Journal of the Brazilian Computer Society, 2025, 31:1, doi: 10.5753/jbcs.2023.3982

This work is licensed under a Creative Commons Attribution 4.0 International License.

A Human-Centered Multiperspective and Interactive Visual Tool For Explainable Machine Learning

Bárbara Lopes [**Federal University of Minas Gerais** | ***barbaragcol@dcc.ufmg.br***]
Liziane Santos Soares [**Federal University of Minas Gerais** | ***liziane.soares@dcc.ufmg.br***]
Marcos André Gonçalves [**Federal University of Minas Gerais** | ***mgoncalv@dcc.ufmg.br***]
Raquel Oliveira Prates [**Federal University of Minas Gerais** | ***rprates@dcc.ufmg.br***]

Federal University of Minas Gerais, Av. Antonio Carlos 6627, Belo Horizonte, MG, Brazil

Received: 28 December 2023 • **Accepted:** 12 November 2024 • **Published:** 20 January 2025

Abstract Understanding why a trained machine learning model makes some decisions is paramount to trusting the model and applying its recommendations in real-world applications. In this article, we present the design and development of an interactive and visual approach to support the use, interpretation and refinement of ML models, whose development was guided by user's needs. We also present **Explain-ML**, an interactive tool that implements a visual *multi-perspective* approach to the support interpretation of ML models. Explain-ML development followed a Human-Centered Machine Learning strategy guided by the target (knowledgeable) users' demands, resulting in a multi-perspective approach in which interpretability is supported by a set of complementary visualizations under several perspectives (e.g., global and local). We performed a qualitative evaluation of the tool´s approach to interpretation with a group of target users, focused on their perspective regarding Explain-ML helpfulness and usefulness in comprehending the outcomes of ML models. The evaluation also explored users' capability in applying the knowledge obtained from the tool's explanations for adapting/improving the current models. Results show that Explain-ML provides a broad account of the model's execution (including historical), offering users an ample and flexible exploration space to make different decisions and conduct distinct analyses. Users stated the tool was very useful and that they would be interested in using it in their daily activities.

Keywords: Interpretability, Machine Learning Model, Computer Interaction, Visualization

1 Introduction

Machine learning (ML) models have been widely used nowadays, though mostly as "magic black boxes". They are applied in many different domains, with multiple and distinct objectives (predictive, analytical, etc) – but the way they generate their results is far from being well understood, specially by the average or final user [Ramos *et al.*, 2019]. This understanding directly impacts the reliability of models and their predictions, particularly when they are used for making decisions that impact human life in domains such as medical and biomedical, healthcare, cybersecurity, finances, law, crime/terrorism prevention, recommender systems, credit analysis, fraud detection and anomaly detection [Caruana *et al.*, 2015; Linardatos *et al.*, 2021; Choung *et al.*, 2023; Nakao *et al.*, 2023; Yang *et al.*, 2023]. In general, models are evaluated through a single performance metric (e.g., accuracy) on standard benchmarks. But real-world data may show variations or have particular and very specific idiosyncrasies that affect model behavior and performance [Ribeiro *et al.*, 2016]. In this context, providing explanations for model predictions favors users' interpretability and acceptance [Tolomei *et al.*, 2017], which is aligned with the field of eXplainable Artificial Intelligence (XAI)[Linardatos *et al.*, 2021; Rong *et al.*, 2022; Schneider, 2024; Longo *et al.*, 2024].

Interpretable explanations can be used to understand how the model itself makes decisions to generate its results. Increasingly, users and businesses that employ ML models are expressing the need to understand how those models generate their predictions. If users do not trust a model or its predictions, it is likely that they will have caveats in using them. Moreover, the need for model interpretability meets the demand for transparency advocated by various international and national government institutions. The European Parliament, for example, has adopted the General Data Protection Regulation (GDPR) which became law in May 2018 [Goodman and Flaxman, 2017; Shneiderman, 2020]. It defends the right of users to have access to an explanation of the logic involved in automated decision-making systems, many of which use ML techniques [Guidotti *et al.*, 2018b]. The explanation behind a prediction can also help to refine the model in order to improve it because it can be used as feedback to the model itself regarding why it hit or missed a result. Thus, the lack of these explanations can be related to practical and ethical issues [Guidotti *et al.*, 2018b].

We highlight that the concepts of interpretability and explainability have been broadly used, but not always consistently. In some works, authors make an explicit distinction between them[1], whereas in others, these concepts are used interchangeably and sometimes even considered synonymous [Mohseni *et al.*, 2021; Vilone and Longo, 2021; Zhou *et al.*, 2021; Cao *et al.*, 2024]. In this paper, we will use **interpretability and explainability as synonymous, as**

[1] Mohseni *et al.* (2021) define interpretability as the ability to support users in understanding the model decision making process and predictions; and explainability as the ability to explain the underlying model and its reasoning with accurate and user comprehensible explanations.

the ability to explain or present the results of ML models by means of elements understandable by a human being [Guidotti *et al.*, 2018a] such as terms, image fragments, graphics and visualizations [Doshi-Velez and Kim, 2017; Guidotti *et al.*, 2018b; Hall and Gill, 2018].

Accordingly, several efforts have been devoted to the problem of explaining ML-based decision models [Linardatos *et al.*, 2021]. Some works have focused on models that are more explainable or understandable to the user [Caruana *et al.*, 2015; Lakkaraju *et al.*, 2017; Guidotti *et al.*, 2018b]. Other approaches focus on explaining ML models as black boxes in a model-agnostic way [Ribeiro *et al.*, 2016; Kononenko *et al.*, 2010; Turner, 2016; Ribeiro *et al.*, 2018; Lundberg and Lee, 2017; Ming *et al.*, 2019; Vidovic *et al.*, 2016; Guidotti *et al.*, 2018a; Erik and Kononenko, 2010; Guidotti *et al.*, 2018b; Singh *et al.*, 2016]. All of these efforts are related to the need to make models more understandable and reliable.

Nevertheless, the *explicit needs* of users of ML systems, being them ML specialists or end-users that only consume the output of the system, concerning interpretability and decision-making aids, have rarely been considered while designing the solutions. Consequently, there is a lack of works focused on the intersection between user demands and ML systems. This intersection has recently fostered the Human-Centered Machine Learning (HCML) area, which advocates that ML research and systems should be considered from a more human-centered perspective. HCML aims to be an interdisciplinary area that brings together the perspectives of HCI (*Human-Computer Interaction*) and ML (Machine Learning) [Ramos *et al.*, 2019; Gillies *et al.*, 2016; Dudley and Kristensson, 2018], and presents several challenges [Capel and Brereton, 2023].

In this work, we present an HCML design process to define an interpretability approach to support users in better comprehending the ML model results and thus being able to improve or adjust the model. Our focus was on assisting *knowledgeable users*, i.e. users that have some knowledge of ML models, acquired by having employed ML models in the past in some application. As a proof-of-concept, based on this approach, a version of Explain-ML instantiated with Random Forests (as ML model) applied to automatic text classification (as ML task) consolidated approach was implemented. Finally, a qualitative study was conducted with a group of six target users using the implemented version of Explain-ML. The results show that users found the tool helpful and useful in understanding the outcomes of the ML models and in using this information to decide strategies to be adopted to improve the model.

Briefly, the main contributions of the work are:

1. *The design and development of an interactive and visual approach to support the use, interpretation and refinement of ML models and can be useful to other researchers/developers interested in explainability systems who can build on them to generate new systems or evaluate existing systems;*
2. *The tool that implements the proposed multi-perspective approach that can be applied to many known ML models and has epistemic potential;*
3. *A qualitative evaluation of the tool with a group of target users, which allowed collecting indicators relating to the support offered by the tool in defining the RF model, providing users with a broad exploration space for interpretability questions that can be flexibly exploited considering the users' own goals, background and expectations.*

This article is structured as follows: Section 2 covers related work and discusses the concepts related to this work. Section 3 describes how the requirements gathering and design process were carried out. Section 4 introduces the proposed approach. Section 5 introduces the Explain-ML tool, that implements the proposed approach. Section 6 presents details on the user study conducted to evaluate the proposed approach, followed by a discussion on the results (Section 7). Finally, Section 8 presents our final considerations and future works.

2 Related work

In this section we present concepts that support this work. We address the context of interpretability of ML models, existing approaches for ML model building, and the HCML and Interactive Machine Learning (IML) Systems Design.

2.1 Interpretability approaches

According to Guidotti *et al.* [2018b], in the context of interpretability of ML models, there are two main scopes of work: (i) Transparent Box Design: focused on the problem of developing interpretable and transparent models along with their predictions; and (ii) Black Box Explanation (also called Post-Hoc [Linardatos *et al.*, 2021]): focused on explaining how the model generates its results, considering it as a black box. The latter can be further broken down into three other scopes of work: (ii.1) *Model Explanation*: aims at providing a global explanation of the black box model; (ii.2) *Outcome explanation*: based on an input instance and a model, aims at presenting an explanation for the model outcome on specific instances; (ii.3) *Model inspection*: aims at providing visual or textual representations to help in the understanding of the black box model and its predictions.

Our approach focuses on the **Black Box Explanation**, more specifically, on **Outcome explanation** and **Model Inspection**. Specifically concerning Model Inspection, [Tamagnini *et al.*, 2017] highlighted the differences among approaches, considering the following activities: inspection of models that perform multi-classification; prediction inspection of a given instance within its space; approaches that allow the active participation of the user who interacts with some visual interface and provides feedback to adjust the model; use of the importance of features as an inspection tool.

The last three aforementioned activities are covered in the approach proposed by this paper. Our approach addresses several elements, such as terms, graphics and visualizations for the interpretability of the results.

The **outcome explanation approachs** focuses on providing an explanation for the Black Box outcome of a single instance. The overall and internal logic of the ML Model is not the concern in this case. Several approaches explain specific types of models such as Deep Neural Network, SVM, Non-Linear Models and Tree Ensemble. Others, are model agnostic so they can be applied with any ML model [Ribeiro *et al.*, 2016; Erik and Kononenko, 2010; Guidotti *et al.*, 2018a,b; Ribeiro *et al.*, 2018; Lundberg and Lee, 2017; Turner, 2016; Singh *et al.*, 2016; Vidovic *et al.*, 2016; Ming *et al.*, 2019].

The approach proposed in this paper is also agnostic (independent of models), since the visualizations of the aspects that involve the explanations mostly apply to several other models (e.g., database statistics, effectiveness metrics, importance of features for the model and instances, confusion matrix, etc.). However, different from the others, it has an advantage related to its ease of use, since it does not require the user to perform source code manipulations. The manipulation and interpretation of the ML models are all performed through the interaction with Explain-ML visualizations and model/executions tuning controls, in a stepwise guided manner.

Still in the scope of the Black Box explanation, **model inspection approaches** are dedicated to providing representations to understand the behavior of ML models. The representations aim to help users in understanding why the model "prefers" a certain outcome (e.g., class) in detriment to others. Some available approaches to Model Inspection are focused on specific models. Others, are model-agnostic [Cortez and Embrechts, 2011; Hooker, 2004; Goldstein *et al.*, 2015; Krause *et al.*, 2016b,a; Guidotti *et al.*, 2018b; Adebayo and Kagal, 2016; Adler *et al.*, 2018; Zhang *et al.*, 2019].

Most of these approaches are evaluated on an available tabular database, with low dimensionality. Our approach supports text databases, with high dimensionality and provides different (complementary) visualizations, besides offering interactive support to build and adjust models based on the understanding of the explanations.

2.2 Overall View of Black Box Explanation Approaches

Several efforts have been devoted to the problem of explaining ML-based decision models [Linardatos *et al.*, 2021]. Table 1 presents a summarized classification of existing approaches mentioned in the literature that provide a Black Box Explanation. The dimensions used to characterize each approach include the three Black Box Explanation scopes of work: outcome explanation, model inspection, and model building, as well as dimensions based on those discussed by Guidotti *et al.* (2018b). We have also included in the last line of the Table 1, the system proposed in this paper – Explain-ML, in order to compare it with the others.

Our goal here is not present an extensive literature review of interpretability approaches or Interactive Machine Learning systems, which has already been done in [Dudley and Kristensson, 2018; Guidotti *et al.*, 2018b; Linardatos *et al.*, 2021] and in [Wondimu *et al.*, 2022; Mosqueira-Rey *et al.*, 2022; Rong *et al.*, 2022; Capel and Brereton, 2023; Cao *et al.*, 2024; Schneider, 2024; Yang *et al.*, 2023]. Our focus here is on contrasting Explain-ML with other existing approaches regarding their scope in addition to the other dimensions listed in the table.

Summarizing, most approaches available in the literature focus on a specific point in the life cycle of a machine learning model: building, debugging or interpreting. Particularly, considering the target user, tools such as [Berthold *et al.*, 2009], which aim at the process workflow, are broad and demand a high specialized knowledge to put resources into practice. The multiperspective characteristic of our proposed approach favors the interpretation of models that deal with text bases, with high dimensionality. Several approaches, such as those based on matrix metaphors or feature distribution [Zhao *et al.*, 2019; Linardatos *et al.*, 2021; Ming *et al.*, 2019; Zhang *et al.*, 2019], for example, have their understandability compromised if applied to datasets with high dimensionality, which is our current focus.

2.3 Approaches for ML model building

In most cases, building ML models is a task assigned to specialized users (e.g., data scientists) who are familiar with Machine Learning. This task usually requires some amount of coding. In this scope of work, the authors of [Han *et al.*, 2016] propose a visual approach to assist users in inspecting and interacting with co-training, a semi-supervised learning method. Their approach, however, is not model-agnostic, focusing on co-training with linear SVM models. The approach was evaluated on an automatic text classification task with participants in an online recruitment system.

In [Demiralp, 2016], an interactive tool - *Clustrophile*, is presented with visualizations to help analysts in running clustering methods. In [Smilkov *et al.*, 2016], the authors discuss an interactive open source visualization tool - *TensorFlow Playground*, specific for neural networks, that allows users to directly manipulate the model with no coding on four datasets. The tool only mentions general characteristics of the dataset related to the distribution of data and not to the nature of the data. No formal user evaluation is performed in either.

The authors of [Wang *et al.*, 2019] propose an interactive visualization tool - *ATMSeer* to assist users in performing automated machine learning methods, including tasks such as selecting machine learning algorithms and tuning hyperparameters. *ATMSeer* was evaluated through a case study with experts and end users. Although the tool presents visualizations to analyze the search models and refine the search space, it was not designed to have the interpretability of the generated models as a main goal, focusing on the selection of the best models.

2.4 Interactive and Human-Centered Machine Learning

ML models have been extensively utilized in different domains, however their use remains more restricted to users familiar with such context or experts. There is a gap in approaches that provide functionality without explicit programming. Recently, interest in ML has grown considerably, and thus, the demand for systems that allow non-expert users to better apply and understand these techniques.

Table 1. Dimensions used to describe existing systems. **Name**: model name given by authors; **Reference**: reference number as listed in the References section;**Year**: publication year (of first paper);**Model Agnostic**: the proposed approach is model agnostic, e.g., works with any ML model; **Dataset**: tab- *Tabular*, txt- *Text*, any - *Any type of data*; **Scope of work**: scope of work addressed by the approach. **OE - *Outcome Explanation Problem***, MI -*Model Inspection Problem*, MB -*Model Building*; **Interactive**: the approach is interactive; **MI - without Programming**: the approach does not requires explicit programming for model interpretation; **MB - without Programming**: the approach does not requires explicit programming for model building and/or control; **Visualizations perspective**: the approach presents visualizations (when it does) that convey aspects related to instance-specific information (L-*Local*), the overall model (G-*Global*) and the training set (D-*Dataset*); **HC Development**: the development approach was accomplished with the explicit participation of users; **HC Evaluation**: a formal evaluation or case study was performed with users on the approach.

Name	**Reference**	**Year**	**Model Agnostic**	**Dataset**	**Scope of Work**	**Interactive**	**MI - Without programming**	**MB - Without programming**	**Visualizations perspective**	**HC Development**	**HC Evaluation**
VIN	[Hooker, 2004]	2004	x	tab	MI				g		
VEC	[Cortez and Embrechts, 2011]	2011	x	tab	MI				l,g		
–	[Erik and Kononenko, 2010]	2010	x	tab	OE				l		
ICE	[Goldstein *et al.*, 2015]	2015	x	tab	MI				l,g		
MFI	[Vidovic *et al.*, 2016]	2016	x	tab	OE				l,g		
MES	[Turner, 2016]	2016	x	any	OE				l		
Lime	[Ribeiro *et al.*, 2016]	2016	x	any	OE				l		x
–	[Singh *et al.*, 2016]	2016	x	tab	OE				–		
Prospector	[Krause *et al.*, 2016b,a]	2016	x	tab	MI	x			l,g		x
OPIA	[Adebayo and Kagal, 2016]	2016	x	tab	MI				–		
–	[Han *et al.*, 2016]	2016		txt	MB	x		x	l		x
Clustrophile	[Demiralp, 2016]	2016		tab	MB	x		x	l,g		
TensorFlow Playground	[Smilkov *et al.*, 2016]	2016		–	MB	x		x	g		
SHAP	[Lundberg and Lee, 2017]	2017	x	any	OE				l		x
Anchors	[Ribeiro *et al.*, 2018]	2018	x	any	OE				l		x
LORE	[Guidotti *et al.*, 2018a]	2018	x	tab	OE				l		
BlackBoxAuditing	[Adler *et al.*, 2018]	2018	x	tab	MI				g		
–	[Krause *et al.*, 2018]	2018		tab	MI	x			l,g		x
ATMSeer	[Wang *et al.*, 2019]	2019		tab	MB	x		x	g	x	x
Manifold	[Zhang *et al.*, 2019]	2019	x	tab	MI	x			g		
RuleMatrix	[Ming *et al.*, 2019]	2019	x	tab	MI,OE	x			l,g		x
IForest	[Zhao *et al.*, 2019]	2019		tab	OE	x			l,d		x
Explainable Matrix	[Neto and Paulovich, 2021]	2021		tab	MI, OE				g,l		x
Explain-ML		**2021**	**x**	**txt**	**MI, OE, MB**	**x**	**x**	**x**	**l,g,d**	**x**	**x**

Interactive Machine Learning (IML) is directly related to human-computer interaction as it puts human interactions in perspective. It was introduced by Fails and Olsen Jr. (2003). It treats the model training process as an HCI task, receiving users input in the example selection, creation and labeling process [Fails and Olsen Jr, 2003]. A user more familiar with ML models may be required for the model's deployment but is not essential in the training process. It differs from classical ML in that it considers user participation through an iterative process of changing and revising the model during its training, and these modifications may be small. In more traditional ML, the user usually performs a wholesale pre-selection of training data and significantly changes every model execution [Dudley and Kristensson, 2018; Fails and Olsen Jr, 2003].

As mentioned before, the *Human-Centered Machine Learning* (HCML) research area has recently emerged based on an identified lack of consideration of the explicit needs of users concerning interpretability and decision-making aids, from a more human-centered perspective [Ramos *et al.*, 2019; Gillies *et al.*, 2016; Fiebrink and Gillies, 2018; Capel and Brereton, 2023]. The approach proposed in this paper is in line with the HCML and IML areas. The approach was developed and evaluated considering human-centered perspective, and offers functionalities that support tasks under the IML paradigm.

3 The Human-Centered Design Process Conducted

In order to propose an ML explainability tool, we conducted an interactive design process that allowed us to take into account users' views and perspectives on which aspects would be relevant in such a tool. In Figure 1, we represent our design process based on the simple lifecycle model for interaction design proposed by [Preece *et al.*, 2023] based on principles of people-centered design by indicating for 2 iterative cycles the steps taken for each design activity. Our first step was to interview target users, and identify factors that were deemed relevant to participants' experiences and design requirements for an ML explainability tool. We then generated a persona that represented our primary user [Nielsen, 2013], and scenarios that described our understanding of how they would use the system [Carroll, 2000]. Next, we developed a low-fidelity prototype to represent our solution based on those requirements and conducted an evaluation with other target users. As a result, we were able to fine-tune our understanding of users requirements and generate: (1) a workflow model for an agnostic approach for an interactive system to support the life-cycle of an ML model; (2) a tool based on this model that would allow us to collect indicators about its usefulness. In this section, we briefly describe the main steps in the developing stage of our user-centered approach.

The target users of our work are *knowledgeable users*, *i.e.* those who are familiar with ML models by having learned and used ML models in the past (*e.g.*, through a course). Our definition is similar to that of the *data expert* category proposed by [Mohseni *et al.*, 2021]. Accordingly, our target user refers to end-users who use AI products in daily tasks and have some expertise with ML systems such as data scientists and domain experts who use Machine Learning for analysis, decision-making, or research.

In this subsection, we briefly describe the steps conducted in our first design cycle in our human-centered design process[2]

3.1 Initial Requirements

In order to elicit our initial requirements for an explicability tool, we conducted semi-structured interviews [Lazar *et al.*, 2017] with target users in order to better understand the process followed by them (e.g., pre-processing, parameter tuning, result analysis) while employing ML models in their applications. The interviews were conducted between July 21 and September 12, 2018, via video conference and included 7 participants. The participants were all male and Computer Science graduate students. All of them had experience with Machine Learning and used ML models on a daily basis.

As a result we identified five main factors that were relevant to participants' experiences with ML models that were used to define our requirements:

1. Integration with used tools: in special it would be desirable for the proposed explainability system to be integrated with ScikitLearn;
2. Including strategies for measuring the reliability of results: among the main evaluation metrics used to assess the performance of the models, the most recurrent ones were accuracy, recall and F1 (micro and macro);
3. Use of reference values for the hyperparameters: which was considered as a complicated step by the respondents, since pre-processing, folding, parameterization and other data treatment procedures depend on the dataset;
4. Parametrization strategies: the new tool should allow folded division and cross validation (including nested) for hyperparameter tuning;
5. Experience with available explicability resources: participants did not have much experience with other explicability systems (2 mentioned having used LIME). Being shown an example generated in LIME, they thought that LIME could help them understand the model as a black box, providing insight regarding the confidence of the classification, as well as decide whether the model application is reliable. They also pointed out aspects they felt could be improved - e.g. they did not think that the explanations were intuitive enough, nor that all relevant metrics were presented.

Regarding the planned functionalities, proposed for explaining and interacting with the ML models, the participants considered as the most relevant (in no particular order): (i) comparative display of several evaluation metrics; (ii) support for folded division and cross-validation parameterization; (iii) overview of best parameters found for a dataset; (iv) visual comparison between runs; and (v) display of the importance of the variables, both globally (whole model) and locally (instance-based).

With the compiled results, it became clear that an ML tool focused on interpretability would have the potential to be largely used by researchers and ML practitioners. Summarizing , the following requirements were extracted from the interviews for developing our tool:

- **R1 – ScikitLearn integration**: As all participants use ScikitLearn the ideal would be that our tool had the same functionalities and steps in ML modeling or used it as basis.

[2] A more detailed description can be found in [Lopes *et al.*, 2021] and [Lopes *et al.*, 2022].

Figure 1. Explain-ML interactive design process depicting two iterative cycles – based on the simple interaction design lifecycle model [Preece *et al.*, 2023]

- **R2 – Support for cross validation**: All participants used cross validation to train the model, so it would be important to support this technique, as well as let users decide on the number of folds.
- **R3 – Support for hyperparameter tuning**: Integrate hyperparameter tuning as part of the process supported by the tool and provide an explanation of each available hyperparameter.
- **R4 – Display of a set of evaluation metrics (including confusion matrix)**: Need of different metrics to be able to perform a more accurate analysis of the execution.
- **R5 – Model-based feature importance**: Its important to exhibit the feature importance of the model as it helps the user understand the model's predictions.
- **R6 – Instance-based feature impact**: It is important to show the instance-based feature impact as it supports user's understanding of the model's prediction per instance.
- **R7 – Visualization of the execution history**: regarding the control of routine tasks in ML, participants mentioned the difficulty in assessing model reliability due to the challenge to compare different results. Thus, it pointed to the need to include a mechanism for the visualization of the history of experiments and monitoring of the evolution of the models based on performed changes (e.g., parameterization, feature selection, etc).

In the end, all participants mentioned that they would use a tool that had such functionalities in their daily activities, confirming a real need for this kind of interpretability framework.

3.2 Persona, scenario and prototype

Once the initial requirements were defined, we created a persona to represent our target users, as well as some use scenarios Carroll [2000]. as can be seen in Figure 2.

Based on the requirements, we developed a low-fidelity prototype to present our proposed visualizations to the target users and perform a formative evaluation of the tool [Preece *et al.*, 2023]. The prototype was developed using Balsamiq Wireframing Tool[3] and presented the structure and content of the interface, allowing users to interact with it. Users could navigate through the interfaces and visualizations, which simulated an execution – the functionality was not yet implemented, but the prototype showed the data for one example as if it were being executed for that one model. Figure 3 illustrates some of the screens created in this prototype.

3.3 User Evaluation of Low-Fidelity Prototype

In order to collect feedback on our proposed tool represented through the Balsamiq Prototype, we conducted a formative evaluation with 3 participants (2 males and 1 female). The participants were all Computer Science graduate students and expert users who use ML Models on a daily basis as part of their research. The evaluation was conducted between December 5 and 26, 2018, and two of them were face-to-face, whereas the other was via videoconference.

In the evaluation session, participants interacted with the prototype of Explain-ML tool as if they were creating a model. To do so, they were asked to (i) Create a new project; (ii) Access the project created; (iii) Create a new execution to the project created; (iv) Analyze the results of the new execution; and (v) Analyze the project results history. Participants were interviewed about their interactive experience and thoughts about the prototype.

The users' interactions and audio in their sessions were recorded, and transcribed, and an analysis conducted. Overall, participants liked the solution conveyed through the prototype and said they would use such a tool in their ML modeling, which was in line with the results of the initial interviews.

As a result of the evaluation, participants raised a few problems, improvements and suggestions related to the explainability model, as well as the interface.

3.4 Second Design Cycle

The results from the users' evaluation of the low-fidelity prototype were used to fine-tune and validate our initial requirements. Furthermore, they allowed us to review some of the initial ideas on how to implement these requirements and improve them in order to implement the actual tool.

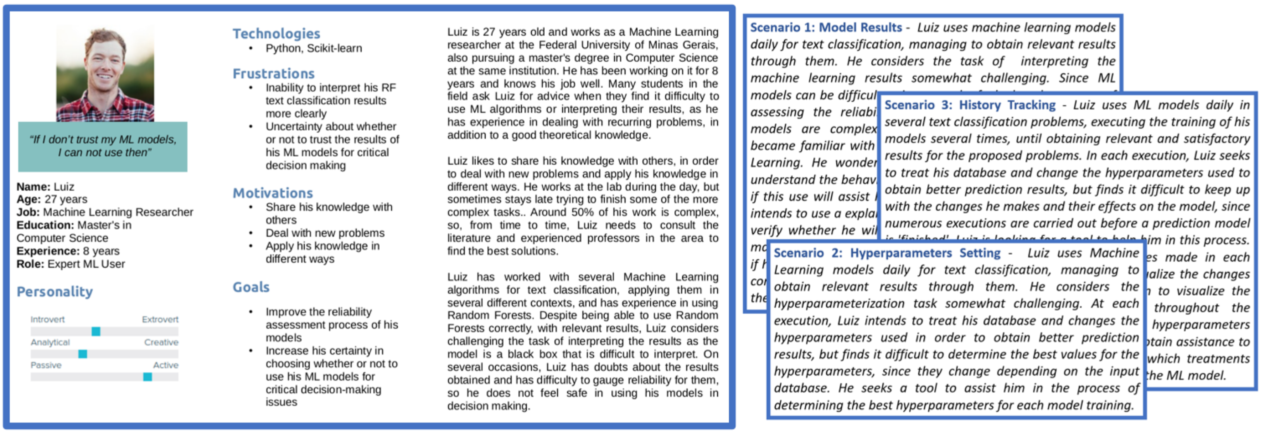
In the second design cycle, we developed a multi-perspective approach to describe the workflow involved in the creation of a project and its execution in an interpretable system that would fulfill our requirements. Next, we developed the Explain-ML tool and performed user evaluation in order to collect information about their perspective and experience with the tool. In the following sections we present the results of each of these steps.

4 The Proposed Multi-perspective Approach

In this section we present the approach proposed in this work based on the requirements generated in our first design cycle. The proposed approach is agnostic regarding the ML model, as the workflow and explanations implemented would apply to different models. Following is a broad overview of the proposed approach.

The approach is organized to support a workflow in which the user can interactively perform the stages of the life cycle of an ML model: definition and optimization of hyperparameters, model training as well as testing, evaluation, adjustments for refining the model and model re-execution. In particular, the interpretability of models provides the core support to the model evaluation activity.

[3] `https://balsamiq.com/`



Technologies
- Python, Scikit-learn

Frustrations
- Inability to interpret his RF text classification results more clearly
- Uncertainty about whether or not to trust the results of his ML models for critical decision making

Motivations
- Share his knowledge with others
- Deal with new problems
- Apply his knowledge in different ways

Goals
- Improve the reliability assessment process of his models
- Increase his certainty in choosing whether or not to use his ML models for critical decision-making issues

"If I don't trust my ML models, I can not use then"

Name: Luiz
Age: 27 years
Job: Machine Learning Researcher
Education: Master's in Computer Science
Experience: 8 years
Role: Expert ML User

Personality

Luiz is 27 years old and works as a Machine Learning researcher at the Federal University of Minas Gerais, also pursuing a master's degree in Computer Science at the same institution. He has been working on it for 8 years and knows his job well. Many students in the field ask Luiz for advice when they find it difficulty to use ML algorithms or interpreting their results, as he has experience in dealing with recurring problems, in addition to a good theoretical knowledge.

Luiz likes to share his knowledge with others, in order to deal with new problems and apply his knowledge in different ways. He works at the lab during the day, but sometimes stays late trying to finish some of the more complex tasks.. Around 50% of his work is complex, so, from time to time, Luiz needs to consult the literature and experienced professors in the area to find the best solutions.

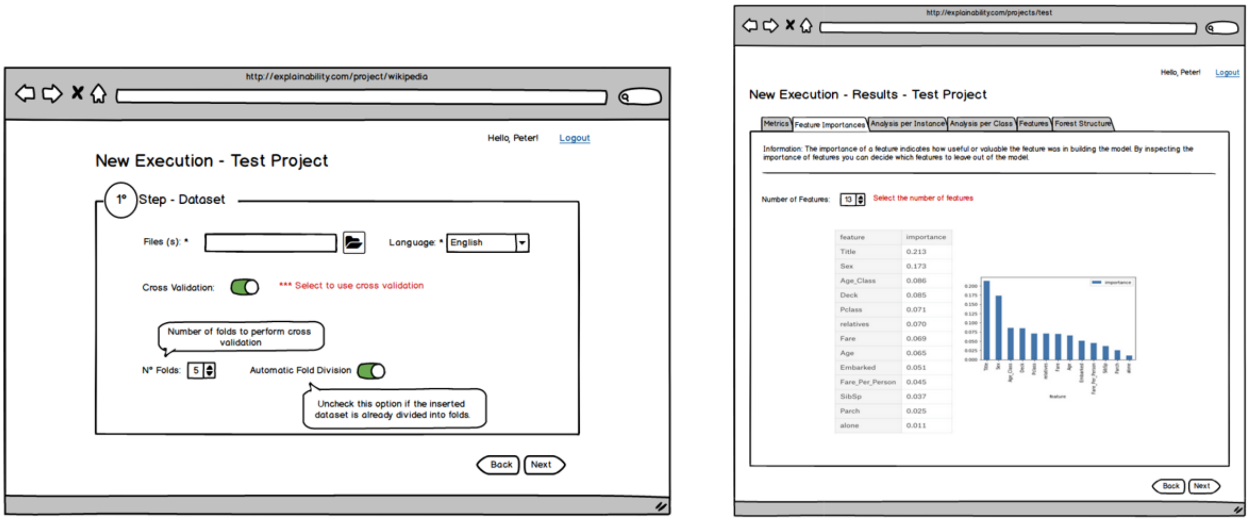
Luiz has worked with several Machine Learning algorithms for text classification, applying them in several different contexts, and has experience in using Random Forests. Despite being able to use Random Forests correctly, with relevant results, Luiz considers challenging the task of interpreting the results as the model is a black box that is difficult to interpret. On several occasions, Luiz has doubts about the results obtained and has difficulty to gauge reliability for them, so he does not feel safe in using his models in decision making.

Scenario 1: Model Results - *Luiz uses machine learning models daily for text classification, managing to obtain relevant results through them. He considers the task of interpreting the machine learning results somewhat challenging. Since ML models can be difficul*
assessing the reliabi
models are complex
became familiar with
Learning. He wonder
understand the behav
if this use will assist
intends to use a explai
verify whether he wil
mo
if h
co
the

Scenario 3: History Tracking - *Luiz uses ML models daily in several text classification problems, executing the training of his models several times, until obtaining relevant and satisfactory results for the proposed problems. In each execution, Luiz seeks to treat his database and change the hyperparameters used to obtain better prediction results, but finds it difficult to keep up with the changes he makes and their effects on the model, since numerous executions are carried out before a prediction model*
him in this process.
es made in each
alize the changes
to visualize the
throughout the
hyperparameters
tain assistance to
which treatments
the ML model.

Scenario 2: Hyperparameters Setting - *Luiz uses Machine Learning models daily for text classification, managing to obtain relevant results through them. He considers the hyperparameterization task somewhat challenging. At each execution, Luiz intends to treat his database and changes the hyperparameters used in order to obtain better prediction results, but finds it difficult to determine the best values for the hyperparameters, since they change depending on the input database. He seeks a tool to assist him in the process of determining the best hyperparameters for each model training.*

Figure 2. Persona and scenarios created during the first design cycle.



http://explainability.com/project/wikipedia

Hello, Peter! Logout

New Execution - Test Project

1º Step - Dataset

Files (s): *

Language: * English

Cross Validation:

*** Select to use cross validation

Number of folds to perform cross validation

Nº Folds: 5

Automatic Fold Division

Uncheck this option if the inserted dataset is already divided into folds.

http://explainability.com/projects/test

Hello, Peter! Logout

New Execution - Results - Test Project

Information: The importance of a feature indicates how useful or valuable the feature was in building the model. By inspecting the importance of features you can decide which features to leave out of the model.

Number of Features: 13 Select the number of features

feature	importance
Title	0.213
Sex	0.173
Age_Class	0.086
Deck	0.085
Pclass	0.071
relatives	0.070
Fare	0.069
Age	0.065
Embarked	0.051
Fare_Per_Person	0.045
SibSp	0.037
Parch	0.025
alone	0.011

Figure 3. Some screens of the low-fidelity prototype generated.

In this approach, users have their own access area, which is organized according to two important concepts within the tool:

- *Project*: consists of a set of one or more model executions, performed by a user, with the projects' execution history being recorded.
- *Execution*: covers the process of defining and optimizing hyperparameters, model training, and testing. The information regarding each execution is recorded by the tool and the corresponding visualizations are made available.

Figure 4 presents the workflow to create a project and ***one*** execution. To perform an execution, the user first creates it and sets values for the model's hyperparameters (or selects the hyperparameters to be tuned with cross-validation along with the respective parameters' options/ranges). After the model is trained, the user applies it to a test set and accesses the interactive visualizations regarding the current execution. If desired, the user can also review the history of previous project executions. *In order to insert a new execution, users should repeat the activities related to: Execution creation, Execution analysis, Project analysis and Model adjustments* (Figure 4: *ii, iii, iv, v*).

5 Explain-ML Overview

In this section, we present in detail the Explain-ML tool that implements the approach proposed in this work. Explain-ML is a web based application, developed using the Python Web framework *Django*[4], using *Sqlite*[5]. The screenshots and visualizations presented in this section are related to a version of *Explain-ML*, instantiated with Random Forests (as ML model) applied to automatic text classification (as ML task). The instantiated version employs the *Scikit-Learn*[6] implementation of the Random Forests Model. Note that, as pointed out, the proposed approach is agnostic regarding the ML model, as the workflow and explanations implemented would apply to other models, which could therefore be incorporated into the tool. Finally, the example shown depicts the model of the the *WebKB-Course dataset*[7]. Following is

[4] `https://www.djangoproject.com/`

[5] `https://www.sqlite.org/index.html`

[6] `https://scikit-learn.org/stable/`

[7] `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/`

Figure 4. Multi-perspective approach workflow for the creation of a project and one execution.

a broad overview of the tool.

The initial interaction with Explain-ML is accomplished by means of the main screen, where users can manage their projects (Figure 5 - A). A new execution for an existing project can be created through the main screen (Figure 5 - A(3)) or by visualizing the project details (Figure5 - A(1)). In the latter, the user is directed to the project information where the list of executions can be visualized (Figure 5 - B), along with other project information.

5.1 Model Execution

Creating an execution is comprised of three steps (Figure 6):

- *Step1*: The user fills in the data and selects the file containing the data-set, data format, number of folds, and whether to perform cross validation.
- *Step2*: Subsequent to step 1, the user configures the execution by selecting whether to tune the hyperparameters and, if so, selects which hyperparameters to perform, along with the range/options.
- *Step3*: The tool displays the hyperparameters (obtained through tuning or user-defined) and allows the user to change or not these values. From there, the user can proceed to model execution. The results are stored and feed the execution visualizations.

In step 2, the hyperparameters are respective to the Random Forest model, the model being used in this instantiation of the approach. If the approach is used with another ML model (e.g. a neural network), the hyperparameters will be different and related to the corresponding model. After every model execution in step 3, the tool stores the execution data, which feeds the respective execution visualizations and execution history. Next we present an overview of Explain-ML[8], describing its visualizations and history.

5.2 Interactive Visualizations

For each execution, the approach presents a set of visualizations that convey aspects related to the overall model, the training set, and instance-specific information. Explain-ML presents five consecutive tabs, which are complementary to each other, and with which the user can interact: *Evaluation Metrics, Class Distributions, Feature Importance , Instance Analysis* and *Analysis by class.*

Next, we present each one of the visualizations. For illustrative purposes, the figures concerns *Execution 3* (Estimators = *200*. Criterion = *gini* and other Scikit Learn default parameters) presented in the Execution List of Figure 5 - B.

5.2.1 Visualization: Evaluation Metrics

This visualization presents aspects of the model, presenting well-known ML effectiveness metrics and its value for the execution (see Figure 7). For each metric contextualized help (depicted by a question mark) is available by demand to users. For instance, the contextualized help associated with Accuracy will present an explanation (*"It represents, in general, how often the classifier is correct. Range [0..1]"*); and for F1-Score (*"F1 is the harmonic mean between accuracy and recall. It shows how accurate your classifier is (how many instances are correctly classified) and how robust it is (it does not lose a significant number of instances). Range [0..1]."*).

This visualization also depicts the comparison of the value of each metric regarding the previous execution. This helps the user quickly perceive whether the new execution has improved or not model performance for each of the metrics. For example, in Figure 7, we observe that metric Accuracy had value of *0.788* in Execution 3 and value *0.777* in the previous one (Execution 2), potentially indicating a (small) improvement in that metric[9].

5.2.2 Visualization: Class Distribution

This visualization presents aspects of the dataset used in the model training (Figure 8). The graph shows the distribution of dataset instances by class. When interacting with each bar, the user sees the name of the respective class, the number of documents belonging to the class and some class specific metrics (F1, Precision, Recall). The user can confront these values with the general metrics of the model (Figure 7) and understand how it behaves globally and specifically, for each class.

The bar graph allows users to easily grasp whether there is any dominant class, the major and minor classes, and other characteristics that may insert bias into the model, or even make it more difficult (or easier) to classify instances of a given class. Those observations, when aggregated with other tool views, assist the user in the overall model interpretability.

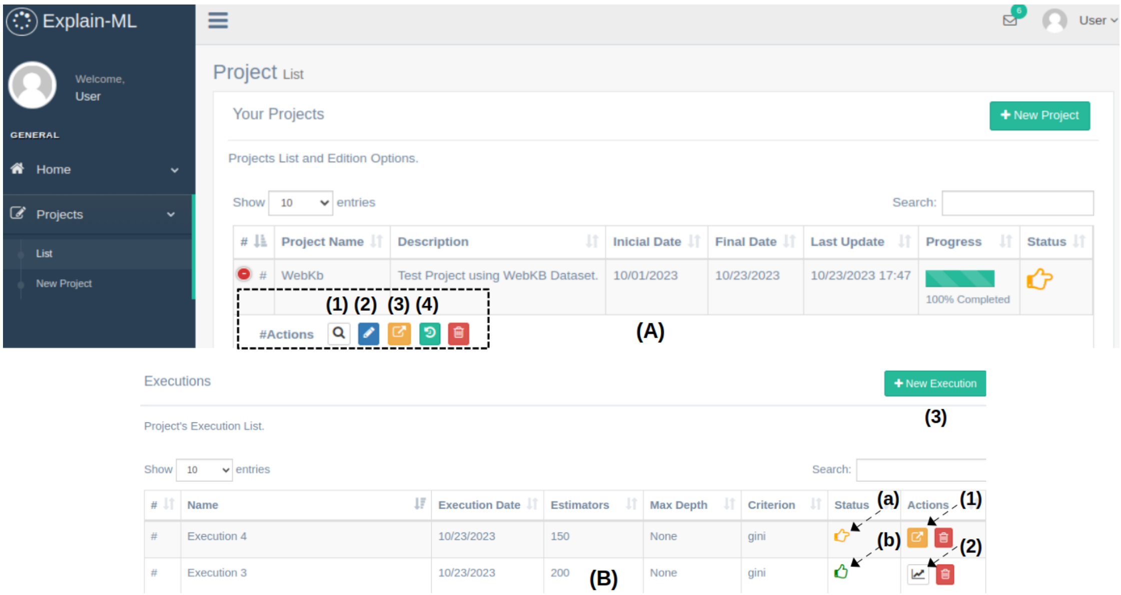
5.2.3 Visualization: Feature Importance

This visualization presents overall aspects of the model. Several types of complementary information are presented regarding the importance of features[10] in the model (Figure 9). At the top of the tab (Figure 9 - A), we see some reference information: (A.1) total number of features, (A.2) and (A.3) lowest and the highest value of importance, (A.4) number of features with importance value zero, (A.5) number of

[8] Explain-ML is not yet available for public use, but for a brief description and video of its interaction are available in: https://github.com/BarbaraGCOL/explain-ml?tab=readme-ov-file

[9] In these results, and in the current version of Explain-ML in general, we are not considering statistical significance tests for accounting for statistical ties, but this function is in the planning.

[10] In the current instantiation of Explain-ML, the value of feature importance considered is given by the decrease in decision tree node impurity weighted by the probability of reaching that node, but any other measure such as information gain or chi-square could have been used.



Project List

Your Projects

+ New Project

Projects List and Edition Options.

Show 10 entries Search:

#	Project Name	Description	Inicial Date	Final Date	Last Update	Progress	Status
#	WebKb	Test Project using WebKB Dataset.	10/01/2023	10/23/2023	10/23/2023 17:47	100% Completed	👉

Executions

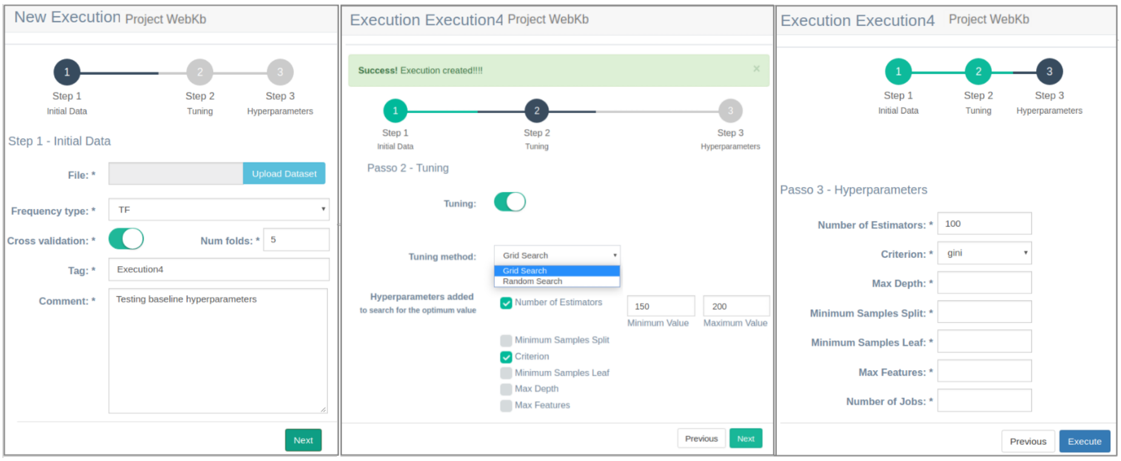
+ New Execution

Project's Execution List.

Show 10 entries Search:

#	Name	Execution Date	Estimators	Max Depth	Criterion	Status	Actions
#	Execution 4	10/23/2023	150	None	gini	👉	
#	Execution 3	10/23/2023	200	None	gini	👍	

Figure 5. (A) A screenshot of Explain-ML: Users can list previous projects or create a new one in the *left side*. In the *right side*, the projects are listed along with their information such as description, period, status and progress, and the available actions the users can take regarding the model: (A.1) visualize the project details, (A.2) edit project, (A.3) create a new execution, and (A.4) visualize history of project executions. A new execution can only be created on *In Progress* projects.
(B) Execution list of a project, with information such as name, execution date, hyperparameter values, and execution status. (B.1) *Continue Execution* - for executions where hyperparameter tuning has been performed and the model has not yet been applied (*Status: Step 2 of execution completed* (B.a)). (B.2) *See explanation* - for executions where hyperparameter tuning has been performed and the model has been executed (*Status: Execution Completed* (B.b)). (B.3) Create a new execution.



New Execution Project WebKb

Step 1 - Initial Data

File: *

Frequency type: * TF

Cross validation: * Num folds: * 5

Tag: * Execution4

Comment: * Testing baseline hyperparameters

Execution Execution4 Project WebKb

Success! Execution created!!!!

Passo 2 - Tuning

Tuning:

Tuning method: Grid Search (Grid Search, Random Search)

Hyperparameters added to search for the optimum value

☑ Number of Estimators 150 Minimum Value 200 Maximum Value

☐ Minimum Samples Split

☑ Criterion

☐ Minimum Samples Leaf

☐ Max Depth

☐ Max Features

Execution Execution4 Project WebKb

Passo 3 - Hyperparameters

Number of Estimators: * 100

Criterion: * gini

Max Depth: *

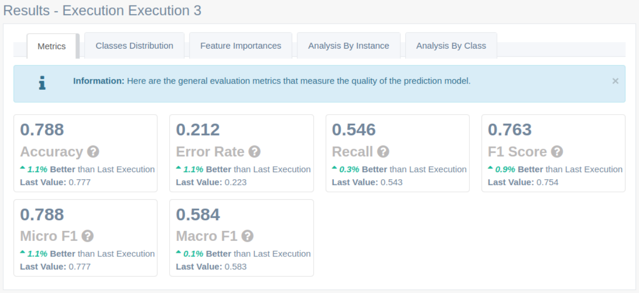
Minimum Samples Split: *

Minimum Samples Leaf: *

Max Features: *

Number of Jobs: *

Figure 6. Creation of a Model Execution - Step 1 , 2 and 3



Results - Execution Execution 3

Information: Here are the general evaluation metrics that measure the quality of the prediction model.

0.788 Accuracy — 1.1% Better than Last Execution — Last Value: 0.777

0.212 Error Rate — 1.1% Better than Last Execution — Last Value: 0.223

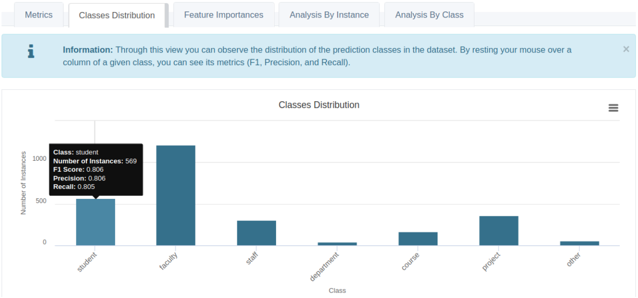
0.546 Recall — 0.3% Better than Last Execution — Last Value: 0.543

0.763 F1 Score — 0.9% Better than Last Execution — Last Value: 0.754

0.788 Micro F1 — 1.1% Better than Last Execution — Last Value: 0.777

0.584 Macro F1 — 0.1% Better than Last Execution — Last Value: 0.583

Figure 7. Visualization: Evaluation Metrics



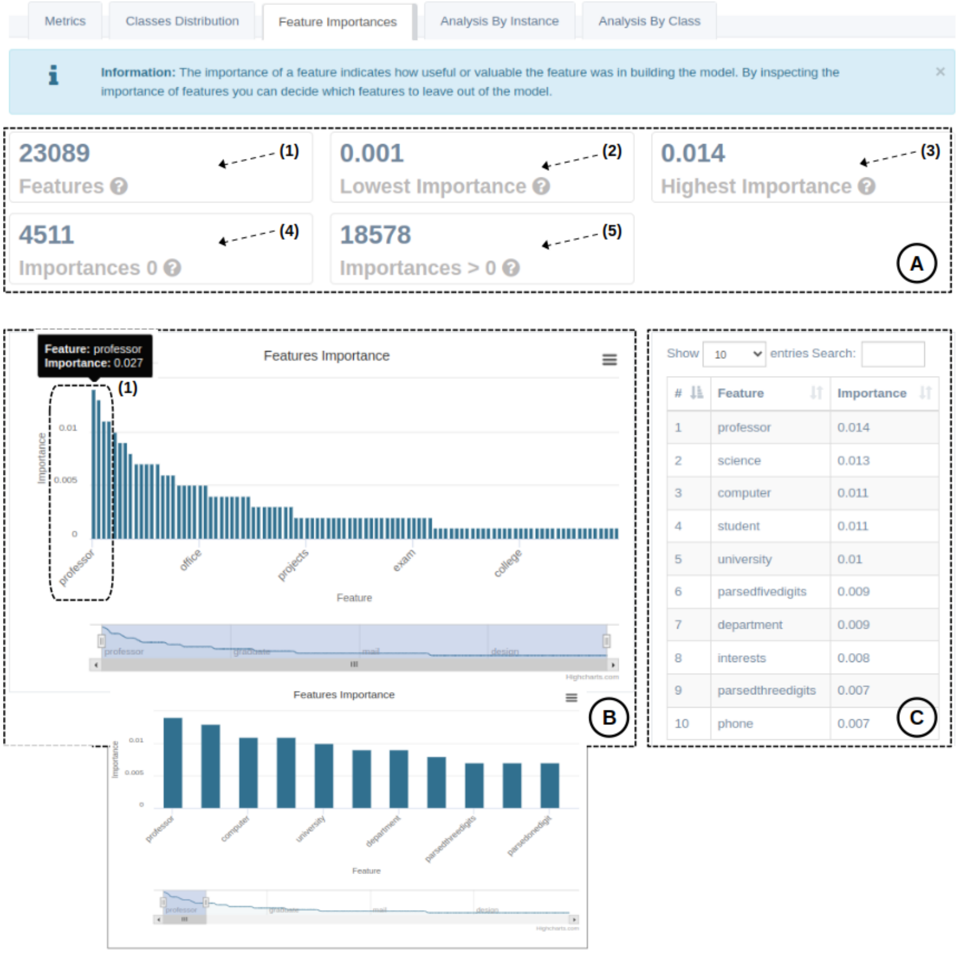
Information: Through this view you can observe the distribution of the prediction classes in the dataset. By resting your mouse over a column of a given class, you can see its metrics (F1, Precision, and Recall).

Class: student
Number of Instances: 569
F1 Score: 0.806
Precision: 0.806
Recall: 0.805

Figure 8. Visualization: Class Distribution

features with importance value greater than zero. These values are important references for understanding the other elements available on the tab.

At the bottom left (Figure 9 - B), a bar chart is shown with the most important features. Users can interact by visualizing more or less features in the graph (B.2) and accessing the value and importance of each feature in the respective bar in the graph (B.1). In addition, the user can perform a feature



Information: The importance of a feature indicates how useful or valuable the feature was in building the model. By inspecting the importance of features you can decide which features to leave out of the model.

23089 Features

0.001 Lowest Importance

0.014 Highest Importance

4511 Importances 0

18578 Importances > 0

Show 10 entries Search:

#	Feature	Importance
1	professor	0.014
2	science	0.013
3	computer	0.011
4	student	0.011
5	university	0.01
6	parsedfivedigits	0.009
7	department	0.009
8	interests	0.008
9	parsedthreedigits	0.007
10	phone	0.007

Figure 9. Visualization: Feature Importance

search in the list of features shown at the bottom right (Figure 9 - C).

This visualization assists in observing the role of features in the model. The user can identify features that are more/less relevant to the model, and how they compare to other features. It is possible to observe the amount of features with some relevance (non-zero importance) or that do not affect the model (zero importance). The tool also allows identifying features whose meaning is not aligned with the predicted class, but have non-zero importance. Or the opposite, meaning compatible with the predicted class, but with zero importance. Especially in text bases, with high dimensionality, these observations can be very significant, allowing users to focus on more relevant features that aid in the interpretation of the model. All of these observations may produce interesting insights for feedback in the models' re-executions.

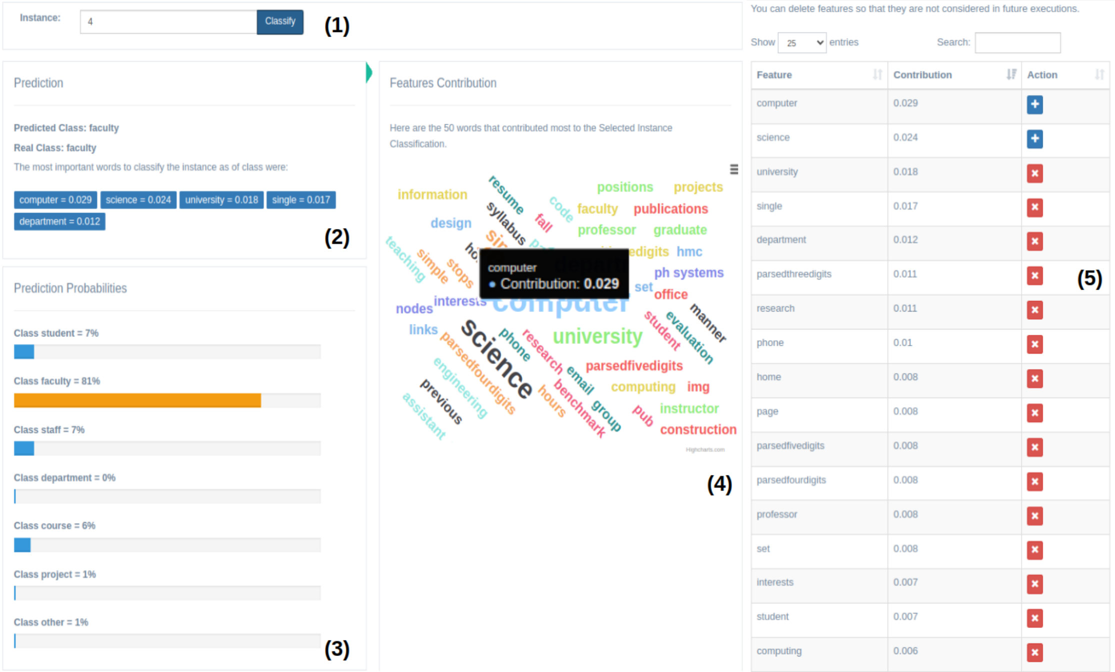
5.2.4 Visualization: Instance Analysis

This visualization presents aspects related to instance-specific information. It aims to assist the user in understanding the instance classification. Figure 10 - (1) shows the visualization for *Instance 4*. The tool presents the *Real* and *Predicted Class* for the instance, along with its most important features and their respective importance values (Figure 10 - (2)). This allows users to check if the model hit the classification and if the most important features have meaning compatible with the context of the predicted class. Otherwise, users may have insights into features that can be excluded for model re-execution.

Besides, the probability for each class in the classification is shown (Figure 10 - (3)). Thus, it is possible to analyze whether the *predicted class* obtained a high probability in relation to the others. It may occur, for example, that two classes have very close probabilities. In this case, the user can use the complementary views (Figure 10 - (4) and (5)) to obtain insights about features that, if removed, may change the classification. This also provides feedback for model re-execution.

The *Word Cloud* allows users to have a general view about contribution of the main features to the result. That feature importance is relative to the instance[11], unlike the overall feature importance, presented on Feature Importances tab. When the user interacts with the tool, the value of the feature's importance is shown. For example, the value of the feature *computer (importance: 0.029)* is shown in the Word Cloud (Figure (10 - (4)) and can also be viewed in the Feature Table (Figure 10 - (5)). The table allows user to sort features by name or importance value, and provides a feature search. The table also has the *Action* column, whereby each feature can be excluded (*red button with x*) / included (*blue button*

[11] Contributions are generated through the *TreeInterpreter* package (`https://github.com/andosa/treeinterpreter`).



Instance: 4 Classify

Prediction

Predicted Class: faculty

Real Class: faculty

The most important words to classify the instance as of class were:

computer = 0.029 science = 0.024 university = 0.018 single = 0.017 department = 0.012

Prediction Probabilities

Class student = 7%

Class faculty = 81%

Class staff = 7%

Class department = 0%

Class course = 6%

Class project = 1%

Class other = 1%

Features Contribution

Here are the 50 words that contributed most to the Selected Instance Classification.

computer
• Contribution: 0.029

You can delete features so that they are not considered in future executions.

Show 25 entries Search:

Feature	Contribution	Action
computer	0.029	+
science	0.024	+
university	0.018	✖
single	0.017	✖
department	0.012	✖
parsedthreedigits	0.011	✖
research	0.011	✖
phone	0.01	✖
home	0.008	✖
page	0.008	✖
parsedfivedigits	0.008	✖
parsedfourdigits	0.008	✖
professor	0.008	✖
set	0.008	✖
interests	0.007	✖
student	0.007	✖
computing	0.006	✖

Figure 10. Visualization: Instance analysis. (1) *Instance*: *Instance 4* selected; (2) *Prediction*: presents the real and predicted class of *Instance 4*, and the most important features; (3) *Prediction Probabilities*: presents the probability for each class in the classification; (4)*Features Contribution*: presents a visual representation (word cloud) for the contribution of the main features of *Instance 4*. (5) Features Table.

Table 2. Information of instances from Execution 3 (Figure 5 - B), with all dataset features. *(Due to space constraints, only the main three 3 features are shown)*. The predicted class is presented along with the probability of its classification.

Instance	Real Class	Predicted Class	Main Features
0	*faculty*	*faculty* (66%)	***computer***, *washington*, ***science***
1	*student*	*student* (46%)	*personal, bookmarks*, **science**
4	*faculty*	*faculty* (81%)	***computer, science***, *university*
5	*staff*	*staff* (47%)	*hours, handouts, final*
7	*student*	*student* (55%)	*interests, graduate, parsedfivedigits*
9	*student* (17%)	*faculty* (42%)	***science***, *single, parsedonedigit*

Table 3. Similar to Table 2, this table is relative to the execution after Execution 3, without features *computer* and *science*.

Instance	Real Class	Predicted Class	Main Features
0	*faculty*	*faculty (68%)*	*university, department, washington*
1	*student (39%)*	*faculty (53%)*	*university, washington, parsedthreedigits*
4	*faculty*	*faculty* (80%)	*university, department, research*
5	*staff*	*staff* (47%)	*handouts, grades, exam*
7	*student*	*student* (48%)	*candidate, graduate, parsedfivedigits*
9	*student* (18%)	*faculty* (40%)	*form, single, parsedthreedigits*

with plus sign) the next model executions. Here, the user can perform several simulations and re-executions of the model, in order to understand its behavior.

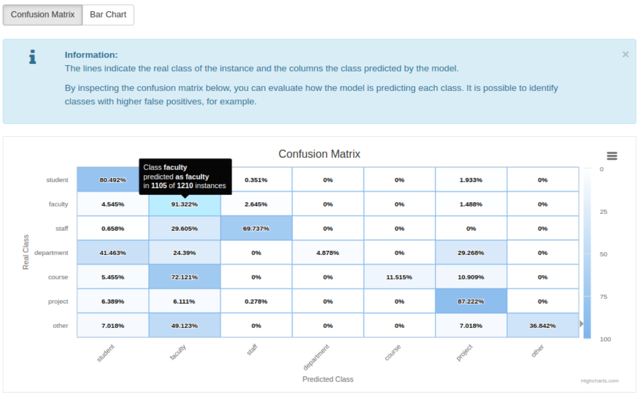
Relying on this *Instance Visualization*, Table 2 contains some information on a few illustrative instances selected *ad-hoc*. The information assists the user in model interpretability, fostering some insights, for instance:

- *Instances 0, 4, 5* e *7* are correctly classified and the main features (those with the highest importance values) are related with the predicted class under the point of view of common sense.
- *Instance 9* is misclassified and the main features have no real meaning regarding the predicted class.
- Features *computer* and *science* are among the main features of *Instances 0* e *4*, whose classes are *faculty*. Note that class *faculty* is the major class of the dataset (Figure 8) and features *computer* and *science* are in the main features of the overall model (Figure 9).

All this complementary information could lead the user to question: *is it possible to specify more relevant features for each class?* New model executions could be performed to better evaluate this. For instance, through the *Feature Table* (Figure 10 - (5)) the user could exclude features *computer* and *science* and rerun model to analyze the effect of this feedback. These features would not be considered for the next model training. This adjustment would cause a slight worsening in the model metrics. In addition, the user could observe changes in the instance analysis (Table 3). Comparing Tables 2 and 3, the user would notice that *Instance 0* remained in the same class with lower probability, and *Instances 4 and 7* remained in the same classes with lower probabilities. The meaning of their main features is more related to the predicted class, which may indicate actual relevance of excluded features. *Instance 5* were slightly affected. The model continues to misclassify *Instance 9*, but the probability for the real class has increased, and started to misclassify *Instance 1*. This example, illustrates some examples of interactions, insights and feedbacks that can be performed on the model, many others are possible. The tool, for instance, allows users to undelete features for subsequent executions.

5.2.5 Visualization: Analysis by Class

This visualization presents aspects related to the model's predictions for instances and their ground truth. Here the focus is to understand how the model behaves for each class and whether it is more successful in predicting certain classes than others. Figure 11 illustrates what is presented to the user in this view. It is based on a confusion matrix that provides an overview of how the model behaves with regards to each class.



Information:
The lines indicate the real class of the instance and the columns the class predicted by the model.

By inspecting the confusion matrix below, you can evaluate how the model is predicting each class. It is possible to identify classes with higher false positives, for example.

Confusion Matrix

	student	faculty	staff	department	course	project	other
student	80.492%	[illegible]	0.351%	0%	0%	1.933%	0%
faculty	4.545%	91.322%	2.645%	0%	0%	1.488%	0%
staff	0.658%	29.605%	69.737%	0%	0%	0%	0%
department	41.463%	24.39%	0%	4.878%	0%	29.268%	0%
course	5.455%	72.121%	0%	0%	11.515%	10.909%	0%
project	6.389%	6.111%	0.278%	0%	0%	87.222%	0%
other	7.018%	49.123%	0%	0%	0%	7.018%	36.842%

Figure 11. Visualization: Analysis by class - *Confusion Matrix*

The matrix is organized in terms of real classes (rows) versus predicted classes (columns). Each element represents the percentage of instances of the real class (row-related) that was predicted as the column-related class. When interacting with the matrix elements, users visualize the absolute number of instances and a brief description, for example, for the first element of the matrix: *"Class Faculty predicted as Faculty in 1105 out of 1210 documents"*.

A quick look at the matrix can provide several insights:

- classes in which the model performs best (based on the diagonal elements);
- for a given real class (line), when the model misclassifies instances, which other class is predicted more frequently as being the real class (based on the elements with the highest values on the line, except from the one in the main diagonal);
- for a given real class (line), when a model misclassifies instances, which class it predicts less frequently as being the real class (based on the elements with the lowest values on the line, except from the one in the main diagonal).

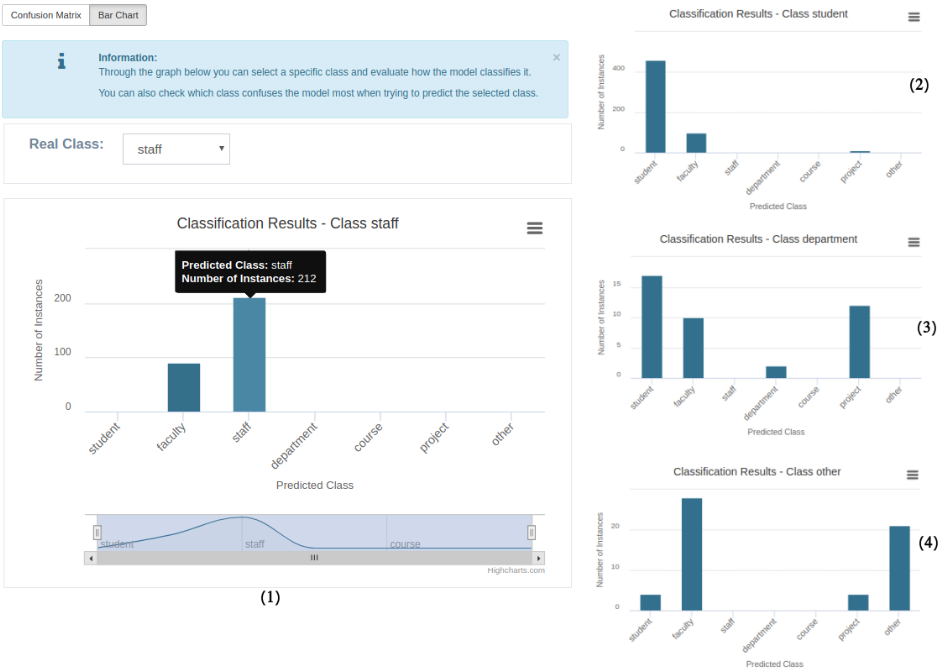
In addition, this visualization also provides a bar graph that represents the behavior of the model for a given real class (selected by the user), as shown in Figure 12. The graph allows the user to inspect each class separately. User interacts selecting the class she wants to inspect. In the graph, it is possible to verify the absolute number of instances related to each bar.

As mentioned, the Explain-ML provides complementary visualizations under several perspectives. For example, analyzing the model information depicted in Figure 12, we can see it behaves well for the *Faculty* and *Student* classes. If we compare the information with that available in the *Class distribution* tab (Figure 8), we notice that these classes are the ones with the largest number of instances. Thus, we can infer that a higher number of instances in the dataset ultimately contributes to the model learning better to predict instances of these classes. If we look, for example, at the *Department* class, we would observe the opposite: the model does not perform well on instances of this class, however, the dataset only contains a few instances of this class. This helps to explain the lower values of the evaluation metric MacroF1 (average of the performance in each individual class) when compared to MicroF1 (global performance regardless of the classes). This would also give hints to users about how to improve the model (for instance, by means of under or oversampling techniques to improve the balancing [Chawla *et al.*, 2004; Batista *et al.*, 2004]).

5.3 Project Execution History

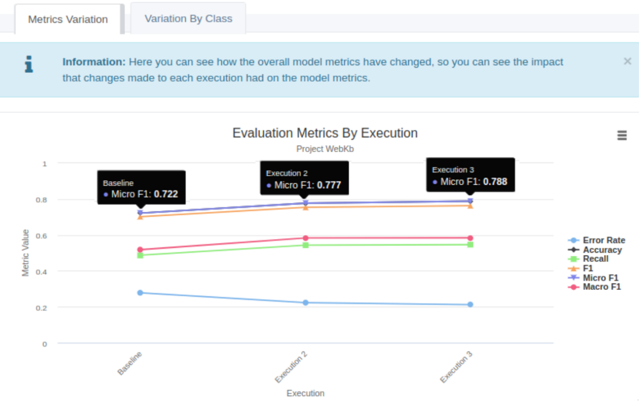
As described at the beginning of Section 5, Explain-ML is organized according to the concepts of *project* and *execution*. A project can have multiple model executions, storing the executions history and presenting a comparative analytical graph between them. The Project execution history provides two types of charts. The first one, *Metrics Variation* (Figure 13), represents the variation in the metrics along the executions. The chart includes some well-known performance metrics such as *Error Rate, Accuracy, Recall, F1, Micro F1,* and *Macro F1*. In the graph, each metric is represented by a color/dot type. The user can interact with the graph and verify the value of a given metric in a given execution. Is is possible to observe the improvements or worsening of the metrics throughout the executions.

In addition, the other chart - *Variation by class* (Figure 14), provides a bar graph that represents the behavior of the model for a given real class (selected by the user). Each bar is an instance of what was shown in the *Analysis by class* tab (Figure 12), but in this case, the bars for all the executions are shown side by side. When interacting with each bar, the user sees the name of the respective real class, predicted class and the number of documents belonging to the real class. The graph summarizes the evolution of the model behavior regarding each class.



Information:
Through the graph below you can select a specific class and evaluate how the model classifies it.
You can also check which class confuses the model most when trying to predict the selected class.

Figure 12. Visualization: Analysis by class - *Bar Chart* . (1) graph presenting the classification results for selected class - *staff*; (2)(3)(4) graphs resulting from the selection of classes *student, department* and *other*, respectively.



Information: Here you can see how the overall model metrics have changed, so you can see the impact that changes made to each execution had on the model metrics.

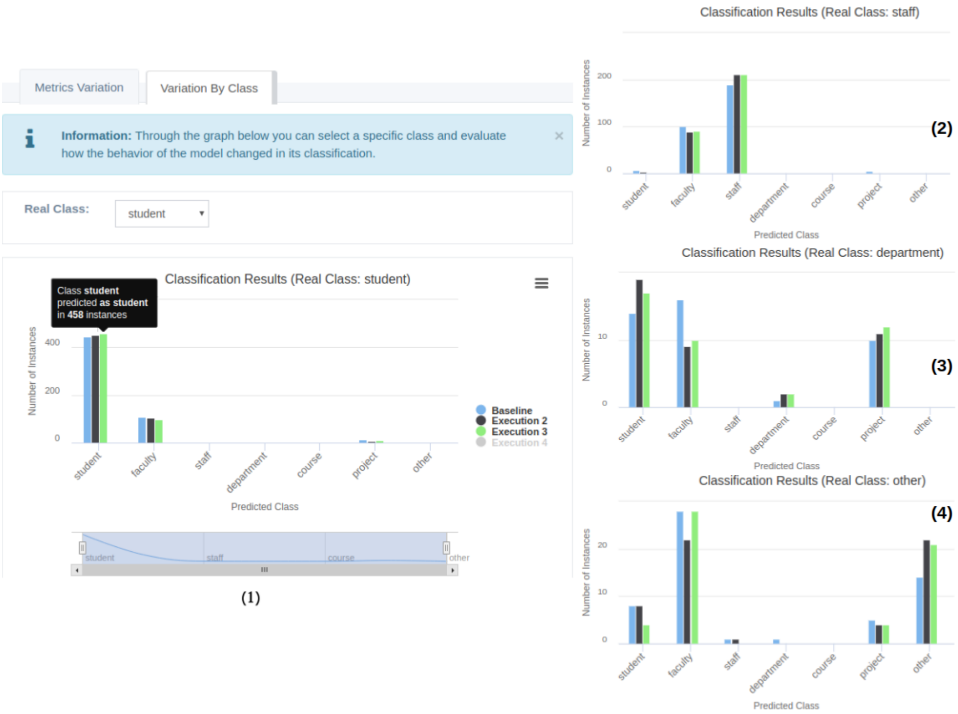
Figure 13. Project execution history - *Metrics Variation* chart . Users visualize each metric in each execution, interacting with the respective point in the graph, as it is done for the metric *Micro F1* in *Executions 1, 2* and *3*. Here, *Micro F1* overlaps *Accuracy*, since both present the same values in the three executions. The graph does not show metrics of the Execution 4 (Figure 5) as it has not been completed.

5.4 Considerations on Explain-ML Use and Proposal

Explain-ML offers different perspectives on the model accuracy, class accuracy, the dataset used to build it, its most discriminative features, predictions probabilities as well as the metrics related to the prediction results at a global and local level. These perspectives are presented through the different visualizations available. Considering the multi-perspective nature of the tool, we hypothesize that the diversity and complementarity of the explanations, help the user to form a mental model about the behavior of the model. Thus, it is aligned with one of the design goals for Explainable Artificial Intelligence Systems described by Mohseni *et al.* (2021), which consists in providing comprehensible transparency for the complex intelligent algorithms (Algorithmic Transparency).

We use standard visualizations for covering the *HOW-WHY-WHAT-IF* types of an explanation [Mohseni *et al.*, 2021]. In more details, for the *'How Explanations'* Explain-ML presents graphs about the model such as a feature importance graph and word cloud, among others; for the *'Why Explanations'* the tool presents the views per instance tab, containing specific views for this purpose and that would fit this type of explanation; and for the *'What-If Explanations'* the interaction itself allows the user to make adjustments to the features and to the parameters, retraining the model and regenerating the result that would consist of a *What-if* explanation.

Explain-ML through its workflow, allows the entire process to be performed from hyperparameter tuning to model enhancements, interactively and without coding. The tool presented in this section is a version of the Explain-ML, instantiated with Random Forests. But the proposed approach can be considered predominantly agnostic, since the views of aspects involving explanations mostly apply to several other models (e.g., database statistics, effectiveness metrics, importance features for the model and for instances, confusion matrix, etc). For the configuration of other models, e.g., neural networks, the parameterization issue would be different, but this is not the main point of the tool. For the first version of Explain-ML, we opted for an iterative process with the Random Forests model, which is naturally more interpretable, before includig others. For the inclusion of other models, in addition to the parameters issue, some more specific visualization of the model itself can be included (e.g., visualization of network layers), but this would be complementary to the existing visualizations, corroborating the multi-perspective purpose of the tool. Explain-ML provides complementary visualizations that allow users to acquire insights regarding the models. Several examples of these insights



Information: Through the graph below you can select a specific class and evaluate how the behavior of the model changed in its classification.

Real Class: student

Class **student** predicted **as student** in **458** instances

Figure 14. Project execution history: *Variation by Class*. (1) graph presenting the behavior of the model for the given real class selected by the user - *student*; (2)(3)(4) graphs resulting from the selection of classes *staff, department* and *other*, respectively.

were discussed throughout the visualizations being presented in this section.

The development of Explain-ML using a human-centric approach is one of its original contributions. In addition, it has a broader scope than most of the existing similar tools: it is interactive, designed based on human-centric principles and spans three scopes of work: (i) Model Building; (ii) Model Interpretability without programming; (iii) and Outcome Explanation.

Furthermore, when compared to previous approaches, ours was developed by explicitly considering user requirements towards interpretability and ease of use. Users can perform tasks related to the entire life cycle of a machine learning model: model training, predictions, model evaluation, model adjustments and model re-execution. The implemented tool, Explain-ML, helps users to build models interactively, without the need of source code manipulations. Besides, the user may execute the model several times, making adjustments. For each execution, information on the overall model, the training dataset and instance level information are stored for visualization purposes. The tool provides a history of the executions with comparative graphs on the evolution of the model in the different executions. There is a common belief that to augment interpretability, ML models have to become simpler and thus less effective [Guidotti *et al.*, 2018b; Breiman *et al.*, 2001]. We take a different perspective, in which we allow users to understand the model and improve it, potentially increasing its effectiveness [Krause *et al.*, 2016a]. To do so, our tool allows users to inspect the model: (i) as a black box (despite its complexity) by understanding its outcomes, its successes and failures; (ii) from a global (e.g., whole dataset) and/or local (e.g., instance) perspective; (iii) through different and complementary representations and visualizations.

6 User Evaluation

In order to evaluate Explain-ML we conducted a qualitative study that allowed us to analyze in depth the users perspective and perceptions on the tool. The methodology adopted combines user interaction with the tool along with semi-structured interviews aimed at exploring how they perceived the visualizations that explain the model's outcomes. As mentioned, the study was conducted using the previously described instantiation of Explain-ML with RFs applied to text classification. In the next sections we explain in more detail the methodology adopted, how the evaluation was conducted and the results obtained.

6.1 Methodology

For the evaluation, users were asked to analyze the results of a specific classification task on a predefined dataset. The interaction was guided and participants were asked to train the model and go through all the visualizations that explained the application of the trained model to the chosen dataset. Then they were asked to interact with the tool to improve the model, and finally to analyze the execution history shown in Explain-ML. During their interaction, participants were encouraged to think aloud [Preece *et al.*, 2019] and at the end of each step a short semi-structured interview about their views on the tool was conducted.

The WebKB dataset was selected for our evaluation[12]. It contains 8.282 webpages classified under the following cate-

[12] For more information on the dataset see: `http://www.cs.cmu.edu/~webkb/` (Last visit: May 2021).

gories: student (1641 pages), faculty (1124), staff (137), department (182), course (930), project (504), and other (3764). It is an interesting dataset for the analysis, as it contains many classification challenges: skewed (imbalanced) class distribution; non-trivial semantic overlap among classes; noise and ambiguity in the text of the pages, etc. Thus, our goal was to analyze whether Explain-ML could help users to identify issues related to class imbalance and corresponding biases (e.g., with the *Class Distribution* visualization), which evaluation metrics would be more adequate to dataset (e.g., with the *Evaluation Metrics* visualization), which words help and which ones confound the decisions of the classifiers, among others.

In order to contextualize the evaluation tasks for the participants an evaluation scenario [Carroll, 2000] was created. The scenario described a situation in which users were participating in a research project that required the automatic classification of their University's webpages. The scenario motivated them to use Explain-ML (a new tool they had just heard of) with a similar dataset (WebKB) to assess whether it would be useful to apply it to their own context (described in the scenario).

To do so, they used the tool to create a Random Forest based ML model for the problem, determine their trust in the model and its results and improve their model based on hints gathered from the explanations. An overall brief explanation of the WebKB dataset – with explanation about the meaning of the classes and corresponding statistics – was also presented to them.

Next we present the steps used to guide participants in their interaction with Explain-ML, and the interview guide associated with each of them. Initially, we asked participants about their experience with ML and Random Forest, specifically. Then we presented Explain-ML to participants, describing its goal to explain the model, and we answered whatever questions they had about the tool. Once the participants had a general view of the tool we guided their interaction through it. We next describe the 4 main tasks that comprised the interaction steps, as well as the semi-structured interview associated to each one of them:

- **T1: Initial model training:** In this step participants were asked to perform the hyperparameter tuning and train the classifier (as shown in Figures 6). Although the tool allows for the tuning of a number of different parameters. – *Interview guide: After this step, participants were asked if they usually performed a hyperparameter tuning, and if so, if they used any tools to do so, and which one(s). They were also asked what they thought about the set of parameters available for tuning.*
- **T2: Analysis of model execution visualizations:** the visualizations of the tool act as explanations for interpreting the model. Participants were asked to analyze each one of the explanation visualizations available (presented in Section 5.2). – *Interview guide: For each visualization, participants were asked about their understanding and opinion about the explanation. After having examined all the explanation visualizations, participants were asked to comment on their analysis of the model, pointing out problems they identified, changes they would make, as well as how they perceived the usefulness of the set of visualizations provided to explain the model.*
- **T3: Model improvement:** Participants were asked to indicate changes they thought could improve the model and execute at least two of them. They were expected to interact with Explain-ML to make the changes they felt would be the most relevant in the model (not necessarily all of them, in the interest of the time of the evaluation), and run a new execution of the model. – *Interview guide: Once the re-execution was performed, they were asked about the changes made and whether they could see their impact in any of the visualizations and, if so, what it meant in terms of the model. If they felt that they were able to improve the model, and whether the changes increased their trust in the model.*
- **T4: Analysis of history results:** Finally, participants were asked to explore the execution history (even though they only conducted two executions of the model) available (presented in Section 5.3) – *Interview guide: They were asked to comment on the changes in the model being conveyed through the history execution visualization and their usefulness.*

After the guided exploration of the tool was over, participants were asked about their overall perception of the tool and its explicability. They were asked if they had knowledge of or had ever used other explicability tools, such as Lime [Ribeiro *et al.*, 2016] or SHAP [Lundberg and Lee, 2017]. Finally, they were encouraged to make any other comments or suggestions they had about their experience in exploring the tool.

6.2 Conducting the Evaluation

As the tool was aimed at ML knowledgeable users, the invitation to participate in the evaluation was sent to Computer Science graduate students who had some knowledge of Machine Learning or whose research was related to the topic. The invitation was sent by email, and those who responded agreeing to participate, were contacted to schedule a convenient time for their participation.

Among the people who responded, we were able to schedule six[13] of them to participate in an evaluation session. Sessions were conducted in an office at the University, between 24 and 30 of October, 2019.

Before starting the session, participants were informed about the goal of the research and asked to sign an informed consent form. The Explain-ML - Random Forest version of the tool was set up in a computer made available to users, and the user's interaction and audio for all sessions were recorded. The audios recorded of the interviews were transcribed, and an analysis conducted. All participants were male and had their background in Computer Science. One of them was postdoctoral fellow, and the other 5 were PhD students. All of them had already experience with Machine

[13] When the goal of an evaluation is to yield insight about a system, a qualitative study with around 5 users is enough [Preece *et al.*, 2019].

Learning model during their graduate studies, but their experience varied in terms of how long they had used ML, which models and for which purposes.

Vilone and Longo [2021] cites that Human-centered evaluation for explainable artificial intelligence can rely on Qualitative studies. According to them, qualitative studies are based upon open-ended questions aimed at achieving deeper insights and they do not provide any guideline towards the ideal number of participants for a qualitative study.

Indeed, qualitative research as the evaluation conducted in our article usually relies on a low number of participants, focusing on delving deeper into the issues that emerge as relevant [Preece *et al.*, 2023]. In our analysis, the use of excerpts from users' speech are used as evidence to illustrate the discussion. In the analysis carried out, all statements from all participants were considered and one of the statements that illustrated well the evidence that supported the analysis was selected.

6.3 Results

We present the main results of our evaluation[14], regarding users understanding of the model based on Explain-ML's explanations and their perspective on its support and usefulness in improving the model. We present the results organized by the aspects explored by participants in each task.

6.3.1 Initial model training

All participants were able to understand and perform the tuning process without difficulties. They said that this was a step that they usually performed through SciKit or by using a script they had written previously. Although they were asked to only make decisions regarding the number of estimators, they observed and commented on the other hyperparameters made available. They mentioned that all the hyperparameters they usually used were depicted in the interface and that usually they did not even use all of the ones available. In short, participants thought it was useful to have the hyperparameter tuning integrated into the Explain-ML tool, and had no difficulties in understanding the step in the workflow, or how it was presented in the interface.

6.3.2 Analysis of model execution visualizations

The first visualization explored was the one that shows the overall **Evaluation Metrics** for the classifier model (see Figure 7). Most participants had a good knowledge of the metrics shown. P6 was not sure about the meaning of MicroF1 and MacroF1 and used the explanation available about each metric to get a clear understanding of the difference between them. All participants were able to explain how they could use the metrics to analyze their confidence in the model, as described by P4:

> *"Micro is a general average and the macro is an average per class… One can already get a general sense..., for example, we know that this dataset here is imbalanced because it [the classifier model] couldn't learn from some of the classes." (P4)*

They all agreed that the most used metrics were being shown in the interface, as explicitly mentioned by P5: *"From what I see, the most used metrics are available."* However, they did not all agree that this was the most interesting set of metrics to be shown. Some participants thought that not all metrics depicted were equally important (P3) or even that some would not be needed (P4, P5). In this direction, P5 suggested that users should be able to configure which metrics they would like to see (or not) in the interface.

On the other hand, other participants mentioned other metrics they thought would be interesting to see. They mentioned general metrics, such as Entropy, Precision or AUC (Area Under the Curve), as well as providing more information about the metrics that were being depicted. Both P4 and P6 made suggestions of adding information to the metrics that would allow them to analyze their variability, such as standard deviation and confidence interval[15]. Participants also suggested that metrics specific to the ML classifier at hand could be interesting to be depicted, as mentioned by P6 regarding Random Forests:

> *"There are some [metrics] specific to the classifier, that might be the case to include..., for example, what was the average depth of the tree, or the gini index of the trees. The problem is that when you generate too many trees, it could be difficult to visualize, right? But at least one way to present the distribution of indices, feature relevance [...]" (P6)*

Finally, one participant (P5) thought that it would be interesting to have the metrics depicted not only for the classifier model, but also for each one of the classes.

Next, participants examined the **Class Distribution** visualization (see Figure 8). All users were able to visualize and understand the distribution of classes. All of them considered this visualization of the distribution very relevant. They used the class distribution to verify issues regarding the skewness (i.e., imbalance) of the dataset used, as exemplified in P4's analysis: *"Its possible to see that the base is very imbalanced, as I had mentioned before."*.

Learning (or confirming) that the dataset was imbalanced helped users to further analyze and better comprehend the previous evaluation metrics values, as well as to identify possible biases in the model. P1 describes how this knowledge informed him about which metrics to look at closely: *"Indeed, if classes are imbalanced, you need to look at those metrics, micro and macro F1."*.

P2's analysis also makes it clear how the class distribution visualization relates to the metrics visualization: *"I look at this distribution, how many documents are in each class, to know which metric will be the most interesting one to inform me whether my model is effective or not. Because, for instance, looking at accuracy on a very imbalanced dataset doesn't mean much … "*.

[14] Our study was conducted in Portuguese, participants' 1st language, and excerpts shown were translated by the authors.

[15] Though planned, this capability was not implemented in the tested version of Explain-ML.

These strategies described by the participants corroborates our *multiperspective hypothesis* that one needs to see the "big picture" (in our case, a comprehensive set of complementary visualizations) to better comprehend ML models and their decisions.

The following visualizations depicted **Feature Importance** (see Figure 9) and were considered extremely relevant by the participants. Four participants (P1, P2, P5 and P6) highlighted how these visualizations have assisted them to better understand (i) how their model made decisions; (ii) which were the most discriminative features of the model, as well as (iii) the noisy features in the classification task.

P2 describes how the visualization helps him understand the model: *"I can use these explanations to understand how the model makes its decisions and to define whether an instance is from one class or another. And when I look at a graph like this... Every time a very meaningful word comes up, I can map it very assertively to the decision model. Then you can get which words, here at the end of the list, do not necessarily have this assertive capability."*.

P1 comments on how that a word that he would expect to be relevant to the context of the database, but through the explanation notices that it has not influenced the model: *"The word "college" surprised me (it didn't appear in the top of the rank among the most discriminative), maybe it's because it is spread across all classes, so it won't help discriminate."*.

P5 describes how he would use Explain-ML to explore and understand the model: *"I think this is amazing, because now, with what you [the tool] told me , I can say... Hmm, let's suppose the model is misclassifying because of the feature [word] "student", then you say: I'm taking "student" out... And it's among the most important features."*.

Three of the participants (P1, P3 and P4) considered the visualization of the low-ranked features as very important to assist them in selecting features for the model. Based on this, they pointed out feature selection as an important aspect for dimensionality reduction, understanding of the model and for potential improvements. The following quotes from P1 and P4 describe their strategies to improve their model by analyzing the low-ranked features:

> *"I usually look at this feature importance table to find out which features are the most discriminative, the ones that really help the classifier ... but it's also very nice to see the least discriminative features – so, if I want to make a feature selection, I know these are the ones I can eliminate from the model without much of an impact." (P1)*
>
> *"It's cool because you can perform feature selection based on this. For instance, you can set a threshold of 0.002 and you do not keep values smaller than this. Then you can remove a lot of features ... It produces a "good" reduction of dimensionality. (P4)"*

It is interesting to note that most interpretability tools, mainly when explaining individual instances, only present the most discriminative features.

Next, participants examined the visualizations **by Instance** (see Figure 10). The visualizations were considered extremely relevant by the participants, and all users were able to visualize and understand them. Participants indicated that these visualizations have assisted them to (i) understand how the model made decisions for each instance, (ii) which were the most discriminative features for particular instances, as well as (iii) which features cause confusion to the classifier.

P2's quote illustrates how this visualization is useful to understand the model decisions for the instances presented: *"I found this one super cool. I can see what was decisive for the model to predict that instance as being in that class."* Participants also highlighted that visualizing the probabilities helped them to identify biases and understand the certainty level of model for each classification, as described by P1: *"I can see the probabilities of each class. I can see for which ones it [the model] is most certain of the result."*

Notice, that the Visualization by Instance is interactive, and users can eliminate features from the model at the interface. Nonetheless, some participants mentioned that it would also be interesting to offer users some more automatic strategies to do so. P1 describes how he usually goes about the issue: *"I try to look and see what is generating noise. But we usually do it more automatically. I would take a script and already eliminate the top X words."* Based on the strategies they described, it would be useful to consider allowing users to determine ranges of features to be eliminated, based on a threshold (e.g. 30% worst) or on the value of their importance (e.g. all features that have importance zero or below a certain value).

The last visualization of the execution results shows the results **by Class** for the classifier model (see Figures 11 and 12). Most participants had a good knowledge of the confusion matrix shown and considered that it was relevant to depict it, as commented by P2: *"It's a concept I'm already used to, the confusion matrix. I liked the colors, I think they highlight ranges that better direct your analysis."*.

The analysis of the results by class allowed participants to identify which classes were the easiest and hardest to predict, as illustrated by P1's analysis: *"The class 'other' is really hard to classify, it is spread here [pointed to the line for 'other' in the confusion matrix]. It seems that the 'faculty' class is the easiest to classify, it gets it right more often."*. They were also able to identify biases in the data such as mentioned in P6's analysis: *"It [the model] is extremely biased by the amount of examples... I noticed based on the classes 'college' and 'department'. 'Department' was the one misclassified the most and 'college' was the one it [the model] got right the most. (P6)."*

Moreover, two participants (P3 and P5) reported that these visualizations helped them identify the need for class balancing. P5 describes how he uses the classes visualization to identify which classes the model is not able to correctly classify, and how he would combine this information with the visualization by Instance to better understand the problem: *"I always want to know the following: Where is my model missing the most and why? So here [seeing the class visualization] I can see which classes it's getting wrong the most, and maybe in the previous step [by Instance] I'll see why it's missing... because of some of the features it's choosing. So this might help me balance the classes."*

Based on identifying the need to balance classes, P3 describes how he would go about it, and conveys his expectation that Explain-ML could also support this task: *"Did the tool have the option to balance? From each class, take the same amount of documents… I would lessen this impact… I would balance, but it would have to balance at the same rate, otherwise it would be different from the real world."*

Finally, the visualization by class supported participants in making decisions about the classifier, not only in a more immediate level regarding which the next step would be (as mentioned by P3), but also about their overall strategies towards improving the model (as indicated by P2).

> *"P3: From the analysis, I have to greatly improve my classifier for 'department' and 'course', right? The ones that are getting misclassified the most."*

> *"P2: It allows you to make decisions like, instead of making a multiclass classifier, making an essemble, and having separate classifiers where you could put the most discriminating thing analyzed separately. This [the visualizations] allows the researcher to make this kind of decision."*

After having explored each one of the 5 visualizations, participants were asked whether they believed the visualizations were useful, and if so, how they had helped them better understand the model. All of the participants considered the explainability offered by the visualizations to be useful. P4 makes it clear that he considered Explain-ML tool gave a very complete account of the model: *"I could understand very well [the model]. Even though I already knew the dataset, this here [the visualizations] is a complete explanation of the dataset. I don't see any other information beyond what's here, you know ?!"* P3 also describes the overall benefits he perceives: *"The tool was good for visualizing the data and proposing improvements."*

Although they all thought it was very useful in general, some of the participants highlighted different aspects of Explain-ML they thought were specially interesting. P1 commented about the usefulness of the model, and emphasized the value of the word cloud: *"I could understand the result. I can see the model well with the word cloud, for text [classification] it is very good."* P6, on his turn, stresses the value of some of the other visualizations: *"Sure [the visualizations were helpful], for understanding, of course [...] The analysis by class, matrix of confusion, sure. Regarding the importance of features, I think this graph gives a good idea of how the importance of features is distributed."*

P2 highlighted on the benefits of the Feature Importance visualization: *"They [the visualizations] are very useful, [...] Specially when you have this very large volume of features, it's hard for you to understand these results... how the classifier is using this information to generate the results."* He also commented that *"I think all the proposed visualizations add a lot and they are quite complementary."*

P5 expresses as a positive point Explain-ML's multiperspective approach: *"I think the information I got here would enable me to refine my model much more easily than I would if I had to do testing, training... Here I have a tool that helps me. [...] Besides doing ... tuning the hyperparameters I would have to perform a large evaluation just to get to the same point that I would by just executing your tool. It really helps not only fine tuning the parameters, but also fine tuning the model. ".*

Its interesting to notice that although participants highlighted different aspects of Explain-ML, they explicitly mentioned that the tool supported them in the main tasks related to modeling a classifier, namely understanding (i) the results of the trained model, (ii) how the classifier uses the information to generate the results, (iii) problems and (iv) biases in the trained model, (v) as well as assisting in the identification of possible improvements in the model.

All participants considered Explain-ML useful. Nonetheless, some of them made suggestions on other aspects they thought could improve the tool. Three participants suggested including a visualization of the importance of features for each prediction class, as illustrated by P6: *"I don't know if there is a concept of the importance of features per class, so maybe it would be the case to include it.".* P2 also suggested it would be interesting to present the relationship between the model features: *"I think this work has a huge potential to expand if we consider getting into how features interact, especially when you have other [machine learning] models.".* Finally, specifically for the use of Random Forest models, P1 suggested that it would be worth considering the possibility of displaying a simplification of the forest structure generated: *"One thing I'd like to see in the models, which I know is very difficult, is either the whole tree or a piece of it."*

The suggestions are interesting, as they provide us with other aspects to consider that could complement the current explainability provided. The suggestion of visualizing the tree when using the Random Forest model has been considered, and is on our intended list of visualizations worth adding. The challenges encountered in including it in this version of Explain-ML were related to the large size of the trees, and providing a visualization that in fact would be useful. The fact that three out of seven participants made the same suggestion regarding the visualizations of features per class could be perceived as a sign of its relevance and the need to consider including it as suggested. Finally, P2's suggestion considers also dealing with other machine learning models. As Explain-ML is intended as an agnostic model tool, these comments are especially useful and will certainly be considered in the next steps of our research.

6.3.3 Model improvement

After having analyzed the model, participants were asked to indicate and perform some model improvements (at least two changes) they had identified in the previous steps as potentially useful to improve the model. Participants adopted different strategies in their attempt to improve the model: (i) removal of less important features (P1, P2, P4), (ii) removal of most "common" features (P3), (iii) removal of features that appear to be noise in the dataset (P5, P6) and (iv) change of hyperparameters (P5).

P1, P2 and P4 removed some features they considered less important. Based on the explanation provided by the visualizations resulting from their strategy, participants were able to easily evaluate the impact of the changes they made. P4's comments on the results of this strategy: *"Here [Evaluation* *metrics visualization] it gives me the information about what has improved and what has gotten worse. [...] It is noticeable that the rest of the features also differed in importance. [...] Improved in micro but got worse in macro.”*. Although P1, P2 and P4 adopted the same strategy, they chose different features to remove, and obtained different results for their changes.

P5 and P6 decided to remove features that seemed to them to be noise in the dataset, and P5 also made changes in the hyperparameters. In the case of P5, the performed changes had higher impact on the model, albeit negative as he expressed: *”Wow, I made the whole thing worse. See... I removed a very important word for it [the model], see what happened ?!”*. As he explored the different visualizations, he commented on the various impacts he noticed of the changes: *“Gosh, accuracy is the same value. Got a little worse too.”*; *“It seems like the number of [features with] importance zero has decreased.”*, and *“It makes more sense now, the words you use to predict.”* Based on P5 comments it seems like, even though the changes decreased the model’s performance metrics, it made sense as a step to improve the model, as he mentioned: *“It improved the confidence in the model, how to explain it to others... I can explain it much better.”*. He concluded, commenting on Explain-ML’s support: *“It showed me why it got worse.”*

P3 adopted a strategy that may not seem obvious at first, as he removed an important feature from the model. He explained that, as the feature was a generic word in the domain (‘computer’), he removed it to see if other features would become more discriminative. Analyzing the changes he thought it improved the model: *”It has improved... the error has decreased...”*. Nonetheless, P3 commented he would have liked a little more information to analyze if the strategy had been successful: *”I wonder if it increased the difference? When I took it [feature ‘computer’] out, did this one here [pointed at feature ‘science’] become more important?”*. Specifically he mentioned it would be interesting to be able to compare feature importance from one execution with the previous one.

It is interesting to notice that although all the participants were examining the same model, based on the visualizations, they chose different actions they believed could improve the model. Even though some of them were not successful, participants were able to identify the low or negative impact of their changes and make considerations of what would be the next steps they would take in improving the model.

The different strategies adopted depend not only on the tool being used, but also on participants’ previous experiences. Although all of them were knowledgeable ML users, they had different levels of expertise. Furthermore, we could argue that because Explain-ML provides a broad account of the model’s execution, it does not guide users to any specific path to improve the model, but rather offers users a broad exploration space. This provides users with autonomy to make different decisions and conduct distinct investigations and analysis according to their contexts (i.e. dataset, classification goal or ML classifier).

6.3.4 Analysis of history results

Lastly, participants were asked to explore the execution results history. Even though participants compared only 2 executions, they all considered the execution history useful in helping them to generate an overall view of their tuning of the model, as expressed by P2: *“So here I could have several executions and check how much is changing according to these changes that I am making in my model [...] Very cool, because then you can see this variation in terms of execution [...] I think it is cool to see the impact of my changes on the model.”*.

The participants made it clear *how* the execution history helped them better understand the impact of their changes in model, as illustrated by P1 and P5: *“I could see the impact here, 0.1%, because I eliminated only a few features. (P1)”* and *“Overall, I realized that some classes have not changed at all. (P5)”*.

Finally, based on the execution history some of them commented how it could provide them with insights regarding their next steps in tuning the model, as described by P6: *“It is possible to have an intuition of what can be improved, but that I haven’t improved yet [...] It was possible to have more confidence, but I still don’t feel confident with this result.”*.

6.3.5 Overall comments

In the last part of the interview, participants were asked about their overall view of the system and whether they thought it would be useful to them considering their activities and strategies to use ML modeling. All participants stated that they would use the tool, as illustrated by P6 and P2: *“I found it amazing [...] It really makes experimentation a lot easier. (P6)”* and *“I would super use this. I think the usefulness of this is huge. I think all the proposed views add a lot and they are quite complementary. (P2)”*

As users explained why they would use Explain-ML, many of them highlighted some of the aspects they believed were the most interesting in the tool. It worth noting that the participants had different views on what was the “most” interesting. For instance, P3 focused on the metrics, whereas P4 mentioned the execution history:

> *”I would use it [Explain-ML] to improve the model. [...] The impact of a feature on classification, I think this is the most important [visualization].... And the metrics also... Exporting the confusion matrix is also important.”*

> *”I would use it [Explain-ML] to know where the error came from and where it went [...] This possibility to keep the execution histories is very good, because we usually do it through a table, but in a table you forget... you don’t put parameters, don’t put a lot of things, and here in the system there is all that. It’s a way to keep it that way, an organized way, and it’s not a hassle keeping everything you need. I think everything I do [manually] is already there.”*

Some participants had experience with other explainability tools and compared them to Explain-ML:

> *"I've already used SHAP. Actually I don't use all it offers, because I know it does a lot of things, but the things I have used I think you're doing [in Explain-ML].(P2)"*

> *"I used another tool with less features than this, which is Lemonade*[16]*. Everything it has yours already has. Yours has more features [...] Parameter tuning is one thing that is not available there [in Lemonade], there we select the parameters we want and execute it with these parameters. [...] Being able to see the results for instance, and everything integrated, the word cloud, the graphs, the metrics, is not something I [usually] see. (P1)"*

One relevant shortcoming of Explain-ML pointed out by participants is that it only offers Random Forest model, while other explainability tools cover a varied set of ML classifiers. P1 illustrates this point: *"It [Lemonade] has already incorporated other models (cluster, frequent patterns, other machine learning algorithms), but it doesn't have as many views as it has here. So I would incorporate that, include other model types."*.

As advantages and benefits of Explain-ML, besides its usefulness, some participants pointed out that it offers an easy-to-use interactive user interface, as highlighted by P5: *"I liked that... besides being useful, a system has to have a friendly face [...if it] doesn't have such a simplified usability... [then] the person stops using it. Seeing these graphs, seeing this facilitated interaction, it pleases the eyes to see. Even if you get stuck at some point, you want to keep moving, because it gives you good information in a way that your brain easily assimilates, so it's very cool."* Participant 2 emphasized that as it did not require any programming knowledge from the user, he considered it could be interesting and useful to a wider audience than most tools: *"P2: I would use it for sure... Me, from computing, would use it... But I think it has an even greater potential when you deal with what I have already commented... When we start thinking about outsiders [from the computing field]... SHAP and these other tools have their limitations: the guy will have to generate the model, will need to implement it in python, he will need to know how to manipulate [the tool]... To use this here [Explain-ML] I do not need to know python ... So it has a very interesting potential, a wider audience."*

Finally, P5 mentioned that he thought Explain-ML would be useful not only for ML modeling, but also he thought it could be useful for the research in the field: *"I wish it were for all the other models [...] Most 'state of the art' works today don't have that. The guy [author] just says: my metric beat the state of the art, but he doesn't say why, he doesn't show why... If you had such a tool to explain the model, you would know where it is weak, what are the constraints of that model, where it is strong, where it is weak [sic]. And that is what is missing in most articles today."*

Next, we discuss the main implications of the results of the evaluation for Explain-ML and our research.

7 Discussion

Our evaluation of Explain-ML allowed us to explore in depth the perspectives the participants had of the tool, as well as how the proposed solution contributed to ML modeling and their activities in general. Based on participants perspectives and experience (albeit short) with Explain-ML, we collected positive indicators about our intended solution, as well as how it was presented to users.

Participants explicitly expressed the advantages they perceived in Explain-ML taking a **multi-perspective approach** and integrating steps ranging from hyperparameter tuning, to model execution, the exploration of its visualizations/ explanation, all the way to, and model improvement (including comparing different executions). In exploring the explanations offered, they considered them complementary and informative. Based on the insights they gained through their exploration and that were expressed in their interviews, it is possible to see that the tool allowed them to understand the outcomes of the model, as well as how it was working. Furthermore, they were able to identify relevant strategies they could take to improve the model, and make changes in that direction. In other words, Explain-ML's **black-box explanation** approach, based on **outcome explanation** and **model inspection** was well received by users and generated positive indicators of how it could be successfully used in ML modeling.

The current implemented version of Explain-ML offers only Random Forest modeling. The reason for this decision was that, in a human-centered machine learning approach it made sense to evaluate one model, and its potential use and adoption by users, before investing in including all the other models. Nonetheless, the visualizations available are not specific to RF model and could be equally useful with many other ML models.

The approach adopted to evaluate the explainability of the tool can be considered in most part model agnostic (independent of models), since the visualizations of the aspects that involve the explanations mostly apply to several other models (e.g., database statistics , effectiveness metrics, importance of features for the model and instances, confusion matrix, etc.). For the configuration of other models, e.g. neural networks, the parameterization issue would be different, but that is not the main point of the tool.

For the inclusion of other models, in addition to the issue of parameters, it is possible to include some more specific visualization of the model itself (e.g., visualization of network layers), but this would be complementary to the already existing visualizations, helping incrementally from a perspective multiperspective.

Participants expressed great interest in Explain-ML and considered it relevant and useful, which indicates that it would be valuable to continue this research and the development of the tool.

The goal to continue developing Explain-ML as a **model agnostic** tool was also reinforced by the (spontaneous) suggestions made by many of the participants to include other classification models in the tool. Furthermore, throughout the evaluation participants made other relevant suggestions on how to improve not only the explanations – suggesting

[16] The participant was taking a Datamining course in which the Lemonade was used (`https://www.lemonade.org.br/index-en.html` for more details). The tool presents certain visualizations, but in another scope.

more information in some of the visualizations, or even new visualizations, but also on the interactive process, commenting on aspects that would be interesting to allow users to tailor to their own needs. Moving forward with our research and the development of Explain-ML, all of them will be carefully considered.

Finally, participants praised the focus on the tool's interactivity – not requiring any coding in any step of the process, and usability. They pointed out that these decisions would benefit not only the intended knowledgeable user, but could also allow for a broader use of Explain-ML.

Although the user evaluation was very useful to generate insights about our proposed multi-perspective approach and the Explain-ML tool, we must acknowledge its limitations. Our research adopted a qualitative method with the goal of exploring users' perspectives in-depth. However, as is the case with qualitative methodologies [Lazar *et al.*, 2017; Flick, 2008], the evaluation involved a small number of participants (6), in a specific context, which by design is not generalizable. Furthermore, even though a small number of participants is adopted both for interaction design with focus on the interface [Preece *et al.*, 2019], and qualitative research [Lazar *et al.*, 2017], it may be biased and not represent all potential users of the system. As future steps in this research, we intend to conduct broader evaluations, including more participants, with more diverse backgrounds and levels of experience in ML. These new studies are intended in the future steps of this research, and their results will be contrasted with the current ones.

8 Conclusion

In this work, we have presented our HCML approach to generating a multi-perspective interpretable system to support the generation of an ML model interactively and a system based on our approach - Explain-ML. Our in-depth evaluation of Explain-ML has shown that it provides users with a broad exploration space for interpretability questions. This space can be flexibly exploited considering the users' own goals, background and expectations. Results also demonstrated that we have achieved the vast majority of the goals we have set for our tool, as ultimately demonstrated by the users' enthusiasm in using Explain-ML in their own activities. We held a discussion about the evaluation performed [Lopes *et al.*, 2021], grounded by IML principles [Dudley and Kristensson, 2018]. The discussion describes how Explain-ML fulfills each of those principles and the target users needs regarding ML interpretability. And how those needs align with the principles.

The requirements generated from this research can be useful to other researchers and developers interested in explainability systems who can build on them to generate new systems or evaluate existing systems. Our multi-perspective approach proposes a combination of broad and complementary perspectives that can be useful to guide the analysis, evaluation and reposting of Machine Learning models in research. They also can be applied to multiple models and contexts. Our proposed tool, although not yet available, has epistemic potential, and its development is guided by demands from ML model users. Our in-depth evaluation of Explain-ML has shown that it provides users with a broad exploration space for interpretability questions. These results can be useful to other researchers in considering different perspectives for explainability visualizations and tools.

We believe that we have advanced the state-of-the-art in Explainable Learning by putting the user center stage, from the inception, to the design and evaluation of a multi-perspective approach, and system based on this approach. Explain-ML allows users to, based on the knowledge they acquired through the visual explanations (including the historical views), make changes in the models in a virtuous cycle of "understand-by-doing".

Furthermore, Explain-ML's ease-of-use, with no need for any coding, along with its appreciated aesthetics, were highlighted several times as major positive assets. This means that a natural venue for investigation is experimenting with the tool with less 'knowledgeable' users in real-world ML projects and different tasks.

Another venue for investigation is to compare Explain-ML to other tools with similar goals. In a previous study [Lopes *et al.*, 2022], we have compared Explain-ML to our analysis of three other systems: Rulematrix [Ming *et al.*, 2019], Explanation Explorer [Krause *et al.*, 2017] and ATM-Seer [Wang *et al.*, 2019]; regarding how they address the Interactive Machine Learning Principles proposed by [Dudley and Kristensson, 2018]. Moving forward it would be interesting to compare (qualitatively and quantitatively) Explain-ML and other tools with similar goals (e.g. LIME, SHAP and Lemonade) regarding their support to users modeling decisions.

We are currently working on a new version of Explain-ML that includes additional ML models and visualizations. In the future, we will make available an open Python Code for this project, to allow users to make personalizations and easily manipulate the tool.

With the new version of Explain-ML we will perform a more extensive evaluation were users will evaluate the tool for a longer period and compare its results with other explainability tools, in order to evaluate the ML results obtained using each tool. With that we will be able to evaluate with qualitative and quantitative measures: (1) If the multi-perspective approach contributes to a better understanding of ML models and their results; (2) which perspectives do we draw most conclusions individually or combined; (3) the most appropriate perspectives inherent to the model and the problem; (4) how do perspectives relate to explaining the ML model and (5) what information is gained from each visualization presented.

Other future tasks involve taking into the account several suggestions often mentioned in the experiments, such as the inclusion of visualizations for statistical significance tests, visualization of features importance per class and the inclusion of other ML models, taking into consideration their specificities (hyperparameters, specific visualizations).

Declarations

Acknowledgements

The authors thank all the potential users who kindly accepted our invitation to participate in any stages of our research: interviews, prototype evaluation, Explain-ML evaluation. They also thank the funding agencies CNPq and Fapemig for their partial support to this research.

Funding

This research was partially funded by CNPq and Fapemig.

Authors' Contributions

All authors participated in the conception of the approach, conception of the workflow and Explain-ML tool. Bárbara Lopes implemented the tool and conducted the users evaluations. Bárbara Lopes conducted the analysis of the evaluations and discussed them with Raquel Prates. Bárbara Lopes advised by Marcos Gonçalves organized the related work and comparison of existing systems. Bárbara Lopes and Liziane Soares selected and tested the other similar tools. Liziane inspected the other tools and discussed the results with Bárbara Lopes and Raquel Prates. All authors participated in the writing and reviewing of the article.

Bárbara Lopes: *Conceptualization, Data curation, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing.* **Liziane Soares:** *Conceptualization, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing.* **Raquel Prates:** *Conceptualization, Methodology, Project administration, Supervision, Validation, Writing – original draft, Writing – review & editing.* **Marcos Gonçalves:** *Conceptualization, Methodology, Project administration, Supervision, Validation, Writing – original draft, Writing – review & editing.*

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Explain-ML is not yet publicly available for use, but the material available, including a video demo of the system can be found in: `https://github.com/BarbaraGCOL/explain-ml`.

References

Adebayo, J. and Kagal, L. (2016). Iterative orthogonal feature projection for diagnosing bias in black-box models. *arXiv preprint arXiv:1611.04967*. DOI: 10.48550/arXiv.1611.04967.

Adler, P., Falk, C., Friedler, S. A., Nix, T., Rybeck, G., Scheidegger, C., Smith, B., and Venkatasubramanian, S. (2018). Auditing black-box models for indirect influence. *Knowledge and Information Systems*, 54(1):95–122. DOI: 10.48550/arXiv.1602.07043.

Batista, G. E., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29. DOI: 10.1145/1007730.1007735.

Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., Ohl, P., Thiel, K., and Wiswedel, B. (2009). Knime-the konstanz information miner: version 2.0 and beyond. *AcM SIGKDD explorations Newsletter*, 11(1):26–31. DOI: 10.1145/1656274.1656280.

Breiman, L. *et al.* (2001). Statistical modeling: The two cultures. *Statistical science*, 16(3):199–231. DOI: 10.1214/ss/1009213726.

Cao, S., Sun, X., Widyasari, R., Lo, D., Wu, X., Bo, L., Zhang, J., Li, B., Liu, W., Wu, D., *et al.* (2024). A systematic literature review on explainability for machine/deep learning-based software engineering research. *arXiv preprint arXiv:2401.14617*. DOI: 10.48550/arXiv.2401.14617.

Capel, T. and Brereton, M. (2023). What is human-centered about human-centered ai? a map of the research landscape. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–23. DOI: 10.1145/3544548.3580959.

Carroll, J. (2000). Introduction to this Special Issue on "Scenario-Based System Development". *Interacting with Computers*, 13(1):41–42. DOI: 10.1016/S0953-5438(00)00022-9.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of KDD '15*, pages 1721–1730. DOI: 10.1145/2783258.2788613.

Chawla, N. V., Japkowicz, N., and Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter*, 6(1):1–6. DOI: 10.1145/1007730.1007733.

Choung, H., David, P., and Ross, A. (2023). Trust in ai and its role in the acceptance of ai technologies. *International Journal of Human–Computer Interaction*, 39(9):1727–1739. DOI: 10.1080/10447318.2022.2050543.

Cortez, P. and Embrechts, M. J. (2011). Opening black box data mining models using sensitivity analysis. In *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 341–348. IEEE. DOI: 10.1109/CIDM.2011.5949423.

Demiralp, Ç. (2016). Clustrophile: A tool for visual clustering analysis. In *KDD 2016 Workshop on Interactive Data Exploration and Analytics*, pages 37–45. DOI: 10.48550/arXiv.1710.02173.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*. DOI: 10.48550/arXiv.1702.08608.

Dudley, J. J. and Kristensson, P. O. (2018). A review of user interface design for interactive machine learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2):8. DOI: 10.1145/3185517.

Erik, S. and Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11(Jan):1–18. `https://www.jmlr.org/papers/volume11/strumbelj10a/strumbelj10a.pdf`.

Fails, J. A. and Olsen Jr, D. R. (2003). Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, pages 39–45. ACM. DOI: 10.1145/604045.604056.

Fiebrink, R. and Gillies, M. (2018). Introduction to the special issue on human-centered machine learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2):7. DOI: 10.1145/3205942.

Flick, U. (2008). *Designing qualitative research*. Sage Sage Publications Ltd., 1th edition. Book.

Gillies, M., Fiebrink, R., Tanaka, A., Garcia, J., Bevilacqua, F., Heloir, A., Nunnari, F., Mackay, W., Amershi, S., Lee, B., *et al.* (2016). Human-centred machine learning. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 3558–3565. ACM. DOI: 10.1145/2851581.2856492.

Goldstein, A., Kapelner, A., Bleich, J., and Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *J. of Comput. and Graphical Statistics*, 24(1):44–65. DOI: 10.48550/arXiv.1309.6392.

Goodman, B. and Flaxman, S. (2017). European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57. DOI: 10.48550/arXiv.1606.08813.

Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., and Giannotti, F. (2018a). Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*. DOI: 10.48550/arXiv.1805.10820.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018b). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):93. DOI: 10.1145/3236009.

Hall, P. and Gill, N. (2018). *Introduction to Machine Learning Interpretability*. O'Reilly Media, Incorporated. Book.

Han, Q., Zhu, W., Heimerl, F., Koch, S., and Ertl, T. (2016). A visual approach for interactive co-training. In *KDD 2016 Workshop on Interactive Data Exploration and Analytics*, pages 46–52. `https://poloclub.gatech.edu/idea2016/papers/p46-han.pdf`.

Hooker, G. (2004). Discovering additive structure in black box functions. In *Proc. of ACM SIGKDD*, pages 575–580. ACM. DOI: 10.1145/1014052.1014122.

Kononenko, I. *et al.* (2010). An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11(Jan):1–18. `https://www.jmlr.org/papers/volume11/strumbelj10a/strumbelj10a.pdf`.

Krause, J., Dasgupta, A., Swartz, J., Aphinyanaphongs, Y., and Bertini, E. (2017). A workflow for visual diagnostics of binary classifiers using instance-level explanations. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 162–172. IEEE. DOI: 10.48550/arXiv.1705.01968.

Krause, J., Perer, A., and Bertini, E. (2016a). Using visual analytics to interpret predictive machine learning models. In *Proceedings of the 2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 2016)*. DOI: 10.48550/arXiv.1606.05685.

Krause, J., Perer, A., and Bertini, E. (2018). A user study on the effect of aggregating explanations for interpreting machine learning models. In *ACM KDD Workshop on Interactive Data Exploration and Analytics*. `https://perer.org/papers/adamPerer-userStudy-IDEA2018.pdf`.

Krause, J., Perer, A., and Ng, K. (2016b). Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5686–5697. ACM. DOI: 10.1145/2858036.2858529.

Lakkaraju, H., Kamar, E., Caruana, R., and Leskovec, J. (2017). Interpretable & explorable approximations of black box models. *arXiv preprint arXiv:1707.01154*. DOI: 10.48550/arXiv.1707.01154.

Lazar, J., Feng, J. H., and Hochheiser, H. (2017). *Research methods in human-computer interaction*. Morgan Kaufmann. Book.

Linardatos, P., Papastefanopoulos, V., and Kotsiantis, S. (2021). Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18. DOI: 10.3390/e23010018.

Longo, L., Brcic, M., Cabitza, F., Choi, J., Confalonieri, R., Del Ser, J., Guidotti, R., Hayashi, Y., Herrera, F., Holzinger, A., *et al.* (2024). Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Information Fusion*, 106:102301. DOI: https://doi.org/10.48550/arXiv.2310.19775.

Lopes, B. G., Soares, L. S., Prates, R. O., and Gonçalves, M. A. (2021). Analysis of the user experience with a multiperspective tool for explainable machine learning in light of interactive principles. In *Proceedings of the XX Brazilian Symposium on Human Factors in Computing Systems*, pages 1–11. DOI: 10.1145/3472301.3484360.

Lopes, B. G. C., Soares, L. S., Prates, R. O., and Gonçalves, M. A. (2022). Contrasting explain-ml with interpretability machine learning tools in light of interactive machine learning principles. *Journal on Interactive Systems*, 13(1):313–334. DOI: 10.5753/jis.2022.2556.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774. DOI: 10.48550/arXiv.1705.07874.

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

Mohseni, S., Zarei, N., and Ragan, E. D. (2021). A multidisciplinary survey and framework for design and evaluation of explainable ai systems. *ACM TIIS*, 11(3-4):1–45. DOI: 10.1145/3387166.

Mosqueira-Rey, E., Pereira, E. H., Alonso-Ríos, D., and Bobes-Bascarán, J. (2022). A classification and review of tools for developing and interacting with machine learning systems. In *Proc. of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 1092–1101. DOI: 10.1145/3477314.3507310.

Nakao, Y., Strappelli, L., Stumpf, S., Naseer, A., Regoli, D., and Gamba, G. D. (2023). Towards responsible ai: A design space exploration of human-centered artificial intelligence user interfaces to investigate fairness. *International Journal of Human–Computer Interaction*, 39(9):1762–1788. DOI: 10.48550/arXiv.2206.00474.

Neto, M. P. and Paulovich, F. V. (2021). Explainable matrix - visualization for global and local interpretability of random forest classification ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1427–1437. DOI: 10.48550/arXiv.2005.04289.

Nielsen, L. (2013). *Personas In "The Ency-*

clopedia of Human-Computer Interaction, 2nd Ed.”. http://www.interaction-design.org/encyclopedia/personas.html.

Preece, J., Rogers, Y., and Sharp, H. (2019). *Interaction Design: Beyond Human - Computer Interaction*. Wiley Publishing, 5th edition. Book.

Preece, J., Rogers, Y., and Sharp, H. (2023). *Interaction Design: Beyond Human-Computer Interaction*. John Wiley Sons, 6th edition. Book.

Ramos, G., Suh, J., Ghorashi, S., Meek, C., Banks, R., Amershi, S., Fiebrink, R., Smith-Renner, A., and Bansal, G. (2019). Emerging perspectives in human-centered machine learning. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, page W11. ACM. DOI: 10.1145/3290607.3299014.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD*, pages 1135–1144. DOI: 10.48550/arXiv.1602.04938.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*. https://homes.cs.washington.edu/~marcotcr/aaai18.pdf.

Rong, Y., Leemann, T., Nguyen, T.-t., Fiedler, L., Seidel, T., Kasneci, G., and Kasneci, E. (2022). Towards human-centered explainable ai: user studies for model explanations. *arXiv preprint arXiv:2210.11584*. DOI: 10.48550/arXiv.2210.11584.

Schneider, J. (2024). Explainable generative ai (genxai): A survey, conceptualization, and research agenda. *arXiv preprint arXiv:2404.09554*. DOI: 10.1007/s10462-024-10916-x.

Shneiderman, B. (2020). Bridging the gap between ethics and practice: Guidelines for reliable, safe, and trustworthy human-centered ai systems. *ACM Trans. Interact. Intell. Syst.*, 10(4). DOI: 10.1145/3419764.

Singh, S., Ribeiro, M. T., and Guestrin, C. (2016). Programs as black-box explanations. *arXiv preprint arXiv:1611.07579*. DOI: 10.48550/arXiv.1611.07579.

Smilkov, D., Carter, S., Sculley, D., Viégas, F. B., and Wattenberg, M. (2016). Direct-manipulation visualization of deep networks. In *KDD 2016 Workshop on Interactive Data Exploration and Analytics*, pages 115–119. DOI: 10.48550/arXiv.1708.03788.

Tamagnini, P., Krause, J., Dasgupta, A., and Bertini, E. (2017). Interpreting black-box classifiers using instance-level visual explanations. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, page 6. ACM. DOI: 10.1145/3077257.3077260.

Tolomei, G., Silvestri, F., Haines, A., and Lalmas, M. (2017). Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the 23rd ACM SIGKDD*, pages 465–474. DOI: 10.48550/arXiv.1706.06691.

Turner, R. (2016). A model explanation system. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE. DOI: 10.48550/arXiv.1606.09517.

Vidovic, M. M.-C., Görnitz, N., Müller, K.-R., and Kloft, M. (2016). Feature importance measure for non-linear learning algorithms. *arXiv preprint arXiv:1611.07567*. DOI: 10.48550/arXiv.1611.07567.

Vilone, G. and Longo, L. (2021). Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion*, 76:89–106. DOI: 10.1016/j.inffus.2021.05.009.

Wang, Q., Ming, Y., Jin, Z., Shen, Q., Liu, D., Smith, M. J., Veeramachaneni, K., and Qu, H. (2019). Atmseer: Increasing transparency and controllability in automated machine learning. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 681. ACM. DOI: 10.48550/arXiv.1902.05009.

Wondimu, N. A., Buche, C., and Visser, U. (2022). Interactive machine learning: A state of the art review. *arXiv preprint arXiv:2207.06196*. DOI: 10.48550/arXiv.2207.06196.

Yang, W., Wei, Y., Wei, H., Chen, Y., Huang, G., Li, X., Li, R., Yao, N., Wang, X., Gu, X., *et al.* (2023). Survey on explainable ai: From approaches, limitations and applications aspects. *Human-Centric Intelligent Systems*, 3(3):161–188. DOI: 10.1007/s44230-023-00038-y.

Zhang, J., Wang, Y., Molino, P., Li, L., and Ebert, D. S. (2019). Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):364–373. DOI: 10.1109/TVCG.2018.2864499.

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

Zhou, J., Gandomi, A. H., Chen, F., and Holzinger, A. (2021). Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 10(5):593. DOI: 10.3390/electronics10050593.