Understanding the Forest: A Visualization Tool to Support Decision Tree Analysis

Catarina Maçãs and João R. Campos and Nuno Lourenço
University of Coimbra
CISUC, DEI
Coimbra, Portugal
{cmacas, jrcampos, naml}@dei.uc.pt

***Abstract*—Decision Trees (DTs) are one of the most widely used supervised Machine Learning algorithms. The algorithm constructs binary tree data structures that partition the data into smaller segments according to different rules. Hence, DTs can be used as a learning process of finding the optimal rules to separate and classify all items of a dataset. Since the algorithm relies on a decision process similar to rule-based decisions, they are easily interpretable. However, DTs can be difficult to analyse when dealing with large datasets and/or with multiple trees, i.e. ensembles. To ease the analysis and validation of these models, we developed a visual tool which includes a set of visualizations that overview and give details of a set of trees. Our tool aims to provide different perspectives over the same data and provide further insights on how decisions are being made. In this article, we overview our design process, present the different visualization models and their iterative validation. We present a use case in the telecommunications domain. In concrete, we use the visual tool to help understand how a model based on DTs decides which is the best channel (i.e., phonecall, e-mail, SMS) to contact a client.**

***Index Terms*—Visual Analytics, Heat maps, Random Forests, Decision Trees**

I. Introduction

Machine Learning (ML) algorithms are often used in classification problems so it is possible to predict the class of objects whose class is unknown [1, 2]. One example of such algorithms are Random forests (RF), an ensemble ML model that consists of many independent Decision Trees (DTs) [3]. In short, the RF model receives a testing data as input and generates the final classification/prediction by feeding the input to all DTs and summarising their results [4]. A DT is a classifier with a tree structure with leaf and decision nodes. The decision nodes split the data samples according to a rule, defined by a split feature and threshold. These splits originate each branch of the tree structure that ends in leaf nodes, containing the class label attached to them.

Despite RFs' good performance, its interpretability is usually difficult as it may contain hundreds of independent DTs. In each DT, the same feature may appear in different depth levels of the tree or even not showing at all. The comparison of hundreds DTs and their paths (i.e., rules) may be a complex and time-consuming task [4]. Still, for classification tasks, the understanding of how the ML models classified the samples is of great importance, to increase data scientists confidence in such models. For these reasons, a summary of the RF may be provided to the users to enhance its interpretability. To facilitate this task, visualization models can be employed so the users can analyse RFs more efficiently. In relation to what to visualise, several aspects may be of importance: (i) the RF's feature importance may reveal insights about their relevance in the classification process; (ii) the relationship between feature and classification may provide insights on the range of values associated with each class; and (iii) with each independent tree may enable the improvement of the model.

To explore the aforementioned challenges, we propose a visual tool that aims to ease the analysis and validation of RF models. In our tool, we use Information Visualization to overview the RF model and give details of the corresponding trees. Our main aim is to provide different perspectives over the RF model, so the user can understand how decisions are being made. To do so, we use a set of visualization models and apply multiple views technique to uncover the relationships between features and classification. We validate our approach with a use case in the telecommunications domain. More precisely, we use our tool to help data scientists understand how a RF model decides which is the best channel (i.e., phonecall, e-mail, SMS) that an operator of a telecommunication company should use to contact a client. Nonetheless, our model can be applied to any RF model. In summary, our contributions are: (1) the analysis of RF through feature and classification relationships; (2) a pixel-based matrix grid that enables the comparison of pairs of features; (3) the demonstration of our tools insights over a use case and a user study to evaluate our models and justify our decisions.

This work is funded by the project POWER (grant number POCI-01-0247-FEDER-070365), co-financed by the European Regional Development Fund (FEDER), through Portugal 2020 (PT2020), and by the Competitiveness and Internationalization Operational Programme (COMPETE 2020). This work is funded by national funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the project CISUC-UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020

II. Related Work

A. Decision Trees

Decision Trees (DTs) are usually visualised through node-link diagrams, such as tree-maps, radial trees, and icicle plots [1, 5–7]. A DT model contains two kinds of information: the tree structure (e.g., number of levels, decision and leaf nodes) and each node's data (e.g., number of items and their class). Usually, each node is represented by glyphs [8].

Tree-maps are the most common visualization model for DTs. However, it may require more canvas space. To overcome this, Ambarsari et al. [9] propose the Phytagoras Tree to position different squares, representing the data subdivisions at each level of the tree. Pham et al. [10] propose a radial node-link diagram with a fish-eye zoom technique. Regarding a more pixel-based approach, Ankerst et al. [11] use pixel bar charts, which although being scalable, cannot represent gaps between classes. Wlodyka et al. [12] created a matrix of features that shows only the percentage of the majority class at each leaf, not giving insights over the class distribution. Wang et al. [13] also propose a matrix view, enabling the analysis of the node distribution. Finally, Liu and Salvendy [1] use icicle plots to represent the distribution of items per node.

Regarding the use of multiple views, Barlow and Neville [14] created a tool to overview the structure of a selected tree through icicle plots and tree views. Liu and Salvendy [1] propose an Interactive Visual Decision Tree (IVDT) tool to improve the effectiveness of DTs in terms of classification accuracy and to reduce the tree sizes, enhancing their interpretability. IVDT combines parallel coordinates [15] and mosaic displays [16]. Also, in the IVDT process, the decision tree is visualised using the icicle plot structure as it facilitates the identification of tree topology, node relationships and sizes. A similar approach is the work of van den Elzen and van Wijk [8], in which a tree-map, a streamgraph, and a visual confusion matrix are applied to support domain experts in growing and optimising DTs.

From the related work overview, we can highlight two main characteristics: (i) the representation of node-link diagrams (i.e., tree maps, icicle plots) to enable the analysis of the DT structure; and (ii) the nodes represent only the distribution of samples per class. From our review, only [8] tackles a more complex visualization. However it may be too complex when dealing with dense tree structures. To overcome scalability issues, matrix-like visual metaphors have been used [17, 18]. However, these models may also be complex to analyse when dealing with large and complex trees.

B. Random Forest Visualization

Random Forest (RF) is an ensemble model composed of independent decision trees and, by aggregating their outputs, it can outperform individual decision tree algorithms [19, 20]. A RF model can be represented by the multiple tree graphs, but this may hinder model interpretability [4]. To interpret the RF model, it is important to understand and compare all tree's properties, which can be challenging [4].

Although the visualization of entire models is still an open challenge, some attempts have been made [4, 20–22]. For example, Min Yang et al. [23], creates a forest where each DT is represented as a 3D tree. Welling et al. [20] present an overview by splitting features and projecting them into a 3D space. ReFINE [22] combines a set of visualization models (e.g., icicle plots and scatterplots) to show the connections between proximity measures, interactions, and prototypes.

Another example is the iForest [4] that deals only with binary classification problems. It applies a multidimensional projection technique to overview path similarities between DTs. To give more information about selected features, it also presents a histogram of feature ranges by tree layer. Finally, ExMatrix [21] uses a similar approach to RuleMatrix [18] by employing a matrix structure, where rows are rules, columns are features, and cells are rules predicates.

We can conclude that visualization tools which focus only on the representation of each DT or focus only on features may not provide a good analysis of the RF results. To provide a more complete analysis of the data, multiple views can be used. Additionally, the overview of all features through a matrix metaphor may enable a better analysis and comparison of all features and their impact on the RF. However, this model may be improved by also enabling the analysis of how the values of each feature impacts the classification.

III. Random Forest Data

Random Forest algorithms combine multiple weak decision trees to achieve better results and address some of the limitations inherent to DTs, such as overfitting [2]. For the visualization of DTs, Parr and Grover [24] define a set of elements that should be highlighted in the visualization: (i) the number and distribution of samples within the decision and node, to perceive how separable the values at each feature and threshold split and to know where most samples are being routed through the decision nodes; (ii) decision node feature and split value, to understand which feature was used and where the split of observations occurred; (iii) purity of nodes, to be able to understand the confidence in each split—regarding classification, leaves with a majority target class are more reliable; and (iv) number of samples in leaf nodes, so we can distinguish trees with fewer, larger, and purer leaves (the goal) from nodes with few samples, that may indicate overfitting. In our work, we focus on the visualization of these elements, but also on the presentation of a summary view of each DT and of the whole RF model.

IV. Understanding the Forest

Our tool's main challenge is to support the analysis of RF. We provide a summary view of all DTs results, and also provide the visualization of each DT structure and classification distribution. Contrary to previous works [21] that focus on rules, we focus on each individual feature. By doing so, we aim to give the user a more interpretable view on feature relevance. This decision was also made as our tool will be used by data scientists, who aim to improve their models or gain confidence in them, but also by non-data scientists that will need to make sense of them. RF models will be applied to enable marketing operators from a telecommunication company to identify which is the best channel to contact each client. Nonetheless, we argue that this perspective is also relevant for other use cases so our tool can be applied to any RF in any domain. Link to video of tool: https://cdv.dei.uc.pt/understandingforest/.

Our tool is web-based and uses the library D3.js [25] to implement the visualizations. It is divided into two main areas: the upper part and the lower part. In the first (Fig. 5), there is a fixed dashboard, which represents the resulting feature importance of the RF model—represented with a bar chart—and a scatterplot of all DTs positioned in the x-axis according to their impurity and in the y-axis according to the maximum feature importance value. To differentiate each feature we use different colours. In the second, we present the "Classification Grid" (Fig. 5). Then, the user can change this view to the "Pyramid Matrix".A more thorough description of these models is given in Section IV-C. Additionally, the user can select any dot of the scatterplot, from the upper part, and analyse the resulting tree structure in two different ways: a Tree Visualization or "Pyramid Matrix".

A. Goals and Tasks

Our main aim is to visualise a RF model in order to give insights about the data structure and features relationships. We divided our tool in two analysis moments: an overview and a detailed view. These two moments were defined according to two goals: (**G1**) reveal relationships between features and classification; and (**G2**) understand the underlying rules. The first goal aims to fulfil the user's need to understand the model results in general and to evaluate its predictions [26]. To do so, we visualise the relationships between input features and outcome classification, enabling the user to detect important features and their influences on classification. The second goal focuses on the need to inspect the DTs and make sense of

Fig. 1. Different approaches to represent the class distribution.

classification rules [27]. By analysing rule paths, the user can drill down on the structure and detect similarities among trees. To accomplish these goals, we defined the design tasks:

T1: Encode class distribution in different moments of the tree path (**G1**). To reveal relationships between features, class distribution, and tree depth level is important to understand the influence of each feature in the classification. By summarising this information, we are able to overview for each feature how the classification changes from the upper levels of the tree (with less influence of other features) to the lower levels (with higher influence of others).

T2: Encode feature importance (**G1** and **G2**). The representation of the RF's feature importances can reveal the importance of each individual feature in the model's classification. With this, the data scientist can improve its model by discarding non important features, and a marketer may understand better which feature is more important to analyse.

T3: Encode prevalence of different classes in different feature ranges (**G1** and **G2**). Understanding the number of split occurrences by feature and corresponding values, may lead to new insights. For example, a feature with various similar split threshold values may indicate that that feature is sensitive to values in that range, having high impact on classification.

T4: Encode the features influence on classification (**G1** and **G2**). Representing the relationship between feature and classification may provide insights on the range of values associated with each class, which in turn may assist data scientists who know the context of application, determining the correct labelling of the model. The analysis of the proper functioning of the model can also be enhanced by the visualization of each independent tree and their summary.

T5: Support DTs model analysis (**G2**). Each DT has its own characteristics (e.g., number of features used, number of decision and leaf nodes). These structures may provide information on the concrete rules used to classify the data.

B. *Classification Grid*

The "Classification Grid" visualises the class distribution per feature (rows) and tree depth level (columns) (Fig. 5) to ease the understanding of how each feature can influence the classification individually (**T1**). Although the rule is what defines the final classification, we aimed to give another perspective on the features influence. Our goal is to see the class distribution along depth and the differences between depth level (**G1**). On the right side of the "Classification Grid", we visualise a histogram of the importance values by feature. In this histogram, the higher the bar, the higher the number of DTs in the corresponding importance value. Due to the small size of this graph, bars representing a reduced number of values (i.e., with smaller heights) would be difficult to notice. To overcome this, we draw a thin grey line below each bar, highlighting visually values with a reduced number of occurrences. For the "Classification Grid", we explored three approaches (Fig. 1) to represent the influence of a certain feature in the classification. Hereafter, we overview each approach in terms of visualization model and insight retrieval.

Before representing the distribution we needed to generate a summary of all DTs. To do so, we defined the "relevant class" per decision node through the true child-nodes. It is computed as follows. For each node, we retrieve the class with the higher number of samples and divide that number by the total number of samples in that node. With this, we get a percentage of relevance where nodes with only one class will have more weight than nodes with more classes. Then, we calculate the class distribution per tree depth level and feature.

Regarding the visualization, our first approach was to create a stacked bar (Fig. 1, Top). By doing so, we can perceive the features' influence in classification per tree depth level. However, we lack the understanding of how many times each feature appeared in the decision nodes.

The second approach follows the same idea, but we change the width of the stacked bar according to the number of times the feature is used in the decision nodes (Fig. 1, Middle). In this approach, we are able to understand which features are more used in the different levels, highlighting their relative importance for the RF model. However, in the first levels, as there are a smaller number of feature splits, the corresponding stacked bar is too small.

Finally, we aimed to understand if the threshold split value also influences the classification. Hence, we created a "Bar of Occurrences", in which we map the length of the bar to the minimum and maximum values of threshold splits, and then, place each split accordingly (Fig. 1 Bottom). The splits are represented by vertical bars, coloured according to the "relevant class" described earlier. A qualitative user study can be seen in Section V.

C. *Pyramid Matrix*

To represent the influence of the different features, we created the "Pyramid Matrix" (**T4**). This model applies a similar concept as the node adjacency visualization [28]. As the node adjacency matrix is symmetric, we divide the matrix in half along the diagonal and rotate 45 degrees, creating a pyramid shape. To visualise the data and explore humans' pattern recognition ability [29] and to analyse how the different pairs of features influence the classification (G1), we defined two different approaches: a Heat Map and a Pie Chart.

1) Pie Chart: We opted for a pie chart due to the fact that the number of classes are commonly small. The pie charts are placed in the corresponding cell of the Pyramid Matrix. The formula to calculate the class distribution is the same as the one described in Section IV-B. An image of this approach can be seen in Fig. 6. The aim is to ease the identification of the features that influence the classification the most.

2) Heat Map Grid: In this approach, we aimed to give more details on the range of values that may influence the classification (**T3**). In short, we created a grid in which each cell represents a range of values (Fig. 2). We calculate the class relevance (see Section IV-B) in each cell and paint each cell according to the most relevant class. We add a gradient to represent cells in which classification is more or less evenly distributed. This means that, for example, a cell in which

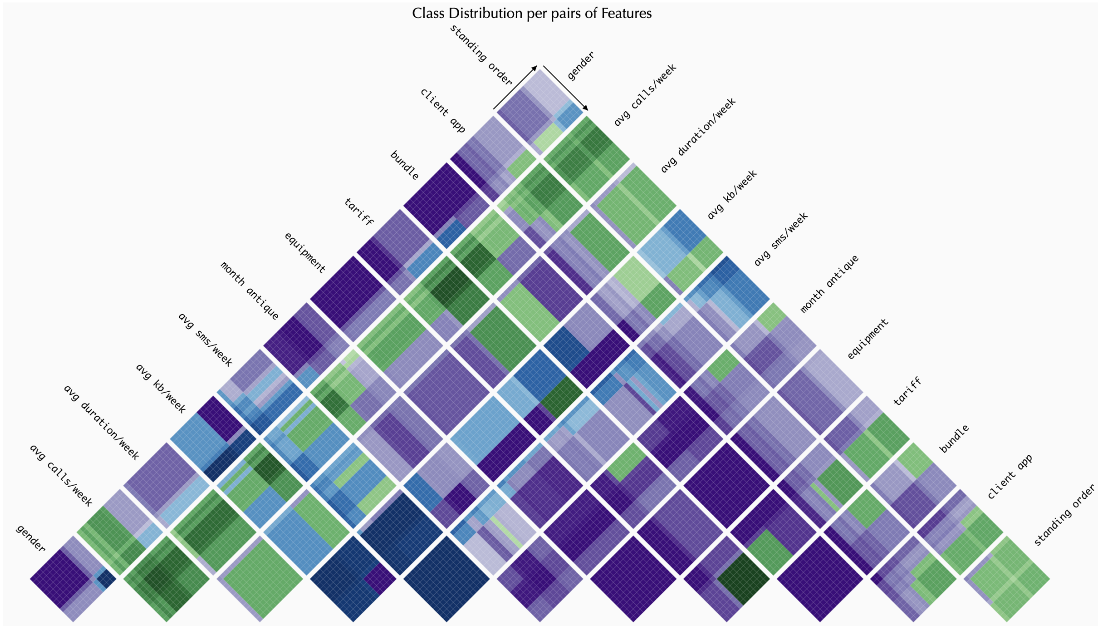

Fig. 2. The Heat Map Grid for the Pyramid Matrix. In this visualization we are able to analyse the distribution of classes and a more detailed analysis on the range of values. Each colour represent one class. The darker the colour, the more relevant the class in that cell.

there is only one class, will be painted in the colour of that class in darker colours than a cell with evenly distributed classifications. With this, ranges with higher certainty of being attributed to one class will stand out in relation to others.

To define the grid of cells, we first computed the range of feature values. As we had no access to the samples values, we calculated as follows: $(MaxValue - MinValue)/numberCells$. Then, as the split signals can differ (i.e., "bigger than" or "smaller than"), our final grid was defined by $numberCells + 2$, so values smaller or bigger than the grid limits can be represented.

D. Tree Visualization

To visualise the DTs (**T5**) we used a node-link diagram to represent single tree structures (Fig. 7). In our visualization, nodes are divided into decision nodes and leaves. Both nodes have represented the respective impurity through an horizontal bar, with a grey scale, the lower the impurity value the lighter the bar, and vice versa. Below this horizontal bar we write the number of the feature, the signal of the split, and the split threshold. We opted to write the number instead of the name to prevent overlapping texts. The samples in leaf nodes are visualised with a pie chart, representing the class distribution.

Regarding the decision nodes, we aimed to represent the class distribution (i.e., which class has a higher number of samples) but also to represent the number of samples at each tree depth level. We tried two approaches of bar charts. The first one is the application of a typical bar chart. This bar chart appears above the impurity bar and has as many bars as classes, representing the number of samples through height. This method is simple to analyse, however, with high ranges of values (nodes with many samples and nodes with fewer samples) the smaller values are difficult to see. To overcome this issue, we propose the application of a horizon bar chart. Its representation is similar to the horizon graphs [30], but applied in bars (Fig. 3). With this strategy, we aim to highlight higher values through colour, but improve the analysis of smaller values in relation to a simple bar chart. In Section V, we provide a qualitative study on both approaches.

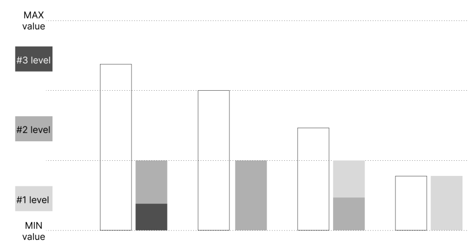

Fig. 3. Scheme of the horizon bars. First, the white bars are divided according to the dashed lines. Then, each part is collapsed and coloured as defined on the left. Smaller and darker parts are placed in front.

In our tree visualization, the links between nodes also represent information. Their colour represents the split feature used in the parent node and its thickness represents the number of samples in each link. Hence, splits which are able to divide the samples in subsets of different sizes are highlighted.

V. User Study

We conducted a user study to analyse the three visual approaches on the "Classification Grid", the horizon bar charts, and the two visualizations for the "Pyramid Grid". This study was carried out using google forms and had a total of 11 participants, aged between 26 and 36 years, of which 3 are professors, 8 are PhD students, and only two participants are female. Regarding their backgrounds, 6 participants are from Graphic and Multimedia Design, 3 are from the Artificial Intelligence area, and 2 are from Information Visualization and Data Science areas. The majority of the participants (6) referred to have basic knowledge on Decision Tree algorithms, and 6 participants referred to have a medium knowledge on

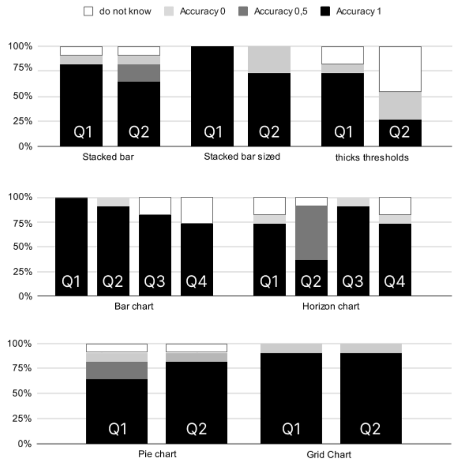

Fig. 4. Distribution of the accuracy answers in Part 1 (top), Part 2 (middle), and Part 3 (bottom). Answers are categorised as: do not know, accuracy 0 (wrong answer), 0.5 (close to correct), and 1 (correct answer).

Fig. 5. Screenshot of the initial state of ”Understanding the Forest”. In this stage, the user can visualise a Feature Importance bar chart, a scatterplot with all DTs and below, the ”Classification Grid”.

Information Visualization. The wide range of backgrounds enable us to test our models with a diverse public, which is a key point in understanding their efficiency.

We divided our test into three main parts: (Part 1) analysis of distribution, in which we analysed the ”Classification Grid” approaches; (Part 2) analysis of magnitudes, in which we tested the horizon bars with a typical bar chart; (Part 3) analysis of correlations, in which we analysed the ”Pyramid Grid” approaches. The different parts were divided into sections, mixed throughout the test, so the participant would not copy the answer from the previous question. In Part 1, we applied the same data for the three models and asked three open questions: two concerning the analysis of the feaure’s distributions, and a third one in which the participants could freely refer to what insights they could get from the image. In Part 2, we defined a set of questions of multiple choice with a “Do not know” possibility. We used a total of four images: two with a typical bar chart, and two with the horizon bar chart. We use the same data, but with different difficulties.

Finally, Part 3 is divided into two sections, one with the ”Pyramid Grid” with the pie charts and the other with the heatmap grid. As we wanted to understand if the users could properly analyse each model, the questions for each image were slightly different. We added one last open question to understand what could be analysed by the users.

A. Results

An overview of the answers in all parts of the test is provided in Fig. 4. Regarding Part 1, both questions (Q1 and Q2) were the same. We can see that, in general, the stacked bar and the sized stacked bar overpass the third approach (i.e., “Bar of Occurrences”). In the case of Q1 (“In which feature the distribution of class SMS is bigger”), we can see that all participants answered correctly. In the open answer, all participants stated that they could easily perceive the distribution of classes in the first two approaches, whereas in the “Bar of occurrences” this task proved to be more difficult. However, the last approach was referred to as a good overview of the frequency of splits along the feature values.

In Part 2 (Fig. 4), we can see that the bar chart outperforms the horizon chart in Q1 and Q2—questions about images easy to compare. From our analysis, the horizon chart performed worse, as the colours used for the different levels were not distinct enough. Then, in Q3 and Q4—questions about bar charts with similar similar—the horizon graph is slightly better than the bar chart. We argue that in the horizon chart, smaller values can be compared more easily.

Finally, in Part 3 (Fig. 4), we were able to perceive that both models were understood correctly and most participants had no doubts when answering the questions. Also, the pie chart version has less correct answers than the Heat Map Grid.

VI. Use Case

To demonstrate the interaction with our tool, we present a use case in which the RF model uses data concerning the clients characteristics from a telecommunication company. Each tree from the RF model has the following information: importance of each feature, tree impurity, and a list of nodes. Each node has the following information: id, depth, split feature, split threshold, list of child nodes, impurity, and number of samples for each class. In total, we have three classes which define the contact channel: “email”, “SMS”, and “voice”. Regarding the features, they are the following: gender, average calls per week, average GPRS duration per week, average kb per week, average SMS per week, month antique, equipment, tariff, bundle, existence of client app, and standing order. Also, the RF model was trained with input data from 1 194 clients and resulted in 500 DTs all with 5 levels of depth.

In the beginning, users can see, at the upper part, the Feature Importance Values of the RF model (Fig. 5). In this case, importance values do not vary much, however, as the bar chart is sorted from left to right by magnitude, we can easily rank all features by importance, being “tariff” the most relevant feature and “gender”, the less relevant (**T2**). On the right of this chart, we can see the DTs distribution along two axes—impurity tree value in the x-axis and maximum importance value in the y-axis (**T5**). With this distribution, we can easily see that the features “avg sms/week” and “tariff” are the ones which achieve higher values of importance in multiple DTs (i.e., are positioned in the upper part of the scatterplot). Also the features “client app” and “equipment” tend to appear in the lower part of the graph, due to lower values of importance. By hovering a certain bar, all dots whose maximum importance

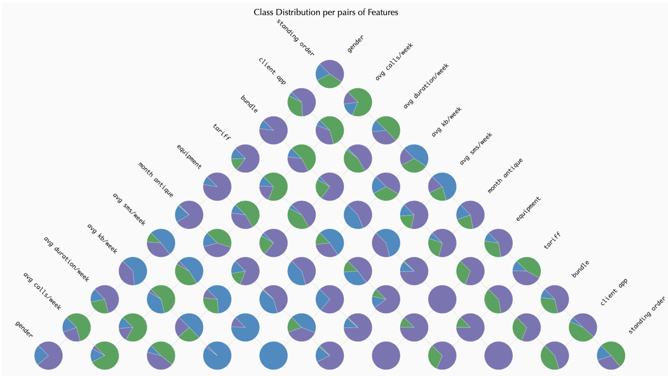

Fig. 6. The Pie Grid for the Pyramid Matrix. In this visualization we are able to analyse the distribution of classes.

value corresponds to the hovered feature are highlighted. With this, we can see that "month antique", "avg kb/week", and "avg calls/week" appear less than 4 times, which means that their importance values are low in the majority of the DTs.

To analyse in more detail the influence of each feature in each class (i.e., email, sms, or voice) the users can view the "Classification Grid", represented in the lower part of the webpage (**T1**). By analysing the distribution by tree depth level (columns), we can perceive that as we go lower in the tree, the class distribution tends to look similar between features. This is due to the fact that at each depth level, specially in the lower ones, the rules get more complex. However, we can see different distributions in the last level. For example, "month antique" presents a distribution in which there are less classification as "email", and more as "voice". This may be due to the fact that the older the clients the less prone they are to use the e-mail. When analysing all the levels of this feature, we can see a predominance of "voice" class.

On the right side of the "Classification Grid", we can see a histogram of importance values (**T2**). We can see that the majority of the importance values in features like "client app" and "bundle" are placed in the beginning of the histogram, which means that the majority of their values are low. We highlight the features "tariff" and "avg sms/week", which are the ones with higher values of feature importance. This endorses the analysis made previously over the scatterplot.

To analyse the influence of pairs of features,the user can click on the button "Pyramid Grid" (Fig. 2). When analysing the grid, it is possible to see that the "voice" class is the one that appears the most (**T3**). We can see the influence of the "avg kb/week" on the "email" class, as this feature is the one that presents more blue cells (i.e., "email" class) in the corresponding grid. Regarding the pairs of features, we can analyse the ones in which only one class appears (i.e., ["bundle", "avg sms/week"] and ["avg sms/week", "equipment"]).

To have a more general analysis of the class distribution, the user can click on the "Pyramid Pie" button (Fig. 6) (**T4**). With this, the user more easily detects pairs of features in which the classes are more even distributed (e.g., ["standing order", "gender"]) and other features in which there are more elements of one class than the others, as is the case of ["avg sms/week, "avg kb/week"], in which the main class is "email".

The user can also analyse each tree independently (**T5**). To do so, the user can click on any dot represented in the upper scatterplot. When a dot is clicked, a tree structure appears in the lower part of the tool. Besides, on the feature importance values bar chart (upper left) the feature importance values of the selected tree is represented by a black line.

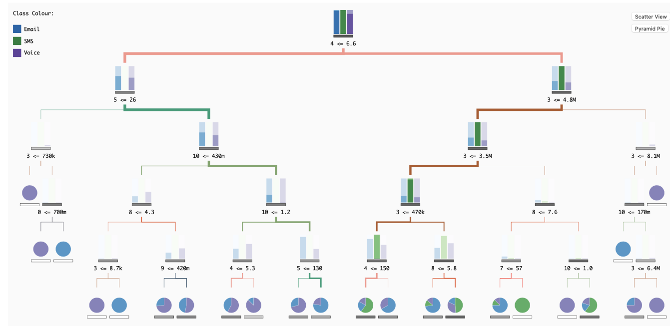

Fig. 7. The resulting tree of a DT previously selected by the user.

By selecting one tree on the scatter plot (Fig. 7), the user can perceive that right in the beginning the root node the DT separated all samples classified as "sms" from the other samples—in the child node on the left there is no green bar.By clicking in the "Pyramid Grid", we can see that this DT does not use all the features, as there are blank spaces. Also, we can view similar patterns to the RF models "Pyramid Grid". Other conclusions can be made, but for the sake of simplicity we will not continue the description.

VII. Discussion

Regarding the User Study, we could perceive that from the three approaches, the sized stacked bar was the one with more positive comments. However, this representation does not work well when the number of occurrences is low. Although the "Occurrences Bar" had lower accuracy, the participants said it was important to represent the number of splits in each feature. Hence, we should combine both stacked bars and split threshold methods, or enable the user to chose from the different methods. The feedback given at the end of the user test and their different backgrounds may provide some hints over the usefulness and efficiency of our tool. It can be generally understood, if properly contextualised, and can provide to a data scientist the necessary tools to improve the RF models. It can also be used by marketing operators, to interpret the RFs results and use this knowledge to improve their marketing campaigns.

In terms of generalisation, we argue that our tool can be used in any other tree-based ensemble models. Also, our tool can be used both for the analysis and interpretation as well for verification and improvement of RF models. Another aspect to be considered is the scalability of our tool. In the "Classification Grid", the higher the number of tree depth levels the wider the grid. This may be an issue as all depths may no longer be visible. To overcome this, we can reduce the length of the bars and add a horizontal scroll. Regarding the "Pyramid Grid" approach, the addition of more features would create a too complex shape. One possible solution would be to decrease the grid density (i.e., reduce the number of cells) so the squares for each feature pair would be smaller.

VIII. Conclusion

We presented a visualization tool for the analysis and interpretation of RF models. We focused on the analysis of features and their influence in classification. Our tool is divided into several visualization models with different purposes. First, we aim to give context on the RF results, regarding feature importance. Then, we provide a summary view on feature, class, and tree depth level. We also overview relations of features and their impact on classification through: a pie chart grid, to overview the distribution of classes among pairs of features, and a heatmap grid, to give more details on the influence of different feature values in classification. Additionally, we provide a Tree visualization, so the user can analyse each individual DT. Finally, we also presented a qualitative evaluation and a use case of our tool.

Although a more thorough analysis of the tool is recommended, from our qualitative user studies and use case, we argue that this is an important starting point to analyse RFs in different perspectives, allowing and widening its interpretability. We reckon our model requires improvements, such as the aggregation of the "Occurrence Bars" and stacked bars in the "Classification Grid". As future work, we aim to improve colour scales and the interactive details on demand. We argue that more research on how to make sense of DTs and RF is of high importance, especially due to its high applicability.

References

[1] Yan Liu and Gavriel Salvendy. Design and evaluation of visualization support to facilitate decision trees classification. *International journal of human-computer studies*, 65(2):95–110, 2007.

[2] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach, Global ed.* Pearson, 2021.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Xun Zhao, Yanhong Wu, Dik Lun Lee, and Weiwei Cui. iforest: Interpreting random forests via visual analytics. *IEEE transactions on visualization and computer graphics*, 25(1):407–416, 2018.

[5] Jianchao Han and Nick Cercone. Interactive construction of decision trees. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 575–580. Springer, 2001.

[6] Malcolm Ware, Eibe Frank, Geoffrey Holmes, Mark Hall, and Ian H Witten. Interactive machine learning: letting users build classifiers. *International Journal of Human-Computer Studies*, 55(3):281–292, 2001.

[7] Jianting Zhang, Le Gruenwald, and Michael Gertz. Vdm-rs: A visual data mining system for exploring and classifying remotely sensed images. *Computers & Geosciences*, 35(9):1827–1836, 2009.

[8] Stef Van Den Elzen and Jarke J Van Wijk. Baobabview: Interactive construction and analysis of decision trees. In *2011 IEEE conference on visual analytics science and technology (VAST)*, pages 151–160, 2011.

[9] Erlin Windia Ambarsari, Aulia Ar Rakhman Awaludin, Andri Suryana, Purni Munah Hartuti, and Robbi Rahim. Basic concept pythagoras tree for construct data visualization on decision tree learning. *Journal of Applied Engineering Science*, 17(4), 2019.

[10] Nguyen-Khang Pham, Thanh Nghi Do, François Poulet, and Annie Morin. Interactive exploration of decision tree results. In *Applied Stochastic Model and Data Analysis International Conference*, pages 152–160, 2007.

[11] Mihael Ankerst. Visual data mining with pixel-oriented visualization techniques. In *ACM SIGKDD Workshop on Visual Data Mining Proc.*, page 23, 2001.

[12] A Wlodyka, R Mlynarski, G Ilczuk, E Pilat, and W Kargul. Visualization of decision rules-from the cardiologist's point of view. In *2008 Computers in Cardiology*, pages 645–648. IEEE, 2008.

[13] Jun Wang, Bei Yu, and Les Gasser. Concept tree based clustering visualization with shaded similarity matrices. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 697–700, 2002.

[14] S Todd Barlow and Padraic Neville. Case study: Visualization for decision tree analysis in data mining. In *infovis*, pages 149–152, 2001.

[15] Alfred Inselberg. The plane with parallel coordinates. *The visual computer*, 1(2):69–91, 1985.

[16] Michael Friendly. Visualizing categorical data: Data, stories, and pictures. In *Proceedings of the Twenty-Fifth Annual SAS Users Group International Conf.*, 2000.

[17] Federica Di Castro and Enrico Bertini. Surrogate decision tree visualization interpreting and visualizing black-box classification models with surrogate decision tree. In *CEUR Workshop Proc.*, volume 2327. CEUR-WS, 2019.

[18] Yao Ming, Huamin Qu, and Enrico Bertini. Rulematrix: Visualizing and understanding classifiers with rules. *IEEE transactions on visualization and computer graphics*, 25(1):342–352, 2018.

[19] Michael Behrisch, Benjamin Bach, Nathalie Henry Riche, Tobias Schreck, and Jean-Daniel Fekete. Matrix reordering methods for table and network visualization. In *Computer Graphics Forum*, volume 35, pages 693–716. Wiley Online Library, 2016.

[20] Soeren H Welling, Hanne HF Refsgaard, Per B Brockhoff, and Line H Clemmensen. Forest floor visualizations of random forests. *arXiv preprint:1605.09196*, 2016.

[21] Mário Popolin Neto and Fernando V Paulovich. Explainable matrix-visualization for global and local interpretability of random forest classification ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1427–1437, 2020.

[22] Natalia Kuznetsova, Michel Westenberg, K Buchin, K Dinkla, and SJ van den Elzen. Random forest visualization. *Eindhoven University of Technology*, 2014.

[23] Min Yang, Hexin Xu, Dingju Zhu, and Huijuan Chen. Visualizing the random forest by 3d techniques. In *Internet of things*, pages 639–645. Springer, 2012.

[24] Terence Parr and Prince Grover. How to visualize decision tree, 2020.

[25] Mike Bostock. D3.js - data-driven documents, 2012.

[26] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[27] Or Biran and Kathleen R McKeown. Human-centric justification of machine learning predictions. In *IJCAI*, pages 1461–1467, 2017.

[28] Mershack Okoe, Radu Jianu, and Stephen Kobourov. Node-link or adjacency matrices: Old question, new insights. *IEEE Transactions on Visualization and Computer Graphics*, 25(10):2940–2952, 2019.

[29] Colin Ware. *Information visualization: perception for design*. Morgan Kaufmann, 2019.

[30] Jeffrey Heer, Nicholas Kong, and Maneesh Agrawala. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1303–1312, 2009.