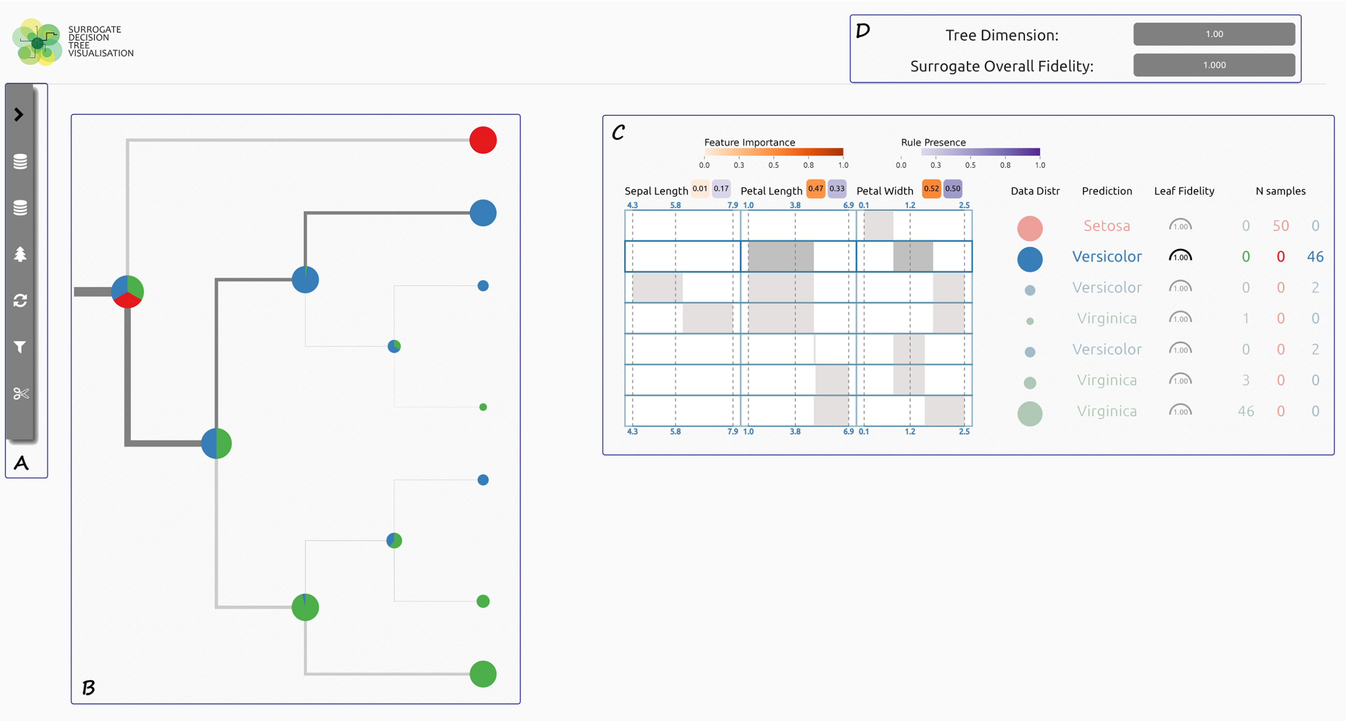
Surrogate Decision Tree Visualization

Interpreting and Visualizing Black-Box Classification Models with Surrogate Decision Tree

Federica Di Castro
Università di Roma La Sapienza
Rome, Italy
dicastro.1597561@studenti.uniroma1.it

Enrico Bertini
New York University Tandon School of Engineering
Brooklyn, New York
enrico.bertini@nyu.edu



Tree Dimension: 1.00

Surrogate Overall Fidelity: 1.000

Data Distr	Prediction	Leaf Fidelity	N samples		
●	Setosa	1.00	0	50	0
●	Versicolor	1.00	0	0	46
●	Versicolor	1.00	0	0	2
●	Virginica	1.00	1	0	0
●	Versicolor	1.00	0	0	2
●	Virginica	1.00	3	0	0
●	Virginica	1.00	46	0	0

Figure 1: Understanding the behavior of a black-box Machine Learning model using the explanatory visual interface of our proposed technique. Interacting with the user-interaction-panel (A) a user can choose a combination of model and dataset that they desire to analyze. The built model is a surrogate decision tree whose structure can be analyzed in the tree Panel (B) and the details of the leaves are featured in the Rule Panel (C). The user can interact with the tree, collapsing any node they see fit and automatically updating the performance Overview (D).

ABSTRACT

With the growing interest towards the application of Machine Learning techniques to many application domains, the need for transparent and interpretable ML is getting stronger. Visualizations methods can help model developers understand and refine ML models by making the logic of a given model visible and interactive. In this paper we describe a visual analytics tool we developed to support developers and domain experts (with little to no expertise in ML) in understanding the logic of a ML model without having access to the internal structure of the model (i.e., a model-agnostic method). The method is based on the creation of a "surrogate" decision tree which simulates the behavior of the black-box model of interest and presents readable rules to the end-users. We evaluate the effectiveness of the method with a preliminary user study and analysis of the level of fidelity the surrogate decision tree can reach with respect to the original model.

CCS CONCEPTS

• **Human-centered computing** → **Dendrograms**; *User interface toolkits*; • **Information systems** → *Data analytics.*

ACM Reference Format:

Federica Di Castro and Enrico Bertini. 2019. Surrogate Decision Tree Visualization: Interpreting and Visualizing Black-Box Classification Models with Surrogate Decision Tree. In *Proceedings of (IUI Workshops'19).* ACM, New York, NY, USA, 5 pages.

Author Keywords

Machine Learning; Interpretability; Classification; Explanation; Visual Analytic; Decision Tree; Dendrograms; User Interface

ACM Classification Keywords

Information interfaces and presentation: User Interfaces.
Software: Design Tools and Techniques.

IUI Workshops'19, March 20, 2019, Los Angeles, USA

INTRODUCTION

In this paper we propose an interactive visualization based on decision rules: treating every model as an unknown black-box we use a Decision Tree to replicate the prediction done by a model in a classification problem and we visualize it with the purpose of using the rules of the decision tree to propose simple yet effective explanations towards the logic that the model adopts for its classification.

With the growing adoption of machine learning techniques, there is an increasing demand for research towards making machine learning models transparent and interpretable [7]; especially in critical areas such as medicine [1], security and law.

In this paper we will follow two definitions of interpretability: (1) *interpretability is the degree to which a human can understand the cause of a decision* [4] and (2) *interpretability is the degree to which a human can consistently predict the model's result* [3]. These definitions provide an intuition regarding the type of user who may be in need of interpretability methods: data scientists or developers for model debugging and validation; end-users (typically domain experts) for understanding and gaining trust in the model in the process of decision making; and regulators and lawmakers for making sure a given system is fair and transparent.

RELATED WORK

In this section we discuss related work that may share some of our goals through different techniques; both on the approach in the generation of rules to describe a model and the need for interpretability in Machine Learning.

Rule Generation and Visualization

Many attempts have been performed in summarizing a model though simple and effective rules: rule lists, rule tables, decision rules have been used in the community to describe ML models. Very established in the community is also LIME (Local interpretable model-agnostic explanations) which creates a local surrogate model and computes local weights which one can use for interpretation of single instances [7]. Using LIME allows to have short, simple, human-friendly explanations that can help any user gain insights about how the model computes a prediction of a specific instance. The same authors later developed Anchors [8], an improved version that computes local explanatory rules instead of weights.

Methods exists also for *global* explanations of model. A current method is to learn if-then rules that globally explain the behavior of black-box models by first gathering conditions that are important at instance level and then generalizing them into rules that are meant to be descriptive of the overall behavior of the model [6]. Another project that has much in common with this proposal is *RuleMatrix* by Ming et al. [5], which derives surrogate rules from an existing black-box model and visualizes them with a custom matrix visualization. Our solution is a follow-up of *RuleMatrix* with an additional innovation: the use of a decision tree structure to compute and visualize the rules. The tree structure, which is explicitly visualized, helps navigate the rules in a hierarchical fashion and as such makes it easier to spot rules of interest.

Interpretable Machine Learning

Understanding a computer-induced model is often a prerequisite for users to trust the model's predictions and follow the recommendations associated with those predictions [2]. In order for a user to trust a model in the process of decision making, it is necessary that the model be transparent or that methods are used to enable its users to verify and understand its behavior. A clear example of the necessity of interpretability is presented in [9][1], where a interpretability method enabled a group of experts to identify a major fault in a model used for medical predictions.

Ilknur Kaynar Kabul, a Senior Manager in the SAS Advanced Analytics division, describes in a post about interpretability desirable characteristics of an interpretable model: *Transparent* - it can explain how it works and/or why it gives certain predictions; *Trustworthy* - it can handle different scenarios in the real world without continuous control; *Explainable* - it can convey useful information about its inner workings, for the patterns that it learns and for the results that it gives. These are goals we took into consideration when building our Surrogate Tree Visualization.

BUILDING SURROGATE TREES

In the following section we introduce our steps in creating our 'Surrogate Decision Tree Visualization'.

Goals and Targets users

In our paper we target as potential user of our tool not only model developers but also domain experts that are impacted by the machine learning techniques (e.g., health care, finance, security, and policymakers). Model developers use interpretability with the goal of model debugging: understanding a model with the final goal of refining and improving its classification. Domain experts, who may have little to no-knowledge in ML, have the goal to understand how the model behaves and what conclusions it draws when making its classification. In both cases, there is a need for profound and deep understanding of what the model does.

Our tool aims to facilitate the answer to the following questions:

Q1 What rules did the model learn?

Q2 Which of these rules can be considered descriptive of the model?

Q3 What are the behaviors of the model that the surrogate is not able to simulate?

Q4 What are the most important features used by the model?

Decision Trees

A decision tree is a simple recursive structure that expresses a sequential process of classification. Every tree-based model splits the data multiple times according to multiple threshold values of the features. At each node a splitting of the dataset occurs: going forward the dataset keeps getting split into multiple subsets until each subset, if every leaf in the tree is pure, contains instances from one class only.

The reason why we chose to use a Decision Tree as the surrogate model is the simplicity of its rules and the natural tree-based visual representation one can build with it. Starting from the root node one can check the next nodes and trace the path down to the leaf to form a rule.

The following formula describes the relationship between outcome $\hat{y}$ and the features x:

$$\hat{y_i} = \hat{f}(x_i) = \sum_{j=1}^{N} c_j I\{x_i \in R_j\}$$

Each instance x_i reaches exactly one leaf node which can be described as a subset R_j of the dataset. The identity function $I\{.\}$ has the purpose of representing the combination of rules at each of the internal nodes.

It's important to clarify that we use decision trees as a way to simulate a black-box model. To achieve this purpose we do not train the tree using the original data but rather use the labels obtained from the original model as training data for the decision tree. This, in turn, allows us to build a tree whose rules simulate the original model.

Feature Importance. The overall importance of a feature in a decision tree can be computed by going through all the splits for which the feature was used and adding up how much it has improved the predictions in the child nodes compared to the parent node (e.g., measured as decrease of Gini index). The sum of all the values of importance is scaled to 100, so that the interpretation for each feature importance is the percentage of the overall importance.

Rule Presence. In addition to feature importance we compute a second metric that we call *rule presence*. The purpose of this metric is to give more weight to features that appear more often in the tree (in multiple splits). The metric is computed as follows:

$$RP_{feat_i} = \frac{Number\ of\ Nodes\ involving\ feature_i}{Number\ of\ Internal\ Nodes}.$$

Disadvantages

Decision trees have a number of disadvantages as model interpretation tools. First, the number of nodes increases exponentially with depth, therefore the more terminal nodes, the more difficult it becomes to understand the decision rules of a tree. Even with a moderate number of features it is not unlikely to have trees with hundreds of nodes and links. Second, the same feature may occur multiple times at different levels in the tree; making it hard for the a viewer to understand how a feature is used by the model across all rules it generates.

In our solution, we provide two techniques to mitigate this issue: (1) We enable the user to interactively contract and expand the tree at different levels of granularity; (2) We provide a coordinated supplementary view which visualizes the rules generated by the tree in a tabular format. As explained in the section explaining how our visualization works, our design aligns rules so that a viewer can see how a given feature is used across the whole set of rules.

Performance Evaluation

There are three main aspects we take into account when evaluating the performance of a surrogate model:

- *Fidelity.* The accuracy with which the tree can simulate the original black-box model;
- *Speed.* The time needed to generate the tree as well as the time performance of the interactive functions (to explore the tree interactively);
- *Complexity.* The overall complexity of the surrogate tree, measured as the number of nodes in the tree.

The overall fidelity of the tree is computed as the ratio of samples for which the tree predicts the outcome of the simulated model correctly. The fidelity of a single node computes the same measure restricting it to the samples that fall into the node.

Table 1: Mean Time and Complexity requirements to reach maximum fidelity with the available datasets.

Dataset	Mean Time (s)	Mean Nodes
Iris	0.131	13
Fico	22.841	190
Housing	13.818	200
Demographic	25.262	319
Car	19.141	238
Cancer	1.128	32

Table 2: Fidelity, time and Complexity for the FICO Dataset with different models and values of max Depths

Model	maxDepth	fidelity	time (s)	nNodes
KNN	6	0.940	13.01	111
KNN	8	0.976	27.71	235
KNN	10	0.996	39.62	337
KNN	12	1.000	41.57	353
LDA	6	0.954	13.08	113
LDA	8	0.988	26.22	221
LDA	10	0.998	31.59	259
LDA	13	1.000	31.95	271
MLPC	6	0.942	12.02	101
MLPC	8	0.977	23.99	203
MLPC	10	0.993	34.49	291
MLPC	12	0.999	37.73	319
MLPC	14	1.000	39.59	325

Time performance is calculated as the combination of time necessary to perform the fitting of the model and the fitting of the surrogate but also the time necessary to explore the tree and gather the necessary information for the visualization.

Complexity is measured as the number of nodes in the tree: in fact, this impacts not only the time needed to generate the tree data, but also the amount of information displayed in the visualization. As we will see in the following discussion, there is a trade-off between complexity and fidelity.

We tested our tool with 6 datasets: Iris Flower, Cancer, Car Safety, Housing Price, Demographic Life Quality, FICO Risk. We also used 5 distinct type of models to simulate as black-boxes: KNN, Linear Discriminant Analysis, MultiLayer Perceptron, AdaBoost, Support Vector Machine. Table 1 and table 2 show the results of our tests.

As we can see from Table 1, it is possible with reasonable complexity and time requirements to reach maximum fidelity with any combination of model and dataset.

What is more interesting is that for a user that is willing to trade-off fidelity in order to obtain a less complex visualization it is possible to do so. In Table 2, we analyze the FICO Dataset (one of the most complex datasets used in our evaluation). As one can see, for almost every model it is possible to save about 20-30 seconds and a complexity of 160-240 nodes by reducing fidelity only of 4%-6%. Therefore, depending on the need of the user there may be plenty of room for compromising on fidelity without reaching too low values.

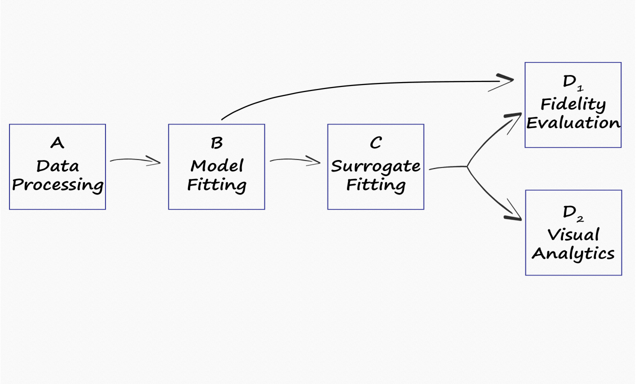


Figure 2: The pipeline the tool follows once a dataset and a model have been chosen: there is a data pre-processing step (A) followed by the fitting of the model that needs to be simulated (B). Once the model is fitted, its prediction is then given as input to the surrogate model block (C) and to the Fidelity Evaluation Block (D_1). Once the surrogate has been fitted, its prediction reached the Fidelity Evaluation Block and is compared with the output of the Model, which it is meant to replicate. Simultaneously, the tree and its data are visualized in block (D_2).

VISUALIZING SURROGATE TREES

To gather all necessary data for the visualization, the tool follows a specific pipeline (see Fig. 2) which has the purpose of obtaining both information about the performance of the surrogate in replicating the model (Fidelity Evaluation) and the data contained in each node of the tree, necessary for the Visual Analytics step. The visualization itself is made of 4 components (see Fig. 1). A user interaction panel (A), a tree panel (B), a rule panel (C) and performance overview (D).

User interaction panel

The interaction panel allows the user to select the combination of model and dataset to use as well as parameters such the preferred value for Max Depth for the decision tree. Being able to choose in advance the max depth is helpful for those users who want to guarantee the maximum fidelity possible, or to those users who want to start with a small tree. Once the surrogate is generated, the user can use the *fidelity slider* to automatically cut the tree at the level that provides the desired fidelity.

Performance Overview

The two bars in the top-right have the purpose of keeping track of updates performed by the user in case of interaction with the tree. *Surrogate Overall Fidelity* provides the measure of the starting fidelity of the surrogate: this is the maximum value that can be reached, in fact any interaction with the tree itself that does not involve an increment of max-depth can only negatively affect the overall fidelity. *Tree Dimension* shows the current proportion of the tree that is shown with respect to the original dimension. When the tree is loaded the first time tree dimension is 100%, but with any toggling of the tree it decreases to represent the current proportion. The combination of the two bars is meant to help users find a good balance between tree size (i.e., complexity) and fidelity.

Tree Panel

The *tree panel* shows the tree as a node-link diagram. Each node is depicted by a pie chart which shows number of instances with size and proportion of labels with the segments of the pie chart.

Each edge has a width proportional to the number of instances that follow the path. Hovering a node one can see its details in a tooltip which includes: number of samples, fidelity and the rule that characterizes the node. When a node is clicked it collapses and becomes a leaf node generating an update of our surrogate model. As a consequence, the performance numbers, as well as the panel that shows them, updates to reflect the change.

Rule Panel

The *rule panel* is structured as a table: each column is a feature of the dataset and each row is a leaf in the tree and, as such, a rule. Reading the table by row it is possible to identify the rules that describe the model in its entirety thus providing an answer to question Q1.

The predicates of the rules are depicted as rectangular ranges, which show, for a given rule-feature combination, what the constrains of the rule are, that is, the range depicts constraints of the type $lb \leq feature \leq ub$.

For each row/leaf we also provide details of the distribution of instances that satisfy the rule, including information about rule cardinality and distribution across the existing classes, the prediction of the rule, and the fidelity. These details are crucial to judge the fidelity and relevance of the rules: questions Q2 and Q3.

Finally, next to every feature we provide details about feature relevance using the two metrics *feature importance* and *rule presence* which characterize each feature in terms of how relevant it is (according to relevance computed from the decision tree) and in how many rules it is used. These details provide the user with information necessary to understand the role of each feature, which covers the needs expressed in question Q4.

USABILITY STUDY

In order to provide preliminary evidence of the utility and usability of our proposed solution, we conduced a small user study. The purpose of the study was to verify whether people would be able to understand the interface and solve the most basic interpretability problems it proposes to solve.

Which Users

We selected 5 users, ranging from 25 to 35 years of age, with extended experience in Machine Learning: Master's degree students in Data Science, PhD students in Computer Engineering and Data Visualization. We chose expert users to target the population who would be interested in using our method as a model verification and debugging method.

Tasks and Results

We divided the study in two phases: a training phase in which users were shown the IRIS flower dataset and asked a series of question, and a testing phase in which users where asked only three questions to test the usability of the system.

The questions were aimed at testing whether the users were able to formally express the rules, by reading the table, and to evaluate their fidelity.

The training phase was meant to introduce the interface to the participants to make sure they were ready to use it in the test phase. An important goal of this first phase was to make sure that every component of the tool was readable and that the participants understood the need to find a balance between complexity and fidelity. The participants were asked specific questions on how to read the rules in the rule panel and their characteristics. All of them provided correct answers to these questions, confirming they had understood how to use it.

In the testing phase we gave our participants time to load the FICO dataset (a much more complex one than the previously shown one) and to interact with the tool for the time they felt necessary to provide a description the model. Two of the subjects spent almost an hour observing and analyzing the dataset and they were able to draw many significant conclusions regarding the relationships between some of the features and the classification performed from the datasets. One user in particular was able to spot many outliers observing the pie chart that summarizes each leaf and the features involved in the corresponding rules.

CONCLUSIONS AND FUTURE WORKS

We presented a description and a preliminary evaluation of our surrogate tree visualization. This visualization can be used by domain experts as well as data scientists to gain an understanding of model behavior, including what the model does right and where issues may arise. Application of such method includes healthcare professionals who want to understanding ML models for diagnostics and other medical outcomes; finance experts who use models for prediction or credit scoring; or even in security to understand how a model detects fraudulent transactions or other illegal activities.

The work presented here is very preliminary. We indent to extend the tool further in many ways. We want to provide a tighter link between the tree and the table and find a way to make it scale to a higher number of features. We also need to test the methods with a much larger number of data sets and models. In particular, we need to verify the extent to which decision trees can simulate more complex models and how fidelity changes when some of the parameters we use change (e.g., tree complexity).

The most important aspect that we will be working on in the future is a wider testing of our tool. We need to test the methods more formally to better understand the extent to which it helps answer the questions we outlines in the introduction. More importantly, we need to better understand how interpretability is affected by visual representation. For this purpose we plan to develop controlled experiments to compare our alternative visual representations. In particular, we deem it important o better understand how the tree representation and the tabular representation of the rules compare and how they score in comparison to a simple textual list of rules.

REFERENCES

[1] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*. ACM, New York, NY, USA, 1721–1730. https://doi.org/10.1145/2783258.2788613

[2] Alex A. Freitas. 2014. Comprehensible Classification Models: A Position Paper. *SIGKDD Explor. Newsl.* 15, 1 (March 2014), 1–10. https://doi.org/10.1145/2594473.2594475

[3] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. 2016. Examples are not enough, learn to criticize! Criticism for Interpretability. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 2280–2288. http://papers.nips.cc/paper/6300-examples-are-not-enough-learn-to-criticize-criticism-for-interpretability.pdf

[4] Tim Miller. 2017. Explanation in Artificial Intelligence: Insights from the Social Sciences. *CoRR* abs/1706.07269 (2017). arXiv:1706.07269 http://arxiv.org/abs/1706.07269

[5] Yao Ming, Huamin Qu, and Enrico Bertini. 2018. RuleMatrix: Visualizing and Understanding Classifiers with Rules. *CoRR* abs/1807.06228 (2018). arXiv:1807.06228 http://arxiv.org/abs/1807.06228

[6] Nikaash Puri, Piyush Gupta, Pratiksha Agarwal, Sukriti Verma, and Balaji Krishnamurthy. 2017. MAGIX: Model Agnostic Globally Interpretable Explanations. *CoRR* abs/1706.07160 (2017). arXiv:1706.07160 http://arxiv.org/abs/1706.07160

[7] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *CoRR* abs/1602.04938 (2016). arXiv:1602.04938 http://arxiv.org/abs/1602.04938

[8] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-Precision Model-Agnostic Explanations. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[9] G. Richards, V.J. Rayward-Smith, P.H. Sönksen, S. Carey, and C. Weng. 2001. Data mining for indicators of early mortality in a database of clinical records. *Artificial Intelligence in Medicine* 22, 3 (2001), 215 – 231. https://doi.org/10.1016/S0933-3657(00)00110-X