Overview of existing approaches for the interpretation of machine learning models

Akif Cinar
University of Applied Sciences Esslingen
Department of Information Technology
Esslingen, Germany
akciit01@hs-esslingen.de

Abstract—In recent years, machine learning methods have taken a firm place in society and their use continues to grow. The challenge here is their little to almost non-existent interpretability. The aim of this paper is to uncover the possibilities of interpreting machine learning. The novel mechanisms and procedures of the emerging field of interpretable machine learning are presented. In a two-part analysis, intrinsically interpretable machine learning methods and established post-hoc interpretation methods are examined in more detail. The focus is on their functionality, properties and boundary conditions. Finally, a use case will be used as an example to demonstrate how post-hoc interpretation methods can contribute to the explainability of an image classifier and systematically provide new insights into a model.

Index Terms—interpretable machine learning, interpretability, traceability, post-hoc interpretation methods, intrinsically interpretable machine learning

1 Introduction

The use of machine learning methods has grown significantly over the last two decades and has thus taken a firm place in society. Reasons for this are among other things the successes of Google with AlphaGo, IBM with Watson or Amazon with its language assistant Alexa. These successes motivate many companies and organizations to use machine learning methods along their value chain. Business enterprises see a large potential in mechanical learning drives regarding their suitability for automation and data-controlled decision making. The applications are versatile and range from individual product recommendations to the detection of credit card fraud and the decision as to whether a patient should be discharged from hospital.

The increasing speed of innovation of the AI as well as the competition in the economy require the use of more precise models. For this reason, analysts and data scientists are developing increasingly complex machine learning models to meet the demands of the market. With increasing complexity, the interpretability of these models becomes more difficult. The interpretability of the models and their results, however, plays a decisive role for their acceptance, documentation and compliance with legal regulations. Therefore, from a legal, commercial and sociological point of view, it is of great importance to develop interpretable, fair and reliable machine learning models. Trust in AI and thus in machine learning can only be established through interpretable models and modelling results.

With this motivation, the mechanisms and procedures of the emerging field of ”interpretable machine learning” are presented in this paper. First, the necessity of interpretability from a commercial, legal and sociological point of view is explained. The methodology and evaluation of interpretable machine learning will then be defined. Subsequently, the approaches of interpretable machine learning will be examined in more detail. It will be examined whether and to what extent these allow an interpretation and comprehension of the decision-making process of machine learning. In the case of interpretable machine learning, a distinction is made between intrinsic and post-hoc interpretability. Therefore, it is necessary in the analysis to consider these sub-areas separately from each other.

Intrinsically interpretable machine learning procedures are inherent in the system, i.e. due to their internal structure they are considered transparent by nature. In the first part of the analysis, the most common intrinsically interpretable machine learning procedures are explained and examined with a focus on their interpretability. In the second part of the analysis, established post-hoc interpretation methods are presented and analyzed with regard to their functionality, properties and boundary conditions. Finally, their applicability is demonstrated by means of a hypothetical use case.

2 Machine learning interpretability

In the context of machine learning, there is no clear definition of interpretability. Doshi-Velez and Kim [2] define interpretability as the ability to present and explain machine learning models and modelling results for a human being in an understandable way [2]. Interpretibility

is also referred to as the point at which a person can understand the cause of a decision or prediction [3]. The higher the interpretability of a machine learning model, the easier it will be for humans to understand the cause of certain decisions or predictions [4].

The novel approach of interpretable machine learning provides methods and procedures with which the functionality of machine learning models can be understood. Lipton [5] provides a taxonomy for the methods of interpretable machine learning. This elaboration is based on Lipton's view and uses the terms interpretability and explainability synonymously. Furthermore, according to Lipton [5] it is considered reasonable to distinguish between interpretability/explainability and explanation. As in the publications of Doshi-Velez and Kim [2], Miller [3] and Molnar [4], the term "explanation" is regarded as the result of the procedures for post-hoc interpretability. A detailed explanation of the taxonomy can be found in section 2.1.

2.1 Taxonomy of interpretability

Methods of interpretable machine learning are basically divided into two categories: intrinsic and post-hoc interpretability.

2.1.1 Intrinsic and post-hoc Interpretability

The first category, *intrinsic interpretability*, refers to machine learning methods that are considered interpretable due to their simple structure and low algorithmic complexity [5]. These are often referred to as white-boxes. Section 3 is focussed on the most common intrinsically interpretable models.

Post-hoc interpretability includes procedures that can be applied to more complex machine learning models to make them as explainable as possible. As the name suggests, post-hoc interpretation methods are applied after model training. Post-hoc interpretation methods are usually used to explain non-intrinsically interpretable models [5]. Non-intrinsically interpretable machine learning techniques include artificial neural networks, random forests, and non-linear support vector machines. When applying a post-hoc approach, the further procedure depends on which learning algorithm is to be analyzed. The next sections 2.1.2 and 2.1.3 explain the further procedure for the realization of post-hoc explanations.

2.1.2 Model-specific and model-agnostic interpretability

The choice of the appropriate post-hoc interpretation method depends on the machine learning method to be analyzed. Model-specific interpretation methods exist for certain machine learning methods. For example, DeepLIFT is a model-specific interpretation method intended for use on artificial neural networks. Chapter 4.2 explains and examines the most common methods for model-specific interpretation.

If there is no model-specific possibility to interpret certain learning algorithms, a model-agnostic procedure can be used. Model-agnostic interpretation methods can be applied to a number of different learning algorithms and represent the majority of available interpretation methods [7]. Examples are presented in chapter 4.1.

2.1.3 Global and local interpretability

The analysis of a machine learning model with regard to its interpretability requires a distinction between global and local considerations. Interpretations from a global perspective can help to understand the overall behavior of the model as well as the relations between the input parameters and its model predictions. In local interpretation, however, only the predictions of individual or a group of similar data points are explained [7].

For the best possible explanation of a machine learning model, it may be useful to combine the results of global and local methods of interpretation [4].

2.2 Need of interpretability

The ability to explain to others the reasons for their decisions is an important aspect of human intelligence. In addition, the explanation of one's own decisions is often a prerequisite for building a relationship of trust between people. Such social aspects may be of little importance for machine learning systems, but there are enough arguments for interpretability in machine learning [8].

So-called black box models, whose decisions and predictions are intransparent, cannot necessarily be trusted [8]. It is often not sufficient to evaluate the performance of a trained model using a quantitative metric such as accuracy. Accuracy and other performance measures merely provide information on how well the trained model generalizes the learned data or how well it can transfer what it has learned to new data. Key performance indicators cannot say how distorted or unbalanced the training data is with respect to the problem, since a machine learning algorithm only learns from the data and does not evaluate it with respect to the problem. In this way, a bias in the training data can often remain undiscovered [4]. A bias in the training data can be seen as a discrepancy between the problem and the training data. Recognizing bias in the machine learning model or data set becomes easier when you understand how the model behaves and makes its predictions. Especially with critical decision support systems, decision makers need to be able to rely on them and verify the results. Therefore, the use of methods of interpretable machine learning to verify the modelling results makes sense [8]. In this way it will also be possible to identify the weak points of a model and to optimize it.

Another important argument for interpretability in machine learning is information extraction. Today's machine learning systems are trained with millions of samples and can observe patterns in data that cannot be captured by humans. By using interpretable machine learning systems, new insights can be extracted from machine learning models [8]. Extracted information and findings can then be reused.

The need for interpretable machine learning has become urgent. With the increasing use of machine learning, legal aspects, such as the allocation of responsibilities in the event of wrong decisions, have also received increased attention. Particularly when using black box models, it may not be possible to give satisfactory answers to such legal questions. Individual rights play an important role here. People who are directly affected by decisions of a machine learning system will wonder why the system has decided in a certain way (e.g. when refusing a loan from the bank). For this reason, machine learning procedures must inevitably be made more explainable [8]. These concerns prompted the European Parliament to adapt the General Data Protection Regulation (GDPR) and to adopt the "right to explanation" (Article 22 - GDPR). Article 22 GDPR entered into force in mid-2018.

2.3 Evaluation of interpretability

Despite an increasing interest in interpretable machine learning and its necessity, there is no uniform procedure for the evaluation or evaluation of interpretability/explainability or explanations according to the current state of the art. In contrast to a performance indicator such as accuracy, the interpretability of a machine learning model or its results can often be difficult to measure quantitatively [2].

Doshi-Velez and Kim name three ways to evaluate machine learning models and their decisions and predictions in terms of their interpretability. The first two evaluation possibilities follow a human-based approach with partly high domain knowledge. However, this is a scarce and expensive resource that often does not exist. For this reason, this paper makes use of the possibility of function-based evaluation. No people are required to carry out the evaluation. The evaluation is carried out by the developer by the use of use cases. This method is normally used if the model has previously been evaluated by humans for its interpretability [2]. For example, it is generally known that linear and logistic regression models as well as decision trees are more comprehensible than artificial neural networks [4] [7] [9] [10]. In this case, qualitative features can be used to evaluate explanations of post-hoc interpretation methods. The methodology of the post-hoc procedures and the quality of the generated explanations are considered on the basis of different criteria.

Criteria for post-hoc interpretation methods have been established as follows:

- **Transparancy:** Transparency refers to the degree to which the interpretation method describes the functioning of the machine learning model.
- **Transferability:** Transferability describes to how many machine learning methods and models the interpretation method can be applied.
- **Complexity:** Complexity refers to the implementation effort of the interpretation method to generate the explanation.

Generated explanations are considered with regard to the following properties:

- **Plausibility:** Plausibility deals with the correctness of the explanation that the post-hoc interpretation method generated the machine learning model or its decision or prediction. For the evaluation of this criterion, several methods can be applied to a model and then compared.
- **Expressiveness:** Expressiveness refers to the structure of the explanations. Explanations can be represented by histograms, IF-THEN instructions or decision paths.
- **Stability:** Stability considers the similarity of explanations for similar samples.
- **Understandability:** Understandability of an explanation often depends on the viewer and is therefore considered a difficult criterion to measure. For the evaluation of this criterion, one can, for example, consider the scope of the explanation or the number of characteristics depicted in an explanation.

3 Interpretable Models

In the following sections, the most common intrinsically interpretable machine learning methods are explained. The focus is on the interpretation of the model parameters and their mutual influence as well as marginal effects. A marginal effect is the influence of an explanatory variable x on a target variable y if the explanatory variable changes by one unit and the remaining explanatory variables remaining constant [6].

3.1 Linear Regression

Linear regression can be used to model the relationship between one or more explanatory variables and a continuous target variable. Furthermore, linear regression can help to identify trends in the data set and to make predictions for continuous values. Depending on the number of explanatory variables, a distinction is made between two basic types of linear regression. In the case of an xxx, it is also called as

In the case of an explanatory variable, we speak of univariate linear regression. If linear regression is applied to any number of explanatory variables, it is referred to as multiple linear regression. In the interpretation, however, no distinction is made between the two types. The equation of linear regression is defined for a sample i as follows:

$$\hat{y} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j \tag{1}$$

The $\hat{y}$ in equation 1 is the declared target variable or to be precise the weighted sum of the p explanatory variables. The constant β_0 represents the y-interception. In connection with machine learning, β_0 is also called bias. This means

that if all weights of the explanatory variables are zero, then β_0 is the average expected value for y. The β_j represent the determined weights of the explanatory variable x_j. The goal is to determine the weights of the linear regression equation in such a way that they describe the relationship between the p explanatory variables and the continuous target variable and thereby minimize the error. In this way, the value of the target variable can be predicted for other values of the explanatory variable that were not part of the training data set.

The property of linearity makes the linear regression model interpretable. The interpretation of linear regression depends on the explanatory variables and their weights. In univariate linear regression, the weight β_0 is often referred to as the y-intercept. In general, regardless of the number of explanatory variables, β_0 is the average expected value for the target variable if all explanatory variables are zero. If the explanatory variables will not be zero under any circumstances, then there is no meaning in the interpretation of β_0 because it does not contain information about the relationship between the explanatory variables and the target variable y.

The weight β_j of an explanatory variable x_j determines its influence on the model. This means the increase or decrease of the value of a numerical explanatory variable x_j by one unit, increases or decreases the contribution to the target variable $\hat{y}$ by the factor β_j. Unlike a numeric explanatory variable, a categorical explanatory variable has a discrete number of values. If necessary, it is needed to transform the values of a categorical explanatory variable, i.e. to transform the individual categories into a binary format. For this purpose, a column is created in the data set for each category, which can assume the values zero or one. It follows that changing the categorical explanatory variable variable x_j from zero to one, or vice versa, changes the estimate for $\hat{y}$ by the weight β_j of the explanatory variable.

It should also be noted that the weights of the explanatory variables in the regression model influence each other. Therefore, a single weight cannot measure the overall effect on the model. Rather, each weights represents an additional effect on the model.

3.2 Logistic Regression

Logistic regression is an extension of linear regression. It is a linear model that is often used for binary classification problems. Similar to linear regression, logistic regression is the sum of explanatory variables. The target variable is not continuous but categorical. Therefore, logistic regression regards the result as a logistic function. The explanatory variables can be both numerical and categorical. Logistic regression is mathematically defined by the following equation.

$$log\left(\frac{P(y=1|x_j)}{P(y=0|x_j)}\right) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j \tag{2}$$

Logistic regression estimates the conditional probability of a given sample belonging to the class $y = 1$ or $y = 0$ for a given explanatory variable x_j. The probability for the occurrence of the event and thus the assignment of a class is described by the odds ratio. As shown in equation 2, the odds ratio is the ratio between the probability of the occurrence of the event $y = 1$ and its counter-event $y = 0$. The odds ratio is logarithmized to compress the output of the model between zero and one. Mathematically, the logarithm of the odds ratio can be considered as $log(\frac{p}{1-p})$, also called logit function. The p here, is the probability of occurrence for the positive event with the label $y = 1$.

Similar to linear regression, the interpretation depends on the explanatory variables and their weights. However, it differs in interpretation because the outputs of a logistic regression are probabilities in the interval between zero and one. The weights β_j of the explanatory variable x_j determine their influence on the logistic regression model. Depending on the type of explanatory variable, the logistic regression model can be interpreted differently. If the value of a numerical explanatory variable x_j is increased by one unit, the odds ratio ($\frac{p}{1-p}$) changes by the factor exp(β_j).

For a binary categorical explanatory variable x_j, the odds ratio changes by the factor exp(β_j) if the explanatory variable takes the value 1. If the explanatory variable assumes the value zero, this explanatory variable has no influence.

If all explanatory variables, both numerical and categorical, are equal to zero, the odds ratio is at least exp(β_j).

3.3 Decision Trees

There are several algorithms for training decision trees. Most learning algorithms are based on the so-called Hunt's algorithm [11], such as ID3, C4.5 [12] or CART [13] [14]. In practice, the CART algorithm *(Classification and Regression Trees)* is the most commonly used, since it is also used by Scikit-Learn by default. Therefore, this section focuses on the interpretation of the CART algorithm. The interpretation does not differ much for other algorithms, since they are also based on the Hunt's algorithm.

The CART algorithm constructs binary decision trees for regression and classification problems. The only difference is the target variable. The explanatory variables that make up the root node and the inner nodes can be categorical or numerical. The following equation describes the relation between the target variable $\hat{y}$ and the explanatory variables x.

$$\hat{y} = \sum_{m=1}^{M} c_m I(x \in R_m) \tag{3}$$

The $\hat{y}$ in equation 3 represents the target variable. If it is a regression task, the target variable is numeric. For a classification task, $\hat{y}$ assumes categorical values. The x stands for the explanatory variable. R_m represents the subsets of the feature, where M is the number of subsets. Each sample is assigned to exactly one subset R_m. The function $I(x \in R_m)$ is the so called identity, which exactly

returns one argument, 0 or 1. If a sample is an element of the subset R_m, 1 is returned, otherwise 0. This means that the target variable $\hat{y}$ corresponds to the constant c_m. The constant c_m corresponds to the average value of the samples in the training dataset of the subset R_m.

The resulting decision tree is relatively simple to interpret. Starting from the root node, the sample is assigned to the next child node depending on the threshold value. As soon as the sample reached a leaf node, the node can be interpreted as the result for the target variable $\hat{y}$.

The nodes of the decision tree are linked by the boolean operator "AND". Therefore, the resulting decision tree can be considered as a set of consecutive rules (IF-THEN). In this way, the decision path for reproducing the prediction can be identified. Starting from the root node, the decision path can be used to determine which explanatory variables influenced the prediction. If the decision tree consists of a single node or explanatory variable, the target variable will take its mean as the result.

3.4 k-Nearest-Neighbor

The k-nearest-neighbor algorithm belongs to the family of supervised machine learning methods and is suitable for regression and classification tasks. The k-nearest-neighbor algorithm considers the nearest sample or so called neighbor of a new sample for its prediction. In a classification task, the algorithm considers the previously defined k nearest neighbors and their classes. The most common class is assigned to the new sample. In case the majority decision is undecided, the algorithm prefers the label of the next sample with the smallest distance. If the distances of the samples in the neighborhood are the same, the label that appears first in the training data is selected.

For regression problems, the average value of the target variable $\hat{y}$ of the nearest neighbors is used. The following equation describes the relationship between the target $\hat{y}$ variable and the explanatory x variables.

$$\hat{y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \tag{4}$$

The term $N_k(x)$ is the neighborhood of a sample defined by the k closest points x_i in the training samples. The k next samples are determined by a distance measure, such as the Euclidean distance, to form the neighborhood.

The interpretation of the k-nearest-neighbor algorithm differs from the previous learning algorithms because it is an instance-based learning algorithm. This means that there are no parameters or weights that can be learned during a training process. This property of the k-nearest-neighbor algorithm allows interpretation only at the local level, since there are no explicitly learned global parameters or weights. From a local point of view, individual predictions of the k-nearest-neighbor algorithm can be explained by considering the hyper parameter k. The interpretability of the model depends on the interpretability of the individual sample in the data set. The dimension of the sample to be interpreted plays an important role. If a sample consists of several hundreds or thousands of explanatory variables, interpretation is not possible. If there is a manageable number of explanatory variables or the dimension can be reduced to the most important features, good explanations can be obtained from the k-nearest-neighbor algorithm.

3.5 Naive Bayes classifier

Another intrinsically interpretable learning algorithm is the Naive Bayes classifier. The Naive Bayes classifier is a probabilistic learning algorithm that belongs to the family of supervised learning methods. It is designed to solve classification problems.

The Naive Bayes classifier is based on the Bayes theorem. The Bayes theorem makes it possible to determine the probability $P(y|x_1, x_2, ..., x_p)$ of a sample's class y affiliation using the given p explanatory variables of a sample. The Naive Bayes classifier is based on the strong $(= naive)$ assumption of conditional independence. This means that the naive classifier assumes that the effect of an explanatory variable x_j on a given target variable or class is conditionally independent of the values of other explanatory variables. This makes it possible to estimate the conditional probability for each explanatory variable for a given class y. For the classification of an unknown sample i from the test data set, the Naive Bayes classifier generates the conditional probability for each class y:

$$P(y|X) = \frac{P(y) \prod_{j=1}^{p} P(x_j|y)}{P(X)} \qquad X = (x_1, x_2, ..., x_p) \tag{5}$$

The interpretability of the Naive Bayes classifier is given by the assumption of conditional independence. The assumption of conditional independence allows the conditional probabilities to be calculated for each explanatory variable, which allows their contribution to a particular class to be determined.

Another way of determining the contribution of individual explanatory variables is to determine their information gain. According to Kononenko, the Information Gain (IG) can help to determine which of the explanatory variables is most useful for classification [15]. The value of the information content is calculated as:

$$IG(x_j|y) = log_2 P(y|x_j) - log_2 P(y) \tag{6}$$

4 Post-hoc interpretation methods

This section is devoted to the analysis of post-hoc interpretation methods, aimed at ensuring the global and local interpretability of black box models. The post-hoc interpretation methods are examined with regard to their functionality, properties and boundary conditions. In addition, these are evaluated and categorized according to their application spectrum.

4.1 Model-agnostic interpretation methods

Model-agnostic post-hoc interpretation methods are not limited to a specific learning algorithm. They are able to generate explanations for a number of complex learning algorithms. In the following sections, the most common and most cited post-hoc model-agnostic interpretation methods are listed.

4.1.1 Partial Dependence

Partial Dependence (PD) describes the average marginal effect of one or more explanatory variables on the prediction of a machine learning model. This property characterizes this method as a global interpretation method that is applied after model training.

The graphical visualization of the dependencies between the explanatory variables and the model prediction is called *Partial Dependence Plot (PDP)*. A Partial Dependence Plot visualizes whether the relationship between the model prediction and an explanatory variable is linear, monotonous, or complex. For example, a Partial Dependence Plot will always show a linear relation when applied to a linear regression model [4].

Partial Dependence plots can be used to interpret different black box models such as neural networks, non-linear support vector machines or complex random forest models [14]. However, the application is limited to the procedures of monitored machine learning. Furthermore, Partial Dependence plots are limited to the visualization of a maximum of two explanatory variables, since higher dimensional correlations are difficult to capture [16].

The advantage of Partial Dependence Plots is their simplicity of implementation. The Python library for machine learning, Scikit-Learn, offers the possibility to generate Partial Dependence Plots via the module sklearn.inspections with only few parameters.

4.1.2 Global Surrogate Modeling

Global Surrogate Modeling is considered the simplest method to explain and interpret non-intrinsically interpretable models, such as neural networks or support vector machines. A surrogate is the replacement of an object. In this case the replacement of a black box model by an interpretable model.

A *Global Surrogate Model* is an intrinsically interpretable model that is trained to approximate a non-intrinsically interpretable model and thus its predictions [4]. Therefore, this approach is also known as *Behavioral Modeling* [29].

The aim of a surrogate model is to approximate the underlying black box model as accurately and simultaneously as possible. In this way, new insights can be gained about the black box model. The prerequisite for this is the intrinsic interpretability of the surrogate model. This can be any intrinsically interpretable machine learning procedure listed in section 3. Linear regression models or decision trees are frequently used surrogate models. The concept is model-agnostic because it does not require any information about the black box model to be approximated and its structure. Only the training data of the black box model are required [4]. If necessary, a subset of the original training data set can also be used. After training the surrogate model, its quality can be measured by the coefficient of determination R^2. The coefficient of determination R^2 indicates the variance between the predictions of the surrogate model and the black box model. In this way it can be ensured whether the trained surrogate model represents a good approximation of the black box model [4]. Equation 7 represents the coefficient of determination.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(\hat{y}_i^* - \hat{y}_i\right)^2}{\sum_{i=1}^{n} \left(\hat{y}_i - \bar{\hat{y}}_i\right)^2} \tag{7}$$

The $\hat{y}_i^*$ represents the prediction of the surrogate model for a sample *i*. A prediction of the black box model for a sample *i* is represented by $\hat{y}_i$. The mean of the predictions of the black box model is represented by $\bar{\hat{y}}_i$. The result of the coefficient of determination R^2 can also be interpreted as a percentage of the variance. If R^2 is close to 1, the black box model is well approximated by the surrogate model by the surrogate model. This means that the black box model can be replaced by the surrogate model and used for global interpretation. The surrogate model cannot represent the black box model well if R^2 is close to 0 [4].

4.1.3 LIME

Local Interpretable Model-agnostic Explanations (LIME) is a model-agnostic approach for generating local explanations. The LIME library is available in both Python and R. There is no possibility to interpret the global model behavior with LIME. It can provide explanations for the predictions of classifiers or regression analyses. The basic idea of LIME for generating explanations is relatively simple and intuitive. Explanations are generated by locally approximating the underlying black box model by linear regression [19].

First, a series of similar data is generated for the sample to be explained. This creates a new data set whose column values are based on those of the original sample. With this new data set, predictions are again made based on the original black box model. The goal is to observe how the predictions or predicted values change when different values are used for the explanatory variables of the sample to be explained. A linear regression model is formed from the samples that have a similar result to the original sample. The more similar the result of a sample is, the higher its explanatory variables are weighted when training the linear regression model. The weights of the linear regression model are learned using the least squares method. The weights learned by the linear regression model for each explanatory variable represent their contribution to the prediction of the black box model [19].

According to this basic principle, LIME generates explanations for individual predictions of a black box

model. The target variable of the black box model can be categorical or continuous. Basically, LIME supports three data structures. It is suitable for tabular data, images and text data. Depending on the data used, the generation of representative samples used to train the weighted linear regression model differs [19].

For classification and regression problems based on tabular data, the samples to be generated depend on the type of explanatory variable. For continuous explanatory variables, LIME generates normal-distributed data based on the mean and standard deviation of the explanatory variables. For categorial variables, the frequency of values is considered. For image files, the generation of representative samples is different. In the image classification to be explained, the image is divided into interpretable components, called superpixels, using the Quickshift segmentation algorithm. Depending on the number of superpixels in an image, a data set is formed. The number of superpixels in an image determines the dimension of the data set to be generated. Each column of the data set represents one superpixel from the image to be explained. The columns are binary explanatory variables. If an explanatory variable of the sample assumes the value 0, the superpixel represented by this explanatory variable is hidden. Conversely, a 1 means the insertion of a superpixel. Subsequently, the class probability of the generated image samples is calculated. Then the process is repeated to form a linear regression model [19].

The LIME algorithm behaves similarly to images when generating explanations for text classifications. It tries to determine which words in the text are the motives for the specific class assignment. Based on the original text sample, LIME generates similar samples by fading in and out different words [19].

The higher the number of data to be generated, the more accurate and reliable the explanation for interpretation will be. However, a high number of samples requires more resources, since for each of the generated samples a prediction is made by the black box model [19].

4.1.4 Shapley Values

The Shapley Value is an approach from cooperative game theory and is based on the analogy of machine learning models and games. Cooperative game theory is a subfield of mathematical game theory. Cooperative game theory assumes that a group of players, also called a coalition, are the primary decision units and enforce cooperative behaviour. This means that cooperative games can be seen as competition between coalitions of players and not between individual players. Based on their contribution, each player in the coalition receives a certain share of the profits from this cooperation. Shapley values can be used to determine the contribution each player in a coalition makes to the result.

By analogy, it is assumed that each explanatory variable of a sample is a player in a game. The game is the determination of the model prediction for a sample. This can be a classification problem or a regression problem. Thus, the result of the game is the model prediction. Shapley values can be used to determine the contribution of each feature to the predicted model result. Shapley values are the average contributions the explanatory variables have to the model result. To form the average contribution, all coalition possibilities are considered. That is, for the generation of the Shapley values, the model prediction of a sample is permuted by all possible values of the explanatory variables. Therefore, Shapley values are very computationally intensive. For this reason, the complete data set is usually not used for permutation, but only a subset [4].

The interpretation of the Shapley values is based on the average model prediction. The average model prediction is calculated by passing the data set or a specific subset of it. The Shapley Value indicates how the value of an explanatory variable affects the average model prediction. The higher the Shapley value, the greater the contribution of the explanatory variable of the sample to the model prediction. However, Shapley values can also be negative. This means that Shapley values can determine both the positive and negative effects of the explanatory variables on the model result [20].

Inspired by game theory and based on Shapley values, a framework for generating explanations called SHAP has been developed. SHAP (Shapley Additive Explanations) offers a local interpretation possibility for various machine learning models. The SHAP framework is suitable for generating visual explanations for classification and regression problems [20].

4.2 Model-specific interpretation methods

The following sections explain model-specific interpretation methods that have gained importance in the field of interpretable machine learning.

4.2.1 DeepLIFT

Deep Learning Important Features, also known as *DeepLIFT*, is a model-specific post-hoc interpretation method that specifically addresses the local explainability of deep learning algorithms. DeepLIFT tries to find out which input parameters were decisive for the model prediction. To achieve this, DeepLIFT uses the layer architecture and backpropagation mechanism of the neural network. DeepLIFT decomposes the model output of the neural network to certain input parameters. The backpropagation mechanism determines the contributions of individual neurons of the neural network depending on their output. In this way, it is attempted to draw conclusions as to which input parameters influenced the model prediction. DeepLIFT determines the contributions based on the difference between the activation of a neuron and its reference activation. The reference activations of all neurons are determined by propagating the input parameters forward through the net, also called *forward pass*. Once

the reference activation is known, the sample whose explanation is desired is propagated via the output layer back to the input layer by the neural net. This process is also called *backward pass*. The differences in all neurons between the reference activation and the current activation are calculated. Then all effects are summed and mapped to the input parameters [30].

DeepLIFT is implemented in Python and is available via GitHub. It is applicable to neural networks of various architectures based on Keras and Tensorflow [21]. Furthermore DeepLIFT is integrated in the SHAP framework under the name Deep SHAP [22]. The SHAP framework has extended the DeepLIFT algorithm by the concept of Shapley values.

4.2.2 Class Activation Maps

Convolutional Neural Networks (CNN) and other deep neural networks have proven themselves in practice and have enabled numerous breakthroughs in image processing and object recognition. While these deep neural networks master complex tasks, they are difficult to interpret due to their lack of decomposability into comprehensible and understandable components.

Class Activation Maps (CAM) are a simple technique to provide visual explanations for image classifications that allow local interpretation. The basic idea behind CAM is simple. To identify the areas in the image that were critical for classification, the contributions of the neurons are considered. Zhou shows in her publication [24] that CNN with a *Global Average Pooling Layer (GAP)* [17], trained for image classification, can also be used to localize objects in an image. This means that a CNN with a Global Average Pooling Layer after the last convolutional layer can determine the position of the object image in addition to predicting which object is mapped in an image. The decisive factor is that the Global Average Pooling Layer is after the last convolutional layer in the layer architecture. The last convolutional layer is selected because it contains detailed spatial information about the objects in the image. The neurons in these layers search for semantic class-specific information in the image [23]. In this way, the attempt is made to mark the pixels in the image that contributed most to the CNN output.

The feature maps in the last convolutional layer before the GAP act as a kind of pattern detector. Each node in the GAP is connected to a feature map from the previous layer. The weights between the output layer and the GAP determine the contributions of individual feature maps. The Class Activation Map is then generated from the weighted sum of the feature maps. This shows which image areas were used by the network for classification [26]. Red markings in the image represent important areas for the classification. Green areas in the Class Activation Map symbolize less important areas. On the other hand, blue image areas mark less important features for the classification of the image.

A major disadvantage of this method is the requirement of a Global Average Pooling Layer in the architecture of CNN. Therefore, a new approach was developed based on this method, which makes the requirement of a Global Average Pooling Layer superfluous. This new approach uses the backpropagation mechanism of CNN and propagates the model prediction back to the last convolutional layer. Another restriction is that the current implementation is limited to neural networks based on Keras. However, the extended version of the Class Activation Maps can be used for both monitored and encouraging learning [23]. The implementation is freely accessible via GitHub, under the name Keras-Vis [23].

4.2.3 Saliency Maps

Saliency Maps are based on the same idea as Class Activation Maps. They are also a visual explanation method that is especially used in image processing [25]. In the context of interpretable machine learning, this approach is a local model-specific post-hoc interpretation method specifically for the predictions of a convolutional neural network. This means that it can only be used to explain individual images.

This method is derived from the concept of saliency in images. The term Saliency means *conspicuousness* in this context. This refers to unique features, such as pixels or resolution of the image in the context of the visual processing of the Convolutional Neural Network. The goal of Saliency Maps is to identify the characteristics or areas in an image that were relevant to a particular prediction [1]. In contrast to Class Activation Maps, color images are converted into black and white images. This tries to emphasize the strongest influences in the image. Saliency Maps can be generated independently from the architecture of the Convolutional Neural Network. Their implementation is freely accessible via GitHub under the name Keras-Vis [25]. This is the same library that also provides an implementation for Class Activation Maps.

5 Use Cases for post-hoc approaches

In the following sections, selected post-hoc interpretation methods are prototypically implemented using a hypothetical use case. The aim is to demonstrate the applicability of post-hoc interpretation methods and thus to generate explanations. The implementation will be realized on the basis of the analysis from section 4. The selection of the post-hoc interpretation methods to be implemented depends on their suitability for the use cases.

5.1 Interpretability of an image classifier

5.1.1 Use Case

Machine learning models cannot evaluate their training data with respect to the problem for which they are intended. Therefore, distortions or bias in the training data may remain undetected. Machine learning algorithms record these distortions in the training data [4]. As a result, models can emerge that generalize the data provided well, but do not represent the real world sufficiently [19]. Depending on the problem, a bias in the training data can even lead to systematic discrimination.

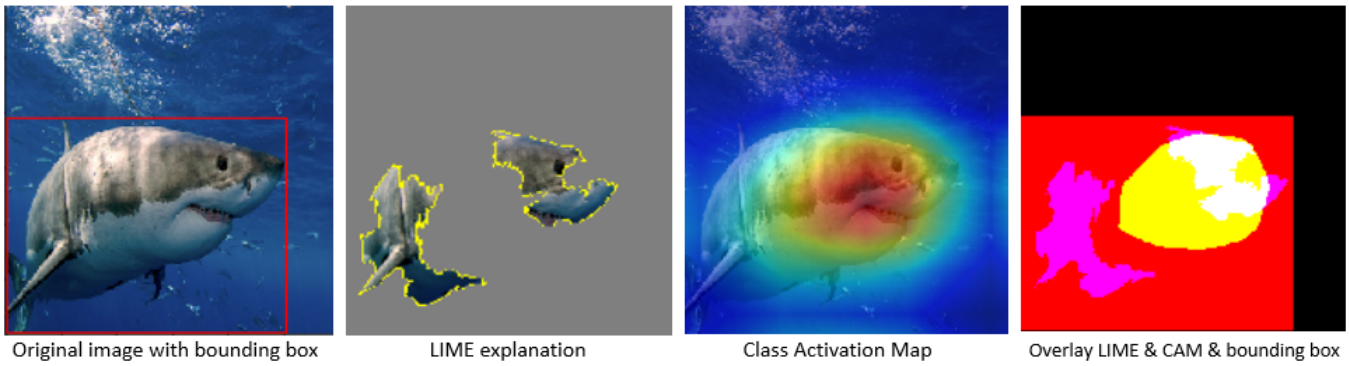

Fig. 1: Approach to identify weaknesses of InveptionV3 using LIME and Class Activation Maps

Many advocates of interpretable machine learning see post-hoc interpretation methods as a way to discover bias and weaknesses in machine learning models. Using post-hoc interpretation methods and their explanations, machine learning models that are non-intrinsically interpretable are to be debugged and their deficits identified. In this way, attempts are made to identify optimization potentials [4] [19].

In this hypothetical use case this assertion shall be investigated. The predictions of the Convolutional Neural Network InceptionV3 [18] for image classification with regard to their weaknesses will be examined using the post-hoc interpretation methods LIME and Class Activation Maps. Both methods support the generation of local explanations for image classifications and thus convolutional neural networks, independent of their architecture.

For the implementation of the use case, images are obtained from the freely accessible online database *ImageNet* [28].

The complete documentation of the implementation is provided on GitHub and can be viewed under the following reference [27].

5.1.2 Systematic analysis for optimization potentials

It is very computationally and time-consuming to examine a model by looking at each explanation individually to make sure that the model has recognized the correct characteristics in the image. Therefore, a more efficient approach is needed to discover optimization potentials in a neural network such as InceptionV3. This section presents such an approach using LIME and Class Activation Maps.

This approach to the systematic analysis of the InceptionV3 with regard to its optimization potential is based on bounding boxes. Bounding boxes are rectangles in an image, which frame the objects to be recognized in an image. The goal is to systematically ensure whether the model actually makes its predictions based on the objects framed in the bounding boxes or whether it concentrates on other areas in the image and thus has a bias. This means that if the areas identified by LIME and Class Activation Maps are not within the bounding box, it may indicate a bias in the training data of the model. The masks of the bounding boxes, class activation maps and LIME explanations are calculated for this purpose. Figure 1 illustrates from left to right the original image, the LIME explanation, the Class Activation Map and their overlapping masks. If one looks at the mask, one sees that in this image classification the masks of the LIME explanation (purple) and the Class Activation Map (yellow) lie within the bounding box (red). This suggests that the image was classified according to the characteristics that were also framed by the bounding box. If the masks of the explanations were outside the red area, this could indicate a bias in the training data of the neural network.

The InceptionV3 was used to determine the classes for 11100 images from 222 different categories. After each classification, the principle shown in Figure 1 was applied. Subsequently, the intersections or number of overlapping pixels of the masks were determined by comparing the pixel coordinates. On the basis of the intersection it should be possible to say what percentage of the LIME explanation or Class Activation Map lies in the bounding box.

For all 222 predicted classes, the average intersection in percent per class was determined. The results were visualized as bar charts and then examined. Figure 2 illustrates a section of the bar charts. Blue columns represent the accuracy in each class. The average intersection between the LIME declarations and the bounding boxes is shown in orange. Green columns visualize the average intersection between the Class Activation Maps and the Bounding Boxes. All values are given in percent. The x-axis represents the coded class names.

The bar chart shows that the explanations of LIME (orange) and Class Activation Maps (green) are predominantly in a similar proportion in the bounding boxes. High values for the explanations in combination with a high accuracy (blue) exclude a bias. A high accuracy in combination with low values for the explanations could indicate a bias in the training data. The values of class n04264628 stand out in the diagram due to very low values in the explanations. The accuracy for this class also performs worse than others. A closer examination revealed that this is the class space_bar.

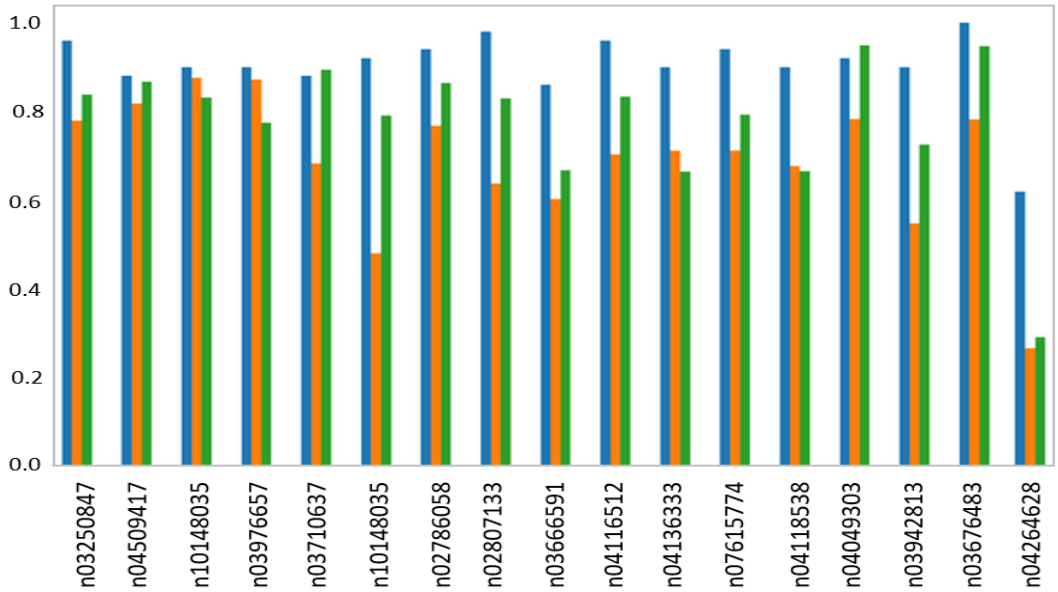

Fig. 2: Average intersections in percent compared to accuracy per class

Figure 3 below illustrates the visual analysis of this class for the first image. The complete output can be viewed in more detail in the Jupyter Notebook [27]. In this edition the background for LIME explanations has not been greyed out for technical reasons.

6 Conclusion

In recent years, the use of machine learning methods has risen sharply and has become an integral part of everyday life. Increasingly complex algorithms are being developed in order to offer the best possible solutions. With increasing complexity, the interpretability of these machine learning algorithms becomes more and more difficult. However, the interpretability of the models and their results is important for their comprehensibility and thus their acceptance. Therefore, it is necessary to achieve the highest possible degree of interpretability in machine learning processes.

With this motivation, the present work aimed at presenting the mechanisms and procedures of the emerging field ”interpretable machine learning” and to investigate them. The necessity of interpretability from a commercial, legal and sociological point of view was explained. Subsequently, the two approaches of interpreting machine learning were examined.

In the first part of the thesis, the most common intrinsically interpretable machine learning methods, such as linear regression, logistic regression or Naive Bayes, were examined. The analysis revealed the basic functioning of intrinsically interpretable machine learning methods. The focus was on the interpretation of the model parameters and their mutual influence.

In the second part of the paper, established model agnostic and model-specific post-hoc interpretation methods were presented and examined with regard to their functionality, properties and boundary conditions. Finally, their applicability was demonstrated by means of a hypothetical use case. In this use case, the neural network *InceptionV3*, which is intended for the classification of images, was examined with regard to its model outputs in order to identify optimization potentials and possible bias in the training data. By a systematic application of the post-hoc interpretation methods LIME and Class Activation Maps deficits in the training data could be determined.

However, for many post-hoc interpretation methods there is still a high degree of optimization potential, since many developers of these methods are not aware of the industrial applications. However, these will become better and better with increasing use. One challenge is the lack of standards and best practices. By the decree of the article 22 of the GDPR, also called ”right to explanation” many developers and large companies like Microsoft or Oracle have started to develop interpretation methods for black box models. Many try to set or enforce a standard. Therefore, the post-hoc interpretation method SHAP offers a good approach, since the post-hoc interpretation methods of this library share the same basis and provide explanations based on Shapley values. However, it is clear that the current approaches, including SHAP, are relatively time-consuming and require a high level of background knowledge. Therefore, an automation mechanism is needed for the targeted and systematic generation of explanations.

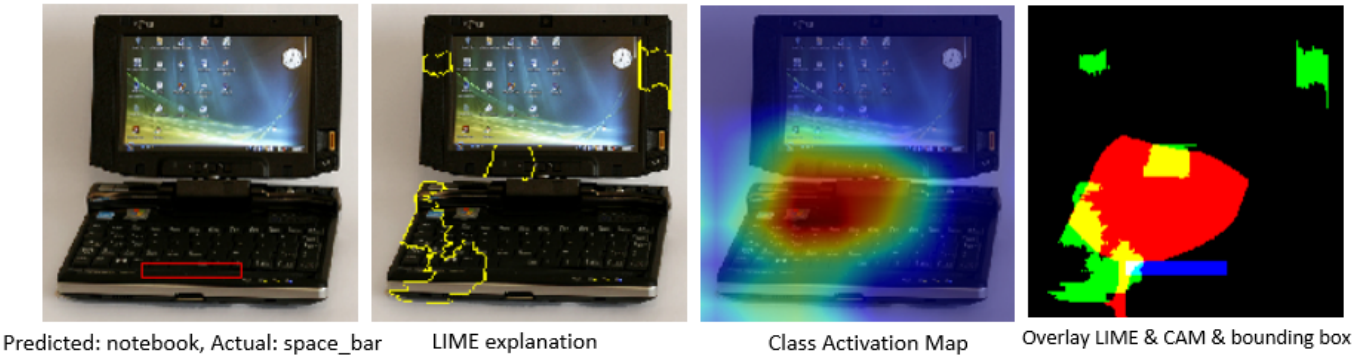

Fig. 3: Identified weaknesses of InceptionV3

REFERENCES

[1] Abishek Sharma (2018). "What are Saliency Maps in Deep Learning?", URL: https:// www.analyticsindiamag.com/what-are-saliency-maps-in-deeplearning/ (Accessed on: 20.08.2019)

[2] Finale Doshi-Velez and Been Kim (2018). "Considerations for Evaluation and Generalization in Interpretable Machine Learning", Springer International Publishing, p. 3 17. DOI: 10.1007/978-3-319-98131-4_1.

[3] Tim Miller (2017). "Explanation in artificial intelligence: Insights from the social sciences", In: Artificial Intelligence 267, p. 138. DOI: 10.1016/j.artint.2018. 07.007.

[4] Christoph Molnar (2019). "Interpretable Machine Learning. A Guide for Making Black Box Models Explainable", URL: https://christophm.github.io/interpret able-ml-book (Accessed on: 17.07.2019)

[5] Zachary Chase Lipton (2016). "The Mythos of Model Interpretability, In: CoRR abs/1606.03490. arXiv: 1606.03490. URL: http://arxiv.org/abs/1606. 03490.

[6] May Boggess (2018). "Methods for obtaining marginal effects", URL: https://www.stata.com/support/faqs/statistics/marginal-effects-methods/ (Accessed on: 11. 06. 2019)

[7] Patrick Hall und Navdeep Gill (2018). An Introduction to Machine Learning Interpretability. Sebastopol, CA: OReilly Media.

[8] Thomas Wiegand and Klaus-Robert MllerWojciech Samek (2017). Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models. In: CoRR abs/1708.08296. URL: http : / / arxiv . org / abs / 1708.08296.

[9] Patrick Hall, Wen Phan and Katie Whitson (2016). "The Evolution of Analytics: Opportunities and Challenges for Machine Learning in Business", Sebastopol, CA: OReilly Media. URL: http://oreil.ly/2DIBefK.

[10] Alex A. Freitas (2014). "Comprehensible classification models", Bd. 15. 1. Association for Computing Machinery (ACM), p. 110. DOI: 10.1145/2594473.2594475.

[11] Janet Marin Earl Hunt and Philip Stone (1966). "Experiments in Induction".

[12] J. R. Quinlan (1986). "Induction of decision trees". Bd. 1. 1. Springer Nature, p. 81106. DOI: 10.1007/bf00116251.

[13] L. Breiman et al. (1984). "Classification and Regression Trees".

[14] Trevor Hastie, Robert Tibshirani and Jerome Friedman (2009). "The Elements of Statistical Learning", Springer New York. DOI: 10.1007/978-0-387-84858-7.

[15] Igor Kononenko (2001). "Machine learning for medical diagnosis: history, state of the art and perspective", Bd. 23. 1. Elsevier BV, p. 89109. DOI: 10 . 1016 / s0933 - 3657(01)00077-x.

[16] Shirin Glander (2018). "Klares Urteil. Erklrbarkeit von Machine-Learning Modellen", Bd. 12, p. 5659.

[17] Keras Documentation (2019). "Keras Pooling Layers". URL: https://keras.io/layers/pooling/ (Accessed on: 19. 08. 2019).

[18] Keras Documentation (2019a). "Models for image classfication with weights trained on ImageNet", URL: https://keras.io/applications/#inceptionv3/ (Accessed on: 19. 08. 2019).

[19] Marco Tulio Ribeiro, Sameer Singh and Carlos Guestrin (2016). "Why Should I Trust You?: Explaining the Predictions of Any Classifier", p. 11351144.

[20] Scott M Lundberg and Su-In Lee (2017a). A Unified Approach to Interpreting Model Predictions. In: Advances in Neural Information Processing Systems 30. Hrsg. von I. Guyon u. a. Curran Associates, Inc., p. 47654774. URL: http://papers .nips.cc/paper/7062-a- unified- approach- to- interpretingmodel- predictions.pdf.

[21] GitHub (2017). "DeepLIFT: Deep Learning Important FeaTures", URL: https://github.com/kundajelab/deeplift#can-you-provide-a-brief-intuit ion-for-how-deeplift-works (Accessed on: 19. 08. 2019).

[22] GitHub (2017a). "SHAP: A unified approach to explain the output of any machine learning model." URL: https://github.com/slundberg/shap (Accessed on: 19. 08. 2019).

[23] Ramprasaath R. Selvaraju et al. (2016). Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization, In: CoRR abs/1610.02391. arXiv: 1610.02391. URL: http://arxiv.org/abs/1610. 02391.

[24] Bolei Zhou et al. (14. Dez. 2015). "Learning Deep Features for Discriminative Localization", arXiv: http://arxiv.org/abs/1512.04150v1 [cs.CV].

[25] Kotikalapudi and Raghavendra (2017). "Keras-vis", https://github.com/raghakot/keras-vis.

[26] Alexis Cook (2017). "Global Average Pooling Layers for Object Localization", URL: ht tps://alexisbcook.github.io/2017/global-average- poolinglayers- for-object-localization/ (Accessed on: 19. 08. 2019).

[27] Akif Cinar (2019). "Interpreting Image Classification of Keras InceptionV3", URL: https://nbviewer.jupyter.org/github/akifcinar/Machine_Learning_Interpretability/blob/master/Interpreting_Image_Classification/Interpreting_Image_Classification.ipynb (Accessed on: 30. 08. 2019).

[28] ImageNet (2019). Image Database ImageNet. URL: http://image-net.org/index (Accessed on: 19. 08. 2019).

[29] Wikipedia (2019). Surrogate Model. URL: https://en.wikipedia.org/wiki/Surrogate_model (Accessed on: 15. 08. 2019).

[30] Avanti Shrikumar, Peyton Greenside and Anshul Kundaje (2017). Learning Important Features Through Propagating Activation Differences. In: CoRR abs/1704.02685. arXiv: 1704.02685. URL: http://arxiv.org/abs/1704.02685.