Explain Variable Influence in Black-box Models through Pattern Mining

Xiaoqi Ma
xiaoqi.ma@rwth-aachen.de
Matriculation number: 383420

Supervisor: Prof. Dr. Markus Strohmaier
Second Examiner: Prof. Dr. Bastian Leibe
Advisor: Dr. Florian Lemmerich

Chair of Computational Social Sciences and Humanities
RWTH Aachen Faculty of Mathematics, Computer Science and Natural Sciences
RWTH Aachen University

This thesis is submitted for the degree of
M.Sc. Media Informatics

Aachen, Germany
4^{th} December, 2019

Abstract

People have a blind belief that complex models can achieve high performance but they fail to explain why the model makes such decisions. Hence, we focus more on the model interpretability rather than the model performance in this thesis, trying to interpret the behavior of black-box models. To tackle this task, we propose an overarching framework to comprehend complex models by inspecting the variable influence. Although existing methods to interpret model predictions relying on the global or local feature influence have been investigated, the expected interpretations from previously studied methods are either too general or too excessive. In response, the meso-level interpretation of black-box models is presented by employing a novel approach that combines the pattern mining technique with the local interpretation method. In specific, the local variable impact is measured for each instance at first, e.g. estimated by LIME or KernelSHAP, then interesting subgroups are detected to show exceptional feature influence through the subgroup discovery approach. It is illustrated that we are able to recover subgroups with extraordinary feature impact from the synthetic dataset. Additionally, three real-world case studies are performed to elaborately analyze black-box models, including neural networks and gradient boosting trees. Our work is relevant for researchers and practitioners interested in interpreting the predictions of black-box models.

Keywords: *Black-box model; Interpretation Framework; Pattern Mining*

Contents

1 Introduction

In recent decades, machine learning fields have been studied extensively. Simply to elucidate, machine learning is a set of methods that are used to teach computers to perform different tasks without hard-coding instructions. It has attracted much attention due to its powerful application, especially in the "Big data" era. Thanks to the boosting computational power, machine learning algorithms can make use of large volumes of data to achieve numerous tasks which are not expected before. For instance, a myriad of classification or regression tasks could be solved efficiently by applying machine learning algorithms. A simple regression task could be predicting the weather temperature by using logistic regression based on historical data, and a more complicated task could look like a language translation problem.

Since there are various kinds of machine learning models, a considerable barrier for human engineers is how to choose the right models for specific problems. Generally, concerning the evaluation of machine learning models, people tend to pay more attention to the model performance rather than the model interpretability. The model performance is definitely very fundamental to assess the model, which typically can be measured by metrics like accuracy, precision, recall, etc. Nevertheless, we should not neglect the importance of model interpretability, which shows "the degree for a human to understand model decisions and the ability to consistently predict the results"[1]. Therefore, one of the major topics to be investigated in the machine learning field is *Interpretable Machine Learning*. It is defined as the use of machine learning models for the extraction of relevant knowledge about domain relationships contained in data. [2]

Broadly speaking, machine learning models can be categorized into white-box models and black-box models judging from the model interpretability. White-box models can be roughly considered as interpretable models, which maintain high model interpretability. Usually, they contain simple structures, a limited number of model parameters, and most importantly, the decisions made by white-box models are interpretable by a human. For example, interpretable models include linear regression, logistic regression, and decision tree model. Those models are human-understandable since the prediction results could be interpreted by examining the model parameters. On the contrary, black-box models usually have more complex structures and a substantial number of parameters which are not intrinsically understandable. Ensemble models or neural networks are normally regarded as black-box models for the reason that decisions made by black-box models cannot be understood by looking at their parameters, which is a major disadvantage for complex models. Typically, those complicated models can achieve better performance for the sake of less interpretability. However, proper model interpretability is crucial to provide explanations to the decisions made by the model and especially important for decision-makers. Besides, "right to explanation", meaning the right to be given an explanation for an output of automated algorithms was stated by General Data Protection Regulation(GDPR), which requires businesses to provide understandable justifications to their users [3]. One scenario is that the bank manager is obligated to clarify reasons to the user about the loan rejection if requested.

Since white-box models are intrinsically interpretable, a more challenging problem that arises in this domain is how to explain the black-box models. In other words, it is of paramount importance to investigate methods to give reasonable explanations to model predictions. Recent theoretical developments have revealed that there are approaches to interpret black-box models, which can be summarized as global interpretation methods and local interpretation methods concerning different viewpoints. As the name suggests, the global interpretation focus on the global view of the input variables while local interpretation is operated on the instance level.

1.1 Purpose of this thesis

The foremost problem we are facing is how to interpret the black-box models. Undoubtedly, many interpretation methods have already come to the surface to facilitate model explanation, but they are not sufficient to deal with complex situations. The global interpretation methods give a too broad interpretation view while local interpretation methods may become too sensitive to reveal the underlying cause due to the excessive interpretation of the target instance. Indeed, the insight gained from a single instance map might be too brittle, and lead to a false sense of understanding [4]. Therefore, our idea is to construct a meso-level interpretation view on the model such that the insight gained from it is more fine-grained than the global interpretation but more general than the local interpretation.

Actually, this thesis aims to develop an overarching framework to provide reasonable explanations for black-box model predictions from multiple model perspectives, given the urgent need to obtain decent justifications for decisions made by the algorithms. To our knowledge, many studies have focused on the local interpretation frameworks but they are just applicable to one type of data or to a specific kind of black-box model, which fails to satisfy our initial purpose. Hence, in our black-box interpretation framework, we will implement diverse approaches that can be employed to various data types and any black-box models. Specifically, we will provide three different interpretation views on the model, i.e. global interpretation, local interpretation, and meso-level interpretation of the model. And in particular, we will devote more efforts to the meso-level interpretation in the framework, which is our main contributions in this thesis.

In fact, the meso-level interpretation is achieved by combining the local interpretation methods with the pattern mining technique. We introduce the subgroup discovery technique to demonstrate the feasibility of discovering patterns that can facilitate us to better understand the feature influence, i.e. identifying patterns in data where a selected variable imposes a significant influence. As an example, assume the adult income dataset is given, and we intend to inspect the influence of gender in a neural network. By employing the meso-level interpretation method to the neural network, patterns could be found out that the attribute gender has an exceptional large impact on persons who are married, indicating by the subgroup description "marital-status=Married". Similar application examples could also be found in later experiments.

In summary, the overall goal of this thesis is to develop a multifaceted interpretation framework to explain the inner behaviors of black-box models from multiple views, which should be furnished with various model interpretation methods.

1.2 Thesis Structure

The remainder of this thesis is structured as follows.

Section 2 focuses on previous work on related fields, such as *Interpretable Machine Learning* and *Subgroup Discovery* field. It starts with the definition of model interpretability, then various of model interpretation methods to explain black-box models are outlined. In particular, it is dedicated to review some existing global interpretation methods as well as local interpretation methods. Subsequently, the fundamentals of the subgroup discovery technique are discussed briefly including the selection of interestingness measure and applied algorithms.

Section 3 is concerned with the approaches in the proposed interpretation framework. In this framework, several model-agnostic local interpretation methods are explored, which includes promising approaches such as LIME and KernelSHAP. Afterwards, a thorough introduction to the subgroup discovery technique is described. Nevertheless, more attention is laid on the proposed novel technique which combines the local interpretation methods and the pattern mining technique to present a meso-level interpretation of the model.

The effectiveness of the black-box interpretation framework is illustrated in Section 4 by conducting experiments on synthetic dataset and empirical dataset. In particular, three case studies are performed to demonstrate the usage of the framework. And the detailed experimental as well as several variations of the experiments are presented.

Finally, Section 5 offers a critical discussion about the topic and Section 6 concludes the work with a summary of results and ideas on future work.

2 Background

It has been stated that there is a trade-off between the model performance and the model interpretability. As declared before, the research goal of the thesis is to examine the interpretability of black-box models and try to explain the model decisions. At first, we will provide an overview to verify key concept–*Model interpretability* (see Section 2.1). It could be further categorized into global interpretability and local interpretability. Accordingly, the global and local interpretation methods are introduced in Section 2.2. Later, a list of existing interpretation frameworks are reviewed in Section 2.3, but there is no single framework could satisfy our desire. After realizing the gap in the current literature, a pristine solution is proposed by mainly introducing a new perspective of model interpretation, i.e. meso-level model interpretation. It is achieved by combining the local interpretation methods (see Section 2.2.2) with the pattern mining technique (see Section 2.4). In addition, *Decision tree* itself is an interpretation model, meanwhile it could also be considered as an alternative of pattern mining technique to discover local patterns, whose background knowledge is covered in Section 2.5.

2.1 Model interpretability

Considerable research efforts have been devoted to the interpretable machine learning area with the pressing need to understand the behaviors of black-box models. In other words, people would like to ascertain why a black-box model makes such predictions. And the extent to explain the model behavior or its predictions in a human-understandable way is termed as interpretability [1]. In this thesis, we will use explainability or comprehensibility as its interchangeable term.

Model explainability can be roughly categorized into two types: intrinsic interpretability and post-hoc interpretability [5]. Intrinsic interpretability refers to models that are inherently interpretable, meaning that their predictions could be explained by model structures and model parameters. On the contrary to that, post-hoc interpretability is achieved by constructing a new model to provide explanations for the black-box model. Particularly, post-hoc interpretability is mainly considered in the current context. Aside from the cognitive definition, it has to be noticed that there is no wide-spread mathematical formula to define or measure the model interpretability. And the assumption that smaller models are more comprehensible than large models concerning the model size is problematic as pointed out by Freitas [6]. More specifically, we might argue that the model complexity is a determining factor to address the measurement of model interpretability. Nonetheless, how to assess the model complexity and how to link these two concepts are beyond our scope.

Basically, models maintaining intrinsic interpretability are interpretable, which are known as interpretable models, including linear models or decision tree models. And it is not surprising that we can easily interpret model predictions through its parameters. For example, we train a logistic regression model to predict the house price. Evidently, we can decompose the house price prediction into the attributions

of each feature, weighted by the coefficients. In this regard, an explanation for this prediction could be inferred from the feature impacts. In contrast, black-box models usually have low comprehensibility, with complex model structures and tremendous parameters. For instance, in the booming field of computer vision, practitioners prefer to apply deep neural networks to achieve sufficient performance with complex neural architectures, training procedures, regularization methods, and hyperparameters. Consequently, it is hardly possible for engineers to interpret the result.

Due to the fact the models with intrinsic interpretability are normally interpretable, we hence devote our efforts to the post-hoc interpretability on black-box models. And it can be further classified as global interpretability and local interpretability [7]. Correspondingly, global interpretation methods and local interpretation methods are introduced as follows.

2.2 Model interpretation methods

The main difference between global interpretability and local interpretability lies in the view of the dataset to be investigated, as displayed in Fig 1. The former highlights the impact of input variables based on the entire dataset, leading to an overall understanding of features. And the latter implies the justification for a specific decision, targeting at the instance level interpretation. There is a number of papers that have imbued explainability in their methodology, and most techniques could be grouped into global interpretation methods and local interpretation methods, respectively.

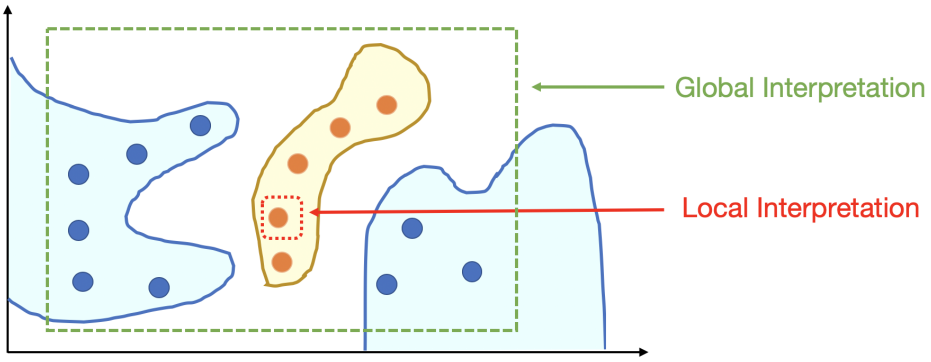

Figure 1: Global interpretation and Local interpretation of a black-box model. Global interpretation explains the feature influence from a global view, based on the whole dataset. Local explanation focuses on the justification for each single instance by inspecting feature impacts (Source from [8]).

2.2.1 Global interpretation methods

The global interpretation methods concentrate on the global view of the input variables, more specifically, they identify the most significant features that can largely

influence the model predictions. Friedman proposed that the *Partial Dependence Plot (PDP)* was a global interpretation method that showed the marginal effect of a feature on the model predictions[9]. This method made clear the relationship between the selected feature and the predictions by adapting the values of the selected feature, and to characterize the feature impact on model predictions. Typically, some simple relationships such as linear or monotonic relation could be inferred from the plot directly.

Another popular approach was called feature importance measure. There were many methods for assessment of feature importance. The default feature importance mechanism was proposed and implemented by the inventor of the *RandomForest* algorithm, which was to add up the Gini decreases for each variable over all trees in the forest and got the average. However, Strobl et al. had demonstrated that this method was biased and was not reliable in scenarios when the selected variable was biased in terms of the scale of measurement[10]. Later, an improved strategy called *Permutation Feature Importance* was described by Fisher et al. [11]. In his approach, the feature importance was estimated by the drop of prediction accuracy of the model after permuting the selected feature. A feature is regarded as "important" if prediction accuracy drops extensively after shuffling feature values as the model depends on the feature for the prediction. Conversely, a feature is "unimportant" if the accuracy is slightly dropped, which means the feature is hardly relied on for the model.

An alternative to permutation feature importance was *SHAP feature importance* measure, estimated by the average feature contributions using SHAP values [12]. To elucidate the idea, we could assume that the model prediction of an individual instance could be decomposed into feature attributions, and each attribution was estimated by the SHAP value. Each feature had a corresponding SHAP value for each instance. Thus, over the entire dataset, the SHAP feature importance was indicated by the mean absolute SHAP values.

2.2.2 Local Interpretation methods

Local interpretation methods aim at the instance level explanation which means each instance should be supplied with an explanation identifying the cause to the prediction. Following this idea, it leads us to the local surrogate methods, which are able to explain individual predictions of any black-box models faithfully. As a concrete implementation of local surrogate models, *Local interpretable model-agnostic explanations (LIME)* was initially proposed by Ribeiro et al. [13], which would be further explored in Section 3.2.3.

Another possible approach was to calculate the individual contribution of each feature in an instance to compose the final prediction as described in paper [14]. Inspired by this idea and the theoretical knowledge from the coalitional game theory, the *Shapley value* approach was highlighted to explain instance-level predictions with contributions of each feature values [15]. Basically, each feature was assigned an importance score for a particular prediction, and the explanation could be derived from feature importance to some extent. More detailed information could be

found in Section 3.2.4.

However, by exploiting the Shapley value approach, it was noticed that only a list of Shapley values corresponding to each feature was generated to form an explanation for each model prediction, rather than an explanation model such as LIME, which failed to make judgments about the connections between input changes and prediction changes. To address those problems, Lundberg and Lee [16] proposed a unified framework for explaining predictions, which was based on the Shapley value, and it was named *SHAP (SHapley Additive exPlanations)*. In this unified framework, there was a novel approach called KernelSHAP, which was the combination of linear LIME and Shapley values. In this way, the intuitive connections between these two methods made this approach more promising, which will be intensively investigated in Section 3.2.5.

2.3 Existing interpretation frameworks

In previous part, the interpretation methods are classified from the interpretation views. Apart from that, Alvarez-Melis concluded that they could also be roughly categorized as salience-based and perturbation-based approaches by observing the traits of those interpretation methods [4]. The former method group was also known as gradient-based attribution method, which computed the partial derivatives of the output with respect to each input feature, e.g. *Integrated Gradients* [17][18]. In contrast, perturbation-based approaches first generated a bunch of neighborhood data points surrounding the instance to be explained, then calculated the contribution of each input features towards the output by fitting a local interpretable model, e.g. *LIME* [19].

Based on the two aforementioned method categories, a list of black-box interpretation frameworks were surfaced. Take an example, DeepExplain was a unified framework of perturbation and gradient-based attribution methods for understanding deep neural networks. The detailed description of this framework could be discovered in paper [20], and the key algorithm in this framework was presented by Marco [21]. Another frequently mentioned interpretation framework was LIME, which supported explaining the predictions of any machine learning classifiers, founding on the perturbation-based approach. And the concrete implementation of LIME was accessible online [22]. More recently, another promising interpretation framework called SHAP emerged, which connected the game theory with the local interpretation to provide understandable explanations for any black-box models. It was accessible and open-sourced online [23].

After seeing these frameworks, it could be easily observed there are huge drawbacks for each single framework. DeepExplain was mainly designed for image classifier that was usually a deep neural network model. Therefore, it failed to apply this framework on tabular data or textual data. In contrast, LIME was applicable to tabular data, textual data, and image data, but the interpretation methods for complicated models such as deep neural network in LIME framework were not well implemented. And another critical problem was caused by the unstable explanations using LIME

framework. Besides, even though SHAP was considered as a promising unified interpretation framework, the problem concerning the computational complexity was still needed to be tackled. What is worse, none of those frameworks included the global interpretation view on the black-box models.

Normally, the existing interpretation frameworks only focus on the local interpretation of the model. However, it is not sufficient to fully understand the model due to its excessive interpretation. Meanwhile, the global explanation methods could only provide a broad interpretation view. Therefore, the idea is to construct a new framework that can not only absorb the advantages in global or local interpretation methods, but also create a novel perspective to inspect the model, called meso-level interpretation. It can be accomplished by combining the pattern mining technique with the local interpretation methods. And the pattern mining technique was reviewed in the following section.

2.4 Subgroup discovery technique

Developments in knowledge discovery field had attracted much attention, where numerous methods were proposed to extract local patterns from large volumes of data [24]. Apart from the methods for mining local patterns such as discriminative patterns [25] and emerging patterns [26], *Subgroup Discovery (Pattern Mining)* was established as a supervised and descriptive data mining technique. As defined in [27], in the subgroup discovery task, assuming we have a population of individuals and the corresponding property of interest, it aims to discover subgroups that are statistically "most interesting". To put it another way, the interesting subgroups have the most unusual distributional characteristics concerning the certain property of interest given by the target variable [28].

In a formal definition, the fundamental concepts of the subgroup discovery task could be summarized by a quadruple (D, Σ, T, Q) [29]. In the quadruple, D represents the dataset, which is formed by a group of instances. Σ means the search space, consisting of a set of selection expressions, and the search space covers all the patterns that are going to be traversed through. Take an example, one of the selectors could look like: "sex=Male AND age>30". T implies the target concepts being exploited in the pattern mining task. Commonly, a single target concept, e.g. binary or numeric, is applied to the mining task. Nevertheless, multi-target concepts are also allowed given the existence of an exceptional model mining framework [30]. Concerning the quality measure criteria, symbolized as Q, it is specified depending on the target concept.

2.4.1 Interestingness measure

The interestingness measure were intensively studied by many researchers because the criteria determined the discovered patterns. Since considerable research efforts have been devoted to study the binary target concept, the quality measure for the binary target was well-investigated. One variant of the quality measure for the binary

target could be easily estimated by the parameters contained in a contingency table, which described the distribution of positive/negative instances for the observed pattern and its complement subgroup, respectively. According to an investigation by Kloesgen et al. [31], they proposed a prevalent family of interestingness measure for binary target, relating to the size of the subgroup and the difference between the target share in the subgroup and the target share in the general population.

Correspondingly, several criteria to measure the quality of numeric attributes had been proposed, which could be found in paper [32]. Since a numeric attribute has certain characteristics, such as mean value or median value, therefore, the quality measure for a numeric target could be formalized by slightly adapting the quality function that was designed for the binary target. To be specific, chances were that the share of the target in the subgroup and in the entire population could be replaced by the characteristic of the target. Generally, there were five categories of interestingness measure for numeric target, concluded by Lemmerich [29], which were mean-based measures, median-based measures, variance-based measures, distribution-based measures, and rank-based measures. Furthermore, as for a multi-target concept, the quality function had been described in a number of studies. And a general framework for multi-target quality functions was the exceptional model mining framework reported by Leman et al. [30], proposing a variety of model classes, which contained the correlation model, the regression model, and the classification model class.

2.4.2 Subgroup discovery algorithms

Unlike the choice of the quality measure which is mainly determined by a target concept in the subgroup discovery task, the mining algorithms are almost equivalent. And for a specific algorithm, three algorithmic components should be verified, which are enumeration strategy, data structure, and pruning strategy. Various enumeration strategies could be used, e.g. exhaustive methods, seeking to acquire the optimal subgroup by traversing through the whole search space. In contrast, heuristic approaches, normally a beam search strategy, was often used for subgroup discovery due to its efficiency, which aimed to find interesting patterns but not necessarily the optimal patterns in a short time [33]. From data structure perspective, data was normally stored in a horizontal layout, e.g. tabular-formatted database. Instead, vertical data representations could also be used, which was covered in paper [34]. Besides, referring to the wide-spread FP-Growth algorithm, the FP-tree structure was also applicable to data [35]. Furthermore, considering the efficiency of algorithms, the pruning strategies were of critical importance. To determine the upper-quality bounds and safely prune parts of the search space, optimistic estimates could be explored as initially stated by Wrobel et al. [36]. In addition, to shrink the search space of the subgroup discovery task, minimal support pruning strategy was useful by exploiting anti-monotone constraints [37].

2.5 Decision trees

Decision trees [38] have been widely discussed in machine learning field [39]. It is a supervised machine learning method used for classification and regression tasks with high performance, computational efficiency. and flexibility. It is called "Decision tree" because the structure of each decision path is a tree-like graph and the model is constructed to predict the target by learning simple decision rules inferred from data attributes. In literature, numerous algorithms to create decision trees had been proposed and the main difference laid in the attribute split criteria [40]. The standard algorithms to grow a decision tree were greedy, which means the decision tree was built recursively by optimizing the split function one node at a time from top down. For example, the ID3 algorithm tried to maximize the information gain of the split attribute while the CART algorithm employed the Gini index as the split metric. However, pointed out by Nowozin [41], the common greedy algorithms were biased depending on the scale of the attribute values. And he demonstrated an improved information gain estimation approach for decision tree induction with enhanced estimators of the discrete and the differential entropy. Yet the predictive performance was increased, the problem still held that high chances were given that the trees grown by greedy algorithms were locally optimal.

To address this issue, non-greedy decision tree induction strategies were also proposed over the last decades. Bennett initially developed a non-greedy algorithm for constructing globally optimal decision trees [42]. Instead of splitting the attribute on each decision node once a time, the presented approach considered all decisions in the tree concurrently to minimize the error of the entire tree. But it was only designed for binary classification tasks with 1-0 loss functions. In addition, learning the parameter of decision tree growth could be treated as a maximum likelihood problem. In particular, an Expectation-Maximization (EM) algorithm was employed to globally tune the parameter of the decision tree [43]. More recently, Norouzi et al. transformed the decision tree induction problem as the optimization task. Hence, a well-characterized optimization framework for training decision trees in a non-greedy way was developed. In this algorithm, the upper bound for the training loss was optimized using stochastic gradient descent such that the training loss for the entire tree was also minimized globally.

Meanwhile, the decision tree itself is an interpretable model that can provide human-understandable decisions by exploring the decision rules [44]. In addition, as an alternative to the subgroup discovery technique, the decision tree algorithm is also capable to mine local patterns through its decision path, where each path is traversed from the root node to a leaf node. But the major difference is that there might be a huge overlap in the antecedents across the decision path, e.g. the attribute "age" might used multiple times as the splitting node, leading redundancy in the decision patterns. Besides, decision trees could also be applied in face recognition field. Commonly, features of the image could be characterized as local binary patterns and a decision tree based approach was presented to discover the the most discriminative features for each facial region [45].

3 Approach

Given a tabular or textual dataset and a black-box model that is trained on the provided dataset, the primary goal of this thesis is to develop a multi-faceted interpretation framework that facilitates non-expert to comprehend the black-box model through investigating the feature influence. Therefore, it can be formalized as a model interpretation problem, and the pipeline in this framework to cope with the task is presented in Fig 2. Subsequently, the building blocks in our proposed interpretation framework will be illustrated.

To begin with, the structure of the black-box interpretation framework is described (see Section 3.1). It provides multiple views to comprehend the black-box model, i.e. global, meso-level, and local interpretations. And our attention has been paid to the local interpretation methods at first (see Section 3.2), since the global interpretation is too general to provide specific explanations. The local interpretation methods contain some variable-specific approaches, such as binary flip or numeric perturbation approach. In addition, LIME and KernelSHAP approaches are intensely investigated due to their appealing properties. Afterwards, the pattern mining technique is discussed thoroughly. Then, we propose the novel technique that combines the local interpretation methods with the pattern mining technique in our black-box interpretation framework (see Section 3.4.1). More importantly, it provides a model interpretation view that is more fine-grained than global interpretation but more general than local interpretation. In a variation of the meso-level interpretation, the decision tree algorithm can also be employed to discover local patterns instead of applying the subgroup discovery approach (see Section 3.4.2).

3.1 Structure of black-box interpretation framework

As seen from Section 2.3, the existing model interpretation frameworks alone are not sufficient to fulfill the desire to explain any black-box models and inspect multiple data types. For instance, DeepExplain is only limited to explain the black-box models trained on image data but not on tabular data or textual data. Although the LIME approach is suitable for the mentioned three data types, the explanation is not stable as expected, and it is not fully supported for deep neural networks. Moreover, the drawback of another appealing framework SHAP is the expensive computations to estimate the SHAP values. Above all, these aforementioned frameworks just concentrate on the local interpretation of the model but ignoring the global interpretation, let alone the meso-level interpretation. Therefore, we pursue to construct an overarching interpretation framework to deal with the above shortcomings.

The pipeline of the interpretation framework is visualized in Fig 2. For input data, any tabular data or textual data are accepted. Then, assumed that a black-box model has been given, the objective is to apply the interpretation framework to inspect the variable influence in the model, constructing the connections between the prediction and the explanation. After all, the explanation is used to interpret the model behavior. As presented, the core elements in the framework could cast three

different views on the model. There are two options to choose when explaining a model from the global view by ranking the feature importance. More importantly, the local interpretation module contains several existing local methods like LIME and KernelSHAP, which extends the existing framework by providing more choices. Above all, the meso-level interpretation is achieved by combining the pattern mining technique with the various local interpretation methods. We manage to discover interesting patterns in the dataset where the inspected variable could reveal a significant impact by executing the subgroup discovery task. And the explored patterns could help non-expert to interpret the model.

In the following, the core components in the framework will be introduced separately. Since the details of the global interpretation methods have been covered in Section 2.2.1, we will start to describe the local interpretation methods comprehensively in the subsequent section.

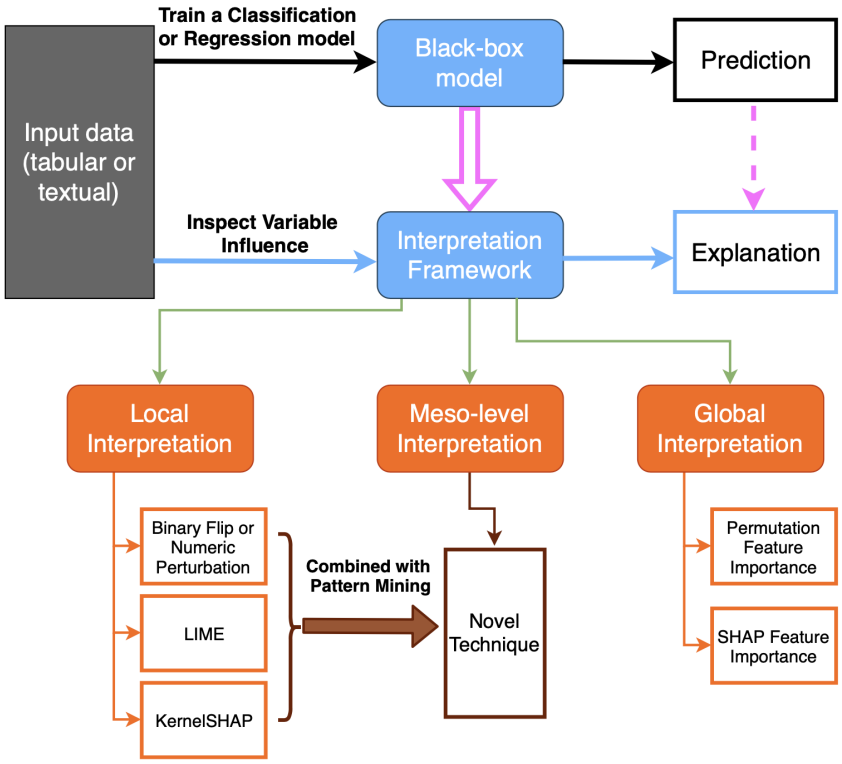

Figure 2: Black-box interpretation framework pipeline. Input data could be either tabular or textual data. The interpretation framework is applied to the black-box model, in order to obtain the explanation that is used to comprehend the model prediction. Three different levels of interpretation on the model are incorporated, i.e. global, meso-level and local interpretation. Various local interpretation methods are provided, including LIME and KernelSHAP. And by employing the novel approach that combines the local methods with the pattern mining technique, the meso-level interpretation of the model could be obtained.

3.2 Local interpretation methods

Global interpretation methods are dedicated to verify the most important features by comprehending the entire dataset, which can largely influence the model predictions. Admittedly, they can be utilized to select features when training the model, but they cannot provide detailed explanations to the model prediction. In addition, this group of methods can be unreliable especially when there is a bias in the feature being explained. Conversely, it is able to generate plausible explanations for each instance by applying the local interpretation methods. Therefore, it is more interesting to investigate the local interpretation methods. And the subsequent contents will cover several variants of local interpretation methods in detail.

3.2.1 Binary feature flip

Binary feature implies that the feature only contains two unique values. In other words, if it is encoded as discrete numeric number, the feature value should be either 1 or 0. The simple idea behind is that we want to measure the feature effect by observing the the output change if we flip a binary feature value. Flipping the binary feature value means to convert from 1 to 0 or the other way around. In practice, we could also use the XOR operation to map from 1 to 0 or vice versa. For instance, gender is usually regarded as a binary feature which only holds value "male" or "female". Thus, the value "male" can be encoded as 1 and "female" can be encoded as 0, and then we can apply the flip operation to this binary feature.

As previously mentioned, the assumption is that we have the dataset and the corresponding black-box model being explained. Initially, we could obtain the model predictions for each instance by applying the black-box classifier. Then, a value is flipped on a chosen binary feature and afterwards a new model prediction is generated by applying the same model to the modified instance. In this case, for each instance we could obtain two model predictions, denoted as probability by convention. Therefore, as a simple measurement, the impact of this binary feature could be estimated by the difference between these two model outputs.

However, in practice, we would consider two variants to assess the binary variable influence. One way is to calculate the absolute difference between two predictions, and in this way we could ignore the bias of this binary feature on the original dataset. Literally to say, the binary feature is more influential if the difference becomes larger. In contrast, we could also compute the output difference in a pre-defined direction. For example, we just care about the effect of gender when it changes from "male" to "female". In this case, not only the magnitude of the gender effect is obtained, but also the positive or negative behavior towards the prediction change. Likewise, if the magnitude of the prediction change is large, then we could assume that this binary feature plays an decisive role in the selected instance when making the model prediction.

3.2.2 Numeric feature perturbation

As the name suggests, this technique is specially applicable to features whose data type is numeric. The idea is that we could apply binary operations to the input values to produce new values, which operates the way similar as injecting noises into the original dataset. In principle, all binary operations are allowable, however, only addition and subtraction are considered in our experiment settings for convenience. Generally, we perturb a numeric feature value by increasing or decreasing by a certain value.

The procedure of measuring the effect of an input numeric feature is similar to that in binary feature value flip approach. For classification or regression tasks, we could make predictions with the existing model on the instance we desire to explain. Afterwards, a new prediction is made on the adapted instance which is produced through perturbation on the selected numeric feature. And the impact of this numeric feature could be approximately evaluated by the absolute difference of two output predictions. Roughly to say, larger prediction differences might imply the numeric feature has a stronger effect on the corresponding instance.

For example, given a simple scenario that a dataset has a numeric feature called "age" and we want to measure the impact of "age" when the age value is increased by 5 unit for a chosen data instance. By utilizing the proposed approach, we therefore could obtain the two model predictions and the difference is considered as a rough influence measurement of adding 5 unit on the "age" attribute.

3.2.3 LIME

As stated before, we would like to focus on post-hoc interpretability, which indicates to explain model decisions after the model has been trained. In order to improve the post-hoc interpretability in black-box models, multitude of interpretation methods are applicable. But among those approaches, we are particular interested in the model-agnostic interpretation method. It is pretty flexible in terms of models, and it can work with any type of machine learning models, which provides a great advantage over model-specific methods [19]. And the local surrogate approach is one of the model-agnostic methods, which is capable to explain individual predictions of any black-box models in a faithful way. As a concrete implementation of the local surrogate model, Local interpretable model-agnostic explanations (LIME) was initially proposed by Marco et al. [13].

The key point behind LIME is actually pretty straightforward. It is intended to explain individual explanations by fitting a simple interpretable model to locally approximate the underlying black-box model. The typical choice of the interpretable model could be regularized linear models like lasso or decision trees. To gain more insight of LIME approach, the toy example is shown in Fig 3. This is a binary classification task and the regions colored with yellow or white are regarded as two distinct decisions. Evidently, this decision function can not be easily interpreted by a linear model. In this scenario, we are interested to gain reasonable explanation

for an individual instance, which is marked with a bold red cross. Thus, in order to approximately evaluate the prediction for the selected instance, a local interpretable model is fitted and the surrounding artificial data points are created by perturbing the original data point. As can be seen from the figure, the learned local model, marked by the dashed line, could in principle provide a faithful interpretation to explain why the prediction for the chosen instance is in the pink region.

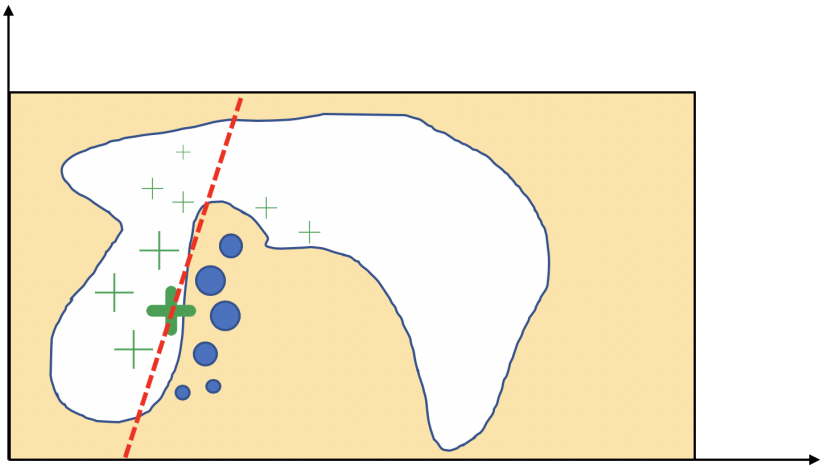

Figure 3: Binary classification task for a black-box model. Decision regions are colored with a yellow or white background. Instance to be explained are marked in a bold green cross. Artificial points, marked as crosses and circles, are created by perturbing the instance of interest, whose size are weighted by the proximity to the instance. The dashed line expresses the fitted local interpretable model which could give faithful explanations (cf. [13]).

After seeing the intuition behind LIME approach, it is necessary to have a deep insight about the procedure to train an explanation model. And the general recipe of LIME to interpret local instances is summarized as follows:

1. Select an instance whose model prediction is going to be explained
2. Perturb the instance and predict the outputs of all permuted observations
3. Calculate the distance from all permuted observations to the original instance as a proximity measure
4. Choose top features best describing the behavior of black-box model from the permuted data
5. Train a weighted, interpretable model to the permuted data, which is weighted by its proximity to the original instance

6. Extract the feature weights from the local interpretable model and use its coefficients to explain the model prediction

From the above procedure, it is noticed that the data permutation is a critical step. Actually, it depends on the type of input data when it comes to permuting an instance. When dealing with tabular data, new data points are generated by sampling around the selected instance with the mean and standard deviation of each feature. And these characteristics are drew from a normal distribution. However, it is a different story in the case of textual data or image data. For textual data, the permutations are performed by randomly removing words from the original instance. Basically, each word is assumed as a feature, therefore, each sampling data could be represented as a binary vector. If the word exists in a sampling instance, then the corresponding value is 1, otherwise the value is 0, meaning that the word has been removed from the original instance. And image data can be considered as a composition of a vast number of super-pixels, where each super-pixel is regarded as a feature. The permutation occurs that we randomly choose a operation for each super-pixel whether to turn it on or turn it off, representing as 1 or 0.

Since it is desired to train a weighted linear model whose weight is determined by the proximity between the neighborhood instance and the instance being explained, the essence of the next step is to choose a proper function to measure the proximity. Generally, the distance is computed using euclidean distance for tabular data, but for textual or image data, the cosine distance might be used. Concerning the spareness of data, such distances are mapped to the proximity measure with a value between 0 and 1 by applying a kernel function. Currently, the exponential smoothing kernel function is used to define the proximity around the chosen instance, and the kernel width is set default as 0.75. Usually, a small kernel width indicates that only close neighborhood could have an influence on the local model, while a larger kernel width implies that instances that are far way also have an impact on the local model. Nevertheless, it has to be reminded that there are no reliable instructions about how to choose the kernel width and even how to define the proximity.

Before fitting a local interpretable model, the simplicity of the explanation model has to be considered first. Imagine a scenario that the fitted model has hundreds or thousands feature weights, even though we can interpret those feature weights, the explanation model still remains too complex, which goes against our initial intention to provide human-understandable explanations. Therefore, we should choose the best features that can be exploited to explain the black-box model. In fact, a number of feature selection approaches are provided in LIME algorithm. For instance, forward selection is one of the feature selection methods, which performs by adding features one by one to the fitted model, e.g. ridge regression model, and those features contained in the best model are selected for the explanation model. The alternative is the highest weights approach. It picks the features with the highest absolute weight in a ridge regression fit of the black-box model [46].

Once the permutation has been performed and the proximity of the neighborhood instances are computed, it is time to train a weighted linear model around the instance of interest. It is assumed that the local model could mimic behavior of the black-box model at that locality. Then we can use this interpretable model to

comprehend the model prediction at that instance.

After knowing the detailed procedure to train an explanation model, we could also argue for the faithfulness of the explanation model from a mathematical perspective. The constraint of LIME could be represented as Eq. 1.

$$\xi = \arg\min_{g \in G} L\left(f, g, \pi_x\right) + \Omega(g) \tag{1}$$

As formally defined in the equation, f is the black-box model, g is the local explanation model needs to be figured out, and G is a group of interpretable models, which includes linear models, decision trees, or falling rule lists [47]. As depicted in the figure, the weight is measured by the proximity between the instance being explained and the surrounding artificial instances, which is defined as $\pi_x(z)$. In specific, let $\pi_x(z) = \exp\left(-D(x,z)^2/\sigma^2\right)$ be an exponential kernel defined on some distance function D (e.g. cosine distance for text, L2 distance for images) with width σ [13]. The complexity of explanation model g is described as $\Omega(g)$. For example, the complexity could be estimated by the depth of trees for decision trees models or by restricting the maximum number of features in linear models. As seen from the formula, in order to obtain the local explanation model for instance x, the loss L (e.g. mean squared error) should be minimized while maintaining the complexity as low as possible.

After literature review, it is found out that LIME is one of the few methods that work for tabular data, textual data, and image data, which is a very promising approach. The python implementation is currently available online [22], which is still in active development and needs further exploring.

3.2.4 Shapley value

As can be viewed from the literature, numerous approaches have been recently proposed to explain the black-box models. Though they adopt diverse techniques to deal with the local interpretation issue, you might notice that one common thing for those approaches is that they will assign a score to each feature and to explain the model predictions relying on those feature scores. The scores could be the weights of a local interpretable model as we have seen in the LIME approach. Basically, we could extract the weights for each feature and the weights represent the importance of each feature in the instance of interest. Alternatively, the score could be regarded as a variable attribution. Biecek proposed a method to assess the local variable attributions based on the contributions of explanatory features to the model prediction [48]. To put it another way, the model prediction could be decomposed as different feature attributions and the magnitude of that attribution value is measured as the difference between the output value and the average output over all perturbations. And the attribution value indicates the feature impact on the instance. The way to compute the feature attributions actually is similar to the principle of using additive model to explain the feature effects. A framework called ExplainD was designed to explain the model decisions using additive evidence, which was introduced by Poulin

et al. a decade ago [49].

To have a better understanding of how to compute the feature attributions, let's first have a look at the additive models. For additive models, the model prediction is aggregated by the marginal effects of each feature. As an example, a linear regression could be considered as a simple additive model, which could be expressed as follows:

$$f(x) = f(x_1, \ldots, x_n) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

In the above formula, x is the instance of interest. The coefficient β_i can be interpreted as the ith feature weight. Nevertheless, it is more interesting to investigate a particular feature contribution, and the contribution for ith feature is expressed as Eq. 2.

$$\varphi_i(x) = \beta_i x_i - \beta_i E[X_i] \tag{2}$$

Clearly, $E[X_i]$ is the expectations of all possible values for feature i. As can be seen, the feature attribution $\varphi_i(x)$ is estimated by the difference between the effects caused by feature i and the expected feature effects. It is obvious to acknowledge that if the feature attribution is greater than 0, then it means this feature has a positive effect for making the model prediction. Otherwise, there is a negative effect influencing the model prediction or even no impact on the prediction.

As a clarification, this approach is particular for additive models. But in reality, many models are non-additive, especially for black-box models. And for those non-additive models, the idea could be adapted to compute the feature contributions by perturbing the selected feature values. However, it was pointed out by Štrumbelj and Kononenko that the results could be unreliable by just permuting one feature at a time [50]. Besides, another major issue still exists for the above presented method, in which the feature interactions and redundancies are not taken into account. To address this problem, an interactions-based approach was initially put forward by Štrumbelj et al, which solved the problem by explicitly considering all feature combinations [50]. This method was inspired by the coalitional game theory which was aimed to fairly distribute the "payout" among the "players". To capture the fairness in distributing feature attributions, Shapley values are used as the solution, which is named after its inventor Lloyd S. Shapley [51].

The Shapley value is usually formulated with respect to the grand coalition, which is consisted of all players $P = \{1, ..., p\}$. Each subset of players $S \in \{1, ...p\}$ is a coalition. And there is a characteristic function val describing the worth ϕ_j of each coalition. The task in the game theory is to find a way to fairly distribute the worth among all players. Actually, a unique solution can be discovered, which is the Shapley value. Therefore, the Shapley value is calculated as a representation of feature attributions in the aforementioned interactions-based approach. Moreover, the insight of Shapley value is that the feature dependence could be eliminated by averaging over all possible coalitions or permutations of the players [52].

Apart from the abstract concept, an illustration taken from Molnar's book might help us intuitively understand the Shapley value [5]. Imagine there is a room and all feature values of an individual instance enter the room in a random order. All feature values, considered as players here, need to collaborate with each other to participate the game, where each player contributes to receive the final prediction. And each feature ordering represents a coalition. Consequently, the Shapley value of a feature corresponds to the average change in the prediction when the feature joins an existing coalition. In other words, the Shapley value is the average marginal contribution of a feature across all possible coalitions. Referring to the Shapley value of a feature, the formal definition is given by Eq. 3, where S is the subset of the features in an individual instance, p is the number of features, and x is the vector of feature values of the instance to be interpreted. As for characteristic function val, it describes the contribution of feature j in each coalition. Recall from previous explanation, it could also be told from the equation that the Shapley value for feature j is the average marginal contribution taken over all permutations of coalitions.

$$\phi_j(val) = \sum_{S \subseteq \{x_1,\ldots,x_p\} \setminus \{x_j\}} \frac{|S|!(p-|S|-1)!}{p!} \left(\text{val}\left(S \cup \{x_j\}\right) - \text{val}(S)\right) \tag{3}$$

Referred to the original paper [51], Lloyd claimed that the Shapley value was the only attribution approach that adhered to the following properties: Efficiency, Symmetry, Dummy, and Additivity, which guaranteed that the distribution of worth among the grand coalition was fair. These four axioms are listed as follows and the proof shall be found in the book "Computational Aspects of Cooperative Game Theory" written by Chaallkiiaadaakiis et al, but the details are omitted here [52].

Efficiency: The sum of feature contributions must equal to the difference between the final prediction for x and the expected prediction considering all permutations of coalitions. As an equation, it is expressed as: $\sum_{j=1}^{p} \phi_j = \hat{f}(x) - E_X(\hat{f}(X))$

Symmetry: Interchangeable features should receive the same worth, i.e., for any instance of interest x, if feature i and feature j are interchangeable, then the contributions are the same. It can be represented as: $if\ \ val\left(S \cup \{x_j\}\right) = val\left(S \cup \{x_k\}\right)$ then $\ \phi_j = \phi_k$

Dummy: Dummy players receives nothing, i.e. the contribution of feature j is 0. In other words, this feature does not change the prediction when it joins any coalitions. This property is demonstrated as: $if\ \ val\left(S \cup \{x_j\}\right) = val\left(S\right)$ then $\ \phi_j = 0$

Additivity: For any pair of games v and w, the combined worth should equal to the sum of two individual worth. It can be formulated as follows: $\phi_j(v+w) = \phi_j(v) + \phi_j(w)$ where $\ (v+w)(S) = v(S) + w(S)$. For example, if a random forest is trained and the additivity axiom could guarantees that the Shapley value for each tree could be computed individually first and then average them to obtain the final Shapley value.

As can be inferred from "Efficiency" axiom, the Shapley value is the marginal feature contribution to the difference between the prediction for the instance being explained

and the average model predictions taken over all instances. Therefore, in the context of a local explanation for an individual instance, the feature attributions measured by Shapley value can be interpreted as follows: Assume the feature contributions are normalized already. If a feature has a positive contribution, it means this feature pushes the prediction of the instance to be explained away from the expected model prediction. In other words, this feature could increase the prediction probability with respect to the target label. On the other hand, a feature could have a inverse impact on the prediction if the feature contribution is negative.

However, as you probably have noticed, the time complexity to compute the Shapley value is exponential according to Eq. 3, which makes the method infeasible for practical use. Later, in year of 2014, improvements had been made to reduce the time complexity by introducing the efficient approximation algorithm. The approximation algorithm was first illustrated by Štrumbelj in a form of Monte Carlo integration [50][53].

Recall that we want to compute the Shapley value for each feature. The approximation algorithm works in the following: First, choose the instance x to be explained, the feature j to be inspected and the number of iterations M. For each iteration, another instance z is chosen randomly from the dataset and a random permutation is set as o. Afterwards, the data instance is shuffled according to the feature ordering. Then, two new instances are constructed by connecting the values from instance x and sample instance z by considering the position of feature j. The first instance is consisted of preceding j-th value from x and succeeding values from z. As a comparison, the second instance includes the preceding j-1-th value from x and the rest values from z. The key difference is that the first instance contains the j-th feature value from the selected instance, while the second excludes it by randomly choose another value from the dataset to replace it. Basically, the prediction difference is the j-th feature attribution in each iteration. Finally, the Shapley value of feature j can be evaluated by averaging over all iterations. And the pseudo code algorithm is described in Algorithm 1.

3.2.5 KernelSHAP

As seen from the previous section, the classical Shapley value is a promising local interpretation method to explain individual model outputs. Nevertheless, according to the algorithm presented above, only a list of Shapley values could be calculated and each value corresponds to a feature attribution. Admittedly, those feature attributions can be interpreted as an explanation for the instance to be explained, but they are just simple values which fails to capture the connections between input changes and model prediction changes. Therefore, the better way is to train a local explanation model like LIME approach to interpret local instances. To address this problem, Lundberg and Lee [16] proposed a unified framework for explaining predictions, which was based on the Shapley value, and they named it SHAP(SHapley Additive exPlanations) values. Particularly, SHAP values are considered as a unified measure for feature importance. In this framework, a kernel-based explanation method based on the Shapley value and LIME approach was put forward. Similar to LIME approach, a linear explanation model is fitted and the coefficients in the

Algorithm 1 Approximating the jth features contribution for model $\hat{f}$, instance x of interest, and draw M samples

Result: Shapley value for the value of the j-th feature

for $m = 1, ..., M$ **do**

1. Draw random instance z from the data
2. Choose a random permutation o of the feature values
3. Shuffle instances x : $x_o = \left(x_{(1)}, \ldots, x_{(j)}, \ldots, x_{(p)}\right)$
4. Shuffle instances z: $z_o = \left(z_{(1)}, \ldots, z_{(j)}, \ldots, z_{(p)}\right)$
5. Construct two new instances:

$$x_{+j} = \left(x_{(1)}, \ldots, x_{(j-1)}, x_{(j)}, z_{(j+1)}, \ldots, z_{(p)}\right)$$

$$x_{-j} = \left(x_{(1)}, \ldots, x_{(j-1)}, z_{(j)}, z_{(j+1)}, \ldots, z_{(p)}\right)$$

6. Compute marginal contribution: $\phi_j^m = \hat{f}\left(x_{+j}\right) - \hat{f}\left(x_{-j}\right)$

end

Compute Shapley value as the average: $\phi_j(x) = \frac{1}{M} \sum_{m=1}^{M} \phi_j^m$

model are referred to the SHAP values. In addition, TreeSHAP was also included in the framework, which was a very efficient estimation approach particular for tree-based machine learning models, such as random forests and gradient boosting trees. In comparison to the KernelSHAP which is too computationally expensive to compute the SHAP values owing to computations for the exponential number of possible coalitions, TreeSHAP is way more efficient and can significantly reduce the time complexity from exponential to polynomial. Besides, by leveraging the difficulty for interpreting neural networks, an alternative method called DeepSHAP was also proposed in the framework. Again, the approximation algorithm to compute the SHAP values is optimized within the polynomial time bound. Since the KernelSHAP is a model-agnostic method while TreeSHAP and DeepSHAP are only suitable for tree-based models and deep neural networks respectively, it is worth to devote more efforts on the KernelSHAP approach and the details will be described in the subsequent content. Nevertheless, the TreeSHAP and DeepSHAP approaches are also supported in our black-box interpretation framework.

As told before, KernelSHAP is aimed to train a local interpretation model to explain local model outputs, and the estimated coefficients of the model are the SHAP values. In other words, the model prediction of an instance could be decomposed as sum of feature attributions, which can be expressed as: $f(x) = g\left(x'\right) = \phi_0 + \sum_{j=1}^{M} \phi_j x'_j$, where f is the black-box model, g is the local explanation model, ϕ_0 is the expected model prediction over all samples, and the product of ϕ_j and x_j represents the contribution for each feature j. By the way, the SHAP values ϕ_j is the weight extracted from the fitted linear model. It can be inferred that the innovation of this attribution-based approach is combination of LIME and Shapley value. From the game theory domain, it is proved that Shapley value is the only solution that

satisfies properties of Efficiency, Symmetry, Dummy and Additivity. Obviously, these properties are also met in the KernelSHAP approach since the Shapley value is calculated for each feature in the context of a local explanation. Beyond that, three more properties derived from above axioms, namely Local accuracy, Missingness, and Consistency, are supposed to be satisfied by the KernelSHAP approach claimed by Lundberg and Lee. Due to the space limitations, the comprehensive description is omitted here and it shall be found from the original paper [16].

It might be a great concern to figure out how to train the local interpretable model using KernelSHAP method. Actually, the thorough procedure has already been summarized in LIME approach. After picking the instance desired to explain, the next step is to generate neighborhood instances by permuting the original instance. The perturbation method for KernelSHAP is different from what has been introduced, it is therefore described as follows: Firstly, a random binary vector z'_k so called "coalition vector" is constructed by repeating coin flips multiple times, expressed as: $z'_k \in \{0,1\}^M, k \in \{1,\ldots,K\}$, where M is the number of features of each instance. Then, new valid instances are generated by using a transformation function h_x where $x' = h_x(z'_k)$. Basically, the idea is to map the binary space to the original feature space, where the new instance is valid. And the transformation function works in a way that the feature value in the original instance is remained if the corresponding binary vector has value 1, otherwise the feature value is replaced by randomly choose a value from the dataset. In other words, the original instance is perturbed by substituting feature values which is considered missing, meaning the coin flip has value 0. With all the permuted instances, the outputs of those neighborhood instances could be obtained by applying the underlying black-box model. Afterwards, a critical step is to measure the weight of each neighborhood instance with respect to the original instance, which corresponds to the proximity measure as shown in LIME. In order to learn more about individual feature effects, it is desirable to assign more weight on those instances with only few feature values absent or few feature values present, i.e. the binary vector has almost all 0's or all 1's. In this case, to achieve this goal, Lundberg and Lee proposed the following kernel function to estimate the weight:

$$\pi_{x'}(z') = \frac{(M-1)}{(M \text{ choose } |z'|)\,|z'|\,(M-|z'|)}$$

where M is the number of features, $|z'|$ is the number of non-zero elements in z'.

Given the weighted instances, a linear interpretable model could be fitted. Considering the feature sizes and model sparseness, the regularization terms could also be added to avoid overfitting. As a result, the coefficients of the linear model are returned as the SHAP values for each feature. And they can be interpreted as an explanation for the instance being explained.

Actually, all the aforementioned approaches, including KernelSHAP, TreeSHAP, and DeepSHAP are implemented in the unified framework called SHAP. It is currently in active development and is accessible online [23]. It is seemed to be a very optimistic approach and we are quite interested in this novel model explanation method, therefore, further experiments will be conducted later.

3.3 Pattern mining technique

It appears from the aforementioned sections that a bunch of local interpretation methods which could be exploited to explain black-box model predictions are incorporated into our black-box interpretation framework. Attempts were made to investigated those local interpretation methods. In principle, it is assumed that we could obtain a local explanation model for each model prediction independent from the underlying black-box model. To put it another words, the instance-level explanation for a chosen prediction shall remain consistent even though the underlying black-box model is modified. Nevertheless, situations could happen that two disparate explanations are provided for the same instance prediction when two different underlying black-box models are applied. This unexpected results may be attributed to the excessive interpretation of the selected instance, which is merely one sample instance rather than a representative of all instances. Moreover, insights extracted from an individual instance might be too specific to train a well-fitted explanation model, causing unstable and unreliable explanations.

Recall from previous part, global interpretation methods and local interpretation methods could provide explanations from a global view point and a local view point on the dataset, respectively. Therefore, to overcome the drawbacks from the global view or local view, we would naturally consider to interpret the black-box model from somewhere between the global view and the local view. Inspiration from the idea that black-box models could detect hidden patterns in a dataset such that they could exhibit a good performance when executing classification tasks, it comes to our mind that pattern mining is exactly the technique that could be utilized to discover hidden patterns in a dataset, which coincides with the former idea somehow. Therefore, a novel method by combining the local interpretation methods and pattern mining technique is proposed in this thesis. In this regard, it is presumed that the novel approach could provide a meso-level interpretation of model's behavior.

To be frankly, it brings more clarity if we discuss these two individual components separately, which also correspond to two techniques. Since we have already covered the discussion about local interpretation methods, it is intended to elaborate the subgroup discovery technique in the following section.

3.3.1 Subgroup discovery task

Note that the terminology pattern mining, whose interchangeable name is subgroup discovery, refers to a data mining technique which pursues to find subgroups of data instances that exhibit interesting characteristics with respect to a predefined target variable [27]. Moreover, subgroup discovery is a descriptive technique, which describes details such that the results are understandable by human experts.

In a more formal definition, four elements could be considered the fundamental components to compose the subgroup discovery task, which is defined by a quadruple (D, Σ, T, Q). These elements are illustrated as below [54] [29]:

- D is a dataset and is formed by a set of instances, and each instance consists of a set of attributes

- Σ constrains the search space, which is made up of subgroup descriptions (patterns). And patterns consist of a set of selection expressions, also known as selectors.

- T represents the target variable for the discovery task. Various types of target could be identified, including binary target, numeric target, or complex target.

- Q defines the quality measure criteria. Different quality measure criteria are specified for different types of target concept.

In principle, the dataset could be any kind of data type, nevertheless, we will primarily concentrate on tabular data and textual data. Actually, the detailed description about the datasets that are used in this thesis will be discussed in the next section, and thereby it will not be deeply investigated here. As for the search space, commonly it is accepted as conjunctive combinations of selectors for the reason that such subgroup descriptions are interpretable by practitioners. As an example, a pattern could be formatted as: $P = sel_1 \wedge sel_2 \wedge ... \wedge sel_d$, where all selection expressions are evaluated to be true. Loosely speaking, the full search space is exponential to the number of input features of the dataset, which can significantly affect the pattern mining efficiency. By taking the size of the search space into account, beam search strategy is adopted to shrink the search space in order to speed up the subgroup discovery task and more detailed information will be illustrated later.

The choice of the target concept is normally task driven and closely related to the dataset. Using a binary variable as the target of subgroup discovery is a more simple and general situation. Since the binary variable only contains two values (True or False), it is aimed to identify interesting subgroups for each of the possible value. Basically, the idea is to discover patterns whose target share is either remarkably high or remarkably low. But in this thesis domain, it is desired to discover patterns where the inspected variable could reveal a significant effect, implying by the importance scores. Therefore, the importance score of the selected attribute is considered as the target concept, which belongs to the numeric data type. Generally, pattern mining for numeric target is more complicated because the attribute values could be handled by a numerous approaches such as numeric target discretization in a predefined number of intervals, or dividing the numeric domain into two ranges with respect to the average. And frequently mentioned discretization methods includes equal-width discretization, equal-frequency discretization, and etc. An overview of discretization methods for a numeric attribute was reported by Garcia et al. [55]. In this thesis, it is more inclined to apply the equal-frequency discretization method.

Without any doubt, it is critical to choose the quality measure carefully, since the results of subgroup discovery are mostly controlled by the quality measure criteria. In light of the fact that the interestingness measure plays a decisive role in the subgroup discovery task, thus, a comprehensive discussion about quality measure for numeric target will be presented in the following subsection.

3.3.2 Interestingness measure for numeric target

As was said before, the interestingness measure for numeric target becomes more complex to investigate than situations where binary feature is chosen as the target. Nevertheless, a list of interestingness measures for numeric target was reviewed by Pieters et al. [32]. As could be summarized, those interestingness measures are heavily relied on the basis of the statistical distribution of numeric values, such as mean value, median value, or variance. And the general idea behind is to design the interestingness measure for numeric target with respect to those predefined data statistics. More specific, interesting subgroups would be discovered if the computed data characteristic in the subgroup is significantly deviating from the value calculated in the entire population. Referred to paper [29], five categories of interestingness measure for numeric target are outlined, which included mean-based measures, median-based measures, variance-based measures, distribution-based measures, and rank-based measures. Since the mean-based measure is widely accepted and applied in many applications, therefore, it is selected as the primary quality measure for numeric target in later experiments.

In point of fact, within this mean-based measure family, several concrete interestingness measures could be further explored, distinguishing by the evaluation functions. Take an example, one simple evaluation function is *Average function*, which calculates the difference between the mean value in the subgroup and the mean value in the entire dataset, denoted as: $q_{mean} = \mu_p - \mu_\emptyset$. However, subgroup size is not considered in the former measure, which fails to perform well in certain circumstances. And it was actually observed from literature that *Generic mean function* was the most prevalent mean-based interestingness measure due to its simplicity to be interpreted. And the general formulation is denoted in Eq. 4.

$$q^a_{mean}(P) = i^a_P \cdot (\mu_P - \mu_\emptyset)\,, a \in [0, 1] \tag{4}$$

It appears from the equation where i_P is the size of the subgroup, a is a parameter which weights the subgroup size and deviations, and $\mu_P, \mu_\emptyset$ represent the average value in the subgroup and the average value in the dataset, respectively. In particular, the choice of parameter a could be selected in an iterative process. For example, a is required an increment if the subgroup size is too small to have a significant score, conversely, low parameter values for a is preferred with a high deviation of mean target values between the subgroup and the overall dataset. Therefore, after calculating the interestingness score for each subset, those subgroups with significantly higher or lower mean values are considered as interesting and the descriptions of them are our desirable interesting patterns.

Likewise, the mean value could be substituted by other data characteristics, such as median value. Then the subgroup quality could be estimated by the generic median function in a similar way. Theoretically, other variants from the aforementioned interestingness measure family are applicable as well, but the mean-based generic function is implemented and mainly used through all the subgroup discovery tasks in our experiments.

3.3.3 Algorithmic components

As was briefly mentioned, the algorithms for pattern mining are composed of three algorithmic components, which are data structures, pruning strategy, and enumeration strategy. And the major differences between subgroup discovery algorithms are owing to the variation on one of the components.

Concerning the data structures, mostly applied structure is horizontal layout, e.g. tabular dataset. In addition, other variants like vertical data representations or FP-tree representations could also be applied in distinct algorithms. However, the details of data representations in various algorithms are beyond the scope and would not be explored further.

As a matter of algorithm performance and memory usage, the pruning strategy plays a decisive role. Frequently reviewed techniques such as anti-monotone constraints and optimistic estimate pruning could be employed. And the intuition behind the anti-monotone strategy is that if a pattern did not fulfill a given constraint, then any of its specializations would not be satisfied either. In doing so, the search space is largely shrunk. But in our black-box interpretation framework, the optimistic estimate pruning strategy is exploited. The principle is that the optimistic estimate value of a pattern P is greater than any interestingness score of its specializations S, as denoted by Eq. 5, where $oe_q(P)$ stands for the optimistic estimate interestingness score of pattern P. As is known, the general purpose of pattern mining task is to discover the top-k most interesting subgroups. To put differently, subgroups with the top-k highest interestingness score are discovered. In this way, a dynamic minimum quality threshold could be set as the lowest score from the top-k subgroups, and any pattern with smaller quality score than this threshold is pruned. By doing so, a considerable number of patterns could be skipped and the efficiency of the algorithm could be improved.

$$\forall S \supset P : q(S) \leq oe_q(P) \tag{5}$$

Actually, the essence of a subgroup discovery algorithm is the enumeration strategy, also known as the search strategy, since the pattern mining task could be considered as a search problem in the search space of patterns. Commonly, two main categories of search strategy are identified, which are exhaustive search strategy and heuristic search strategy. The former search strategy family aims to acquire the optimal results of the most interesting subgroups by traversing through the whole search space. And the exemplary strategies are depth-first-search and breadth-first-search. In contrast, heuristic approaches seek to discover interesting patterns but not necessarily the optimal patterns by evaluating only promising candidates. Take an example, beam search strategy belongs to this genre and it is also the main search strategy utilized in the framework due to its efficiency. Beam search is a level-wise approach and it usually utilizes a full refinement operator, which means the pattern on the next level is generated by adding non-used selectors to the current level of pattern. And the intuition behind is that it is assumed that the next level patterns are more likely to be interesting if the current level patterns are also interesting. Therefore, the search starts with an empty hypothesis, then it tries to find the best patterns

with size k (corresponding to beam width) by evaluating all non-used selectors in the search space. In other words, patterns on each level with top-k highest interestingness score are selected. Following that, at each search iteration, the hypotheses contained in the beam are expanded but only the currently best k hypotheses are kept using a hill-climbing greedy search [56]. In the end, the remaining hypotheses are considered as the most interesting patterns in this dataset with respect to the target concept.

3.3.4 Redundancy avoidance

So far, we have already discussed the numeric target concept and the corresponding interestingness measure. Afterwards, there have been a few discussions about the subgroup discovery algorithms, including the pruning strategy and the enumeration strategy. Naturally, the next step is to execute the subgroup discovery task by choosing a proper algorithm to generate the most interesting patterns. However, it is observed that the result set after pattern mining task contains strongly overlapping subgroups, which is not as good as expected because only little information could be extracted from redundant patterns.

The problem of redundant subgroups might be attributed to the fact that only the subgroup size and statistics difference between subgroups and entire dataset are considered to estimate the interestingness score but the selectors in search space are ignored. For example, assume that the average contributions of *age* for the entire dataset is at $M_{\emptyset} = 0.50$. And the mean value in the subgroup P with the expression "$age > 40 \wedge gender = male$" is $M_{age>40 \wedge gender=male} = 0.80$. It seems that the pattern P has a high quality score and is supposed to be an interesting pattern. However, it is probably not interesting enough if given the information that its generalization S has nearly the same value, e.g. $M_{age>40} = 0.78$, which means pattern P does not deviate significantly from its generalizations S, and should not be regarded as interesting.

To avoid that such subgroups are included in the result set, *Generalization-awareness* interestingness measure could be applied to improve the traditional selection criteria for pattern mining by considering the statistics of the subgroup and also to its all generalizations. Referred to paper [57], Grosskreutz et al. proposed to estimate the quality of a pattern P as the minimum of the quality of P with respect to the extension of all its generalizations. And the incorporation of generalizations into the interestingness measure is formatted as Eq 6, where q^{Δ} is the incremental version of q, DB is the dataset, P is the subgroup and H includes its all generalizations.

$$q^{\Delta}(DB, P) = \min_{H \subset P} q\left(DB\left[H\right], P\right) \tag{6}$$

Since the generic mean function is mainly exploited to assess the quality of the subgroup, the above equation could be formalized in another way, as shown in Eq. 7. By doing so, redundant patterns are avoided and more interesting subgroups are discovered.

$$q^a_{\text{mean}}(P) = i^a_P \cdot \left(\mu_P - \max_{H \subset P} \mu_H\right), a \in [0,1] \tag{7}$$

3.4 Meso-level interpretation methods

The meso-level interpretation approach can provide a more fine-grained explanation of variable influence than the global interpretation method and a relative stable explanation than the local interpretation method. It is achieved by incorporating the pattern mining technique with the local interpretation methods (see Section 3.4.1). Besides, it has been stated that the decision trees could also be used to detect local patterns. Hence, a variant that combines the decision trees with the local interpretation methods is also put forward to cast the meso-level explanation on the feature influence (see Section 3.4.2).

3.4.1 Pattern mining with local interpretation methods

It has been announced that aside from the interpretation methods that are used to explain the black-box model from a global or local perspective, we further extend the black-box interpretation framework by introducing a novel technique to interpret the model from the meso-level point of view. The innovation is that the proposed approach combines the local interpretation methods and the pattern mining technique. In doing so, we could identify the impact of an inspected variable not only on an individual instance, but also on groups of instances. Theoretically, it is observable that a selected feature in a black-box model could present extraordinary influence in some particular patterns. And we are passionate for those discovered patterns, which could provide explanations for the model predictions.

Firstly, the influence for a selected variable in each instance could be computed by applying the local interpretation methods on instance level. The impact of the feature is measured by the prediction change if binary feature flip approach or numeric perturbation approach are utilized. Alternatively, the impact of the feature could be indicated by the feature weight when an local interpretable model is fitted by using the LIME approach. Furthermore, it could also be estimated by the SHAP values of the feature by considering feature attributions to the model prediction. In this way, the selected feature impact on each instance is represented by an importance score. Therefore, by taking the feature influence as the numeric target, we could execute the subgroup discovery tasks to discover patterns where the feature has a remarkable impact.

3.4.2 Decision trees with local interpretation methods

Apart from the pattern mining technique, an alternative method to mine local patterns is decision tree algorithm. Commonly, we apply the greedy algorithm to create our decision tree, i.e. CART algorithm. Inspecting the model from the interpretability perspective, it is usually considered as an interpretable model that provides

human-understandable decisions. In that regard, we can explain the model decisions by looking at the decision rules. Beyond that, the decision trees could also be applied to find local patterns inferring from the input features. Generally to say, each decision pattern is defined by a decision path, which is the path from the root node to a leaf node.

Likewise, we could easily obtain the local variable influence for each instance by employing the local interpretation methods. As for the local variable influence estimation, those aforementioned local interpretation methods can be employed, e.g. LIME approach or KernelSHAP method. By taking the local influence of a specific feature as the target label, exceptional patterns of that feature could be found following the decision path and those local patterns could help us to explain the model decisions better. However, in comparison to the pattern mining technique, the local patterns mined by decision trees can lead to redundant feature selectors, i.e. the same attribute may occur multiple times in the decision pattern.

4 Experiments & Evaluations

The effectiveness of the black-box interpretation framework was illustrated in this section by conducting experiments on a synthetic dataset and empirical datasets. As for the parameter setup, if not stated otherwise, we adopted the KernelSHAP as our main approach to measure the local variable influence for all experiments. In the subgroup discovery task, the search space was formed by attribute-value pairs of categorical features and intervals of numeric features. Besides, the numeric intervals were constructed by employing the equal-frequency discretization and by default we split each numeric values into 10 intervals. Evidently, the local influence of a feature to be explained was considered as the target concept. And concerning the quality measure, the generalization-awareness interestingness measure was chosen by default, with parameter setting $a = 0.5$. The maximum combination of selectors in the searching algorithm was constrained to 3. Lastly, the beam search strategy was exploited in detecting patterns.

And from the evaluation of the synthetic dataset, it could be observed that the artificial pattern in the synthetic dataset would be recovered through the meso-level interpretation on feature's influence (see Section 4.1). Furthermore, case study researches were performed to examine the proposed interpretation framework. First of all, the well-known *Adult Income* dataset from the UCI repository was evaluated in Section 4.2. Then, another publicly available dataset extracted from the amazon reviews was examined and the possible outcomes could be found in Section 4.3. Later, it was testified that the framework could not only interpret classification models, but also regression models. And the interpretation for a regression model could be seen in Section 4.4. In addition, more experiments based on the variation of the framework pipeline were performed. Particularly, one variation was to compare the various local interpretation methods, focusing on the local level interpretation of the model (see Section 4.5.1). Another experiment variation was to consider the different interestingness measures in the pattern mining technique, and the findings could be seen in Section 4.5.2. Moreover, we wanted to investigate the difference in local patterns that were found using the subgroup discovery technique and the decision tree algorithm (see Section 4.5.3).

4.1 Synthetic dataset evaluation

Presumably, there were hidden patterns in a synthetic dataset where the selected attribute had extraordinary feature impacts. And in this section, the aim was to justify whether the exceptional subgroups could be recovered from the artificial dataset by inspecting the variable influence. The procedure to construct the artificial dataset, as well as the procedure to perform the experiments, were described as follows.

We created the synthetic dataset based on the popular *German Credit* dataset that included 9 original attributes. Then, it was modified by adding an "Artificial" attribute with random binary values. For each instance, we additionally generated

10 binary noise attributes. And the purpose was to predict whether people have good or bad credit risks. As can be seen in Fig 4, the sample of the dataset was shown. It was known that the the coefficients in the logistic regression model had a straightforward interpretation, indicating the feature influence to some degree. And we could manually modify the coefficients with freedom. Hence, we would try to infer the gender influence from a logistic regress model in this case. Firstly, we trained a baseline logistic regression model using the original attributes, where the weight for attribute "Sex" was 0.4. Then we would just adopt the coefficient of "Sex" in the baseline model to artificially create two logistic regression models depending on the "Artificial" attribute, denoted as LR1 and LR2 respectively. In LR1 model, the influence weight of "Sex" was increased to 0.8, while the weight was reduced to 0.1 in LR2 model. In that regard, the local gender influence would be estimated by employing the LR1 model to the instances with value "Artificial=1" and LR2 model to the instances with value "Artificial=0". By assigning more weight to the attribute meant that the attribute played a more decisive role in the model, and on the contrary, the attribute revealed less significance in the model. By saying that, the gender influence were supposed to be significant on instances with pattern "Artificial=1". Therefore, our approach should detect the pattern "Artificial=1" as the most relevant one concerning the gender influence.

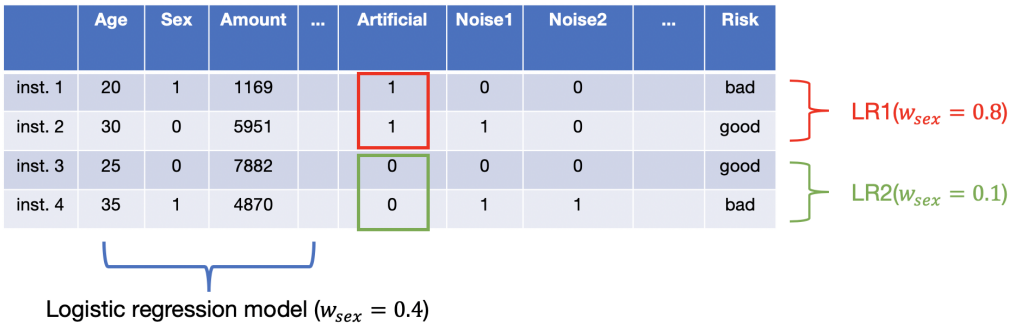

Figure 4: Synthetic dataset evaluation. The synthetic dataset consists of 9 original attributes from the popular German Credit dataset, one binary "Artificial" attribute and 10 more binary noises attributes. A baseline logistic regression model was trained based on the original attributes. And the weight of "sex" was assigned as 0.4. Based on the split feature, we would artificially generate two models, i.e. LR1, LR2. The model LR1 was applied to instances with value "Artificial=1", where the weight of "sex" in the model would increase to 0.8. Otherwise, the model LR2 was applied, where the weight of "sex" was reduced to 0.1.

To measure the gender effect, we could use the KernelSHAP approach. For each instance, we would get an importance score assigned to the feature "sex", indicating its local influence. Then, by treating the gender influence as the target concept, the subgroup discovery technique was applied to the artificial dataset to discover interesting subgroups. And the findings are revealed in Table 1. It could be observed that the first subgroup with the highest interestingness score can recover the pattern "Artificial=1". And from the next subgroups, we could also observe its specifications of the desired pattern, e.g. "Housing=1 AND Artificial=1".

Table 1: Top exceptional subgroups for the synthetic dataset. Each subgroup is described by a conjunction of selectors. Accompanying with the subgroup description, the size of the subgroup (denoted as # Inst.), the average effect of "sex" in the subgroup (denoted as μ_{sex}), and the interestingness score (denoted as q) for the subgroup are presented. The average effect of gender overall is 0.065. And the top-ranked subgroups with the highest interestingness score are listed. It is noticed that the pattern "Artificial=1" could be recovered from the results.

Subgroup description	# Inst.	μ_{sex}	q (score)
Artificial=1	516	0.113	0.024
Housing=1	713	0.079	0.009
Housing=1 $\wedge$ Artificial=1	362	0.138	0.009
Duration>=36	170	0.107	0.007
Duration>=36 $\wedge$ Artificial=1	95	0.176	0.006

4.2 Case study: Adult Income dataset

The first comprehensive case study was demonstrated as follows. The dataset utilized in the case study was first introduced. Then, the experimental setup was elaborated afterwards. It consisted of the black-box model's choice, the default parametric setup, and the detailed experimental procedure. Subsequently, the experimental results were shown.

4.2.1 Tabular dataset

Recall from the previous assumption on the proposed black-box interpretation framework, it supports the comprehension of any underlying black-box models that are trained on tabular datasets or textual datasets. And the datasets are carefully selected in order to get rid of the data quality issues. Generally, the chosen datasets are categorized as tabular datasets and textual datasets, and most of them could be found in *UCI Machine Learning Repository* [58][59], which is usually considered as a reliable source of data.

The first study case is going to examine a tabular dataset, which contains data in a column formatted table. Each column of the dataset is regarded as a distinct feature of the dataset with the same data type and each row is an individual instance. Normally, each column could have an either numerical or categorical data type, but each instance could contain multiple data types. Taken the quality of a tabular dataset into account, the *Adult Census Income* dataset will be explored during the experimental phase, which was extracted from census bureau database. It contains 14 descriptive features and more than 40 thousand instances. Most features are personal demographics, e.g. age, gender, educational level, etc. And the label of the dataset is to determine whether a person earns more than $50K or less than $50K a year.

4.2.2 Experimental setup

It started with a brief overview of the black-box model that was being interpreted, followed by a default parametric setup. Then, a series of steps to conduct the experiments were listed.

Black-box model. In a recent study, Rudin pointed out that people had a blind belief in the myth of the accuracy-interpretability trade-off, telling that there was a widespread acceptance that the models with high complexity were supposed to have excellent performance [60]. Probably, this is one of the reasons why *Deep Neural Network (DNN)* is applied to many fields and is believed to provide the state-of-the-art performance since it has a high model complexity. Since the purpose is to interpret any kind of black-box models with the assistance of the black-box interpretation framework, it is thereby worthy of investigating the deep neural network. By definition, DNN learns the accurate mathematical transformation from input to output no matter the transformation is linear or non-linear. And there are many types of architectures for deep neural network, e.g. *Convolutional Neural Network* or *Recurrent Neural Network* [61]. Nevertheless, in the following experiments, *Fully Connected Neural Networks* will be investigated due to its "structure agnostic" property, i.e. no special assumptions have to be made about the input data [62]. Therefore, a simple Keras-style fully-connected neural network (or known as Multi-layer Perceptron) was trained on the dataset and the goal was to interpret this black-box model. To be specific, we decided to train a three-layer fully-connected neural network using the Keras library [63]. The choice of hyperparameters was defined in the following.

Parametric setup. In our case, we had to pre-train a neural network and tried to explain the trained model. During the model training process, most hyperparameters were set as default except that we adopted the batch size as 50 and the adam optimizer to minimize the loss function. In addition, l2 regularization was used to avoid model overfitting.

In the pattern mining task, the numeric feature values were split into 10 intervals by equal-frequency discretization to construct subgroup descriptions. And the generalization-awareness interestingness measure was applied to estimate the quality of the discovered subgroups. In particular, the parameter of a=0.5 was utilized for the interestingness measure in order to balance between mean effects and subgroup size. In addition, the maximum depth of the search space was constrained to 3, which meant that the subgroup description could have at most three combinations of selectors. And to improve the subgroup discovery efficiency, the beam search strategy was adopted by default.

Experimental procedure. Since the dataset contained both categorical and numerical data types, it was required to encode the data by performing either label encoding or dummy encoding. Later on, data processing had to be done before training, e.g. missing data removal or data standardization. Then a Keras-style neural network was trained on the processed dataset.

Generally, the variable influence in the black-box model could be interpreted from three viewpoints. Firstly, the global variable importance was examined on the

dataset. It was estimated by the permutation feature importance approach, generating an importance ranking plot. By using this approach, each variable importance was estimated by computing the model accuracy drop after shuffling the selected feature values. Even though the permutation procedure was random, this operation was performed multiple times so that the results could be stabilized by averaging the feature importance overall repetitions. Next, we considered local-level inspections on variables. As an assumption, it was determined to inspect the impact of variable "sex" in the dataset. Naturally, the local interpretation approaches were employed to explain the output of a selected instance. In this case, the contribution of feature "sex" in an individual instance could be estimated by its marginal feature effect, which was denoted as the SHAP value for this feature, calculating by the KernelSHAP method. Subsequently, the meso-level inspection on a variable was cast on. According to the calculation from the previous part, the impact of feature "sex" for each instance was measured locally. By taking the local variable influence as the target concept, the subgroup discovery technique was applied to the adapted dataset. Eventually, some interesting patterns could be discovered where the selected feature had a significant impact.

4.2.3 Results

The experimental results for the first case study were reported as follows, corresponding to the global, local, and meso-level interpretation of the black-box model respectively.

Global interpretation. As seen from Fig 5, each feature importance score measured by the permutation feature importance method was denoted on the plot. Due to the randomness of the permutation, the importance measure was repeated 20 times to obtain a relatively stable result. Hence, each score was composed of two parts, i.e. the average importance score and the variance. From the plot, it can be observed that the "education-num" is the most important feature in this model since the model accuracy drops about 5% on average by permuting this feature. And the importance of attribute "capital-gain" follows behind closely. Also, it is noticed that the attribute "sex" can only reveal a tiny influence on the model's global behavior. In contrast, the attributes "occupation" and "work-class" could even have a negative impact, meaning that the prediction accuracy increases if the corresponding feature values are shuffled. One plausible reason might be that these two features are not given high weights in the black-box model when making the decisions, leading to the result that the model can even gain more accuracy on the shuffled dataset. In addition, the global variable importance can also be evaluated by SHAP feature importance approach, and the results can be found in the Appendix (see Fig 13).

Local interpretation. For the model's local interpretation, KernelSHAP method was applied to explain the local influence of gender. The SHAP value for each feature is regarded as the feature attribution, which represents the importance weight of that feature. Pictorially, a nice reasoning plot showing the feature's local influence on the model prediction is depicted in Fig 6. The SHAP value could be visualized as "forces" and each feature value is a force either increases or decreases the prediction. As can be seen, there is a base value denoted as -1.889, which is the average model

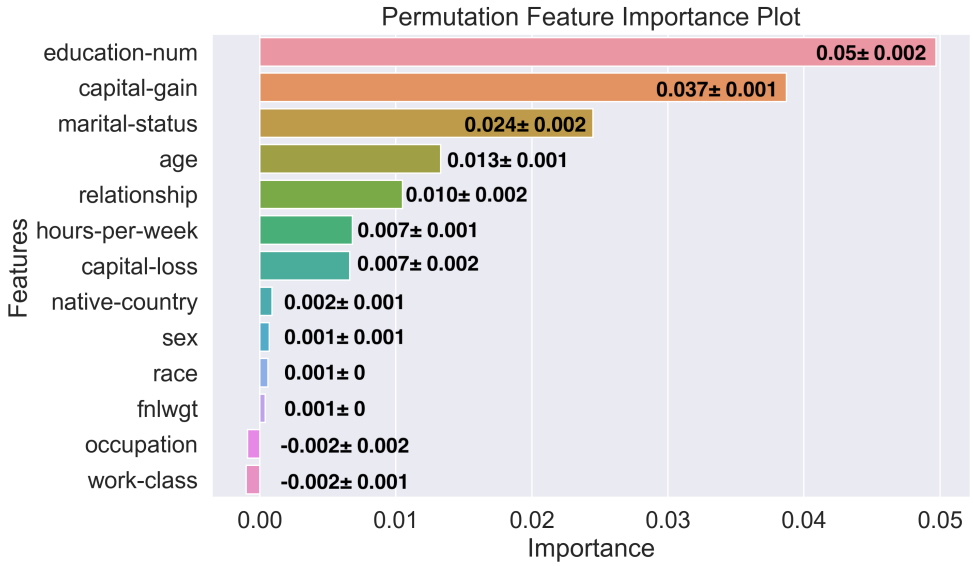

Figure 5: Feature importance ranking plot. The importance score is measured by the permutation feature importance method. A feature is regarded as "important" if prediction accuracy drops extensively after shuffling feature values. Hence, attribute "education-num" is considered as the most important feature in the model globally, causing about 5% accuracy drop. Conversely, the feature "sex" can only contribute tiny influence to the model prediction.

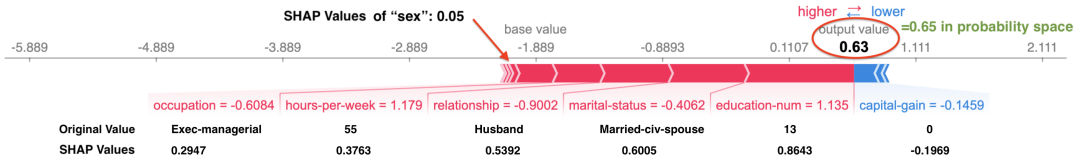

Figure 6: Local variable influence for adult dataset. It interprets the feature influence as attributions to the model prediction, represented by the SHAP value of each feature. The base value is the average model output overall instances but it is calculated in log odds unit, which is -1.889. And the output value is 0.63 in logit space while it is transformed to 65% in probability space. SHAP values are attached for each feature value as well. Feature values shown in red have a positive effect in increasing the chance of earning more than \$50K per year, while feature values marked in blue areas declines the probability.

output over all instances in the logit units. The below explanation shows features each contributing to push the model output from the base value to the actual model output. Features pushing the prediction higher are shown in red, those pushing the prediction lower are in blue. Actually, the "force" of each feature is estimated by the SHAP value. As noticed, the original value and the SHAP value for each feature are also attached in the figure. One more thing needed to be clarified is that all the SHAP values of the selected instance sum up to the difference between the model output (output value) and the expected model prediction (base value). Regarding that, the output value could be calculated by adding the sum of SHAP values to the base value. By convention, the values computed by this approach is in the logit space, therefore, they can be converted to the probability space, e.g. the log odds of the output value is 0.63, but it can be converted to 65% in probability space. This plot also highlights features that largely contribute to the model prediction no matter they are positive contributions or negative contributions, e.g. "education-num" contributes positively to the output while the attribute "capital-gain" has an inverse impact in this case. Besides, our concerned feature "sex" has a positive contribution towards the output with the SHAP value 0.05, which could cause about 1% change in the model prediction.

Meso-level interpretation. In the context of the meso-level variable influence interpretation of a black-box model, the pattern mining technique combined with the KernelSHAP estimation was employed. Basically, the local influence of attribute "sex" of each instance had been measured by the KernelSHAP method, which was further treated as the numeric target in the subgroup discovery task. Then, by applying the pattern mining technique to the adapted dataset, those remarkably interesting patterns could be found. As exhibited in Table 2, the top-ranked subgroups are discovered with respect to the gender effect measurement. And each subgroup is expressed as a combination of selectors, known as subgroup description as well. Specifically, each subgroup is characterized by its size, its average gender effect, and the computed interestingness score. Besides, it can be observed from the table that the above three subgroups have the largest quality score, indicating that gender has an extraordinary impact on those patterns. It turns out that gender has very strong effects for people who are married or serve as a husband in a family. Contradictory to that, another three patterns below the table are listed, where the gender could have inverse influence, especially for housewives in a family or single persons.

Table 2: Top subgroups for the adult income dataset. Each subgroup is described by a conjunction of selectors. Accompanying with the subgroup description, the size of the subgroup (denoted as # Inst.), the average effect of gender in the subgroup (denoted as μ_{gender}), and the interestingness score (denoted as q) for the subgroup are presented. The generalization-awareness interestingness measure is used, where parameter a is set as 0.5. The average effect of gender overall is -0.0001. Among the discovered subgroups, the top three with the largest interestingness score and three subgroups with the smallest quality score are listed. It shows that the gender has a particular strong effect on people who are married or taking responsibility as a husband, but the effect dramatically decreases for people who are unmarried.

Subgroup description	# Inst.	μ_{gender}	q (score)
relationship=Husband	3907	0.020	0.012
marital-status=Married-civ-spouse	4452	0.014	0.009
occupation=Craft-repair	1211	0.013	0.005
...	...	$\mu_\phi = -0.0001$	...
marital-status=Widowed	310	-0.021	-0.004
marital-status=Never-married	3192	-0.009	-0.005
relationship=Wife $\wedge$ marital-st.=Married	484	-0.039	-0.012

4.3 Case study: Amazon Review dataset

In the second case study, a black-box model trained on a textual dataset was explored. Following the same procedure as we did in the previous case-based research, an introduction to the dataset was first demonstrated. Then, we described the experimental setup. It started with the process to pre-train a complex gender classifier based on product reviews. Afterwards, the choice of parameters was notified, followed by a series of steps to perform the experiments. Finally, the results were illustrated.

4.3.1 Textual dataset

As the name suggests, textual data comprise textual resources, which include lexicons, words, or documents. By applying proper processing techniques to the textual data, the meaningful information could be extracted and afterwards a text classifier could be trained. The publicly available data source to be studied is *Amazon Review* dataset. It comprises more than 80 million amazon product reviews commented between May 1996 and July 2014 by about 21 million users. Each review contains a user's demographic information, full review text, and other meta-information about the product, e.g. the product categories. Despite all the information, the essence of future experiments is to build a gender classifier merely based on product reviews, and use it to infer the gender of an anonymous user.

4.3.2 Experimental setup

To began with, instructions were given to tell how to train a gender classifier just based on reviews, which was the model to be interpreted as well. Then the parameters were set up across the entire experiments. At last, the experimental procedure was described step by step.

Black-box model. The applied black-box model in this case is *Gradient Boosting Trees* [64], which construct an ensemble of decision trees to perform classification or regression tasks, where each decision tree is a weak prediction model. However, unlike random forests algorithm that fully grown decision trees are created, in *Gradient Boosting Trees* algorithm, each tree is a shallow tree, sometimes even as small as decision stumps (trees with two leaves). The main idea behind is to add new decision trees to the ensemble sequentially. At each iteration of the training process, those data instances with high prediction errors are emphasized by the next decision tree in order to correct the errors. And the final prediction is determined by the weighted average of each decision tree, where the weight depends on the performance of the corresponding tree [65]. There is a rich variety of libraries that implement the gradient boosting trees algorithms. In the context of this case study, *LightGBM* [66] library was used, which implements fast, distributed, high-performance gradient boosting algorithms. As claimed, it can outperform existing frameworks on both efficiency and accuracy with significantly lower memory consumption [67].

Parametric setup. During training the boosting tree model, it was considered reasonable to set the learning rate as 0.01. And the number of boosted trees to fit was chosen as 500, where each tree had a maximum number of 10 leaves. Other hyperparameters were selected by default. In the subgroup discovery task, the generalization-awareness interestingness measure was also used. Likewise, the parameter a was chosen as 0.5 to balance between the subgroup size and the mean effects. Keep in mind that the maximum depth of the search space was limited to three.

Experimental procedure. Before training a black-box model, the text preprocessing was an essential step. In general, a good pipeline for processing text included the following steps: tokenization, text normalization, punctuation removal, stop words removal, and lemmatization. Tokenization was the method of breaking the text into words or sentences. And text normalization was to convert any non-text information to formal texts, e.g. transforming dates to texts or converting currency sings to texts. Afterwords, the punctuation and stop words were removed from the original texts since they contained less meaningful messages in the context. In addition, each word was lemmatized to keep the root of the word (known as lemma), which was considered as the canonical form of the word. In practice, the pipeline was constructed based on a popular Natural Language Processing library, called spaCy [68]. Actually, one more step was still required after text processing, which was to map the cleaned texts into a fixed-length vector representation owing to that the gender classifier couldn't be trained directly on texts. As disclosed previously, the tf-idf approach was exploited. In this case, a collection of text reviews were converted to a matrix of tf-idf values. Basically, each review was represented as a fixed-length vector corresponding to one row in the matrix, and the feature length was bounded by

the number of unique words in the corpus because each distinct word was regarded as a feature. Besides, in order to avoid the long tail distribution of word frequency, those words that had occurred less than 20 times in the test dataset were excluded from the feature set. Even though a large number of features were ruled out, still there were more than 18 thousand features. Ultimately, a gradient boosting tree model was trained for demonstration purposes.

In the context of gender detection from text reviews, the variable influence in the complex classifier was indeed meant to be the word influence. Therefore, the global word importance was supposed to be inspected at first by exploiting the SHAP feature importance method. In principle, the SHAP value of a selected word could be calculated for each review. Then all the absolute SHAP values across the whole product reviews were averaged and the ultimate score was an indication for this word influence. The word importance ranking could be perceived from a summary plot.

From the local-level interpretation of the model, partial words would be highlighted for any randomly selected review that they influenced the judgment whether the review was written by a female user or a male user. Each word could positively contribute to the model prediction or revealed an inverse impact, where the impact could be visualized in a SHAP force plot. Particularly, the SHAP value of each word evaluated by the KernelSHAP method represented the word influence. In practice, due to the huge sparseness of the matrix converted from the product reviews and the computational limitations from our working stations, the testing dataset was split into multiple chunks and each chunk with 100 thousand instances. Then we exploited the KernelSHAP method to each chunk of the data to obtain the SHAP value for each word. Besides, the word influence could also be interpreted from a local interpretable classifier which was considered as an approximate model to the original model around the given text, i.e. the coefficients in the local explanation model indicated the word effect. It was accomplished by using the LIME approach.

Insight had to be gained from the meso-level interpretation of the model as well. Before that, the original dataset had to be reconstructed first for the preparation of pattern mining tasks. It was stated before that each product review contained meta-information about the product, therefore, product categories could be attached to each review as the meta-data in this experiment. Basically, each product review could belong to several categories. In this case, each review could be represented as a binary vector, where the value was 1 if the review belonged to the corresponding category and otherwise was denoted as 0. And the vector length was bounded by the number of distinct categories. In addition, the label for each review was the SHAP value of the word being inspected, which was considered as the target concept in a subgroup discovery task. By the way, it had to be reminded that we were only interested in reviews where the inspected word indeed occurred. Then a subgroup discovery task could be executed. Eventually, it was expected to see that the selected word could be a strong indicator for the gender if the review was commented under some specific categories.

4.3.3 Results

The experimental results for the second case study were presented in the following. Likewise, the global, local, and meso-level interpretation of the gender classifier based on the text were explored respectively.

Global interpretation. Each word was taken as a single feature in the model, and it was aimed to study the word influence in this case. Obviously, the first step was to gain a global insight into the word influence. As displayed in Fig 7, the present findings suggest that the word "love" has the largest average impact on predicting the gender of a commenter. In other words, on average the chance of the commenter to be predicted as female could increase by 4% in probability space if the word "love" occurs in a text review. Moreover, other words that reveal a huge impact on gender prediction are presented as well. By the way, the word influence was estimated by the SHAP feature importance approach, which was computed by taking the average of the word's marginal effect in each review.

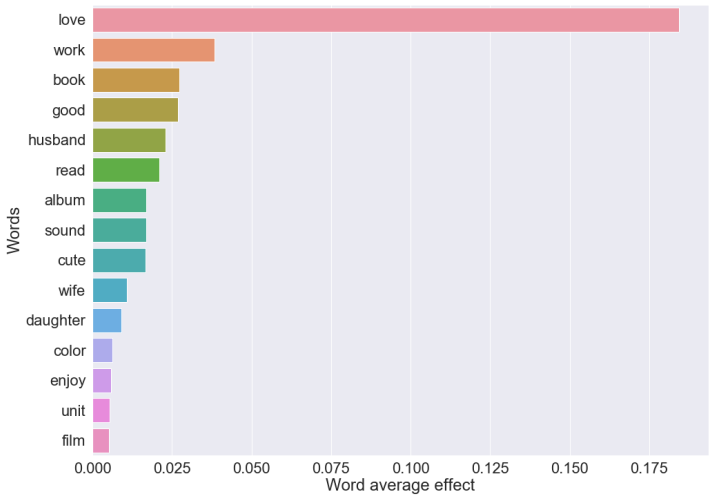

Figure 7: Word influence ranking plot. The importance score is estimated by the SHAP feature importance method. In each review, the SHAP value is calculated for each word. And the word influence is considered "important" if the average word's SHAP value is large. Top important words are highlighted in the plot, e.g. the word "husband" can be a strong indicator for a female commenter.

Local interpretation. To investigate the local behavior of the gender predictor, it was desired to figure out which words could determine gender inferring. To put it another way, the distinctive writing patterns of female and male users might be inferred from the usage of some particular words. An explanatory case is exhibited in Fig 8. As can be observed, the word influence in a text review which is likely to be written by a female user with high probability 0.85 is demonstrated by LIME and KernelSHAP method respectively. Interestingly, from the first two cases, the words "love" and "husband" are both given high weights to predict the gender as female, which are visualized from two different techniques. In contrast, other words

are highlighted in the next two plots, such as "war", "american", which indicate that the review is likely to be commented by a male user. Fortunately, the explanation of the word influence in a local text review interpreted by two techniques is quite consistent, which can be assumed as a reliable interpretation of the model's local behavior.

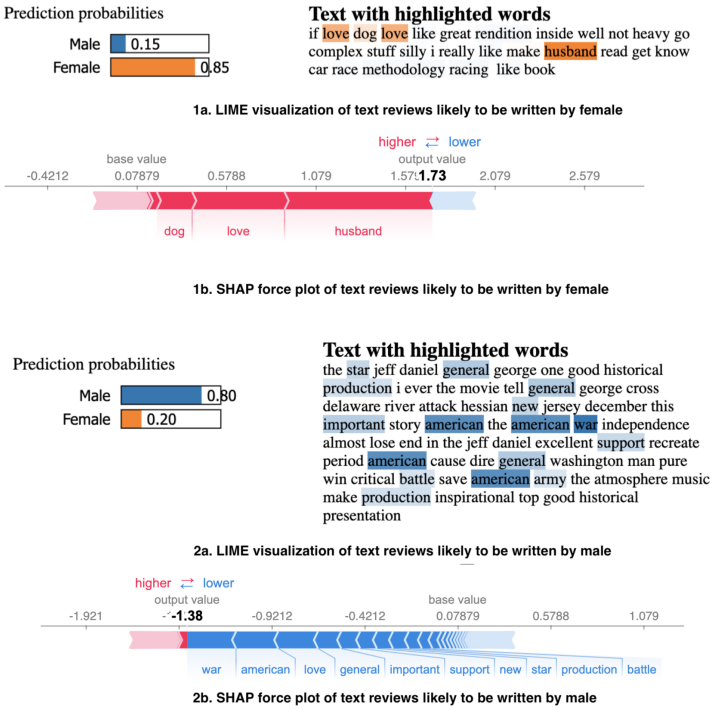

Figure 8: Representations of local word influence. 1a)1b): LIME visualization and SHAP force plot for a review likely written a female user. In this review, the word "love" and "husband" are given high weights to infer a female writer. 2a)2b): LIME visualization and SHAP force plot for a review likely written a male user. In this case, the word "war" and "american" are a strong indicator for a male commenter.

Meso-level interpretation. From the previous experiment, the word influence in each review had already been estimated by the KernelSHAP method, i.e. represented as the SHAP value for each word. Therefore, in the subgroup discovery task, the word influence was considered as the target for the mining task. The results are described in Table 3. On the left part of the table, the findings are based on the influence of the word "Husband", which is frequently used by female users. As noticed, the average weight assigned to the word "Husband" is 0.0765 taking all product reviews into account. But if the word occurs in a product review and the product belongs to the categories appeared in the top three subgroups such as "Health & Care" or "Dietary Supplementary", then the gender classifier would assign higher weight to the word, making it become a stronger indicator for a female commenter. In contrast, the word "Build" is considered as a signal pointing at a male writer, which can be told from the negative average weight assigned to the word, i.e. -0.0026. Likewise, we can draw a conclusion that the word "Build" can be a determining factor if a product review contains the word and the corresponding product is under categories like "Computers" or "Batteries". In the other

Table 3: Top subgroups for the amazon review dataset. For each discovered pattern, we have presented its subgroup description, its size (denoted as # Inst.), and the average weight assigned to the selected word in the subgroup (denoted as $\mu_{husband}$ or μ_{build}). The generalization-awareness interestingness measure is used, where parameter a is set as 0.5. The average weight assigned to the word "Husband" overall 0.0765, and it is -0.0026 for the word "Build". The word "Husband" would become a strong indicator for female commenter if the word occurs in a product review and the product is under categories like "Health & Care" or "Dietary Supplementary". Conversely, the review is likely to be written by a male user if the word "Build" appears and the review is commented below categories such as "Computers" or "Batteries".

Word: Husband $\mu_\phi = 0.0765$			Word: Build $\mu_\phi = -0.0026$		
Description	# Inst.	$\mu_{husband}$	Description	# Inst.	μ_{build}
Health & Care=1	815	0.0978	Computers=1	1367	-0.0045
Dietary Supp.=1	133	0.1416	Cases & Sleeves=1	120	-0.0054
Accessories=1	609	0.0907	Batteries=1	87	-0.0046
...	...	...	...	...	...
Jewelry=1	242	0.0625	Romance& Books=1	267	-0.0011
Kindle Store=1	435	0.0667	Movies & TV=1	717	-0.0017
Books=1	5140	0.0663	Books=1	7526	-0.0023

way around, if the product belongs to genres like "Books", the word becomes less influential.

4.4 Case study: Diamonds dataset

Indeed, classification and regression are two main sub-problems under the same umbrella of supervised learning. The main difference between them is that the output in regression is numerical (or continuous) while that for classification is categorical (or discrete). From a mathematical perspective, supervised machine learning defines a mapping function from the input variable X to the output variable Y. An algorithm is then employed to learn the mapping function $X \rightarrow Y$ by solving a classification problem if Y is discrete and a regression task if $Y \in R$ [69].

From the pipeline of the framework (see Fig 2), it is clear that we can exploit the framework to interpret a black-box classifier or a regressor, which is trained depending on the dataset. From previous case studies, we have already seen how to apply the framework to a complex classifier whether the input is tabular or textual data. Subsequently, experiments were performed to testify the effectiveness of the framework when it tried to solve a regression task.

In this context, we decided to use a publicly available dataset, named *Diamonds*. It consists of over 50 thousand diamonds, and each diamond has 10 distinct variables,

which belong to either numerical or categorical data type. One prominent feature is "Carat", representing the weight of the diamond in a special unit. More features like the color or dimension are also given for each diamond. Based on those features, the task is to predict the price of the diamond, which is considered as a regression task since the price is a continuous value. After encoding the dataset, a Keras-style neural network with three layers was trained. For parametric setup, the mean squared error was adopted as the training metric and all training examples were fed to the model for 100 epochs with each batch size as 50. In addition, the learning rate was set by default.

Afterwards, the local interpretation of the model on a chosen instance was expressed in Fig 9. As can be observed, the four features marked on the plot have a large impact on the price prediction. In particular, the features shown in red devote positive contributions to the price, such as "carat", "volume" and "color". In contrast, the feature "clarity" decreases the price prediction. The original value corresponding to each feature is denoted in the figure, as well as the SHAP value that is estimated by the KernelSHAP method. The base value -0.039 represents the average diamonds price where the price has been standardized. Therefore, after inverse-scaling on the output value and adding back the mean of the scaler, the predicted diamond price is about 10600, which is approximate to the diamond's true price 11743.

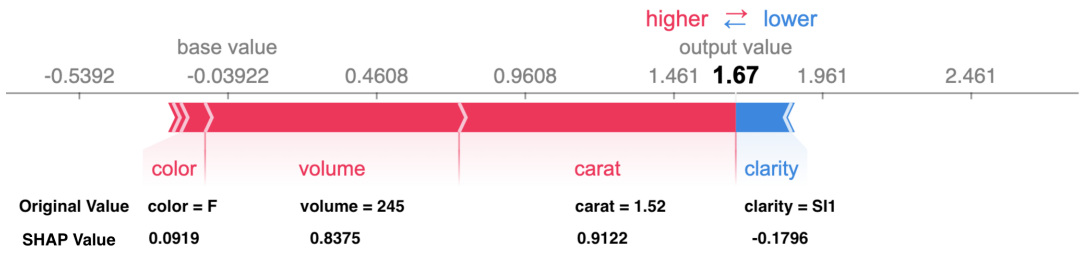

Figure 9: Local variable influence for diamonds dataset. It interprets the feature influence as attributions to the model prediction, represented by the SHAP value of each feature. Original value, as well as the SHAP value for each feature, are denoted in the plot. The base value is the average diamond price where the price has been standardized, which is -0.039 but it can be transformed to 3775 after inverse scaling. Likewise, the predicted diamond price for this instance is 10600 after proper transformation. Feature values shown in red have a positive effect in increasing diamond price, while feature values marked in blue areas decrease the price.

We were particularly interested in how does the volume of a diamond influence the price. In this case, we performed the pattern mining task based on the local influence of the attribute "volume". By default, we employed the generalization-awareness interestingness measure with parameter $a = 0.5$. Table 4 presents the top-ranked patterns where feature "volume" has exceptional behaviors. It is observable from the results that if the carat of a diamond is high, usually the local effect of "volume" is also large. The above finding is understandable because a diamond with more carat usually indicates the larger volume, which also implies a higher price. Conversely, in the pattern where the carat is very small, the local influence of "volume" also

decreases, leading to small contributions to the price. Surprisingly, the color of the diamonds can affect the influence of "volume", causing lower diamond price.

Table 4: Top exceptional patterns for the diamonds dataset. Each subgroup is described by a conjunction of selectors. Accompanying with the subgroup description, the size of the subgroup (denoted as # Inst.), the average effect of "volume" in the subgroup (denoted as μ_{volume}), and the interestingness score (denoted as q) for the subgroup are presented. The generalization-awareness interestingness measure is used, where parameter a is equal to 0.5. The average effect of "volume" overall is -0.059. The attribute "volume" has particular strong impacts on diamonds whose carat is large while the effects are small if the diamonds have little carat.

Subgroup description	# Inst.	μ_{volume}	q (score)
carat>=1.50	1830	1.100	0.192
carat: [1.12:1.50]	1457	0.405	0.059
clarity=SI2	2811	0.226	0.056
...	...	$\mu_\phi = -0.059$	...
color=D	2067	-0.229	-0.044
color=E	2915	-0.225	-0.057
carat: [0.31:0.35]	1867	-0.465	-0.083

4.5 Experimental variations

More experiments with variations were conducted in this section. First, a comparison between various local interpretation methods was illustrated. Then, the attention was paid to the patterns that were discovered using different interestingness measures. Moreover, the decision tree model, as an alternative approach to mine local patterns, was explored.

4.5.1 Comparison of various local interpretation methods

From the pipeline of the black-box interpretation framework, it is noticed that there are several options to choose the local interpretation methods. In the following, a comprehensive comparison between these local interpretation methods would be presented. Keep in mind that the parametric setup is the same as in the first case study (see Section 4.2).

The first local interpretation method is the binary flip approach, which is only applicable to a binary attribute. The impact of a binary feature is estimated by the model prediction change when the flip operation is performed on that feature. Assumed that the binary flip approach was used on a selected instance, e.g second instance from the testing dataset, the model prediction could be easily observed. Before flipping operation, this instance was predicted to have 65% chance to earn more than 50K$ per year, however, the chance reduced to 61% after changing the gender.

Overall, the average model prediction change across the entire testing dataset was 0.026. Thus, it could roughly argue that gender might influence the model prediction by 4% on the chosen instance.

An alternative approach to inspect the impact of gender on an instance was using LIME technique. After fixing the instance of interest, it was required to generate neighborhood observations by perturbing around the original instance. Normally, they were perturbed by sampling from a Normal(0,1) distribution and performed the inverse operation of mean-centering and scaling, according to the means and standard deviations in the training dataset. Additionally, those sampling instances were weighted according to the proximity, which was estimated as the distance to the original instance. Afterwards, a weighted intrinsic interpretable model was fitted with an R-squared score of 0.76. And the coefficients of this estimated linear model could be used to understand how changes in the variables affected the model prediction for the instance being explained. As displayed in Fig 10, the weights of each feature are presented. In general, the classifier predicts that the selected person has 65% chance to earn an annual income more than 50K$. From the middle part, it can be noticed that the most striking feature causing the prediction to lean towards the label "income<50K" is attribute "capital-gain", while attributes "education-num" and "hours-per-week" have inverse impacts with weights 0.07 and 0.05 respectively. And one plausible interpretation for the feature weight of "education-num" could be that if one unit is increased on this feature, the chance of the person to make more money would increase by 7%. Nevertheless, the feature "sex" is not found in the plot, which means that this feature has a tiny influence so that it is discarded from the feature selection procedure. Therefore, the impact of feature "sex" is considered as 0 in this case. Moreover, the feature weights inflected the local variable influence on the instance, which was a comprehension of the model's local behavior. And on the right side of the figure, the feature value pairs of the instance to be explained are displayed in a table format. The feature column shows the feature names, the value column displays the values after processing, and another column is attached to show the original feature values.

Another explanation method called KernelSHAP was also adopted to interpret the local behavior of gender, and the results were already shown in the previous section, as depicted in Fig 6. The finding shows that the attribute "sex" has a positive impact on the model prediction.

After the local influence of attribute "sex" was measured by previously mentioned methods, the subgroup discovery task was executed to uncover interesting patterns. As exhibited in Table 5, the five top-ranked subgroups are discovered with respect to the approaches that are used to estimate the gender effect. To be more clear, the first column shows the highly ranked subgroups where gender has an extraordinary impact when gender influence is measured by the binary flip approach. Likewise, the following two columns uncover the most interesting subgroups when the gender effect is estimated by LIME or KernelSHAP approach respectively. As can be observed from the table, there is a high correlation between the patterns that are discovered by employing the binary flip and KernelSHAP method, i.e. the top two subgroups are the same. Surprisingly, the patterns discovered through LIME and KernelSHAP approach are nearly the same, with only one tiny difference in the

ranking of subgroups. It seems the results are quite good, which makes us more skeptical about them. Therefore, we further conduct an experiment to calculate the similarity between the results generated by different approaches by using the rank-biased overlap metric.

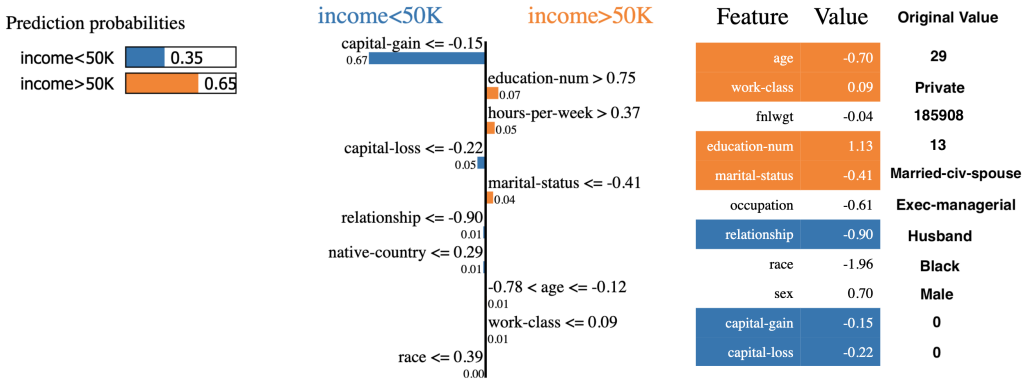

Figure 10: LIME local interpretation plot. On the left side, the prediction probability is shown, telling that the selected instance is predicted to have 65% chance earning more than 50K$ per year. In the middle, the feature weights in the fitted model are displayed. The most striking feature caused the positive results is the attribute "education-num" while the feature "sex" is regarded as no impact in this case. Note that the feature value pairs of the instance being explained are listed on the right side.

Table 5: Top subgroups for the adult income dataset. The target concept in the subgroup discovery task is the gender effect, which can be estimated by several approaches. The generalization-awareness interestingness measure is applied, where parameter a is set as 0.5. The first column shows the highly ranked subgroups where gender has an extraordinary impact when gender influence is measured by the binary flip approach. Similarly, the next two columns show the subgroups that are discovered with respect to the approaches to measure the gender effect.

Binary flip approach	LIME approach	KernelSHAP approach
relationship=Husband	relationship=Husband	relationship=Husband
marital-status=Married	marital-status=Married	marital-status=Married
education-num>=13	occupation=Craft-repair	occupation=Craft-repair
occupation=Prof-specialty	occupation= Transporter	hours-per-week>=55
occupation=Manager	hours-per-week>=55	occupation=Transporter

Rank-biased overlap (RBO) is a similarity measure for indefinite ranking lists, which was proposed by Webber et al. [70]. It compares two ranked lists and return a value in the range between 0 and 1, where a RBO value 0 implies that the two lists are totally distinguished, and a RBO value 1 means identical. For practical usage, the traditional RBO measure with proper extrapolation is applied to give us a point estimate. During the computation, the choice of parameter p is of critical importance, which determines the degree of top-weightedness of the RBO metric.

For example, p=0.8 indicates that the top-five ranks are assigned 86% of the weight and p=0.9 tells that the top-ten subgroups are given 86% of the weight. To provide the top-thirty subgroups with the same weight, p=0.95 should be set. We have employed the RBO similarity measure to two list of ranked subgroups where the target (local variable influence) is measured by KernelSHAP and LIME respectively. The results are depicted in Fig 11. It can be observed the RBO score is about 0.9 when the gender effect is investigated, indicating a high similarity in the two list of subgroups, which also coincides with the result in Table 5. However, the meso-level interpretation of the influence of features such as "work-class" or "race" can result in two completely different list of subgroups, with the RBO score 0. Generally, it can be assumed that the subgroup discovery results are relative stable for some specific feature's influence measure but they are not always consistent as we expect. In addition, the RBO measure between two lists of subgroups concerning the KernelSHAP or binary flip approach can be found in the Appendix (See Fig 14).

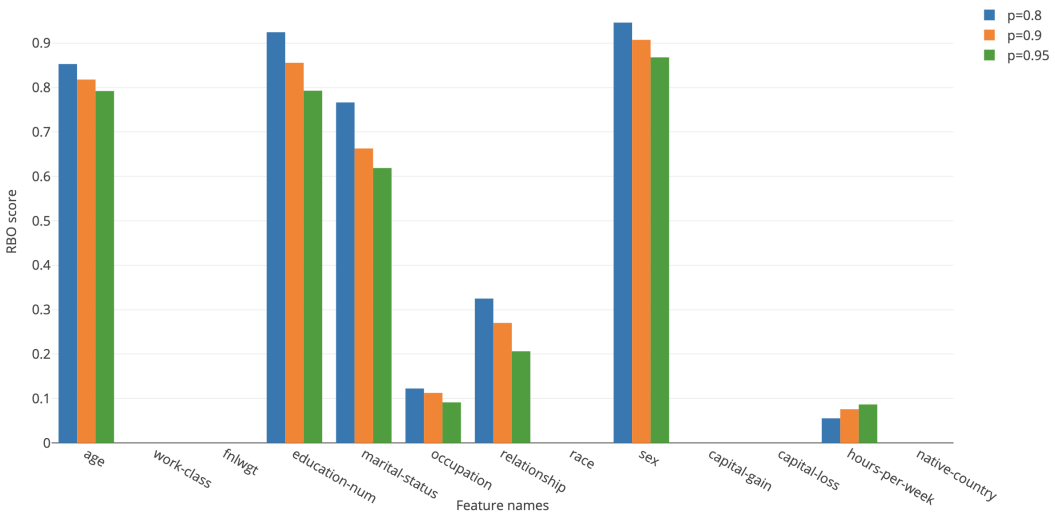

Figure 11: Rank-biased overlap between two list of ranked subgroups. The subgroups are discovered with respect to a local variable influence which is either measured by KernelSHAP or LIME approach. For each feature, we obtain two lists of top-ranked subgroups, and the RBO similarity score is calculated. Parameter p=0.8 indicates that the top-five ranks are assigned 86% of the weight; p=0.9 implies the top-ten with the same weight; p=0.95 involves the top-thirty subgroups. The discovered subgroups are pretty similar for the influence measure of some specific feature, e.g. "sex", having RBO score about 0.9. But they can be completely different for some features, e.g. "race", with RBO score 0.

4.5.2 Comparison of different interestingness measure

It is observable that patterns discovered in the subgroup discovery task are distinct by employing different interestingness measures. In this black-box interpretation framework, two groups of quality measure families are provided. The standard one is the generic-mean based interestingness measure, whose formula is denoted in Eq. 4. However, it brings out a huge concern about the redundancy in the subgroup

descriptions. For example, one case is that pattern P is considered as an interesting subgroup by just taking the quality score into account, as well as its generalization S. In this case, both patterns P and S will be included in the final outcomes, nevertheless, the pattern P doesn't reveal any useful information. Therefore, to avoid including such subgroups in the result set, the generalization-awareness interestingness measure could be employed instead. It can filter out redundant subgroup descriptions in the results by considering the statistics of the subgroup as well as to its all generalizations. Mathematically, the mean-based generalization-awareness quality measure is defined in Eq. 7.

Table 6: Top exceptional patterns using different interestingness measure. Each subgroup is described by a conjunction of selectors. Table 6a shows patterns discovered by applying the standard generic-mean function as the interestingness measure, while the subgroups extracted by employing the generalization-awareness interestingness measure are presented in Table 6b. Experiments are performed when the parameter a is chosen as 0.1, 0.5, 1.0 respectively.

(a) Standard generic-mean based interestingness with different parameter a. Notice that "rel=H" is the abbreviation for "relationship=Husband"

$a = 0.1$	$a = 0.5$	$a = 1.0$
rel=H ∧ edu-num>=13	rel=H ∧ gain>=0	rel=H ∧ loss>=0
rel=H ∧ edu-num>=13 ∧ gain>=0	rel=H	rel=H
rel=H ∧ edu-num>=13 ∧ loss>=0	rel=H ∧ loss>=0	rel=H ∧ gain>=0
rel=H ∧ marital=Married ∧ edu-num>=13	rel=H ∧ gain>=0 ∧ loss>=0	rel=H ∧ loss>=0 ∧ gain>=0

(b) Generalization-awareness interestingness with different parameter a

$a = 0.1$	$a = 0.5$	$a = 1.0$
relationship=Husband	relationship=Husband	relationship=Husband
marital-st.=Married	marital-st.=Married	marital-st.=Married
occupation=Craft-repair	occupation=Craft-repair	occupation=Craft-repair
occupation=Armed-Force	hours-per-week>=55	hours-per-week>=55
occupation=Transporter	occupation=Transporter	hours-per-week=[50:55]

Table 6 displays our findings when we apply different interestingness measures in the subgroup discovery task. As seen from Table 6a, the patterns found through the standard generic-mean based quality measure are reported regarding the choice of parameter a. Specifically, the experiments were performed when parameter a was chosen as 0.1, 0.5, 1.0 respectively. As a notice, the pattern "relationship=Husband"

is abbreviated as "rel=H" due to the limited space. Comparing the results of different choice of a, the subgroup descriptions are quite similar despite different orderings. Therefore, we argue that the choice of a won't influence the result sets to a large degree. But for a specific setting of a, we notice that the discovered patterns are very redundant. Basically, only the general pattern "relationship=Husband" is uncovered and the rest are just its specifications. In this way, no additional information is gained, which is a bad practice. Hence, we could improve the result sets by employing the generalization-awareness quality measure, whose experimental results are described in Table 6b. As can be observed, there are less redundancy in the patterns and the subgroup description are more diverse. From a horizontal comparison of the results, the top three subgroups are exactly the same even though different parameter a is selected. All of them suggest that gender has a strong impact when the patterns "relationship=Husband" or "marital-status=Married" appear. It can be explained by the fact that the marriage status and the gender of a person are determining factors to the person's annual income. By comparing these two tables, we can argue that the generalization-awareness interestingness measure is more suitable for the task since more information could be revealed and the redundancy in patterns is avoided as well.

4.5.3 Subgroup discovery vs. Decision trees

For the purpose of explaining variable impact from a new viewpoint namely the meso-level interpretation, the proposed approach which combines the local interpretation methods and pattern mining technique is explored. By combining the subgroup discovery technique, the aim is to discover interesting patterns from the dataset with respect to the target, which is the influence score of the selected attribute in this case. An interpretation for those discovered subgroups is that the selected feature could have a significant impact on them, telling how the feature influences the model decisions. Nevertheless, not only the subgroup discovery technique can be utilized to discover patterns, but the decision tree algorithm is also capable to mine local patterns through its decision path, where each path is traversed from the root node to a leaf node. In addition, the decision tree model itself is an interpretable model, which can provide evidence about how features influence the model decisions. Thus, it is worth to compare the results from both techniques and to see whether there are similarities or differences. The Adult Income dataset was experimented in this case, and the local influence of gender was measured by the KernelSHAP method.

Notice that before training a decision tree model, the categorical features in the available dataset were pre-processed by label encoding. Then, a simple decision tree model was fitted using CART algorithms in order to identify the split patterns with regard to the influence score of gender. However, due to the limitation of space, only the first three layers of the decision tree graph were visualized (see Fig 12). The results suggest that the first optimal feature to be split is "relationship", which is a categorical feature before label encoding. By transforming labels back to original encoding, we notice that "relationship=Husband" is the first split node. As clarified before, a local pattern could be assumed as the decision path from the

root node to the leaf node, and a decision is made on each internal node. For example, as indicated by the red arrows, one interesting pattern could be formatted as "relationship=Husband AND age>36.5 AND hours-per-week>34.5". For data instances complied with this pattern, the average effect of gender is 0.066, which was significantly larger than the mean effect over the entire dataset. To put it another way, the attribute "sex" could positively influence the model decision to a large degree for instances within this subgroup, i.e. increasing the chance to earn more than 50K$ per year. Conversely, a pattern discovered by following the rightmost decision path reveals a negative impact on the result.

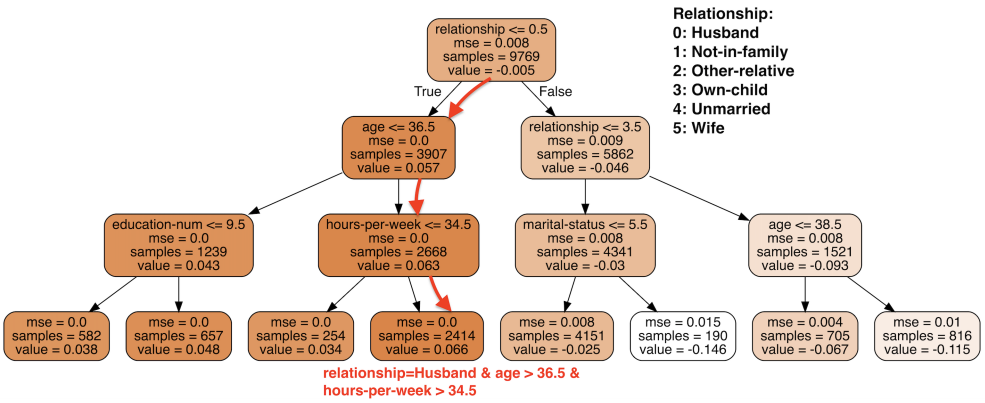

Figure 12: Patterns visualized by the decision tree model. The decision tree is built using the CART algorithm and the Gini index is regarded as the node splitting criteria. Each local pattern is discovered by following a decision tree path. Hence, following the red arrows, an interesting pattern "relationship=Husband AND age>36.5 AND hours-per-week>34.5" is discovered with the average gender effect significantly deviating from the mean effect that is calculated over the entire dataset.

Comparatively, the results discovered by pattern mining technique have already been illustrated in Table 2, showing the top-ranked interesting patterns. One thing to note is that a feature selector *relationship=Husband* can lead to a significant increase in the average gender effect. Interestingly, a similar pattern is discovered by both techniques. From the decision tree model, this pattern is considered as the first optimal split, while it is also the most interesting subgroup discovered by pattern mining task.

5 Discussions

In this section, we will summarize our experimental results and present the underlying meaning of our research findings based upon a logical analysis. Additionally, we will explore possible improvements that can be made to further deal with our research concerns.

5.1 Implications

The global interpretation methods can identify important features globally from the model while the local interpretation methods are targeted at the instance level interpretation of the variable influence. Nevertheless, the insight gained from the global view is too vague and the variable influence obtained from the local view might be too brittle due to the excessive local interpretation. In order to fill the gap, we are passionate about creating a novel interpretation viewpoint somewhere between global and local, i.e. meso-level interpretation. It is achieved by combining the local interpretation methods with the subgroup discovery technique, which is a descriptive data mining method. With this approach, we are able to detect subgroups with exceptional behaviors. In addition, the pattern mining technique has never been explored to explain black-box models, which is worthy of investigating. And our major contribution is that we have successfully constructed a unified black-box interpretation framework that covers three interpretation views. The effectiveness of our proposed framework has been verified in our experiments.

In Section 4.1, an evaluation for a synthetic dataset was performed. By employing the meso-level interpretation on feature "sex", we could successfully recover the desired pattern referring to the gender effect. As a result, we could argue that the effectiveness of the meso-level interpretation can be proved and it indicates that we can indeed obtain a deeper insight into the variable influence through those exceptional patterns.

Next, we performed a case study research on a tabular dataset, i.e. Adult Income dataset. The experimental results had been shown in Section 4.2.3. Again, we were passionate about the gender effect in the black-box model. In the context of the meso-level interpretation of gender's influence, the top-ranked subgroups were discovered in Table 2. As a consequence, the effect of gender is significantly large for people who are married or take responsibility as a husband in a marriage. In contrast, if a person is single or being as a wife, the attribute gender could have a negative impact. In this case, it is believed that married men can earn more money with a high probability. A similar finding could also be found in paper [71], but they just manually define the subgroup by clustering the local interpretation results, which fails to perform the automatic subgroup discovery.

Afterwards, our next case study was researched on a textual dataset to infer a user's gender. Likewise, it is also possible to infer gender from movie reviews [72], Tweets [73], or user names [74]. And in our case, we aimed to inspect the word influence in a gender classifier that was trained on amazon product reviews. The experimental

results indicate that some words in reviews are given high weights to infer a female user, such as "husband" and "hair". Conversely, words like "war" or "build" are a strong indicator for a male user (see Section 4.3.3). Furthermore, the meso-level word influence can be interpreted. Take the influence of the word "build" as an example. Interesting results are found that the word "build" has an extraordinary impact on gender referring if the review is commented under the categories "Computers" or "Batteries". To put it another way, a product review about computers or batteries is highly likely to be written by a male user if the word "build" appears. It is understandable because males are more likely to buy electronics and give feedback. However, we notice that the different choice of the interestingness measure in the subgroup discovery task would cause the change of the output patterns. And it's a pity that we have no advanced method to determine the quality measure, leading to the instability in the results.

Another case study was conducted on a regression dataset. The objective was to predict a diamond's price based upon its geometric attributes. To understand the price fluctuation regarding the influence of a diamond's volume, it is discovered that the impact of attribute volume is particularly large for diamonds with high carat. But if a diamond has a "bad" color, the influence would drop dramatically, leading to lower price evaluation (see Section 4.4).

A further study was performed to compare the various local interpretation methods. For the meso-level interpretation of gender, we could apply either the KernelSHAP or the LIME approach to estimate the local gender effect. After that, we could obtain two lists of subgroups and these subgroups are observed to be quite consistent regarding the gender influence, meaning the rank-biased overlap score is high. Nonetheless, the two lists of ranked subgroups might be totally different concerning other variable's influence, e.g. the RBO score between two lists of subgroups concerning the influence of attribute "race" is 0. As seen in Fig 11, the RBO scores are pretty high for some features, meaning that the results are quite consistent. But for others, the RBO score can be very low. A similar phenomenon has also been found when we apply the meso-level interpretation of variable influence to another dataset. Interestingly, it is often found out that the RBO score is relatively high for those features with high global feature importance scores. It implies that the results generated by the meso-level interpretation approach are more stable when we try to interpret the influence of a feature that is considered as an important feature from a global interpretation.

Moreover, we have a comparison between the subgroup discovery and the decision trees. Technically, both of them can be applied to discover local patterns, but the former one contains patterns with distinct attribute selectors while the latter one could have redundant attribute selectors in one decision path. Basically, the subgroup discovery is an enumeration or a search problem such that we could find the optimal results with a certain interestingness measure. In contrast, the decision trees use the greedy algorithm and the results might be just local optimal but not globally. As a consequence, it is observed that the local patterns discovered from the two techniques are not quite consistent except for a pattern "relationship=Husband". It serves as the first optimal split in the decision trees, meanwhile it is also discovered as the most interesting subgroup. It is still questionable which results are more

convincing, but we argue that the subgroup discovery has its advantage to find our desired patterns. Hence, the subgroup discovery is adopted in our meso-level interpretation to discover local patterns regarding the feature influence.

5.2 Limitations

However, the presented results are not perfect. Taking that into account, a few limitations of this work has to be stated. As clarified before, we are only focusing on the tabular data and textual data, which means the interpretation framework does not apply to the image data currently. Therefore, the extensibility of the framework is one concern. Additionally, only the classification or regression tasks are considered, otherwise they cannot be handled by the framework. Although it claims that our framework could interpret any model-agnostic black-box model, the full support for the deep neural networks and other very complicated models are not implemented. In particular, the local interpretation of deep neural networks is not natively implemented in LIME. As for the KernelSHAP approach, the interpretation of the SHAP value is a challenge, leading to some confusions in understanding the patterns discovered by the meso-level interpretation approach. And one more thing needs to be considered is the dimension of the input features in the model. For example, if a complex model has high-dimensional input features, the feature influence could be infeasible to interpret since each feature might contribute only a tiny part to the model prediction. Hence, it remains a challenge for the complex model with high-dimensional features, namely the scalability problem. Nevertheless, the biggest challenge is how to verify the rationality of our explanations. For example, there are chances that the discovered patterns are distinct if we employ different feature's local influence measure, but we cannot determine which explanations are more reasonable. Furthermore, our framework also suffers from the stability problem, e.g. the randomness in the permutation-based methods or the parameter choice in the associated methods could lead to completely different results. And the prevalent shortcomings in permutation-based methods should not be ignored, which refers to the invalidity of the permuted instances.

Despite all the limitations, we still believe that our research could help fill the gap in the literature and our interpretation framework is effective to facilitate non-expert to understand the model decisions. It is also a good pointer for further research.

6 Conclusion

In this thesis, we propose an overarching interpretation framework to establish a way to explain the variable influence of a black-box model from comprehensive perspectives, i.e. global-level, meso-level, and local-level interpretations views. Ideally, it supports the interpretation for any classifiers or regressors that are trained on tabular dataset or textual data. Besides, various approaches for each interpretation view are implemented.

For the global interpretation of feature influence, two methods are mainly developed, including the permutation feature importance measure and SHAP feature importance measure. Additionally, to leverage the capabilities of inspecting the local variable influence, multiple algorithms are developed as well. Simply, the binary flip or numeric perturbation approach is employed by performing a binary operation, and the difference of the model prediction change for each instance is regarded as the local influence for the specified attribute. More properly, the idea that we train an interpretable model to locally approximate the black-box model is achieved using LIME method. And the feature influence is assessed by the weights in the fitted linear model. Alternatively, a promising method called KernelSHAP is also integrated into the framework, which assigns each feature an importance score. Each score is calculated as the SHAP value, which implies the variable importance degree. Lastly, the meso-level interpretation of the model consists of two stages. The first step is to measure the selected feature's local influence for each instance using one of the aforementioned local interpretation methods. And the next step is to employ the pattern mining technique to discover unusual patterns where the selected feature has a significant impact. And those patterns can help us to better understand the model decisions. More importantly, the meso-level interpretation of the feature influence is our major contribution in this thesis and this approach can provide more fine-grained analyses than global feature interpretation and more reliable explanations than the local variable interpretation.

In the experiments, we have already demonstrated the potential usage of our proposed interpretation framework by employing it to both the synthetic data and the real-world data. From the evaluation of the synthetic dataset, we could successfully recover the artificial pattern that reveals remarkable feature impacts. And the effectiveness of the interpretation framework has also been verified in our real-word case studies. The first case study aims to interpret a neural network model that is trained on a tabular dataset, while the second case focuses on the interpretation of a text classifier. In addition, the framework could also be extended to explain the variable influence in a regressor, which is presented in the third case study.

In the future, one of the possible directions of further work is extending the black-box interpretation framework such that it can interpret image classifiers. And more efforts could be devoted to supporting the explanation of deep neural networks due to their wide usage. Besides, a huge challenge concerning the scalability of those interpretation methods should be discussed, e.g. the computational efficiency. Furthermore, the stability of the explanations provided by the framework is worthy of further exploring. Regarding the effectiveness of the pattern mining technique,

improving the interestingness measures for automatic discovery is still a challenge. Thus, the challenges could serve as a good starting point for future research in this area.

Acknowledgement. First, I would like to express my sincere gratitude to my advisor Florian Lemmerich for the continuous support of my master thesis and research, for his patience, motivation, and immense knowledge, also to my supervisor Prof. Markus Strohmaier for his insightful comments and encouragement, and last but not least to the second supervisor Prof. Bastian Leibe for his kindness and support. Their guidance helped me in all the time of research and writing of this thesis. Second, I owe the success of my master study and this thesis to my parents, Xuyong Li and Xianlin Ma, and my girlfriend friend, Mingjiao Zheng. They supported me through the difficulties that I encountered during the work.

References

[1] B. Kim, R. Khanna, and O. O. Koyejo, "Examples are not enough, learn to criticize! criticism for interpretability," in *Advances in Neural Information Processing Systems*, 2016, pp. 2280–2288.

[2] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, "Interpretable machine learning: definitions, methods, and applications," *arXiv preprint arXiv:1901.04592*, 2019.

[3] M. Goddard, "The eu general data protection regulation (gdpr): European regulation that has a global impact," *International Journal of Market Research*, vol. 59, no. 6, pp. 703–705, 2017.

[4] D. Alvarez-Melis and T. S. Jaakkola, "On the robustness of interpretability methods," *arXiv preprint arXiv:1806.08049*, 2018.

[5] C. Molnar, *Interpretable Machine Learning*, 2019, https://christophm.github.io/interpretable-ml-book/.

[6] A. A. Freitas, "Comprehensible classification models: a position paper," *ACM SIGKDD explorations newsletter*, vol. 15, no. 1, pp. 1–10, 2014.

[7] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *arXiv preprint arXiv:1808.00033*, 2018.

[8] D. Sarkar, "The need and importance of model interpretation," 2018, accessed: 2019-11-10. [Online]. Available: https://www.kdnuggets.com/2018/06/human-interpretable-machine-learning-need-importance-model-interpretation.html

[9] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[10] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC bioinformatics*, vol. 8, no. 1, p. 25, 2007.

[11] A. Fisher, C. Rudin, and F. Dominici, "Model class reliance: Variable importance measures for any machine learning model class, from the" rashomon" perspective," *arXiv preprint arXiv:1801.01489*, 2018.

[12] S. M. Lundberg, B. Nair, M. S. Vavilala, M. Horibe, M. J. Eisses, T. Adams, D. E. Liston, D. K.-W. Low, S.-F. Newman, J. Kim *et al.*, "Explainable machine-learning predictions for the prevention of hypoxaemia during surgery," *Nature biomedical engineering*, vol. 2, no. 10, p. 749, 2018.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.

[14] M. Robnik-Šikonja and I. Kononenko, "Explaining classifications for individual instances," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 589–600, 2008.

[15] I. Kononenko *et al.*, "An efficient explanation of individual classifications using game theory," *Journal of Machine Learning Research*, vol. 11, no. Jan, pp. 1–18, 2010.

[16] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017, pp. 4765–4774.

[17] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *International Conference on Computer Vision*, 2017, pp. 618–626.

[18] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning*, 2017, pp. 3319–3328.

[19] M. T. Ribeiro, S. Singh, and C. Guestrin, "Model-agnostic interpretability of machine learning," *arXiv preprint arXiv:1606.05386*, 2016.

[20] M. Ancona. (2019) Deepexplain: attribution methods for deep learning. [Online]. Available: https://github.com/marcoancona/DeepExplain

[21] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, "Towards better understanding of gradient-based attribution methods for deep neural networks," *arXiv preprint arXiv:1711.06104*, 2017.

[22] S. S. Marco Tulio Ribeiro and C. Guestrin. (2019) Lime: Explaining the predictions of any machine learning classifier. [Online]. Available: https://github.com/marcotcr/lime

[23] S. M. Lundberg and S.-I. Lee. (2019) shap: A unified approach to explain the output of any machine learning model. [Online]. Available: https://github.com/slundberg/shap

[24] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, pp. 37–37, 1996.

[25] H. Cheng, X. Yan, J. Han, and S. Y. Philip, "Direct discriminative pattern mining for effective classification," in *International Conference on Data Engineering*, 2008, pp. 169–178.

[26] G. Dong and J. Li, "Efficient mining of emerging patterns: Discovering trends and differences," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 43–52.

[27] F. Herrera, C. J. Carmona, P. González, and M. J. Del Jesus, "An overview on subgroup discovery: foundations and applications," *Knowledge and information systems*, vol. 29, no. 3, pp. 495–525, 2011.

[28] M. Atzmueller and F. Lemmerich, "Fast subgroup discovery for continuous target concepts," in *International Symposium on Methodologies for Intelligent Systems*, 2009, pp. 35–44.

[29] F. Lemmerich, "Novel techniques for efficient and effective subgroup discovery," 2014.

[30] D. Leman, A. Feelders, and A. Knobbe, "Exceptional model mining," in *Joint European conference on machine learning and knowledge discovery in databases*, 2008, pp. 1–16.

[31] W. Klösgen, "Explora: A multipattern and multistrategy discovery assistant," in *Advances in knowledge discovery and data mining*, 1996, pp. 249–271.

[32] B. F. Pieters, A. Knobbe, and S. Dzeroski, "Subgroup discovery in ranked data, with an application to gene set enrichment," in *ECML PKDD*, vol. 10, 2010, pp. 1–18.

[33] P. Clark and T. Niblett, "The cn2 induction algorithm," *Machine learning*, vol. 3, no. 4, pp. 261–283, 1989.

[34] M. J. Zaki, "Scalable algorithms for association mining," *IEEE transactions on knowledge and data engineering*, vol. 12, no. 3, pp. 372–390, 2000.

[35] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *ACM sigmod record*, vol. 29, no. 2, 2000, pp. 1–12.

[36] S. Wrobel, "An algorithm for multi-relational discovery of subgroups," in *European Symposium on Principles of Data Mining and Knowledge Discovery*, 1997, pp. 78–87.

[37] F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi, "Efficient breadth-first mining of frequent pattern with monotone constraints," *Knowledge and Information Systems*, vol. 8, no. 2, pp. 131–153, 2005.

[38] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees chapman & hall," *New York*, 1984.

[39] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning. springer series in statistics," in ∴ Springer, 2001.

[40] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[41] S. Nowozin, "Improved information gain estimates for decision tree induction," *arXiv preprint arXiv:1206.4620*, 2012.

[42] K. P. Bennett, "Global tree optimization: A non-greedy decision tree algorithm," *Computing Science and Statistics*, pp. 156–156, 1994.

[43] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the em algorithm," *Neural computation*, vol. 6, no. 2, pp. 181–214, 1994.

[44] B. Letham, C. Rudin, T. H. McCormick, D. Madigan *et al.*, "Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model," *The Annals of Applied Statistics*, vol. 9, no. 3, pp. 1350–1371, 2015.

[45] D. Maturana, D. Mery, and A. Soto, "Face recognition with decision tree-based local binary patterns," in *Asian Conference on Computer Vision*. Springer, 2010, pp. 618–629.

[46] T. L. Pedersen and M. Benesty, "Understanding lime," https://cran.r-project.org/web/packages/lime/vignettes/Understanding_lime.html, accessed: 2019-10-08.

[47] F. Wang and C. Rudin, "Falling rule lists," in *Artificial Intelligence and Statistics*, 2015, pp. 1013–1022.

[48] T. B. Przemyslaw Biecek. (2019) Break down: Model agnostic explainers for individual predictions. [Online]. Available: https://github.com/pbiecek/breakDown

[49] B. Poulin, R. Eisner, D. Szafron, P. Lu, R. Greiner, D. S. Wishart, A. Fyshe, B. Pearcy, C. MacDonell, and J. Anvik, "Visual explanation of evidence with additive classifiers," in *National Conference on Artificial Intelligence*, vol. 21, no. 2, 2006, p. 1822.

[50] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowledge and information systems*, vol. 41, no. 3, pp. 647–665, 2014.

[51] L. Shapley, "A value for n-person games," *Contributions to the Theory of Games*, pp. 31–40, 1953.

[52] G. Chalkiadakis, E. Elkind, and M. Wooldridge, "Computational aspects of cooperative game theory," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 5, no. 6, pp. 1–168, 2011.

[53] P. Jäckel, *Monte Carlo methods in finance*. J. Wiley, 2002, vol. 71.

[54] M. Atzmueller, F. Puppe, and H.-P. Buscher, "Towards knowledge-intensive subgroup discovery." in *LWA*, 2004, pp. 111–117.

[55] S. Garcia, J. Luengo, J. A. Sáez, V. Lopez, and F. Herrera, "A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 4, pp. 734–750, 2012.

[56] M. Atzmueller, "Subgroup discovery," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 1, pp. 35–49, 2015.

[57] H. Grosskreutz, M. Boley, and M. Krause-Traudes, "Subgroup discovery for election analysis: a case study in descriptive data mining," in *International Conference on Discovery Science*, 2010, pp. 57–71.

[58] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.

[59] D. Dua and C. Graff, "UCI machine learning repository," 2017, accessed: 2019-11-15. [Online]. Available: http://archive.ics.uci.edu/ml

[60] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, p. 206, 2019.

[61] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[62] B. Ramsundar and R. B. Zadeh, *TensorFlow for deep learning: from linear regression to reinforcement learning.* " O'Reilly Media, Inc.", 2018.

[63] F. Chollet *et al.*, "Keras," https://keras.io, 2015, accessed: 2019-10-15.

[64] J. Ye, J.-H. Chow, J. Chen, and Z. Zheng, "Stochastic gradient boosted distributed decision trees," in *ACM conference on Information and knowledge management*, 2009, pp. 2061–2064.

[65] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in neurorobotics*, vol. 7, p. 21, 2013.

[66] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, 2017, pp. 3146–3154.

[67] G. Ke, Q. Meng, T. Finley, T. Wang, and W. Chen. (2019) Lightgbm: Light gradient boosting machine. https://github.com/microsoft/LightGBM.

[68] M. Honnibal and I. Montani, "spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing," *To appear*, vol. 7, 2017.

[69] M. Cord and P. Cunningham, "Machine learning techniques for multimedia," *2008*, pp. 251–262, 2007.

[70] W. Webber, A. Moffat, and J. Zobel, "A similarity measure for indefinite rankings," *ACM Transactions on Information Systems (TOIS)*, vol. 28, no. 4, p. 20, 2010.

[71] S. M. Lundberg, G. G. Erion, and S.-I. Lee, "Consistent individualized feature attribution for tree ensembles," *arXiv preprint arXiv:1802.03888*, 2018.

[72] J. Otterbacher, "Inferring gender of movie reviewers: exploiting writing style, content and metadata," in *ACM International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 369–378.

[73] D. Rao, D. Yarowsky, A. Shreevats, and M. Gupta, "Classifying latent user attributes in twitter," in *Search and mining user-generated contents*, 2010, pp. 37–44.

[74] F. Karimi, C. Wagner, F. Lemmerich, M. Jadidi, and M. Strohmaier, "Inferring gender from names on the web: A comparative evaluation of gender detection methods," in *International Conference Companion on World Wide Web*, 2016, pp. 53–54.

Appendices

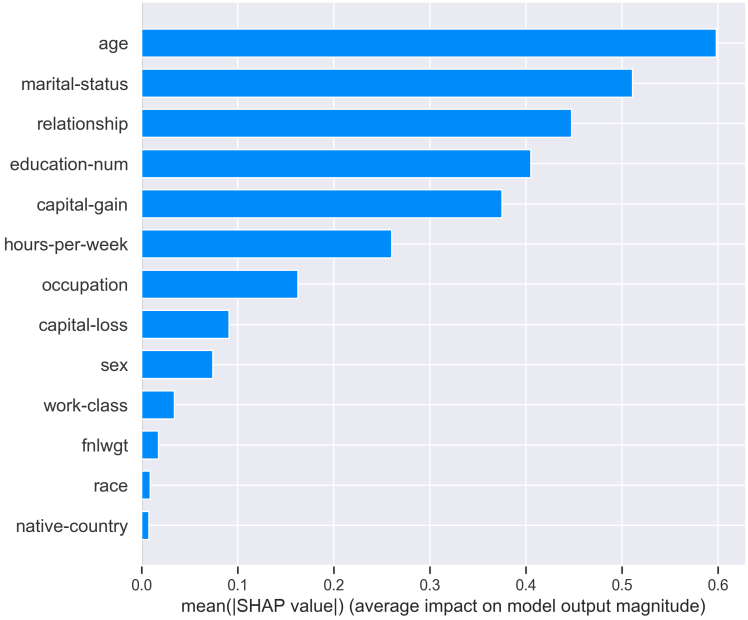

Figure 13: SHAP feature importance ranking plot. The importance score is measured by the SHAP feature importance method. It indicates the average feature impact on the model output. It can be observed that attribute "age" is considered as the most important feature in this case.

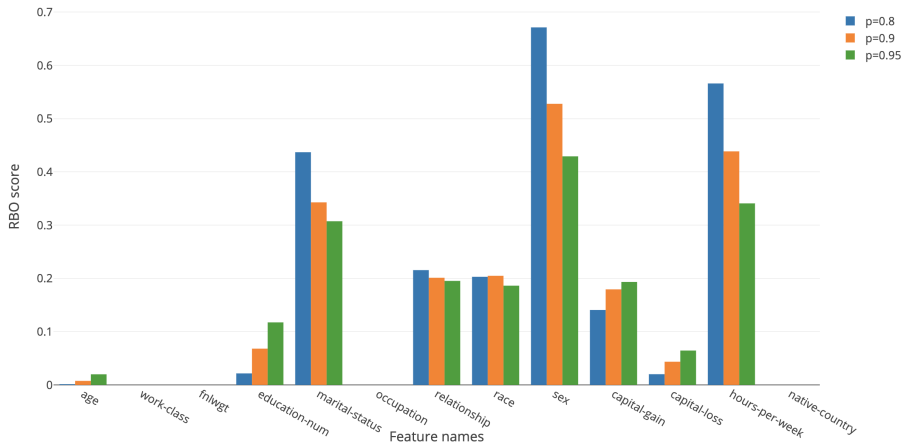

Figure 14: Rank-biased overlap between two list of ranked subgroups. The subgroups are discovered with respect to a local variable influence which is either measured by KernelSHAP or binary flip approach. For each feature, we obtain two lists of top-ranked subgroups, and the RBO similarity score is calculated. Parameter p=0.8 indicates that the top-five ranks are assigned 86% of the weight; p=0.9 implies the top-ten with the same weight; p=0.95 involves the top-thirty subgroups. The discovered subgroups are similar for the influence measure of some specific feature, e.g. "sex", with RBO score about 0.6. But they can be completely different for some features, e.g. "race", with RBO score 0.