustration of permutation for PFI. Table (a) shows the features x_1, \dots, x_p , target y and the loss L . In (b), the values for x_j are permuted, which changes the loss L .

(a)							(b)						
x_1	...	x_j	...	x_p	y	L	x_1	...	\tilde{x}_j	...	x_p	y	L
0.7	...	1.3	...	12.1	1.1	0.012	0.7	...	1.8	...	12.1	1.1	0.166
1.7	...	1.8	...	7.1	4.9	0.119	1.7	...	8.2	...	7.1	4.9	1.222
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
1.3	...	1.9	...	2.1	3.2	0.044	1.3	...	1.3	...	2.1	3.2	0.208
2.7	...	8.2	...	17.0	2.2	0.92	2.7	...	1.9	...	17.0	2.2	0.478

At first glance PFI and PDP are rather different methods. However, as we showed in Scholbeck et. al (2020), they both work by the framework of sampling, intervention, prediction and aggregation. We could show that there are even more parallels between both methods, which was leveraged to propose novel visualizations of PFI (Casalicchio et. al, 2019).

2.4.5. Counterfactual Explanations

Counterfactual¹ explanations can be used to explain individual predictions of ML models. The “factual” in “counterfactual” stands for the prediction that was observed for a specific instance. Counterfactual refers to a prediction that we did not observe. In that sense, counterfactual explanations reverse the approach of other methods: Counterfactual explanations explain which values the features would have to take on to yield a different prediction. A counterfactual explanation is represented by a new data point, for which a few features are changed (compared to the original data point).

Counterfactual explanations are therefore contrastive, and selective (focus on a few feature changes), which makes them an ideal candidate for explanations for humans (Miller, 2019). Both model-agnostic and model-specific versions of counterfactual explanations exist (Wachter et al., 2017; Joshi et al., 2019; Looveren and Klaise, 2019; Poyiadzi et al., 2019; Sharma et al., 2019; Grath et al., 2018; Dhurandhar et al., 2019; White and d’Avila Garcez, 2019; Karimi et al., 2019).

Most approaches propose an optimization function that combines various objectives: The prediction for the counterfactual should be as close as possible to the desired prediction; As few features as possible should be changed; The feature value changes should be kept at a minimum; The resulting data point should be as realistic as possible, according to the joint distribution of the data. In Dandl et. al (2020) we were the first to formulate this search for counterfactuals as a multi-objective optimization problem.

¹Not to be confused with counterfactuals in the causal inference literature

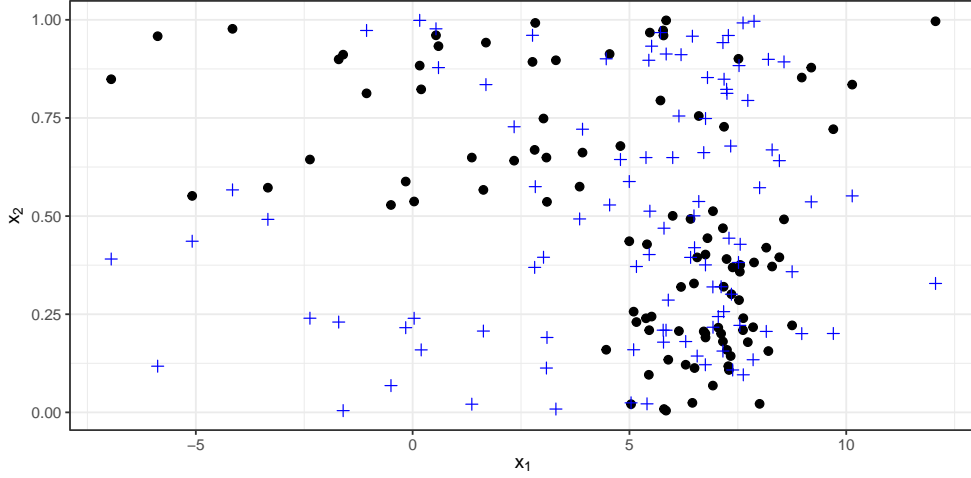


Figure 2.7.: Features X_1 and X_2 (black dots). Permutation of feature X_1 yields new data (blue crosses), which extrapolate to areas outside the original data distribution

2.5. Dependent Features and Extrapolation

Model-agnostic, permutation-based methods take the data $X \sim \mathbb{P}_X$ and apply an intervention (Scholbeck et. al, 2020). For PDP, this intervention means replacing a feature with a fixed value. For (marginal) PFI, the feature of interest is sampled / permuted. We call this interventional data \tilde{X} . An interpretation method can be true to the data, if the intervention preserves the joint distribution (Definition 1).

Definition 1. An intervention $I : \mathcal{X} \mapsto \mathcal{X}$ is true to the data when the generated data follows the same distribution $\tilde{X} = I(X) \sim \mathbb{P}_X$.

The marginal version of PDP, PFI and so on are true-to-the-data when features are independent, as X_S are drawn from the marginal distribution. However, when the features are dependent, marginal versions of PDP and PFI are no longer true to the data. This dependence between two individual features can be a simple linear correlation, but also other, more complex dependencies that are non-linear and involve more than two features are possible. As a consequence, practitioners have to use non-linear dependence measures such as HSIC to detect possible dependencies (Molnar et. al, 2020). The intervention of PDP and PFI produce a distribution that does not match the original joint distribution any longer.

By computing marginal PDP or PFI on dependent data, the feature values are extrapolated to regions outside of the data distribution (see Figure 2.7). Extrapolation leads to an emphasis on predictions for feature values with low probability or at worst can produce impossible data points. The extrapolation problem affects all interpretation method that rely on permutation or sampling from the marginal distribution of features. Extrapolation is problematic for two reasons:

1. **Uncertain model predictions.** The extrapolated data points lie outside of the training data. This means that the model was never trained on data in this region of the feature space. As a result, an extreme prediction of the model in this region of the feature space would be possible as the model was never “controlled” in this area (via the loss function).

2.5 Dependent Features and Extrapolation

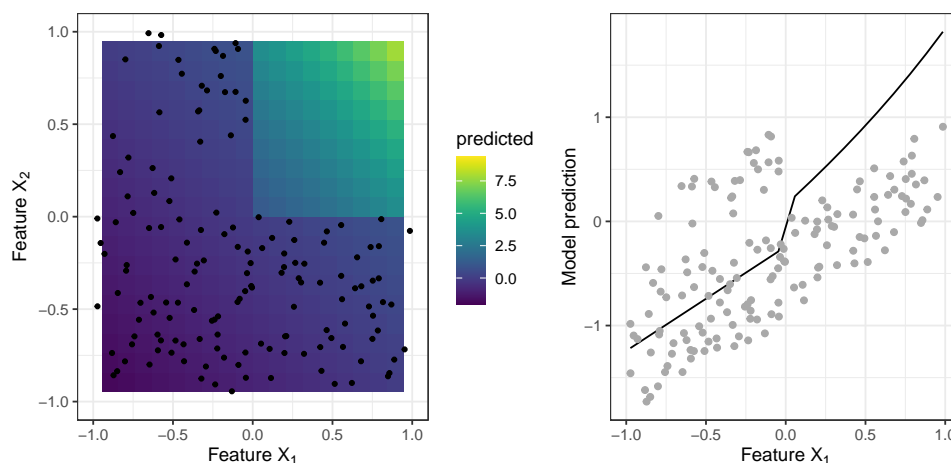


Figure 2.8.: Misleading PDP due to extrapolation. Simulated are features X_1 and X_2 which follow a joint uniform distribution from -1 to 1, except that the density for points with $X_1 < 0, X_2 < 0$ is zero, and therefore the features are dependent. The prediction function is a linear model with an additional interaction term $Y = X_1 + X_2 + I(X_1 < 0, X_2 < 0) \cdot \exp(X_1 \cdot X_2)$. The left plot shows the data sample and the prediction function surface, the right plot shows the PDP for feature X_1 .

2. **Undefined data-generating process.** The newly generated data points might represent impossible entities, like a 20 year old person with 30 years of professional experience. When these impossible data points are used to compute PDP and PFI, their interpretations become problematic.

Therefore, feature dependence poses one of the biggest conceptual problems for permutation-based interpretation methods such as PDP and PFI. Figure 2.8 shows a case of a misleading PDP due to extrapolation. The features X_1 and X_2 are dependent, and the PDP estimate is strongly influenced by model predictions in areas where the data density is zero. We could call the PDP misleading, since it suggests that for values of X_1 above 0.5, the average prediction is above 1, although for the entire training set no prediction larger than 1 was ever observed. The PDP, however, is not necessarily “wrong”, since it gives interesting insights into the prediction function when we knowingly intervene on X_1 and potentially extrapolate. The PDP shows that the prediction function changes from linear to exponential at $X_1 = 0$, which we could only notice because of the extrapolation. There is a tension between staying true to the data (by not extrapolating), and staying true to the model (by showing what the function looks like).

The exact effect of extrapolation on interpretation cannot be stated a-priori. The effect depends on the chosen model class (e.g. trees, linear model, ...), the dependence structure, the strength of dependence and so on. The model must represent an interaction in the extrapolated area between the features for the interpretation to be affected by extrapolation.

Conditional Versions of PDP and PFI

A solution to the extrapolation problem is to adjust the intervention so that it becomes true-to-the-data. These versions of PFI and PDP, the conditional PFI (Molnar et al., 2020; Fisher et al., 2019; Watson and Wright, 2019; Hooker and Mentch, 2019) and the conditional PDP, also called M-Plot (Apley and Zhu, 2020), replace feature values using the conditional distribution instead of the marginal distribution. Their sampling mechanisms changes from $\tilde{X}_S \sim X_S$, to a conditional sampling mechanism so that $\tilde{X}_S \sim X_S|X_C$ (or $\tilde{X}_C \sim X_C|X_S$ for the case of M-Plot). This conditional sampling mechanism respects the conditional distribution and produces data points where the joint distribution is respected and makes sure that the intervention is true to the data.

The conditional distribution is typically unknown and has to be estimated. For conditional PFI, various approaches were suggested, such as using knockoffs (Watson and Wright, 2019; Candès et al., 2018) and imputation (Fisher et al., 2019). We suggested an approach based on permutation in subgroups (Molnar et. al, 2020).

While conditional interpretation methods avoid the extrapolation problem, they change the interpretation itself, which can results in unintuitive behavior (if a marginal interpretation was expected). There is an inherent tension between staying true to the data and staying true to the model (Chen et al., 2020). The tension only arises when features are dependent, as independence between two subsets of features X_S and X_C means that the marginal and conditional distribution of $X_S|X_C \sim X_S$ and $X_C|X_S \sim X_C$ coincide. The marginal PDP and PFI allow an isolated interpretation of feature effect or importance which can fully be attributed to that particular feature unaffected by other features (except through interactions). By switching to the conditional PDP / PFI, the interpretation also becomes conditional, meaning that the interpretation becomes entangled between features.

The conditional PDP can be defined in analogue with the marginal PDP (Equation 2.1), but with a conditional expectation:

$$cPDP_S(x) = \int_{X_C|X_S} \hat{f}(x, X_C) d\mathbb{P}_X(X_C|X_S = x). \quad (2.9)$$

The conditional PDP for a feature mixes the (marginal) effects of the feature of interest with all other features that are dependent on it. This is because at a certain grid value of feature X_S , instances with values of X_C that are more likely get a larger weight.

The conditional PFI is defined similarly to the marginal PFI (Equation 2.7):

$$cPFI_S = \mathbb{E}_{Y, X_C, X_S \tilde{X}_S|X_C} \left[L(\hat{f}(\tilde{X}_S, X_C), Y) - L(\hat{f}(X), Y) \right], \quad (2.10)$$

While the PDP combines effects of dependent features, the conditional PFI of a feature shrinks if other features are correlated with the feature of interest. The conditional PFI can be interpreted as the drop in performance for removing X_S , but given that we know the values of X_C . In the extreme case that two features are copies of each other, and both are used by the model, the conditional PFI can approach zero, since the other feature encodes the same information. It would be a pitfall (Molnar et. al, 2020) to conclude that both features were irrelevant to the model. It is not set in stone that the conditioning must consider all features, or only features in the model. We explored the idea of *relative feature importance* (König et. al, 2021) as a framework for analyzing the importance of a feature

2.5 Dependent Features and Extrapolation

relative to an arbitrary set of other features. Choosing a meaningful set of features to condition on can be used, for example, to study indirect influences of features. Indirect influence plays a role in fairness considerations, and allows to study whether, for example, gender influences the model prediction via other features.

3. Concluding Remarks and Future Work

Since I started my dissertation a few years ago, the hype around interpretable machine learning has grown a lot. The increased need for interpretability is not surprising. With the increased use of ML, the demand for interpretability grows as well. This makes IML a fast growing field: A lot of new interpretation methods have been introduced. In my opinion, the field of interpretability has reached a first “plateau” of maturity – as proven by a large body of research, established interpretation methods, software implementations, even startups that sell “interpretability-as-a-service”.

Instead of developing new methods, I deliberately in my dissertation focused on consolidation and deepening our understanding of established methods. Also, my conclusion and ideas of future work on IML revolve around the notion that the IML field needs more rigor, more consolidation and addressing more fundamental questions. In the following, I discuss a few themes that I think future work should focus on.

3.1. Dependent Features – A More Fundamental Problem?

Correlated or dependent features are a big issue when interpreting ML models. The dependence problem has been a constant throughout all articles contributing to this thesis. Other researchers have noticed the problem as well, and suggested various approaches to juggle the delicate trade-off between avoiding extrapolation, but also disentangling the interpretation of features. But still, none of these methods seem to completely fix the dependence problem once and for all. Can the dependence problem not be fixed? I believe the way we are approaching the dependence problem can only address the problem superficially, be it using conditional variants, grouping features together or working with disentangled representations of features. The dependence problem is inherent to the current ML paradigm. This paradigm treats feature dependence as a technical problem, a nuisance to be dealt with. But maybe the right question to ask would be: Why are these features dependent? What does their dependence mean for the interpretation? Is there even a real world equivalent for disentangling the features?

In my opinion, feature dependence is not a solely technical problem to solve, but the modeling approach must be adapted to treat such dependencies as part of the data-generating process. The merely associative nature of most predictive models prohibit a more causal interpretation of importance and effects of features. Causal inference is a modeling field that treats dependent features not as a mere nuisance, but requires explicit causal assumptions of the data structure. My believe is that to “fix” the dependence problem also means that we have to “fix” our approach to modeling. Causal inference is a research direction that could be helpful in this regard.

3.2. Do We Really Have to Define Interpretability?

One of the biggest critique of IML is that “interpretability” lacks a definition and therefore the field would lack rigor. A lot of ink has been spent on discussing the definition of model interpretability. But what if interpretability can never be defined with a simple definition or metric? Could we again be asking the wrong question? I believe there are three ways forward in face of this critique: (1) acknowledging multi-dimensionality of interpretability, (2) case studies and simulations, and (3) axiomatization.

Multi-dimensionality of Interpretability

A possible way forward could be to let go of the desire to have a single definition of interpretability and embrace that there are various dimensions of interpretability. Examples of such dimensions are sparsity, monotonicity, linearity, time reduction in human task completion, faithfulness of an explanation, and so on. In Molnar et. al (2020), we tried to define three such dimensions in a model-agnostic way: sparsity, main effect complexity and interaction strength, in our case to quantify the complexity / interpretability of ML models themselves. An embrace of multi-dimensionality of interpretability would shift the burden away from showing that an IML method fulfills a fuzzy notion of interpretability, and towards concrete aspects of interpretability. This would also increase the rigor of the research, since it forces the researchers to be explicit about how their models or explanation methods are interpretable: It would not be enough to just claim interpretability, but required to define and compare methods or models across various metrics. This is already happening in the field, but could be done with more emphasis.

Case Studies and Simulations

When describing the distribution of a random variable, we have various choices: We can describe the center of the mass with the average value, with the median, with a weighted average, the mode, ... We can describe the width of the distribution by calculating the range between minimum and maximum value, or the variance, or the interquartile distance, ... However, there is no scientifically correct way, the choice is subject to preferences and, very importantly, limitations of each descriptor. For example, the mean of a distribution can be heavily influenced by outliers, so that for a very skewed distribution, such as household income, the median might be a more useful descriptor.

With IML, I would argue that we are in a similar situation. Both descriptive statistics and IML describe either distribution or model and both provide various descriptors to choose from. Instead of requiring “proof” of whether an IML method matches some fuzzy definition of interpretability, we could collect use cases and simulations to study limitations of IML methods and catalogue how IML methods behaves for specific tasks, models and data-generating processes.

These type of method-focused studies have already had an impact on IML research and lead to a better understanding of limitations and improvement of methods. Thanks to Goldstein et al. (2015), we know that there are situations where the PDP is basically flat, but still the feature influences the prediction function. The feature effect is fully mediated through interactions with other features, so that the main effect is zero. Groemping (2020) showed that ALE plots can deliver non-intuitive results for dependent and interacting features in a simple simulated setting. Hooker (2007) studied a simple simulation with two dependent features and a simple model to show limitations of the PDP.

3.3 Can IML Reveal Insights About the Real World?

Scenario by scenario, these studies generate a body of practical knowledge for IML methods. This additional rigor can ultimately lead to concrete recommendations on whether to use a certain IML method in a certain scenario and how to interpret the results.

Axiomatization

Shapley values for explaining predictions are a quite popular IML method. While many factors have contributed to this method's success, the axiomatic framework on which it is based is one of the key ingredients. Shapley values are based on game theory, a system of four axioms – efficiency, dummy, symmetry and additivity – for which Shapley values provide a unique solution. These axioms are not only useful for mathematical reasons, but they also define how we can interpret the Shapley values. For example, the dummy axiom implies that a feature that does not change the prediction, no matter how much the feature is changed, receives an attribution of zero. Other methods also fulfill the dummy axiom, for example the PDP. But in contrast, the PDP did not emerge from an axiomatic framework. Thus, axiomatization can help us in two ways: For established methods, we can, post-hoc, check whether they fulfill certain axioms. Or, as in the case with Shapley values, we can start with a set of axioms and build a method from scratch that fulfills these axioms.

In this sense, axiomatization is a complementary approach to use cases and simulations: Instead of describing the properties of an explanation method, we can first think about desirable axioms and then develop and/or test interpretation methods accordingly. While some research papers are already guided by axiomatization, there is room for more axiom-based research.

3.3. Can IML Reveal Insights About the Real World?

Classical statistical modeling is state-of-the-art for inferring properties of the real world from data in many scientific disciplines. Statistics, especially statistical modeling and hypothesis testing, is accepted in most quantitative fields, like medicine or ecology, as a method to generate knowledge. Statistical modeling puts a lot of emphasis on thinking about the data-generating process and adapting the models accordingly. In some fields, for example ecology, classical statistical modeling is slowly being replaced, or at least challenged, by ML models with an additional application of IML methods (Roscher et al., 2020). A goal of research is, ultimately, to generate knowledge, and therefore researchers more and more often interpret output of IML methods as real world effects.

However, most IML methods are first and foremost designed to describe the model, not the real world. Currently, our ML models and IML methods are not ready for this use case. We need more research to bridge the gap between IML being model descriptors and IML being used for real world inference. For PFI and the PDP, we made some first steps with our contribution Molnar et. al (2021) by proposing a ground truth equivalent of PFI and PDP in the data-generating process and providing estimators that respect model variance and confidence intervals.

3.4. Which Uncertainty?

The question of real world inference is coupled with the question of uncertainty. All of the IML methods in this thesis assumed a fixed model that is trained once and the IML method only works by

manipulating the data. Therefore the only uncertainty that we can measure is the approximation error of the IML method, which for many IML methods is just the variance of the Monte Carlo integration. But this ignores the uncertainty in the training process itself. The uncertainty stems from the training data being a random sample, but also from stochastic steps in the training, such as stochastic gradient descent for neural networks, or bootstrapping for random forests. Especially when our goal is to draw conclusions about the real world, the uncertainty of the entire procedure, including training, have to be taken into account. As most IML methods are designed for a fixed model, there is room for future research to adapt IML methods to take model uncertainty into account.

IML has reached a first plateau of usability, but IML methods are already being used outside their intended use, as addressed in the paragraphs before. I am confident that the IML research community can address these shortcomings, bringing more rigor to IML and making IML a powerful and useful tool for practitioners and researchers.

Part II.

Contributions

Contributing Publications

Contributions sorted by topic: (1) consolidation (2) PDP and PFI (3) other IML methods

- Molnar, C., Casalicchio, G., and Bischl, B. (2020). Interpretable Machine Learning—A Brief History, State-of-the-Art and Challenges. In *ECML PKDD 2020 Workshops* (p. 417 – 431). Springer International Publishing (Cham). https://doi.org/10.1007/978-3-030-65965-3_28
- Molnar, C., König, G., Herbringer, J., Freiesleben, T., Dandl, S., Scholbeck, C., Casalicchio, G., Grosse-Wentrup, M., and Bischl, B. (2020). General Pitfalls of Model-Agnostic Interpretable Machine Learning. *To appear in: xxAI – Beyond explainable Artificial Intelligence. Lecture Notes in Artificial Intelligence, vol. 13200. Springer, Cham..*
- Scholbeck, C., Molnar, C., Heumann, C., Bischl, B., and Casalicchio, G. (2019). Sampling, Intervention, Prediction, Aggregation: A Generalized Framework for Model-Agnostic Interpretations. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 205-216).
- Molnar, C., Casalicchio, G., and Bischl, B. (2018). iml: an R package for Interpretable Machine Learning. *Journal of Open Source Software*, 3, 786.
- Molnar, C., König, G., Bischl, B., and Casalicchio, G. (2020). Model-agnostic Feature Importance and Effects with Dependent Features - A Conditional Subgroup Approach. *arXiv preprint arXiv:2006.04628*.
- Molnar, C., König, G., Freiesleben, T., Wright, M., Casalicchio, G., and Bischl, B. (2021). Relating the Partial Dependence Plot and Permutation Feature Importance to the Data Generating Process. *arxiv preprint arXiv:2109.01433*.
- König, G., Molnar, C., Bischl, B., and Grosse-Wentrup, M. (2021). Relative Feature Importance. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 9318-9325).
- Casalicchio, G., Molnar, C., and Bischl, B. (2018). Visualizing the Feature Importance for Black Box Models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 655-670).
- Molnar, C., Casalicchio, G., and Bischl, B. (2019). Quantifying Model Complexity via Functional Decomposition for Better Post-Hoc Interpretability. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 193-204).
- Dandl, S., Molnar, C., Binder, M., and Bischl, B. (2020). Multi-objective counterfactual explanations. In: *Bäck T. et al. (eds) Parallel Problem Solving from Nature – PPSN XVI. PPSN 2020. Lecture Notes in Computer Science, vol 12269., vol 12269*, (pp. 448-469). Springer, Cham. https://doi.org/10.1007/978-3-030-58112-1_31

4. Interpretable Machine Learning–A Brief History, State-of-the-Art and Challenges

Contributing article:

Molnar, C., Casalicchio, G., and Bischl, B. (2020). Interpretable Machine Learning–A Brief History, State-of-the-Art and Challenges. In *ECML PKDD 2020 Workshops* (p. 417 – 431). Springer International Publishing (Cham). https://doi.org/10.1007/978-3-030-65965-3_28

Copyright information:

©Springer Nature Switzerland AG 2020

Author contributions:

Christoph Molnar wrote the whole paper. All authors added valuable input, suggested several notable modifications, proofread and revised the paper.

Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges*

Christoph Molnar¹[0000–0003–2331–868X], Giuseppe Casalicchio¹[0000–0001–5324–5966], and Bernd Bischl¹[0000–0001–6002–6980]

Department of Statistics, LMU Munich
Ludwigstr. 33, 80539 Munich, Germany
`christoph.molnar@stat.uni-muenchen.de`

Abstract. We present a brief history of the field of interpretable machine learning (IML), give an overview of state-of-the-art interpretation methods and discuss challenges. Research in IML has boomed in recent years. As young as the field is, it has over 200 years old roots in regression modeling and rule-based machine learning, starting in the 1960s. Recently, many new IML methods have been proposed, many of them model-agnostic, but also interpretation techniques specific to deep learning and tree-based ensembles. IML methods either directly analyze model components, study sensitivity to input perturbations, or analyze local or global surrogate approximations of the ML model. The field approaches a state of readiness and stability, with many methods not only proposed in research, but also implemented in open-source software. But many important challenges remain for IML, such as dealing with dependent features, causal interpretation, and uncertainty estimation, which need to be resolved for its successful application to scientific problems. A further challenge is a missing rigorous definition of interpretability, which is accepted by the community. To address the challenges and advance the field, we urge to recall our roots of interpretable, data-driven modeling in statistics and (rule-based) ML, but also to consider other areas such as sensitivity analysis, causal inference, and the social sciences.

Keywords: Interpretable Machine Learning · Explainable Artificial Intelligence

1 Introduction

Interpretability is often a deciding factor when a machine learning (ML) model is used in a product, a decision process, or in research. Interpretable machine learning (IML)¹ methods can be used to discover knowledge, to debug or justify

* This project is funded by the Bavarian State Ministry of Science and the Arts and coordinated by the Bavarian Research Institute for Digital Transformation (bidt) and supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibilities for its content.

¹ Sometimes the term Explainable AI is used.

the model and its predictions, and to control and improve the model [1]. In this paper, we take a look at the historical building blocks of IML and give an overview of methods to interpret models. We argue that IML has reached a state of readiness, but some challenges remain.

2 A Brief History of IML

A lot of IML research happened in the last couple of years. But learning interpretable models from data has a much longer tradition. Linear regression models were used by Gauss, Legendre, and Quetelet [109, 64, 37, 90] as early as the beginning of the 19th century and have since then grown into a vast array of regression analysis tools [115, 98], for example, generalized additive models [45] and elastic net [132]. The philosophy behind these statistical models is usually to make certain distributional assumptions or to restrict the model complexity beforehand and thereby imposing intrinsic interpretability of the model.

In ML, a slightly different modeling approach is pursued. Instead of restricting the model complexity beforehand, ML algorithms usually follow a non-linear, non-parametric approach, where model complexity is controlled through one or more hyperparameters and selected via cross-validation. This flexibility often results in less interpretable models with good predictive performance. A lot of ML research began in the second half of the 20th century with research on, for example, support vector machines in 1974 [119], early important work on neural networks in the 1960s [100], and boosting in 1990 [99]. Rule-based ML, which covers decision rules and decision trees, has been an active research area since the middle of the 20th century [35].

While ML algorithms usually focus on predictive performance, work on interpretability in ML – although underexplored – has existed for many years. The built-in feature importance measure of random forests [13] was one of the important IML milestones.² In the 2010s came the deep learning hype, after a deep neural network won the ImageNet challenge. A few years after that, the IML field really took off (around 2015), judging by the frequency of the search terms "Interpretable Machine Learning" and "Explainable AI" on Google (Figure 1, right) and papers published with these terms (Figure 1, left). Since then, many model-agnostic explanation methods have been introduced, which work for different types of ML models. But also model-specific explanation methods have been developed, for example, to interpret deep neural networks or tree ensembles. Regression analysis and rule-based ML remain important and active research areas to this day and are blending together (e.g., model-based trees [128], RuleFit [33]). Many extensions of the linear regression model exist [45, 25, 38] and new extensions are proposed until today [26, 14, 27, 117]. Rule-based ML also remains an active area of research (for example, [123, 66, 52]). Both regres-

² The random forest paper has been cited over 60,000 times (Google Scholar; September 2020) and there are many papers improving the importance measure ([110, 111, 44, 56]) which are also cited frequently.

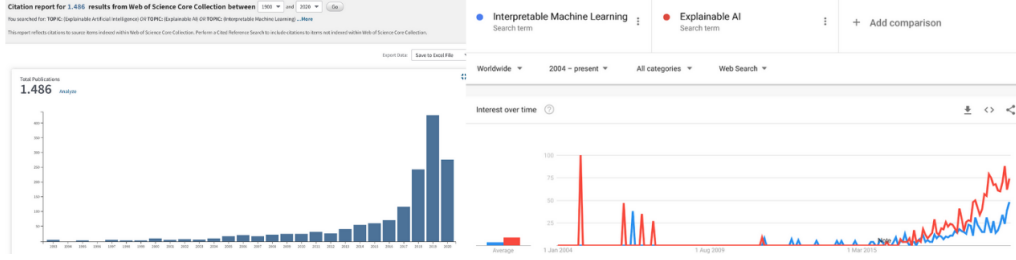


Fig. 1. Left: Citation count for research articles with keywords “Interpretable Machine Learning” or “Explainable AI” on Web of Science (accessed August 10, 2020). Right: Google search trends for “Interpretable Machine Learning” and “Explainable AI” (accessed August 10, 2020).

sion models and rule-based ML serve as stand-alone ML algorithms, but also as building blocks for many IML approaches.

3 Today

IML has reached a first state of readiness. Research-wise, the field is maturing in terms of methods surveys [75, 41, 120, 96, 1, 6, 23, 15], further consolidation of terms and knowledge [42, 22, 82, 97, 88, 17], and work about defining interpretability or evaluation of IML methods [74, 73, 95, 49]. We have a better understanding of weaknesses of IML methods in general [75, 79], but also specifically for methods such as permutation feature importance [51, 110, 7, 111], Shapley values [57, 113], counterfactual explanations [63], partial dependence plots [51, 50, 7] and saliency maps [2]. Open source software with implementations of various IML methods is available, for example, *iml* [76] and *DALEX* [11] for R [91] and *Alibi* [58] and *InterpretML* [83] for Python. Regulation such as GDPR and the need for ML trustability, transparency and fairness have sparked a discussion around further needs of interpretability [122]. IML has also arrived in industry [36], there are startups that focus on ML interpretability and also big tech companies offer software [126, 8, 43].

4 IML Methods

We distinguish IML methods by whether they analyze model components, model sensitivity³, or surrogate models, illustrated in Figure 4.⁴

³ Not to be confused with the research field of sensitivity analysis, which studies the uncertainty of outputs in mathematical models and systems. There are methodological overlaps (e.g., Shapley values), but also differences in methods and how input data distributions are handled.

⁴ Some surveys distinguish between *ante-hoc* (or *transparent design*, *white-box models*, *inherently interpretable model*) and *post-hoc* IML method, depending on whether

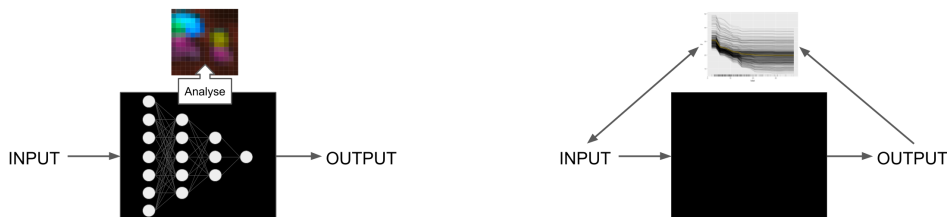


Fig. 2. Some IML approaches work by assigning meaning to individual model components (left), some by analyzing the model predictions for perturbations of the data (right). The surrogate approach, a mixture of the two other approaches, approximates the ML model using (perturbed) data and then analyzes the components of the interpretable surrogate model.

4.1 Analyzing Components of Interpretable Models

In order to analyze components of a model, it needs to be decomposable into parts that we can interpret individually. However, it is not necessarily required that the user understands the model in its entirety (simulatability [82]). Component analysis is always model-specific, because it is tied to the structure of the model.

Inherently interpretable models are models with (learned) structures and (learned) parameters which can be assigned a certain interpretation. In this context, linear regression models, decision trees and decision rules are considered to be interpretable [30, 54]. Linear regression models can be interpreted by analyzing components: The model structure, a weighted sum of features, allows to interpret the weights as the effects that the features have on the prediction.

Decision trees and other rule-based ML models have a learned structure (e.g., “IF feature $x_1 > 0$ and feature $x_2 \in \{A, B\}$, THEN predict 0.6”). We can interpret the learned structure to trace how the model makes predictions.

This only works up to a certain point in high-dimensional scenarios. Linear regression models with hundreds of features and complex interaction terms or deep decision trees are not that interpretable anymore. Some approaches aim to reduce the parts to be interpreted. For example, LASSO [98, 115] shrinks the coefficients in a linear model so that many of them become zero, and pruning techniques shorten trees.

4.2 Analyzing Components of More Complex Models

With a bit more effort, we can also analyze components of more complex black-box models.⁵ For example, the abstract features learned by a deep convolutional neural network (CNN) can be visualized by finding or generating images that

interpretability is considered at model design and training or after training, leaving the (black-box) model unchanged. Another category separates model-agnostic and model-specific methods.

⁵ This blurs the line between an “inherently interpretable” and a “black-box” model.

activate a feature map of the CNN [84]. For the random forest, the minimal depth distribution [85, 55] and the Gini importance [13] analyze the structure of the trees of the forest and can be used to quantify feature importance. Some approaches aim to make the parts of a model more interpretable with, for example, a monotonicity constraint [106] or a modified loss function for disentangling concepts learned by a convolutional neural network [130].

If an ML algorithm is well understood and frequently used in a community, like random forests in ecology research [19], model component analysis can be the correct tool, but it has the obvious disadvantage that it is tied to that specific model. And it does not combine well with the common model selection approach in ML, where one usually searches over a large class of different ML models via cross-validation.

4.3 Explaining Individual Predictions

Methods that study the sensitivity of an ML model are mostly model-agnostic and work by manipulating input data and analyzing the respective model predictions. These IML methods often treat the ML model as a closed system that receives feature values as an input and produces a prediction as output. We distinguish between local and global explanations.

Local methods explain individual predictions of ML models. Local explanation methods have received much attention and there has been a lot of innovation in the last years. Popular local IML methods are Shapley values [69, 112] and counterfactual explanations [122, 20, 81, 116, 118]. Counterfactual explanations explain predictions in the form of what-if scenarios, which builds on a rich tradition in philosophy [108]. According to findings in the social sciences [71], counterfactual explanations are “good” explanations because they are contrastive and focus on a few reasons. A different approach originates from collaborative game theory: The Shapley values [104] provide an answer on how to fairly share a payout among the players of a collaborative game. The collaborative game idea can be applied to ML where features (i.e., the players) collaborate to make a prediction (i.e., the payout) [112, 69, 68].

Some IML methods rely on model-specific knowledge to analyze how changes in the input features change the output. Saliency maps, an interpretation method specific for CNNs, make use of the network gradients to explain individual classifications. The explanations are in the form of heatmaps that show how changing a pixel can change the classification. The saliency map methods differ in how they backpropagate [114, 69, 80, 107, 105]. Additionally, model-agnostic versions [95, 69, 129] exist for analyzing image classifiers.

4.4 Explaining Global Model Behavior

Global model-agnostic explanation methods are used to explain the expected model behavior, i.e., how the model behaves on average for a given dataset. A useful distinction of global explanations are feature importance and feature effect.

Feature importance ranks features based on how relevant they were for the prediction. Permutation feature importance [28, 16] is a popular importance measure, originally suggested for random forests [13]. Some importance measures rely on removing features from the training data and retraining the model [65]. An alternative are variance-based measures [40]. See [125] for an overview of importance measures.

The feature effect expresses how a change in a feature changes the predicted outcome. Popular feature effect plots are partial dependence plots [32], individual conditional expectation curves [39], accumulated local effect plots [7], and the functional ANOVA [50]. Analyzing influential data instances, inspired by statistics, provides a different view into the model and describes how influential a data point was for a prediction [59].

4.5 Surrogate Models

Surrogate models⁶ are interpretable models designed to “copy” the behavior of the ML model. The surrogate approach treats the ML model as a black-box and only requires the input and output data of the ML model (similar to sensitivity analysis) to train a surrogate ML model. However, the interpretation is based on analyzing components of the interpretable surrogate model. Many IML methods are surrogate model approaches [89, 75, 72, 95, 34, 10, 18, 61] and differ, e.g., in the targeted ML model, the data sampling strategy, or the interpretable model that is used. There are also methods for extracting, e.g., decision rules from specific models based on their internal components such as neural network weights [5, 9]. LIME [95] is an example of a local surrogate method that explains individual predictions by learning an interpretable model with data in proximity to the data point to be explained. Numerous extensions of LIME exist, which try to fix issues with the original method, extend it to other tasks and data, or analyze its properties [53, 93, 92, 121, 47, 94, 103, 12].

5 Challenges

This section presents an incomplete overview of challenges for IML, mostly based on [79].

5.1 Statistical Uncertainty and Inference

Many IML methods such as permutation feature importance or Shapley values provide explanations without quantifying the uncertainty of the explanation. The model itself, but also its explanations, are computed from data and hence are subject to uncertainty. First research is working towards quantifying uncertainty of explanations, for example, for feature importance [124, 28, 4], layer-wise relevance propagation [24], and Shapley values [127].

⁶ Surrogate models are related to knowledge distillation and the teacher-student model.

In order to infer meaningful properties of the underlying data generating process, we have to make structural or distributional assumptions. Whether it is a classical statistical model, an ML algorithm or an IML procedure, these assumptions should be clearly stated and we need better diagnostic tools to test them. If we want to prevent statistical testing problems such as p-hacking [48] to reappear in IML, we have to become more rigorous in studying and quantifying the uncertainty of IML methods. For example, most IML methods for feature importance are not adapted for multiple testing, which is a classic mistake in a statistical analysis.

5.2 Causal Interpretation

Ideally, a model should reflect the true causal structure of its underlying phenomena, to enable causal interpretations. Arguably, causal interpretation is usually the goal of modeling if ML is used in science. But most statistical learning procedures reflect mere correlation structures between features and analyze the surface of the data generation process instead of its true inherent structure. Such causal structures would also make models more robust against adversarial attacks [101, 29], and more useful when used as a basis for decision making. Unfortunately, predictive performance and causality can be conflicting goals. For example, today’s weather directly causes tomorrow’s weather, but we might only have access to the feature “wet ground”. Using “wet ground” in the prediction model for “tomorrow’s weather” is useful as it has information about “today’s weather”, but we are not allowed to interpret it causally, because the confounder “today’s weather” is missing from the ML model. Further research is needed to understand when we are allowed to make causal interpretations of an ML model. First steps have been made for permutation feature importance [60] and Shapley values [70].

5.3 Feature Dependence

Feature dependence introduces problems with attribution and extrapolation. Attribution of importance and effects of features becomes difficult when features are, for example, correlated and therefore share information. Correlated features in random forests are preferred and attributed a higher importance [110, 51]. Many sensitivity analysis based methods permute features. When the permuted feature has some dependence with another feature, this association is broken and the resulting data points extrapolate to areas outside the distribution. The ML model was never trained on such combinations and will likely not be confronted with similar data points in an application. Therefore, extrapolation can cause misleading interpretations. There have been attempts to “fix” permutation-based methods, by using a conditional permutation scheme that respects the joint distribution of the data [78, 110, 28, 51]. The change from unconditional to conditional permutation changes the respective interpretation method [78, 7], or, in worst case, can break it [57, 113, 62].

5.4 Definition of Interpretability

A lack of definition for the term "interpretability" is a common critique of the field [67, 22]. How can we decide if a new method explains ML models better without a satisfying definition of interpretability? To evaluate the predictive performance of an ML model, we simply compute the prediction error on test data given the groundtruth label. To evaluate the interpretability of that same ML model is more difficult. We do not know what the groundtruth explanation looks like and have no straightforward way to quantify how interpretable a model is or how correct an explanation is. Instead of having one groundtruth explanation, various quantifiable aspects of interpretability are emerging [87, 86, 77, 46, 131, 3, 102, 87, 21, 31].

The two main ways of evaluating interpretability are objective evaluations, which are mathematically quantifiable metrics, and human-centered evaluations, which involve studies with either domain experts or lay persons. Examples of aspects of interpretability are sparsity, interaction strength, fidelity (how well an explanation approximates the ML model), sensitivity to perturbations, and a user's ability to run a model on a given input (simulatability). The challenge ahead remains to establish a best practice on how to evaluate interpretation methods and the explanations they produce. Here, we should also look at the field of human-computer interaction.

5.5 More Challenges Ahead

We focused mainly on the methodological, mathematical challenges in a rather static setting, where a trained ML model and the data are assumed as given and fixed. But ML models are usually not used in a static and isolated way, but are embedded in some process or product, and interact with people. A more dynamic and holistic view of the entire process, from data collection to the final consumption of the explained prediction is needed. This includes thinking how to explain predictions to individuals with diverse knowledge and backgrounds and about the need of interpretability on the level of an institution or society in general. This covers a wide range of fields, such as human-computer interaction, psychology and sociology. To solve the challenges ahead, we believe that the field has to reach out horizontally – to other domains – and vertically – drawing from the rich research in statistics and computer science.

References

1. Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access* **6**, 52138–52160 (2018)
2. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. In: *Advances in Neural Information Processing Systems*. pp. 9505–9515 (2018)
3. Akaike, H.: Information theory and an extension of the maximum likelihood principle. In: *Selected papers of Hirotugu Akaike*, pp. 199–213. Springer (1998)

4. Altmann, A., Tološi, L., Sander, O., Lengauer, T.: Permutation importance: a corrected feature importance measure. *Bioinformatics* **26**(10), 1340–1347 (2010)
5. Andrews, R., Diederich, J., Tickle, A.B.: Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-based systems* **8**(6), 373–389 (1995)
6. Anjomshoe, S., Najjar, A., Calvaresi, D., Främling, K.: Explainable agents and robots: Results from a systematic literature review. In: 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019. pp. 1078–1088. International Foundation for Autonomous Agents and Multiagent Systems (2019)
7. Apley, D.W., Zhu, J.: Visualizing the effects of predictor variables in black box supervised learning models. *arXiv preprint arXiv:1612.08468* (2016)
8. Arya, V., Bellamy, R.K., Chen, P.Y., Dhurandhar, A., Hind, M., Hoffman, S.C., Houde, S., Liao, Q.V., Luss, R., Mojsilovic, A., et al.: AI explainability 360: An extensible toolkit for understanding data and machine learning models. *Journal of Machine Learning Research* **21**(130), 1–6 (2020)
9. Augasta, M.G., Kathirvalavakumar, T.: Rule extraction from neural networks—a comparative study. In: International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012). pp. 404–408. IEEE (2012)
10. Bastani, O., Kim, C., Bastani, H.: Interpreting blackbox models via model extraction. *arXiv preprint arXiv:1705.08504* (2017)
11. Biecek, P.: DALEX: explainers for complex predictive models in r. *The Journal of Machine Learning Research* **19**(1), 3245–3249 (2018)
12. Botari, T., Hvilshøj, F., Izbicki, R., de Carvalho, A.C.: MeLIME: Meaningful local explanation for machine learning models. *arXiv preprint arXiv:2009.05818* (2020)
13. Breiman, L.: Random forests. *Machine learning* **45**(1), 5–32 (2001)
14. Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N.: Intelligent models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1721–1730 (2015)
15. Carvalho, D.V., Pereira, E.M., Cardoso, J.S.: Machine learning interpretability: A survey on methods and metrics. *Electronics* **8**(8), 832 (2019)
16. Casalicchio, G., Molnar, C., Bischl, B.: Visualizing the feature importance for black box models. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 655–670. Springer (2018)
17. Chromik, M., Schuessler, M.: A taxonomy for human subject evaluation of black-box explanations in XAI. In: ExSS-ATEC@ IUI (2020)
18. Craven, M., Shavlik, J.W.: Extracting tree-structured representations of trained networks. In: Advances in neural information processing systems. pp. 24–30 (1996)
19. Cutler, D.R., Edwards Jr, T.C., Beard, K.H., Cutler, A., Hess, K.T., Gibson, J., Lawler, J.J.: Random forests for classification in ecology. *Ecology* **88**(11), 2783–2792 (2007)
20. Dandl, S., Molnar, C., Binder, M., Bischl, B.: Multi-objective counterfactual explanations. *arXiv preprint arXiv:2004.11165* (2020)
21. Dhurandhar, A., Iyengar, V., Luss, R., Shanmugam, K.: TIP: typifying the interpretability of procedures. *arXiv preprint arXiv:1706.02952* (2017)
22. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (2017)
23. Du, M., Liu, N., Hu, X.: Techniques for interpretable machine learning. *Communications of the ACM* **63**(1), 68–77 (2019)

24. Fabi, K., Schneider, J.: On feature relevance uncertainty: A Monte Carlo dropout sampling approach. arXiv preprint arXiv:2008.01468 (2020)
25. Fahrmeir, L., Tutz, G.: Multivariate statistical modelling based on generalized linear models. Springer Science & Business Media (2013)
26. Fasiolo, M., Nedellec, R., Goude, Y., Wood, S.N.: Scalable visualization methods for modern generalized additive models. *Journal of computational and Graphical Statistics* **29**(1), 78–86 (2020)
27. Fasiolo, M., Wood, S.N., Zaffran, M., Nedellec, R., Goude, Y.: Fast calibrated additive quantile regression. *Journal of the American Statistical Association* pp. 1–11 (2020)
28. Fisher, A., Rudin, C., Dominici, F.: All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research* **20**(177), 1–81 (2019)
29. Freiesleben, T.: Counterfactual explanations & adversarial examples—common grounds, essential differences, and potential transfers. arXiv preprint arXiv:2009.05487 (2020)
30. Freitas, A.A.: Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter* **15**(1), 1–10 (2014)
31. Friedler, S.A., Roy, C.D., Scheidegger, C., Slack, D.: Assessing the local interpretability of machine learning models. arXiv preprint arXiv:1902.03501 (2019)
32. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. *Annals of statistics* pp. 1189–1232 (2001)
33. Friedman, J.H., Popescu, B.E., et al.: Predictive learning via rule ensembles. *The Annals of Applied Statistics* **2**(3), 916–954 (2008)
34. Frosst, N., Hinton, G.: Distilling a neural network into a soft decision tree. arXiv preprint arXiv:1711.09784 (2017)
35. Fürnkranz, J., Gamberger, D., Lavrač, N.: Foundations of rule learning. Springer Science & Business Media (2012)
36. Gade, K., Geyik, S.C., Kenthapadi, K., Mithal, V., Taly, A.: Explainable AI in industry. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. pp. 3203–3204 (2019)
37. Gauss, C.F.: *Theoria motus corporum coelestium in sectionibus conicis solem ambientium*, vol. 7. Perthes et Besser (1809)
38. Gelman, A., Hill, J.: *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press (2006)
39. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics* **24**(1), 44–65 (2015)
40. Greenwell, B.M., Boehmke, B.C., McCarthy, A.J.: A simple and effective model-based variable importance measure. arXiv preprint arXiv:1805.04755 (2018)
41. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* **51**(5), 1–42 (2018)
42. Hall, M., Harborne, D., Tomsett, R., Galetic, V., Quintana-Amate, S., Nottle, A., Preece, A.: A systematic method to understand requirements for explainable AI(XAI) systems. In: *Proceedings of the IJCAI Workshop on eXplainable Artificial Intelligence (XAI 2019)*, Macau, China (2019)
43. Hall, P., Gill, N., Kurka, M., Phan, W.: Machine learning interpretability with h2o driverless AI. H2O. ai. URL: <http://docs.h2o.ai/driverless-ai/latest-stable/docs/booklets/MLIBooklet.pdf> (2017)

44. Hapfelmeier, A., Hothorn, T., Ulm, K., Strobl, C.: A new variable importance measure for random forests with missing data. *Statistics and Computing* **24**(1), 21–34 (2014)
45. Hastie, T.J., Tibshirani, R.J.: Generalized additive models, vol. 43. CRC press (1990)
46. Hauenstein, S., Wood, S.N., Dormann, C.F.: Computing AIC for black-box models using generalized degrees of freedom: A comparison with cross-validation. *Communications in Statistics-Simulation and Computation* **47**(5), 1382–1396 (2018)
47. Haunschmid, V., Manilow, E., Widmer, G.: audioLIME: Listenable explanations using source separation. *arXiv preprint arXiv:2008.00582* (2020)
48. Head, M.L., Holman, L., Lanfear, R., Kahn, A.T., Jennions, M.D.: The extent and consequences of p-hacking in science. *PLoS Biol* **13**(3), e1002106 (2015)
49. Hoffman, R.R., Mueller, S.T., Klein, G., Litman, J.: Metrics for explainable AI: Challenges and prospects. *arXiv preprint arXiv:1812.04608* (2018)
50. Hooker, G.: Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics* **16**(3), 709–732 (2007)
51. Hooker, G., Mentch, L.: Please stop permuting features: An explanation and alternatives. *arXiv preprint arXiv:1905.03151* (2019)
52. Hothorn, T., Hornik, K., Zeileis, A.: ctree: Conditional inference trees. *The Comprehensive R Archive Network* **8** (2015)
53. Hu, L., Chen, J., Nair, V.N., Sudjianto, A.: Locally interpretable models and effects based on supervised partitioning (LIME-SUP). *arXiv preprint arXiv:1806.00663* (2018)
54. Huysmans, J., Dejaeger, K., Mues, C., Vanthienen, J., Baesens, B.: An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems* **51**(1), 141–154 (2011)
55. Ishwaran, H., Kogalur, U.B., Gorodeski, E.Z., Minn, A.J., Lauer, M.S.: High-dimensional variable selection for survival data. *Journal of the American Statistical Association* **105**(489), 205–217 (2010)
56. Ishwaran, H., et al.: Variable importance in binary regression trees and forests. *Electronic Journal of Statistics* **1**, 519–537 (2007)
57. Janzing, D., Minorics, L., Blöbaum, P.: Feature relevance quantification in explainable AI: A causality problem. *arXiv preprint arXiv:1910.13413* (2019)
58. Klaise, J., Van Looveren, A., Vacanti, G., Coca, A.: Alibi: Algorithms for monitoring and explaining machine learning models. URL <https://github.com/SeldonIO/alibi> (2020)
59. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730* (2017)
60. König, G., Molnar, C., Bischl, B., Grosse-Wentrup, M.: Relative feature importance. *arXiv preprint arXiv:2007.08283* (2020)
61. Krishnan, S., Wu, E.: Palm: Machine learning explanations for iterative debugging. In: *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*. pp. 1–6 (2017)
62. Kumar, I.E., Venkatasubramanian, S., Scheidegger, C., Friedler, S.: Problems with Shapley-value-based explanations as feature importance measures. *arXiv preprint arXiv:2002.11097* (2020)
63. Laugel, T., Lesot, M.J., Marsala, C., Renard, X., Detyniecki, M.: The dangers of post-hoc interpretability: Unjustified counterfactual explanations. *arXiv preprint arXiv:1907.09294* (2019)

64. Legendre, A.M.: Nouvelles méthodes pour la détermination des orbites des comètes. F. Didot (1805)
65. Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R.J., Wasserman, L.: Distribution-free predictive inference for regression. *Journal of the American Statistical Association* **113**(523), 1094–1111 (2018)
66. Letham, B., Rudin, C., McCormick, T.H., Madigan, D., et al.: Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics* **9**(3), 1350–1371 (2015)
67. Lipton, Z.C.: The mythos of model interpretability. *Queue* **16**(3), 31–57 (2018)
68. Lundberg, S.M., Erion, G.G., Lee, S.I.: Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888* (2018)
69. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: *Advances in neural information processing systems*. pp. 4765–4774 (2017)
70. Ma, S., Tourani, R.: Predictive and causal implications of using Shapley value for model interpretation. In: *Proceedings of the 2020 KDD Workshop on Causal Discovery*. pp. 23–38. PMLR (2020)
71. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* **267**, 1–38 (2019)
72. Ming, Y., Qu, H., Bertini, E.: Rulematrix: Visualizing and understanding classifiers with rules. *IEEE transactions on visualization and computer graphics* **25**(1), 342–352 (2018)
73. Mohseni, S., Ragan, E.D.: A human-grounded evaluation benchmark for local explanations of machine learning. *arXiv preprint arXiv:1801.05075* (2018)
74. Mohseni, S., Zarei, N., Ragan, E.D.: A multidisciplinary survey and framework for design and evaluation of explainable AI systems. *arXiv pp. arXiv-1811* (2018)
75. Molnar, C.: Interpretable Machine Learning (2019), <https://christophm.github.io/interpretable-ml-book/>
76. Molnar, C., Bischl, B., Casalicchio, G.: iml: An R package for interpretable machine learning. *JOSS* **3**(26), 786 (2018)
77. Molnar, C., Casalicchio, G., Bischl, B.: Quantifying model complexity via functional decomposition for better post-hoc interpretability. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. pp. 193–204. Springer (2019)
78. Molnar, C., König, G., Bischl, B., Casalicchio, G.: Model-agnostic feature importance and effects with dependent features—a conditional subgroup approach. *arXiv preprint arXiv:2006.04628* (2020)
79. Molnar, C., König, G., Herbringer, J., Freiesleben, T., Dandl, S., Scholbeck, C.A., Casalicchio, G., Grosse-Wentrup, M., Bischl, B.: Pitfalls to avoid when interpreting machine learning models. *arXiv preprint arXiv:2007.04131* (2020)
80. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.R.: Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition* **65**, 211–222 (2017)
81. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. pp. 607–617 (2020)
82. Murdoch, W.J., Singh, C., Kumbier, K., Abbasi-Asl, R., Yu, B.: Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences* **116**(44), 22071–22080 (2019)
83. Nori, H., Jenkins, S., Koch, P., Caruana, R.: Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223* (2019)

84. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. *Distill* (2017). <https://doi.org/10.23915/distill.00007>, <https://distill.pub/2017/feature-visualization>
85. Paluszynska, A., Biecek, P., Jiang, Y.: randomForestExplainer: Explaining and Visualizing Random Forests in Terms of Variable Importance (2020), <https://CRAN.R-project.org/package=randomForestExplainer>, r package version 0.10.1
86. Philipp, M., Rusch, T., Hornik, K., Strobl, C.: Measuring the stability of results from supervised statistical learning. *Journal of Computational and Graphical Statistics* **27**(4), 685–700 (2018)
87. Poursabzi-Sangdeh, F., Goldstein, D.G., Hofman, J.M., Vaughan, J.W., Wallach, H.: Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810* (2018)
88. Preece, A., Harborne, D., Braines, D., Tomsett, R., Chakraborty, S.: Stakeholders in explainable AI. *arXiv preprint arXiv:1810.00184* (2018)
89. Puri, N., Gupta, P., Agarwal, P., Verma, S., Krishnamurthy, B.: Magix: Model agnostic globally interpretable explanations. *arXiv preprint arXiv:1706.07160* (2017)
90. Quetelet, L.A.J.: Recherches sur la population, les naissances, les décès, les prisons, les dépôts de mendicité, etc. dans le royaume des Pays-Bas (1827)
91. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2020), <https://www.R-project.org/>
92. Rabold, J., Deininger, H., Siebers, M., Schmid, U.: Enriching visual with verbal explanations for relational concepts—combining LIME with Aleph. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. pp. 180–192. Springer (2019)
93. Rabold, J., Siebers, M., Schmid, U.: Explaining black-box classifiers with ilp—empowering LIME with aleph to approximate non-linear decisions with relational rules. In: *International Conference on Inductive Logic Programming*. pp. 105–117. Springer (2018)
94. Rahnema, A.H.A., Boström, H.: A study of data and label shift in the LIME framework. *arXiv preprint arXiv:1910.14421* (2019)
95. Ribeiro, M.T., Singh, S., Guestrin, C.: ” why should i trust you?” explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. pp. 1135–1144 (2016)
96. Rosenfeld, A., Richardson, A.: Explainability in human–agent systems. *Autonomous Agents and Multi-Agent Systems* **33**(6), 673–705 (2019)
97. Samek, W., Müller, K.R.: Towards explainable artificial intelligence. In: *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 5–22. Springer (2019)
98. Santosa, F., Symes, W.W.: Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing* **7**(4), 1307–1330 (1986)
99. Schapire, R.E.: The strength of weak learnability. *Machine learning* **5**(2), 197–227 (1990)
100. Schmidhuber, J.: Deep learning in neural networks: An overview. *Neural networks* **61**, 85–117 (2015)
101. Schölkopf, B.: Causality for machine learning. *arXiv preprint arXiv:1911.10500* (2019)

102. Schwarz, G., et al.: Estimating the dimension of a model. *The annals of statistics* **6**(2), 461–464 (1978)
103. Shankaranarayana, S.M., Runje, D.: ALIME: Autoencoder based approach for local interpretability. In: *International Conference on Intelligent Data Engineering and Automated Learning*. pp. 454–463. Springer (2019)
104. Shapley, L.S.: A value for n-person games. *Contributions to the Theory of Games* **2**(28), 307–317 (1953)
105. Shrikumar, A., Greenside, P., Shcherbina, A., Kundaje, A.: Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713* (2016)
106. Sill, J.: Monotonic networks. In: *Advances in neural information processing systems*. pp. 661–667 (1998)
107. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013)
108. Starr, W.: *Counterfactuals* (2019)
109. Stigler, S.M.: *The history of statistics: The measurement of uncertainty before 1900*. Harvard University Press (1986)
110. Strobl, C., Boulesteix, A.L., Kneib, T., Augustin, T., Zeileis, A.: Conditional variable importance for random forests. *BMC bioinformatics* **9**(1), 307 (2008)
111. Strobl, C., Boulesteix, A.L., Zeileis, A., Hothorn, T.: Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics* **8**(1), 25 (2007)
112. Štrumbelj, E., Kononenko, I.: Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems* **41**(3), 647–665 (2014)
113. Sundararajan, M., Najmi, A.: The many Shapley values for model explanation. *arXiv preprint arXiv:1908.08474* (2019)
114. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365* (2017)
115. Tibshirani, R.: Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* **58**(1), 267–288 (1996)
116. Tolomei, G., Silvestri, F., Haines, A., Lalmas, M.: Interpretable predictions of tree-based ensembles via actionable feature tweaking. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. pp. 465–474 (2017)
117. Ustun, B., Rudin, C.: Supersparse linear integer models for optimized medical scoring systems. *Machine Learning* **102**(3), 349–391 (2016)
118. Ustun, B., Spangher, A., Liu, Y.: Actionable recourse in linear classification. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. pp. 10–19 (2019)
119. Vapnik, V., Chervonenkis, A.: *Theory of pattern recognition* (1974)
120. Vilone, G., Longo, L.: Explainable artificial intelligence: a systematic review. *arXiv preprint arXiv:2006.00093* (2020)
121. Visani, G., Bagli, E., Chesani, F.: Optilime: Optimized LIME explanations for diagnostic computer algorithms. *arXiv preprint arXiv:2006.05714* (2020)
122. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.* **31**, 841 (2017)
123. Wang, F., Rudin, C.: Falling rule lists. In: *Artificial Intelligence and Statistics*. pp. 1013–1022 (2015)

124. Watson, D.S., Wright, M.N.: Testing conditional independence in supervised learning algorithms. arXiv preprint arXiv:1901.09917 (2019)
125. Wei, P., Lu, Z., Song, J.: Variable importance analysis: a comprehensive review. *Reliability Engineering & System Safety* **142**, 399–432 (2015)
126. Wexler, J., Pushkarna, M., Bolukbasi, T., Wattenberg, M., Viégas, F., Wilson, J.: The what-if tool: Interactive probing of machine learning models. *IEEE transactions on visualization and computer graphics* **26**(1), 56–65 (2019)
127. Williamson, B.D., Feng, J.: Efficient nonparametric statistical inference on population feature importance using Shapley values. arXiv preprint arXiv:2006.09481 (2020)
128. Zeileis, A., Hothorn, T., Hornik, K.: Model-based recursive partitioning. *Journal of Computational and Graphical Statistics* **17**(2), 492–514 (2008)
129. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: *European conference on computer vision*. pp. 818–833. Springer (2014)
130. Zhang, Q., Nian Wu, Y., Zhu, S.C.: Interpretable convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 8827–8836 (2018)
131. Zhou, Q., Liao, F., Mou, C., Wang, P.: Measuring interpretability for different types of machine learning models. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. pp. 295–308 (2018)
132. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)* **67**(2), 301–320 (2005)

5. General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models

Contributing article:

Molnar, C., König, G., Herbringer, J., Freiesleben, T., Dandl, S., Scholbeck, C., Casalicchio, G., Grosse-Wentrup, M., and Bischl, B. (2020). General Pitfalls of Model-Agnostic Interpretable Machine Learning. *To appear in: xxAI – Beyond explainable Artificial Intelligence. Lecture Notes in Artificial Intelligence*, vol. 13200. Springer, Cham..

Copyright information:

Creative Commons Attribution 4.0 International License (CC BY 4.0).

Author contributions:

Christoph Molnar wrote parts of abstract and introduction, and the chapters on bad model generalization, unnecessary use of complex models, and ignoring estimation uncertainty. He also initiated and coordinated the project. All other chapters were mainly written by the co-authors. All authors, including Christoph Molnar, added input to the chapters in which they were not involved as authors, and proofread and revised the paper.

General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models^{*}

Christoph Molnar^{1,7}[0000–0003–2331–868X], Gunnar König^{1,4}[0000–0001–6141–4942],
 Julia Herbinger¹[0000–0003–0430–8523], Timo Freiesleben^{2,3}[0000–0003–1338–3293],
 Susanne Dandl¹[0000–0003–4324–4163], Christian A.
 Scholbeck¹[0000–0001–6607–4895], Giuseppe Casalicchio¹[0000–0001–5324–5966],
 Moritz Grosse-Wentrup^{4,5,6}[0000–0001–9787–2291], and Bernd
 Bischl¹[0000–0001–6002–6980]

¹ Department of Statistics, LMU Munich, Munich, Germany

² Munich Center for Mathematical Philosophy, LMU Munich, Munich, Germany

³ Graduate School of Systemic Neurosciences, LMU Munich, Munich, Germany

⁴ Research Group Neuroinformatics, Faculty for Computer Science, University of
 Vienna, Vienna, Austria

⁵ Research Platform Data Science @ Uni Vienna, Vienna, Austria

⁶ Vienna Cognitive Science Hub, Vienna, Austria

⁷ Leibniz Institute for Prevention Research and Epidemiology – BIPS GmbH, Bremen,
 Germany

{christoph.molnar.ai}@gmail.com

Abstract. An increasing number of model-agnostic interpretation techniques for machine learning (ML) models such as partial dependence plots (PDP), permutation feature importance (PFI) and Shapley values provide insightful model interpretations, but can lead to wrong conclusions if applied incorrectly. We highlight many general pitfalls of ML model interpretation, such as using interpretation techniques in the wrong context, interpreting models that do not generalize well, ignoring feature dependencies, interactions, uncertainty estimates and issues in high-dimensional settings, or making unjustified causal interpretations, and illustrate them with examples. We focus on pitfalls for global methods that describe the average model behavior, but many pitfalls also apply to local methods that explain individual predictions. Our paper addresses ML practitioners by raising awareness of pitfalls and identifying solutions for correct model interpretation, but also addresses ML researchers by discussing open issues for further research.

Keywords: Interpretable Machine Learning · Explainable AI

^{*} This work is funded by the Bavarian State Ministry of Science and the Arts (coordinated by the Bavarian Research Institute for Digital Transformation (bidtt)), by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A, by the German Research Foundation (DFG), Emmy Noether Grant 437611051, and by the Graduate School of Systemic Neurosciences (GSN) Munich. The authors of this work take full responsibilities for its content.

1 Introduction

In recent years, both industry and academia have increasingly shifted away from parametric models, such as generalized linear models, and towards non-parametric and non-linear machine learning (ML) models such as random forests, gradient boosting, or neural networks. The major driving force behind this development has been a considerable outperformance of ML over traditional models on many prediction tasks [32]. In part, this is because most ML models handle interactions and non-linear effects automatically. While classical statistical models – such as generalized additive models (GAMs) – also support the inclusion of interactions and non-linear effects, they come with the increased cost of having to (manually) specify and evaluate these modeling options. The benefits of many ML models are partly offset by their lack of interpretability, which is of major importance in many applications. For certain model classes (e.g. linear models), feature effects or importance scores can be directly inferred from the learned parameters and the model structure. In contrast, it is more difficult to extract such information from complex non-linear ML models that, for instance, do not have intelligible parameters and are hence often considered black boxes. However, model-agnostic interpretation methods allow us to harness the predictive power of ML models while gaining insights into the black-box model. These interpretation methods are already applied in many different fields. Applications of interpretable machine learning (IML) include understanding pre-evacuation decision-making [124] with partial dependence plots [36], inferring behavior from smartphone usage [106,105] with the help of permutation feature importance [107] and accumulated local effect plots [3], or understanding the relation between critical illness and health records [70] using Shapley additive explanations (SHAP) [78]. Given the widespread application of interpretable machine learning, it is crucial to highlight potential pitfalls, that, in the worst case, can produce incorrect conclusions.

This paper focuses on pitfalls for model-agnostic IML methods, i.e. methods that can be applied to any predictive model. Model-specific methods, in contrast, are tied to a certain model class (e.g. saliency maps [57] for gradient-based models, such as neural networks), and are mainly considered out-of-scope for this work. We focus on pitfalls for global interpretation methods, which describe the expected behavior of the entire model with respect to the whole data distribution. However, many of the pitfalls also apply to local explanation methods, which explain individual predictions or classifications. Global methods include the partial dependence plot (PDP) [36], partial importance (PI) [19], accumulated local effects (ALE) [3], or the permutation feature importance (PFI) [12,33,19]. Local methods include the individual conditional expectation (ICE) curves [38], individual conditional importance (ICI) [19], local interpretable model-agnostic explanations (LIME) [94], Shapley values [108] and SHapley Additive exPlanations (SHAP) [78,77] or counterfactual explanations [115,26]. Furthermore, we distinguish between feature effect and feature importance methods. A feature effect indicates the direction and magnitude of a change in predicted outcome due to changes in feature values. Effect methods include Shapley values, SHAP, LIME, ICE, PDP, or ALE. Feature importance methods quantify the contribution of a

		Local	Global
	Feature Effects	ICE LIME Counterfactuals Shapley Values SHAP	PDP ALE
	Importance	ICI	PI PFI SAGE

Fig. 1. Selection of popular model-agnostic interpretation techniques, classified as local or global, and as effect or importance methods.

feature to the model performance (e.g. via a loss function) or to the variance of the prediction function. Importance methods include the PFI, ICI, PI, or SAGE. See Figure 1 for a visual summary.

The interpretation of ML models can have subtle pitfalls. Since many of the interpretation methods work by similar principles of manipulating data and “probing” the model [100], they also share many pitfalls. The sources of these pitfalls can be broadly divided into three categories: (1) application of an unsuitable ML model which does not reflect the underlying data generating process very well, (2) inherent limitations of the applied IML method, and (3) wrong application of an IML method. Typical pitfalls for (1) are bad model generalization or the unnecessary use of complex ML models. Applying an IML method in a wrong way (3) often results from the users’ lack of knowledge of the inherent limitations of the chosen IML method (2). For example, if feature dependencies and interactions are present, potential extrapolations might lead to misleading interpretations for perturbation-based IML methods (inherent limitation). In such cases, methods like PFI might be a wrong choice to quantify feature importance.

Sources of pitfall	Sections
Unsuitable ML model	3, 4
Limitation of IML method	5.1, 6.1, 6.2, 9.1, 9.2
Wrong application of IML method	2, 5.2, 5.3, 7, 8, 9.3, 10

Table 1. Categorization of the pitfalls by source.

Contributions: We uncover and review general pitfalls of model-agnostic interpretation techniques. The categorization of these pitfalls into different sources is provided in Table 1. Each section describes and illustrates a pitfall, reviews possible solutions for practitioners to circumvent the pitfall, and

discusses open issues that require further research. The pitfalls are accompanied by illustrative examples for which the code can be found in this repository: https://github.com/compstat-lmu/code_pitfalls_uml.git. In addition to reproducing our examples, we invite readers to use this code as a starting point for their own experiments and explorations.

Related Work: Rudin et al. [96] present principles for interpretability and discuss challenges for model interpretation with a focus on inherently interpretable models. Das et al. [27] survey methods for explainable AI and discuss challenges with a focus on saliency maps for neural networks. A general warning about using and explaining ML models for high stakes decisions has been brought forward by Rudin [95], in which the author argues against model-agnostic techniques in favor of inherently interpretable models. Krishnan [64] criticizes the general conceptual foundation of interpretability, but does not dispute the usefulness of available methods. Likewise, Lipton [73] criticizes interpretable ML for its lack of causal conclusions, trust, and insights, but the author does not discuss any pitfalls in detail. Specific pitfalls due to dependent features are discussed by Hooker [54] for PDPs and functional ANOVA as well as by Hooker and Mentch [55] for feature importance computations. Hall [47] discusses recommendations for the application of particular interpretation methods but does not address general pitfalls.

2 Assuming One-Fits-All Interpretability

Pitfall: Assuming that a single IML method fits in all interpretation contexts can lead to dangerous misinterpretation. IML methods condense the complexity of ML models into human-intelligible descriptions that only provide insight into specific aspects of the model and data. The vast number of interpretation methods make it difficult for practitioners to choose an interpretation method that can answer their question. Due to the wide range of goals that are pursued under the umbrella term “interpretability”, the methods differ in which aspects of the model and data they describe.

For example, there are several ways to quantify or rank the features according to their relevance. The relevance measured by PFI can be very different from the relevance measured by the SHAP importance. If a practitioner aims to gain insight into the relevance of a feature regarding the model’s generalization error, a loss-based method (on unseen test data) such as PFI should be used. If we aim to expose which features the model relies on for its prediction or classification – irrespective of whether they aid the model’s generalization performance – PFI on test data is misleading. In such scenarios, one should quantify the relevance of a feature regarding the model’s prediction (and not the model’s generalization error) using methods like the SHAP importance [76].

We illustrate the difference in Figure 2. We simulated a data-generating process where the target is completely independent of all features. Hence, the features are just noise and should not contribute to the model’s generalization

error. Consequently, the features are not considered relevant by PFI on test data. However, the model mechanistically relies on a number of spuriously correlated features. This reliance is exposed by marginal global SHAP importance.

As the example demonstrates, it would be misleading to view the PFI computed on test data or global SHAP as one-fits-all feature importance techniques. Like any IML method, they can only provide insight into certain aspects of model and data.

Many pitfalls in this paper arise from situations where an IML method that was designed for one purpose is applied in an unsuitable context. For example, extrapolation (Section 5.1) can be problematic when we aim to study how the model behaves under realistic data but simultaneously can be the correct choice if we want to study the sensitivity to a feature outside the data distribution.

For some IML techniques – especially local methods – even the same method can provide very different explanations, depending on the choice of hyperparameters: For counterfactuals, explanation goals are encoded in their optimization metrics [34,26] such as sparsity and data faithfulness; The scope and meaning of LIME explanations depend on the kernel width and the notion of complexity [8,37].

Solution: The suitability of an IML method cannot be evaluated with respect to one-fits-all interpretability but must be motivated and assessed with respect to well-defined interpretation goals. Similarly, practitioners must tailor the choice of the IML method and its respective hyperparameters to the interpretation context. This implies that these goals need to be clearly stated in a detailed manner *before* any analysis – which is still often not the case.

Open Issues: Since IML methods themselves are subject to interpretation, practitioners must be informed about which conclusions can or cannot be drawn given different choices of IML technique. In general, there are three aspects to be considered: (a) an intuitively understandable and plausible algorithmic construction of the IML method to achieve an explanation; (b) a clear mathematical axiomatization of interpretation goals and properties, which are linked by proofs and theoretical considerations to IML methods, and properties of models and data characteristics; (c) a practical translation for practitioners of the axioms from (b) in terms of what an IML method provides and what not, ideally with implementable guidelines and diagnostic checks for violated assumptions to guarantee correct interpretations. While (a) is nearly always given for any published method, much work remains for (b) and (c).

3 Bad Model Generalization

Pitfall: Under- or overfitting models can result in misleading interpretations with respect to the true feature effects and importance scores, as the model does not match the underlying data-generating process well [39]. Formally, most IML

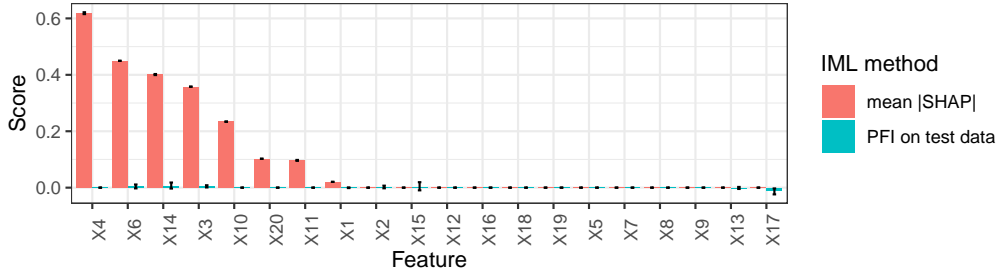


Fig. 2. Assuming one-fits-all interpretability. A default `xgboost` regression model that minimizes the mean squared error (MSE) was fitted on 20 independently and uniformly distributed features to predict another independent, uniformly sampled target. In this setting, predicting the (unconditional) mean $\mathbb{E}[Y]$ in a constant model is optimal. The learner overfits due to a small training data size. Mean marginal SHAP (red, error bars indicate 0.05 and 0.95 quantiles) exposes all mechanistically used features. In contrast, PFI on test data (blue, error bars indicate 0.05 and 0.95 quantiles) considers all features to be irrelevant, since no feature contributes to the generalization performance.

methods are designed to interpret the model instead of drawing inferences about the data-generating process. In practice, however, the latter is often the goal of the analysis, and then an interpretation can only be as good as its underlying model. If a model approximates the data-generating process well enough, its interpretation should reveal insights into the underlying process.

Solution: In-sample evaluation (i.e. on training data) should not be used to assess the performance of ML models due to the risk of overfitting on the training data, which will lead to overly optimistic performance estimates. We must resort to out-of-sample validation based on resampling procedures such as holdout for larger datasets or cross-validation, or even repeated cross-validation for small sample size scenarios. These resampling procedures are readily available in software [67,89], and well-studied in theory as well as practice [4,11,104], although rigorous analysis of cross-validation is still considered an open problem [103]. Nested resampling is necessary, when computational model selection and hyperparameter tuning are involved [10]. This is important, as the Bayes error for most practical situations is unknown, and we cannot make absolute statements about whether a model already optimally fits the data.

Figure 3 shows the mean squared errors for a simulated example on both training and test data for a support vector machine (SVM), a random forest, and a linear model. Additionally, PDPs for all models are displayed, which show to what extent each model’s effect estimates deviate from the ground truth. The linear model is unable to represent the non-linear relationship, which is reflected in a high error on both test and training data and the linear PDPs. In contrast, the random forest has a low training error but a much higher test error, which indicates overfitting. Also, the PDPs for the random forest display overfitting behavior, as the curves are quite noisy, especially at the lower and upper value

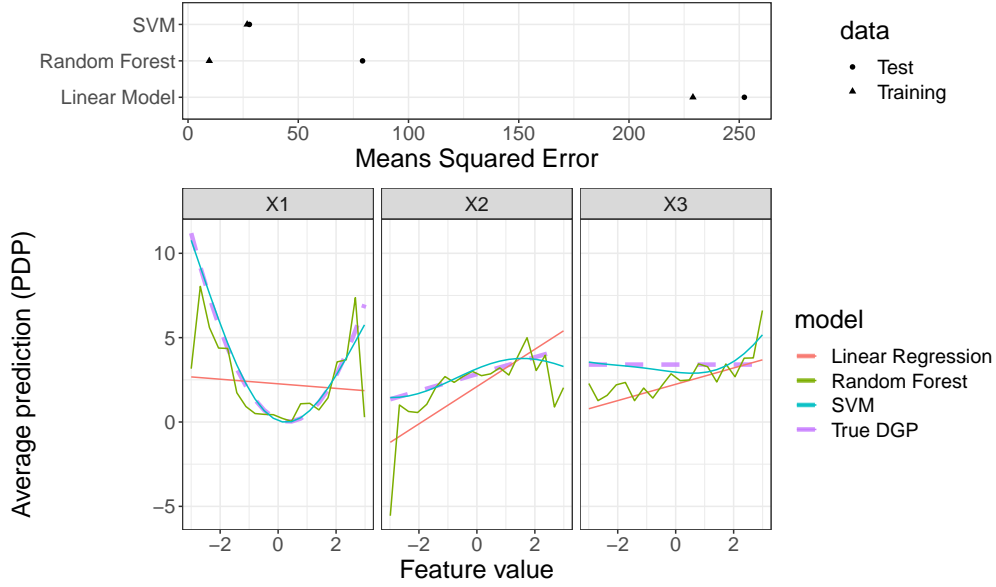


Fig. 3. Bad model generalization. Top: Performance estimates on training and test data for a linear regression model (underfitting), a random forest (overfitting) and a support vector machine with radial basis kernel (good fit). The three features are drawn from a uniform distribution, and the target was generated as $Y = X_1^2 + X_2 - 5X_1X_2 + \epsilon$, with $\epsilon \sim N(0, 5)$. **Bottom:** PDPs for the data-generating process (DGP) – which is the ground truth – and for the three models.

ranges of each feature. The SVM with both low training and test error comes closest to the true PDPs.

4 Unnecessary Use of Complex Models

Pitfall: A common mistake is to use an opaque, complex ML model when an interpretable model would have been sufficient, i.e. when the performance of interpretable models is only negligibly worse – or maybe the same or even better – than that of the ML model. Although model-agnostic methods can shed light on the behavior of complex ML models, inherently interpretable models still offer a higher degree of transparency [95] and considering them increases the chance of discovering the true data-generating function [23]. What constitutes an interpretable model is highly dependent on the situation and target audience, as even a linear model might be difficult to interpret when many features and interactions are involved.

It is commonly believed that complex ML models always outperform more interpretable models in terms of accuracy and should thus be preferred. However, there are several examples where interpretable models have proven to be serious competitors: More than 15 years ago, Hand [49] demonstrated that simple

models often achieve more than 90% of the predictive power of potentially highly complex models across the UCI benchmark data repository and concluded that such models often should be preferred due to their inherent interpretability; Makridakis et al. [79] systematically compared various ML models (including long-short-term-memory models and multi-layer neural networks) to statistical models (e.g. damped exponential smoothing and the Theta method) in time series forecasting tasks and found that the latter consistently show greater predictive accuracy; Kuhle et al. [65] found that random forests, gradient boosting and neural networks did not outperform logistic regression in predicting fetal growth abnormalities; Similarly, Wu et al. [120] have shown that a logistic regression model performs as well as AdaBoost and even better than an SVM in predicting heart disease from electronic health record data; Baesens et al. [7] showed that simple interpretable classifiers perform competitively for credit scoring, and in an update to the study the authors note that “the complexity and/or recency of a classifier are misleading indicators of its prediction performance” [71].

Solution: We recommend starting with simple, interpretable models such as linear regression models and decision trees. Generalized additive models (GAM) [50] can serve as a gradual transition between simple linear models and more complex machine learning models. GAMs have the desirable property that they can additively model smooth, non-linear effects and provide PDPs out-of-the-box, but without the potential pitfall of masking interactions (see Section 6). The additive model structure of a GAM is specified before fitting the model so that only the pre-specified feature or interaction effects are estimated. Interactions between features can be added manually or algorithmically (e.g. via a forward greedy search) [18]. GAMs can be fitted with component-wise boosting [99]. The boosting approach allows to smoothly increase model complexity, from sparse linear models to more complex GAMs with non-linear effects and interactions. This smooth transition provides insight into the tradeoffs between model simplicity and performance gains. Furthermore, component-wise boosting has an in-built feature selection mechanism as the model is build incrementally, which is especially useful in high-dimensional settings (see Section 9.1). The predictive performance of models of different complexity should be carefully measured and compared. Complex models should only be favored if the additional performance gain is both significant and relevant – a judgment call that the practitioner must ultimately make. Starting with simple models is considered best practice in data science, independent of the question of interpretability [23]. The comparison of predictive performance between model classes of different complexity can add further insights for interpretation.

Open Issues: Measures of model complexity allow quantifying the trade-off between complexity and performance and to automatically optimize for multiple objectives beyond performance. Some steps have been made towards quantifying model complexity, such as using functional decomposition and quantifying the complexity of the components [82] or measuring the stability of predictions [92].

However, further research is required, as there is no single perfect definition of interpretability, but rather multiple depending on the context [30,95].

5 Ignoring Feature Dependence

5.1 Interpretation with Extrapolation

Pitfall: When features are dependent, perturbation-based IML methods such as PFI, PDP, LIME, and Shapley values extrapolate in areas where the model was trained with little or no training data, which can cause misleading interpretations [55]. This is especially true if the ML model relies on feature interactions [45] – which is often the case. Perturbations produce artificial data points that are used for model predictions, which in turn are aggregated to produce global or local interpretations [100]. Feature values can be perturbed by replacing original values with values from an equidistant grid of that feature, with permuted or randomly subsampled values [19], or with quantiles. We highlight two major issues: First, if features are dependent, all three perturbation approaches produce unrealistic data points, i.e. the new data points are located outside of the multivariate joint distribution of the data (see Figure 4). Second, even if features are independent, using an equidistant grid can produce unrealistic values for the feature of interest. Consider a feature that follows a skewed distribution with outliers. An equidistant grid would generate many values between outliers and non-outliers. In contrast to the grid-based approach, the other two approaches maintain the marginal distribution of the feature of interest.

Both issues can result in misleading interpretations (illustrative examples are given in [55,84]), since the model is evaluated in areas of the feature space with few or no observed real data points, where model uncertainty can be expected to be very high. This issue is aggravated if interpretation methods integrate over such points with the same weight and confidence as for much more realistic samples with high model confidence.

Solution: Before applying interpretation methods, practitioners should check for dependencies between features in the data, e.g. via descriptive statistics or measures of dependence (see Section 5.2). When it is unavoidable to include dependent features in the model (which is usually the case in ML scenarios), additional information regarding the strength and shape of the dependence structure should be provided. Sometimes, alternative interpretation methods can be used as a workaround or to provide additional information. Accumulated local effect plots (ALE) [3] can be applied when features are dependent, but can produce non-intuitive effect plots for simple linear models with interactions [45]. For other methods such as the PFI, conditional variants exist [17,84,107]. In the case of LIME, it was suggested to focus in sampling on realistic (i.e. close to the data manifold) [97] and relevant areas (e.g. close to the decision boundary) [69]. Note, however, that conditional interpretations are often different and should not be used as a substitute for unconditional interpretations (see Section 5.3).

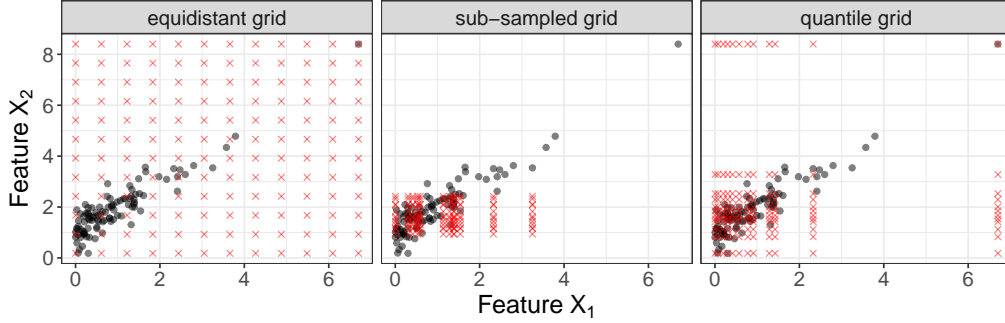


Fig. 4. Interpretation with extrapolation. Illustration of artificial data points generated by three different perturbation approaches. The black dots refer to observed data points and the red crosses to the artificial data points.

Furthermore, dependent features should not be interpreted separately but rather jointly. This can be achieved by visualizing e.g. a 2-dimensional ALE plot of two dependent features, which, admittedly, only works for very low-dimensional combinations. Especially in high-dimensional settings where dependent features can be grouped in a meaningful way, grouped interpretation methods might be more reasonable (see Section 9.1).

We recommend using quantiles or randomly subsampled values over equidistant grids. By default, many implementations of interpretability methods use an equidistant grid to perturb feature values [41,81,89], although some also allow using user-defined values.

Open Issues: A comprehensive comparison of strategies addressing extrapolation and how they affect an interpretation method is currently missing. This also includes studying interpretation methods and their conditional variants when they are applied to data with different dependence structures.

5.2 Confusing Linear Correlation with General Dependence

Pitfall: Features with a Pearson correlation coefficient (PCC) close to zero can still be dependent and cause misleading model interpretations (see Figure 5). While independence between two features implies that the PCC is zero, the converse is generally false. The PCC, which is often used to analyze dependence, only tracks linear correlations and has other shortcomings such as sensitivity to outliers [113]. Any type of dependence between features can have a strong impact on the interpretation of the results of IML methods (see Section 5.1). Thus, knowledge about the (possibly non-linear) dependencies between features is crucial for an informed use of IML methods.

Solution: Low-dimensional data can be visualized to detect dependence (e.g. scatter plots) [80]. For high-dimensional data, several other measures of depen-

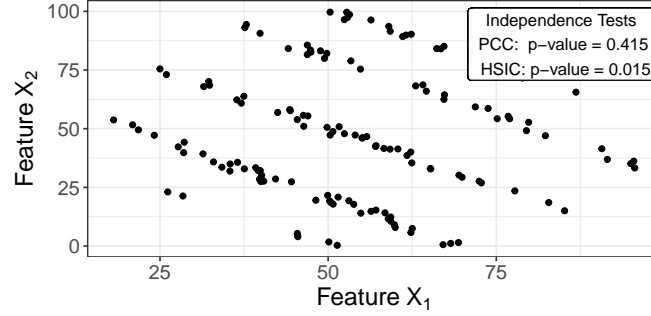


Fig. 5. Confusing linear correlation with dependence. Highly dependent features X_1 and X_2 that have a correlation close to zero. A test (H_0 : Features are independent) using Pearson correlation is not significant, but for HSIC, the H_0 -hypothesis gets rejected. Data from [80].

dence in addition to PCC can be used. If dependence is monotonic, Spearman’s rank correlation coefficient [72] can be a simple, robust alternative to PCC. For categorical or mixed features, separate dependence measures have been proposed, such as Kendall’s rank correlation coefficient for ordinal features, or the phi coefficient and Goodman & Kruskal’s lambda for nominal features [59].

Studying non-linear dependencies is more difficult since a vast variety of possible associations have to be checked. Nevertheless, several non-linear association measures with sound statistical properties exist. Kernel-based measures, such as kernel canonical correlation analysis (KCCA) [6] or the Hilbert-Schmidt independence criterion (HSIC) [44], are commonly used. They have a solid theoretical foundation, are computationally feasible, and robust [113]. In addition, there are information-theoretical measures, such as (conditional) mutual information [24] or the maximal information coefficient (MIC) [93], that can however be difficult to estimate [116,9]. Other important measures are e.g. the distance correlation [111], the randomized dependence coefficient (RDC) [74], or the alternating conditional expectations (ACE) algorithm [14]. In addition to using PCC, we recommend using at least one measure that detects non-linear dependencies (e.g. HSIC).

5.3 Misunderstanding Conditional Interpretation

Pitfall: Conditional variants of interpretation techniques avoid extrapolation but require a different interpretation. Interpretation methods that perturb features independently of others will extrapolate under dependent features but provide insight into the model’s mechanism [56,61]. Therefore, these methods are said to be true to the model but not true to the data [21].

For feature effect methods such as the PDP, the plot can be interpreted as the isolated, average effect the feature has on the prediction. For the PFI, the importance can be interpreted as the drop in performance when the feature’s information is “destroyed” (by perturbing it). Marginal SHAP value functions [78] quantify a feature’s contribution to a specific prediction, and marginal SAGE

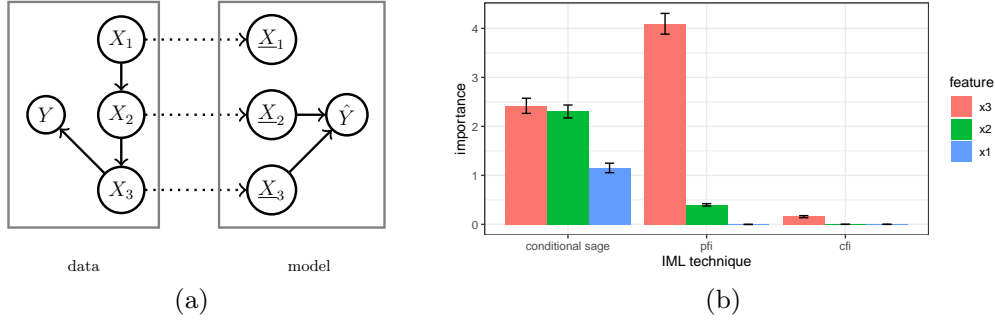


Fig. 6. Misunderstanding conditional interpretation. A linear model was fitted on the data-generating process modeled using a linear Gaussian structural causal model. The entailed directed acyclic graph is depicted on the left. For illustrative purposes, the original model coefficients were updated such that not only feature X_3 , but also feature X_2 is used by the model. PFI on test data considers both X_3 and X_2 to be relevant. In contrast, conditional feature importance variants either only consider X_3 to be relevant (CFI) or consider all features to be relevant (conditional SAGE value function).

value functions [25] quantify a feature’s contribution to the overall prediction performance. All the aforementioned methods extrapolate under dependent features (see also Section 5.1), but satisfy sensitivity, i.e. are zero if a feature is not used by the model [56,25,61,110].

Conditional variants of these interpretation methods do not replace feature values independently of other features, but in such a way that they conform to the conditional distribution. This changes the interpretation as the effects of all dependent features become entangled. Depending on the method, conditional sampling leads to a more or less restrictive notion of relevance.

For example, for dependent features, the Conditional Feature Importance (CFI) [17,117,84,107] answers the question: “How much does the model performance drop if we permute a feature, *but given that we know the values of the other features?*” [107,63,84].⁸ Two highly dependent features might be individually important (based on the unconditional PFI), but have a very low conditional importance score because the information of one feature is contained in the other and vice versa.

In contrast, the conditional variant of PDP, called marginal plot or M-plot [3], violates sensitivity, i.e. may even show an effect for features that are not used by the model. This is because for M-plots, the feature of interest is not sampled conditionally on the remaining features, but rather the remaining features are sampled conditionally on the feature of interest. As a consequence, the distribution of dependent covariates varies with the value of the feature of interest. Similarly, conditional SAGE and conditional SHAP value functions sample the remaining

⁸ While for CFI the conditional independence of the feature of interest X_j with the target Y given the remaining features X_{-j} ($Y \perp X_j | X_{-j}$) is already a sufficient condition for zero importance, the corresponding PFI may still be nonzero [63].

features conditional on the feature of interest and therefore violate sensitivity [56,109,25,61].

We demonstrate the difference between PFI, CFI, and conditional SAGE value functions on a simulated example (Figure 6) where the data-generating mechanism is known. While PFI only considers features to be relevant if they are actually used by the model, SAGE value functions may also consider a feature to be important that is not directly used by the model if it contains information that the model exploits. CFI only considers a feature to be relevant if it is both mechanistically used by the model and contributes unique information about Y .

Solution: When features are highly dependent and conditional effects and importance scores are used, the practitioner must be aware of the distinct interpretation. Recent work formalizes the implications of marginal and conditional interpretation techniques [21,56,63,61,25]. While marginal methods provide insight into the model’s mechanism but are not true to the data, their conditional variants are not true to the model but provide insight into the associations in the data.

If joint insight into model and data is required, designated methods must be used. ALE plots [3] provide interval-wise unconditional interpretations that are true to the data. They have been criticized to produce non-intuitive results for certain data-generating mechanisms [45]. Molnar et al. [84] propose a subgroup-based conditional sampling technique that allows for group-wise marginal interpretations that are true to model and data and that can be applied to feature importance and feature effects methods such as conditional PDPs and CFI. For feature importance, the DEDACT framework [61] allows to decompose conditional importance measures such as SAGE value functions into their marginal contributions and vice versa, thereby allowing global insight into both: the sources of prediction-relevant information in the data as well as into the feature pathways by which the information enters the model.

Open Issues: The quality of conditional IML techniques depends on the goodness of the conditional sampler. Especially in continuous, high-dimensional settings, conditional sampling is challenging. More research on the robustness of interpretation techniques regarding the quality of the sample is required.

6 Misleading Interpretations due to Feature Interactions

6.1 Misleading Feature Effects due to Aggregation

Pitfall: Global interpretation methods, such as PDP or ALE plots, visualize the average effect of a feature on a model’s prediction. However, they can produce misleading interpretations when features interact. Figure 7 A and B show the marginal effect of features X_1 and X_2 of the below-stated simulation example. While the PDP of the non-interacting feature X_1 seems to capture the true underlying effect of X_1 on the target quite well (A), the global aggregated effect of the interacting feature X_2 (B) shows almost no influence on the target, although an effect is clearly there by construction.

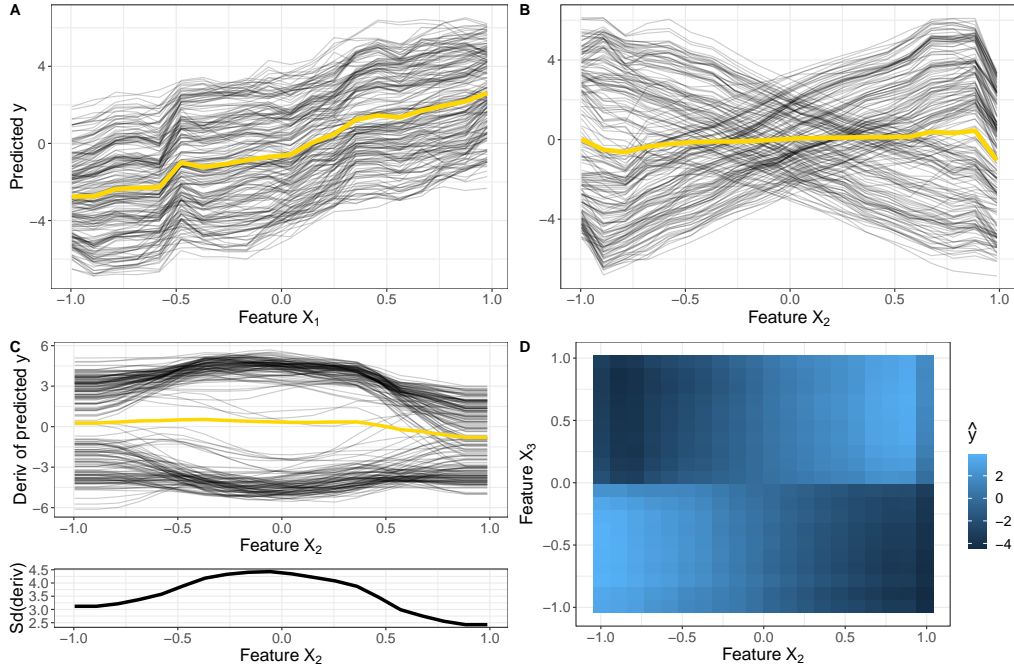


Fig. 7. Misleading effect due to interactions. Simulation example with interactions: $Y = 3X_1 - 6X_2 + 12X_2\mathbb{1}_{(X_3 \geq 0)} + \epsilon$ with $X_1, X_2, X_3 \stackrel{i.i.d.}{\sim} U[-1, 1]$ and $\epsilon \stackrel{i.i.d.}{\sim} N(0, 0.3)$. A random forest with 500 trees is fitted on 1000 observations. Effects are calculated on 200 randomly sampled (training) observations. **A, B:** PDP (yellow) and ICE curves of X_1 and X_2 ; **C:** Derivative ICE curves and their standard deviation of X_2 ; **D:** 2-dimensional PDP of X_2 and X_3 .

Solution: For the PDP, we recommend to additionally consider the corresponding ICE curves [38]. While PDP and ALE average out interaction effects, ICE curves directly show the heterogeneity between individual predictions. Figure 7 A illustrates that the individual marginal effect curves all follow an upward trend with only small variations. Hence, by aggregating these ICE curves to a global marginal effect curve such as the PDP, we do not lose much information. However, when the regarded feature interacts with other features, such as feature X_2 with feature X_3 in this example, then marginal effect curves of different observations might not show similar effects on the target. Hence, ICE curves become very heterogeneous, as shown in Figure 7 B. In this case, the influence of feature X_2 is not well represented by the global average marginal effect. Particularly for continuous interactions where ICE curves start at different intercepts, we recommend the use of derivative or centered ICE curves, which eliminate differences in intercepts and leave only differences due to interactions [38]. Derivative ICE curves also point out the regions of highest interaction with other features. For example, Figure 7 C indicates that predictions for X_2 taking values close to 0 strongly depend on other features' values. While these methods show that interactions are present with regards to the feature of interest but do not reveal other features with which

it interacts, the 2-dimensional PDP or ALE plot are options to visualize 2-way interaction effects. The 2-dimensional PDP in Figure 7 D shows that predictions with regards to feature X_2 highly depend on the feature values of feature X_3 .

Other methods that aim to gain more insights into these visualizations are based on clustering homogeneous ICE curves, such as visual interaction effects (VINE) [16] or [122]. As an example, in Figure 7 B, it would be more meaningful to average over the upward and downward proceeding ICE curves separately and hence show that the average influence of feature X_2 on the target depends on an interacting feature (here: X_3). Work by Zon et al. [125] followed a similar idea by proposing an interactive visualization tool to group Shapley values with regards to interacting features that need to be defined by the user.

Open Issues: The introduced visualization methods are not able to illustrate the type of the underlying interaction and most of them are also not applicable to higher-order interactions.

6.2 Failing to Separate Main from Interaction Effects

Pitfall: Many interpretation methods that quantify a feature’s importance or effect cannot separate an interaction from main effects. The PFI, for example, includes both the importance of a feature and the importance of all its interactions with other features [19]. Also local explanation methods such as LIME and Shapley values only provide additive explanations without separation of main effects and interactions [40].

Solution: Functional ANOVA introduced by [53] is probably the most popular approach to decompose the joint distribution into main and interaction effects. Using the same idea, the H-Statistic [35] quantifies the interaction strength between two features or between one feature and all others by decomposing the 2-dimensional PDP into its univariate components. The H-Statistic is based on the fact that, in the case of non-interacting features, the 2-dimensional partial dependence function equals the sum of the two underlying univariate partial dependence functions. Another similar interaction score based on partial dependencies is defined by [42]. Instead of decomposing the partial dependence function, [87] uses the predictive performance to measure interaction strength. Based on Shapley values, Lundberg et al. [77] proposed SHAP interaction values, and Casalicchio et al. [19] proposed a fair attribution of the importance of interactions to the individual features.

Furthermore, Hooker [54] considers dependent features and decomposes the predictions in main and interaction effects. A way to identify higher-order interactions is shown in [53].

Open Issues: Most methods that quantify interactions are not able to identify higher-order interactions and interactions of dependent features. Furthermore, the

presented solutions usually lack automatic detection and ranking of all interactions of a model. Identifying a suitable shape or form of the modeled interaction is not straightforward as interactions can be very different and complex, e.g., they can be a simple product of features (multiplicative interaction) or can have a complex joint non-linear effect such as smooth spline surface.

7 Ignoring Model and Approximation Uncertainty

Pitfall: Many interpretation methods only provide a mean estimate but do not quantify uncertainty. Both the model training and the computation of interpretation are subject to uncertainty. The model is trained on (random) data, and therefore should be regarded as a random variable. Similarly, LIME’s surrogate model relies on perturbed and reweighted samples of the data to approximate the prediction function locally [94]. Other interpretation methods are often defined in terms of expectations over the data (PFI, PDP, Shapley values, ...), but are approximated using Monte Carlo integration. Ignoring uncertainty can result in the interpretation of noise and non-robust results. The true effect of a feature may be flat, but – purely by chance, especially on smaller datasets – the Shapley value might show an effect. This effect could cancel out once averaged over multiple model fits. Figure 8 shows that a single PDP (first plot) can be

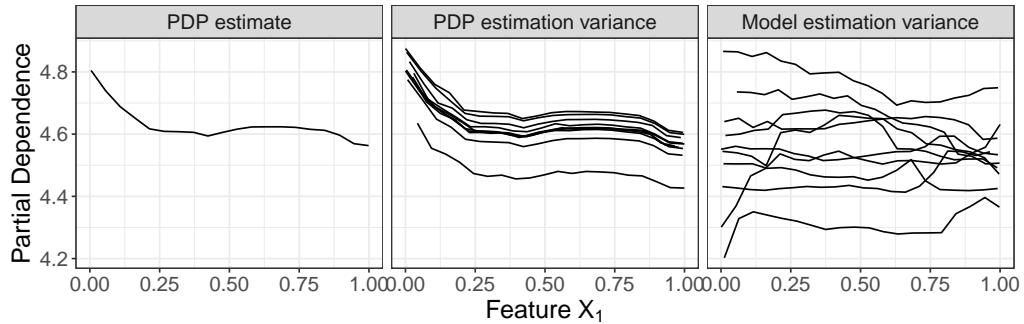


Fig. 8. Ignoring model and approximation uncertainty. PDP for X_1 with $Y = 0 \cdot X_1 + \sum_{j=2}^{10} X_j + \epsilon_i$ with $X_1, \dots, X_{10} \sim U[0, 1]$ and $\epsilon_i \sim N(0, 0.9)$. **Left:** PDP for X_1 of a random forest trained on 100 data points. **Middle:** Multiple PDPs (10x) for the model from left plots, but with different samples (each $n=100$) for PDP estimation. **Right:** Repeated (10x) data samples of $n=100$ and newly fitted random forest.

misleading because it does not show the variance due to PDP estimation (second plot) and model fitting (third plot). If we are not interested in learning about a specific model, but rather about the relationship between feature X_1 and the target (in this case), we should consider the model variance.

Solution: By repeatedly computing PDP and PFI with a given model, but with different permutations or bootstrap samples, the uncertainty of the estimate

can be quantified, for example in the form of confidence intervals. For PFI, frameworks for confidence intervals and hypothesis tests exist [117,2], but they assume a fixed model. If the practitioner wants to condition the analysis on the modeling process and capture the process’ variance instead of conditioning on a fixed model, PDP and PFI should be computed on multiple model fits [83].

Open Issues: While Moosbauer et al. [85] derived confidence bands for PDPs for probabilistic ML models that cover the model’s uncertainty, a general model-agnostic uncertainty measure for feature effect methods such as ALE [3] and PDP [36] has (to the best of our knowledge) not been introduced yet.

8 Ignoring the Rashomon Effect

Pitfall: Sometimes different models explain the data-generating process equally well, but contradict each other. This phenomenon is called the Rashomon effect, named after the movie “Rashomon” from the year 1950. Breiman formalized it for predictive models in 2001 [13]: Different prediction models might perform equally well (Rashomon set), but construct the prediction function in a different way (e.g. relying on different features). This can result in conflicting interpretations and conclusions about the data. Even small differences in the training data can cause one model to be preferred over another.

For example, Dong and Rudin [29] identified a Rashomon set of equally well performing models for the COMPAS dataset. They showed that the models differed greatly in the importance they put on certain features. Specifically, if criminal history was identified as less important, race was more important and vice versa. Cherry-picking one model and its underlying explanation might not be sufficient to draw conclusions about the data-generating process. As Hancox-Li [48] states “just because race happens to be an unimportant variable in that one explanation does not mean that it is objectively an unimportant variable”.

The Rashomon effect can also occur at the level of the interpretation method itself. Differing hyperparameters or interpretation goals can be one reason (see Section 2). But even if the hyperparameters are fixed, we could still obtain contradicting explanations by an interpretation method, e.g., due to a different data sample or initial seed.

A concrete example of the Rashomon effect is counterfactual explanations. Different counterfactuals may all alter the prediction in the desired way, but point to different feature changes required for that change. If a person is deemed uncreditworthy, one corresponding counterfactual explaining this decision may point to a scenario in which the person had asked for a shorter loan duration and amount, while another counterfactual may point to a scenario in which the person had a higher income and more stable job. Focusing on only one counterfactual explanation in such cases strongly limits the possible epistemic access.

Solution: If multiple, equally good models exist, their interpretations should be compared. Variable importance clouds [29] is a method for exploring variable

importance scores for equally good models within one model class. If the interpretations are in conflict, conclusions must be drawn carefully. Domain experts or further constraints (e.g. fairness or sparsity) could help to pick a suitable model. Semenova et al. [102] also hypothesized that a large Rashomon set could contain simpler or more interpretable models, which should be preferred according to Section 4.

In the case of counterfactual explanations, multiple, equally good explanations exist. Here, methods that return a set of explanations rather than a single one should be used – for example, the method by Dandl et al. [26] or Mothilal et al. [86].

Open Issues: Numerous very different counterfactual explanations are overwhelming for users. Methods for aggregating or combining explanations are still a matter of future research.

9 Failure to Scale to High-Dimensional Settings

9.1 Human-Intelligibility of High-Dimensional IML Output

Pitfall: Applying IML methods naively to high-dimensional datasets (e.g. visualizing feature effects or computing importance scores on feature level) leads to an overwhelming and high-dimensional IML output, which impedes human analysis. Especially interpretation methods that are based on visualizations make it difficult for practitioners in high-dimensional settings to focus on the most important insights.

Solution: A natural approach is to reduce the dimensionality before applying any IML methods. Whether this facilitates understanding or not depends on the possible semantic interpretability of the resulting, reduced feature space – as features can either be selected or dimensionality can be reduced by linear or non-linear transformations. Assuming that users would like to interpret in the original feature space, many feature selection techniques can be used [46], resulting in much sparser and consequently easier to interpret models. Wrapper selection approaches are model-agnostic and algorithms like greedy forward selection or subset selection procedures [60,5], which start from an empty model and iteratively add relevant (subsets of) features if needed, even allow to measure the relevance of features for predictive performance. An alternative is to directly use models that implicitly perform feature selection such as LASSO [112] or component-wise boosting [99] as they can produce sparse models with fewer features. In the case of LIME or other interpretation methods based on surrogate models, the aforementioned techniques could be applied to the surrogate model.

When features can be meaningfully grouped in a data-driven or knowledge-driven way [51], applying IML methods directly to grouped features instead of single features is usually more time-efficient to compute and often leads to more appropriate interpretations. Examples where features can naturally be

grouped include the grouping of sensor data [20], time-lagged features [75], or one-hot-encoded categorical features and interaction terms [43]. Before a model is fitted, groupings could already be exploited for dimensionality reduction, for example by selecting groups of features by the group LASSO [121].

For model interpretation, various papers extended feature importance methods from single features to groups of features [5,43,114,119]. In the case of grouped PFI, this means that we perturb the entire group of features at once and measure the performance drop compared to the unperturbed dataset. Compared to standard PFI, the grouped PFI does not break the association to the other features of the group, but to features of other groups and the target. This is especially useful when features within the same group are highly correlated (e.g. time-lagged features), but between-group dependencies are rather low. Hence, this might also be a possible solution for the extrapolation pitfall described in Section 5.1.

We consider the PhoneStudy in [106] as an illustration. The PhoneStudy dataset contains 1821 features to analyze the link between human behavior based on smartphone data and participants’ personalities. Interpreting the results in this use case seems to be challenging since features were dependent and single feature effects were either small or non-linear [106]. The features have been grouped in behavior-specific categories such as app-usage, music consumption, or overall phone usage. Au et al. [5] calculated various grouped importance scores on the feature groups to measure their influence on a specific personality trait (e.g. conscientiousness). Furthermore, the authors applied a greedy forward subset selection procedure via repeated subsampling on the feature groups and showed that combining app-usage features and overall phone usage features were most of the times sufficient for the given prediction task.

Open Issues: The quality of a grouping-based interpretation strongly depends on the human intelligibility and meaningfulness of the grouping. If the grouping structure is not naturally given, then data-driven methods can be used. However, if feature groups are not meaningful (e.g. if they cannot be described by a super-feature such as app-usage), then subsequent interpretations of these groups are purposeless. One solution could be to combine feature selection strategies with interpretation methods. For example, LIME’s surrogate model could be a LASSO model. However, beyond surrogate models, the integration of feature selection strategies remains an open issue that requires further research.

Existing research on grouped interpretation methods mainly focused on quantifying grouped feature importance, but the question of “how a group of features influences a model’s prediction” remains almost unanswered. Only recently, [5,15,101] attempted to answer this question by using dimension-reduction techniques (such as PCA) before applying the interpretation method. However, this is also a matter of further research.

9.2 Computational Effort

Pitfall: Some interpretation methods do not scale linearly with the number of features. For example, for the computation of exact Shapley values the number

of possible coalitions [25,78], or for a (full) functional ANOVA decomposition the number of components (main effects plus all interactions) scales with $\mathcal{O}(2^p)$ [54].⁹

Solution: For the functional ANOVA, a common solution is to keep the analysis to the main effects and selected 2-way interactions (similar for PDP and ALE). Interesting 2-way interactions can be selected by another method such as the H-statistic [35]. However, the selection of 2-way interactions requires additional computational effort. Interaction strength usually decreases quickly with increasing interaction size, and one should only consider d -way interactions when all their $(d-1)$ -way interactions were significant [53]. For Shapley-based methods, an efficient approximation exists that is based on randomly sampling and evaluating feature orderings until the estimates converge. The variance of the estimates reduces in $\mathcal{O}(\frac{1}{m})$, where m is the number of evaluated orderings [25,78].

9.3 Ignoring Multiple Comparison Problem

Pitfall: Simultaneously testing the importance of multiple features will result in false-positive interpretations if the multiple comparisons problem (MCP) is ignored. The MCP is well known in significance tests for linear models and exists similarly in testing for feature importance in ML. For example, suppose we simultaneously test the importance of 50 features (with the H_0 -hypothesis of zero importance) at the significance level $\alpha = 0.05$. Even if all features are unimportant, the probability of observing that at least one feature is significantly important is $1 - \mathbb{P}(\text{'no feature important'}) = 1 - (1 - 0.05)^{50} \approx 0.923$. Multiple comparisons become even more problematic the higher the dimension of the dataset.

Solution: Methods such as Model-X knockoffs [17] directly control for the false discovery rate (FDR). For all other methods that provide p-values or confidence intervals, such as PIMP (Permutation IMPortance) [2], which is a testing approach for PFI, MCP is often ignored in practice to the best of our knowledge, with some exceptions [105,117]. One of the most popular MCP adjustment methods is the Bonferroni correction [31], which rejects a null hypothesis if its p-value is smaller than α/p , with p as the number of tests. It has the disadvantage that it increases the probability of false negatives [90]. Since MCP is well known in statistics, we refer the practitioner to [28] for an overview and discussion of alternative adjustment methods, such as the Bonferroni-Holm method [52].

As an example, in Figure 9 we compare the number of features with significant importance measured by PIMP once with and once without Bonferroni-adjusted significance levels ($\alpha = 0.05$ vs. $\alpha = 0.05/p$). Without correcting for multiple comparisons, the number of features mistakenly evaluated as important grows

⁹ Similar to the PDP or ALE plots, the functional ANOVA components describe individual feature effects and interactions.

considerably with increasing dimension, whereas Bonferroni correction results in only a modest increase.

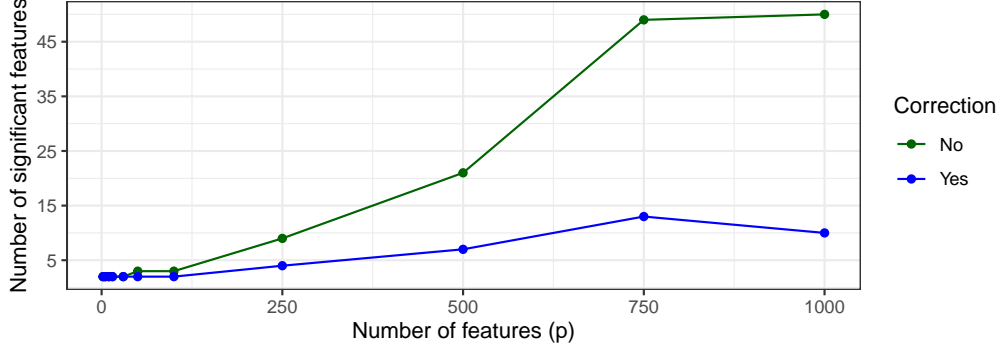


Fig. 9. Failure to scale to high-dimensional settings. Comparison of the number of features with significant importance - once with and once without Bonferroni-corrected significance levels for a varying number of added noise variables. Datasets were sampled from $Y = 2X_1 + 2X_2^2 + \epsilon$ with $X_1, X_2, \epsilon \sim N(0, 1)$. $X_3, X_4, \dots, X_p \sim N(0, 1)$ are additional noise variables with p ranging between 2 and 1000. For each p , we sampled two datasets from this data-generating process – one to train a random forest with 500 trees on and one to test whether feature importances differed from 0 using PIMP. In all experiments, X_1 and X_2 were correctly identified as important.

10 Unjustified Causal Interpretation

Pitfall: Practitioners are often interested in causal insights into the underlying data-generating mechanisms, which IML methods do not generally provide. Common causal questions include the identification of causes and effects, predicting the effects of interventions, and answering counterfactual questions [88]. For example, a medical researcher might want to identify risk factors or predict average and individual treatment effects [66]. In search of answers, a researcher can therefore be tempted to interpret the result of IML methods from a causal perspective.

However, a causal interpretation of predictive models is often not possible. Standard supervised ML models are not designed to model causal relationships but to merely exploit associations. A model may therefore rely on causes and effects of the target variable as well as on variables that help to reconstruct unobserved influences on Y , e.g. causes of effects [118]. Consequently, the question of whether a variable is relevant to a predictive model (indicated e.g. by $\text{PFI} > 0$) does not directly indicate whether a variable is a cause, an effect, or does not stand in any causal relation to the target variable. Furthermore, even if a model would rely solely on direct causes for the prediction, the causal structure between features must be taken into account. Intervening on a variable in the real world may

affect not only Y but also other variables in the feature set. Without assumptions about the underlying causal structure, IML methods cannot account for these adaptations and guide action [58,62].

As an example, we constructed a dataset by sampling from a structural causal model (SCM), for which the corresponding causal graph is depicted in Figure 10. All relationships are linear Gaussian with variance 1 and coefficients 1. For a linear model fitted on the dataset, all features were considered to be relevant based on the model coefficients ($\hat{y} = 0.329x_1 + 0.323x_2 - 0.327x_3 + 0.342x_4 + 0.334x_5$, $R^2 = 0.943$), although x_3 , x_4 and x_5 do not cause Y .

Solution: The practitioner must carefully assess whether sufficient assumptions can be made about the underlying data-generating process, the learned model, and the interpretation technique. If these assumptions are met, a causal interpretation may be possible. The PDP between a feature and the target can be interpreted as the respective average causal effect if the model performs well and the set of remaining variables is a valid adjustment set [123]. When it is known whether a model is deployed in a causal or anti-causal setting – i.e. whether the model attempts to predict an effect from its causes or the other way round – a partial identification of the causal roles based on feature relevance is possible (under strong and non-testable assumptions) [118]. Designated tools and approaches are available for causal discovery and inference [91].

Open Issues: The challenge of causal discovery and inference remains an open key issue in the field of ML. Careful research is required to make explicit under which assumptions what insight about the underlying data-generating mechanism can be gained by interpreting an ML model.

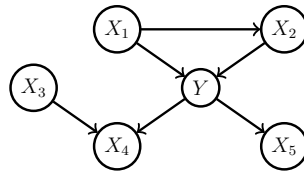


Fig. 10. Causal graph

11 Discussion

In this paper, we have reviewed numerous pitfalls of local and global model-agnostic interpretation techniques, e.g. in the case of bad model generalization, dependent features, interactions between features, or causal interpretations. We have not attempted to provide an exhaustive list of all potential pitfalls in ML model interpretation, but have instead focused on common pitfalls that apply to various model-agnostic IML methods and pose a particularly high risk.

We have omitted pitfalls that are more specific to one IML method type: For local methods, the vague notions of neighborhood and distance can lead to misinterpretations [69,68], and common distance metrics (such as the Euclidean distance) are prone to the curse of dimensionality [1]; Surrogate methods such as LIME may not be entirely faithful to the original model they replace in interpretation. Moreover, we have not addressed pitfalls associated with certain data types (like the definition of superpixels in image data [98]), nor those related to human cognitive biases (e.g. the illusion of model understanding [22]).

Many pitfalls in the paper are strongly linked with axioms that encode desiderata of model interpretation. For example, pitfall 5.3 (misunderstanding conditional interpretations) is related to violations of sensitivity [56,110]. As such, axioms can help to make the strengths and limitations of methods explicit. Therefore, we encourage an axiomatic evaluation of interpretation methods.

We hope to promote a more cautious approach when interpreting ML models in practice, to point practitioners to already (partially) available solutions, and to stimulate further research on these issues. The stakes are high: ML algorithms are increasingly used for socially relevant decisions, and model interpretations play an important role in every empirical science. Therefore, we believe that users can benefit from concrete guidance on properties, dangers, and problems of IML techniques – especially as the field is advancing at high speed. We need to strive towards a recommended, well-understood set of tools, which will in turn require much more careful research. This especially concerns the meta-issues of comparisons of IML techniques, IML diagnostic tools to warn against misleading interpretations, and tools for analyzing multiple dependent or interacting features.

References

1. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: Van den Bussche, J., Vianu, V. (eds.) *Database Theory — ICDT 2001*. pp. 420–434. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
2. Altmann, A., Tološi, L., Sander, O., Lengauer, T.: Permutation importance: a corrected feature importance measure. *Bioinformatics* **26**(10), 1340–1347 (2010). <https://doi.org/10.1093/bioinformatics/btq134>
3. Apley, D.W., Zhu, J.: Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **82**(4), 1059–1086 (2020). <https://doi.org/10.1111/rssb.12377>
4. Arlot, S., Celisse, A.: A survey of cross-validation procedures for model selection. *Statist. Surv.* **4**, 40–79 (2010). <https://doi.org/10.1214/09-SS054>
5. Au, Q., Herbinger, J., Stachl, C., Bischl, B., Casalicchio, G.: Grouped feature importance and combined features effect plot. *arXiv preprint arXiv:2104.11688* (2021)
6. Bach, F.R., Jordan, M.I.: Kernel independent component analysis. *Journal of Machine Learning Research* **3**(Jul), 1–48 (2002)
7. Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J.: Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society* **54**(6), 627–635 (2003). <https://doi.org/10.1057/palgrave.jors.2601545>

8. Bansal, N., Agarwal, C., Nguyen, A.: Sam: The sensitivity of attribution methods to hyperparameters. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 8673–8683 (2020)
9. Belghazi, M.I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., Hjelm, D.: Mutual information neural estimation. In: *International Conference on Machine Learning*. pp. 531–540 (2018). https://doi.org/10.1007/978-3-642-02962-2_49
10. Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A.L., Deng, D., Lindauer, M.: Hyperparameter optimization: Foundations, algorithms, best practices and open challenges. *arXiv preprint arXiv:2107.05847* (2021)
11. Bischl, B., Mersmann, O., Trautmann, H., Weihs, C.: Resampling methods for meta-model validation with recommendations for evolutionary computation. *Evolutionary Computation* **20**(2), 249–275 (2012). https://doi.org/10.1162/EVCO_a.00069
12. Breiman, L.: Random forests. *Machine learning* **45**(1), 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
13. Breiman, L.: Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Statistical Science* **16**(3), 199 – 231 (2001). <https://doi.org/10.1214/ss/1009213726>
14. Breiman, L., Friedman, J.H.: Estimating optimal transformations for multiple regression and correlation. *Journal of the American statistical Association* **80**(391), 580–598 (1985). <https://doi.org/10.1080/01621459.1985.10478157>
15. Brenning, A.: Transforming feature space to interpret machine learning models. *arXiv:2104.04295* (2021)
16. Britton, M.: Vine: Visualizing statistical interactions in black box models. *arXiv preprint arXiv:1904.00561* (2019)
17. Candès, E., Fan, Y., Janson, L., Lv, J.: Panning for gold: ‘model-x’ knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **80**(3), 551–577 (2018). <https://doi.org/10.1111/rssb.12265>
18. Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N.: Intelligent models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. pp. 1721–1730 (2015). <https://doi.org/10.1145/2783258.2788613>
19. Casalicchio, G., Molnar, C., Bischl, B.: Visualizing the feature importance for black box models. In: Berlingerio, M., Bonchi, F., Gärtner, T., Hurley, N., Ifrim, G. (eds.) *Machine Learning and Knowledge Discovery in Databases*. pp. 655–670. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-10925-7_40
20. Chakraborty, D., Pal, N.R.: Selecting useful groups of features in a connectionist framework. *IEEE Transactions on Neural Networks* **19**(3), 381–396 (mar 2008). <https://doi.org/10.1109/TNN.2007.910730>
21. Chen, H., Janizek, J.D., Lundberg, S., Lee, S.I.: True to the model or true to the data? *arXiv preprint arXiv:2006.16234* (2020)
22. Chromik, M., Eiband, M., Buchner, F., Krüger, A., Butz, A.: I think i get your point, ai! the illusion of explanatory depth in explainable ai. In: *26th International Conference on Intelligent User Interfaces*. p. 307–317. IUI ’21, Association for Computing Machinery, New York, NY, USA (2021). <https://doi.org/10.1145/3397481.3450644>
23. Claeskens, G., Hjort, N.L., et al.: Model selection and model averaging. Cambridge Books (2008). <https://doi.org/10.1017/CBO9780511790485>

24. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons (2012). <https://doi.org/10.1002/047174882X>
25. Covert, I., Lundberg, S.M., Lee, S.I.: Understanding global feature contributions with additive importance measures. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 17212–17223. Curran Associates, Inc. (2020)
26. Dandl, S., Molnar, C., Binder, M., Bischl, B.: Multi-objective counterfactual explanations. In: International Conference on Parallel Problem Solving from Nature. pp. 448–469. Springer (2020). https://doi.org/10.1007/978-3-030-58112-1_31
27. Das, A., Rad, P.: Opportunities and challenges in explainable artificial intelligence (xai): A survey. arXiv preprint arXiv:2006.11371 (2020)
28. Dickhaus, T.: Simultaneous Statistical Inference. Springer-Verlag Berlin Heidelberg (2014). <https://doi.org/10.1007/978-3-642-45182-9>
29. Dong, J., Rudin, C.: Exploring the cloud of variable importance for the set of all good models. Nature Machine Intelligence **2**(12), 810–824 (Dec 2020). <https://doi.org/10.1038/s42256-020-00264-0>
30. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608 (2017)
31. Dunn, O.J.: Multiple comparisons among means. Journal of the American Statistical Association **56**(293), 52–64 (1961). <https://doi.org/10.1080/01621459.1961.10482090>
32. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems. Journal of Machine Learning Research **15**(1), 3133–3181 (2014). <https://doi.org/10.5555/2627435.2697065>
33. Fisher, A., Rudin, C., Dominici, F.: All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. Journal of Machine Learning Research **20**(177), 1–81 (2019)
34. Freiesleben, T.: Counterfactual explanations & adversarial examples—common grounds, essential differences, and potential transfers. arXiv preprint arXiv:2009.05487 (2020)
35. Friedman, J.H., Popescu, B.E.: Predictive learning via rule ensembles. Annals of Applied Statistics **2**(3), 916–954 (09 2008). <https://doi.org/10.1214/07-AOAS148>
36. Friedman, J.H., et al.: Multivariate adaptive regression splines. The Annals of Statistics **19**(1), 1–67 (1991). <https://doi.org/10.1214/aos/1176347963>
37. Garreau, D., von Luxburg, U.: Looking deeper into tabular lime. arXiv preprint arXiv:2008.11092 (2020)
38. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. Journal of Computational and Graphical Statistics **24**(1), 44–65 (2015). <https://doi.org/10.1080/10618600.2014.907095>
39. Good, P.I., Hardin, J.W.: Common errors in statistics (and how to avoid them). John Wiley & Sons (2012). <https://doi.org/10.1002/9781118360125>
40. Gosiewska, A., Biecek, P.: Do not trust additive explanations. arXiv preprint arXiv:1903.11420 (2019)
41. Greenwell, B.M.: pdp: An R package for constructing partial dependence plots. The R Journal **9**(1), 421–436 (2017). <https://doi.org/10.32614/RJ-2017-016>
42. Greenwell, B.M., Boehmke, B.C., McCarthy, A.J.: A simple and effective model-based variable importance measure. arXiv:1805.04755 (2018)

43. Gregorutti, B., Michel, B., Saint-Pierre, P.: Grouped variable importance with random forests and application to multiple functional data analysis. *Computational Statistics & Data Analysis* **90**, 15–35 (oct 2015). <https://doi.org/10.1016/j.csda.2015.04.002>
44. Gretton, A., Bousquet, O., Smola, A., Schölkopf, B.: Measuring statistical dependence with hilbert-schmidt norms. In: *International Conference on Algorithmic Learning Theory*. pp. 63–77. Springer (2005). https://doi.org/10.1007/11564089_7
45. Grömping, U.: Model-agnostic effects plots for interpreting machine learning models. *Reports in Mathematics, Physics and Chemistry* **Report 1/2020** (2020)
46. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of machine learning research* **3**(Mar), 1157–1182 (2003)
47. Hall, P.: On the art and science of machine learning explanations. *arXiv preprint arXiv:1810.02909* (2018)
48. Hancox-Li, L.: Robustness in machine learning explanations: Does it matter? In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. p. 640–647. FAT* '20, Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3351095.3372836>
49. Hand, D.J.: Classifier Technology and the Illusion of Progress. *Statistical Science* **21**(1), 1 – 14 (2006). <https://doi.org/10.1214/0883423060000000060>
50. Hastie, T., Tibshirani, R.: Generalized Additive Models. *Statistical Science* **1**(3), 297 – 310 (1986). <https://doi.org/10.1214/ss/1177013604>
51. He, Z., Yu, W.: Stable Feature Selection for Biomarker Discovery, vol. 34 (4), pp. 215–225. *Computational Biology and Chemistry* (aug 2010). <https://doi.org/10.1016/j.compbiolchem.2010.07.002>
52. Holm, S.: A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* **6**(2), 65–70 (1979)
53. Hooker, G.: Discovering additive structure in black box functions. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. p. 575–580. KDD '04, Association for Computing Machinery, New York, NY, USA (2004). <https://doi.org/10.1145/1014052.1014122>
54. Hooker, G.: Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics* **16**(3), 709–732 (2007). <https://doi.org/10.1198/106186007X237892>
55. Hooker, G., Mentch, L.: Please stop permuting features: An explanation and alternatives. *arXiv preprint arXiv:1905.03151* (2019)
56. Janzing, D., Minorics, L., Blöbaum, P.: Feature relevance quantification in explainable ai: A causality problem. *arXiv preprint arXiv:1910.13413* (2019)
57. Kadir, T., Brady, M.: Saliency, scale and image description. *International Journal of Computer Vision* **45**(2), 83–105 (Nov 2001). <https://doi.org/10.1023/A:1012460413855>
58. Karimi, A.H., Schölkopf, B., Valera, I.: Algorithmic Recourse: from Counterfactual Explanations to Interventions. *arXiv: 2002.06278* (2020)
59. Khamis, H.: Measures of association: how to choose? *Journal of Diagnostic Medical Sonography* **24**(3), 155–162 (2008). <https://doi.org/10.1177/8756479308317006>
60. Kohavi, R., John, G.H.: Wrappers for feature subset selection. *Artificial intelligence* **97**(1-2), 273–324 (1997)
61. König, G., Freiesleben, T., Bischl, B., Casalicchio, G., Grosse-Wentrup, M.: Decomposition of global feature importance into direct and associative components (dedact). *arXiv preprint arXiv:2106.08086* (2021)
62. König, G., Freiesleben, T., Grosse-Wentrup, M.: A causal perspective on meaningful and robust algorithmic recourse. *arXiv preprint arXiv:2107.07853* (2021)

63. König, G., Molnar, C., Bischl, B., Grosse-Wentrup, M.: Relative feature importance. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 9318–9325. IEEE (2021). <https://doi.org/10.1109/ICPR48806.2021.9413090>
64. Krishnan, M.: Against interpretability: a critical examination of the interpretability problem in machine learning. *Philosophy & Technology* (08 2019). <https://doi.org/10.1007/s13347-019-00372-9>
65. Kuhle, S., Maguire, B., Zhang, H., Hamilton, D., Allen, A.C., Joseph, K., Allen, V.M.: Comparison of logistic regression with machine learning methods for the prediction of fetal growth abnormalities: a retrospective cohort study. *BMC Pregnancy and Childbirth* **18**(1), 1–9 (2018). <https://doi.org/10.1186/s12884-018-1971-2>
66. König, G., Grosse-Wentrup, M.: A Causal Perspective on Challenges for AI in Precision Medicine (2019)
67. Lang, M., Binder, M., Richter, J., Schratz, P., Pfisterer, F., Coors, S., Au, Q., Casalicchio, G., Kotthoff, L., Bischl, B.: mlr3: A modern object-oriented machine learning framework in R. *Journal of Open Source Software* (dec 2019). <https://doi.org/10.21105/joss.01903>
68. Laugel, T., Lesot, M.J., Marsala, C., Renard, X., Detyniecki, M.: The dangers of post-hoc interpretability: Unjustified counterfactual explanations. In: Twenty-Eighth International Joint Conference on Artificial Intelligence {IJCAI-19}. pp. 2801–2807. International Joint Conferences on Artificial Intelligence Organization (2019)
69. Laugel, T., Renard, X., Lesot, M.J., Marsala, C., Detyniecki, M.: Defining locality for surrogates in post-hoc interpretability. *arXiv preprint arXiv:1806.07498* (2018)
70. Lauritsen, S.M., Kristensen, M., Olsen, M.V., Larsen, M.S., Lauritsen, K.M., Jørgensen, M.J., Lange, J., Thiesson, B.: Explainable artificial intelligence model to predict acute critical illness from electronic health records. *Nature communications* **11**(1), 1–11 (2020). <https://doi.org/10.1038/s41467-020-17431-x>
71. Lessmann, S., Baesens, B., Seow, H.V., Thomas, L.C.: Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research* **247**(1), 124–136 (2015). <https://doi.org/10.1016/j.ejor.2015.05.030>
72. Liebetrau, A.: Measures of Association. No. Bd. 32;Bd. 1983 in 07, SAGE Publications (1983)
73. Lipton, Z.C.: The mythos of model interpretability. *Queue* **16**(3), 31–57 (2018). <https://doi.org/10.1145/3236386.3241340>
74. Lopez-Paz, D., Hennig, P., Schölkopf, B.: The randomized dependence coefficient. In: *Advances in Neural Information Processing Systems*. pp. 1–9 (2013). <https://doi.org/10.5555/2999611.2999612>
75. Lozano, A.C., Abe, N., Liu, Y., Rosset, S.: Grouped graphical granger modeling for gene expression regulatory networks discovery. *Bioinformatics* **25**(12), i110–i118 (05 2009). <https://doi.org/10.1093/bioinformatics/btp199>
76. Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.I.: From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence* **2**(1), 56–67 (2020). <https://doi.org/10.1038/s42256-019-0138-9>
77. Lundberg, S.M., Erion, G.G., Lee, S.I.: Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888* (2018)
78. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: *NIPS*, vol. 30, pp. 4765–4774. Curran Associates, Inc. (2017). <https://doi.org/10.5555/3295222.3295230>

79. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one* **13**(3) (2018). <https://doi.org/10.1371/journal.pone.0194889>
80. Matejka, J., Fitzmaurice, G.: Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. pp. 1290–1294 (2017). <https://doi.org/10.1145/3025453.3025912>
81. Molnar, C., Casalicchio, G., Bischl, B.: iml: An R package for interpretable machine learning. *Journal of Open Source Software* **3**(26), 786 (2018). <https://doi.org/10.21105/joss.00786>
82. Molnar, C., Casalicchio, G., Bischl, B.: Quantifying model complexity via functional decomposition for better post-hoc interpretability. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. pp. 193–204. Springer (2019). https://doi.org/10.1007/978-3-030-43823-4_17
83. Molnar, C., Freiesleben, T., König, G., Casalicchio, G., Wright, M.N., Bischl, B.: Relating the partial dependence plot and permutation feature importance to the data generating process. *arXiv preprint arXiv:2109.01433* (2021)
84. Molnar, C., König, G., Bischl, B., Casalicchio, G.: Model-agnostic feature importance and effects with dependent features—a conditional subgroup approach. *arXiv preprint arXiv:2006.04628* (2020)
85. Moosbauer, J., Herbringer, J., Casalicchio, G., Lindauer, M., Bischl, B.: Towards explaining hyperparameter optimization via partial dependence plots. *8th ICML Workshop on Automated Machine Learning (AutoML)* (2020)
86. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. *CoRR* **abs/1905.07697** (2019), <http://arxiv.org/abs/1905.07697>
87. Oh, S.: Feature interaction in terms of prediction performance. *Applied Sciences* **9**(23) (2019). <https://doi.org/10.3390/app9235191>
88. Pearl, J., Mackenzie, D.: *The ladder of causation. The book of why: the new science of cause and effect*. New York (NY): Basic Books pp. 23–52 (2018). <https://doi.org/10.1080/14697688.2019.1655928>
89. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011). <https://doi.org/10.5555/1953048.2078195>
90. Perneger, T.V.: What’s wrong with bonferroni adjustments. *BMJ* **316**(7139), 1236–1238 (1998). <https://doi.org/10.1136/bmj.316.7139.1236>
91. Peters, J., Janzing, D., Scholkopf, B.: *Elements of Causal Inference - Foundations and Learning Algorithms*. The MIT Press (2017). <https://doi.org/doi/10.5555/3202377>
92. Philipp, M., Rusch, T., Hornik, K., Strobl, C.: Measuring the stability of results from supervised statistical learning. *Journal of Computational and Graphical Statistics* **27**(4), 685–700 (2018). <https://doi.org/10.1080/10618600.2018.1473779>
93. Reshef, D.N., Reshef, Y.A., Finucane, H.K., Grossman, S.R., McVean, G., Turnbaugh, P.J., Lander, E.S., Mitzenmacher, M., Sabeti, P.C.: Detecting novel associations in large data sets. *Science* **334**(6062), 1518–1524 (2011). <https://doi.org/10.1126/science.1205438>
94. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should i trust you?: Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD*

- International Conference on Knowledge Discovery and Data Mining. pp. 1135–1144. ACM (2016). <https://doi.org/10.1145/2939672.2939778>
95. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* **1**(5), 206–215 (2019). <https://doi.org/10.1038/s42256-019-0048-x>
96. Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., Zhong, C.: Interpretable machine learning: Fundamental principles and 10 grand challenges. *arXiv preprint arXiv:2103.11251* (2021)
97. Saito, S., Chua, E., Capel, N., Hu, R.: Improving lime robustness with smarter locality sampling. *arXiv preprint arXiv:2006.12302* (2020)
98. Schallner, L., Rabold, J., Scholz, O., Schmid, U.: Effect of superpixel aggregation on explanations in lime—a case study with biological data. *arXiv preprint arXiv:1910.07856* (2019)
99. Schmid, M., Hothorn, T.: Boosting additive models using component-wise p-splines. *Computational Statistics & Data Analysis* **53**(2), 298–311 (2008). <https://doi.org/10.1016/j.csda.2008.09.009>
100. Scholbeck, C.A., Molnar, C., Heumann, C., Bischl, B., Casalicchio, G.: Sampling, intervention, prediction, aggregation: A generalized framework for model-agnostic interpretations. *Communications in Computer and Information Science* p. 205–216 (2020). https://doi.org/10.1007/978-3-030-43823-4_18
101. Seedorff, N., Brown, G.: totalvis: A principal components approach to visualizing total effects in black box models. *SN Computer Science* **2**(3), 1–12 (2021). <https://doi.org/10.1007/s42979-021-00560-5>
102. Semenova, L., Rudin, C., Parr, R.: A study in rashomon curves and volumes: A new perspective on generalization and model simplicity in machine learning. *arXiv preprint arXiv:1908.01755* (2021)
103. Shalev-Shwartz, S., Ben-David, S.: Understanding machine learning: From theory to algorithms. Cambridge university press (2014)
104. Simon, R.: Resampling strategies for model assessment and selection. In: *Fundamentals of data mining in genomics and proteomics*, pp. 173–186. Springer (2007). https://doi.org/10.1007/978-0-387-47509-7_8
105. Stachl, C., Au, Q., Schoedel, R., Buschek, D., Völkel, S., Schuwerk, T., Oldemeier, M., Ullmann, T., Hussmann, H., Bischl, B., et al.: Behavioral patterns in smartphone usage predict big five personality traits. *PsyArXiv* (2019). <https://doi.org/10.31234/osf.io/ks4vd>
106. Stachl, C., Au, Q., Schoedel, R., Gosling, S.D., Harari, G.M., Buschek, D., Theres, S., Völkel, Schuwerk, T., Oldemeier, M., Ullmann, T., Hussmann, H., Bischl, B., Bühner, M.: Predicting personality from patterns of behavior collected with smartphones. *Proceedings of the National Academy of Sciences* (2020). <https://doi.org/10.1073/pnas.1920484117>
107. Strobl, C., Boulesteix, A.L., Kneib, T., Augustin, T., Zeileis, A.: Conditional variable importance for random forests. *BMC bioinformatics* **9**(1), 307 (2008). <https://doi.org/10.1186/1471-2105-9-307>
108. Štrumbelj, E., Kononenko, I.: Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems* **41**(3), 647–665 (2014). <https://doi.org/10.1007/s10115-013-0679-x>
109. Sundararajan, M., Najmi, A.: The many shapley values for model explanation. *arXiv preprint arXiv:1908.08474* (2019)
110. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: *International Conference on Machine Learning*. pp. 3319–3328. PMLR (2017)

111. Székely, G.J., Rizzo, M.L., Bakirov, N.K., et al.: Measuring and testing dependence by correlation of distances. *The Annals of Statistics* **35**(6), 2769–2794 (2007). <https://doi.org/10.1214/009053607000000505>
112. Tibshirani, R.: Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* **58**(1), 267–288 (1996). <https://doi.org/10.1111/j.1467-9868.2011.00771.x>
113. Tjøstheim, D., Otneim, H., Støve, B.: Statistical dependence: Beyond pearson’s p . *arXiv preprint arXiv:1809.10455* (2018)
114. Valentin, S., Harkotte, M., Popov, T.: Interpreting neural decoding models using grouped model reliance. *PLOS Computational Biology* **16**(1), e1007148 (2020). <https://doi.org/10.1371/journal.pcbi.1007148>
115. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.* **31**, 841 (2017). <https://doi.org/10.2139/ssrn.3063289>
116. Walters-Williams, J., Li, Y.: Estimation of mutual information: A survey. In: *International Conference on Rough Sets and Knowledge Technology*. pp. 389–396. Springer (2009). https://doi.org/10.1007/978-3-642-02962-2_49
117. Watson, D.S., Wright, M.N.: Testing Conditional Independence in Supervised Learning Algorithms. *arXiv preprint arXiv:1901.09917* (2019)
118. Weichwald, S., Meyer, T., Özdenizci, O., Schölkopf, B., Ball, T., Grosse-Wentrup, M.: Causal interpretation rules for encoding and decoding models in neuroimaging. *NeuroImage* **110**, 48–59 (2015). <https://doi.org/10.1016/j.neuroimage.2015.01.036>
119. Williamson, B.D., Gilbert, P.B., Simon, N.R., Carone, M.: A unified approach for inference on algorithm-agnostic variable importance. *arXiv:2004.03683* (2020)
120. Wu, J., Roy, J., Stewart, W.F.: Prediction modeling using ehr data: challenges, strategies, and a comparison of machine learning approaches. *Medical Care* pp. S106–S113 (2010). <https://doi.org/10.1097/MLR.0b013e3181de9e17>
121. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**(1), 49–67 (2006). <https://doi.org/10.1111/j.1467-9868.2005.00532.x>
122. Zhang, X., Wang, Y., Li, Z.: Interpreting the black box of supervised learning models: Visualizing the impacts of features on prediction. *Applied Intelligence* pp. 1–15 (2021). <https://doi.org/10.1007/s10489-021-02255-z>
123. Zhao, Q., Hastie, T.: Causal interpretations of black-box models. *Journal of Business & Economic Statistics* pp. 1–10 (2019). <https://doi.org/10.1080/07350015.2019.1624293>
124. Zhao, X., Lovreglio, R., Nilsson, D.: Modelling and interpreting pre-evacuation decision-making using machine learning. *Automation in Construction* **113**, 103140 (2020). <https://doi.org/10.1016/j.autcon.2020.103140>
125. van der Zon, S., Duivesteijn, W., van Ipenburg, W., Veldsink, J., Pechenizkiy, M.: ICIE 1.0: A novel tool for interactive contextual interaction explanations. In: Alzate, C., Monreale, A., Bioglio, L., Bitetta, V., Bordino, I., Caldarelli, G., Ferretti, A., Guidotti, R., Gullo, F., Pascolutti, S., Pensa, R.G., Robardet, C., Squartini, T. (eds.) *ECML PKDD 2018 Workshops - MIDAS 2018 and PAP 2018*, Dublin, Ireland, September 10–14, 2018, Proceedings. *Lecture Notes in Computer Science*, vol. 11054, pp. 81–94. Springer (2018). https://doi.org/10.1007/978-3-030-13463-1_6

6. iml: an R package for Interpretable Machine Learning

Contributing article:

Molnar, C., Casalicchio, G., and Bischl, B. (2018). iml: an R package for Interpretable Machine Learning. *Journal of Open Source Software*, 3, 786.

Copyright information:

Paper: Creative Commons Attribution 4.0 International License (CC BY NC 4.0).

Software: MIT License

Author contributions:

Christoph Molnar designed and implemented the iml R package and wrote the paper. Bernd Bischl and Giuseppe Casalicchio provided feedback for the design and scope of the software and proofread the paper.

iml: An R package for Interpretable Machine Learning

Christoph Molnar¹, Giuseppe Casalicchio¹, and Bernd Bischl¹

¹ Department of Statistics, LMU Munich

DOI: [10.21105/joss.00786](https://doi.org/10.21105/joss.00786)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Submitted: 19 June 2018

Published: 27 June 2018

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

Complex, non-parametric models, which are typically used in machine learning, have proven to be successful in many prediction tasks. But these models usually operate as black boxes: While they are good at predicting, they are often not interpretable. Many inherently interpretable models have been suggested, which come at the cost of losing predictive power. Another option is to apply interpretability methods to a black box model after model training. Given the velocity of research on new machine learning models, it is preferable to have model-agnostic tools which can be applied to a random forest as well as to a neural network. Tools for model-agnostic interpretability methods should improve the adoption of machine learning.

iml is an R package (R Core Team 2016) that offers a general toolbox for making machine learning models interpretable. It implements many model-agnostic methods which work for any type of machine learning model. The package covers following methods:

- Partial dependence plots (Friedman 2001): Visualizing the learned relationship between features and predictions.
- Individual conditional expectation (Goldstein et al. 2015): Visualizing the learned relationship between features and predictions for individual instances of the data.
- Feature importance (Fisher, Rudin, and Dominici 2018): Scoring features by contribution to predictive performance.
- Global surrogate tree: Approximating the black box model with an interpretable decision tree.
- Local surrogate models (Ribeiro, Singh, and Guestrin 2016): Explaining single predictions by approximating the black box model locally with an interpretable model.
- Shapley value (Strumbelj et al. 2014): Explaining single predictions by fairly distributing the predicted value among the features.
- Interaction effects (Friedman, Popescu, and others 2008): Measuring how strongly features interact with each other in the black box model.

iml was designed to provide a class-based and user-friendly way to make black box machine learning models interpretable. Internally, the implemented methods inherit from the same parent class and share a common framework for the computation. Many of the methods are already implemented in other packages (e.g. (Greenwell 2017), (Goldstein et al. 2015), (Pedersen and Benesty 2017)), but the **iml** package implements all of the methods in one place, uses the same syntax and offers consistent functionality and outputs. **iml** can be used with models from the R machine learning libraries **mlr** and **caret**, but the package is flexible enough to work with models from other packages as well. Similar projects are the R package **DALEX** (Biecek 2018) and the Python package **Skater** (Choudhary, Kramer, and team 2018). The difference to **iml** is that the other two projects do not implement the methods themselves, but depend on other packages. **DALEX** focuses more on model comparison, and **Skater** additionally includes interpretable models and has less model-agnostic interpretability methods compared to **iml**.

Acknowledgements

This work is funded by the Bavarian State Ministry of Science and the Arts in the framework of the Centre Digitisation.Bavaria (ZD.B)

References

- Biecek, Przemyslaw. 2018. *DALEX: Descriptive mACHine Learning Explanations*. <https://CRAN.R-project.org/package=DALEX>.
- Choudhary, Pramit, Aaron Kramer, and contributors datascience.com team. 2018. "Skater: Model Interpretation Library." <https://doi.org/10.5281/zenodo.1198885>.
- Fisher, Aaron, Cynthia Rudin, and Francesca Dominici. 2018. "Model Class Reliance: Variable Importance Measures for any Machine Learning Model Class, from the "Rashomon" Perspective." <http://arxiv.org/abs/1801.01489>.
- Friedman, Jerome H. 2001. "Greedy Function Approximation: A Gradient Boosting Machine." *Annals of Statistics*. JSTOR, 1189–1232. <https://doi.org/10.1214/aos/1013203451>.
- Friedman, Jerome H, Bogdan E Popescu, and others. 2008. "Predictive Learning via Rule Ensembles." *The Annals of Applied Statistics* 2 (3). Institute of Mathematical Statistics:916–54. <https://doi.org/10.1214/07-AOAS148>.
- Goldstein, Alex, Adam Kapelner, Justin Bleich, and Emil Pitkin. 2015. "Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation." *Journal of Computational and Graphical Statistics* 24 (1):44–65. <https://doi.org/10.1080/10618600.2014.907095>.
- Greenwell, Brandon M. 2017. "Pdp: An R Package for Constructing Partial Dependence Plots." *The R Journal* 9 (1):421–36. <https://journal.r-project.org/archive/2017/RJ-2017-016/index.html>.
- Pedersen, Thomas Lin, and Michaël Benesty. 2017. *Lime: Local Interpretable Model-Agnostic Explanations*. <https://CRAN.R-project.org/package=lime>.
- R Core Team. 2016. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.
- Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?: Explaining the Predictions of Any Classifier." In *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 1135–44. ACM. <https://doi.org/10.1145/2939672.2939778>.
- Strumbelj, Erik, Igor Kononenko, Erik Štrumbelj, and Igor Kononenko. 2014. "Explaining prediction models and individual predictions with feature contributions." *Knowledge and Information Systems* 41 (3):647–65. <https://doi.org/10.1007/s10115-013-0679-x>.

7. Sampling, Intervention, Prediction, Aggregation: A Generalized Framework for Model-Agnostic Interpretations

Contributing article:

Scholbeck, C., Molnar, C., Heumann, C., Bischl, B., and Casalicchio, G. (2019). Sampling, Intervention, Prediction, Aggregation: A Generalized Framework for Model-Agnostic Interpretations. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 205-216).

Copyright information:

Springer Nature Switzerland AG 2020

Author contributions:

Christoph Molnar developed the initial idea for the SIPA framework, after which also the *iml* R package was implemented. Christian Scholbeck further developed the idea and wrote the paper. All authors added input, suggested modifications and proofread the paper.

Sampling, Intervention, Prediction, Aggregation: A Generalized Framework for Model-Agnostic Interpretations

Christian A. Scholbeck (✉), Christoph Molnar, Christian Heumann, Bernd
Bischl, Giuseppe Casalicchio

Department of Statistics, Ludwig-Maximilians-University Munich,
Ludwigstr. 33, 80539 Munich, Germany
`christian.scholbeck@stat.uni-muenchen.de`

Abstract. Model-agnostic interpretation techniques allow us to explain the behavior of any predictive model. Due to different notations and terminology, it is difficult to see how they are related. A unified view on these methods has been missing. We present the generalized SIPA (sampling, intervention, prediction, aggregation) framework of work stages for model-agnostic interpretations and demonstrate how several prominent methods for feature effects can be embedded into the proposed framework. Furthermore, we extend the framework to feature importance computations by pointing out how variance-based and performance-based importance measures are based on the same work stages. The SIPA framework reduces the diverse set of model-agnostic techniques to a single methodology and establishes a common terminology to discuss them in future work.

Keywords: Interpretable Machine Learning | Explainable AI | Feature Effect | Feature Importance | Model-Agnostic | Partial Dependence

1 Introduction and Related Work

There has been an ongoing debate about the lacking interpretability of machine learning (ML) models. As a result, researchers have put in great efforts developing techniques to create insights into the workings of predictive black box models. Interpretable machine learning [15] serves as an umbrella term for all interpretation methods in ML. We make the following distinctions:

- (i) *Feature effects or feature importance:* Feature effects indicate the direction and magnitude of change in predicted outcome due to changes in feature values. Prominent methods include the individual conditional expectation (ICE) [9] and partial dependence (PD) [8], accumulated local effects (ALE) [1], Shapley values [19] and local interpretable model-agnostic explanations (LIME) [17]. The feature importance measures the importance of a feature to the model behavior. This includes variance-based measures like the feature importance ranking measure (FIRM) [10], [20]

and performance-based measures like the permutation feature importance (PFI) [7], individual conditional importance (ICI) and partial importance (PI) curves [4], as well as the Shapley feature importance (SFIMP) [4]. Input gradients were proposed by [11] as a model-agnostic tool for both effects and importance that essentially equals marginal effects (ME) [12], which have a long tradition in statistics. They also define an average input gradient which corresponds to the average marginal effect (AME).

- (ii) *Intrinsic or post-hoc interpretability*: Linear models (LM), generalized linear models (GLM), classification and regression trees (CART) or rule lists [18] are examples for intrinsically interpretable models, while random forests (RF), support vector machines (SVM), neural networks (NN) or gradient boosting (GB) models can only be interpreted post-hoc. Here, the interpretation process is detached from and takes place after the model fitting process, e.g., with the ICE, PD or ALEs.
- (iii) *Model-specific or model-agnostic interpretations*: Interpreting model coefficients of GLMs or deriving a decision rule from a classification tree is a model-specific interpretation. Model-agnostic methods such as the ICE, PD or ALEs can be applied to any model.
- (iv) *Local or global explanations*: Local explanations like the ICE evaluate the model behavior when predicting for one specific observation. Global explanations like the PD interpret the model for the entire input space. Furthermore, it is possible to explain model predictions for a group of observations, e.g., on intervals. In a lot of cases, local and global explanations can be transformed into one another via (dis-)aggregation, e.g., the ICE and PD.

Motivation: Research in model-agnostic interpretation methods is complicated by the variety of different notations and terminology. It turns out that deconstructing model-agnostic techniques into sequential work stages reveals striking similarities. In [14] the authors propose a unified framework for model-agnostic interpretations called SHapley Additive exPlanations (SHAP). However, the SHAP framework only considers Shapley values or variations thereof (KernelSHAP and TreeSHAP). The motivation for this research paper is to provide a more extensive survey on model-agnostic interpretation methods, to reveal similarities in their computation and to establish a framework with common terminology that is applicable to all model-agnostic techniques.

Contributions: In Section 4 we present the generalized SIPA (sampling, intervention, prediction, aggregation) framework of work stages for model-agnostic techniques. We proceed to demonstrate how several methods to estimate feature effects (MEs, ICE and PD, ALEs, Shapley values and LIME) can be embedded into the proposed framework. Furthermore, in Section 5 and 6 we extend the framework to feature importance computations by pointing out how variance-based (FIRM) and performance-based (ICI and PI, PFI and SFIMP) importance measures are based on the same work stages. By using a unified notation, we also reveal how the methods are related.

2 Notation and Preliminaries

Consider a p -dimensional feature space $\mathcal{X}_P = \mathcal{X}_1 \times \dots \times \mathcal{X}_p$ with the feature index set $P = \{1, \dots, p\}$ and a target space \mathcal{Y} . We assume an unknown functional relationship f between \mathcal{X}_P and \mathcal{Y} . A supervised learning model \hat{f} attempts to learn this relationship from an i.i.d. training sample that was drawn from the unknown probability distribution \mathcal{F} with the sample space $\mathcal{X}_P \times \mathcal{Y}$. The random variables generated from the feature space are denoted by $X = (X_1, \dots, X_p)$. The random variable generated from the target space is denoted by Y . We draw an i.i.d. sample of test data \mathcal{D} with n observations from \mathcal{F} . The vector $x^{(i)} = (x_1^{(i)}, \dots, x_p^{(i)}) \in \mathcal{X}_P$ corresponds to the feature values of the i -th observation that are associated with the observed target value $y^{(i)} \in \mathcal{Y}$. The vector $x_j = (x_j^{(1)}, \dots, x_j^{(n)})^\top$ represents the realizations of X_j . The generalization error $GE(\hat{f}, \mathcal{F})$ corresponds to the expectation of the loss function \mathcal{L} on unseen test data from \mathcal{F} and is estimated by the average loss on \mathcal{D} .

$$GE(\hat{f}, \mathcal{F}) = \mathbb{E} \left[\mathcal{L}(\hat{f}(X_1, \dots, X_p), Y) \right]$$

$$\widehat{GE}(\hat{f}, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}(\hat{f}(x_1^{(i)}, \dots, x_p^{(i)}), y^{(i)})$$

A variety of model-agnostic techniques is used to interpret the prediction function $\hat{f}(x_1, \dots, x_p)$ with the sample of test data \mathcal{D} . We estimate the effects and importance of a subset of features with index set S ($S \subseteq P$). A vector of feature values $x \in \mathcal{X}_P$ can be partitioned into two vectors x_S and $x_{\setminus S}$ so that $x = (x_S, x_{\setminus S})$. The corresponding random variables are denoted by X_S and $X_{\setminus S}$. Given a model-agnostic technique where S only contains a single element, the corresponding notations are $X_j, X_{\setminus j}$ and $x_j, x_{\setminus j}$.

The partial derivative of the trained model $\hat{f}(x_j, x_{\setminus j})$ with respect to x_j is numerically approximated with a symmetric difference quotient [12].

$$\lim_{h \rightarrow 0} \frac{\hat{f}(x_j + h, x_{\setminus j}) - \hat{f}(x_j, x_{\setminus j})}{h} \approx \frac{\hat{f}(x_j + h, x_{\setminus j}) - \hat{f}(x_j - h, x_{\setminus j})}{2h}, \quad h > 0$$

A term of the form $\hat{f}(x_j + h, x_{\setminus j}) - \hat{f}(x_j - h, x_{\setminus j})$ is called a finite difference (FD) of predictions with respect to x_j .

$$FD_{\hat{f}, j}(x_j, x_{\setminus j}) = \hat{f}(x_j + h, x_{\setminus j}) - \hat{f}(x_j - h, x_{\setminus j})$$

3 Feature Effects

Partial dependence (PD) and individual conditional expectation (ICE): First suggested by [8], the PD is defined as the dependence of the prediction function on x_S after all remaining features $X_{\setminus S}$ have been marginalized out [9]. The PD is estimated via Monte Carlo integration.

$$\begin{aligned}
PD_{\hat{f},S}(x_S) &= \mathbb{E}_{X_{\setminus S}} \left[\hat{f}(x_S, X_{\setminus S}) \right] = \int \hat{f}(x_S, X_{\setminus S}) d\mathcal{P}(X_{\setminus S}) \\
\widehat{PD}_{\hat{f},S}(x_S) &= \frac{1}{n} \sum_{i=1}^n \hat{f}(x_S, x_{\setminus S}^{(i)})
\end{aligned} \tag{1}$$

The PD is a useful feature effect measure when features are not interacting [8]. Otherwise it can obfuscate the relationships in the data [4]. In that case, the individual conditional expectation (ICE) can be used instead [9]. The i -th ICE corresponds to the expected value of the target for the i -th observation as a function of x_S , conditional on $x_{\setminus S}^{(i)}$.

$$\widehat{ICE}_{\hat{f},S}^{(i)}(x_S) = \hat{f}(x_S, x_{\setminus S}^{(i)})$$

The ICE disaggregates the global effect estimates of the PD to local effect estimates for single observations. Given $|S| = 1$, the ICE and PD are also referred to as ICE and PD curves. The ICE and PD suffer from extrapolation when features are correlated, because the permutations used to predict are located in regions without any training data [1].

Accumulated local effects (ALE): In [1] ALEs are presented as a feature effect measure for correlated features that does not extrapolate. The idea of ALEs is to take the integral with respect to X_j of the first derivative of the prediction function with respect to X_j . This creates an accumulated partial effect of X_j on the target variable while simultaneously removing additively linked effects of other features. The main advantage of not extrapolating stems from integrating with respect to the conditional distribution of $X_{\setminus j}$ on X_j instead of the marginal distribution of $X_{\setminus j}$ [1]. Let $z_{0,j}$ denote the minimum value of x_j . The first order ALE of the j -th feature at point x is defined as:

$$\begin{aligned}
ALE_{\hat{f},j}(x) &= \int_{z_{0,j}}^x \mathbb{E}_{X_{\setminus j}|X_j} \left[\frac{\partial \hat{f}(X_j, X_{\setminus j})}{\partial X_j} \Big| X_j = z_j \right] dz_j - constant \\
&= \int_{z_{0,j}}^x \left[\int \frac{\partial \hat{f}(z_j, X_{\setminus j})}{\partial z_j} d\mathcal{P}(X_{\setminus j}|z_j) \right] dz_j - constant
\end{aligned} \tag{2}$$

A constant is subtracted in order to center the plot. We estimate the first order ALE in three steps. First, we divide the value range of x_j into a set of intervals and compute a finite difference (FD) for each observation. For each i -th observation, $x_j^{(i)}$ is substituted by the corresponding right and left interval boundaries. Then the predictions with both substituted values are subtracted in order to receive an observation-wise FD. Second, we estimate local effects by averaging the FDs inside each interval. This replaces the inner integral in Eq. (2). Third, the accumulation of all local effects up to the point of interest replaces the outer integral in Eq. (2), i.e., the interval-wise average FDs are summed up.

The second order ALE is the bivariate extension of the first order ALE. It is important to note that first order effect estimates are subtracted from the second order estimates. In [1] the authors further lay out the computations necessary for higher order ALEs.

Marginal effects (ME): MEs are an established technique in statistics and often used to interpret non-linear functions of coefficients in GLMs like logistic regression. The ME corresponds to the first derivative of the prediction function with respect to a feature at specified values of the input space. It is estimated by computing an observation-wise FD. The average marginal effect (AME) is the average of all MEs that were estimated with observed feature values [2]. Although there is extensive literature on MEs, this concept was suggested by [11] as a novel method for ML and referred to as the input gradient. Derivatives are also often utilized as a feature importance metric.

Shapley value: Originating in coalitional game theory [19], the Shapley value is a local feature effect measure that is based on a set of desirable axioms. In coalitional games, a set of p players, denoted by P , play games and join coalitions. They are rewarded with a payout. The characteristic function $v : 2^P \rightarrow \mathbb{R}$ maps all player coalitions to their respective payouts [4]. The Shapley value is a player's average contribution to the payout, i.e., the marginal increase in payout for the coalition of players, averaged over all possible coalitions. For Shapley values as feature effects, predicting the target for a single observation corresponds to the game and a coalition of features represents the players. Shapley regression values were first developed for linear models with multicollinear features [13]. A model-agnostic Shapley value was first introduced in [19].

Consider the expected prediction for a single vector of feature values x , conditional on only knowing the values of features with indices in K ($K \subseteq P$), i.e., the features $X_{\setminus K}$ are marginalized out. This essentially equals a point (or a line, surface etc. depending on the power of K) on the PD from Eq. (1).

$$\mathbb{E}_{X_{\setminus K}} [\hat{f}(x_K, X_{\setminus K})] = \int \hat{f}(x_K, X_{\setminus K}) d\mathcal{P}(X_{\setminus K}) = \widehat{PD}_{\hat{f}, K}(x_K) \quad (3)$$

Eq. (3) is shifted by the mean prediction and used as a payout function $v_{PD}(x_K)$, so that an empty set of features ($K = \emptyset$) results in a payout of zero [4].

$$\begin{aligned} v_{PD}(x_K) &= \mathbb{E}_{X_{\setminus K}} [\hat{f}(x_K, X_{\setminus K})] - \mathbb{E}_{X_{K \cup (P \setminus K)}} [\hat{f}(X_K, X_{\setminus K})] \\ &= \widehat{PD}_{\hat{f}, K}(x_K) - \widehat{PD}_{\hat{f}, \emptyset}(x_{\emptyset}) \\ &= \widehat{PD}_{\hat{f}, K}(x_K) - \frac{1}{n} \sum_{i=1}^n \hat{f}(x_K^{(i)}, x_{\setminus K}^{(i)}) \end{aligned}$$

The marginal contribution $\Delta_j(x_K)$ of a feature value x_j joining the coalition of feature values x_K is:

$$\Delta_j(x_K) = v_{PD}(x_{K \cup \{j\}}) - v_{PD}(x_K) = \widehat{PD}_{\hat{f}, K \cup \{j\}}(x_{K \cup \{j\}}) - \widehat{PD}_{\hat{f}, K}(x_K)$$

The exact Shapley value of the j -th feature for a single vector of feature values x corresponds to:

$$\begin{aligned}
\widehat{Shapley}_{f,j} &= \sum_{K \subseteq P \setminus \{j\}} \frac{|K|!(|P| - |K| - 1)!}{|P|!} \Delta_j(x_K) \\
&= \sum_{K \subseteq P \setminus \{j\}} \frac{|K|!(|P| - |K| - 1)!}{|P|!} \left[\widehat{PD}_{\hat{f}, K \cup \{j\}}(x_{K \cup \{j\}}) - \widehat{PD}_{\hat{f}, K}(x_K) \right]
\end{aligned}$$

Shapley values are computationally expensive because the PD function has a complexity of $\mathcal{O}(N^2)$. Computations can be sped up by Monte Carlo sampling [19]. Furthermore, in [14] the authors propose a distinct variant to compute Shapley values called SHapley Additive exPlanations (SHAP).

Local interpretable model-agnostic explanations (LIME): In contrast to all previous techniques which are based on interpreting a single model, LIME [17] locally approximates the black box model with an intrinsically interpretable surrogate model. Given a single vector of feature values x , we first perturb x_j around a sufficiently close neighborhood while $x_{\setminus j}$ is kept constant. Then we predict with the perturbed feature values. The predictions are weighted by the proximity of the corresponding perturbed values to the original feature value. Finally, an intrinsically interpretable model is trained on the weighted predictions and interpreted instead.

4 Generalized Framework

Although the techniques presented in Section 3 are seemingly unrelated, they all work according to the exact same principle. Instead of trying to inspect the inner workings of a non-linear black box model, we evaluate its predictions when changing inputs. We can deconstruct model-agnostic techniques into a framework of four work stages: sampling, intervention, prediction, aggregation (SIPA). The software package `iml` [16] was inspired by the SIPA framework.

We first sample a subset (**sampling stage**) to reduce computational costs, e.g., we select a random set of available observations to evaluate as ICEs. In order to change the predictions made by the black box model, the data has to be manipulated. Feature values can be set to values from the observed marginal distributions (ICEs and PD or Shapley values), or to unobserved values (FD based methods such as MEs and ALEs). This crucial step is called the **intervention stage**. During the **prediction stage**, we predict on previously intervened data. This requires an already trained model, which is why model-agnostic techniques are always post-hoc. The predictions are further aggregated during the **aggregation stage**. Often, the predictions resulting from the prediction stage are local effect estimates, and the ones resulting from the aggregation stage are global effect estimates.

In Fig. 1, we demonstrate how all presented techniques for feature effects are based on the SIPA framework. Although LIME is a special case as it is based on training a local surrogate model, we argue that it is also based on the SIPA framework as training a surrogate model can be considered an aggregation of the training data to a function.

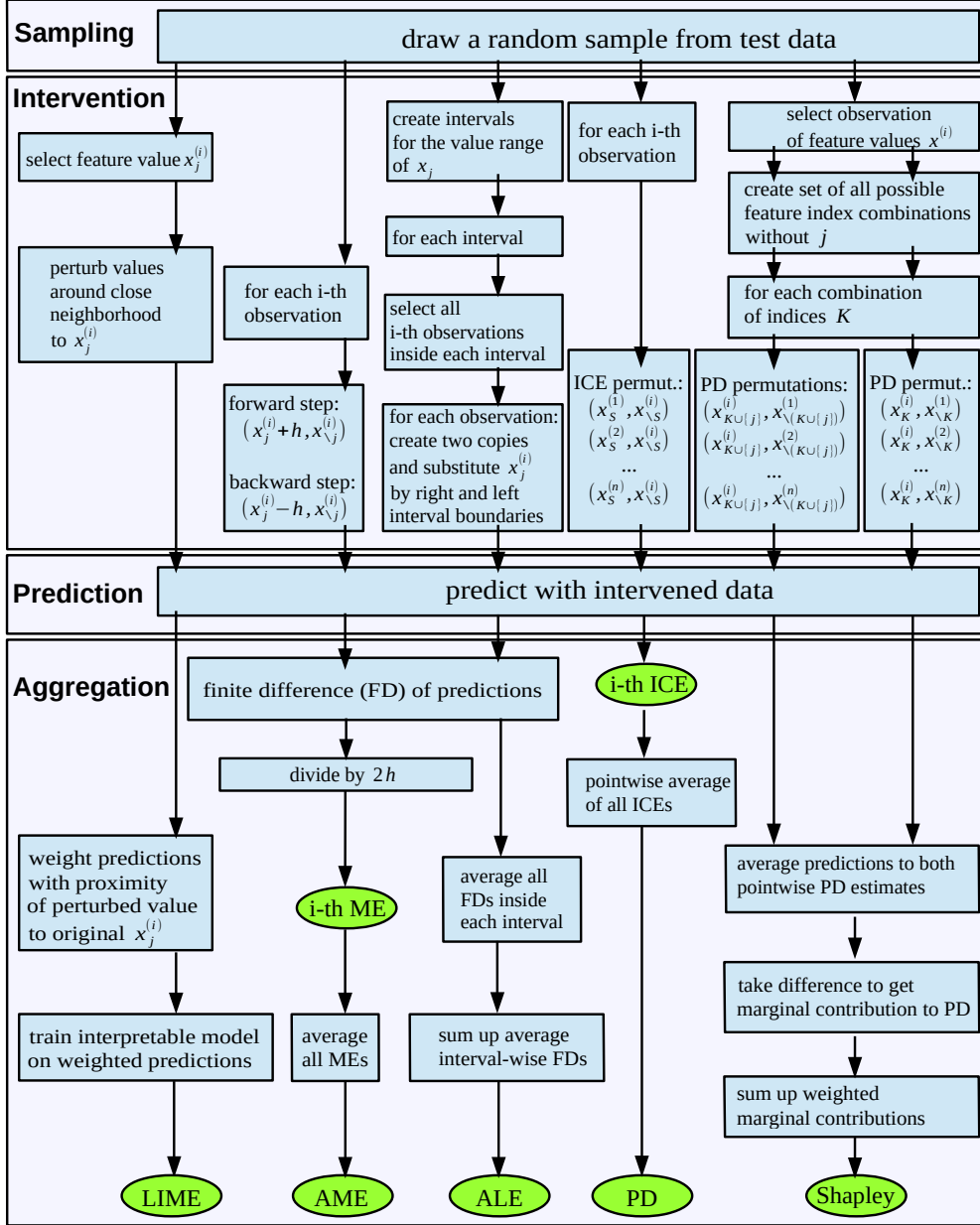


Fig. 1. We demonstrate how all presented model-agnostic methods for feature effects are based on the SIPA framework. For every method, we assign each computational step to the corresponding generalized SIPA work stage. Contrary to all other methods, LIME is based on training an intrinsically interpretable model during the aggregation stage. We consider training a model to be an aggregation, because it corresponds to an optimization problem where the training data is aggregated to a function. For reasons of simplicity, we do not differentiate between the actual functions or values and their estimates.

5 Feature Importance

We categorize model-agnostic importance measures into two groups: variance-based and performance-based.

Variance-based: A mostly flat trajectory of a single ICE curve implies that in the underlying predictive model, varying x_j does not affect the prediction for this specific observation. If all ICE curves are shaped similarly, the PD can be used instead. In [10] the authors propose a measure for the curvature of the PD as a feature importance metric. Let the average value of the estimated PD of the j -th feature be denoted by $\widehat{PD}_{\hat{f},j}(x_j) = \frac{1}{n} \sum_{i=1}^n \widehat{PD}_{\hat{f},j}(x_j^{(i)})$. The estimated importance $\widehat{IMP}_{\widehat{PD},j}$ of the j -th feature corresponds to the standard deviation of the feature's estimated PD function. The flatter the PD, the smaller its standard deviation and therefore the importance metric. For categorical features, the range of the PD is divided by 4. This is supposed to represent an approximation to the estimate of the standard deviation for small to medium sized samples [10].

$$\widehat{IMP}_{\widehat{PD},j} = \begin{cases} \sqrt{\frac{1}{n-1} \sum_{i=1}^n \left[\widehat{PD}_{\hat{f},j}(x_j^{(i)}) - \widehat{PD}_{\hat{f},j}(x_j) \right]^2} & x_j \text{ continuous} \\ \frac{1}{4} \left[\max \left\{ \widehat{PD}_{\hat{f},j}(x_j) \right\} - \min \left\{ \widehat{PD}_{\hat{f},j}(x_j) \right\} \right] & x_j \text{ categorical} \end{cases} \quad (4)$$

In [20] the authors propose the feature importance ranking measure (FIRM). They define a conditional expected score (CES) function for the j -th feature.

$$CES_{\hat{f},j}(v) = \mathbb{E}_{X_{\setminus j}} \left[\hat{f}(x_j, X_{\setminus j}) \mid x_j = v \right] \quad (5)$$

It turns out that Eq. (5) is equivalent to the PD from Eq. (1), conditional on $x_j = v$.

$$\begin{aligned} CES_{\hat{f},j}(v) &= \mathbb{E}_{X_{\setminus j}} \left[\hat{f}(v, X_{\setminus j}) \right] \\ &= PD_{\hat{f},j}(v) \end{aligned}$$

The FIRM corresponds to the standard deviation of the CES function with all values of x_j used as conditional values. This in turn is equivalent to the standard deviation of the PD. The FIRM is therefore equivalent to the feature importance metric in Eq. (4).

$$\widehat{FIRM}_{\hat{f},j} = \sqrt{\text{Var}(\widehat{CES}_{\hat{f},j}(x_j))} = \sqrt{\text{Var}(\widehat{PD}_{\hat{f},j}(x_j))} = \widehat{IMP}_{\widehat{PD},j}$$

Performance-based: The permutation feature importance (PFI), originally developed by [3] as a model-specific tool for random forests, was described as a model-agnostic one by [6]. If feature values are shuffled in isolation, the relationship between the feature and the target is broken up. If the feature is important

for the predictive performance, the shuffling should result in an increased loss [4]. Permuting x_j corresponds to drawing from a new random variable \tilde{X}_j that is distributed like X_j but independent of $X_{\setminus j}$ [4]. The model-agnostic PFI measures the difference between the generalization error (GE) on data with permuted and non-permuted values.

$$PFI_{\hat{f},j} = \mathbb{E} \left[\mathcal{L}(\hat{f}(\tilde{X}_j, X_{\setminus j}), Y) \right] - \mathbb{E} \left[\mathcal{L}(\hat{f}(X_j, X_{\setminus j}), Y) \right]$$

Let the permutation of x_j be denoted by \tilde{x}_j . Consider the sample of test data \mathcal{D}_j where x_j has been permuted, and the non-permuted sample \mathcal{D} . The PFI estimate is given by the difference between GE estimates with permuted and non-permuted values.

$$\begin{aligned} \widehat{PFI}_{\hat{f},j} &= \widehat{GE}(\hat{f}, \mathcal{D}_j) - \widehat{GE}(\hat{f}, \mathcal{D}) \\ &= \frac{1}{n} \sum_{i=1}^n \mathcal{L}(\hat{f}(\tilde{x}_j^{(i)}, x_{\setminus j}^{(i)}), y^{(i)}) - \frac{1}{n} \sum_{i=1}^n \mathcal{L}(\hat{f}(x_j^{(i)}, x_{\setminus j}^{(i)}), y^{(i)}) \end{aligned} \quad (6)$$

In [4] the authors propose individual conditional importance (ICI) and partial importance (PI) curves as visualization techniques that disaggregate the global PFI estimate. They are based on the same principle as the ICE and PD. The ICI visualizes the influence of a feature on the predictive performance for a single observation, while the PI visualizes the average influence of a feature for all observations. Consider the prediction for the i -th observation with observed values $\hat{f}(x_j^{(i)}, x_{\setminus j}^{(i)})$ and the prediction $\hat{f}(x_j^{(l)}, x_{\setminus j}^{(i)})$ where $x_j^{(i)}$ was replaced by a value $x_j^{(l)}$ from the marginal distribution of observed values x_j . The change in loss is given by:

$$\Delta \mathcal{L}^{(i)}(x_j^{(l)}) = \mathcal{L}(\hat{f}(x_j^{(l)}, x_{\setminus j}^{(i)})) - \mathcal{L}(\hat{f}(x_j^{(i)}, x_{\setminus j}^{(i)}))$$

The ICI curve of the i -th observation plots the value pairs $(x_j^{(l)}, \Delta \mathcal{L}^{(i)}(x_j^{(l)}))$ for all l values of x_j . The PI curve is the pointwise average of all ICI curves at all l values of x_j . It plots the value pairs $(x_j^{(l)}, \frac{1}{n} \sum_{i=1}^n \Delta \mathcal{L}^{(i)}(x_j^{(l)}))$ for all l values of x_j . Substituting values of x_j essentially resembles shuffling them. The authors demonstrate how averaging the values of the PI curve results in an estimation of the global PFI.

$$\widehat{PFI}_{\hat{f},j} = \frac{1}{n} \sum_{l=1}^n \frac{1}{n} \sum_{i=1}^n \Delta \mathcal{L}^{(i)}(x_j^{(l)})$$

Furthermore, a feature importance measure called Shapley feature importance (SFIMP) was proposed in [4]. Shapley importance values based on model refits with distinct sets of features were first introduced by [5] for feature selection. This changes the behavior of the learning algorithm and is not helpful to evaluate a single model, as noted by [4]. The SFIMP is based on the same computations as the Shapley value but replaces the payout function with one that is sensitive to the model performance. The authors define a new payout $v_{GE}(x_j)$

that substitutes the estimated PD with the estimated GE. This is equivalent to the estimated PFI from Eq. (6).

$$v_{GE}(x_j) = \widehat{GE}(\hat{f}, \mathcal{D}_j) - \widehat{GE}(\hat{f}, \mathcal{D}) = \widehat{PFI}_{\hat{f},j} = v_{PFI}(x_j)$$

We can therefore refer to $v_{GE}(x_j)$ as $v_{PFI}(x_j)$ and regard the SFIMP as an extension to the PFI [4].

6 Extending the Framework to Importance Computations

Variance-based importance methods measure the variance of feature effect estimates, which we already demonstrated to be based on the SIPA framework. Therefore, we simply add a variance computation during the aggregation stage. Performance-based techniques measure changes in loss, i.e., there are two possible modifications. First, we predict on non-intervened or intervened data (prediction stage). Second, we aggregate predictions to the loss (aggregation stage). In Fig. 2, we demonstrate how feature importance computations are based on the same work stages as feature effect computations.

7 Conclusion

In recent years, various model-agnostic interpretation methods have been developed. Due to different notations and terminology it is difficult to see how they are related. By deconstructing them into sequential work stages, one discovers striking similarities in their methodologies. We first provided a survey on model-agnostic interpretation methods and then presented the generalized SIPA framework of sequential work stages. First, there is a sampling stage to reduce computational costs. Second, we intervene in the data in order to change the predictions made by the black box model. Third, we predict on intervened or non-intervened data. Fourth, we aggregate the predictions. We embedded multiple methods to estimate the effect (ICE and PD, ALEs, MEs, Shapley values and LIME) and importance (FIRM, PFI, ICI and PI and the SFIMP) of features into the framework. By pointing out how all demonstrated techniques are based on a single methodology, we hope to work towards a more unified view on model-agnostic interpretations and to establish a common ground to discuss them in future work.

Acknowledgments

This work is supported by the Bavarian State Ministry of Science and the Arts as part of the Centre Digitisation.Bavaria (ZD.B) and by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibilities for its content.

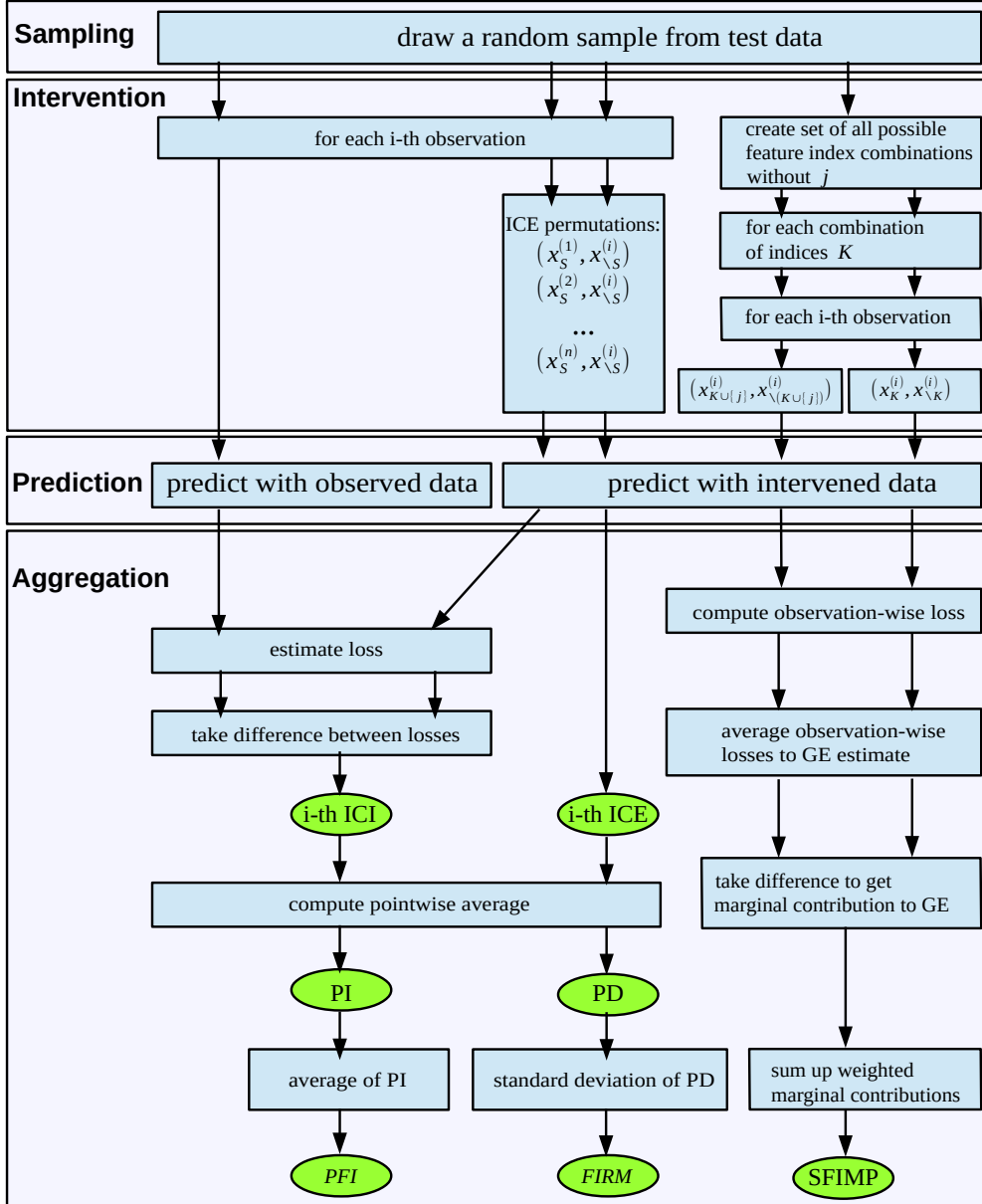


Fig. 2. We demonstrate how importance computations are based on the same work stages as effect computations. In the same way as in Fig. 1, we assign the computational steps of all techniques to the corresponding generalized SIPA work stages. Variance-based importance measures such as FIRM measure the variance of a feature effect, i.e., we add a variance computation during the aggregation stage. Performance-based importance measures such as ICI, PI, PFI and SFIMP are based on computing changes in loss after the intervention stage. For reasons of simplicity, we do not differentiate between the actual functions or values and their estimates.

References

1. Apley, D.W.: Visualizing the effects of predictor variables in black box supervised learning models. ArXiv e-prints arXiv:1612.08468 (Dec 2016)
2. Bartus, T.: Estimation of marginal effects using margeff. *The Stata Journal* **5**(3), 309 – 329 (2005)
3. Breiman, L.: Random forests. *Machine Learning* **45**(1), 5–32 (Oct 2001)
4. Casalicchio, G., Molnar, C., Bischl, B.: Visualizing the feature importance for black box models. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. pp. 655–670. Springer (2018)
5. Cohen, S., Dror, G., Ruppin, E.: Feature selection via coalitional game theory. *Neural Computation* **19**(7), 1939–1961 (2007)
6. Fisher, A., Rudin, C., Dominici, F.: Model class reliance: Variable importance measures for any machine learning model class, from the “Rashomon” perspective. ArXiv e-prints arXiv:1801.01489 (Jan 2018)
7. Fisher, A., Rudin, C., Dominici, F.: All models are wrong but many are useful: Variable importance for black-box, proprietary, or misspecified prediction models, using model class reliance. arXiv e-prints arXiv:1801.01489 (Jan 2018)
8. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. *Ann. Statist.* **29**(5), 1189–1232 (10 2001)
9. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics* **24** (09 2013)
10. Greenwell, B.M., Boehmke, B.C., McCarthy, A.J.: A simple and effective model-based variable importance measure. ArXiv e-prints arXiv:1805.04755 (May 2018)
11. Hechtlinger, Y.: Interpretation of prediction models using the input gradient. arXiv e-prints arXiv:1611.07634 (Nov 2016)
12. Leeper, T.J.: margins: Marginal effects for model objects (2018)
13. Lipovetsky, S., Conklin, M.: Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry* **17**(4), 319–330 (October 2001)
14. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in Neural Information Processing Systems* 30, pp. 4765–4774. Curran Associates, Inc. (2017)
15. Molnar, C.: *Interpretable Machine Learning* (2019), <https://christophm.github.io/interpretable-ml-book/>
16. Molnar, C., Bischl, B., Casalicchio, G.: iml: An R package for interpretable machine learning. *JOSS* **3**(26), 786 (2018)
17. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should I trust you?: Explaining the predictions of any classifier. In: *Knowledge Discovery and Data Mining (KDD)* (2016)
18. Rudin, C., Ertekin, Ş.: Learning customized and optimized lists of rules with mathematical programming. *Mathematical Programming Computation* **10**(4), 659–702 (Dec 2018)
19. Štrumbelj, E., Kononenko, I.: Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems* **41**(3), 647–665 (Dec 2014)
20. Zien, A., Krämer, N., Sonnenburg, S., Rätsch, G.: The feature importance ranking measure. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) *Machine Learning and Knowledge Discovery in Databases*. pp. 694–709. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)

8. Model-agnostic Feature Importance and Effects with Dependent Features - A Conditional Subgroup Approach

Contributing article:

Molnar, C., König, G., Bischl, B., and Casalicchio, G. (2020). Model-agnostic Feature Importance and Effects with Dependent Features - A Conditional Subgroup Approach. *arXiv preprint arXiv:2006.04628*.

Copyright information:

Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

Author contributions:

Christoph Molnar wrote most of the paper. Gunnar König wrote a big part of Section 4 and provided the proofs in Appendix A and B. All authors added input, suggested modifications proofread and revised the paper.

Model-agnostic Feature Importance and Effects with Dependent Features—A Conditional Subgroup Approach

Christoph Molnar · Gunnar König ·
Bernd Bischl · Giuseppe Casalicchio

Abstract The interpretation of feature importance in machine learning models is challenging when features are dependent. Permutation feature importance (PFI) ignores such dependencies, which can cause misleading interpretations due to extrapolation. A possible remedy is more advanced conditional PFI approaches that enable the assessment of feature importance conditional on all other features. Due to this shift in perspective and in order to enable correct interpretations, it is therefore important that the conditioning is transparent and humanly comprehensible. In this paper, we propose a new sampling mechanism for the conditional distribution based on permutations in conditional subgroups. As these subgroups are constructed using decision trees (transformation trees), the conditioning becomes inherently interpretable. This not only provides a simple and effective estimator of conditional PFI, but also local PFI estimates within the subgroups. In addition, we apply the conditional subgroups approach to partial dependence plots (PDP), a popular method for describing feature effects that can also suffer from extrapolation when features are dependent and interactions are present in the model. We show that PFI and PDP based on conditional subgroups often outperform methods such as conditional PFI based on knockoffs, or accumulated local effect plots. Furthermore, our approach allows for a more fine-grained interpretation of feature effects and importance within the conditional subgroups.

C. Molnar¹, G. König², B. Bischl³, G. Casalicchio⁴
Department of Statistics, Ludwig-Maximilians-University Munich, Munich, Germany

C. Molnar¹
Leibniz Institute for Prevention Research and Epidemiology – BIPS GmbH, Bremen, Germany
E-mail: christoph.molnar.ai@gmail.com

CRediT taxonomy: Conceptualization: 1, 2, 3, 4; Methodology: 1, 2, 4; Formal analysis and investigation: 1, 2; Writing - original draft preparation: 1, 2; Writing - review and editing: 2, 3, 4; Visualization: 1; Validation: 1, 2; Software: 1; Funding acquisition: 1, 3, 4; Supervision: 3, 4

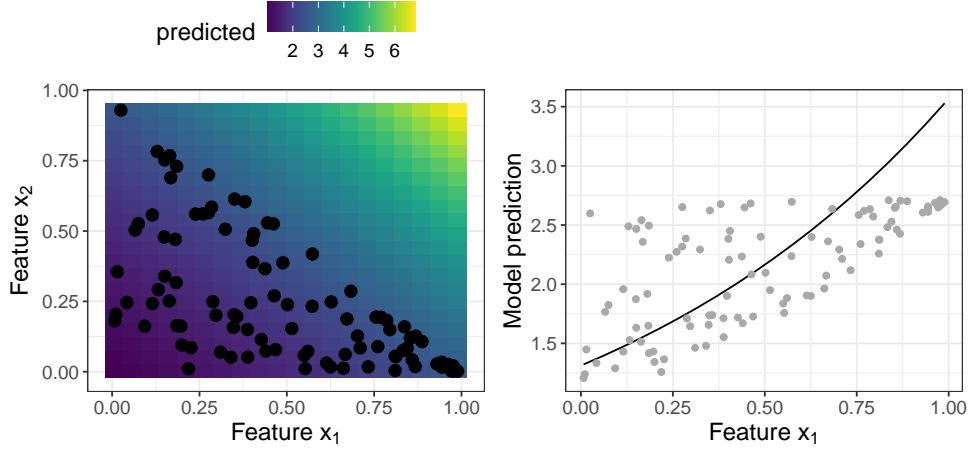


Fig. 1 Misleading PDP. Simulation of features $x_1 \sim U(0, 1)$, $x_2 \sim U(0, 1 - x_1)$ and a non-additive prediction model $\hat{f}(x) = \exp(x_1 + x_2)$. **Left:** Scatter plot with 100 data points and the prediction surface of \hat{f} . **Right:** PDP of x_1 . The grey dots are observed $(x_1, \hat{f}(x_1, x_2))$ -pairs. For $x_1 > 0.75$ the PDP suggests higher average predictions than the maximum prediction observed in the data.

Keywords Interpretable Machine Learning, Explainable AI, Permutation Feature Importance, Partial Dependence Plot

1 Introduction

Many model-agnostic machine learning (ML) interpretation methods (see Molnar (2019); Guidotti et al. (2018) for an overview) are based on making predictions on perturbed input features, such as permutations of features. The partial dependence plot (PDP) (Friedman et al., 1991) visualizes how changing a feature affects the prediction on average. The permutation feature importance (PFI) (Breiman, 2001; Fisher et al., 2019) quantifies the importance of a feature as the reduction in model performance after permuting a feature. PDP and PFI change feature values without conditioning on the remaining features. If features are dependent, such changes can lead to extrapolation to areas of the feature space with low density. For non-additive models such as tree-based methods or neural networks, extrapolation can result in misleading interpretations (Strobl et al., 2008; Tološi and Lengauer, 2011; Hooker and Mentch, 2019; Molnar et al., 2020). An illustration of the problem is given in Figure 1.

Extrapolation can be avoided by sampling a feature conditional on all other features and thereby preserving the joint distribution (Strobl et al., 2008; Hooker and Mentch, 2019). This yields conditional variants of the PDP and PFI that have to be interpreted differently. While the interpretation of marginal PDP and PFI is *independent* of the other features, the interpretation of conditional PDP and PFI is *conditional* on other features.

Figure 2 shows how conditional PFI can be misinterpreted: Features X_1 and X_3 have the same coefficient in a linear model and the same marginal PFI, but X_1 has a lower conditional PFI since it is correlated with feature X_2 . The conditional PFI must be interpreted as the additional, unique contribution of a feature given all features we conditioned on (König et al., 2020; Fisher et al., 2019). It therefore has also been called “partial importance” (Debeer and Strobl, 2020). If interpreted incorrectly, this can lead to the wrong conclusion that, for example, two strongly dependent features are irrelevant for the prediction model (Figure 2). The correct conclusion would be that a feature is less relevant given knowledge of the dependent feature.

In Figure 2, the conditional PDP shows a positive effect for a feature that has a negative coefficient in a linear regression model. The discrepancy is due to correlation of the feature with another feature with a large positive coefficient. The conditional effect of a feature is a mix of its marginal effect and the marginal effects of all dependent features (Hooker and Mentch, 2019; Apley and Zhu, 2016). While conditional PFI might assign a low importance to a feature on which the model relied heavily, the conditional PDP has the opposite pitfall: it can show an effect for a feature that was not used by the model. This interpretation might be undesirable and is similar to the omitted variable bias phenomenon, which also happens in Figure 2: regressing \hat{f} from X_2 , while ignoring X_1 (Apley and Zhu, 2016).

The interpretation of conditional PFI and PDP requires knowledge of the dependence structure between the feature of interest and the other features. Such knowledge of dependence structures would help explain differences between a feature’s marginal and conditional PFI and break down the conditional PDP into the effect of the feature of interest and that of the dependent features. However, state-of-the-art conditional sampling mechanisms such as knockoffs (Barber et al., 2015; Candès et al., 2018; Watson and Wright, 2019) do not provide a readily interpretable conditioning.

Our **contributions** are the following. We propose the conditional subgroup PDPs (cs-PDPs) and PFIs (cs-PFIs). Both are based on conditional subgroup permutation (cs-permutation), a sampling method for the conditional distribution. Standard (i.e., marginal) PDPs and PFIs are computed and interpreted within subgroups of the data, enabling a local interpretation of feature effect and importance while handling the problem of extrapolation. We construct the subgroups for a feature by training a decision tree in which the distribution of the feature becomes less dependent on other features. The tree structure allows interpretation of how other features influence the effect and importance of the feature at hand. We show that the conditional PFI estimate based on cs-PFIs can recover the ground truth in simulations and often outperforms related approaches. In addition, we study how well different conditional PDP/PFI approaches retain the joint distribution of data sets from the OpenML-CC18 benchmarking suite (Bischl et al., 2019) and show that cs-permutation achieves state-of-the-art data fidelity. We demonstrate that the cs-PDPs have a high model fidelity, that is, they are closer to the model prediction than other feature effect methods. By inspecting the cs-PFIs and cs-PDPs in combination

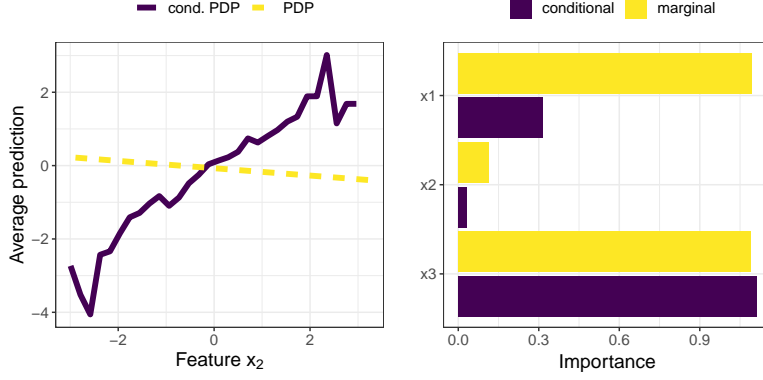


Fig. 2 Simulation of a linear model $\hat{f}(x) = x_1 - 0.1 \cdot x_2 + x_3$ with $x_1, x_2, x_3 \sim N(0, 1)$ and a correlation of 0.978 between x_1 and x_2 . **Left:** PDP and conditional PDP for feature x_2 . The conditional PDP mixes the effects of x_1 and x_2 and thus shows a positive effect. **Right:** PFI and conditional PFI of x_1 , x_2 and x_3 . The PFI of x_1 decreases when x_1 is permuted conditional on x_2 and vice versa.

with the respective subgroup descriptions, insights into the model and the dependence structure of the data are possible. We show how we can trade off human-intelligibility of the subgroups for extrapolation by choosing the granularity of the grouping. In an application, we illustrate how cs-PDPs and cs-PFIs can reveal new insights into the ML model and the data.

2 Notation and Background

We consider ML prediction functions $\hat{f} : \mathbb{R}^p \mapsto \mathbb{R}$, where $\hat{f}(\mathbf{x})$ is a model prediction and $\mathbf{x} \in \mathbb{R}^p$ is a p -dimensional feature vector. We use $\mathbf{x}_j \in \mathbb{R}^n$ to refer to an observed feature (vector) and X_j to refer to the j -th feature as a random variable. With \mathbf{x}_{-j} we refer to the complementary feature space $\mathbf{x}_{\{1, \dots, p\} \setminus \{j\}} \in \mathbb{R}^{n \times (p-1)}$ and with X_{-j} to the corresponding random variables. We refer to the value of the j -th feature from the i -th instance as $x_j^{(i)}$ and to the tuples $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$ as data.

The **Permutation Feature Importance (PFI)** is defined as the increase in loss when feature X_j is permuted:

$$PFI_j = \mathbb{E}[L(Y, \hat{f}(\tilde{X}_j, X_{-j}))] - \mathbb{E}[L(Y, \hat{f}(X_j, X_{-j}))] \quad (1)$$

If the random variable \tilde{X}_j has the same marginal distribution as X_j (e.g., permutation), the estimate yields the marginal PFI. If \tilde{X}_j follows the conditional distribution $\tilde{X}_j \sim X_j | X_{-j}$, we speak of the conditional PFI. The PFI is estimated with the following formula:

$$\widehat{PFI}_j = \frac{1}{n} \sum_{i=1}^n \left(\frac{1}{M} \sum_{m=1}^M \tilde{L}^{m(i)} - L^{(i)} \right) \quad (2)$$

where $L^{(i)} = L(y^{(i)}, \hat{f}(\mathbf{x}^{(i)}))$ is the loss for the i -th observation and $\tilde{L}^{(i)} = L(y^{(i)}, \hat{f}(\tilde{x}_j^{(i)}, \mathbf{x}_{-j}^{(i)}))$ is the loss where $x_j^{(i)}$ was replaced by the m -th sample $\tilde{x}_j^{m(i)}$. The latter refers to the i -th feature value obtained by a sample of \mathbf{x}_j . The sample can be repeated M -times for a more stable estimation of $\tilde{L}^{(i)}$. Numerous variations of this formulation exist. Breiman (2001) proposed the PFI for random forests, which is computed from the out-of-bag samples of individual trees. Subsequently, Fisher et al. (2019) introduced a model-agnostic PFI version.

The marginal **Partial Dependence Plot (PDP)** (Friedman et al., 1991) describes the average effect of the j -th feature on the prediction.

$$PDP_j(x) = \mathbb{E}[\hat{f}(x, X_{-j})], \quad (3)$$

If the expectation is conditional on X_j , $\mathbb{E}[\hat{f}(x, X_{-j})|X_j = x]$, we speak of the conditional PDP. The marginal PDP evaluated at feature value x is estimated using Monte Carlo integration:

$$\widehat{PDP}_j(x) = \frac{1}{n} \sum_{i=1}^n \hat{f}(x, \mathbf{x}_{-j}^{(i)}) \quad (4)$$

3 Related Work

In this section, we review conditional variants of PDP and PFI and other approaches that try to avoid extrapolation.

3.1 Related Work on Conditional PDP

The marginal plot (M-Plot) (Apley and Zhu, 2016) averages the predictions locally on the feature grid and mixes effects of dependent features.

Hooker (2007) proposed a functional ANOVA decomposition with hierarchically orthogonal components, based on integration using the joint distribution of the data, which in practice is difficult to estimate.

Accumulated Local Effect (ALE) plots by Apley and Zhu (2016) reduce extrapolation by accumulating the finite differences computed within intervals of the feature of interest. By definition, interpretations of ALE plots are thus only valid locally within the intervals. Furthermore, there is no straightforward approach to derive ALE plots for categorical features, since ALE requires ordered feature values. Our proposed approach can handle categorical features.

Another PDP variant based on stratification was proposed by Parr and Wilson (2019). However, this stratified PDP describes only the data and is independent of the model.

Individual Conditional Expectation (ICE) curves by Goldstein et al. (2015) can be used to visualize the interactions underlying a PDP, but they also suffer from the extrapolation problem. The “conditional” in ICE refers to conditioning on individual observations and not on certain features. As a solution,

Hooker and Mentch (2019) suggested to visually highlight the areas of the ICE curves in which the feature combinations are more likely.

3.2 Related Work on Conditional PFI

We review approaches that modify the PFI (Breiman, 2001; Fisher et al., 2019) in presence of dependent features by using a conditional sampling strategy.

Strobl et al. (2008) proposed the conditional variable importance for random forests (CVIRF), which is a conditional PFI variant of Breiman (2001). CVIRF was further analyzed and extended by Debeer and Strobl (2020). Both CVIRF and our approach rely on permutations based on partitions of decision trees. However, there are fundamental differences. CVIRF is specifically developed for random forests and relies on the splits of the underlying individual trees of the random forest for the conditional sampling. In contrast, our cs-PFI approach trains decision trees for each feature using X_{-j} as features and X_j as the target. Therefore, the subgroups for each feature are constructed from their conditional distributions (conditional on the other features) in a separate step, which is decoupled from the machine learning model to be interpreted. Our cs-PFI approach is model-agnostic, independent of the target to predict and not specific to random forests.

Hooker and Mentch (2019) made a general suggestion to replace feature values by estimates of $\mathbb{E}[X_j|X_{-j}]$.

Fisher et al. (2019) suggested to use matching and imputation techniques to generate samples from the conditional distribution. If X_{-j} has few unique combinations, they suggested to group $x_j^{(i)}$ by unique $\mathbf{x}_{-j}^{(i)}$ combinations and permute them for these fixed groups. For discrete and low-dimensional feature spaces, they suggest non-parametric matching and weighting methods to replace X_j values. For continuous or high-dimensional data, they suggest imputing X_j with $\mathbb{E}[X_j|X_{-j}]$ and adding residuals (under the assumption of homogeneous residuals). Our approach using permutation in subgroups can be seen as a model-driven, binary weighting approach extended to continuous features.

Knockoffs (Candes et al., 2018) are random variables which are “copies” of the original features that preserve the joint distribution but are otherwise independent of the prediction target. Knockoffs can be used to replace feature values for conditional feature importance computation. Watson and Wright (2019) developed a testing framework for PFI based on knockoff samplers such as Model-X knockoffs (Candes et al., 2018). Our approach is complementary since Watson and Wright (2019) is agnostic to the sampling strategy that is used. Others have proposed to use generative adversarial networks for generating knockoffs (Romano et al., 2019). Knockoffs are not transparent with respect to how they condition on the features, while our approach creates interpretable subgroups.

Conditional importance approaches based on model retraining have been proposed (Hooker and Mentch, 2019; Lei et al., 2018; Gregorutti et al., 2017).

8. Model-agnostic Feature Importance and Effects with Dependent Features - A Conditional Subgroup Approach

Importance and Effects with Dependent Features

7

Sampling Strategy	Used/Suggested By	Assumptions
No intervention on X_j	Drop-and-Refit, LOCO (Lei et al., 2018)	
Permute X_j	Marginal PFI (Breiman, 2001; Fisher et al., 2019), PDP (Friedman et al., 1991)	$X_j \perp\!\!\!\perp X_{-j}$
Replace X_j by knockoff Z_j with $(Z_j, X_{-j}) \sim (X_j, X_{-j})$ and $Z_j \perp\!\!\!\perp Y$	Knockoffs (Candes et al., 2018), CPI (Watson and Wright, 2019)	$(X_j, X_{-j}) \sim N$
Move each $x_j^{(i)}$ to left and right interval bounds	ALE (Apley and Zhu, 2016)	$X_j \perp\!\!\!\perp X_{-j}$ in intervals
Permute X_j in subgroups	cs-PFI, cs-PDP	$X_j \perp\!\!\!\perp X_{-j}$ in subgroups
Permute X_j in random forest tree nodes	CVIRF (Strobl et al., 2008; Debeer and Strobl, 2020)	$X_j \perp\!\!\!\perp X_{-j}$ cond. on tree splits in X_{-j} to predict Y
Impute X_j from X_{-j}	(Fisher et al., 2019)	Homogeneous residuals

Table 1 Sampling strategies for model-agnostic interpretation techniques.

Retraining the model can be expensive, and answers a fundamentally different question, often related to feature selection and not based on a fixed set of features. Hence, we focus on approaches that compute conditional PFI for a fixed model without retraining.

None of the existing approaches makes the dependence structures between the features explicit. It is unclear which of the features in X_{-j} influenced the replacement of X_j the most and how. Furthermore, little attention has been paid on evaluating how well different sampling strategies address the extrapolation problem. We address this gap with an extensive data fidelity experiment on the OpenML-CC18 benchmarking suite. To the best of our knowledge, our paper is also the first to conduct experiments using ground truth for the conditional PFI. Our approach works with any type of feature, be it categorical, numerical, ordinal and so on, since we rely on decision trees to find the subgroups used for conditioning. The differences between the different (conditional) PDP and PFI approaches ultimately boil down to how they sample from the conditional distribution. Table 1 lists different sampling strategies of model-agnostic interpretation methods and summarizes their assumptions to preserve the joint distribution.

4 Conditional Subgroups

We suggest approaching the dependent feature problem by constructing an interpretable grouping G_j such that the feature of interest X_j becomes less dependent on remaining features X_{-j} within each subgroup. In the best case the features become independent: $(X_j \perp\!\!\!\perp X_{-j})|G_j$. Assuming that we find a grouping in which $(X_j \perp\!\!\!\perp X_{-j})|G_j$ holds, sampling from the group-wise marginal distribution removes extrapolation (see Figure 3) and within each

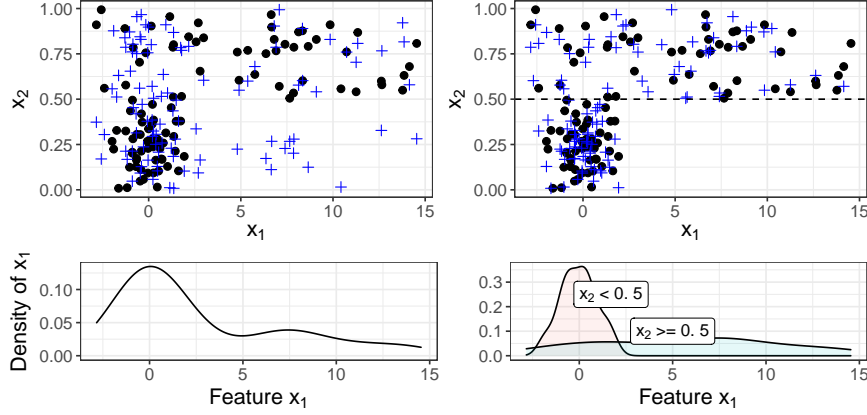


Fig. 3 Features $X_2 \sim U(0, 1)$ and $X_1 \sim N(0, 1)$, if $X_2 < 0.5$, else $X_1 \sim N(4, 4)$ (black dots). **Top left:** The crosses are permutations of X_1 . For $X_2 < 0.5$, the permutation extrapolates. **Bottom left:** Marginal density of X_1 . **Top right:** Permuting X_1 within subgroups based on X_2 ($X_2 < 0.5$ and $X_2 \geq 0.5$) reduces extrapolation. **Bottom right:** Densities of X_1 conditional on the subgroups.

group, the samples from the marginal and the conditional distribution would coincide. Such groupings exist when, for example, the features in X_{-j} are categorical, or when the conditional distribution of X_j only depends on discrete changes in features X_{-j} . Such a grouping would consequently enable (1) the application of standard PFI and PDP within each group *without extrapolation* and (2) sampling from the global conditional distribution $P(X_j|X_{-j})$ using group-wise permutation and aggregation. With our approach we exploit these properties to derive both a group-wise marginal interpretation and, for the PFI, a global conditional interpretation. Even when such a discrete grouping does not exist, e.g., when the true dependence is linear, the cs-permutation reduces extrapolation, see Figure 4. Moreover, an accurate interpretation requires the groupings to be human-intelligible. We can gain insight into how the model behaves within specific subgroups which is not possible with approaches that directly sample X_j conditional on all features X_{-j} (Candes et al., 2018; Strobl et al., 2008; Aas et al., 2019; Fisher et al., 2019; Watson and Wright, 2019).

For our approach, any algorithm can be used that splits the data in X_{-j} so that the distribution of X_j becomes more homogeneous within a group and more heterogeneous between groups. We consider decision tree algorithms for this task, which predict X_j based on splits in X_{-j} . Decision tree algorithms directly or indirectly optimize splits for heterogeneity of some aspects of the distribution of X_j in the splits. The partitions in a decision tree can be described by decision rules that lead to that terminal leaf. We leverage this partitioning to construct an interpretable grouping \mathcal{G}_j^k based on random variable G_j for a specific feature X_j . The new variable can be calculated by assigning every observation the indicator of the partition that it lies in (mean-

ing for observation i with $x_{-j}^{(i)} \in \mathcal{G}_j^k$ the group variable's value is defined as $g_j^{(i)} := k$.

Transformation trees (trtr) (Hothorn and Zeileis, 2017) are able to model the conditional distribution of a variable. This approach partitions the feature space so that the distribution of the target (here X_j) within the resulting subgroups \mathcal{G}_j^k is homogeneous, which means that the group-wise parameterization of the modeled distribution is independent of X_{-j} . Transformation trees directly model the target's distribution $\mathbb{P}(X_j \leq x) = F_Z(h(x))$, where F_Z is the chosen (cumulative) distribution function and h a monotone increasing transformation function (hence the name transformation trees). The transformation function is defined as $\mathbf{a}(y)^T \boldsymbol{\theta}$ where $\mathbf{a} : \mathcal{X}_j \mapsto \mathbb{R}^k$ is a basis function of polynomials or splines. The task of estimating the distribution is reduced to estimating $\boldsymbol{\theta}$, and the trees are split based on hypothesis tests for differences in $\boldsymbol{\theta}$ given X_{-j} , and therefore differences in the distribution of X_j . For more detailed explanations of transformation trees please refer to Hothorn and Zeileis (2017).

In contrast, a simpler approach would be to use **classification and regression trees (CART)** (Breiman et al., 1984), which, for regression, minimizes the variance within nodes, effectively finding partitions with different means in the distribution of X_j . However, CART's split criterion only considers differences in the expectation of the distribution of X_j given X_{-j} : $\mathbb{E}[X_j|X_{-j}]$. This means CART could only make X_j and X_{-j} independent if the distribution of X_j only depends in its expectation on X_{-j} (and if the dependence can be modeled by partitioning the data). Any differences in higher moments of the distribution of X_j such as the variance of $X_j|X_{-j}$ cannot be detected.

We evaluated both trtr which are theoretically well equipped for splitting distributions and CART, which are established and well-studied. For the remainder of this paper, we have set the default minimum number of observations in a node to 30 for both approaches. For the transformation trees, we used the Normal distribution as target distribution and we used Bernstein polynomials of degree five for the transformation function. Higher-order polynomials do not seem to increase model fit further (Hothorn, 2018).

We denote the subgroups by $\mathcal{G}_j^k \subset \mathbb{R}^{p-1}$, where $k \in \{1, \dots, K_j\}$ is the k -th subgroup for feature j , with K_j groups in total for the j -th feature. The subgroups per feature are disjoint: $\mathcal{G}_j^l \cap \mathcal{G}_j^k = \emptyset, \forall l \neq k$ and $\bigcup_{k=1}^{K_j} \mathcal{G}_j^k = \mathbb{R}^{p-1}$. Let $(\mathbf{y}_j^k, \mathbf{x}_j^k)$ be a subset of (\mathbf{y}, \mathbf{x}) that refers to the data subset belonging to the subgroup \mathcal{G}_j^k . Each subgroup can be described by the decision path that leads to the respective terminal node.

4.1 Remarks

4.1.1 Continuous Dependencies

For conditional independence $X_j \perp X_{-j} | G_j^k$ to hold, the chosen decision tree approach has to capture the (potentially complex) dependencies between X_j

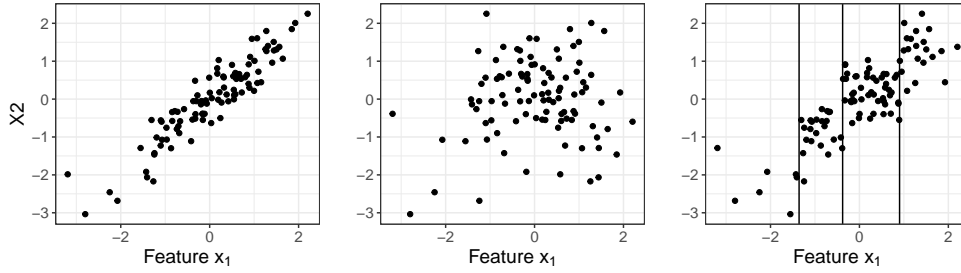


Fig. 4 **Left:** Simulation of features $X_1 \sim N(0, 1)$ and $X_2 \sim N(0, 1)$ with a covariance of 0.9. **Middle:** Unconditional permutation extrapolates strongly. **Right:** Permuting on partitions found by CART (predicting X_2 from X_1) has greatly reduces extrapolation, but cannot get rid of it completely. x_1 and x_2 remain correlated in the partitions.

and X_{-j} . CART can only capture differences in the expected value of $X_j|X_{-j}$ but are insensitive to changes in, for example, the variance. Transformation trees are in principle agnostic to the specified distribution and the default transformation family of distributions is very general, as empirical results suggest (Hothorn and Zeileis, 2017). However, the approach is based on the assumption that the dependence can be modeled with a discrete grouping. For example, in the case of linear Gaussian dependencies, the corresponding optimal variable would be linear Gaussian itself, and would be in conflict with our proposed interpretable grouping approach. Even in these settings the approach allows an approximation of the conditional distribution. In the case of simple linear Gaussian dependencies, partitioning the feature space will still **reduce extrapolation**. But we never get rid of it completely, unless there are only individual data points left in each partition, see Figure 4.

4.1.2 Sparse Subgroups

Fewer subgroups are generally desirable for two reasons: (1) for a good approximation of the marginal distribution within a subgroup, a sufficient number of observations per group is required, which might lead to fewer subgroups, and (2) a large number of subgroups leads to more complex groups, which reduces their human-intelligibility and therefore forfeits the added value of the local, subgroup-wise interpretations. As we rely on decision trees, we can adjust the granularity of the grouping using hyperparameters such as the maximum tree depth. By controlling the maximum tree depth, we can control the trade-off between the depth of the tree (and hence its interpretability) and the homogeneity of the distribution within the subgroups.

4.2 Conditional Subgroup Permutation Feature Importance (cs-PFI)

We estimate the cs-PFI of feature X_j within a subgroup \mathcal{G}_j^k as:

$$PFI_j^k = \frac{1}{n_k} \sum_{i: \mathbf{x}^{(i)} \in \mathcal{G}_j^k} \left(\frac{1}{M} \sum_{m=1}^M L(y^{(i)}, \hat{f}(\tilde{x}_j^{m(i)}, \mathbf{x}_{-j}^{(i)})) - L(y^{(i)}, \hat{f}(\mathbf{x}^{(i)})) \right), \quad (5)$$

where $\tilde{x}_j^{m(i)}$ refers to a feature value obtained from the m -th permutation of x_j within the subgroup k_j . This estimation is exactly the same as the marginal PFI (Equation 2), except that it only includes observations from the given subgroup. Algorithm 1 describes the estimation of the cs-PFIs for a given feature on unseen data.

Algorithm 1: Estimate cs-PFI

Input: Model f ; data $\mathcal{D}_{train}, \mathcal{D}_{test}$; loss L ; feature j ; no. permutations M

- 1 Train tree T_j with target X_j and features X_{-j} using \mathcal{D}_{train}
 - 2 Compute subgroups \mathcal{G}_j^k for \mathcal{D}_{test} based on terminal nodes of T_j , $k \in \{1, \dots, K_j\}$
 - 3 **for** $k \in \{1, \dots, K_j\}$ **do**
 - 4 $L_{orig} := \frac{1}{n_k} \sum_{i: \mathbf{x}^{(i)} \in \mathcal{G}_j^k} L(y^{(i)}, \hat{f}(\mathbf{x}^{(i)}))$
 - 5 **for** $m \in \{1, \dots, M\}$ **do**
 - 6 Generate $\tilde{\mathbf{x}}_j^m$ by permuting feature values \mathbf{x}_j within subgroup \mathcal{G}_j^k
 - 7 $L_{perm}^m := \frac{1}{n_k} \sum_{i: \mathbf{x}^{(i)} \in \mathcal{G}_j^k} L(y^{(i)}, \hat{f}(\tilde{x}_j^{m(i)}, \mathbf{x}_{-j}^{(i)}))$
 - 8 cs-PFI $_j^k = \frac{1}{M} \sum_{m=1}^M L_{perm}^m - L_{orig}$
 - 9 cs-PFI $_j = \frac{1}{n} \sum_{k=1}^{K_j} n^k PFI_j^k$
-

The algorithm has two outcomes: We get local importance values for feature X_j for each subgroup (cs-PFI $_j^k$; Algorithm 1, line 8) and a global conditional feature importance (cs-PFI $_j$; Algorithm 1, line 9). The latter is equivalent to the weighted average of subgroup importances regarding the number of observations within each subgroup (see proof in Appendix A).

$$\text{cs-PFI}_j = \frac{1}{n} \sum_{k=1}^{K_j} n^k PFI_j^k$$

The cs-PFIs needs the same amount of model evaluations as the PFI ($O(nM)$). On top of that comes the cost for training the respective decision trees and making predictions to assign a subgroup to each observation.

Theorem 1 *When feature X_j is independent of features X_{-j} for a given dataset \mathcal{D} , each cs-PFI $_j^k$ has the same expectation as the marginal PFI, and an n/n_k -times larger variance, where n and n_k are the number of observations in the data and the subgroup \mathcal{G}_j^k .*

The proof of Theorem 1 is shown in Appendix B. Theorem 1 has the practical implication that even in the case of applying cs-PFI to an independent feature, we will retrieve the marginal PFI, and not introduce any problematic interpretations. Equivalence in expectation and higher variance under the independence of X_j and X_{-j} holds true even if the partitions \mathcal{G}_j^k would be randomly chosen. Theorem 1 has further consequences regarding overfitting: Assuming a node has already reached independence between X_j and X_{-j} , then further splitting the tree based on noise will not change the expected cs-PFIs.

4.3 Conditional Subgroup Partial Dependence Plots (cs-PDPs)

The conditional PDP has a different interpretation than the marginal PDP, as the motivating example in Figure 2 showed: The conditional PDP can be interpreted as the effect of a feature on the prediction, given that all other features would change according to the joint distribution. This violates a desirable property that the effect of features that were not used by the model should have a zero effect curve. This poses a dilemma for dependent features: Either extrapolate using the marginal PDP, or use the conditional PDP with undesirable properties for interpretation. Our proposed cs-PDPs reduces extrapolation while allowing a marginal interpretation *within* each subgroup. We compute the cs-PDP $_j^k$ for each subgroup \mathcal{G}_j^k using the marginal PDP formula in Equation 4.

$$\text{cs-PDP}_j^k(x) = \frac{1}{n^k} \sum_{i: x^{(i)} \in \mathcal{G}_j^k} \hat{f}(x, x_{-j}^{(i)})$$

This results in multiple cs-PDPs per feature, which can be displayed together in the same plot as in Figure 12. As shown in Figure 5, even features that do not contribute to the prediction at all can have a conditional PDP different from zero. We therefore argue that an aggregation of the cs-PDPs to the conditional PDP is not meaningful for model interpretation, and we suggest to plot the group-wise curves. For the visualization of the cs-PDPs, we suggest to plot the PDPs similar to boxplots, where the dense center quartiles are indicated with a bold line (see Figure 6). We restrict each cs-PDP $_j^k$ to the interval $[\min(\mathbf{x}_j), \max(\mathbf{x}_j)]$, with $\mathbf{x}_j = (x_j^{(1)}, \dots, x_j^{(n_j^k)})$.

Equivalently to PFI, the subgroup PDPs approximate the true marginal PDP even if the features are independent.

Theorem 2 *When feature X_j is independent of features X_{-j} for a given dataset \mathcal{D} , each cs-PDP $_j^k$ has the same expectation as the marginal PDP, and an n/n_k -times larger variance, where n and n_k are the number of observations in the data and the subgroup \mathcal{G}_j^k .*

The proof of Theorem 2 is shown in Appendix C. Theorem 2 has the same practical implications as Theorem 1: Even if the features are independent, we

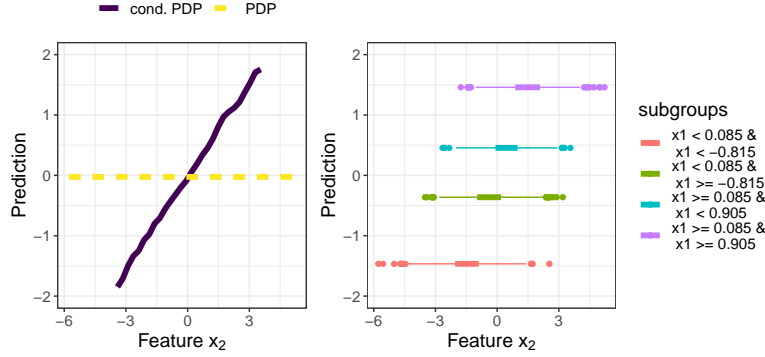


Fig. 5 We simulated a linear model of $y = x_1 + \epsilon$ with $\epsilon \sim N(0,1)$ and an additional feature X_2 which is correlated with X_1 (≈ 0.72). The conditional PDP (left) gives the false impression that X_2 has an influence on the target. The cs-PDPs help in this regard, as the effects due to X_1 (changes in intercept) are clearly separated from the effect that X_2 has on the target (slope of the cs-PDPs), which is zero. Unlike the marginal PDP, the cs-PDPs reveals that for increasing X_2 we expect that the prediction increases due to the correlation between X_1 and X_2 .

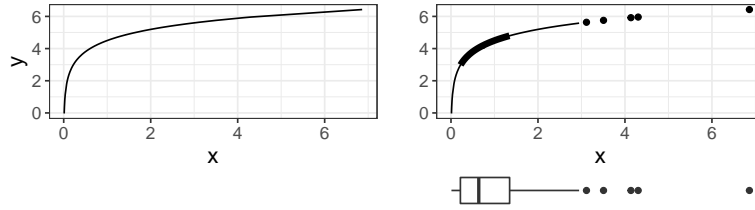


Fig. 6 Left: Marginal PDP. Bottom right: Boxplot showing the distribution of feature X . Top right: PDP with boxplot-like emphasis. In the x -range, the PDP is drawn from $\pm 1.58 \cdot IQR / \sqrt{n}$, where IQR is the range between the 25% and 75% quantile. If this range exceeds $[\min(x_j), \max(x_j)]$, the PDP is capped. Outliers are drawn as points. The PDP is bold between the 25% and 75% quantiles.

will, in expectation, get the marginal PDPs. And when trees are grown deeper than needed, in expectation the cs-PDPs will yield the same curve.

Both the PDP and the set of cs-PDPs need $O(nM)$ evaluations, since $\sum_{k=1}^{K_j} n^k = n$ (and worst case $O(n^2)$ if evaluated at each $x_j^{(i)}$ value). Again, there is an additional cost for training the respective decision trees and making predictions.

5 Training Conditional Sampling Approaches

To ensure that sampling approaches are not overfitting, we suggest to separate training and sampling, where training covers all estimation steps that involve data. For this purpose, we refer to the training data with \mathcal{D}_{train} and to the data for importance computation with \mathcal{D}_{test} . This section both describes how

we compared the sampling approaches in the following chapters and serves as a general recommendation for how to use the sampling approaches.

For our cs-permutation, we trained the CART / transformation trees on \mathcal{D}_{train} and permuted X_j of \mathcal{D}_{test} within the terminal nodes of the tree. For CVIRF (Strobl et al., 2008; Debeer and Strobl, 2020), which is specific to random forests, we trained the random forest on \mathcal{D}_{train} to predict the target y and permuted X_j of \mathcal{D}_{test} within the terminal nodes. For Model-X knockoffs (Candes et al., 2018), we fitted the second-order knockoffs on \mathcal{D}_{train} and replaced X_j in \mathcal{D}_{test} with its knockoffs. For the imputation approach (Fisher et al., 2019), we trained a random forest on \mathcal{D}_{train} to predict X_j from X_{-j} , and replaced values of X_j in \mathcal{D}_{test} with their random forest predictions plus a random residual. For the interval-based sampling (Apley and Zhu, 2016), we computed quantiles of X_j using \mathcal{D}_{train} and perturbed X_j in \mathcal{D}_{test} by moving each observation once to the left and once to the right border of the respective intervals. The marginal permutation (PFI, PDP) required no training, we permuted (i.e., shuffled) the feature X_j in \mathcal{D}_{test} .

6 Conditional PFI Ground Truth Simulation

We compared our cs-PFI approach using CART (tree cart) and transformation trees (tree trtr), CVIRF (Strobl et al., 2008; Debeer and Strobl, 2020), Model-X knockoffs (ko) (Candes et al., 2018) and the imputation approach (impute rf) (Fisher et al., 2019) in ground truth simulations. We simulated the following data-generating process: $y^{(i)} = f(x^{(i)}) = x_1^{(i)} \cdot x_2^{(i)} + \sum_{j=1}^{10} x_j^{(i)} + \epsilon^{(i)}$, where $\epsilon^{(i)} \sim N(0, \sigma_\epsilon)$. All features, except feature X_1 followed a Gaussian distribution: $X_j \sim N(0, 1)$. Feature X_1 was simulated as a function of the other features plus noise: $x_1^{(i)} = g(x_{-1}^{(i)}) + \epsilon_x$. We simulated the following scenarios by changing g and ϵ_x :

- In the **independent** scenario, X_1 did not depend on any feature: $g(x_{-1}^{(i)}) = 0$, $\epsilon_x \sim N(0, 1)$. This scenario served as a test how the different conditional PFI approaches handle the edge case of independence.
- The **linear** scenario introduces a strong correlation of X_1 with feature X_2 : $g(x_{-1}^{(i)}) = x_2^{(i)}$, $\epsilon_x \sim N(0, 1)$.
- In the **non-linear** scenario, we simulated X_1 as a non-linear function of multiple features: $g(x_{-1}^{(i)}) = 3 \cdot \mathbb{1}(x_2^{(i)} > 0) - 3 \cdot \mathbb{1}(x_2^{(i)} \leq 0) \cdot \mathbb{1}(x_3^{(i)} > 0)$. Here also the variance of $\epsilon_x \sim N(0, \sigma_x)$ is a function of x : $\sigma_x(x^{(i)}) = \mathbb{1}(x_2^{(i)} > 0) + 2 \cdot \mathbb{1}(x^{(i)} \leq 0) \cdot \mathbb{1}(x_3^{(i)} > 0) + 5 \cdot \mathbb{1}(x_2^{(i)} \leq 0) \cdot \mathbb{1}(x_3^{(i)} \leq 0)$.
- For the **multiple linear dependencies** scenario, we chose X_1 to depend on many features: $g(x_{-1}^{(i)}) = \sum_{j=2}^{10} x_j^{(i)}$, $\epsilon_x \sim N(0, 5)$.

For each scenario, we varied the number of sampled data points $n \in \{300, 3000\}$ and the number of features $p \in \{9, 90\}$. To “train” each of the cPFI methods, we used $2/3 \cdot n$ (200 or 2000) data points and the rest (100

/ 1000) to compute the cPFI. The experiment was repeated 1000 times. We examined two settings.

- In setting (I), we assumed that the model recovered the true model $\hat{f} = f$.
- In setting (II), we trained a random forest with 100 trees (Breiman, 2001).

In both settings, the true conditional distribution of X_1 given the remaining features is known (function g and error distribution is known). Therefore we can compute the ground truth conditional PFI, as defined in Equation 2. We generated the samples of X_1 according to g to get the \tilde{X}_1 values and compute the increase in loss. The conditional PFIs differed in settings (I) and (II) since in (I) we used the true f , and in (II) the trained random forest \hat{f} .

6.1 Conditional PFI Ground Truth Results

For setting (I), the mean squared errors between the estimated conditional PFIs and the ground truth are displayed in Table 2, and the distributions of conditional PFI estimates in Figure 7. In the *independent scenario*, where conditional and marginal PFI are equal, all methods performed equally well, except in the low n , high p scenario, where the knockoffs sometimes failed. As expected, the variance was higher for all methods when $n = 300$. In the *linear scenario*, the marginal PFI was clearly different from the conditional PFI. There was no clear best performing conditional PFI approach, as the results differ depending on training size n and number of features p . For low n and low p , knockoffs performed best. For high p , regardless of n , the cs-permutation approaches worked best, which might be due to the feature selection mechanism inherent to trees. The *multiple linear dependencies scenario* was the only scenario in which the cs-PFI approach was consistently outperformed by the other methods. Decision trees already need multiple splits for recovering linear relationships, and in this scenario, multiple linear relationships had to be recovered. Imputation with random forest worked well when multiple linear dependencies are present. For knockoffs, the results were mixed. As expected, the cs-PFI approach worked well in the *non-linear scenario*, and outperformed all other approaches. Knockoffs and imputation with random forests both overestimated the conditional PFI (except for knockoffs for $n = 300$ and $p = 90$). In addition to this bias, they had a larger variance compared to the cs-PFI approaches.

Generally, the transformation trees performed equal to or outperformed CART across all scenarios, except for the multiple linear dependencies scenario. Our cs-PFI approaches worked well in all scenarios, except when multiple (linear) dependencies were present. Even for a single linear dependence, the cs-PFI approaches were on par with knockoffs and imputation, and clearly outperformed both when the relationship was more complex.

In setting (II), a random forest was analyzed, which allowed us to include the conditional variable importance for random forests (CVIRF) by Strobl et al. (2008); Debeer and Strobl (2020) in the benchmark. The MSEs are

Table 2 MSE comparing estimated and true conditional PFI (scenario I). Legend: impute rf: Imputation with a random forest, ko: Model-X knockoffs, mPFI: (marginal) PFI, tree cart: cs-permutation based on CART, tree trtr: cs-permutation based on transformation trees.

setting	cs-PFI (cart)	cs-PFI (trtr)	impute rf	ko	mPFI
independent					
n=300, p=10	1.33	1.35	1.67	1.47	1.39
n=300, p=90	1.50	1.29	1.46	5.81	1.31
n=3000, p=10	0.14	0.15	0.16	0.13	0.15
n=3000, p=90	0.15	0.14	0.14	0.18	0.13
linear					
n=300, p=10	4.62	4.30	3.64	2.03	44.83
n=300, p=90	5.55	5.26	17.53	11.63	45.36
n=3000, p=10	0.40	0.26	0.26	0.63	37.40
n=3000, p=90	0.45	0.31	3.55	0.38	36.32
multi. lin.					
n=300, p=10	2443.67	2623.54	1276.41	1583.69	2739.83
n=300, p=90	2574.54	2896.47	2141.01	6607.73	2988.68
n=3000, p=10	1031.83	900.68	140.98	810.78	1548.37
n=3000, p=90	1075.95	1041.10	438.25	185.13	1599.59
non-linear					
n=300, p=10	22.00	17.76	265.73	668.34	1204.17
n=300, p=90	19.99	19.81	504.53	131.77	1248.74
n=3000, p=10	1.18	1.00	144.77	626.80	1156.32
n=3000, p=90	1.17	1.13	206.01	579.02	1136.83

displayed in Appendix D, Table 7, and the distribution of conditional PFI estimates in Appendix D in Figure 14. The results for all other approaches are comparable to setting (I). For the low n settings, CVIRF worked as well as the other approaches in the *independent scenario*. It outperformed the other approaches in the *linear scenario* and the *multiple linear scenario* (when n was small). The CVIRF approach consistently underestimated the conditional PFI in all scenarios with high n , even in the *independent scenario*. Therefore, we would recommend to analyze the conditional PFI for random forests using cs-PFI for lower dimensional dependence structures, and imputation for multiple (linear) dependencies.

7 Trading Interpretability for Accuracy

In an additional experiment, we examined the trade-off between the depth of the trees and the accuracy with which we recover the true conditional PFI. For scenario (I), we trained decision trees with different maximal depths (from 1 to 10) and analyzed how the resulting number of subgroups influenced the conditional PFI estimate. The experiment was repeated 1000 times. Figure 8 shows that the deeper the transformation trees (and the more subgroups), the better the true conditional PFI was approximated. The plot also shows

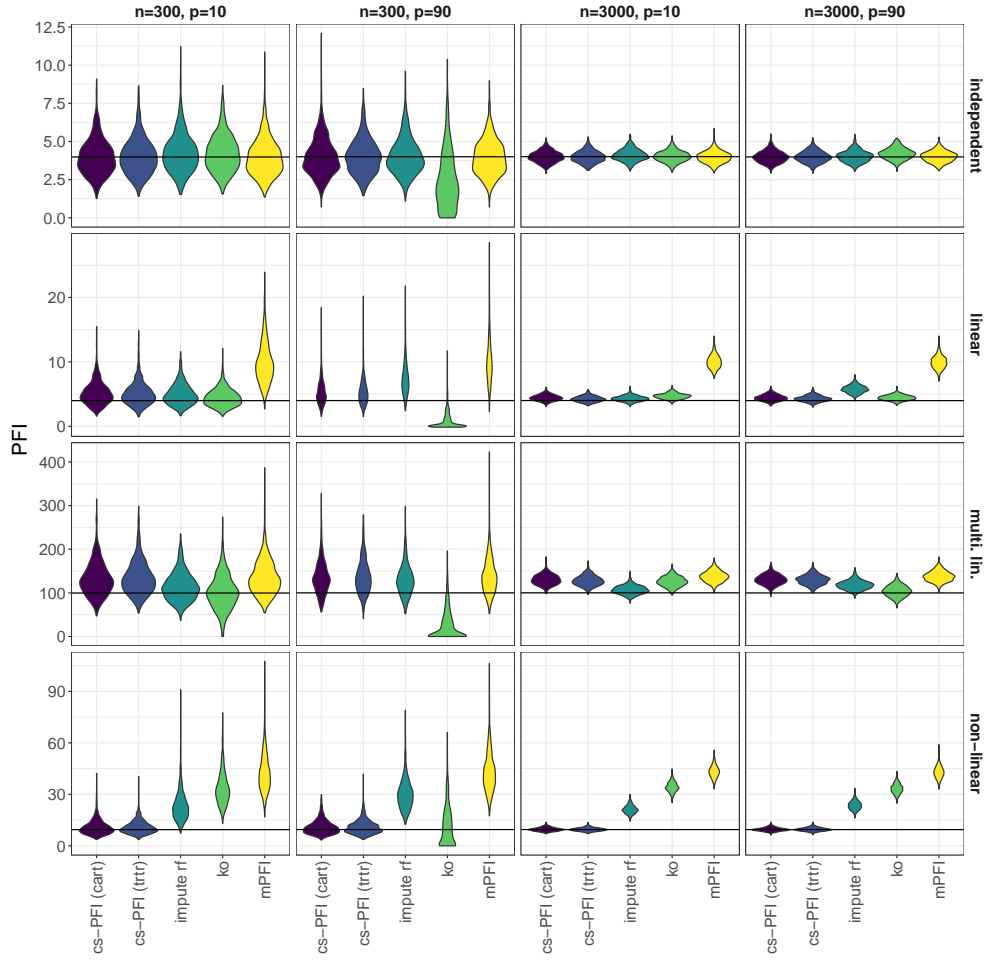


Fig. 7 Setting (I) comparing various conditional PFI approaches on the true model against the true conditional PFI (horizontal line) based on the data generating process.

that no overfitting occurred, which is in line with theoretical considerations in Theorem 1.

8 Data Fidelity Evaluation

PDP and PFI work by data intervention, prediction, and subsequent aggregation (Scholbeck et al., 2019). Based on data \mathcal{D} , the intervention creates a new data set. In order to compare different conditional sampling approaches, we define a measure of data fidelity to quantify the ability to preserve the joint distribution under intervention. Failing to preserve the joint distribution leads to extrapolation when features are dependent. Model-X knockoffs, for example, are directly motivated by preserving the joint distribution, while others, such as accumulated local effect plots do so more implicitly.

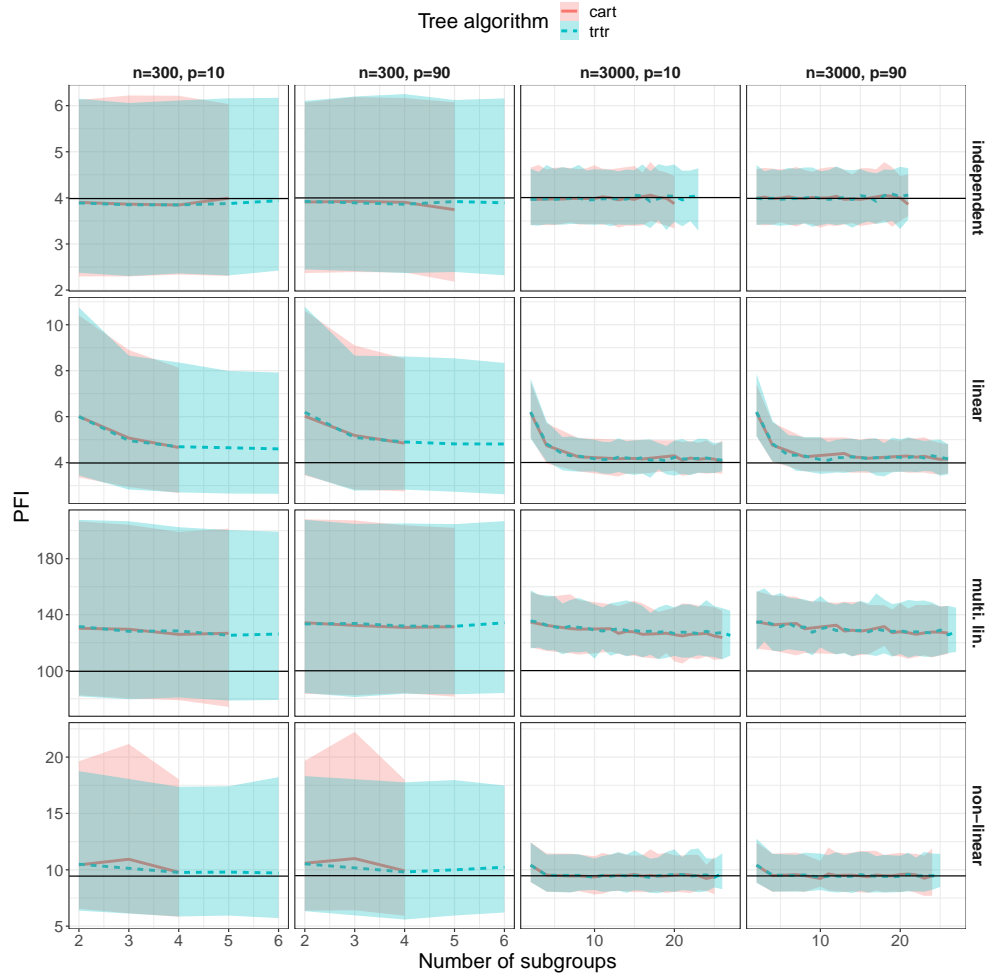


Fig. 8 Conditional PFI estimate using cs-PFI (**cart** / **transformation tree**) with increasing number of subgroups (simulation scenario I). Displayed is the median PFI over 1000 repetitions along with the 5% and 95% quartiles.

Data fidelity is the degree to which a sample \tilde{X}_j of feature X_j preserves the joint distribution, that is, the degree to which $(\tilde{X}_j, X_{-j}) \sim (X_j, X_{-j})$. In theory, any measure that compares two multivariate distributions can be used to compute the data fidelity. In practice, however, the joint distribution is unknown, which makes measures such as the Kullback-Leibler divergence impractical. We are dealing with two samples, one data set without and one with intervention.

In this classic two-sample test-scenario, the maximum mean discrepancy (MMD) can be used to compare whether two samples come from the same distribution (Fortet and Mourier, 1953; Smola et al., 2007; Gretton et al.,

2007, 2012). The empirical MMD is defined as:

$$\text{MMD}(\mathcal{D}, \tilde{\mathcal{D}}) = \frac{1}{n^2} \sum_{x, z \in \mathcal{D}} k(x, z) - \frac{2}{nl} \sum_{x \in \mathcal{D}, z \in \tilde{\mathcal{D}}} k(x, z) + \frac{1}{l^2} \sum_{x, z \in \tilde{\mathcal{D}}} k(x, z) \quad (6)$$

where $\mathcal{D} = \{x_j^{(i)}, x_{-j}^{(i)}\}_{i=1}^n$ is the original data set and $\tilde{\mathcal{D}} = \{\hat{x}_j^{(i)}, x_{-j}^{(i)}\}_{i=1}^l$ a data set with perturbed $x_j^{(i)}$. For both data sets, we scaled numerical features to a mean of zero and a standard deviation of one. For the kernel k we used the radial basis function kernel for all experiments. For parameter σ of the radial basis function kernel, we chose the median L2-distance between data points which is a common heuristic (Gretton et al., 2012). We measure data fidelity as the negative logarithm of the MMD ($-\log(\text{MMD})$) to obtain a more condensed scale where larger values are better.

Definition 1 (MMD-based Data Fidelity) Let \mathcal{D} be a dataset, and $\tilde{\mathcal{D}}$ be another dataset from the same distribution, but with an additional intervention. We define the data fidelity as: Data Fidelity = $-\log(\text{MMD}(\mathcal{D}, \tilde{\mathcal{D}}))$.

We evaluated how different sampling strategies (see Table 1) affect the data fidelity measure for numerous data sets of the OpenML-CC18 benchmarking suite (Bischl et al., 2019). We removed all data sets with 7 or fewer features and data sets with more than 500 features. See Appendix E for an overview of the remaining data sets. For each data set, we removed all categorical features from the analysis, as the underlying sampling strategies of ALE plots and Model-X knockoffs are not well equipped to handle them. We were foremost interested in two questions:

- A) How does cs-permutation compare with other sampling strategies w.r.t. data fidelity?
- B) How do choices of tree algorithm (CART vs. transformation trees) and tree depth parameter affect data fidelity?

In each experiment, we selected a data set, randomly sampled a feature and computed the data fidelity of various sampling strategies as described in the pseudo-code in Algorithm 2.

For an unbiased evaluation, we split the data into three pieces: \mathcal{D}_{train} (40% of rows), \mathcal{D}_{test} (30% of rows) and \mathcal{D}_{ref} (30% of rows). We used \mathcal{D}_{train} to “train” each sampling method (e.g., train decision trees for cs-permutation, see Section 5). We used \mathcal{D}_{ref} , which we left unchanged and \mathcal{D}_{test} , for which the chosen feature was perturbed to estimate the data fidelity. For each data set, we chose 10 features at random, for which sampling was applied. Marginal permutation (which ignores the joint distribution) and “no perturbation” served as lower and upper bounds for data fidelity. For CVIRF, we only used one tree per random forest as we only compared the general perturbation strategy which is the same for each tree.

We repeated all experiments 30 times with different random seeds and therefore different data splits. All in all this produced 12210 results (42 data

Algorithm 2: Data Fidelity Experiments

Input: OpenML-CC18 data sets, sampling strategies

```

1 for data set  $\mathcal{D}$  in OpenML-CC18 do
2   Remove prediction target from  $\mathcal{D}$  (only keep it for CVIRF)
3   Randomly order features in  $\mathcal{D}$ 
4   for features  $j \in \{1, \dots, 10\}$  do
5     for repetition  $\in \{1, \dots, 30\}$  do
6       Sample  $\min(10,000, n)$  rows from  $\mathcal{D}$ 
7       Split sample into  $\mathcal{D}_{train}$  (40%),  $\mathcal{D}_{test}$  (30%) and  $\mathcal{D}_{ref}$  (30%)
8       for each sampling do
9         “Train” sampling approach using  $\mathcal{D}_{train}$  (e.g., construct
          subgroups, fit knockoff-generator, ...)
10        Generate conditional sample  $\tilde{X}_j$  for  $\mathcal{D}_{test}$ 
11        Estimate data fidelity as  $-\log(MMD(\mathcal{D}_{ref}, \mathcal{D}_{test}))$ 
12 return Set of data fidelity estimates

```

sets \times (up to) 10 features \times 30 repetitions) per sampling method. All results are shown in detail in Appendix E (Figures 15, 16, 17, 18).

Since the experiments are repeated across the same data sets and the same features, the data fidelity results are not independent. Therefore, we used a random intercept model (Bryk and Raudenbush, 1992) to analyze the differences in data fidelity between different sampling approaches. The random intercepts were nested for each data set and each feature. We chose “Marginal Permutation” as the reference category. We fitted two random intercept models: One to compare cs-permutation with fully-grown trees (CART, trtr) with other sampling methods and another one to compare different tree depths.

8.1 Results A) State-of-the-art comparison

Figure 9 shows the effect estimates of different sampling approaches modeled with a random intercept model. The results show that cs-permutation performed better than all other methods. Model-X knockoffs and the imputation approach (with random forests) came in second place and outperformed ALE and CVIRF. Knockoffs were proposed to preserve the joint distribution, but are based on multivariate Gaussian distribution. This seems to be too restrictive for the data sets in our experiments. CVIRF does not have much higher data fidelity than marginal permutation. However, results for CVIRF must be viewed with caution, since data fidelity regards all features equally – regardless of their impact on the model prediction. For example, a feature can be highly correlated with the feature of interest, but might not be used in the random forest. A more informative experiment for comparing CVIRF can be found in Section 6. Figure 15 and Figure 16 in Appendix E show the individual data fidelity results for the OpenML-CC18 data sets. Not perturbing the feature at all has the highest data fidelity and serves as the upper bound. The marginal permutation serves as a lower baseline. For most data

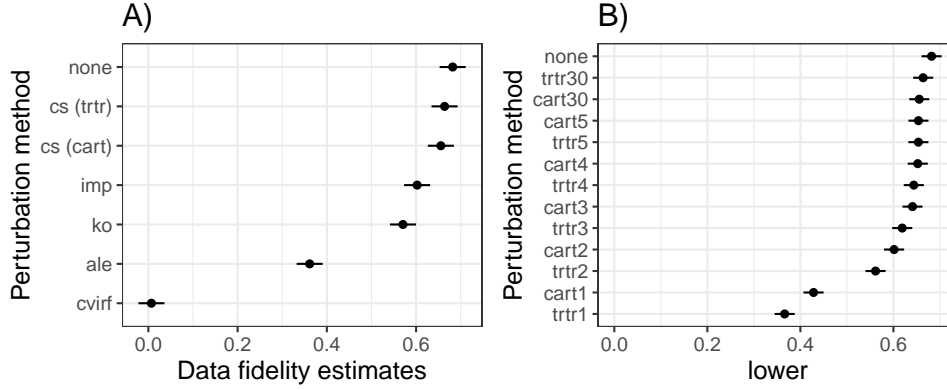


Fig. 9 Linear regression model coefficients and 95% confidence intervals for the effect of different sampling approaches on data fidelity, with (nested) random effects per data set and feature. **A)** Comparing different sampling approaches. No perturbation (“none”) and permutation (“perm”) serve as upper and lower bounds. **B)** Comparing cs-permutation using either CART or transformation trees and different tree depths (1,2,3,4,5 and 30). Marginal permutation is the reference category.

sets, cs-permutation has a higher data fidelity compared to all other sampling approaches. For all the other methods there is at least one data set on which they reach a low data fidelity (e.g., “semeion”, “qsar-biodeg” for ALE; “node-simulation”, “churn” for imputation; “jm1”, “pcl” for knockoffs). In contrast, cs-permutation achieves a consistently high data fidelity on all these data sets.

Additionally, we review the data fidelity rankings of the sampling methods in Table 3. The rankings show a similar picture as the random intercept model estimates, except that Model-X knockoffs have a better average ranking than imputation. This could be the case since on a few data sets (bank-marketing, electricity, see Figure 15 in Appendix E) Model-X knockoffs have a very low data fidelity but on most others a higher model fidelity than the imputation method.

	none	cs (trtr)	ko	cs (cart)	imp	ale	perm	cvirf
Mean ranks	2.50	3.51	3.70	3.76	4.25	4.61	6.82	6.84
SD	0.73	0.87	1.32	0.91	1.37	2.07	1.14	1.14

Table 3 Mean ranks and their standard deviation based on data fidelity of various perturbation methods over data sets, features and repetitions. **Legend:** none: No intervention, which serves as upper benchmark. cart30: cs-permutation with CART with maximal depth of 30. trtr30: cs-permutation with transformation trees with maximal depth of 30. imp: Imputation approach. ko: Model-X knockoffs Candes et al. (2018) . ale: ALE perturbation Apley and Zhu (2016). cvirf: Conditional variable importance for random forests Strobl et al. (2008). perm: Unconditional permutation.

8.2 Results B) tree configuration

We included shallow trees with maximum depth parameter from 1 to 5 to analyze the trade-off between tree depth and data fidelity. We included trees with a maximum depth parameter of 30 (“fully-grown” trees as this was the software’s limit) as an upper bound for each decision tree algorithm. Figure 9 B) shows that the deeper the trees (and the more subgroups), the higher the data fidelity. This is to be expected, since deeper trees allow for a more fine-grained separation of distributions. More importantly, we are interested in the trade-off between depth and data fidelity. Even splitting with a maximum depth of only 1 (two subgroups) strongly improves data fidelity over the simple marginal permutation for most data sets. A maximum depth of two means another huge average improvement in data fidelity, and already puts cs-permutation on par with knockoffs. A depth of three to four is almost as good as a maximum depth parameter of 30 and already outperforms all other methods, while still being interpretable due to their shortness. CART slightly outperforms transformation trees clearly when trees are shallow, which is surprising since transformation trees are, in theory, better equipped to handle changes in the distribution. Deeply grown transformation trees (max. depth of 30) slightly outperform CART. Figure 17 and Figure 18 in Appendix E show data fidelity aggregated by data set.

9 Model Fidelity

Model fidelity has been defined as how well the predictions of an explanation method approximate the ML model (Ribeiro et al., 2016). Similar to Szepannek (2019), we define model fidelity for feature effects as the mean squared error between model prediction and the prediction of the partial function f_j (which depends only on feature X_j) defined by the feature effect method, for example $f_j(x) = PDP_j(x)$. For a given data instance with observed feature value $x_j^{(i)}$, the predicted outcome of, for example, a PDP can be obtained by the value on the y-axis of the PDP at the observed x_j value.

$$\text{Model_Fidelity}(\hat{f}, f_j) = \frac{1}{n} \sum_{i=1}^n (\hat{f}(x^{(i)}) - f_j(x_j^{(i)}))^2, \quad (7)$$

where f_j is a feature effect function such as ALE or PDP. In order to evaluate ALE plots, they have to be adjusted such that they are on a comparable scale to a PDP (Apley and Zhu, 2016): $f_j^{ALE,adj} = f_j^{ALE} + \frac{1}{n} \sum_{i=1}^n \hat{f}(x^{(i)})$.

We trained random forests (500 trees), linear models and k-nearest neighbours models ($k = 7$) on various regression data sets (Table 4). 70% of the data were used to train the ML models and the transformation trees / CARTs. This ensure that results are not over-confident due to overfitting, see also Section 5. The remaining 30% of the data were used to evaluate model fidelity. For each model and each data set, we measured model fidelity between effect prediction and model prediction (Equation 7), averaged across observations and features.

8. Model-agnostic Feature Importance and Effects with Dependent Features - A Conditional Subgroup Approach

Importance and Effects with Dependent Features

23

	wine	satellite	wind	space	pollen	quake
No. of rows	6497	6435	6574	3107	3848	2178
No. of features	12	37	15	7	6	4

Table 4 We selected data sets from OpenML Vanschoren et al. (2014); Casalicchio et al. (2017) having 1000 to 8000 instances and a maximum of 50 numerical features. We excluded data sets with categorical features, since ALE cannot handle them.

Table 5 shows that the model fidelity of ALE and PDP is similar, while the cs-PDPs have the best model fidelity. This is an interesting result since the decision trees for the cs-PDPs are neither based on the model nor on the real target, but solely on the conditional dependence structure of the features. However, the cs-PDPs have the advantage that we obtain multiple plots. We did not aggregate the plots to a single conditional PDP, but computed the model fidelity for the PDPs within the subgroups (visualized in Figure 12). Our cs-PDPs using trees with a maximum depth of 2 have a better model fidelity than using a maximum depth of 1. We limited the analysis to interpretable conditioning and therefore allowed only trees with a maximum depth of 2, since a tree depth of 3 already means up to 8 subgroups which is already an impractical number of PDPs to have in one plot. CART sometimes beats trtr (e.g., on the “satellite” data set) but sometimes trtr has a lower loss (e.g., on the “wind” data set). Using different models (knn or linear model) produced similar results, see Appendix F.

	pollen	quake	satellite	space	wind	wine
PDP	9.61	0.04	4.80	0.03	44.84	0.75
ALE	9.91	0.04	4.81	0.03	44.83	0.75
cs-PDP trtr1	8.44	0.04	4.49	0.03	29.96	0.71
cs-PDP cart1	8.44	0.04	3.71	0.03	31.38	0.73
cs-PDP trtr2	8.17	0.04	3.25	0.03	26.56	0.70
cs-PDP cart2	8.29	0.04	3.05	0.03	25.96	0.71

Table 5 Median model fidelity averaged over features in a random forest for various data sets. The cPDPs always had a lower loss (i.e. higher model fidelity) than PDP and ALE. The loss monotonically decreases with increasing maximum tree depth for subgroup construction.

10 Application

In the following application, we demonstrate that cs-PDPs and cs-PFI are valuable tools to understand model and data beyond insights given by PFI, PDPs, or ALE plots. We trained a random forest to predict daily bike rentals (Dua and Graff, 2017) with given weather and seasonal information. The data ($n = 731$, $p = 9$) was divided into 70% training and 30% test data.

10.1 Analyzing Feature Dependence

The features in the bike data are dependent. For example, the correlation between temperature and humidity is 0.13. The data contains both categorical and numerical features and we are interested in the multivariate, non-linear dependencies. Thus, correlation is an inadequate measure of dependence. We therefore indicate the degree of dependence by showing the extent to which we can predict each feature from all other features in Table 6. This idea is based on the proportional reduction in loss (Cooil and Rust, 1994). Per feature, we trained a random forest to predict that feature from all other features. We measured the proportion of loss explained to quantify the dependence of the respective feature on all other features. For numerical features, we used the R-squared measure. For categorical features, we computed $1 - MMCE(y_{class}, rf(X)) / MMCE(y_{class}, x_{mode})$, where $MMCE$ is the mean misclassification error, y_{class} the true class, $rf()$ the classification function of the random forest and x_{mode} the most frequent class in the training data. We divided the training data into two folds and trained the random forest on one half. Then, we computed the proportion of explained loss on the other half and vice versa. Finally, we averaged the results. The feature “work” can be fully predicted by weekday and holiday. Season, temperature, humidity and weather can be partially predicted and are therefore not independent.

season	yr	holiday	weekday	temp	hum	work	weather	wind
45%	8%	29%	14%	66%	43%	100%	46%	12%

Table 6 Percentage of loss explained by predicting a feature from the remaining features with a random forest.

10.2 cs-PDPs and cs-PFI

To construct the subgroups, we used transformation trees with a maximum tree depth of 2 which limited the number of possible subgroups to 4. Figure 10 shows that for most features the biggest change in the estimated conditional PFI happens when moving from a maximum depth of 0 (= marginal PFI) to a depth of 2. This makes a maximum depth of 2 a reasonable trade-off between limiting the number of subgroups and accurately approximating the conditional PFI. We compared the marginal and conditional PFI for the bike rental predictions, see Figure 11.

The most important features, according to (marginal) PFI, were temperature and year. For the year feature, the marginal and conditional PFI are the same. Temperature is less important when we condition on season and humidity. The season already holds a lot of information about the temperature, so this is not a surprise. When we know that a day is in summer, it is not as important to know the temperature to make a good prediction. On humid

8. Model-agnostic Feature Importance and Effects with Dependent Features - A Conditional Subgroup Approach

Importance and Effects with Dependent Features

25

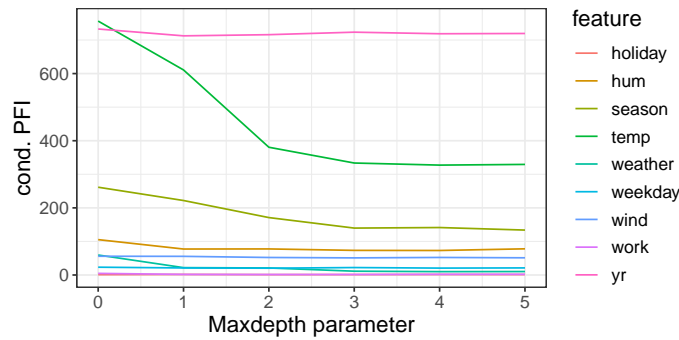


Fig. 10 Conditional feature importance by increasing maximum depth of the trees.

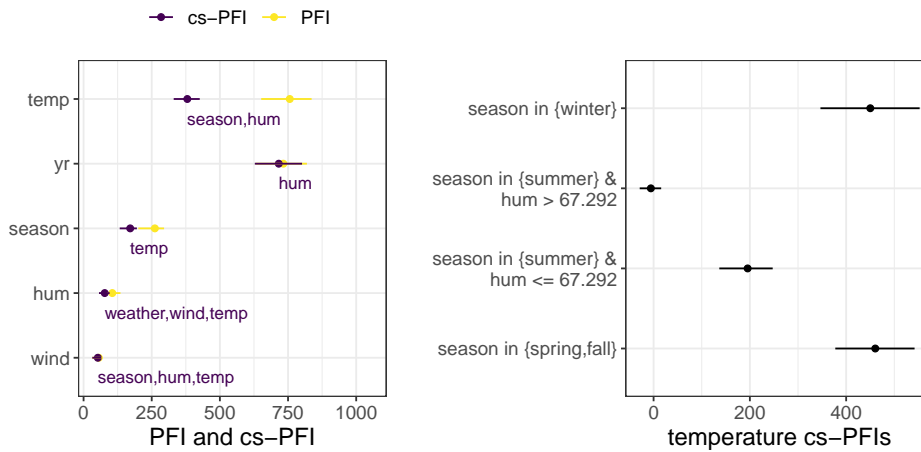


Fig. 11 Left: Comparison of PFI and cs-PFI for a selection of features. For cs-PFI we also show the features that constitute the subgroups. **Right:** Local cs-PFI of temperature within subgroups. The temperature feature is important in spring, fall and winter, but neglectable on summer days, especially humid ones.

summer days, the PFI of temperature is zero. However, in all other cases, it is important to know the temperature to predict how many bikes will be rented on a given day. The disaggregated cs-PFI in a subgroup can be interpreted as “How important is the temperature, given we know that the season and the humidity”.

Both ALE and PDP show a monotone increase of predicted bike rentals up until a temperature of 25 °C and a decrease beyond that. The PDP shows a weaker negative effect of very high temperatures which might be caused by extrapolation: High temperature days are combined with e.g. winter. A limitation of the ALE plot is that we should only interpret it locally within each interval that was used to construct the ALE plot. In contrast, our cs-PDP is explicit about the subgroup conditions in which the interpretation of the cs-PDP is valid and shows the distributions in which the feature effect may be interpreted. The local cs-PDPs in subgroups reveal a more nuanced picture:

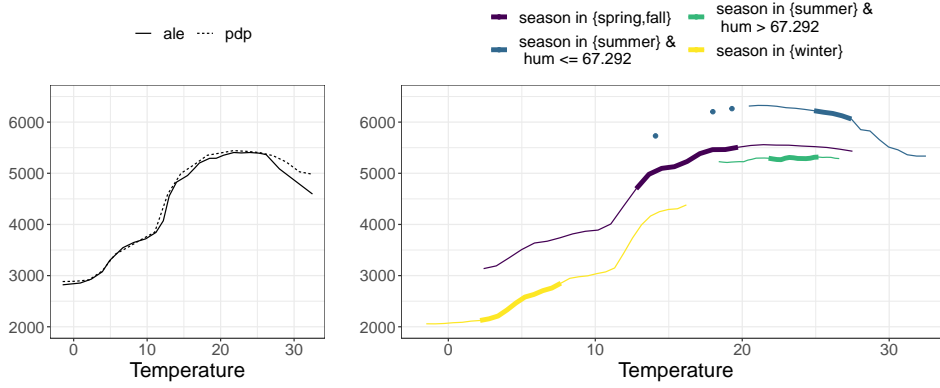


Fig. 12 Effect of temperature on predicted bike rentals. **Left:** PDP and ALE plot. **Right:** cs-PDPs for 4 subgroups.

For humid summer days, the temperature has no effect on the bike rentals, and the average number of rentals are below that of days with similar temperatures in spring, fall and drier summer days. The temperature has a slightly negative effect on the predicted number of bike rentals for dry summer days (humidity below 70.75). The change in intercepts of the local cs-PDP can be interpreted as the effect of the grouping feature (season). The slope can be interpreted as the temperature effect within a subgroup.

We also demonstrate the local cs-PDPs for the season, a categorical feature. Figure 13 shows both the PDP and our local cs-PDPs. The normal PDP shows that on average there is no difference between spring, summer and fall and only slightly less bike rentals in winter. The PDP with four subgroups conditional on temperature shows that the marginal PDP is misleading. The PDP indicates that in spring, summer and fall, around 4500 bikes are rented and in winter around 1000 fewer. The cs-PDPs in contrast show that, conditional on temperature, the differences between the seasons are much greater, especially for low temperatures. Only at high temperatures is the number of rented bikes similar between seasons.

11 Discussion

We proposed the cs-PFIs and cs-PDPs, which are variants of PFI and PDP that work when features are dependent. Both cs-PFIs and cs-PDPs rely on permutations in subgroups based on decision trees. The approach is simple: Train a decision tree to predict the feature of interest and compute the (marginal) PFI / PDP in each terminal node defined by the decision tree.

Compared to other approaches, cs-PFIs and cs-PDPs enable a human comprehensible grouping, which carries information how dependencies affect feature effects and importance. As we showed in various experiments, our methods are on par or outperform other methods in many dependence settings. We therefore recommend using cs-PDPs and cs-PFIs to analyze feature effects and

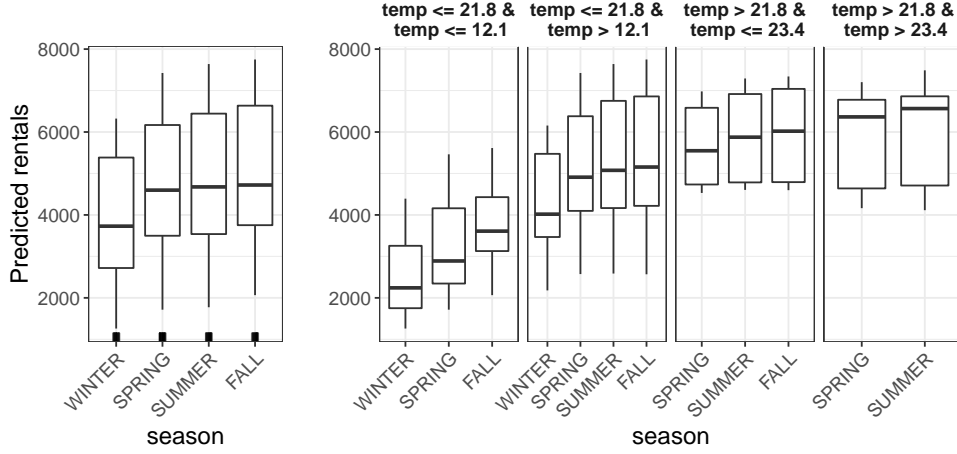


Fig. 13 Effect of season on predicted rentals. **Left:** PDP. **Right:** Local cs-PDPs. The cs-PDPs are conditioned on temperature, in which the tree split at 21.5 and at 9.5.

importances when features are dependent. However, due to their construction with decision trees, cs-PFIs and cs-PDPs do not perform well when the feature of interest depends on many other features, but only if it depends on a few features. We recommend analyzing the dependence structure beforehand, using the imputation approach with random forests in the case of multiple dependencies, and cs-PFIs in all other cases.

Our framework is flexible regarding the choice of partitioning and we leave the evaluation of the rich selection of possible decision tree and decision rules approaches to future research.

Reproducibility: All experiments were conducted using *mlr* (Lang et al., 2019) and R (R Core Team, 2017). We used the *iml* package (Molnar et al., 2018) for ALE and PDP, *party*/*partykit* (Hothorn and Zeileis, 2015) for CVIRF and *knockoff* (Patterson and Sesia, 2020) for Model-X knockoffs. The code for all experiments is available at https://github.com/christophM/paper_conditional_subgroups.

Acknowledgements This project is funded by the Bavarian State Ministry of Science and the Arts, by the Bavarian Research Institute for Digital Transformation (bidt) and supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A and by the German Research Foundation (DFG), Emmy Noether Grant 437611051. The authors of this work take full responsibilities for its content.

References

- Aas K, Jullum M, Løland A (2019) Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. arXiv preprint arXiv:190310464
- Apley DW, Zhu J (2016) Visualizing the effects of predictor variables in black box supervised learning models. arXiv preprint arXiv:161208468
- Barber RF, Candès EJ, et al. (2015) Controlling the false discovery rate via knockoffs. *The Annals of Statistics* 43(5):2055–2085
- Bischl B, Casalicchio G, Feurer M, Hutter F, Lang M, Mantovani RG, van Rijn JN, Vanschoren J (2019) Openml benchmarking suites. arXiv preprint arXiv:170803731
- Breiman L (2001) Random forests. *Machine learning* 45(1):5–32
- Breiman L, Friedman J, Olshen R, Stone C (1984) *Classification and Regression Trees*. Wadsworth and Brooks
- Bryk AS, Raudenbush SW (1992) *Hierarchical linear models: Applications and data analysis methods*. Sage Publications, Inc
- Candes E, Fan Y, Janson L, Lv J (2018) Panning for gold: ‘model-x’ knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 80(3):551–577
- Casalicchio G, Bossek J, Lang M, Kirchhoff D, Kerschke P, Hofner B, Seibold H, Vanschoren J, Bischl B (2017) OpenML: An R package to connect to the machine learning platform OpenML. *Comput Stat*
- Cooil B, Rust RT (1994) Reliability and expected loss: A unifying principle. *Psychometrika* 59(2):203–216
- Debeer D, Strobl C (2020) Conditional permutation importance revisited. *BMC bioinformatics* 21(1):1–30
- Dua D, Graff C (2017) UCI machine learning repository. URL <http://archive.ics.uci.edu/ml>
- Fisher A, Rudin C, Dominici F (2019) All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research* 20(177):1–81
- Fortet R, Mourier E (1953) Convergence de la répartition empirique vers la répartition théorique. In: *Annales scientifiques de l’École Normale Supérieure*, vol 70, pp 267–285
- Friedman JH, et al. (1991) Multivariate adaptive regression splines. *The annals of statistics* 19(1):1–67
- Goldstein A, Kapelner A, Bleich J, Pitkin E (2015) Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *J Comput Graph Stat* 24(1):44–65
- Gregorutti B, Michel B, Saint-Pierre P (2017) Correlation and variable importance in random forests. *Statistics and Computing* 27(3):659–678
- Gretton A, Fukumizu K, Teo CH, Song L, Schölkopf B, Smola AJ, et al. (2007) A kernel statistical test of independence. In: *Nips, Citeseer*, vol 20, pp 585–592

- Gretton A, Borgwardt KM, Rasch MJ, Schölkopf B, Smola A (2012) A kernel two-sample test. *The Journal of Machine Learning Research* 13(1):723–773
- Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D (2018) A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* 51(5):1–42
- Hooker G (2007) Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *J Comput Graph Stat* 16(3)
- Hooker G, Mentch L (2019) Please stop permuting features: An explanation and alternatives. *arXiv preprint arXiv:190503151*
- Hothorn T (2018) Top-down transformation choice. *Statistical Modelling* 18(3-4):274–298
- Hothorn T, Zeileis A (2015) partykit: A modular toolkit for recursive party-tioning in r. *The Journal of Machine Learning Research* 16(1):3905–3909
- Hothorn T, Zeileis A (2017) Transformation forests. *arXiv preprint arXiv:170102110*
- König G, Molnar C, Bischl B, Grosse-Wentrup M (2020) Relative feature importance. *arXiv preprint arXiv:200708283*
- Lang M, Binder M, Richter J, Schratz P, Pfisterer F, Coors S, Au Q, Casalicchio G, Kotthoff L, Bischl B (2019) mlr3: A modern object-oriented machine learning framework in R. *Journal of Open Source Software*
- Lei J, G'Sell M, Rinaldo A, Tibshirani RJ, Wasserman L (2018) Distribution-free predictive inference for regression. *Journal of the American Statistical Association* 113(523):1094–1111
- Molnar C (2019) Interpretable Machine Learning. <https://christophm.github.io/interpretable-ml-book/>
- Molnar C, Bischl B, Casalicchio G (2018) iml: An R package for interpretable machine learning. *JOSS* 3(26):786
- Molnar C, König G, Herbringer J, Freiesleben T, Dandl S, Scholbeck CA, Casalicchio G, Grosse-Wentrup M, Bischl B (2020) Pitfalls to avoid when interpreting machine learning models. *arXiv preprint arXiv:200704131*
- Parr T, Wilson JD (2019) A stratification approach to partial dependence for codependent variables. *arXiv preprint arXiv:190706698*
- Patterson E, Sesia M (2020) knockoff: The Knockoff Filter for Controlled Variable Selection. URL <https://CRAN.R-project.org/package=knockoff>, r package version 0.3.3
- R Core Team (2017) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria
- Ribeiro MT, Singh S, Guestrin C (2016) Why should i trust you?: Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, ACM, pp 1135–1144
- Romano Y, Sesia M, Candès E (2019) Deep knockoffs. *Journal of the American Statistical Association* pp 1–12
- Scholbeck CA, Molnar C, Heumann C, Bischl B, Casalicchio G (2019) Sampling, intervention, prediction, aggregation: A generalized framework for model-agnostic interpretations. In: *Joint European Conference on Machine*

- Learning and Knowledge Discovery in Databases, Springer, pp 205–216
- Smola A, Gretton A, Song L, Schölkopf B (2007) A hilbert space embedding for distributions. In: International Conference on Algorithmic Learning Theory, Springer, pp 13–31
- Strobl C, Boulesteix AL, Kneib T, Augustin T, Zeileis A (2008) Conditional variable importance for random forests. *BMC bioinformatics* 9(1):307
- Szepannek G (2019) How much can we see? A note on quantifying explainability of machine learning models. *arXiv preprint arXiv:191013376*
- Toloşi L, Lengauer T (2011) Classification with correlated features: unreliability of feature ranking and solutions. *Bioinformatics* 27(14):1986–1994
- Vanschoren J, Van Rijn JN, Bischl B, Torgo L (2014) OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter* 15(2):49–60
- Watson DS, Wright MN (2019) Testing conditional independence in supervised learning algorithms. *arXiv preprint arXiv:190109917*

A Decompose conditional PFI into cs-PFIs

Assuming a perfect construction of G_j , it holds that $X_j \perp X_{-j}|G_j$ and also that $X_j \perp G_j|X_{-j}$ (as G_j is a compression of X_{-j}). Therefore

$$P(X_j|X_{-j}) = P(X_j|X_{-j}, G_j) = P(X_j|G_j). \quad (8)$$

When we sample the replacement $\tilde{x}_j^{(i)}$ for an $x_j^{(i)}$ from the marginal within a group ($P(X_j|G_j = g_j^{(i)})$, e.g., via permutation) we also sample from the conditional $P(X_j|X_{-j} = x_{-j}^{(i)})$. Every data point from the global sample can therefore equivalently be seen as a sample from the marginal within the group, or as a sample from the global conditional distribution. As follows, the weighted sum of marginal subgroup PFIs coincides with the conditional PFI (cPFI).

$$cPFI = \sum_{i=1}^n \frac{1}{n} \left(L(f(\tilde{x}_j^{(i)}, x_{-j}^{(i)}), y^{(i)}) - L(\hat{f}(x_j^{(i)}, x_{-j}^{(i)}), y^{(i)}) \right) \quad (9)$$

$$= \sum_{k=1}^K \frac{n^k}{n} \sum_{i \in \mathcal{G}_k} \frac{1}{n^k} \left(L(f(\tilde{x}_j^{(i)}, x_{-j}^{(i)}), y^{(i)}) - L(\hat{f}(x_j^{(i)}, x_{-j}^{(i)}), y^{(i)}) \right) \quad (10)$$

$$(11)$$

$$= \sum_{k=1}^K \frac{n^k}{n} PFI^k \quad (12)$$

B Expectation and Variance of the PFI in a Subgroup

We show that under feature independence the PFI and a PFI in an arbitrary subgroup have the same expected value and the subgroup k PFI has a higher variance. Let $\tilde{L}^{(i)} = \frac{1}{M} \sum_{m=1}^M L(y^{(i)}, \hat{f}(\tilde{x}_j^{m(i)}, x_{-j}^{(i)}))$ and $L^{(i)} = L(y^{(i)}, \hat{f}(x_j^{m(i)}, x_{-j}^{(i)}))$.

Proof

$$\begin{aligned} \mathbb{E}_{X_{-j}}[PFI_j] &= \mathbb{E}_{X_{-j}} \left[\frac{1}{n} \sum_{i=1}^n (\tilde{L}^{(i)} - L^{(i)}) \right] \\ &= \mathbb{E}_{X_{-j}} [\tilde{L}^{(i)} - L^{(i)}] \\ \mathbb{E}[PFI_j^k]_{X_{-j}} &= \mathbb{E}_{X_{-j}} \left[\frac{1}{n_k} \sum_{i: x^{(i)} \in \mathcal{G}_j^k} (\tilde{L}^{(i)} - L^{(i)}) \right] \\ &= \frac{1}{n_k} \mathbb{E}_{X_{-j}} \left[\sum_{i: x^{(i)} \in \mathcal{G}_j^k} (\tilde{L}^{(i)} - L^{(i)}) \right] \\ &= \frac{1}{n_k} n_k \mathbb{E}_{X_{-j}} [\tilde{L}^{(i)} - L^{(i)}] \\ &= \mathbb{E}_{X_{-j}}[PFI_j] \end{aligned}$$

$$\begin{aligned}
\mathbb{V}_{X_{-j}}[PFI_j] &= \mathbb{V}_{X_{-j}} \left[\frac{1}{n} \sum_{i=1}^n (\tilde{L}^{(i)} - L^{(i)}) \right] \\
&= \frac{1}{n^2} n \mathbb{V}_{X_{-j}} [\tilde{L}^{(i)} - L^{(i)}] \\
&= \frac{1}{n} \mathbb{V}_{X_{-j}} [\tilde{L}^{(i)} - L^{(i)}] \\
\mathbb{V}_{X_{-j}}[PFI_j^k] &= \mathbb{V}_{X_{-j}} \left[\frac{1}{n_k} \sum_{i=1}^{n_k} (\tilde{L}^{(i)} - L^{(i)}) \right] \\
&= \frac{1}{n_k^2} n_k \mathbb{V}_{X_{-j}} [\tilde{L}^{(i)} - L^{(i)}] \\
&= \frac{1}{n_k} \mathbb{V}_{X_{-j}} [\tilde{L}^{(i)} - L^{(i)}] \\
\frac{\mathbb{V}_{X_{-j}}[PFI_j^k]}{\mathbb{V}_{X_{-j}}[PFI_j]} &= \frac{n}{n_k}
\end{aligned}$$

C Expectation and Variance of the PDP in a Subgroup

We show that under feature independence the PDP and a PDP in an arbitrary subgroup have the same expected value and the subgroup k PDP has a higher variance.

Proof

$$\begin{aligned}
\mathbb{E}_{X_{-j}}[PDP_j(x)] &= \mathbb{E}_{X_{-j}} [\hat{f}(x, X_{-j})] \\
\mathbb{E}_{X_{-j}}[PDP_j^k(x)] &= \mathbb{E}_{X_{-j}} \left[\frac{1}{n_k} \sum_{i=1}^{n_k} \hat{f}(x, x_{-j}^{(i)}) \right] = \frac{1}{n_k} n_k \mathbb{E}_{X_{-j}} [\hat{f}(x, X_{-j})] = \\
&= \mathbb{E}_{X_{-j}} [\hat{f}(x, X_{-j})] \\
\mathbb{V}_{X_{-j}}[PDP_j(x)] &= \mathbb{V}_{X_{-j}} \left[\frac{1}{n} \sum_{i=1}^n \hat{f}(x, x_{-j}^{(i)}) \right] \\
&= \frac{1}{n^2} n \mathbb{V}_{X_{-j}} [\hat{f}(x, X_{-j})] \\
&= \frac{1}{n} \mathbb{V}_{X_{-j}} [\hat{f}(x, X_{-j})] \\
\mathbb{V}_{X_{-j}}[PDP_j^k(x)] &= \mathbb{V}_{X_{-j}} \left[\frac{1}{n_k} \sum_{i=1}^{n_k} \hat{f}(x, x_{-j}^{(i)}) \right] \\
&= \frac{1}{n_k^2} n_k \mathbb{V}_{X_{-j}} [\hat{f}(x, X_{-j})] \\
&= \frac{1}{n_k} \mathbb{V}_{X_{-j}} [\hat{f}(x, X_{-j})] \\
\frac{\mathbb{V}_{X_{-j}}[PDP_j^k(x)]}{\mathbb{V}_{X_{-j}}[PDP_j(x)]} &= \frac{n}{n_k}
\end{aligned}$$

8. Model-agnostic Feature Importance and Effects with Dependent Features - A Conditional Subgroup Approach

Importance and Effects with Dependent Features

33

Table 7 MSE comparing estimated and true conditional PFI (for random forest, scenario II). Legend: impute rf: Imputation with a random forest, ko: Model-X knockoffs, mPFI: (marginal) PFI, tree cart: cs-permutation based on CART, tree trtr: cs-permutation based on transformation trees, CVIRF: conditional variable importance for random forests.

setting	cs-PFI (cart)	cs-PFI (trtr)	cvirf	impute rf	ko	mPFI
independent						
n=300, p=10	0.26	0.28	0.22	0.27	0.25	0.27
n=300, p=90	0.19	0.17	0.14	0.18	0.19	0.17
n=3000, p=10	0.07	0.07	1.39	0.07	0.06	0.08
n=3000, p=90	0.08	0.08	1.37	0.08	0.08	0.08
linear						
n=300, p=10	1.79	1.69	0.45	1.87	1.10	7.11
n=300, p=90	1.93	1.88	1.36	4.25	2.93	7.06
n=3000, p=10	0.29	0.22	5.41	0.25	0.40	6.80
n=3000, p=90	0.32	0.24	6.98	1.66	0.26	7.02
multi. lin.						
n=300, p=10	667.79	744.48	275.58	335.40	377.35	726.15
n=300, p=90	972.42	1098.74	301.26	823.89	1473.67	1065.26
n=3000, p=10	715.41	625.99	1790.45	114.71	454.26	1017.53
n=3000, p=90	974.37	945.19	5090.09	532.44	110.94	1416.30
non-linear						
n=300, p=10	1.40	1.29	1.37	3.96	12.35	18.51
n=300, p=90	1.06	1.03	2.05	6.77	2.38	12.32
n=3000, p=10	0.17	0.16	6.53	1.55	15.29	17.56
n=3000, p=90	0.15	0.14	9.09	3.28	8.00	11.30

D cPFI Ground Truth Scenario II

This chapter contains the results for the conditional PFI ground truth simulation, scenario II with an intermediate random forest.

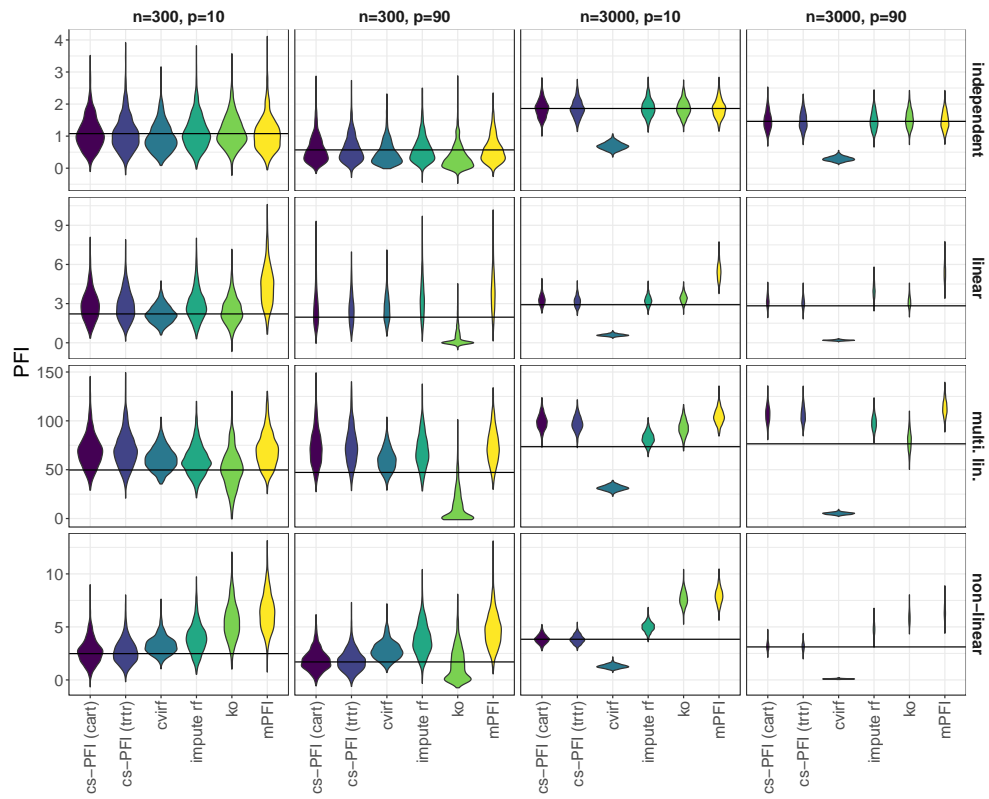


Fig. 14 Experiment (II) comparing various conditional PFI approaches with an intermediary random forest against the true conditional PFI based on the data generating process.

8. Model-agnostic Feature Importance and Effects with Dependent Features - A Conditional Subgroup Approach

Importance and Effects with Dependent Features

35

E Data Fidelity on OpenML-CC18 data sets

An overview of data sets from the OpenML-CC18 benchmarking suit. We used a subset of 42 out of 72 data sets with 7 to 500 continuous features.

OpenML ID	Name	No. Obs.	No. numerical feat.	No. feat.
1049	pc4	1458	38	38
1050	pc3	1563	38	38
1053	jml	10880	22	22
1063	kc2	522	22	22
1067	kc1	2109	22	22
1068	pc1	1109	22	22
12	mfeat-factors	2000	217	217
14	mfeat-fourier	2000	77	77
1461	bank-marketing	45211	8	17
1475	first-order-theorem-proving	6118	52	52
1480	ilpd	583	10	11
1486	nomao	34465	90	119
1487	ozone-level-8hr	2534	73	73
1494	qsar-biodeg	1055	42	42
1497	wall-robot-navigation	5456	25	25
15	breast-w	683	10	10
1501	semeion	1593	257	257
151	electricity	45312	8	9
1510	wdbc	569	31	31
16	mfeat-karhunen	2000	65	65
182	satimage	6430	37	37
188	eucalyptus	641	15	20
22	mfeat-zernike	2000	48	48
23517	numerai28.6	96320	22	22
28	optdigits	5620	63	65
307	vowel	990	11	13
31	credit-g	1000	8	21
32	pendigits	10992	17	17
37	diabetes	768	9	9
40499	texture	5500	41	41
40701	churn	5000	17	21
40966	MiceProtein	552	78	82
40979	mfeat-pixel	2000	241	241
40982	steel-plates-fault	1941	28	28
40984	segment	2310	19	20
40994	climate-model-simulation-crashes	540	21	21
44	spambase	4601	58	58
4538	GesturePhaseSegmentationProcessed	9873	33	33
458	analcata_data_authorship	841	71	71
54	vehicle	846	19	19
6	letter	20000	17	17
6332	cylinder-bands	378	19	40

Table 8 Overview of OpenML CC18 data sets used for the data fidelity experiment.

E.1 Data Fidelity Results

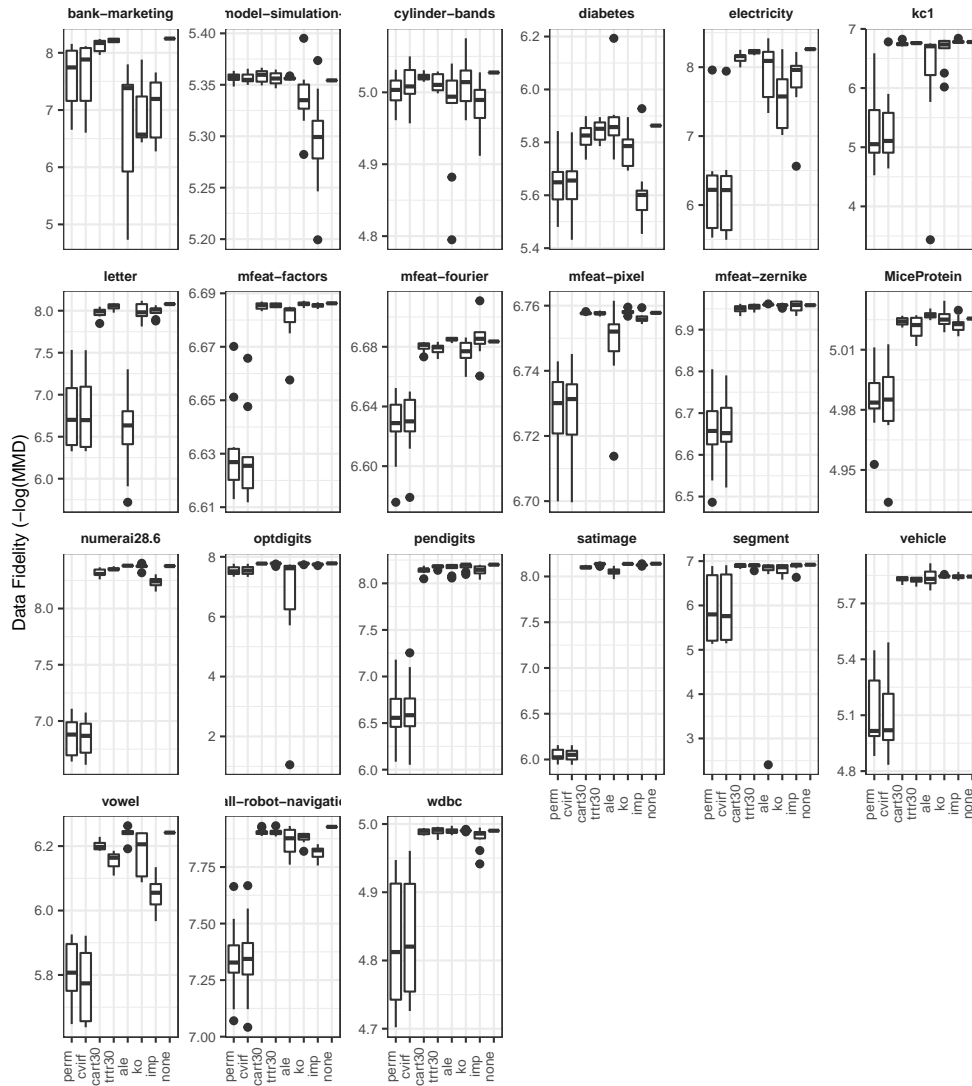


Fig. 15 Data Fidelity experiment with OpenML-CC18 data sets (1/2). Different sampling types are compared: unconditional permutation (perm), cs-permutation (maximal tree depth) with CART (cart30) or transformation trees (trtr30), Model-X knockoffs (ko), data imputation with a random forest (imp), ALE (ale), conditional variable importance for random forests (cvirf) and no permutation (none). Each data point in the boxplot represents one feature and one data set. Results from repeated experiments have been averaged (mean) before using them in the boxplots.

8. Model-agnostic Feature Importance and Effects with Dependent Features - A Conditional Subgroup Approach

Importance and Effects with Dependent Features

37

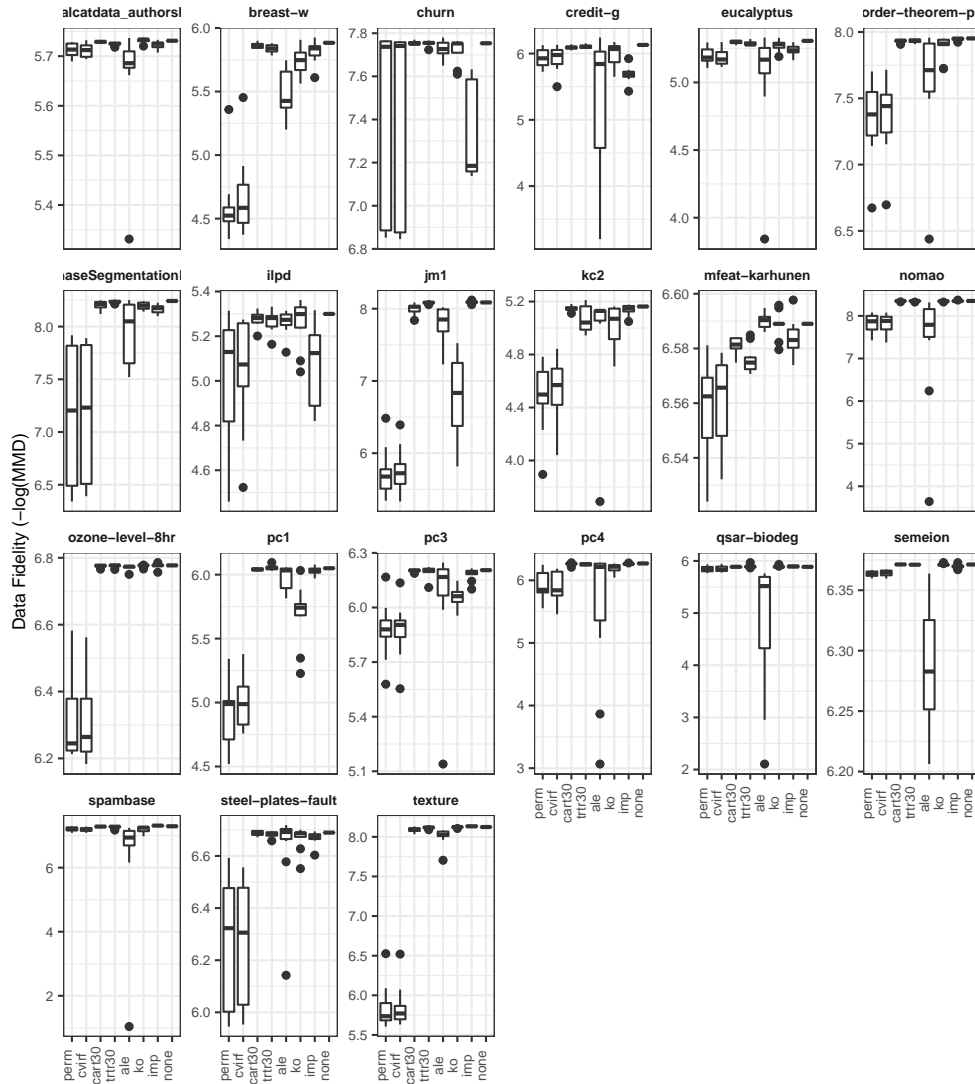


Fig. 16 Data Fidelity experiment with OpenML-CC18 data sets (2/2). Different sampling types are compared: unconditional permutation (perm), cs-permutation (maximal tree depth) with CART (cart30) or transformation trees (trtr30), Model-X knockoffs (ko), data imputation with a random forest (imp), ALE (ale), conditional variable importance for random forests (cvirf) and no permutation (none). Each data point in the boxplot represents one feature and one data set. Results from repeated experiments have been averaged (mean) before using them in the boxplots.

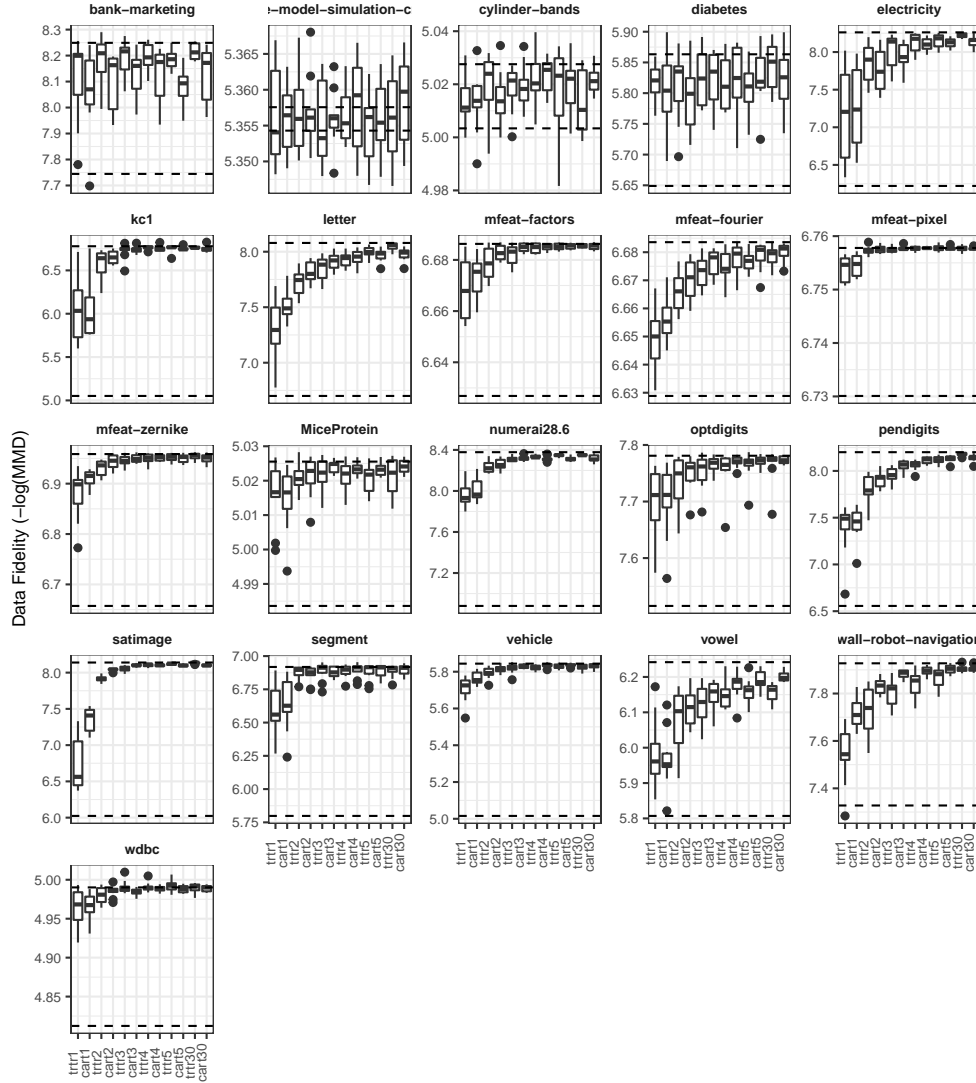


Fig. 17 Data Fidelity experiment with OpenML-CC18 data sets (1/2). Different tree depths and tree types (CART and Transformation Trees) are compared. Unconditional permutation and lack of permutation serve as lower and upper bound for data fidelity and their median data fidelity is plotted as dotted lines. Each data point in the boxplot represents one feature and one data set. Results from repeated experiments have been averaged (mean) before using them in the boxplots.

8. Model-agnostic Feature Importance and Effects with Dependent Features - A Conditional Subgroup Approach

Importance and Effects with Dependent Features

39

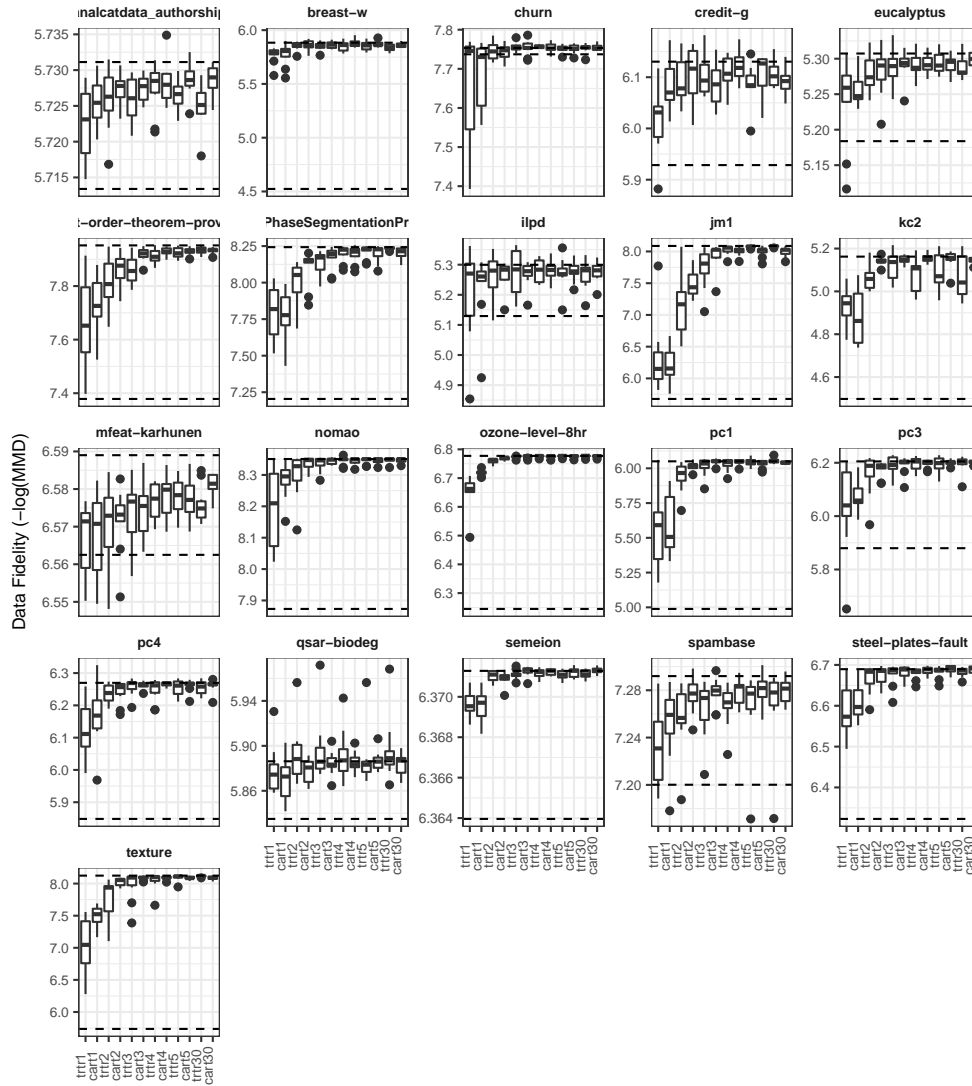


Fig. 18 Data Fidelity experiment with OpenML-CC18 data sets (2/2). Different tree depths and tree types (CART and Transformation Trees) are compared. Unconditional permutation and lack of permutation serve as lower and upper bound for data fidelity and their median data fidelity is plotted as dotted lines. Each data point in the boxplot represents one feature and one data set. Results from repeated experiments have been averaged (mean) before using them in the boxplots.

F Model Fidelity Plots

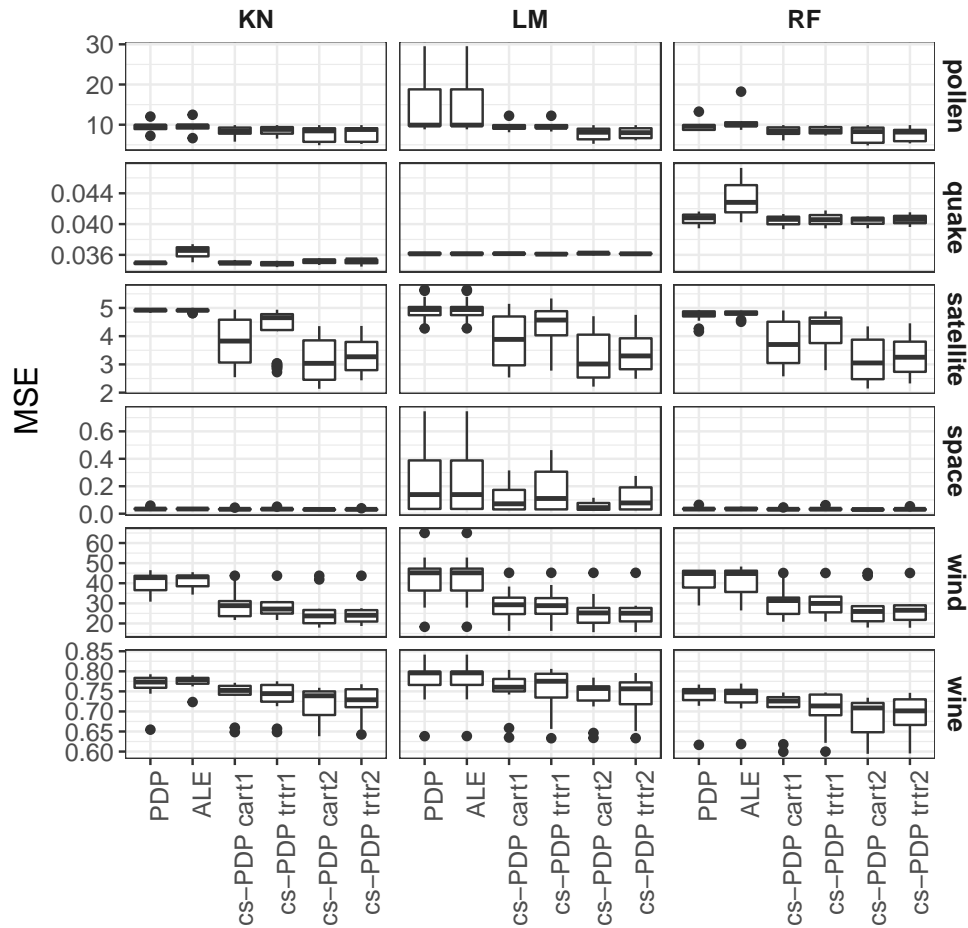


Fig. 19 Comparing the loss between model f and various feature effect methods. Each instance in the boxplot is MSE for one feature, summed over the test data.

9. Relating the Partial Dependence Plot and Permutation Feature Importance to the Data Generating Process

Contributing article:

Molnar, C., König, G., Freiesleben, T., Wright, M., Casalicchio, G., and Bischl, B. (2021). Relating the Partial Dependence Plot and Permutation Feature Importance to the Data Generating Process. arxiv preprint arXiv:2109.01433.

Copyright information:

Creative Commons Attribution 4.0 International (CC BY 4.0)

Author contributions:

Christoph Molnar wrote most of the paper. Timo Freiesleben and Gunnar König developed some of the proofs. Marvin Wright helped implement the simulation study. All authors added input, suggested modifications, proofread and revised the paper.

Relating the Partial Dependence Plot and Permutation Feature Importance to the Data Generating Process

Christoph Molnar · Timo Freiesleben ·
Gunnar König · Giuseppe Casalicchio ·
Marvin N. Wright · Bernd Bischl

Abstract Scientists and practitioners increasingly rely on machine learning to model data and draw conclusions. Compared to statistical modeling approaches, machine learning makes fewer explicit assumptions about data structures, such as linearity. However, their model parameters usually cannot be easily related to the data generating process. To learn about the modeled relationships, partial dependence (PD) plots and permutation feature importance (PFI) are often used as interpretation methods. However, PD and PFI lack a theory that relates them to the data generating process. We formalize PD and PFI as statistical estimators of ground truth estimands rooted in the data generating process. We show that PD and PFI estimates deviate from this ground truth due to statistical biases, model variance and Monte Carlo approximation errors. To account for model variance in PD and PFI estimation, we propose the learner-PD and the learner-PFI based on model refits, and propose corrected variance and confidence interval estimators.

Keywords Interpretable Machine Learning, Explainable AI, Permutation Feature Importance, Partial Dependence Plot, Statistical Inference, Uncertainty Quantification

This project is funded by the Bavarian State Ministry of Science and the Arts and coordinated by the Bavarian Research Institute for Digital Transformation (bidt) and supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A, by the German Research Foundation (DFG) – Emmy Noether Grant 437611051 to MNW, and by the Graduate School of Systemic Neurosciences (GSN) Munich. The authors of this work take full responsibilities for its content.

C. Molnar, T. Freiesleben, G. König, G. Casalicchio, B. Bischl
Ludwig-Maximilian University Munich, Germany

G. König
University of Vienna, Austria

C. Molnar, M. Wright
Leibniz Institute for Prevention Research and Epidemiology – BIPS, Bremen, Germany

M. Wright
University of Bremen, Germany

1 Introduction

Statistical models such as linear or logistic regression models are frequently used to learn about relationships in data. Assuming that a statistical model reflects the data generating process (DGP) well, we may interpret the model coefficients in place of the DGP and draw conclusions about the data. An important part of interpreting the coefficients is the quantification of their uncertainty via standard errors, which allows to separate random noise (non-significant coefficients) from real effects. Statistical biases and violation of assumptions are well studied for many model classes, such as heterogeneous residuals, deviations from normality, and non-additivity for linear models (Fahrmeir et al., 2007).

Increasingly, machine learning approaches such as gradient-boosted trees, random forests or neural networks are used instead of or in addition to statistical models. Compared to statistical models that are driven by considerations of the data generating process, the machine learning approaches often lack a mapping between model parameters and properties of the DGP. Due to the ability of many machine learning models to address highly non-linear relationships and interactions, they often outperform more restrictive statistical models. Scientific applications of machine learning are widespread and range from modeling volunteer labor supply (Bair et al., 2013), mapping fish biomass (Esselman et al., 2015), analyzing urban reservoirs (Obringer and Nateghi, 2018), identifying disease-associated genetic variants (Boulesteix et al., 2020), and inferring behavior from smartphone use (Stachl et al., 2020). In these scientific applications, the model is only the means to an end: a better understanding of the data generating process, in particular the conditional expectation of the target variables as a function of the features.

Model-agnostic interpretation methods (Ribeiro et al., 2016) are a (partial) remedy to the lack of interpretable parameters of more complex models. Model-agnostic methods follow a general procedure of 1) sampling data, 2) manipulating this data, 3) predicting and 4) aggregating the predictions (Scholbeck et al., 2019). Since none of these steps depend on specific model properties, model-agnostic interpretation techniques allow us to study the behavior of arbitrary models. Partial dependence (PD) plots (Friedman, 1991) and permutation feature importance (PFI) (Breiman, 2001; Fisher et al., 2019) are popular model-agnostic methods for describing the relationship between input features and model outcome on a global level. PD plots visualize the average effect features have on the prediction, and PFI estimates how much each feature improves the model performance and therefore how relevant a feature is. However, PD and PFI merely describe the prediction (or classification) function, but lack a theory that connects them to the data generating process. Treating PD and PFI as statistical estimators (like coefficients in a regression model) would require a theoretical counterpart in the DGP: a ground truth estimand that these interpretation methods are supposed to retrieve. Furthermore, for proper inference about the DGP, we need to quantify the uncertainty of PD and PFI estimators. Linear regression models, for exam-

ple, provide variance estimates for the coefficients, which help to distinguish true effects from randomness and allow confidence interval estimation and hypothesis testing. Most machine learning approaches, however, do not provide variance estimates for their predictions or model parameters. Yet, the training process itself can be a relevant source of variance as the trained model heavily depends on the specific training data.

We propose to treat PD and PFI as statistical estimators of a ground truth, which allows us to relate the model interpretation to the data generating process. In Section 2, we introduce related work and in Section 3 we introduce notation and background on PD and PFI. In Section 4, we formulate PD and PFI as estimators of (proposed) ground truth estimands in the DGP. By treating PD and PFI as statistical estimators, we can apply the bias and variance decomposition and identify the different sources of uncertainty. To reflect the different uncertainty sources, we distinguish between model-PD/PFI and learner-PD/PFI. The model-PD/PFI (Section 6) follows the standards definitions of PD and PFI. We propose confidence intervals and variance estimators for model-PD/PFI and show that they neglect the model variance originating from the training process. In Section 7, we propose the learner-PD and learner-PFI which take the model variance into account, study their statistical biases and propose variance estimators and confidence intervals. For models that lack variance estimates, multiple model refits are required to capture the variance due to the learning process. Data size is often a limiting factor, so that model refits are based on resampled data with overlapping observations. This overlap can lead to an underestimation of variance and thus to confidence intervals that are too narrow. We leverage a variance correction approach from model performance estimation to improve the variance estimation. In Section 8, we analyze the coverage of the confidence intervals for learner-PD and learner-PFI with and without the correction. In the application in Section 9 we demonstrate the use of confidence intervals for PD and PFI and illustrate the importance of taking the model variance into account.

2 Related Work

For PD plots, model-specific confidence intervals exist that rely on models with inherent variance estimators such as Bayesian additive regression trees (Cafri and Bailey, 2016; Zhao and Hastie, 2021). Furthermore, various applied articles contain computations of PD confidence bands (Bair et al., 2013; Grange and Carslaw, 2019; Esselman et al., 2015; Emrich and Pierdzioch, 2016; Page et al., 2018; Obringer and Nateghi, 2018). These approaches either quantify only the error due to Monte Carlo approximation or, when they cover model variance, they do not account for underestimation of the variance. This demonstrates the need for a theoretical underpinning of this inferential tool for practical research. For PFI and related approaches, multiple suggestions for confidence intervals and variance estimation are available. Some contributions are specific to the random forest PFI (Ishwaran and Lu, 2019; Archer and Kimes, 2008;

Janitz et al., 2018), for which a test for null importance was proposed by Altmann et al. (2010).

Model-agnostic PFI confidence intervals that are similar to ours are proposed by Watson and Wright (2019); Williamson et al. (2019, 2020). We additionally correct for variance underestimation arising from resampling (Nadeau and Bengio, 2003) and relate the estimators to the proposed ground truth PFI. An alternative approach for providing bounds on PFI is proposed by Fisher et al. (2019) via Rashomon sets, which are sets of models with similar near-optimal prediction accuracy. Furthermore, alternative approaches of “model-free” inference exist (Parr et al., 2020; Parr and Wilson, 2019; Zhang and Janson, 2020), which aim to infer properties of the data without an intermediary ML model.

3 Background and Notation

We denote the joint distribution induced by the data generating process as \mathbb{P}_{XY} , where X is a p -dimensional random variable and Y a 1-dimensional random variable. We describe the true mapping from features X to the target Y with $f(X) = \mathbb{E}[Y|X = x]$. We denote a single random draw from the DGP with $x^{(i)}$ and $y^{(i)}$. A dataset consisting of multiple draws from \mathbb{P}_{XY} will be called $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$, where n is the number of samples and with each $(x^{(i)}, y^{(i)}) \sim \mathbb{P}_{XY}$, $i \in \{1, \dots, n\}$. An ML model \hat{f} is a function ($\hat{f} : \mathcal{X} \rightarrow \mathcal{Y}$) that maps a feature vector to a prediction (e.g. $\mathcal{Y} = \mathbb{R}$ for regression). The model \hat{f} is induced based on a dataset \mathcal{D}_n , using a loss function $L : \mathcal{Y} \times \mathbb{R}^p \rightarrow \mathbb{R}_0^+$. As the true function f is unknown, the model \hat{f} is interpreted instead of f , for example, with PD plots and PFI. The model \hat{f} is learned by an ML learner $I : \mathcal{D} \times \Lambda \rightarrow \mathcal{H}$ that maps from the space of datasets and the space of hyperparameters Λ to the function hypothesis space \mathcal{H} . The learning process contains two sources of randomness: the training data being a random sample from \mathbb{P}_{XY} and (possibly) the inherent randomness of the training process (Bouthillier et al., 2021).¹ Thus, a model \hat{f} can be seen as realization of a random variable F with distribution \mathbb{P}_F . We assume that the model is evaluated with a risk function $\mathcal{R}(\hat{f}) = \mathbb{E}_{XY}[L(Y, \hat{f}(X))] = \int L(y, \hat{f}(x)) d\mathbb{P}_{XY}$, based on a loss function L . To get unbiased estimates of the risk, model training and evaluation use different datasets. The dataset \mathcal{D}_n is split into \mathcal{D}_{n_1} for model training and \mathcal{D}_{n_2} for evaluation, with $n_1 + n_2 = n$. The empirical risk is estimated with $\hat{\mathcal{R}}(\hat{f}_{\mathcal{D}_{n_2}, \lambda}) := \frac{1}{n_2} \sum_{i=1}^{n_2} L(y^{(i)}, \hat{f}_{\mathcal{D}_{n_2}, \lambda}(x^{(i)}))$.

We distinguish between the “simulation” and the “real world” scenario (Hothorn et al., 2005). In the simulation scenario, we can generate a quasi-infinite number of datasets, which allows us to refit the model multiple times using fresh data each time. In the real world setting, we assume that a single dataset of size n is available. To fit multiple models (of the same class) and to

¹ For example, stochastic gradient descent and weight initialization in neural networks or bootstrap and feature sampling in random forests are sources of randomness.

obtain multiple estimates of the risk, resampling techniques such as bootstrapping, cross-validation and repeated subsampling have to be used. We denote by B_d the set of indices for the training data in the d -th split repetition and with B_{-d} the corresponding test data indices, where $B_d \cup B_{-d} = \{1, \dots, n\}$, $b \in \{1, \dots, m\}$, and m is the number of models trained with different data.

We distinguish between the interpretation of a single model and the distribution of models produced by a learner. Often a fixed trained model \hat{f} is the subject of interpretation. Any interpretation of a fixed model neglects the model variance originating from the learning process. Often we are interested in extending the interpretation to the distribution of models produced by a learner. For example, the importance of a feature in a decision tree might be zero because it was never selected for a split. However, if we were to train the tree on a slightly different sample from the same distribution, it might obtain a non-zero importance. A similar distinction between model and learner can be made for performance estimation, where model performance is estimated with a test set, but learner performance requires averaging performance over m repetitions and thus model refits.

3.1 Partial Dependence (PD)

The partial dependence function (Friedman, 1991) of a model \hat{f} describes the expected effect of a feature after marginalizing out the effects of all other features. Partial dependence of a feature set X_S , $S \subseteq \{1, \dots, p\}$ (usually $|S| = 1$) is defined as:

$$PD_S = \mathbb{E}_{X_C}[\hat{f}(x, X_C)], \quad (1)$$

where X_C are the remaining features so that $S \cup C = \{1, \dots, p\}$ and $S \cap C = \emptyset$. The PD is estimated using Monte Carlo integration:

$$\widehat{PD}_S(x) = \frac{1}{n_2} \sum_{i=1}^{n_2} \hat{f}(x, x_C^{(i)}) \quad (2)$$

For simplicity, we write PD instead of PD_S , and \widehat{PD} instead of \widehat{PD}_S when we refer to an arbitrary PD. The PD plot consists of a line connecting the points $\{(x^{(g)}, \widehat{PD}_S(x^{(g)}))\}_{g=1}^G$, with G grid points that are usually equidistant or quantiles of \mathbb{P}_{X_S} . See Figure 6 for an example of a PD plot.

3.2 Permutation Feature Importance (PFI)

The PFI (Breiman, 2001; Fisher et al., 2019) of a model \hat{f} is defined as the increase in loss L when the feature set X_S (usually just one feature) is permuted:

$$PFI_S = \mathbb{E}_{\tilde{X}_S X_C Y} [L(Y, \hat{f}(\tilde{X}_S, X_C))] - \mathbb{E}_{XY} [L(Y, \hat{f}(X))], \quad (3)$$

where \tilde{X}_S is a random variable based on the distribution of X_S . There are two versions of PFI, the marginal PFI and the conditional PFI, which have different strategies to replace X_S and also different interpretations. The marginal PFI can be interpreted as the importance of the feature, ignoring dependencies with other features and also ignoring that the data used may differ greatly from the original joint distribution \mathbb{P}_X (extrapolation). For the marginal PFI we take the expected value over the distribution $\mathbb{P}_{X_S} \cdot \mathbb{P}_{X_C Y}$, which means that \tilde{X}_S follows the marginal distribution of X_S and is independent of X_C and Y ($\tilde{X}_S \perp\!\!\!\perp X_C, Y$). This means that the marginal PFI breaks the association between the feature(s) X_S and the target Y , but also between X_S and all other features X_C . For the conditional PFI (cPFI) (Molnar et al., 2020; Watson and Wright, 2019; Hooker and Mentch, 2019; Candès et al., 2018), the expectation is taken over the distribution $\mathbb{P}_{X_S|X_C} \cdot \mathbb{P}_{X_C Y}$, so that \tilde{X}_S follows the conditional distribution of X_S given X_C but is still independent of Y . The interpretation of the conditional PFI of a feature is therefore also conditional on all features that are correlated with the feature of interest. Conditional PFI may be interpreted as the *additional* importance of a feature *given that we already know the other feature values*.

PFI and cPFI are estimated with Monte Carlo integration:

$$\widehat{PFI}_S = \frac{1}{n} \sum_{i=1}^{n_2} \left(\frac{1}{l} \sum_{k=1}^l L(y^{(i)}, \hat{f}(\tilde{x}_S^{(k,i)}, x_C^{(i)})) - L(y^{(i)}, \hat{f}(x^{(i)})) \right), \quad (4)$$

where $\tilde{x}_S^{(k,i)}$ with $k \in \{1, \dots, l\}$ is the k -th sample of x_S for the i -th observation. For the marginal PFI, $\tilde{x}_S^{(k,i)}$ can be a permutation of the original vector x_S . The conditional PFI requires a conditional sampling mechanism for the feature, such as subgroups (Molnar et al., 2020) or knockoffs (Candès et al., 2018; Watson and Wright, 2019). The estimation of \widehat{PFI} requires unseen data, so that the loss estimates deliver unbiased results (Zheng and van der Laan, 2011; Chernozhukov et al., 2018). If not stated otherwise, mathematical derivations in this paper apply to both marginal and conditional PFI. We assume that the loss used for PFI can be computed per instance, which excludes losses such as AUC. See Figure 6 for a PFI example. As with PD, we use PFI instead of PFI_S and \widehat{PFI} instead of \widehat{PFI}_{Ss} .

4 Relating Model to Data Generating Process

The goal of statistical inference is to gain knowledge about the DGP. Therefore, the modeler aims to establish relationships between properties of the model and the DGP. For example, under certain assumptions, the coefficients of a generalized linear model (= model properties) can be related to parameters of the respective conditional distribution defined by the DGP, such as conditional mean and covariance structure (= DGP properties). Machine learning models such as random forests or neural networks lack such a mapping between learned

model parameters and properties of the data generating process. This lack of counterparts in the DGP make it difficult to interpret complex machine learning models and to draw conclusions about the real world. Interpretation methods such as PD and PFI provide **external descriptors** of how features affect the model predictions. However, PD and PFI are estimators that lack a counterpart estimand in the DGP. We propose an inference approach for these external descriptors. We define a ground truth version of PD and PFI directly on the DGP, namely the DGP-PD and the DGP-PFI. The DGP-PD and the DGP-PFI are defined as the PD and PFI, but applied to the true function f instead of \hat{f} . This means that the DGP-PD becomes the feature effect of features X_S on the underlying function f :

Definition 1 (DGP-PD) The DGP-PD is the PD applied to function $f : \mathcal{X} \mapsto \mathcal{Y}$ of the data generating process.

$$\text{DGP-PD}(x) = \mathbb{E}_{X_C}[f(x, X_C)]$$

Similarly, for the DGP-PFI we replace \hat{f} for f and compute the expected losses. We compute the difference between the loss for the permuted distribution and the loss on the joint distribution. Since we work with the true f , the “original” loss is the aleatoric uncertainty (without any bias or variance).

Definition 2 (DGP-PFI) The DGP-PFI is the PFI applied to function $f : \mathcal{X} \mapsto \mathcal{Y}$ of the data generating process.

$$\text{DGP-PFI} = \mathbb{E}_{\tilde{X}_S X_C Y}[L(Y, f(\tilde{X}_S, X_C))] - \mathbb{E}_{XY}[L(Y, f(X))]$$

The function f is usually unknown. If it were known in an application, we would not need machine learning in the first place. However, Definitions 1 and 2 immediately enable at least two useful applications: It allows scientists to compare the PD/PFI of a model with the PD/PFI of the DGP **in simulation studies** and research statistical biases. More importantly, the ground truth definitions of DGP-PD and DGP-PFI allow us to treat PD and PFI as statistical estimators of properties of the data generating process.

This paper studies PD and PFI as statistical estimators of the ground truth DGP-PD and DGP-PFI, including bias and variance decompositions, and confidence interval estimators. Whether the estimands themselves are desirable in specific data scenarios and model choices is out-of-scope for this work. Others have done work in limitations of PFI and PD: For example Molnar et al. (2020); Hooker and Mentch (2019); Strobl et al. (2008) show that interpretation methods produce misleading results under strongly dependent features (e.g. large correlation between features), Zhao and Hastie (2021) assess whether PDs can be used to estimate causal effects, and Groemping (2020) studied whether PDs recover the linear relationship of the DGP when the relationship between target and features is linear. Extrapolation when features are dependent might be one of the biggest issue for PD and PFI. As one possible remedy, conditional variants of PDP and PFI (Molnar et al., 2020; Fisher

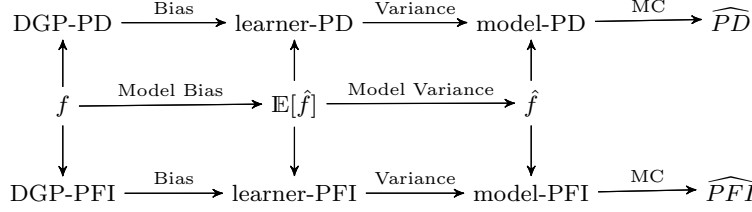


Figure 1 A model \hat{f} deviates from f due to model bias and variance. Similarly \widehat{PD} and \widehat{PFI} estimates deviate from their ground truth versions DGP-PD and DGP-PFI due to bias, variance, and Monte Carlo integration (MC).

et al., 2019; Watson and Wright, 2019; Apley and Zhu, 2020) have been proposed. For PD, the conditional variant is also called M-Plot (Apley and Zhu, 2020) and weights predictions according to how likely their respective feature values are for a given PD grid point. Our proposed variance and confidence interval estimators and other results apply to both the original and conditional variants of PD and PFI, if not stated otherwise.

5 Bias-Variance Decomposition

The definition of DGP-PD and DGP-PFI gives us a ground truth to which the PD and PFI of a model can be compared – at least in theory and simulation. The error of the estimation (mean squared error between estimator and estimand) can be decomposed into the systematic deviation from the true estimand (statistical bias) and the variance due to model variance. PD and PFI are both expectations over the – usually unknown – joint distribution of the data. The expectations are therefore usually estimated from data using Monte Carlo integration, which adds another source of variance to the PFI and PD estimates. Figure 1 visualizes the chain of errors that stand between the estimand (DGP-PD, DGP-PFI) and the estimates (\widehat{PD} , \widehat{PFI}).

For the PD, we compare the MSE between the true DGP-PD (PD_f as defined in Equation 1) with the theoretical PD of a model instance \hat{f} ($PD_{\hat{f}}$) at position x .

$$\mathbb{E}_F[(PD_f(x) - PD_{\hat{f}}(x))^2] = \underbrace{(PD_f(x) - \mathbb{E}_F[PD_{\hat{f}}(x)])^2}_{Bias^2} + \underbrace{\mathbb{V}_F[PD_{\hat{f}}(x)]}_{Variance}$$

Here, F is the distribution of the trained models, which can be treated as a random variable. The bias-variance decomposition of the MSE of estimators is a well known result (Geman et al., 1992). For completeness, we provide a proof in Appendix A. Figure 2 visualizes bias and variance of a PD curve, and the variance due to Monte Carlo integration.

Similarly, the MSE of the theoretical PFI of a model (Equation 3) can be decomposed into squared bias and variance. The proof can be found in

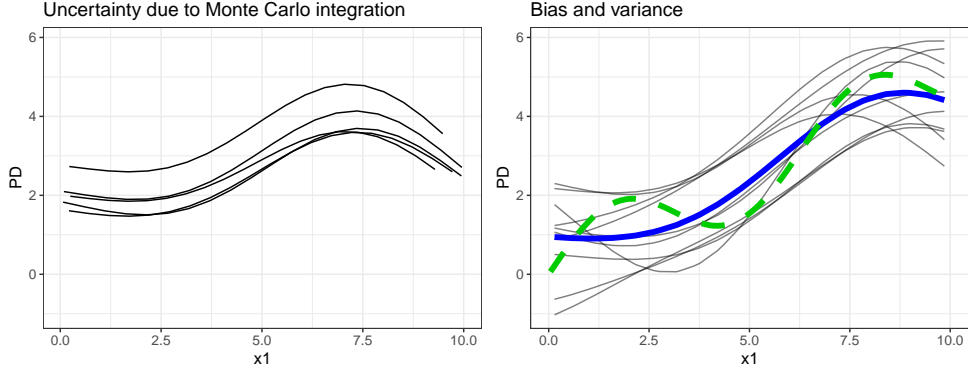


Figure 2 Illustration of bias, variance and Monte Carlo approximation for the PD. Left: Various PDPs using different data for the Monte Carlo integration, but keeping the model fixed. Right: The green dashed line shows the DGP-PD plot of a toy example. Each thin line is the PD plot for the model fitted with a different sample, and the thick blue line is the average thereof. Deviation of the expected PDP from the DGP-PDP are due to bias, deviations of the individual model-PDPs to the expected PDP are due to model variance.

Appendix B.

$$\mathbb{E}_F[(PFI_{\hat{f}} - PFI_f)^2] = Bias_F^2[PFI_{\hat{f}}] + \mathbb{V}_F[PFI_{\hat{f}}]$$

The model variance of PD/PFI stems from variance in the model fit, which depends on the training sample \mathcal{D} and on randomness in the model training such as weight initialization or feature and observation sampling. When constructing confidence intervals, we have to take into account the variance of PFI and PDP across model fits, and not just the error due to Monte Carlo integration. As we show in an application (Section 9), whether PD and PFI are based on a single model or are averaged across model refits can impact the interpretation, and especially the certainty of the interpretation. We therefore distinguish between model-PD/PFI and learner-PD/PFI, which are averaged over refitted models. Variance estimators for model-PD/PFI only account for variance due to Monte Carlo integration.

6 Model-PD and Model-PFI

In this section, we study the model-PD and the model-PFI, and provide variance and confidence interval estimators. With model-PD and model-PFI, we refer to the original proposals for PD (Friedman, 1991) and PFI (Breiman, 2001; Fisher et al., 2019) for fixed models. Conditioning on a given model \hat{f} ignores the model variance due to the learning process. Only the variance due to Monte Carlo integration can be considered in this case.

The model-PD estimator (Equation (2)) is unbiased regarding the theoretical model-PD (Equation (1)). Also, the estimated model-PFI (Equation (4)) is unbiased with respect to the theoretical model-PFI (Equation (3)). These

findings are general properties of Monte Carlo integration, which state that Monte Carlo integration converges to the integral due to the law of large numbers. Proofs can be found in Appendix C and E. In addition, model-PD and model-PFI are unbiased estimator of the DGP-PD (Theorem 1) and DGP-PFI (Theorem 2), under certain conditions.

To quantify the variance due to Monte Carlo integration and to construct confidence intervals, we calculate the variance across the test data instances. For the model-PD, the variance can be estimated with:

$$\widehat{\mathbb{V}}(\widehat{PD}(x)) = \frac{1}{n_2(n_2 - 1)} \sum_{i=1}^{n_2} \left(\hat{f}(x, x_C^{(i)}) - \widehat{PD}(x) \right)^2.$$

Similarly, for the model-PFI the variance is:

$$\widehat{\mathbb{V}}(\widehat{PFI}) = \frac{1}{n_2(n_2 - 1)} \sum_{i=1}^n \left(L^{(i)} - \widehat{PFI} \right)^2,$$

where $L^{(i)} = \frac{1}{l} \sum_{k=1}^l L(y^{(i)}, \hat{f}(\tilde{x}_S^{(k,i)}, x_C^{(i)})) - L(y^{(i)}, \hat{f}(x^{(i)}))$.

Model-PD and model-PFI are mean estimates of independent samples with estimated variance. As such, they follow a t-distribution with $n_2 - 1$ degrees of freedom. This allows us to construct point-wise confidence bands for the model-PD and confidence intervals for the model-PFI, that capture the Monte Carlo approximation uncertainty. We define point-wise α -confidence bands around the estimated model-PD:

$$CI_{\widehat{PD}(x)} = \left[\widehat{PD}(x) - t_{1-\frac{\alpha}{2}} \sqrt{\widehat{\mathbb{V}}(\widehat{PD}(x))}; \widehat{PD}(x) + t_{1-\frac{\alpha}{2}} \sqrt{\widehat{\mathbb{V}}(\widehat{PD}(x))} \right]. \quad (5)$$

where $t_{1-\frac{\alpha}{2}}$ is the $1 - \alpha/2$ quantile of the t-distribution with $n_2 - 1$ degrees of freedom. We proceed in the same manner for PFI:

$$CI_{\widehat{PFI}} = \left[\widehat{PFI} - t_{1-\frac{\alpha}{2}} \sqrt{\widehat{\mathbb{V}}(\widehat{PFI})}; \widehat{PFI} + t_{1-\frac{\alpha}{2}} \sqrt{\widehat{\mathbb{V}}(\widehat{PFI})} \right]. \quad (6)$$

Confidence intervals for model-PD and model-PFI ignore the model variance. The interpretation, therefore, is limited to variance regarding the Monte Carlo approximation, and we cannot generalize results to the data generating process. Model-PD/PFI and its confidence bands/intervals are applicable when the focus is a fixed model (e.g. in a model audit).

7 Learner-PD and Learner-PFI

To account for the model variance, we propose the learner-PD and the learner-PFI, which average the PD/PFI over m model fits $\hat{f}_d, d \in \{1, \dots, m\}$ produced by the same learning algorithm, but trained on different data samples. The

learner-variants are averages of the model-variants, where for each model-PD/PFI the model is repeatedly “sampled” from the distribution of models.

The learner-PD is therefore the expected PD over the distribution of models generated by the learning process: $\mathbb{E}_F[PD(x)]$. We estimate the learner-PD with:

$$\widehat{PD}(x) = \frac{1}{m} \sum_{d=1}^m \frac{1}{|B_{-d}|} \sum_{i \in B_{-d}} \hat{f}_d(x, x_C^{(i)}), \quad (7)$$

where \hat{f}_d is trained on sample indices B_d and the PD estimated using samples B_{-d} so that $B_d \cap B_{-d} = \emptyset$.

Following the PD, the learner-PFI is the expected PFI over the distribution of models produced by the learner: $\mathbb{E}_F[PFI]$. We propose the following estimator for the learner-PFI:

$$\widehat{PFI} = \frac{1}{m} \sum_{d=1}^m \frac{1}{|B_{-d}|} \sum_{i \in B_{-d}} \left(\tilde{L}_d^{(i)} - L_d^{(i)} \right), \quad (8)$$

where losses $L_d^{(i)} = L(y^{(i)}, \hat{f}_d(x^{(i)}))$ and $\tilde{L}_d^{(i)} = \frac{1}{l} \sum_{k=1}^l L(y^{(i)}, \hat{f}_d(\tilde{x}_S^{(k,i)}, x_C^{(i)}))$ are estimated with data B_{-d} for a model trained on data B_d . Marginal and conditional versions can also be distinguished for the learner-PFI, depending on how \tilde{X}_S was sampled. A similar estimator has been proposed by Janitza et al. (2018) for random forests.

7.1 Bias of Learner-PD

The learner-PD is an unbiased estimator of the expected PD over the distribution of models F , since $\mathbb{E}_F[\widehat{PD}(x)] = \mathbb{E}_F \left[\frac{1}{m} \sum_{d=1}^m \widehat{PD}_d(x) \right] = \frac{m}{m} \mathbb{E}_F[PD_{\hat{f}}(x)] = \mathbb{E}_F[PD_{\hat{f}}(x)]$. The bias of the learner-PD *regarding the DGP-PD* is linked to the bias of the model. If the ML model is unbiased, the PDs are unbiased as well.

Theorem 1 *Model unbiasedness implies PD unbiasedness:*

$$\mathbb{E}_F[\hat{f}(x)] = f(x) \implies \mathbb{E}_F[\mathbb{E}_{X_C}[\hat{f}]] = \mathbb{E}_{X_C}[f]$$

Proof Sketch 1 *Applying Fubini’s Theorem allows us to switch the order of integrals. Further replacing $\mathbb{E}_F[\hat{f}]$ with f proves the unbiasedness. A full proof can be found in Appendix D.*

By model bias, we refer to the deviation between the estimated \hat{f} and f . Inductive bias, i.e. the preference of one generalization over another, is necessary for learning (Mitchell, 1980). A wrong choice of inductive bias, such as assuming a linear \hat{f} for a non-linear f , leads to deviations of \hat{f} from f . But there are also other reasons why a bias of \hat{f} from f may occur, for example a too small training data size. We discuss the critical assumption of model unbiasedness further in Section 10.

7.2 Bias of Learner-PFI

The learner-PFI is unbiased regarding the expected learner-PFI over the distribution of models F , since the learner-PFI is a simple mean estimate. However, unlike the learner-PD, model unbiasedness does not, in general, imply unbiasedness of the learner-PFI *regarding the DGP-PFI*. In the following, we study the PFI bias when the squared error is used for loss L (L2-loss).

Theorem 2 *If model \hat{f} is unbiased with $\mathbb{E}_F[\hat{f}] = f$ and the L2-loss is used, then the conditional model-PFI and conditional learner-PFI are unbiased estimators of the conditional DGP-PFI.*

Corollary 1 *If model \hat{f} is unbiased, the L2-loss is used and the features X_S are independent of features X_C , then the marginal model-PFI and marginal learner-PFI are unbiased estimators of the DGP-PFI. If the features are dependent, the following bias is introduced: $\text{PFI}_{\hat{f}} - \text{DGP-PFI} = \mathbb{E}_{\tilde{X}_S X}[\mathbb{V}_F[\hat{f}]] - \mathbb{E}_X[\mathbb{V}_F[\hat{f}]]$.*

Proof Sketch 2 *Both L and \tilde{L} can be decomposed into bias, variance, and irreducible error. Due to the subtraction, the irreducible error vanishes and the differences of biases and variances remain. Model unbiasedness sets the bias terms to zero, but the difference in variance only becomes zero if either $X_S \perp\!\!\!\perp X_C$ or conditional PFI is used. The extended proof can be found in Appendix F.*

Sampling feature X_S creates a new distribution (\tilde{X}_S, X_C) , with a (possibly) different variance for a given point across models. If the variance of \hat{f} changes for \tilde{X}_S , this leads to a bias in the PFI estimate. Besides this bias due to the extrapolation variance, the assumption of model unbiasedness is critical or even unreasonable for regions outside of \mathbb{P}_{XY} , since there is no feedback whether the model matches the DGP in these regions. Furthermore, the DGP might have a probability density of zero for regions of extrapolation. This means that the marginal PFI for dependent features can have a conceptual problem, as the permutation might create data points that are in conflict with the DGP (Hooker and Mentch, 2019; Molnar et al., 2020).²

Intuitively, the model-PFI and learner-PFI should tend to have a negative bias and therefore underestimate the DGP-PFI. A model cannot use more information about the target than is encoded in the DGP (except for dependent features in combination with marginal PFI). However, as Theorem 3 shows the (conditional) PFI can be larger than the DGP-PFI.

Theorem 3 *The difference between the conditional PFI ($c\text{PFI}_{\hat{f}}$) and the conditional DGP-PFI ($c\text{PFI}_f$) of a model \hat{f} is given by:*

$$c\text{PFI}_f - c\text{PFI}_{\hat{f}} = 2\mathbb{E}_{X_C}[\mathbb{V}_{X_S|X_C}[f] - \text{Cov}_{X_S|X_C}[f, \hat{f}]].$$

² Imagine a person with a weight of 4kg and a height of 2m.

Proof Sketch 3 For the L2 loss, the expected loss of a model \hat{f} can be decomposed into the expected loss between \hat{f} and f and the expected variance of Y given X . Due to the subtraction, the latter term vanishes. The remainder can be simplified using that $Y \perp\!\!\!\perp \tilde{X}_S \mid X_C$ and $P(\tilde{X}_S, X_C) = P(X_S, X_C)$. The extended proof can be found in Appendix G.

However, for an overestimation of the PFI to occur, the expected conditional variance of \hat{f} must be greater than the one of f . Moreover, \hat{f} and f must have a large expected conditional covariance, meaning that \hat{f} has learned something about f .

7.3 Variance Estimation

The learner-PD and learner-PFI vary due to model variance (refitted models), but also due to using different samples each time for the Monte Carlo integration. Their variance estimates therefore capture the entire modeling process. Insofar, learner-PD/PFI along with their variance estimators bring us closer to the DGP-PD/PFI and only the systematic bias remains unknown.

We can estimate this point-wise variance of the learner-PD with:

$$\hat{\mathbb{V}}(\widehat{PD}(x)) = \left(\frac{1}{m} + c \right) \cdot \frac{1}{(m-1)} \sum_{d=1}^m (\widehat{PD}_d(x) - \widehat{PD}(x))^2$$

And equivalently for learner-PFI:

$$\hat{\mathbb{V}}(\widehat{PFI}) = \left(\frac{1}{m} + c \right) \cdot \frac{1}{(m-1)} \sum_{d=1}^m (\widehat{PFI}_d - \widehat{PFI})^2$$

The correction term c depends on the data setting. In simulation settings that allow us to draw new training and test sets for each model, we can use $c = 0$, yielding the standard variance estimators. In real world settings, we usually have a fixed dataset of size n and models are refitted using resampling techniques. Consequently, data are shared by model refits and variance estimators will underestimate the true variance (Nadeau and Bengio, 2003). To correct the variance estimate of the generalization error for bootstrapped or subsampled models, Nadeau and Bengio (2003) suggested the correction term $c = \frac{n_2}{n_1}$ (where n_2 and n_1 are sizes of test and training data). However, the correction remains a rough correction, relying on the strongly simplifying assumption that the correlation between model refits depends only on the number of shared observations in the respective training datasets, and not on the specific observations that they share. While this assumption is usually wrong, we show in Section 8 that the correction term offers a vast improvement for variance estimation – compared to using no correction.

7.4 Confidence Bands and Intervals

Since learner-PD and learner-PFI are means with estimated variance, we can use the t-distribution with $m - 1$ degrees of freedom to construct confidence bands/intervals, where m is the number of model fits. The point-wise confidence band for learner-PD is:

$$CI_{\widehat{PD}(x)} = \left[\widehat{PD}(x) - t_{1-\frac{\alpha}{2}} \sqrt{\widehat{V}(\widehat{PD}(x))}; \widehat{PD}(x) + t_{1-\frac{\alpha}{2}} \sqrt{\widehat{V}(\widehat{PD}(x))} \right],$$

where $t_{1-\frac{\alpha}{2}}$ is the respective $1 - \alpha/2$ quantile of the t-distribution with $m - 1$ degrees of freedom. Equivalently, we propose a confidence interval for the learner-PFI:

$$CI_{\widehat{PFI}} = \left[\widehat{PFI} - t_{1-\frac{\alpha}{2}} \sqrt{\widehat{V}(\widehat{PFI})}; \widehat{PFI} + t_{1-\frac{\alpha}{2}} \sqrt{\widehat{V}(\widehat{PFI})} \right].$$

Respecting the model variance can make a difference in the interpretation as we show in the application, Section 9. Resampling strategies make better use of the data, in the sense that a bigger share of the data ends up being used as test data compared to the holdout strategy.

8 Confidence Interval Coverage Simulation

In simulations we compared confidence interval performance between bootstrapping and subsampling, with and without variance correction. We simulated two data generating processes: a *linear* DGP was defined as $y = f(x) = x_1 - x_2 + \epsilon$ and a *non-linear* DGP as $y = f(x) = x_1 - \sqrt{1 - x_2} + x_3 \cdot x_4 + (x_4/10)^2 + \epsilon$. All features were uniformly sampled from the unit interval $[0; 1]$ and for both DGPs we set $\epsilon \sim N(0, 1)$. We studied the two settings “simulation” and “real world”. In both settings, we trained (each 15 times) linear models (lm), regression trees (tree) and random forests (rf), and computed confidence intervals for learner-PD and learner-PFI across the 15 refitted models. In the “simulation” setting, we sampled $n \in \{100, 1,000\}$ fresh data points for each model refit, where 63.2% of the data were used for training and the remaining 36.8% for PDP and PFI estimation.

In the “real world” setting, we sampled $n \in \{100, 1,000\}$ data points **once** per experiment, and generated 15 training data sets using bootstrap (sample size n with replacement, which yields $0.632 \cdot n$ unique data points in expectation) or subsampling (sample size $0.632 \cdot n$ without replacement). In both settings, learner-PD and learner-PFI plus their respective confidence intervals were computed over the 15 retrained models. We repeated the experiment 10,000 times and counted how often the estimated confidence intervals covered the expected PD or PFI ($\mathbb{E}_{\hat{f}}[PD]$ and $\mathbb{E}_{\hat{f}}[PFI]$) over the distribution of

9. Relating the Partial Dependence Plot and Permutation Feature Importance to the Data Generating Process

Table 1 Coverage Probability of the 95% PDP Confidence Bands. boot = bootstrap, subs = subsampling, * = with adjustment.

dgp	model	n	boot	boot*	subs	subs*	ideal
linear	lm	100	0.41	0.89	0.34	0.82	0.95
linear	lm	1000	0.41	0.89	0.33	0.80	0.95
linear	rf	100	0.39	0.86	0.36	0.83	0.95
linear	rf	1000	0.38	0.87	0.35	0.83	0.95
linear	tree	100	0.54	0.96	0.47	0.92	0.95
linear	tree	1000	0.57	0.96	0.48	0.91	0.95
non-linear	lm	100	0.43	0.90	0.36	0.84	0.95
non-linear	lm	1000	0.41	0.89	0.33	0.81	0.95
non-linear	rf	100	0.39	0.87	0.36	0.84	0.95
non-linear	rf	1000	0.38	0.86	0.36	0.83	0.95
non-linear	tree	100	0.58	0.98	0.51	0.95	0.95
non-linear	tree	1000	0.59	0.97	0.51	0.94	0.95

Table 2 Coverage Probability of the 95% PFI Confidence Intervals. boot = bootstrap, subs = subsampling, * = with adjustment.

dgp	model	n	boot	boot*	subs	subs*	ideal
linear	lm	100	0.27	0.70	0.23	0.63	0.94
linear	lm	1000	0.25	0.68	0.21	0.60	0.95
linear	rf	100	0.44	0.92	0.39	0.88	0.95
linear	rf	1000	0.42	0.90	0.38	0.86	0.95
linear	tree	100	0.52	0.97	0.42	0.90	0.95
linear	tree	1000	0.42	0.90	0.34	0.81	0.95
non-linear	lm	100	0.31	0.81	0.25	0.72	0.94
non-linear	lm	1000	0.25	0.67	0.21	0.59	0.95
non-linear	rf	100	0.47	0.94	0.43	0.91	0.95
non-linear	rf	1000	0.41	0.89	0.38	0.86	0.95
non-linear	tree	100	0.68	0.99	0.56	0.96	0.94
non-linear	tree	1000	0.58	0.97	0.46	0.92	0.95

models F .³ These expected values were computed using 10,000 separate runs. The coverage estimates were averaged across features per scenario, and, for PD also across grid points ($\{0.1, 0.3, 0.5, 0.7, 0.9\}$ for all features).

Table 2 and Table 1 show that in the “simulation” setting (“ideal”), we can recover confidence intervals using the standard variance estimation with the desired coverage probability. However, in the “real-world”, setting the confidence intervals for both learner-PD and learner-PFI are too narrow across all scenarios and both resampling strategies, when the intervals are based on naive variance estimates. Some coverage probabilities are especially low, such as for linear models with 30% – 40%.

The coverage probabilities drastically improve when the correction term is used, see Figure 3. However, in the simulated scenarios, they are still somewhat too narrow. For the linear model, the confidence intervals were the most narrow

³ The coverage is not regarding the DGP-PD/PFI, but regarding the expected learner-PD/PFI, as we studied the choices of resampling and correction for the model variance.

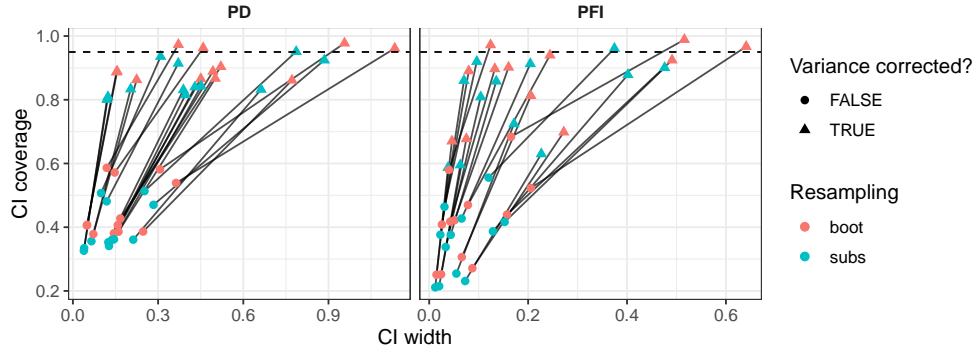


Figure 3 Confidence interval width vs. coverage for *bootstrapping* and *subsampling*, comparing before and after correction. Segments connect identical scenarios.

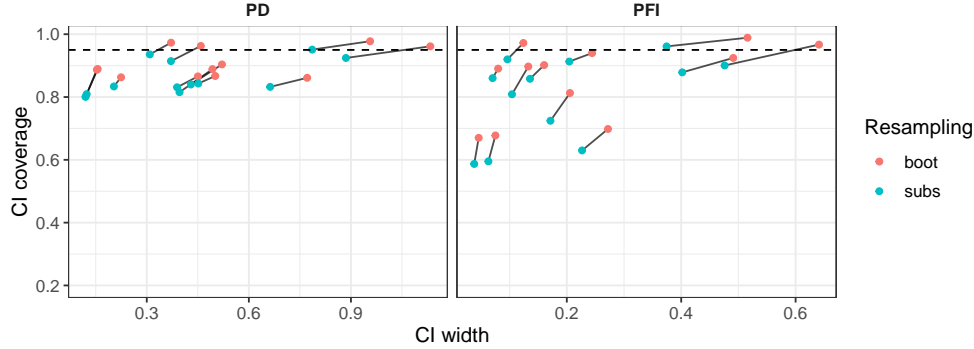


Figure 4 Confidence interval width vs. coverage for *bootstrap* and *subsampling*, both with correction. Segments connect identical scenarios.

with coverage probabilities of around 80% – 90% for PD and 60% – 80% for PFI across DGPs and sample sizes. The PD confidence bands were not much affected by increasing sample size n , but the PFI estimates became slightly more narrow in most cases. In the case of decision trees, the adjusted confidence intervals were sometimes too large, especially for adjusted bootstrap.

Except for trees on the *non-linear* DGP, bootstrap outperformed subsampling in terms of coverage, meaning the coverage was closer to the 95% level and rather erred on the side of “caution” with wider confidence intervals (see Figure 4). As recommended in Nadeau and Bengio (2003), we used 15 refits. We additionally analyzed how the coverage and interval width changed by increasing refits from 2 to 30 and noticed that the coverage worsened with more refits, while the width of the confidence intervals decreased. Increasing the number of refits comes with an inherent trade-off between interval width and coverage: The more refits are considered, the more accurate the learner-PFI and learner-PD become and also the more certain the variance estimates become, scaling with $1/m$. But there is a limit to the information in the data, so

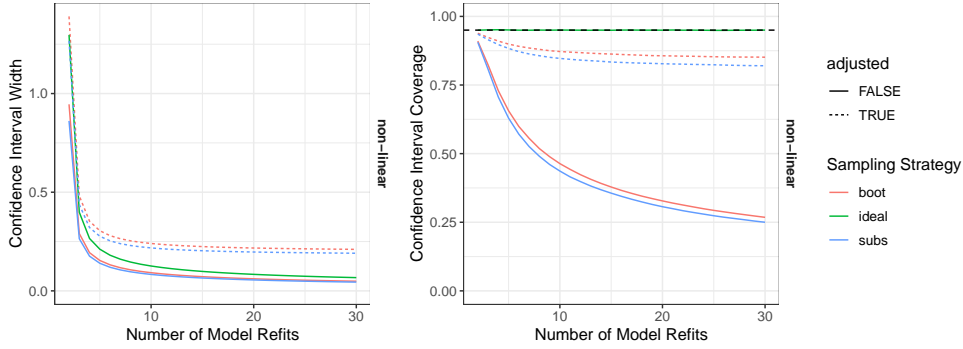


Figure 5 Average PD confidence band width (left) and coverage (right) as a function of number of refitted models for the random forest on the *non-linear* DGP.

that additional refits falsely reduce the variance estimate and the confidence intervals become too narrow. To refit the model 10 - 20 times seemed to be an acceptable trade-off between coverage and interval width, see for example Figure 5. Below ~ 10 refits, the confidence intervals were large, and also the mean PD/PFI estimates have a high variance. Above ~ 20 refits, the widths did not decrease by much anymore. The figures for the other scenarios can be found in Appendix H. With our simulation results we could show that confidence intervals using the naive variance estimation (without correction) results in way too narrow intervals. While the simple correction term by Nadeau and Bengio (2003) does not always provide the desired coverage probability, it is a vast improvement over the naive approach. We therefore recommend using the correction when computing confidence intervals for learner-PD and learner-PFI – it is currently the best approach available. We also recommend refitting the model around 15 times. For more “cautious” confidence intervals we recommend using confidence intervals based on resampling with replacement (bootstrap) over sampling without replacement (subsampling). However, beside wider confidence intervals, the bootstrap requires additional attention when model tuning with internal resampling is used, as data points may otherwise end up in both training and validation data.

9 Application

We apply our proposed estimators to predict wine quality (Cortez et al., 2009) ($n = 1599$) from physicochemical features such as alcohol content and acidity. We compared the performance (mean squared error) of a linear regression model, a regression tree (CART) (Breiman et al., 1984) and a random forest (Breiman, 2001) using 15 bootstrap samples (sample size n with replacement). The MSEs for the different models were: 0.425 (Linear regression), 0.342 (Random Forest) and 0.456 (Tree). The random forest was significantly better than the other models based on an adjusted t-test of the performance difference (Nadeau and Bengio, 2003), with a 95% confidence interval of $[-0.098; -0.069]$

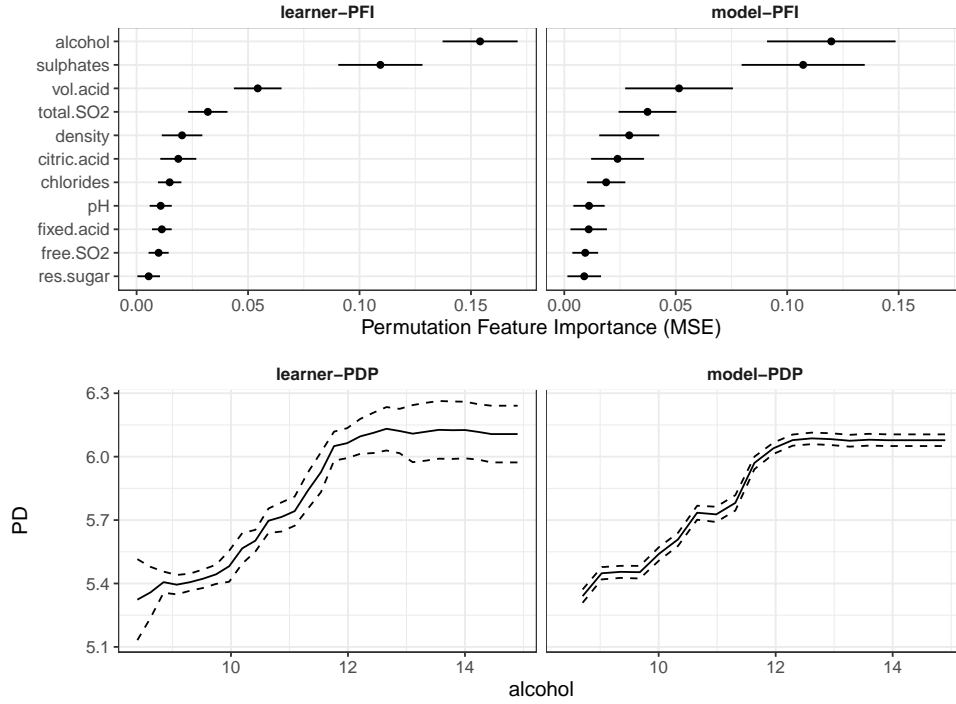


Figure 6 Top: Lerner-PFI and model-PFI with point-wise 95%-confidence intervals for the random forest. Bottom: Lerner-PDP and model-PDP with point-wise 95%-confidence bands for the random forest and feature "alcohol".

for the difference to the linear model MSE and $[-0.158; -0.071]$ for the difference to the decision tree. We reused the 15 random forests from the bootstrap to estimate the learner-PD and learner-PFI including their confidence intervals based on adjusted variance estimates. Figure 6, top row, shows that the most important features were alcohol, sulphates and volatile acidity. The model-PFI quantifies how important each feature was for a fixed random forest, and the confidence intervals show the variance of the approximation of the model-PFI due to Monte Carlo integration. The model-PFI, however, cannot tell us how much the estimate varies due to model variance. The learner-PFI quantifies this model variance. Both model-PFI and learner-PFI gave a similar ordering for the top features. The learner-PFI shows that alcohol is more important than sulphates (with no overlap in the confidence intervals), for which the model-PFI would suggest that the importance is almost equal.

Figure 6, bottom row, shows both the model-PDP and the learner-PDP for the alcohol feature. Notably, the confidence bands of the learner-PDP are wider than of the model-PDP. Especially for very low and for high alcohol volumes the models have a high variance. Neglecting the model variance would mean being overconfident about the partial dependence curve. In particular, the Monte Carlo approximation error decreases with $1/n$ as the sample size n for PD and PFI estimation increases. Wrongly interpreted, this can lead to a

false sense of confidence in the estimated effects and importance, even though only one model is considered and model variance is ignored.

10 Discussion

We related the PD and the PFI to the data generating process (DGP), proposed variance and confidence intervals, and discussed conditions for inference. Our derivations were motivated by taking an external view of the statistical inference process, and postulating that there is a ground truth counterpart to PD/PFI in the data generating process. To the best of our knowledge, statistical inference via model-agnostic interpretable machine learning is already used in practice, but under-explored in theory.

A critical assumption for inference of effects and importance using interpretable machine learning is unbiasedness of the model. The model bias is difficult to test, and can be introduced by, e.g. choice of model class, regularization and feature selection. For example, regularization techniques such as LASSO introduce a small bias *on purpose* (Tibshirani, 1996) to decrease model variance and improve predictive performance. We have to better understand how specific biases affect the prediction function and therefore PD and PFI estimates. Another crucial limitation for inference of PD and PFI is the underestimation of variance due to data sharing between model refits. While we could show that a simple correction of the variance (Nadeau and Bengio, 2003) vastly improves the coverage, a proper estimation of the variance remains an open issue. A promising approach relying on repeated nested cross validation to correctly estimate the variance was recently proposed by Bates et al. (2021). However, this approach is more computationally intensive by up to a factor of 1,000.

References

- Altmann A, Tološi L, Sander O, Lengauer T (2010) Permutation importance: a corrected feature importance measure. *Bioinformatics* 26(10):1340–1347
- Apley DW, Zhu J (2020) Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 82(4):1059–1086
- Archer KJ, Kimes RV (2008) Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis* 52(4):2249–2260
- Bair E, Ohrbach R, Fillingim RB, Greenspan JD, Dubner R, Diatchenko L, Helgeson E, Knott C, Maixner W, Slade GD (2013) Multivariable modeling of phenotypic risk factors for first-onset tmd: the oppera prospective cohort study. *The Journal of Pain* 14(12):T102–T115
- Bates S, Hastie T, Tibshirani R (2021) Cross-validation: what does it estimate and how well does it do it? *arXiv preprint arXiv:210400673*

- Boulesteix AL, Wright MN, Hoffmann S, König IR (2020) Statistical learning approaches in the genetic epidemiology of complex diseases. *Human Genetics* 139(1):73–84
- Bouthillier X, Delaunay P, Bronzi M, Trofimov A, Nichyporuk B, Szeto J, Sepah N, Raff E, Madan K, Voleti V, et al. (2021) Accounting for variance in machine learning benchmarks. *arXiv preprint arXiv:210303098*
- Breiman L (2001) Random forests. *Machine Learning* 45(1):5–32
- Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) *Classification and Regression Trees*. CRC Press, Boca Raton
- Cafri G, Bailey BA (2016) Understanding variable effects from black box prediction: Quantifying effects in tree ensembles using partial dependence. *Journal of Data Science* 14(1):67–95
- Candès E, Fan Y, Janson L, Lv J (2018) Panning for gold: ‘model-X’ knock-offs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 80(3):551–577
- Chernozhukov V, Chetverikov D, Demirer M, Duflo E, Hansen C, Newey W, Robins J (2018) Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal* 21(1):C1–C68
- Cortez P, Cerdeira A, Almeida F, Matos T, Reis J (2009) Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems* 47(4):547–553
- Emrich E, Pierdzioch C (2016) Public goods, private consumption, and human capital: Using boosted regression trees to model volunteer labour supply. *Review of Economics/Jahrbuch für Wirtschaftswissenschaften* 67(3)
- Esselman PC, Stevenson RJ, Lupi F, Riseng CM, Wiley MJ (2015) Landscape prediction and mapping of game fish biomass, an ecosystem service of michigan rivers. *North American Journal of Fisheries Management* 35(2):302–320
- Fahrmeir L, Kneib T, Lang S, Marx B (2007) *Regression*. Springer
- Fisher A, Rudin C, Dominici F (2019) All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research* 20(177):1–81
- Friedman JH (1991) Multivariate adaptive regression splines. *The Annals of Statistics* pp 1–67
- Geman S, Bienenstock E, Doursat R (1992) Neural networks and the bias/variance dilemma. *Neural Computation* 4(1):1–58
- Grange SK, Carslaw DC (2019) Using meteorological normalisation to detect interventions in air quality time series. *Science of The Total Environment* 653:578–588
- Groemping U (2020) Model-agnostic effects plots for interpreting machine learning models. *Reports in Mathematics, Physics and Chemistry, Department II, Beuth University of Applied Sciences Berlin Report 1/2020*
- Hooker G, Mentch L (2019) Please stop permuting features: An explanation and alternatives. *arXiv preprint arXiv:190503151*
- Hothorn T, Leisch F, Zeileis A, Hornik K (2005) The design and analysis of benchmark experiments. *Journal of Computational and Graphical Statistics* 14(3):675–699

- Ishwaran H, Lu M (2019) Standard errors and confidence intervals for variable importance in random forest regression, classification, and survival. *Statistics in Medicine* 38(4):558–582
- Janitza S, Celik E, Boulesteix AL (2018) A computationally fast variable importance test for random forests for high-dimensional data. *Advances in Data Analysis and Classification* 12(4):885–915
- Mitchell TM (1980) The need for biases in learning generalizations. Department of Computer Science, Laboratory for Computer Science Research . . .
- Molnar C, König G, Bischl B, Casalicchio G (2020) Model-agnostic feature importance and effects with dependent features—a conditional subgroup approach. *arXiv preprint arXiv:200604628*
- Nadeau C, Bengio Y (2003) Inference for the generalization error. *Machine Learning* 52(3):239–281
- Obringer R, Nateghi R (2018) Predicting urban reservoir levels using statistical learning techniques. *Scientific Reports* 8(1):1–9
- Page WG, Wagenbrenner NS, Butler BW, Forthofer JM, Gibson C (2018) An evaluation of ndfd weather forecasts for wildland fire behavior prediction. *Weather and Forecasting* 33(1):301–315
- Parr T, Wilson JD (2019) A stratification approach to partial dependence for codependent variables. *arXiv preprint arXiv:190706698*
- Parr T, Wilson JD, Hamrick J (2020) Nonparametric feature impact and importance. *arXiv preprint arXiv:200604750*
- Ribeiro MT, Singh S, Guestrin C (2016) Model-agnostic interpretability of machine learning. *ICML WHI '16* URL <http://arxiv.org/abs/1606.05386>, 1606.05386
- Scholbeck CA, Molnar C, Heumann C, Bischl B, Casalicchio G (2019) Sampling, intervention, prediction, aggregation: A generalized framework for model agnostic interpretations. *arXiv preprint arXiv:190403959*
- Stachl C, Au Q, Schoedel R, Gosling SD, Harari GM, Buschek D, Völkel ST, Schuwerk T, Oldemeier M, Ullmann T, Hussmann H, Bischl B, Bühner M (2020) Predicting personality from patterns of behavior collected with smartphones. *Proceedings of the National Academy of Sciences* 117(30):17680–17687
- Strobl C, Boulesteix AL, Kneib T, Augustin T, Zeileis A (2008) Conditional variable importance for random forests. *BMC Bioinformatics* 9(1):307
- Tibshirani R (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1):267–288
- Watson DS, Wright MN (2019) Testing conditional independence in supervised learning algorithms. *arXiv preprint arXiv:190109917*
- Williamson BD, Gilbert PB, Carone M, Simon N (2019) Nonparametric variable importance assessment using machine learning techniques. *Biometrics*
- Williamson BD, Gilbert PB, Simon NR, Carone M (2020) A unified approach for inference on algorithm-agnostic variable importance. *arXiv preprint arXiv:200403683*
- Zhang L, Janson L (2020) Floodgate: inference for model-free variable importance. *arXiv preprint arXiv:200701283*

-
- Zhao Q, Hastie T (2021) Causal interpretations of black-box models. *Journal of Business & Economic Statistics* 39(1):272–281
- Zheng W, van der Laan MJ (2011) Cross-validated targeted minimum-loss-based estimation. In: *Targeted Learning*, Springer, pp 459–474

Supplementary Material

A Bias and Variance of PD

The expected squared difference between model-PD and DGP-PD can be decomposed into bias and variance.

Proof

$$\begin{aligned}
 \mathbb{E}_F[(PD_f - PD_{\hat{f}})^2] &= \mathbb{E}_F[(\mathbb{E}_{X_C}[f] - \mathbb{E}_{X_C}[\hat{f}])^2] \\
 &= \mathbb{E}_F[\mathbb{E}_{X_C}[f]^2] - 2\mathbb{E}_F[\mathbb{E}_{X_C}[f] \cdot \mathbb{E}_{X_C}[\hat{f}]] \\
 &\quad + \mathbb{E}_F[\mathbb{E}_{X_C}[\hat{f}]^2] \\
 &= \mathbb{E}_F^2[\mathbb{E}_{X_C}[f]] + \underbrace{\mathbb{V}_F[\mathbb{E}_{X_C}[f]]}_{=0} \\
 &\quad - 2\mathbb{E}_{X_C}[f]\mathbb{E}_F[\mathbb{E}_{X_C}[\hat{f}]] \\
 &\quad + \mathbb{E}_F^2[\mathbb{E}_{X_C}[\hat{f}]] + \mathbb{V}_F[\mathbb{E}_{X_C}[\hat{f}]] \\
 &= \underbrace{(\mathbb{E}_{X_C}[f] - \mathbb{E}_F[\mathbb{E}_{X_C}[\hat{f}]])^2}_{\text{Bias}} \\
 &\quad + \underbrace{\mathbb{V}_F[\mathbb{E}_{X_C}[\hat{f}]]}_{\text{Variance}} \\
 &= \underbrace{(PD_f - \mathbb{E}_F[PD_{\hat{f}}])^2}_{\text{Bias}} + \underbrace{\mathbb{V}_F[PD_{\hat{f}}]}_{\text{Variance}}
 \end{aligned}$$

B Bias and Variance of PFI

The expected squared difference between model-PFI and DGP-PFI can be decomposed into bias and variance.

Proof

$$\begin{aligned}
 \mathbb{E}_F[(PFI_{\hat{f}} - PFI_f)^2] &= \mathbb{E}_F[PFI_{\hat{f}}^2] + \mathbb{E}_F[PFI_f^2] \\
 &\quad - 2\mathbb{E}_F[PFI_{\hat{f}}PFI_f] \\
 &= \mathbb{V}_F[PFI_{\hat{f}}] + \mathbb{E}_F[PFI_{\hat{f}}]^2 \\
 &\quad + PFI_f^2 - 2\mathbb{E}_F[PFI_{\hat{f}}PFI_f] \\
 &= (PFI_f - \mathbb{E}_F[PFI_{\hat{f}}])^2 + \mathbb{V}_F[PFI_{\hat{f}}] \\
 &= \text{Bias}_F^2[PFI_{\hat{f}}] + \mathbb{V}_F[PFI_{\hat{f}}]
 \end{aligned}$$

C Model-PD Unbiasedness Regarding Theoretical PD

Proof By the law of large numbers, the Monte Carlo integration converges with $n_2 \rightarrow \infty$ to the true integral. Assuming n_2 identically distributed random draws $X_C^{(1)}, \dots, X_C^{(n_2)} \sim X_C$

and model \hat{f} , the estimate is:

$$\begin{aligned}\mathbb{E}_{X_C}[\widehat{PD}(x)] &= \mathbb{E}_{X_C} \left[\frac{1}{n_2} \sum_{i=1}^{n_2} \hat{f}(x, X_C^{(i)}) \right] \\ &= \frac{1}{n_2} n_2 \mathbb{E}_{X_C} [\hat{f}(x, X_C)] \\ &= PD(x)\end{aligned}$$

and therefore unbiased for the interval, i.e., the theoretical PD of the model.

D Model-PD Unbiasedness Regarding DGP-PD

Proof Unbiasedness of the model \hat{f} implies unbiasedness of the model-PD.

$$\begin{aligned}\mathbb{E}_F[\mathbb{E}_{X_C}[\hat{f}]] &\stackrel{Def}{=} \int_F \int_{X_C} \hat{f}(x_S, X_C) d\mathbb{P}(X_C) d\mathbb{P}(F) \\ &\stackrel{Fub}{=} \int_{X_C} \int_F \hat{f}(x_S, X_C) d\mathbb{P}(F) d\mathbb{P}(X_C) \\ &\stackrel{Def}{=} \mathbb{E}_{X_C}[\mathbb{E}_F[\hat{f}]] \\ &\stackrel{Unbiased}{=} \mathbb{E}_{X_C}[f]\end{aligned}$$

Fubini's theorem requires that $\int_F \int_{X_C} |\hat{f}| d\mathbb{P}_F d\mathbb{P}_{X_C} < \infty$. This is given when it can be guaranteed that the model predictions have an upper bound c : $|\hat{f}(x)| < c < \infty$.

E Model-PFI Regarding theoretical PFI

Proof As a function of random variables, the loss L itself is a random variable. We assume that the loss $L^{(i)}$ of observation i is a sample from the distribution of losses: $L^{(i)} \sim L$ and, similarly for the permuted loss: $\tilde{L}^{(k,i)} \sim \tilde{L}$, where $L^{(i)} = L(y^{(i)}, \hat{f}(x^{(i)}))$ and $\tilde{L}^{(k,i)} = L(y^{(i)}, \hat{f}(\tilde{x}_S^{(k,i)}, x_C^{(i)}))$.

The expectation of our estimator is:

$$\begin{aligned}\mathbb{E}_{\tilde{X}_S X_S X_C Y}[\widehat{PFI}_{\hat{f}}] &= \mathbb{E}_{\tilde{X}_S X_S X_C Y} \left[\frac{1}{n_2} \sum_{i=1}^{n_2} \left(\frac{1}{l} \sum_{k=1}^l (\tilde{L}^{(k,i)} - L^{(i)}) \right) \right] \\ &= \frac{1}{n_2} n_2 \mathbb{E}_{\tilde{X}_S X_S X_C Y} \left[\left(\frac{1}{l} \sum_{k=1}^l \tilde{L} - L \right) \right] \\ &= \mathbb{E}_{\tilde{X}_S X_C Y}[\tilde{L}] - \mathbb{E}_{X_S X_C Y}[L] \\ &= PFI_{\hat{f}}\end{aligned}$$

In expectation, we retrieve the theoretical PFI of the model.

F PFI Biases for L2

We assume that L is the squared loss $L(y, \hat{f}) = (y - \hat{f}(x))^2$ and that $\mathbb{E}[Y|X]$ can be described by f with some additive, irreducible, error ϵ with $\mathbb{E}(\epsilon) = 0$ and $\mathbb{V}(\epsilon) = \sigma^2$. To further examine the bias for PFI, we apply the Bias-Variance Decomposition also on the loss itself: In addition, we use that $\mathbb{E}_{XY}[Y] = \mathbb{E}_X[f(X)]$, $\mathbb{V}_Y[Y] = \sigma^2$ and $\mathbb{E}[A^2] = \mathbb{V}[A] + \mathbb{E}[A]^2$. We

9. Relating the Partial Dependence Plot and Permutation Feature Importance to the Data Generating Process

first derive the bias-variance decomposition of (i) permuted loss and (ii) original loss and derive from that the expected PFI.

For the permuted loss (i):

$$\begin{aligned}
 \mathbb{E}_{F\tilde{X}_SXY}[\tilde{L}] &= \mathbb{E}_{F\tilde{X}_SXY}[(Y - \tilde{f})^2] \\
 &= \mathbb{E}_{\tilde{X}_SXY}[Y^2 - 2Y\mathbb{E}_F[\tilde{f}] + \mathbb{E}_F[\tilde{f}^2]] \\
 &= \mathbb{E}_{\tilde{X}_SXY}[Y^2 - 2Y\mathbb{E}_F[\tilde{f}] + \mathbb{E}_F[\tilde{f}]^2 + \mathbb{V}_F[\tilde{f}]] \\
 &= \mathbb{V}_Y[Y] + \mathbb{E}_{\tilde{X}_SX}[f^2 - 2f\mathbb{E}_F[\tilde{f}] + \mathbb{E}_F[\tilde{f}]^2 + \mathbb{V}_F[\tilde{f}]] \\
 &= \underbrace{\sigma^2}_{\text{Data Var}} + \underbrace{\mathbb{E}_{\tilde{X}_SX}[(f - \mathbb{E}_F[\tilde{f}])^2]}_{\text{Bias}^2} + \underbrace{\mathbb{E}_{\tilde{X}_SX}[\mathbb{V}_F[\tilde{f}]]}_{\text{Variance}}
 \end{aligned}$$

For the original loss (ii):

$$\begin{aligned}
 \mathbb{E}_{FXY}[L] &= \mathbb{E}_{FXY}[(Y - \hat{f})^2] \\
 &= \mathbb{E}_{XY}[Y^2 - 2Y\mathbb{E}_F[\hat{f}] + \mathbb{E}_F[\hat{f}^2]] \\
 &= \mathbb{E}_{XY}[Y^2 - 2Y\mathbb{E}_F[\hat{f}] + \mathbb{E}_F[\hat{f}]^2 + \mathbb{V}_F[\hat{f}]] \\
 &= \mathbb{V}_Y[Y] + \mathbb{E}_X[f^2 - 2f\mathbb{E}_F[\hat{f}] + \mathbb{E}_F[\hat{f}]^2 + \mathbb{V}_F[\hat{f}]] \\
 &= \underbrace{\sigma^2}_{\text{Data Var}} + \underbrace{\mathbb{E}_X[(f - \mathbb{E}_F[\hat{f}])^2]}_{\text{Bias}^2} + \underbrace{\mathbb{E}_X[\mathbb{V}_F[\hat{f}]]}_{\text{Variance}}
 \end{aligned}$$

The expected PFI for feature X_S then is:

$$\begin{aligned}
 PFI &= \mathbb{E}_{F\tilde{X}_SXY}[\tilde{L}] - \mathbb{E}_{FXY}[L] \\
 &\stackrel{(i)+(ii)}{=} \sigma^2 + \mathbb{E}_{\tilde{X}_SX}[(f - \mathbb{E}_F[\tilde{f}])^2] + \mathbb{E}_{\tilde{X}_SX}[\mathbb{V}_F[\tilde{f}]] \\
 &\quad - (\sigma^2 + \mathbb{E}_X[(f - \mathbb{E}_F[\hat{f}])^2] + \mathbb{E}_X[\mathbb{V}_F[\hat{f}]]) \\
 &= \mathbb{E}_{\tilde{X}_SX}[(f - \mathbb{E}_F[\tilde{f}])^2] - \mathbb{E}_X[(f - \mathbb{E}_F[\hat{f}])^2] \\
 &\quad + \mathbb{E}_{\tilde{X}_SX}[\mathbb{V}_F[\tilde{f}]] - \mathbb{E}_X[\mathbb{V}_F[\hat{f}]]
 \end{aligned}$$

We can derive the same L2 decomposition for the DGP-PFI by replacing \hat{f} with f in the equation above. This yields $PFI_f = \mathbb{E}_{\tilde{X}_SX}[(f(X) - f(\tilde{X}_S, X_C))^2]$, since $\mathbb{V}_F[f] = \mathbb{V}_F[\tilde{f}] = 0$ and $\mathbb{E}_F[f] = f$ and $\mathbb{E}_F[\tilde{f}] = \tilde{f}$.

The bias of the model-PFI, compared to the DGP-PFI, is:

$$PFI_{\tilde{f}} - PFI_f = \underbrace{\mathbb{E}_{\tilde{X}_SX}[(f - \mathbb{E}_F[\tilde{f}])^2 - (f - \tilde{f})^2]}_{\text{Permutation Loss Bias}} \quad (9)$$

$$\underbrace{- \mathbb{E}_X[(f - \mathbb{E}_F[\hat{f}])^2]}_{(\text{Model Bias})^2} + \underbrace{\mathbb{E}_{\tilde{X}_SX}[\mathbb{V}_F[\tilde{f}]] - \mathbb{E}_X[\mathbb{V}_F[\hat{f}]]}_{\text{Variance Inflation}} \quad (10)$$

The permutation loss bias and the squared model bias from the equation above are zero when the model is not biased, i.e., $\hat{f} = f$. The variance inflation term is zero if $\tilde{X}_S \sim X_S|X_C$, which is the case when conditional PFI is used, or when marginal PFI is used and features X_S are independent from features X_C . If the features in X_S and X_C are dependent, the marginal PFI might be biased, even when the underlying model is unbiased.

G DGP-PFI minus model-PFI for L2

$$\begin{aligned}
cPFI_f - cPFI_{\hat{f}} &= \mathbb{E}_{\tilde{X}_S X_C Y}[(Y - f)^2] - \mathbb{E}_{X_S X_C Y}[(Y - f)^2] \\
&\quad - \left(\mathbb{E}_{\tilde{X}_S X_C Y}[(Y - \hat{f})^2] - \mathbb{E}_{X_S X_C Y}[(Y - \hat{f})^2] \right) \\
&= \underbrace{\left(\mathbb{E}_{X_S X_C Y}[(Y - \hat{f})^2] - \mathbb{E}_{X_S X_C Y}[(Y - f)^2] \right)}_{T1:=} \\
&\quad + \underbrace{\left(\mathbb{E}_{\tilde{X}_S X_C Y}[(Y - f)^2] - \mathbb{E}_{\tilde{X}_S X_C Y}[(Y - \hat{f})^2] \right)}_{T2:=}
\end{aligned}$$

We know that for any $g : X \rightarrow Y$ holds:

$$\mathbb{E}_{X,Y}[(Y - g)^2] = \mathbb{E}_X[\mathbb{V}_{Y|X}[Y]] + \mathbb{E}_X[(\mathbb{E}_{Y|X}[Y] - g)^2]$$

Since $f = \mathbb{E}_{Y|X_S, X_C}[Y]$ we can conclude for our first term T1 that:

$$\begin{aligned}
T1 &= \mathbb{E}_{X_S X_C}[\mathbb{V}_{Y|X_S, X_C}[Y]] + \mathbb{E}_{X_S X_C}[(f - \hat{f})^2] \\
&\quad - \left(\mathbb{E}_{X_S X_C}[\mathbb{V}_{Y|X_S, X_C}[Y]] + \underbrace{\mathbb{E}_{X_S X_C}[(f - \hat{f})^2]}_{=0} \right) \\
&= \mathbb{E}_{X_S X_C}[(f - \hat{f})^2]
\end{aligned}$$

We apply the same trick to T2. Moreover, $Y \perp\!\!\!\perp \tilde{X}_S \mid X_C$.

$$\begin{aligned}
T2 &= \mathbb{E}_{\tilde{X}_S X_C}[\mathbb{V}_{Y|\tilde{X}_S, X_C}[Y]] + \mathbb{E}_{\tilde{X}_S X_C}[(\mathbb{E}_{Y|\tilde{X}_S, X_C}[Y] - f)^2] \\
&\quad - \left(\mathbb{E}_{\tilde{X}_S X_C}[\mathbb{V}_{Y|\tilde{X}_S, X_C}[Y]] + \mathbb{E}_{\tilde{X}_S X_C}[(\mathbb{E}_{Y|\tilde{X}_S, X_C}[Y] - \hat{f})^2] \right) \\
&= \mathbb{E}_{\tilde{X}_S X_C}[(\mathbb{E}_{Y|X_C}[Y] - f)^2] - \mathbb{E}_{\tilde{X}_S X_C}[(\mathbb{E}_{Y|X_C}[Y] - \hat{f})^2]
\end{aligned}$$

If we now set together the two terms again and use in the first step that $P(X_S, X_C) = P(\tilde{X}_S, X_C)$, we get:

$$\begin{aligned}
T1+T2 &= \mathbb{E}_{X_S X_C}[(f - \hat{f})^2] + \mathbb{E}_{X_S X_C}[(\mathbb{E}_{Y|X_C}[Y] - f)^2] \\
&\quad - \mathbb{E}_{X_S X_C}[(\mathbb{E}_{Y|X_C}[Y] - \hat{f})^2] \\
&= \mathbb{E}_{X_S X_C} \left[f^2 - 2f\hat{f} + \hat{f}^2 + \mathbb{E}_{Y|X_C}[Y]^2 - 2\mathbb{E}_{Y|X_C}[Y]f + f^2 \right. \\
&\quad \left. - \mathbb{E}_{Y|X_C}[Y]^2 + 2\mathbb{E}_{Y|X_C}[Y]\hat{f} - \hat{f}^2 \right] \\
&= 2\mathbb{E}_{X_S X_C} \left[(f^2 - \mathbb{E}_{Y|X_C}[Y]f) - (f\hat{f} - \mathbb{E}_{Y|X_C}[Y]\hat{f}) \right] \\
&= 2\mathbb{E}_{X_C} \left[\mathbb{E}_{X_S|X_C} \left[(f^2 - \mathbb{E}_{Y|X_C}[Y]f) - (f\hat{f} - \mathbb{E}_{Y|X_C}[Y]\hat{f}) \right] \right] \\
&\stackrel{*}{=} 2\mathbb{E}_{X_C} \left[(\mathbb{E}_{X_S|X_C}[f^2] - \mathbb{E}_{Y|X_C}[Y]\mathbb{E}_{X_S|X_C}[f]) \right. \\
&\quad \left. - (\mathbb{E}_{X_S|X_C}[f\hat{f}] - \mathbb{E}_{Y|X_C}[Y]\mathbb{E}_{X_S|X_C}[\hat{f}]) \right] \\
&\stackrel{**}{=} 2\mathbb{E}_{X_C} \left[(\mathbb{E}_{X_S|X_C}[f^2] - \mathbb{E}_{X_S|X_C}[f]^2) \right. \\
&\quad \left. - (\mathbb{E}_{X_S|X_C}[f\hat{f}] - \mathbb{E}_{X_S|X_C}[\hat{f}]\mathbb{E}_{X_S|X_C}[f]) \right] \\
&= 2\mathbb{E}_{X_C} [\mathbb{V}_{X_S|X_C}[f] - Cov_{X_S|X_C}[f, \hat{f}]]
\end{aligned}$$

At *, we use the fact that the random variable $\mathbb{E}_{Y|X_C}[Y]$ is measurable by the σ -Algebra generated from X_C and we are inclined to pull it out of the expectation. In **, we use that from $f = \mathbb{E}_{Y|X_S, X_C}[Y]$ follows $\mathbb{E}_{X_S|X_C}[f] = \mathbb{E}_{Y|X_C}[Y]$.

9. Relating the Partial Dependence Plot and Permutation Feature Importance to the Data Generating Process

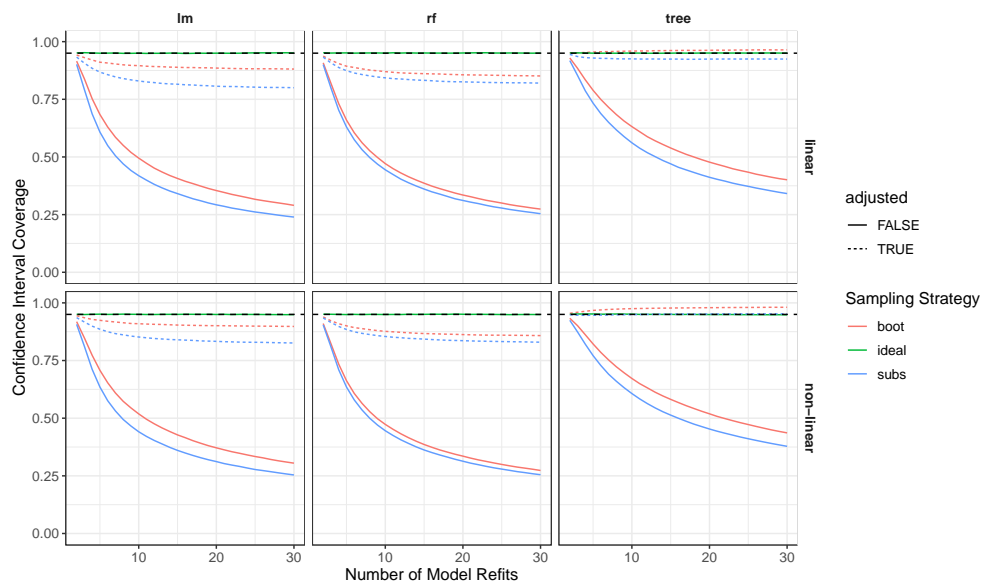


Figure 7 CI coverage for PD with $n=100$.

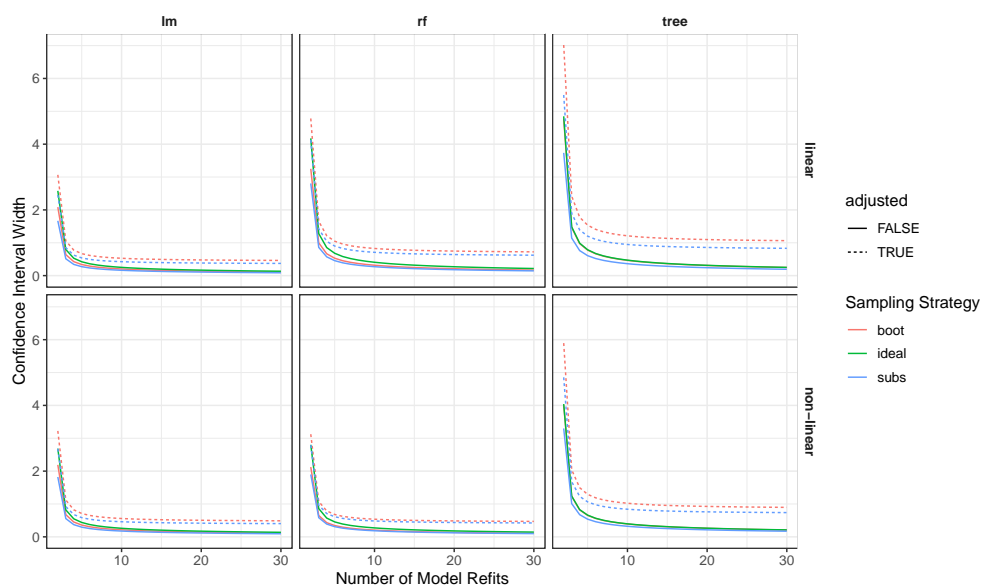


Figure 8 CI width for PD with $n=100$.

H CI simulation results

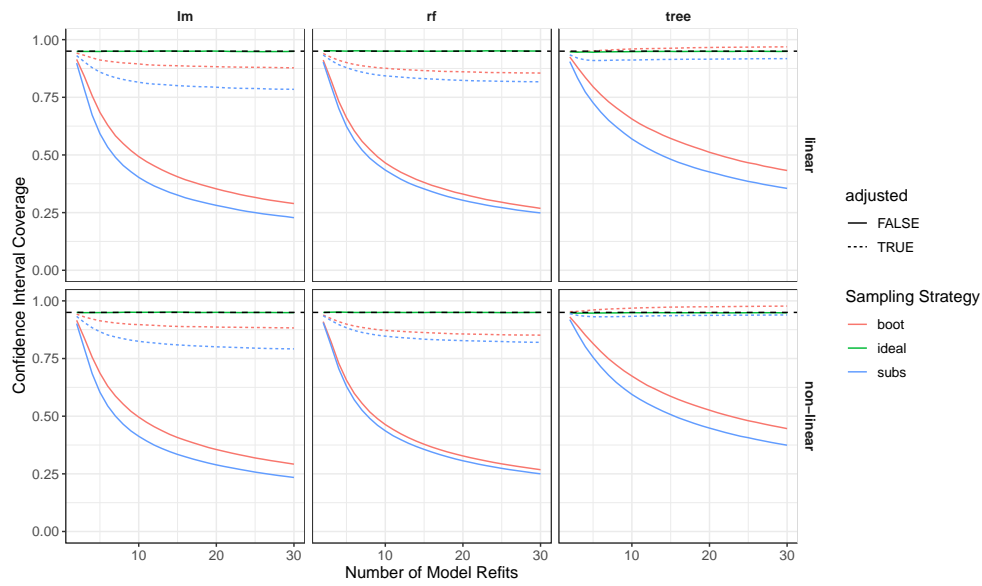


Figure 9 CI coverage for PD with $n=1,000$.

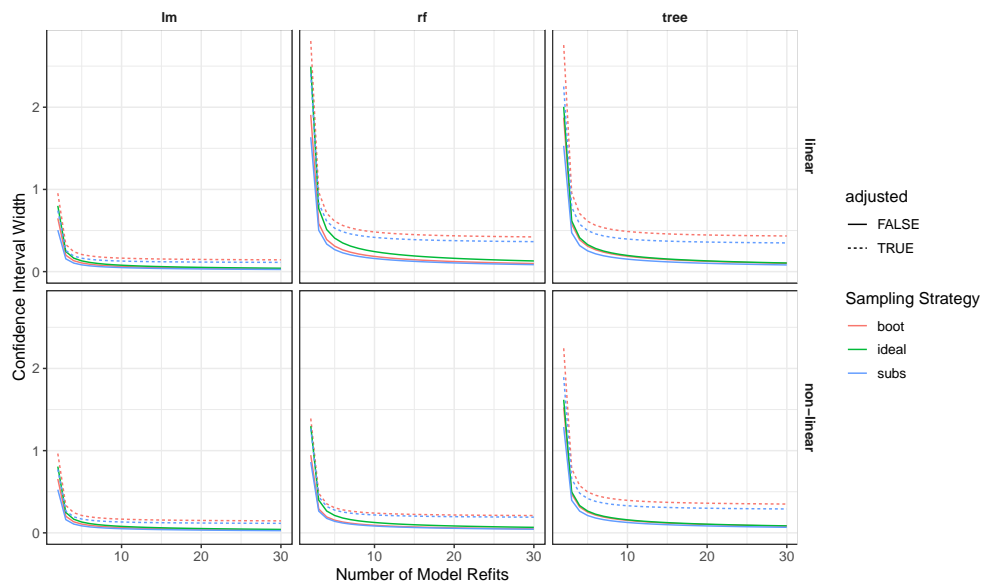


Figure 10 CI width for PD with $n=1,000$.

9. Relating the Partial Dependence Plot and Permutation Feature Importance to the Data Generating Process

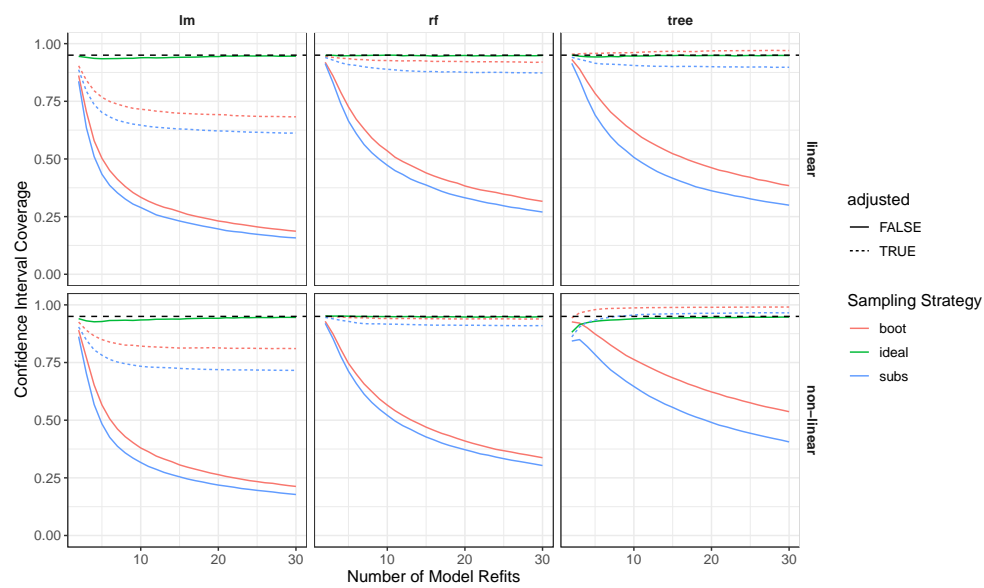


Figure 11 CI coverage for PFI with $n=100$.

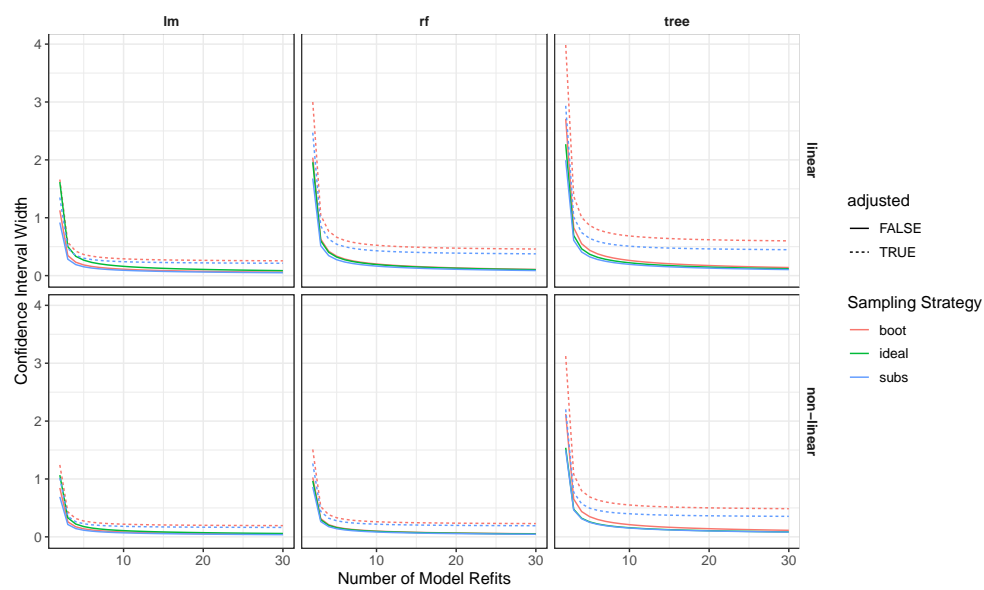


Figure 12 CI width for PFI with $n=100$.

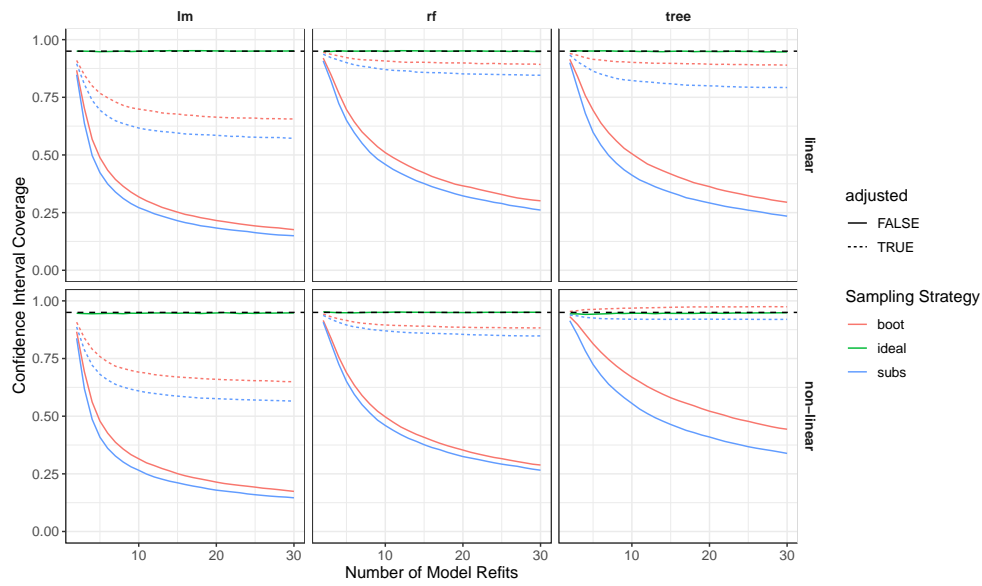


Figure 13 CI coverage for PFI with $n=1,000$.

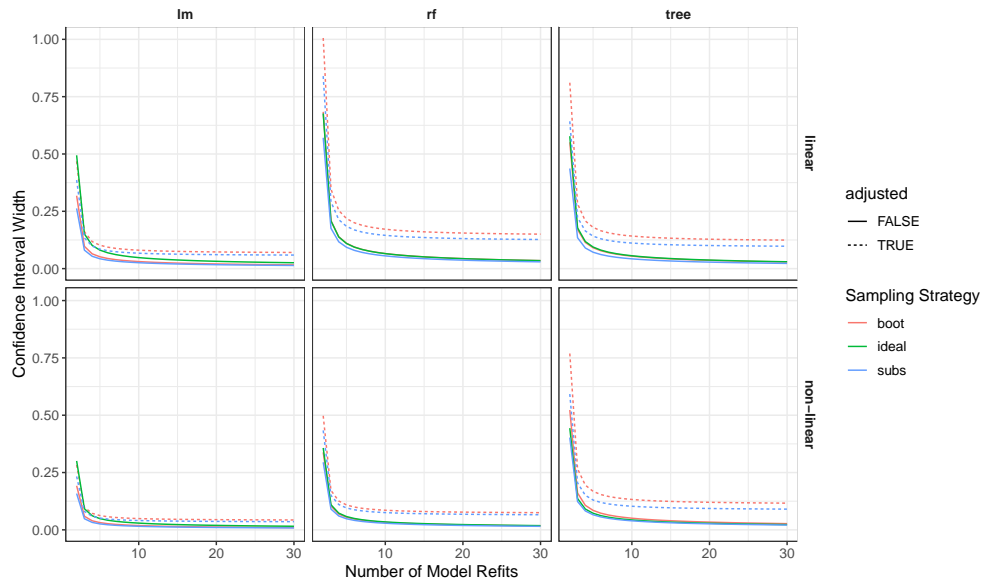


Figure 14 CI width for PFI with $n=1,000$.

10. Relative Feature Importance

Contributing article:

König, G., Molnar, C., Bischl, B., and Grosse-Wentrup, M. (2021). Relative Feature Importance. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 9318-9325).

Copyright information:

©2021 IEEE. Reprinted, with permission, from all authors, Relative Feature Importance, 2020 25th International Conference on Pattern Recognition (ICPR), January 2021.

Author contributions:

Gunnar König mainly wrote the paper. Christoph Molnar partly wrote the section on estimation and testing, reviewed the software code and extensively proofread the mathematical proofs. All authors added input, suggested modifications proofread and revised the paper.

Relative Feature Importance

Gunnar König^{1,2}, Christoph Molnar¹, Bernd Bischl¹, Moritz Grosse-Wentrup^{2,3,4}

¹Institute for Statistics, LMU Munich, ²Research Group Neuroinformatics, University of Vienna,

³Research Platform Data Science @ Uni Vienna, ⁴Vienna Cognitive Science Hub

Abstract—Interpretable Machine Learning (IML) methods are used to gain insight into the relevance of a feature of interest for the performance of a model. Commonly used IML methods differ in whether they consider features of interest in isolation, e.g., Permutation Feature Importance (PFI), or in relation to all remaining feature variables, e.g., Conditional Feature Importance (CFI). As such, the perturbation mechanisms inherent to PFI and CFI represent extreme reference points. We introduce Relative Feature Importance (RFI), a generalization of PFI and CFI that allows for a more nuanced feature importance computation beyond the PFI versus CFI dichotomy. With RFI, the importance of a feature relative to any other subset of features can be assessed, including variables that were not available at training time. We derive general interpretation rules for RFI based on a detailed theoretical analysis of the implications of relative feature relevance, and demonstrate the method’s usefulness on simulated examples.

Index Terms—feature importance, interpretable machine learning, explainable artificial intelligence, causality

I. INTRODUCTION

Predictive modelling is increasingly deployed in high-stakes environments, e.g., in the criminal justice system [11], loan approval [32], recruiting [9] and medicine [27]. Due to legal regulations [10], [29] and ethical considerations, ML methods need not only perform robustly in such environments but also be able to justify their recommendations in a human-intelligible fashion. This development has given rise to the field of interpretable machine learning (IML) that involves studying methods that provide insight into the relevance of features for model performance, referred to as feature importance. Prominent feature importance techniques include permutation feature importance (PFI) [5], [12] and conditional feature importance (CFI) [12], [19], [25]. PFI is based on replacing the feature of interest X_j with a perturbed version sampled from the marginal distribution $P(X_j)$ while CFI perturbs X_j such that the conditional distribution with respect to the set R of remaining features $P(X_j|X_R)$ is preserved. The sampling strategy defines the method’s reference point and therefore affects the method’s implicit notion of relevance. While PFI quantifies the overall reliance of the model on the feature of interest, CFI quantifies its unique contribution given

all remaining features.

While both PFI and CFI are useful, they fail to answer more nuanced questions of feature importance. For instance, a stakeholder may be interested in the importance of a feature relative to a subset of features. Also, the user may want to know how important a feature is relative to variables that had not been available at training time. We suggest relative feature importance (RFI) as a generalization of PFI and CFI that moves beyond the dichotomy between PFI, which breaks all dependencies with features, and CFI, which preserves all dependencies with features. In contrast to PFI and CFI, RFI is based on a perturbation that is restricted to preserve the relationships with a set of variables G that can be chosen arbitrarily. We show that RFI is (1) semantically meaningful and (2) practically useful.

We demonstrate the semantical meaning of RFI in Section IV. In particular, we derive general interpretation rules that link nonzero RFI to (1) the conditional dependence of the feature of interest with the target and non-conditioned features X_R given the conditioned variables X_G in the data and (2) the conditional dependence of the input to the feature of interest X_j with the model’s prediction \hat{Y} given fixed inputs to the remaining features X_R (Theorem 1). Furthermore, we show that a nonzero difference between RFI_j^G and $\text{RFI}_j^{G \cup N}$, with N being an arbitrary set disjoint with G , implies the conditional dependence $X_j \not\perp\!\!\!\perp X_N | X_G$ (Theorem 2).

In Section V, we provide an implementation of RFI estimation that is based on recent results from the related knockoff research field [7], [23]. Furthermore, we translate the testing framework developed for conditional feature importance [30] to RFI. We support our theoretical analysis and findings by various simulation studies in Section VI. In particular, we show that RFI can expose the indirect contribution of variables that are not directly used by the model but provide information via dependent variables (Section VI-A). Similarly, we show how RFI can be used to assess feature importance with respect to variables not included at training time (Section VI-B).

A. Contributions and Related Work

While conditioning on subsets of variables has been suggested before [12], [25], the implications of this generalized variant of CFI have not yet been rigorously analyzed. Some IML methods perturb or hide subsets

This work is funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A and supported by the Bavarian State Ministry of Science and the Arts in the framework of the Centre Digitisation.Bavaria (ZD.B). The authors of this work take full responsibility for its content.

of features, e.g., in the context of multiple regression relative importance analysis is a model-specific technique that averages over all importances of models trained on feature subsets [6], [16]. Model-agnostic, local approximations to the respective feature effect that avoid retraining and instead perturb subsets of features have also been proposed [17], [33]. A very recent global, model-agnostic feature importance proposal called SAGE quantifies feature importance by perturbing multiple features [8].

While the aforementioned approaches are all based on removing several features to provide more nuanced insight into the model, our proposal only modifies the feature of interest. Our approach is model-agnostic and global, while most aforementioned approaches are model-specific or local. The exception is the global, model-agnostic SAGE [8], however the approaches are not only computationally but also semantically different. E.g. our method assigns an importance of zero for features that are not used by the model¹, which is not the case for SAGE. While our approach aims to provide nuanced insights into variable importance relative to a specific set, SAGE aims to quantify the overall importance of variables for the model.

Feature importance relative to variables that have not been included in the training set has not been studied before. The indirect influence of variables that the model does not computationally rely but statistically depend on has been studied e.g. in [1].

II. BACKGROUND AND NOTATION

A. Notation

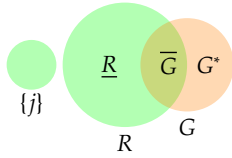


Fig. 1. Overview of our notation.

We denote the target variable, i.e., the variable the model predicts, as Y and feature variables by $X_{(\cdot)}$. We refer to the variables as features to emphasize when they were used in model training. Their observations are denoted by y and $x_{(\cdot)}$. We use $D := \{1, \dots, p\}$ for the index set of all features included in model training and j for the index of our feature of interest, X_j . The index set of the remaining variables is denoted as $R := D \setminus \{j\}$ (rest, remainder). The index set of features, relative to which the importance of X_j is considered, is denoted as G . As G can refer to any index set of variables, we denote its intersection with R as $\bar{G} = R \cap G$ and its complement as $\underline{R} = R \setminus G$. We denote the index set of

conditioning variables that were not made available to the model during training as $G^* = G \setminus R$.

In case we add new elements to the conditioning set G , we will denote this set as N . The set may include variables within and outside D . The respective components are denoted as $N^* = N \setminus R$ and as $\bar{N} = R \cap N$. The remainder of R without G and N is denoted as $\underline{R} = \underline{R} \setminus N$. We denote perturbed variables of interest relative to G as \tilde{X}_j^G . We refer to the original and perturbed probability distribution of X_j as the observational and interventional distribution $P(X_j, \dots)$ and $P(\tilde{X}_j^G, \dots)$. The inspected model is denoted as f , its prediction as \hat{Y} . Independence of Y and X conditional on Z is denoted using $X \perp\!\!\!\perp Y|Z$, the respective conditional dependence as $X \not\perp\!\!\!\perp Y|Z$.

B. Feature Importance

Performance-based feature importance methods assess the relevance of a feature of interest X_j by assessing the impact of a perturbation of X_j on the model's performance. Local feature importance methods focus on the importance of features for specific data points, whereas global feature importance methods assess the impact over the whole domain. In the following, we focus on global methods.

Global feature importance is computed according to the following general schemata:

$$FI_j = \tilde{\mathcal{R}}^j - \mathcal{R} \text{ or } FI_j = \frac{\tilde{\mathcal{R}}^j}{\mathcal{R}}$$

where we denote the original risk of the model and the risk after perturbing X_j as \mathcal{R} and $\tilde{\mathcal{R}}^j$, respectively. For estimation, the true risk \mathcal{R} is replaced with the empirical risk \mathcal{R}_{emp} .

Feature importance methods furthermore differ in how they perturb and whether they rely on retraining the model. While some methods retrain the model after the perturbation (e.g. LOCO, [15]), others evaluate the impact of the perturbation on the same original model (e.g. [5], [25]). In this work, we focus on methods that avoid retraining.

For methods that avoid retraining, we observe a dichotomy between two general perturbation approaches: resampling that preserves the *marginal* and resampling that preserves the *conditional* distribution. Marginal resampling was originally proposed to compute perturbed versions of X_j by permuting the observations $x_j^{(i)}$ within the sample [5]. The respective sample breaks the dependence between X_j and (Y, X_R) while preserving the marginal distribution $P(X_j)$. More recently, Model Reliance was proposed [12], which takes the expectation over all possible permutations. Resampling from the marginal distribution has been criticized to introduce bias, in particular because it overestimates the importance of correlated variables

¹A proof of this property is given in Lemma 2.

[25], resulting in incorrect feature rankings [26]. It also leads to extrapolation under dependent features [14], [19], i.e. conclusions about the model are being drawn using unrealistic data points on which the model was not trained. CFI, on the other hand, samples from the conditional distribution $P(X_j|X_R)$ [2], [7], [12], [14], [19], [25], [28]. A large variety of model-specific methods exist [13], [31]. Conditional variants quantify the importance of a feature given the information that all remaining features R contain about X_j [20], thereby avoiding evaluation of the model on unrealistic datapoints [19].

III. RELATIVE FEATURE IMPORTANCE

Relative Feature Importance is a general framework that assesses feature importance relative to arbitrary variable sets G . The framework subsumes PFI and CFI as two extreme special cases.

In PFI, X_j is replaced with a perturbed version that preserves the marginal distribution $P(X_j)$ while breaking the dependencies with Y and all features. In CFI, a perturbed version of X_j is used that preserves the conditional distribution $P(X_j|X_R)$, thereby only breaking conditional dependence between X_j and Y given all features. As our analysis in Section IV establishes, the replacement strategies of PFI and CFI define extreme reference points. CFI quantifies the contribution relative to *all* remaining features R , whereas PFI regards a feature in isolation. We go beyond the PFI versus CFI dichotomy. We argue that it is (1) meaningful (Section IV) and (2) practically useful (Section VI) to replace X_j with perturbed versions that preserve the conditional distribution $P(X_j|X_G)$ with respect to *arbitrary* sets G while requiring $\tilde{X}_j^G \perp\!\!\!\perp (X_R, Y)|X_G$. G can be a subset of R , but can also include variables not available at training time such that $G \setminus R \neq \emptyset$. We term the resulting method Relative Feature Importance (RFI):

Definition 1 (Relative Feature Importance – RFI): We define Relative Feature Importance with respect to a feature set G with $Y \notin G$ and a fixed model f as

$$RFI_j^G := \tilde{\mathcal{R}}^{j|G} - \mathcal{R},$$

where $\tilde{\mathcal{R}}^{j|G} := \mathcal{R}(Y, f(X_R, \tilde{X}_j^G))$ is the risk w.r.t. to a replacement variable \tilde{X}_j^G and $\mathcal{R} = \mathcal{R}(Y, f(X_j, X_R))$ refers to the original risk. The replacement variable has to satisfy

- $\tilde{X}_j^G \sim P(X_j|X_G)$ and
- $\tilde{X}_j^G \perp\!\!\!\perp (X_R, Y)|X_G$.

In the following section, we discuss the semantic meaning of RFI. The estimation of RFI is discussed in Section V.

IV. INTERPRETING RELATIVE FEATURE IMPORTANCE

IML techniques aim to provide insight into the model and, possibly, into the underlying data generating mechanism. However, IML techniques themselves are subject to interpretation. The characterization of an IML method by its mathematical definition is computationally precise, but has limited aid in guiding users to make conclusions about the underlying model and data. In this section we provide a (non-comprehensive) list of interpretation rules for RFI, that *characterize the method by how it behaves in its context*. This context includes *both the model and the underlying data generating mechanism*. More specifically, we link RFI to (conditional) independence in the underlying data set as well as to whether the model's prediction \hat{Y} is constant in the argument x_j for a fixed x_R . While RFI can be used for quantification of feature importance, we focus our analysis on relevance as a binary property and characterize relative feature relevance ($RFI \neq 0$). We show that the implicit notion of relevance of RFI is defined by the choice of G . By modifying the conditioning set G beyond the PFI versus CFI dichotomy, we are able to gain insight into more nuanced aspects of the model and the data generating mechanism. The main results are given in Theorem 1 and Theorem 2. Furthermore, we highlight limitations stemming from the choice of the loss function L and the model fit for the interpretation, which are, in our humble opinion, underrepresented in the current discussion. We structure our analysis by taking the user's perspective and asking "What can we infer from relative feature relevance?".

A. Implications of Relative Feature Relevance

In the following, we analyze the implications of RFI without further assumptions about model and data. We thereby distinguish between two levels of explanation. Relative feature relevance provides insight, both into *model* and *data*.

Theorem 1: If $RFI_j^G \neq 0$ then

- $X_j \not\perp\!\!\!\perp (Y, X_R)|X_G$ in the underlying distribution (data level)
- $\tilde{X}_j \not\perp\!\!\!\perp \hat{Y}|X_R$ w.r.t. the interventional distribution $P(X_j|X_G)P(X_G, X_R) > 0$ (model level)

We prove Theorem 1 in two steps. First, we assess the implications of the respective independence for the underlying data set (Lemma 1). Then, we assess the implications of the respective independence for the model (Lemma 2). The contrapositions yield Theorem 1.

Lemma 1: If $X_j \perp\!\!\!\perp (Y, X_R)|X_G$ for any G with $Y \notin G$ then $RFI_j^G = 0$.

We base the proof of Lemma 1 on the insight that (because the model f is fixed) an equivalence in distribution implies an equivalence in risk (Proposition 1). Therefore conditions under which the interventional distribution $P(\tilde{X}_j^G, X_R, Y)$ coincides with the original distribution $P(X_j, X_R, Y)$ are sufficient for $RFI = 0$.

Proposition 1: If observational and interventional distribution coincide, then risks with and without perturbation are equal:

$$P(Y, X_j, X_R) = P(Y, \tilde{X}_j^G, X_R) \Rightarrow \mathcal{R}(f) = \tilde{\mathcal{R}}^{jG}(f)$$

Proof of Proposition 1: Given that $P(Y, X_j, X_R) = P(Y, \tilde{X}_j^G, X_R)$ we can write

$$\begin{aligned} \mathcal{R}(f) &= \mathbb{E}_{Y, X_j, X_R} [L(Y, f(X_j, X_R))] \\ &= \mathbb{E}_{Y, \tilde{X}_j^G, X_R} [L(Y, f(\tilde{X}_j^G, X_R))] = \tilde{\mathcal{R}}(f). \end{aligned}$$

We show next that the conditional independence $X_j \perp\!\!\!\perp (X_R, Y) | X_G$ is a sufficient condition for identity of both distributions.

Proof of Lemma 1: It holds that

$$\begin{aligned} P(Y, X_j, X_R, X_G) &= P(X_j | Y, X_R, X_G) P(Y, X_R, X_G) \\ &\stackrel{X_j \perp\!\!\!\perp (X_R, Y) | X_G}{=} P(X_j | X_G) P(Y, X_R, X_G) \\ &\stackrel{(\text{def})}{=} P(\tilde{X}_j^G | X_G) P(Y, X_R, X_G) \\ &= P(\tilde{X}_j^G, Y, X_R, X_G). \end{aligned}$$

Using Proposition 1 we can infer that $RFI_j^G = 0$. \blacksquare

So far, we have assessed implications for the underlying data generating mechanism. Next, we assess implications for the inspected model f .

Lemma 2: If $\tilde{X}_j^G \perp\!\!\!\perp \hat{Y} | X_R$ w.r.t. the interventional distribution $P(\tilde{X}_j^G, X_G, X_R)$ then $RFI_j^G = 0$ for any G .

Proof of Lemma 2: If the prediction for an observation (x_1, \dots, x_p) is independent of the value x_j' w.r.t. the interventional distribution, the prediction is unaffected when replacing x_j with any value x_j' with $P(x_j' | X_G = x_G) P(X_G = x_G, X_R = x_R) > 0$. Consequently, any sample from \tilde{X}_j^G yields the same prediction. Furthermore values x_j' with nonzero probability over the interventional distribution also have nonzero probability over the observational distribution. The interventional distribution can be rewritten as

$$\begin{aligned} P(\tilde{X}_j^G, X_G, X_R) &= P(\tilde{X}_j^G | X_G, X_R) P(X_G, X_R) \\ &= P(\tilde{X}_j^G | X_G) P(X_G, X_R) \\ &= P(X_j | X_G) P(X_G, X_R). \end{aligned}$$

Similarly, the observational distribution can be factorized into $P(X_j | X_G, X_R) P(X_G, X_R)$. As $P(X_j | X_G, X_R) > 0 \Rightarrow P(X_j | X_G) > 0$ (which can be derived from, e.g., the law of total probability) it follows that $P(\tilde{X}_j^G, X_G, X_R) > 0 \Rightarrow P(X_j, X_G, X_R) > 0$.

Consequently the prediction \hat{y} for any value x_j with positive probability $P(X_j = x_j | X_R = x_R)$ is identical given unchanged x_R .

As the conditional distributions of X_j and \tilde{X}_j^G overlap and the distribution of X_R is unaffected, the prediction \hat{Y} is identical with and without perturbation. Therefore $\mathcal{R} = \tilde{\mathcal{R}}^{jG}$ and $RFI_j^G = 0$. \blacksquare

To summarize, we have shown that independence on the dataset and on the model level respectively imply $RFI_j^G = 0$ and can thereby prove Theorem 1.

Proof of Theorem 1: The result follows from contraposition of Lemma 1 and contraposition of Lemma 2. \blacksquare

Theorem 1 shows that nonzero RFI_j^G implies dependencies between sets of variables on the model level as well as on the data level. Which dependencies are relevant for RFI_j^G can be controlled with the conditioning set G . Consequently, the conditioning set G determines the method's implicit definition of relevance. I.e., on the data level, if $X_j \perp\!\!\!\perp (X_R, Y) | X_G$ holds, RFI_j^G is zero irrespective of any other dependencies that may hold, e.g. with X_G (Lemma 1). Nonzero RFI, a difference in performance on interventional and observational distribution, can only be caused by dependencies that have been destroyed in the interventional distribution, the dependencies with and via X_G are preserved by the replacement \tilde{X}_j^G and can therefore not be responsible for $RFI_j^G \neq 0$. Similarly, on the model level, $\tilde{X}_j^G \perp\!\!\!\perp \hat{Y} | X_R$ over the interventional distribution $P(X_j | X_G) P(X_G, X_R)$ yields zero RFI (Lemma 2). The behavior of the model outside the domain in which it is evaluated is irrelevant for RFI_j^G . What domain the model is evaluated over depends on the choice of G .

Because we can control RFI's implicit definition of relevance with G , RFI allows more nuanced insights into model and data than PFI or CFI alone. In Theorem 1, we aim to make the implicit definition of relevance explicit. On the data level, nonzero RFI implies the dependence of X_j with the tuple (Y, X_R) given X_G ($X_j \not\perp\!\!\!\perp (Y, X_R) | X_G$). In order to understand the aforementioned dependence, using the graphoid axioms contraction and weak union [22], the equivalent formulation below can be adduced:

$$(X_j \not\perp\!\!\!\perp Y | X_G) \vee (X_j \not\perp\!\!\!\perp X_R | X_G, Y).$$

At least one of the two conditional dependencies has to hold for nonzero RFI_j^G . The first dependence can be rephrased as: X_j is informative of Y , even if we already know X_G . It is more difficult to make sense of the second

dependence. Under dependent features ($X_j \not\perp X_R | X_G, Y$), the distribution of X_j with X_R is not preserved under perturbation \tilde{X}_j^G . In the interventional distribution $P(\tilde{X}_j^G, X_R)$ observations that are improbable or impossible w.r.t. the observational distribution $P(X_j, X_R)$ can be possible and probable (and vice versa). Consequently, in the interventional distribution the feature distribution differs from the observation feature distribution. Even if $X_j \perp Y | X_G$ holds, the model may perform suboptimally due to this distribution shift and cause RFI_j^G nonzero². If the conditioning set is a superset of R ($G \supseteq R$), such that set of remaining variables X_R is empty, it holds that $(X_j \perp X_R | X_G, Y)$. Therefore nonzero RFI must be attributed to $(X_j \not\perp Y | X_G)$ for $G \supseteq R$.

On the model level, nonzero RFI implies that the model's predictions are conditionally dependent on \tilde{X}_j^G given the remaining features R are fixed. E.g. for a linear model that has coefficient zero for all terms involving X_j , this dependence would not be fulfilled, and RFI_j^G would be zero (Lemma 2). The model is evaluated over the interventional distribution $P(X_j | X_G)P(X_G, X_R) > 0$, which varies depending on G . If G contains a nearly perfect correlate of X_j , X_j can be reconstructed well. In contrast, if $G = \emptyset$, for every possible x_R the model is evaluated over the whole marginal distribution of X_j . Although choosing a smaller set $G \subset R$ leads to extrapolation under dependent features, it allows more insight into the model's mechanism. For interpretation purposes like safety, this is highly desirable.

In the preceding paragraphs we have highlighted the importance of the conditioning set G for the method's implicit notion of relevance and illustrated the results from Theorem 1. We have argued that the conditioning set controls which potential dependencies can be responsible for nonzero RFI_j^G . The insights lead to a further, interesting application of RFI. By assessing the difference $\Delta \text{RFI}_j^{G \rightarrow GUN} = \text{RFI}_j^G - \text{RFI}_j^{GUN}$ when modifying the conditioning set G by adding new elements N , we are able to assess the role of the dependencies with variables in N relative to a baseline G . While for RFI_j^G only dependencies of X_j with and via G are preserved, for RFI_j^{GUN} also dependencies with and via N are maintained. If $\Delta \text{RFI}_j^{G \rightarrow GUN}$ is nonzero, this change has to be due to dependencies involving N , but not G . We substantiate this claim with Theorem 2. In order for $\Delta \text{RFI}_j^{G \rightarrow GUN}$ to be positive, the dependence $X_j \not\perp X_N | X_G$ has to hold.

Theorem 2: If the difference $\Delta \text{RFI}_j^{G \rightarrow GUN} = \text{RFI}_j^G - \text{RFI}_j^{GUN} \neq 0$, then $X_j \not\perp X_N | X_G$.

²Let e.g. X_1, X_2 be perfectly correlated and independent of Y . Then adding $X_1 - X_2$ does not alter its prediction performance, unless the dependence between the variables is broken. Also see [14] for a discussion in PFI.

Proof of Theorem 2: Under independence $X_j \perp X_N | X_G$ it holds that

$$\begin{aligned} P(\tilde{X}_j^G, Y, X_R, X_G, X_N) &= P(\tilde{X}_j^G | Y, X_R, X_G, X_N) P(Y, X_R, X_G, X_N) \\ &\stackrel{(\text{def } \tilde{X}_j^G)}{=} P(X_j | X_G) P(Y, X_R, X_G, X_N) \\ &\stackrel{X_j \perp X_N | X_G}{=} P(X_j | X_G, X_N) P(Y, X_R, X_G, X_N) \\ &\stackrel{(\text{def } \tilde{X}_j^{GUN})}{=} P(\tilde{X}_j^{GUN} | X_G, X_N) P(Y, X_R, X_G, X_N) \\ &\stackrel{(\text{def } \tilde{X}_j^{GUN})}{=} P(\tilde{X}_j^{GUN} | Y, X_G, X_N, X_R) P(Y, X_R, X_G, X_N) \\ &= P(\tilde{X}_j^{GUN}, Y, X_R, X_G, X_N) \end{aligned}$$

The equality $P(\tilde{X}_j^G, Y, X_R, X_G, X_N) = P(\tilde{X}_j^{GUN}, Y, X_R, X_G, X_N)$ implies $P(\tilde{X}_j^G, Y, X_R) = P(\tilde{X}_j^{GUN}, Y, X_R)$. Invoking Proposition 1 it holds that the corresponding risks \mathcal{R}^{jG} and \mathcal{R}^{jGUN} are equal. As $\text{RFI}_j^G - \text{RFI}_j^{GUN} = \mathcal{R}^{jG} - \mathcal{R}^{jGUN}$ it holds that $X_j \not\perp X_N | X_G \Rightarrow \Delta \text{RFI}_j^{G \rightarrow GUN} = 0$. Contraposition proves Theorem 2. \blacksquare

While nonzero RFI_j^G as well as nonzero $\Delta \text{RFI}_j^{G \rightarrow GUN}$ have clear implications, interpreting zero RFI_j^G or zero $\Delta \text{RFI}_j^{G \rightarrow GUN}$ is difficult. For example, we may be tempted to interpret $\text{RFI}_j^G = 0$ as conditional independence in the data. However, the general principle that absence of evidence is no evidence for absence also applies in the context of RFI. A dependence in the data may not be captured by the model when it has a poor fit and does not rely on the respective variable. Similarly, although f may be optimal, a dependence in higher moments may simply not be modeled by f or captured by the loss L . As all aforementioned causes of nonzero RFI are potentially sufficient, but not necessary, it is unclear which of the causes nonzero RFI can be attributed to. Furthermore, the related problem of conditional independence testing is provably hard [24].

The theoretical insights that we derive in this Section (Theorem 1 and 2) are applied and illustrated in a simulation study in Section VI.

V. ESTIMATION AND TESTING

Estimating and sampling from the conditional distribution is in general difficult, especially in high-dimensional continuous settings. Various approaches for replacing X_j with samples from its conditional distribution exist, e.g., knockoff approaches [2], [7], [23], imputation and weighting [12] or permutation within decision tree leaves [18]. We used Model-X knockoffs [7] in this work, but note that the RFI approach is agnostic to its algorithmic implementation.

Using (standard) empirical risk estimates, our RFI estimate is

$$\hat{\text{RFI}}_j^G = \frac{1}{n} \sum_{i=1}^n L(y^{(i)}, f(\hat{x}_j^{(i)}, x_R^{(i)})) - \frac{1}{n} \sum_{i=1}^n L(y^{(i)}, f(x_j^{(i)}, x_R^{(i)}))$$

where $\hat{x}_j^{(i)}$ is a sample from \tilde{X}_j^G . We can then test for nonzero RFI_j^G using procedures for conditional independence tests, e.g., [30], thereby quantifying the uncertainty coming from empirical risk minimization. Because of the central limit theorem, the empirical risk converges (in probability) to a Gaussian distribution with increasing number of observations. Therefore, one-sided, paired t-tests can be used to infer tests and confidence intervals [30]. The test procedures proposed in [30] are agnostic to the conditioning set for the perturbation \tilde{X}_j^G . For smaller samples, the Exact Test by Fisher may be used.

The t-test and Fisher Exact Test ignore uncertainty and bias of the estimation procedures, i.e. the ML model and the knockoff-sampler are treated as “fixed”. E.g. misspecified, suboptimal models may not capture dependencies. Or dependencies are in higher moments that are not captured by the loss. Consequently, without further assumptions, the framework does not provide a test for conditional independence in the dataset.

The popular testing procedures for knockoffs proposed by [7] provide FDR over all features, but does not test the significance of the importance of individual features.

VI. SIMULATION STUDIES

In the following, we demonstrate the usefulness of RFI on two simulation studies. In the first example, we use RFI to expose indirect influence of variables that are not computationally used by the model. In the second example, we assess feature importance relative to a confounder that was unavailable at training time. In both examples, we represent the underlying data generating mechanism, that gives rise to the dependencies in the data, with a causal directed acyclic graph (DAG). The code for the examples is available online³.

A. Indirect Influence

A prominent application of interpretable machine learning is auditing models regarding its reliance on protected attributes A like age or sex. A reliance on the respective attributes may result in unfair discrimination and requires further inspection. With approaches like fairness through unawareness [3], the model does not rely on protected attributes directly. However, by implicitly reconstructing the sensitive attributes using seemingly harmless correlates, the model can indirectly make use of the protected attribute resulting in potentially harmful, unfair discrimination [3].

³Link to Code: <https://github.com/gcskoenig/icpr2020-rfi>

PFI and CFI cannot expose such indirect influence. As Lemma 2 proves, RFI_A^G is zero for a model that does not (directly) use the feature of interest A for the prediction for any conditioning set G . Furthermore, from PFI and CFI alone, we cannot infer whether the importance of a variable can be attributed to its dependence with an indirect influence. Using RFI_j^G with $G = A$ we preserve the influence of A on the prediction and can thereby restrict the attribution of importance to contributions stemming from dependencies not involving A (Theorem 1, Lemma 1). The difference to $\Delta \text{RFI}_j^{G \rightarrow \text{GUN}}$ with $G = \emptyset$ and $N = A$ exposes the indirect influence.

Not every indirect influence from a sensitive attribute is considered undesirable. Certain correlates of A may indeed be valid criteria for a decision (e.g. [4]). Importance stemming from dependencies with A via such resolving variables Z would be considered acceptable. We can assess the indirect influence beyond contributions stemming from dependence via Z by comparing to a baseline $G = Z$. In this baseline, contributions via Z are preserved and therefore irrelevant for RFI. Consequently, when setting $N = A$, the difference $\Delta \text{RFI}_j^{G \rightarrow \text{GUN}}$ only quantifies indirect influence that is not resolved by Z .

We demonstrate the usefulness of RFI to expose indirect influence in a simulation study. The dataset is a sample drawn from the distribution induced by a structural causal model (SCM) depicted in Figure 2. All relationships are additive linear with coefficients 1 and Gaussian noise terms ($\sigma_1 = \sigma_2 = \sigma_4 = 1$, $\sigma_3 = 0.3$ and $\sigma_y = 0.5$). An ordinary least squares linear regression model was fit to predict Y from X_1, \dots, X_4 ($\text{MSE} = 0.25$, $f(x_1, x_2, x_3, x_4) = 0.00x_1 - 0.01x_2 + 1.01x_3 + 1.00x_4$). We trained model-X knockoffs [7] on the training data and evaluated RFI on test data. Sample size is 10^5 with 10% test data.

In order to quantify the direct influence of the features we compute PFI. As we can see in Figure 3, X_1 and X_2 are considered irrelevant. In order to expose their indirect influence, we additionally compute RFI with respect to $G = \{X_1\}$ and $G = \{X_2\}$ respectively. For both variables we observe a drop in importance of X_3 and X_4 . Consequently both X_1 and X_2 have an indirect influence on the target (Theorem 2).

Furthermore we are interested in whether the indirect influence of X_1 can be resolved by X_2 . We therefore compute $\text{RFI}_j^{\text{GUN}}$ with $G = \{X_2\}$ and $N = \{X_1\}$. We see that for X_3 no change in importance can be observed. This is due to the independence $X_1 \perp\!\!\!\perp X_3 | X_2^4$ (Theorem 2). The indirect influence is resolved. However, for X_4 the importance decreases further and is therefore not resolved by X_2 . This is in alignment with the dependence $X_1 \not\perp\!\!\!\perp X_4 | X_2$ implied by the graph (Figure 2).

⁴As faithfulness and causal markov condition hold, d -separation in the graph and (conditional) independence coincide [21]. We can therefore read the independence structures off Figures 2 and 4.

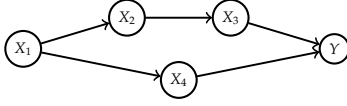


Fig. 2. Variable X_1 influences Y both via the chain $X_2 \rightarrow X_3$ and via X_4 . X_1 may be some undesired influence, and X_2 a variable resolving the undesired influence. We find that the prediction can nevertheless be influenced via X_4 by comparing $RFI_{X_4}^{X_2}$ with $RFI_{X_4}^{X_2, X_1}$ (Figure 3). All relationships are additive linear Gaussian with all coefficients being equal to 1 and $\sigma_1 = \sigma_2 = \sigma_4 = 1$, $\sigma_3 = 0.3$ and $\sigma_Y = 0.5$.

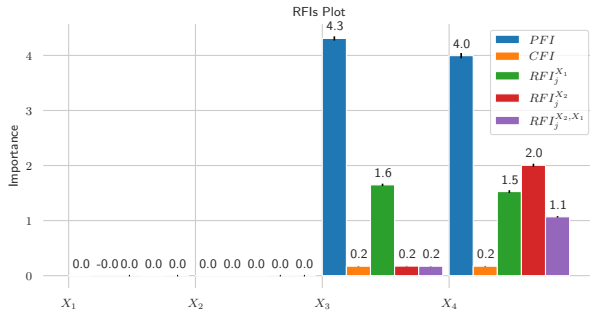


Fig. 3. RFI's for a linear regression model fitted on the dataset illustrated in Figure 2. Feature importance values are averaged over 30 runs and rounded. Feature importance values are averaged over 30 runs and rounded. We evaluated significance using a t-test for the first run. All positive features were significant at $\alpha = 0.01$, whereas for all zero RFI values the null could not be rejected. For X_1 and X_2 all RFIs are zero, whereas for X_3 and X_4 RFIs are positive. We see that X_1 and X_2 both have an indirect influence on X_3 and X_4 , but that X_2 can resolve the influence of X_1 on X_3 .

B. Variables Outside Training Set

When designing a model f , a practitioner may have decided to exclude a variable from the feature set, e.g., because it was then considered irrelevant, it belongs to a different modality or would have required further preprocessing. Furthermore, when auditing a machine learning model f , variables that have not been available for the training of the model may be accessible.

In this example, we demonstrate that variables outside the training set can be included in the conditioning set for RFI. Consequently, importance of the features relative to variables outside the training set and the indirect influence of such variables can be assessed. More specifically, we simulate a hypothetical situation where the influence of a previously unknown confounder C shall be evaluated. This variable C is available for the model audit. In particular, we wonder whether the features X_1 , X_2 and X_3 are only or partly important due to a dependence via C .

The dataset was sampled from a structural causal model (SCM) depicted in Figure 4. Assuming faithfulness and the causal Markov condition, this DAG implies the following (conditional) (in-)dependencies: X_1 is independent of C , X_3 is independent of Y conditional on C , and

X_2 is dependent on Y . Note that the dependence between X_2 and Y is due to the common cause C as well as due to a direct effect of X_2 on Y . All relationships are additive linear with coefficients 1 and additive Gaussian noise ($\sigma_1 = \sigma_2 = \sigma_C = 1.0$ and $\sigma_Y = 0.5$). We fit an ordinary least squares linear regression model on X_1 , X_2 and X_3 to predict Y ($MSE = 0.40$, $f(x_1, x_2, x_3) = 1.0x_1 + 1.17x_2 + 0.67x_3$). C was not available for model training. We trained Model-X knockoffs [7] on training data and sampled from \tilde{X}_j^G on test data. Sample size is 10^5 with 10% test data.

When computing RFI_j^C ($G = \{C\}$) for each variable, the different relationships with C become apparent. The respective results are depicted in Figure 5. For X_1 the feature importance relative to C remains unchanged as the variables are pairwise independent (Theorem 2). For X_3 , that is only dependent with Y via C , it completely vanishes (Lemma 1). For X_2 the feature importance decreases but remains nonzero, as X_2 is dependent with Y directly and via C .

Consequently, using RFI, we can (1) identify variables that are important due to a variable unavailable at training time and (2) distinguish between variables that only depend on Y via C from those that do not. With PFI ($G = \emptyset$) or CFI ($G = R$) such a distinction is in general not possible.

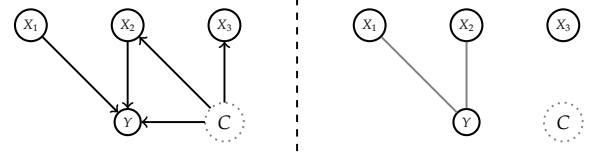


Fig. 4. Left: We see the causal graph \mathcal{G} corresponding to the Structural Causal Model that was used to generate the dataset used in Figure 5. All relationships are additive linear Gaussian with all coefficients equal to 1 and $\sigma_1 = \sigma_2 = \sigma_C = 1.0$ and $\sigma_Y = 0.5$. Right: Pairwise dependencies after conditioning on C .

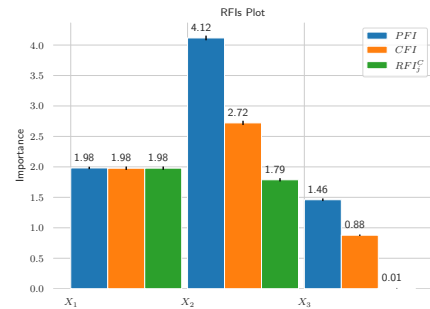


Fig. 5. Feature Importance results corresponding to the dataset depicted in Figure 4. We averaged RFI over 30 runs. RFI for X_1 is unaffected by changes in G , for X_2 RFI drops with C is added to G . For X_3 RFI vanishes relative to C . For all except for $RFI_{X_3}^C$ the null can be rejected at $\alpha = 0.01$ in the first run.

VII. DISCUSSION

We proposed relative feature importance (RFI), a general conditional feature importance framework which allows to condition on arbitrary sets of other features, including features outside the training set. We underpin the method with theoretical results allowing insight into both model and underlying dataset. In a simulation study, the usefulness of the method for the exposure of indirect influence is demonstrated.

Relative feature importance requires sampling from (unknown) conditional distributions. For continuous variables and in high-dimensional settings this task is challenging and an open area of research [7], [23]. Uncertainty stemming from inaccurate sampling may affect the interpretation. The quality of insight into the underlying dataset strongly depends on the training and evaluation of the model. Dependencies in higher moments are usually not modeled and not captured by standard loss functions and can therefore not be detected. Especially the interpretation of zero RFI requires careful assessment of the model specification. Further research is needed to assess necessary assumptions for the interpretation of RFI. These challenges are not unique to RFI, but apply more generally in the field of interpretable machine learning [20].

REFERENCES

- [1] Philip Adler, Casey Falk, Sorelle A. Friedler, Tionney Nix, Gabriel Rybeck, Carlos Scheidegger, Brandon Smith, and Suresh Venkatasubramanian. Auditing black-box models for indirect influence. *Knowledge and Information Systems*, 54(1):95–122, 2018. arXiv: 1602.07043.
- [2] Rina Foygel Barber, Emmanuel J Cands, and others. Controlling the false discovery rate via knockoffs. *The Annals of Statistics*, 43(5):2055–2085, 2015. Publisher: Institute of Mathematical Statistics.
- [3] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. <http://www.fairmlbook.org>.
- [4] Vence L Bonham, Shawneequa L Callier, and Charmaine D Royal. Will precision medicine move us beyond race? *The New England journal of medicine*, 374(21):2003, 2016.
- [5] Leo Breiman. Random forests. *Machine Learning*, pages 1–122, 2001.
- [6] David V Budesu. Dominance analysis: a new approach to the problem of relative importance of predictors in multiple regression. *Psychological bulletin*, 114(3):542, 1993.
- [7] Emmanuel Cands, Yingying Fan, Lucas Janson, and Jinchi Lv. Panning for gold: model-X knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 80(3):551–577, 2018. arXiv: 1610.02351.
- [8] Ian Covert, Scott Lundberg, and Su-In Lee. Understanding Global Feature Contributions Through Additive Importance Measures. arXiv preprint arXiv:2004.00668, 2020.
- [9] Jeffrey (Reuters) Dastin. Amazon scraps secret AI recruiting tool that showed bias against women. *Reuters*, 2018.
- [10] Lydia de la Torre. A Guide to the California Consumer Privacy Act of 2018. *SSRN Electronic Journal*, pages 1–17, 2018.
- [11] Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science Advances*, 4(1):1–6, 2018.
- [12] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.
- [13] Ulrike Grmping. Variable importance assessment in regression: linear regression versus random forest. *The American Statistician*, 63(4):308–319, 2009. Publisher: Taylor & Francis.
- [14] Giles Hooker and Lucas Mentch. Please Stop Permuting Features: An Explanation and Alternatives. arXiv preprint arXiv:1905.03151v, pages 1–15, 2019. arXiv: 1905.03151v1.
- [15] Jing Lei, Max Gsell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.
- [16] Stan Lipovetsky and Michael Conklin. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330, 2001. Lipovetsky2001.
- [17] Scott M. Lundberg and Su In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 2017-Decem(Section 2):4766–4775, 2017. arXiv: 1705.07874.
- [18] Christoph Molnar, Gunnar König, Bernd Bischl, and Giuseppe Casalicchio. Model-agnostic feature importance and effects with dependent features—a conditional subgroup approach. arXiv preprint arXiv:2006.04628, 2020.
- [19] Christoph Molnar, Gunnar König, Bernd Bischl, and Giuseppe Casalicchio. Model-agnostic feature importance and effects with dependent features—a conditional subgroup approach. arXiv preprint arXiv:2006.04628, 2020.
- [20] Christoph Molnar, Gunnar König, Julia Herbringer, Timo Freiesleben, Susanne Dandl, Christian A. Scholbeck, Giuseppe Casalicchio, Moritz Grosse-Wentrup, and Bernd Bischl. Pitfalls to avoid when interpreting machine learning models. arXiv preprint arXiv:2007.04131, 2020.
- [21] Judea Pearl. *Causality*. Cambridge university press, 2009.
- [22] Judea Pearl and Azaria Paz. Graphoids: A graph-based logic for reasoning about relevance relations. University of California (Los Angeles). Computer Science Department, 1985.
- [23] Yaniv Romano, Matteo Sesia, and Emmanuel Cands. Deep knockoffs. *Journal of the American Statistical Association*, pages 1–12, 2019. Publisher: Taylor & Francis.
- [24] Rajen D Shah and Jonas Peters. The hardness of conditional independence testing and the generalised covariance measure. arXiv preprint arXiv:1804.07203, 2018.
- [25] Carolin Strobl, Anne Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. Conditional variable importance for random forests. *BMC Bioinformatics*, 9:1–11, 2008.
- [26] Laura ToloÁi and Thomas Lengauer. Classification with correlated features: unreliability of feature ranking and solutions. *Bioinformatics*, 27(14):1986–1994, 2011. Publisher: Oxford University Press.
- [27] Eric J Topol. High performance medicine: the convergence of human and artificial intelligence. *Nature Medicine*, 25(January), 2019. Publisher: Springer US.
- [28] Eugene Tuv, Alexander Borisov, George Runger, and Kari Torkkola. Feature selection with ensembles, artificial variables, and redundancy elimination. *Journal of Machine Learning Research*, 10(Jul):1341–1366, 2009.
- [29] Paul Voigt and Axel dem Bussche. The eu general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing, 2017. Publisher: Springer.
- [30] David S. Watson and Marvin N. Wright. Testing Conditional Independence in Supervised Learning Algorithms. arXiv preprint arXiv:1901.09917, 2019. arXiv: 1901.09917.
- [31] Pengfei Wei, Zhenzhou Lu, and Jingwen Song. Variable importance analysis: a comprehensive review. *Reliability Engineering & System Safety*, 142:399–432, 2015. Publisher: Elsevier.
- [32] Yufei Xia, Chuanzhe Liu, Yu Ying Li, and Nana Liu. A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, 78:225–241, 2017. Publisher: Elsevier Ltd.
- [33] Erik Átrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014. Publisher: Springer.

11. Visualizing the Feature Importance for Black Box Models

Contributing article:

Casalicchio, G., Molnar, C., and Bischl, B. (2018). Visualizing the Feature Importance for Black Box Models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 655-670).

Copyright information:

Springer Nature Switzerland AG 2019

Author contributions:

The whole paper and the accompanying R package were written by Giuseppe Casalicchio. Christoph Molnar and Giuseppe Casalicchio discussed many theoretical concepts in-depth which shaped the paper, especially concerning Shapley values, Shapley feature importance and the commonalities between the PDP and PFI. All authors suggested modification, proofread and revised the paper.

Supplementary material available at:

R package: <https://github.com/giuseppec/featureImportance>

Visualizing the Feature Importance for Black Box Models

Giuseppe Casalicchio (✉), Christoph Molnar, and Bernd Bischl

Department of Statistics
Ludwig-Maximilians-University Munich
Ludwigstraße 33, 80539 Munich, Germany
`giuseppe.casalicchio@stat.uni-muenchen.de`

Abstract. In recent years, a large amount of model-agnostic methods to improve the transparency, trustability, and interpretability of machine learning models have been developed. Based on a recent method for model-agnostic global feature importance, we introduce a local feature importance measure for individual observations and propose two visual tools: partial importance (PI) and individual conditional importance (ICI) plots which visualize how changes in a feature affect the model performance on average, as well as for individual observations. Our proposed methods are related to partial dependence (PD) and individual conditional expectation (ICE) plots, but visualize the expected (conditional) feature importance instead of the expected (conditional) prediction. Furthermore, we show that averaging ICI curves across observations yields a PI curve, and integrating the PI curve with respect to the distribution of the considered feature results in the global feature importance. Another contribution of our paper is the Shapley feature importance, which fairly distributes the overall performance of a model among the features according to the marginal contributions and which can be used to compare the feature importance across different models.

Keywords: Interpretable Machine Learning · Explainable AI · Feature Importance · Variable Importance · Feature Effect · Partial Dependence.

1 Introduction and Related Work

Machine learning (ML) algorithms such as neural networks and support vector machines (SVM) are often considered to produce black box models because they do not provide any direct explanation for their predictions. However, these methods often outperform simple linear models or decision trees in predictive performance as they can model complex relationships in the data. Nevertheless, such simple models are still preferred in areas such as life sciences and social sciences due to their simplicity and interpretability [14]. Many researchers have therefore developed and implemented several model-agnostic interpretability tools, which quantify or visualize feature effects or feature importance [9, 11, 17].

In our context, the terms *feature effect*, *feature contribution* and *feature attribution* describe how or to what extent each feature contributes to the *prediction*

of the model, either on a local or a global level. Methods for feature effects include partial dependence (PD) plots [10], individual conditional expectation (ICE) plots [11] and, more recently, SHAP values [15]. These methods visualize or quantify the relationship and contribution of each feature to the prediction of a model without requiring knowledge about the true values of the target variable. A method that measures feature effects based on the Shapley value [19] from coalitional game theory was first presented for classification in [21] and has been extended to regression and global analysis in [22]. Further developments, visualizations, and generalizations were introduced by [15, 16]. Similar work proposing a general notion of a quantity of interest for the characteristic function of the Shapley value and focusing on the joint and marginal contributions of feature sets was introduced by [8].

In biomedical research, for example, measuring the effects of biomedical markers w.r.t. model prediction is as essential as measuring their added value regarding model performance [4]. We use the term *feature importance*¹ to describe how important the feature was for the *predictive performance* of the model, regardless of the shape (e.g., linear or nonlinear relationship) or direction of the feature effect. This implies that measures of feature importance require knowledge of the true values of the target variable. The most prominent approach is the permutation importance introduced by Breiman [3] for random forests. It computes the drop in out-of-bag performance after permuting the values of a feature. A model-agnostic global permutation-based feature importance (PFI) was recently introduced in [9].

Contributions: We review model-agnostic global PFI and propose an efficient approximation based on Monte-Carlo integration. We then introduce a local version of the global PFI, which measures the feature importance of individual observations. We provide visualizations for local and global PFI, which illustrate how changes in the considered feature affect model performance. We also relate our new visual tools to PD plots, ICE plots and show that the integral of our PI curve results in the global PFI measure. Furthermore, we propose a permutation-based Shapley feature importance (SFIMP) measure that fairly distributes the model performance among features and allows the comparison of feature importances across different models.

2 Preliminaries and Background on Feature Effects

In this section, we introduce the notation and describe methods focusing on feature effects, which we transfer to feature importance in Section 4 and 5.

General Notation: Consider a p -dimensional feature space $\mathcal{X}_P = (\mathcal{X}_1 \times \dots \times \mathcal{X}_p)$ with the feature index set $P = \{1, \dots, p\}$ and a target space \mathcal{Y} . Suppose that there is an unknown functional relationship f between \mathcal{X}_P and \mathcal{Y} . ML algorithms try to learn this relationship using training data with observations

¹ In the literature, the term feature importance is sometimes also used for methods that only work with model predictions. In our context, however, we would categorize them under feature effects as they do not take into account the model performance.

that have been drawn i.i.d. from an unknown probability distribution \mathcal{P} on the joint space $\mathcal{X}_P \times \mathcal{Y}$. We consider an arbitrary prediction model \hat{f} , fitted on some training data to approximate f and analyze it with model-agnostic interpretability methods. Let $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$ be a test data set sampled i.i.d. from \mathcal{P} where n is the number of observations in the test set. We denote the corresponding random variables generated from the feature space by $X = (X_1, \dots, X_p)$ and the random variable generated from the target space by Y . In our notation, the vector $\mathbf{x}^{(i)} = (x_1^{(i)}, \dots, x_p^{(i)})^\top \in \mathcal{X}_P$ refers to the i -th observation, which is associated with the target variable $y^{(i)} \in \mathcal{Y}$, and $\mathbf{x}_j = (x_j^{(1)}, \dots, x_j^{(n)})^\top$ denotes the realizations of the j -th feature. We denote the generalization error of a fitted model, which is measured by a loss function L on unseen test data from \mathcal{P} , by $GE(\hat{f}, \mathcal{P}) = \mathbb{E}(L(\hat{f}(X), Y))$. It can be estimated using the test data \mathcal{D} by

$$\widehat{GE}(\hat{f}, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^n L(\hat{f}(\mathbf{x}^{(i)}), y^{(i)}). \quad (1)$$

A better estimate for the generalization error of an ML algorithm can be obtained using resampling techniques such as cross-validation or bootstrap [1].

PD Plots [10] visualize the marginal relationship between features of interest and the expected prediction of a fitted model on a global level. Consider a subset of feature indices $S \subseteq P$ and its complement C . Each observation $\mathbf{x} \in \mathcal{X}_P$ can be partitioned into $\mathbf{x}_S \in \mathcal{X}_S$ and $\mathbf{x}_C \in \mathcal{X}_C$ containing only features from S and C , respectively. Let X_S and X_C be the corresponding random variables and let the prediction function using features in S , marginalized over features in C be the PD function defined by $f_S(\mathbf{x}_S) = \mathbb{E}_{X_C}(\hat{f}(\mathbf{x}_S, X_C))$. This definition also covers $f_\emptyset(\mathbf{x}_\emptyset)$ and results in a constant, the average prediction over \mathcal{P} . We can estimate the PD function using Monte-Carlo integration by averaging over feature values $\mathbf{x}_C^{(i)}$ in order to marginalize out features in C :

$$\hat{f}_S(\mathbf{x}_S) = \frac{1}{n} \sum_{i=1}^n \hat{f}_S^{(i)}(\mathbf{x}_S) = \frac{1}{n} \sum_{i=1}^n \hat{f}(\mathbf{x}_S, \mathbf{x}_C^{(i)}). \quad (2)$$

Here, $\hat{f}_S^{(i)}(\mathbf{x}_S) = \hat{f}(\mathbf{x}_S, \mathbf{x}_C^{(i)})$ can be read in two ways: a) the prediction of the i -th observation with replaced feature values in S taken from \mathbf{x} or b) the prediction of \mathbf{x} with replaced values in C taken from the i -th observation. Plotting the pairs $\{(\mathbf{x}_S^{*(k)}, \hat{f}_S(\mathbf{x}_S^{*(k)}))\}_{k=1}^m$ using (often $m < n$) grid points denoted by $\mathbf{x}_S^{*(1)}, \dots, \mathbf{x}_S^{*(m)}$ yields a PD curve. Fig. 1 illustrates the PD principle for a simple example.

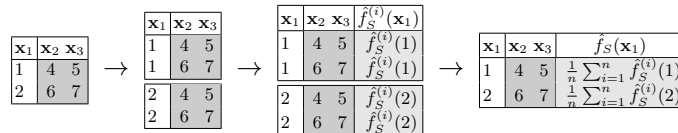


Fig. 1. PD plot for an example with $n = 2$, $p = 3$ and $S = \{1\}$ and $C = \{2, 3\}$ (marginal effect of \mathbf{x}_1 on \hat{f}). We construct a grid using each observed value from \mathbf{x}_1 , i.e., $x_1^{(1)} = 1$ and $x_1^{(2)} = 2$, and compute the PD function using these grid points.

ICE Plots [11]: The averaging in Eq. (2) of the PD function can obfuscate more complex relationships resulting from feature interactions, i.e. when the partial relationship of one or more observations depends on other features. ICE plots address this problem by visualizing to what extent the prediction of a single observation changes when the value of the considered feature changes. Instead of plotting the pairs $\{(\mathbf{x}_S^{*(k)}, \hat{f}_S(\mathbf{x}_S^{*(k)}))\}_{k=1}^m$, ICE plots visualize the pairs $\{(\mathbf{x}_S^{*(k)}, \hat{f}_S^{(i)}(\mathbf{x}_S^{*(k)}))\}_{k=1}^m$ for each observation indexed by $i \in \{1, \dots, n\}$.

Shapley Value: A coalitional game is defined by a set of players P , which can form coalitions $S \subseteq P$. Each coalition S achieves a certain payout. The characteristic function $v : 2^P \rightarrow \mathbb{R}$ maps all 2^p possible coalitions to their payouts. The Shapley value [19] now fairly assigns a value to each player depending on their contribution in all possible coalitions. This concept was transferred to feature effect estimation in [21]. We could explain the prediction of a single, fixed observation \mathbf{x} by regarding features as players, who form various coalitions (subsets) S to achieve the prediction $\hat{f}(\mathbf{x})$. For each coalition S , we are only allowed to access values of features from S . A natural definition of the payout is the PD value $f_S(\mathbf{x}_S)$, which we shift so that an empty set of no features is assigned a value of 0 – which is required by the general Shapley value definition:

$$v(\mathbf{x}_S) = \mathbb{E}_{X_C}(\hat{f}(\mathbf{x}_S, X_C)) - \mathbb{E}_X(\hat{f}(X)) = f_S(\mathbf{x}_S) - f_\emptyset(\mathbf{x}_\emptyset). \quad (3)$$

The marginal contribution of feature j , joining a coalition S , is defined as

$$\Delta_j(\mathbf{x}_S) = v(\mathbf{x}_{S \cup \{j\}}) - v(\mathbf{x}_S) = f_{S \cup \{j\}}(\mathbf{x}_{S \cup \{j\}}) - f_S(\mathbf{x}_S).$$

Let Π be the set of all possible permutations over the index set P . For a permutation $\pi \in \Pi$, we denote the set of features that are in order *before* feature j as $B_j(\pi)$. For example, for $p = 4$, if we consider feature $j = 4$ and permutation $\pi = \{2, 3, 4, 1\}$, then $B_4(\pi) = \{2, 3\}$. For an observation \mathbf{x} and its feature value for feature j , the Shapley value can be estimated by

$$\begin{aligned} \hat{\phi}_j(\mathbf{x}) &= \frac{1}{p!} \sum_{\pi \in \Pi} \hat{\Delta}_j(\mathbf{x}_{B_j(\pi)}) \\ &= \frac{1}{p!} \sum_{\pi \in \Pi} \hat{f}_{B_j(\pi) \cup \{j\}}(\mathbf{x}_{B_j(\pi) \cup \{j\}}) - \hat{f}_{B_j(\pi)}(\mathbf{x}_{B_j(\pi)}) \\ &= \frac{1}{p! \cdot n} \sum_{\pi \in \Pi} \sum_{i=1}^n \hat{f}_{B_j(\pi) \cup \{j\}}^{(i)}(\mathbf{x}_{B_j(\pi) \cup \{j\}}) - \hat{f}_{B_j(\pi)}^{(i)}(\mathbf{x}_{B_j(\pi)}), \end{aligned}$$

where $\hat{f}_{B_j(\pi)}$ and $\hat{f}_{B_j(\pi) \cup \{j\}}$ are estimated by Eq. (2). An efficient approximation based on Monte-Carlo integration using m rather than $p! \cdot n$ summands was proposed by [22]. Consider the following example to illustrate the Shapley value: The features enter a room in a random order specified by the permutation π . All features in the room participate in the game, i.e., they contribute to the model prediction. The Shapley value ϕ_j is the average additional contribution of feature j by joining whatever features already entered the room before.

3 Permutation-based Feature Importance

Background. The permutation importance for random forests introduced in [3] measures the performance, e.g., the mean squared error (MSE), of each tree

within a random forest using out-of-bag samples. The performance is measured once with and once without permuted values of the feature of interest. The difference between those two performance values is computed for each tree and averaged to yield the feature importance. Permuting the values of a feature breaks the association between the feature and the target variable and results in a large drop in performance if the considered feature is important. A model-agnostic global PFI for features included in S can be defined as

$$PFI_S = \mathbb{E}(L(\hat{f}(\tilde{X}_S, X_C), Y)) - \mathbb{E}(L(\hat{f}(X), Y)) \quad (4)$$

where \tilde{X}_S refers to an independent replication of X_S , which is also independent of X_C and Y . This implies that \tilde{X}_S is a new (multivariate) random variable, which is distributed as X_S , but independent of everything else. This definition is analogous to the permutation-based model reliance introduced by [9] and relates to the definition in [12] where the authors focus on random forests. The larger the value of PFI_S , the more substantial the increase in error when we permute feature values in S , and the more important we deem the feature set S . According to [9], the use of the ratio $PFI_S = \mathbb{E}(L(\hat{f}(\tilde{X}_S, X_C), Y)) / \mathbb{E}(L(\hat{f}(X), Y))$ instead of the difference might be more comparable across different models, as it always refers to the relative drop in performance with respect to the standard generalization error. However, using the ratio can result in numerically unstable estimations if the denominator is close or equal to zero. Thus, both definitions have drawbacks that we try to address in Section 5.

Estimating and Approximating the PFI. The first term of Eq. (4) encodes the expected generalization error under perturbation of features in feature set S , which can be formulated as:

$$\begin{aligned} \mathbb{E}(L(\hat{f}(\tilde{X}_S, X_C), Y)) &= \mathbb{E}_{(X_C, Y)}(\mathbb{E}_{\tilde{X}_S | (X_C, Y)}(L(\hat{f}(\tilde{X}_S, X_C), Y))) \\ &= \mathbb{E}_{(X_C, Y)}(\mathbb{E}_{\tilde{X}_S}(L(\hat{f}(\tilde{X}_S, X_C), Y))) \\ &= \mathbb{E}_{(X_C, Y)}(\mathbb{E}_{X_S}(L(\hat{f}(X_S, X_C), Y))) \end{aligned}$$

In the derivation above, the first equality follows from the “law of total expectation”, the second from the independence of \tilde{X}_S from (X_C, Y) , and the third because \tilde{X}_S is distributed as X_S . We can plug in an estimator for the inner expected value and denote the estimate of this quantity by

$$\widehat{GE}_C(\hat{f}, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{n} \sum_{k=1}^n L(\hat{f}(\mathbf{x}_S^{(k)}, \mathbf{x}_C^{(i)}), y^{(i)}). \quad (5)$$

The index C in GE_C emphasizes that the set of features in C were not replaced with a perturbed random variable and can thus be seen as the model performance using features in C (and ignoring those in S). The above estimator is analogous to the V-statistic [18] and may also be replaced by the unbiased U-statistic using $\frac{1}{n} \sum_{i=1}^n \frac{1}{n-1} \sum_{k \neq i} L(\hat{f}(\mathbf{x}_S^{(k)}, \mathbf{x}_C^{(i)}), y^{(i)})$ as proposed by [9].² The estimator scales

² For the sake of simplicity, we consider the V-statistic throughout the article. However, all calculations and approximations based on Eq. (5) still apply – with slight modifications – when using the U-statistic.

with $O(n^2)$ (for a given set C , and assuming \hat{f} can be computed in constant time), which can be expensive when n is large. However, we can use a different formulation to motivate an approximation for Eq. (5): Let $\{\tau_1, \dots, \tau_{n!}\}$ be the set of all possible permutation vectors over the observation index set $\{1, \dots, n\}$. As shown by [9], we can replace Eq. (5) by the equivalent formulation

$$\widehat{GE}_{C,\text{perm}}(\hat{f}, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{n!} \sum_{k=1}^{n!} L(\hat{f}(\mathbf{x}_S^{(\tau_k^{(i)})}, \mathbf{x}_C^{(i)}), y^{(i)}).$$

If we approximate $\widehat{GE}_{C,\text{perm}}$ by Monte-Carlo integration using only m randomly selected permutations rather than all $n!$ permutations, we obtain

$$\widehat{GE}_{C,\text{approx}}(\hat{f}, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{k=1}^m L(\hat{f}(\mathbf{x}_S^{(\tau_k^{(i)})}, \mathbf{x}_C^{(i)}), y^{(i)}). \quad (6)$$

The approximation refers to permuting features in S repeatedly (i.e., m times) and averaging the resulting model performances.³ The PFI from Eq. (4) can be estimated using Eq. (5) for the first term and using Eq. (1) for the last term. Including the summands into an iterated sum yields the estimate

$$\widehat{PFI}_S = \frac{1}{n^2} \sum_{i=1}^n \sum_{k=1}^n \left(L(\hat{f}(\mathbf{x}_S^{(k)}, \mathbf{x}_C^{(i)}), y^{(i)}) - L(\hat{f}(\mathbf{x}^{(i)}), y^{(i)}) \right). \quad (7)$$

If we use Eq. (6) rather than Eq. (5), we obtain the approximation

$$\widehat{PFI}_{S,\text{approx}} = \frac{1}{n \cdot m} \sum_{i=1}^n \sum_{k=1}^m \left(L(\hat{f}(\mathbf{x}_S^{(\tau_k^{(i)})}, \mathbf{x}_C^{(i)}), y^{(i)}) - L(\hat{f}(\mathbf{x}^{(i)}), y^{(i)}) \right). \quad (8)$$

Eq. (8) is identical to the permutation importance of random forests formalized in [12] if we consider m as the number of trees, replace n with the number of out-of-bag samples per tree and replace the model \hat{f} with the individual trees fitted within a random forest, i.e., \hat{f}_k .

4 Visualizing Global and Local Feature Importance

Consider the summands in Eq. (7) and denote them by

$$\Delta L^{(i)}(\mathbf{x}_S) = L(\hat{f}(\mathbf{x}_S, \mathbf{x}_C^{(i)}), y^{(i)}) - L(\hat{f}(\mathbf{x}^{(i)}), y^{(i)}).$$

This quantity refers to the change in performance between the i -th observation with and without replaced feature values \mathbf{x}_S . Inspired by ICE plots, we introduce *individual conditional importance* (ICI) plots which visualize the pairs $\{(\mathbf{x}_S^{(k)}, \Delta L^{(i)}(\mathbf{x}_S^{(k)}))\}_{k=1}^n$ for all observations $i = 1, \dots, n$. We define the local feature importance of the i -th observation (regarding features in S) as the integral of

³ By the same logic, we could also directly approximate Eq. (5) by summing over m randomly selected feature values for features in S instead of using all of them. We here opted for Eq. (6), due to the in our opinion interesting relation to the random forest permutation importance explained at the end of this section.

the corresponding ICI curve with respect to the distribution of X_S . It is estimated by $\widehat{PFI}_S^{(i)} = \frac{1}{n} \sum_{k=1}^n \Delta L^{(i)}(\mathbf{x}_S^{(k)})$ and can be interpreted as the expected change in performance of the i -th observation after marginalizing its features in S . It also refers to the contribution of the i -th observation to the global PFI (see later in Eq. (9)). To the best of our knowledge, a similar definition for local feature importance only exists in the context of random forests, e.g., in [7].

Analogous to the PD function from Eq. (2), we introduce the *partial importance* (PI) function as the expected change in performance at a specific value \mathbf{x}_S , which can be estimated by $\widehat{PI}_S(\mathbf{x}_S) = \frac{1}{n} \sum_{i=1}^n \Delta L^{(i)}(\mathbf{x}_S)$. Consequently, a PI plot visualizes the pairs $\{(\mathbf{x}_S^{(k)}, \widehat{PI}_S(\mathbf{x}_S^{(k)}))\}_{k=1}^n$ and refers to the pointwise average of all ICI curves across all observations at fixed grid points \mathbf{x}_S .

Fig. 2 illustrates the computation of ICI and PI curves for the first feature. It also shows the n grid points for which $\Delta L^{(i)}(\mathbf{x}_S^{(i)}) = 0 \forall i$. We can omit these points by plotting the pairs $\{(\mathbf{x}_S^{(k)}, \Delta L^{(i)}(\mathbf{x}_S^{(k)}))\}_{k \in \{1, \dots, n\} \setminus \{i\}}$ to visualize the unbiased estimation of the feature importance proposed by [9]. Visualizing the ICI curves for the approximation in Eq. (8) implies that some grid points are randomly skipped because the feature values used as grid points are implicitly determined by the randomly selected permutations in Eq. (8). The ICI curves, the PI curve, and the global PFI are related: Averaging all ICI curves pointwise yields a PI curve. Integrating the PI curve (as well as averaging the integral of all ICI curves) using Monte-Carlo integration over all points $\{\mathbf{x}_S^{(k)}\}_{k=1}^n$ yields an equivalent estimate of the global PFI from Eq. (7):

$$\widehat{PFI}_S = \frac{1}{n} \sum_{i=1}^n \widehat{PFI}_S^{(i)} = \frac{1}{n} \sum_{k=1}^n \widehat{PI}_S(\mathbf{x}_S^{(k)}). \quad (9)$$

We propose to additionally inspect the PI and ICI curves instead of focusing on a single PFI value. PI curves enable the user to identify regions in which the feature importance is higher or lower than its global PFI. ICI curves additionally enable the user to identify (suspicious) observations that impact the global PFI strongly and can reveal heterogeneity in the feature importance among the observations, which remain hidden in the PI plots (see also Section 6).

Algorithm 1 describes a procedure for obtaining PI and PD plots, which also allows to return ICI and ICE plots by visualizing $\{(\mathbf{x}_S^{*(k)}, \Delta L^{(i)}(\mathbf{x}_S^{*(k)}))\}_{k=1}^m$ and $\{(\mathbf{x}_S^{*(k)}, \hat{f}_S^{(i)}(\mathbf{x}_S^{*(k)}))\}_{k=1}^m$ for all observations i . Similar to PD and ICE plots, we can use all $k = 1, \dots, n$ or a random sample (of size $m < n$) of feature values from S as grid points for PI and ICI plots.

5 Shapley Feature Importance

In this section, we introduce the *Shapley Feature IMPortance* (SFIMP) measure, which allows to easily visualize and interpret the contribution of each feature to the model performance. Our goal is to fairly distribute the performance difference among the individual features between the scenario when all features are used and when all features are ignored, which is illustrated in Fig. 3.

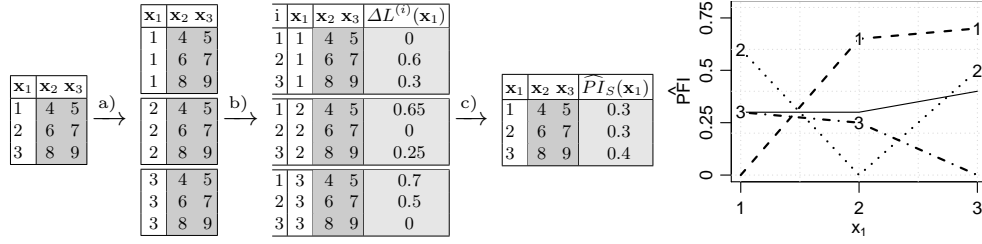


Fig. 2. The tables on the left side illustrate the required steps to create ICI curves and PI plots as described in Algorithm 1. The right plot visualizes the ICI curves of individual observations for $i = 1, 2, 3$ (dotted and dashed lines) and the PI curve (solid line) which is the average of ICI curves at each point of the abscissa. All points belonging to the same observation are connected by a line to produce the ICE curves.

Algorithm 1: PD plot and PI plot

1. Choose m grid points $\mathbf{x}_S^{*(1)}, \dots, \mathbf{x}_S^{*(m)}$.
 2. Repeat the following steps for the k -th grid point:
 - a) Modify the data by replacing all observed values in \mathbf{x}_S with the constant values from the k -th grid point $\mathbf{x}_S^{*(k)}$.
 - b) Use the modified data from a), the prediction function \hat{f} and the loss function L and calculate for all individual observations:
 - i) $\hat{f}_S^{(i)}(\mathbf{x}_S^{*(k)}) = \hat{f}(\mathbf{x}_S^{*(k)}, \mathbf{x}_C^{(i)})$
 - ii) $\Delta L^{(i)}(\mathbf{x}_S^{*(k)}) = L(\hat{f}(\mathbf{x}_S^{*(k)}, \mathbf{x}_C^{(i)}), y^{(i)}) - L(\hat{f}(\mathbf{x}^{(i)}), y^{(i)})$
 - c) Aggregate the individual values:
 - i) $\hat{f}_S(\mathbf{x}_S^{*(k)}) = \frac{1}{n} \sum_{i=1}^n \hat{f}_S^{(i)}(\mathbf{x}_S^{*(k)})$
 - ii) $\widehat{PI}_S(\mathbf{x}_S^{*(k)}) = \frac{1}{n} \sum_{i=1}^n \Delta L^{(i)}(\mathbf{x}_S^{*(k)})$
 3. Plot the pairs $\{(\mathbf{x}_S^{*(k)}, \hat{f}_S(\mathbf{x}_S^{*(k)}))\}_{k=1}^m$ and $\{(\mathbf{x}_S^{*(k)}, \widehat{PI}_S(\mathbf{x}_S^{*(k)}))\}_{k=1}^m$.
-

The Shapley value was used in [6] for a fair attribution of the difference in model performance. However, the authors focused on feature selection which requires refitting the model by leaving out or including features. This can lead to different results of the learning algorithm since different relationships can be learned due to the absence of features. This is reasonable in the context of feature selection. However, as we measure the feature importance of an already fitted model, we prefer marginalizing over features rather than omitting them completely. Inspired by Eq. (3), we define the characteristic function of the coalition of features in $S \subseteq P$ based on Eq. (5) as:

$$v_{GE}(S) = \widehat{GE}_S(\hat{f}, \mathcal{D}) - \widehat{GE}_\emptyset(\hat{f}, \mathcal{D}). \quad (10)$$

The characteristic function measures the change in performance between using features in S (i.e., ignoring features in its complement C by marginalizing over them) and ignoring all features. This is similar to Eq. (7) which, in contrast, measures the change in performance between ignoring features in S and using all features. Since the error $\widehat{GE}_\emptyset(\hat{f}, \mathcal{D})$ (no features are considered, i.e., all

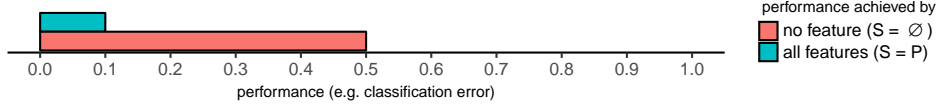


Fig. 3. Illustration of the difference in model performance that we want to fairly distribute among the features. The model performance (e.g., classification error) is 0.1 when using all features (green bar) and 0.5 when ignoring all features (red bar). Our goal is to fairly distribute the resulting performance difference of 0.4 among all involved features based on their marginal contribution.

features are marginalized out) is usually greater than $\widehat{GE}_S(\hat{f}, \mathcal{D})$, $v_{GE}(S)$ will have negative values.⁴ The marginal contribution of a feature j to a coalition of features in S is given by

$$\Delta_j(S) = v_{GE}(S \cup \{j\}) - v_{GE}(S) = \widehat{GE}_{S \cup \{j\}}(\hat{f}, \mathcal{D}) - \widehat{GE}_S(\hat{f}, \mathcal{D}).$$

If we consider a permuted order $\pi \in \Pi$ of the features, where $B_j(\pi)$ is the set of features occurring before feature j , we obtain the Shapley value estimation

$$\begin{aligned} \hat{\phi}_j(v_{GE}) &= \frac{1}{p!} \sum_{\pi \in \Pi} \Delta_j(B_j(\pi)) \\ &= \frac{1}{p!} \sum_{\pi \in \Pi} \widehat{GE}_{B_j(\pi) \cup \{j\}}(\hat{f}, \mathcal{D}) - \widehat{GE}_{B_j(\pi)}(\hat{f}, \mathcal{D}), \end{aligned} \quad (11)$$

which refers to the SFIMP measure of feature j . Computing Eq. (11) is computationally expensive when the number of features p is large, even if we use the approximation of the model performance from Eq. (6). We therefore suggest an efficient procedure in Algorithm 2. The Shapley value satisfies the following four desirable properties as already worked out in [6]:

1. Efficiency: $\sum_{j=1}^p \phi_j = v_{GE}(P)$. All SFIMP values add up to $v_{GE}(P)$, i.e., the difference in performance between the scenario when all features are used and when all features are ignored. This allows us to calculate the proportion of explained importance for each feature j using $\frac{\phi_j}{\sum_{j=1}^p \phi_j}$.
2. Symmetry: If $v_{GE}(S \cup \{j\}) = v_{GE}(S \cup \{k\})$ for all $S \subseteq \{1, \dots, p\} \setminus \{j, k\}$, then $\phi_j = \phi_k$. Two features j and k have the same SFIMP values if their marginal contribution to all possible coalitions is the same.
3. Dummy property: If $v_{GE}(S \cup \{j\}) = v_{GE}(S)$ for all $S \subseteq P$, then $\phi_j = 0$. The SFIMP value of a feature j is zero if its marginal contribution does not change no matter to which coalition S the feature is added.
4. Additivity: $\phi_j(v_{GE} + w_{GE}) = \phi_j(v_{GE}) + \phi_j(w_{GE})$. The SFIMP value resulting from a single game with two combined performance measures $\phi_j(v_{GE} + w_{GE})$ is the same as the sum of the two SFIMP values resulting from two separate games with corresponding characteristic functions, i.e., $\phi_j(v_{GE}) + \phi_j(w_{GE})$. Linearity: $\phi_j(c \cdot v_{GE}) = c \cdot \phi_j(v_{GE})$. Any multiplication of the performance measure with a constant c does not affect the feature ranking.

⁴ We prefer the definition in Eq. (10) as it directly shows the relation to Eq. (3), however, we could also swap the sign as discussed at the end of this section.

Algorithm 2: Approximation of SFIMP values: Contribution of j -th feature towards the model performance.

Input: $m_{\text{feat}}, m_{\text{obs}}, \hat{f}, L, \mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$

- 1 **forall** $k \in \{1, \dots, m_{\text{feat}}\}$ **do**
- 2 choose a random permutation of the feature indices $\pi \in \Pi$.
- 3 set $S = B_j(\pi)$ containing features that won't be permuted.
- 4 set $\widehat{GE}_{S, \text{perm}} = 0$ and $\widehat{GE}_{S \cup \{j\}, \text{perm}} = 0$.
- 5 **forall** $l \in \{1, \dots, m_{\text{obs}}\}$ **do**
- 6 choose a random permutation of observation indices $\tau \in \{\tau_1, \dots, \tau_n\}$.
- 7 measure performance by permuting features w.r.t. $\tau = (\tau^{(1)}, \dots, \tau^{(n)})$:

$$\begin{aligned} \widehat{GE}_{S, \text{perm}} &= \widehat{GE}_{S, \text{perm}} + \frac{1}{n} \sum_{i=1}^n L(\hat{f}(\mathbf{x}_S^{(i)}, \mathbf{x}_C^{(\tau^{(i)})}), y^{(i)}) \\ \widehat{GE}_{S \cup \{j\}, \text{perm}} &= \widehat{GE}_{S \cup \{j\}, \text{perm}} + \frac{1}{n} \sum_{i=1}^n L(\hat{f}(\mathbf{x}_{S \cup \{j\}}^{(i)}, \mathbf{x}_{C \setminus \{j\}}^{(\tau^{(i)})}), y^{(i)}) \end{aligned}$$
- 8 compute marginal contribution for feature j in iteration k :

$$\Delta_j^{(k)}(S) = \frac{1}{m_{\text{obs}}} \cdot (\widehat{GE}_{S \cup \{j\}, \text{perm}} - \widehat{GE}_{S, \text{perm}})$$
- 9 **return** $\hat{\phi}_j = \frac{1}{m_{\text{feat}}} \sum_{k=1}^{m_{\text{feat}}} \Delta_j^{(k)}(S)$

The properties above imply that fairly distributing the drop in performance using $v_{PFI}(S) = \widehat{PFI}_S = \widehat{GE}_C(\hat{f}, \mathcal{D}) - \widehat{GE}_P(\hat{f}, \mathcal{D})$ results in the same Shapley values (except for the sign) and is equivalent to using $-v_{GE}(P)$. The SFIMP measure can thus be seen as an extension of the PFI measure in the sense that it additionally fairly distributes the importance values among features. The PFI measure ignores features in S by permuting or marginalizing over them, which destroys any correlation and interaction of features in C with features in S . Consequently, the PFI of a feature also includes the importance of any interaction with that feature and features in C and therefore an interaction will be fully attributed to all involved features. The SFIMP measure solves this issue as it considers the marginal contribution of a feature and equally distributes the importance of interactions among the interacting features. This allows comparing feature importances across different models.

6 Simulations and Application

For full reproducibility, all our proposed methods are available in the R package `featureImportance`⁵. The repository also contains the R code, which is partly based on `batchtools` [13], for the application and simulation in this section.

6.1 Simulations

PI and ICI Plots. Consider the following data-generating model:

$$Y = X_1 + X_2 + 10X_1 \cdot \mathbb{1}_{X_3=0} + 10X_2 \cdot \mathbb{1}_{X_3=1} + \epsilon,$$

⁵ <https://github.com/giusepppec/featureImportance>.

$$X_1, X_2 \stackrel{i.i.d}{\sim} \mathcal{N}(0, 1), X_3 \sim \mathcal{B}(1, 0.5), \epsilon \sim \mathcal{N}(0, 0.5).$$

We simulate a training data set with 10000 observations, train a random forest and compute the global PFI on 100 test sets of size $n = 100$ sampled from the same distribution. We demonstrate that, by merely inspecting the global PFI, the features X_1 and X_2 would be considered equally important. However, due to the interactions, it is clear that feature X_1 should be considered more important than X_2 when $X_3 = 0$ and vice-versa when $X_3 = 1$.

According to Eq. (9), averaging the local feature importances (i.e., the integral of all ICI curves) results in the global PFI. Having at hand the local feature importance of each observation allows calculating the PFI conditional on other features. This does not require additional time-consuming calculations, as we only have to average the already computed local feature importances according to the condition considered in the conditional PFI. The relevance of conditional feature importance in the case of random forests with correlated features was discussed in [20]. In Fig. 4, we illustrate the usefulness of a model-agnostic conditional PFI in case of interactions by showing the PI curves of X_1 and X_2 conditional on the binary feature X_3 . The integral of these conditional PI curves refers to the PFI conditional on X_3 . Its value differs depending on the two groups introduced by feature X_3 , which suggests that there is an interaction between the features X_1 and X_3 as well as X_2 and X_3 .

Table 1 shows that feature X_1 and X_2 are almost equally important if we consider the unconditional global PFI. However, a different ranking of features is obtained when we compute the PFI conditional on X_3 . Thus, inspecting PI and ICI curves conditional on other feature values may help in detecting interactions.

Table 1. The mean and the standard deviation (numbers in brackets) of the PFI values estimated using the 100 simulated test data sets.

	X_1	X_2
global PFI	77.976 (14.15)	76.764 (13.89)
PFI for $X_3 = 0$	152.49 (26.06)	1.428 (1.32)
PFI for $X_3 = 1$	1.261 (1.03)	151.489 (24.69)

Shapley Feature Importance. We illustrate how the SFIMP measure can be used to compare the feature importance across different models and present the results of a small simulation study to compare the SFIMP measure introduced in Section 5 with the difference-based and the ratio-based PFI discussed in Section 3. Consider the following data-generating linear model with a simple interaction:

$$Y = X_1 + X_2 + X_3 + X_1 \cdot X_2 + \epsilon, \quad X_1, X_2, X_3 \stackrel{i.i.d}{\sim} \mathcal{N}(0, 1), \epsilon \sim \mathcal{N}(0, 0.5).$$

All three features and the interaction of X_1 and X_2 have the same linear effect on the target Y . We simulate training data with 10000 observations and train four

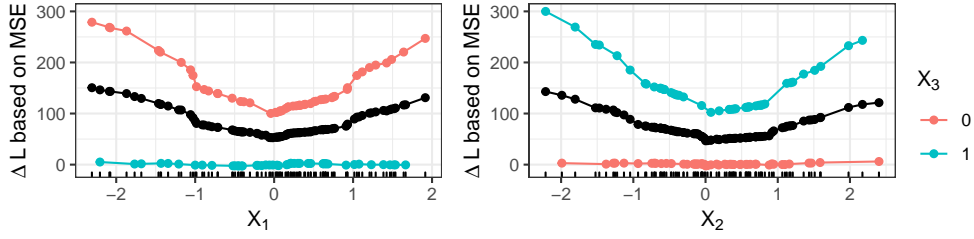


Fig. 4. PI curves of X_1 and X_2 calculated using all observations (black line) and conditional on $X_3 = 0$ (red line) and $X_3 = 1$ (green line). The points plotted on the lines refer to the observed feature values that were used as grid points to produce the corresponding PI curves as described in Algorithm 1.

learning algorithms using the `mlr` R package [2] in their defaults: An SVM with Gaussian kernel (`ksvm`), a random forest (`randomForest`), a simple linear model (`lm`) and another one that considers 2-way interaction effects (`rsm`). We use a test set with $n = 100$ observations sampled from the same distribution and compute the SFIMP values according to Eq. (11). Panel (a) of Fig. 5 displays how the SFIMP measure distributes the total explainable performance among all features and shows the proportion of explained importance for each feature. We repeat the experiment 500 times on different test sets of equal size and additionally compute the difference-based and ratio-based PFI. The results are shown in panel (b) of Fig. 5. For the linear model without interaction effects, the calculated importance of all three features is equal (median ratio of 1). For all other models, we obtained a higher importance for the interacting features, indicating that these models were able to grasp the interaction effect. However, as permuting a feature destroys any interaction with that feature, the PFI values of a feature will also include the importance of any interaction with that feature. Thus, the importance of the interaction between X_1 and X_2 is contained in the PFI value for feature X_1 as well as in the PFI value for feature X_2 . This will overestimate the importance of X_1 and X_2 with respect to X_3 since X_1 and X_2 share the same interaction. In panel (b), we thus show the ratio of the importance values with respect to X_3 . The results suggest that the difference-based PFI considers X_1 and X_2 twice as important as X_3 as the median ratio is around 2. In contrast, the median ratio of SFIMP is around 1.5 as the importance of the interaction is fairly distributed among X_1 and X_2 .

6.2 Application on Real Data

We demonstrate our graphical tools on the Boston housing data, which is publicly available on OpenML [23] with data set ID 531. The data set contains 13 features that may affect the median home price of 506 metropolitan areas of Boston. We used the `OpenML` R package [5] and created the OpenML task with ID 167147 containing a holdout split ($\frac{2}{3}$ vs. $\frac{1}{3}$) for training a random forest and producing the PI and ICI plots on the test set.

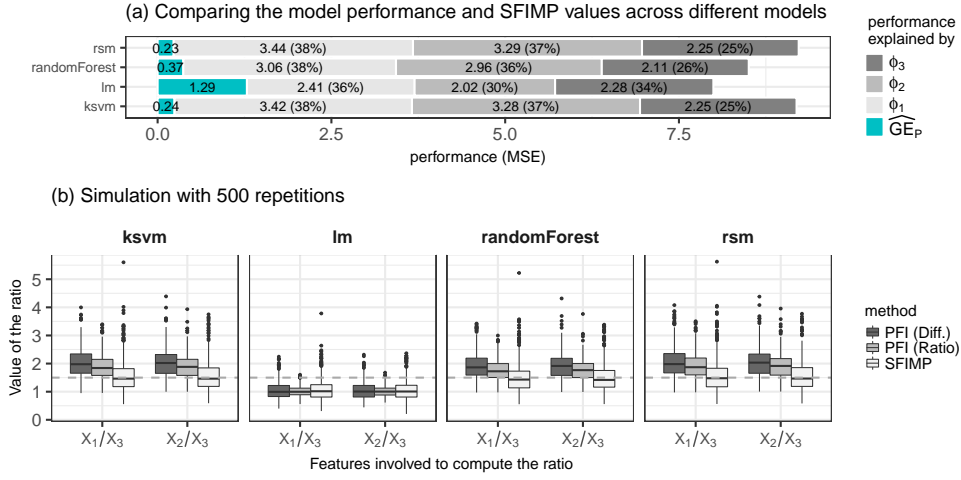


Fig. 5. Panel (a) shows the results of a single run, consisting of sampling test data and computing the importance on the previously fitted models. The first numbers on the left refer to the model performance (MSE) using all features. The other numbers are the SFIMP values which sum up to the total explainable performance $v_{GE}(P)$ from Eq. (10). The percentages refer to the proportion of explained importance. Panel (b) shows the results of 500 repetitions of the experiment. The plots display the distribution of ratios of the importance values for X_1 and X_2 with respect to X_3 computed by SFIMP, by the difference-based PFI, and by the ratio-based PFI.

Row (1) of Table 2 shows the global PFI values of all features. They are estimated using Eq. (7) by taking into account all $166 \cdot 166$ points of the test data. Fig. 6 shows the corresponding PI and ICI curves for the two most important features (LSTAT and RM). They visualize which regions of each feature and which observations have a high impact on the computed PFI values on a global and local level, which follows from the relation in Eq. (9).

PI plots visualize the expected change in performance at each position of the abscissa. An expected change close to zero across the whole range of the feature values suggests an unimportant feature. The PI plot of LSTAT in Fig. 6 suggests that the feature is more important if $LSTAT < 10$. For illustration purposes, we omit all observations for which $LSTAT \geq 10$ and recompute the conditional PFI values, which are displayed in Row (2) of Table 2. The resulting conditional PFI values are smaller, i.e., excluding observations for which $LSTAT \geq 10$ makes the LSTAT feature less important. Note that omitting observations change the empirical distribution of the features and thus also affects the importance of other features when the PFI values are recomputed.

ICI curves additionally reveal the most (and the least) influential observations for the feature importance by considering their integral (see highlighted lines in Fig. 6). We can, for example, omit observations with a negative ICI curve integral. In our test set, we observe 18 of 166 ICI curves with a negative integral

for the LSTAT feature. These observations have a negative impact on the global PFI according to the relation in Eq. (9). We omit them and recompute the PFI values. The results are listed in row (3) of Table 2 and show an increased PFI value for LSTAT.

Table 2. PFI values calculated for a random forest trained on the Boston housing training set and using the MSE on the test data. The PFI values in row (1) are based on all observations from the test set, in row (2) on a subset where $LSTAT < 10$ and in row (3) after removing observations having a negative ICI integral.

	LSTAT	RM	NOX	DIS	CRIM	PTRATIO	AGE	INDUS	TAX	RAD	B	ZN	CHAS
(1)	32.0	15.6	3.9	2.7	2.6	2.2	1.2	1.0	1.0	0.8	0.8	0.1	0.1
(2)	10.4	29.6	1.5	3.3	0.8	2.3	0.8	0.5	1.2	1.1	0.6	0.2	0.2
(3)	35.3	17.0	4.3	2.4	2.5	2.5	1.1	1.2	0.8	0.9	0.8	0.1	0.1

7 Conclusion and Future Work

It is essential for practitioners to peek inside black box models to get a better understanding of how features contribute to model predictions or how they affect the model performance. Model-agnostic visualization methods can simplify this task tremendously. Regarding the feature importance, the PI and ICI curves are a convenient choice for visualizing how features affect model performance. We demonstrated how to disaggregate the global PFI into its individual local PFI components, which enabled us to visualize the feature importance on a local and global level. It also allows practitioners to analyze and compare the feature importance across different groups of observations in the data, e.g., by subsetting the data according to other feature values and computing a conditional feature importance similar to [20] on the subsetted data which may reveal interactions. Another interesting aspect, which we leave for future work, is aggregating the local feature importances of individual observations (i.e., the integral of ICI curves) across different features to obtain a measure for the importance of individual observations. This could be used to find clusters of observations in the data that were important for the model performance similar to [15], but based on feature importance rather than feature effects. Furthermore, it is also possible to disaggregate the Shapley feature importance introduced in Section 5 and produce plots similar to Shapley dependency plots that were recently introduced in [15], but we leave this for future work. Our proposed methods serve as an evaluation tool that is applied to a data set *after* a model has been fitted. As a consequence, our methods can be used to either assess the feature importance based on the “in-sample performance” or based on the “out-of-sample performance” of a fitted model. In the former case, the same data could be used to fit the model and to calculate the quantities involved in the definition of our methods. We focused on the latter case with independent test data. However, we could also investigate

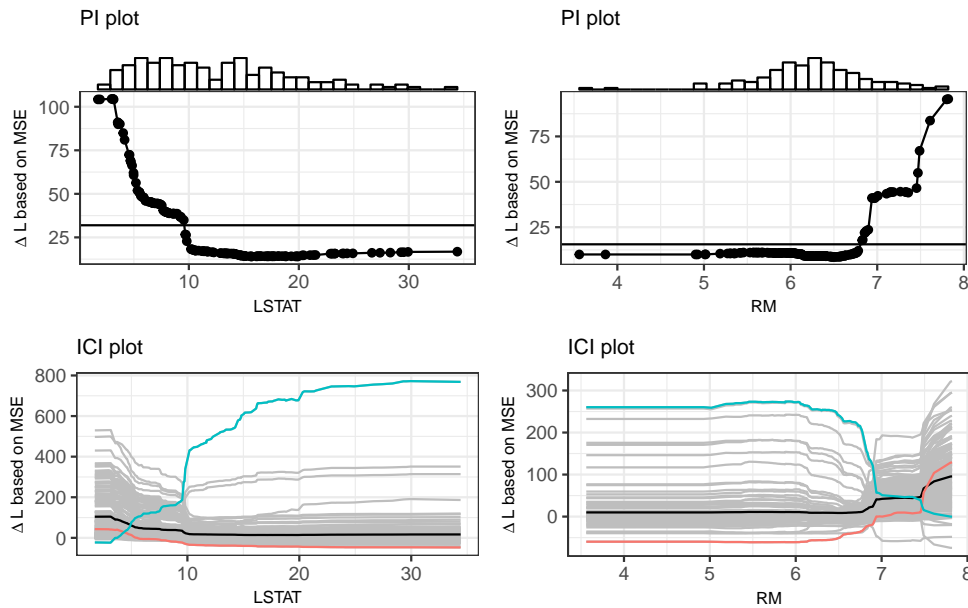


Fig. 6. PI and ICI plots for a random forest and the two most important features of the Boston housing data (LSTAT and RM). The horizontal lines in the PI plots represent the value of the global PFI (i.e., the integral of the PI curve). Marginal distribution histograms for features are added to the PI margins. The ICI curve with the largest integral is highlighted in green and the curve with the smallest integral in red.

the variability introduced by the estimation of the model itself via resampling and plot or aggregate the resulting set of quantities.

Acknowledgments

This work is funded by the Bavarian State Ministry of Education, Science and the Arts in the framework of the Centre Digitisation.Bavaria (ZD.B).

References

- [1] Bischl, B., Mersmann, O., Trautmann, H., Weihs, C.: Resampling methods for meta-model validation with recommendations for evolutionary computation. *Evol. Comput.* **20**(2), 249–275 (2012)
- [2] Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Studerus, E., Casalicchio, G., Jones, Z.M.: mlr: Machine learning in R. *J. Mach. Learn. Res.* **17**(170), 1–5 (2016)
- [3] Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
- [4] Casalicchio, G., Bischl, B., Boulesteix, A.L., Schmid, M.: The residual-based predictiveness curve: A visual tool to assess the performance of prediction models. *Biometrics* **72**(2), 392–401 (2016)

-
- [5] Casalicchio, G., Bossek, J., Lang, M., Kirchhoff, D., Kerschke, P., Hofner, B., Seibold, H., Vanschoren, J., Bischl, B.: OpenML: An R package to connect to the machine learning platform OpenML. *Comput. Stat.* (2017). <https://doi.org/10.1007/s00180-017-0742-2>
 - [6] Cohen, S., Dror, G., Ruppin, E.: Feature selection via coalitional game theory. *Neural Comput.* **19**(7), 1939–1961 (2007)
 - [7] Cutler, A., Cutler, D.R., Stevens, J.R.: Random forests. In: *Ensemble Machine Learning*, pp. 157–175. Springer (2012)
 - [8] Datta, A., Sen, S., Zick, Y.: Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. *Proceedings - 2016 IEEE Symposium on Security and Privacy, SP 2016* pp. 598–617 (2016)
 - [9] Fisher, A., Rudin, C., Dominici, F.: Model class reliance: Variable importance measures for any machine learning model class, from the "Rashomon" perspective. *arXiv preprint arXiv:1801.01489* (2018)
 - [10] Friedman, J.H.: Greedy function approximation: A gradient boosting machine. *Annals of statistics* pp. 1189–1232 (2001)
 - [11] Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *J. Comput. Graph. Stat.* **24**(1), 44–65 (2015)
 - [12] Gregorutti, B., Michel, B., Saint-Pierre, P.: Correlation and variable importance in random forests. *Stat. Comput.* **27**(3), 659–678 (2017)
 - [13] Lang, M., Bischl, B., Surmann, D.: batchtools: Tools for R to work on batch systems. *The Journal of Open Source Software* **2**(10) (2017)
 - [14] Lipton, Z.C.: The mythos of model interpretability. *ICML WHI '16* (2016)
 - [15] Lundberg, S.M., Erion, G.G., Lee, S.I.: Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888* (2018)
 - [16] Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: *NIPS*, vol. 30, pp. 4765–4774. Curran Associates, Inc. (2017)
 - [17] Molnar, C., Casalicchio, G., Bischl, B.: iml: An R package for interpretable machine learning. *The Journal of Open Source Software* **3**(786) (2018)
 - [18] Serfling, R.J.: *Approximation Theorems of Mathematical Statistics*, vol. 162. John Wiley & Sons (2009)
 - [19] Shapley, L.S.: A value for n-person games. *Contributions to the Theory of Games* **2**(28), 307–317 (1953)
 - [20] Strobl, C., Boulesteix, A.L., Kneib, T., Augustin, T., Zeileis, A.: Conditional variable importance for random forests. *BMC Bioinf.* **9**, 307 (2008)
 - [21] Štrumbelj, E., Kononenko, I., Wrobel, S.: An efficient explanation of individual classifications using game theory. *J. Mach. Learn. Res.* **11**(Jan), 1–18 (2010)
 - [22] Štrumbelj, E., Kononenko, I.: A general method for visualizing and explaining black-box regression models. In: *Int. Conf. on Adaptive and Natural Computing Algorithms*. pp. 21–30. Springer (2011)
 - [23] Vanschoren, J., Van Rijn, J.N., Bischl, B., Torgo, L.: OpenML: Networked science in machine learning. *ACM SIGKDD Explor. Newsl.* **15**(2), 49–60 (2014)

12. Quantifying Model Complexity via Functional Decomposition for Better Post-Hoc Interpretability

Contributing article:

Molnar, C., Casalicchio, G., and Bischl, B. (2019). Quantifying Model Complexity via Functional Decomposition for Better Post-Hoc Interpretability. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 193-204).

Copyright information:

©Springer Nature Switzerland AG 2020

Author contributions:

The paper was drafted and written by Christoph Molnar. He also implemented the measures, the simulations and the application example. Bernd Bischl proposed to use a multi-objective optimization example in the application. All authors added input, suggested modifications, proofread and revised the paper.

Quantifying Model Complexity via Functional Decomposition for Better Post-Hoc Interpretability

Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl

Department of Statistics, LMU Munich,
Ludwigstr. 33, 80539 Munich, Germany
`christoph.molnar@stat.uni-muenchen.de`

Abstract. Post-hoc model-agnostic interpretation methods such as partial dependence plots can be employed to interpret complex machine learning models. While these interpretation methods can be applied regardless of model complexity, they can produce misleading and verbose results if the model is too complex, especially w.r.t. feature interactions. To quantify the complexity of arbitrary machine learning models, we propose model-agnostic complexity measures based on functional decomposition: number of features used, interaction strength and main effect complexity. We show that post-hoc interpretation of models that minimize the three measures is more reliable and compact. Furthermore, we demonstrate the application of these measures in a multi-objective optimization approach which simultaneously minimizes loss and complexity.

Keywords: Model Complexity · Interpretable Machine Learning · Explainable AI · Accumulated Local Effects · Multi-Objective Optimization

1 Introduction

Machine learning models are optimized for predictive performance, but it is often required to understand models, e.g., to debug them, gain trust in the predictions, or satisfy regulatory requirements. Many post-hoc interpretation methods either quantify effects of features on predictions, compute feature importances, or explain individual predictions, see [17, 24] for more comprehensive overviews. While model-agnostic post-hoc interpretation methods can be applied regardless of model complexity [30], their reliability and compactness deteriorates when models use a high number of features, have strong feature interactions and complex feature main effects. Therefore, model complexity and interpretability are deeply intertwined and reducing complexity can help to make model interpretation more reliable and compact. Model-agnostic complexity measures are needed to strike a balance between interpretability and predictive performance [4, 31].

Contributions. We propose and implement three model-agnostic measures of machine learning model complexity which are related to post-hoc interpretability. To our best knowledge, these are the first model-agnostic measures that describe the global interaction strength, complexity of main effects and number

of features. We apply the measures to different datasets and machine learning models. We argue that minimizing these three measures improves the reliability and compactness of post-hoc interpretation methods. Finally, we illustrate the use of our proposed measures in multi-objective optimization.

2 Related Work and Background

In this section, we introduce the notation, review related work, and describe the functional decomposition on which we base the proposed complexity measures.

Notation: We consider machine learning prediction functions $f : \mathbb{R}^p \mapsto \mathbb{R}$, where $f(x)$ is a prediction (e.g., regression output or a classification score). For the decomposition of f , we write $f_S : \mathbb{R}^{|S|} \mapsto \mathbb{R}$, $S \subseteq \{1, \dots, p\}$, to denote a function that maps a vector $x_S \in \mathbb{R}^{|S|}$ with a subset of features to a marginal prediction. If subset S contains a single feature j , we write f_j . We refer to the training data of the machine learning model with the tuples $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$ and refer to the value of the j -th feature from the i -th instance as $x_j^{(i)}$. We write X_j to refer to the j -th feature as a random variable.

Complexity and Interpretability Measures: In the literature, model complexity and (lack of) model interpretability are often equated. Many complexity measures are model-specific, i.e., only models of the same class can be compared (e.g., decision trees). Model size is often used as a measure for interpretability (e.g., number of decision rules, tree depth, number of non-zero coefficients) [3, 16, 20, 22, 31–34]. Akaike's Information Criterion (AIC) and the Bayesian Information Criterion (BIC) are more widely applicable measures for the trade-off between goodness of fit and degrees of freedom. In [26], the authors propose model-agnostic measures of model stability. In [27], the authors propose explanation fidelity and stability of local explanation models. Further approaches measure interpretability based on experimental studies with humans, e.g., whether humans can predict the outcome of the model [8, 13, 20, 28, 35].

Functional Decomposition: Any high-dimensional prediction function can be decomposed into a sum of components with increasing dimensionality:

$$f(x) = \underbrace{f_0}_{\text{Intercept}} + \underbrace{\sum_{j=1}^p f_j(x_j)}_{\text{1st order effects}} + \underbrace{\sum_{j < k}^p f_{jk}(x_j, x_k)}_{\text{2nd order effects}} + \dots + \underbrace{f_{1, \dots, p}(x_1, \dots, x_p)}_{\text{p-th order effect}} \quad (1)$$

This decomposition is only unique with additional constraints regarding the components. Accumulated Local Effects (ALE) were proposed in [1] as a tool for visualizing feature effects (e.g., Figure 1) and as unique decomposition of the prediction function with components $f_S = f_{S, ALE}$. The ALE decomposition is unique under an orthogonality-like property described in [1].

The ALE main effect $f_{j, ALE}$ of a feature $x_j, j \in \{1, \dots, p\}$ for a prediction function f is defined as

$$f_{j, ALE}(x_j) = \int_{z_{0,j}}^{x_j} \mathbb{E} \left[\frac{\partial f(X_1, \dots, X_p)}{\partial X_j} \middle| X_j = z_j \right] dz_j - c_j \quad (2)$$

Here, $z_{0,j}$ is a lower bound of X_j (usually the minimum of x_j) and the expectation \mathbb{E} is computed conditional on the value for x_j and over the marginal distribution of all other features. The constant c_j is chosen so that the mean of $f_{j,ALE}(x_j)$ with respect to the marginal distribution of X_j is zero, so that the ALE components sum to the full prediction function. By integrating the expected derivative of f with respect to X_j the effect of x_j on the prediction function f is isolated from the effects of all other features. ALE main effects are estimated with finite differences, i.e., access to the gradient of a prediction function is not required (see [1]). We base our proposed measures on the ALE decomposition, because ALE are computationally cheap (worst case $O(n)$ per main effect), they can be computed sequentially instead of simultaneously, they do not require knowledge of the joint distribution, and several software implementations exist [2, 25].

3 Functional Complexity

In this section, we motivate complexity measures based on functional decomposition. Based on Equation 1, we decompose the prediction function into a constant (estimated as $f_0 = \frac{1}{n} \sum_{i=1}^n f(x^{(i)})$), main effects (estimated by ALE), and a remainder term containing interactions (i.e., the difference between the full model and constant + main effects).

$$f(x) = f_0 + \underbrace{\sum_{j=1}^p \overbrace{f_{j,ALE}(x_j)}^{\text{MEC: How complex?}} + \overbrace{IA(x)}^{\text{IAS: Interaction strength?}}}_{\text{NF: How many features were used?}} \quad (3)$$

This arrangement of components emphasizes a decomposition of the prediction function into a main effect model and an interaction remainder. We can analyze how well the main effect model itself approximates f by looking at the magnitude of the interaction measure IAS. The average main effect complexity (MEC) captures how many parameters are needed to describe the one-dimensional main effects on average. The number of features used (NF) describes how many features were used in the full prediction function.

3.1 Number of Features (NF)

We propose an approach based on feature permutation to determine how many features are used by a model. We regard features as "used" when changing a feature changes the prediction. If available, the model-specific number of features is preferable. The model-agnostic version is useful when the prediction function is only accessible via API or when the machine learning pipeline is complex.

The proposed procedure is formally described in Algorithm 1. To estimate whether the j -th feature was used, we sample instances from data \mathcal{D} , replace their j -th feature values with random values from the distribution of X_j (e.g., by

Algorithm 1: Number of Features Used (NF)

Input: Number of samples M , data \mathcal{D}

```

1 NF = 0
2 for  $j \in 1, \dots, p$  do
3   Draw  $M$  instances  $\{x^{(m)}\}_{m=1}^M$  from dataset  $\mathcal{D}$ 
4   Create  $\{x^{(m)*}\}_{m=1}^M$  as a copy of  $\{x^{(m)}\}_{m=1}^M$ 
5   for  $m \in 1, \dots, M$  do
6     Sample  $x_j^{(new)}$  from  $\{x_j^{(i)}\}_{i=1}^n$  with the constraint that  $x_j^{(new)} \neq x_j^{(m)}$ 
7     Set  $x_j^{(m)*} = x_j^{(new)}$ 
8   if  $f(x^{(m)*}) \neq f(x^{(m)})$  for any  $m \in \{1, \dots, M\}$  then  $NF = NF + 1$ .
9 return NF

```

sampling x_j from other instances from \mathcal{D}), and observe whether the predictions change. If the prediction of any sample changes, the feature was used.

We tested the NF heuristic with the Boston Housing data. We trained decision trees (CART) with maximum depths $\in \{1, 2, 10\}$ leading to 1, 2 and 4 features used and an L1-regularized linear model with penalty $\lambda \in \{10, 5, 2, 1, 0.1, 0.001\}$ leading to 0, 2, 3, 4, 11 and 13 features used. For each model, we estimated NF with sample sizes $M \in \{10, 50, 500\}$ and repeated each estimation 100 times. For the elastic net models, NF was always equal to the number of non-zero weights. For CART, the mean absolute differences between NF and number of features used in the trees were 0.280 ($M = 10$), 0.020 ($M = 50$) and 0.000 ($M = 500$).

3.2 Interaction Strength (IAS)

Interactions between features mean that the prediction cannot be expressed as a sum of independent feature effects, but the effect of a feature depends on values of other features [24]. We propose to measure interaction strength as the scaled approximation error between the ALE main effect model and the prediction function f . Based on the ALE decomposition, the ALE main effect model is defined as the sum of first order ALE effects:

$$f_{ALE1st}(x) = f_0 + f_{1,ALE}(x_1) + \dots + f_{p,ALE}(x_p)$$

We define interaction strength as the approximation error measured with loss L :

$$IAS = \frac{\mathbb{E}(L(f, f_{ALE1st}))}{\mathbb{E}(L(f, f_0))} \geq 0 \quad (4)$$

Here, f_0 is the mean of the predictions and can be interpreted as the functional decomposition where all feature effects are set to zero. IAS with the $L2$ loss equals 1 minus the R-squared measure, where the true targets y_i are replaced with $f(x^{(i)})$.

$$IAS = \frac{\sum_{i=1}^n (f(x^{(i)}) - f_{ALE1st}(x^{(i)}))^2}{\sum_{i=1}^n (f(x^{(i)}) - f_0)^2} = 1 - R^2$$

If $IAS = 0$, then $L(f, f_{ALE1st}) = 0$, which means that the first order ALE model perfectly approximates f and the model has no interactions.

3.3 Main Effect Complexity (MEC)

To determine the average shape complexity of ALE main effects $f_{j,ALE}$, we propose the main effect complexity (MEC) measure. For a single ALE main effect, we define MEC_j as the number of parameters needed to approximate the curve with piece-wise linear models. For the entire model, MEC is the average MEC_j over all main effects, weighted with their variance. Figure 1 shows an ALE plot (= main effect) and its approximation with two linear segments.

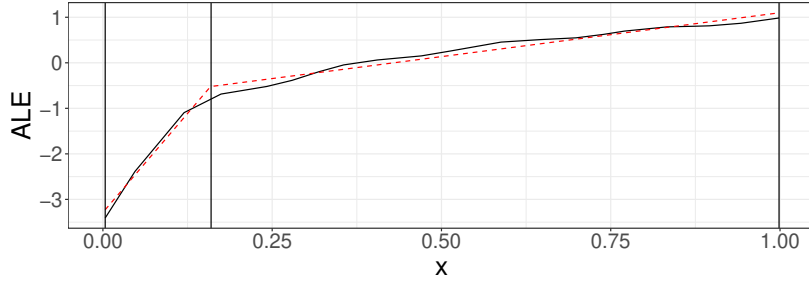


Fig. 1. ALE curve (solid line) approximated by two linear segments (dotted line).

We use piece-wise linear regression to approximate the ALE curve. Within the segments, linear models are estimated with ordinary least squares. The break-points that define the segments are found by greedy and exhaustive search along the interval boundaries of the ALE curve. Greedy here means that we first optimize the first breakpoint, then the second breakpoint with the first breakpoint fixed and so on. We measure the degrees of freedom as the number of non-zero coefficients for intercepts and slopes of the linear models. The approximation allows some error, e.g., an almost linear main effect may have $MEC_j = 1$, even if dozens of parameters would be needed to describe it perfectly. The approximation quality is measured with R-squared (R^2), i.e., the proportion of variance of $f_{j,ALE}$ that is explained by the approximation with linear segments. An approximation has to reach an $R^2 \geq 1 - \epsilon$, where ϵ is the user defined maximum approximation error. We also introduced parameter max_{seg} , the maximum number of segments. In the case that an approximation cannot reach an $R^2 \geq 1 - \epsilon$ with a given max_{seg} , MEC_j is computed with the maximum number of segments. The selected maximum approximation error ϵ should be small, but not too small. We found ϵ between 0.01 and 0.1 visually meaningful (i.e. a subjectively good approximation) and used $\epsilon = 0.05$ throughout the paper. We apply a post-processing step that greedily sets slopes of the linear segments to zero, as long as $R^2 \in \{1 - \epsilon, 1\}$. The post-processing potentially decreases the MEC_j ,

especially for models with constant segments like decision trees. MEC_j is averaged over all features to obtain the global main effect complexity. Each MEC_j is weighted with the variance of the corresponding ALE main effect to give more weight to features that contribute more to the prediction. Algorithm 2 describes the MEC computation in detail.

Algorithm 2: Main Effect Complexity (MEC).

Input: Model f , approximation error ϵ , max. segments max_{seg} , data \mathcal{D}

- 1 Define $R^2(g_j, f_{j,ALE}) := \sum_{i=1}^n (g_j(x_j^{(i)}) - f_{j,ALE}(x_j^{(i)}))^2 / \sum_{i=1}^n (f_{j,ALE}(x_j^{(i)}))^2$
- 2 **for** $j \in \{1, \dots, p\}$ **do**
- 3 Estimate $f_{j,ALE}$
- 4 // Approximate ALE with linear model
- 5 Fit $g_j(x_j) = \beta_0 + \beta_1 x_j$ predicting $f_{j,ALE}(x_j^{(i)})$ from $x_j^{(i)}$, $i \in 1, \dots, n$
- 6 Set $K = 1$
- 7 // Increase nr. of segments until approximation is good enough
- 8 **while** $K < max_{seg}$ AND $R^2(g_j, f_{j,ALE}) < (1 - \epsilon)$ **do**
- 9 // Find intervals Z_k through exhaustive search along ALE curve breakpoints
- 10 // For categorical feature, set slopes $\beta_{1,k}$ to zero
- 11 $g_j(x_j) = \sum_{k=1}^{K+1} \mathbb{I}_{x_j \in Z_k} \cdot (\beta_{0,k} + \beta_{1,k} x_j)$
- 12 Set $K = K + 1$
- 13 Greedy set slopes to zero while $R^2 > 1 - \epsilon$
- 14 // Sum of non-zero coefficients minus first intercept
- 15 $MEC_j = K + \sum_{k=1}^K \mathbb{I}_{\beta_{1,k} > 0} - 1$
- 16 $V_j = \frac{1}{n} \sum_{i=1}^n (f_{j,ALE}(x_j^{(i)}))^2$
- 17 **return** $MEC = \frac{1}{\sum_{j=1}^p V_j} \sum_{j=1}^p V_j \cdot MEC_j$

4 Application of Complexity Measures

In the following experiment, we train various machine learning models on different prediction tasks and compute the model complexities. The goal is to analyze how the complexity measures behave across different datasets and models. The dataset are: Bike Rentals [10] (n=731; 3 numerical, 6 categorical features), Boston Housing (n=506; 12 numerical, 1 categorical features), (down-sampled) Superconductivity [18] (n=2000; 81 numerical, 0 categorical features) and Abalone [9] (n=4177; 7 numerical, 1 categorical features).

Table 1 shows performance and complexity of the models. As desired, the main effect complexity for linear models is 1 (except when categorical features with 2+ categories are present as in the bike data), and higher for more flexible methods like random forests. The interaction strength (IAS) is zero for additive models (boosted GAM, (regularized) linear models). Across datasets we observe

learner	bike				Boston Housing				superconductivity				abalone			
	MSE	MEC	IAS	NF	MSE	MEC	IAS	NF	MSE	MEC	IAS	NF	MSE	MEC	IAS	NF
cart	923035	1.1	0.07	6	23.7	1.9	0.12	4	325.0	1.0	0.23	8	6.0	2.8	0.09	3
cart2	1245105	1.0	0.01	2	29.8	1.7	0.02	2	417.6	1.0	0.22	3	6.7	3.0	0.02	1
cvglmnet	667291	1.1	0.00	9	27.4	1.0	0.00	8	351.1	1.0	0.00	50	5.1	1.0	0.00	6
gamboost	539538	1.6	0.00	8	17.7	2.5	0.00	10	360.3	1.7	0.00	14	5.3	1.1	0.00	4
ksvm	424184	1.6	0.04	8	13.7	1.7	0.09	13	256.0	2.2	0.25	81	4.6	1.0	0.12	8
lm	629144	1.5	0.00	9	23.4	1.0	0.00	13	337.4	1.0	0.00	81	4.9	1.0	0.00	8
rf	478115	1.8	0.06	9	13.2	2.5	0.10	13	167.4	3.0	0.25	81	4.6	1.7	0.30	8

Table 1. Model performance and complexity on 4 regression tasks for various learners: linear models (lm), cross-validated regularized linear models (cvglmnet), kernel support vector machine (ksvm), random forest (rf), gradient boosted generalized additive model (gamboost), decision tree (cart) and decision tree with depth 2 (cart2).

that the underlying complexity measured as the range of MEC and IAS across the models varies. The bike dataset seems to be adequately described by only additive effects, since even random forests, which often model strong interactions show low interaction strength here. In contrast, the superconductivity dataset is better explained by models with more interactions. For the abalone dataset there are two models with low MSE: the support vector machine and the random forest. We might prefer the SVM, since main effects can be described with single numbers ($MEC = 1$) and interaction strength is low.

5 Improving Post-hoc Interpretation

Minimizing the number of features (NF), the interaction strength (IAS), and the main effect complexity (MEC) improves reliability and compactness of post-hoc interpretation methods such as partial dependence plots, ALE plots, feature importance, interaction effects and local surrogate models.

Fewer features, more compact interpretations. Minimizing the number of features improves the readability of post-hoc analysis results. The computational complexity and output size of most interpretation methods scales with $O(NF)$, like feature effect plots [1, 14] or feature importance [6, 11]. As demonstrated in Table 2, a model with fewer features has a more compact representation. If additionally $IAS = 0$, the ALE main effects fully characterize the prediction function. Interpretation methods that analyze 2-way feature interactions scale with $O(NF^2)$. A complete functional decomposition requires to estimate $\sum_{k=1}^{NF} \binom{NF}{k}$ components which has a computational complexity of $O(2^{NF})$.

Less interaction, more reliable feature effects. Feature effect plots such as partial dependence plots and ALE plots visualize the marginal relationship between a feature and the prediction. The estimated effects are averages across instances. The effects can vary greatly for individual instances and even have opposite directions when the model includes feature interactions.

In the following simulation, we trained three models with different capabilities of modeling interactions between features: a linear regression model, a support vector machine (radial basis kernel, $C=0.05$), and gradient boosted trees. We

simulated 500 data points with 4 features and a continuous target based on [15]. Figure 2 shows an increasing interaction strength depending on the model used. More interaction means that the feature effect curves become a less reliable summary of the model behavior.

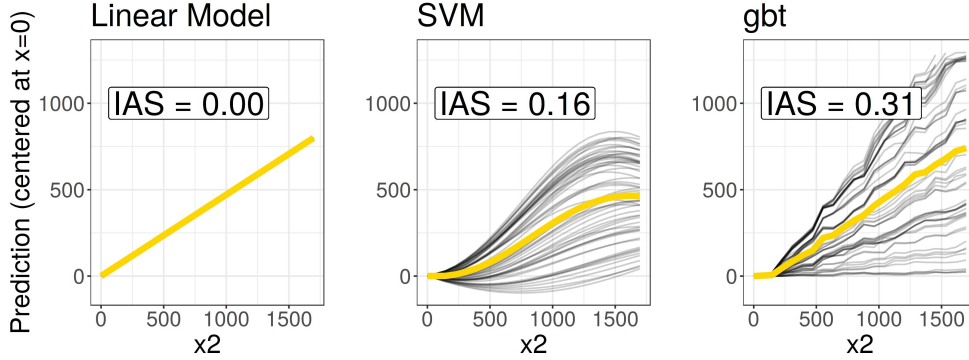


Fig. 2. The higher the interaction strength in a model (IAS increases from left to right), the less representative the Partial Dependence Plot (light thick line) becomes for individual instances represented by their Individual Conditional Expectation curves (dark thin lines).

The less complex the main effects, the better summarizable. In linear models, a feature effect can be expressed by a single number, the regression coefficient. If effects are non-linear the method of choice is visualization [1, 14]. Summarizing the effects with a single number (e.g., using average marginal effects [23]) can be misleading, e.g., the average effect might be zero for U-shaped feature effects. As a by-product of MEC, there is a third option: Instead of reporting a single number, the coefficients of the segmented linear model can be reported. Minimizing MEC means preferring models with main effects that can be described with fewer coefficients, offering a more compact model description.

6 Application: Multi-objective Optimization








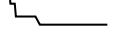
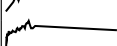

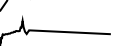

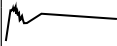

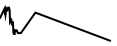



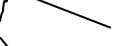




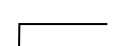

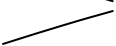



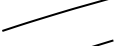
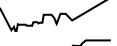


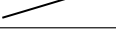










We demonstrate model selection for performance and complexity in a multi-objective optimization approach. For this example, we predict wine quality (scale from 0 to 10) [7] from the wines physical-chemical properties such as alcohol and residual sugar of 4870 white wines. It is difficult to know the desired compromise between model complexity and performance before modeling the data. A solution is multi-objective optimization [12]. We suggest searching over a wide spectrum of model classes and hyperparameter settings, which allows to select a suitable compromise between model complexity and performance.

We used the mlrMBO model-based optimization framework [19] with ParEGO [21] (500 iterations) to find the best models based on four objectives: number of

features used (NF), main effect complexity (MEC), interaction strength (IAS) and cross-validated mean absolute error (MAE) (5-fold cross-validated). We optimized over the space of following model classes (and hyperparameters): **CART** (maximum tree-depth and complexity parameter cp), support vector machine (cost C and inverse kernel width sigma), **elastic net** regression (regularization alpha and penalization lambda), **gradient boosted trees** (maximum depth, number of iterations), **gradient boosted generalized additive model** (number of iterations nrounds) and **random forest** (number of split features mtry).

Results. The multi-objective optimization resulted in 27 models. The measures had the following ranges: MAE 0.41 – 0.63, number of features 1 – 11, mean effect complexity 1 – 9 and interaction strength 0 – 0.71. For a more informative visualization, we propose to visualize the main effects together with the measures in Table 2. The selected models show different trade-offs between the measures.

Table 2. A selection of four models from the Pareto optimal set, along with their ALE main effect curves. From left to right, the columns show models with 1) lowest MAE, 2) lowest MAE when $MEC = 1$, 3) lowest MAE when $IAS \leq 0.2$, and 4) lowest MAE with $NF \leq 7$.

	gbt (maxdepth:8, nrounds:269)	svm (C:23.6979, sigma:0.0003)	gbt (maxdepth:3, nrounds:98)	CART (maxdepth:14, cp:0.0074)
MAE	0.41	0.58	0.52	0.59
MEC	4.2	1	4.5	2
IAS	0.64	0	0.2	0.2
NF	11	11	11	4
fixed.acidity				
volatile.acidity				
citric.acid				
residual.sugar				
chlorides				
free.sulfur.dioxide				
total.sulfur.dioxide				
density				
pH				
sulphates				
alcohol				

7 Discussion

We proposed three measures for machine learning model complexity based on functional decomposition: number of features used, interaction strength and main effect complexity. Due to their model-agnostic nature, the measures allow model selection and comparison across different types of models and they can be used as objectives in automated machine learning frameworks. This also includes "white-box" models: For example, the interaction strength of interaction terms in a linear model or the complexity of smooth effects in generalized additive models can be quantified and compared across models. We argued that minimizing these measures for a machine learning model improves its post-hoc interpretation. We demonstrated that the measures can be optimized directly with multi-objective optimization to make the trade-off between performance and post-hoc interpretability explicit.

Limitations. The proposed decomposition of the prediction function and definition of the complexity measures will not be appropriate in every situation. For example, all higher order effects are combined into a single interaction strength measure that does not distinguish between two-way interactions and higher order interactions. However, the framework of accumulated local effect decomposition allows to estimate higher order effects and to construct different interaction measures. The main effect complexity measure only considers linear segments but not, e.g., seasonal components or other structures. Furthermore, the complexity measures quantify machine learning models from a functional point of view and ignore the structure of the model (e.g., whether it can be represented by a tree). For example, main effect complexity and interaction strength measures can be large for short decision trees (e.g. in Table 1).

Implementation. The code for this paper is available at https://github.com/compstat-lmu/paper_2019_iml_measures. For the examples and experiments we relied on the `mlr` package [5] in R [29].

Acknowledgements. This work is funded by the Bavarian State Ministry of Science and the Arts in the framework of the Centre Digitisation.Bavaria (ZD.B) and supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibilities for its content.

References

- [1] Apley, D.: Visualizing the effects of predictor variables in black box supervised learning models. arXiv preprint arXiv:1612.08468 (2016)
- [2] Apley, D.: Aleplot: Accumulated local effects (ale) plots and partial dependence(pd) plots. CRAN (2017)
- [3] Askira-Gelman, I.: Knowledge discovery: comprehensibility of the results. In: Proceedings of the thirty-first Hawaii international conference on system sciences. vol. 5, pp. 247–255. IEEE (1998)
- [4] Bibal, A., Frénay, B.: Interpretability of machine learning models and representations: an introduction. In: Proceedings on ESANN. pp. 77–82 (2016)

- [5] Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Studerus, E., Casalicchio, G., Jones, Z.M.: mlr: Machine learning in R. *Journal of Machine Learning Research* **17**(170), 1–5 (2016)
- [6] Casalicchio, G., Molnar, C., Bischl, B.: Visualizing the feature importance for black box models. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. pp. 655–670. Springer (2018)
- [7] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., Reis, J.: Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems* **47**(4), 547–553 (2009)
- [8] Dhurandhar, A., Iyengar, V., Luss, R., Shanmugam, K.: TIP: typifying the interpretability of procedures. *arXiv preprint arXiv:1706.02952* (2017)
- [9] Dua, D., Graff, C.: UCI machine learning repository (2017), <http://archive.ics.uci.edu/ml>
- [10] Fanaee-T, H., Gama, J.: Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence* pp. 1–15 (2013)
- [11] Fisher, A., Rudin, C., Dominici, F.: All Models are Wrong but many are Useful: Variable Importance for Black-Box, Proprietary, or Misspecified Prediction Models, using Model Class Reliance. *arXiv preprint arXiv:1801.01489* (2018)
- [12] Freitas, A.A.: Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter* **15**(1), 1–10 (2014)
- [13] Friedler, S.A., Roy, C.D., Scheidegger, C., Slack, D.: Assessing the local interpretability of machine learning models. *arXiv preprint arXiv:1902.03501* (2019)
- [14] Friedman, J.H.: Greedy function approximation: a gradient boosting machine. *Annals of statistics* pp. 1189–1232 (2001)
- [15] Friedman, J.H., et al.: Multivariate adaptive regression splines. *The annals of statistics* **19**(1), 1–67 (1991)
- [16] Fürnkranz, J., Gamberger, D., Lavrač, N.: *Foundations of rule learning*. Springer Science & Business Media (2012)
- [17] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* **51**(5), 93 (2018)
- [18] Hamidieh, K.: A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science* **154**, 346–354 (2018)
- [19] Horn, D., Bischl, B.: Multi-objective parameter configuration of machine learning algorithms using model-based optimization. In: *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. pp. 1–8. Ieee (2016)
- [20] Huysmans, J., Dejaeger, K., Mues, C., Vanthienen, J., Baesens, B.: An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems* **51**(1), 141–154 (2011)
- [21] Knowles, J.: Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* **10**(1), 50–66 (2006)

-
- [22] Lakkaraju, H., Kamar, E., Caruana, R., Leskovec, J.: Interpretable & explorable approximations of black box models. arXiv preprint arXiv:1707.01154 (2017)
 - [23] Leeper, T.J.: Interpreting regression results using average marginal effects with R's margins. CRAN (2017)
 - [24] Molnar, C.: Interpretable Machine Learning (2019), <https://christophm.github.io/interpretable-ml-book/>
 - [25] Molnar, C., Bischl, B., Casalicchio, G.: iml: An R package for interpretable machine learning. *JOSS* **3**(26), 786 (2018)
 - [26] Philipp, M., Rusch, T., Hornik, K., Strobl, C.: Measuring the stability of results from supervised statistical learning. *Journal of Computational and Graphical Statistics* **27**(4), 685–700 (2018)
 - [27] Plumb, G., Al-Shedivat, M., Xing, E., Talwalkar, A.: Regularizing black-box models for improved interpretability. arXiv preprint arXiv:1902.06787 (2019)
 - [28] Poursabzi-Sangdeh, F., Goldstein, D.G., Hofman, J.M., Vaughan, J.W., Wallach, H.: Manipulating and measuring model interpretability. arXiv preprint arXiv:1802.07810 (2018)
 - [29] R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2018)
 - [30] Ribeiro, M.T., Singh, S., Guestrin, C.: Model-agnostic interpretability of machine learning. arXiv preprint arXiv:1606.05386 (2016)
 - [31] Rüping, S., et al.: Learning interpretable models. Univ. Dortmund (2006), <http://d-nb.info/997491736>
 - [32] Schielzeth, H.: Simple means to improve the interpretability of regression coefficients. *Methods in Ecology and Evolution* **1**(2), 103–113 (2010)
 - [33] Ustun, B., Rudin, C.: Supersparse linear integer models for optimized medical scoring systems. *Machine Learning* **102**(3), 349–391 (2016)
 - [34] Yang, H., Rudin, C., Seltzer, M.: Scalable bayesian rule lists. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. pp. 3921–3930. JMLR. org (2017)
 - [35] Zhou, Q., Liao, F., Mou, C., Wang, P.: Measuring interpretability for different types of machine learning models. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. pp. 295–308 (2018)

13. Multi-objective counterfactual explanations

Contributing article:

Dandl, S., Molnar, C., Binder, M., and Bischl, B. (2020). Multi-objective counterfactual explanations. In: *Bäck T. et al. (eds) Parallel Problem Solving from Nature – PPSN XVI. PPSN 2020. Lecture Notes in Computer Science, vol 12269.*, vol 12269, (pp. 448-469). Springer, Cham. https://doi.org/10.1007/978-3-030-58112-1_31

Copyright information:

Creative Commons Attribution 4.0 International License (CC BY 4.0)

Author contributions:

The paper was mainly written by Susanne Dandl, who also implemented most of the software and application. Christoph Molnar implemented the benchmarks for DiCE and Recourse in Python, and ported the results to R. Christoph Molnar and Susanne Dandl had many discussions about technical and methodological details regarding the counterfactual objectives, evaluation metrics and software comparisons. All authors added input and suggested several notable modifications, proofread and revised the paper.

Multi-Objective Counterfactual Explanations^{*}

Susanne Dandl^[0000–0003–4324–4163], Christoph Molnar^[0000–0003–2331–868X],
Martin Binder, and Bernd Bischl^[0000–0001–6002–6980]

Department of Statistics, LMU Munich, Ludwigstr. 33, 80539 Munich, Germany
`susanne.dandl@stat.uni-muenchen.de`

Abstract. Counterfactual explanations are one of the most popular methods to make predictions of black box machine learning models interpretable by providing explanations in the form of ‘what-if scenarios’. Most current approaches optimize a collapsed, weighted sum of multiple objectives, which are naturally difficult to balance a-priori. We propose the Multi-Objective Counterfactuals (MOC) method, which translates the counterfactual search into a multi-objective optimization problem. Our approach not only returns a diverse set of counterfactuals with different trade-offs between the proposed objectives, but also maintains diversity in feature space. This enables a more detailed post-hoc analysis to facilitate better understanding and also more options for actionable user responses to change the predicted outcome. Our approach is also model-agnostic and works for numerical and categorical input features. We show the usefulness of MOC in concrete cases and compare our approach with state-of-the-art methods for counterfactual explanations.

Keywords: Interpretability · Interpretable machine learning · Counterfactual explanations · Multi-objective optimization · NSGA-II.

1 Introduction

Interpretable machine learning methods have become very important in recent years to explain the behavior of black box machine learning (ML) models. A useful method for explaining *single* predictions of a model are counterfactual explanations. ML credit risk prediction is a common motivation for counterfactuals. For people whose credit applications have been rejected, it is valuable to know why they have not been accepted, either to understand the decision making process or to assess their actionable options to change the outcome. Counterfactuals provide these explanations in the form of “if these features had different values, your credit application would have been accepted”. For such explanations to be plausible, they should only suggest small changes in a few features. Therefore, counterfactuals can be defined as close neighbors of an actual

^{*} This work has been partially supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A and by the Bavarian State Ministry of Science and the Arts in the framework of the Centre Digitisation.Bavaria (ZD.B). The authors of this work take full responsibility for its content.

data point, but their predictions have to be sufficiently close to a (usually quite different) desired outcome. Counterfactuals explain why a certain outcome was not reached, can offer potential reasons to object against an unfair outcome and give guidance on how the desired prediction could be reached in the future [35]. Note that counterfactuals are also valuable for predictive modelers on a more technical level to investigate the pointwise robustness and the pointwise bias of their model.

2 Related Work

Counterfactuals are closely related to adversarial perturbations. These have the aim to deceive ML models instead of making the models interpretable [30]. Attribution methods such as Local Interpretable Model-agnostic Explanations (LIME) [27] and Shapley Values [22] explain a prediction by determining how much each feature contributed to it. Counterfactual explanations differ from feature attributions since they generate data points with a different, desired prediction instead of attributing a prediction to the features.

Counterfactual methods can be model-agnostic or model-specific. The latter usually exploit the internal structure of the underlying ML model, such as the trained weights of a neural network, while the former are based on general principles which work for arbitrary ML models - often by only assuming access to the prediction function of an already fitted model. Several model-agnostic counterfactual methods have been proposed [8,11,16,18,25,29,37]. Apart from Grath et al. [11], these approaches are limited to classification. Unlike the other methods, the method of Poyiadzi et al. [25] can obtain plausible counterfactuals by constructing feasible paths between data points with opposite predictions.

A model-specific approach was proposed by Wachter et al. [35], who also introduced and formalized the concept of counterfactuals in predictive modeling. Like many model-specific methods [15,20,24,28,33] their approach is limited to differentiable models. The approach of Tolomei et al. [32] generates explanations for tree-based ensemble binary classifiers. As with [35] and [20], it only returns a single counterfactual per run.

3 Contributions

In this paper, we introduce Multi-Objective Counterfactuals (MOC), which to the best of our knowledge is the first method to formalize the counterfactual search as a multi-objective optimization problem. We argue that the mathematical problem behind the search for counterfactuals should be naturally addressed as multi-objective. Most of the above methods optimize a collapsed, weighted sum of multiple objectives to find counterfactuals, which are naturally difficult to balance a-priori. They carry the risk of arbitrarily reducing the solution set to a single candidate without the option to discuss inherent trade-offs – which should be especially relevant for model interpretation that is by design very hard to precisely capture in a (single) mathematical formulation.

Compared to Wachter et al. [35], we use a distance metric for mixed feature spaces and two additional objectives: one that measures the number of feature changes to obtain sparse and therefore more interpretable counterfactuals, and one that measures the closeness to the nearest observed data points for more plausible counterfactuals. MOC returns a Pareto set of counterfactuals that represents different trade-offs between our proposed objectives, and which are constructed to be diverse in feature space. This seems preferable because changes to different features can lead to a desired counterfactual prediction¹ and it is more likely that some counterfactuals meet the (hidden) preferences of a user. A single counterfactual might even suggest a strategy that is interpretable but not actionable (e.g., ‘reduce your number of pregnancies’) or counterproductive in more general contexts (e.g., ‘increase your age to reduce the risk of diabetes’). In addition, if multiple otherwise quite different counterfactuals suggest changes to the same feature, the user may have more confidence that the feature is an important lever to achieve the desired outcome. We refer the reader to Appendix A for two concrete examples illustrating the above.

Compared to other counterfactual methods, MOC is model-agnostic and handles classification, regression and mixed feature spaces, which furthermore increases its practical usefulness in general applications. Together with [16], our paper also includes one of the first benchmark studies that compares multiple counterfactual methods on multiple, heterogeneous datasets.

4 Methodology

[35] loosely define counterfactuals as:

“You were denied a loan because your annual income was 30,000. If your income had been 45,000, you would have been offered a loan. Here the statement of decision is followed by a counterfactual, or statement of how the world would have to be different for a desirable outcome to occur. Multiple counterfactuals are possible, as multiple desirable outcomes can exist, and there may be several ways to achieve any of these outcomes.”

We now formalize this statement by stating four objectives, which a counterfactual should adhere to. In the subsequent section we provide detailed definitions of these objectives and tie them together as a multi-objective optimization problem in order to generate a diverse set of different trade-off solutions.

4.1 Multi-Objective Counterfactuals

Definition 1 (Counterfactual Explanation). *Let $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$ be a prediction function, \mathcal{X} the feature space and $Y' \subset \mathbb{R}$ a set of desired outcomes. The latter can either be a single value or an interval of values. We define a counterfactual explanation \mathbf{x}' for an observation \mathbf{x}^* as a data point fulfilling the following: (1)*

¹ Rashomon effect [5]

its prediction $f(\mathbf{x}')$ is close to the desired outcome set Y' , (2) it is close to \mathbf{x}^* in the \mathcal{X} space, (3) it differs from \mathbf{x}^* only in a few features, and (4) it is a plausible data point according to the probability distribution $\mathbb{P}_{\mathcal{X}}$. For classification models, we assume that \hat{f} returns the probability for a user-selected class and Y' has to be the desired probability (range).

This can be translated into a multi-objective minimization task:

$$\min_{\mathbf{x}} \mathbf{o}(\mathbf{x}) := \min_{\mathbf{x}} (o_1(\hat{f}(\mathbf{x}), Y'), o_2(\mathbf{x}, \mathbf{x}^*), o_3(\mathbf{x}, \mathbf{x}^*), o_4(\mathbf{x}, \mathbf{X}^{obs})), \quad (1)$$

with $\mathbf{o} : \mathcal{X} \rightarrow \mathbb{R}^4$ and \mathbf{X}^{obs} as the observed (i.e. training) data. The first component o_1 quantifies the distance between $\hat{f}(\mathbf{x})$ and Y' . We define it as:²

$$o_1(\hat{f}(\mathbf{x}), Y') = \begin{cases} 0 & \text{if } \hat{f}(\mathbf{x}) \in Y' \\ \inf_{y' \in Y'} |\hat{f}(\mathbf{x}) - y'| & \text{else} \end{cases}.$$

The second component o_2 quantifies the distance between \mathbf{x}^* and \mathbf{x} using the Gower distance to account for mixed features [10]:

$$o_2(\mathbf{x}, \mathbf{x}^*) = \frac{1}{p} \sum_{j=1}^p \delta_G(x_j, x_j^*) \in [0, 1]$$

with p being the number of features. The value of δ_G depends on the feature type:

$$\delta_G(x_j, x_j^*) = \begin{cases} \frac{1}{\hat{R}_j} |x_j - x_j^*| & \text{if } x_j \text{ is numerical} \\ \mathbb{I}_{x_j \neq x_j^*} & \text{if } x_j \text{ is categorical} \end{cases}$$

with \hat{R}_j as the value range of feature j , extracted from the observed dataset.

Since the Gower distance does not take into account how many features have been changed, we introduce objective o_3 , which counts the number of changed features using the L_0 norm:

$$o_3(\mathbf{x}, \mathbf{x}^*) = \|\mathbf{x} - \mathbf{x}^*\|_0 = \sum_{j=1}^p \mathbb{I}_{x_j \neq x_j^*}.$$

The fourth objective o_4 measures the weighted average Gower distance between \mathbf{x} and the k nearest observed data points $\mathbf{x}^{[1]}, \dots, \mathbf{x}^{[k]} \in \mathbf{X}^{obs}$ as an empirical approximation of how likely \mathbf{x} originates from the distribution of \mathcal{X} :

$$o_4(\mathbf{x}, \mathbf{X}^{obs}) = \sum_{i=1}^k w^{[i]} \frac{1}{p} \sum_{j=1}^p \delta_G(x_j, x_j^{[i]}) \in [0, 1] \text{ where } \sum_{i=1}^k w^{[i]} = 1.$$

² We chose the L_1 norm over the L_2 norm for a natural interpretation. Its non-differentiability is negligible for evolutionary optimization.

Throughout this paper, we set k to 1. Further procedures to increase the plausibility of the counterfactuals are integrated into the optimization algorithm and are described in Section 4.3.

Balancing the four objectives is difficult since the objectives contradict each other. For example, minimizing the distance between counterfactual outcome and desired outcome Y' (o_1) becomes more difficult when we require counterfactual feature values close to \mathbf{x}^* (o_2 and o_3) and to the observed data (o_4).

4.2 Counterfactual Search

Our proposed method MOC uses the *Nondominated Sorting Genetic Algorithm II* (NSGA-II) [7] with modifications specific to the problem considered. First, unlike the original NSGA-II, it uses *mixed integer evolutionary strategies* (MIES) [19] to work with the mixed discrete and continuous search space. Furthermore, a different crowding distance sorting algorithm is used, and we propose some optional adjustments tailored to the counterfactual search in the upcoming section.

For MOC, each candidate is described by its feature vector (the ‘genes’) and the objective values of the candidates are evaluated by Eq. (1). Features of candidates are recombined and mutated with predefined probabilities – some of the control parameters of MOC. Numerical features are recombined by the simulated binary crossover recombinator [6], all other feature types by the uniform crossover recombinator [31]. Based on [19], numerical features are mutated by the scaled Gaussian mutator. Categorical features are altered by uniformly sampling from their admissible levels, while binary and logical features are simply flipped. After recombination and mutation, some feature values are randomly set to the values of \mathbf{x}^* with a given (low) probability – another control parameter – to prevent all features from deviating from \mathbf{x}^* .

Contrary to NSGA-II, the crowding distance is computed not only in the objective space \mathbb{R}^4 (L_1 norm) but also in the feature space \mathcal{X} (Gower distance), and the distances are summed up with equal weighting. As a result, candidates are more likely kept if they differ greatly from another candidate in their feature values although they are similar in the objective values. Diversity in \mathcal{X} is desired because the chances of obtaining counterfactuals that meet the (hidden) preferences of users are higher. This approach is based on Avila et al. [2].

MOC stops if either a predefined number of generations is reached (default) or the performance no longer improves for a given number of successive generations.

4.3 Further Modifications

Initialization Naively, we could initialize a population by uniformly sampling some feature values from their full range of possible values, while randomly setting other features to the values of \mathbf{x}^* to induce sparsity. However, if a feature has a large influence on the prediction, it should be more likely that the counterfactual values differ from \mathbf{x}^* . The importance of a feature for an entire dataset can be measured as the standard deviation of the partial dependence

plot [12]. Analogously, we propose to measure the feature importance for a single prediction with the standard deviation of the Individual Conditional Expectation (ICE) curve of \mathbf{x}^* . ICE curves show for one observation and for one feature how the prediction changes when the feature is changed, while other features are fixed to the values of the considered observation [9]. The greater the standard deviation of the ICE curve, the higher we set the probability that the feature value is initialized with a different value than the one of \mathbf{x}^* . Therefore, the standard deviation σ_j^{ICE} of each feature x_j is transformed into probabilities within $[p_{min}, p_{max}] \cdot 100\%$:

$$P(\text{value differs}) = \frac{(\sigma_j^{ICE} - \min(\sigma^{ICE})) \cdot (p_{max} - p_{min})}{\max(\sigma^{ICE}) - \min(\sigma^{ICE})} + p_{min}$$

with $\sigma^{ICE} := (\sigma_1^{ICE}, \dots, \sigma_p^{ICE})$. p_{min} and p_{max} are control parameters with default values 0.01 and 0.99.

Actionability To get more actionable counterfactuals, extreme values of numerical features outside a predefined range are capped to the upper or lower bound after recombination and mutation. The ranges can either be derived from the minimum and maximum values of the features in the observed dataset or users can define these ranges. In addition, users can identify non-actionable features such as the country of birth or gender. The values of these features are permanently set to the values of \mathbf{x}^* for all candidates within MOC.

Penalization Furthermore, candidates whose predictions are further away from the target than a predefined distance $\epsilon \in \mathbb{R}$ can be penalized. After the candidates have been sorted into fronts F_1 to F_K using nondominated sorting, the candidate that violates the constraint least will be reassigned to front F_{K+1} , the candidate with the second smallest violation to F_{K+2} , and so on. The concept is based on Deb et al. [7]. Since the constraint violators are in the last fronts, they are less likely to be selected for the next generation.

Mutation Since the aforementioned mutators do not take the data distribution into account and can potentially generate unlikely new candidates, we suggest a conditional mutator. It generates plausible feature values conditional on the values of the other features. For each input feature, we trained a transformation tree [14] on X^{obs} , which is then used to sample values from the conditional distribution. We mutate the feature in randomized order since a feature mutation now depends on the previous changes.

How our proposed strategies for initialization and mutation affect MOC is later examined in a benchmark study (Sections 6 & 7).

4.4 Evaluation Metric

We use the popular hypervolume indicator (HV) [38] to evaluate the quality of our estimated Pareto front, with reference point $\mathbf{s} = (\inf_{y' \in Y'} |\hat{f}(\mathbf{x}^*) - y'|, 1, p, 1)$,

representing the maximal values of the objectives. We compute the HV always over the complete archive of evaluated solutions.

4.5 Tuning of Parameters

We also use HV, when we tune MOC’s control parameters – population size, the probabilities for recombining and mutating a feature of a candidate – with iterated F-racing [21]. Furthermore, we let iterated F-racing decide whether our proposed strategies for initialization and mutation of Section 4.3 are preferable. Tuning is performed on six binary classification datasets from OpenML [34] – which were not used in the benchmark. A summary of the tuning setup and results can be found in Table 5 in Appendix B. Iterated F-racing found both our initialization and mutation strategy to be advantageous. The tuned parameters were used for the credit data application and the benchmark study.

5 Credit Data Application

This section demonstrates the usefulness of MOC to explain the prediction of credit risk using the German credit dataset [13]. The dataset has 522 complete observations and nine features containing credit and customer information. Categories with few case numbers were combined. The binary target indicates whether a customer has a ‘good’ or ‘bad’ credit risk. We chose the first observation of the dataset as \mathbf{x}^* with the following feature values:

Age	Sex	Job	Housing	Saving accounts	Checking account	Credit amount	Duration	Purpose
22	female	2	own	little	moderate	5951	48	radio/TV

We tuned a support vector machine (with radial-basis (RBF) kernel) on the remaining data with the same tuning setup as for the benchmark (Appendix C). To obtain a single numerical outcome, only the predicted probability for the class ‘good’ credit risk was returned. We obtained an accuracy of 0.64 for the model using two nested cross-validations (CV) (5-fold CV in outer and inner loop) and a predicted probability for ‘good’ credit risk of 0.41 for \mathbf{x}^* .

We set the desired outcome interval to $Y' = [0.5, 1]$, which indicates a change to a ‘good’ credit risk. We generated counterfactuals using MOC with the parameter setting selected by iterated F-racing. Candidates with a prediction below 0.5 were penalized.

A total of 136 counterfactuals were found by MOC. In the following, we focus upon the 82 of them with predictions within $[0.5, 1]$. Credit *duration* was changed for all counterfactuals, followed by *credit amount* (86%). Since a user might not want to investigate all returned counterfactuals individually (in feature space), we provide a visual summary of the Pareto set in Figure 1, either as a parallel coordinate plot or a response surface plot³ along two features. All counterfactuals had values equal to or smaller than the values of \mathbf{x}^* for *duration* and *credit amount*. The response surface plot illustrates why these feature changes were

³ This is equivalent to a 2-D ICE-curve through \mathbf{x}^* [9]. We refer to Section 4.3 for a general definition of ICE curves.

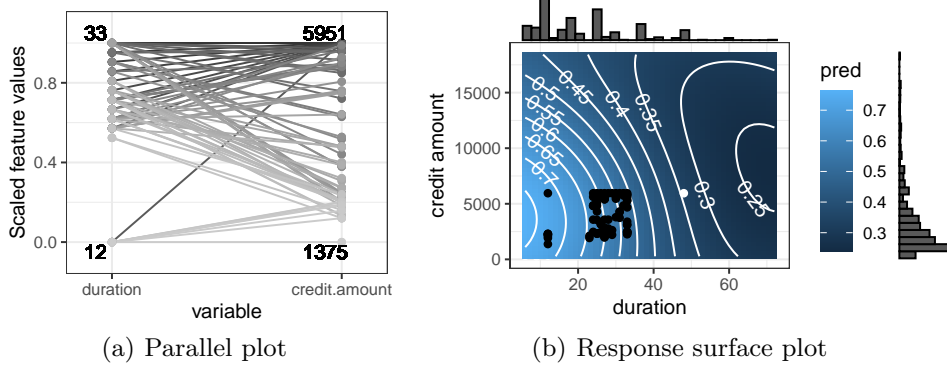


Fig. 1. Visualization of counterfactuals for the first data point \mathbf{x}^* of the credit dataset. (a) Feature values of the counterfactuals. Only changed features are shown. The given numbers indicate the minimum and maximum feature values of the counterfactuals. (b) Response surface plot for the model prediction along features duration and credit amount, holding other feature values constant at the value of \mathbf{x}^* . Colors and contour lines indicate the predicted value. The white point is \mathbf{x}^* and the black points are the counterfactuals that only proposed changes in duration and/or credit amount. The histograms show the marginal distributions of the features in the observed dataset.

recommended. The color gradient and contour lines indicate that either *duration* or both *credit amount* and *duration* must be decreased to reach the desired outcome. Due to the fourth objective and the conditional mutator, we obtained counterfactuals in high density areas (indicated by histograms). Counterfactuals in the lower left corner seem to be in a less favorable region far from \mathbf{x}^* , but they are close to the training data.

6 Experimental Setup

In this section, the performance of MOC is evaluated in a benchmark study for binary classification. The datasets are from the OpenML platform [34] and are briefly described in Table 1. We selected datasets with no missing values, with up to 3500 observations and a maximum of 40 features. We randomly selected ten observed data points per dataset as \mathbf{x}^* and excluded them from the training data. For each dataset, we tuned and trained the following models: logistic regression, random forest, xgboost, RBF support vector machine and a one-hidden-layer neural network. The tuning parameter set and the performance using nested resampling are in Table 8 in Appendix C. Each model returned only the probability for one class. The desired target for each \mathbf{x}^* was set to the opposite of the predicted class:

$$Y' = \begin{cases}]0.5, 1] & \text{if } \hat{f}(\mathbf{x}^*) \leq 0.5 \\ [0, 0.5] & \text{else} \end{cases}.$$

Table 1. Description of benchmark datasets. Legend: *task*: OpenML task id; *Obs*: Number of rows; *Cont/Cat*: Number of continuous/categorical features.

Task	Name	Obs	Cont	Cat
3718	boston	506	12	1
3846	cmc	1473	2	7
145976	diabetes	768	8	0
9971	ilpd	583	9	1
3913	kc2	522	21	0
3	kr-vs-kp	3196	0	36
3749	no2	500	7	0
3918	pc1	1109	21	0
3778	plasma_retinol	315	10	3
145804	tic-tac-toe	958	0	9

Table 2. MOC’s coverage rate of methods to be compared per dataset averaged over all models. The number of nondominated counterfactuals for each method are given in parentheses. Higher values of coverage indicate that MOC dominates the other method. The * indicates that the binomial test with $H_0 : p < 0.5$ that a counterfactual is covered by MOC is significant at the 0.05 level.

	DiCE	Recourse	Tweaking
boston	1* (36)	0.92* (24)	0.9* (10)
cmc	1* (17)		0.75 (8)
diabetes	1* (64)	0.45 (40)	1 (3)
ilpd	1* (26)	1* (37)	0.83 (6)
kc2	1* (53)	0.31 (55)	1 (2)
kr-vs-kp	1* (8)		0.2 (10)
no2	1* (58)	0.5 (12)	0.9* (10)
pc1	1* (60)	0.66* (38)	
plasma_retinol	1* (7)		0.89* (9)
tic-tac-toe	1* (20)		0.75 (8)

The benchmark study aimed to answer two research questions:

- Q1) How does MOC perform compared to other state-of-the-art methods for counterfactuals?
- Q2) How do our proposed strategies for initialization and mutation of Section 4.3 influence the performance of MOC?

For the first one, we compared MOC – once with and once without our proposed strategies for initialization and mutation – with ‘DiCE’ by Mothilal et al. [24], ‘Recourse’ by Ustun et al. [33] and ‘Tweaking’ by Tolomei et al. [32]. We chose DiCE, Recourse and Tweaking because they are implemented in general open source code libraries.⁴ The methods are only applicable to certain models: DiCE can handle neural networks and logistic regressions, Recourse can handle logistic regressions and Tweaking can handle random forests. Since Recourse can only process binary and numerical features, we did not train logistic regression on cmc, tic-tac-toe, kr-vs-kp and plasma_retinol. As a baseline, we selected the closest observed data point to \mathbf{x}^* (according to the Gower distance) that has a prediction equal to our desired outcome. Since this approach is part of the *What-If Tool* [36], we call this approach ‘Whatif’.

The parameters of DiCE, Recourse and Tweaking were set to the default values recommended by the authors (Appendix D). To allow for a fair comparison, we initialized MOC with the parameters of iterated F-racing which were tuned on other binary classification datasets (Appendix B). While MOC can potentially return several hundreds of counterfactuals, the other methods are designed to either return one or a few. We have therefore limited the maximum number of

⁴ Most other counterfactual methods are implemented for specific examples, but cannot be easily used for other datasets.

counterfactuals to ten for all approaches.⁵ Tweaking and Whatif generated only one counterfactual by design. For MOC we reduced the number of counterfactuals by preferring the ones that achieved the target prediction Y' and/or the highest HV contribution.

For all methods, only nondominated counterfactuals were considered for the evaluation. Since we are interested in a diverse set of counterfactuals, we evaluate the methods based on the size of their counterfactual set, its objective values, and the coverage rate derived from the coverage indicator by Zitzler and Thiele [38]. The coverage rate is the relative frequency with which counterfactuals of a method are dominated by MOC's counterfactuals for a certain model and \mathbf{x}^* . A counterfactual covers another counterfactual if it dominates it, and it does not cover the other if both have the same objective values or the other has lower values in at least one objective. A coverage rate of 1 implies that for each generated counterfactual of a method MOC generated at least one dominating counterfactual. We only computed the coverage rate over counterfactuals that met the desired target Y' .

To answer the second research question, we compared the dominated HV over the generations of MOC with and without our proposed strategies for initialization and mutation. As a baseline, we used a random search approach that has the same population size (20) and number of generations (175) as MOC. In each generation, some feature values were uniformly sampled from their set of possible values derived from the observed data and \mathbf{x}^* , while other features were set to the values of \mathbf{x}^* . The HV for one generation was computed over the newly generated candidates combined with the candidates of the previous generations.

7 Results

Q1) MOC vs. State-of-the-Art Counterfactual Methods

Table 2 shows the coverage rate of each method (to be compared) by the tuned MOC per dataset. Some fields are empty because Recourse could not process features with more than two classes and Tweaking never achieved the desired outcome for pc1. MOC's counterfactuals dominated all counterfactuals of DiCE for all datasets. The same holds for Tweaking except for kr-vs-kp and tic-tac-toe because the counterfactuals of Tweaking had the same objective values as the ones of MOC. MOC's coverage rate of Recourse only exceeded 90% for boston and ilpd since Recourse's counterfactuals often deviated less from \mathbf{x}^* (but performed worse in other objectives).

Figure 2 compares MOC (with (*mocmod*) and without (*moc*) our proposed strategies for initialization and mutation) with the other methods for the datasets diabetes and no2 and for each model separately. The resulting boxplots for all other datasets are shown in Figures 4 and 5 in the Appendix. They agree with the results shown here. Compared to the other methods, both versions of MOC found the most nondominated solutions, which met the target and

⁵ Note that this artificially penalizes our approach in the benchmark comparison.

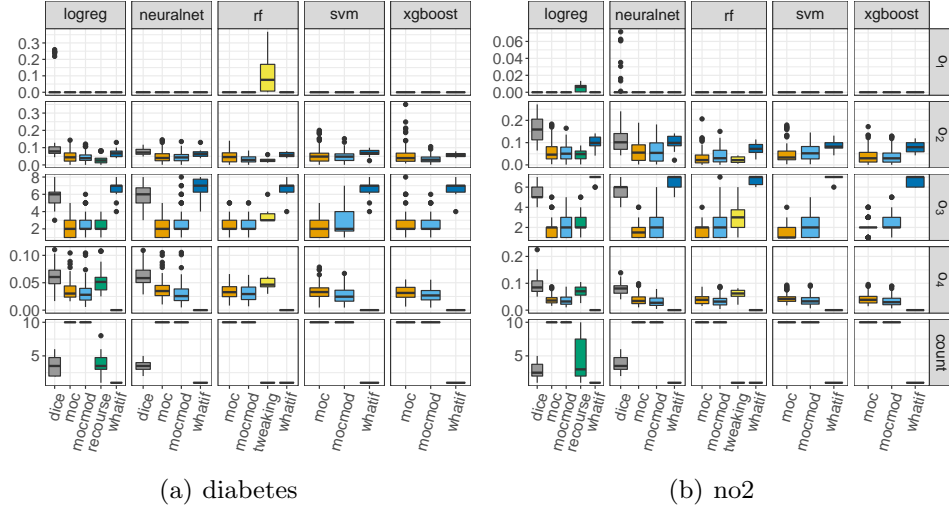


Fig. 2. Boxplots of the objective values and number of nondominated counterfactuals (*count*) per model for MOC with our proposed strategies for initialization and mutation (*mocmod*), MOC without these modifications, Whatif, DiCE, Recourse and Tweaking for the datasets diabetes and no2. Lower values are better except for *count*.

changed the least features. DiCE performed worse than MOC in all objectives. Tweaking’s counterfactuals were often closer to \mathbf{x}^* , but they were further away from the nearest training data point and more features were changed. Tweaking’s counterfactuals often did not reach the desired outcome because they stayed too close to \mathbf{x}^* . The MOC with our proposed modifications found counterfactuals closer to \mathbf{x}^* and the observed data, but required more feature changes compared to MOC without the modifications.

Q2) MOC Strategies for Initialization and Mutation

Figure 3 shows the ranks of the dominated HVs for MOC without modifications, for each modification of MOC and random search. Ranks were calculated per dataset, model, \mathbf{x}^* and generation, and were averaged over all datasets, models and \mathbf{x}^* . We transformed HVs to ranks because the HVs are not comparable across \mathbf{x}^* . It can be seen that the MOC with our proposed modifications clearly outperforms the MOC without these modifications. The ranks of the initial population were higher when the ICE curve variance was used to initialize the candidates. The use of the conditional mutator led to higher dominated HVs over the generations. We received the best performance over the generations when both modifications were used. At each generation, all versions of MOC outperformed random search. Figure 6 in the Appendix shows the ranks over the generations for each dataset separately. They largely agree with the results shown here. The performance gains of MOC compared to random search were particularly evident for higher-dimensional datasets.

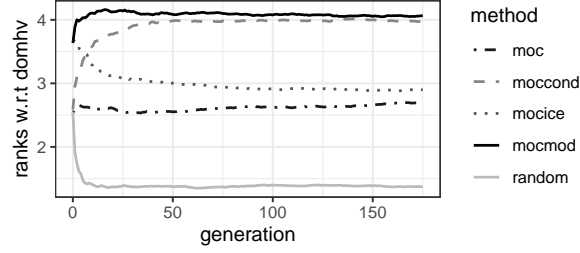


Fig. 3. Comparison of the ranks w.r.t. the dominated HV (*domhv*) per generation averaged over all models and datasets. For each approach, the population size of each generation was 20. A higher HV and therefore a higher rank is better. Legend: *moc*: MOC without our proposed modifications; *moccond*: MOC with the conditional mutator; *mocice*: MOC with the ICE curve variance initialization; *mocmod*: MOC with both modifications; *random*: random search.

8 Conclusion and Outlook

In this paper, we introduced Multi-Objective Counterfactuals (MOC), which to the best of our knowledge is the first method to formalize the counterfactual search as a multi-objective optimization problem. Compared to state-of-the-art approaches, MOC returns a diverse set of counterfactuals with different trade-offs between our proposed objectives. Furthermore, MOC is model-agnostic and suited for classification, regression and mixed feature spaces. We demonstrated the usefulness of MOC to explain a prediction on the German credit dataset and showed in a benchmark study that MOC finds more counterfactuals than other counterfactual methods that are closer to the training data and required fewer feature changes. Our proposed initialization strategy (based on ICE curve variances) and our conditional mutator resulted in higher performance in fewer evaluations and in counterfactuals that were closer to the data point we were interested in and to the observed data.

MOC has only been evaluated on binary classification, and only with respect to the dominated HV and the individual objectives. It is an open question how to let users select the counterfactuals that meet their – a-priori unknown – trade-off between the objectives. We leave these investigations to future research.

9 Electronic Submission

The complete code of the algorithm and the code to reproduce the experiments and results of this paper are available at <https://github.com/susanne-207/moc>. The implementation of MOC is based on our implementation of [19], which we also used for [3]. We will provide an open source R library with our implementation of the method based on the *iml* package [23].

References

1. Allaire, J., Chollet, F.: keras: R Interface to 'Keras' (2019), <https://keras.rstudio.com>, R package version 2.3.0
2. Avila, S.L., Krähenbühl, L., Sareni, B.: A Multi-Niching Multi-Objective Genetic Algorithm for Solving Complex Multimodal Problems. In: OIPE. Sorrento, Italy (2006), <https://hal.archives-ouvertes.fr/hal-00398660>
3. Binder, M., Moosbauer, J., Thomas, J., Bischl, B.: Multi-Objective Hyperparameter Tuning and Feature Selection using Filter Ensembles (2019), accepted at GECCO 2020
4. Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Studerus, E., Casalicchio, G., Jones, Z.M.: mlr: Machine Learning in R. *Journal of Machine Learning Research* **17**(170), 1–5 (2016), <http://jmlr.org/papers/v17/15-066.html>, R package version 2.17
5. Breiman, L.: Statistical Modeling: The Two Cultures. *Statistical Science* **16**(3), 199–231 (08 2001). <https://doi.org/10.1214/ss/1009213726>, <https://doi.org/10.1214/ss/1009213726>
6. Deb, K., Agarwal, R.B.: Simulated Binary Crossover for Continuous Search Space. *Complex Systems* **9**, 115–148 (1995)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197 (April 2002). <https://doi.org/10.1109/4235.996017>
8. Dhurandhar, A., Pedapati, T., Balakrishnan, A., Chen, P., Shanmugam, K., Puri, R.: Model Agnostic Contrastive Explanations for Structured Data. *CoRR abs/1906.00117* (2019), <http://arxiv.org/abs/1906.00117>
9. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics* **24**(1), 44–65 (2015). <https://doi.org/10.1080/10618600.2014.907095>, <https://doi.org/10.1080/10618600.2014.907095>
10. Gower, J.C.: A General Coefficient of Similarity and Some of its Properties. *Biometrics* **27**(4), 857–871 (1971)
11. Grath, R.M., Costabello, L., Van, C.L., Sweeney, P., Kamiab, F., Shen, Z., Lécué, F.: Interpretable Credit Application Predictions With Counterfactual Explanations. *CoRR (abs/1811.05245)* (2018), <http://arxiv.org/abs/1811.05245>
12. Greenwell, B.M., Boehmke, B.C., McCarthy, A.J.: A simple and effective model-based variable importance measure. *arXiv preprint arXiv:1805.04755* (2018)
13. Hofmann, H.: German Credit Risk (2016), <https://www.kaggle.com/uciml/german-credit>, last accessed 25.01.2020
14. Hothorn, T., Zeileis, A.: Transformation Forests (2017)
15. Joshi, S., Koyejo, O., Vijitbenjaronk, W., Kim, B., Ghosh, J.: Towards Realistic Individual Recourse and Actionable Explanations in black-box decision making systems. *CoRR abs/1907.09615* (2019), <http://arxiv.org/abs/1907.09615>
16. Karimi, A., Barthe, G., Balle, B., Valera, I.: Model-Agnostic Counterfactual Explanations for Consequential Decisions. *CoRR (abs/1905.11190)* (2019), <http://arxiv.org/abs/1905.11190>
17. Kingma, D., Ba, J.: Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* (12 2014)
18. Laugel, T., Lesot, M.J., Marsala, C., Renard, X., Detyniecki, M.: Comparison-Based Inverse Classification for Interpretability in Machine Learning. *CoRR (abs/1712.08443)* (2017), <http://arxiv.org/abs/1712.08443>

19. Li, R., Emmerich, M.T., Eggermont, J., Bäck, T., Schütz, M., Dijkstra, J., Reiber, J.H.: Mixed Integer Evolution Strategies for Parameter Optimization. *Evolutionary Computation* **21**(1), 29–64 (2013)
20. Looveren, A.V., Klaise, J.: Interpretable Counterfactual Explanations Guided by Prototypes. *CoRR* **abs/1907.02584** (2019), <http://arxiv.org/abs/1907.02584>
21. López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L.P., Birattari, M., Stützle, T.: The irace Package: Iterated Racing for Automatic Algorithm Configuration. *Operations Research Perspectives* **3**, 43 – 58 (2016). <https://doi.org/https://doi.org/10.1016/j.orp.2016.09.002>, <http://www.sciencedirect.com/science/article/pii/S2214716015300270>, R package version 3.4.1
22. Lundberg, S.M., Lee, S.I.: A Unified Approach to Interpreting Model Predictions. In: *Advances in Neural Information Processing Systems*. pp. 4765–4774 (2017)
23. Molnar, C., Bischl, B., Casalicchio, G.: iml: An R package for Interpretable Machine Learning. *JOSS* **3**(26), 786 (2018). <https://doi.org/10.21105/joss.00786>, <http://joss.theoj.org/papers/10.21105/joss.00786>
24. Mothilal, R.K., Sharma, A., Tan, C.: Explaining Machine Learning Classifiers through Diverse Counterfactual explanations. *CoRR* (abs/1905.07697) (2019), <http://arxiv.org/abs/1905.07697>
25. Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., Bie, T.D., Flach, P.: FACE: Feasible and Actionable Counterfactual Explanations (2019)
26. Radulescu, A., López-Ibáñez, M., Stützle, T.: Automatically Improving the Anytime Behaviour of Multiobjective Evolutionary Algorithms. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) *Evolutionary Multi-Criterion Optimization*. pp. 825–840. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
27. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 1135–1144 (2016)
28. Russell, C.: Efficient Search for Diverse Coherent Explanations. *CoRR* (abs/1901.04909) (2019), <http://arxiv.org/abs/1901.04909>
29. Sharma, S., Henderson, J., Ghosh, J.: CERTIFAI: Counterfactual Explanations for Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence models. *CoRR* **abs/1905.07857** (2019), <http://arxiv.org/abs/1905.07857>
30. Su, J., Vargas, D.V., Sakurai, K.: One Pixel Attack for Fooling Deep Neural Networks. *IEEE Transactions on Evolutionary Computation* **23**, 828–841 (2017)
31. Syswerda, G.: Uniform Crossover in Genetic Algorithms. In: *Proceedings of the 3rd International Conference on Genetic Algorithms*. p. 29. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1989)
32. Tolomei, G., Silvestri, F., Haines, A., Lalmas, M.: Interpretable Predictions of Tree-based Ensembles via Actionable Feature Tweaking. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 465–474. KDD '17, ACM, New York, NY, USA (2017). <https://doi.org/10.1145/3097983.3098039>, <http://doi.acm.org/10.1145/3097983.3098039>
33. Ustun, B., Spangher, A., Liu, Y.: Actionable Recourse in Linear Classification. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. pp. 10–19. FAT* '19, ACM, New York, NY, USA (2019). <https://doi.org/10.1145/3287560.3287566>, <http://doi.acm.org/10.1145/3287560.3287566>

34. Vanschoren, J., van Rijn, J.N., Bischl, B., Torgo, L.: OpenML: Networked Science in Machine Learning. *SIGKDD Explorations* **15**(2), 49–60 (2013). <https://doi.org/10.1145/2641190.2641198>, <http://doi.acm.org/10.1145/2641190.2641198>
35. Wachter, S., Mittelstadt, B.D., Russell, C.: Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *CoRR* (abs/1711.00399) (2017), <http://arxiv.org/abs/1711.00399>
36. Wexler, J., Pushkarna, M., Bolukbasi, T., Wattenberg, M., Viégas, F.B., Wilson, J.: The What- If Tool: Interactive Probing of Machine Learning Models. *CoRR* **abs/1907.04135** (2019), <http://arxiv.org/abs/1907.04135>
37. White, A., d’Avila Garcez, A.: Measurable Counterfactual Local Explanations for Any Classifier (2019)
38. Zitzler, E., Thiele, L.: Multiobjective Optimization Using Evolutionary Algorithms — a Comparative Case Study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.P. (eds.) *Parallel Problem Solving from Nature — PPSN V*. pp. 292–301. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)

A Illustration of MOC's Benefits

This section illustrates the benefits of having a *diverse set* of counterfactuals using the diabetes dataset of the benchmark study (Section 6). We will compare the counterfactuals returned by MOC with the ones of Recourse [33] and Tweaking [32]. Due to space constraints, we only show the six counterfactuals of MOC with the highest HV contribution for both examples.

Table 3 contrasts MOC's counterfactuals with the three counterfactuals of Recourse for the prediction of observation 741. A logistic regression predicted a probability of having diabetes of 0.89 for this observation. The desired target is a prediction of less than 0.5, which indicates having no diabetes. All counterfactuals

Table 3. Counterfactuals and corresponding objective values of MOC and Recourse for the prediction of a logistic regression for observation 741 of the diabetes dataset. Shaded fields indicate values that differ from the value of observation 741 in brackets.

Feature (\mathbf{x}^*)	MOC ₁	MOC ₂	MOC ₃	MOC ₄	MOC ₅	MOC ₆	Recourse ₁	Recourse ₂	Recourse ₃
preg (11)	11.00	6.35	11.00	11.00	11.00	6.35	11.00	11.00	10.92
plas (120)	27.78	3.29	79.75	94.85	79.75	3.18	57.00	57.00	57.00
pres (80)	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00
skin (37)	37.00	37.00	37.00	37.00	37.00	37.00	37.00	36.81	37.00
insu (150)	150.00	150.00	17.13	150.00	40.61	150.00	150.00	150.00	150.00
mass (42.3)	42.30	42.30	29.17	15.36	29.17	42.30	42.30	42.30	42.30
pedi (0.78)	0.78	0.78	0.31	0.78	0.17	0.78	0.78	0.78	0.78
age (48)	48.00	41.61	44.42	48.00	48.00	48.00	28.36	28.36	28.36
o_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
o_2	0.06	0.12	0.10	0.07	0.10	0.11	0.08	0.08	0.08
o_3	1.00	3.00	5.00	2.00	4.00	2.00	2.00	3.00	3.00
o_4	0.10	0.05	0.03	0.07	0.04	0.07	0.09	0.09	0.09

of Recourse suggest the same reduction in *age* and plasma concentration (*plas*), with two counterfactuals additionally suggesting a minimal reduction in the number of pregnancies (*preg*) or the skin fold thickness (*skin*).⁶ Apart from that a reduction in *age* or *preg* is impossible, they do not offer many options for users. Instead, MOC returned a larger set of counterfactuals that provide more options for actionable user responses and are closer to the observed data than Recourse's counterfactuals (o_4). Counterfactual MOC₁ has overall lower objective values than all counterfactuals of Recourse. MOC₃ suggested changes to five features so that it is especially close to the nearest training data point (o_4).

Table 4 compares the set of counterfactuals found by MOC with the single counterfactual found by Tweaking for the prediction of observation 268. A random forest classifier predicted a probability of having diabetes of 0.62 for this observation. Again, the desired target is a prediction of less than 0.5. Tweaking suggested reducing the number of children and plasma glucose concentration (*plas*) while increasing the *age* so that the probability of diabetes decreases. This

⁶ By reclassifying *age* and *preg* as integers (instead of decimals), integer changes would be recommended by MOC, Recourse and Tweaking.

Table 4. Counterfactuals and corresponding objective values given by MOC and Tweaking for the prediction of a random forest for observation 268 of the cmc dataset. Shaded fields indicate values that differ from the value of observation 268 in brackets.

Feature (\mathbf{x}^*)	MOC ₁	MOC ₂	MOC ₃	MOC ₄	MOC ₅	MOC ₆	Tweaking ₁
preg (2)	2.00	2.00	2.00	2.00	2.00	2.00	1.53
plas (128)	121.50	90.21	126.83	128.00	88.44	120.64	119.71
pres (64)	64.00	64.00	64.00	64.00	64.00	64.00	64.00
skin (42)	42.00	42.00	42.00	42.00	42.00	42.00	42.00
insu (0)	0.00	0.00	0.00	0.00	0.00	90.93	0.00
mass (40)	40.00	40.00	40.00	40.00	40.00	40.00	40.00
pedi (1.1)	1.10	0.48	1.10	0.17	0.46	1.10	1.10
age (24)	24.00	24.00	24.00	24.00	25.85	24.00	28.29
o_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00
o_2	0.00	0.06	0.00	0.05	0.06	0.02	0.02
o_3	1.00	2.00	1.00	1.00	3.00	2.00	3.00
o_4	0.05	0.02	0.05	0.04	0.01	0.03	0.06

is contradictory and not plausible. In contrast, MOC’s counterfactuals suggest various strategies, e.g., only a decrease of *plas*, which is easier to realize. In addition, MOC₁, MOC₃ and MOC₆ dominate the counterfactual of Tweaking. Since five of six counterfactuals suggest changes to *plas*, the user may have more confidence that *plas* is an important lever to achieve the desired outcome.

B Iterated F-racing

We used iterated F-racing (irace) [21] to tune the parameters of MOC for binary classification. The parameters and considered ranges are given in Table 5. The number of generations was not part of the parameter set because it would be always tuned to the upper bound. Instead, the number of generations was determined after the other parameters were tuned with irace. Irace was initialized with a maximum budget of 3000 evaluations equal to 3000 runs of MOC. In every step, irace randomly selected one of 300 instances. Each instance consisted of a trained model, a randomly selected data point from the observed data as \mathbf{x}^* and a desired outcome. The desired target for each \mathbf{x}^* was the opposite of the predicted class:

$$Y' = \begin{cases}]0.5, 1] & \text{if } \hat{f}(\mathbf{x}^*) \leq 0.5 \\ [0, 0.5] & \text{else} \end{cases}.$$

The trained model was either logistic regression, random forest, xgboost, RBF support vector machine or a two-hidden-layer neural network. Each model estimated only the probability for one class. The models were trained on datasets obtained from the OpenML platform [34] (without the sampled \mathbf{x}^*) and are briefly described in Table 7. While these datasets were not used in the benchmark study (Section 6), the same preprocessing steps were conducted and the models were tuned with the same setup (see Section C for details).

In each step of irace, parameter configurations were evaluated by running MOC on the same selected instance. MOC stopped after evaluating 8000 candidates

Table 5. Parameter space investigated with iterated F-racing, as well as the resulting optimized configuration (*Result*).

Name	Description	Range	Result
M	Population size	[20, 100]	20
initialization	Initialization strategy	[Random, ICE curve]	ICE curve
conditional	Whether to use the conditional mutator	[TRUE, FALSE]	TRUE
p.rec	Probability a pair of parents is chosen to recombine	[0.3, 1]	0.57
p.rec.gen	Probability a feature is recombined	[0.3, 1]	0.85
p.rec.use.orig	Probability the indicator for feature changes is recombined	[0.3, 1]	0.88
p.mut	Probability a child is chosen to be mutated	[0.05, 0.8]	0.79
p.mut.gen	Probability one feature is mutated	[0.05, 0.8]	0.56
p.mut.use.orig	Probability indicator for a feature change is flipped	[0.05, 0.5]	0.32

with Eq. (1), which should be enough to ensure convergence of the HV in most cases. The integral of the first order spline approximation of the dominated HV over the evaluations was the performance criterion as recommended by [26]. The integral takes into account not only the extent but also the rate of convergence of the dominated HV. A Friedman test was used to discard less promising configurations. The first Friedman test was conducted after initial configurations were evaluated on 15 instances; afterward, the test was conducted after evaluating the remaining configurations on a single instance to accelerate the exclusion process. The best configuration returned is given in Table 5.

To obtain a default parameter for the number of generations for the benchmark study, we determined for the 300 instances after how many generations of the tuned MOC the dominated HV has not increased for 10 generations. We chose the maximum of 175 generations as a default for the study.

Table 6. Tuning search space per model. The hyperparameters *ntrees* and *nrounds* were log-transformed.

Model	Hyperparameter	Range
randomforest	ntrees	[0, 1000]
xgboost	nrounds	[0, 1000]
svm	cost	[0.01, 1]
logreg	lr	[0.0005, 0.1]
neuralnet	lr	[0.0005, 0.1]
	layer_size	[1, 6]

Table 7. Description of datasets for tuning with iterated F-racing. Legend: *Task*: OpenML task id; *Obs*: Number of rows; *Cont/Cat*: Number of continuous/categorical features.

Task Name	Obs	Cont	Cat
3818 tae	151	3	2
3917 kc1	2109	21	0
52945 breastTumor	277	0	6
3483 mammography	11183	6	0
3822 nursery	12960	0	8
3586 abalone	4177	7	1

C Model Hyperparameters for the Benchmark Study

We used random search (with 200 iterations for neural networks and 100 iterations for all other models) and 5-fold CV (with misclassification error as performance measure) to tune the hyperparameters of the models on the training data. The tuning search space was the same as for iterated F-racing and is shown in Table 6. Numerical features were scaled (standardization (Z-score) for random forest, min-max-scaling (0-1-range) for all other models) and categorical features were one-hot encoded. For neural network and logistic regression, ADAM [17] was the optimizer, the batch size was 32 with a 1/3 validation split and early stopping was conducted after 5 patience steps. Logistic regression needed these configurations because we constructed the model as a zero-hidden-layer neural network. For all other hyperparameters of the models, we chose the default values of the `mlr` [4] and `keras` [1] R packages. Table 8 shows the accuracies of the trained models using nested resampling (5-fold CV in outer and inner loop).

Table 8. Accuracy using nested resampling per benchmark dataset and model. Legend: *Name*: OpenML task name; *rf*: random forest. Logistic regression (*logreg*) was only trained on datasets with numerical or binary features.

Name	rf	xgboost	svm	logreg	neuralnet
boston	0.90	0.89	0.87	0.86	0.87
cmc	0.70	0.72	0.67		0.68
diabetes	0.76	0.74	0.75	0.63	0.68
ilpd	0.69	0.67	0.65	0.53	0.58
kc2	0.81	0.80	0.79	0.75	0.72
kr-vs-kp	0.99	0.99	0.97		0.99
no2	0.63	0.59	0.58	0.55	0.54
pc1	0.93	0.93	0.91	0.91	0.88
plasma_retinol	0.53	0.52	0.58		0.55
tic-tac-toe	0.99	0.99	0.98		0.97

D Control Parameters of Counterfactual Methods

For Tweaking [32], we only changed ϵ , a positive threshold that limits the tweaking of each feature. It was set to 0.5 because it obtained better results for the authors on their data example on Ad Quality in comparison to the default value 0.1. We used the R implementation of Tweaking on Github: <https://github.com/katokohaku/featureTweakR> (commit 6f3e614). For Recourse [33], we left all parameters at their default settings. We used the Python implementation of Recourse on Github: <https://github.com/ustunb/actionable-recourse> (commit aaaa8fa). For DiCE [24], we used the ‘DiverseCF’ version proposed by the authors [24] and left the control parameters at their defaults. We used the inverse mean absolute deviation for the feature weights. For datasets where the mean absolute deviation of a feature was zero, we set the feature weight to 10. We used the Python implementation of DiCE available on Github: <https://github.com/microsoft/DiCE> (commit fed9d27).

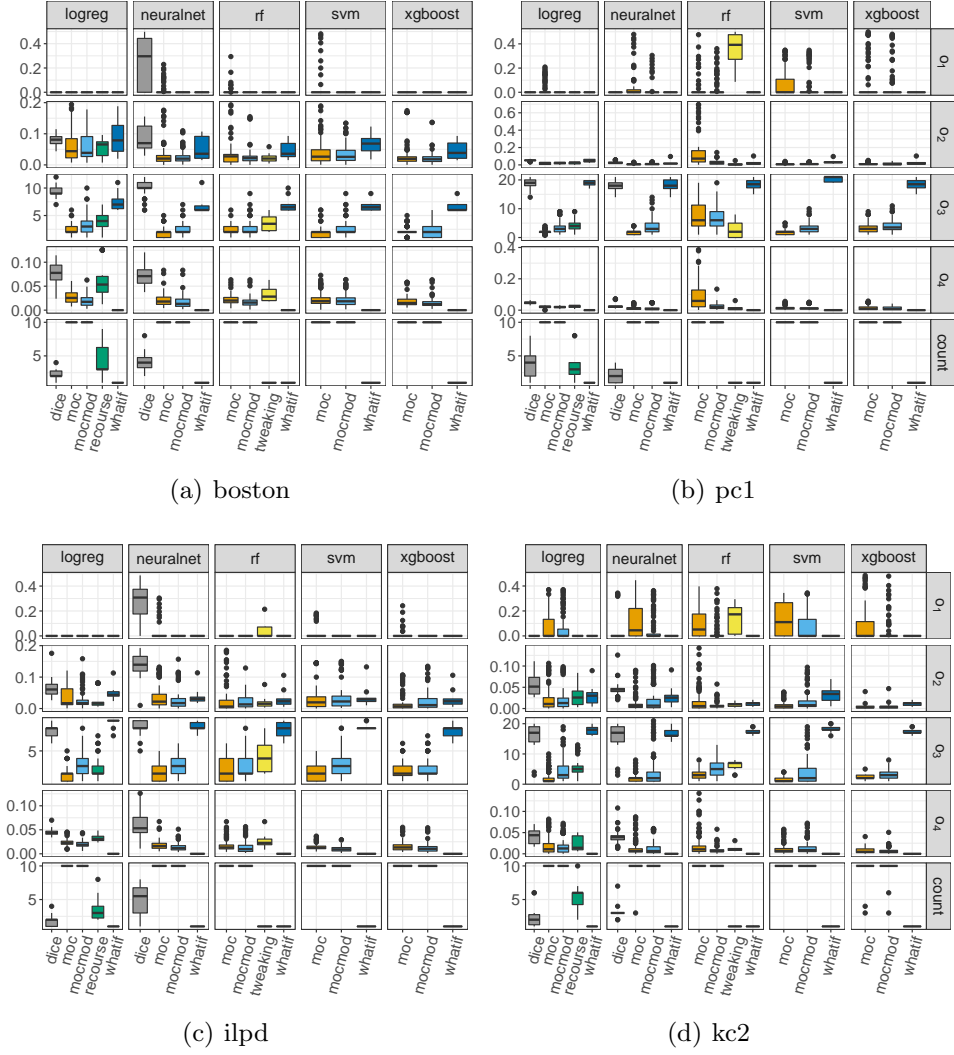


Fig. 4. Boxplots of the objective values and number of nondominated counterfactuals (*count*) per dataset and model for MOC with our proposed strategies for initialization and mutation (*mocmod*), MOC without these modifications, Whatif, DiCE, Recourse and Tweaking. Lower values are better except for *count*.

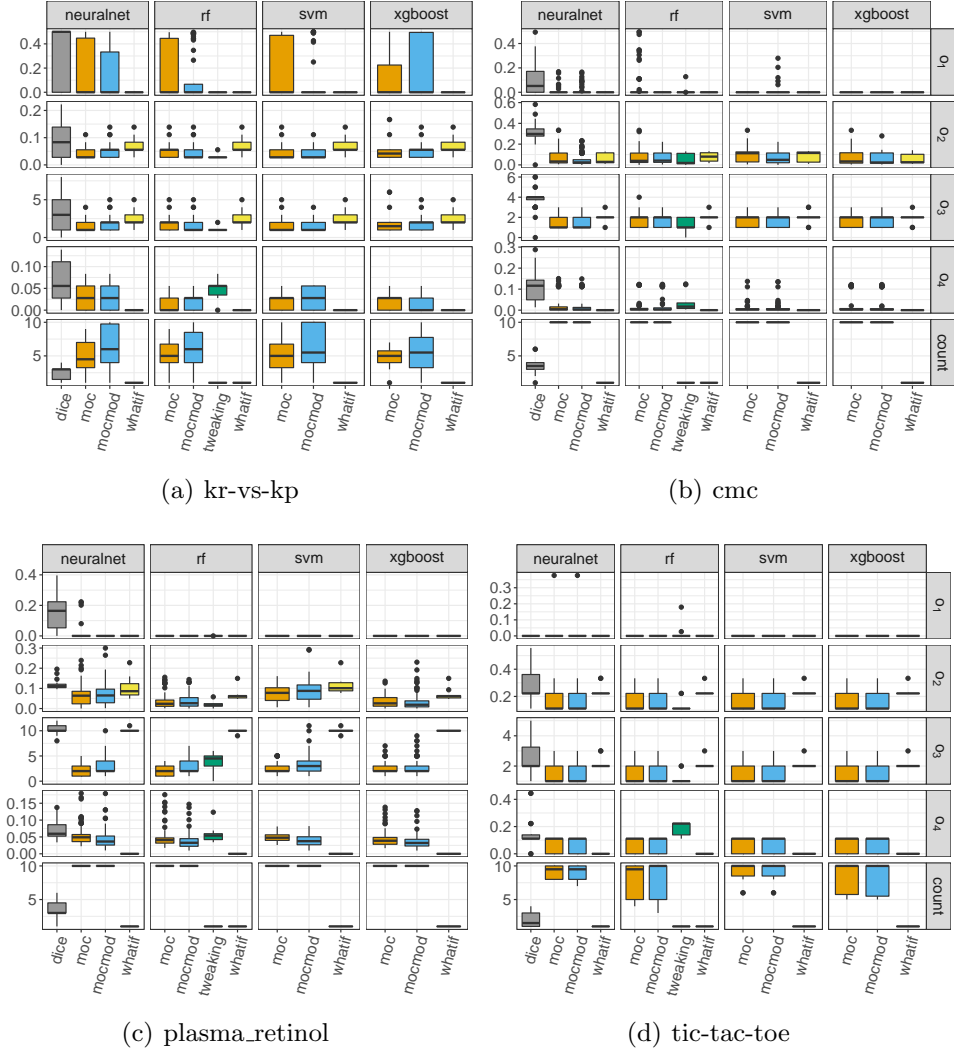


Fig. 5. Boxplots of the objective values and number of nondominated counterfactuals (*count*) per dataset and model for MOC with our proposed strategies for initialization and mutation (*mocmod*), MOC without these modifications, Whatif, DiCE, Recourse and Tweaking. Lower values are better except for *count*.

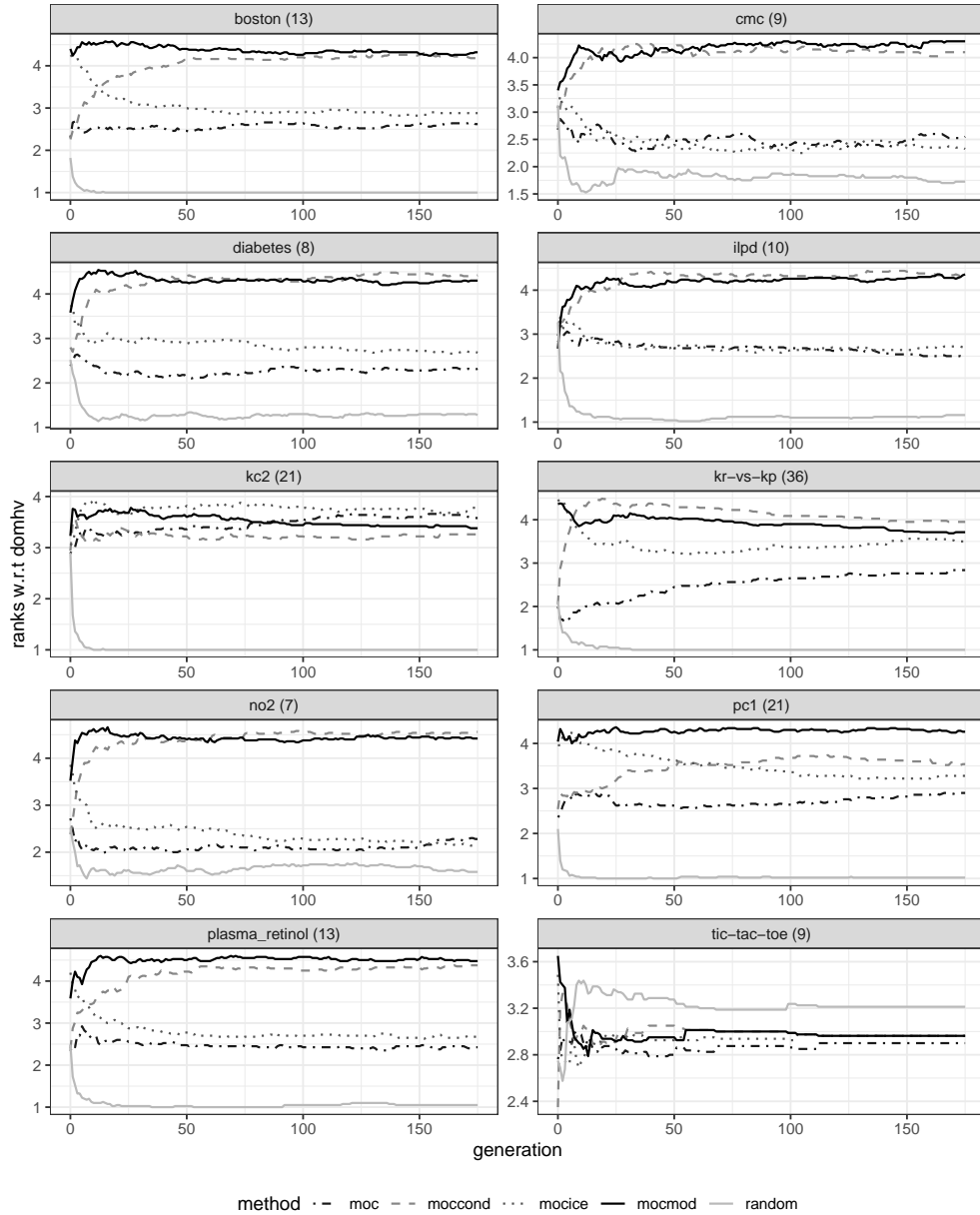


Fig. 6. Comparison of the ranks w.r.t. the dominated HV (*domhv*) per generation and per benchmark dataset averaged over all models. The numbers in parentheses indicate the number of features. For each approach, the population size of each generation was 20. Higher ranks are better. Legend: *moc*: MOC without modifications; *mocond*: MOC with the conditional mutator; *mocice*: MOC with the ICE curve variance initialization; *mocmod*: MOC with both modifications; *random*: random search.

Further References

- Apley, D. (2018). *ALEPlot: Accumulated Local Effects (ALE) Plots and Partial Dependence (PD) Plots*. R package version 1.1.
- Apley, D. W. and J. Zhu (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 82(4), 1059–1086.
- Archer, K. J. and R. V. Kimes (2008). Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis* 52(4), 2249–2260.
- Askira-Gelman, I. (1998). Knowledge discovery: comprehensibility of the results. In *Proceedings of the thirty-first Hawaii international conference on system sciences*, Volume 5, pp. 247–255. IEEE.
- Bair, E., R. Ohrbach, R. B. Fillingim, J. D. Greenspan, R. Dubner, L. Diatchenko, E. Helgeson, C. Knott, W. Maixner, and G. D. Slade (2013). Multivariable modeling of phenotypic risk factors for first-onset tmd: the opera prospective cohort study. *The Journal of Pain* 14(12), T102–T115.
- Boulesteix, A.-L., S. Janitza, J. Kruppa, and I. R. König (2012). Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2(6), 493–507.
- Boulesteix, A.-L., M. N. Wright, S. Hoffmann, and I. R. König (2020). Statistical learning approaches in the genetic epidemiology of complex diseases. *Human Genetics* 139(1), 73–84.
- Breiman, L. (2001). Random forests. *Machine learning* 45(1), 5–32.
- Breiman, L. et al. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science* 16(3), 199–231.
- Candès, E., Y. Fan, L. Janson, and J. Lv (2018). Panning for gold: ‘model-x’ knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 80(3), 551–577.
- Caruana, R., Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1721–1730.
- Chen, H., J. D. Janizek, S. Lundberg, and S.-I. Lee (2020). True to the model or true to the data? *arXiv preprint arXiv:2006.16234*.
- Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. IEEE.

- Dhurandhar, A., V. Iyengar, R. Luss, and K. Shanmugam (2017). TIP: typifying the interpretability of procedures. *arXiv preprint arXiv:1706.02952*.
- Dhurandhar, A., T. Pedapati, A. Balakrishnan, P. Chen, K. Shanmugam, and R. Puri (2019). Model Agnostic Contrastive Explanations for Structured Data. *CoRR abs/1906.00117*.
- Doshi-Velez, F. and B. Kim (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.
- Elish, M. C. and E. A. Watkins (2020). Repairing innovation: A study of integrating ai in clinical care.
- Esselman, P. C., R. J. Stevenson, F. Lupi, C. M. Riseng, and M. J. Wiley (2015). Landscape prediction and mapping of game fish biomass, an ecosystem service of michigan rivers. *North American Journal of Fisheries Management* 35(2), 302–320.
- Fisher, A., C. Rudin, and F. Dominici (2019). All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research* 20(177), 1–81.
- Friedler, S. A., C. D. Roy, C. Scheidegger, and D. Slack (2019). Assessing the local interpretability of machine learning models. *arXiv preprint arXiv:1902.03501*.
- Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, 1–67.
- Friedman, J. H., B. E. Popescu, et al. (2008). Predictive learning via rule ensembles. *Annals of Applied Statistics* 2(3), 916–954.
- Fürnkranz, J., D. Gamberger, and N. Lavrač (2012). *Foundations of rule learning*. Springer Science & Business Media.
- Goldstein, A., A. Kapelner, J. Bleich, and E. Pitkin (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics* 24(1), 44–65.
- Grath, R. M., L. Costabello, C. L. Van, P. Sweeney, F. Kamiab, Z. Shen, and F. Lécué (2018). Interpretable Credit Application Predictions With Counterfactual Explanations. *CoRR (abs/1811.05245)*.
- Greenwell, B. M., B. C. Boehmke, and A. J. McCarthy (2018). A simple and effective model-based variable importance measure. *arXiv preprint arXiv:1805.04755*.
- Groemping, U. (2020). Model-agnostic effects plots for interpreting machine learning models. *Reports in Mathematics, Physics and Chemistry, Department II, Beuth University of Applied Sciences Berlin. Report 1/2020*.
- Hastie, T. J. and R. J. Tibshirani (2017). *Generalized additive models*. Routledge.
- Hooker, G. (2004). Discovering additive structure in black box functions. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 575–580. ACM.
- Hooker, G. (2007). Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics* 16(3).

Further References

- Hooker, G. and L. Mentch (2019). Please stop permuting features: An explanation and alternatives. *arXiv preprint arXiv:1905.03151*.
- Huysmans, J., K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens (2011). An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems* 51(1), 141–154.
- Ishwaran, H., U. B. Kogalur, E. Z. Gorodeski, A. J. Minn, and M. S. Lauer (2010). High-dimensional variable selection for survival data. *Journal of the American Statistical Association* 105(489), 205–217.
- Ishwaran, H. and M. Lu (2019). Standard errors and confidence intervals for variable importance in random forest regression, classification, and survival. *Statistics in Medicine* 38(4), 558–582.
- Janitza, S., E. Celik, and A.-L. Boulesteix (2018). A computationally fast variable importance test for random forests for high-dimensional data. *Advances in Data Analysis and Classification* 12(4), 885–915.
- Johner, C., C. Molnar, A. Purde, A. Rad, C. Dierks, S. Bunk, and S. Piechottka (2021). Guideline for ai for medical products. <https://github.com/johner-institut/ai-guideline>.
- Joshi, S., O. Koyejo, W. Vijitbenjaronk, B. Kim, and J. Ghosh (2019). Towards Realistic Individual Recourse and Actionable Explanations in black-box decision making systems. *CoRR abs/1907.09615*.
- Karimi, A., G. Barthe, B. Balle, and I. Valera (2019). Model-Agnostic Counterfactual Explanations for Consequential Decisions. *CoRR (abs/1905.11190)*.
- Kim, B., O. Koyejo, R. Khanna, et al. (2016). Examples are not enough, learn to criticize! criticism for interpretability. In *NIPS*, pp. 2280–2288.
- Lakkaraju, H., E. Kamar, R. Caruana, and J. Leskovec (2017). Interpretable & explorable approximations of black box models. *arXiv preprint arXiv:1707.01154*.
- Lapuschkin, S., S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller (2019). Unmasking clever hans predictors and assessing what machines really learn. *Nature communications* 10(1), 1–8.
- Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue* 16(3), 31–57.
- Looveren, A. V. and J. Klaise (2019). Interpretable Counterfactual Explanations Guided by Prototypes. *CoRR abs/1907.02584*.
- Lou, Y., R. Caruana, J. Gehrke, and G. Hooker (2013). Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 623–631.
- Lundberg, S. M., G. G. Erion, and S.-I. Lee (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.
- Lundberg, S. M. and S.-I. Lee (2017). A unified approach to interpreting model predictions. In *NIPS*, Volume 30, pp. 4765–4774. Curran Associates, Inc.

- Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence* 267, 1–38.
- Molnar, C. (2019). *Interpretable Machine Learning*. <https://christophm.github.io/interpretable-ml-book/>.
- Molnar, C., G. König, B. Bischl, and G. Casalicchio (2020). Model-agnostic feature importance and effects with dependent features—a conditional subgroup approach. *arXiv preprint arXiv:2006.04628*.
- Nguyen, A., J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski (2017). Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4467–4477.
- Nguyen, A., A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune (2016). Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *arXiv preprint arXiv:1605.09304*.
- Obringer, R. and R. Nateghi (2018). Predicting urban reservoir levels using statistical learning techniques. *Scientific Reports* 8(1), 1–9.
- Olah, C., A. Mordvintsev, and L. Schubert (2017). Feature visualization. *Distill*. <https://distill.pub/2017/feature-visualization>.
- Paluszynska, A., P. Biecek, and Y. Jiang (2020). *randomForestExplainer: Explaining and Visualizing Random Forests in Terms of Variable Importance*. R package version 0.10.1.
- Pintelas, E., M. Liaskos, I. E. Livieris, S. Kotsiantis, and P. Pintelas (2020). Explainable machine learning framework for image classification problems: case study on glioma cancer prediction. *Journal of Imaging* 6(6), 37.
- Plumb, G., M. Al-Shedivat, E. Xing, and A. Talwalkar (2019). Regularizing black-box models for improved interpretability. *arXiv preprint arXiv:1902.06787*.
- Poursabzi-Sangdeh, F., D. G. Goldstein, J. M. Hofman, J. W. Vaughan, and H. Wallach (2018). Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810*.
- Poyiadzi, R., K. Sokol, R. Santos-Rodriguez, T. D. Bie, and P. Flach (2019). FACE: Feasible and Actionable Counterfactual Explanations.
- Prates, M. O., P. H. Avelar, and L. C. Lamb (2019). Assessing gender bias in machine translation: a case study with google translate. *Neural Computing and Applications*, 1–19.
- Ribeiro, M. T., S. Singh, and C. Guestrin (2016a). Model-agnostic interpretability of machine learning. *ICML WHI '16*.
- Ribeiro, M. T., S. Singh, and C. Guestrin (2016b). Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144. ACM.
- Roscher, R., B. Bohn, M. F. Duarte, and J. Garcke (2020). Explainable machine learning for scientific insights and discoveries. *IEEE Access* 8, 42200–42216.
- Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1(5), 206–215.

Further References

- Rudin, C., C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong (2021). Interpretable machine learning: Fundamental principles and 10 grand challenges. *arXiv preprint arXiv:2103.11251*.
- Rüping, S. et al. (2006). Learning interpretable models. *Univ. Dortmund*.
- Schielzeth, H. (2010). Simple means to improve the interpretability of regression coefficients. *Methods in Ecology and Evolution* 1(2), 103–113.
- Sendak, M. P., W. Ratliff, D. Sarro, E. Alderton, J. Futoma, M. Gao, M. Nichols, M. Revoir, F. Yashar, C. Miller, et al. (2020). Real-world integration of a sepsis deep learning technology into routine clinical care: implementation study. *JMIR medical informatics* 8(7), e15182.
- Sharma, S., J. Henderson, and J. Ghosh (2019). CERTIFAI: Counterfactual Explanations for Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence models. *CoRR abs/1905.07857*.
- Stachl, C., Q. Au, R. Schoedel, S. D. Gosling, G. M. Harari, D. Buschek, S. T. Völkel, T. Schuwerk, M. Oldemeier, T. Ullmann, H. Hussmann, B. Bischl, and M. Bühner (2020). Predicting personality from patterns of behavior collected with smartphones. *Proceedings of the National Academy of Sciences* 117(30), 17680–17687.
- Stiglic, G., P. Kocbek, N. Fijacko, M. Zitnik, K. Verbert, and L. Cilar (2020). Interpretability of machine learning-based prediction models in healthcare. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 10(5), e1379.
- Strobl, C., A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis (2008). Conditional variable importance for random forests. *BMC Bioinformatics* 9(1), 307.
- Strobl, C., A.-L. Boulesteix, A. Zeileis, and T. Hothorn (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics* 8(1), 25.
- Štrumbelj, E. and I. Kononenko (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems* 41(3), 647–665.
- Tomsett, R., D. Braines, D. Harborne, A. Preece, and S. Chakraborty (2018). Interpretable to whom? a role-based model for analyzing interpretable machine learning systems. *arXiv preprint arXiv:1806.07552*.
- Ustun, B. and C. Rudin (2016). Supersparse linear integer models for optimized medical scoring systems. *Machine Learning* 102(3), 349–391.
- Wachter, S., B. D. Mittelstadt, and C. Russell (2017). Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *CoRR* (abs/1711.00399).
- Watson, D. S. and M. N. Wright (2019). Testing conditional independence in supervised learning algorithms. *arXiv preprint arXiv:1901.09917*.
- Wei, P., Z. Lu, and J. Song (2015). Variable importance analysis: A comprehensive review. *Reliability Engineering & System Safety* 142, 399–432.
- Weisberg, M. (2012). *Simulation and similarity: Using models to understand the world*. Oxford University Press.

- White, A. and A. d'Avila Garcez (2019). Measurable Counterfactual Local Explanations for Any Classifier.
- Yang, H., C. Rudin, and M. Seltzer (2017). Scalable bayesian rule lists. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3921–3930. JMLR. org.
- Zhao, X., X. Yan, A. Yu, and P. Van Hentenryck (2020). Prediction and behavioral analysis of travel mode choice: A comparison of machine learning and logit models. *Travel behaviour and society* 20, 22–35.
- Zhou, Q., F. Liao, C. Mou, and P. Wang (2018). Measuring interpretability for different types of machine learning models. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 295–308.

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12. Juli 2011, § 8 Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

München, den 15.02.2022

Christoph Molnar

