Journal on Interactive Systems, 2022, 13:1, doi: 10.5753/jis.2022.2556

Contrasting Explain-ML with Interpretability Machine Learning Tools in Light of Interactive Machine Learning Principles

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

Abstract

The way Complex Machine Learning (ML) models generate their results is not fully understood, including by very knowledgeable users. If users cannot interpret or trust the predictions generated by the model, they will not use them. Furthermore, the human role is often not properly considered in the development of ML systems. In this article, we present the design, implementation and evaluation of Explain-ML, an Interactive Machine Learning (IML) system for Explainable Machine Learning that follows the principles of Human-Centered Machine Learning (HCML). We assess the user experience with the Explain-ML interpretability strategies, contrasting them with the analysis of how other IML tools address the IML principles. To do so, we have conducted an analysis of the results of the evaluation of Explain-ML with potential users in light of principles for IML systems design and a systematic inspection of three other tools – Rulematrix, Explanation Explorer and ATMSeer – using the Semiotic Inspection Method (SIM). Our results generated positive indicators regarding Explain-ML and the process that guided its development. Our analyses also highlighted aspects of the IML principles that are relevant from the users' perspective. By contrasting the results with Explain-ML and SIM inspections of the other tools we were able to identify common interpretability strategies. We believe that the results reported in this work contribute to the understanding and consolidation of the IML principles, ultimately advancing the knowledge in HCML.

Keywords: *Human-centered computing, User studies; Information visualization, Computing methodologies, Machine Learning, Semiotic inspection method.*

1 Introduction

We live in a society permeated by intelligent machines and algorithms, applied to different sectors of society. In this scenario, we increasingly observe the application of techniques and models whose inner workings cannot be easily explained (Labs, 2020). Indeed, most Machine Learning (ML) models, including the most recent and complex ones based on deep neural networks, are considered as "black boxes" due to the inherent difficult of interpreting their outputs (Linardatos et al., 2021)[1].

The reliability of the models and their results are very important, particularly when they are used to support decision making in critical areas as diverse as medical diagnostics, recommender systems, credit analysis, fraud detection and anomaly detection (Linardatos et al., 2021). In general, models are evaluated based on metrics related to the accuracy on datasets available for validation. But real-world data can have idiosyncracies that profoundly affect the behavior and performance of ML models (Ribeiro et al., 2016).

Providing explanations for the predictions generated by a model favors its interpretability and its acceptance by the user (Tolomei et al., 2017). More and more users and companies that adopt Machine Learning models to support decision making point out the need to understand how the model generates its predictions (Labs, 2020). The lack of these explanations is a practical and also an ethical issue (Guidotti et al., 2018b), meeting the demand for transparency defended by several international and national governmental institutions. The European Parliament, for example, adopted the General Data Protection Regulation (*GDRP - General Data Protection Regulation*), which defends the right of users to have access to explanations about the logic involved in automated decision making systems (Goodman and Flaxman, 2017). The explanation behind a prediction can also contribute to improving the model, as it can be used as *feedback* for the model itself, relative to why it produced some results, right or wrong.

At this point, it is worth noting that the concepts of interpretability and explainability have been broadly used, but not always in a consistent way. In some works, these concepts are used interchangeably and sometimes even considered synonymous (Mohseni et al., 2021; Vilone and Longo, 2021; Zhou et al., 2021), whereas in others, authors make an explicit distinction between them[2]. In this paper, we will use **interpretability** and explainability as synonymous, as 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,

[1] Although some models such as decision trees and linear regression are considered as "more interpretable´´, their outputs are still hard to comprehend when applied to large datasets.

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

graphics and visualizations (Doshi-Velez and Kim, 2017; Guidotti et al., 2018b; Hall and Gill, 2018).

Many efforts have been devoted to the problem of explaining the behavior of a ML model or on developing more explainable models (Guidotti et al., 2018b; Linardatos et al., 2021). According to the *Cloudera Fast Forward Labs* report (2020), in interviews with organizations that are concerned with the interpretability of their systems, many of them prioritize white-box models (those that are already designed to be more interpretable), in order to maintain its interpretability, even if there is a "tolerable loss" from the point of view of the accuracy of the results.

All the efforts mentioned are related to the need to make Machine Learning systems more understandable and reliable. However, we observed that the preferences and needs of users of these systems (whether they are ML specialists or end users who just consume the results), related to interpretability and support in decision making, are rarely considered in the design of such systems.

Indeed, there are not many works focused on the intersection between user demands and Machine Learning systems (Ramos et al., 2019; Gillies et al., 2016; Dudley and Kristensson, 2018). The gaps in this intersection have fostered the field of HCML (*Human Centered Machine Learning*), which argues that ML research and system development should be considered from a more human-centered perspective. HCML is an interdisciplinary field that encompasses the perspectives of HCI (*Human-Computer Interaction*) and ML (Ramos et al., 2019; Gillies et al., 2016; Dudley and Kristensson, 2018). At this intersection of HCI and ML, the concept of interactive Machine Learning - IML (*Interactive Machine Learning*) has emerged. In IML, the model training process is treated as an HCI task, in which the user provides input to the tasks of selecting, creating and labeling the instances, and actively participates in an iterative process of modifying and revising the model during its training and deployment (Fails and Olsen Jr, 2003).

Dudley and Kristensson (2018) discuss the design context of IML systems, addressing the key challenges and elements involved. Discussion of the design space of this type of system fosters design considerations that contribute to more efficient and productive IML systems. In this direction, the authors identify **six IML principles** for the design of IML interfaces. We used these principles to guide our evaluations and discussions in this work.

More specifically, in this article, we evaluate a specific IML tool developed by our group – Explain-ML (Lopes, 2020), contrasting it with other Interpretability ML Tools in light of IML Principles. Explain-ML is a multi-perspective tool for Machine Learning interpretability, aimed for knowledgeable users, that is, those having some knowledge about ML models, acquired through the use of ML models in the past. Its development adopted an HCML approach, as it took into account the target users' perspectives and needs. Explain-ML's first version, focused on the Random Forest (RF) model due to the better interpretability of this type of ML model.

We present an overview of the Explain-ML tool and its development (Lopes, 2020) and perform an analysis on the results obtained in an evaluation of the users' experience with the tool. To assess the users' experience (UX) with the interpretability strategies of Explain-ML, a qualitative assessment was carried out with a group of target users, focusing on their perspective on the usefulness of the developed approach. From there, we held a discussion about the evaluation performed, grounded by IML principles (Lopes et al., 2021). 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.

In this article, we extend and enrich the Explain-ML assessment by contrasting it with the analysis of how other IML tools available in the literature address the IML principles. In order to contrast Explain-ML with other interpretability tools, we used the Semiotic Inspection Method (SIM). SIM is a qualitative and interpretative inspection method aimed at assessing the communicability of an interactive system by inspecting the meta-message sent by the designer to users.

Our goal was to use SIM to systematically carry out the reconstruction of the design meta-message of each tool, in order to identify: (i) the interpretability strategies offered, and (ii) the compliance of the meta-message with the IML principles. We selected three tools: RuleMatrix (Ming et al., 2019), Explanation Explorer (Krause et al., 2017) and ATMSeers (Wang et al., 2019). The selection considered the literature review performed and was based on three criteria: (i) Work Scope: in line to the Explain-ML scope; (ii) Interactiveness: required for SIM application; (iii) Availability for inspection, which ideally meant the system being available to be installed/used or providing other materials or means to inspect it. We present the main parts of the meta-messages of each of the three tools and discuss the compliance of each one with the IML principles, thus providing an overview of how they compare to Explain-ML while meeting the IML principles.

In sum, our main contributions include:

- A presentation of Explain-ML development process, illustrating our approach to an HCML process. This presentation includes an overview of the system developed and how it offers interpretability of RF-based models through mulitperspective views.
- An analysis of Explain-ML in light of the IML principles (Dudley and Kristensson, 2018), based on an analysis of the users' evaluation of the system.
- A reconstruction of the designer's meta-message for the three selected interpretability tools together with the identification of interpretability strategies offered by each tool, and their compliance with IML principles.
- A contrast between the results of the analysis performed with Explain-ML and the other tools. There are commonalities among the tools inspected with SIM, related to: interaction strategy, types of explored signs and common problems identified. We also observed that Explain-ML and the other three tools presented several strategies in line with the IML principles.
- A discussion regarding how our analyses corroborate the relevance of the principles, helping to consolidate them in HCML realm.

The analysis of Explain-ML, carried out with users, under the perspective of IML principles showed the compliance of

Explain-ML with them. The works that describe RuleMatrix, Explainer Explorer and ATMSeer do not explicitly mention the use of IML principles. The three tools were designed to serve non-ML specialist target audience. In this way, we observe that the attempts of designers to develop more appropriate tools for this audience are in line with the IML principles. And despite not being listed as an explicit objective, the three tools meet many of the principles fully or partially. This reiterates the importance of these principles.

This article is structured as follows. Section 2 discusses the concepts that support this work, organized into two subsections: subsection 2.1 presents Interactive Machine Learning and Human Centered Machine Learning concepts, including the IML principles that will be used as guide to our analysis; and subsection 2.2 describes the Semiotic Inspection Method adopted in our analysis. Section 3 presents close related work. Section 4 presents the development process and strategies adopted by our tool – Explain-ML – as well as its interpretability approach. Section 5 presents the adopted methodology. Section 6 presents the analysis of Explain-ML, carried out with users, under the perspective of IML principles and the analysis of three interpretability tools through SIM. Section 7 discusses our results while Section 8 presents some limitations of our work. Finally, we conclude in the Section 9, with glimpses at future work.

2 Background

In this section we present concepts that support this work. First, we address the HCML and IML Systems Design. Then, we address The Semiotic Inspection Method (SIM), used in this work to inspect and analyze three IML tools.

2.1 Interactive Machine Learning and Human Centered Machine Learning

HCML (*Human Centered Machine Learning*) research area has recently emerged based on an identified demand for considering explicit user needs for interpretability and decision making assistance from a more human-centered perspective (Ramos et al., 2019; Gillies et al., 2016; Fiebrink and Gillies, 2018). In this context of HCML and IML, (Dudley and Kristensson, 2018) present a review and characterization of Interactive Machine Learning research. The authors describe the generalization of a structural and behavioral model for IML systems and discuss principles to guide the construction of more effective (explainable) interfaces. Our work relies on these principles (which will be detailed in section 2.1.1) to discuss the results of the user-led assessment of their experience with our tool.

Interactive Machine Learning is directly related to Human-Computer Interaction as it puts human interactions into perspective. It was introduced by Fails and Olsen Jr (2003) and treats the model training process as an HCI task, receiving user input in the process of selecting, creating and labeling instances. A user more familiar with ML models may be required for model deployment, but is not essential in the training process. IML differs from classical ML as it considers user participation through an iterative process of modifying and revising the model during its training, usually in small steps. In the more traditional ML, the user usually performs a complete pre-selection of training data and significant changes at each execution of the model are performed (Dudley and Kristensson, 2018; Fails and Olsen Jr, 2003).

2.1.1 Principles of IML Systems Design

Considering the context where ML systems are increasingly used by non-ML users, together with the fact that several systems in this area try to increase the user's capabilities when dealing with the system, Dudley and Kristensson (2018) discuss the design context of IML systems, addressing key challenges and involved elements.

For the authors, the main **challenges** include: (i) users may be imprecise or inconsistent, implying some kind of bias in the model training process; (ii) there is a certain level of uncertainty between intent and user input; (iii) the interaction with a ML model is different from the interaction with a traditional system, since the evolution of a ML model is not always intuitive; and finally; (iv) training a model is not an exact task and may remain open, as training a model with 100 % accuracy may be impossible. From a structural point of view, an IML system is composed of four **elements**: (i) user; (ii) model; (iii) data; and (iv) interface; which should guide the designer in the design of the system. From a behavioral point of view, the interactive process of building an ML model can be subdivided into **subtasks**: (i) selection of *features*, (ii) selection of models, (iii) targeting the model , (iv) quality assessment, (v) completion assessment; and (vi) implementation.

Discussion of the design space of this type of system fosters design considerations that contribute to more efficient and productive IML systems, aimed also at non-ML specialists. In this way, (Dudley and Kristensson, 2018) identify **six principles** for the design of IML interfaces. Later on, we will present an analysis of the outcome of the evaluation of the Explain-ML tool with users in the light of these six principles.

Principle 1 - Make task goals and constraints explicit: the refinement of an ML model consists of an iterative process (optimization of parameters, training, evaluation, model adjustments, re-execution) and demands a significant role from the user. An IML system must clearly establish the objectives, as well as the restrictions, of each task, especially for non-expert users. This principle is related to challenges (i) and (iv) (subsection 2.1.1): users may be imprecise or inconsistent; and training a model may remain open.

Principle 2 - Support user understanding of model uncertainty and confidence: uncertainty and trust are inherent elements of ML models and, therefore, also inherent to IML systems. The user must be aware of this, so that he can manage his expectations regarding the system and its results. This principle is related to challenge (iii) (subsection 2.1.1): interacting with a ML model is different from interacting with a traditional system.

Principle 3 - Capture intent instead of input: this principle is related to challenge (ii) (subsection 2.1.1): There is a certain level of uncertainty between intent and the user input. In particular, in the context of IML, uncertain user input can

be quite damaging to the ML model training process.

Principle 4 - Provide Effective Data Representations: providing effective IML interfaces stimulates user perception of the tool and models. The analysis of the result at the instance level contributes to the understanding of the behavior of the model. Furthermore, interfaces designed to interact with complex data, through some kind of data simplification, increase the user cognitive capacity.

Principle 5 - Explore interactivity and promote rich interactions: the development of ML models can be more efficient if the system offers richer forms of interaction so that the users can express their intentions and *insights* through inputs to the system. IML systems should also provide ways for the user to reverse actions and contribute value to their entries.

Principle 6 - Engage the use: the system must provide the user with elements to monitor the tasks being performed, but in a way that does not overload their perception. This helps keeping the user motivated when interacting with the system.

It is worth noting that Dudley and Kristensson (2018) proposed these IML design principles focused on non-expert users. However, in our work the intended user is knowledgeable in ML models. Nonetheless, as the principles focus on overall principles that would allow users to interact (e.g. guaranteeing that users can express their intention or are not overloaded) we argue that they are equally relevant to our intended audience.

2.2 The Semiotic Inspection Method (SIM)

The *Semiotic Inspection Method* (De Souza et al., 2006; De Souza and Leitão, 2009) is an inspection method grounded in Semiotic Engineering theory (De Souza, 2005) that aims to identify the designers' intentions and principles that are communicated through the system interfaces.

Semiotic Engineering is an explanatory HCI theory that considers a system's interface as a meta-message from designers to users, that can be structured by a meta-communication template: "*Here is my understanding of who you are, what I've learned you want or need to do, in which preferred ways, and why. This is the system that I have therefore designed for you, and this is the way you can or should use it in order to fulfill a range of purposes that fall within this vision.*" (De Souza, 2005).

SIM is a qualitative and interpretative inspection method that aims to asses the communicability of an interactive system by inspecting the meta-message sent by the designer to the users, segmenting the interface into metalinguistic, static and dynamic signs[3] that are analyzed separately (De Souza et al., 2006; De Souza and Leitão, 2009).

A metalinguistic signs refer to other signs in the system interface, like help systems and tooltips. A static signs represent a state of the system that that does not depend on causal or temporal relations, like the status of a button on the screen. Dynamic signs represent the systems behavior, and depends on causal and temporal relations, like, for example, a new window opening after the user presses a button.

[3]A sign is anything that means something to someone (Hall et al., 2006)

In order to apply the SIM method, the following steps need to be taken:

1. Analysis of the interface's metalinguistic signs and reconstruction of the meta-message transmitted by the designer related to this sign.
2. Analysis of the interface's static signs and reconstruction of the meta-message transmitted by the designer related to this sign.
3. Analysis of the interface's dynamic signs and reconstruction of the meta-message transmitted by the designer related to this sign.
4. Compare and contrast the meta-messages conveyed by the three sign categories, in order to identify eventual inconsistencies and generate a unified meta-message.
5. Evaluate the system, after define a scope of inspection and scenarios to guide the evaluation.

SIM aims at identifying the designers' intentions and principles that are communicated through the system's interface. Nonetheless, other works have shown that during the analysis, evaluators may focus on specific aspects of the meta-message being conveyed (e.g. (Pereira et al., 2017)). In this work, we have used the IML principles as "inspection lens", in order to highlight how these principles were being addressed in the designer´s meta-message for the three analyzed IML tools.

3 Related Work

Guidotti et al. (2018b) has proposed that 'black box explanation' approaches, can be classified according to the scope of work in three dimensions: outcome explanation - OE, model inspection - MI and model building - MB. In Table 1 we present a summary of the classification of existing studies that present model-agnostic approaches according to these dimensions.

As our focus is on IML tools, we have also indicated in the last column of Table 1 an indication whether the described approach is interactive or not. Notice that only 10 out of the 24 related studies depicted in Table 1 describe interactive systems, most of them focused on a specific point in the life cycle of a Machine Learning model: building, debugging or interpreting. Finally, we have included Explain-ML (presented in section 4) as the last line of the table in order to compare it with the others.

We have also investigated other studies aimed at analyzing IML approaches. To the best of our knowledge other studies have not conducted an analysis based on the IML principles, as presented in this work. Most of them have analyzed or classified how other studies have considered IML. Wondimu et al. (2022), for instance, categorized IML studies according to (i) functional aspects related to the ML scope, such as: Visual Analytics, Searching, Explainability, HCI Interface Design; or (ii) the application area, such as Health, Education, Agriculture. Mosqueira-Rey et al. (2022) carried out a recent review of tools for developing and interacting with ML systems, performing a classification according to the development life cycle of an ML system. Both studies do not

perform individual analysis of the tools from the perspective of the IML principles nor any systematic analysis of the tools.

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); **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. *The tools whose names are in bold are evaluated in this work.*

Name	Reference	Year	Scope of work	Interactive
VIN	(Hooker, 2004)	2004	MI	
VEC	(Cortez and Embrechts, 2011)	2011	MI	
–	(Erik and Kononenko, 2010)	2010	OE	
ICE	(Goldstein et al., 2015)	2015	MI	
MFI	(Vidovic et al., 2016)	2016	OE	
MES	(Turner, 2016)	2016	OE	
Lime	(Ribeiro et al., 2016)	2016	OE	
–	(Singh et al., 2016)	2016	OE	
Prospector	(Krause et al., 2016b,a)	2016	MI	x
Explanation Explorer	**(Krause et al., 2017)**	**2017**	**MI**	**x**
OPIA	(Adebayo and Kagal, 2016)	2016	MI	
–	(Han et al., 2016)	2016	MB	x
Clustrophile	(Demiralp, 2016)	2016	MB	x
TensorFlow Playground	(Smilkov et al., 2016)	2016	MB	x
SHAP	(Lundberg and Lee, 2017)	2017	OE	
Anchors	(Ribeiro et al., 2018)	2018	OE	
LORE	(Guidotti et al., 2018a)	2018	OE	
BlackBoxAuditing	(Adler et al., 2018)	2018	MI	
–	(Krause et al., 2018)	2018	MI	x
ATMSeer	**(Wang et al., 2019)**	**2019**	**MB**	**x**
Manifold	(Zhang et al., 2019)	2019	MI	x
RuleMatrix	**(Ming et al., 2019)**	**2019**	**MI,OE**	**x**
IForest	(Zhao et al., 2019)	2019	OE	x
Explainable Matrix	(Neto and Paulovich, 2021)	2021	MI,OE	
Explain-ML	**(Lopes, 2020)**	**2021**	**MI, OE, MB**	**x**

Finally, we would like to point out that it is not the purpose of this work to present a comprehensive review of interpretability approaches, which has already been done in (Dudley and Kristensson, 2018; Guidotti et al., 2018b; Linardatos et al., 2021). Our focus is on contrasting Explain-ML with other existing approaches in light of the IML principles (Dudley and Kristensson, 2018), and discuss the principles themselves.

4 Explain-ML

In this section we present the Explain-ML tool and its development process. Explain-ML was developed as a proof of concept of an approach of the same name, proposed by (Lopes, 2020).

As shown in Table 1, the tool has multiple scopes of work: MB (Model Building) MI (Model Inspection) and OE (Outcome Explanation). In more details, with Explain-ML, users can perform tasks related to the lifecycle of a ML model: model training, predictions, model evaluation, model tuning, and re-execution. The tool helps users to build models interactively, without the need for source code manipulations. Users can execute the model several times, making adjustments. For each execution, information about the overall model, training dataset, and instance-level information is 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.

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 Forest (as ML model) applied to automatic text classification (ATC) as an ML task. The instantiated version employs the *Scikit-Learn*[6] implementation of the Random Forests Model. The initial design of Explain-ML focused on demonstrating our hypotheses that a multi-view and multi-perspective visualization approach for interpreting results could aid users to better interpret the results of a ML model.

Notice that, although, in theory, Explain-ML could be used with any ML model, initially it was instantiated for Random Forests (RF). The rationale behind this decision was to analyze the results for RF models, which can be considered more interpretable, but yet represent very effective text classification model (Cunha et al., 2021), before tackling other, more challenging models.

4.1 Designing EXPLAIN-ML

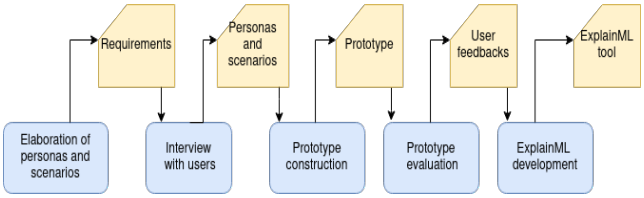

Figure 1. Explain-ML Development Process

The design process adopted to develop Explain-ML was a user-centric design, which allowed us to consider users' views and perspectives on what aspects would be relevant in an ML explainability tool. Thus, our design process included (i) Persona and Scenarios definition to guide the design process, (ii) interviews with users to better define our requirements, (iii) development of a prototype to represent our requirements-based solution and a (iv) evaluation of this prototype to better guide the (v) tool development.

4.1.1 Target User Description

The target users of our work are those somewhat familiar with ML models, acquired through the use of ML models in the past (e.g., through a course), which we call here as *knowledgeable user*. 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. It may include data scientists and domain experts who use Machine Learning for analysis, decision making, or research. With this profile in mind, we generated the persona[7] from a knowledgeable user representing the target users of the proposed tool (Figure 2).

[4] `https://www.djangoproject.com/`
[5] `https://www.sqlite.org/index.html`
[6] `https://scikit-learn.org/stable/`
[7] "Persona is the description of a fictitious user, based on research data from the user, capturing in detail the user of the system to be designed for which the designers will guide the design process." (Preece et al., 2019b)
[8] `https://xtensio.com/`

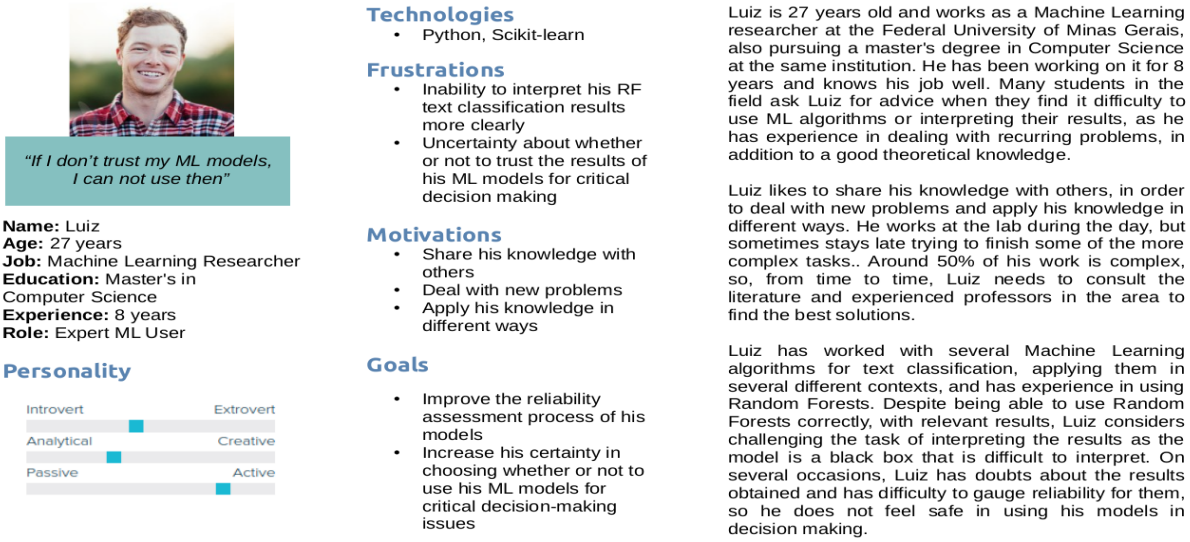

Figure 2. Persona: User with knowledge of ML models - Luiz. *(Generated through the Xtensio[8] platform.)*

4.1.2 Requirements Gathering

We conducted semi-structured (Lazar et al., 2017) interviews with target users in order to better understand the process followed by them (e.g. pre-processing, parameter tuning, analysis of results) while employing ML models in their applications. As a result, we have identified some important requirements for understanding and using Machine Learning results in relation to model explainability.

In preparation for the interviews, we conducted a pilot study to review our topic guide and made minor adjustments. Recruitment was based on email and social media invitations targeted at ML researchers. We recruited, from the target group, people who responded motivated to participate in the interviews. The interviews were conducted through videoconference and had 7 participants. The participants were all male Computer Science students (undergraduate or PhD). All of them had ML experience and use ML models on a daily basis. Before the interviews, all participants received the Free and Informed Consent Term explaining the study, and were free to ask any questions about the study.

The interviews were organized into six main blocks: (i) demographic data of the participants; (ii) contextualization regarding the objective of the interview and the scope of this work; (iii) the participant's knowledge about the specific ML Model we explore (i.e., Random Forests). The interview explored the user experience with ML in general and then focused on experience and analysis using Random Forests, as one of our goals was to test an instantiation of the tool with a specific classifier; (iv) the process participants adopted to analyze ML models and their results; (v) participants' opinions about a specific interpretability tool - LIME (Ribeiro et al., 2016); (vi) any other comments that the participants considered worthy of mention regarding the topics covered in the interview. The interviews were transcribed and analyzed (Lazar et al., 2017).

In our analysis of the interviews, our objective was to understand the participants' strategy for creating ML models and what would be the aspects in which an explainability tool could or should support them. Thus, to define the requirements of our explainability tool, we looked for strategies that were commonly adopted by the participants, as well as differences in how they worked to build their models.

We identified factors relevant to participants' experiences with the ML models based on the interviews, which we classified into five categories: (i) integration with tools, already used by the participants; (ii) strategies to measure the reliability of results; (iii) use of reference values utilized by participants to verify the quality of the trained model; (iv) parameterization strategies; and (v) experience with available explainability resources. Based on the identified factors, the following design decisions were made regarding the tool being developed:

1. Integration with Scikit-Learn (category i): We decided to use Scikit-Learn in our tool, since all participants declared to use it and stated that the integration with it would be very advantageous for an explainability tool.
2. Support for cross validation (categories ii and iv): Users reported its use to verify the variability of results and assess their confidence level in the trained model (ii). The use of cross-validation as part of the strategy for hyperparameter adjustment (iv) has also been reported.
3. Hyperparameter tuning support (categories iii and iv): The tool shall support hyperparameter tuning, providing an explanation of each available hyperparameter. It was proposed to use reference values for the hyperparameters to help in the parameterization step.
4. Display of evaluation metrics (category ii): Users pointed to evaluation metrics used to assess the performance (effectiveness) of the ML model.
5. Model Features Importance Display (category ii): Displaying the importance of features helps in understanding the model outputs.
6. Feature Importance Display per instance (category ii): Displaying the importance of features helps to understand the model classifications for each instance.
7. Visualization of execution history (categories ii and v): Participants mentioned their difficulty in assessing the reliability of the model due to the challenge of

comparing different results. When asked about tools they used for ML, participants highlighted the fact that they could not compare the results of model executions, which would be useful to assess reliability.

From the design decisions, we generate a prototype to validate, with the users, the decisions and requirements.

4.1.3 Requirements Validation

At this stage our goal was to generate a prototype that would allow us to conduct a formative evaluation of the prototype with target users to validate our proposed requirements and design. For this, based on the defined persona, we generated usage scenarios.(Madsen and Nielsen, 2010) for shared understandings and design ideas. We then developed a prototype to present our proposed views.

We described three usage scenarios[9] (Interpret Model Results Scenario, Hyperparameter Definition Scenario, and History Tracing Scenario), based on the needs highlighted during the interviews. From the described requirements and scenarios, a low-fidelity prototype was developed to present our proposed visualizations to target users and perform a formative evaluation of the tool (Preece et al., 2019b). The prototype was developed using Balsamiq Wireframing Tool[10] and presented the structure and content of the interface, allowing users to interact with it. Users could navigate through interfaces and views (as shown in Figure 3), simulating an execution.

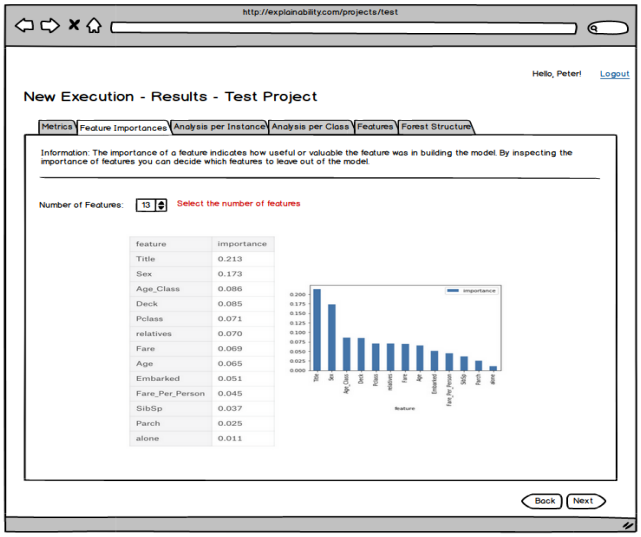

Figure 3. Prototype: visualization of the importance of the features. Results for the new execution

In order to collect *feedback* about our proposed tool represented through the Balsamiq Prototype, we carried out a formative evaluation with 3 participants (2 men and 1 woman). The participants were all Computer Science students (one Master's student, one Ph.D candidate, and one Ph.D.) who use ML models daily as part of their research. Recruitment was based on email and social media interactions directed at Machine Learning researchers in our department.

In preparation for the evaluation, we conducted a pilot study to review our topic guide and small adjustments were made to the roadmap. Before the evaluations, all participants received the Free and Informed Consent Form, and were free to ask any questions about the study.

In the evaluation session, participants interacted with the tool prototype as if they were creating an ML model. They were asked to (i) create a new project, (ii) access the created project; (iii) create a new execution for the created project; (iv) analyze the results of the new execution; and (v) analyze the history of project results. Users were guided and observed while performing the predetermined tasks and encouraged to think aloud (Preece et al., 2019b), about their interactive experience, possible difficulties and reflections on the prototype.

User's interaction (user video and prototype interaction) and audio from all sessions were recorded, transcribed and analyzed. Participants liked the solution presented by the prototype and said that they would use this tool in their ML modeling, confirming the *feedback* of the interview participants. As a result of the evaluation, the participants raised some problems, improvements and suggestions related to the explainability model, as well as the interface. These results allowed us to review some of the initial ideas and improve them in order to implement the real tool.

4.2 The Explain-ML Multiperspective Approach to Explainability

Explain-ML[11] was designed to implement a workflow in which the user can interactively carry out the steps of the life cycle of an ML model: definition and optimization of hyperparameters, model training, testing, evaluation, adjustments to refine the model and re-execution of the model. Users have an access area, in which they maintain different projects where one or more executions are created. An execution covers the previously mentioned stages of the model lifecycle.To do so, users must perform three steps. In the first step, users provide the input dataset and information such as number of *folds* for partitioning, data format and cross-validation. Then, users select options related to whether or not to optimize the parameters, type of optimization and possible ranges of values for the parameters of the model. In the last step, they can see the defined (or optimized) parameters, make adjustments to the parameters (if desired) and proceed to training and generating the visualizations.

For each execution, the approach presents a set of views that conveys aspects related to the model, the model training dataset and specific information about instances. These views act as explanations for the model. The set of views was designed to provide different perspectives that complement each other (global, dataset and local), which help the user to interpret the results of the model. The following briefly describes the views provided by Explain-ML:

[9] A scenario "is an informal narrative description of human activities or tasks in a story that allows exploring a discussion of contexts, needs and requirements" (Preece et al., 2019b). The scenario can be used at several steps in the (Rosson and Carroll, 2002) design process.)

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

[11] 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`.

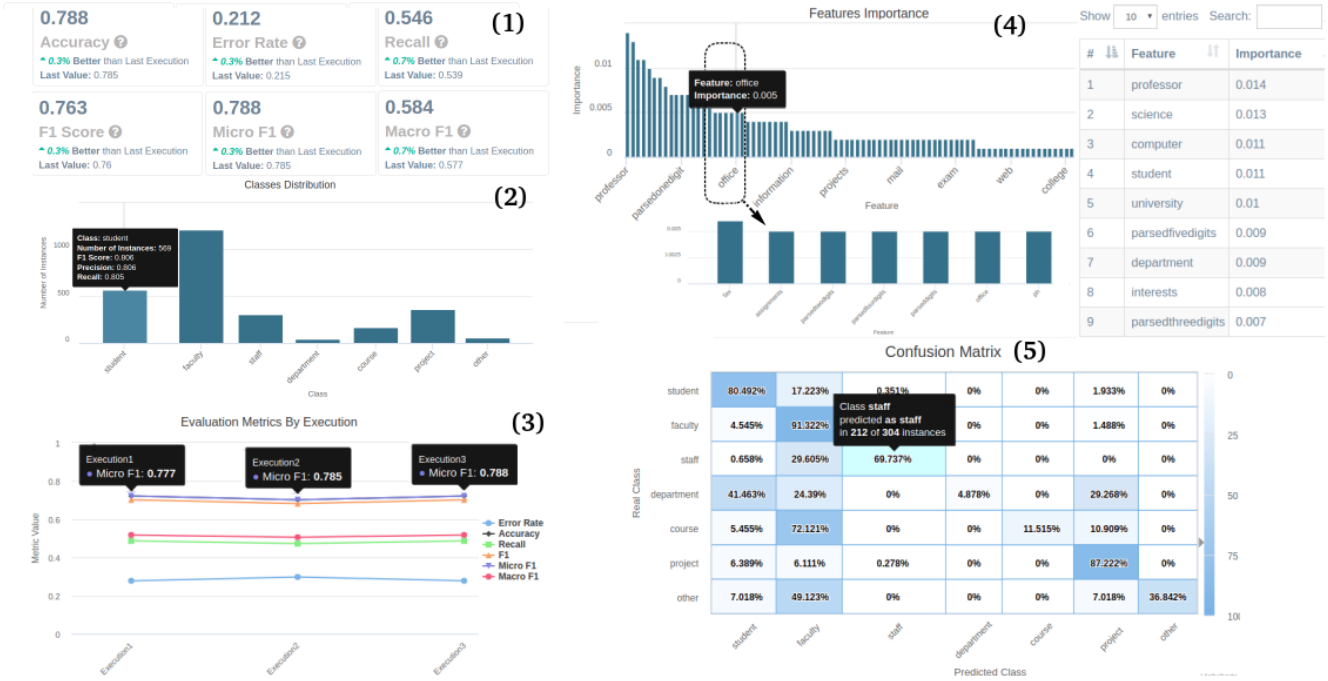

Figure 4. Available Views in the Explain-ML approach: (1) Evaluation metrics, (2) Class distribution, (3) Project execution history, (4) Feature Importances, (5) Analysis by class.

- **Evaluation Metrics:** This view conveys a model perspective, presenting the metrics *Accuracy, Error Rate, Recall, F1-Score, Micro-F1* and *Macro-F1*. The value of each metric is presented along with an explanation of the metric and its value relative to the previous run (it is possible to make adjustments to the model and generate new runs). This view allows the user to have an overview about the performance of the model and its improvement/deterioration in relation to the previous execution (Figure 4 - (1)) [12].
- **Class Distribution:** This view conveys a perspective on the dataset used in training the model. The graph shows the distribution of instances by class, and some information inherent to each class such as name, number of instances, and metrics such as F1, Precision and Class Revocation. This view allows the user to understand which are the main classes and other characteristics that can introduce bias in the model or even make it more difficult (or easier) to classify instances of a certain class (Figure 4 - (2)).
- **Project Execution History:** As stated earlier, the proposed approach allows a project to contain several model executions. This view presents a graph with the history of the model's performance metrics (*Accuracy, Error Rate, Recall, F1-Score, Micro-F1* and *Macro-F1*) in the different runs. Furthermore, the user can visualize the model's prediction behavior in different executions, for a certain selected class (Figure 4 - (3)).
- **Feature Importance:** This view conveys a perspective of the model. Additional information about the importance of features[14] include the total number of features, the lowest and highest importance value, number of features with zero importance value and with a value greater than zero. The visualization also adds an interactive graph with the importance of features which are also presented by means of a list in which the user can search for specific features. Particularly, in textual bases with high dimensionality, our current focus, these observations can be quite significant, allowing the user to focus on the most relevant features that help in model interpretation (Figure 4 - (4)).
- **Analysis by class:** This view covers aspects related to model predictions for real instances and their classes. It allows the user to understand how the model behaves for each class and if it is more successful in predicting certain classes. It is based on a confusion matrix that provides an overview of how the model behaves in relation to each class. The visualization also provides a bar graph that allows the user to inspect the behavior of the model for a given specific real class, selected by the user (Figure 4 - (5)).
- **Instance Analysis**: This view has a local perspective, focused on instances. For each selected instance, the approach presents the actual and predicted classes, the probabilities of each class, the most important *features* of the instance, a word cloud with the features and a list of features through which the user can search for features and add or delete one or more features. Adding or removing features consists of adjustments to the model, which are considered in the next run (Figure 5).

The Explain-ML approach allows the entire ML process to be performed, from hyperparameter tuning to model im-

[12] The example shown in Figures 4 and 5 refers to the WebKB-Course[13].

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

[14] In the current instantiation of Explain-ML, the importance of features is obtained from the decrease in the impurity of the node in each Decision Tree of the Random Forest, weighted by the probability of reaching that node, but any other measure, such as information gain or chi-square could have been used.

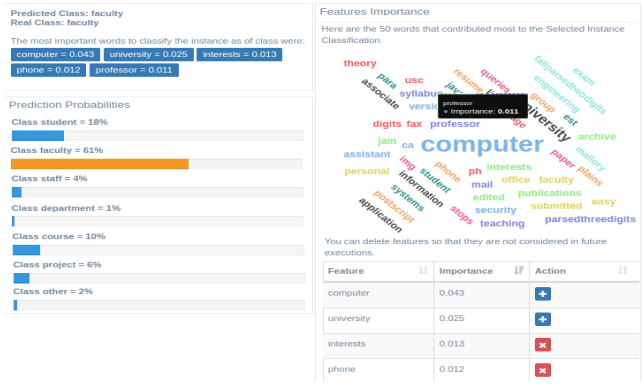

Figure 5. Instance-level views

provements, interactively and without coding. The approach provides complementary views that allow users to obtain interesting insights that drive feedbacks on model reruns and assist the user in the general interpretability of the model.

It is worth mentioning that although the current version of the tool only includes the Random Forests model, it can be considered mostly agnostic (independent of models), since the views of the aspects that involve the explanations are mostly applicable to several other models (e.g., dataset statistics, effectiveness metrics, importance of the features for the model and for the instances, confusion matrix, etc). For the configuration of other models, e.g. neural networks, the parameterization issue would be different[15].

5 Methodology

The first step of our study involved the evaluation of Explain-ML in light of the IML principles (Dudley and Kristensson, 2018) presented in subsection 2.1.1. In order to broaden our analysis, we also examined how other tools, also intended for ML interpretability, complied (or not) to these principles. We selected three IML interpretability tools and used the Semiotic Inspection Method (SIM) (De Souza et al., 2006; De Souza and Leitão, 2009), to analyze them. The choice of using SIM was based on the fact that it allowed us not only to perform a systematic analysis of the solution proposed by the tool, but also to take into consideration specifically if and how the IML principles were addressed in the tool.

Figure 6 depicts the steps of the qualitative analysis performed: (1) evaluation of Explain-ML in light of IML principles; (2) the selection of the other tools to be considered in the study; (3) the analysis of the selected interpretability tools using SIM; (4) the contrast of Explain-ML evaluation results and the tools evaluation by SIM. In the next subsections, steps 1 to 3 of the adopted methodology are presented. The contrast of the results is discussed in Section 7.

5.1 Explain-ML Evaluation

To evaluate Explain-ML we carried out a qualitative study that allowed us to analyze in depth the perspective and per-

[15]Firstly, we opted for the Random Forests model, which is naturally more interpretable. 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 existing visualizations.

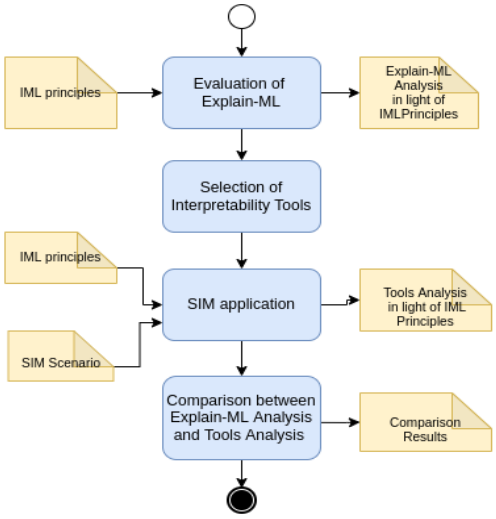

Figure 6. Overview of the methodology adopted.

ceptions of users about the tool. The adopted methodology combines user interaction with the tool and semi-structured interviews, with the aim of exploring how they perceive the visualizations that explain the model results. The study was conducted using the tool instantiated with the *Random Forest* model, applied to text classification.

As the tool was intended for users with some ML knowledge, the invitation to participate in the evaluation was sent to graduate students in Computer Science 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 their participation.

Among the people who responded, we were able to schedule six of them to participate in an evaluation session. The sessions were held in Portuguese in person in a (meeting) room at the University[16]. The Explain-ML - *Random Forest* version of the tool was deployed on a computer made available to users, and user interaction and audio from all sessions were recorded. All participants were male and had training in Computer Science. One of them was a postdoctoral fellow and the other 5 were doctoral students. All of them had experience with the Machine Learning models during graduate program, but the experience varied in terms of time and knowledge about the models.

During the evaluation conducted with users, we asked them to perform analysis on a specific classification task on a predefined dataset and its results. We guided participants to train a model and look at all the explanation views of the results generated by Explain-ML. Then they were asked to interact with the tool to improve the model and finally analyze the run history displayed in Explain-ML. During the interaction, the participants were encouraged to think out loud (Preece et al., 2019b) and at the end of each stage a short semi-structured interview was carried out about their opinions regarding the tool.

The WebKB[17] dataset was selected for our evaluation. It contains 8282 web pages classified into the following categories: student (1641 pages), faculty (1124), staff (137),

[16]When the objective of an evaluation is to generate *insights* about a system, a qualitative study with about 5 users is enough (Preece et al., 2019b).

[17]`http://www.webkb.org/`

department (182), course (930), project (504) and others (3764). This is a very interesting dataset from an analytical point of view, as it contains many classification challenges found in real applications: skewed (unbalanced) class distribution; non-trivial semantic overlap between classes; noise and ambiguity in the page text, etc. Our goal was analyze which of our Explain-ML views could help users to identify issues and devise strategies to improve models.

To contextualize the tasks for the participants, an assessment scenario (Carroll, 2000) was created. The scenario described a situation where users were participating in a research project that required automatic classification of their university pages. They were motivated to use Explain-ML with a similar dataset (WebKB) to assess whether it would be useful to apply it to their own context.

Subsequently, we present the steps used to guide participants in their interaction with Explain-ML. Initially, we asked participants about their experiences with ML and *Random Forest* in particular. We then introduced the participants to Explain-ML, describing its goal of explaining the model, and answered any questions they had about the tool. After participants had an overview of the tool, we guided their interaction through it. We describe the 4 main tasks that made up the interaction stages, as well as the semi-structured interview associated with each of them: (i) initial model training (T1) through hyperparameter tuning; (ii) analysis of views of the model´s results (T2); (iii) model improvements (T3) by means of the definition and execution of at least two changes in the model that they judged capable of improving it; and (iv) analysis and exploration of the results history (T4) in order to analyze the impact of the changes made in T3. We recorded the audios of the interviews.

For this paper, the focus of the analysis was on the IML principles. Thus, for each principle we describe how Explain-ML considered it as well as the results of the evaluation, indicating how users' perceived how the principle was addressed in the tool. The results of this analysis are presented in section 6.1.

5.2 ML Interpretability Tools

The selection of interpretability tools to contrast with the Explain-ML analysis was based on the literature review performed and presented in the table 1, in addition to three criteria:

- *Work Scope:* our selection focused on approaches related to the problem of 'black box explanation´ and that had one or more scopes of work: Outcome Explanation Problem (OE), Model Inspection Problem (MI), Model Building (MB);
- *Interactive*: considering the interactive nature of Explain-ML and the goal to analyze in light of the IML principles, we focused on tools and interfaces that also were interactive.
- *Availability for inspection*: several tools proposed in the literature are not available for use. In order to be able to perform the systematic evaluation of the tools through the Semiotic Inspection Method, only tools that could be installed/used or provided material that thoroughly presented it, including videos that allowed dynamic signs to be examined, were considered.

As a result of this analysis, we obtained a list of three tools (*RuleMatrix, Explanation Explorer and ATMSeers*) to be systematically analyzed. Note that the goal of the analyses was to identify the strategies used by designers of each tool to provide interpretability and their level of "compliance" with the IML principles defined by (Dudley and Kristensson, 2018) for the design of IML (Interactive Machine Learning) systems.

5.3 SIM Application

To perform the analysis of the selected tools, we used the *Semiotic Inspection Method* (De Souza et al., 2006; De Souza and Leitão, 2009), an inspection method based on Semiotic Engineering theory (De Souza, 2005) that aims to identify the designers' intentions and principles that are communicated through a system's interface.

In this article, we analyze the meta-message conveyed by the three selected tools for ML interpretability – RuleMatrix, Explanation Explorer and ATMSeer –, focusing on identifying interpretability strategies used by their designers and analyzing their "compliance" with interactive principles. To do so, we have combined SIM with the principles for the design of IML systems (Dudley and Kristensson, 2018). This combination requires the evaluator to analyze and register, at each step of SIM, what the designer is conveying regarding each one of these principles.

SIM was carried out by one of the authors, who had academic and research experience with the method. One inspection scenario was created for the systems that would be analyzed, considering knowledgeable users that decided to explore the systems in order to obtain help in interpreting ML models results.

> *Scenario: Luiz is 27 years old and works as a Machine Learning researcher at an University, also pursuing a Masters Degree in Computer Science at the same institution. He has been working in this field for 8 years and is good at his job. Many students in the field ask Luiz for advice when they find themselves having difficulty using ML algorithms or interpreting their results, as he has experience in dealing with recurring problems, as well as a good theoretical framework. About 50% of his work is complex, so from time to time he needs to consult the literature and experienced professors in the field to find new solutions. Luiz has already worked with several ML algorithms, in several different contexts, but, despite being able to use them correctly and obtaining relevant results, he finds the task of interpreting ML results somewhat challenging, since the models can be black boxes difficult to interpret. Luiz constantly has doubts about the results obtained by his ML models and finds it difficult to assess their reliability, so he doesn't feel safe using them for critical decision making. Luiz became aware of systems that support the task of*

> *explaining Machine Learning models. Thus, he decided to use them to facilitate the assessment of the reliability of his models and to verify the usefulness of these systems in his context.*

The three selected systems (RuleMatrix, Explanation Explorer and ATMSeer) were inspected during February 2022. The SIM steps were followed, considering the user's perspective as that of a person who chooses to use the system for the purpose of interpreting ML models. All of the metalinguistic signs available as part of the help system, system documentation or within the system were inspected. As these were tools developed as part of scientific researches, all papers about the tools were also considered during the metalinguistic analysis. All the screens available to the users and the static and dynamic signs associated were examined. For each system, all the main aspects of the meta-message were taken note of, especially what was communicated regarding the principles for the design of IML systems. In sections 6.2, 6.3 and 6.4, we present the key points of the unified meta-message of each system and our analyses.

6 Results

In this section, we present the results of our analysis of how Explain-ML addresses the IML principles, based on the user evaluation of the system (subsection 6.1). We then present the results of our analysis of RuleMatrix (subsection 6.2), Explanation Explorer (subsection 6.3) and ATMSeer (subsection 6.4). For each of these systems, we present the main points of the consolidated meta-message generated through SIM, followed by our analysis of how they address each of the IML principles. The contrast of the results is discussed in Section 7.

6.1 Analysis of Explain-ML in Light of IML Principles

In this section, we present the results of the analysis of the Explain-ML tool in the context of IML principles. To do so, we present the analysis of Explain-ML considering each of these principles taking into account both the system design decisions and how they were received by users in the evaluation. We include participants quotes[18] that illustrate our analysis of their perspective.

Principle 1 - Make task goals and constraints explicit: Explain-ML offers an intuitive interface, where the construction of the model is carried out in a guided way, making it clear to users what must be done at each step. Options are offered for users to configure models and how parameters are optimized and defined. Views with various perspectives (global, local and dataset level) are also offered, acting as model explanations. Users can interact with the views and make adjustments to the model for future executions. The tool also helps users to build models interactively, without the need for source code manipulations.

[18]The evaluation was conducted in Portuguese, participants native language, and the quotes translated by authors.

The comments of participants 1 and 3 during the evaluation illustrate the relevance of complementary visualizations in order to guide them in the interpretation and improvement of the ML model:

> *"The ability to see the results by instance, and everything integrated, the word cloud, the graphs, the metrics, is not something I see [normally]."* (P1) *"The program was good for visualizing the data and proposing improvements."* (P3)

We were also able to observe that Explain-ML visualizations guided participants in carrying out tasks to achieve their goals or in identifying strategies to achieve them.

> *"I think the information I got here would allow me to refine my model much more easily than I would if I had to keep testing, training… Here I have a tool that helps guide me. " (P5)*

Explain-ML provides multiple perspectives of model execution, it does not direct users to any specific path to improve the model, but it does give users ample space for exploration. This gives users autonomy to make different decisions and conduct different investigations and analyses.

Principle 2 - Support user understanding of model uncertainty and confidence: Explain-ML displays model effectiveness metrics such *Accuracy, Error Rate, Recall, F1-Score, Micro-F1* along with metrics related to previous model execution (Figure 4 - (1)). An explanation is associated with each metric that can be viewed by the user on demand (indicated by the "?" icon after the metric). Users can view the model's execution history (Figure 4 - (3)), being able to better understand the uncertainty and confidence level of the model by analyzing the current and previous executions, focusing on performed adjustments and parameterizations.

Through the evaluation, we were able to find evidence that the various Explain-ML views allowed users to better understand the model (and system) they were interacting with. Each of the views addressed a relevant aspect regarding the global understanding of the model.

The **Evaluation metrics** visualization allowed users to assess their level of confidence in the model prediction, as highlighted by P4: *"The micro (F1) is an overall average and the macro (F1) is an average by classes… to get an idea, for example, we know that this base here is a little unbalanced, because he couldn't learn from some of the classes."* Visualizing the **Class Distribution**, in turn, allowed users to identify possible biases in the model, in order to help them determine which metrics should be better observed, as illustrated in the comments of P2 and P1: *"[...] talking about accuracy on an unbalanced basis doesn't solve anything..." (P2). "If the classes are unbalanced, you need to look at those metrics, micro and macro F1." (P1).*

The **Importance of features** view makes it possible for users to better understand model behavior: *"It seems that the model is cool because all these first features that are appearing here have everything to do with the idea from the dataset you passed at the beginning" (P5)*. This view also helped participants to identify possible noise in the classifier, despite

model's high accuracy: *"5-digit numbers are the sixth most important item in the model?! This is weird..." (P6)*. Finally, the **Analysis by Class** view allowed the identification of a bias in the model, as well as possible strategies to improve it: *"You have to see which features are most important for the department (class), because they are not correct." (P3)*.

Principle 3 - Capture intent instead of input: in Explain-ML, the optimization and definition of models are performed without the need for source code manipulations. The interface guides users' input through fields with pre-set ranges and offers help messages about the tool and ML concepts. As users can focus on the parameters, and not on how to code it, we can consider that it allows them to focus more on their intentions as they make changes, than on how to make those changes.

In the evaluation carried out, we found evidence that this principle is met:

> *"[...] other tools have this limitation: The guy will have to create a model, he will need to implement it in python, he will need to know how to manipulate... To use this here I don't need to know python... So it has a very interesting potential, I think its audience is wider." (P2)*

Principle 4 - Provide Effective Data Representations: Explain-ML provides views at the global (model), local (instance) and dataset levels. Visualizations include graphs, word clouds, listings, confusion matrix, among other elements. Thus, users access the characteristics and results of the models more intuitively, as complementary details among the different perspectives can be verified.

In the evaluation carried out, we noticed that the participants adopted strategies that corroborated the relevance of our multiperspective approach, highlighting the importance of complementary visualizations in obtaining a general/comprehensive view of the prediction model. P2's comments illustrate the effectiveness of Explain-ML representations in satisfying Principle 4:

> *"[...] especially when you have this very large volume of features, it is difficult for you to understand on your own [without using the tool] the results [of the ML model]." (P2)*

> *"I would really use this… I think its usefulness is enormous… I think that all the proposed views add a lot and they are quite complementary." (P2)*

Principle 5 - Explore interactivity and promote rich interactions: the possibility to exclude or add *features* for future executions of the model is in line with this principle. The model development stage includes broad user participation in terms of optimization and parameterization definition.

In the evaluation carried out, users were able to adopt strategies to improve the model. They were asked to indicate and make some improvements (at least two changes) that they identified during their analysis as potentially useful for improving the model. Participants adopted different strategies in an 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) hyperparameter change (P5).

> *"So here I could have several executions and check how much is varying according to the changes I make to my model… I want my error to always decrease and the accuracy or another metric to increase." (P2)*

At each interaction with Explain-ML, participants were able to obtain feedback on the effects of their actions on the model, in order to analyze the impact of each change:

> *"Cool, so I can see that deleting that feature was good for my model." (P2)*

> *"You could see the impact here, 0.1%, because I eliminated a few features." (P1)*

By observing the impact of their actions on the model, participants were able to determine whether their adopted strategy was functional or whether they should take a different course of action:

> *"It made the rating for the department worse… Removing that feature did not improve the rating for the department… If the objective was to improve the rating for the department, it would not be interesting to do that." (P3)*

> *"It started marking a few more documents as college, isn't that right?! It started to confuse more. Yes, now I understand [...]" (P6)*

Principle 6 - Engage the user: Explain-ML allows the user to train models in a guided way, showing the progress of the most time-consuming processes (optimization of parameters and training of models). The available visualizations are presented through a structure of tabs that organize the elements according to the perspectives to which they refer. The tabs allow the user to navigate through the views quickly and without information overload.

P5's comments illustrate the motivations in using the tool, in line with principle 6:

> *"I liked that, in addition to being useful, the system has this friendly face, this nice interface that your system has... A system can be very useful sometimes, but the tool may not have a nice look, it doesn't have this simplified usability... People stop using them... 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 pretty cool." (P5)*

> *"I lost a lot of information [referring to a previous analysis without the system] that I'm seeing here, that I could have had. And that's exactly what I wanted to see there, back then. If I had this framework that I have here, I think it would be much easier to explain the model I was generating." (P5)*

6.2 RuleMatrix Inspection

In order to inspect this RuleMatrix, we accessed the project available on Github[19] and generated an executable version for our inspection. In our analysis we focused on the interactive version of RuleMatrix (Ming et al., 2019) and on understanding its model behavior. We did not include in our evaluation scope the activities related to model training (which is done through programming).

6.2.1 Meta-message for RuleMatrix

In this sub-section, we present the meta-message from the designers of RuleMatrix (Figure 7) reconstructed through SIM.

"We (designers of Rulematrix) understand that you": are a domain expert with little knowledge of ML and who work with ML systems.

"Our view about what you want or need to do": You want to understand, explore and validate predictive models.

"How and why you want it to be done": You want an application, with a visual and interactive interface that allows you to explore the details of the decisions that gave rise to the model results, even if you do not have much knowledge about the ML model used.

"This is the system we (RuleMatrix designers) designed for you, and this is how you can or should use it in order to fulfill a range of purposes that fall within this vision."

The tool involves using substitution rules and matrix-style visualization. The rules are generated from the dataset and outputs of the original model, and try to reproduce the results generated by the original model. Each rule contemplates one or more logical condition composed of features and thresholds. The matrix presents each rule as a row and each feature as a column. Initially, one needs to perform some programming to use the available package, train the model, generate the rules and generate the interactive interface containing the views. The interactive interface itself is more aligned with the user profile, as it does not require programming or in-depth knowledge of ML.

The tool provides several interpretability strategies, incorporated in a single screen, without menus. All possible actions are visible on the screen. The main element consists of the matrix visualization which is divided into three parts (Figure 7 - (2)):

- *Dataflow*: which shows the data flow feeding the matrix and each rule. The user can inspect the amount of instances per class that are input to the matrix and the data flow that goes into each rule.
- *RuleMatrix*: which presents each rule as a row and each feature as a column. The user can: (i) check the rules by inspecting each row, (ii) analyze the features used in the rules by inspecting the columns of the matrix; (iii) see the constraint imposed on each feature through the gray box, (iv) expand it to see the distribution of instances according to their classes and (v) inspect the probability value of the classification, with the color of the resulting class. Some tooltips about matrix elements are available.
- *Support View*: which shows the fidelity of the rule expressing how faithful the rules are to the original model. It also shows the evidence of the rules' predictions through a horizontal bar showing the proportions of classified instances in each class and striped boxes to represent error predictions.

Besides the Rules matrix, the interface provides a control panel, through which one can adjust the style of some matrix elements, the settings of matrix output conditions and details, as well as the rules filters (Figure 7 - (1)). A data filter panel (Figure 7 - (3)) is also available and allows users to (i) filter input instances being considered by the matrix, adjusting value ranges of the instances features; and (ii) set instance-specific values to be classified according to the rules. RuleMatrix also allows one to select the dataset used in the matrix among the options: train, test, sample train and sample test. And, finally, the tool presents a table with the raw data of instances (Figure 7 - (4)).

RuleMatrix's designers include a few metalinguistic signs, focused only on the elements of the rules matrix and what they represent, in the format of a help accessed by demand. The exploration strategy they offer users is mostly based on a trial and error in a direct manipulation interface. The tool does not offer tooltips nor a help system to clarify ML concepts considering non-ML expert users. In this case, if users do not have experience in this type of interface or are not interested in exploring it, they may not understand some of the available features. This may lead users not to perceive or use of all available features, thus compromising the understanding of the visualizations made available for interpretability.

Another drawback of the interface design is that it is not very clear to users the effect of some of the controls. Thus, one will need to explore these controls and observe their effects on the matrix to infer their purpose. However, this might not be an easy (or even feasible) task for users who have little experience with ML.

Considerations. The tool is well organized, although the designers' meta-message presents some inconsistencies, considering the few metalinguistic signs and the profile of the target audience. The designers explore the dynamic signs extensively to communicate their meta-message to the user. The main style of interaction used is direct manipulation. All interface items can be explored through mouse clicks, to use controls that mostly consist of range sliders and checkboxes. With no metalinguistic signs about the screen controls, the effect of using them can be perceived in the matrix, only after their use. So, for experienced users willing to explore the interface, the meta-message will likely be transmitted successfully. However, the challenge of this strategy is that if users are not experienced in ML, they may not perceive the details related to the behavior of the rules and the original model, which are precisely what must be perceived in order to be able to interpret the system.

[19] RuleMatrix: available on https://github.com/rulematrix/rule-matrix-py

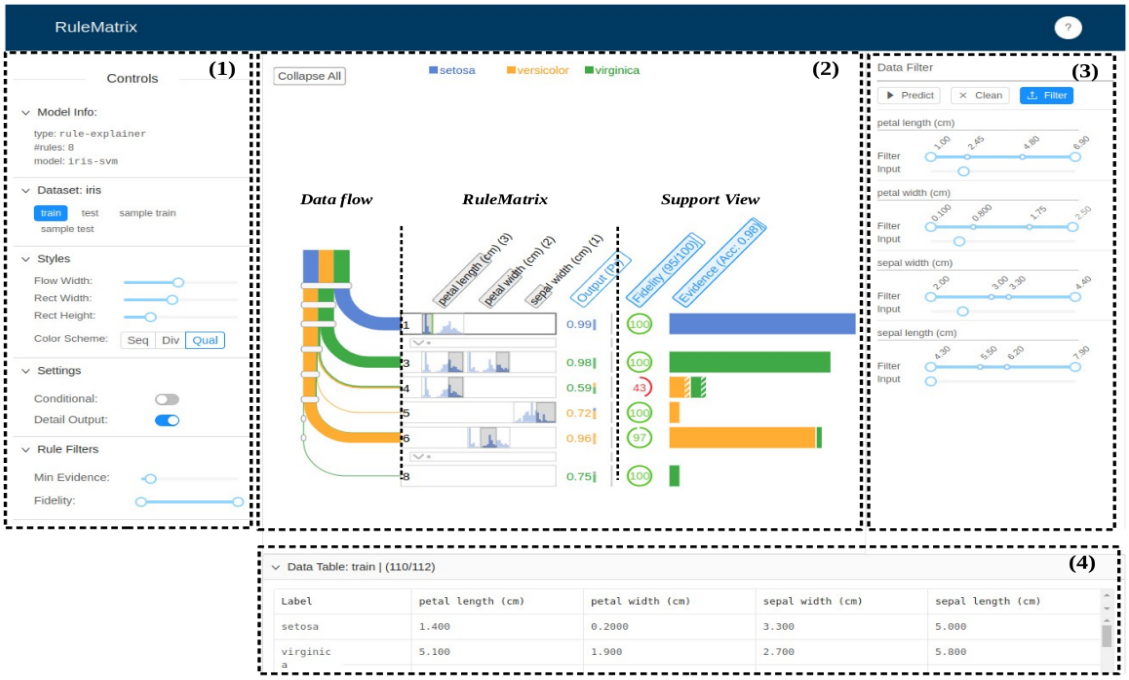

Figure 7. RuleMatrix interactive interface.

6.2.2 Analysis of the RuleMatrix in Light of the IML Principles

Principle 1 - Make task goals and constraints explicit: RuleMatrix *partially complies* with this principle. On one hand the limited possibility of the interaction is well communicated through the set of elements used in the interface. On the other hand, the effects of the available controls are not clear to users, who have to explore their effects to make sense of them.

Principle2 - Support user understanding of model uncertainty and confidence: RuleMatrix allows users to perceive the uncertainty associated with the rule regarding the selected dataset portion. Fidelity (i.e. expresses the error between the rule and the model) is also presented, as well as the evidence of the original model on the real data, in addition to a tooltip showing the model's accuracy. However, the tool does not present common global metrics such as *Accuracy, Error Rate, Recall, F1-Score, Micro-F1* and *Macro-F1*; related to the global behavior of the system on the dataset submitted to the rules matrix. Thus, we have considered that it *partially complies* with principle 2 as it focuses only on a local perspective, neglecting the global and dataset ones.

Principle 3 - Capture intent instead of input: The tool offers a certain degree of flexibility for users to provide their input, while avoiding leaving users free enough to provide uncertain inputs that generally exist between intent and user-provided input. Thus, the tool *complies* with principle 3.

Principle 4 - Provide Effective Data Representations: In RuleMatrix the matrix consists of the available view. However in some settings it is possible to adapt which/how some elements of the matrix are seen. The tool allows simulating the classification of an instance according to the rules,thus, the involved rules can be considered an instance-level explanation. A table with raw data is also displayed at the bottom of the screen. However, all these elements may not be so helpful in cases with high dimensionality datasets (e.g., textual). Thus, the tool *partially complies* with principle 4.

Principle 5 - Explore interactivity and promote rich interactions: RuleMatrix allows users to express their intentions and insights through inputs to the system. Although it is not possible to feed the model through the interactive interface, the user can vary the inputs and follow the behavior of the rules matrix, simulating behaviors. Thus, it *complies* with principle 5.

Principle 6 - Engage the user: RuleMatrix *complies* with this principle as it allows users to interact with the tool, causing effects that can be undone or reset. These features encourage users to explore the tool and engage them in the objective of understanding the behavior of the model.

6.3 Explanation Explorer Inspection

Explanation Explorer is intended to be a visual analytic workflow to help domain experts and data scientists in the activities of exploring, diagnosing, and understanding the decisions made by a binary classifier (Krause et al., 2017). In order to inspect this tool, we accessed the project available on Github[20] and generated an executable version of the system.

Next, we present the main parts of the consolidated meta-message from the Explanation Explorer designers (Figure 8), reconstructed through SIM. This is followed by the analysis of the tool design in light of the IML Principles.

6.3.1 Meta-message for *Explanation Explorer*

"We (Explanation Explorer designers) understand that you": are a data scientist, a domain expert or both at the same time,

[20] Explanation Explorer: available on `https://github.com/nyuvis/explanation_explorer`

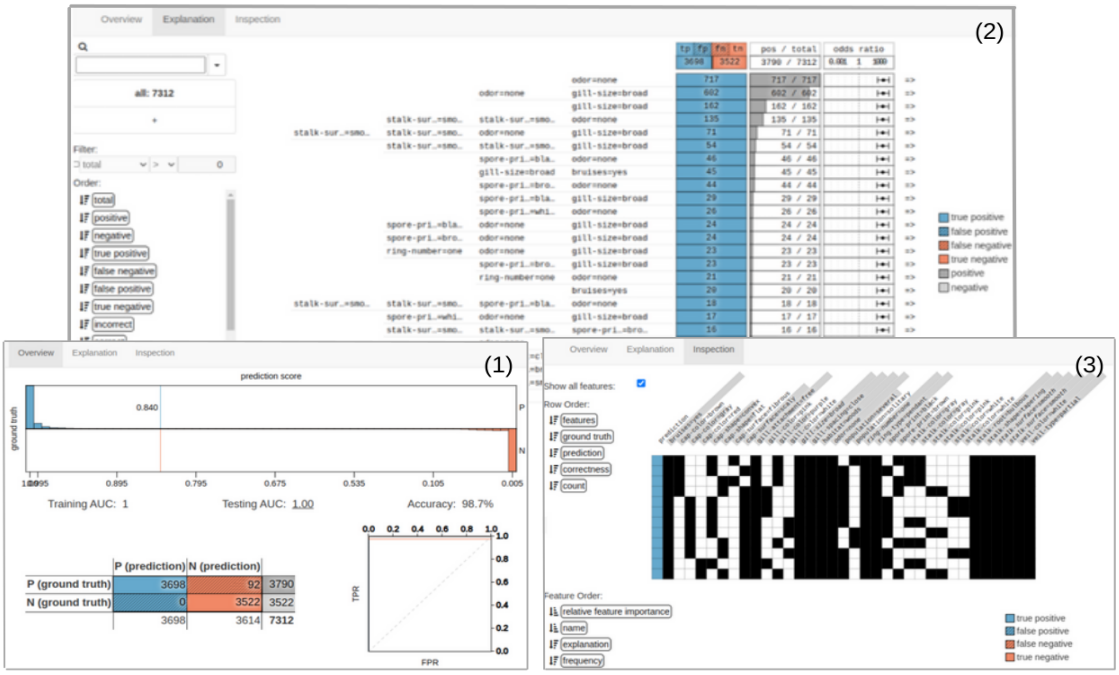

Figure 8. Explanation Explorer: (1) Overview panel, (2) Exploration panel, (3) Inspection panel.

and have experience with interactive software systems.

"Our view about what you want or need to do": You want to understand the behavior of a binary classifier.

"How and why you want it to be done": You want a system, with a visual and interactive interface that allows you to explore, diagnose, and understand the decisions behind the results of a binary classifier.

"This is the system we (Explanation Explorer designers) designed for you, and this is how you can or should use it in order to fulfill a range of purposes that fall within this vision."

Explanation Explorer does not include menus, so all available actions are visible on the screen. The user can navigate through three panels:

- *Overview*: presents views related to statistical summaries of the overall model performance. Users can inspect histograms showing the distribution of prediction scores. The direction of the bars indicates the ground truth and the position relative to the threshold line indicates the predicted label. When one hovers the mouse over the top precision score graph, it is possible to visualize corresponding changes in the graphs at the bottom-right. Users can also check the confusion matrix which shows the number of correct and incorrect predictions. The ROC (*Receiver Operating Characteristics*) curve shows the prediction quality (Figure 8 - (1)).
- *Explanations*: focused on providing an overview of the decisions made by the classifier, through the computed explanations and their precision. Users can explore explanations consisting of a set of features. Each row shows a group of data items explained by a set of features. Explanations longer than three features are flagged and can be fully visualized through tooltips. One can inspect three columns: the first one shows the distribution of true/false and positive/negative data items within the group, through label colors and hatching patterns indicating incorrect predictions. The second column shows the number of items captured by the explanation. The third column shows the odds ratio of the group on a logarithmic scale, with whiskers for the confidence interval. It is possible to select explanations (lines) that are positioned first. One can navigate to the item Level Inspector through the arrows on the right. Users can also use the controls on the left to filter and order data items according to various aspects. One can also type a feature to obtain explanations that contain that feature (Figure 8 - (2)).
- *Inspection*: focused on presenting user-selected explanation and the instances it explains. Users can view a matrix with data items as rows and features as columns for the explanations. Rows gather identical instances and count them on the left side. Features are sorted by their relative feature importance, and show how labels can be separated from left to right. It is possible to expand the matrix to show all features, or only those used in the explanation. Finally, users can sort rows and features according to some specific parameters (Figure 8 - (3)).

Considerations. The system offers practically no metalinguistic signs on the screen to clarify the meaning of the elements such as graphics, lines or columns of features, nor does it clarify the effect of certain controls or ML concepts. This is clearly related to the exploration strategy they offer users which is based on a trial and error strategy in a direct manipulation interface. The designers extensively explores dynamic signs to communicate their meta-message to the user. All interface items can be explored through mouse clicks. Thus, for experienced users (which is the target

audience) willing to explore the interface, the meta-message will likely be transmitted successfully.

6.3.2 Analysis of the Explanation Explorer design in Light of IML Principles

Principle 1 - Make task goals and constraints explicit: In Explanation Explorer, the task goals and constraints are well communicated through the interface elements, which drive interactions via mouse clicks to select explanations or filters, and sort rows or features. On the other hand, the effects of the available controls are not clear to users, who have to explore their effects to make sense of them. Thus, Explanation Explorer *partially complies* with this principle.

Principle2 - Support user understanding of model uncertainty and confidence: Explanation Explorer allows users to perceive the uncertainty associated with the model and explanations. The tool presents elements such as confusion matrix, ROC (*Receiver Operating Characteristics*) curve and AUC (*Area Under The Curve*) metric, related to model; and also odds ratio with whiskers showing the confidence interval for explanations. Thus, the tool *complies* with the principle.

Principle 3 - Capture intent instead of input: Explanation Explorer, through its interface, offers controls that allow the user to interact in a guided way, through mouse clicks and two text fields. Users do not have the possibility to provide noisy inputs that may misrepresent their real intention. Thus, Explanation Explorer *complies* with the principle.

Principle 4 - Provide Effective Data Representations: Explanation Explorer provides global and local views. Users can interact with views sorting and ordering the explanations in order of better understand the model. The Explanation Explorer does not provide data simplification or visualization of the raw data. Some elements presented may not be so helpful in cases with high dimensionality datasets. Thus, the tool *partially complies* with this principle.

Principle 5 - Explore interactivity and promote rich interactions: Explanation Explorer *partially complies* with this principle. The tool allows users to express their intentions through inputs to the system. Users can change the visualization of the explanations by ordering them according to different parameters. They can also choose which instances to inspect and follow the impact of the decisions on the screen. However, no resources are offered for simulations or model adjustments, nor to select the features that make up the explanations. As a consequence, interactions are limited.

Principle 6 - Engage the user: The tool *complies* with this principle. Explanation Explorer offers elements and controls which allow users to interact and perform actions that can be undone or reset, thus contributing to help users engage with the goal of understanding the model behavior.

6.4 ATMSeer Inspection

The ATMSeer tool consists of an interactive visualization tool to help users to monitor an AutoML process, analyze running models, and refine the AutoML search space in real time (Wang et al., 2019). It is implemented as a client-server system where the server accomplishes the AutoML process and data/model storage.

Our inspection focused on the client which consists of the visual interface, with graphical controls to coordinate the AutoML process and the views.

Although the ATMSeer code was availabe on Github [21], we were not able to to generate an executable version of the system. The code available seem not to be updated, and for many of the libraries used, the current version generated errors. We were unable to identify which versions or fixes were necessary to generate an executable version. Nonetheless, in ATMSeer's github there was a video containing a video of the system and how to use it, as well as other materials such the article that presented the system (Wang et al., 2019), and screenshots. The inspection was conducted based on the explanations and information about the system available, and not based on the inspection of the system itself. Even so, we considered that the material allowed us to apply SIM (even if we were not able to explore it ourselves), as it provided a detailed presentation of the system and its execution.

Next, we present the main parts of the consolidated meta-message from the ATMSeer designers (Figure 9), reconstructed through SIM. This is followed by the analysis of the tool design in light of the IML Principles.

6.4.1 Meta-message for *ATMSeer*

"We (designers of ATMSeer) understand that you": are a person with a certain level of expertise in ML, and have experienced the task of searching the most suitable ML Models manually, in a time-consuming and error-prone way.

"Our view about what you want or need to do": You want to search, analyze, and choose Machine Learning models for your own tasks, refining the search space of AutoML efficiently.

"How and why you want it to be done": You want a system with a visual and interactive interface, that allows you to supervise the AutoML process, analyze searched models, and refine the searching.

"This is the system we (ATMSeer designers) designed for you, and this is how you can or should use it in order to fulfill a range of purposes that fall within this vision."

The tool provides users with several views incorporated in a single screen, without menus. All possible actions are visible on the screen, through three parts (Figure 9):

- *Control panel (Figure 9 (1))*: through which users can perform the upload of a new dataset or select an existing dataset and create or resume an AutoML process.
- *Overview panel (Figure 9 (2))*: in this panel, users can follow the execution progress of the models and hyperpartitions and check high-level information about the performance of the best model. One can visualize and compare the top k models and activate the *focus mode* to highlight the corresponding algorithms and hyperpartitions details in the *AutoML Profiler* panel.

[21] ATMSeer: available on https://github.com/HDI-Project/ATMSeer

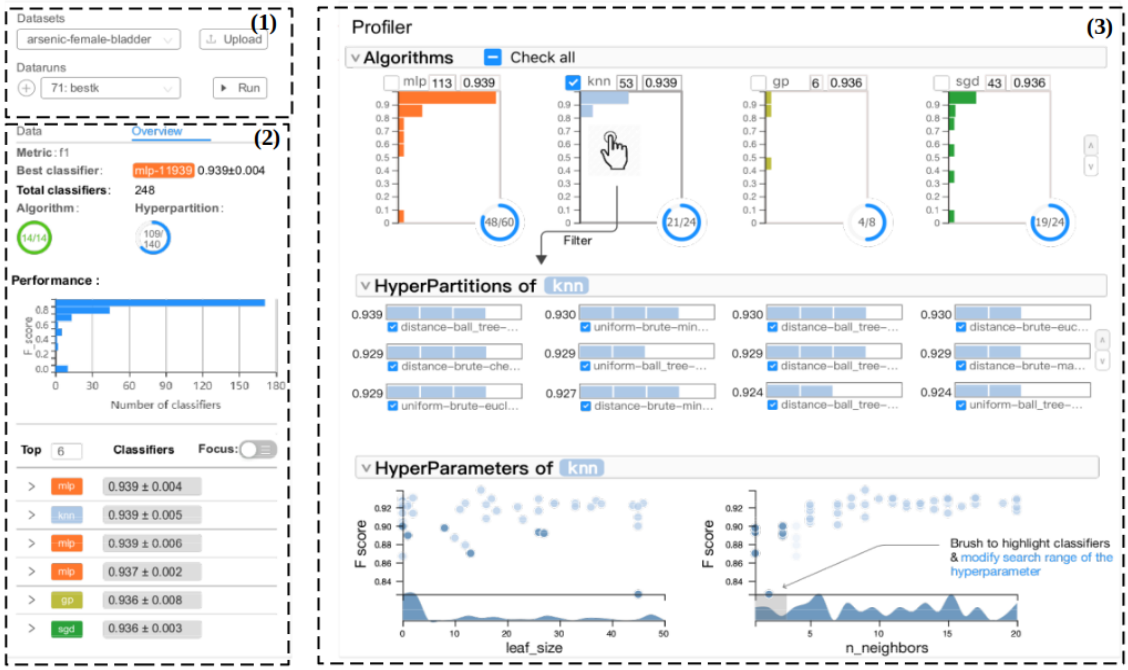

Figure 9. ATMSeer

- *The AutoML Profiler (Figure 9 (3)).*: through this panel, users can inspect the AutoML process using three different granularity levels: algorithm, hyperpartition, and hyperparameter-level. Users can observe how their choices of algorithms, hyperpartitions and hyperparameters influence model performance.

In the *Algorithm-level View* users can visualize the general model performance distribution of each ML algorithm using a histogram. *The Hyperpartition-level View* shows different hyperpartitions of a selected algorithm in the Algorithm-level View. Users can visualize the different properties of the algorithm, helping to analyze the search space and compare hyperpartitions. For each algorithm, its hyperpartitions are presented by a list of progress bars, where the height denotes the model's effectiveness.

Hyperparameter-level View presents the relation between performance and each tunable hyperparameter through a scatter plot where each model is visualized as a point in the plot. Users can realize how each hyperparameter influences the performance and use it to improve the search space.

ATMSeer provides *real-time control* to monitor, analyze, pause and reconfigure the AutoML process and restart it from a previous state. At that point, the ATMSeer interface is updated dynamically. Another resource consists of In-Situ Search Space configuration, which allows users to modify the search space at the same place they observe and analyze the search models, using the closest checkboxes and on-screen elements.

Considerations. The metalinguistic signs in the tool consist of tooltips about the screen elements, generated values related to algorithm executions or progress of the AutoML process. The tool does not offer tooltips or explanations about ML concepts. A disadvantage of the interface design is that it is not clear to users what each of the controls represents. The designers extensively explores static and dynamic signs to communicate their meta-message to users, and the expectation is that users will learn about the system through trial and error. Considering the target audience consist of knowledgeable users, the designers' meta-message has no inconsistencies and is well organized. The main style of interaction used is direct manipulation. All interface items can be explored through mouse clicks. So, for experienced users willing to explore the interface, the meta-message will likely be transmitted successfully.

6.4.2 Analysis of the ATMSeer design in Light of the IML Principles

Principle 1 - Make task goals and constraints explicit: ATMSeer *partially complies* with this principle. The task goals and constraints are well communicated through the interface elements and labels which drive interactions via mouse clicks and there are some tooltips. On the other hand, the interface does not make very clear to users what the controls represent, thus users have to explore the effects of these controls to make sense of them.

Principle 2 - Support user understanding of model uncertainty and confidence: ATMSeer allows users to perceive the uncertainty associated with the models. The tool uses *F1* metric, with confidence range and it is possible to inspect overall algorithmic performance through the effectiveness of the executed models. Users can analyze the impact of the hyperparameters and hyperpartitions on the overall algorithm performance. Thus, the tool *complies* with the principle.

Principle 3 - Capture intent instead of input: the tool *complies* with principle 3 given that it offers a certain degree of flexibility for users to provide their input and also supports them with rich views on the ongoing process. This prevents users from providing input with a certain level of uncertainty that generally exists between intent and user-provided input.

Principle 4 - Provide Effective Data Representations: The tool *complies* with Principle 4 given that it provides several views related to the overall performance of algorithms and search space, in the level of algorithm, hyperpartitions and hyperparameters (present in the interface). The tool does not include an instance level view, though this type of view makes no sense to the scope of work of Model Building. So, we can consider that this does not compromise the compliance of the tool with this principle.

Principle 5 - Explore interactivity and promote rich interactions: ATMSeer *complies* with this principle. The tool allows users to express their intentions through inputs to the system. The user can choose aspects of how and which views are presented. They present the real time picture of the AutoML process and users can use them as a basis for making adjustments to control the views and restart the process in progress, so that their actions are reversible.

Principle 6 - Engage the user: The tool *complies* with this principle. ATMSeer offers elements and controls which allow users to interact and perform actions that can be undone or reset, thus contributing to help users engage with the goal of understanding what is going on with the AutoML process. These elements and controls also help users to interact with the process of finding the best ML Algorithm for the dataset in case.

7 Discussion

Explain-ML is an interactive system that allows knowledgeable ML users to train their models for Automatic Text Classification (ATC) tasks. Our analysis, following the general principles of IML interface design (Dudley and Kristensson, 2018), allows a discussion of how the principles are represented in the tool and the impact they have on the vision and user experience.

It is worth mentioning that the Explain-ML development started before the publication of the proposed IML principles of (Dudley and Kristensson, 2018), thus, the principles were not used to guide the system design decisions. Thus, in the context of this work, the principles-based analysis of Explain-ML allows us to make a critical analysis of design decisions, considering an IML perspective. The analysis described the aspects of the system that met or were in line with the proposed principles. In addition, we identified, in the participants' comments, indicators that these aspects were considered positive by the participants in their perception and experience with the tool (subsection 6.1). Thus, the analysis highlights the relevant points for the users, corroborates the relevance of the principles and helps to consolidate them.

To further enrich the analysis carried out with Explain-ML users in light of the IML principles (Dudley and Kristensson, 2018), we carried out an analysis of how other IML tools meet those principles. For this purpose, we used SIM (Semiotic Inspection Method), having as reference the same IML principles, to evaluate the tools (subsections 6.2, 6.3, 6.4), contrasting the resulting analysis with the analysis of Explain-ML (subsection 6.1).

A comparison between the results of the analyses shows that there are commonalities among the tools inspected with SIM, related to their interaction strategy, types of interface signs and common problems identified. We also observed that Explain-ML and the other three tools presented several strategies that are in line with the IML principles.

Regarding the complete unified meta-message of the three tools, we noticed some common aspects in all of them. The tools provide several interpretability strategies presented in a single screen, without menus, and all possible actions are visible on the screen. We also observed the use of few or no metalinguistic signs, and the few existing ones are generally tooltips, focused on interface elements and not related to ML concepts. The meta-message is usually well organized and the designers of the tools extensively explore dynamic signs to communicate their meta-message to the users. The design is based on a trial and error strategy, in a direct-manipulation interface. The target audience varies between domain expert with little ML knowledge, but work with ML systems (RuleMatrix), to data scientist and domain experts (ATMSeer). Accordingly, it would be desirable for the tools to explore more metalinguistic signs, focused on interface elements and ML concepts.

In contrast, Explain-ML was aimed at knowledgeable ML users and provided a multi-perspective view to support users' interpretability. Each perspective is shown in a different tab, that allows users to navigate through them by clicking on the desired tab. Different from the other tools, Explain-ML made use of metalinguistic signs, both to explain the interface itself, but also to help users remember concepts (e.g. meaning of each metric shown) and results (e.g. comparing a result to that of a previous execution).

Concerning the IML principles, Explain-ML tool fully met all the six principles. For the other three tools -Rulematrix, Explanation Explorer and ATMSeer - principles 3 (*Capture intent instead of input*) and 6 (*Engage the user*) are fully met. The other principles are met either partially or fully, depending on each tool. In this context, we observe that the design of the analyzed tools already shows a tendency to fulfill what is defended by these principles, independently whether they have been known and taken into account during their design. Table 2 summarizes the results of the analysis of how each system in light of the IML.

Table 2. Overview of how the tools meet IML principles. (**N**: does not meet the principle, **P**: partially meets the principle, **Y**: yes, meets the principle).

Tools	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**
RuleMatrix	P	P	Y	P	Y	Y
Explanation Explorer	P	Y	Y	P	P	Y
ATMSeer	P	Y	Y	Y	Y	Y
Explain-ML	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*

As shown in Table 2, Explain-ML was the only system to be considered compliant to all 6 principles, ATMSeer complied to 5 principles and partially complied to 1 (*Principle 1 - Make task goals and constraints explicit*), and the other 2 complied to 3 principles, and partially complied to other 3 (not the same ones). We cannot pinpoint the factors that led to these results, especially because we do not have enough information regarding the design process of the inspected

tools. Nonetheless, the fact that the Explain-ML design process adopted a user-centric view, involving the user to elicit their needs and preferences and through prototype formative evaluation might have played a role in this result.

Considering the principles aimed at guiding the design of IML interfaces, the proposal is recent, and, to the best of our knowledge, they have not been used to analyze existing systems. The analysis of how each tool complies (fully or partially) with each principle, can be useful not only to compare the interactive aspects of the tools, but also guide designers in improving them.

Considering each principle in isolation (each column in Table 2), we can see that Principles 1 and 4 (*Principle 1 - Make task goals and constraints explicit and Principle 4 - Provide Effective Data Representations*) were the ones that more than one system was not able to fully comply with. It would be worth investigating if this was due only to design choices of the respective tools, or whether these principles pose specific challenges to achieving them in the interface. At any rate, researchers or designers working ML interpretability systems can pay special attention to them.

Finally, our work describes the analyses based on the IML principles, allowing readers to better understand them, and illustrating how different design decisions made in the systems impacted the compliance to the principles. Therefore, our work is useful in fostering a discussion about the principles themselves and in the process of their consolidation.

8 Limitations

Our research adopted qualitative methods, throughout its development. As is the case with qualitative methodologies (Lazar et al., 2017; Flick, 2008a), the goal here is to provide in-depth results, focused on a specific context, which are by design not generalizable.

During the Explain-ML development, the initial motivation for a system that could support users in creating ML models, including interpretability, came from the literature. As we understood the relevance for the HCML approach, we interviewed users to better understand their practices, preferences and needs to guide our design decisions, and performed a formative evaluation of the system prototype that illustrated our proposal. In both cases, only a few participants were included (7 and 3, respectively). Our recruiting was opportunistic and through researchers' contacts. Thus, due the small number of participants and their recruitment, our results may not represent the needs or views of other ML knowledgeable users intended as users of the system.

In the same direction, the final evaluation of Explain-ML, adopted a qualitative method and included only 6 participants, who had some relation, at the time or previously, with the ML research group developing the system. Although, the small number of participants is adopted both for interaction design with focus on the interface (Preece et al., 2019a), and qualitative research (Lazar et al., 2017; Flick, 2008a), it may be biased and not represent all potential users the system. Although we can argue that the results are relevant to potential users of the system, they may be limited by the participants' experience and context, and not include other perspectives.

The in-depth analysis conducted and their results generated valuable insights to the development and evaluation of Explain-ML. Nonetheless, we will conduct broader evaluations, including more participants, with more diverse backgrounds and levels of experience in ML. The results of these new studies will be contrasted with the current ones.

Regarding our methodology, the analysis of the IML principles considered different methods. Although the decision was justified, not only by the bias of designers applying SIM to their own system, but also by the research design, the different perspectives of each method may impact the results. To mitigate this potential impact, while discussing the results of our analysis with each method, we have also presented the collected evidence that supports our results, allowing readers to understand how they were achieved (Flick, 2008b).

Finally, regarding the systems selected for inspection, our goal was to use the executable version of all of them. Although they all had their code available in GitHub, generating an executable version was a challenge. For all three cases we ran across multiple issues, such as the incompatible or unspecified versions of the libraries. We had ourselves to identify versions that would make possible to generate an executable version. For RuleMatrix and Explanation Explorer we were successful, but actions taken might have impacted other aspects of the the executable version generated that we are not aware of. Finally, for ATMSeer, it was not possible to generate an executable version of the system. Nonetheless, as a complete and detailed demo video was available, as well as other materials presenting the system, we performed the analysis of the system based on them. On one hand, this guaranteed that our changes did not introduce any impacts on the system; on the other, we were not able to conduct the inspection of the system itself, and the material may not have shown every sign or interactive path available to users.

9 Conclusions and Future Work

In this article, we have presented Explain-ML (Lopes, 2020), an interactive multiperspective visual tool for interpretability of Machine Learning, as well as it design process based on a Human-Centered Machine Learning (HCML) approach. We have analyzed how Explain-ML complies or not to the IML principles proposed by Dudley and Kristensson (2018), that target users needs regarding ML interpretability. The analysis was based on the results of users' evaluation of Explain-ML (Lopes, 2020).

The evaluation of Explain-ML was extended by contrasting it with the analysis of how other IML tools available in the literature address the IML principles. We used the Semiotic Inspection Method to systematically inspect three other tools (Rulematrix, Explanation Explorer and ATMSeer).

Considering each analyzed system, our investigation can be useful to pointing directions that it may be interesting to continue the research or their development. For Explain-ML, the results may be an indicator of the positive impact of the HCML design process that guided its development. Furthermore, it indicates that it would be worth investing in including other ML models in the system, as initially intended. This process would benefit from continuing an HCML approach, but also from including in the design process considerations

about the IML principles. The results for the other tools can be useful for their designers to identify how to improve interactive aspects of their systems. Furthermore, the detailed analysis of each tool and the contrast between them, allows readers to compare their strengths and weaknesses .

By describing the systems, and how they did or not comply to each IML principle, we contribute to the consolidation of the proposed IML principles. We also generate indicators of the value to use the IML principles not only to guide design, as proposed (Dudley and Kristensson, 2018), but also to guide evaluation and even contrast exsiting systems. In this sense, our work advances the knowledge in HCML.

Our next steps will involve a more extensive evaluation of Explain-ML, considering both a larger group of participants, as well as its use in real contexts. We also aim to propose, implement and evaluate new visualizations to provide ML interpretability, as well as specific visualizations for different ML models. Finally, as one of our current focus is on textual datasets, we will exploit state-of-art Neural Transformer architectures, especially Attention Models, such as BERT and derivatives, and adapt/extend Explain-ML with approaches to interpret those models.

CRedit author statement

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.*

Acknowledgements

We thank all the participants who agreed to contribute to the evaluation of the Explain-ML tool and its development. This work was partially supported by CAPES, CNPq and FAPEMIG.

References

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

Carroll, J. (2000). Introduction to this Special Issue on "Scenario-Based System Development". *Interacting with Computers*, 13(1):41–42.

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.

Cunha, W., Mangaravite, V., Gomes, C., Canuto, S., Resende, E., Nascimento, C., Viegas, F., França, C., Martins, W. S., Almeida, J. M., et al. (2021). On the cost-effectiveness of neural and non-neural approaches and representations for text classification: A comprehensive comparative study. *Information Processing & Management*, 58(3):102481.

De Souza, C. S. (2005). *The semiotic engineering of human-computer interaction*. MIT press.

De Souza, C. S. and Leitão, C. F. (2009). Semiotic engineering methods for scientific research in hci. *Synthesis Lectures on Human-Centered Informatics*, 2(1):1–122.

De Souza, C. S., Leitão, C. F., Prates, R. O., and Da Silva, E. J. (2006). The semiotic inspection method. In *Proc. of VII Brazilian symposium on Human factors in computing systems*, pages 148–157.

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

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

Dudley, J. J. and Kristensson, P. O. (2018). A review of user interface design for interactive machine learning. *ACM TIIS*, 8(2):8.

Erik, S. and Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11(Jan):1–18.

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

Fiebrink, R. and Gillies, M. (2018). Introduction to the special issue on human-centered machine learning. *ACM TIIS*, 8(2):7.

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

Flick, U. (2008b). *Managing quality in qualitative research*. Sage Publications Ltd., 1th edition.

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 *Proc. of the 2016 CHI*, pages 3558–3565.

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.

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

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*.

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.

Hall, A., Bosevski, D., and Larkin, R. (2006). Blogging by the dead. In *Proc. of the 4th Nordic conference on Human-computer interaction: changing roles*, pages 425–428.

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

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.

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

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.

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

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*.

Krause, J., Perer, A., and Ng, K. (2016b). Interacting with predictions: Visual inspection of black-box machine learning models. In *Proc. of the 2016 CHI*, pages 5686–5697.

Labs, C. F. F. (2020). Interpretability, Report FF06. Technical report, Cloudera Fast Forward Labs. `https://ff06-2020.fastforwardlabs.com/`.

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

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

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 *Proc. of the XX Brazilian Symposium on Human Factors in Computing Systems*, pages 1–11.

Lopes, B. G. C. O. (2020). Explain-ml: A human-centered multiperspective and interactive visual tool for explainable machine learning. Master's thesis, Universidade Federal de Minas Gerais, Belo Horizonte, Minas Gerais, Brasil.

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.

Madsen, S. and Nielsen, L. (2010). Exploring persona-scenarios - using storytelling to create design ideas. In Katre, D., Orngreen, R., Yammiyavar, P., and Clemmesen, T., editors, *Human Work Interaction Design*, pages 57–66.

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.

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.

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.

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.

Pereira, F. H. S., Prates, R. O., Maciel, C., and Pereira, V. C. (2017). Combining configurable interaction anticipation challenges and volitional aspects in the analysis of digital posthumous communication systems. *SBC Journal on Interactive Systems*, 8(2):77–88.

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

Preece, J., Sharp, H., and Rogers, Y. (2019b). *Interaction design: beyond human-computer interaction*, page 408. John Wiley & Sons.

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*, page W11.

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

Ribeiro, M. T., Singh, S., and Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Rosson, M. B. and Carroll, J. M. (2002). Scenario-based usability engineering. In *Proc. of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 413–413.

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

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

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

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*.

Vilone, G. and Longo, L. (2021). Notions of explainability and evaluation approaches for explainable artificial intel-

ligence. *Information Fusion*, 76:89–106.

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 *Proc. of the 2019 CHI*, page 681.

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

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.

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.

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.