Noname manuscript No.
(will be inserted by the editor)

Editable Machine Learning Models? A Rule-based Framework for User Studies of Explainability

Received: date / Accepted: date

Abstract So far, most user studies dealing with comprehensibility of machine learning models have used questionnaires or surveys to acquire input from participants. In this article, we argue that compared to questionnaires, the use of an adapted version of a real machine learning interface can yield a new level of insight into what attributes make a machine learning model interpretable, and why. Also, we argue that interpretability research also needs to consider the task of humans editing the model, not least due to the existing or forthcoming legal requirements on the right of human intervention. In this paper, we focus on rule models as these are directly interpretable as well as editable. We introduce an extension of the EasyMiner system for generating classification and explorative models based on association rules. The presented web-based rule editing software allows the user to perform common editing actions such as modify rule (add or remove attribute), delete rule, create new rule, or reorder rules. To observe the effect of a particular edit on predictive performance, the user can validate the rule list against a selected dataset using a scoring procedure. The system is equipped with functionality that facilitates its integration with crowdsourcing platforms commonly used to recruit participants.

Keywords Rule learning · User experiment · Crowdsourcing · Explainable Artificial Intelligence · Cognitive Computing · Legal compliance

1 Introduction

While rule-based models are not currently considered as a main-stream topic due to their lower predictive performance than achieved by neural networks or random forests (Fernández-Delgado et al, 2014), this is changing as explainability of machine learning models is gaining on importance. For example, neural networks and random forests have the best predictive performance but are considered uninterpretable. Common examples of interpretable models are Bayesian networks and decision trees (Miller, 2019). The advantage of the rule-based representation is not only good interpretability; rules are well-suited for learning and classification over relational data, which they can represent more naturally than most other methods. The use of rules is not limited to classification, as, e.g., association rule learning is commonly used for exploratory data analysis (descriptive data mining) and finding interesting patterns (Fürnkranz et al, 2012). Also, outside machine learning, there is a large and long-established eco-system of software tools and algorithmic solutions which work with rules. There are already standards, such as PMML[1] or RuleML (Boley et al, 2010), allowing

[1] http://dmg.org/pmml/v4-4/GeneralStructure.html

Accepted to be published in the journal of Advances in Data Analysis and Classification, Springer

the transfer of rule learning results to industry business rule management systems, which earlier relied only on manual input of rules. While rules can be learnt from historical data, valuable inputs for the learning process can also be directly elicited from domain experts, as it is natural for people to express their knowledge in the form of rules. Overall, rule learning can offer a solution to end-to-end machine learning workflow, where a single interpretable representation is used to elicit existing knowledge, to encode the results of learning and deploy these to production systems. Rules also have excellent properties in terms of potential for compliance with recently introduced regulations and guidelines. For example, the EU Guidelines for Trustworthy Artificial Intelligence foresee a right to intervention to the machine learning model, a requirement difficult to meet for most machine learning models, including random forests and neural networks. Since a rule model is composed of multiple individual rules acting as local classifiers, rule-based models can be more easily edited.

While rules are, in principle, intrinsically comprehensible, there is a paucity of research on how to optimise the presentation of rules and the interaction with them so that possible misinterpretations are avoided, and users spend only the necessary amount of time on these tasks. While many studies are primarily interested in assessing to what degree humans comprehend the generated explanations, some user studies also have the understanding of the participants' mental models applied in the actual induction process as one of the explicit sub-objectives. Psychological research specifically on hypothesis testing in the rule discovery task has been performed in cognitive science at least since the 1960's (Wason, 1960, 1968), but this topic is underexplored in the area of machine learning. According to Miller (2019), one of the most successful studies in this domain was done by Kulesza et al (2015), who report on an experiment with a between-subject design focused on 'explanatory debugging' of a Naive Bayes classifier. After Bayesian reasoning, rule learning is the next contact point between machine learning and cognitive science. The most recent research in this area is a study focused on human-in-the-loop-analytics for text categorisation, where classification rules are automatically extracted from data, and then domain experts (natural language processing engineers) are explained the individual rules and asked to refine them (Yang et al, 2019). The results of this study showed considerable improvements in text classification performance as well as a generalisation potential. What this study also showed is that obtaining detailed information on user behaviour is demanding in terms of resources if the screen and audio recordings have to be manually analysed.

In this paper, we introduce a rule editor designed specifically for empirical studies of explainability, but the editor could be also adapted for the study of mental models. We believe that there is a connection between human thought processes and rules as machine learning models, and with the presented software framework, our aim is to facilitate focused investigation of this connection. The editor is an extension of an existing system for interactive rule learning called EasyMiner (Vojíř et al, 2018). This system, combined with the editor, provides a comprehensive web framework that allows the user to learn a rule-based model from provided data, and then review the model and modify it according to user preferences. The user gets feedback on the predictive performance of the model. To enhance the model, the user can use the editor, for example, to delete a condition from a rule. The system saves data on user interaction which can be analysed by interpretability researchers.

This paper is organised as follows. Section 2 briefly reviews related work. Section 3 presents the EasyMiner system and the editor. Section 4 provides a methodology for using the proposed system as a software aid in laboratory and crowdsourcing experiments. Section 5 reports the first insights from an empirical study with test users. Section 6 briefly describes the architecture of the system and its availability. The conclusions briefly summarise the contributions and provide an outlook for future work.

2 Related work

Related research applicable to this paper comes from several principal areas: quantifying explainability of machine learning models, research specifically on the interpretability of rule-based models, the software utilised in prior user studies of explainability, and possible applications of such user studies.

In our brief review of each of these areas presented below, we concentrate on research on explaining symbolic (“white-box”) models, such as decision trees and rules. Research focusing on explaining “black-box” models, such as neural networks or random forests, is only marginally covered.

Quantifying explainability The main line of research focuses on gauging the size of the model, following the assumption that smaller models are more understandable. For rule-based models, the commonly applied metrics include the number of conditions (or cognitive chunks) in the rule and the number of rules in the model (García et al, 2009; Lakkaraju et al, 2016; Lage et al, 2019).

The intrinsic explainability metrics, which can be derived directly from the model, are often complemented by user studies that use quantifiable criteria such as the time taken by the user to apply the model, or a percentage of correct answers when manually using the model to predict the class of unseen instances.

In the related context of decision trees, Piltaver et al (2016) proposed a methodology for measuring comprehensibility of decision trees, which includes six tasks (classify, explain, validate, discover, rate and compare). The first four tasks were designed to objectively quantify to what degree comprehensibility is affected by selected parameters of the decision tree learning algorithm. The rate and compare task were designed to assess the subjective perception of comprehensibility.

In the context of rule learning, the aim of the study performed by Lakkaraju et al (2016) was to compare comprehensibility of rule lists with the comprehensibility of rule sets. The study is composed of a mix of descriptive and multiple-choice questions. These aim at evaluating user understanding of decision boundaries of two types of rule models. The descriptive questions were evaluated by judges, the multiple-choice answers were assessed automatically, as they primarily correspond to the ability to mechanically apply the model to classify a specific instance. In some cases, the classification was not possible due to incomplete information provided to the user. Comprehensibility was measured in terms of average user accuracy and the average time spent. A more refined approach to combining the accuracy of human predictions with the required time was taken in recent research focused on rule-based programs learned by Inductive Logical Programming (ILP) approaches (Muggleton et al, 2018). In this study, the authors define *inspection time* as the mean time that a participant spent studying the model before applying it.

The scope of recent research has broadened the focus of user studies to psychological and pragmatic aspects of explanation (Schmid and Finzel, 2020; Páez, 2019; Fürnkranz and Kliegr, 2018). The problems studied include effects of cognitive biases on the interpretability of rule-based models (Kliegr et al, 2018), as well as cognitive preferences that determine perceived plausibility of rules (Fürnkranz et al, 2020). Some results suggest that while smaller models may take a shorter time for users to apply, they can also be perceived as less understandable (plausible).

Rule-based explanations as lingua franca Research in inductive rule learning has been pursued for more than 50 years (Michalski, 1969; Hájek et al, 1966), and is still an active area of research. Modern algorithms, such as Bayesian Rule Sets (Wang et al, 2016) and Fuzzy Association Rule Classifiers (Elkano et al, 2014), combine rule learning with other machine learning approaches. The rule-based representation is also not limited to algorithms that directly learn rules. Other types of models can also be converted to rules, or rules can be extracted from them. Decision trees can be represented as a rule model when one rule is generated for each path from the root of the tree to a leaf. Also, the decision table, which was found to be the most comprehensible type of model in the empirical study of Huysmans et al (2011), can be represented as a set of rules.

Rule models can also be used to explain some “black-box models”. Notably, rule extraction from neural networks has already been studied for several decades, cf., e.g., Towell and Shavlik (1993); Zilke et al (2016). Other types of models, such as Support Vector Machines and Random Forests, are also amenable to rule extraction (Barakat and Bradley, 2010; Rapp et al, 2019).

Table 1 Overview of prior empirical research on explainability of machine learning models. Value N denotes the number of participants across all empirical studies in the cited paper

study	cohort	elicitation software	N	models
Lage et al (2019)	crowdsourcing	questionnaire	900	rules
Muggleton et al (2018)	students with knowledge of ILP	questionnaire	121	rule models
Fürnkranz et al (2020)	crowdsourcing	questionnaire	390	rules
Piltaver et al (2016)	machine learning experts, IT specialists and IT students	proprietary web interface	52	decision trees
Lakkaraju et al (2016)	data science students	questionnaire	47	rules
Huysmans et al (2011)	graduate business students, doctoral researchers	proprietary web interface	42	decision tree, decision table

Software support for empirical studies of explainability Research fields that involve human-computer interaction (HCI) have developed software tools that help to execute user studies in realistic environments. These tools support tasks as diverse as eye-tracking experiments (Dalmaijer et al, 2014), studying users' color preferences (Tomanová et al, 2019), or performing psychophysics experiments (Brainard and Vision, 1997).

While there were already multiple user studies on the comprehensibility of machine learning models, these typically relied on questionnaires or used proprietary software that was not publicly released. Such studies include those focused on rule models (Lakkaraju et al, 2016; Lage et al, 2019), and decision tables (Huysmans et al, 2011).

Table 1 provides an overview of these studies, their size, and the type of software support used. It can be observed that the benefit of recent studies relying on crowdsourcing platforms are much larger user samples. On the other hand, the data elicitation method in these studies is limited to questionnaires. For example, one research highly relevant for the presented tool is an empirical study by Muggleton et al (2018) showing that ILP-generated rule models meet the requirements for ultra-strong machine learning, since they enhance human classification performance on unseen data. In this study, the authors have asked participants – students with knowledge of ILP – to manually design classification rules based on presented training data. The hand-designed rules created by the participants were found to be on average of lower accuracy than rules induced by an ILP system. For authoring the rules, the participants had to rely on a generic web-based questionnaire system.

We hypothesise that a combination of crowdsourcing with user interfaces specifically designed for research on explainability of machine learning models can provide additional insights compared to research relying on questionnaires only.[2]

Applications of results of psychological studies on explainability Insights obtained in user-studies on explainability can be used to define optimality criteria for learning algorithms. For example, the Interpretable Decision Sets algorithm (Lakkaraju et al, 2016) allows for user-set weights for various facets of interpretability, including overlap between rules and whether – and to what extent – larger rule lists should be penalised.

Another line of applicable work is related to the presentation of information, as follows from research expanding on the conversational rules set out by Grice (1975). While conversational rules have considerably influenced the design of other domains of human-computer interaction, such as question answering, they have not yet been used, with a few exceptions, in machine learning. However, such use shows promise as demonstrated on their use in rule learning in Sorower et al (2011).

Studies focused on avoidance of misinterpretation of rules caused by cognitive biases can improve interpretability of rules. In Experiment 3 presented in Fürnkranz et al (2020), we describe a user

[2] It should be noted that the current version of the editor does not meet the requirements of the particular study by Muggleton et al (2018), since some syntactical constructs necessary for the expression of ILP rules are not supported.

study supporting the conclusion that if rule confidence and rule support are presented simultaneously, the users will almost disregard the support. This finding is consistent with the insensitivity to sample size effect.

3 EasyMiner and the Rule Editor

The rule editor introduced in this paper is a part of EasyMiner (Vojíř et al, 2018), which is a web-based data mining framework supporting mining of association rules, rule pruning and building models for anomaly detection. The editor provides a user-friendly graphical user interface usable in all modern web browsers, as well as a comprehensive REST API for integration with other applications.

An initial version of this rule editor was developed for learning business rules (Vojíř et al, 2014). The work presented here is substantially reworked and extended. In addition to new functionality specific to use in empirical studies with human subjects (logging of user actions, anonymous/external users), the changes include the code base partly rewritten to reflect the new mining backend used in EasyMiner, better integration of the editor with EasyMiner, and new export functionality.

Setting up rule mining task Association rules discovered in EasyMiner/R are of the form *antecedent* → *consequent*, where antecedent and consequent are conjunctions of *literals* (attributes with concrete values).[3] The user selects attributes that can appear in the antecedent and consequent of the discovered rules and defines minimum threshold values of interest measures. The most commonly applied measures, which are also supported by EasyMiner, are *confidence*, *support* and *lift*. Refer, e.g., to Fürnkranz and Kliegr (2015), for the definition of these measures.

Building rule-based classifiers For the building of classification models, the consequent needs to contain only the target attribute. EasyMiner uses the Classification Based on Associations (CBA) algorithm (Liu et al, 1998) to convert the discovered list of rules to a rule-based classifier. This algorithm sorts the rules according to strength and consequently removes redundant rules.[4] A default rule (rule with empty antecedent) is also placed at the end of the rule list. This rule classifies instances not covered by any of the previous rules.

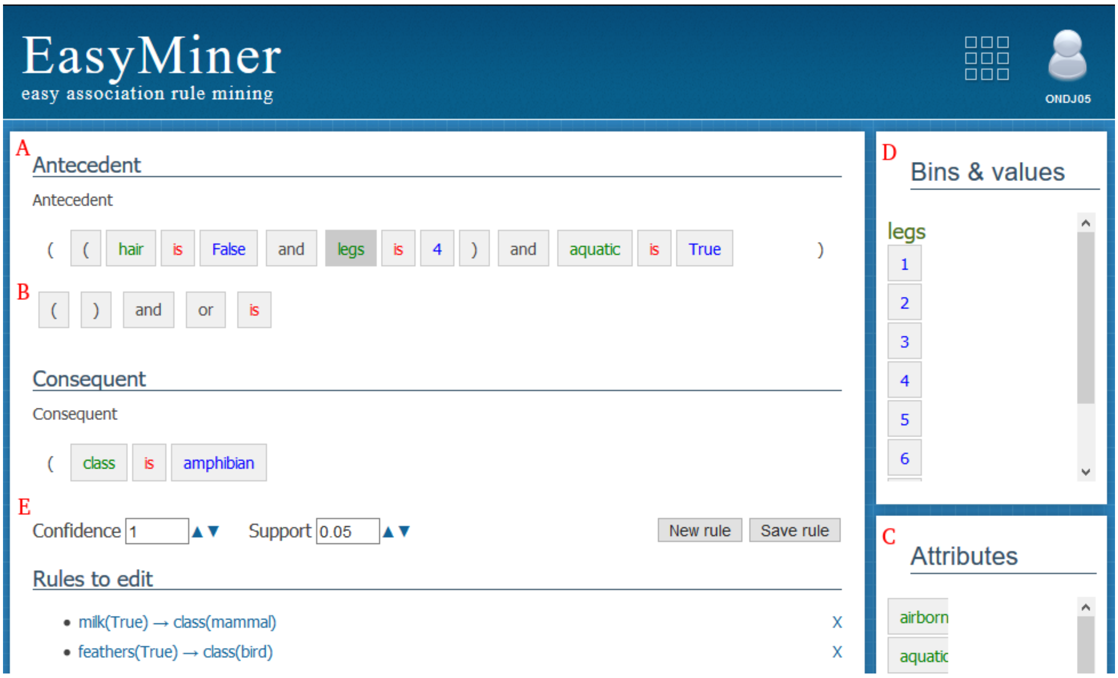
Rule editor The user has the option to save discovered rules to a knowledge base. Subsequently, the rule editor (see Fig. 1) can be activated. The user can edit existing rules, add new rules, and remove existing rules. The user can also change positions of rules in the rule list by changing their confidence and support (reordering rules). To modify an existing rule, the user clicks on the rule in the list. The rule is loaded into the main editor area and decomposed to modifiable elements - attributes, values, boundaries (open, closed interval) and logical connectives.

Model evaluation The user can check the quality of the model against a chosen data set. An exemplary result of the model evaluation part is shown in Figure 2. To apply the rule list on a new instance, the system takes the class predicted by the consequent of the first rule[5] whose antecedent matches the instance.

[3] In this paper, we do not consider the more expressive rules based on the GUHA method (Hájek et al, 1966) that earlier versions of EasyMiner could also process.

[4] The rules are sorted by *confidence*, *support* and *antecedent length*, which is the number of attribute-value pairs in the condition of the rule. For confidence and support, the higher value is better. For antecedent length, the shorter (and simpler) antecedent is preferred.

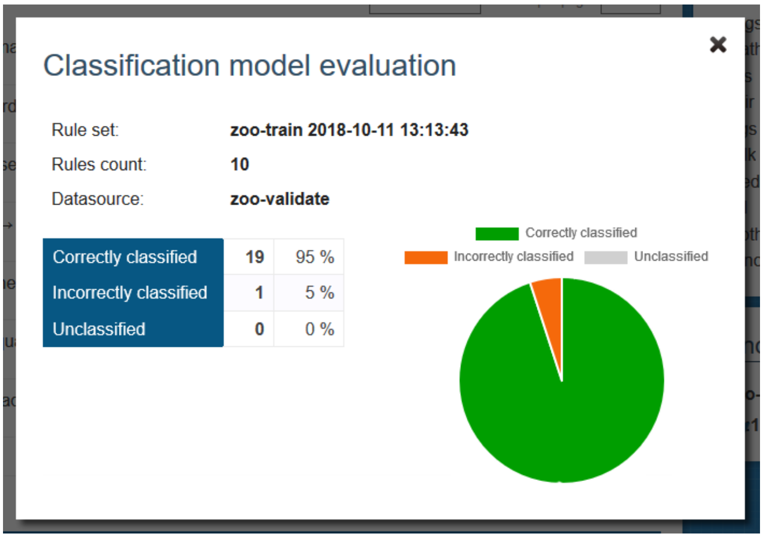
[5] Sorted by confidence, support and length as noted above.



Confidence 1 Support 0.05

- milk(True) → class(mammal) X
- feathers(True) → class(bird) X

Fig. 1 Rule editor with one loaded rule (zoo dataset). Fig. 1A shows a list of conditions of the currently edited rule decomposed to individual blocks corresponding to attribute names (hair, legs, aquatic), values (such as False, True and 4) and syntactic elements (round brackets, is, and). The user can delete selected block (block legs is selected), insert new syntactic blocks by dragging them from the palette displayed below (Fig. 1B), insert new attributes (Fig. 1C) and their values (Fig. 1D), or edit the values of Confidence and Support (Fig. 1E).



Rule set: **zoo-train 2018-10-11 13:13:43**

Rules count: **10**

Datasource: **zoo-validate**

Correctly classified	19	95 %
Incorrectly classified	1	5 %
Unclassified	0	0 %

Fig. 2 Evaluation result for a classification model. In the example shown in the figure, a rule model consisting of 10 rules was applied on 20 instances from the UCI zoo dataset. For 19 the class of the animal was predicted correctly.

4 Usage in Empirical Studies

EasyMiner with our Rule Editor can principally support cognitive experiments in two ways (or modes). In the first mode, the users perform the complete data mining cycle. The second mode involves studies involving only the rule editor. This section describes these two modes of operation in detail.

4.1 Complete Data Mining Workflow Mode

In this mode, study participants perform all tasks as in a real data mining assignment including data preprocessing, setup of the mining task and model evaluation. The last step are the edits to the generated rule lists in the rule editor. The EasyMiner system saves selected intermediate results of the user interaction on the server for further inspection by the experimenter.

A significant limitation of this option is that while the user interface is intuitive, performing the complete data mining cycle presupposes that the participants either possess prior knowledge of data mining or have received detailed instructions.

4.2 Crowdsourcing Mode

Already the European Union's General Data Protection Regulation (GDPR) provides the right to obtain *human intervention* for some decisions based on data analysis (Roig, 2018). There is a growing emphasis on presenting machine learning models so that they are understandable to people without training in machine learning (HLEGAI, 2019). For example, a user may want to review a set of preference rules comprising her user profile created based on her past shopping behaviour by a recommender system.

The EasyMiner *Crowdsourcing mode* is designed to support empirical studies focusing on the interaction between non-expert users and rule-based models. The interface aims to facilitate integration with crowdsourcing workflows but can also be used for laboratory user studies.

The difference from the Complete data mining workflow mode described in the previous subsection is that in the Crowdsourcing mode, the experimenter is responsible for generating the initial rule list, which is presented to the user within the rule editor. The interactivity entails the ability to manipulate with the rules and to obtain feedback in terms of evaluation of model quality.

The web-based nature of the system makes user studies easier to setup and administer. The system needs to be installed only once on the server and is accessed through regular internet browsers. This facilitates use in crowdsourcing user studies with remote participants.

4.3 Setup of Crowdsourcing User Study by Experimenter

Design of a user experiment with the presented software framework has several phases, which are schematically depicted in Figure 3. In the following, we will describe them in greater detail.

Generation of initial rule list In the first phase, the experimenter uploads a dataset into EasyMiner, prepares data and executes data mining tasks and adds rules to the knowledge base. Subsequently, the experimenter transforms the knowledge base into a user study - selects testing data set (for model evaluation) and obtains *Experiment task URL* to be distributed to participants.

Setup of crowdsourcing task In the second phase, the experimenter sets up a task in the crowdsourcing platform, such as Amazon Mechanical Turk. The experimenter gives instructions for completing the task and inserts into the instructions the Experiment task URL, which is the same for all participants.

Additionally, the experimenter sets the common attributes of crowdsourcing tasks, such as the number of participants that should be recruited by the crowdsourcing platform, their geographic location and the level of payment. Ethical recommendations typically suggest that the level of remuneration set by the experimenter is related to the average time spent with completing the task. Referring to values collected using a test run, the average time and consequently, the costs can be determined from timestamped logs of user actions that are created by EasyMiner.

Quiz mode In line with recommendations for designing crowdsourcing empirical studies, we suggest that the instructions contain a quiz session that each participant has to pass to become eligible for the main task. The quiz tests whether the participant understands the instructions and can work with the EasyMiner environment. Several types of quiz questions that we used in prior crowdsourcing research on rule plausibility can be found in Fürnkranz et al (2020).

Motivational bonuses In addition to the quiz mode, an essential element that helps keeping participants focused on the task is an assignment of monetary bonuses that depend on performance. Research has shown that psychological studies with monetary bonuses for correct answers are more realistic (Yin et al, 2014). In the rule learning context, a bonus can be tied to performing edits that improve the accuracy of the model by at least a specific number of percentage points.

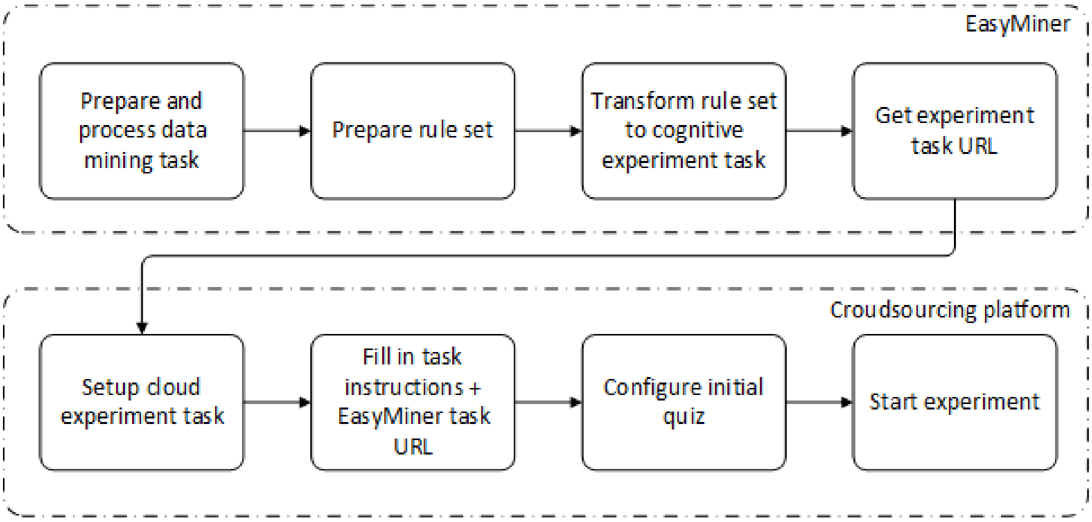

EasyMiner
Prepare and process data mining task
Prepare rule set
Transform rule set to cognitive experiment task
Get experiment task URL
Croudsourcing platform
Setup cloud experiment task
Fill in task instructions + EasyMiner task URL
Configure initial quiz
Start experiment

Fig. 3 Setup of an empirical study by the experimenter

4.4 Actions performed by the participant

The actions performed by participants are shown in Fig. 4. The participant first reads a brief description of the task and decides whether to accept it. Then the participant completes the quiz mode if enabled by the experimenter.

In EasyMiner, the user is welcomed with a customisable introduction text. When the user starts the task, EasyMiner creates a copy of the rule set prepared by the experimenter and automatically generates a new *External User ID* – the participant is not required to create any user account. The participant is then redirected to a simplified user interface, which has two principal components: rule editor and model tester, which allows evaluating the edited model against a test set uploaded by the experimenter.

After completion of all operations, the user copies the EasyMiner External User ID and inserts it into the crowdsourcing platform. In the crowdsourcing platform, the user can be presented with an additional questionnaire. Note that some crowdsourcing platforms, such as `Prolific.ac` do not provide any direct means for collecting participant input. This needs to be collected through an external survey tool. In this case, the survey needs to ask the participants to input both the External User ID assigned by EasyMiner and the user ID assigned by the crowdsourcing platform.

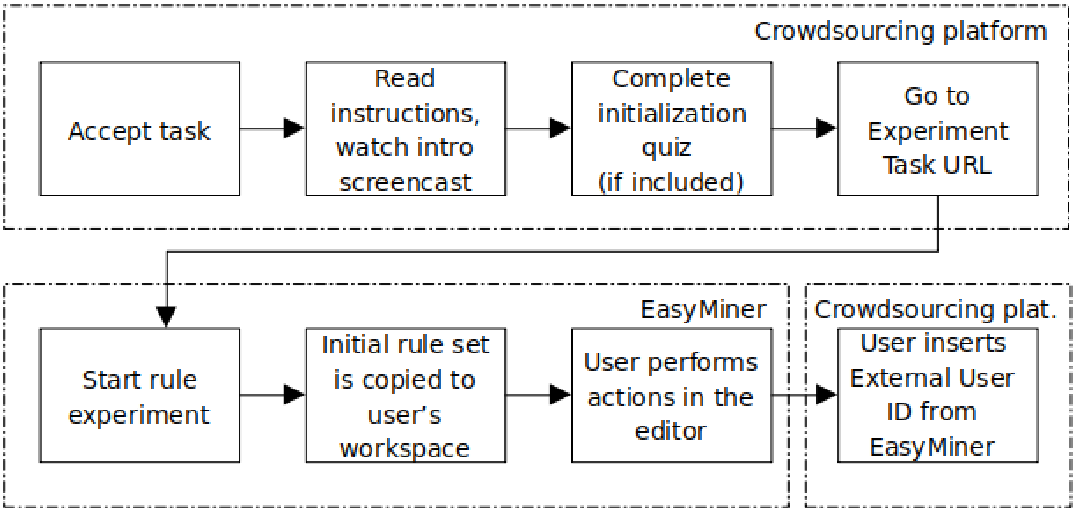

Crowdsourcing platform
Accept task
Read instructions, watch intro screencast
Complete initialization quiz (if included)
Go to Experiment Task URL
EasyMiner
Start rule experiment
Initial rule set is copied to user's workspace
User performs actions in the editor
Crowdsourcing plat.
User inserts External User ID from EasyMiner

Fig. 4 Participant workflow

4.5 Data Logged, Export Options

EasyMiner offers several export options. Rule lists can be exported in plain text.

The system stores a log file containing a list of the main operations performed by the user exemplified in Figure 5. It includes entries for saving rule, adding a new rule, removing rule, and evaluation of the rule model. Analysis of this data shows, for example, how many partial steps the user performed or how much time the user spent working on the assignment.

5 First Experience from User Studies

As part of internal testing we used the rule editor for two empirical studies investigating the semantic coherence hypothesis, which states that rules or models with semantically similar attributes will be considered to be more understandable by users (Gabriel et al, 2014). We used the editor in both modes. For the Complete Data Mining Workflow Mode, the experiment involved

```
{
    "testId": 15,
    "testName": "zoo PROLIFIC V",
    "testUserId": "bqo5CeIhIw10394707w",
    "operations": [
        {
            "created": "2020-01-31T11:42:53+01:00",
            "message": "User created"
        },
        {
            "created": "2020-01-31T11:44:36+01:00",
            "message": "Save rule",
            "data": {
                "ruleId": 454893,
                "rule": "milk(true) & catsize(true) → class(mammal)",
                "confidence": 28.9998,
                "support": 1
            }
        },
        {
            "created": "2020-01-31T11:57:18+01:00",
            "message": "Remove rule",
            "data": {
                "ruleId": 454918,
                "rule": "milk(false) & domestic(false) & animal(frog) → class(amphibian)",
                "state": "removed"
            }
        }
    ]
}
```

Fig. 5 Excerpt from a log file created by Rule Editor in a test crowdsourcing user study. To enhance understanding by the participants, the initial confidence values were replaced by an integer number expressing the order of the rule in the list. Due to the fact that confidence values are internally saved in terms of a contingency table, the confidence value shown is not integer number 29, but a float approximation. We intend to address this in a future release.

undergraduate students of an introductory data science course. Before the experiment commenced, the students passed a lecture and a hands-on lab exercise on rule learning, classification and the use of EasyMiner software.

In the experiment, participants were randomly assigned to one of the following datasets used in the original study by Gabriel et al (2014): autos, baloons, bridges, flag, glass, hepatitis, primary-tumor, and zoo. While we collected too few valid submissions for a meaningful statistical analysis, the lesson learnt from this study was that the zoo dataset posed the least challenge for the participants. A rule learnt on this dataset is included in Figure 1. As can be observed, the purpose of this dataset is clear; there is no specialised domain knowledge required to interpret the individual attributes.

The second user study was performed using the crowdsourcing mode. Participants were recruited through the `Prolific.ac` platform. To introduce the operation of the rule editor to the participants, we used a short screencast, which is available on EasyMiner website. This explained what a rule list is and how it can be applied to new instances to obtain a prediction.

We avoided explaining rule confidence and support in the crowdsourcing study. For prediction with the CBA algorithm, only the order of rules is important. We thus replaced the values of confidence, which belong to the interval $[0,1]$, with integer values that reflect the order of the rule in the rule list (cf. Figure 5). Participants were instructed that they can change the position of the rule in the rule list by changing the value of confidence.

Several iterations of the crowdsourcing study were performed. In the initial version, we observed the tendency to perform only the easiest operation in the rule editor, which is a removal of the rule.

In the subsequent version, we asked the participants to perform at least several operations of each kind (remove an attribute, add an attribute, change position of a rule in the rule list, delete rule).

Initial inspection of the logs indicates that participants found it difficult to relate the results of the evaluation of a user model (in terms of accuracy) to the actions they performed. For example, the current version of the editor does not easily allow it to undo the last edit operation. A future version of the editor could thus show the effect of each edit operation on model accuracy before the user confirms the edit.

6 Architecture, Availability and Installation

EasyMiner is composed of modules communicating via REST APIs. The main programming languages used for development were Java, Scala and R for the data-mining backend and PHP and JavaScript for the integration component and user interfaces. The data is stored in a relational database such as MySQL or MariaDB. The main repository on GitHub.com is `https://github.com/KIZI/EasyMiner`. The version described in this paper corresponds to release *v2.7*. A screencast of the system is available via a link in the attached supplementary material. The system can be installed either from source code or via docker containers.

7 Conclusion

In this paper, we presented a software framework, consisting of a web-based machine learning interface and a rule editor, which can be used to carry out empirical studies focused on the understandability of rule-based models.

The editor builds upon a proof-of-concept system developed for business rule learning (Vojíř et al, 2014). Compared to this previous release, the codebase was substantially reworked and extended with functionalities supporting empirical user studies. As far as we know, a unique feature of the system is the ability of study participants to change (edit) machine learning models. The system also provides information on the effects of the operation in terms of basic measures of predictive model quality. Besides classification, the framework can also be used for studies involving explorative data mining with association rules.

In future work, we plan to support externally generated rule models. This would allow performing an evaluation of explainability of recently proposed rule learning algorithms, such as Bayesian rule sets (Wang et al, 2016). Also, several usability enhancements would be desirable. In particular, participants could be informed on the effects of an edit on model quality measures already when the edit operation is in progress, allowing them to revert or interrupt the operation when they observe adverse effects on model performance.

Acknowledgements This research was supported by long term institutional support of research activities and grant IGA 33/2018 of the University of Economics, Prague. Author contributions: SV has implemented the system and edited the article, TK has conceived the research, wrote the article and organised the internal user study.

References

Barakat N, Bradley AP (2010) Rule extraction from support vector machines: a review. Neurocomputing 74(1-3):178–190

Boley H, Paschke A, Shafiq O (2010) RuleML 1.0: the overarching specification of web rules. In: International Workshop on Rules and Rule Markup Languages for the Semantic Web, Springer, pp 162–178

Brainard DH, Vision S (1997) The psychophysics toolbox. Spatial Vision 10:433–436

Dalmaijer ES, Mathôt S, Van der Stigchel S (2014) Pygaze: An open-source, cross-platform toolbox for minimal-effort programming of eyetracking experiments. Behavior Research Methods 46(4):913–921

Elkano M, Galar M, Sanz JA, Fernández A, Barrenechea E, Herrera F, Bustince H (2014) Enhancing multiclass classification in FARC-HD fuzzy classifier: On the synergy between n-dimensional overlap functions and decomposition strategies. IEEE Transactions on Fuzzy Systems 23(5):1562–1580

Fernández-Delgado M, Cernadas E, Barro S, Amorim D (2014) Do we need hundreds of classifiers to solve real world classification problems? The Journal of Machine Learning Research 15(1):3133–3181

Fürnkranz J, Kliegr T (2015) A brief overview of rule learning. In: International Symposium on Rules and Rule Markup Languages for the Semantic Web, Springer, pp 54–69

Fürnkranz J, Kliegr T (2018) The need for interpretability biases. In: International Symposium on Intelligent Data Analysis, Springer, pp 15–27, DOI https://doi.org/10.1007/978-3-030-01768-2_2

Fürnkranz J, Gamberger D, Lavrač N (2012) Foundations of rule learning. Springer Science & Business Media

Fürnkranz J, Kliegr T, Paulheim H (2020) On cognitive preferences and the plausibility of rule-based models. Machine Learning pp 853–898

Gabriel A, Paulheim H, Janssen F (2014) Learning semantically coherent rules. In: Proceedings of the 1st International Workshop on Interactions between Data Mining and Natural Language Processing co-located with The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (DMNLP@ PKDD/ECML), CEUR Workshop Proceedings, Nancy, France, pp 49–63

García S, Fernández A, Luengo J, Herrera F (2009) A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Computing 13(10):959–977, DOI https://doi.org/10.1007/s00500-008-0392-y

Grice HP (1975) Logic and conversation. In: Speech Acts, Brill, pp 41–58

Hájek P, Havel I, Chytil M (1966) The GUHA method of automatic hypotheses determination. Computing 1(4):293–308

HLEGAI (2019) Ethics guidelines for trustworthy artificial intelligence. Retrieved from High-Level Expert Group on Artificial Intelligence (AI HLEG), URL https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai

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

Kliegr T, Bahník Š, Fürnkranz J (2018) A review of possible effects of cognitive biases on interpretation of rule-based machine learning models. arXiv preprint arXiv:180402969

Kulesza T, Burnett M, Wong WK, Stumpf S (2015) Principles of explanatory debugging to personalize interactive machine learning. In: Proceedings of the 20th International Conference on Intelligent User Interfaces, Association for Computing Machinery, New York, NY, USA, IUI'15, pp 126–137, DOI 10.1145/2678025.2701399, URL https://doi.org/10.1145/2678025.2701399

Lage I, Chen E, He J, Narayanan M, Kim B, Gershman S, Doshi-Velez F (2019) An evaluation of the human-interpretability of explanation. arXiv preprint arXiv:190200006

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

Liu B, Hsu W, Ma Y (1998) Integrating classification and association rule mining. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, AAAI Press, KDD'98, pp 80–86

Michalski RS (1969) On the quasi-minimal solution of the general covering problem. In: Proceedings of the V International Symposium on Information Processing (FCIP 69)(Switching Circuits), pp

125–128

Miller T (2019) Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence 267:1–38

Muggleton SH, Schmid U, Zeller C, Tamaddoni-Nezhad A, Besold T (2018) Ultra-strong machine learning: comprehensibility of programs learned with ILP. Machine Learning 107(7):1119–1140

Páez A (2019) The pragmatic turn in explainable artificial intelligence (XAI). Minds and Machines pp 1–19

Piltaver R, Lustrek M, Gams M, Martincic-Ipsic S (2016) What makes classification trees comprehensible? Expert Systems with Applications 62:333 – 346, DOI https://doi.org/10.1016/j.eswa.2016.06.009

Rapp M, Mencía EL, Fürnkranz J (2019) Simplifying random forests: On the trade-off between interpretability and accuracy. arXiv preprint arXiv:191104393

Roig A (2018) Safeguards for the right not to be subject to a decision based solely on automated processing (article 22 GDPR). European Journal of Law and Technology 8(3)

Schmid U, Finzel B (2020) Mutual explanations for cooperative decision making in medicine. KI-Künstliche Intelligenz pp 1–7

Sorower MS, Doppa JR, Orr W, Tadepalli P, Dietterich TG, Fern XZ (2011) Inverting grice's maxims to learn rules from natural language extractions. In: Advances in neural information processing systems, pp 1053–1061

Tomanová P, Hradil J, Sklenák V (2019) Measuring users' color preferences in CRUD operations across the globe: a new software ergonomics testing platform. Cognition, Technology & Work pp 1–11

Towell GG, Shavlik JW (1993) Extracting refined rules from knowledge-based neural networks. Machine Learning 13(1):71–101

Vojíř S, Zeman V, Kuchař J, Kliegr T (2018) Easyminer.eu: Web framework for interpretable machine learning based on rules and frequent itemsets. Knowledge-Based Systems 150:111–115, DOI https://doi.org/10.1016/j.knosys.2018.03.006

Vojíř S, Duben PV, Kliegr T (2014) Business rule learning with interactive selection of association rules. In: Patkos T, Wyner AZ, Giurca A (eds) Proceedings of the RuleML 2014 Challenge and the RuleML 2014 Doctoral Consortium hosted by the 8th International Web Rule Symposium, Challenge+DC@RuleML 2014, Prague, Czech Republic, August 18-20, 2014, CEUR-WS.org, CEUR Workshop Proceedings, vol 1211, URL `http://ceur-ws.org/Vol-1211/paper5.pdf`

Wang T, Rudin C, Velez-Doshi F, Liu Y, Klampfl E, MacNeille P (2016) Bayesian rule sets for interpretable classification. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), IEEE, pp 1269–1274

Wason PC (1960) On the failure to eliminate hypotheses in a conceptual task. Quarterly Journal of Experimental Psychology 12(3):129–140

Wason PC (1968) Reasoning about a rule. Quarterly Journal of Experimental Psychology 20(3):273–281

Yang Y, Kandogan E, Li Y, Sen P, Lasecki W (2019) A study on interaction in human-in-the-loop machine learning for text analytics. In: IUI Workshops, CEUR-WS.org, (CEUR Workshop Proceedings), vol 2327

Yin M, Chen Y, Sun YA (2014) Monetary interventions in crowdsourcing task switching. In: Second AAAI Conference on Human Computation and Crowdsourcing (HCOMP), AAAI, pp 234–242

Zilke JR, Mencía EL, Janssen F (2016) DeepRED–rule extraction from deep neural networks. In: International Conference on Discovery Science, Springer, pp 457–473