Handling Missing Values in Local Post-hoc Explainability

Martina Cinquini[1,2](✉), Fosca Giannotti[2,3], Riccardo Guidotti[1,2], and Andrea Mattei[1]

[1] University of Pisa, Pisa, Italy
{martina.cinquini,andrea.mattei}@unipi.it
[2] ISTI-CNR, Pisa, Italy
riccardo.guidotti@isti.cnr.it
[3] Scuola Normale Superiore, Pisa, Italy
fosca.giannotti@sns.it

Abstract. Missing data are quite common in real scenarios when using Artificial Intelligence (AI) systems for decision-making with tabular data and effectively handling them poses a significant challenge for such systems. While some machine learning models used by AI systems can tackle this problem, the existing literature lacks post-hoc explainability approaches able to deal with predictors that encounter missing data. In this paper, we extend a widely used local model-agnostic post-hoc explanation approach that enables explainability in the presence of missing values by incorporating state-of-the-art imputation methods within the explanation process. Since our proposal returns explanations in the form of feature importance, the user will be aware also of the importance of a missing value in a given record for a particular prediction. Extensive experiments show the effectiveness of the proposed method with respect to some baseline solutions relying on traditional data imputation.

Keywords: Explainable AI · Local Post-hoc Explanation · Decision-Making · Missing Values · Missing Data · Data Imputation

1 Introduction

Missing data is a pervasive problem across various domains that arises when some values in a dataset, typically tabular datasets, are unavailable due to factors such as measurement errors, incomplete data collection, or the intrinsic nature of the data [7,14]. The presence of missing values, and therefore the absence of some information, can significantly affect the performance of Machine Learning (ML) models used by Artificial Intelligence (AI) systems for decision-making in these contexts, often resulting in biased outcomes and inferior accuracy [13]. In particular, issues related to missing values are particularly relevant in applications where accurate data is critical for decision-making, such as medical diagnosis, risk assessment, and credit scoring [17,38,46]. Hence, addressing missing values

L. Longo (Ed.): xAI 2023, CCIS 1902, pp. 256–278, 2023.
https://doi.org/10.1007/978-3-031-44067-0_14

is crucial to improve the reliability and usefulness of ML models used by AI systems in real-world scenarios. In the last years, researchers realized various data preprocessing methods to impute missing values [12,29], and designed predictive ML models which can deal by design with datasets affected by missing values such as XGBoost [8], LightGBM [26], and CatBoost [41].

Besides missing values, another issue against which researchers are fighting nowadays is the eXplainability of AI systems (XAI), particularly when ML techniques are employed to model the logic of the AI system in high-stakes decision fields [20,28]. Indeed, some of the most effective ML predictors are considered "black-box" models [20,37] due to their complexity, which causes the non-interpretability of the decision process [28,34]. However, explainability is a fundamental requirement in sensitive domains where the AI system is meant to offer support to experts instead of making decisions for them [15,33].

Even though the current research in XAI is flourishing [1,5,30,50], there is an apparent research vacuum at the intersection between these two issues in AI and ML, i.e., XAI approaches able to deal with missing values. To understand a real practical scenario in which it may be important to have an explanation method also working in the presence of missing values, we can think of a predictive model in the healthcare context that tries to assess the severity of a disease or to recommend a treatment plan. Indeed, models in such contexts are typically trained and applied on incomplete patient data due to missing values [17,38,46]. For instance, the record describing a patient can be incomplete because they cannot undergo particular medical examinations. In such cases, the AI recommendation should be questioned and inspected thoroughly to check the correctness of the decision process. In addition, even if the model's performance may appear promising, the model might be biased towards a particular group of patients due to missing data and thus make incorrect predictions and recommend wrong treatments. Consequently, an explainer working in this context is needed to verify the decision logic learned and applied by the AI system.

Since the literature shows a lack of efforts toward the design of XAI methods able to handle missing values, we extend one of the most widely used and applied post-hoc explanation approaches. In this paper, we propose LIMEMV for Local Interpretable Model-Agnostic Explanations with Missing Values. LIMEMV extends LIME [42] by removing the need for imputing missing data before explaining the record under analysis. Indeed, LIMEMV handle missing values within the explanation process by employing state-of-the-art imputation methods. Specifically, *(i)* we replace the synthetic data generation performed by LIME with a neighborhood generation strategy creating synthetic records with missing data, and *(ii)* we substitute the linear model adopted by LIME with a surrogate model able to handle missing values. As a result, LIMEMV is able to return an explanation in the form of feature importance for a record with missing values, for a predictive model working on missing values, and for considering a dataset with missing values. We highlight that our proposal for a missing-value-compliant explanation method can be easily adapted to extend and improve other model-agnostic explainers like LORE [19], or SHAP [31]. However, we restrict our investigation and the enabling of the native treatment of missing values into the explanation extraction process of LIME due to the easiness of the integration. Our experiments on various datasets show that LIMEMV explanations are relatively similar to those of LIME and that it approximates well the decisions taken by the black-box classifier without impacting the fidelity in mimicking the black-box.

The rest of this paper is organized as follows. Section 2 describes the state-of-the-art related to missing values and XAI. Section 3 formalizes the problem treated and recalls basic notions for understanding our proposal that is defined in Sect. 4. Section 5 presents experimental results. Conclusions, limitations and future works are discussed in Sect. 6.

2 Related Works

In this section, we provide the reader with a brief review of XAI approaches, taking into account missing values and LIME, that is at the basis of our proposal.

In [2] are presented the challenges of imputation in XAI methods showing different settings where AI models with imputation can be problematic, as optimizing for explainability with post-hoc models while simultaneously optimizing for performance via imputation may lead to unsafe results. Our proposal can be adapted to answer many issues raised in this paper. In [25] is confirmed that the presence of missing values is among the common issues faced by data scientists working with explainability. However, despite the presence of many researchers both in the fields of missing values [7,14] and XAI [15,20,28,33,34] there is a clear lack of effort at the intersection of these two fields. To the best of our knowledge, we can refer to decision trees [49] as interpretable-by-design approaches dealing with missing data. Indeed, during training, if an attribute a has missing values in some of the training instances associated with a node, a way to measure the gain in purity for a is to exclude instances with missing values of the records of a in the counting of instances associated with every child node generated for every possible outcome. Further, suppose a is chosen as the attribute test condition at a node. In that case, training instances with missing values of a can be propagated to the child nodes by distributing the records to every child according to the probability that the missing attribute has a particular value. The same can be done at query time. Obviously, such approaches, despite being interpretable, are only sometimes effective for solving complex decision problems. Another possibility for decision trees is the CHAID approach [24] that treats missing values as separate categorical values. Also, the BEST approach [4] selects a certain feature to split the dataset only when in the current partition there are no missing values. Furthermore, CART trees [47] employ recursive partitioning based on feature thresholds to split data into homogeneous subsets. Recently, in [22] has been presented a procedure for data imputation based on different data type values and their association constraints that not only imputes the missing values but also generates human-readable explanations describing the significance of attributes used for every imputation.

Again to the best of our knowledge, there are no post-hoc local explanation approaches able to handle natively missing values. Consequently, we decided to extend LIME [42], the most well-known model-agnostic explainer that returns local explanations as feature importance vectors. Further details about LIME are presented in Sect. 3. Although LIME is effective and straightforward, it has several weak points. A possible downside is the required transformation of any data into a binary format claimed to be human-interpretable. Another aspect worth highlighting is that the random perturbation method results in shifts in data and instability in explanations. Indeed, for the same record and prediction, LIME can generate different explanations over several iterations [51]. This lack of stability is among the main weaknesses of an interpretable model, especially in critical domains [51]. Lastly, in [16] is shown that *additive* explanations like those returned by LIME cannot be trusted in the presence of noisy interactions introduced in the reference set used to extract the explanations.

Over recent years, numerous researchers have analyzed LIME limitations and proposed several subsequent works extending or improving it. Most of the modifications have been in selecting relevant data for training the local interpretable model. For instance, KLIME [21] runs the K-Means clustering algorithm to partition the training data and then fit local models within each cluster instead of perturbation-based data generation around an instance being explained. A weakness of KLIME is that it is non-deterministic, as the default implementation of K-Means picks initial centroids randomly. In [23] is proposed LIME-SUP that approximates the original LIME better than KLIME by using supervised partitioning. Furthermore, KL-LIME [40] adopts the Kullback-Leibler divergence to explain Bayesian predictive models. Within this constraint, both the original task and the explanation model can be arbitrarily changed without losing the theoretical information interpretation of the projection for finding the explanation model. ALIME [45] presents modifications by employing an autoencoder as a better weighting function for the local surrogate models. In QLIME [6], the authors consider nonlinear relationships using a quadratic approximation. Another approach proposed in [44] utilizes a Conditional Tabular Generative Adversarial Network (CTGAN) to generate more realistic synthetic data for querying the model to be explained. Theoretically, GAN-like methods can learn possible dependencies. However, as empirically demonstrated in [9], these relationships are not directly represented, and there is no guarantee that they are followed in the data generation process. In [51] is proposed DLIME, a Deterministic Local Interpretable Model-Agnostic Explanations. In DLIME, random perturbations are replaced with hierarchical clustering to group the data. After that, a kNN is used to select the cluster where the instance to be explained belongs. The authors showed that DLIME is superior to LIME with respect to three medical datasets. We highlight that, besides this deterministic enhancement, clusters may have a few points affecting the fidelity of explanations. In [52] is presented a Bayesian local surrogate model called BAY-LIME, which exploits prior knowledge and Bayesian reasoning to improve both the consistency in repeated explanations of a single prediction and the robustness to kernel settings. Finally, in [10] is presented CALIME, a causal-aware version of LIME that discover causal relationships and exploits them for the synthetic neighborhood generation.

Although a considerable number of solutions proposed to overcome the limitations of LIME, no state-of-the-art variants allow to handle missing values. This research vacuum motivates our interest in developing such a methodology.

3 Setting the Stage

In this paper, we address the problem of designing a XAI method able to solve the *black-box outcome explanation problem* [20] in the presence of incomplete data. A black-box classifier is defined as a function $b : \mathcal{X}^{(m)} \rightarrow \mathcal{Y}$ that maps data instances $x = \{(a_1, v_1), \ldots, (a_m, v_m)\}$ from a feature space $\mathcal{X}^{(m)}$ with m input features (where a_i is the attribute name and v_i is the corresponding value) to a decision y in a target space $\mathcal{Y}$ of size $l = |\mathcal{Y}|$, i.e., y can assume one of the l different labels ($l = 2$ is binary classification, $l > 2$ is multi-class classification). We write $b(x) = y$ to denote the decision y taken by b, and $b(X) = Y$ as a shorthand for $\{b(x) \mid x \in X\} = Y$. A classifier b is *black-box* when its internals are unknown to the observer or they are known but uninterpretable by humans. If b is a probabilistic classifier, we denote with $b_p(x)$ the vector of probabilities for the different labels. Hence, $b(x) = y$ is the label with the highest probability among the l values in $b_p(x)$. In this paper, we assume that *(i)* some values v_i of the records used to train the classifier b can be missing, i.e., $v_i = *$, *(ii)* b can return a decision $y = b(x)$ even when values $v_i = *$ in x are missing. Let $A = \{a_1, \ldots, a_m\}$ be the set of all the features. We name $M(x) = \{a_j | \forall j = 1, \ldots m \wedge v_j = *\}$ the set of features with a missing value for a record x, and $\neg M(x) = A - M(x)$ the set of features for which values are not missing. We write $M(X)$, respectively $\neg M(X)$, as a shorthand to indicate the set of features for which at least a record has a missing value in X. Thus, we can model the input domain of a predictive model b as $\mathcal{X}^{(m)} = (A_1 \cup \{*\}) \times \cdots \times (A_m \cup \{*\})$ where A_i identifies the set of known values for attribute a_i. We complete our formalism using $|X|$ to indicate the size of a dataset X, and X_j to indicate the j^{th} feature, i.e., column, of X.

Given a black-box b and an instance x classified by b, i.e., $b(x) = y$, the *black-box outcome explanation problem* aims at providing an explanation e belonging to a human-interpretable domain. According to the domain, in our work, we focus on feature importance modeling the explanation as a vector $e = \{e_1, e_2, \ldots, e_m\}$, in which the value $e_i \in e$ is the importance of the i^{th} feature for the decision made by $b(x)$. To understand each feature's contribution, the sign and the magnitude of e_i are considered. If $e_i < 0$, the feature a_i contributes negatively to the outcome y; otherwise, the feature a_i contributes positively. The magnitude represents how significant the feature's contribution is to the prediction.

In this context, our aim is to design an explanation method that can return a valid and meaningful explanation e even in the presence of missing values in x and/or X without requiring any a priori imputation.

We keep this paper self-contained by summarizing in the following the key concepts necessary to comprehend our proposal.

3.1 Missing Values Imputation

In statistics [43], the mechanisms of missing values are categorized into three types depending on the relationship between $M(X)$ and $\neg M(X)$.

First, Missing Completely At Random (MCAR) if $M(X)$ is independent of A, i.e., when the probability of a record having a missing value for an attribute does not depend on either the known values or the missing data itself.

Second, Missing At Random (MAR) if $M(X)$ depends only on $\neg M(X)$, i.e., when the probability of a record having a missing value for an attribute may depend on the value of other attributes without missing values. In other words, MAR occurs when the distribution of a record having missing values for an attribute depends on the observed data. Considering missing data as MAR instead of MCAR is a safer assumption since any analysis valid for MAR data, e.g., multiple imputations, is correct also if the data is MCAR [39].

Finally, Missing Not At Random (MNAR) if $M(X)$ depends on $M(X)$, i.e., MNAR occurs when the probability of a record having a missing value for an attribute may depend on the value of that attribute. MNAR data is also called "non-ignorable" since treating it with techniques designed to work on MCAR or MAR, like imputation, will produce misleading results. A peculiar case of MNAR is when data is structurally missing, i.e., data that is missing for a logical reason. A typical example can be a survey where some questions are only asked participants who answered in a certain way to previous questions. In this case, the mechanism is easy to analyze, while MNAR data can pose more of a challenge since the logic behind the missing data might be difficult to understand. We conduct experiments with the MCAR mechanism only, as most researchers are reported doing in the survey in [29]. The extensive adoption of the MCAR approach underlines its credibility and efficacy in addressing missing data, making it a compelling and well-founded choice for our investigations. In future works, we intend to explore also other settings such as MAR or MNAR.

In the following, we summarize two missing values imputation approaches that we adopted as competitors and as a component of our proposal.

K-Nearest Neighbours. k-Nearest Neighbours (kNN) is a supervised ML method widely employed with good results for imputing missing values [29]. KNN identifies the nearest neighbors of an instance based on a distance function, e.g., the Euclidean distance. The distance computation is performed w.r.t. the features in $\neg M(x)$. A majority vote is then conducted among the top k neighbors to determine the most appropriate value for replacing the missing one.

MICE. Multivariate Imputation by Chained Equations (MICE) [3] is a multiple imputation method [36] that can be used whenever missing data is assumed to be MAR or MCAR. MICE works by imputing values in multiple copies of the dataset and then pooling together the results. On each copy of the available data, MICE performs an iterative process in which, at each iteration, a feature in the dataset is imputed using the knowledge of the other attributes. In particular, at each iteration, the first step replaces the missing values in $M(X)$ with placeholder

Algorithm 1: LIME(x, b, X, k, N)

```
Input  : $x$ - instance to explain, $b$ - classifier, $X$ - reference dataset,
         $k$ - nbr of features $N$ - nbr of samples
Output: $e$ - features importance
1  $Z \leftarrow \emptyset, Z' \leftarrow \emptyset, W \leftarrow \emptyset, S \leftarrow \emptyset$;     // init. empty synth data, weights, ad stats
2  for $j \in [1, m]$ do
3      $S \leftarrow S \cup \{(\mu(X_j), \sigma(X_j))\}$;     // compute statistics
4  for $i \in [1, \ldots, N]$ do
5      $z \leftarrow sampling(x, S)$;     // random permutation
6      $z' \leftarrow \{(a_j, \mathbb{1}(x_{v_j} = z_{v_j}))|j = 1, \ldots, m\}$;     // features changed
7      $Z \leftarrow Z \cup \{z\}$; $Z' \leftarrow Z' \cup \{z'\}$;     // add synthetic instance
8      $W \leftarrow W \cup \{exp(\frac{-\pi(x,z)^2}{\sigma^2})\}$;     // add weights
9  $e \leftarrow solve\_Lasso(Z', b_p(Z), W, k)$;     // get coefficients
10 return $e$;
```

values that do not consider the other features, e.g., the mean of the available data for that attribute or random values. Then, let $X' \subset X$, for each attribute $a \in M(X')$, MICE imputes it with a linear regression model trained on another slice of the dataset $X'' \subset X$ such that $a \in \neg M(X'')$. An iteration is completed when all the features are processed. This process is repeated up to a user-specified number of times or until convergence is reached.

3.2 LIME

A widely adopted, local, model-agnostic, post-hoc explanation method is LIME (Local Interpretable Model-Agnostic Explanations) [42], which acts as a foundation for our proposal. The main idea of LIME is that the explanation can be derived locally from records generated randomly in the synthetic neighborhood Z of the instance x that has to be explained.

Algorithm 1 illustrates the pseudo-code of LIME. In line 1, two empty sets Z and Z' are initialized. Z will be populated with the synthetic data sampled around the instance x represented in the real domain, while Z' will contain a representation of the synthetic records in Z in a binary version signaling the features that have been changed, i.e., given $z' \in Z'$ and $z \in Z$, the value of the j^{th} features in z' is equal to one ($z'_{v_j} = 1$) if $z_{v_j} = x_{v_j}$, $z'_{v_j} = 0$ otherwise. The vector W will contain the weights associated with the records Z generated, expressed in terms of their distance from x. S will contain the statistics of every feature j with $j = \{1, \ldots, m\}$ where m is the number of features. Indeed, the loop in lines 2–3 populates S with the mean μ and standard deviation σ of every feature X_j. Subsequently, LIME runs N times a loop (lines 4–8), populating Z, Z' and W at each iteration with a new synthetic instance. LIME randomly samples N instances similar to x by drawing values according to the statistics S (line 5). The function $\mathbb{1}$(*condition*) in line 6 returns one when the *condition* is verified, zero otherwise. It highlights how LIME creates the binary version Z' of the synthetic records Z. Then, LIME weights proximity of the records z' with

x w.r.t. a certain distance function π and store the result in W (line 8). Finally, LIME adopts the perturbed sample of instances Z to fed to the black-box b and obtain the classification probabilities $b_p(Z)$ with respect to the class $b(x) = y$. The binary synthetic instances Z' with the weights W are used to train a linear regressor with Lasso regularization using the classification probabilities $b_p(Z)$ as the target variable and considering only the top k most essential features (line 9). The k coefficients of the linear regressor are returned as explanation e.

4 Local Explainability with Missing Values

We present LIMEMV (Local Interpretable Model-Agnostic Explanations with Missing Values). LIMEMV extends LIME [42] with the ability to handle incomplete data during the explanation process. This eliminates the need of imputing missing values both on the training dataset and on the records for which the explanation is required. The presence of the missing values in X impacts the calculus of the statistics S used to generate the synthetic neighborhood (line 3, Algorithm 1), while missing values in the record x to be explained impacts the sampling function generating the synthetic records z (line 5, Algorithm 1). LIMEMV is able to deal with both of these issues.

Before outlining the details of LIMEMV, we aim at clarifying when this approach is crucial. Given a dataset X with missing values, a user can decide *(i)* to adopt a model b which is not able to handle missing values, such as a SVM or a Neural Network, *(ii)* to use a model b able to handle missing values, such as XGBoost and LightGBM. In the first case, in order to train b, the user needs to preprocess X by applying a data imputation approach. As a consequence, given a test record x possibly having missing values, the same imputation approach should be applied on x before querying b to obtain the decision $y = b(x)$. Thus, if an explanation e is desired for the decision $y = b(x)$, the classic LIME approach can be used. Instead, in the second case, the dataset X can be directly used to train b, and any test record x can be passed to b without requiring any data imputation. However, if an explanation e is desired, for the decision $y = b(x)$, the classic LIME approach cannot be used, as it cannot work in the presence of missing data. A naive solution consists in applying an imputation approach on X and x before passing them to LIME (see Algorithm 1). However, in this case, the explainer is applied to a dataset and on a record that differ from those adopted by the decision model b. On the other hand, with LIMEMV, the user does not need to apply any imputation approach, and it can be used directly to obtain the explanation e for the decision $y = b(x)$ in the presence of missing data.

The pseudo-code of LIMEMV is reported on Algorithm 2, with the main differences from LIME highlighted in blue. In the following, we detail such differences. Also, Fig. 1 visualizes with an example the various steps of LIMEMV.

4.1 Input Parameters

First, we can notice that *(i)* LIMEMV does not require the user to specify the number of important features k as these are identified by design by the surrogate

```
Algorithm 2: LIMEMV($x$, $b$, $X$, $k$, $N$)
Input : $x$ - instance to explain, $b$ - classifier, $X$ - reference dataset,
        $N$ - nbr of samples, $\psi$ - imputation function
Output: $e$ - features importance
1  $Z \leftarrow \emptyset, W \leftarrow \emptyset, S \leftarrow \emptyset$;        // init. empty synth data, weights, ad stats
2  for $j \in [1, m]$ do
3      $X'_j \leftarrow \{(a_i, v_i) | \forall i = 1, \dots, |X| \wedge v_i \neq *\}$;     // consider only non missing values
4      $S \leftarrow S \cup \{(\mu(X'_j), \sigma(X'_j), 1 - |X'_j|/|X|)\}$;        // compute statistics
5  for $i \in [1, \dots, N]$ do
6      $z \leftarrow sampling\_imputation(x, \psi_X, S)$;     // random permutation with imputation
7      $Z \leftarrow Z \cup \{z\}$;                           // add synthetic instance
8      $W \leftarrow W \cup \{exp(\frac{-\pi'(x,z)^2}{\sigma^2})\}$;              // add weights
9  $T \leftarrow train\_tree(Z, b_p(Z), W)$;                  // train regressor tree
10 $e \leftarrow tree\_feature\_imp(x, T)$;                   // get coefficients
11 return $e$;
```

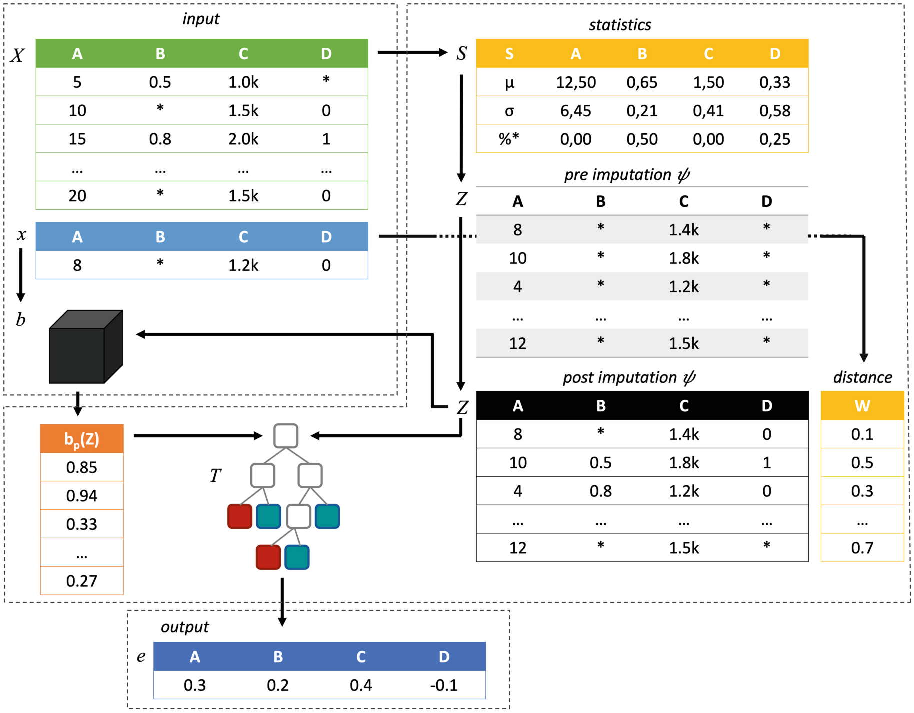
model adopted; *(ii)* it requires as input an imputation function ψ, i.e., a function that given a dataset X fills the missing values using a certain strategy. Examples of ψ functions are kNN [29] and MICE [3]. Other naive approaches may consist in using the mean (or the mode) of each feature to fill in missing values.

4.2 Dataset Statistics

The next difference is in the loop computing the statistics (lines 3–4). Indeed, rather than of calculating the mean and standard deviations for the complete set of features X_j, it calculates them on a subset $X'_j \subseteq X_j$ such that X_j only contains not missing values (as formalized in line 3). This setting solves the possible presence of the missing values in X. Another difference w.r.t. LIME is an addition to the set of computed statistics, i.e., the information about the distribution of missing values in each attribute. Since a priori we need to assume MCAR data, this boils down to the relative number of missing values for each feature, i.e., $1 - |X'_j|/|X|$. However, when dealing with MAR data, information about the relationships with other features can be exploited if available. Figure 1 shows an example of S content resulting from a dataset.

4.3 Synthetic Neighborhood Generation

The knowledge stored in S is then applied when generating the synthetic neighborhood in the subsequent loop (lines 5–8) that is responsible for the synthetic neighborhood generation, where the sampling function has been changed w.r.t. LIME (line 6, Algorithm 2) to fix the possible presence of missing values in the record x. The problem we face is relative to how to sample values around a coordinate that is absent from x. A naive strategy consists in generating the N



A	B	C	D
5	0.5	1.0k	*
10	*	1.5k	0
15	0.8	2.0k	1
…	…	…	…
20	*	1.5k	0

A	B	C	D
8	*	1.2k	0

S	A	B	C	D
μ	12,50	0,65	1,50	0,33
σ	6,45	0,21	0,41	0,58
%*	0,00	0,50	0,00	0,25

A	B	C	D
8	*	1.4k	*
10	*	1.8k	*
4	*	1.2k	*
…	…	…	…
12	*	1.5k	*

A	B	C	D
8	*	1.4k	0
10	0.5	1.8k	1
4	0.8	1.2k	0
…	…	…	…
12	*	1.5k	*

W
0.1
0.5
0.3
…
0.7

$b_p(Z)$
0.85
0.94
0.33
…
0.27

A	B	C	D
0.3	0.2	0.4	-0.1

Fig. 1. LIMEMV takes as input the reference dataset X, the instance to explain x and the black-box b, and returns as output a feature importance explanation e. The workflows highlights the statistics S calculated considering missing values and shows the synthetic neighborhood Z before and after the imputation with ψ. Finally, e is returned as feature importance extracted from a local regressor tree using as target variable the probability $b_p(x)$ for the decision $y = b(x)$.

synthetic neighbors Z only considering the features in $\neg M(x)$. This would practically remove those attributes from the explanation, thus preventing the user from understanding the impact of features with missing values. On the other hand, the *sampling_imputation* adopted by LIMEMV works as follows. The values for the features in $\neg M(x)$ are drawn as in the classic approach exploiting the means and standard deviations in S, while for the features in $M(x)$ the values are set as missing $*$. After that, an imputation function ψ is used to fill a number of missing values in z proportionate with the ratio stored in S, i.e., $1 - |X_j|/|X|$ for feature j. We implemented ψ as kNN [29] and MICE [3]. In other words, with LIMEMV we obtain a set Z of synthetic records where the features without missing values in $\neg M(x)$ are randomly sampled around the observed values or left the same, while some of the records of some of the features with missing values in $M(x)$ are filled w.r.t. the records in X, i.e., with plausible values for non missing features. Hence, in this way, the imputation is performed exclusively at explanation time and to generate a plausible synthetic dataset in the proximity of x and respect the missing values in X. Figure 1 shows an example of a synthetic neighborhood Z before and after the application of the imputation function ψ. We notice that the number of missing values per feature remains coherent with those in the observed dataset X and captured by S.

Like in LIME, the importance of the synthetic records in Z is stored in W and it is evaluated w.r.t. their proximity with x. Differently from LIME, since the synthetic records z have missing values, we employ a function (π') calculating the Euclidean distances in the presence of missing values by ignoring features with a missing value in both x and z and scaling the result as m divided by the features without missing values [11], i.e., $|(\neg M(x)) \cap (\neg M(z))|$.

4.4 Local Interpretable Surrogate Model

At this stage, differently from LIME, the synthetic neighborhood Z contains missing values. As a consequence, the linear Lasso regression model cannot be used as it is not capable of handling training sets containing missing values. Thus, inspired by [19], we decided to employ a decision tree T that is able to deal with missing data by design [49] (line 9). As a side effect, in LIMEMV there is no need for the user to specify the number of important features k for which the explanation is required as the explanation e is going to be formed only by the features appearing in the branch of T responsible for the decision on x. However, differently from LIME instead of training the surrogate tree regressor T on Z', i.e., the binary version of Z modeling the changes w.r.t. x, like in [19], we train the surrogate T directly on Z, permitting in this way to understand in terms of values, and not in terms of presence/absence, the dependencies between the features Z and the prediction probability of the target label $b_p(Z)$.

4.5 Explanation with Missing Values

Finally, LIMEMV extracts the explanation e of x in terms of feature importance with the function *tree_feature_imp* as follows. First, as the magnitude of the feature importance e_j, it is used the normalized total reduction of the impurity criterion brought by the j^{th} feature, i.e., the Gini importance[1] [49]. Second, as the sign of the feature importance e_j, LIMEMV adopts the sign of the difference between the average value on the q^{th} node in the tree (with $q = 0$ indicating the root) and the subsequent one w.r.t. the path from the root to the leaf followed on T for the prediction of the record x, i.e., $sign(T_q(x) - T_{q+1}(x))$, where $T_q(x)$ indicates the average value on the q^{th} node T for the prediction of x. Thus, each feature j receives a score that depends on the decision path followed on x. We should note that, while considering e_j could be reasonable, it requires more investigation and might also be of interest outside the missing data domain.

For example, suppose that a certain local surrogate tree T for the record x in the root separates the data using the attribute a_j =*age*. If the normalized

[1] The Gini importance could be biased regarding cardinality as pointed out in [48] but its effect is mitigated from the normalization.

Gini importance of a_j is 0.4, and, for x, $T_0(x) - T_1(x) < 0$, than, we will have $e_j = -0.4$, meaning that *age* has a negative contribution of 0.4 in the decision taken by b on x. We underline that, according to our definition, the value of *age* for x might also be unspecified in this setting. However, we still have access to its local importance for the decision $y = b(x)$.

5 Experiments

We report here the experiments carried out to validate LIMEMV[2]. We present the evaluation measures adopted, the datasets used, the experimental setup, and the explainers selected as baselines. Then, we demonstrate that LIMEMV outperforms LIME used in pipeline with standard imputation approaches. Since it is not generally possible to access the ground truth for explanations [18], we decided to adopt a controlled experiment to check the validity of the explanations returned by LIMEMV and by the baseline competitors to judge their effectiveness.

In particular, we adopt datasets without missing data in which we insert missing values in a controlled way. Formally, let X be the original dataset, and $\tilde{X}$ the same dataset where some records are modified by inserting missing values for certain features according to a procedure detailed in the following sections. Let b be the black-box able to deal with missing trained on X and $\tilde{b}$ the same black-box trained on $\tilde{X}$. Also, let x be a record to predict and explain and $\tilde{x}$ the same record but with some missing values. Given an explanation method *expl* that is implemented in the experiments by LIMEMV or by one of the baselines, we name e and $\tilde{e}$ the feature importance explanations returned by $expl(x, b, X)$ and $expl(\tilde{x}, \tilde{b}, \tilde{X})$, respectively. Then, by comparing sets of e and $\tilde{e}$, i.e., the explanations obtained for records with and without missing values, we can establish the impact of the treatment of the missing values in the explanation process: the lower the discrepancy between the explanations, the less impactful is the treatment of the missing values made by the explainer.

5.1 Evaluation Measures and Explanations Normalization

In order to compare explanations expressed as feature importance, we normalize the magnitude e_i of the values present in each explanation e. Given an explanation e, we aim at guaranteeing that the following property holds $\sum_{j=1}^{m} |e_j| = 1$. Thus, we normalize the value e_j obtaining the normalized value e'_i as

$$e'_i = e_i / \sum_{j=1}^{m} |e_j|.$$

We underline that this normalization is useful not only to compare explanations, but also to make the explanations more intuitive for human users. In the following, we assume that all the explanations returned by the different explainers tested are normalized as described in this section.

[2] The implementation is available here: https://github.com/marti5ini/LIMEMV..

Table 1. Datasets statistics and classifiers accuracy. Specifically, we present the number of samples of each dataset (n), the number of features (m), the number of labels that can assume the class (l), the number of training records for the black-box (X_b) and the number of records for which we seek predictions (X_t). Additionally, we report the accuracy of the black-box without missing values (b) and with missing values ($\tilde{b}$) across various levels of missingness (p).

	n	m	l	$\lvert X_b\rvert$	$\lvert X_t\rvert$	b	$\tilde{b}$				
							$p = 10$	$p = 20$	$p = 30$	$p = 40$	$p = 50$
`adult`	32561	13	2	2600	50	.88	.87 ± .01	.87 ± .01	.70 ± .01	.87 ± .01	.87 ± .01
`compas`	6907	11	2	1381	50	.81	.79 ± .01	.79 ± .02	.80 ± .01	.79 ± .01	.79 ± .02
`diabetes`	768	8	2	154	50	.73	.73 ± .02	.70 ± .03	.71 ± .04	.69 ± .02	.70 ± .04
`fico`	10459	22	2	1822	50	.67	.70 ± .01	.70 ± .00	.70 ± .00	.70 ± .01	.69 ± .01
`german`	1000	20	2	200	50	.83	.79 ± .02	.77 ± .03	.77 ± .02	.77 ± .04	.76 ± .02
`iris`	150	4	3	30	30	1.0	.96 ± .03	.99 ± .02	.95 ± .08	.98 ± .03	.96 ± .04
`titanic`	715	4	2	143	50	.78	.77 ± .02	.78 ± .02	.76 ± .02	.76 ± .02	.76 ± .02

Given a couple of normalized explanations with and without missing values e and $\tilde{e}$, we adopted the *Cosine Similarity* (CS) [49] and the Kendall Tau (KT) [27] to measure their similarity. The CS ranges in $[-1, 1]$, the closer to one the better it is. In addition, inspired by [16,32], we measure the discrepancy between two explanations by calculating the *Absolute Deviation* (AD) as feature-wise and record-wise means of the vector of differences $\delta = \{|e_1 - \tilde{e_1}|, \ldots, |e_m - \tilde{e_m}|\}$. The AD ranges in $[0, +\infty]$, the closer to zero the better it is. In particular, we group the features to analyze the differences between the features contained in $M(\tilde{X})$ versus those contained in $\neg M(\tilde{X})$. We use ADW to indicate the AD for features *W*ith missing values, and ADO for the AD of features with *O*ut missing values.

Furthermore, in line with the literature in XAI [5,20], we measure the *Fidelity* (FI) of the local surrogate models in approximating the behavior of the black-box as the difference between the predicted probability of the black-box for the decision, i.e., $b_p(x)$, and the prediction of the surrogate, i.e., T. We measure the FI as $1 - |b_p(x) - T(x)|$ such that it is in $[0, 1]$, the closer to one the better it is.

Finally, we also report the *Explanation Time* (ET) expressed in seconds.

5.2 Datasets and Experimental Setting

We experimented on seven benchmarking datasets from UCI Machine Learning Repository and Kaggle[3], namely `iris`, `titanic`, `adult`, `german`, `diabetes`, `compas`, and `fico`, which belong to diverse yet critical real-world applications. These datasets have very different properties in terms of number of records and features and type of features, i.e., their attributes are numeric, categorical, or mixed. Table 1 (left) presents a summary of each dataset. The datasets are pre-processed by removing all the records with missing values, and normalized using

[3] https://archive.ics.uci.edu/ml/index.php, https://www.kaggle.com/.

the Z-Score normalization [49]. Categorical features are label encoded. We split each dataset into two partitions: X_b, is the set of records to train the black-box models b and $\tilde{b}$ when trained on the training with and without missing values, respectively, while is X_t the partition that contains the records for which we want a prediction b and $\tilde{b}$ and an explanation from the explainers detailed in the following section. We highlight that both X_b and X_t are used at training and explanation time in the two versions with and without missing values. We underline that X_b is used to train the black-box but also by LIMEMV and by the baselines tested to gather information to generate the synthetic neighborhood.

Our objective is to re-create a scenario in which missing values are present both in the observable data and in the records for which the explanation is required[4]. In this work, we experiment with the MCAR setting, which is typically assumed in the presence of missing values when additional knowledge is unavailable. We leave the study on MAR and MNAR for future work.

Since often the most important features for individual predictions are in overlap with features globally important for the classifier, we aim at stressing the experimental scenario by inserting missing values among the features most important globally. Thus, for each dataset, we train a Random Forest (RF) classifier on X_b with default hyper-parameters. We exploit the RF to obtain a ranking of the m features $\{j_1, \dots, j_m\}$, where j_r says that the j^{th} features is ranked r^{th} w.r.t. its importance, *which is determined using Gini importance*. Thus, we randomly select $p\%$ features among the most important ones with respect to the ranking obtained with $p \in \{10, 20, 30, 40, 50\}$, i.e., $|M(X)| = |X| * p/100$. Then, for each feature in $M(X)$ we select the percentage q of missing values with $q \in \{4, 8, 16, 32\}$. Hence, we are able to observe the impact of different configurations of missing values in the explanation process.

As black-box we trained an XGBoost [8] implemented as the `xgboost` library[5] using default parameters both in the training set with and without missing values to avoid possible biases. The partitioning sizes and the classification accuracy in presence of missing values and without them are in Table 1 (right). When in presence of missing values for a certain percentage of features p, it is reported the average accuracy w.r.t. the various percentages of missing values in the features q. We notice that, even in presence of missing values with various p, the accuracy of the various black-boxes $\tilde{b}$ remains close to the accuracy of b.

5.3 Baseline Explainers

We compare LIMEMV against some naive approaches that can be adopted to solve the problem faced in this paper without requiring a novel implementation. These solutions consist in using *a data imputation approach* on the dataset X_b and on

[4] In preliminary experimentation considering missing values present only in the observable data X_b or only in the explained records X_t we noticed that the overall performance are similar to those reported in this paper. Thus we preferred to illustrate and discuss results only for the most realistic and complex scenario.

[5] https://xgboost.readthedocs.io/en/stable/.

the test record x, and then relying on the explanation returned by the classic LIME version since there are no missing values disturbing the explanation process. As data imputation approaches we experiment with the mean value of the feature with missing values, kNN, and MICE. We adopt the names LB-M, LB-K, and LB-C, to refer to these baseline explainers relying on mean, kNN, and MICE imputations and LIME, respectively. On the other hand, we use LMV-K and LMV-C to refer to the two versions of LIMEMV implementing the imputation function ψ with kNN and MICE, respectively. For future work, it could be also interesting to investigate LIME with a tree-based local model and pre-hoc imputation as a competitor. To make the LIME and LIMEMV methods comparable, since LIMEMV automatically selects the most important features that will appear with non-zero features importance, for LIME we set $k = m$ such that all the features are considered by the surrogate Lasso regressor. For all the explainers we keep the size of the synthetic neighborhood as in the original LIME implementation, i.e., $N = 5000$. We remark on the fact that with p and q we refer to the percentages of features among the most important ones and those with missing values and they do not impact with k.

5.4 Case Study Explanation

Before presenting the experimental results, we show in Fig. 2 a case study explanation for a record of the `adult` dataset where missing values are inserted among $p = 50\%$ of the most important features according to a RF and such that there are at least $q = 20\%$ missing values for each feature with missing. The features' importance of the explanations is reported as bars for the features having a value e_j different from zero. Thus, the taller the bar, the higher the magnitude of the feature importance e_j. For completeness, we also report the values. The plot on the left shows the feature importance returned by LIME using kNN as data imputation method at the preprocessing time, while the one on the right shows the feature importance returned by LIMEMV using kNN as an imputation function ψ.

In this particular example, the record has two missing values for the attributes *age* and *relationship*. By comparing the two plots, we immediately realize two aspects. First, due to the usage of the Lasso regressor as a local surrogate, LIME returns much more features than LIMEMV with non-zero feature importance e_j[6]. On the other hand, the local surrogate tree regressor adopted by LIMEMV is able to identify by-design the most important features, and indeed, due to the experimental setting adopted, among them, we find also *age* and *relationship*. Second, we visually see a clear discrepancy between the feature importance of the explanations with and without missing values when LIME or LIMEMV are adopted. Indeed, LIMEMV is considerably more adherent than LIME to the explanation without missing values as to *capital-gain* is assigned almost

[6] Such an outcome is due to the choice of $k = m$. However, regardless of how we set k, the same result occurs when k is smaller than m and greater than the minimum number of features required to obtain a high-performing linear regressor surrogate.

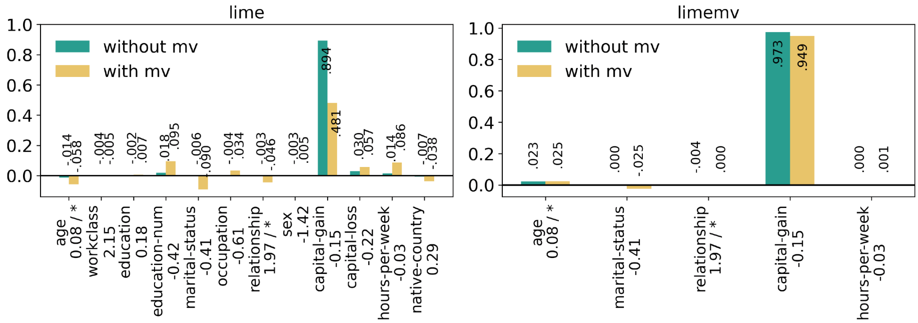


Fig. 2. Explanations of a record of the `adult` dataset with and without missing values for LIME using kNN as imputer at preprocessing time, and LIMEMV using kNN as imputation function ψ. Normalized/missing values (*) of the record are on the x-axis.

the same value. Furthermore, also for the features with missing values, we notice a minor discrepancy for the explanation of LIMEMV: *age* passes from 0.023 to 0.025 for LIMEMV while it changes from −0.14 to −0.58 for LIME, *relationship* passes from −0.004 to 0.00 for LIMEMV while it changes from −0.003 to −0.046 for LIME.

In the next section, we observe numerically these phenomena on various datasets and with various settings for inserting missing values.

5.5 Results

According to our experimental setting, we are able to evaluate the explanations with the measures previously presented for each record in X_t of each dataset, for each explanation method, and for each couple of parameters p and q tested. Table 2 reports the mean and standard deviations for the various settings where the measures obtained for the local explanations of the records in X_t of each setting are aggregated using the interquartile range mean, i.e., the mean of the values in the range defined by the 25^{th} and 75^{th} percentile. The score of the best performer for each dataset and measure is highlighted in bold.

We immediately realize that LMV-K exhibits superior performance in all qualitative measures for `adult`, `compas`, and `titanic`. Furthermore, LMV-K is always the best performer in terms of ADO, i.e., it is the explainer treating missing values with a smaller impact on features without missing values. This characteristic is particularly significant since missing values are usually a minority among the records in a dataset, and therefore, their importance should remain unchanged regardless of their presence or absence. On the contrary, LIMEMV adopting MICE, i.e., LMV-C, is often among the worst in terms of ADO. This underlines how the choice of a certain imputation function ψ can affect the explanation process that is not necessarily the best with more advanced imputation functions.

Concerning the similarity measures CS and KT we notice that there is not a clear winner. Indeed, regarding CS for three datasets, the best approach is LMV-

Table 2. Mean and standard deviation of the evaluation measures observed for each dataset, explainer, and setting of missing values w.r.t. the number of features with missing and percentage of missing values. The best performer is highlighted in bold.

		CS ↑	KT ↑	ADW ↓	ADO ↓	FI ↑	ET ↓
`adult`	LB-M	.024 ± .38	.271 ± .05	.299 ± .20	.029 ± .02	.691 ± .11	.047 ± .00
	LB-K	.465 ± .14	.396 ± .04	.211 ± .09	.026 ± .01	.796 ± .02	.048 ± .00
	LB-C	.132 ± .28	.304 ± .03	.276 ± .16	.027 ± .01	.697 ± .07	.577 ± .00
	LMV-K	**.770 ± .06**	**.437 ± .08**	**.112 ± .05**	**.012 ± .01**	**.834 ± .02**	8.75 ± 1.08
	LMV-C	.071 ± .28	.026 ± .06	.317 ± .18	.031 ± .01	.697 ± .07	**.040 ± .00**
`compas`	LB-M	.314 ± .11	.243 ± .09	.176 ± .04	.072 ± .02	.870 ± .02	.059 ± .01
	LB-K	.360 ± .07	.248 ± .07	.194 ± .05	.041 ± .01	.871 ± .04	.059 ± .01
	LB-C	.258 ± .06	.206 ± .05	.210 ± .05	.057 ± .01	.869 ± .02	.210 ± .01
	LMV-K	**.380 ± .07**	**.387 ± .07**	**.071 ± .10**	**.028 ± .02**	.858 ± 0.03	1.82 ± .26
	LMV-C	.311 ± 0.07	.119 ± .06	.301 ± .10	.058 ± .01	**.871 ± .02**	**.039 ± .00**
`diabetes`	LB-M	.235 ± .15	.143 ± .14	.284 ± .05	.087 ± .02	.772 ± .03	.037 ± .00
	LB-K	.315 ± .12	.228 ± .12	.266 ± .04	.060 ± .01	.771 ± .03	.041 ± .00
	LB-C	**.316 ± .12**	.215 ± .13	**.261 ± .04**	.069 ± .01	.774 ± .02	.099 ± .00
	LMV-K	.248 ± .17	**.303 ± .10**	.364 ± .17	**.030 ± .03**	**.809 ± .05**	.227 ± .03
	LMV-C	.210 ± .07	.165 ± .08	.400 ± .14	.076 ± .02	.774 ± .02	**.035 ± .00**
`fico`	LB-M	.340 ± .14	.334 ± .08	.094 ± .03	.023 ± .00	.816 ± .03	.086 ± .00
	LB-K	.681 ± .11	.466 ± .06	.066 ± .02	.018 ± .00	.842 ± .02	.093 ± .00
	LB-C	**.731 ± .07**	**.500 ± .05**	**.062 ± .01**	.018 ± .00	**.844 ± .02**	1.56 ± .00
	LMV-K	.491 ± .52	.129 ± .17	.223 ± .11	**.010 ± .01**	.781 ± .03	3.19 ± .40
	LMV-C	.273 ± .20	.047 ± .05	.141 ± .04	.038 ± .01	**.844 ± .02**	**.081 ± .00**
`german`	LB-M	.483 ± .10	.405 ± .07	**.079 ± .01**	.027 ± .00	.832 ± .03	.063 ± .00
	LB-K	.521 ± .11	.433 ± .09	**.079 ± .01**	.025 ± .00	.838 ± .02	.074 ± .00
	LB-C	**.526 ± .11**	**.436 ± .08**	.079 ± .02	.024 ± .00	.851 ± .03	.547 ± .00
	LMV-K	.445 ± .13	.396 ± .11	.126 ± .03	**.012 ± .01**	.849 ± .03	.345 ± .04
	LMV-C	.398 ± .07	.210 ± .04	.127 ± .02	.034 ± .01	**.851 ± .01**	**.059 ± .00**
`iris`	LB-M	.565 ± .24	.317 ± .22	.285 ± .11	.162 ± .06	.758 ± .07	.063 ± .00
	LB-K	**.619 ± .18**	**.503 ± .11**	**.232 ± .09**	.087 ± .03	.791 ± .04	.064 ± .00
	LB-C	.599 ± .22	.450 ± .16	.252 ± .11	.097 ± .02	.698 ± .06	.114 ± .00
	LMV-K	.525 ± .36	.446 ± .28	.337 ± .20	**.050 ± .04**	**.834 ± .10**	.110 ± .00
	LMV-C	.456 ± .17	.377 ± .12	.269 ± .13	.134 ± .05	.698 ± .06	**.061 ± .00**
`titanic`	LB-M	.156 ± .17	.151 ± .19	.558 ± .09	.168 ± .06	.618 ± .07	.053 ± .00
	LB-K	.172 ± .14	.157 ± .14	.542 ± .07	.113 ± .05	.658 ± .05	.054 ± .00
	LB-C	.151 ± .12	.133 ± .16	.541 ± .07	.110 ± .04	.630 ± .07	.080 ± .00
	LMV-K	**.182 ± .11**	**.168 ± .16**	**.521 ± .09**	**.100 ± .11**	**.678 ± .05**	.232 ± .02
	LMV-C	.086 ± .18	.075 ± .13	.663 ± .14	.116 ± .05	.630 ± .07	**.051 ± .00**

K, for the other three is LB-C, and for one is LB-K. On the other hand, for KT, LMV-K is the winner on four datasets, LB-C on two, and for one dataset LB-K. The insights from this analysis are the following. First, relying only on the mean at a preprocessing time does not guarantee at all coherence for explanations, and using LMV-C can be even worse. Some approaches are favoring similarity among the scores (measured in terms of CS), while others are favoring that the ordering of the scores, i.e., the order of the importance, is respected. Overall, adopting kNN as an imputation function is a reliable solution at a preprocessing time with LIME but is even better at explanation time with LIMEMV.

Regarding the absolute deviation with missings (ADW), the situation is even more unclear as it is considerably difficult to leave untouched the level of importance of a feature when the value is missing. A possible future research direction might re-frame this measure into a loss function and learn an explanation model from simulated situations of missing values (like the proposed experiment) such that these errors can be avoided by relying on the other features with values to estimate the importance of the features with missing values.

For the fidelity (FI) of the local surrogate, we observe that the proposed approaches in the LIMEMV family have always the best results. This is probably due to the usage *(i)* of the regressor tree that is better in approximating the behavior of the black-box, *(ii)* of a synthetic neighborhood that includes missing values and resembles the real data where the black-box is trained and applied.

Finally, for the explanation time (ET), we observe that LMV-K is the slowest approach compared to the others that always have an ET smaller than a second for explaining a single instance. This is caused by the kNN imputation approach that is applied for each instance in the synthetic neighborhood Z having at least a missing value and every time it needs to calculate the distance with all the other synthetic records in Z. Since the size of Z is $N = 5000$, this causes a not negligible increment in the ET w.r.t. the other explainers for large datasets.

In Fig. 3 and Fig. 4, we observe the impact of the different percentages of features with missing values (p) and different percentages of missing values in features (q) on the evaluation measures CS, KT, ADW, and ADO, for the datasets `adult` and `german`, respectively. Similar behaviors can be observed for the other datasets. In particular, for `compas`, `diabetes`, `titanic` and `iris` results are similar to `adult`, while for `fico` results are similar to `german`. We do not report the same plots for FI and ET as the variation of p and q do not impact these measures significantly enough.

As we know from the previous discussion and from Table 2, LMV-K is, on average, the best performer for the `adult` dataset. However, Fig. 3 unveils that this is not true for all combinations of p and q. Besides highlighting the best performers, through these plots, we can understand that the situation is even more variegated than expected, independently from the explainer we are interested in. Indeed, from Fig. 3, we can realize that for the explainers the increment of the percentage of features with missing values p has an impact w.r.t. certain measures. The measures more impacted by p are CS and ADW, as we observe an increasing performance trend when p grows. Indeed, the explainers are more coherent in explaining the corresponding record without missing values when the number of missing values is smaller. This may seem surprising. However, it makes sense that when there are fewer features with missing values, it is eas-

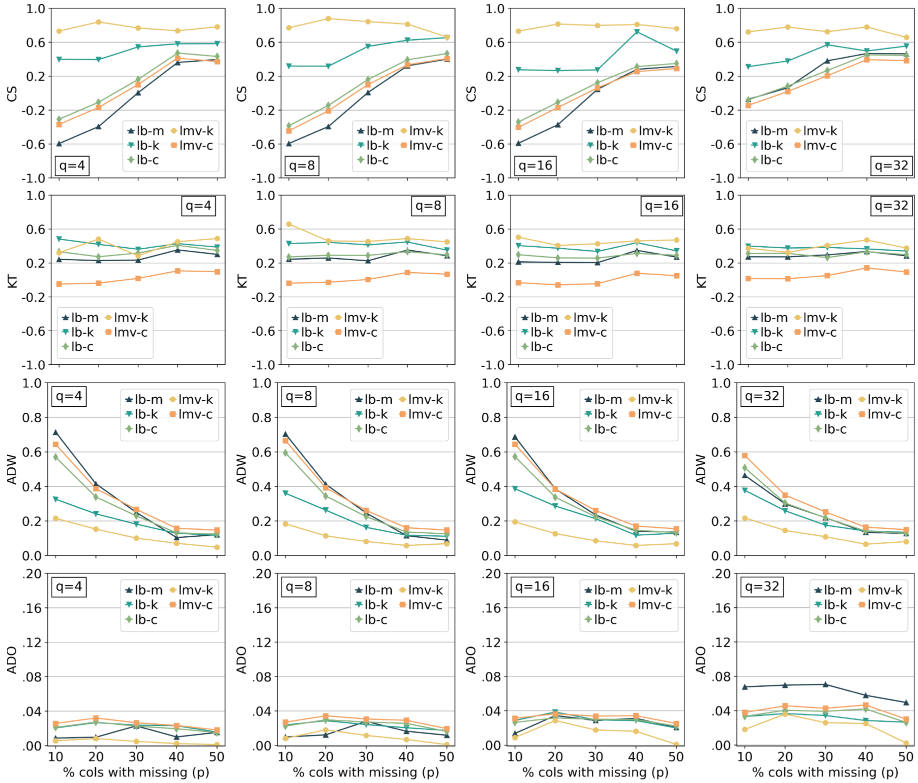


Fig. 3. Impact on evaluation measures of the variation of percentage of features with missing values (p) and percentage of missing values in features (q) for `adult`.

ier to create a discrepancy with the real importance value as by experimental setting, these are the globally most important. In comparison, when there are more features with missing values, their overall relative importance might be balanced among them, and the measures suffer less from their incorrect evaluation. A future research direction might consist in designing unbiased evaluation measures. All the explainers gain an improvement of ADW with LMV-K being constantly the best while concerning CS LMV-K and LB-K seem to be more robust, and their performance remains constant when varying p. On the other hand, KT and ADO are less impacted by the variation of p. Concerning q, we notice that nearly all the plots have slight changes from left to right, except for $q = 32$ for ADO. Indeed, in this case, especially for LB-M, we observe a degradation of the performance in terms of discrepancy for the features without missing values, i.e., having 32% of missing values in the features negatively affects the estimation of the importance of features without missing values.

In Fig. 4 are shown the same results reported in Fig. 3 but for `german`. In this case, we can notice that the percentage of features with missing values p has a

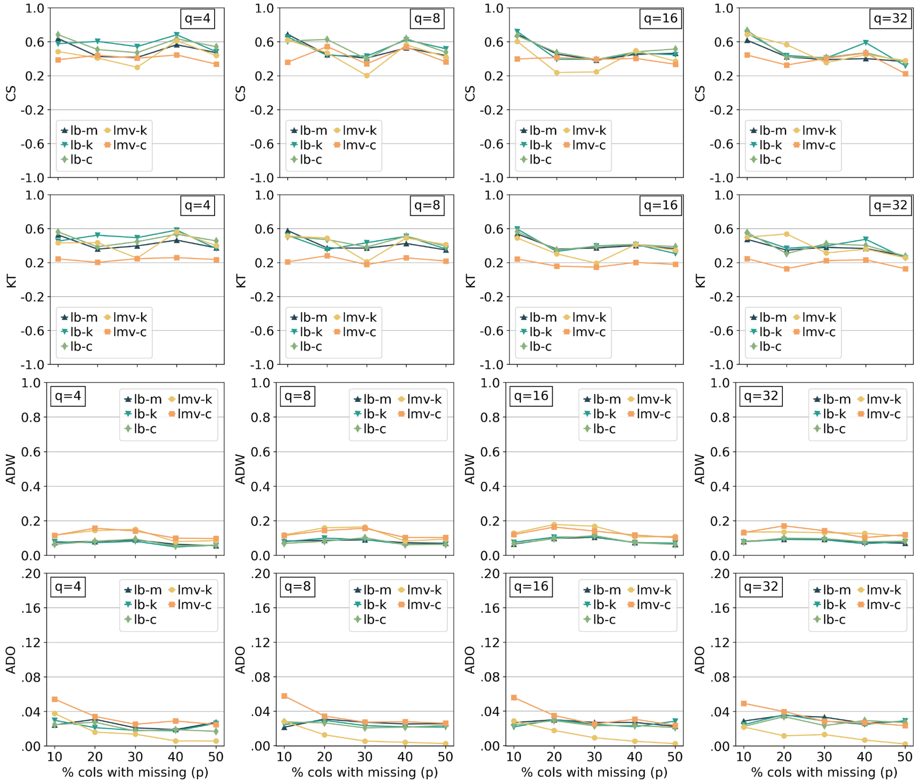


Fig. 4. Impact on evaluation measures of the variation of percentage of features with missing values (p) and percentage of missing values in features (q) for `german`.

negligible impact w.r.t. almost measures. For ADO, we observe an improvement in the performance when p grows, but this is not evident as it was in Fig. 3 for CS and ADW. In addition, the scores of all the explainers are quite similar to each other and do not follow a clear increasing or decreasing trend. Therefore, these approaches are not very sensitive to the characteristics of the missing values for `german`, for the configurations studied.

6 Conclusion

We have presented LIMEMV, the first proposal in the research area of post-hoc local model-agnostic explanation methods that is able to handle the presence of missing values directly in the explanation process. An experimental evaluation empirically proves that using LIMEMV leads to more reliable explanations than using any imputation approach in the pipeline with the classic LIME regarding coherence for features without missing values and fidelity of the local surrogate model. However, we cannot state that LIMEMV is always the best solution as it seems that various issues are tied to the type of dataset processed by the black-box, with the type of missing values and how disruptive their presence is.

As future research direction, we would like to implement the missing value-compliant version of other post-hoc explanation approaches such as SHAP [31], LORE [19] or DICE [35] by following the same strategies used for LIMEMV. Also, we intend to study these techniques not only in the MCAR setting but also in MAR and MNAR. Furthermore, we aim to adapt the neighborhood generation process by extending its capability to handle categorical, continuous, and discrete data simultaneously. Finally, to completely cover LIME applicability, we would like to study to which extent it is possible to handle missing data on data types different from tabular data, such as images, textual data, and time series.

Acknowledgment. This work is partially supported by the EU NextGenerationEU programme under the funding schemes PNRR-PE-AI FAIR (Future Artificial Intelligence Research), "SoBigData.it - Strengthening the Italian RI for Social Mining and Big Data Analytics" - Prot. IR0000013, H2020-INFRAIA-2019-1: Res. Infr. G.A. 871042 *SoBigData++*, G.A. 761758 *Humane AI*, G.A. 952215 *TAILOR*, ERC-2018-ADG G.A. 834756 *XAI*, and CHIST-ERA-19-XAI-010 SAI, by MUR (N. not yet available), FWF (N. I 5205), EPSRC (N. EP/V055712/1), NCN (N. 2020/02/Y/ST6/00064), ETAg (N. SLTAT21096), BNSF (N. KP-06-AOO2/5).

References

1. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
2. Ahmad, M.A., Eckert, C., Teredesai, A.: The challenge of imputation in explainable artificial intelligence models. In: AISafety@IJCAI, vol. 2419 of CEUR Workshop Proceedings. CEUR-WS.org (2019)
3. Azur, M.J., Stuart, E.A., Frangakis, C., Leaf, P.J.: Multiple imputation by chained equations: what is it and how does it work? Int. J. Meth. Psychiatr. Res. **20**(1), 40–49 (2011)
4. Beaulac, C., Rosenthal, J.S.: BEST: a decision tree algorithm that handles missing values. Comput. Stat. **35**(3), 1001–1026 (2020). https://doi.org/10.1007/s00180-020-00987-z
5. Bodria, F., Giannotti, F., Guidotti, R., Naretto, F., Pedreschi, D., Rinzivillo, S.: Benchmarking and survey of explanation methods for black box models. CoRR, abs/2102.13076 (2021)
6. Bramhall, S., Horn, H., Tieu, M., Lohia, N.: Qlime-a quadratic local interpretable model-agnostic explanation approach. SMU Data Sci. Rev. **3**(1), 4 (2020)
7. Brick, J.M., Kalton, G.: Handling missing data in survey research. Stat. Meth. Med. Res. **5**(3), 215–238 (1996)
8. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: KDD, pp. 785–794. ACM (2016)
9. Cinquini, M., Giannotti, F., Guidotti, R.: Boosting synthetic data generation with effective nonlinear causal discovery. In: CogMI, pp. 54–63. IEEE (2021)
10. Cinquini, M., Guidotti, R.: CALIME: causality-aware local interpretable model-agnostic explanations. CoRR, abs/2212.05256 (2022)

11. Dixon, J.K.: Pattern recognition with partly missing data. IEEE Trans. Syst. Man Cybern. **9**(10), 617–621 (1979)
12. Donders, A.R.T., Van Der Heijden, G.J., Stijnen, T., Moons, K.G.: A gentle introduction to imputation of missing values. J. Clin. Epidemiol. **59**(10), 1087–1091 (2006)
13. Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., Tabona, O.: A survey on missing data in machine learning. J. Big Data **8**(1), 1–37 (2021). https://doi.org/10.1186/s40537-021-00516-9
14. Fletcher Mercaldo, S., Blume, J.D.. Missing data and prediction: the pattern submodel. Biostatistics **21**(2), 236–252 (2020)
15. Freitas, A.A.: Comprehensible classification models: a position paper. SIGKDD Explor. **15**(1), 1–10 (2013)
16. Gosiewska, A., Biecek, P.: Do not trust additive explanations. CoRR, abs/1903.11420 (2019)
17. Groenwold, R.H., White, I.R., Donders, A.R.T., Carpenter, J.R., Altman, D.G., Moons, K.G.: Missing covariate data in clinical research: when and when not to use the missing-indicator method for analysis. CMAJ **184**(11), 1265–1269 (2012)
18. Guidotti, R.: Evaluating local explanation methods on ground truth. Artif. Intel. **291**, 103428 (2021)
19. Guidotti, R., Monreale, A., Giannotti, F., Pedreschi, D., Ruggieri, S., Turini, F.: Factual and counterfactual explanations for black box decision making. IEEE Intell. Syst. **34**(6), 14–23 (2019)
20. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(5):93:1–93:42 (2019)
21. Hall, P., Gill, N., Kurka, M., Phan, W.: Machine learning interpretability with H2O driverless AI. H2O. AI (2017)
22. Hans, S., Saha, D., Aggarwal, A.: Explainable data imputation using constraints. In: COMAD/CODS, pp. 128–132. ACM (2023)
23. Hu, L., Chen, J., Nair, V.N., Sudjianto, A.: Locally interpretable models and effects based on supervised partitioning (LIME-SUP). CoRR, abs/1806.00663 (2018)
24. Kass, G.V.: An exploratory technique for investigating large quantities of categorical data. J. Roy. Stat. Soc.: Ser. C (Appl. Stat.) **29**(2), 119–127 (1980)
25. Kaur, H., Nori, H., Jenkins, S., Caruana, R., Wallach, H., Wortman Vaughan, J.: Interpreting interpretability: understanding data scientists' use of interpretability tools for machine learning. In: CHI, pp. 1–14. ACM (2020)
26. Ke, G., et al.: LightGBM: a highly efficient gradient boosting decision tree. In: NeurIPS, pp. 3146–3154 (2017)
27. Kendall, M.G.: A new measure of rank correlation. Biometrika **30**(1/2), 81–93 (1938)
28. Li, X., et al.: A survey of data-driven and knowledge-aware explainable AI. IEEE Trans. Knowl. Data Eng. **34**(1), 29–49 (2022)
29. Lin, W., Tsai, C.: Missing value imputation: a review and analysis of the literature (2006–2017). Artif. Intel. Rev. **53**(2), 1487–1509 (2020)
30. Longo, L., Goebel, R., Lecue, F., Kieseberg, P., Holzinger, A.: Explainable artificial intelligence: concepts, applications, research challenges and visions. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds.) CD-MAKE 2020. LNCS, vol. 12279, pp. 1–16. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57321-8_1
31. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: NIPS, pp. 4765–4774 (2017)

32. Manerba, M.M., Guidotti, R.: Investigating debiasing effects on classification and explainability. In: AIES, pp. 468–478. ACM (2022)
33. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. ACM Comput. Surv. **54**(6), 115:1–115:35 (2022)
34. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. Artif. Intel. **267**, 1–38 (2019)
35. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: FAT*, pp. 607–617. ACM (2020)
36. Murray, J.S.: Multiple imputation: a review of practical and theoretical findings (2018)
37. Pasquale, F.: The black box society: The secret algorithms that control money and information. Harvard University Press (2015)
38. Payrovnaziri, S.N., et al.: The impact of missing value imputation on the interpretations of predictive models: a case study on one-year mortality prediction in ICU patients with acute myocardial infarction. MedRxiv **10**(2020.06), 06–20124347 (2020)
39. Pedersen, A.B. et al.: Missing data and multiple imputation in clinical epidemiological research. Clin. Epidemiol. **9**, 157–166 (2017)
40. Peltola, T.: Local interpretable model-agnostic explanations of Bayesian predictive models via Kullback-Leibler projections. CoRR, abs/1810.02678 (2018)
41. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: CatBoost: unbiased boosting with categorical features. In: NeurIPS, pp. 6639–6649 (2018)
42. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?": explaining the predictions of any classifier. In: HLT-NAACL Demos, pp. 97–101. The Association for Computational Linguistics (2016)
43. Rubin, D.B.: Inference and missing data. Biometrika **63**(3), 581–592 (1976)
44. Saito, S., Chua, E., Capel, N., Hu, R.: Improving LIME robustness with smarter locality sampling. CoRR, abs/2006.12302 (2020)
45. Shankaranarayana, S.M., Runje, D.: ALIME: autoencoder based approach for local interpretability. In: Yin, H., Camacho, D., Tino, P., Tallón-Ballesteros, A.J., Menezes, R., Allmendinger, R. (eds.) IDEAL 2019. LNCS, vol. 11871, pp. 454–463. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33607-3_49
46. Sharafoddini, A., Dubin, J.A., Maslove, D.M., Lee, J., et al.: A new insight into missing data in intensive care unit patient profiles: observational study. JMIR Med. Inform. **7**(1), e11605 (2019)
47. Steinberg, D.: Cart: classification and regression trees. In: The Top Ten Algorithms in Data Mining, pp. 193–216. Chapman and Hall/CRC (2009)
48. Strobl, C., Boulesteix, A.L., Zeileis, A., Hothorn, T.: Bias in random forest variable importance measures: illustrations, sources and a solution. BMC Bioinform. **8**, 1–21 (2007)
49. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley, Boston (2005)
50. Vilone, G., Longo, L.: Explainable artificial intelligence: a systematic review. CoRR, abs/2006.00093 (2020)
51. Zafar, M.R., Khan, N.: Deterministic local interpretable model-agnostic explanations for stable explainability. Mach. Learn. Knowl. Extr. **3**(3), 525–541 (2021)
52. Zhao, X., Huang, W., Huang, X., Robu, V., Flynn, D.: BayLIME: Bayesian local interpretable model-agnostic explanations. In: Uncertainty in Artificial Intelligence, pp. 887–896. PMLR (2021)