A Novel Model Usability Evaluation Framework (MUsE) for Explainable Artificial Intelligence

Jürgen Dieber, Sabrina Kirrane*

Institute for Information Systems and New Media,
Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria

Abstract

When it comes to complex machine learning models, commonly referred to as black boxes, understanding the underlying decision making process is crucial for domains such as healthcare and financial services, as well as when they are used in connection with safety critical systems such as autonomous vehicles. As a result, interest in explainable artificial intelligence (xAI) tools and techniques has increased in recent years. However, the user experience (UX) effectiveness of existing xAI frameworks, especially concerning algorithms that work with data as opposed to images, is still an open research question. In order to address this gap, we examine the UX effectiveness of the Local Interpretable Model-Agnostic Explanations (LIME) xAI framework, one of the most popular model agnostic frameworks found in the literature, with a specific focus on its performance in terms of making tabular models more interpretable. In particular, we apply several state of the art machine learning algorithms on a tabular dataset, and demonstrate how LIME can be used to supplement conventional performance assessment methods. Based on this experience, we evaluate the understandability of the output produced by LIME both via a usability study, involving participants who are not familiar with LIME, and its overall usability via a custom made assessment framework, called Model Usability Evaluation (MUsE), which is derived from the International Organisation for Standardisation 9241-11:2018 standard.

Keywords: Machine learning, Explainable Artificial Intelligence, Model Agnostic Explanations, Usability Study, User Experience

1. Introduction

Since the term was first mentioned in 1956 [1], artificial intelligence (AI), and especially its subset machine learning, has steadily made its way into various kinds of industries and aspects of our lives, like healthcare[1][2], transportation[3] and advertisement[4][5]. While machine learning applications are advancing further, the understanding of how machine learning models work and how decisions are made is not advancing at the same pace. In some applications like recommendation systems or predictive maintenance it may not be necessary to understand the black box decision making, as long as the models' predictions are accurate in the majority of cases. However, in circumstances where human lives are involved, like medical diagnosis or self-driving cars, the ability to understand the decision process is essential in order to establish trust in such systems.

In this context, Arrieta et al. [2] defines understandability, as *"the characteristic of a model to make a human understand its function – how the model works – without any need for explaining its internal structure or the algorithmic means by which the model processes data internally."* Efforts made in the field of Explainable AI (xAI) [3] aim to accomplish just that, by building and using models that generate transparency for their users, thus giving a functional understanding of the model [4]. One approach is to develop powerful and fully explainable models, such as deep k-nearest neighbours [5] and teaching explanations for decisions [6], with an explanation being an accurate proxy of the decision maker, used with the aim to create understandability for humans [7]. Another approach is to tackle the issue of model agnostic post modelling interpretability, hence, the ability to explain the meaning to a person [2], by explaining the output of well established machine learning models, instead of replacing these models entirely (cf., LIME by Ribeiro et al. [8], SHAP by Lundberg and Lee [9], and MAPLE by Plumb et al. [10]).

When it comes to xAI frameworks, the Local Interpretable Model-Agnostic Explanations (LIME) framework is, with 5832 citations[6], one of the predominant tools dis-

*Corresponding author
Email addresses: juergen.dieber@gmail.com (Jürgen Dieber), sabrina.kirrane@wu.ac.at (Sabrina Kirrane)

[1] https://www.entrepreneur.com/article/341626
[2] https://medicus.ai/de/
[3] https://kodiak.ai/
[4] https://instapage.com/blog/machine-learning-in-advertising
[5] https://www.ezoic.com/

[6] https://bit.ly/3hcv4eS

cussed in the literature. For instance, one highly cited publication, by Selvaraju et al. [11] (with 5083 citations), remarks that the method on assessing trust in models, proposed by Ribeiro et al. [8], motivated them to use a similar approach to assess their own model. Another prominent example, the interpretability SHAP framework, by Lundberg and Lee [9] (with 3587 citations), bases its computational method on LIME and also uses LIME as a benchmark for their performance evaluation.

Another indicator for LIMEs popularity is their activity on the biggest repository hosting service GitHub[7][8]. From August 2016 to July 2021 the project has been bookmarked (*starred*) over 9000 times, has been copied (*forked*) over 1500 times and has been used by over 1300 GitHub users. 45 researchers and developers have contributed to the project with over 526 approved commits, with the most recent update being made in June 2021.[9]

Although existing publications primarily use LIME as a benchmarking framework in order to assess their tools [12, 13, 14], they do not evaluate the effectiveness of LIME from a usability perspective, hence its explainability. No extensive assessment of its effectiveness from a user experience (UX) perspective has been conducted to date, thus the overarching goal of this work is to close this gap.

Summarizing our contributions, we: (i) demonstrate how LIME can be used to supplement conventional performance assessment methods; (ii) evaluate the understandability of the output produced by LIME via a usability study; and (iii) propose an assessment framework, which is derived from the International Organisation for Standardisation (ISO) 9241-11:2018 standard, that can be used not only to evaluate the usability of LIME but also other xAI frameworks. In addition, our code and data are made available in a GitHub repository[10].

The reminder of this article is structured as follows: Section 2 summarizes the state of the art with respect to post-modelling interpretability. Section 3 compares the performance of several machine learning models using conventional methods. Section 4 illustrates the value LIME adds when it comes to understating the models output in comparison to conventional performance assessment methods. Section 5 evaluates LIME from a usability perspective via a user-study and by analyzing the experience we have had via a self-assessment. Finally, our conclusions and interesting directions for future work are presented in Section 6.

2. A comparative analysis of existing work on model agnostic explainablility

Existing work relating to xAI can be grouped into two distinct categories: (i) the development of fully explainable models (cf., [5, 6]), which are interpretable by design, without using another framework, and (ii) the development of model agnostic explainability frameworks (cf., [8, 9, 10]), which are used on a model to make it more interpretable. Considering that model agnostic frameworks can be used with any machine learning algorithm, in this paper we focus specifically on the latter. In particular, our integrative literature review, which is summarised in Table 1, focuses on comparing and contrasting existing work with respect to the scope of the interpretability, the type of data the method is tested with, and the evaluation used to assess or compare the methods performance.

In terms of the scope of interpretability, a framework can either be on a *global level*, meaning it makes different models comparable with each other, by summarizing their performance with respect to specific indicators, or on a *local level*, giving insight into how a classification in the case of a single prediction is made. Although the vast majority of works focus on local interpretability [34, 35, 45, 19, 41, 40, 21, 32, 39, 44, 30, 31, 27, 12, 16, 24, 20, 3, 18, 42, 9, 13, 28, 10, 8, 22, 17, 26, 33, 38, 46, 14, 36, 37], several can also be used for a global comparison [45, 19, 41, 32, 31, 47, 15, 18, 10, 23, 8, 29, 46]. Only the activation maximization method [15] and model distillation [29] are exclusively global. Although each of the papers includes some demonstration of the method using a specific data type, the actual data used is very different: twenty-four methods are applied to tabular data [35, 45, 19, 41, 21, 32, 44, 30, 31, 12, 16, 24, 3, 18, 42, 13, 10, 23, 8, 22, 17, 38, 29, 14], sixteen are applied to image data [35, 40, 39, 27, 20, 3, 9, 15, 28, 17, 38, 46, 36, 37], and eight are applied to textual data [34, 3, 42, 8, 22, 26, 33, 38]. Only four publications, Koh and Liang [3], Ribeiro et al. [8, 22] and Sundararajan et al. [38] include an application of all three data types.

Concerning the evaluation technique, where an assessment is performed two different methods are used: a *baseline evaluation* and a *user interview*. A baseline evaluation is a quantitative evaluation technique, where one or more indicators are used for a comparative assessment. For instance, Plumb et al. [10] uses a self defined causal local explanation metric to compare their framework to LIME. In total, eight of the publications apply some sort of baseline evaluation [34, 21, 12, 20, 42, 13, 10, 14]. The second evaluation technique is a qualitative method, either a survey or user interview. Only three publications use this approach. Lakkaraju et al. [47] and Lundberg and Lee [9] include a survey in their evaluation and Dhurandhar et al. [21] ask two professionals to rate a mixed set of interpretability framework outputs given to them. Out of the ten publications who evaluate their framework, six draw a comparison to LIME [21, 12, 42, 9, 10, 14], from which we can assume that LIME constitutes a benchmark for interpretability frameworks. However, when it comes to the evaluation of LIME itself, none of the publications actually use evaluation techniques to assess LIMEs performance and only Sokol and Flach [25] evaluate LIME as a demonstration of their novel explainability taxonomy.

[7] https://github.com/

[8] https://thenewstack.io/i-dont-git-it-tracking-the-source-collaboration-market/

[9] https://github.com/marcotcr/lime

[10] https://github.com/jdieber/WhyModelWhy

Method	Reference	Scope	Data Type	Evaluation Technique
Activation maximization	[15]	Global	Image	-
Counterfactual	[16], [17]	Local	Tabular, Image	-
Feature importance	[18], [19]	Global, Local	Image, Tabular	-
Fisher kernels	[20]	Local	Image	*Baseline evaluation:* Fisher kernels compared to Influence functions
Frequency map	[21]	Local	Tabular	*Baseline evaluation:* MACEM compared to LIME *User interview:* MACEM compared to LIME
if-then rules	[22], [23]	Global, Local	Image, Tabular, Text	-
Influence function	[3]	Local	Image, Tabular, Text	-
LIME	[8], [24], [22], [25]	Global, Local	Image, Tabular, Text	-
LIME extension	[26], [27], [12], [14], [28], [13]	Local	Image, Tabular, Text	*Baseline evaluation:* SUP-LIME compared to K-LIME; SLIME compared to positive saliency map; DLIME compared to LIME
MAPLE	[10]	Global, Local	Tabular	*Baseline evaluation: MAPLE* compared to LIME
Model distillation	[29]	Global	Tabular	-
Parametric statistical tests	[30]	Local	Tabular	-
Partial dependence plot	[31]	Global, Local	Tabular	-
Prototype and criticism	[32]	Global, Local	Tabular	-
Ranking models	[33]	Local	Text	-
Relevance scores	[34]	Local	Text	*Baseline evaluation:* LRP compared to TFIDF and uniform
Saliency map	[35], [36], [37], [38], [39], [40]	Local	Tabular, Text, Image	-
Sensitive analysis	[41]	Global, Local	Tabular	-
Shapley value	[9], [19], [42], [43], [44]	Local	Tabular, Text, Image	*Baseline evaluation:* true shapley value, classical shapley estimations, LIME and ES values *User interview:* SHAP compared to true shapley Value, LIME and shapley sampling
Surrogate models	[8], [45], [46]	Global, Local	Image, Tabular, Text	-
Visualisation	[19]	Global, Local	Tabular	-

Table 1: Existing model agnostic explainablility approaches

Model agnostic frameworks have also been applied in several domains. Within the medical sector, considering that AI systems are used to support the diagnosis, both Gale et al. [48] and Katuwal and Chen [24] identify the need to enhance model comprehensibility for the professionals using them. In the case of Holzinger et al. 2019 [49] they go beyond simply explaining the models, towards uncovering causality. Within the field of news detection, the automatic understanding or processing of text, xAI helps to shed light on the multi-layer deep learning applications used for advanced applications [34]. While, in the music business, content analysis is supported by model agnostic interpretability frameworks in order to gain a better understanding of how certain tones are identified [13].

Although the LIME framework[11], especially its image explainer, is one of the predominant tools discussed in the literature, its tabular explainer has received limited attention to date. In addition, existing work focuses primarily on using LIME as a benchmark as opposed to assessing the usability of LIME itself. In order to fill this gap in this paper we apply LIME on tabular machine learning models and evaluate LIMEs performance in terms of comparabil-

[11] https://github.com/marcotcr/lime

ity, interpretability and usability.

3. Using machine learning to classify tabular data

We start by presenting four state of the art classification models, namely decision tree [50], random forest [51], logistic regression [52] and XGBoost [53]. Following on from this, we make use of conventional methods (i.e., the classification report [54] and receiver operating characteristic curve [55]) to assess the model performance and identify the best performing algorithm.

3.1. Tabular data pre-processing

For our tabular data analysis we use the *Rain in Australia* data-set from Kaggle[12]. Before the algorithm is trained, we work through the different variables step by step to fully understand their meaning and make them processable by our model. Given that *RISK_MM* has a 100% correlation with the target variable, it is removed. Other variables with too many missing values are also excluded. A summary of the full dataset is given in Table 2, while the features we use for training are denoted with an asterisk.

From a preprocessing perspective, we modify several categorical variables, making them numeric so they can be processed by the models. We further build a scikit learn `pipeline` object, to apply the preprocessor on the data and sequentially build our model based on its structure. This enables us to perform a sequence of different transformations and to give each algorithm a customised setting while being able to cross-validate each setting-combination during the training process.

The scikit-learn `train_test_split` function is used to break our data into different parts, namely training and testing data. We assign 70% of our observation to the training dataset and the remaining 30% to the testing dataset. Once the data is prepared, we train our four models with the same training data. For comparability reasons, we mainly used standard parameter settings for the setup of the algorithms.

3.2. The application and interpretation of the machine learning models

Our choice of algorithms (i.e., decision trees, random forest, logistic regression and XGBoost) is based on the different levels of interpretability they pose. While the decision tree and the logistic regression are interpretable on their own, the random forest and XGBoost, as examples of ensemble methods, are black box models [8][56] that require interpretation by a framework such as LIME.

In order to analyse the models performance on the testing data, we utilise the sklearn classification report. A model comparison using conventional methods is presented in Table 3. *Precision*, *recall* and *f1-score* are calculated based on the classification results true positive, true negative, false positive and false negative. True positive and true negative both indicate that the weather was correctly predicted with either it is going to rain or it is not going to rain, respectively. A false positive however indicates a class that should not have been predicted positive and false negative indicates that a class should have been predicted positive. The scores next to the metrics name in Table 3 either refer to the target variable that it is not going to rain (0) or that it is going to rain (1) as well as the weighted scores (w) and the training baseline value (tr) for the receiver operating characteristic curves (ROC). Taking the decision tree as an example, the values are then calculated as follows:

Accuracy: The accuracy gives an average of how often the model classified the target variable correctly, in the decision trees example in 79% of the time.

Precision: The precision describes how often the model was correct in classifying an observation as positive, and is therefore also known as the *positive predictive value*. It is the result of the true positives, divided by the sum of false positives and true positives, adding up to 91% for the outcome that it is not going to rain and 53% for the outcome that it is going to rain.

Recall: For the recall measurement, the performance of the variables is more similar. It consists of the true positives divided by the sum of true positives and false negatives, 81% and 73%, respectively. A popular synonym for recall is the *true positive rate*.

F1-score: The f1-score tells us what percentage of positive prediction is correct, including the *recall* and *precision* into its measurement. The *f1-score* consists of two times the *precision* * *recall* divided by the sum of *precision* and *recall*. The decision tree delivers a *f1-score* of 86% for the outcome that it is not going to rain and 61% for it is going to rain.

Macro score: The macro score represents the overall performance of the indicator, meaning the average. The *macro precision* reaches 82%, the *macro recall* 71% and the *macro f1-score* 74%.

Weighted average score: The weighted average is the respective score times its number of instances, for example, the 0.85% *weighted average precision* result from the target variable not going to rain, having a score of 91% and 53% of target variable going to rain, respectively.

Another state of the art tool to measure the validity of classification results is the ROC curve [57]. Figure 1 displays one ROC curve per model, each graph showing two curves, the upper one is the ROC curve, posing a

[12] https://www.kaggle.com/jsphyg/weather-dataset-rattle-package

variable name	sample input	type	non-null-values
Date	2008-12-03	categorical	142193
Location	Albury	categorical	142193
MinTemp*	13.4	numerical	141556
MaxTemp*	25.1	numerical	141871
Rainfall*	0.00	numerical	140787
Evaporation	23	numerical	81350
Sunshine	11	numerical	74377
WindGustDir*	W	categorical	132863
WindGustSpeed*	44.0	numerical	132923
WindDir9am*	NW	categorical	132180
WindDir3pm*	W	categorical	138415
WindSpeed9am*	25.0	numerical	140845
WindSpeed3pm*	8.0	numerical	139563
Humidity9am*	25.0	numerical	140419
Humidity3pm*	22.0	numerical	138583
Pressure9am*	1007.7	numerical	128179
Pressure3pm*	1007.1	numerical	128212
Cloud9am	2.0	numerical	88536
Cloud3pm	8.0	numerical	85099
Temp9am*	16.9	numerical	141289
Temp3pm*	21.8	numerical	139467
RainToday*	Yes	categorical	140787
RISK_MM	0.2	numerical	142193
RainTomorrow*	No	categorical	142193

Table 2: An overview of the datasets' features (Variables used for the training of the models are marked with a *)

probability, the lower one is the *baseline*, which separates the ROC and the area under the curve (AUC), which is a measurement for separability. The ROC curve uses the false positive rate, *fall-out*, and the true positive rate, *recall*, for its measurement. Due to its graphical display, the curves of different models can be easily compared with each other. Each point on the curve represents the relation between *fall-out* and *recall*. The further to the upper left corner the curve bends, the better the classification. The AUC measures the general accuracy, meaning how well a model can differentiate between classes. It provides an aggregated measure of performance across all possible classification thresholds, which makes it a quality indicator for a model's prediction regardless of what threshold is chosen. For the AUC the following rule holds true: the closer its value is to 1, the better the model is able to correctly classify. If the value is 0.5 it means that the model is not better than randomly guessing and a value of close to 0 means that the model is doing the classification upside down[13][14][15]. For instance, in the case of our decision tree, the *baseline* performs with 0.85 on our test-data and the model can therefore be interpreted as reliable.

3.3. An assessment of the machine learning models

Overall it is notable that the performances of the decision tree, random forest and logistic regression are very similar while the XGBoost performance differs significantly. In this comparison, the XGBoost delivers the highest values with a 85% *accuracy*, *weighted average scores* of 85% *precision*, 85% *recall* as well as 84% *f1-score*. But it's weak performance in classifying that it is going to rain correctly, can be seen in a low *recall (1)* and *f1-score (1)* score with 46% and 58%, respectively. It is worth noting that the high difference in the *recall* scores for the respective target variable might be caused by unbalanced testing data, which is something we would like to further explore in future work. The logistic regression offers the highest *recall (1)*, in the case of 77% of the positive observations it predicts correctly that it is going to rain, with a weighted *recall* of 79%. In terms of *f1-score (1)* the logistic regression and the random forest score equal 62% which is four percent higher than the XGBoost with 58%. Furthermore, comparing the ROC curves shows a similar performance for all models, with XGBoost scoring 88% *ROC baseline*, the logistic regression 87%, the random forest 86% and the decision tree 85%, indicating, that all four models are reasonably reliable when it comes to classifying instances correctly.

To summarize, the decision tree performs worst in all metrics. The random forest and the logistic regression never differ more than two percent in any of the metrics and are therefore performing similarly. Although the XGBoost outperforms the others in several metrics, it scores significantly lower when it comes to predicting the outcome of a positive observation. Thus, in order to decide which model should be deployed, based on this results, requires a trade-off: a higher accuracy and more accurate prediction of true negatives would stand in favor of the XGBoost, while the need for a more accurate prediction of true positives would stand in favor of the random forest or the logistic regression. Furthermore, while the confusion matrix and the ROC give us insight into how the models perform, they do not reveal how the models reach a certain decision.

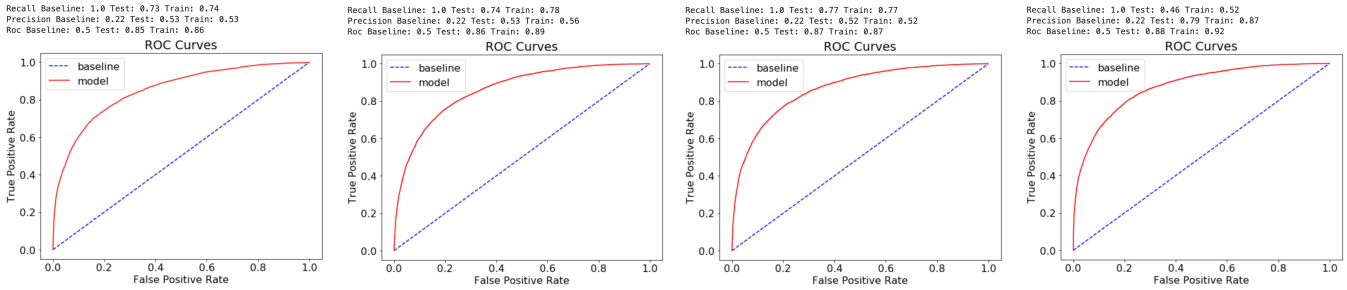
[13] https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/

[14] https://www.jstor.org/stable/2531595?seq=1

[15] https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

model	accuracy	precision (0)	precision (1)	recall (0)	recall (1)	f1-score (0)	f1-score (1)	precision (w)	recall (w)	f1-score (w)	ROC (tr)
decision tree	0.79	0.91	0.53	0.81	0.73	0.86	0.61	0.83	0.79	0.80	0.85
random forest	0.80	**0.92**	0.53	0.81	0.75	0.86	**0.62**	0.83	0.80	0.81	0.86
logistic reg.	0.79	**0.92**	0.52	0.80	**0.77**	0.86	**0.62**	0.84	0.79	0.80	0.87
XGBoost	**0.85**	0.86	**0.79**	**0.96**	0.46	**0.91**	0.58	**0.85**	**0.85**	**0.84**	**0.88**

Table 3: A model comparison using conventional methods



(a) Decision Tree **(b)** Random Forest **(c)** Logistic Regression **(d)** XGB

Figure 1: The ROC curves of the models

4. Applying the LIME xAI framework to tabular data

In order to better understand the behaviour of our four classification models we employ the Local Interpretable Model-Agnostic Explanations (LIME) xAI framework. We start by providing a short introduction to LIME and follow on by applying LIME on our four tabular models and describing the output. Finally, we conduct a quantitative analysis of fifty aggregated LIME observations to further compare performance on a global level.

4.1. A short introduction to LIME

LIME is an open source framework, published by Ribeiro et al. in 2016 [8], which aims to shed light on the decision-making process of machine learning models and therewith establish trust in their usage. LIME is based on the assumption that every model is linear on a local scale. Therefore, it explains individual predictions by creating new, slightly altered data points around the real data and then applies a local linear model on it. In addition, LIME visualises the output, using coloured megapixels on image data and bar charts for tabular and text.
LIME is an acronym for Local Interpretable Model-Agnostic Explanations. *Local* means that the framework analyses specific observations. It does not give a general explanation as to why the model behaves in a certain way, but rather explains how a specific observation is categorised. *Interpretable* means that the user should be able to understand what a model does. Thus, in image classification it shows which part of the picture it considered when it comes to predictions and when working with tabular data it shows which features influence its decision. *Model-Agnostic* means that it can be applied to any black-box algorithm we know today or that we might develop in the future. If the model is a glassbox this is not taken into consideration as LIME treats every model like a blackbox. *Explanations* denote the output, which the LIME framework produces. LIME has three core functionalities: the image explainer interprets image classification models, the text explainer provides insight into text based models[16] and the tabular explainer assesses to what extent features of a tabular dataset are considered when it comes to the classification process[17].

Listing 1: The LIME tabular explainer

```
explainer = LimeTabularExplainer(
    convert_to_lime_format(X_train,
        categorical_names).
    values,
    mode="classification",
    feature_names=X_train.columns.tolist(),
    categorical_names=categorical_names,
    categorical_features=categorical_names.keys(),
    discretize_continuous=True,
    random_state=42)
```

4.2. The application of the LIME Tabular Explainer

The main function that LIME offers is called the `explainer`. As LIME is model agnostic, the explanation happens exclusively on the data level, hence ignoring the

[16] https://www.tensorflow.org/lite/models/text_classification/overview
[17] https://towardsdatascience.com/pytorch-tabular-binary-classification-a0368da5bb89

process within the model. Therefore, the explainer explains predictions on tabular data by perturbing features based on the statistical properties of the training data [58]. A highlevel overview of the LIME explainer is provided below:

The convert to LIME function: Prior to being able to explain an observation, we need to convert the output into a certain format, which we do by creating a list of all possible categorical values per feature. Then, we use the `convert_to_lime_format` function [59] adopted from Kevin Lemagnen's Pycon presentation in 2019[18], as the one included in the LIME documentation only works with older versions of Python. The function converts all existing string variables to integers, such that they can be interpreted.

The explainer: The explainer itself is included in the LIME library and displayed in Listing 1. We set all parameters manually, as the explainer does not possess any default values. First we call our now formatted dataset and set the mode to classification, then we give a list of all features in our dataset (line 3) and with `categorical_names=categorical_names` we specify which of the variables are categorical (line 4), `Categorical_features` (line 5) lists the index of all features with a categorical type and `discretize_continuous` (line 6) is a mathematical function that simply helps to produce a better output by converting continuous attributes to nominal attributes. The final parameter, `random_state`, brings consistency into the function, otherwise it always picks a different number whenever we reload the function.

Displaying one observation: We choose one observation on which we apply the interpretability framework and subsequently print the classification that each model gives for this instance as well as the true label. We can now convert the output to the LIME format, saving it in the observation variable before defining a standard predict function. The `custom_predict_proba` function, is able to transform very simple models but also more complex input. It converts the data so that it is processible by the `LIMETabularExplainer`, which we carry out for every model we wish to interpret. After this we can apply the LIME framework on our classification models. To create a LIME output, we define the explanation as `explainer.explain_instance` and include the observation we chose above, adding the `lr_predict_proba` and five features as this shows us the factors considered the most influential on predicting the target variable.

[18]https://speakerdeck.com/klemag/pycon-2019-introduction-to-model-interpretability-in-python

Running the code presents us with the first of the four LIME outputs, displayed in Figure 2, consisting of four parts: the prediction probabilities on the left side, the feature probabilities in the center, the feature-value table on the right and the r-squared value on the bottom left. The prediction probabilities graph shows the model's decision on that instance, meaning which outcome it predicts and the corresponding probability. In our example it displays the output of the logistic regression and predicts, that it is not going to rain with 92% probability, represented by the blue bar with the number 0 and that it is going to rain with 8%, represented by the orange bar with the number 1. The feature probabilities graph gives insight into how much a feature influences the given decision. For this observation the variable *Humidity3pm* is the most influential factor and supports the prediction, that it is not going to rain tomorrow. The second most important feature is *WindGustSpeed* which weights towards that it is going to rain tomorrow, represented by the number 1. In this case, we display the top five features in our output, but theoretically all the features could be listed that way, ordered by their importance. The last graph is the feature-value table, which also sorts the features by importance, but instead of showing their weight, is given the actual value that this feature possesses in this observation. For example, the forth feature, *Temp3pm*, shows 35.60 in this table, representing 35.60 degrees Celsius, the temperature at 3pm of the day of the observation. It is coloured orange, as it is influencing the model's decision towards rain. The r-squared indicates how well the model fits the observed data and can take a value between 0 and 1, with 1 constituting a perfect fit. For this instance, the value of 0.50 indicates a moderate fit. As demonstrated in Figure 2, LIME does not differentiate between the machine learning model used but displays each of them the same way.

4.3. Evaluating the models on a global level

In order to analyse the LIME output on a global level, we apply the framework on fifty observations. For this we adopt a simple random sampling methodology [60], which is applied by utilizing a random selection function. We then aggregate the output in an excel file to compare the graphs with each other. As we analyse four models, we end up with 200 interpretations in total. Our simple random baseline approach, could be enhanced with more sophisticated sampling mechanisms, such as Submodular Pick LIME (SP-LIME), which can be used to select a diverse yet representative set of explanations.

LIME allows us to look at individual features in more detail and evaluate their influence, the occurrences of the three most relevant features are summarized in table Table 4. In our analysis the framework displays the top five features per observation resulting in 200 total feature counts and 50 top positions per model. Out of this set, *Humidity3pm* occurs most frequently, except for the XGBoost where it is ranked second after *Pressure9am*. It appears 50

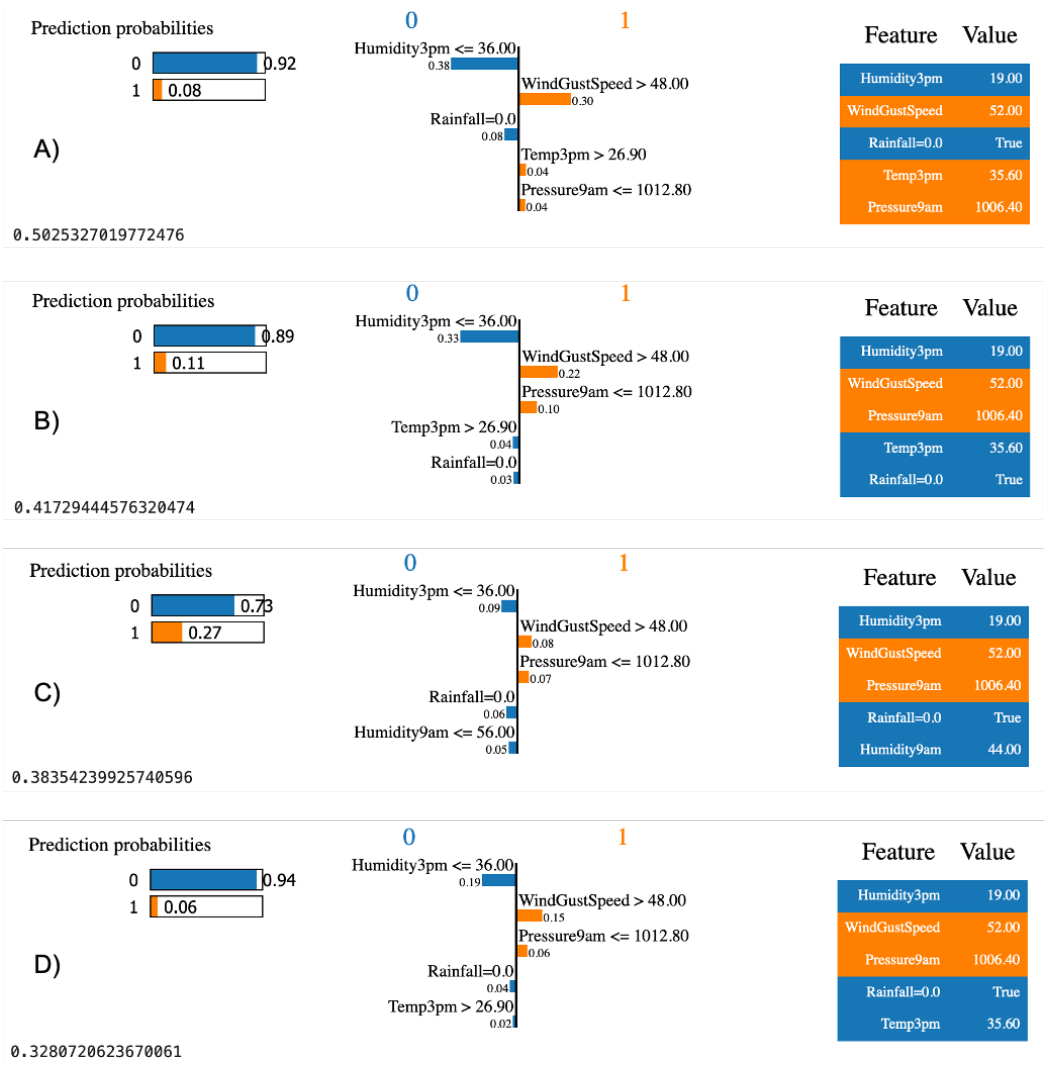


Figure 2: LIME output of the same observation from the (A) Logistic Regression, (B) Decision Tree, (C) Random Forest and (D) XGBoost

times in the analysis of the decision tree and logistic regression, 42 times at the random forest and 48 times at the XGBoost. Furthermore, *Humidity3pm* is not only the most frequent, but is also considered the most important feature, as for the logistic regression it is the most influential feature, meaning it is ranked number one, in all 50 cases and for the decision tree in 42 cases. In case of the random forest, its prediction that it is not going to rain is heavily influenced by `Rainfall`, as whenever it did not rain, it is ranked in first or second position, which happens in 22 and 11 cases, respectively. Nevertheless, *Humidity3pm* is also important for the random forest and occurs in 21 cases on the first rank. In the XGBoost classification *Humidity3pm* is considered the most important feature 38 times. The least considered features are *WindGustDir*, *RainToday* and *Temp9am*, with an occurrence of five, seven and eight times, respectively, none of which are ever ranked within the first or second position. Considering this values, we now know that *Humidity3pm* is highly predictive for our models, bringing us a step closer to developing a usable application.

By displaying the intervals of its classification, LIME enables us to evaluate the accuracy of a single prediction. In terms of a false assessment we calculate the absolute difference between the probabilities assigned to the target variables, measured in percent. This tells us by how much the prediction is wrong and results in another indicator to assess the models. The false classifications are divided into two categories: a wrong prediction with less than 20 percent of absolute difference is called a close miss and a prediction with 20 percent or over more absolute difference is called a far miss. The results are displayed in Table 5. The analysis of all observations results in the following: the decision tree classifies 12 out of 50 instances incorrectly, which are split evenly between close and far misses. The average absolute difference of all wrong classifications is 23 percent. In terms of the amount of incorrect classifications the logistic regression performs better than the decision tree, with eight wrong classifications, of which five are a close and three are a far miss. In absolute differ-

Feature	Decision Tree		Random Forest		Logistic Regression		XGBoost	
	O	TP	O	TP	O	TP	O	TP
Humidity3pm	50	42	42	21	50	50	42	38
Pressure9am	50	4	37	2	20	0	48	5
WindGustSpeed	50	4	29	4	44	11	34	4

Table 4: Summary of the most occurring (O) and highest rated (TP) features

Type	Decision Tree	Random Forest	Logistic Regression	XGBoost
Num. of close Misses (¡ 20%)	6	6	5	1
Num. of far Misses ($\geq$ 20%)	6	3	3	3
Average (in %)	23	15	26	38

Table 5: Summary of close and far misses

ence the logistic regression performs slightly worse, with around 26 percent. The random forest misclassifies nine times, of which six are close and three are far misses and gives us an average of 15 percent. Lastly, the XGBoost predicts incorrectly only four times, one time causing a close miss and three times a far miss, resulting in around 38 percent absolute difference, which is significantly lower in the times of incorrect classifications, but when it fails than by a lot more than other models.

Considering the different evaluations we conducted, XGBoost is superior in the majority of cases. With the highest accuracy of 85%, weighted classification report scores of 85% *precision*, 85% *recall*, 84% *f1-score*, a *ROC-test-baseline* of 88% and the least amount of incorrect classifications, it delivers a better performance than the other models.

5. Evaluating LIME from a usability perspective

After applying LIME on four machine learning models, and testing its local and global functions, we evaluate its usability. This usability assessment consists of two parts: firstly, we perform interviews to get an impression of how LIME is interpreted by people who are not familiar with the concept of explainable AI; secondly, we use a user experience evaluation framework in order to perform a self assessment of LIME's usability based on its criteria.

5.1. The interviews

We interviewed twelve people, equally split between male and female, six with prior knowledge of machine learning, classification models and data modelling, and six with no prior knowledge in these fields. None of them were familiar with the concept of xAI before participating in the interview. The participants were either academics or in the process of pursuing a degree and were chosen for the usability assessment based on the mentioned characteristics. In each interview we wanted to find out how interpretable the LIME output is for a person who has never worked with xAI before. The interviews, which lasted between fifteen and twenty-five minutes, were conducted using the standardised question-catalogue discussed in detail below. An overview of the interview results discussed herein is displayed in Table 6, while the set of anonymous interview notes can in turn be found in our GitHub repository[19].

The interview was split into two sections, both of which started with an explanation from the interviewer. In the first part the interviewees were given a quick introduction into rain prediction, as well as a quick introduction into the applicable machine learning methods. They were subsequently shown the first `LIMETabularExplainer` output graph (cf., Figure 3) and were asked the following four questions.

What do you see in this graph? All interviewees expressed uncertainty about what the illustrations show. All started with identifying the three graphs and tried to make sense of the different numbers. Although a few participants struggled with the prediction-probabilities and the feature-value graph, every participant had difficulties interpreting the feature probabilities as the numbers did not seem to add up and there was too much information given in a badly structured way.

Which feature influences the prediction and how? People without prior machine learning knowledge struggled to see the relation between the prediction probabilities and the classification, but those with prior knowledge in machine learning concluded, that there is a connection between the feature probabilities and the prediction probabilities graph. Five concluded correctly, that the second smaller numbers on the central graph are probabilities, as they are between 0 and 1 and influence the predictability.

Do you know why the model made this prediction? Five out of twelve answered correctly, that the classification is determined by the numbers of the feature probabilities graph.

[19] https://github.com/jdieber/WhyModelWhy

Participant	Prior knowledge	Gender	Understood illustration	Understood prediction	Rating part I	Understanding part II	Rating part II
1	yes	m	yes	yes	3	improved	8
2	no	f	no	no	3	improved	6.5
3	no	m	no	no	4	improved	7.5
4	yes	m	yes	yes	5	improved	9.5
5	no	f	no	no	4	improved	7.5
6	yes	f	no	no	3	improved	7
7	yes	f	yes	yes	8	improved	10
8	yes	m	yes	yes	7	decreased	4
9	yes	m	no	no	5	improved	9
10	no	f	no	no	3	improved	5
11	no	m	yes	yes	4	improved	8
12	no	f	no	no	1	improved	3

Table 6: Summary of the participants' understanding of the LIME output (ratings on a scale from 1-10, increasing)

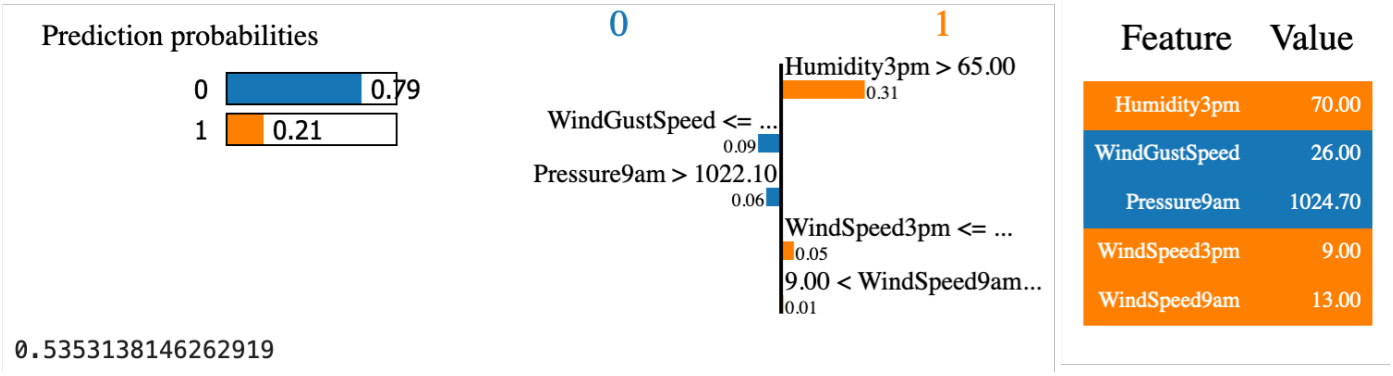


Figure 3: Example of the interview LIME output

How well can you interpret the results of the prediction of the graph, on an increasing scale from 1-10? The interpretability of the LIME output was rated with an average of 4.16. The rating between the subgroups differed significantly, as the participants without prior knowledge gave an average of 3.16 and the participants with prior knowledge 5.16, respectively.

The second section started with a short explanation of each graph of the LIME output as well as an explanation of the meaning of the r-squared value at the bottom of the output. The participants were subsequently shown another LIME output and were asked four more questions.

What do you see in the second graph? After the participants were given the explanation for each graph the answers improved significantly. Seven understood the graphs correctly, but were still uncertain where the probabilities of the prediction probabilities graph came from. Four of the participants with a machine learning background and one without understood the framework after the explanation. Another six pointed out that the r-squared scores of both models were low, which resulted in concerns about the reliability of the prediction.

How well can you interpret the results of the prediction, on an increasing scale from 1-10? Even though several remarks were made in the previous question the interpretability of the graph after the explanation improved significantly, to an average of 7.08. Participants with prior machine learning knowledge again rated it slightly higher with an average of 7.91, compared to an average of 6.25 by the participants without prior knowledge.

What differences do you see between this one and the other graph? All participants noted the different prediction probabilities. Some participants pointed out that there is a big difference on how the features in the different outputs were rated and that the numbers of the feature value graph had changed.

Is there anything that stands out as strange or unusual? Additionally, nine out of twelve participants stated that the central graph was not very interpretable and four mentioned that they found the choice of colours disturbing. Furthermore, six interviewees suggested a legend, titles or a short explanation should be included in the output visualisation to improve its interpretability.

To sum up, the results produced by the framework are difficult to understand without documentation and/or

explanation. Although the participants with a background in machine learning were more effective in terms of interpreting the explanation produced by LIME, usability assessments such as the one described in this paper could be used to significantly improve the user experience.

5.2. Self assessment of the usability

To assess LIME's user experience more broadly, we adopt the definition of usability proposed by the International Organisation for Standardisation (ISO)[20] in their ISO 9241-11:2018 report [61]. Therein, usability is defined as the *"extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use"* [61]. As this definition is too broad to be directly applied in our evaluation context, we improve its applicability by taking into consideration the *"New ISO Standards for Usability, Usability Reports and Usability Measures"* produced by Bevan et al. [62] and the *"Usability Meanings and Interpretations in ISO Standards"* guidelines provided by Abran et al. Abran et al. [63]. Combined they constitute our custom made assessment framework, called Model Usability Evaluation (MUsE).

5.2.1. How effective is LIME in terms of achieving model interpretability?

In terms of effectiveness, Bevan et al. [62] state that *"effectiveness has been associated with completing a task completely and accurately, but it is also important to take account of the potential negative consequences if the task is not achieved correctly"*. From this we extract three effectiveness factors: measure of completion; measure of accuracy; and negative consequences to rate effectiveness. Abran et al. [63] take a more holistic perspective questioning *"how well do users achieve their goal using the system?"*. Thus, we use both the standard and the guidelines in order to develop four UX effectiveness questions tailored specifically to LIME, and subsequently use them to perform our assessment:

(a) How complete is the explanation on a local level? LIME is a local explainability framework, therefore it calculates the influence of every feature and its importance on a local level (i.e., this is done for each prediction). Nevertheless, the connection between the prediction probabilities and the feature probability graph is incomplete as currently only the feature importance score is shown. Additionally, these scores do not add up to the prediction probabilities. As displayed in Figure 3, the feature *Humidity3pm* with a feature probabilities score of 0.31 alone exceeds the total prediction probability of 0.21 that it is going to rain, while the overall classification was in favor of no rain. This can only be explained by assuming that the displayed prediction probabilities are not the sum of the feature probabilities, but the result of another calculation not obvious to a user.

(b) How complete is the explanation on a global level? While LIME is generally used for local interpretability, in this paper we also assess its performance on a global level. It is not surprising that the `LIMETabularExplainer` is less effective globally, as it does not include a function or interface to allow a global evaluation. Thus, we extract several observation outputs manually and analyse them in an Excel file, as we did in the global analysis of Section 5. Considering the importance of global interpretability and the effectiveness of the simple proof of concept presented in this paper, it would be beneficial to: (i) implement performance indicators that allow for a global comparison with other models; and/or (ii) add a function to extract the local outputs of several random observations as a spreadsheet, so the user can calculate indicators necessary for a global comparison themselves.

(c) Could accurate results be misinterpreted? The interpretations of the local predictions appear to be accurate. But we see a risk of misinterpretation when it comes to the tabular explainer, as no comprehensive explanation of it has been published yet [58]. Therefore, we have to rely on third party explanations like online articles[21] [22] or talks on YouTube[23][24]. Ideally such guidance should be incorporated into the LIME documentation.

(d) What negative consequences arise from a misinterpretation? In case of a misinterpretation of the LIME evaluation the severity of the negative consequences depends on the use-case. For example the implication of the predictions produced by our rain prediction model for Australia and an automated defense system [64] differ greatly. In our case a mistake in the interpretation could lead to a faulty feature importance and therefore a wrong rain forecast. In the automated defense system case an incorrect classification could put lives at risk. As the severity of the consequences is not determined by the developers of LIME but rather lies in the hands of the users, reducing the risk that a misinterpretation occurs should be one of the key evaluation criteria when it comes to usability assessments.

[20] https://www.iso.org/home.html

[21] https://medium.com/analytics-vidhya/explain-your-model-with-lime-5a1a5867b423

[22] https://www.oreilly.com/content/introduction-to-local-interpretable-model-agnostic-explanations-lime/

[23] https://www.youtube.com/watch?v=CY3t11vuuOM

[24] https://www.youtube.com/watch?v=C80SQe16Rao

5.2.2. What resources are consumed in order to achieve interpretability?

In order to evaluate resource efficiency Bevan et al. [62] identify the following factors: task time, time efficiency, cost-effectiveness, productive time ratio, unnecessary actions and fatigue. We aggregate them to a list with mutually exclusive components and conclude with the question raised by Abran et al. *"What resources are consumed in order to achieve the goal?"* [63].

(a) How much time does it take to use LIME? Both, the time to set up LIME as well as the time to analyse the output play a role in this context. The setup works well, however the official `LIMETabularExplainer` setup documentation relates to several old packages[25]. Therefore, the initial process of applying the original notebook and trying to find workarounds consumed a lot of time. Additionally, the analysis of the LIME output took a considerable amount of time, as the documentation of the graphs is non-transparent as stated in the effectiveness evaluation. On the up-side, the time it takes to compute and display an observation is minimal.

(b) What other costs are involved? As LIME is an open source tool, no licensing costs are involved and also the publications, documents and videos to understand the tool (where available) are can be freely accessed.

(c) Does this process cause fatigue? Applying LIME to only a few observations can be performed quickly and therefore is not costly from a performance perspective. However, the global interpretation was a tedious process, which entailed hours of repetitive manual work copying and pasting LIME output from the notebook into an Excel file. Also, given that there is no benchmark on the number of observations necessary to evaluate the models globally it is not clear how many outputs are necessary/sufficient.

5.2.3. How satisfying is the application of LIME?

Satisfaction is the least standardised of the three parameters as it is highly dependent on the user and use-case [62]. Based on Bevan et al. satisfaction aims to take *"positive attitudes, emotions and/or comfort resulting from use of a system, product or service"* [62] into account. The question Abran et al. raise to assess satisfaction is *"How well does the user feel about the use of the system?"* [63], which we include in our analysis. Combining both ideas we come up with the following assessment questions:

(a) Do we have a positive or negative attitude towards the tool? At the start of the implementation our attitude was very positive, as LIME's serves to help users to interpret and trust predictions performed by blackbox algorithms. During the setup our attitude deteriorated due to a lack of documentation and support, which posed an even bigger problem during the analysis. LIME gives insight into a model's processes, but here again it takes a lot of effort to get a clear understanding of the framework, which has a negative influence on our attitude. Naturally, once we learned how to apply and interpret LIME, the process was a much more pleasant one.

(b) What emotions arise from using it? The lack of clear and explicit guideline makes understanding LIME a frustrating process. However, reaching the point of a better overall understanding of our classification models raises positive feelings. Especially LIME's short processing time makes it easy to evaluate several instances in a row, which leads to a very pleasant user experience.

(c) How satisfying is the final result? The output of the `LIMETabularExplainer` unquestionably helps to understand the model's classification process, as it offers insights conventional methods can not provide, which causes satisfaction. However, this satisfaction could be increased by eliminating doubt about the relationships between the local indicators and offering a global analysis.

6. Conclusions

Motivated by the lack of limited evaluation of existing post model interpretability tools, in this paper, we evaluated the UX effectiveness of the LIME framework, via both a usability study and a structured self assessment analysis. In particular, we examined the performance of four state of the art classification algorithms on a tabular dataset that is used to predict rain; applied the `LIMETabularExplainer` to analyse single observations on a local level; and used a random sampling approach in order to evaluate the models on a global level. In order to assess the interpretability of the output produced by LIME, we conducted interviews with individuals who had no prior experience with LIME. Whereas, in order to examine the usability of LIME, more generally, we developed a usability assessment framework, Model Usability Evaluation (MUsE), derived from the ISO 9241-11:2018 standard.

Based on our analysis we conclude that LIME could be further enhanced via self explanatory data visualisations, better support for global interpretability, improved documentation, and contextualised accuracy and reliability insights that limit the potential for negative consequences. Additionally, we can conclude that the visualisations provided by LIME is more suitable for users who already have experience working with classification algorithms. Indicating that post model interpretability tools need to consider how best to present their findings to various stakeholder

[25] https://lime-ml.readthedocs.io/en/latest/lime.html

groups (i.e., developers, theorists, ethicists, and users). Some initial insights with respect the the requirements of the various stakeholders are provided by Preece et al. [65] and Tomsett et al. [66]. Taking a broader perspective on usability, there are a number of surveys that focus on usability, from an analysis [67], a design [68], and an evaluation perspective [69] that could provide be used to inform post model interpretability tool enhancement.

When it comes to verification and validation, more generally, there is a need for additional metrics and methodologies that go beyond the baseline evaluations and user interviews that are normally used to evaluate post model interpretability tools. Here researchers have surveyed tools and techniques that can be used to evaluate the effectiveness of machine learning applications [70], expert systems [71], and cyber physical systems [72], to name but a few, that could potentially be used to inform verification and validation for xAI.

From an impact perspective, considering the lack of formal metrics for assessing the effectiveness xAI proposals in general, MUsE, which has been derived from the ISO 9241-11:2018 standard and usability guidelines provided by Bevan et al. [62] and Abran et al. [63], could serve as a means to examine the usability of various post model interpretability tools, and to compare them to one another.

In terms of future work, interviewing experienced LIME users on their user experience with LIME would add another valuable perspective to the usability study. Additionally, an in-depth performance evaluation of LIMEs tabular explainer could close a gap in current research. Besides proposing strategies for improving the interpretability of the output produced by LIME, and the usability of the framework from a global level perspective, we are interested in using MUsE to benchmark alternative model-agnostic explanation frameworks.

References

[1] P. McCorduck, M. Minsky, O. G. Selfridge, H. A. Simon, History of artificial intelligence, in: Proceedings of the 5th International Joint Conference on Artificial Intelligence. Cambridge, MA, USA, August 22-25, 1977, 1977, pp. 951–954. URL: http://ijcai.org/Proceedings/77-2/Papers/083.pdf.

[2] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al., Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information Fusion 58 (2020) 82–115.

[3] P. W. Koh, P. Liang, Understanding black-box predictions via influence functions, 2017. arXiv:1703.04730.

[4] G. Montavon, W. Samek, K.-R. Müller, Methods for interpreting and understanding deep neural networks, Digital Signal Processing 73 (2018) 1–15.

[5] N. Papernot, P. McDaniel, Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning, 2018. arXiv:1803.04765.

[6] M. Hind, D. Wei, M. Campbell, N. C. F. Codella, A. Dhurandhar, A. Mojsilović, K. N. Ramamurthy, K. R. Varshney, Ted: Teaching ai to explain its decisions, 2018. arXiv:1811.04896.

[7] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM computing surveys (CSUR) 51 (2018) 1–42.

[8] M. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?": Explaining the predictions of any classifier, in: In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 97–101. doi:10.18653/v1/N16-3020.

[9] S. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, 2017. arXiv:1705.07874.

[10] G. Plumb, D. Molitor, A. S. Talwalkar, Model agnostic supervised local explanations, in: Advances in Neural Information Processing Systems, 2018, pp. 2515–2524.

[11] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 618–626.

[12] L. Hu, J. Chen, V. N. Nair, A. Sudjianto, Locally interpretable models and effects based on supervised partitioning (lime-sup), arXiv preprint arXiv:1806.00663 (2018).

[13] S. Mishra, B. L. Sturm, S. Dixon, Local interpretable model-agnostic explanations for music content analysis, in: ISMIR, 2017, pp. 100–110.

[14] M. R. Zafar, N. M. Khan, Dlime: a deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems, arXiv preprint arXiv:1906.10263 (2019).

[15] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, J. Clune, Synthesizing the preferred inputs for neurons in neural networks via deep generator networks, in: Advances in neural information processing systems, 2016, pp. 3387–3395.

[16] A.-H. Karimi, G. Barthe, B. Belle, I. Valera, Model-agnostic counterfactual explanations for consequential decisions, arXiv preprint arXiv:1905.11190 (2019).

[17] S. Sharma, J. Henderson, J. Ghosh, Certifai: Counterfactual explanations for robustness, transparency, interpretability, and fairness of artificial intelligence models, arXiv preprint arXiv:1905.07857 (2019).

[18] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, L. Wasserman, Distribution-free predictive inference for regression, Journal of the American Statistical Association 113 (2018) 1094–1111.

[19] G. Casalicchio, C. Molnar, B. Bischl, Visualizing the feature importance for black box models, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2018, pp. 655–670.

[20] R. Khanna, B. Kim, J. Ghosh, O. Koyejo, Interpreting black box predictions using fisher kernels, arXiv preprint arXiv:1810.10118 (2018).

[21] A. Dhurandhar, T. Pedapati, A. Balakrishnan, P.-Y. Chen, K. Shanmugam, R. Puri, Model agnostic contrastive explanations for structured data, arXiv preprint arXiv:1906.00117 (2019).

[22] M. T. Ribeiro, S. Singh, C. Guestrin, Nothing else matters: model-agnostic explanations by identifying prediction invariance, arXiv preprint arXiv:1611.05817 (2016).

[23] N. Puri, P. Gupta, P. Agarwal, S. Verma, B. Krishnamurthy, Magix: Model agnostic globally interpretable explanations, arXiv preprint arXiv:1706.07160 (2017).

[24] G. J. Katuwal, R. Chen, Machine learning model interpretability for precision medicine, arXiv preprint arXiv:1610.09045 (2016).

[25] K. Sokol, P. Flach, Explainability fact sheets, Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (2020). URL: http://dx.doi.org/10.1145/3351095.3372870. doi:10.1145/3351095.3372870.

[26] J. Singh, A. Anand, Exs: Explainable search using local model agnostic interpretability, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019, pp. 770–773.

[27] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, F. Giannotti, Local rule-based explanations of black box decision systems, arXiv preprint arXiv:1805.10820 (2018).

[28] T. Peltola, Local interpretable model-agnostic explanations of bayesian predictive models via kullback-leibler projections, arXiv preprint arXiv:1810.02678 (2018).

[29] S. Tan, R. Caruana, G. Hooker, Y. Lou, Detecting bias in black-box models using transparent model distillation, arXiv preprint arXiv:1710.06169 (2017).

[30] S. García, A. Fernández, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, Soft Computing 13 (2009) 959.

[31] D. P. Green, H. L. Kern, Modeling heterogeneous treatment effects in large-scale experiments using bayesian additive regression trees, in: The annual summer meeting of the society of political methodology, 2010, pp. 100–110.

[32] J. Elith, J. R. Leathwick, T. Hastie, A working guide to boosted regression trees, Journal of Animal Ecology 77 (2008) 802–813.

[33] J. Singh, A. Anand, Model agnostic interpretability of rankers via intent modelling, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 2020, pp. 618–628.

[34] L. Arras, F. Horn, G. Montavon, K.-R. Müller, W. Samek, " what is relevant in a text document?": An interpretable machine learning approach, PloS one 12 (2017).

[35] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, K.-R. MÃžller, How to explain individual classification decisions, Journal of Machine Learning Research 11 (2010) 1803–1831.

[36] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European conference on computer vision, Springer, 2014, pp. 818–833.

[37] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2921–2929.

[38] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 3319–3328.

[39] R. C. Fong, A. Vedaldi, Interpretable explanations of black boxes by meaningful perturbation, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3429–3437.

[40] P. Dabkowski, Y. Gal, Real time image saliency for black box classifiers, in: Advances in Neural Information Processing Systems, 2017, pp. 6967–6976.

[41] P. Cortez, M. J. Embrechts, Opening black box data mining models using sensitivity analysis, in: 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2011, pp. 341–348.

[42] S. Lundberg, S.-I. Lee, An unexpected unity among methods for interpreting model predictions, arXiv preprint arXiv:1611.07478 (2016).

[43] J. Chen, L. Song, M. J. Wainwright, M. I. Jordan, L-shapley and c-shapley: Efficient model interpretation for structured data, arXiv preprint arXiv:1808.02610 (2018).

[44] C. Frye, I. Feige, C. Rowat, Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability, arXiv preprint arXiv:1910.06358 (2019).

[45] O. Bastani, C. Kim, H. Bastani, Interpretability via model extraction, arXiv preprint arXiv:1706.09773 (2017).

[46] J. J. Thiagarajan, B. Kailkhura, P. Sattigeri, K. N. Ramamurthy, Treeview: Peeking into deep neural networks via feature-space partitioning, arXiv preprint arXiv:1611.07429 (2016).

[47] H. Lakkaraju, E. Kamar, R. Caruana, J. Leskovec, Interpretable & explorable approximations of black box models, arXiv preprint arXiv:1707.01154 (2017).

[48] W. Gale, L. Oakden-Rayner, G. Carneiro, A. P. Bradley, L. J. Palmer, Producing radiologist-quality reports for interpretable artificial intelligence, 2018. arXiv:1806.00340.

[49] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, H. Müller, Causability and explainability of artificial intelligence in medicine, WIREs Data Mining and Knowledge Discovery 9 (2019) e1312. doi:10.1002/widm.1312.

[50] J. Morgan, J. Sonquist, Problems in the analysis of survey data, and a proposal, Journal of the American Statistical Association 58 (1963) 415–434.

[51] T. K. Ho, Random decision forests, in: Proceedings of 3rd International Conference on Document Analysis and Recognition, volume 1, 1995, pp. 278–282 vol.1. doi:10.1109/ICDAR.1995.598994.

[52] J. Berkson, Application of the logistic function to bio-assay, Journal of the American statistical association 39 (1944) 357–365.

[53] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[54] S. V. Stehman, Selecting and interpreting measures of thematic classification accuracy, Remote sensing of Environment 62 (1997) 77–89.

[55] J. Fan, S. Upadhye, A. Worster, Understanding receiver operating characteristic (roc) curves, Canadian Journal of Emergency Medicine 8 (2006) 19–20.

[56] S. R. Safavian, D. Landgrebe, A survey of decision tree classifier methodology, IEEE transactions on systems, man, and cybernetics 21 (1991) 660–674.

[57] D. L. Streiner, J. Cairney, What's under the roc? an introduction to receiver operating characteristics curves, The Canadian Journal of Psychiatry 52 (2007) 121–128. URL: https://doi.org/10.1177/070674370705200210. arXiv:https://doi.org/10.1177/070674370705200210.

[58] M. T. Ribeiro, Lime tabular package, https://lime-ml.readthedocs.io/en/latest/lime.html, 2016. URL: https://lime-ml.readthedocs.io/en/latest/lime.html, accessed: 2020-04-18.

[59] K. Lemagnen, helpers.py, https://github.com/charlespwd/project-titlehttps://github.com/klemag/PyconUS_2019-model-interpretability-tutorial/blob/master/helpers.py, 2019.

[60] A. S. Acharya, A. Prakash, P. Saxena, A. Nigam, Sampling: Why and how of it, Indian Journal of Medical Specialties 4 (2013) 330–333.

[61] I. O. for Standardisation, Iso 9241-11:2018(en) ergonomics of human-system interaction — part 11: Usability: Definitions and concepts, 2018. URL: https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en.

[62] N. Bevan, J. Carter, J. Earthy, T. Geis, S. Harker, New iso standards for usability, usability reports and usability measures, in: New ISO Standards for Usability, Usability Reports and Usability Measures, volume 9731, 2016, pp. 268–278. doi:10.1007/978-3-319-39510-4_25.

[63] A. Abran, A. Khelifi, W. Suryn, A. Seffah, Usability meanings and interpretations in iso standards, Software Quality Journal 11 (2003) 325–338. doi:10.1023/A:1025869312943.

[64] D. Gunning, D. Aha, Darpa's explainable artificial intelligence (xai) program, 2019. URL: https://www.aaai.org/ojs/index.php/aimagazine/article/view/2850.

[65] A. Preece, D. Harborne, D. Braines, R. Tomsett, S. Chakraborty, Stakeholders in explainable ai, arXiv preprint arXiv:1810.00184 (2018).

[66] R. Tomsett, D. Braines, D. Harborne, A. Preece, S. Chakraborty, Interpretable to whom? a role-based model for analyzing interpretable machine learning systems, arXiv preprint arXiv:1806.07552 (2018).

[67] A. Følstad, E. Law, K. Hornbæk, Analysis in practical usability evaluation: a survey study, in: proceedings of the SIGCHI conference on human factors in computing systems, 2012, pp. 2127–2136.

[68] E. Folmer, J. Bosch, Architecting for usability: a survey, Journal of systems and software 70 (2004) 61–78.

[69] D. A. Bowman, J. L. Gabbard, D. Hix, A survey of usability evaluation in virtual environments: classification and comparison of methods, Presence: Teleoperators & Virtual Environments 11 (2002) 404–424.

[70] S. Masuda, K. Ono, T. Yasue, N. Hosokawa, A survey of software quality for machine learning applications, in: 2018 IEEE International conference on software testing, verification and validation workshops (ICSTW), IEEE, 2018, pp. 279–284.

[71] R. M. O'Keefe, D. E. O'Leary, Expert system verification and validation: a survey and tutorial, Artificial Intelligence Review 7 (1993) 3–42.

[72] X. Zheng, C. Julien, Verification and validation in cyber physical systems: Research challenges and a way forward, in: 2015 IEEE/ACM 1st International Workshop on Software Engineering for Smart Cyber-Physical Systems, IEEE, 2015, pp. 15–18.