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 are able to model complex relationships in the data. Nevertheless, trees and linear models are still preferred in areas such as life sciences and social sciences due to their simplicity and interpretability [14]. Some researchers have therefore developed model-agnostic interpretability tools for black box models, which often measure or visualize feature effects or feature importance [9, 11, 20].

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 on 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 [18] from coalitional game theory was first presented for classification in [20] and has been extended to regression and global analysis in [21]. 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 important 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 $\mathcal{X}_P = (\mathcal{X}_1 \times \ldots \times \mathcal{X}_p)$ with the feature index set $P = \{1, \ldots, p\}$ 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 that have been drawn i.i.d. from an unknown probability distribution $\mathcal{P}$ on the joint space

[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.

$\mathcal{X}_P \times \mathcal{Y}$. 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

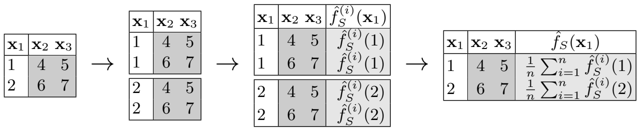
$$\widehat{GE}(\hat{f}, \mathcal{D}) = \tfrac{1}{n} \textstyle\sum_{i=1}^n L(\hat{f}(\mathbf{x}^{(i)}), y^{(i)}). \tag{1}$$

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:

$$\hat{f}_S(\mathbf{x}_S) = \tfrac{1}{n} \textstyle\sum_{i=1}^n \hat{f}_S^{(i)}(\mathbf{x}_S) = \tfrac{1}{n} \textstyle\sum_{i=1}^n \hat{f}(\mathbf{x}_S, \mathbf{x}_C^{(i)}). \tag{2}$$

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 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.



$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$
1	4	5
2	6	7

→

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$
1	4	5
1	6	7
2	4	5
2	6	7

→

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\hat{f}_S^{(i)}(\mathbf{x}_1)$
1	4	5	$\hat{f}_S^{(i)}(1)$
1	6	7	$\hat{f}_S^{(i)}(1)$
2	4	5	$\hat{f}_S^{(i)}(2)$
2	6	7	$\hat{f}_S^{(i)}(2)$

→

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\hat{f}_S(\mathbf{x}_1)$
1	4	5	$\frac{1}{n}\sum_{i=1}^n \hat{f}_S^{(i)}(1)$
2	6	7	$\frac{1}{n}\sum_{i=1}^n \hat{f}_S^{(i)}(2)$

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 extend the prediction of one observation changes when the value of a 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 individual observation indexed by $i \in \{1, \ldots, n\}$.

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 $v : 2^P \to \mathbb{R}$ maps all 2^p possible coalitions to their payouts. The Shapley value [18] 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 [20]. 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

$$\varDelta_j(\mathbf{x}_S) = v(\mathbf{x}_{S \cup \{j\}}) - v(\mathbf{x}_S) = f_{S \cup \{j\}}(\mathbf{x}_{S \cup \{j\}}) - f_S(\mathbf{x}_S).$$

Let $\varPi$ be the set of all possible permutations over the index set P. For a permutation $\pi \in \varPi$, 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 $\pi = \{2, 3, \mathbf{4}, 1\}$, 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

$$\begin{aligned} \hat{\phi}_j(\mathbf{x}) &= \tfrac{1}{p!} \textstyle\sum_{\pi \in \varPi} \hat{\varDelta}_j(\mathbf{x}_{B_j(\pi)}) \\ &= \tfrac{1}{p!} \textstyle\sum_{\pi \in \varPi} \hat{f}_{B_j(\pi) \cup \{j\}}(\mathbf{x}_{B_j(\pi) \cup \{j\}}) - \hat{f}_{B_j(\pi)}(\mathbf{x}_{B_j(\pi)}) \\ &= \tfrac{1}{p! \cdot n} \textstyle\sum_{\pi \in \varPi} \sum_{i=1}^n \hat{f}^{(i)}_{B_j(\pi) \cup \{j\}}(\mathbf{x}_{B_j(\pi) \cup \{j\}}) - \hat{f}^{(i)}_{B_j(\pi)}(\mathbf{x}_{B_j(\pi)}), \end{aligned}$$

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 [21]. 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 individual 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 individual 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

$$PFI_S = \mathbb{E}(L(\hat{f}(\tilde{X}_S, X_C), Y)) - \mathbb{E}(L(\hat{f}(X), Y)) \tag{4}$$

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 are 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

$$\widehat{GE}_C(\hat{f}, \mathcal{D}) = \tfrac{1}{n}\textstyle\sum_{i=1}^n \tfrac{1}{n}\sum_{k=1}^n L(\hat{f}(\mathbf{x}_S^{(k)}, \mathbf{x}_C^{(i)}), y^{(i)}). \tag{5}$$

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 [17] 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 $\{\boldsymbol{\tau}_1, \dots, \boldsymbol{\tau}_{n!}\}$ 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

$$\widehat{GE}_{C,\text{perm}}(\hat{f}, \mathcal{D}) = \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \tfrac{1}{n!} \sum_{k=1}^{n!} L(\hat{f}(\mathbf{x}_S^{(\tau_k^{(i)})}, \mathbf{x}_C^{(i)}), y^{(i)}).$$

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

$$\widehat{GE}_{C,\text{approx}}(\hat{f}, \mathcal{D}) = \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \tfrac{1}{m} \sum_{k=1}^{m} L(\hat{f}(\mathbf{x}_S^{(\tau_k^{(i)})}, \mathbf{x}_C^{(i)}), y^{(i)}). \tag{6}$$

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

$$\widehat{PFI}_S = \tfrac{1}{n^2} \textstyle\sum_{i=1}^{n} \sum_{k=1}^{n} \left(L(\hat{f}(\mathbf{x}_S^{(k)}, \mathbf{x}_C^{(i)}), y^{(i)}) - L(\hat{f}(\mathbf{x}^{(i)}), y^{(i)}) \right). \tag{7}$$

If we use Eq. (6) rather than Eq. (5), we obtain the approximation

$$\widehat{PFI}_{S,\text{approx}} = \tfrac{1}{n \cdot m} \textstyle\sum_{i=1}^{n} \sum_{k=1}^{m} \left(L(\hat{f}(\mathbf{x}_S^{(\tau_k^{(i)})}, \mathbf{x}_C^{(i)}), y^{(i)}) - L(\hat{f}(\mathbf{x}^{(i)}), y^{(i)}) \right). \tag{8}$$

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

[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.

of 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):

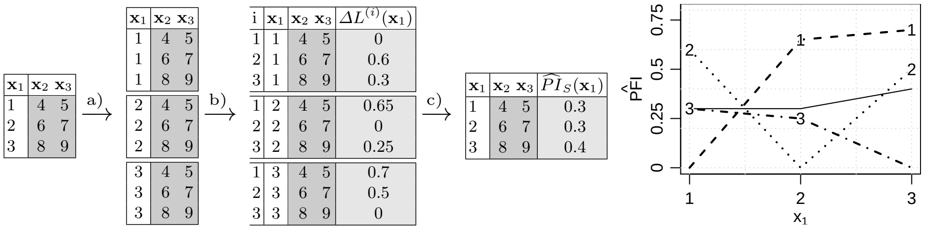
$$\widehat{PFI}_S = \frac{1}{n} \sum_{i=1}^{n} \widehat{PFI}_S^{(i)} = \frac{1}{n} \sum_{k=1}^{n} \widehat{PI}_S(\mathbf{x}_S^{(k)}). \tag{9}$$

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 *S*hapley *F*eature *IMP*ortance (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.



$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$
1	4	5
2	6	7
3	8	9

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$
1	4	5
1	6	7
1	8	9
2	4	5
2	6	7
2	8	9
3	4	5
3	6	7
3	8	9

i	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\Delta L^{(i)}(\mathbf{x}_1)$
1	1	4	5	0
2	1	6	7	0.6
3	1	8	9	0.3
1	2	4	5	0.65
2	2	6	7	0
3	2	8	9	0.25
1	3	4	5	0.7
2	3	6	7	0.5
3	3	8	9	0

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\widehat{PI}_S(\mathbf{x}_1)$
1	4	5	0.3
2	6	7	0.3
3	8	9	0.4

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 absence of features. This is reasonable in the context of feature selection. However, as we are measuring 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:

$$v_{GE}(S) = \widehat{GE}_S(\hat{f}, \mathcal{D}) - \widehat{GE}_{\emptyset}(\hat{f}, \mathcal{D}). \qquad (10)$$

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

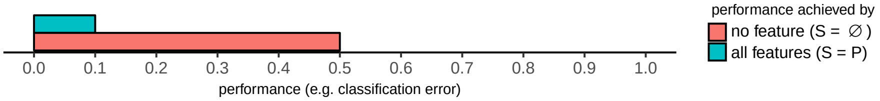


Fig. 3. Illustration of the difference in model performance that we want to fairly distribute among the features. The model performance (e.g. classification error) is 0.1 when using all features (green bar) and 0.5 when ignoring all features (red bar). Our goal is to fairly distribute the resulting performance 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

$$\begin{aligned} \hat{\phi}_j(v_{GE}) &= \tfrac{1}{p!} \textstyle\sum_{\pi \in \Pi} \Delta_j(B_j(\pi)) \\ &= \tfrac{1}{p!} \textstyle\sum_{\pi \in \Pi} \widehat{GE}_{B_j(\pi) \cup \{j\}}(\hat{f}, \mathcal{D}) - \widehat{GE}_{B_j(\pi)}(\hat{f}, \mathcal{D}), \end{aligned} \tag{11}$$

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: $m_{\text{feat}}$, $m_{\text{obs}}$, $\hat{f}$, $L$, $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$
forall $k \in \{1, \ldots, m_{\text{feat}}\}$ do
    choose a random permutation of the feature indices $\pi \in \Pi$.
    set $S = B_j(\pi)$ containing features that won't be permuted.
    set $\widehat{GE}_{S,\text{perm}} = 0$ and $\widehat{GE}_{S \cup \{j\},\text{perm}} = 0$.
    forall $l \in \{1, \ldots, m_{\text{obs}}\}$ do
        choose a random permutation of observation indices $\boldsymbol{\tau} \in \{\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_{n!}\}$.
        measure performance by permuting features w.r.t. $\boldsymbol{\tau} = (\tau^{(1)}, \ldots, \tau^{(n)})$:
          $\widehat{GE}_{S,\text{perm}} = \widehat{GE}_{S,\text{perm}} + \frac{1}{n} \sum_{i=1}^n L(\hat{f}(\mathbf{x}_S^{(i)}, \mathbf{x}_C^{(\tau^{(i)})}), y^{(i)}))$
          $\widehat{GE}_{S \cup \{j\},\text{perm}} = \widehat{GE}_{S \cup \{j\},\text{perm}} + \frac{1}{n} \sum_{i=1}^n L(\hat{f}(\mathbf{x}_{S \cup \{j\}}^{(i)}, \mathbf{x}_{C \setminus \{j\}}^{(\tau^{(i)})}), y^{(i)}))$
    compute marginal contribution for feature $j$ in iteration $k$:
      $\varDelta_j^{(k)}(S) = \frac{1}{m_{\text{obs}}} \cdot (\widehat{GE}_{S \cup \{j\},\text{perm}} - \widehat{GE}_{S,\text{perm}})$
return $\hat{\phi}_j = \frac{1}{m_{\text{feat}}} \sum_{k=1}^{m_{\text{feat}}} \varDelta_j^{(k)}(S)$
```

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 to compare 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/giuseppec/featureImportance.

$$X_1, X_2 \overset{i.i.d}{\sim} \mathcal{N}(0,1), X_3 \sim \mathcal{B}(1,0.5), \epsilon \sim \mathcal{N}(0,0.5).$$

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 individual observation allows to calculate 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 case of a random forest with correlated features was discussed in [19]. In Figure 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:

$$Y = X_1 + X_2 + X_3 + X_1 \cdot X_2 + \epsilon, \qquad X_1, X_2, X_3 \overset{i.i.d}{\sim} \mathcal{N}(0,1), \epsilon \sim \mathcal{N}(0,0.5).$$

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

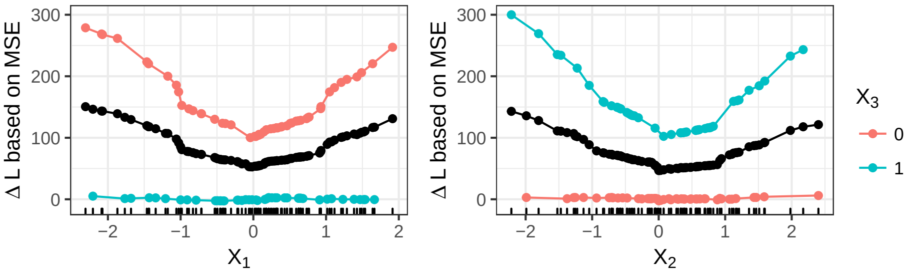


Fig. 4. PI curves of X_1 and X_2 calculated using all observations (black line) and conditional on $X_3 = 0$ (red line) and $X_3 = 1$ (green line). The points plotted on the lines refer to the observed feature values that were used as grid points to produce the corresponding PI curves as described in Algorithm 1.

four 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, a higher importance was calculated for the interacting features, indicating that they 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 therefore 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 [22] with data set ID 531. The data set contains 13 features that may affect the median home price of 506 metropolitan areas of

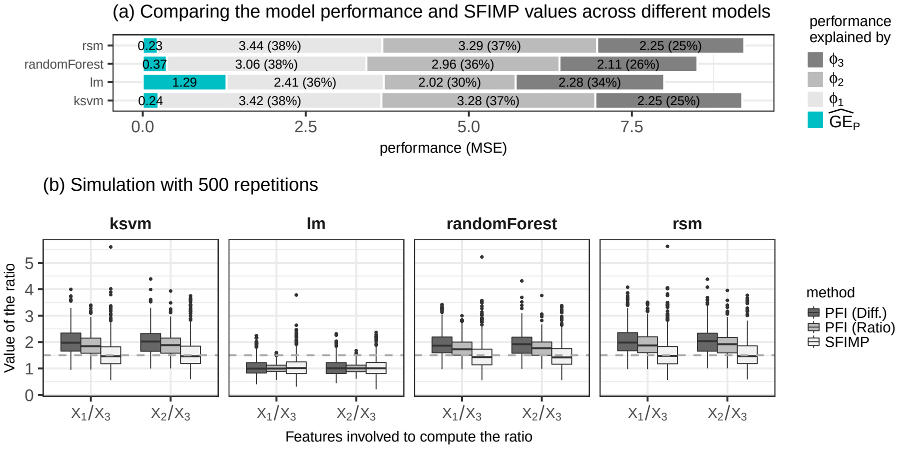


Fig. 5. Panel (a) shows the results of a single run, consisting of sampling test data and computing the importance on the previously fitted models. The first numbers on the left refer to the model performance (MSE) using all features. The other numbers are the SFIMP values which sum up to the total explainable 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.

Boston. We use 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.

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 ≥ 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 ≥ 10 makes the LSTAT feature less important. Note that omitting observations changes 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 how features contribute to model predictions or how they affect the model performance. The availability of model-agnostic visualization methods can simplify this task tremendously. In terms of 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 [19] on the subsetted data which may reveal interactions. Another interesting aspect, which we leave for future work, is aggregating the local feature importances of an observation (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

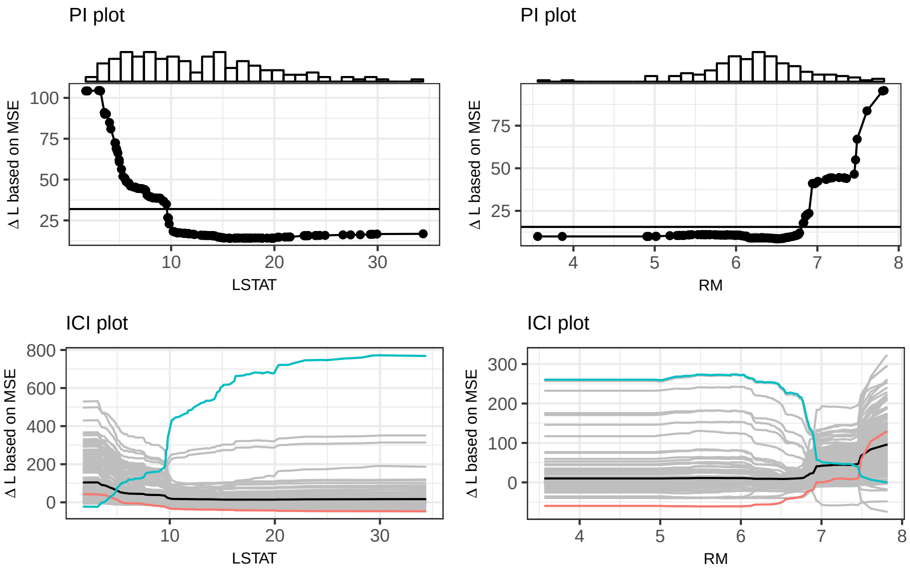


Fig. 6. PI and ICI plots for a random forest and the two most important features of the Boston housing data (LSTAT and RM). The horizontal lines in the PI plots represent the value of the global PFI (i.e. the integral of the PI curve). Marginal distribution histograms for features are added to the PI margins. The ICI curve with the largest integral is highlighted in green and the curve with the smallest integral in red.

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 the variability introduced by 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] Serfling, R.J.: Approximation Theorems of Mathematical Statistics, vol. 162. John Wiley & Sons (2009)

[18] Shapley, L.S.: A value for n-person games. Contributions to the Theory of Games **2**(28), 307–317 (1953)

[19] Strobl, C., Boulesteix, A.L., Kneib, T., Augustin, T., Zeileis, A.: Conditional variable importance for random forests. BMC Bioinf. **9**, 307 (2008)

[20] Štrumbelj, E., Kononenko, I., Wrobel, S.: An efficient explanation of individual classifications using game theory. J. Mach. Learn. Res. **11**(Jan), 1–18 (2010)

[21] 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)

[22] 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)