Explainable Matrix – Visualization for Global and Local Interpretability of Random Forest Classification Ensembles

Mário Popolin Neto and Fernando V. Paulovich, *Member, IEEE*

Abstract—Over the past decades, classification models have proven to be one of the essential machine learning tools given their potential and applicability in various domains. In these years, the north of the majority of the researchers had been to improve quality metrics, notwithstanding the lack of information about models' decisions such metrics convey. Recently, this paradigm has shifted, and strategies that go beyond tables and numbers to assist in interpreting models' decisions are increasing in importance. Part of this trend, visualization techniques have been extensively used to support the interpretability of classification models, with a significant focus on rule-based techniques. Despite the advances, the existing visualization approaches present limitations in terms of visual scalability, and large and complex models, such as the ones produced by the Random Forest (RF) technique, cannot be entirely visualized without losing context. In this paper, we propose *Explainable Matrix (ExMatrix)*, a novel visualization method for RF interpretability that can handle models with massive quantities of rules. It employs a simple yet powerful matrix-like visual metaphor, where rows are rules, columns are features, and cells are rules predicates, enabling the analysis of entire models and auditing classification results. ExMatrix applicability is confirmed via different usage scenarios, showing how it can be used in practice to increase trust in the classification models.

Index Terms—Random forest visualization, Logic rules visualization, Classification model interpretability, Explainable AI

1 Introduction

Imagine a machine learning classification model based on patients' records with 99% accuracy for cancer prediction, prognosticating positive breast cancer for a specific patient. Even though we are far from reaching such level of precision, we (researchers, companies, among others) have been trying to convince the general public to trust classification models, using the premise that machines are more precise than humans [12]. However, in most cases, yes or no, are not satisfactory answers. A doctor or patient inevitably may want to know why positive? What are the determinants of the outcome? What are the changes in patient records that may lead to a different prediction? Although standard instruments for building classification models, quantitative metrics such as accuracy and error cannot tell much about the model prediction, failing to provide detailed information to support understanding [33].

We are not advocating against machine learning classification models, since there is no questioning about their potential and applicability in various domains [7, 17]. The point is the acute need to go beyond tables and numbers to assist in understanding models' decisions, increasing trust in the produced results. Typically, this is called model interpretability and has become the concern of many researchers in recent years [8, 51]. Model interpretability is an open challenge and opportunity for researchers [17] and also a government concern, as the *European General Data Protection Regulation* requires explanations about decisions automatically made regarding individuals [8, 22, 34].

Model interpretability strategies are typically classified as global or local approaches. Global techniques aim at explaining the entire model, while the local ones give support for understanding the reasons for the classification of a single instance [16]. In both cases, interpretability can be attained using inherent interpretable models such as Decision Trees, Rules Sets, and Decision Tables [26], or through surrogate models [14], where black-box models, like Artificial Neural Networks, Support Vector Machines, or Random Forests [21, 22, 30], are (locally) replaced by interpretable models. The common factor in both cases is to produce logic rules to explain decisions made by a model.

Recently, visualization techniques have been used to support the interpretability of rule-based classification models. Given the nature of these rules (connections of predicates), one of the most popular visual metaphor is the node-link diagram. In this case, visual scalability is a limitation, and only small models with few rules are supported [23, 31, 43]. Attempts have been made to overcome such a hurdle by filtering rules to show only subsets of interest but paying the price of losing the overall picture of a model [52]. Besides node-link, matrix-like visual metaphors have been recently used, supporting the visualization of larger models [36]. However, visual scalability limitation still exists, and large and complex models cannot be entirely visualized, remaining as a challenge [33].

In this paper, we propose *Explainable Matrix (ExMatrix)*, a novel method for Random Forest interpretability. ExMatrix is based on a visual metaphor for logic rules [14], allowing global and local explanations for models overview and classification process auditing. The key idea is to explore logic rules by demand using matrix visualizations, where rows are logic rules, columns are features, and cells are rules predicates. ExMatrix allows reasoning on a considerable number of rules at once, helping users to build insights by employing different ordering criteria of the rules/rows and features/columns, not only supporting the analysis of subsets of rules used on a particular prediction but also the minimum changes at the instance level to make a prediction to change. The visual scalability is addressed in our solution using a simple yet powerful representation that allows us to display entire large and complex models avoiding problems related to losing context in the visualization. In summary, the main contributions of this paper are:

- A new matrix-like visual metaphor that supports the visualization of Random Forest models;
- A strategy to support Global Interpretation of large and complex Random Forest models without losing context; and
- A strategy to promote Local Interpretation of Random Forest models, supporting auditing models' decisions.

2 Related Work

Typically, visualization techniques aid in classification tasks in two different ways. One is on supporting parametrization and labeling processes aiming to improve model performance [1, 15, 25, 29, 33, 44, 46, 48]. The other is on understanding the model as a whole or the reasons for a particular prediction. In this paper, our focus is on the latter group, usually named model interpretability or just interpretability.

Interpretability techniques can be divided into pre-model, in-model, or post-model strategies, regarding support to understand classification results before, during, or after the model construction [8]. Pre-model strategies usually give support to data exploration and understanding

- *M. Popolin Neto is with Federal Institute of São Paulo. E-mail: mariopopolin@ifsp.edu.br*
- *F.V. Paulovich is with Dalhousie University. E-mail: paulovich@dal.ca.*

before model creation [8, 11, 35, 38]. In-model strategies involve the interpretation of intrinsically interpretable models, such as decision trees, during their conception, and post-model strategies concerns interpretability of complete built models, and can be model-specific [39, 50] or model-agnostic. Both in-model and post-model approaches aim to provide interpretability by producing global and/or local explanations [16].

2.1 Global Explanation

Global explanation techniques produce overviews of classification models aiming at improving users' trust in the classification model [40]. For inherently interpretable models, the global explanation is attained through visual representations of the model. For more complex non-interpretable black-box models, such as artificial neural networks, support vector machines, or random forests, it can be attained through a surrogate process where such models are (locally) approximated by interpretable ones [41]. Decision trees are commonly used as surrogate models [14, 23] given the straightforward nature of interpreting rules.

Whether a decision tree is used as a surrogate model or as a classification model per se, the most common visual metaphor for global explanation is the node-link [36, 52], such as the BaobaView technique [48]. The problem with the node-link metaphor is its scalability, mainly when it is used to create visual representations for Random Forests (which are considered black-boxes [22]), limiting the model to be small with shallow decision trees [23, 31] in a small number (e.g., between three and five) [43]. Creating a scalable visual representation for an entire Random Forest model, presenting all paths (root node to leaf node), remains a challenge even with a considerably small number of trees [33].

Although the node-link metaphor is the straightforward representation, logic rules extracted from decision trees, called decision paths, have also been used to help on interpretation. Indeed, short and disjoint rules have shown to be more suitable for user interpretation than hierarchical representations [28], and user evaluation experiments comparing the node-link metaphor, with different logic rule representations, showed that the Decision Table [19, 32] rule representation offers better comprehensibility properties [19, 24]. Nonetheless, such strategy use text information and have as drawback model size [19]. Similarly to Decision Tables [26], our method does not lean on the hierarchical property of decision trees. However, instead of using text to represent logic rules, we used a matrix-based visual metaphor, where rows are rules, columns are features, and cells are rules predicates, capable of displaying a much larger number of rules at once than the textual representations.

The idea of using a matrix metaphor to represent a classification model is not new [14, 36] and has been used before by the RuleMatrix technique [36]. RuleMatrix also represents rules in rows, features in columns, and predicates in cells using histograms representing predicates. As data histograms require a certain display space to support human cognition, the number of rules that can be displayed at once is reduced, and the rules with low coverage (how many instances a rule satisfies) and accuracy are omitted. Therefore, not being able to present the entire model or even parts of a complex model in a single visual representation. In our approach, we use a simpler icon to represent the predicates, filling rectangular shapes, colored by class, and sized proportionally to logical statements bounds in the feature space, considerably improving the scalability of the visual representation. This allows us not only to display entire models but also to display more complex ones, such as the resulting from Random Forests. Besides, we allow users to order the matrix rows and columns using varying criteria to seek for different visual patterns enabled by the icon representation we use, promoting analytical tasks not supported by the RuleMatrix, such as the holistic analysis of RF models through complete overviews.

2.2 Local Explanation

Differently from the model overview of global explanations, local explanation techniques focus on reasoning on a particular instance classification result, aiming to improve users' trust in the prediction [40]. Local explanation, as in global strategies, can be provided using inherently interpretable models or using surrogates of black-boxes. In general, local explanations are constructed using the logic rule applied to classify the instance along with its properties, such as coverage, certainty, and fidelity, providing additional information for prediction reasoning [28, 36].

One example of a visualization technique that supports local explanation is the RuleMatrix [36]. RuleMatrix was applied to support the analysis of surrogate logic rules of artificial neural networks and support vector machine models, where local explanations are taken by analyzing the employed rules, observing the instance feature values, rules predicates, and rule properties. Another interactive system closely related to our method is the iForest [52], combining techniques for Random Forest models local explanations. The iForest [52] system focus on binary classification problems, and for each instance, it allows us to explore the decision paths from decision trees using multidimensional projection techniques. By selecting a decision path of interest (a circle in the projection), a summarized decision path is built and displayed as a node-link diagram.

As discussed before, node-link diagrams are prone to present scalability issues and, although by summarizing similar decision paths, the iForest reduces the associate issues, it fails on presenting the overall picture of the voting committee of random forest classification models. Our approach shows the voting committee by displaying all rules (decision paths) used by a Random Forest model when classifying a particular instance, allowing insights on the feature space and class association through ordering the rules in different ways. Also, our approach can be applied to multi-class problems, not only binary classifications, and, similarly to iForest, it supports contrastive analysis by displaying the rules that, with the smallest changes, cause the instance under analysis to switch its final classification.

3 ExMatrix

In this section, we present *Explainable Matrix (ExMatrix)*, a visualization framework to support Random Forest (RF) model global and local interpretability.

3.1 Overview

To create a classifier, classification techniques take a labelled dataset $X = \{x_1, ..., x_N\}$ with N instances and its classes $Y = \{y_1, ..., y_N\}$, where $y_n \in C = \{c_1, ..., c_{J \leq 2}\}$ and x_n consists of a vector $x_n = \{x_n^1, ..., x_n^M\}$ with M features $F = \{f_1, ..., f_M\}$ values, and build a mathematical model to compute a class y_n when new instances $x_n \notin X$ are given as input. In this process, X is usually split into two different sets, one X_{train} to build the model and one X_{test} to test it. The existing classification techniques have adopted many different strategies to build a classifier. The Random Forest (RF) is an ensemble approach that creates multiple Decision Tree (DT) models $DT_1, ..., DT_K$ of randomly selected subsets of features or subsets of training instances, and combines them to classify an instance using a voting strategy [4, 6, 45]. Therefore, an RF model can be viewed as a collection of decision paths (or logic rules), belonging to different DTs, used or combined to classify an instance.

Aiming at supporting users to examine RF models and enable results audit, ExMatrix presents the decision paths extracted from DTs as logic rules using a matrix visual metaphor, supporting global and local explanations. ExMatrix arranges logic rules $R = \{r_1, ..., r_Z\}$ as rows, features $F = \{f_1, ..., f_M\}$ as columns, and rule predicates $r_z = \{r_z^1, ..., r_z^M\}$ as cells, inspired by similar user-friendly and powerful matrix-like solutions [9, 10, 49]. Fig. 1 depicts our method overview. ExMatrix is composed mainly of two steps. One involving the vector rules extraction, where all decision paths of the decision trees DT_k are converted into vectors with elements representing logic predicates, and a second one where these vectors are displayed using a matrix metaphor to support explanations. The next sections detail these steps, starting with the vector rule extraction process.

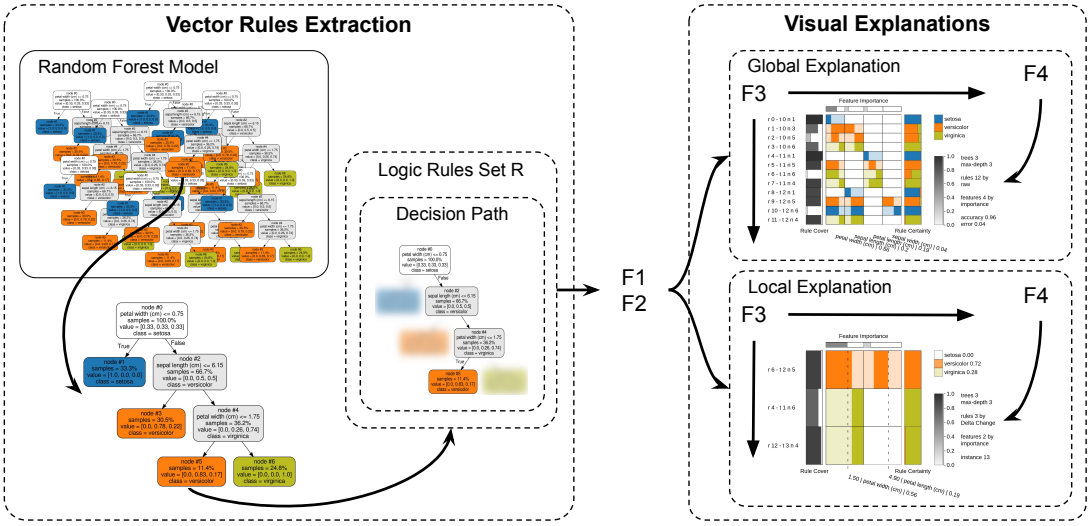


Fig. 1. Explainable Matrix overview. ExMatrix is composed of two main steps. In the first, decision paths of the Random Forest model under analysis are converted into logic rules, then, in the second, these rules are displayed using a interactive matrix metaphor to support local and global explanations.

3.2 Vector Rules Extraction

The process to convert an RF model into vector rules first extracts for every tree DT_k in the model one logic rule r_z per decision path $p_{(o,d)}$, the path from the root node o to a leaf node d. More formally, a decision path is denoted as $p_{(o,d)} = \{(f_o \bigotimes \theta_o), ..., (f_v \bigotimes \theta_v)\}$ with $\bigotimes \in \{\text{“}\leq\text{”}, \text{“}>\text{”}\}$, where node o contains an oblique cut $\bigotimes$ on feature f_o by threshold θ_o and node v is parent of node d [52]. In textual format, the decision path $p_{(o,d)}$ is converted into a logic rule as

$$IF \quad f_o \bigotimes \theta_o \quad AND \quad ... \quad AND \quad f_v \bigotimes \theta_v \quad THEN \quad c_d$$

This process results in a set of disjoint rules $R = \{r_1, ..., r_Z\}$, where each rule r_z classifies an instance x_n belonging to a class r_z^{class} if its predicates $r_z = \{r_z^1, ..., r_z^M\}$ are all true for the feature values in x_n. Each rule in R is then converted into a vector in which the elements represents the limits covered by the rule's predicate in each feature. That is, real sets $r_z^m = [\alpha_z^m, \beta_z^m]$, with lower bound α_z^m defined by Equation 1 and upper bound β_z^m defined by Equation 2 if and only if $f^m \in p_{(o,d)}$. If a logic rule r_z does not have a predicate on feature f^m, that is $f^m \notin p_{(o,d)}$, $r_z^m = \varnothing$. For the case when $f^m \in p_{(o,d)}$ and there is no $\theta_{t_h} \in p_{(o,d)}$ for $\bigotimes = \text{“}>\text{”}$, α_z^m is the lower value of $x^m \in X$, if there is no $\theta_{t_h} \in p_{(o,d)}$ for $\bigotimes = \text{“}\leq\text{”}$, β_z^m is the higher value of $x^m \in X$.

$$\alpha_z^m = \begin{cases} Max(\theta_{t_h} | f_{t_h} = f^m, \bigotimes = \text{“}>\text{”}, t_h \in t(p)) & \theta_{t_h} \in p_{(o,d)} \\ Min(x^m | x^m \in X) & \text{Otherwise} \end{cases} \quad (1)$$

$$\beta_z^m = \begin{cases} Min(\theta_{t_h} | f_{t_h} = f^m, \bigotimes = \text{“}\leq\text{”}, t_h \in t(p)) & \theta_{t_h} \in p_{(o,d)} \\ Max(x^m | x^m \in X) & \text{Otherwise} \end{cases} \quad (2)$$

Beyond predicates and classes, two other properties are obtained from each logic rule r_z. The rule cover r_z^{cover} and rule certainty $r_z^{certainty}$. Let r_z be a rule extracted from a decision path $p_{(o,d)}$. Its coverage r_z^{cover} is the number of instances of class c_d residing on the leaf node d, over the number of instances of class c_d residing on the root node o. So that, r_z^{cover} reflects the relation between total number of instances belonging to r_z^{class} used on DT_k and how many of these instances make r_z valid. Its certainty $r_z^{certainty}$ is the vector calculated at the leaf node d containing the probability of each class, calculated taking the number of instances of each class over the total number of instance in d.

As an example of rule vector extraction, consider the zoom-in decision tree in Fig. 1. This is a decision tree for the Iris dataset [18], a multi-class dataset $C = \{setosa, versicolor, virginica\}$, with $N = 150$ instances and $M = 4$ features, $F = \{sepal\ length\ ,\ sepal\ width\ ,\ petal\ length\ ,\ petal\ width\ \}$. From this tree, the decision path $p_{(\#0,\#5)}$ is translated into the logic rule $IF \quad petal\ width \quad > \quad 0.75 \quad AND \quad sepal\ length \quad > \quad 6.15 \quad AND \quad petal\ width \quad \leq \quad 1.75 \quad THEN \quad versicolor$, which results into the vector rule $r_3 = \{[6.15, 7.9], \varnothing, \varnothing, [0.75, 1.75]\}$ with $r_3^{class} = versicolor$. Regarding rule cover, $r_3^{cover} = 0.28$ since r_3 is valid for 10 out of 35 versicolor instances. Leaf node #5 has rule certainty equals to $r_3^{certainty} = \{0.0, 0.83, 0.17\}$, indicating that r_3 predicts versicolor class with 83% and 17% for virginica class.

3.3 Visual Explanations

Once the vector rules are extracted, they are used to create the matrix visual representations for global and local interpretation. To guide our design process we adopted the iForest design goals (G1 - G3) [52] and the RuleMatrix target questions (Q1 - Q4) [36] summarized on Table 1. These goals and questions consider classification model reasoning beyond performance measures like accuracy and error, focusing on the model internals. For global explanations, where the focus is an overview of a model, ExMatrix displays both the features space ranges and classes associations (G1 and Q1), and how reliable are these associations (Q2). For local explanations, where the focus is the classification of a particular instance x_n for auditing, ExMatrix allows the analysis of x_n values and features space ranges that resulted into the assigned class y_n (G2 and Q3), and the inspection of the changes in x_n that may lead to a different classification (G3 and Q4).

ExMatrix implements these design goals using a set of four functions:

F1 – Rules of Interest. Function $R' = f_{rules}(R, ...)$ returns a sub-set of rules of interest $R' \subseteq R$. For global explanations $f_{rules}(R, ...)$ returns the entire vector rules set $R' = R$, while for local explanations $f_{rules}(R, x_n, ...)$ returns a subset $R' \subset R$ related to a given instance x_n.

F2 – Features of Interest. Function $F' = f_{features}(R', ...)$ returns features of interest $F' \subseteq F$ considering a set of rules of interest R'. For global explanations $f_{features}(R', ...)$ returns all features used by the RF model, whereas for local explanations

Table 1. Explanations goals.

Global	Local
G1 Reveal the relationships between features and predictions [52]. Q1 What knowledge has the model learned? [36] Q2 How certain is the model for each piece of knowledge? [36]	G2 Uncover the underlying working mechanisms [52]. G3 Provide case-based reasoning [52]. Q3 What knowledge does the model utilize to make a prediction? [36] Q4 When and where is the model likely to fail? [36]

$f_{features}(R', x_n, ...)$ returns the features used to classify a given instance x_n.

F3 – Ordering. Function $L' = f_{ordering}(L, criteria, \ldots)$ returns an ordered version L' of a input set L following a given criteria, where L can be rules R' or features F'. This is used for both local and global explanations aiming at revealing patterns, a central characteristic in matrix-like visualizations [9, 10, 49], where rows and columns can be sorted in different ways, following, for instance, elements properties [27] or similarity measures [3, 20, 42, 47].

F4 – Predicate Icon. Function $\sigma = f_{icon}(r_z^m, \ldots)$ returns a cell icon (visual element) for a predicate r_z^m of the rule r_z and feature f_m. For the global and local explanations, a cell icon is a color-filled rectangular element, allowing our visual metaphor to display a substantial number of logic rules at once. This is an important aspect since the ability of matrix-like visualizations to display a massive number of rows and columns relies on such icons not requiring many pixels [9].

Fig. 1 shows how these four functions are used in conjunction to construct and explore the visual representations for global and local interpretation. Functions **F1** and **F2** are used to select and map rules and features of interest from the entire RF model. Function **F3** is used to change the rows and columns order to help in finding interesting patterns, and function **F4** is used to derive the predicate icon that can vary depending on the type of interpretation task (global or local). Next section, we detail how these functions are used to build the visual representations.

3.3.1 Global Explanation (GE)

Our first visual representation is an overview of RF models presenting all logic rules and all features used by the model and is intended to support *Global Explanation (GE)*. To build this matrix, $R' = f_{rules}(R, \ldots)$ returns R and $F' = f_{features}(R', \ldots)$ returns all features used by at least one rule $r_z \in R'$. As previously explained, matrix rows represent logic rules, columns features, and cells rules' predicates (icons). Columns and rows can be ordered using different criteria ($L' = f_{ordering}(L, criteria, \ldots)$). The rows can be ordered by rules' coverage, certainty, and class & coverage, while columns can be ordered by feature importance, calculated using the Mean Decrease Impurity (MDI) [5] strategy. Both, rules and features, can also be ordered by normalized real set diameter link, using the complete-linkage hierarchical clustering method [37] or the optimal-leaf-ordering [2].

The matrix cell icon σ for the rule predicate r_z^m consists of a rectangle colored according to r_z^{class} and positioned and sized inside the cell proportional to $[\alpha_z^m, \beta_z^m]$ where the cell width is proportional to $[Min(x^m | x^m \in X), Max(x^m | x^m \in X)]$ (goals G1 and Q1).

Rules and features properties are also exhibited using additional rows and columns (goal Q2). The rule coverage r_z^{cover} is shown using an extra column on the left side of the table with cells' color (grayscale) and fill proportional to the coverage. The rules certainty $r_z^{certainty}$ is shown in an extra column in the right side of the table with cells' split into colored rectangles with sizes proportional to the probability of the different classes. The feature importance $f_m^{importance}$ is shown in an extra row on the top of the table with cells' color (grayscale) and fill proportional to the importance. Also, labels are added to the table on the bottom, combining feature name and importance value, and on the left indicating the rule, decision tree, and leaf node ids (e.g., *r5 - t2n1* is the rule 5 of the decision tree 2 and the leaf node 1).

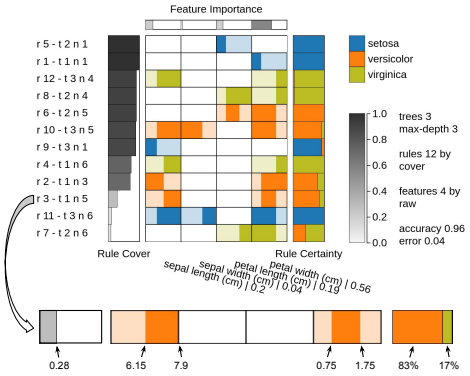


Fig. 2. Global Explanation of an RF model of the Iris dataset containing 3 tree with maximum depth equals to 3. Rows represent logic rules, columns features, and matrix cells the predicates. Additional rows and columns are also used to represent rule coverage and certainty. One matrix row is highlighted to exemplify how the rules' information is transformed into icons.

Fig. 2 presents a GE visualization of an RF model of the Iris dataset with 3 trees with limited depth equals to 3. In this example, the rows (rules) are ordered by coverage, and the columns (features) follows the dataset order. The logic rule $r_3 = \{[6.15, 7.9], \varnothing, \varnothing, [0.75, 1.75]\}$ extracted from the decision path $p_{(\#0,\#5)}$ (see Fig. 1) is zoom-in. It is colored in orange since this is the color we assign to the versicolor class and it classifies 83% of the training instances as belonging to this class (17% belonging to virginica). Also, its coverage is $r_3^{cover} = 0.28$.

3.3.2 Local Explanation Showing the Used Rules (LE/UR)

The second visual representation, called *Local Explanation Showing the Used Rules (LE/UR)*, is a matrix to help in auditing the results of an RF model providing explanations for the classification of a given instance x_n. In this process, $R' = f_{rules}(R, x_n)$ returns all logic rules (decision paths) used by the model to classify x_n (goals G2 and Q3). As in the GE visualization, $F' = f_{features}(R')$ returns all features used by logic rules R', and $f_{ordering}(L, criteria)$ orders rules R' by coverage, certainty, class & cover, and link, and features F' by importance and link, and $f_{predicate\ graph}(r_z^m, X)$ returns cell icons σ for the rule predicate r_z^m consisting of a rectangle colored according to r_z^{class} and positioned and sized inside the cell proportional to $[\alpha_z^m, \beta_z^m]$ where the cell width is proportional to $[Min(x^m | x^m \in X), Max(x^m | x^m \in X)]$.

Fig. 3 presents and example of LE/UR visual representation showing the rules used to classify the instance $x_{13} = \{6.9, 3.1, 4.9, 1.5\}$. We use the same model of Fig. 2 with 3 trees, so the RF committee uses 3 rules in the classification. The resulting matrix rows are ordered by rules coverage and columns by feature importance. The dashed line in each column indicates the values of the features of instance x_{13}. According to the committee, the probability of x_{13} to be versicolor is 72% and 28% to be virginica. Most of the virginica probability comes from the rule r_7, which holds the lowest coverage.

3.3.3 Local Explanation Showing Smallest Changes (LE/SC)

Our final matrix representation, called *Local Explanation Showing Smallest Changes (LE/SC)*, is also designed to support results audit when classifying a given instance x_n. In this visualization, for each DT_k in the model, we display the rule requiring the smallest change to make DT_k to change the classification of x_n. Let r_z be the rule extracted from DT_k that is true when classifying x_n, in this process we seek for the rule $r_e \in DT_k$ with $r_e^{class} \neq r_z^{class}$ that presents the minimum summation of

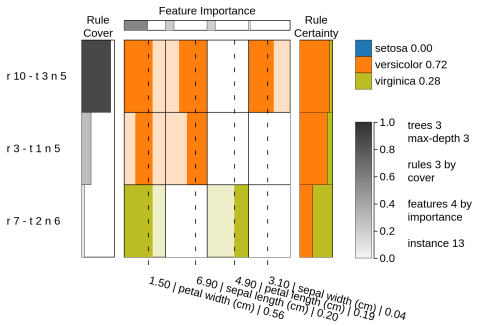


Fig. 3. Local Explanation employing the Used Rules (LE/UR) visualization. Three rules are used by the RF committee to classify a given instance as belonging to the versicolor class with 72% of probability. The dashed line in each column indicates the features' values of the instance.

the smallest changes to the values of x_n that makes r_e true and r_z false, that is, $\Delta_{(r_e,x_n)} = \sum_{m=1}^{M}(\Delta^m_{(r_e,x_n)})$, where

$$\Delta^m_{(r_e,x_n)} = \begin{cases} \frac{Min(|\alpha^m_e - x^m_n|, |\beta^m_e - x^m_n|)}{|Max(x^m|x^m \in X) - Min(x^m|x^m \in X)|} & if\ x^m_n \notin [\alpha^m_e, \beta^m_e] \\ 0 & if\ x^m_n \in [\alpha^m_e, \beta^m_e] \end{cases} \quad (3)$$

Using this formulation, the function $R' = f_{rules}(R, x_n)$ returns the list of logic rules containing the more similar rules r_e extracted from each DT_k in the model that makes x_n to change class. The result is a set of rules that can potentially change the classification process outcome requiring the lowest changes (goals G3 and Q4). The function $F' = f_{features}(R', x_n)$ returns the features with a non zero change $\Delta^m_{(r_e,x_n)}$ in at least one rule $r_e \in R'$, allowing to focus on the features with relevant information for the changing process. Beyond the ordering criteria for rules and features previously used, function $f_{ordering}(L, criteria)$ also allows ordering using the change summation $\sum_{m=1}^{M}(\Delta^m_{(r_e,x_n)})$. Finally, function $\sigma = f_{predicate\ graph}(r^m_e, x_n)$ returns a rectangle positioned and sized proportional to the change $\Delta^m_{(r_e,x_n)}$, with positive changes colored in green and negative in red. To help understand the class swapping, we add another column to the right of the table indicating the classification returned by the original rule r_z, showing the difference to the similar rule r_e that cause the decision tree to change prediction.

Fig. 4 shows an example of visualization for the same model of Fig. 2 given an instance $x_{13} = \{6.9, 3.1, 4.9, 1.5\}$. Features F' are ordered by importance and rules by change sum. The dashed lines represent the instance x_{13} values. Just as an illustration, rule r_6 presents the smallest change in the feature "petal length" to make it change class virginica to class versicolor. Also, it presents high coverage and certainty, so it is an important rule to the committee.

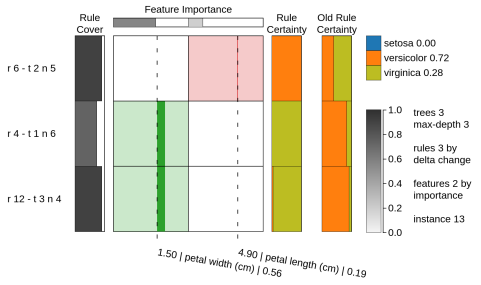


Fig. 4. Local Explanation Showing Smallest Changes (LE/SC) visualization. Three rules with the smallest change to make the decisions trees to change class decisions are displayed. The rule in the first row presents the smallest change need to change the classification of a given instance. Small perturbations may change the RF classification decision.

4 Results and Evaluation

In this section, we present and evaluate our framework through a use-case discussing the proposed features, two usage-scenarios showing ExMatrix being used in practice to explore Random Forest (RF) models, finishing with a formal user test. All datasets employed in this section were downloaded from the *UCI Machine Learning Repository* [13], and the ExMatrix testing implementation is publicly available at $< url >$.

4.1 Use Case: Breast Cancer Diagnostic

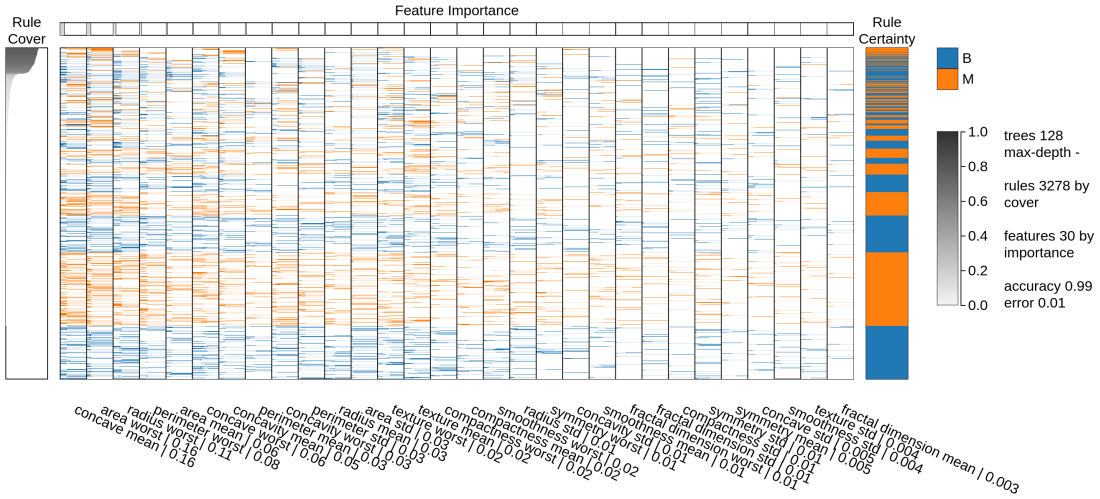
In this use case, we utilize the *Wisconsin Breast Cancer Diagnostic (WBCD)* dataset to discuss how to use ExMatrix local and global explanations to analyze RF models varying the number of employed decision trees and their maximum depth. The WDBC dataset contains samples of breast mass cells of $N = 569$ patients, 357 classified as benign (B) and 212 as malignant (M), with $M = 30$ features (cells properties). The RF models were created randomly, selecting 70% of the instances for training and 30% for testing.

In the first model, we set the number of decision trees to $K = 128$ and do not limit their maximum depth. The result is a model with 3,278 logic rules, 25.6 rules per decision tree, and accuracy of 0.99. Fig. 5(a) presents an overview of the model using the Global Explanation representation (see Sect. 3.3.1). In this visualization, the rules are ordered by coverage and features by importance. Using this ordering scheme, it is possible to see that "concave mean", "area worst", and "radius worst" are the three most important features, whereas "smoothness std", "texture std", and "fractal dimension mean" are less important. Also, all the 30 dataset features were used in the classification, and taking only the high coverage rules and features with more importance ("concave mean' to "texture worst"), it is possible to observe that low feature values appear to be more related to class B while higher values to class M (guidelines G1, Q1, and Q2).

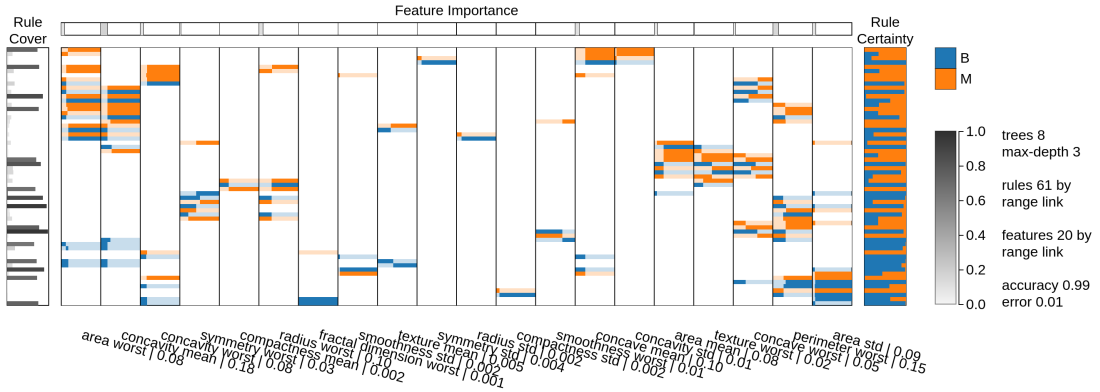
For the second model, we set the number of decision trees to $K = 8$ with a maximum depth $DT_{max\ depth} = 3$. The resulting model also presents 0.99 of accuracy but produces only 61 rules, 7.6 rules per decision tree, with much lower complexity with no more than 3 predicates. Fig. 5(b) presents an overview of this model using the Global Explanation representation. In this representation, rules and features are ordered by hierarchical clustering link. The resulting visualization allows us to observe rules with divergent class sharing features value ranges. This new model presents a similar pattern observed in the previous model. Low values of features are more associated with class B and higher values with class M (guidelines G1, Q1, and Q2) for some specific features, for instance, "symmetry std", "radius std", and "compactness std", not depending on rules coverage.

In both models, the decision trees were created using random selections of features, which result in different feature importance values given the different model parametrizations. In the first model, by not setting a maximum depth, the derived rules maximize certainty but minimize coverage, so the model is more specific with each rule covering a few instances. The opposite can be observed in the less complex model, defining a more generic model. It is also possible to notice through these models overviews that by decreasing K and $DT_{max\ depth}$ values, fewer features are used by the rules, 30 for the more complex model and 20 for the simpler one, indicating that some of the collected features are less relevant for predictions in this specific dataset.

The error rate of 0.01 in the two models is due to a misclassification of only one instance of the test set. In the first model, instance x_{29} was incorrectly classified as class B with a probability of 0.55. Fig. 6(a) shows the Local Explanation Showing the Used Rules representation (see Sect. 3.3.2) using x_{29} as the target instance. In this visualization, it is possible to see that all features are used for this particular prediction. Features value ranges of classes B and M overlap several times for almost all features, except for "fractal dimension std" and "concave std". Also, the "compactness std", "symmetry std", and "symmetry mean" are the features that most contribute to class B result (guidelines G2 and Q3). Analyzing the smallest changes to make the trees to change prediction (see Fig. 6(b)), positive changes on feature "concave mean" may tie or alter the prediction of x_{29} to class M, while negative



(a) Complex model with 128 trees without depth limit, resulting in 3,278 logic rules.



(b) Less complex model with 8 trees with maximum depth of 3, resulting into 61 logic rules.

Fig. 5. Global Explanation representations of two different models of the same dataset. In (a), the model uses more decision trees without limit of depth, while in (b), the number of trees is heavily decreased with limited depth. Both models present the same accuracy level (0.99), but the more complex model is less generic, with many rules showing low coverage and high certainty, the opposite of the low complex model.

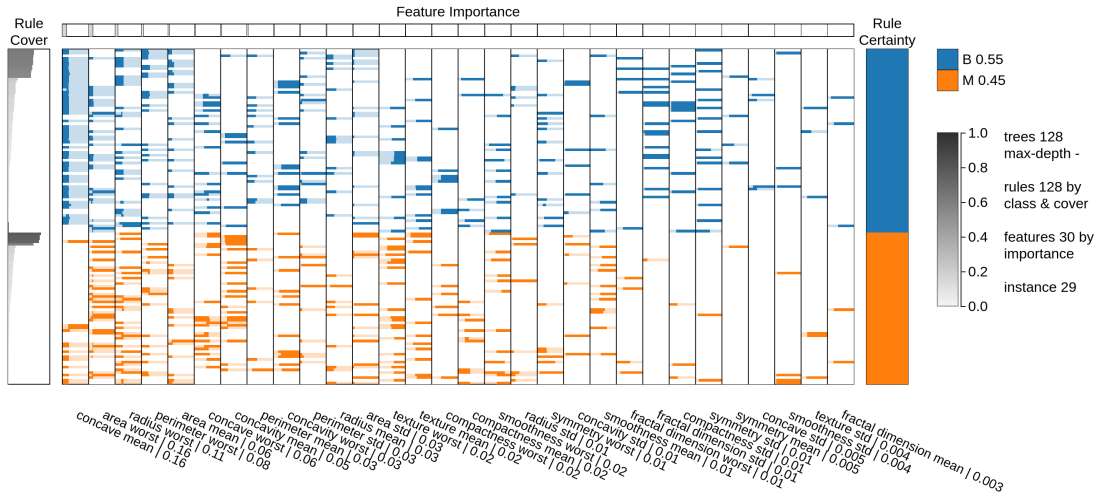
change on "area worst" increases its classification as class B (guidelines G3 and Q4).

In the second model, instance x_{130} was classified as class M incorrectly, with a probability of 0.58. Fig. 7(a) shows the Local Explanation Showing the Used Rules visual representation with x_{130} as the target instance. If only the first four rules with high coverage were considered for prediction, the classification of instance x_{130} would be even more tied between classes M and B since these rules have similar certainty. Although all remaining rules predict class M, they hold very low coverage and uncertainty degrees (guidelines G2 and Q3), so with low classification relevance. So it is clear that the model is not complex enough to decide about x_{130} classification. Analyzing the closest rules to make the trees to change prediction (see Fig. 7(b)), shows that these swapping rules also present high certainty and that the smaller changes are negative on features "area std" and "area worst" to turn x_{130} the prediction to B, while positive changes on "perimeter worst", "area mean", and "texture worst" increase the class M probability outcome (guidelines G3 and Q4). Given even more evidence about the lack of complexity of the model to decide about x_{130}.

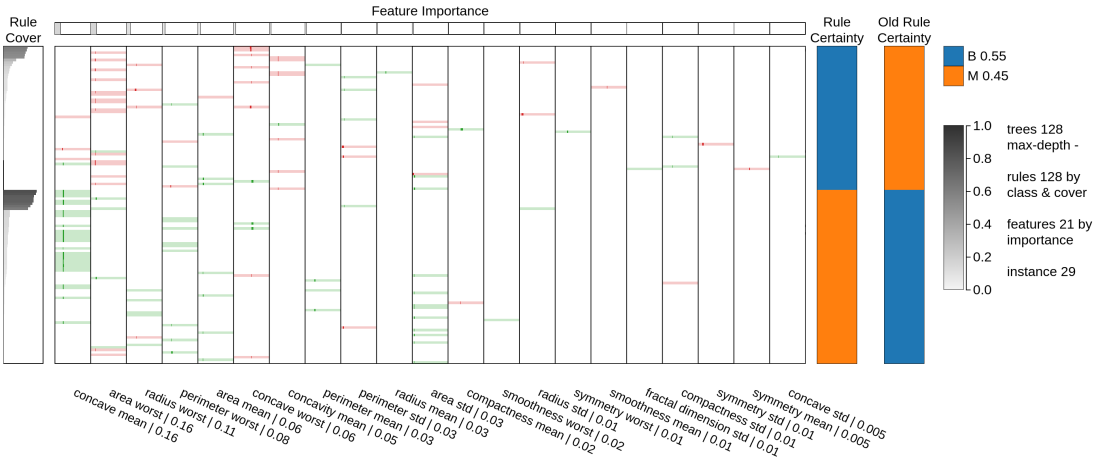
4.2 Usage Scenario I: German Credit Bank

As a first hypothetical usage scenario, we describe a bank manager Sylvia incorporating ExMatrix in her data analytics pipeline. To speed up her process of evaluating loan applications, she sends to a data science team her own dataset of years of experience and asks for a classification system to aid in the decision-making process. Such dataset contains 1,000 instances (customers profiles) and 20 features (customers information), with 700 customers presenting rejected applications and 300 accepted (here we use the German Credit Data from the UCI for illustration). For the implementation of such a system, Sylvia has two main requirements: 1 – the system must be precise in classifying loan applications, and; 2 – the classification results must be interpretable so she can explain the outcome.

To fulfill the requirements, the data science team build a Random Forest model setting the number of decision trees to $K = 32$ with maximum depth $DT_{max\ depth} = 6$. Before creating the model, the data was pre-processed to transform categorical features into numerical ordinal features, and 9 features were selected (we follow [52] approach). The accuracy of the produced model was 0.81, resulting in 1,273 logic rules, 38.7 rules per tree. Using ExMatrix Global Explanation representation (Fig. 8(a)), she observes that the features "Account Balance", "Credit Amount", and "Duration of Credit" are the three most important, whereas "Value Savings/Stocks", "Duration in Current address", and "Instalment per cent" are the three less. Also, she notices that applications requesting credit to be paid in more extended periods tend to be rejected (rules with high coverage in the third column), and applications requesting low amounts of credit are prone to be accepted (second column in general), matching her expectations. However, unexpectedly, customers without account balance (meaning no account in the bank) have less chance to have their application rejected (rules with high coverage in the first column), something she did not anticipate. Although confronting some of her expectations and bias, she trusts her



(a) Rules used to classify a misclassified instance.



(b) Changes in the misclassified instance feature values to make the trees to change class prediction.

Fig. 6. Local Explanation representations of the complex model of Fig. 5(a). Two different visualizations are displayed, one showing the rules employed in the classification of a target instance (a), and one presenting the smallest changes to make the trees of a model to change the prediction of that instance (b). In both cases, the target instance is the only misclassified instance.

data, and the classification accuracy seems convincing, so she decides to put the system in practice.

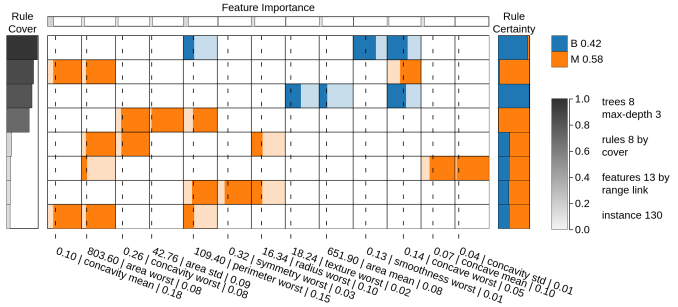
One day she receives a new customer interest in a loan. After filling the system with his data, unfortunately, the application got rejected by the classification system. Based on the new *European General Data Protection Regulation* [8, 22, 34] that requires explanations about decisions automatically made, the customer requests clarification. By inspecting the ExMatrix Local Explanation Showing the Used Rules visualization (see Fig. 8(b)), she notices, besides the denied result with probability of 0.65, that all the 9 features were used in the evaluation. Also, she sees that the feature "Length of current employment" is the most directly related to the denied outcome since it is used only by rules that result in rejection. Using this information, she explains to the customer that since he is working for less than one year in the current job (2.00 as "Length of current employment" corresponds to less than 1 year), the bank recommends denying the application. However, analyzing the smallest changes to make the trees to change prediction (see Fig. 8(c)), she realizes that negative changes on features "Credit Amount" and "Duration of Credit" may turn the outcome to authorized. Thereby, as an alternative, she suggests lowering the requested amount as well as the number of installments. Based on the observable differences to make the rules change class, she notices that upon reducing the credit application from $1,207 to $867 and the number of payments from 24 to 15, the system changes recommendation to approval.

4.3 Usage Scenario II: Contraceptive Method

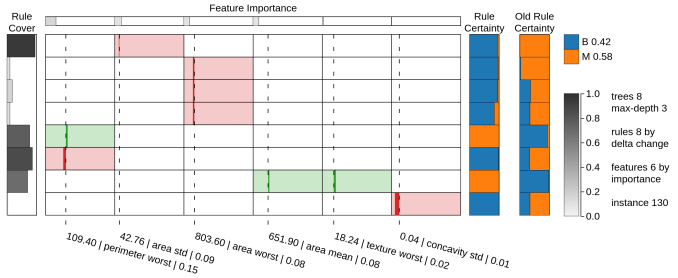
This last usage scenario presents Christine, a public health policy manager who wants to create a contraceptive campaign to advertise a new, safer drug for long term use. To investigate married wives' preferences, Christine's data science team creates a prediction model using a data set with information about contraceptive usage choices her office collected past year (here we use the Contraceptive Method Choice dataset from UCI for illustration). The dataset contains 1,473 samples (married wives profiles) with 9 features, where each instance belongs to one of the classes "No-use", "Long-term", and "Short-term", regarding the contraceptive usage method, with 42.7% of the instances belonging to class No-use, 22.6% to Long-term, and 34.7% to Short-term.

Since interpretability is mandatory in this scenario, so the results can be used in practice, the data science team creates a Random Forest prediction model and employs the ExMatrix to support analysis. To create the model, the team set the number of decision trees to $K = 32$ and maximum depth to $DT_{max\ depth} = 6$, resulting on 1,383 logic rules, 43.2 rules per tree. The model accuracy is 0.63, and, although not ideal for individual classifications, can be used to understand the overall scenario of the collected data since ExMatrix allows the analysis of certainty and generality of individual logic rules.

By inspecting the Global Explanation representation of the model (see Fig. 9), she readily understands that the features "Number of



(a) Rules used to classify a misclassified instance.



(b) Changes in the misclassified instance feature values to make the trees to change class prediction.

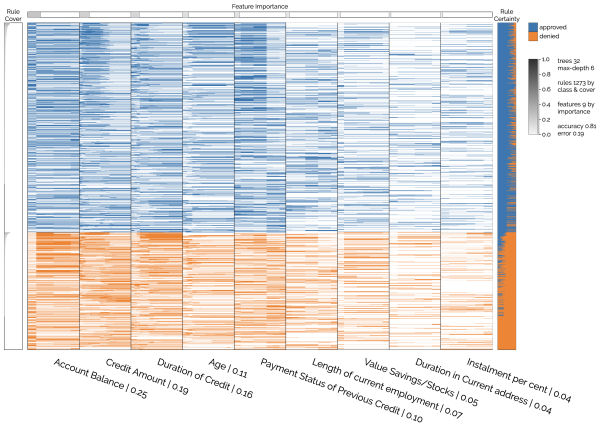
Fig. 7. Local Explanation representations of the less complex model of Fig. 5(b). Two different visualizations are displayed, one showing the rules employed in the classification of a target instance (a), and one presenting the smallest changes to make the trees of a model to change the prediction of that instance (b). In both cases, the target instance is the only misclassified instance.

children ever born", "Wife age", and "Wife education" are the three most relevant for the defining the contraceptive method class, while "Media exposure", "Wife now working?", and "Wife religion" are the three less. Also, ordering the matrix using the rule coverage per class and considering only the rules with high coverage on each class, she notices some interesting patterns regarding features value ranges and classes. For instance, lower values for the feature "Number of children ever born" are more related to class No-use, while lower values are rarely related to class Long-term. Higher values for the feature "Wife age" are related to class Long-term, while mean and lower values are more related to class Short-term. Higher values for "Wife education" are more related to class Long-term, and mean and lower values are associated with classes No-use and Short-term (guidelines G1, Q1, and Q2). Based on these observations, and given the modest budget she received for the campaign, Christine decides to focus on the group of older and highly educated wives with at least one child to target the campaign's first phase.

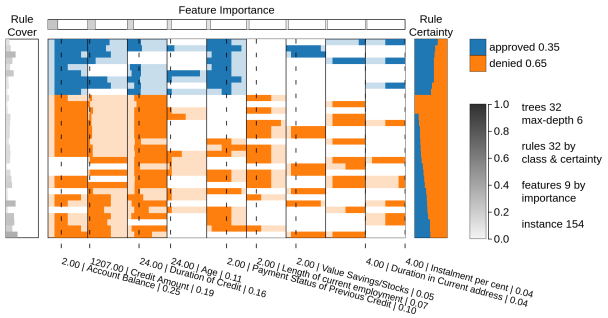
Fig.

To evaluate our framework, we performed a user study to access the proposed visual representations for global and local explanations. In this study, we asked four different questions covering the main features of ExMatrix using real examples extracted from the use-case of Sect. 4.1, focusing on evaluating the guidelines presented in Table 1.

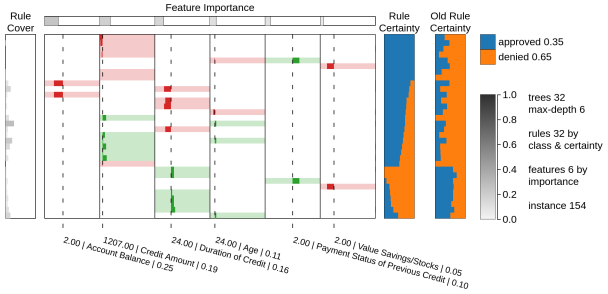
The study starts with video tutorials about Random Forest's basic concepts and how to use ExMatrix to analyze classification processes and results. We test our framework with 13 users, 69.2% male, and 30.8% female, aged between 24 to 36, all with a background in computer science and related fields. For the tests, we use the images presented in Fig. 5(a), Fig. 6(a), and Fig. 6(b). We asked four different questions with multiple choices (see Table 2) replacing the names of the features by "Feature 1" to "Feature 30", and classes names by "Class A" and "Class B" aiming at removing any influence of knowledge domain in the results (our focus is to access the visual metaphors not the results of the classification model).



(a) Overview of the model.



(b) Rules used to classify a given instance.



(c) Changes in a given instance feature values to make the trees to change class prediction.

Fig. 8. ExMatrix representations of a Random Forest model and the German Credit Data dataset. In general (a), applications requesting credit to be paid in longer periods tend to be rejected, while applications requesting low amounts of credit have more chances to be accepted. Analyzing one sample of rejected application (b), it is possible to infer that it is probably rejected due to the (applicant) short period working in the current job. However, lowering the requested amount as well as the number of installments can change the model decision.

Using the Global Explanation representation (Fig. 5(a)), 76.9% of the users were able to identify patterns involving feature space ranges and classes, where, only considering rules with high coverage, low features values are more related to class B, while features with large values are more related to cFig. lass M (Qst 1). Using the Local Explanation representation showing the used rules (Fig. 6(a)), also 76.9% of the users were able to recognize that the feature "concave std" is the most important to classify instance x_{29} as belonging to class A (Qst 2). Using the Local Explanation representations showing the smallest differences for a given instance to change class (see Fig. 6(b)), 61.5% of the users were able to identify that negative changes on instance x_{29} features "area worst" and "concavity mean" values would better support the class B outcome (Qst 3), and 46.2% were able to identify

Table 2. User study questions. Our focus was to evaluate if ExMatrix implements the guidelines presented in Table 1.

Question	Guidelines	Visualization
Qst 1 - About features space ranges and class ASSOCIATIONS. Considering rules with HIGH COVERAGE, and features with HIGH IMPORTANCE, select your answer: (three options of associations)	G1, Q1, and Q2	Fig. 5(a)
Qst 2 - Instance 29 is classified as Class A with probability of 55%, against 45% for Class B. What feature is more related to Class A and less related to Class B? (four options of features)	G2 and Q3	Fig. 6(a)
Qst 3 - Select the pair of features where DELTA CHANGES on instance 29 will potentially INCREASE Class A probability, and by that may SUPPORT its classification as Class A. (four options of features pairs)	G3 and Q4	Fig. 6(b)
Qst 4 - Select the pair of features where DELTA CHANGES on instance 29 will potentially INCREASE Class B probability, and by that may ALTER its classification as Class A. (four options of features pairs)	G3 and Q4	Fig. 6(b)

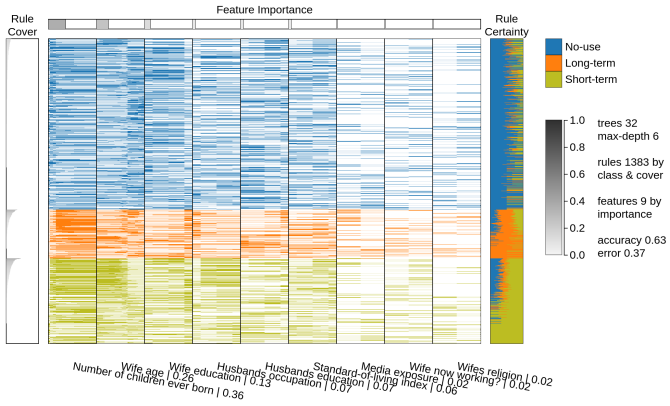


Fig. 9. Global Explanation representation of a Random Forest model and the Contraceptive Method Choice dataset. Focusing on the rules with high coverage per class, some interesting patterns emerge regarding features value ranges and classes. For instance, older women tend to use long-term contraceptive methods, while younger are prone to use short-term. Also, highly educated women make use of long-term methods, while women with fewer years of education tend to use short-term or none methods.

that positive changes for features "concave mean" and "perimeter worst" values may alter the outcome from class B to class M (Qst 4).

In general, the results were very promising for the first two scenarios, but users present slightly worse results when interpreting the Local Explanation representations showing the smallest differences. This is not surprising since this last visual representation requires a much better background about Random Forest theory than the first two. The Global Explanation and Local Explanation showing the used rules are more generic and involve much fewer concepts of how Random Forest models work internally. In contrast, the last one requires a good level of knowledge about ensembles models and how the voting system work when making a prediction. And, although, most of the users self-declared some background in machine learning, only a few are specialists in the Random Forest technique and ensemble methods.

We also have asked subjective, open questions, and, in general, users gave positive feedbacks about ExMatrix explanations, where the visualizations were classified as visually pleasing and useful for understanding Random Forest models.

5 Discussion and Limitations

Although we designed ExMatrix with Random Forest (RF) interpretability in mind, it can be readily applied to Decision Tree models, such as the ones used as surrogates for Artificial Neural Networks [14, 23], or approaches based on logic rules such as Decision Tables since the core of our method is the visualization of logic rules, opening up many different scenarios not explored in this paper. Another potential application scenario not investigated is model construction and improvement. The visual metaphors we proposed can be easily applied to the analysis and comparison of RF models resulted from different parametrizations, for instance, with different numbers of trees and their maximum depth. Therefore, allowing machine learning engineers to go beyond accuracy and error when building a model.

In terms of visual scalability, although ExMatrix supports the analysis of many more rules concomitantly if compared with the state-of-the-art, we still have problems if the number of trees substantially grows, since this exponentially increases the number of rules for a Global Explanation. Although we can represent one rule per line of pixels, we are still limited by the display resolution. Scroll bars can be used, but context can be lost in the process. One potential solution for this issue is to make the height of the rows proportional to coverage or certainty, so that the rules with the lowest coverage or certainty are less prominent (visible) and could even be combined in less than one line of pixels. We have not tested this approach and left it as future work.

Finally, regarding the user study, although the results were satisfactory and within what we expect. The Local Explanation representation showing the smallest differences to change class still needs to be improved to reach the same level of the other representations. Nevertheless, as discussed in the User Study section, the low performance of users is not only resulted from the visual metaphor but also the users' expertise. Among the users we tested, few of them know the RF technique in detail. Based on this, we conclude that people with less expertise can use the Global Explanation and the Local Explanation showing the Used Rules representations, but the Local Explanation showing the changes is more suitable for experts. In general, despite the complexity of the problems we ask users to solve, they acknowledged the ExMatrix potential, expressing encouraging positive remarks, including "*... this solution ... allows a deeper understanding of how each particular rule or feature impacted on the final the decision/classification.*" or "*I think the ExMatrix can be used in a variety of domains, from E-commerce to Healthcare...*".

6 Conclusion and Future Works

In this paper, we present *Explainable Matrix (ExMatrix)*, a novel method for Random Forest (RF) models interpretability. ExMatrix uses a matrix-like visual metaphor, where logic rules are rows, features are columns, and rules predicates are cells, allowing users to obtain overviews of models (Global Explanations) and to audit results (Local Explanations). Although simple in nature, ExMatrix visual representations are powerful and support the execution of tasks that are challenging to perform without a proper interactive visualization. To show ExMatrix usefulness, we present one use-case and two hypothetical usage scenarios employing real datasets, showing that with the proper training, users can better understand RF models beyond what is granted by usual metrics, like accuracy or error rate. Although our primary goal is to aid in RF models global and local interpretability, the ExMatrix method can also be applied for the analysis of complex Decision Trees, such as the ones used as surrogates of Artificial Neural Networks, or any other technique based on logic rules, opening up new possibilities for future development and use.

Acknowledgments

The authors wish to thank the postgaduate studies support from the Qualification Program of the Federal Institute of São Paulo (IFSP).

REFERENCES

[1] M. Ankerst, C. Elsen, M. Ester, and H.-P. Kriegel. Visual classification: An interactive approach to decision tree construction. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pp. 392–396. ACM, New York, NY, USA, 1999. doi: 10.1145/312129.312298

[2] Z. Bar-Joseph, D. K. Gifford, and T. S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17:S22–S29, 06 2001. doi: 10.1093/bioinformatics/17.suppl_1.S22

[3] M. Behrisch, B. Bach, N. Henry Riche, T. Schreck, and J.-D. Fekete. Matrix reordering methods for table and network visualization. *Computer Graphics Forum*, 35(3):693–716, 2016. doi: 10.1111/cgf.12935

[4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324

[5] L. Breiman. Manual on setting up, using, and understanding random forests v3. 1. *Statistics Department University of California Berkeley, CA, USA*, 1:58, 2002.

[6] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

[7] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018. doi: 10.1038/s41586-018-0337-2

[8] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8), 2019. doi: 10.3390/electronics8080832

[9] C.-H. Chen, H.-G. Hwu, W.-J. Jang, C.-H. Kao, Y.-J. Tien, S. Tzeng, and H.-M. Wu. Matrix visualization and information mining. In J. Antoch, ed., *COMPSTAT 2004 — Proceedings in Computational Statistics*, pp. 85–100. Physica-Verlag HD, Heidelberg, 2004.

[10] C.-h. Chen, A. Sinica, and Taipei. Generalized association plots: information visualization via iteratively generated correlation matrices. *Statistica Sinica*, 12:7–29, 01 2002.

[11] J. Choo, H. Lee, J. Kihm, and H. Park. ivisclassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pp. 27–34, Oct 2010. doi: 10.1109/VAST.2010.5652443

[12] J. A. Cruz and D. S. Wishart. Applications of machine learning in cancer prediction and prognosis. *Cancer Informatics*, 2:117693510600200030, 2006. doi: 10.1177/117693510600200030

[13] D. Dheeru and E. Karra Taniskidou. UCI machine learning repository, 2017.

[14] F. Di Castro and E. Bertini. Surrogate decision tree visualization interpreting and visualizing black-box classification models with surrogate decision tree. *CEUR Workshop Proceedings*, 2327, 1 2019.

[15] T.-N. Do. Towards simple, easy to understand, an interactive decision tree algorithm. *College Inf. Technol., Can tho Univ., Can Tho, Vietnam, Tech. Rep*, pp. 06–01, 2007.

[16] M. Du, N. Liu, and X. Hu. Techniques for interpretable machine learning, 2018.

[17] A. Endert, W. Ribarsky, C. Turkay, B. W. Wong, I. Nabney, I. D. Blanco, and F. Rossi. The state of the art in integrating machine learning into visual analytics. *Computer Graphics Forum*, 36(8):458–486, 2017. doi: 10.1111/cgf.13092

[18] R. A. FISHER. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936. doi: 10.1111/j.1469-1809.1936.tb02137.x

[19] A. A. Freitas. Comprehensible classification models: A position paper. *SIGKDD Explor. Newsl.*, 15(1):1–10, Mar. 2014. doi: 10.1145/2594473.2594475

[20] T. Fujiwara, O. Kwon, and K. Ma. Supporting analysis of dimensionality reduction results with contrastive learning. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):45–55, Jan 2020. doi: 10.1109/TVCG.2019.2934251

[21] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti. Local rule-based explanations of black box decision systems, 2018.

[22] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5):93:1–93:42, Aug. 2018. doi: 10.1145/3236009

[23] P. Hall. On the art and science of machine learning explanations, 2018.

[24] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141 – 154, 2011. doi: 10.1016/j.dss.2010.12.003

[25] B. Hferlin, R. Netzel, M. Hferlin, D. Weiskopf, and G. Heidemann. Interactive learning of ad-hoc classifiers for video visual analytics. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 23–32, Oct 2012. doi: 10.1109/VAST.2012.6400492

[26] R. Kohavi. The power of decision tables. In *Proceedings of the 8th European Conference on Machine Learning*, ECML'95, pp. 174–189. Springer-Verlag, Berlin, Heidelberg, 1995.

[27] J. Krause, A. Dasgupta, J. Swartz, Y. Aphinyanaphongs, and E. Bertini. A workflow for visual diagnostics of binary classifiers using instance-level explanations. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 162–172, Oct 2017. doi: 10.1109/VAST.2017.8585720

[28] H. Lakkaraju, S. H. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 1675–1684. ACM, New York, NY, USA, 2016. doi: 10.1145/2939672.2939874

[29] T. Lee, J. Johnson, and S. Cheng. An interactive machine learning framework. *ArXiv*, abs/1610.05463, 2016.

[30] J. Lei, Z. Wang, Z. Feng, M. Song, and J. Bu. Understanding the prediction process of deep networks by forests. In *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, pp. 1–7, Sep. 2018. doi: 10.1109/BigMM.2018.8499073

[31] R. A. G. Leiva, A. F. Anta, V. Mancuso, and P. Casari. A novel hyperparameter-free approach to decision tree construction that avoids overfitting by design. *CoRR*, abs/1906.01246, 2019.

[32] E. Lima, C. Mues, and B. Baesens. Domain knowledge integration in data mining using decision tables: case studies in churn prediction. *Journal of the Operational Research Society*, 60(8):1096–1106, 2009. doi: 10.1057/jors.2008.161

[33] S. Liu, J. Xiao, J. Liu, X. Wang, J. Wu, and J. Zhu. Visual diagnosis of tree boosting methods. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):163–173, Jan 2018. doi: 10.1109/TVCG.2017.2744378

[34] X. Liu, X. Wang, and S. Matwin. Interpretable deep convolutional neural networks via meta-learning. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, July 2018. doi: 10.1109/IJCNN.2018.8489172

[35] M. Migut and M. Worring. Visual exploration of classification models for risk assessment. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pp. 11–18, Oct 2010. doi: 10.1109/VAST.2010.5652398

[36] Y. Ming, H. Qu, and E. Bertini. Rulematrix: Visualizing and understanding classifiers with rules. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):342–352, Jan 2019. doi: 10.1109/TVCG.2018.2864812

[37] D. Mullner. Modern hierarchical, agglomerative clustering algorithms, 2011.

[38] J. G. S. Paiva, W. R. Schwartz, H. Pedrini, and R. Minghim. An approach to supporting incremental visual data classification. *IEEE Transactions on Visualization and Computer Graphics*, 21(1):4–17, Jan 2015. doi: 10.1109/TVCG.2014.2331979

[39] P. E. Rauber, S. G. Fadel, A. X. Falco, and A. C. Telea. Visualizing the hidden activity of artificial neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):101–110, Jan 2017. doi: 10.1109/TVCG.2016.2598838

[40] M. T. Ribeiro, S. Singh, and C. Guestrin. 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, p. 11351144. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2939672.2939778

[41] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. 2018.

[42] S. seok Choi and S. hyuk Cha. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, pp. 43–48, 2010.

[43] G. Stiglic, M. Mertik, V. Podgorelec, and P. Kokol. Using visual interpretation of small ensembles in microarray analysis. In *19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)*, pp. 691–695, June 2006. doi: 10.1109/CBMS.2006.169

[44] J. Talbot, B. Lee, A. Kapoor, and D. S. Tan. Ensemblematrix: Interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pp. 1283–1292. ACM, New York, NY, USA, 2009. doi:

10.1145/1518701.1518895

[45] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to data mining*. Pearson, 1 ed., 2005.

[46] S. T. Teoh and K.-L. Ma. Paintingclass: Interactive construction, visualization and exploration of decision trees. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pp. 667–672. ACM, New York, NY, USA, 2003. doi: 10.1145/956750.956837

[47] S. Tzeng, H. Wu, and C. Chen. Selection of proximity measures for matrix visualization of binary data. In *2009 2nd International Conference on Biomedical Engineering and Informatics*, pp. 1–9, Oct 2009. doi: 10.1109/BMEI.2009.5305137

[48] S. van den Elzen and J. J. van Wijk. Baobabview: Interactive construction and analysis of decision trees. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 151–160, Oct 2011. doi: 10.1109/VAST.2011.6102453

[49] H.-M. Wu, S. Tzeng, and C.-h. Chen. *Matrix Visualization*, pp. 681–708. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. doi: 10.1007/978-3-540-33037-0_26

[50] M. Wu, M. Hughes, S. Parbhoo, M. Zazzi, V. Roth, and F. Doshi-Velez. Beyond sparsity: Tree regularization of deep models for interpretability. 2018.

[51] F. Yang, M. Du, and X. Hu. Evaluating explanation without ground truth in interpretable machine learning. *CoRR*, abs/1907.06831, 2019.

[52] X. Zhao, Y. Wu, D. L. Lee, and W. Cui. iforest: Interpreting random forests via visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):407–416, Jan 2019. doi: 10.1109/TVCG.2018.2864475