Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.6073

Model interpretation using improved local regression with variable importance

Gilson Yuuji Shimizu [**Renato Archer Information Technology Center | Centro Universitário de Jaguariúna | Centro Universitário Max Planck** | ***gyshimizu@cti.gov.br***]
Rafael Izbicki [**Federal University of São Carlos** | ***rafaelizbicki@gmail.com***]
Fernando Rezende Zagatti [**Renato Archer Information Technology Center | Centro Universitário de Jaguariúna | Centro Universitário Max Planck** | ***fzagatti@cti.gov.br***]
Filipe Loyola Lopes [**Renato Archer Information Technology Center | Centro Universitário de Jaguariúna | Centro Universitário Max Planck** | ***flopes@cti.gov.br***]
André Gomes Regino [**Renato Archer Information Technology Center | Centro Universitário de Jaguariúna | Centro Universitário Max Planck** | ***aregino@cti.gov.br***]
Rodrigo Bonacin [**Renato Archer Information Technology Center | Centro Universitário de Jaguariúna | Centro Universitário Max Planck** | ***rbonacin@cti.gov.br***]
Andre C. P. L. F. de Carvalho [**University of São Paulo** | ***andre@icmc.usp.br***]

Computing Methodologies Division, Renato Archer Information Technology Center, Dom Pedro I Highway (SP-65), Km 143,6 - Chácaras Campos dos Amarais, Campinas, SP, 13069-901, Brazil.

Received: 29 May 2025 • **Accepted:** 18 August 2026 • **Published:** 17 September 2026

Abstract. One of the main limitations for the trust in the use of machine learning models is the understanding of how they produce their predictions. This limitation is related to an increasing demand for transparency in decision-making. Interpretability refers to how well a user can understand the model's decision-making process without necessarily knowing its internal mechanisms. Several interpretability methods have emerged in the last years, such as LIME and Shapley values, which are valued for their flexibility, intuitive appeal, and strong theoretical foundations. While these methods have significantly contributed to better model transparency, they face challenges in several model deployment scenarios, such as handling irrelevant features or maintaining stability under small data perturbations. This article introduces two new agnostic interpretability methods, namely VarImp and SupClus, which overcome these issues by using local regressions fits with a weighted distance that takes into account variable importance. Whereas VarImp generates interpretations for each instance and can be applied to datasets with more complex relationships, SupClus interprets data clusters of instances with similar interpretations and can be applied to simpler datasets where data clusters can be found. In this paper, we compare these proposed methods with state-of-the-art methods and show that the proposed methods generate either equal or better interpretations, according to several proposed quantitative metrics (mean square error of coefficients, effect correlation, prediction correlation and ICE effect correlation), particularly in high-dimensional problems with irrelevant features and when the relationship between features and target is non-linear.

Keywords: Explainable AI (XAI), Interpretable Machine Learning, Local interpretation, Model Agnostic

1 Introduction

The increasing popularity of machine learning (ML), together with the growing concern regarding reliability, transparency model interpretability have demanded interpretations for their decision-making processes, mainly model inner works and model predictions. Especially in areas that involve risk, good predictions are not sufficient, since professionals must understand the reason for a decision to ensure trust, fairness, and transparency. Some models such as decision trees and linear models are naturally interpretable; however, the models induced by the ML algorithms with known high predictive power are considered black boxes. The lack of interpretability prevents these models from being applied in a wider range of areas.

Model interpretability concerns how much the human being can understand the cause for a decision or predict the outcome of a model [Miller, 2019; Kim *et al*., 2016; Molnar, 2020]. Linear models are highly interpretable, since the regression coefficients indicate whether the relationship between a feature and a target is positive or negative. It is important to distinguish interpretability from explainability, another important aspect of transparency. According to Barredo Arrieta *et al*. [2020], model explainability refers to how well a user can understand the internal mechanisms of a model, while model interpretability relates to how well a user can understand the model decision-making process without necessarily knowing its internal work; thus, interpretability is more closely associated with model transparency.

Interpretability methods can explain all instances globally or only one instance locally [Molnar, 2020; Hakkoum *et al*., 2024], and can be either specific to a ML algorithm, such as the Integrated Gradients method [Sundararajan *et al*., 2017], or agnostic. Some ML algorithms are intrinsically interpretable (see, e.g., Coscrato *et al*. [2023]; Agarwal *et al*. [2021] and references therein), whereas others require a post hoc method of interpretability (see, e.g., Hechtlinger [2016]; Koh and Liang [2017]; Lundberg and Lee [2017];

Ribeiro *et al*. [2016]; Botari *et al*. [2020]; Saeed and Omlin [2023] for a review). The methodologies of local and agnostic interpretability that have stood out are based on a penalized local linear regression [Ribeiro *et al*., 2016; Botari *et al*., 2020; Molnar *et al*., 2018]. The key idea is that the relationship between features and target is linear in a small region around an instance. To achieve this, the methodology generates perturbed samples by introducing small variations to the feature values of the instance being explained. These perturbed samples are then utilized to fit a weighted local linear regression model, where samples closer to the original instance are assigned higher weights. This weighting ensures that the linear approximation primarily captures the local behavior of the model around the selected instance, enabling a more precise and interpretable explanation of its predictions. Another prominent approach is based on the Shapley values of cooperative game theory [Shapley, 2016; Štrumbelj and Kononenko, 2014; Lundberg and Lee, 2017]. In this approach, features are seen as players and a prediction is the end result of a game; therefore, Shapley values represent the contribution of each feature for a given prediction. Some authors propose methods that interpret sets of instances with similar interpretations as clusters in a partition created via an unsupervised algorithm or as a decision tree [Zafar and Khan, 2021; Hall *et al*., 2017; Hu *et al*., 2018]. In this approach, when clusters are used, interpretable linear models are further fitted for each cluster.

Several interpretability methods have been developed in recent years, each contributing valuable insights into model behavior. Approaches based on perturbed samples, such as LIME and Shapley values, are particularly popular due to their simplicity and intuitive appeal. However, a known challenge is that these methods may sometimes produce less reliable interpretations, as the perturbed samples used do not always reflect the true data distribution [Alvarez-Melis and Jaakkola, 2018; Slack *et al*., 2020]. Methods that use penalization (e.g., LASSO [Tibshirani, 1996]) may also be unreliable, due to their focus on prediction rather than on the inference of regression coefficients, thus requiring further inferential analyses [Lee *et al*., 2016]. Another negative aspect is that the Euclidean distance used in local regressions may not be suitable, since relevant and irrelevant variables contribute equally to the calculation of distances. Finally, although clustering-based methods have the advantage of generating more stable interpretations, they have difficulties to effectively group instances with similar interpretations, since traditional clustering algorithms do not use the target variable. Even decision trees are often not suitable, since they do not take into account the sign of the relationship between variables and target.

Contributions. This article proposes two new interpretability methods, namely VarImp (Variable Importance) and SupClus (Supervised Cluster), to overcome the deficiencies of reliability and quality of interpretations of the current methods. The key idea behind VarImp is to use local variable importance as weights when calculating distances in a local linear regression; this means that variables more relevant to a specific instance receive higher weights in the distance calculation. In this context, SupClus uses the coefficients obtained by VarImp to group similar instances and interpret the resulting clusters through linear models. As a result, whereas VarImp provides instance-level interpretations, SupClus focuses on interpreting clusters to which instances belong. Consequently, VarImp is better suited for complex datasets, while SupClus is more appropriate for simpler datasets where combinations of linear models are sufficient to interpret the original model. The methods proposed in this study are model-agnostic, thus they can be used to explain predictions generated by any model. These methods were compared with other model-agnostic approaches, namely Shapley values, LIME, and IML. Non–model-agnostic methods such as Integrated Gradients, TreeSHAP, and DeepSHAP, as well as other approaches based on approximations of Shapley values, were not considered in this study. An R package with the implementation of VarImp and SupClus is available on Github[1] to allow readers to use the proposed methods.

The article is organized as follows: Sections 2 and 3 introduce VarImp and SupClus, respectively; Section 4 describes the experiments carried out on artificial and real data and presents the measures used for the evaluation of interpretability methods; Section 5 presents and analyzes the experimental results; finally, Section 6 draws concluding remarks.

Notation. For simplicity of notation, we assume our i.i.d. (independent and identically distributed) dataset is split into two parts of the same size n: the set

$$\mathbb{D}' = \{(\mathbf{x}'_1, y'_1), \ldots, (\mathbf{x}'_n, y'_n)\}$$

is used to train the original model (to be interpreted), which we denote by f, while the set

$$\mathbb{D} = \{(\mathbf{x}_1, f(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, f(\mathbf{x}_n))\}$$

is employed to train the local model interpreter. In a regression task, f represents the regression prediction, while in a classification task, f represents the estimated probabilities of the chosen label. We denote the local model interpreter as g. Finally, we denote by $x_{i,j}$ the value of the j-th feature for an instance i, with $j = 1, \ldots, d$, where d represents the total number of features.

2 VarImp method

As LIME [Ribeiro *et al*., 2016], VarImp is based on the assumption that predictions of the original model can be locally approximated by a linear model. However, rather than computing weights needed to implement such model using plain Euclidean (or related) distance, we define a matrix that uses importance weights. Moreover, since LASSO tends to select a single variable from a group of correlated predictors, leading to unstable estimates [Lee *et al*., 2016], a Forward Stepwise (FS) variable selection algorithm [James *et al*., 2013] was adopted, with the possibility of choosing a maximum number of features $M \leq d$. The FS method begins with the null model, which has no covariates. At each iteration of the FS algorithm, a variable is added to the

[1] Available at: `https://github.com/gilsonshimizu/interpretability`

existing model based on a predetermined criterion, such as reducing the residual sum of squares. The process concludes when no further variable enhances the model according to the specified criterion.

It is important to note that in penalized methods the regularization parameter is typically tuned with a focus on predictive performance rather than on the interpretability of the coefficients. Moreover, penalization introduces shrinkage bias into the estimates, which may hinder the direct interpretation of coefficients in local surrogate models. In contrast, FS allows explicit control over the maximum number of variables selected and estimates the coefficients by ordinary least squares on the chosen subset, without additional penalization. As a result, the coefficients obtained through FS more directly reflect the contribution of the selected variables in the neighborhood of the observation being explained, thereby favoring local explanations that are more closely aligned with the interpretability objective of the method.

The details are as follows. For a specific instance i with a feature vector $\mathbf{x}_i$, we fit the following local model:

$$g(\mathbf{x}_i) = \hat{\beta}_{i,0} + \sum_{j=1}^{d} \hat{\beta}_{i,j} x_{i,j},$$

where the coefficients $\hat{\beta}_{i,0}, \hat{\beta}_{i,1}, \ldots, \hat{\beta}_{i,d}$ are estimated by weighted least squares through

$$\underset{\beta_{i,0},\beta_{i,1},\ldots,\beta_{i,d}}{\arg\min} \sum_{l=1}^{n} w_l(\mathbf{x_i}) \left(f(\mathbf{x_l}) - \beta_{i,0} - \sum_{j=1}^{d} \beta_{i,j} x_{l,j} \right)^2.$$

As in LIME, the weights w_l are defined using a Gaussian kernel, i.e., $w_l(\mathbf{x_i}) = \frac{K(\mathbf{x_i},\mathbf{x_l})}{\sum_{l=1}^{n} K(\mathbf{x_i},\mathbf{x_l})}$ with

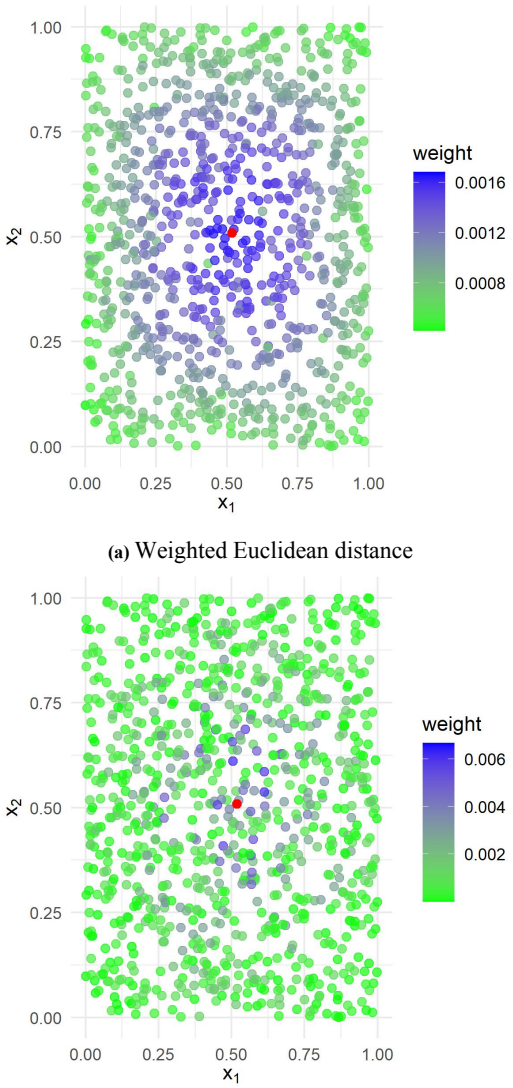
$$K(\mathbf{x_i}, \mathbf{x_l}) = \left(\sqrt{2\pi h^2}\right)^{-1} \exp\{-\frac{\mathrm{d}^2(\mathbf{x_i}, \mathbf{x_l})}{2h^2}\},$$

$h > 0$. However, to better deal with irrelevant variables, and by considering that some variables are more relevant than others, a weighted Euclidean distance

$$\mathrm{d}^2(\mathbf{x_i}, \mathbf{x_l}) = \sum_{j=1}^{d} v_{i,j} \left(x_{i,j} - x_{l,j}\right)^2,$$

where $v_{i,j}$ represents the weight of variable j for instance i, was used, resulting in more stable weights w_l and more reliable local coefficients. Figure 1 illustrates the advantage of using weighted Euclidean distance (versus the unweighted one) on an artificial dataset with two relevant and two irrelevant variables.

Weights $v_{i,j}$ are calculated using the *local variable importance* provided by the Random Forest algorithm [Breiman, 2001] applied in $\mathbb{D}$. That is, for each tree, we consider the sample not used in its construction (out-of-bag data). The values of the variable j are randomly permuted in each sample. Therefore, only trees that used instance i were considered and the differences in mean square error with and without permutation were calculated. Thus, $v_{i,j} = \frac{I_{i,j}}{\sum_j I_{i,j}}$, with $I_{i,j} = \mathrm{MSE}_{\text{with permutation}} - \mathrm{MSE}$.



(a) Weighted Euclidean distance

(b) Standard Euclidean distance

Figure 1. Comparison of weights w by weighted Euclidean distance and standard Euclidean distance around an instance (red) for artificial dataset: $Y = 1+5X_1-5X_2+0X_3+0X_4+\epsilon$, $X_j \sim \mathrm{Uniform}(0,1)$, $\epsilon \sim \mathrm{Normal}(0,1)$ and $d = 4$. The weighted Euclidean distance (left graphic) deals better with irrelevant features.

Instead of an analysis of coefficients $\beta_{i,j}$, an investigation on the effects of $\phi_{i,j} = \beta_{i,j} x_{i,j}$ may be more intuitive Molnar *et al.* [2018]. Since $g(\mathbf{x}_i) = \sum_{j=0}^{d} \hat{\phi}_{i,j}$, each $\hat{\phi}_{i,j}$ is interpreted as the contribution of variable j to prediction $g(\mathbf{x}_i)$. Thus, we use $\phi_{i,j}$ as a measure of importance.

The pseudocode of the VarImp method is described in Algorithm 1.

3 SupClus method

SupClus builds on VarImp, and was designed based on the assumption that some datasets may not need a different interpretation for each instance. Thus, inspired by Hall *et al.* [2017] and Zafar and Khan [2021], we create clusters of the data, and give the same interpretation to all instances that fall into the same cluster. To describe how SupClus works, let us use a synthetic dataset with

$$Y = 20X_1 \mathbf{1}_{X_1 \leq 0.5}(X_1) + (20 - 20X_1)\mathbf{1}_{X_1 > 0.5}(X_1) + \epsilon,$$

Algorithm 1 VarImp

Input: Data $\mathbb{D} = \{(\mathbf{x}_1, f(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, f(\mathbf{x}_n))\}$ with features and predictions, maximum number of features M, kernel bandwidth h (hyperparameter), instance to be explained i

Output: Local coefficients $\hat{\boldsymbol{\beta}}_i$

1: $\mathbf{I} \leftarrow$ varimp($\mathbb{D}$) // matrix with local variable importance for each instance from random forest algorithm
2: $\mathbf{W}_i \leftarrow$ ker($\mathbf{I}, \mathbf{X}, h, i$) // weights for local linear regression using Gaussian kernel and local variable importance
3: $\hat{\boldsymbol{\beta}}_i \leftarrow$ locstep($\mathbb{D}, \mathbf{W}_i, M$) // local linear regression coefficients using forward stepwise
4: **Return** $\hat{\boldsymbol{\beta}}_i$

$X_j \sim \text{Unif}(0,1)$, $\epsilon \sim \text{Normal}(0,1)$ and $d = 20$ (Figure 2). In this example, it would be sufficient to generate interpretations for only two segments of the data. The standard approach creates clusters using only the features. However, this fails to find clusters with different interpretations (Figure 2a, left). To overcome this, SupClus is based on the training of a supervised cluster that considers relationships between variables and predictions in the creation of interpretable clusters (Figure 2b, right).

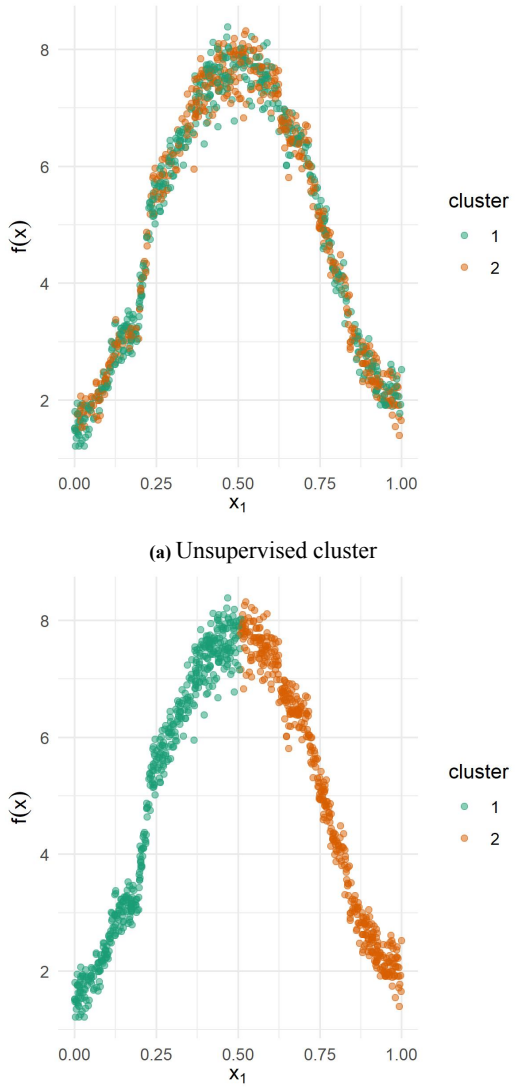
SupClus works as follows. First, we implement VarImp (Section 2) on $\mathbb{D}$ and collect the matrix of local coefficients $\hat{\mathbf{B}}_{n\times d} = \left[\hat{\boldsymbol{\beta}}_1 \ldots \hat{\boldsymbol{\beta}}_n\right]^t$. Then, we cluster the instances using the representation $\hat{\mathbf{B}}$. In order to allow for general shapes for the clusters, we use Spectral Cluster Analysis (the version proposed by Ng *et al*. [2001]), which maps the data into a space where standard clustering techniques can more effectively separate the groups. By construction, instances that fall into the same cluster will share a similar interpretation. Next, we train a different linear model for each cluster using all instances that fall into it. These models use the output $f(\mathbf{x})$ as the target. Again, a Forward Stepwise variable selection algorithm with a maximum number of variables $M \leq d$ is used in each cluster. The interpretation for each cluster is given by the coefficients of such model. To define the optimal number of clusters, we use an adjusted R^2 statistic, weighted by the size of the clusters: $R^2 = \frac{1}{n}\sum_{l=1}^{k}|c_l|R_l^2$, where $R_l^2 = 1 - \frac{(|c_l|-1)\sum_{i\in c_l}(f(\mathbf{x}_i)-g(\mathbf{x}_i))^2}{(|c_l|-M)\sum_{i\in c_l}\left(f(\mathbf{x}_i)-\overline{f_l(\mathbf{x})}\right)^2}$, $\overline{f_l(\mathbf{x})} = \sum_{i\in c_l} f(\mathbf{x}_i)/|c_l|$ and c_l is the size of the cluster l.

Algorithm 2 shows the pseudocode for SupClus.

4 Experimental design

We apply VarImp and SupClus both to artificial datasets, where the true local coefficients are known, and to real datasets. To assess their performance, they were compared with LIME [Ribeiro *et al*., 2016], Shapley values [Shapley, 2016; Štrumbelj and Kononenko, 2014; Lundberg and Lee, 2017], and IML [Molnar *et al*., 2018], a version of LIME with no perturbed samples.

Six artificial datasets (Table 1) were created, each trying to reflect what could be observed on real datasets (e.g., data with irrelevant attributes or data with non-linear relationships between features and target). In all cases, $d = 20$ and an error $\epsilon \sim \text{Normal}(0,1)$ was considered. Only a single run was performed for each dataset. The proposed methods were also applied to five real datasets (Table 2) from different application domains: health, education, economics, mobility, and ecology. Dataset 3 involves a classification problem where methods will be used to explain the predicted probability of heart disease.



(a) Unsupervised cluster

(b) Supervised cluster

Figure 2. Comparison of unsupervised and supervised clusters for artificial dataset: $Y = 20X_1\mathbf{1}_{X_1\leq 0.5}(X_1) + (20 - 20X_1)\mathbf{1}_{X_1>0.5}(X_1) + \epsilon$, $X_j \sim \text{Unif}(0,1)$, $\epsilon \sim \text{Normal}(0,1)$ and $d = 20$. Supervised cluster groups instances with similar interpretations.

First, the model to be interpreted was trained with the use of a sample $\mathbb{D}'$ in all experiments. On the simulated data, four ML algorithms (Random Forest, XGBoost, Multi-layer perceptron and LightGBM) were trained with hyperparameters computed by cross-validation on a grid of values. To evaluate the methods on the real data, the Random Forest algorithm with default hyperparameter values was used, as it offers strong predictive performance even without extensive tuning. Note that since the methods used are agnostic, any supervised algorithm could have been used. For the comparison, the interpretability methods were applied

Table 1. Artificial datasets representing a variety of real datasets.

Dataset	Features	Target
Linear regression w/ irrelevant features (1)	$X_j \sim \text{Unif}(0,1)$	$Y = 1 + 5X_1 - 5X_2 + \epsilon$
Linear regressions combination w/ irrelevant features (2)	$X_j \sim \text{Unif}(0,1)$	$Y = 20X_1\mathbf{1}_{X_1 \leq 0.5}(X_1) + (20 - 20X_1)\mathbf{1}_{X_1 > 0.5}(X_1) + \epsilon$
Relevant continuous features (3)	$X_j \sim \text{Unif}(0,1)$	$Y = \sum_{j=1}^{d}(-1)^{j+1}X_j + \epsilon$
Relevant binary features (4)	$X_j \sim \text{Ber}(1/2)$	$Y = \sum_{j=1}^{d}(-1)^{j+1}X_j + \epsilon$
Step function w/ irrelevant features (5)	$X_j \sim \text{Unif}(0,1)$	$Y = 20X_1\mathbf{1}_{1/3 \leq X_1 \leq 2/3}(X_1) + \epsilon$
Sine function w/ irrelevant features (6)	$X_j \sim \text{Unif}(0,1)$	$Y = 20\sin(2\pi X_1) + \epsilon$

Table 2. Real datasets from different application domains: health, education, economics, mobility, and ecology.

Dataset	Description
Bike rental (1)	Dataset with daily bike rental counts (target) in Washington DC. Features: temperature, humidity, wind speed, etc. Source: http://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset.
Ecology (2)	Above-ground biomass (target) in Australia. Features: temperature, precipitation, soil density, pH, etc. Source: https://datadryad.org/stash/dataset/doi:10.5061/dryad.dr7sqv9w9.
Heart diseases (3)	Data on heart diseases (target) and risk factors in the USA. Features: age, body mass index, other diseases, etc. Source: https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease?resource=download.
HDI (4)	Data with Human Development Index (target) from 5565 Brazilian cities. Features: information on income, education, and health. Source: https://www.ipea.gov.br/ipeageo/arquivos/bases.
Student´s performance (5)	Data on performance (target) of secondary students in Portugal. Features: number of absences, previous grades, age, hours of study, travel time, etc. Source: https://archive.ics.uci.edu/ml/datasets/Student+Performance.

Algorithm 2 SupClus

Input: Data $\mathbb{D} = \{(\mathbf{x}_1, f(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, f(\mathbf{x}_n))\}$ with features and predictions, maximum number of features M, local coefficient matrix $\hat{\mathbf{B}}$ from VarImp, number of clusters k (hyperparameter), instance to be explained i

Output: Local coefficients $\hat{\gamma}_i$ and weighted coefficient of determination r_k

1: $\mathbf{c} \leftarrow \text{specc}(\hat{\mathbf{B}}, M, k)$ // cluster for each instance using local VarImp coefficients

2: $\hat{\mathbf{A}} = [\hat{\boldsymbol{\alpha}}_1 \ldots \hat{\boldsymbol{\alpha}}_M]^t \leftarrow \text{regcluster}(\mathbf{c}, \mathbb{D})$ // local linear regression coefficients ($\hat{\mathbf{A}}$) fitted with forward stepwise for each cluster

3: $\hat{\boldsymbol{\gamma}}_i = \hat{\boldsymbol{\alpha}}_l : i \in c_l$ // finds coefficients of the cluster (c_l) to which i belongs

4: **Return** $\hat{\boldsymbol{\gamma}}_i$

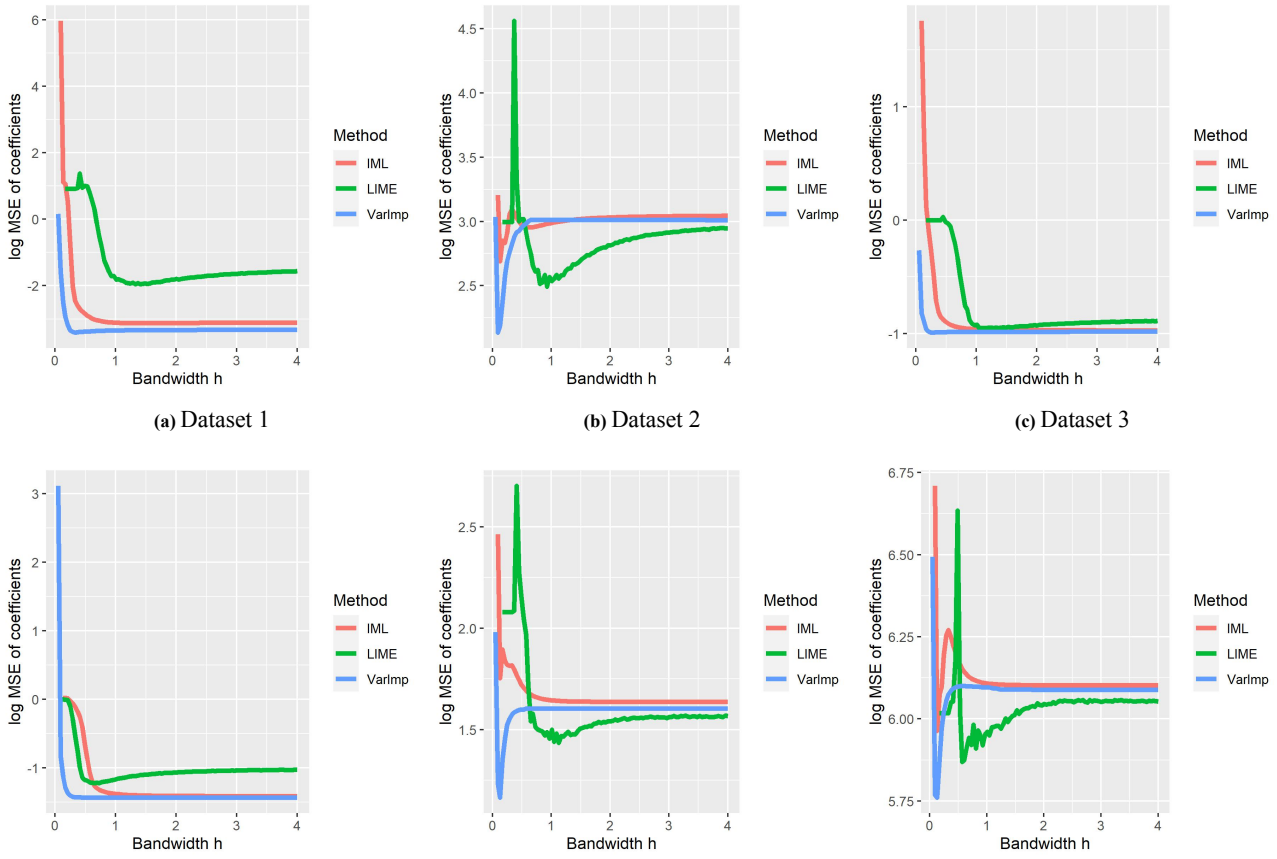
to dataset $\mathbb{D}$ using the predictions from the original model as target.

In the VarImp, LIME and IML methods, a bandwidth h between 0.05 and 4.00 was used. These values were obtained by analyzing local slope plots and mean square error of coefficients (Subsection 4.1). When true coefficients are not known we use coefficients obtained by the ICE method [Goldstein *et al.*, 2015]. Figure 3 shows that VarImp has smaller errors compared to LIME and IML even considering different values of bandwidth h. The number of clusters k in SupClus varied between 1 and 5 and the cluster with the highest R^2 was chosen. All available features were analyzed for all methods, with $M = d$, for comparisons with Shapley values. For the experiments with the real data sets, categorical variables were transformed into indicators and continuous variables were rescaled between 0 and 1 by min-max scaling. LIME, IML and Shapley values were applied using packages available in the programming language R [Molnar *et al.*, 2018; Pedersen and Benesty, 2021].

4.1 Evaluation measures

Measures for evaluating interpretability methods that involve surveys have the limitation of being subjective in addition to using only a few points from a sample. Other measures involve aspects such as complexity and stability and do not directly assess the quality of interpretations. This section presents measures for evaluations and comparisons of the quality of interpretations and the predictive performance of the proposed methods. The measurements take into account all points in a sample $\mathbb{D}$.

Mean square error of coefficients. For artificial data whose values of local coefficients are known, we compute the mean squared error between real and estimated coefficients, $\text{MSE}_{\mathbf{B},\hat{\mathbf{B}}} = \frac{1}{nd}\sum_{l=1}^{n}\sum_{j=1}^{d}\left(\beta_{l,j} - \hat{\beta}_{l,j}\right)^2$. This measure directly assesses the quality of interpretations through local



(a) Dataset 1 **(b)** Dataset 2 **(c)** Dataset 3

(d) Dataset 4 **(e)** Dataset 5 **(f)** Dataset 6

Figure 3. Influence of bandwidth h in MSE of coefficients for artificial datasets. VarImp has smaller errors compared to LIME and IML even considering different values of bandwidth h.

coefficients and therefore cannot be used for Shapley values whose local coefficients are not estimated.

Effect correlation. Still for artificial data, we compute the Pearson correlation between real and estimated effects, $r_{\phi,\hat{\phi}}$. This measure assesses the quality of interpretations through local effects and can be applied to Shapley values.

Prediction correlation. We assess the predictive performance of the interpretations using the correlations between predictions from the original model and the local model, $r_{f,g}$. This measure evaluates the fidelity of the local model as an approximation of the original prediction algorithm f.

ICE effect correlation. The Individual Conditional Expectation method Goldstein *et al.* [2015] generates visualizations of the curves $g_{i,j}(x) = f(x_{i,1}, \ldots, x_{i,j-1}, x, x_{i,j+1}, \ldots, x_{i,d})$ for a feature j and instance i. Derivatives and their effects were obtained by this method and the Pearson correlation between ICE effects and the effects of an interpretability method, $r_{\phi_{\text{ICE}},\hat{\phi}}$, were calculated. This measure assesses the quality of interpretations through local effects and is especially useful in real data where true effects are not known.

Local slope plot. To check how well the local coefficients fit the ML algorithm f, we add line segments to the scatter plot of the predictions $f(\mathbf{x})$ versus each feature x_l. These segments have a slope given by the local coefficient associated with x_l.

4.2 Practical recommendations for selecting the hyperparameters h and k

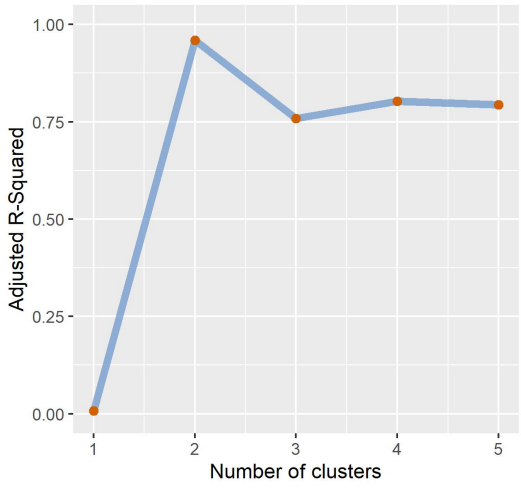
In the SupClus method, the optimal number of clusters (k) can be identified using the coefficient of determination weighted by the size of each cluster. Accordingly, the value of k that maximizes this coefficient is selected. In the VarImp method, the bandwidth (h) can be determined by maximizing the correlation between the estimated effects and the effects obtained using the ICE method.

5 Experimental results

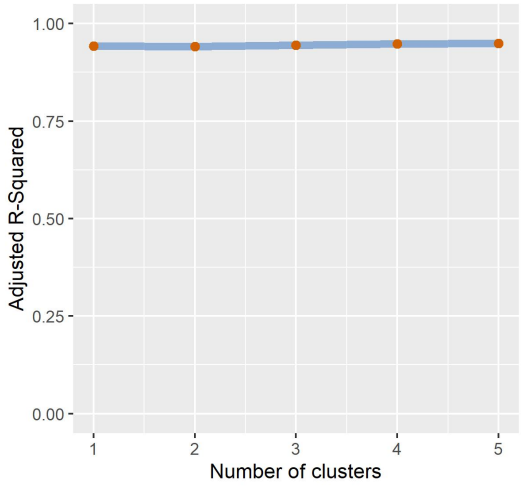
5.1 Artificial data

Number of clusters. As previously mentioned, the main advantage of SupClus when compared with the other methods is that it can identify simpler datasets whose instances can be interpreted by using clusters. As an example, datasets 1, 3, and 4 can be interpreted with only 1 cluster and dataset 2 (or dataset 5) can be interpreted with only 2 or 3 clusters (Figure 2b). On the other hand, dataset 6 requires interpretations for each instance due to its complexity. A weighted coefficient of determination obtained the optimal number of clusters in each dataset (Figure 4).

Mean square error of coefficients. Since the true local coefficients are known for those datasets, the proposed



(a) Dataset 2



(b) Dataset 4

Figure 4. Coefficient of determination with a variable number of clusters for datasets 2 and 4. Dataset 2 can be interpreted with only 2 clusters and dataset 4 can be interpreted with only 1 cluster.

methods can be more accurately compared with other available methods. Table 3 shows that, in general, VarImp presented lower MSE values, especially for more complex datasets, such as dataset 6. All methods have low errors in datasets 3 and 4, where all variables are relevant and the relationships of X_1 with the target are linear. SupClus has the smallest coefficient error in datasets 2 and 5, which contain well-defined clusters and irrelevant variables. LIME and IML have higher errors for datasets with irrelevant variables and when the relationships are not linear. LIME has the largest error in dataset 1. We conclude that our methods do a good job in approximating true local slopes end that the results are independent of the ML algorithm used.

Effect and prediction correlation. Another way of comparing interpretation methods in artificial data is through *effect correlation* from subsection 4.1 (Table 4). All conclusions with MSE are reinforced with this measure. Shapley value does not show good correlations of effects in most datasets, although it yields a good indicator of which variables are the most important (regardless of the sign of the relationship). All methods show good correlations between local and global predictions (Table 4).

Local slope plots. Figure 5 shows that the coefficients estimated by VarImp are closer to the true coefficients for dataset 6 when compared to the other approaches. In general, VarImp has better coefficients for data with non-linear relationships between target and feature (see Appendix 6 for the other plots).

Instance analysis. Finally, Figure 6 shows a comparison of the estimated effects of a specific instance made by VarImp, SupClus, LIME, IML, and Shapley values for dataset 2, where only X_1 is shown to be relevant ($Y = 20X_1\mathbf{1}_{X_1 \leq 0.5}(X_1) + (20 - 20X_1)\mathbf{1}_{X_1 > 0.5}(X_1) + \epsilon$). We analyze an instance with $X_1 \leq 0.5$, for which a positive effect is expected for X_1. VarImp and SupClus provided better interpretations, whereas LIME and IML estimated large effects for irrelevant variables and Shapley value estimated a negative effect for X_1. The same conclusions also hold for the other data sets (see Appendix 6).

5.2 Real data

Comparing interpretability methods using real datasets is more challenging since the true local coefficients are not known. Thus, instead of computing the correlations between estimated and true effects, we compared the estimated effects of the methods with the estimated effects using ICE, as described in Section 4 (Figure 7a). Correlations between local and global predictions were also calculated (Figure 7b).

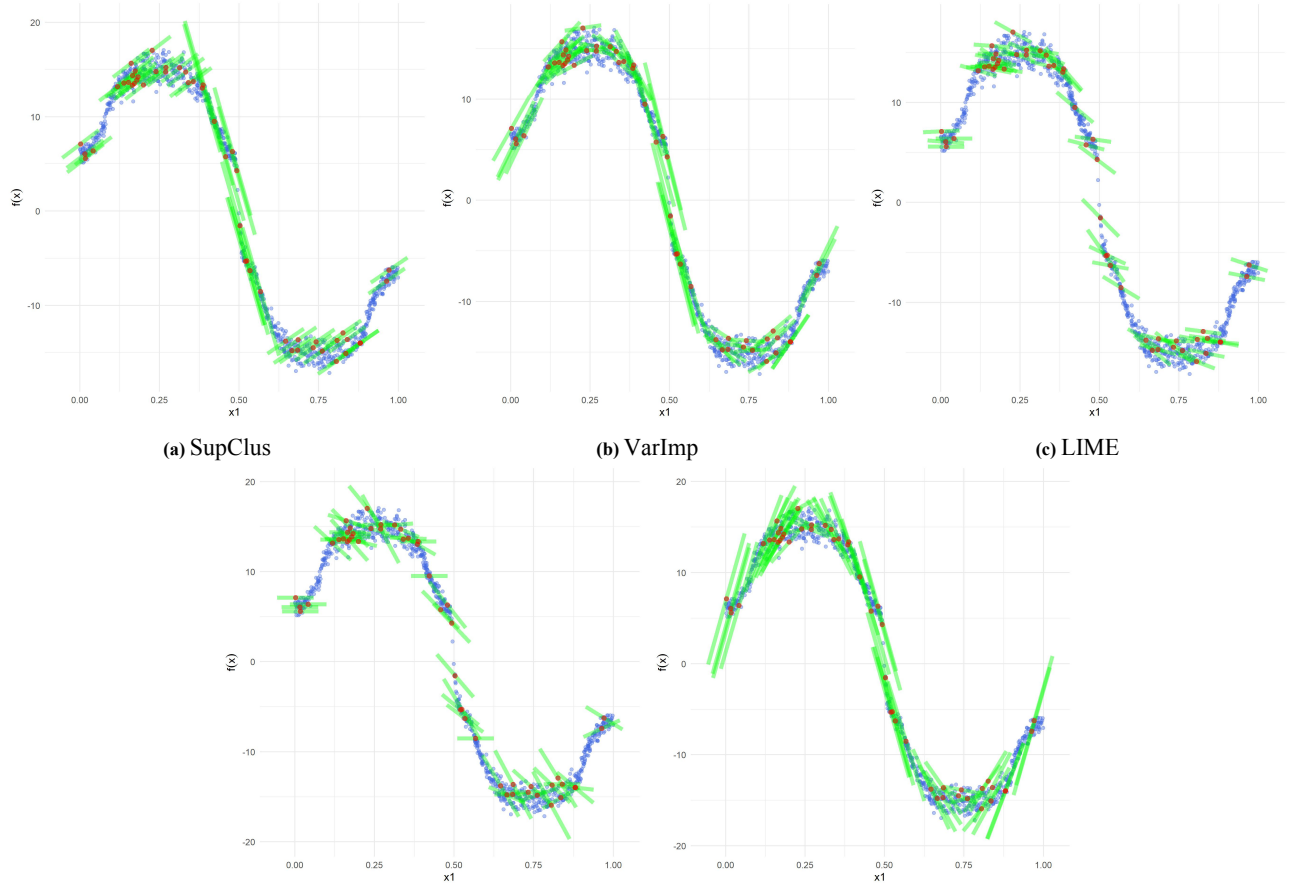
ICE effect and Prediction correlation. Shapley values have the worst correlations of ICE effects in all datasets. VarImp, SupClus, LIME, and IML have the best correlations of effects for datasets 3 and 5. On the other hand, VarImp, LIME, and IML have the best correlations for dataset 1, VarImp and LIME have the best correlations for dataset 2 and VarImp and IML have the best correlations for dataset 4. All methods show good correlations of predictions for all datasets. Overall, VarImp has better ICE effect correlations than the other methods.

Extreme cases. When we analyze instances individually, it is interesting to compare extreme cases (low and high predictions) to evaluate if the interpretations make sense. Using the HDI dataset we compared two cities with extreme predictions using VarImp (Figure 8). The prediction of the low HDI city is mainly due to the high infant mortality rate. On the other hand, the prediction of the city with a high HDI can be explained by the high percentage of households with electricity and garbage collection. Thus, this analysis reinforces the quality of the interpretations.

Table 5 presents a comparison of the processing time required by each method to explain the entire test sample in each experiment. Overall, the Shapley values and LIME methods exhibited the highest processing times, whereas VarImp, SupClus, and IML showed the lowest. According to the experiments, VarImp, SupClus, and IML are more advantageous when the goal is to interpret the full sample; conversely, LIME and Shapley values are more suitable when only a subset of instances needs to be interpreted. All experiments were conducted on a computer equipped with a 13th Gen Intel(R) Core(TM) i7-13700 processor (2.10 GHz), 16.0 GB of installed RAM, and running Windows 11 Pro

Table 3. Mean square error of coefficients for artificial data and four ML algorithms. VarImp presented lower MSE values for more complex datasets. SupClus has the smallest coefficient error in dataset 2 and 5, which contains well-defined clusters and irrelevant variables.

Dataset	Model	$MSE_{B,\hat{B}}$			
		VarImp	**SupClus**	**LIME**	**IML**
Linear regression w/ irrelevant features (1)	Random Forest	0.009	0.010	0.041	0.011
	XGBoost	0.033	0.036	0.054	0.034
	Multi-layer perceptron	0.014	0.014	0.029	0.013
	LightGBM	0.057	0.059	0.079	0.045
Linear regressions combination w/ irrelevant features (2)	Random Forest	**2.381**	**0.256**	11.139	13.834
	XGBoost	**2.446**	**1.116**	11.069	14.218
	Multi-layer perceptron	**2.827**	**0.382**	10.930	13.853
	LightGBM	**2.638**	**0.188**	11.455	16.935
Relevant continuous features (3)	Random Forest	0.387	0.389	0.396	0.390
	XGBoost	0.010	0.010	0.027	0.010
	Multi-layer perceptron	0.000	0.000	0.000	0.001
	LightGBM	0.008	0.009	0.023	0.009
Relevant binary features (4)	Random Forest	0.258	0.258	0.296	0.259
	XGBoost	0.000	0.000	0.012	0.000
	Multi-layer perceptron	0.000	0.000	0.000	0.001
	LightGBM	0.001	0.001	0.019	0.002
Step function w/ irrelevant features (5)	Random Forest	**1.573**	**0.347**	4.541	4.901
	XGBoost	**1.629**	**2.467**	4.454	4.797
	Multi-layer perceptron	**1.771**	**0.106**	4.550	4.922
	LightGBM	**1.560**	**0.484**	4.718	4.964
Sine function w/ irrelevant features (6)	Random Forest	**7.106**	114.900	363.825	376.451
	XGBoost	**6.969**	296.940	380.039	381.046
	Multi-layer perceptron	**7.753**	88.837	374.787	380.571
	LightGBM	**34.106**	88.584	372.643	399.236



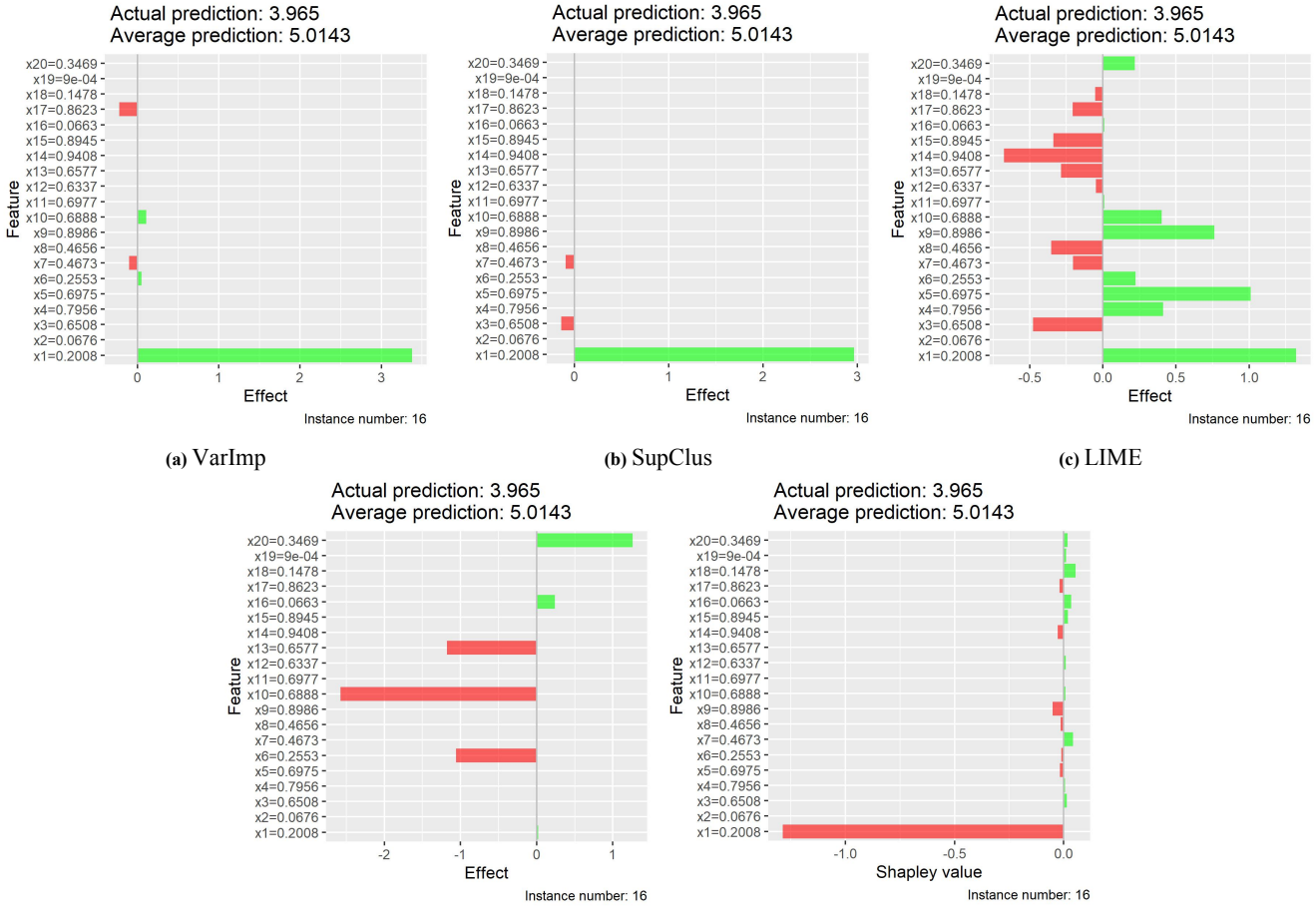
(a) SupClus **(b)** VarImp **(c)** LIME

(d) IML **(e)** True

Figure 5. Local slope plots (green) for sample points (red) from the dataset 6. Coefficients estimated by VarImp are closer to the true coefficients.

Table 4. Correlations of effects and predictions for artificial data and four ML algorithms. VarImp and SupClus provide better or equal measures than other methods.

Dataset	Model	Effect Correlation					Prediction Correlation				
		VarImp	SupClus	LIME	IML	Shapley	VarImp	SupClus	LIME	IML	Shapley
1	Random Forest	1.00	1.00	1.00	1.00	0.49	0.99	0.99	1.00	0.99	0.99
	XGBoost	1.00	1.00	0.99	1.00	0.49	0.99	0.99	1.00	0.99	0.99
	Multi-layer perceptron	1.00	1.00	0.99	1.00	0.49	1.00	1.00	0.99	0.99	0.98
	LightGBM	0.99	1.00	0.99	1.00	0.48	0.98	0.98	0.98	0.97	0.97
2	Random Forest	**0.95**	**1.00**	0.71	0.60	0.21	1.00	1.00	1.00	1.00	0.99
	XGBoost	**0.95**	**0.98**	0.70	0.58	0.21	1.00	0.99	1.00	1.00	0.99
	Multi-layer perceptron	**0.95**	**0.99**	0.72	0.56	0.21	1.00	0.99	1.00	1.00	0.99
	LightGBM	**0.95**	**1.00**	0.68	0.46	0.20	0.99	0.98	0.99	0.99	0.98
3	Random Forest	0.95	0.95	0.93	0.95	0.46	0.96	0.95	1.00	0.96	0.99
	XGBoost	1.00	1.00	0.99	1.00	0.49	0.99	0.99	0.99	0.99	0.98
	Multi-layer perceptron	1.00	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	0.99
	LightGBM	1.00	1.00	0.99	1.00	0.49	0.99	0.99	0.99	0.98	0.98
4	Random Forest	0.97	0.97	0.96	0.97	0.67	0.97	0.97	1.00	0.97	0.99
	XGBoost	1.00	1.00	1.00	1.00	0.70	1.00	1.00	1.00	1.00	0.99
	Multi-layer perceptron	1.00	1.00	1.00	1.00	0.70	1.00	1.00	1.00	1.00	0.99
	LightGBM	1.00	1.00	1.00	1.00	0.70	1.00	1.00	1.00	1.00	0.99
5	Random Forest	**0.87**	**0.96**	0.57	0.53	0.11	0.99	1.00	1.00	0.94	0.99
	XGBoost	**0.87**	**0.80**	0.58	0.53	0.11	0.99	1.00	1.00	0.94	0.99
	Multi-layer perceptron	**0.85**	**0.99**	0.57	0.53	0.11	0.99	0.99	0.99	0.94	0.98
	LightGBM	**0.87**	**0.95**	0.57	0.53	0.10	0.98	0.99	0.99	0.93	0.98
6	Random Forest	**1.00**	**0.85**	0.26	0.35	-0.08	1.00	0.89	1.00	1.00	0.99
	XGBoost	**1.00**	**0.90**	0.23	0.34	-0.08	1.00	0.99	1.00	1.00	0.99
	Multi-layer perceptron	**1.00**	**0.89**	0.25	0.33	-0.08	1.00	0.99	1.00	1.00	0.99
	LightGBM	**0.96**	**0.87**	0.24	0.29	-0.10	1.00	0.99	1.00	1.00	0.99

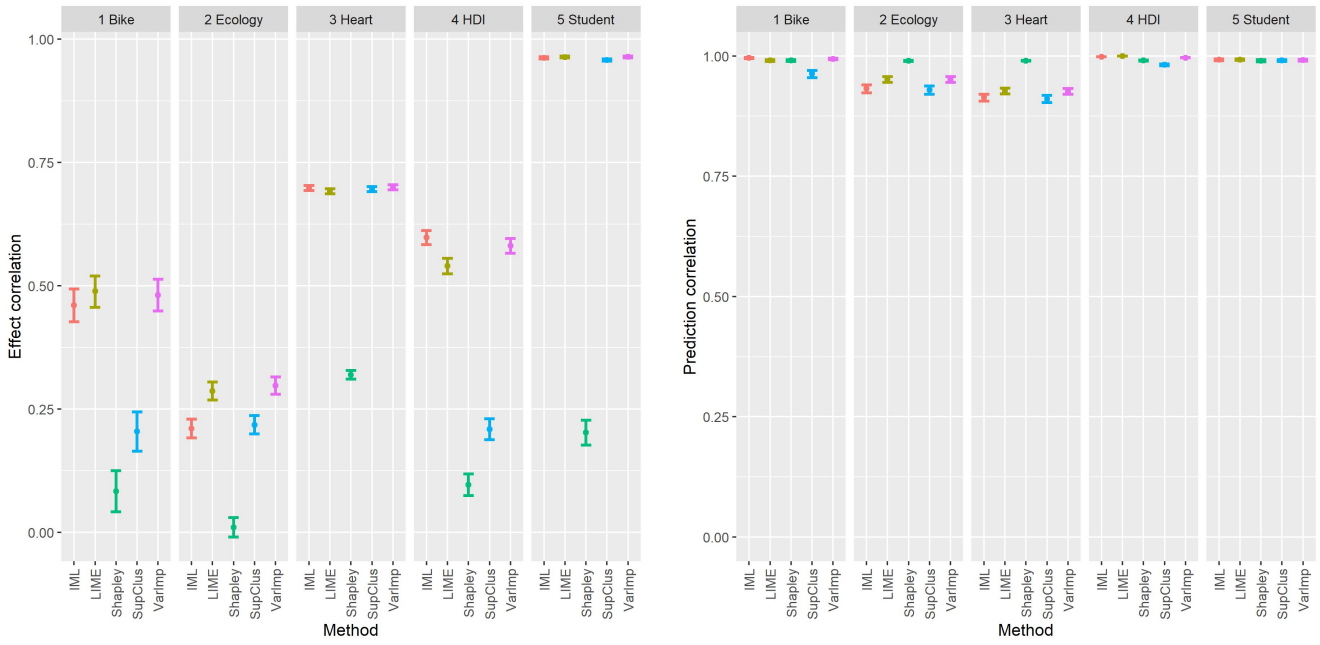


(a) VarImp **(b)** SupClus **(c)** LIME

(d) IML **(e)** Shapley values

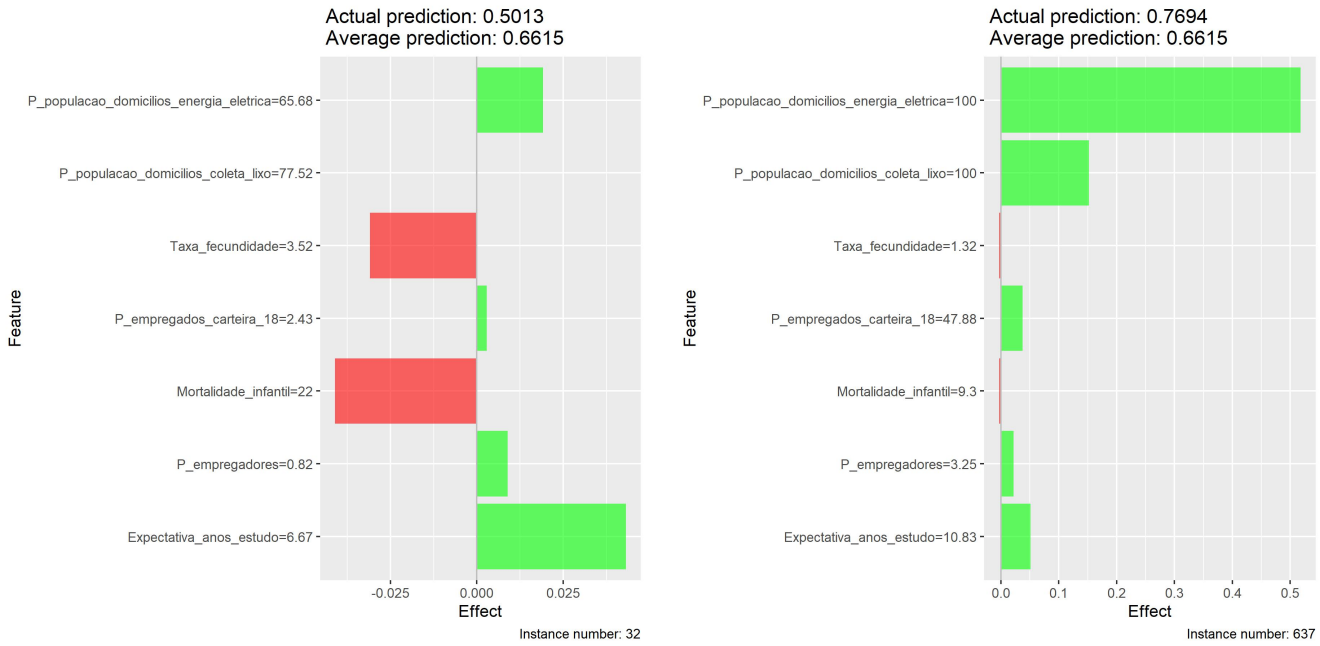
Figure 6. Estimated effects of an instance for dataset 2. VarImp and SupClus provided better interpretations.

(64-bit).



(a) Effect correlation **(b)** Prediction correlation

Figure 7. Correlations of effects and predictions for real data with 95% confidence level. Overall, VarImp has better ICE effect correlations than the other methods.



(a) Low prediction **(b)** High prediction

Figure 8. Interpretations of instances with extreme predictions from the HDI dataset using VarImp. The prediction of the low HDI city is mainly due to the high infant mortality rate. The prediction of the city with a high HDI can be explained by the high percentage of households with electricity and garbage collection.

6 Conclusion

This article proposed two new methods of local and agnostic interpretability, namely VarImp and SupClus, which can better deal with high-dimensional situations with irrelevant variables and when the relationship between input variables and target is not linear. Both are based on a local linear regression improved through the weights of local variables included in the Euclidean distance. While SupClus interprets instances in groups with similar interpretations, VarImp can be applied to data with more complex relationships and that require individual interpretations per instance.

Measures for assessments of the quality of VarImp and SupClus interpretations, and to compare them with other methods, were used. These measures consider not one instance only, but a complete sample of points, and use non-subjective criteria more directly related to the quality of interpretations. A way to visualize the interpretations of a

Table 5. Processing time required by each method to explain the entire test sample in each experiment.

Dataset	Model	Processing time (min)				
		VarImp	**SupClus**	**LIME**	**IML**	**Shapley**
Linear regression w/ irrelevant features	Random Forest	0.137	0.138	8.762	0.627	10.377
	XGBoost	0.133	0.134	8.548	0.617	12.030
	Multi-layer perceptron	0.139	0.139	7.898	0.451	17.489
	LightGBM	0.138	0.138	7.963	0.466	17.638
Linear regressions combination w/ irrelevant features	Random Forest	0.136	0.137	8.795	0.629	10.771
	XGBoost	0.136	0.136	10.470	0.999	12.384
	Multi-layer perceptron	0.139	0.139	7.997	0.456	10.252
	LightGBM	0.138	0.139	8.013	0.470	9.968
Relevant continuous features	Random Forest	0.134	0.134	11.131	1.224	12.743
	XGBoost	0.134	0.134	8.580	0.618	10.505
	Multi-layer perceptron	0.130	0.131	7.894	0.463	9.733
	LightGBM	0.133	0.134	8.037	0.479	9.928
Relevant binary features	Random Forest	0.113	0.113	11.777	1.474	13.377
	XGBoost	0.112	0.113	7.963	0.484	9.888
	Multi-layer perceptron	0.114	0.114	7.937	0.470	9.712
	LightGBM	0.113	0.114	7.892	0.475	9.828
Step function w/ irrelevant features	Random Forest	0.138	0.139	8.879	0.647	10.752
	XGBoost	0.140	0.141	8.133	0.495	10.068
	Multi-layer perceptron	0.141	0.141	7.977	0.457	10.086
	LightGBM	0.143	0.143	7.942	0.464	9.933
Sine function w/ irrelevant features	Random Forest	0.137	0.138	11.342	1.163	12.527
	XGBoost	0.137	0.137	10.583	1.004	12.110
	Multi-layer perceptron	0.139	0.140	8.031	0.442	10.134
	LightGBM	0.139	0.139	8.026	0.457	11.177
Bike rental	Random Forest	0.015	0.030	1.902	0.127	1.492
Ecology	Random Forest	0.087	0.301	7.531	1.493	8.079
Heart diseases	Random Forest	0.318	0.319	26.561	6.309	28.075
HDI	Random Forest	0.080	0.081	9.614	2.615	9.117
Student performance	Random Forest	0.028	0.028	3.077	0.144	3.415

method for each variable is to plot the local coefficients for a sample of points.

Applications in artificial and real data indicated that the proposed methods achieve equal or better performance than known interpretability methods, LIME, IML, and Shapley values. In most datasets, VarImp showed better quality measures, especially for datasets with nonlinear relationships or irrelevant variables. SupClus performed well for simpler datasets, where clusters of instances with similar interpretations can be identified. Both proposed methods, together with LIME and IML provided good results in datasets with all relevant variables and linear relationships (which is common in practice). While Shapley values demonstrated strong theoretical properties and accurate local predictions, their quality measures were less effective in the applications studied. Similarly, LIME and IML, valued for their flexibility and ease of use, showed limitations in capturing complex patterns in datasets with nonlinear relationships.

Future Research. In this study, interpretability methods were compared using quantitative metrics. However, future user studies are needed to assess human understanding of the explanations generated by these methods. Another direction for future work is to compare the proposed methods with non–model-agnostic approaches and with other methods based on approximations of Shapley values.

Declarations

Authors' Contributions

GY: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. RI: Conceptualization, Methodology, Supervision, Writing – review editing. FZ, FL, AR, RB: Writing – review & editing. AC: Conceptualization, Funding acquisition, Methodology, Supervision, Writing – review & editing, Project administration. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Funding

Gilson Y. Shimizu is grateful for the financial support of Pró-Reitoria de Pesquisa da USP - Brazil. Rafael Izbicki is grateful for the financial support of FAPESP (grant 2019/11321-9 and 2023/07068-1) and CNPq (grants 309607/2020-5, 422705/2021-7 and 305065/2023-8). Andre C. P. L. F. de Carvalho is grateful for the financial support from CNPq (grants 309858/2021-6, 422705/2021-7) and 408589/2024-8 and from Fapesp 2020/09835-1.

Availability of data and materials

The datasets (and/or softwares) generated and/or analysed during the current study are available in https://github.com/gilsonshimizu/interpretability. The sources of the publicly available data are listed in Table 2.

References

Agarwal, R., Melnick, L., Frosst, N., Zhang, X., Lengerich, B., Caruana, R., and Hinton, G. E. (2021). Neural additive models: interpretable machine learning with neural nets. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. Curran Associates Inc.. DOI: 10.5555/3540261.3540620.

Alvarez-Melis, D. and Jaakkola, T. S. (2018). On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*. DOI: 10.48550/arXiv.1806.08049.

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115. DOI: 10.1016/j.inffus.2019.12.012.

Botari, T., Hvilshøj, F., Izbicki, R., and de Carvalho, A. C. (2020). Melime: meaningful local explanation for machine learning models. *arXiv preprint arXiv:2009.05818*. DOI: 10.48550/arXiv.2009.05818.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32. DOI: 10.1023/A:1010933404324.

Coscrato, V., Inácio, M. H., Botari, T., and Izbicki, R. (2023). Nls: An accurate and yet easy-to-interpret prediction method. *Neural Networks*, 162:117–130. DOI: 10.1016/j.neunet.2023.02.043.

Goldstein, A., Kapelner, A., Bleich, J., and Pitkin, E. (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. DOI: 10.1080/10618600.2014.907095.

Hakkoum, H., Idri, A., and Abnane, I. (2024). Global and local interpretability techniques of supervised machine learning black box models for numerical medical data. *Engineering Applications of Artificial Intelligence*, 131:107829. DOI: 10.1016/j.engappai.2023.107829.

Hall, P., Gill, N., Kurka, M., and Phan, W. (2017). Machine learning interpretability with h2o driverless ai. *H2O. ai*. Available at: `https://docs.h2o.ai/driverless-ai/latest-stable/docs/booklets/MLIBooklet.pdf`.

Hechtlinger, Y. (2016). Interpretation of prediction models using the input gradient. *arXiv preprint arXiv:1611.07634*. DOI: 10.48550/arXiv.1611.07634.

Hu, L., Chen, J., Nair, V. N., and Sudjianto, A. (2018). Locally interpretable models and effects based on supervised partitioning (lime-sup). *arXiv preprint arXiv:1806.00663*. DOI: 10.48550/arXiv.1806.00663.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer. DOI: 10.1007/978-1-0716-1418-1.

Kim, B., Khanna, R., and Koyejo, O. (2016). Examples are not enough, learn to criticize! criticism for interpretability. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 2288–2296, Red Hook, NY, USA. Curran Associates Inc. Available at: `https://papers.nips.cc/paper_files/paper/2016/file/5680522b8e2bb01943234bce7bf84534-Paper.pdf`.

Koh, P. W. and Liang, P. (2017). Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1885–1894. JMLR.org. Available at: `https://proceedings.mlr.press/v70/koh17a.html`.

Lee, J. D., Sun, D. L., Sun, Y., and Taylor, J. E. (2016). Exact post-selection inference, with application to the lasso. *The Annals of Statistics*, 44(3):907 – 927. DOI: 10.1214/15-AOS1371.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA. Curran Associates Inc. Available at: `https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf`.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38. DOI: 10.1016/j.artint.2018.07.007.

Molnar, C. (2020). *Interpretable machine learning*. Lulu.com. DOI: 10.21105/joss.00786.

Molnar, C., Casalicchio, G., and Bischl, B. (2018). iml: An r package for interpretable machine learning. *Journal of Open Source Software*, 3(26):786. DOI: 10.21105/joss.00786.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: analysis and an algorithm. In *Proceedings of the 15th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, page 849–856, Cambridge, MA, USA. MIT Press. Available at:`https://papers.nips.cc/paper_files/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf`.

Pedersen, T. L. and Benesty, M. (2021). lime: Local interpretable model-agnostic explanations. *R package version 0.5.2*. DOI: 10.32614/CRAN.package.lime.

Ribeiro, M. T., Singh, S., and 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*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/2939672.2939778.

Saeed, W. and Omlin, C. (2023). Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems*, 263:110273. DOI: 10.1016/j.knosys.2023.110273.

Shapley, L. S. (2016). *17. A Value for n-Person Games*, pages 307–318. Princeton University Press, Princeton. DOI: 10.1515/9781400881970-018.

Slack, D., Hilgard, S., Jia, E., Singh, S., and Lakkaraju, H. (2020). Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, page 180–186, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3375627.3375830.

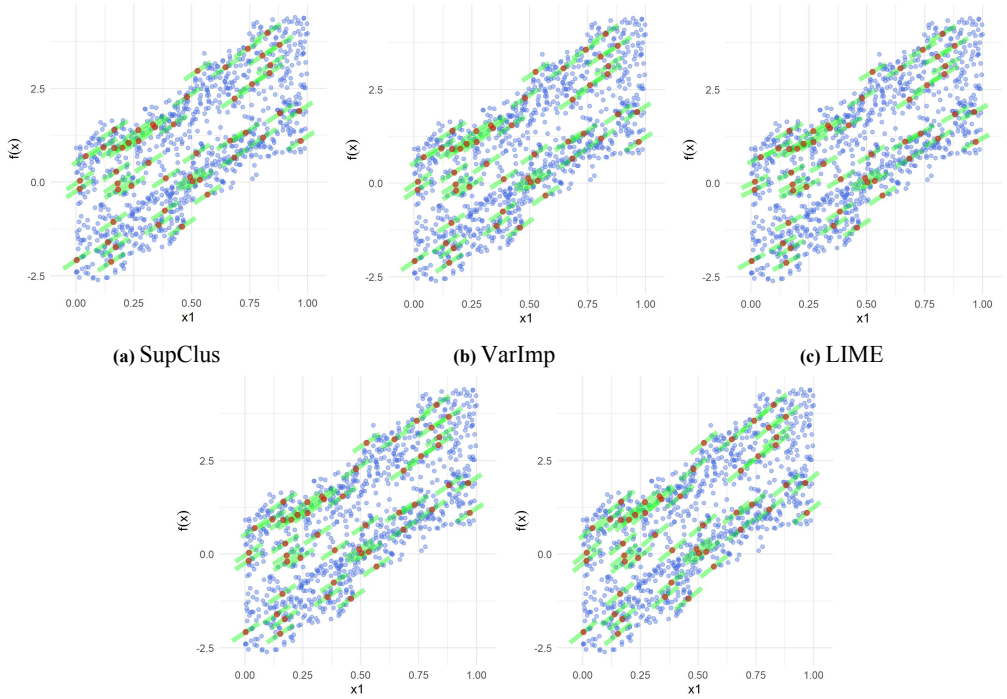
Štrumbelj, E. and Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665. DOI: 10.1007/s10115-013-0679-x.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org. Available at:`https://proceedings.mlr.press/v70/sundararajan17a.html`.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

Zafar, M. R. and Khan, N. (2021). Deterministic local interpretable model-agnostic explanations for stable explainability. *Machine Learning and Knowledge Extraction*, 3(3):525–541. DOI: 10.3390/make3030027.

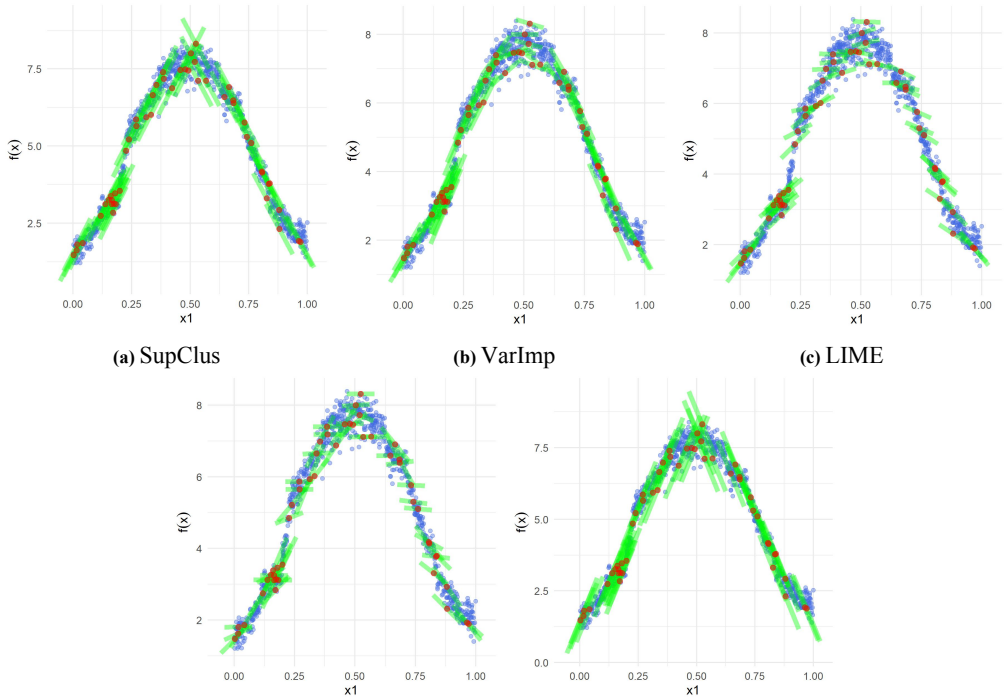
Appendix A. Local slope plots for artificial datasets.



(a) SupClus **(b)** VarImp **(c)** LIME

(d) IML **(e)** True

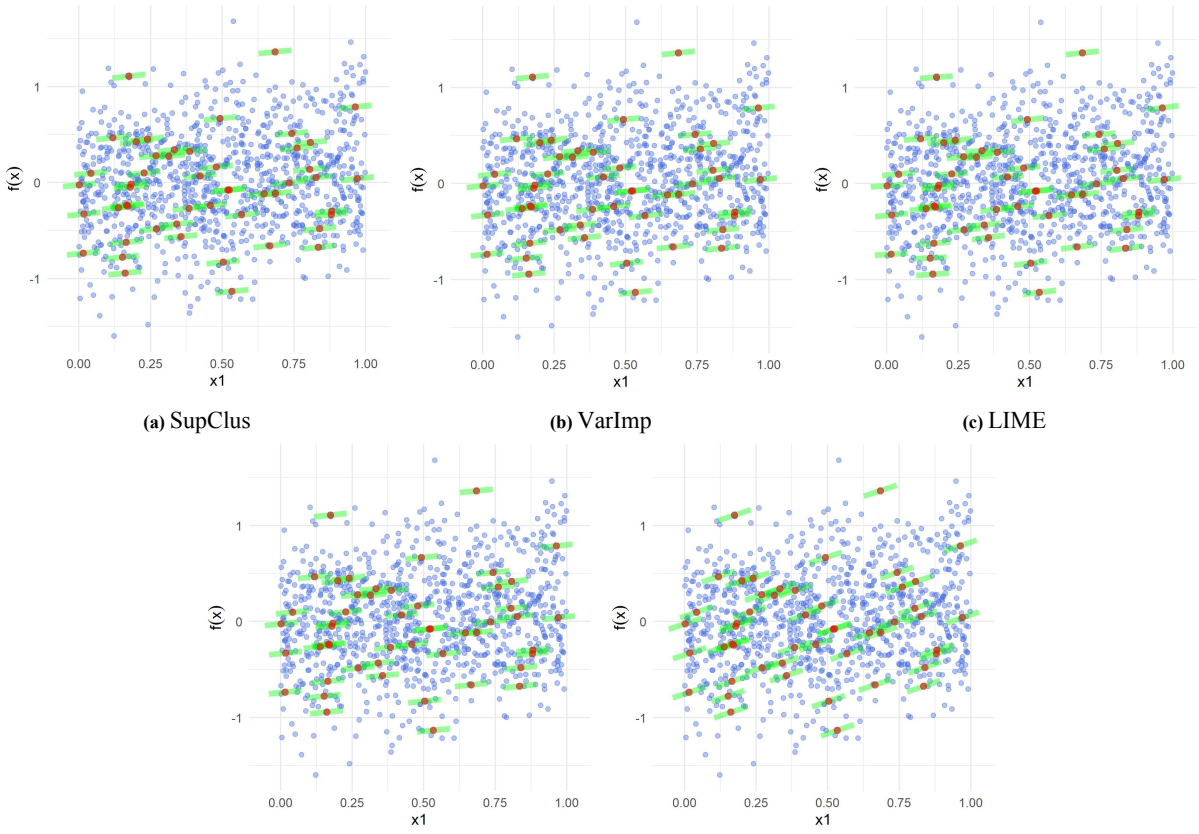
Figure 9. Local slope plots (green) for sample points (red) from the dataset 1.



(a) SupClus **(b)** VarImp **(c)** LIME

(d) IML **(e)** True

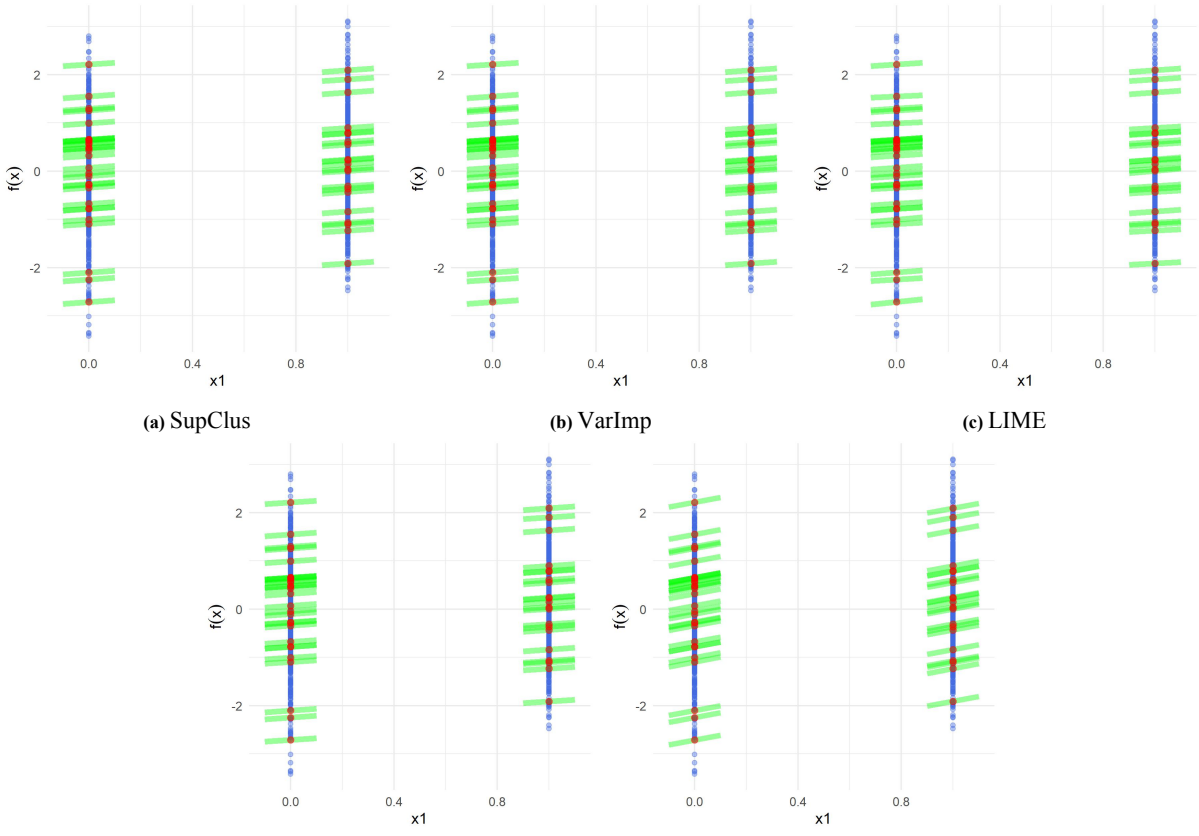
Figure 10. Local slope plots (green) for sample points (red) from the dataset 2.



(a) SupClus **(b)** VarImp **(c)** LIME

(d) IML **(e)** True

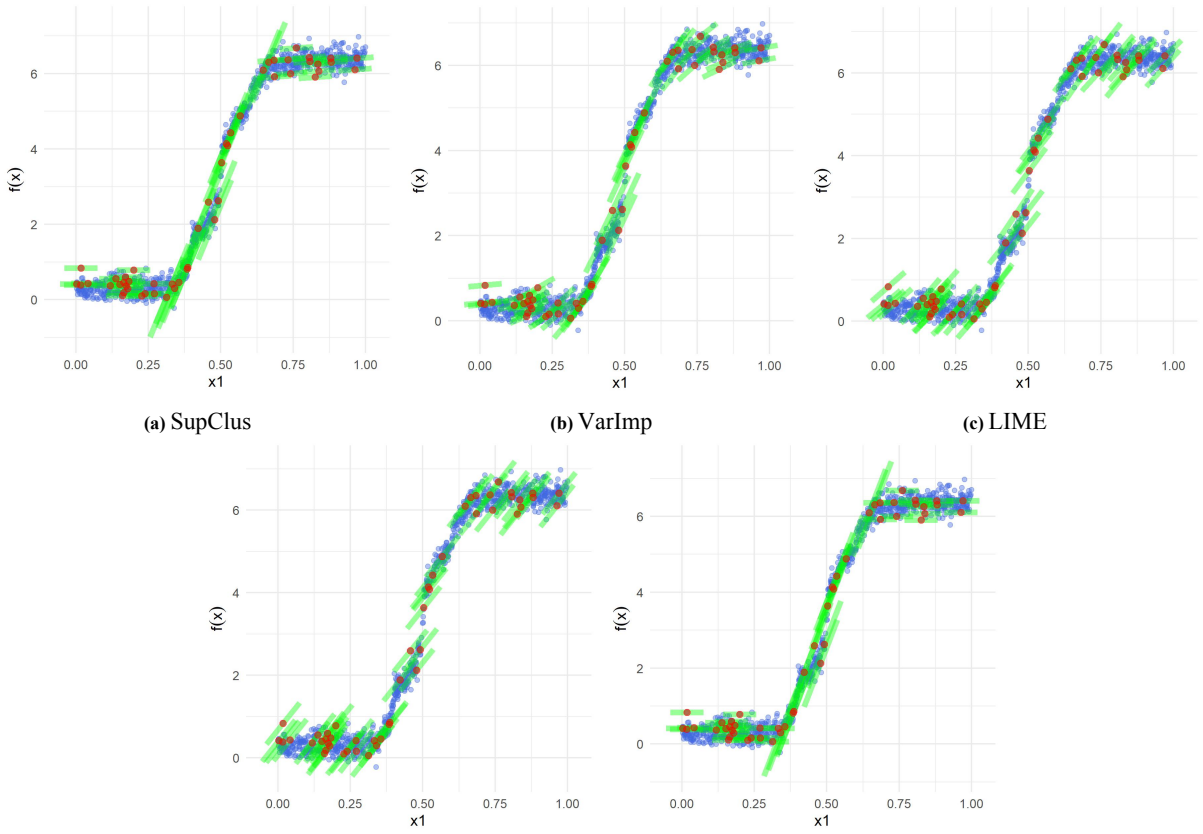
Figure 11. Local slope plots (green) for sample points (red) from the dataset 3.



(a) SupClus **(b)** VarImp **(c)** LIME

(d) IML **(e)** True

Figure 12. Local slope plots (green) for sample points (red) from the dataset 4.

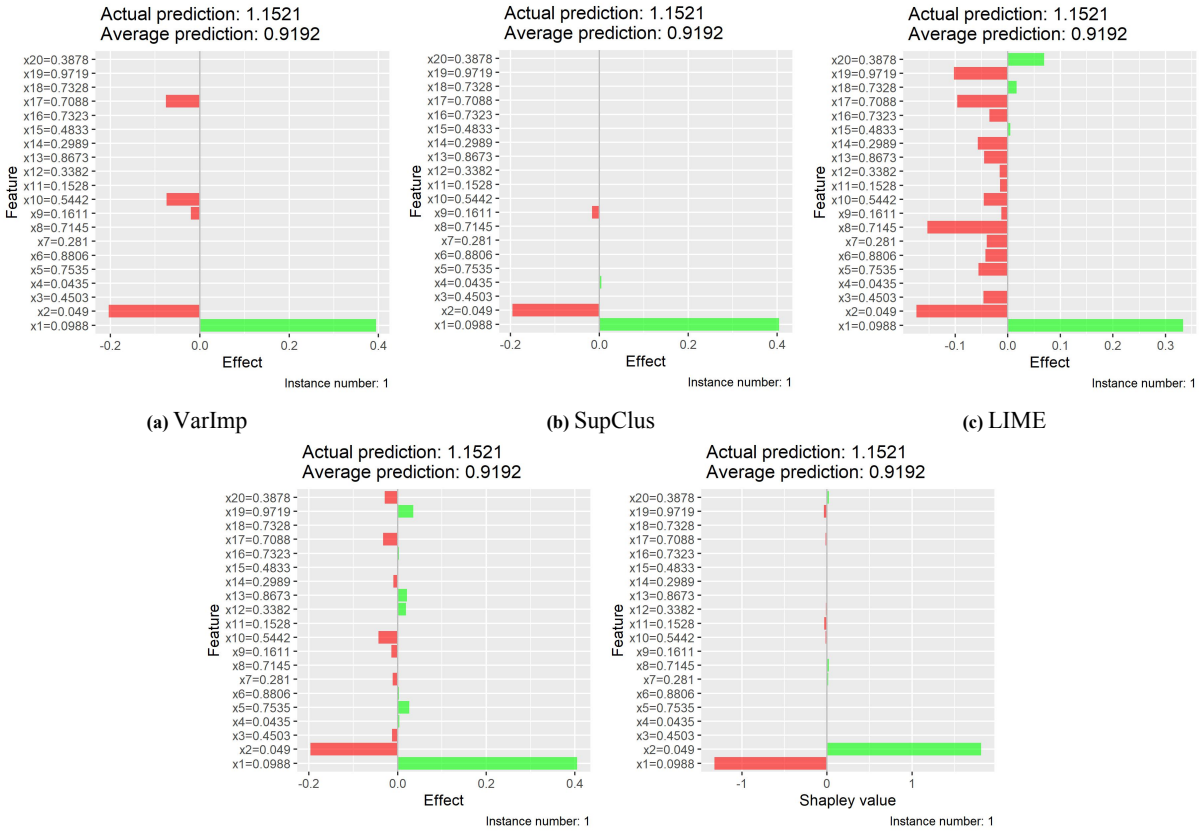


(a) SupClus **(b)** VarImp **(c)** LIME

(d) IML **(e)** True

Figure 13. Local slope plots (green) for sample points (red) from the dataset 5.

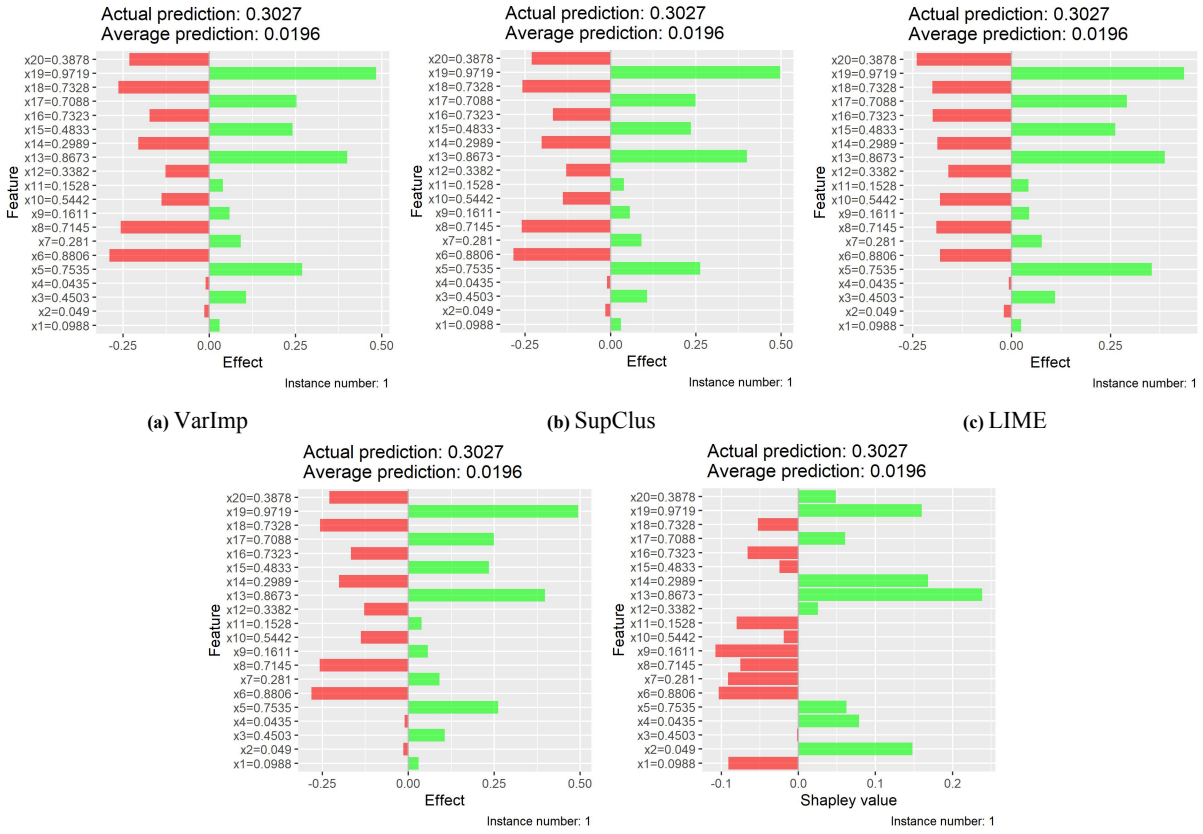
Appendix B. Estimated effects of an instance for artificial datasets.



(a) VarImp **(b)** SupClus **(c)** LIME

(d) IML **(e)** Shapley values

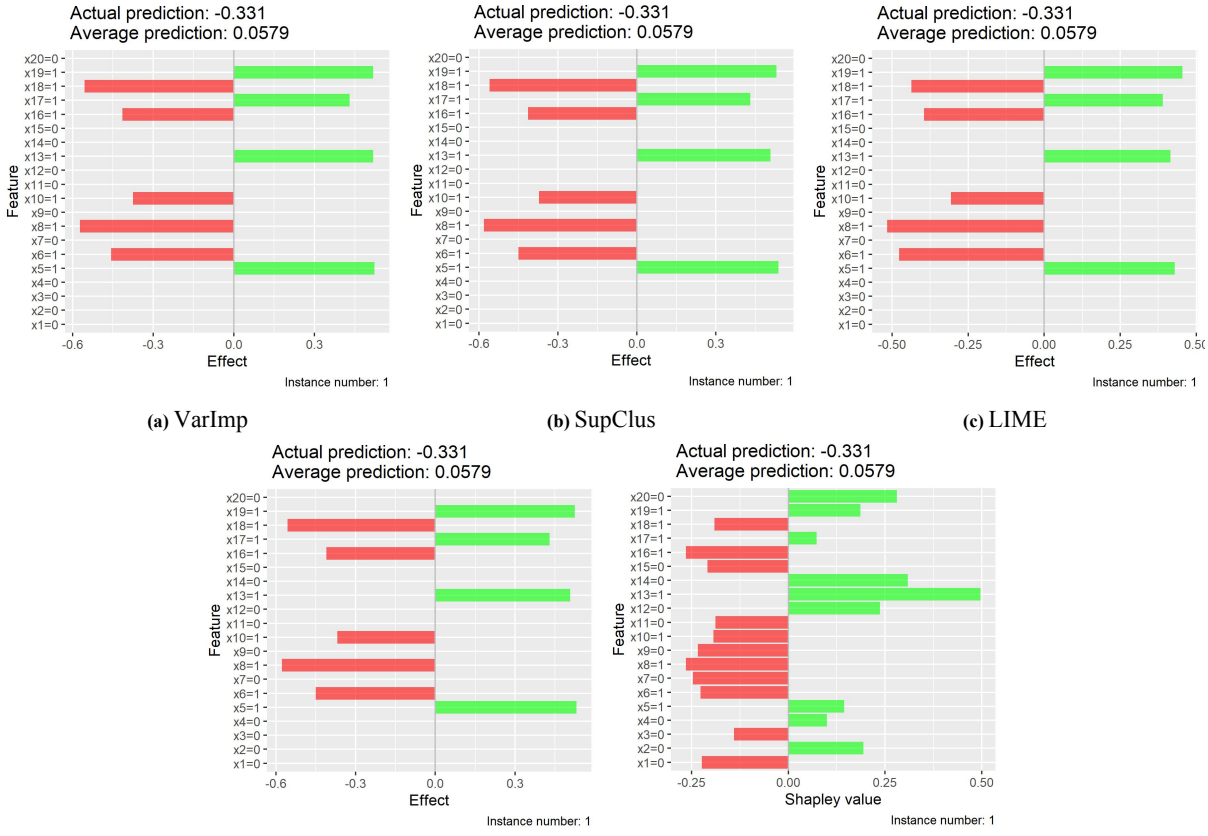
Figure 14. Estimated effects of an instance for dataset 1.



(a) VarImp **(b)** SupClus **(c)** LIME

(d) IML **(e)** Shapley values

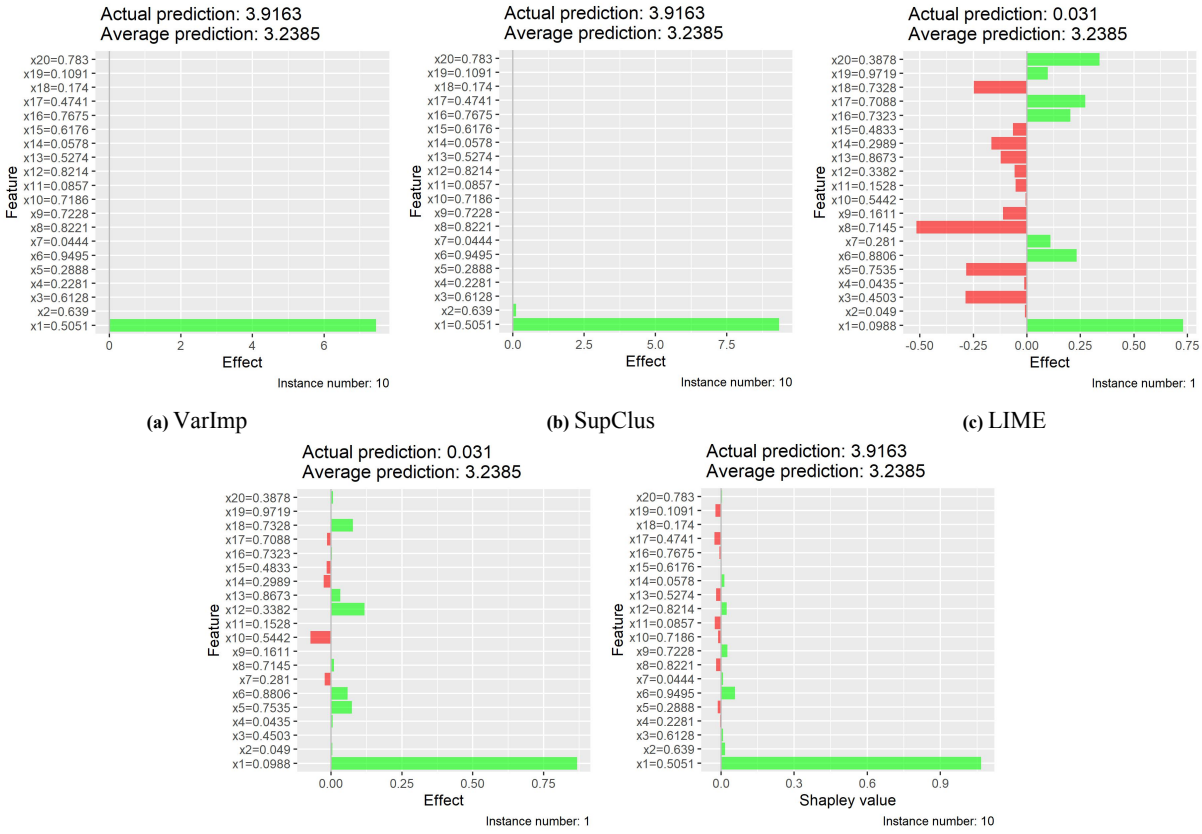
Figure 15. Estimated effects of an instance for dataset 3.



(a) VarImp **(b)** SupClus **(c)** LIME

(d) IML **(e)** Shapley values

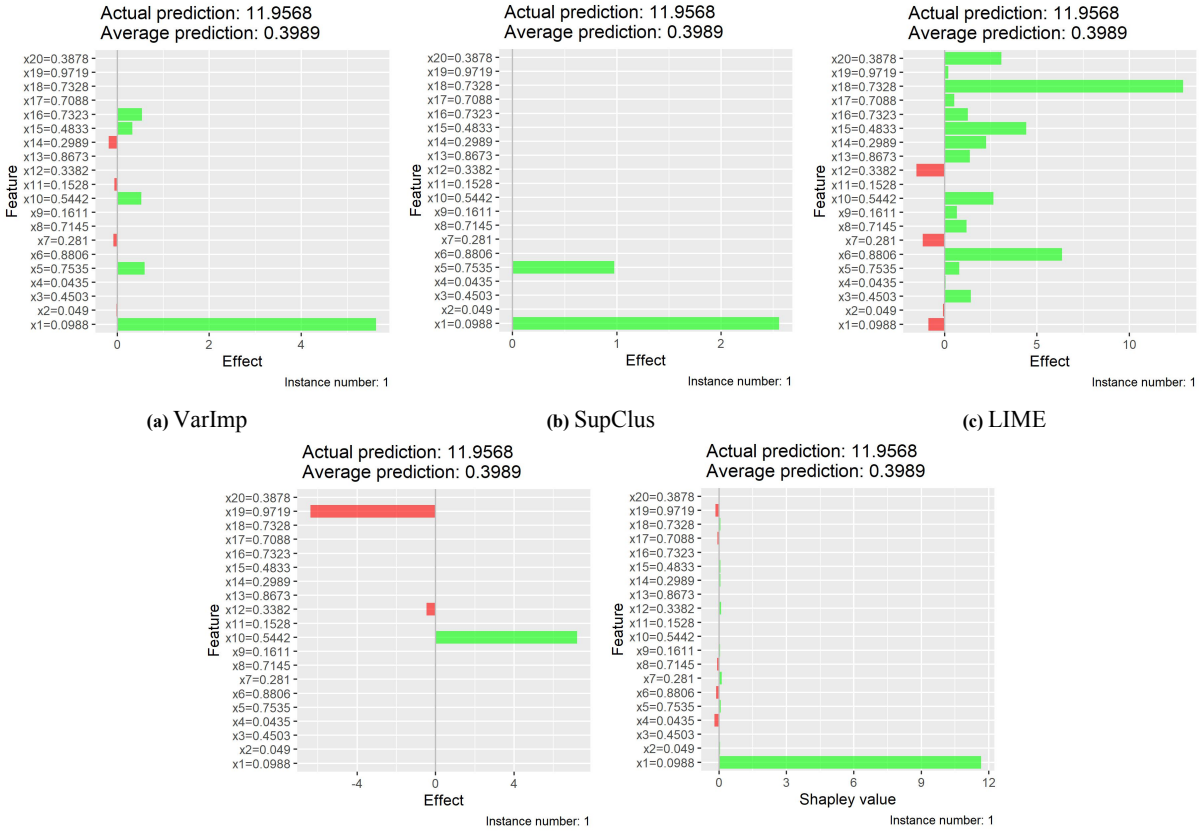
Figure 16. Estimated effects of an instance for dataset 4.



(a) VarImp **(b)** SupClus **(c)** LIME

(d) IML **(e)** Shapley values

Figure 17. Estimated effects of an instance for dataset 5.



(a) VarImp **(b)** SupClus **(c)** LIME

(d) IML **(e)** Shapley values

Figure 18. Estimated effects of an instance for dataset 6.