TU WIEN Informatics

Von lokalen zu globalen Erklärungen und Kommunikation der Entscheidungsunterschiede zweier Black Box Classifier

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Data Science UE 066 645

eingereicht von

Lena Jiricka
Matrikelnummer 01427014

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Rauber

Wien, 4. Dezember 2022

Lena Jiricka	Andreas Rauber

TU WIEN Informatics

From local to global explanations and communication of decision differences between two black box classifiers

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Data Science UE 066 645

by

Lena Jiricka
Registration Number 01427014

to the Faculty of Informatics
at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Rauber

Vienna, 4th December, 2022

Lena Jiricka | Andreas Rauber

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Lena Jiricka

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 4. Dezember 2022

Lena Jiricka

Acknowledgements

First and foremost, I would like to thank my supervisor Andreas Rauber for giving me the opportunity to research on this topic, for inspiration and the continuous input and suggested improvements.
Furthermore, I would also like to give special thanks to my family for their continuous support throughout my whole education. Last but not least, I want to thank my boyfriend Lukas for always listening to me, giving me essential advice and his motivation.

Kurzfassung

Die Verwendung von Neural Networks, Support Vector Machines und Gradient Boosting Models ist den letzten 10 Jahren rasant gestiegen. Vor allem in kritischen Bereichen wie der Medizin oder dem Kreditrisikomanagement ist die Nachvollziehbarkeit und Verständlichkeit der Entscheidungen solcher sogenannten Black Box Modelle ein entscheidender und wesentlicher Punkt.
Die meisten Modelle werden anhand historischer Daten trainiert. Da sich die Verteilung der gesammelten Daten allerdings über die Zeit ändern kann, muss das Modell erneut trainiert und damit aktualisiert werden. Der Vergleich der unterschiedlichen Versionen solcher Modelle kann beim Erkennen von solchen Concept Drifts hilfreich sein.
DiRo2C (Difference Recognition of 2 Classifiers) zielt darauf ab, Entscheidungsunterschiede zweier Black Box Modelle lokal, also für einen bestimmten Datenpunkt, zu erkennen und zu lernen. Mithilfe eines genetischen Algorithmus wird ein synthetischer Datensatz generiert, der aus ähnlichen Datenpunkten und den entsprechenden Entscheidungen der Black Box Modelle besteht. Auf diesem synthetischen Datensatz wird im nächsten Schritt ein Decision Tree trainiert, um die Entscheidungsunterschiede zu lernen und in weiterer Folge zu erklären. Das Ziel dieser Diplomarbeit ist die global erklärbare künstliche Intelligenz, geschaffen durch mehrere lokale Erklärungen in Bezug auf Entscheidungsunterschiede zwischen zwei Black Box Klassifizierungsmodellen. Wir präsentieren Möglichkeiten, globale Erklärungen durch DiRo2Cs lokale Erklärungsansätze zu erzeugen. Darüberhinaus werden Strategien vorgestellt, um die generierten Erklärungen zu präsentieren und kommunizieren. Experimente haben gezeigt, dass ein 'Lokal zu Global' Ansatz mit Clustering Konzepten, um eine Beschreibung der Entscheidungsunterschiede zweier Black Box Modelle zu generieren, eine ähnliche Leistung wie ein auf dem ursprünglichen Datensatz trainierter Decision Tree erzielt und gleichzeitig die Komplexität des Erklärers reduziert.

Abstract

The use of neural networks, support vector machines and gradient boosting models, among others, has increased significantly in the past 10 years. Especially in crucial areas, the understanding and traceability of the decisions of such black box models is of fundamental interest.
Most models are trained on the basis of historical data at a specific point in time. However, the distribution of measured data may change over time and hence the model has to be updated. Comparing different versions of models to detect decision differences supports detecting such concept drifts.
DiRo2C (Difference Recognition of 2 Classifiers) aims at recognizing decision differences locally, that is for a specific instance, between two black box classifiers using a modified genetic algorithm to create a synthetic dataset, consisting of data points similar to the to be explained instance and the corresponding decisions of the two black box classifiers. On this synthetic dataset a decision tree is trained to learn and explain decision differences.
The main focus of this thesis is the problem of global explainable artificial intelligence through local explainers in the setting of difference recognition of two black box classifiers. We propose approaches to derive global explanations using concepts of local explanation generation of DiRo2C in addition to accompanying strategies to communicate those explanations. Experiments show that a 'Local to Global' approach using clustering concepts to derive a description of the characteristics of the decision differences of the black box models, yields similar performance to a decision tree trained on the original dataset while reducing the complexity of the explainer.

Contents

CHAPTER 1

Introduction

1.1 Motivation and Problem Statement

Machine Learning has made its way into our daily lives, providing automated decisions having a great impact on the future. Especially black box models, such as Neural Networks, Support Vector Machines and gradient boosting models, have shown great performance in many complex tasks. However, decisions made by such black box models lack of traceability and understanding [10, 5, 16]. Many model-agnostic methods, that is independently of the model or its internals, have been proposed to explain the decisions of single black box models either globally, to describe the average behavior of the model, or locally for a single instance [27, 7, 28, 22].

The learning of models is based on historical data at a specific point in time and outcomes provided by domain experts. However, the distribution of measured data may change over time and hence the model has to be updated [37]. Comparing different versions of models to detect decision differences support detecting such concept drifts. In another scenario, one might be interested in comparing two black box models trained on the same underlying data or comparing two different Machine Learning models.
DiRo2C (Difference Recognition of 2 Classifiers) [31] aims at recognizing decision differences locally between two black box classifiers. This is achieved by adaption of the LORE (LOcal Rule-based Explanations) approach by Guidotti et al. [7], a model-agnostic method to provide local interpretable explanations. DiRo2C uses a modified genetic algorithm to create a synthetic dataset, consisting of similar data points to the to be explained instance and the corresponding decisions of the two black box classifiers. On this synthetic dataset, a decision tree is trained to learn decision differences. This thesis will focus on two main enhancements of DiRo2C which will be introduced and discussed in the following.
The first part of the thesis deals with the problem of global explainable artificial intelligence through local explainers in the setting of difference recognition of two black box

classifiers.
One approach to capture global patterns of decision differences is to make use of local explanation generation strategies. By combining several synthetic datasets generated for specific instances, for which decision differences are to be explained, a global synthetic dataset can be obtained. On the basis of this global synthetic dataset, a (global) decision tree can be trained [31]. However, it is not clear which and how many instances to select. Too few instances might not result in sufficient global explanations in terms of difference detection accuracy. On the contrary, too many instances increase the complexity since the synthetic dataset has to be generated for each of them.
Research question 1: *To what extent can the performance of global explanations in terms of difference detection accuracy be improved by advanced data synthesis approaches?*

In case of complex or highly non-linear decision boundaries or simply many disjoint areas of decision differences, a single local explainer might become too complex to comprehend areas of decision differences. Another approach to describe the overall logic of the models is to use multiple local explanations in combination with clustering strategies.
A Self-organizing map is used to understand how different clusters of input data are distributed through input space or how an outcome variable changes across this dataset. This concept applied to data in the setting of difference detection recognition yields a structured dataset consisting of clusters with similar instances. For each cluster, a local explanation can be obtained. Furthermore, for each local explainer, the decision level can be projected onto two dimensions and hence visualised with the help of SOMs.
Research question 2: *To what extent can the explanation obtained from a structured combination of local explanations compete with or outperform a single global explanation in terms of difference detection accuracy and complexity?*

The third research questions focuses on the communication of explanations. In the following, we consider two k-class black box classifiers. We obtain k^2 combinations of class predictions of the two black box classifiers as response for the (global) difference detection classifier and hence increased complexity of the explaining tree.
Research question 3: *What is an effective way of communicating decision differences of two multi-class black box classifiers?*

1.2 Structure of the Work

Chapter 2 presents a brief introduction to explainability of single black box models, introduces core principles of DiRo2C and works as a theoretical foundation for the thesis. Additionally, an overview of concepts used in this thesis is provided.
The remainder of the thesis is divided into two main parts:

From Local to Global Section 3.1 presents concepts used in this section. In Section 3.2, approaches to obtain a global explanation from multiple local explanations are introduced. The results are presented and discussed in Section 3.4.

Communication of decision differences in multi-class setting Chapter 4 covers research question 3 by describing the process on how to analyze possible decision differences including approaches to present high-dimensional classification results and argumentative evaluation.

CHAPTER 2

Explainability of Black Box Models and DiRo2C

2.1 Terminology and Definitions

Biran and Cotton [2] define systems as interpretable 'if their operations can be understood by a human, either through introspection or through a produced explanation'. Based on this definition, Miller [20] defines interpretability of a model as 'the degree to which an observer can understand the cause of a decision'. Similarly, Guidotti et al. [8] define interpretability of a black box 'as the ability to explain or to provide the meaning in understandable terms to a human'. Moreover, whether a model is interpretable is dependent on the user seeking understanding of the model's internals or decisions [27].

2.1.1 Dimensions and Taxonomy of Interpretability

The domain of interpretable machine learning can be subdivided based on a variety of properties: a model may be interpretable by design (*transparent box design problem*), such as simple decision trees, rules, or linear models, or explanations can be provided after model training (post-hoc, *black box explanation problem*) [8, 1, 4].
Post-hoc methods can further be subdivided into three main categories of problems: model explanation, outcome explanation, and model inspection. The aim of model explanations is to provide an overall understanding of the logic of the black box model [8] to prevent incorrect decisions by a black box caused by biases in the training data or introduced by the model [24]. On the contrary, the outcome explanation problem focuses on a specific data point and the corresponding black box decision. The model inspection problem aims at understanding a specific property of the model, such as the effect of changes to an attribute [8].
Post-hoc methods are often independent from the model [4] which is another way of distinction of explanation systems: model-agnostic vs. model-specific. Model-agnostic

approaches can be applied to any black box whereas model-specific approaches are tied to specific model classes [4, 8]. Another important distinction is based on the scope of interpretability. A global explanation describes the average behavior of the model and focuses on how the overall logic of the black box works [8]. On the contrary, a local interpretability method provides explanations for a single instance [1, 8].

2.1.2 Definitions

A classifier is defined as a function $M : \mathcal{X}^m \to \mathcal{Y}$ that maps m-tuples x, referred to as instances, of the input space $\mathcal{X}^m$ to decisions y in the target space $\mathcal{Y}$ [7]. A decision y by the classifier M for an instance $x \in \mathcal{X}^m$ is denoted as $M(x) = y$. Throughout this work, M is considered to be a black box model, for example Neural Networks, SVMs and ensemble classifiers [8].
An explanation is e is a decision rule $r = P \to y$, describing the reasons for the decision y using a set of premises in conjunctive form $P = \{p_1, \ldots, p_s\}$. A set of explanations is denoted as explanation theory $E = \{e_1, \ldots, e_q\}$ [29, 7].

2.1.3 Decision Differences

In the following, we consider two k-class black box classifiers, M_A and M_B, trained on datasets A and B, respectively. Hence, there are $k \cdot k$ possible combinations of class predictions as visualized in Figure 2.1 for $k = 2$ and in Figure 2.2 for $k = 3$.

2.1.4 Running Examples

All concepts throughout this thesis are visualized using two-dimensional synthetically generated datasets[1] as shown in Figure 2.1 and Figure 2.2.
For the first running example (Sine), the decision boundaries f_1 and f_2 of the two black boxes M_A and M_B are emulated by two sine functions defined as follows:

$$\begin{aligned} f_1(x) &:= 4 \cdot \sin(x) \\ f_2(x) &:= \frac{\sin(x)}{x} \ \forall x \in \mathbb{R}\backslash\{0\} \text{ and } f_2(0) := 1 \end{aligned} \tag{2.1}$$

Based on these functions the classifier for an instance $x = (x_1, x_2) \in \mathbb{R}^2$ are defined:

$$\begin{aligned} M_A(x) &= \begin{cases} 1, & x_2 > f_1(x_1) \\ 0, & x_2 \leq f_1(x_1) \end{cases} \\ M_B(x) &= \begin{cases} 1, & x_2 > f_2(x_1) \\ 0, & x_2 \leq f_2(x_1) \end{cases} \end{aligned} \tag{2.2}$$

where $x = (x_1, x_2) \in \mathbb{R}^2$ denotes an instance to be classified. 1000 data points were sampled from two bivariate normal distributions with centers $\mu_1 = (-5, 0)$ and $\mu_2 = (5, 0)$

[1] Generating code and data available at `https://github.com/jrckln/DiRo2CLocaltoGlobal`

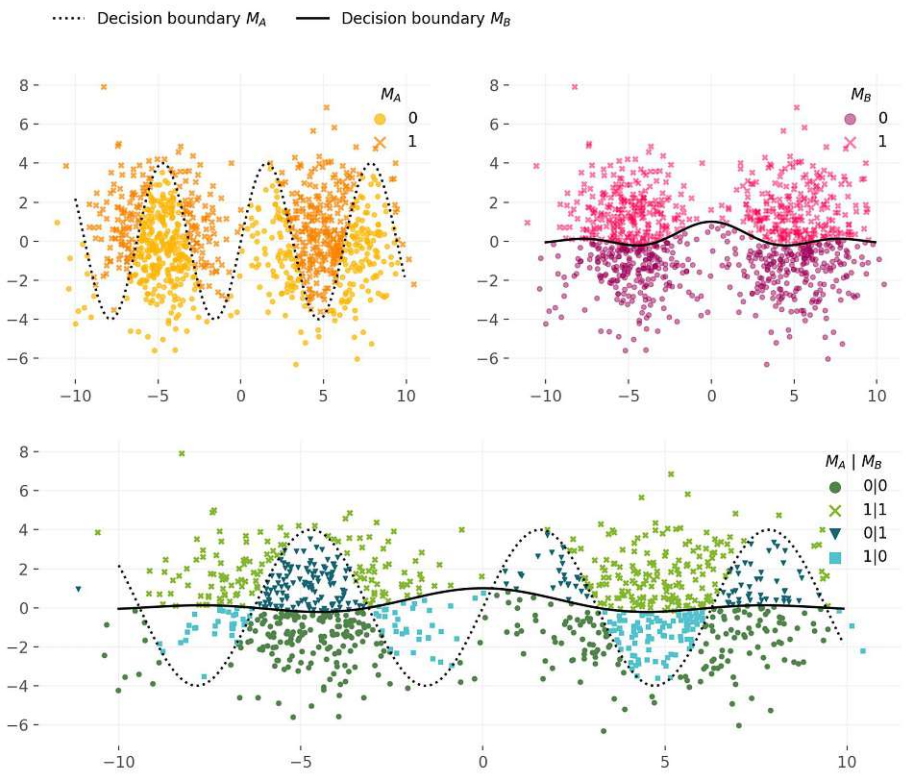


Figure 2.1: Decision differences: Sine running example

and covariance-matrices $\Sigma_i = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}, i = 1, 2.$

The second running example (Spiral) consisting of 1000 instances was generated with 3 classes. Support Vector Machine models with radial basis function kernel and linear kernel are used as black box models M_A and M_B, respectively.

2.2 DiRo2C

DiRo2C [31] aims at recognizing decision differences locally between two black box classifiers. This is achieved by adaption of the LORE approach [7]. LORE is a model-agnostic method to provide local interpretable explanations for a single black box. Given a black box M and an instance x for which decisions are to be explained, a local explanation is obtained from a decision tree trained on a generated dataset of similar instances. The strength of LORE lies in the generation of similar instances by making use of the concepts of genetic algorithms to focus on the decision boundaries in the vicinity of the to be

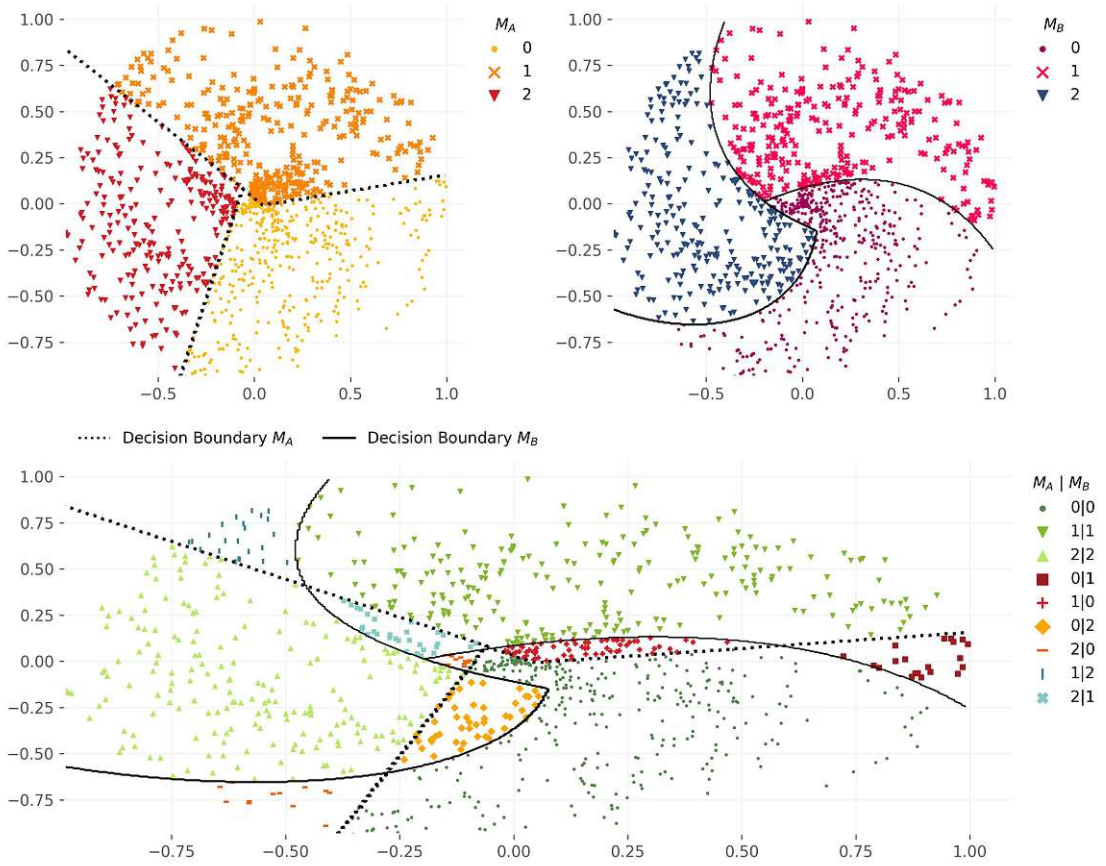
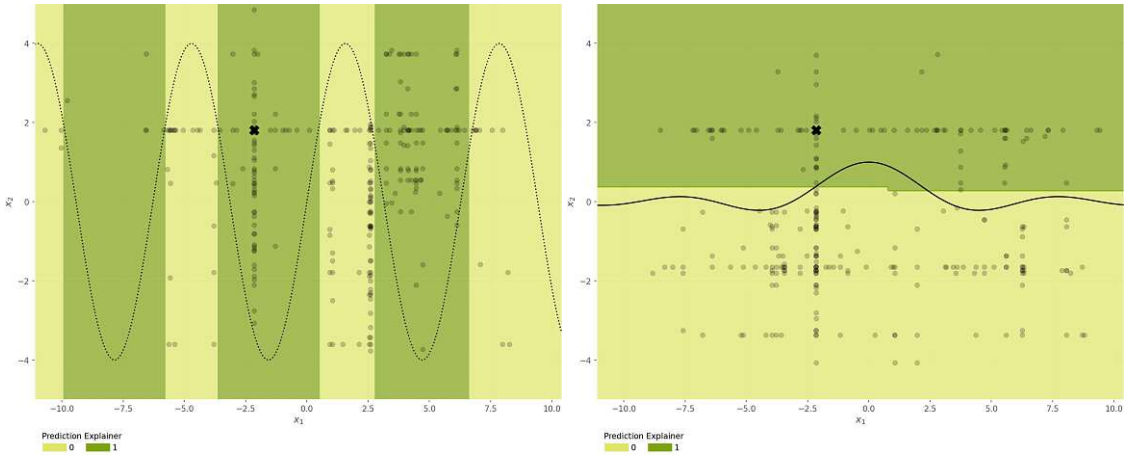


Figure 2.2: Decision differences: Spiral running example

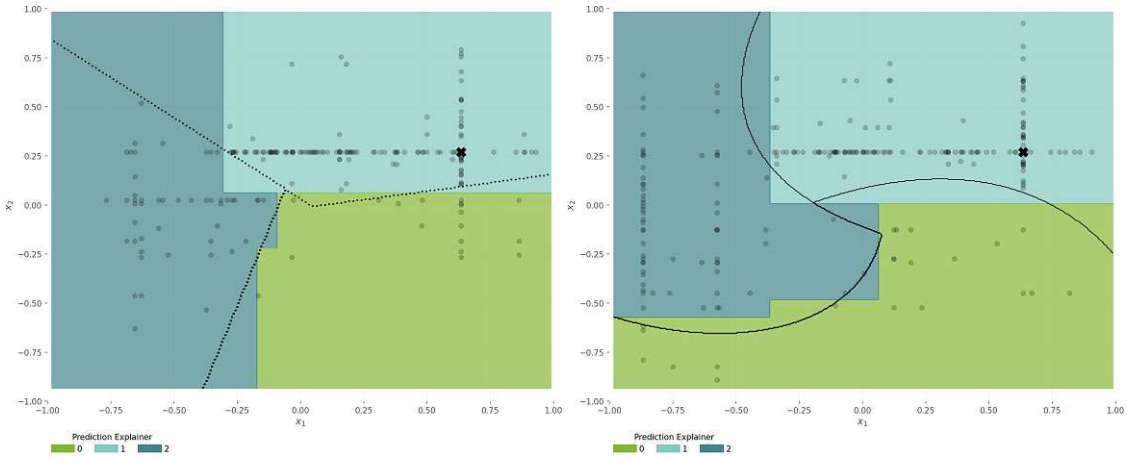
explained instance x [7]. The decision surface of the explainer of each black box model for both running examples are shown in Figure 2.3 and Figure 2.4. In the proximity of the marked instance to be explained, the decision boundaries of the black boxes are well approximated by the explainer which can be seen especially for the sine running example.

DiRo2C extends this concept to decision differences between two black box classifiers. In order to explain differences in decisions of the k-class black boxes, DiRo2C maps these to a two-class (decision differences vs. no decision differences) or a $k \cdot k$-class problem. The concepts of LORE are then used to explain the reasons for the assignment into these classes: a decision tree is trained to solve the two-class or a $k \cdot k$-class problem as interpretable surrogate model. DiRo2C uses a modified genetic algorithm to create a synthetic dataset, consisting of similar data points to the instance for which decision differences are to be explained and the corresponding decisions of the two black box classifiers as basis for the decision tree.



(a) Model A (b) Model B

Figure 2.3: Sine running example: decision surface and generated training data for the explainer by LORE



(a) Model A (b) Model B

Figure 2.4: Spiral running example: decision surface and generated training data for the explainer by LORE

2.2.1 Neighborhood Generation

The goal of the neighborhood generation process is to create a set of instances with similar properties as the instance for which decision differences are to be explained in order to approximate local decision differences [7]. In the following, the core components of genetic algorithms adapted to DiRo2C are presented and discussed.

Fitness function The modified genetic neighborhood of DiRo2C consists of a set of instances $Z_=$ with concordant decisions of M_A and M_B as well as a set of instances $Z_\neq$ with different decisions of M_A and M_B to generate a class-balanced difference detection dataset. This is achieved by a separate modified fitness function for each set of instances that score these instances. The fitness functions of a data point $z \in \mathcal{X}^m$ in the process of explaining decision differences for an instance $x \in \mathcal{X}^m$ are defined as given in Equations (2.3, 2.4) [31] where $d(v, w)$ denotes the distance between two data points $v, w \in \mathcal{X}^m$ using a distance function $d : \mathcal{X}^m \times \mathcal{X}^m \to [0, 1]$ and I denotes the Indicator function.

$$\text{fitness}^x_=(z) = \text{I}_{M_A(z)=M_B(z)} + 1 - d_{(x,z)} - I_{x=z} \tag{2.3}$$

$$\text{fitness}^x_\neq(z) = \text{I}_{M_A(z)\neq M_B(z)} + 1 - d_{(x,z)} - I_{x=z} \tag{2.4}$$

Equation (2.3) favors instances z similar, but not equal to x, for which both black box models M_A and M_B predict the same class. On the contrary, equation (2.4) presents a fitness function which scores instances z similar, but not equal to x, high for which decisions of the black box models are different [7, 31]. For continuous features, normalized Euclidean distance and for categorical features, Simple Match distance is used. The distance between two data points is then defined as the weighted sum of both distances.

Selection The selection operator selects a subset of the current population. Individual instances are chosen probabilistically by assigning a probability proportional to their fitness, measured by the respective fitness function. Therefore, instances with higher fitness have a higher probability to be selected [3, p. 65]. LORE as well as DiRo2C use tournament selection[2] where l times the best instance is selected from a set of k randomly chosen instances from the parent population [3, p. 181].

Mutation and Crossover Subsequent to the selection phase, the genetic operators mutation and crossover are applied to the selected instances in order to produce offsprings [3, p. 65]. LORE and DiRo2C use a two-point crossover operator that randomly chooses two features and swaps the feature values of the parents. Crossover is applied to a fraction of the new population, controlled by a probability pc. A proportion pm of the current population is then mutated by randomly replacing feature values according to the empirical distribution of the feature determined using the training data [7, 31].

The steps to obtain the synthetic dataset are presented in Algorithm 1.

[2] `https://deap.readthedocs.io/en/master/api/tools.html#deap.tools.selTournament`

Algorithm 1 Modified Genetic Neighborhood Generation of DiRo2C

Input

x	Instance for which decision differences are to be explained
f	Fitness function
M_A	Black-box model A
M_B	Black-box model B
n	Population size
ngen	Number of generations
pc	Fraction of the population for which crossover is applied
pm	Fraction of the population for which mutation is applied

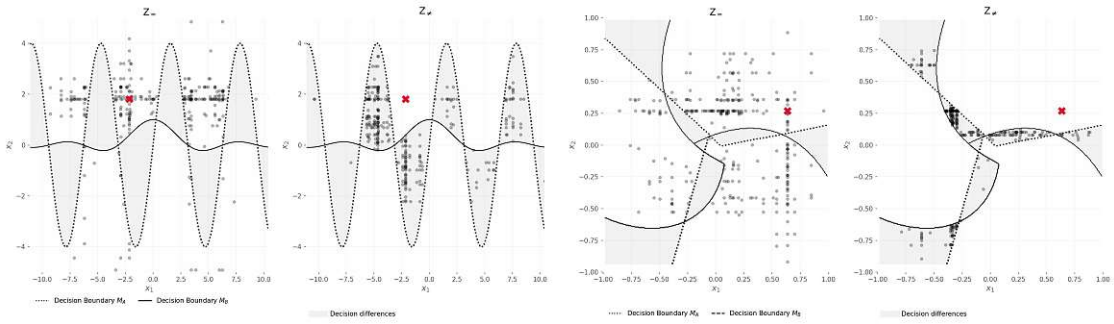
Output

$Z \in \mathcal{X}^m$ Neighborhood

$P_0[k] \leftarrow x,\ k = 1 \dots n$
evaluate(P_0, f, M_A, M_B)
for $i = 1$ to n **do**
  $P_{i+1} \leftarrow$ select(P_i)
  $P'_{i+1} \leftarrow$ crossover(P_{i+1}, pc)
  $P''_{i+1} \leftarrow$ mutate(P'_{i+1}, pm)
  evaluate(P''_{i+1}, f, M_A, M_B)
  $P_{i+1} \leftarrow P''_{i+1}$
end for
$Z \leftarrow P_{i+1}$
return Z

2.2.2 Learning decision differences

For an instance x, two black box models M_A and M_B and a pre-specified population size n, DiRo2C builds the modified genetic neighborhood as presented in Algorithm 1 using the equality fitness function given in Equation (2.3), favoring instances with matching predictions of M_A and M_B. This process is repeated for the second fitness function given in Equation (2.4) focusing on instances with differences in predictions of M_A and M_B. The generated neighborhoods are visualized separately for $Z_=$ and $Z_\neq$ in Figure 2.5a for Sine running example and Figure 2.5b for Spiral running example. For Sine running example, the closest non-linear decision boundary to the left of the marked instance is correctly emulated by synthetic instances of $Z_\neq$. This can also be observed for the spiral running example for the closest non-linear decision boundary of model B as well as the linear decision boundary of model A below the marked instance.
The genetic neighborhoods $Z_=$ and $Z_\neq$ are combined in a next step and duplicates are dropped which no longer guarantees a class-balanced dataset.
To generate the difference detection dataset, the classifier decision of the instances from the combined modified genetic neighborhoods are obtained both from M_A and M_B and compared. To generate the labels of the outcome two concepts are considered. First, the



(a) Sine running example (b) Spiral running example

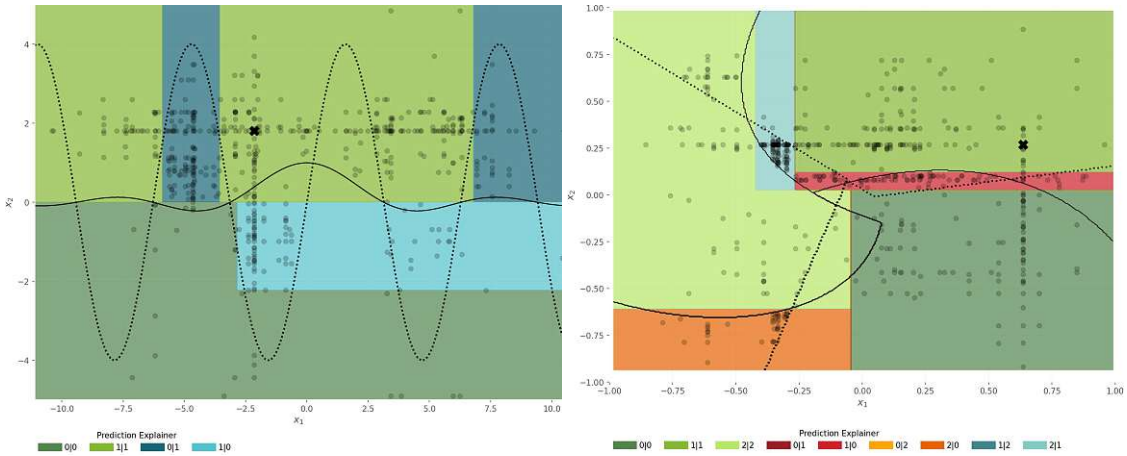
Figure 2.5: DiRo2C's generated neighborhood separately for each fitness function

two classes *difference* and *no difference* in predictions of M_A and M_B are used as labels. As a second approach, all $k \cdot k$ combinations of class predictions of the two classifiers are considered as labels for the response. Based on the generated difference detection dataset, a decision tree is trained to learn and explain decision differences [31]. Figure 2.6a and Figure 2.6b show the decision surfaces of the local explainers in addition to the synthetic neighborhood used for the explainer generated by DiRo2C. The explaining models are shown in Figure 2.7a and Figure 2.7b. The decision trees shown were pre-pruned to reduce complexity and enhance readability.

For the Sine running example, the local explainer is a good linear approximation of the boundaries of the decision difference regions in the proximity of the instance to be explained. According to the explainer for negative x_2, model A predicting class 1 and model B predicting class 0 are the only possible decision differences (first split, see Figure 2.7a). More precisely, this is the case for $x_2 > -2.226$ and $x_1 > -2.846$ as follows from the path *node* #0 - *node* #1 - *node* #3 - *node* #5. The explainer of the Spiral running example predicts decision differences using three rules:

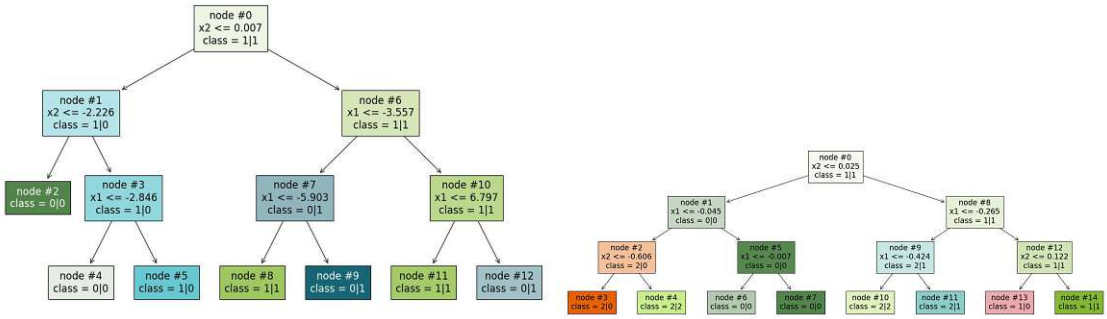
- $x_2 \leq 0.007 \wedge x_2 \leq -2.226 \wedge x_1 > -2.846$ (path: *node* #0 - *node* #1 - *node* #3 - *node* #5)
- $x_2 > 0.007 \wedge x_1 \leq -3.557 \wedge x_1 > -5.903$ (path: *node* #0 - *node* #6 - *node* #7 - *node* #9)
- $x_2 > 0.007 \wedge x_1 > -3.557 \wedge x_1 > 6.797$ (path: *node* #0 - *node* #6 - *node* #10 - *node* #12)

Figure 2.8a and Figure 2.8b visualizing the correctness of the explainers' prediction across the input-space, clearly show that DiRo2C is a **local** explanation generation framework since many areas of actual decision differences are missed or incorrectly predicted. In case one seeks a global explanation, the rules produced by DiRo2C are too inaccurate at further distance from the instance being explained.



(a) Sine running example (b) Spiral running example

Figure 2.6: Decision surface and training data of the local explainer for decision differences of the marked instance



(a) Sine running example (b) Spiral running example

Figure 2.7: Explainer for local decision differences in the proximity of the marked instance

Algorithm 2 Procedure of DiRo2C using a modified genetic neighborhood

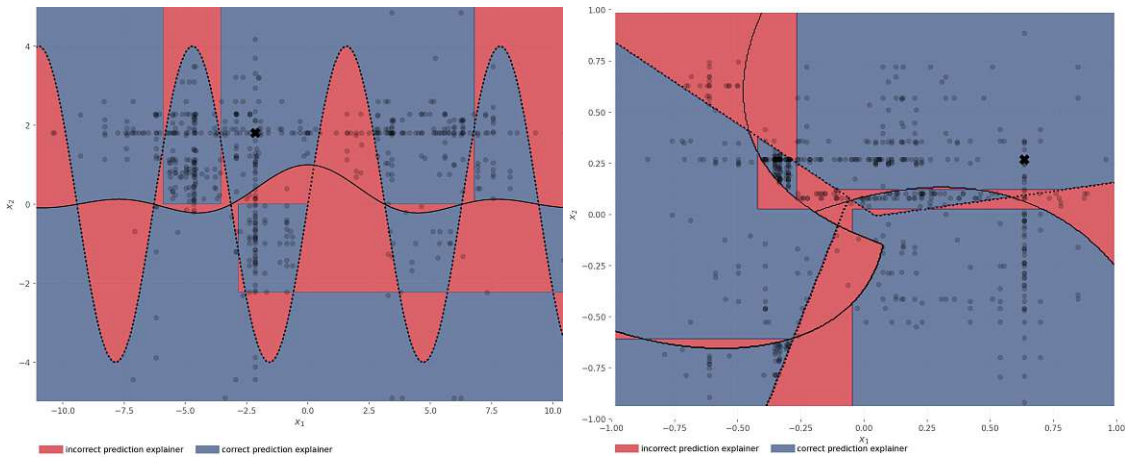
Input

x	Instance for which decision differences are to be explained
M_A	Black box model A
M_B	Black box model B
n	Population size

Output

dc	Difference Classifier
$Z_d \in \mathcal{X}^m$	Combined neighborhood of black box A and B and decision difference as target

$ngen \leftarrow 10$
$pc \leftarrow 0.5$
$pm \leftarrow 0.2$
$Z_{=} \leftarrow$ Genetic Neighborhood using (x, fitness$^x_{=}$, M_A, M_B, $\frac{n}{2}$, ngen, pc, pm)
$Z_{\neq} \leftarrow$ Genetic Neighborhood using (x, fitness$^x_{\neq}$, M_A, M_B, $\frac{n}{2}$, ngen, pc, pm)
$Z \leftarrow Z_{=} \cup Z_{\neq}$
$Z_d \leftarrow$ Build difference detection dataset using (M_A, M_B, Z)
$dc \leftarrow$ Train decision tree using Z_d
return dc, Z_d



(a) Sine running example (b) Spiral running example

Figure 2.8: Correctness of the explainers' prediction

CHAPTER 3

From Local to Global

The following part deals with the problem of global explainable artificial intelligence through local explanation approaches in the setting of difference recognition of two black box classifiers.
In Section 3.1, a review of existing literature on deriving global explanations from local approaches are presented, followed by the introduction of main concepts used in this thesis. Thereafter, bottom-up approaches to obtain global explanations in the setting of decision difference detection between two black box classifiers are proposed and evaluated by means of the running examples in Section 3.2. The proposed approaches are evaluated by means of benchmark datasets frequently used in literature. Experiments are described in Section 3.3 and results are presented in Section 3.4.

3.1 Background and Related work

In this section, results from a literature review of local-to-global techniques in the domain of classification explainability are presented and core concepts used in the first part *From Local to Global* of this thesis are formally introduced.

3.1.1 Local-to-Global Framework

Pedreschi et al. [24] proposed the local-to-global framework for black box explanations based on the following three assumptions:

Logic explanations Explanations should be based on logic

Local explanations The decision boundary of the black box model in the proximity of the instance to be explained can be approximated by a simple and explainable model

Explanation composition Similar instances have similar explanations which can be generalized

According to this framework, at first a local step is conducted that provides an explanation (and ideally also a counterfactual) of why the black box assigned a specific class for any instance of the training dataset. The subsequent local-to-global step consists of an iterative composition and generalization of all available local explanations for each instance of the training dataset, optimizing simplicity and fidelity [24].
GLocalX, based on the local-to-global framework, uses a set of logical rules representing multiple local explanations to infer global explanations. Local explanations are aggregated by hierarchically merging them into a global explanation, accounting for fidelity and accuracy [29].

3.1.2 Self-Organizing Maps

For RQ 2, we aim for multiple local explanations that apply for different parts of the feature space. Apart from the data structuring task, also for various visualizations of high-dimensional data Self-organizing maps are used due to their topology-preserving projections. While Principle Component Analysis provides a simple approach to this task, SOMs allow for a more sophisticated and non-linear projection.
Self-organizing maps, introduced by Kohonen [13], are Neural Networks used mainly for clustering and dimensionality reduction. The basis of a SOM is a pre-defined low-dimensional lattice of nodes arranged as a rectangle or hexagon. The input space is projected onto the lattice in a topology preserving fashion that can be utilized to visualize the high-dimensional input data [35, 9].
SOMs consist of an input and an output layer but no hidden layers. The output layer is organized as a low- dimensional (usually one- or two-dimensional) lattice in rectangular or hexagonal formation. SOMs are completely connected, that is every node in the input layer is connected to every node in the output layer. Each connection is associated with a weight [14].
The formation of SOMs is based on three processes [9, 11]:

1. **Competition:** For each input sample, the neurons in the output layer compete with each other for the best representation measured by a discriminant function. The winner of the competition is the neuron with the largest value of the discriminant function.

2. **Cooperation:** The winning neuron forms the center of the topological neighborhood of cooperating neurons.

3. **Adaptation:** The neurons in the neighborhood learn by adapting their weights to achieve a higher value of the discriminant function.

Competition Let $x \in \mathcal{X}^{(m)}$ denote a random instance of the input data and $w_j \in \mathbb{R}^m$ the weight of the output node l_j. The weights $w_j, j = 1, \ldots\ldots, k$ with k denoting the

number of nodes in the output layer, are initialized either randomly or systematically. The number of nodes strongly influences generalization capabilities and performance of SOMs [35]. Kohonen [13] suggests to use $5 \cdot \sqrt{n}$ nodes. The node or neuron $l_{\iota(x)}$ that best matches the input instance x is defined as the node that minimizes the euclidean distance between x and w_j:

$$\iota(x) = \arg\min_j ||x - w_j||, \ j = 1, \ldots, k \tag{3.1}$$

The neuron that satisfies Equation (3.1) is called best-matching unit or winning neuron. It forms the center of the topological neighborhood of cooperating neurons [9, 12].

Cooperation Around the winning neuron l_ι, a topological neighborhood comprising cooperating neurons l_j at time t is defined [11]. The topological neighborhood function $h_{j,i}$ has to meet the following requirements [9]:

1. The topological neighborhood function should be symmetric around the winning neuron and the maximum should be attained by the winning neuron l_ι.
2. As the distance between two nodes l_j and l_i increases, the topological neighborhood function $h_{j,i}$ should decrease.

An example for a topological neighborhood function fulfilling these requirements is the Gaussian function [9, 11]:

$$h_{j,\iota}(t) = \exp\left(-\frac{d_{j,\iota}^2}{2\sigma(t)^2}\right), \ j = 1, \ldots, k \tag{3.2}$$

In Equation (3.2), $d_{j,\iota}$ denotes the lateral distance of the winning neuron l_ι and the excited neuron l_j defined by $d_{j,\iota} = ||l_j - l_\iota||$. The parameter σ is the effective width [9] or the radius of the topological neighborhood at time t [11]. An essential feature of SOMs is the dependence of the size of the neighborhood on time by utilization of exponential decay:

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\tau_1}\right), \ t = 0, 1, \ldots \tag{3.3}$$

where σ_0 is the value at initialization and τ_1 a time constant [9].

Adaptation The weights w_j of the neuron are updated according to the update rule given in Equation (3.4).

$$w_j(t+1) = w_j(t) + \alpha(t) h_{j,\iota}\left(x(t) - w_j(t)\right) \tag{3.4}$$

The learning rate $\alpha(t) \in [0, 1]$ controls the magnitude of change in weights and also decreases as a function of time t [11, 12].
The fitting process of SOMs with randomly initialized weights is visualized in Figure 3.1 for the Sine running example.

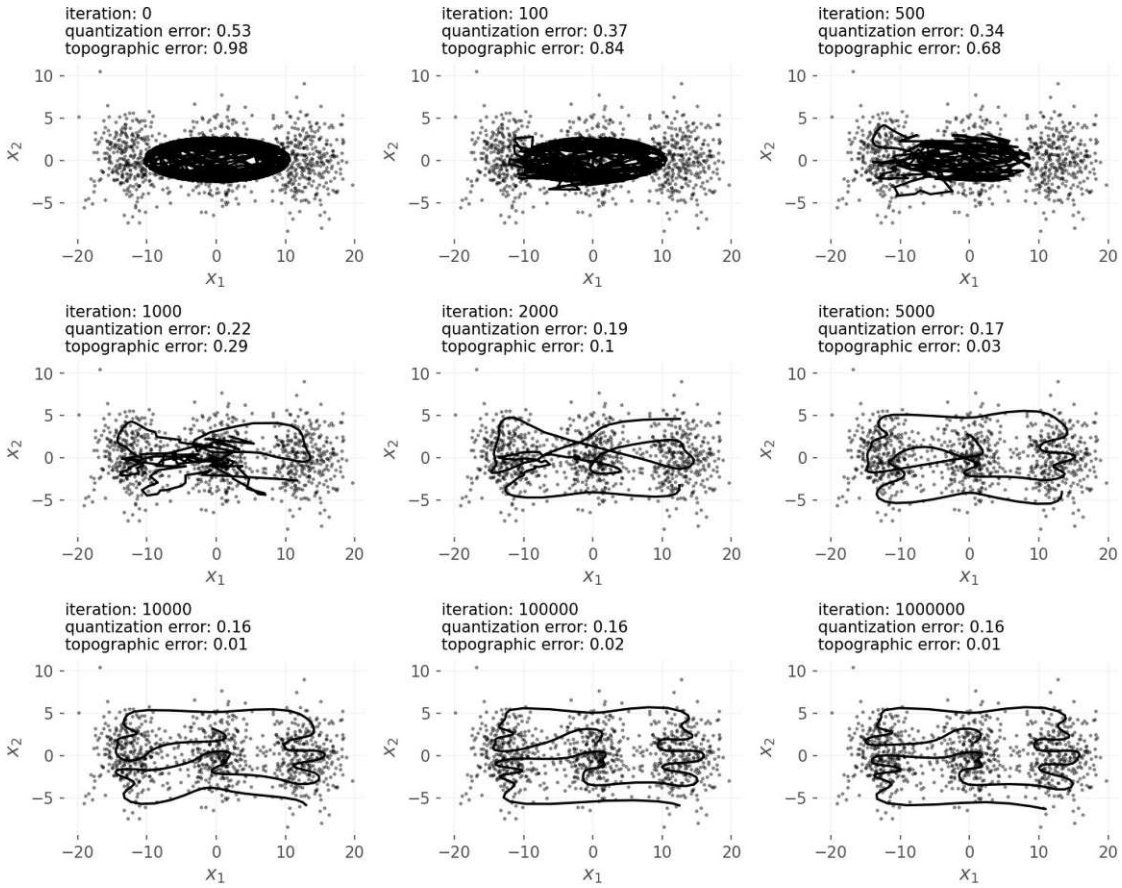


Figure 3.1: Training of SOM per iteration for the Sine running example

Quality Measures

Quantization Error The Quantization Error E_Q is defined as the average distance of the instances and the corresponding nearest nodes of the SOM [13] and measures how well the map fits the data [26]:

$$E_Q = \sum_{i=1}^{n} \|x_i - w_c\|^2 \text{ where } c = \arg\min_j \|x - w_j\|, j = 1, \ldots, k \tag{3.5}$$

Topographic Error The Topographic Error measures how well the map preserves the topology of the data [26]. It is defined as the proportion of instances where the BMU $l_{\iota(x)}$ and the second best matching unit $l_{\iota_2(x)}$ are adjacent nodes:

$$E_T = \frac{1}{n} \sum_{i=1}^{n} I_{l_{\iota(x)} \text{ and } l_{\iota_2(x)} \text{ are neighbors}} \tag{3.6}$$

Visualisation and Clustering

Numerous methods have been proposed to visualize SOMs mainly to reveal cluster boundaries. The most common method is called unified distance matrix (U-Matrix, [33]) and uses the distance of the neighboring weight vectors depicted in a grayscale image to evaluate the similarity of the neighbors [18]. This technique has been extended to a combination of distance and density information by U^*-Matrix technique [32]. Other visualizations rely on the distribution of the underlying data. Hit-histograms show the number of instances for each node with the node as BMU [18]. An overview of possible visualization techniques is given by Vesanto [34]. In the following visualizations each SOM node is colored according to the most frequent occurring label in the set of instances with the respective node as BMU.

3.2 Bottom-up Approaches for Global Explanations

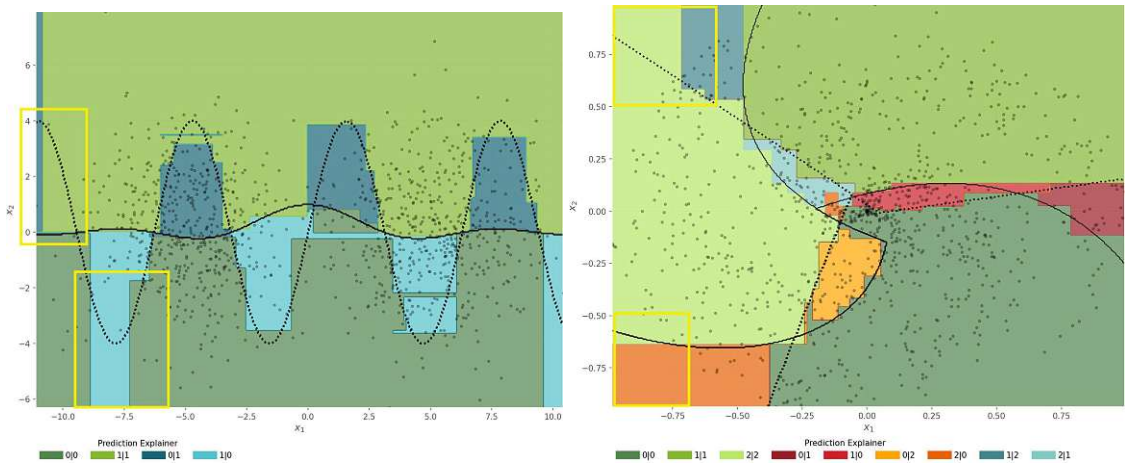
To capture global patterns of decision differences, multiple approaches are considered. In contrast to existing literature, the local-to-global step used in this thesis does not merge the rules of multiple local explanations but rather makes use of the data that was generated to extract rule-based explanations. All approaches described in the following (apart from the Baseline) use multiple independently with DiRo2C generated neighborhoods of pre-specified instances. The neighborhoods are either combined to a global dataset serving as training data for a global explanation model (Approach 1-3), or the data is structured into clusters and a local explanation is provided for each cluster (Approach 4). In this section, the following four bottom-up approaches for global explanations are introduced and compared to the Baseline approach by means of the running examples.

Approach 1: Random sampling
Approach 2: Class-stratified sampling
Approach 3: Cluster-stratified sampling
Approach 4: Structured sampling

This part of the thesis focuses solely on the performance of the approaches to detect decision differences. The decision differences of the constructed running examples are overrepresented in the dataset as compared to real world examples on the one hand and, on the other hand, the corresponding decision boundaries are highly non-linear and therefore complex to approximate by a decision tree. For this reason, to explain the decision differences present in the running examples, longer rules are needed to accurately describe the location of decision differences which might not be easily comprehensible anymore. Therefore, obtained explanations are not presented and discussed in this chapter. The resulting explainer and derived rules are the focus of Chapter 4.

Approach 0: Baseline To capture the whole available input-space, all available data is used to train an explaining model. Figure 3.2 shows the decision surfaces of the

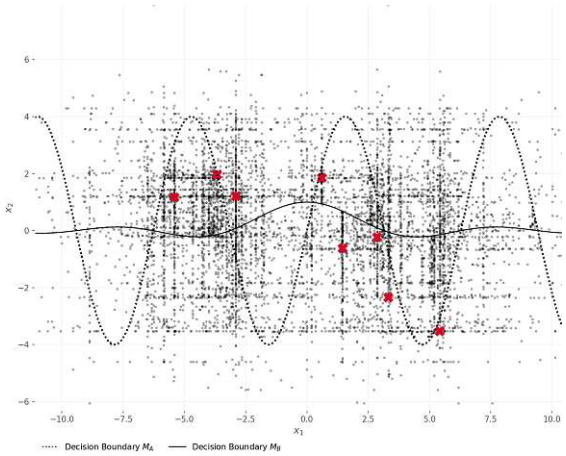
explainers of a pure global approach (Baseline) for the Sine and Spiral running example and the training data used for the decision tree.
Although the main areas of decision differences are overall correctly recognized by the global explainer, the decision tree fails to predict the correct decision difference class in areas with a low number of instances. Examples for such areas are marked in Figure 3.2. For the Spiral running example, this is especially the case at the outer area of the input-space.



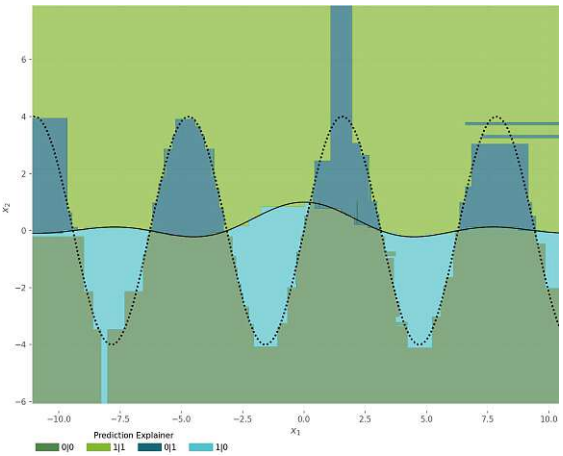
(a) Sine running example (b) Spiral running example

Figure 3.2: Baseline (Approach 0): Decision surfaces of explainers and corresponding training data of a pure global approach

Approach 1: Random sampling DiRo2C's strength lies in the neighborhood generation process that focuses on the boundaries between decisions. To make use of this property, a neighborhood might be generated for all available instances of the input data. However, since this process is time-consuming and would additionally result in redundant information, a straightforward way to reduce the number of instances is to randomly sample N instances from the underlying dataset. For each of these instances, a genetic neighborhood is generated using DiRo2C's neighborhood generation process. The neighborhoods are concatenated in a next step to a global synthetic dataset and decision differences are determined. This dataset is then used as a basis for a decision tree acting as a global explainer.
Figure 3.3 shows the generated synthetic neighborhoods of randomly sampled instances and the explainers' decision surface for both running examples. Compared to the Baseline explainers, random sampling of instances and DiRo2C's neighborhood generation already improves the detection of differences especially in the areas with a low number of training instances for the Baseline as can be seen for example for the Sine running example. For $x_1 \leq -10$ and $x_2 > 0$, the Baseline explainer missed this area of decision differences almost completely, whereas due to the neighborhood generation process, this part is

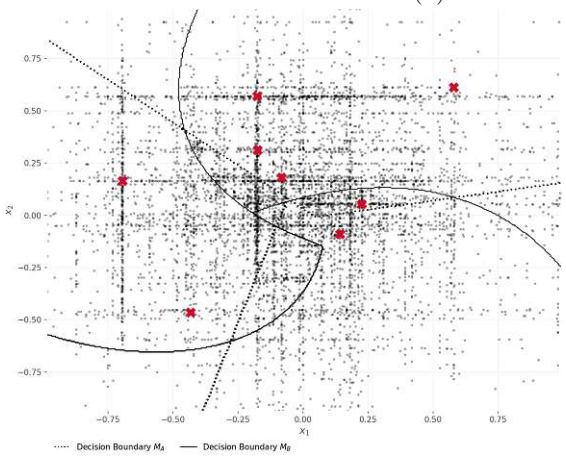


(a.a) Generated neighborhoods

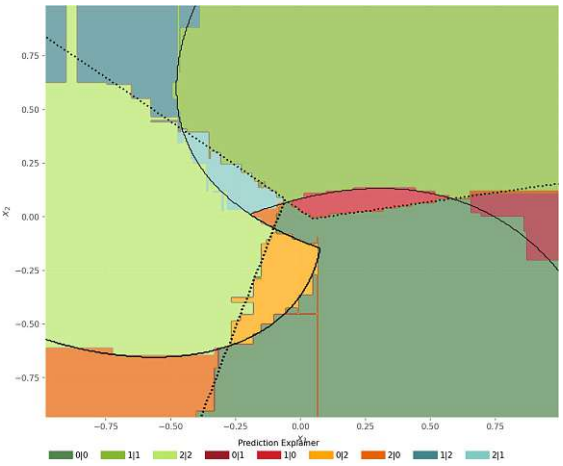


(a.b) Decision surface of explainer

(a) Sine running example



(b.a) Generated neighborhoods



(b.b) Decision surface of explainer

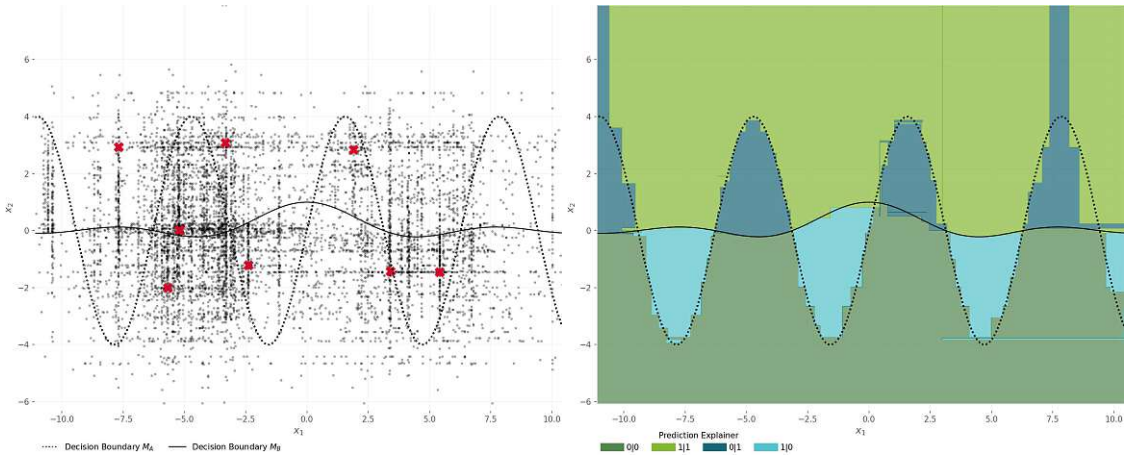
(b) Spiral running example

Figure 3.3: Approach 1: Random sampling

covered by synthetic data points.
For the Sine running example, three instances were sampled in the near proximity of each other resulting in unnecessarily overlapping neighborhoods and hence no information gain. In Figure 3.3b.a of the Spiral running example, it can be seen that the area for $x_1 > 0.6$ is not covered by any instance resulting in poor decision boundaries of the explaining decision tree as shown in Figure 3.3b.b. This is also the case for the Sine running example for negative x_1 and negative x_2.

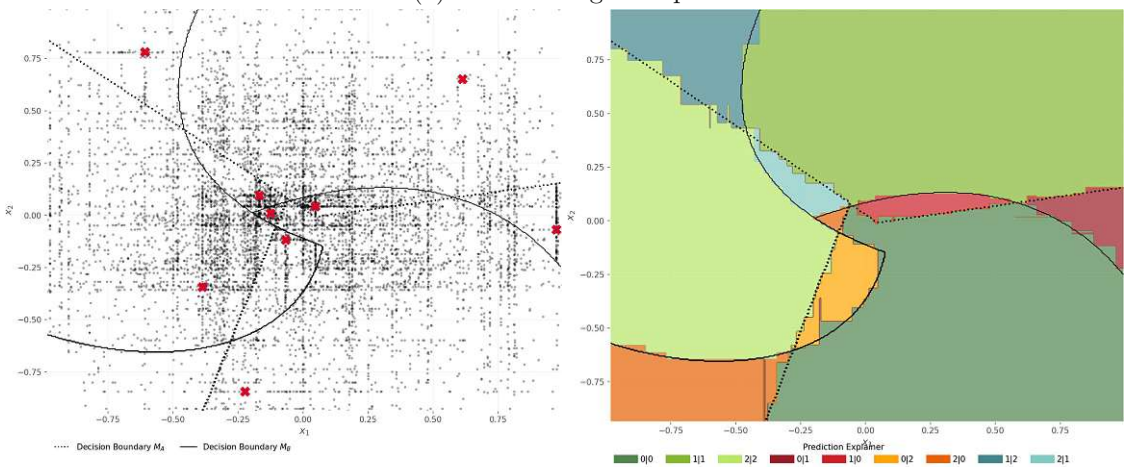
Approach 2: Class-stratified sampling Randomly sampled instances may be unevenly distributed across the input-space. To ensure that every decision difference label is tackled, each instance of the underlying dataset is classified using the black box models M_A and M_B and decision differences are determined. For each of the resulting $k \cdot k$ classes of possible decision differences $\max(1, \lfloor \frac{N}{k \cdot k} \rfloor)$ instances are randomly sampled and a neighborhood is generated via DiRo2C for each instance. The neighborhoods are again concatenated and a decision tree is trained to learn decision differences.
Stratified sampling of instances clearly improves in the issue of unevenly distribution of the random sampling approach. In Figure 3.4a.a showing the Sine running example, sampled instances are more evenly distributed across the input space, however disjoint subregions of areas of decision differences are still missed, for example the area marked in the Figure.
For the Spiral running example boundaries of $M_A|M_B : 1|0$ and $2|0$ are already sufficiently recognized by the explainer due to the sampled instances within this regions and the generated neighborhoods (Figure 3.4b.b). In contrast to random sampling, class stratified sampling ensures at least one sampled instance per decision difference class, hence also a sampled instance within the region of $M_A|M_B : 0|1$ and therefore more accurate and fine-granular approximations of the boundaries as compared to the random sampling approach where no instance was sampled within this region.

Approach 3: Cluster-stratified sampling In contrast to the Spiral running example, for the Sine running example the set of instances of certain combinations of black box predictions is not fully connected but partitioned into multiple regions. The main intuition of this approach is to select at least one instance for each region within the same decision difference label. In case of the Sine running example, the set of instances of the class $M_A|M_B : 1|0$ can be further subdivided into 4 disjoint regions which can be seen in Figure 3.5 showing a SOM-projection of the Sine running example. The higher the distance between two instances within the same decision difference class, the higher the probability that these instances belong to different regions.
For each possible combination of decisions of the black boxes, the corresponding instances are clustered using hierarchical-clustering with single-linkage criterion, and a dendrogram is constructed based on which the number of clusters is determined manually. From each resulting branch, one instance for which a synthetic neighborhood is generated, is randomly sampled. The neighborhoods are again concatenated and a decision tree is trained to learn decision differences.



(a.a) Generated neighborhoods (a.b) Decision surface of explainer

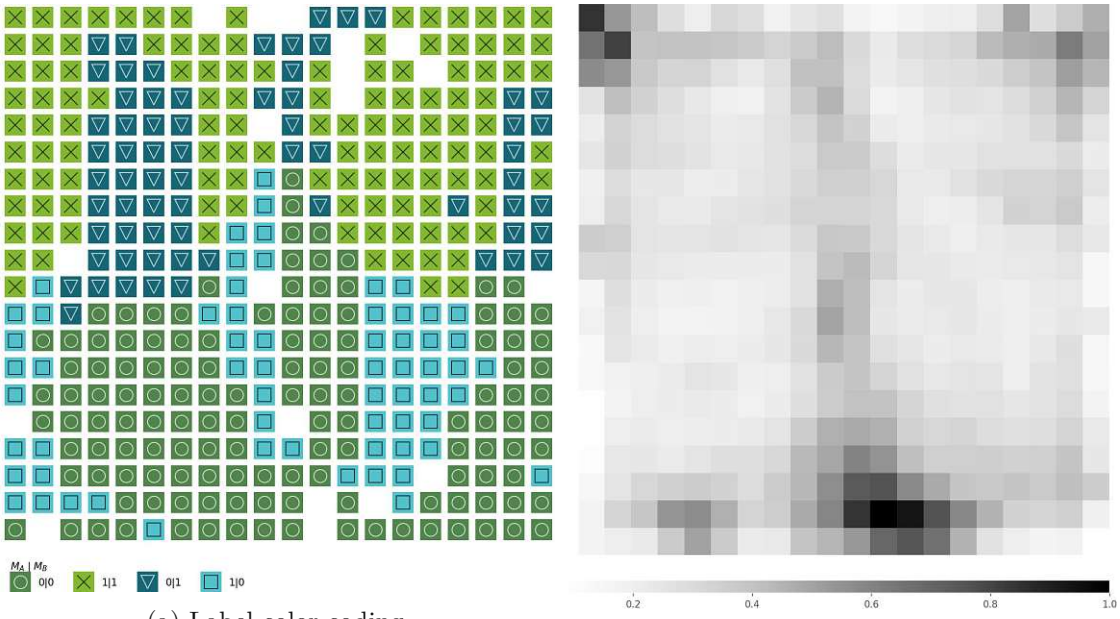
(a) Sine running example



(b.a) Generated neighborhoods (b.b) Decision surface of explainer

(b) Spiral running example

Figure 3.4: Approach 2: Class-stratified sampling



(a) Label color coding

(b) U-Matrix

Figure 3.5: SOM-Projection of the Sine running example

A visualization of disjoint partitions of a certain decision-differences class and a dendrogram of hierarchical clustering is shown in Figure 3.6. In Figure 3.5 and in the dendrogram of Figure 3.6, 4 partitions of region $M_A|M_B : 1|0$ are identifiable.
The dendrograms of hierarchical clustering of each decision difference class are shown in Figure 3.7 for both running examples and the manually chosen distance threshold is added. From each of the resulting clusters, an instance is randomly sampled and DiRo2C's neighborhood for this instance is generated. The neighborhoods are again concatenated and used as training data for a decision tree.
Figure 3.8 shows the sampled instances, generated neighborhoods and the decision surface of the trained explainer for both running examples. This approach to select instances does not have much influence on the Spiral running example, since only the combination $M_A|M_B : 2|0$ is split into two partitions. For the Sine running example, however, at least one instance was sampled from each partition. This sampling strategy results in an even better approximation of the boundaries of decision differences.

Approach 4: Structured sampling In case of complex or highly non-linear decision difference boundaries as emulated by the running examples or simply disjoint areas of decision differences, a single local explainer has to be very complicated to sufficiently locate areas of decision differences. This approach uses multiple ordered local explaining models to draw a global picture. First, the decision differences are structured using a one-dimensional SOM. Subsequently, the weights of the SOM are clustered, and for each of the clusters a separate local explanation is obtained. The explainers are trained on

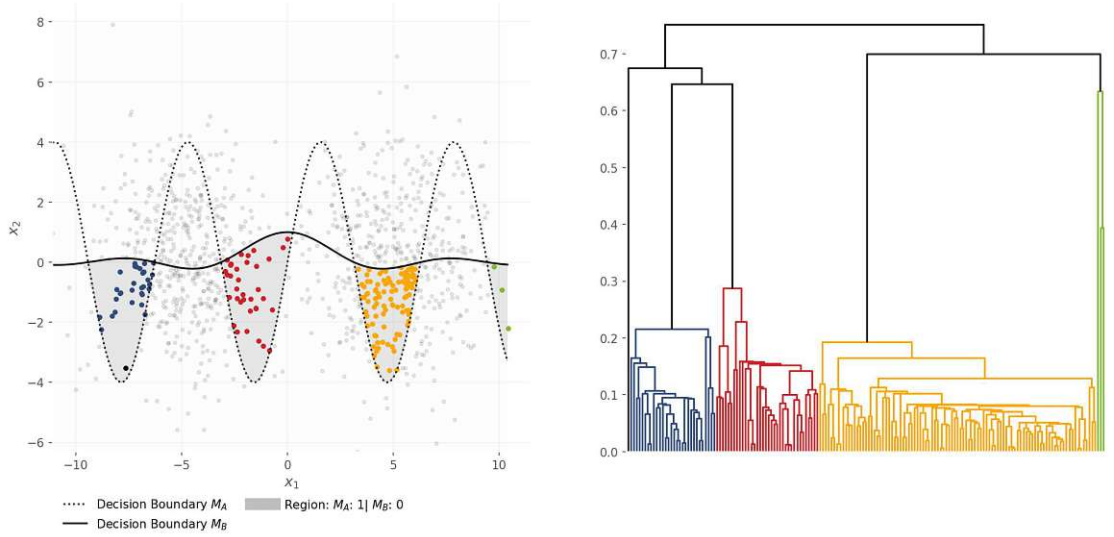
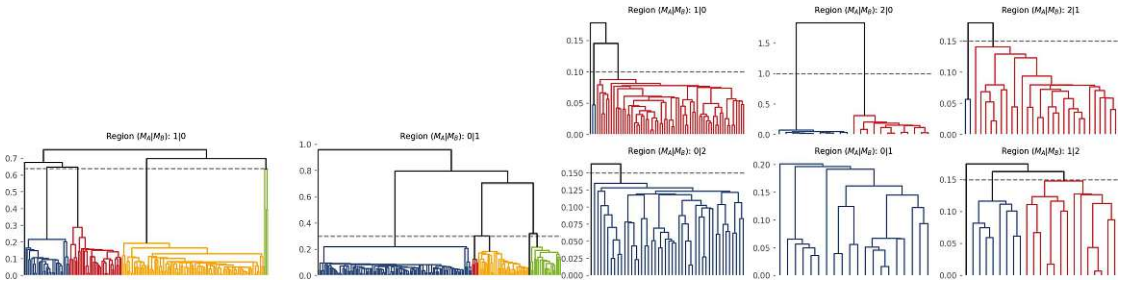


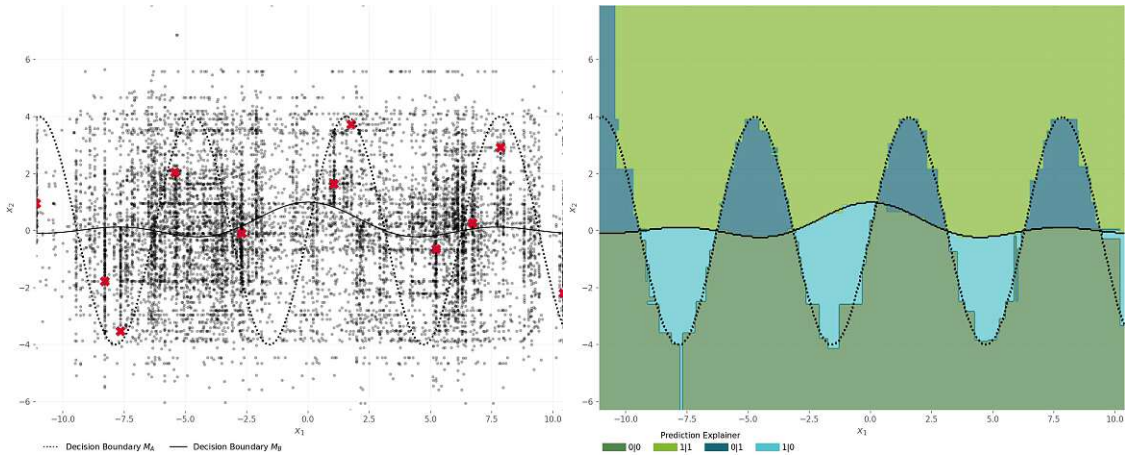
Figure 3.6: Visualization of cluster-stratified sampling (Approach 3) for single-linkage hierarchical-clustering of region $M_A|M_B : 1|0$ of the Sine running example



(a) Sine running example (b) Spiral running example

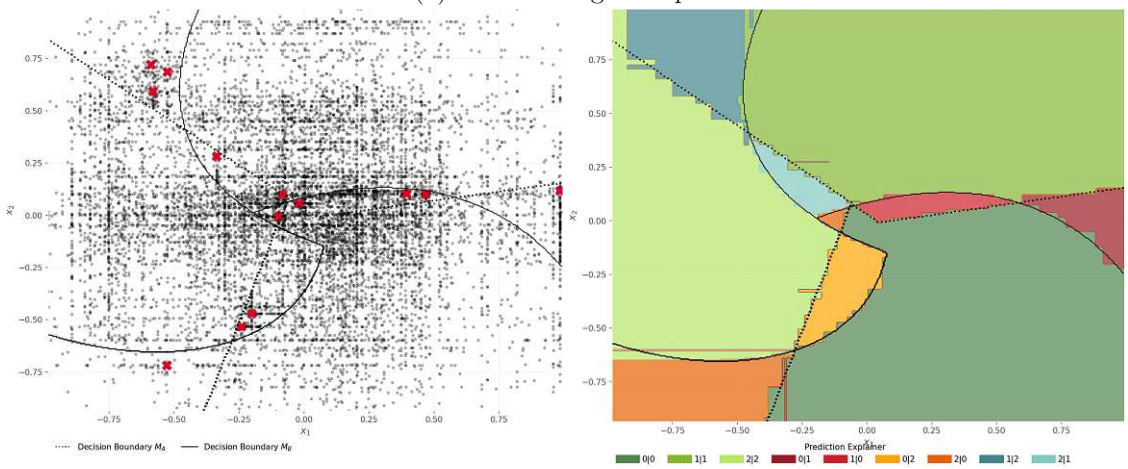
Figure 3.7: Dendrograms of single-linkage hierarchical clustering for each decision difference class

DiRo2C's synthetic neighborhoods generated for sampled instances from the starting, middle and ending SOM node of each cluster. This approach provides a more fine-granular sampling strategy since instances are not randomly sampled from the entire cluster but instead a more guided sampling is devised. The resulting explanations are structured and ordered linearly by the SOM. Using dimensionality-reduction techniques, the original input space can be projected onto two dimensions, and the regions for which a local explainer is responsible can be colored accordingly.
Figure 3.9 shows the trained 1-dim SOM for the Sine and Spiral running example. The dendrograms of single-linkage hierarchical clustering of the SOM nodes are shown in Figure 3.10. Using a distance threshold of 0.25 (Sine) and 0.225 (Spiral) for clustering with connectivity constraints according to the linear SOM relationship, the nodes are clustered into 6 cluster for Sine running example and 8 clusters for Spiral running example



(a.a) Generated neighborhoods (a.b) Decision surface of explainer

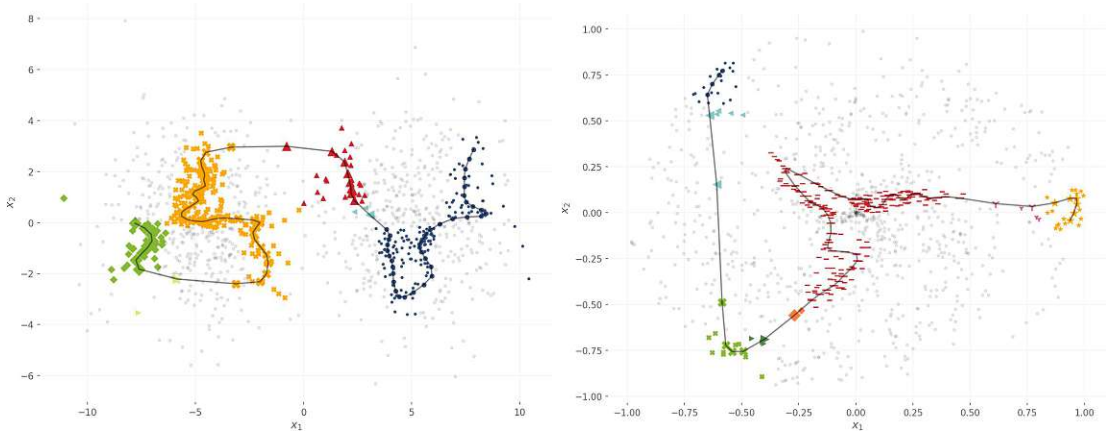
(a) Sine running example



(b.a) Generated neighborhoods (b.b) Decision surface of explainer

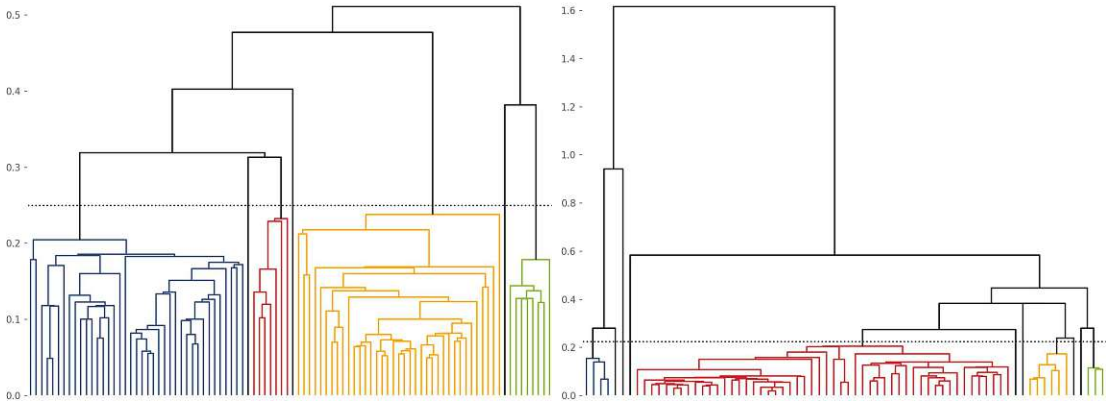
(b) Spiral running example

Figure 3.8: Approach 3: Cluster-stratified sampling



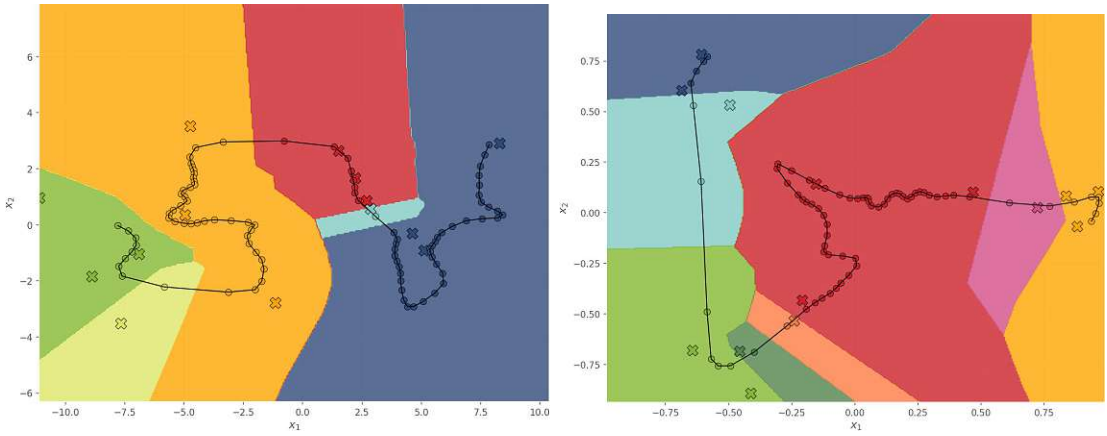
(a) Sine running example (b) Spiral running example

Figure 3.9: Structured sampling (Approach 4): DiRo2C's synthetic neighborhoods for a randomly sampled instance of each cluster



(a) Sine running example (b) Spiral running example

Figure 3.10: Structured sampling (Approach 4): Dendrograms of single-linkage hierarchical clustering of the nodes of the 1-dim structuring SOM

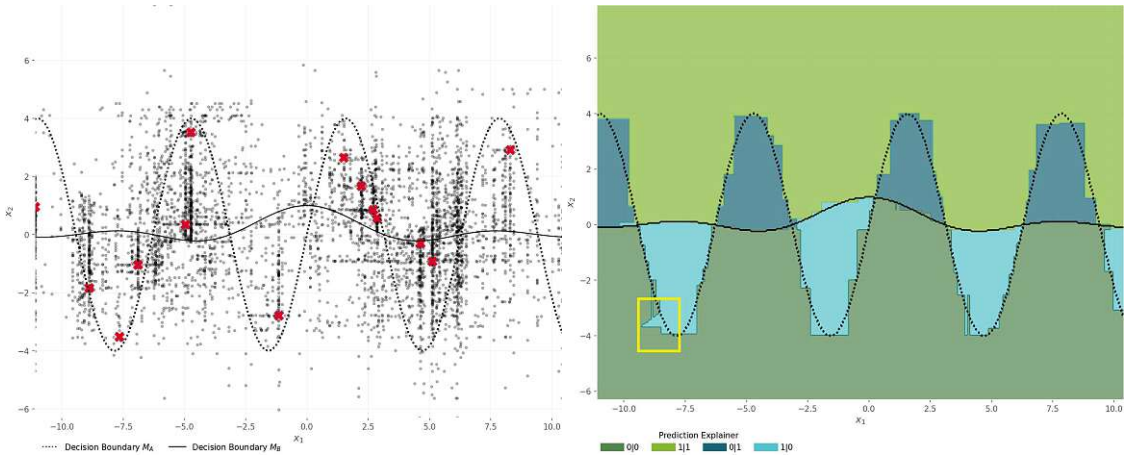


(a) Sine running example (b) Spiral running example

Figure 3.11: Structured sampling (Approach 4): Subdivided data-space according to clustering of nodes of 1-dim SOM. Each Cluster has its own explaining model based on the generated neighborhood around the marked sampled instance.

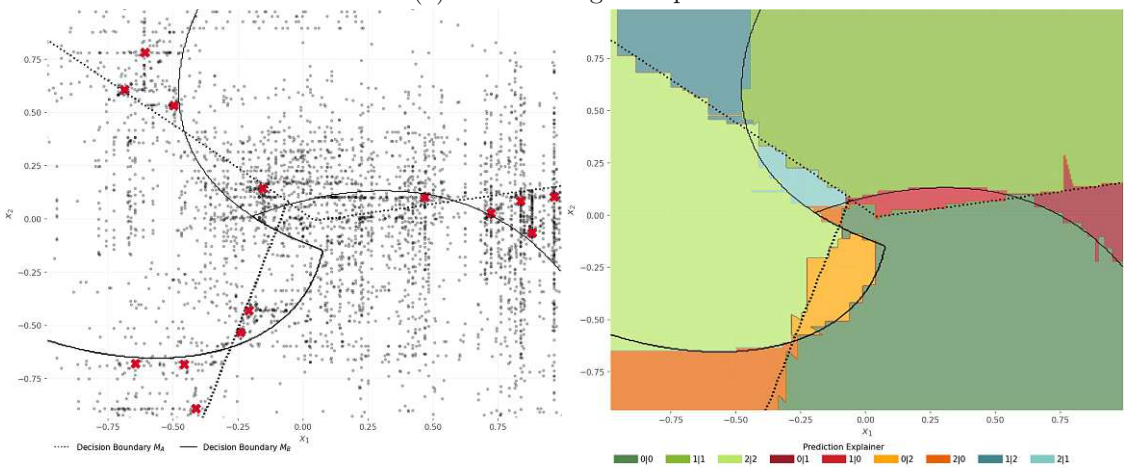
as visualized in 3.9. For whole clusters without training data, the nodes are assigned to their respective nearest cluster. For each cluster the starting, ending and middle SOM-nodes are selected. From each of these nodes an instance of the training set is sampled randomly and a genetic neighborhood via DiRo2C is generated as can be seen in Figure 3.12a.a and Figure 3.12b.a. These neighborhoods are then used to train a local explaining model for each cluster. Figure 3.11 shows the selected instances and the areas of the input-space for which the explainers are responsible, that is the set of data points for which the BMU of the 1-dim SOM is assigned to this specific cluster. In case no training data is assigned to one of the starting, middle or ending node, the next subsequent node is considered. For small clusters consisting of 3-4 nodes, only starting and ending nodes are considered and for clusters of 1-2 nodes a cluster is chosen randomly from which an instance is sampled.
In Figure 3.12 the sampled instances, generated neighborhoods and the decision surface of the structured combination of local explainers is visualized. The decision difference boundaries of the Sine running example are almost correctly recognized. The non-linear cluster boundaries entail non-linear boundaries of decision differences as can be seen for the marked area in Figure 3.12a.b. This incorrect predicted area is a result of the boundary of the second cluster colored in lime in Figure 3.11a.

Performance The datasets introduced in Section 2.1.4 are used as a training set X_{train}. A test set X_{test} for evaluation is generated using the same generation mechanisms as described with $n = 500$ data points for both running examples. Table 3.1 shows the performance of the proposed approaches for both running examples. Overall, with increasing complexity and granularity of the proposed approaches, performance also tends to increase. However, this comes at the cost of comprehensibility as the depth of



(a.a) Generated neighborhoods (a.b) Decision surface of explainer

(a) Sine running example



(b.a) Generated neighborhoods (b.b) Decision surface of explainer

(b) Spiral running example

Figure 3.12: Approach 4: Structured cluster-stratified sampling

Running Example	Approach	Accuracy	Precision	Recall	Depth of explainers
Sine	Approach 0: Baseline	0.904	0.904	0.900	10.000
	Approach 1: Random sampling	0.950	0.946	0.953	13.000
	Approach 2: Class-stratified sampling	0.972	0.972	0.970	13.000
	Approach 3: Cluster-stratified sampling	0.960	0.961	0.959	14.000
	Approach 4: Structured sampling	0.964	0.962	0.965	7.500
Spiral	Approach 0: Baseline	0.951	0.937	0.887	10.000
	Approach 1: Random sampling	0.969	0.946	0.956	12.000
	Approach 2: Class-stratified sampling	0.958	0.926	0.931	15.000
	Approach 3: Cluster-stratified sampling	0.968	0.943	0.946	15.000
	Approach 4: Structured sampling	0.957	0.922	0.932	5.375

Table 3.1: Performances of the proposed approaches: Running examples

* Averaged depth over explainers

the explainer also increases for the neighborhood combination approaches (Approaches 1-3). Nevertheless, all proposed approaches clearly outperform the Baseline. For the Sine running example, structured sampling (Approach 4) of instances yields the second-highest performance and the lowest depth of the explaining trees. Also for the Spiral running example the data structuring approach obtains high performance and the lowest depth of the decision tree.

3.3 Experimental Setup

The Approaches 1-4 introduced in Section 3.2 are compared to the baseline global explaining model (Approach 0). Since the explanation rules are based on the decision tree used as surrogate model, we used the performance of the decision tree as a measurement of correctness of the rules. The approaches are evaluated using accuracy, macro-averaged precision and recall on two benchmark datasets frequently used in literature, namely *Bank-Marketing* and *Compas*. Since we are interested in detecting usually underrepresented decision differences, macro-averaging of precision and recall was chosen in order not to obtain overly optimistic performance values. Both datasets are modified as described in 3.3.1 and 3.3.1 to obtain a basis for training of the second black box. Each dataset, original and modified, is split into three parts with the ratio (4:4:2): $X_{\text{M A/B}}$ to train the black boxes, $X_{\text{train A/B}}$ is used to select a set of instances to be explained or serves as training data for structuring and clustering, and $X_{\text{test A/B}}$ for evaluation. The training sets $X_{\text{train A}}$ and $X_{\text{train B}}$ as well as the test sets $X_{\text{test A}}$ and $X_{\text{test B}}$ are combined for each data set to X_{train} and X_{test}, respectively. An overview is given in Table 3.2.
SVM and Naive-bayes models are used as black box models and trained with parameter grid search with 3-fold Cross Validation. The parameter grids used are shown in Table 3.6. The black boxes are evaluated on X_{train} (i.e. the data that will be used to sample instances for which DiRo2C's neighborhood is generated in a next step) using accuracy, macro-averaged precision and recall. Performances are reported in Table 3.7.

Dataset	Observations	Features	$\lvert X_M \rvert$	$\lvert X_{\text{train}} \rvert$	Proportion of decision differences	$\lvert X_{\text{test}} \rvert$	Proportion of decision differences
Bank-Marketing	41,188	20	16,475	32,950	4.5%	16,476	4.4%
Compas	7,214	8	2,885	5,772	15.7%	2,886	15.9%

Table 3.2: Datasets statistics summary

Parameter	List of values	Compas	Bank-Marketing
Initial effective width σ_0	[0.5, 0.6, 0.7, 0.8, 0.9]	0.9	0.9
Initial learning rate $\alpha(t = 0)$	[0.2, 0.4, 0.6, 0.8, 1]	1.0	1.0

Table 3.3: Parameter grid used for hyper-parameter training of SOM and best parameter

A decision tree trained on X_{train} to learn decision differences with default parameters and evaluated on X_{test} acts as baseline global explainer. The decision tree with default parameter uses Gini impurity to measure the quality of a split, and the tree is expanded until for all leaves either the leave contains less than 2 instances or all instances of the leaves have the same label [1].
For RQ 1, instances are selected from X_{train} as described in Approaches 1-3 in Section 3.2. For each of these instances, a genetic neighborhood consisting of 1,000 data points is generated using DiRo2Cs neighborhood generation process. The neighborhoods are concatenated in a next step to a global synthetic dataset. Each instance of this dataset is classified using M_A and M_B and decision differences are determined. This dataset is then used as a basis for an explaining decision tree fitted with default parameter. To account for randomness, sampling of instances and the subsequent neighborhood generation is repeated 5 times and averaged performance and standard errors are reported.

For RQ 2, categorical features are One-Hot encoded and each continuous feature of X_{train} is normalized. For training of a one-dimensional SOM with $\lfloor 5 \cdot \sqrt{n} \rfloor$ nodes, only the subset $X_{\text{difference train}} \subset X_{\text{train}}$ with decision differences is considered. n denotes the number of instances of $X_{\text{difference train}}$. The initial effective width σ_0 and the initial learning rate $\alpha(0)$ are determined using parameter grid search with 3-fold CV using quantization error as performance measure, as shown in Table 3.3. The resulting $\lfloor 5 \cdot \sqrt{n} \rfloor$ weights are subsequently clustered using hierarchical clustering with single linkage criterion. The linear SOM neighborhood relation is used as a connectivity constraint of the weights in the process. An instance of the training/test set is assigned to a specific cluster by determining the BMU and thereafter the cluster of the respective BMU. BMUs without underlying data assigned to them are merged to the cluster with the nearest node. The number of sampled instances per cluster is made dependent on the cluster's size: in case a cluster contains more than 4 nodes the starting and ending node as well as the node in the middle are selected. For clusters containing 3-4 nodes, only the starting and

[1] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

ending node are considered and otherwise a random node is selected. For each of the nodes, an instance with the respective BMU of $X_{\text{difference train}}$ is randomly sampled and DiRo2C's genetic neighborhood with 1,000 data points is generated. Within each cluster, the resulting synthetic datasets are concatenated and each instance is classified by M_A and M_B and decision differences are determined. For each cluster, the corresponding synthetic dataset is cut off at the cluster boundaries by mapping each instance to its respective cluster as described before and then used as a basis for an explaining decision tree fitted with default parameter.
Using a SOM-projection onto two dimensions, the difference detection dataset is visualized and the regions of each responsible local explainer are colored. According to the number of training instances, the Bank-Marketing dataset is visualized using a 2-dimensional SOM with 30x30 nodes and Compas using 20x20 nodes.

For the generation of all genetic neighborhoods, 100 iterations (generations) are used. Mutation and crossover probabilities were set to 0.3 and 0.2, respectively. Tournament selection with 3 tournaments is used.

All analyses are conducted using scikit-learn [23] version 1.0.2, MiniSom [36] version 2.3.0 and DiRo2C[2] version 1.2. The experiments were performed on Microsoft Windows 10 Enterprise LTSC Version 10.0.17763, 64 GB RAM, 3.80GHz Intel Core(TM) i7-10700K. The source code is available at `https://github.com/jrckln/DiRo2CLocaltoGlobal`.

3.3.1 Data

In this section, the datasets used and their modifications are described.

Bank-Marketing

The Bank-Marketing dataset stems from a Portuguese retail bank from May 2008 to June 2013. It consists of information on clients contacted via phone in the course of a direct marketing campaign. The features include telemarketing attributes, product details and client information and was enriched by external data from the Central Bank of the Portuguese Republic. The aim is to predict whether a term deposit was sold to a client resulting in a binary response variable `y` of successful or unsuccessful contact [21]. The dataset contains 41,188 instances of 21 attributes and includes missing values encoded as 'unknown' which are treated as a separate category.
For the analysis all available features were used:

- `age`: Age of bank client in years
- `job`: Type of job of bank client
- `marital`: Marital status of bank client

[2] `https://gitlab.com/andsta/diro2c`

- `education`: Highest education of bank client
- `default`: Has bank client credit in default?
- `housing`: Has bank client a housing loan?
- `loan`: Has bank client a personal loan?
- `contact`: Contact communication type
- `month`: Month of last contact
- `day_of_week`: Day of the week of last contact
- `duration`: Duration of last contact in seconds
- `campaign`: Number of contacts performed during this campaign
- `pdays`: Number of days passed by after the bank client was last contacted from a previous campaign
- `previous`: Number of contacts performed before this campaign and for this bank client
- `poutcome`: Outcome of the previous marketing campaign
- `emp.var.rate`: Employment variation rate
- `cons.price.idx`: Consumer price index
- `cons.conf.idx`: Consumer confidence index
- `euribor3m`: Euribor 3 month rate
- `nr.employed`: Number of employees

Table 3.4 shows basic statistical properties of the Bank-Marketing dataset. The binary response variable `y` is not evenly distributed among classes (88.73% unsuccessful ('No') and 11.27% successful ('Yes') contact).

Modification To have a ground-truth of expected black box model differences between M_A and M_B, the original Bank-Marketing dataset is modified as follows: The value 5 is added to each value of the variable `pdays` [31]. Additionally, `education` is decreased by one level for clients in management positions and entrepreneurs, if applicable, and `age` is decreased by 10 for bank clients with a personal loan and increased by 10 for bank clients with a housing loan.

Feature	Classes (n)	Top category	Missing values
job	12	admin.	330
martial	4	married	80
education	8	university.degree	1731
default	3	no	8597
housing	3	yes	990
loan	3	no	990
contact	2	cellular	0
month	10	may	0
day_of_week	5	thu	0
p_outcome	3	nonexistent	0
y	2	no	0

Feature	Range	Mean (SD)	Missing values
age	17 - 98	40.02 (10.42)	0
duration	0 - 4918	258.29 (259.28)	0
campaign	1 - 56	2.57 (2.77)	0
pdays	0 - 999	962.48 (186.91)	0
previous	0 - 7	0.17 (0.49)	0
emp.var.rate	(-3.4) - 1.4	0.08 (1.57)	0
cons.price.idx	92.2 - 94.77	93.58 (0.58)	0
cons.conf.idx	(-50.8) - (-26.9)	-40.5 (4.63)	0
euribor3m	0.63 - 5.04	3.62 (1.73)	0
nr.employed	4963.6 - 5228.1	5167.04 (72.25)	0

Table 3.4: Basic properties of the Bank-Marketing dataset

Compas

COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) is a software by Northpointe, Inc. to score the likelihood of a criminal to become a recidivist. ProRepublic's analysis of predicted scores and actual recidivist rates of more than 10,000 criminal defendants in Florida over a two-year period revealed an ethnic bias [15]. Scoring of defendants is based on a questionnaire. The answers are used by the COMPAS software to generate multiple scores from 1-10 including at least 'Risk of Recidivism', 'Risk of Violence' and 'Risk of Failure to Appear' for each defendant [15]. The Compas dataset[3] was obtained from the Broward County Sheriff's Office Florida for 11,757 subjects and contains information on sex, age and race as well as length of stay in jail, degree of charge etc. 'Risk of Recidivism' was considered as response and was labeled according to COMPAS as 'Low' for scores of 1-4, 'Medium' for scores of 5-7 and 'High' for scores 8 to 10 [15]. For the analysis, the following features were used:

[3]https://github.com/propublica/compas-analysis/blob/master/compas-scores-two-years.csv

Feature	Classes (n)	Top category	Missing values
sex	2	Male	0
race	6	African-American	0
c_charge_degree	2	F	0
score_text	3	Low	0

Feature	Range	Mean (SD)	Missing values
age	18 - 96	34.82 (11.89)	0
juv_fel_count	0-20	0.07 (0.47)	0
juv_misd_count	0-13	0.09 (0.49)	0
juv_other_count	0-17	0.11 (0.5)	0
priors_count	0-38	3.47 (4.88)	0

Table 3.5: Basic properties of the Compas dataset

- sex: Gender (Male or Female)
- age: Age in years
- race: Race (African-American, Asian, Caucasian, Hispanic, Native American or Other)
- juv_fel_count: Number of juvenile felonies
- juv_misd_count: Number of juvenile misdemeanors
- juv_other_count: Number of prior juvenile convictions that are not felonies or misdemeanors
- priors_count: Number of prior convictions
- c_charge_degree: Charge degree
- score_text: Risk of Recidivism by COMPAS (Low, Medium or High)

Table 3.5 shows basic statistical properties of the chosen features of the Compas dataset. The response variable score_text is not evenly distributed among classes: 54.03% were classified as 'Low' risk, 26.53% as 'Medium' and 19.45% were classified as 'High' risk of recidivism.

Modification To train model M_B, the dataset was modified so that the number of prior convictions of individuals older than 60 years is increased by 5 and reduced by 5 for individuals younger than 30 years. Additionally, the charge degree of African-Americans with less than 2 juvenile felonies was decreased by one level from felony to misdemeanor.

Model	Parameter	List of values	Bank-Marketing	Compas
Black box A: SVM	C	[0.1, 1, 10, 100]	0.1	100
Black box A: SVM	kernel	['rbf', 'sigmoid', 'linear']	linear	rbf
Black box A: SVM	gamma	['scale', 'auto']	scale	scale
Black box B: Naive Bayes	var_smoothing	[1.e+00 1.e-01 1.e-02 1.e-03 1.e-04 1.e-05 1.e-06 1.e-07 1.e-08 1.e-09]	0.01	0.0001

Table 3.6: Parameter grid used for hyper-parameter training of the black box models and resulting best parameter

	Bank-Marketing		Compas	
Metric	Data A	Data B	Data A	Data B
Accuracy	0.907	0.906	0.638	0.599
Precision (macro)	0.813	0.782	0.581	0.552
Recall (macro)	0.648	0.689	0.515	0.444

Table 3.7: Performances of black box model training

3.4 Results

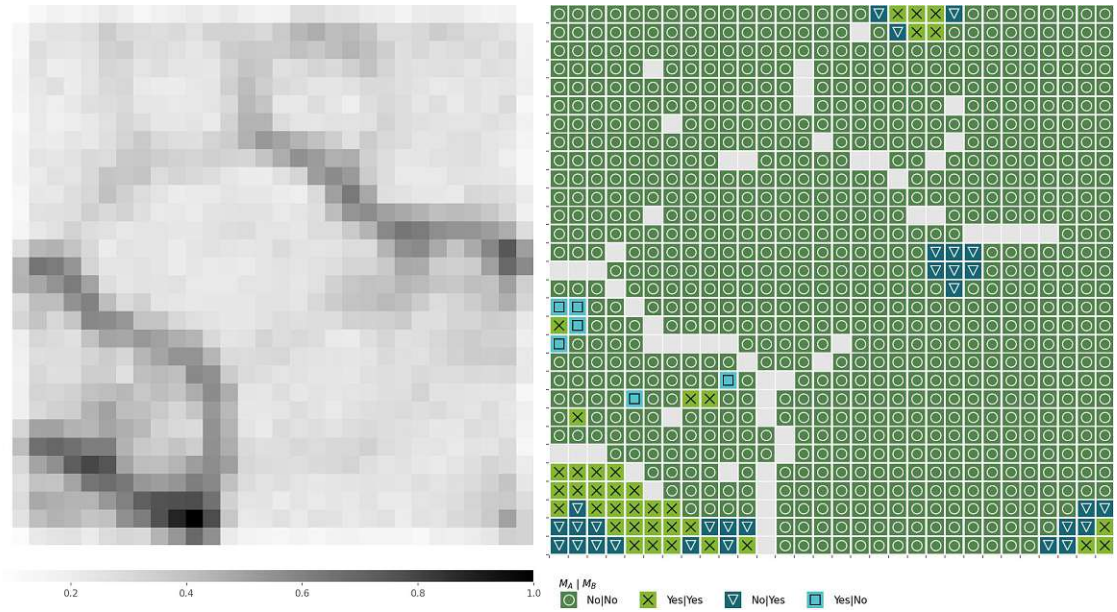
In the following, the results of the conducted experiments are presented. In Section 3.4.1, a brief summary of the black box training results is provided, and preliminaries for the proposed approaches are presented followed by the evaluation of the approaches in Section 3.4.2 by means of the benchmark datasets.

3.4.1 Black Boxes and Preliminary Analysis

Black box training Table 3.6 shows the parameter grids used for grid search with 3-fold CV of the two black box classifiers and the corresponding chosen parameters for training. Performances of the black box models evaluated on the black box test set X_{train} are reported in Table 3.7.

Cluster-stratified sampling (Approach 3): Number of Clusters Figure 3.13 and Figure 3.15 show SOM-projections of Bank-Marketing and Compas datasetd. Figure 3.14 and Figure 3.16 show the dendrograms of hierarchical clustering using Ward's-linkage criterion of the training set of Bank-Marketing and Compas. Based on the projections and the accompanying dendrograms, the number of distinct regions for cluster-stratified sampling of instances (Approach 3) were appointed as stated in Table 3.8, resulting in 26 instances for Compas and 21 instances for Bank-Marketing, for which a synthetic neighborhood is generated.

Structured sampling (Approach 4) Figure 3.17 shows dendrograms of the hierarchical clustering of SOM nodes with single-linkage criterion and connectivity constraints according to linear SOM neighborhood relation. Based on the dendrogram, distance thresholds of 4.5 for Bank-Marketing and 3.0 for Compas were chosen with which the SOM-nodes are clustered into 10 and 8 clusters, respectively. Using a 2-dim SOM, Figure



(a) U-Matrix

(b) Label color coding

Figure 3.13: SOM-Projection of training data of Bank-Marketing

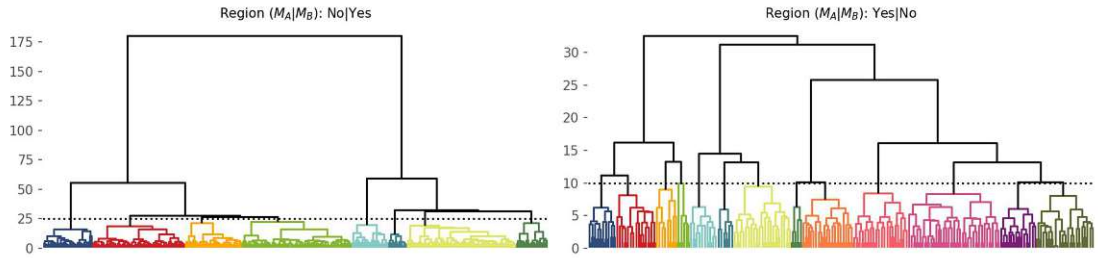
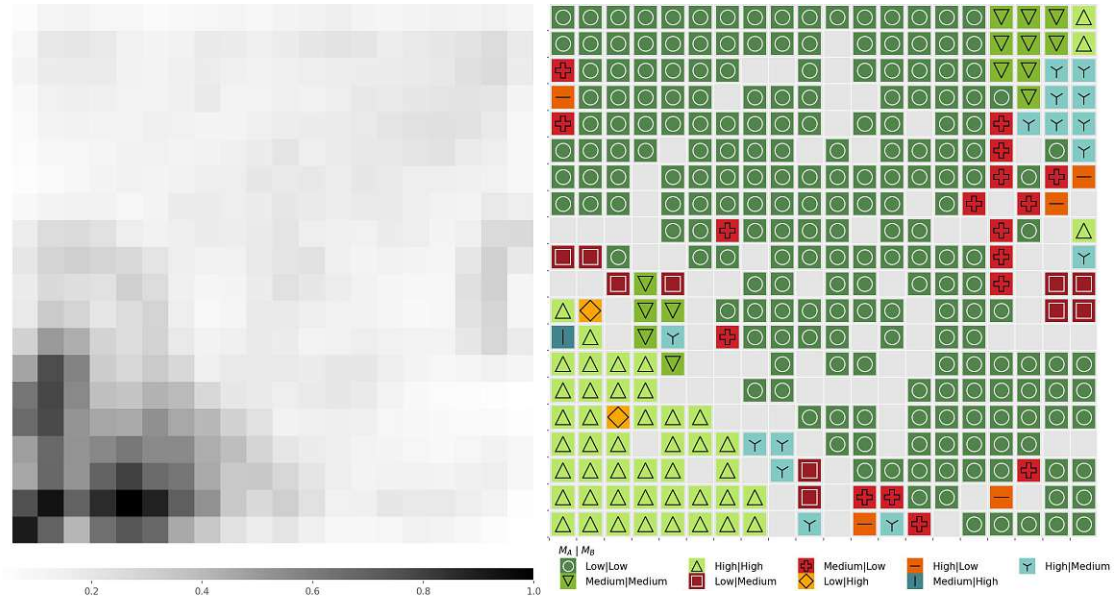


Figure 3.14: Dendrograms of hierarchical clustering of decision difference regions of Bank-Marketing

Region M_A|M_B	Compas
Low|Medium	4
Low|High	5
Medium|Low	2
Medium|High	8
High|Low	3
High|Medium	4

Region M_A|M_B	Bank-Marketing
No|Yes	8
Yes|No	13

Table 3.8: Cluster-stratified sampling (Approach 3): Number of chosen regions of decision differences



(a) U-Matrix (b) Label color coding

Figure 3.15: SOM-Projection of training data of Compas

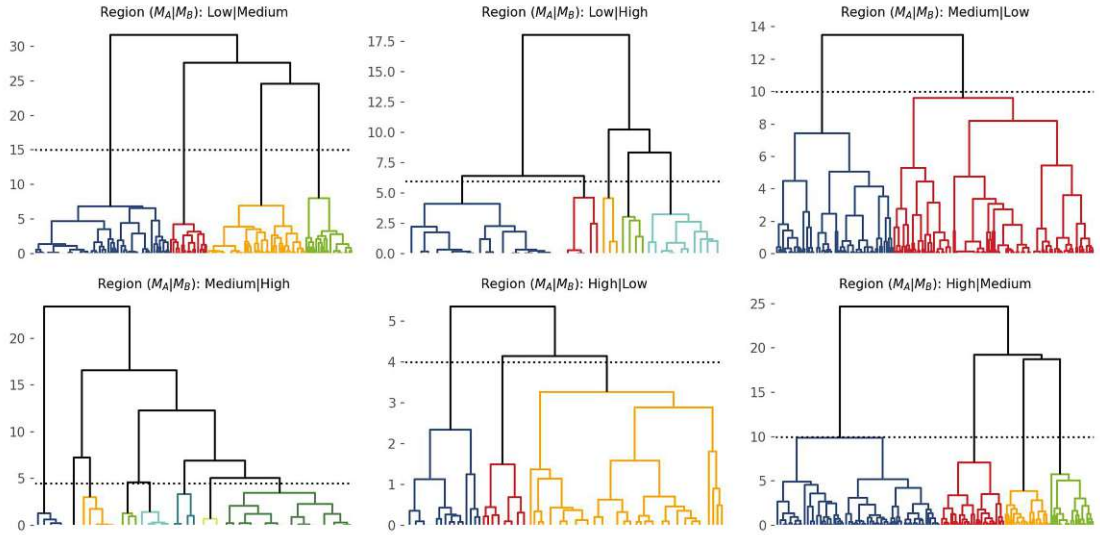
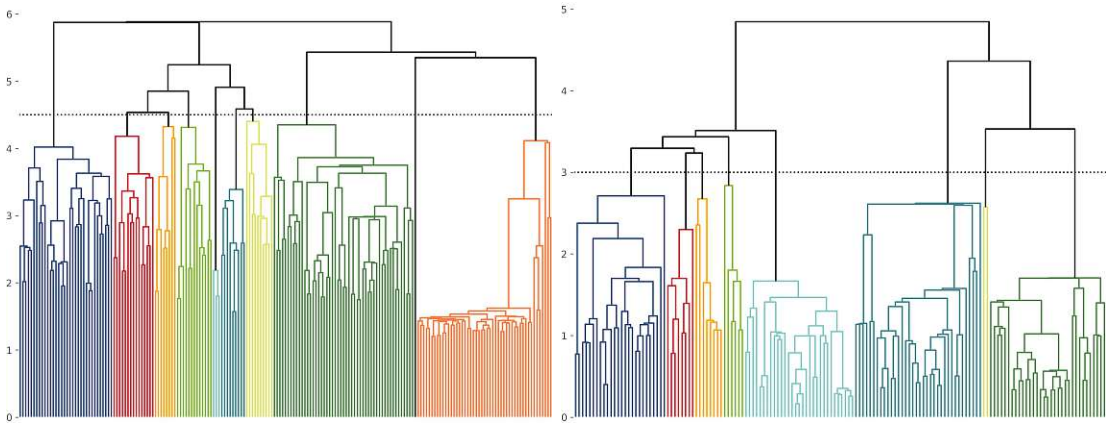


Figure 3.16: Dendrograms of hierarchical clustering of decision difference regions of Compas



(a) Bank-Marketing (b) Compas

Figure 3.17: Structured sampling (Approach 4): Dendrograms of hierarchical clustering of SOM nodes with single-linkage and connectivity constraints according to linear SOM neighborhood relation

Dataset	Approach	Number of instances	Accuracy (SD)	Precision (SD)	Recall (SD)	Depth of explainer (SD)
Bank-Marketing	Approach 0: Baseline	-	0.988	0.895	0.897	12
	Approach 1: Random sampling	21	0.95 (0.007)	0.585 (0.067)	0.572 (0.057)	22.2 (1.924)
	Approach 2: Class-stratified sampling	21	0.945 (0.007)	0.643 (0.025)	0.754 (0.022)	30.6 (2.702)
	Approach 3: Cluster-stratified sampling	21	0.948 (0.007)	0.666 (0.039)	0.798 (0.041)	25.2 (3.493)
	Approach 4: Structured sampling	10*	0.918 (0.026)	0.587 (0.033)	0.739 (0.033)	14.22 (0.638)
Compas	Approach 0: Baseline	-	0.943	0.747	0.759	15
	Approach 1: Random sampling	26	0.899 (0.009)	0.654 (0.029)	0.631 (0.038)	26.8 (2.588)
	Approach 2: Class-stratified sampling	26	0.916 (0.009)	0.713 (0.024)	0.746 (0.025)	25.2 (3.347)
	Approach 3: Cluster-stratified sampling	26	0.914 (0.011)	0.734 (0.028)	**0.784 (0.025)**	25.4 (2.302)
	Approach 4: Structured sampling	8*	0.807 (0.02)	0.57 (0.045)	0.652 (0.048)	**13.325** (0.59)

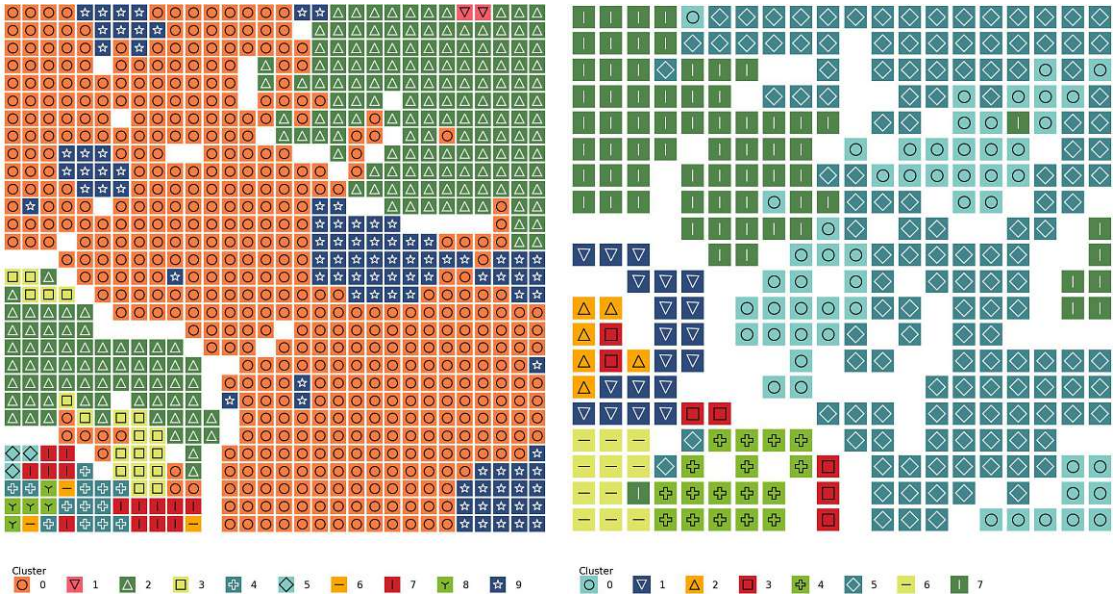
Table 3.9: Performances of the proposed approaches: Benchmark datasets, averaged over 5 runs

* Number of clusters

3.18 visualizes for each node the most frequent cluster of instances with the respective node as BMU.

3.4.2 Local-to-Global

Table 3.9 shows the average performance of the proposed approaches for global explanations over 5 runs. Performance overall could not be improved for both benchmark datasets as compared to the Baseline. Random sampling of instances (Approach 1) yields the highest accuracy for Bank-Marketing of the proposed approaches. However, accuracy of class- and cluster-stratified sampling (Approach 2 and Approach 3) is almost as high. Macro-averaged precision and recall could indeed be improved by advanced and strategic selection of instances, resembling the focus on instances with decision differences. For Compas, cluster-stratified sampling of instances yields a macro-averaged precision



(a) Bank-Marketing (b) Compas

Figure 3.18: Structured sampling (Approach 4): SOM-nodes colored according to most frequent cluster

similar to the Baseline and even outperforms the Baseline for macro-averaged recall. In case of the Bank-Marketing dataset, cluster-stratified sampling of instances could not attain the performance of the Baseline. For both benchmark datasets, accuracy and macro-averaged precision of the data structuring approach (Approach 4) are slightly lower and macro-averaged recall is slightly higher as compared to the performance of random sampling of instances. Nevertheless, the depth of the explaining trees is lower for structured sampling of instances and even lower than the Baseline for Compas.

Effect of complexity of the explainer The complexity of a decision tree is measured by either the number of nodes, the number of leaves, depth of the tree or the number of used attributes [17]. The comprehensibility is often measured by the complexity of a classifier [25]. Grounded by cognitive load theory, the maximum depth of decision trees to be comprehensible was determined to be at seven [6, 19]. Since decision trees with a depth of over 20 cannot be considered explainable anymore, we additionally evaluated performance of explainers pre-pruned to a maximum of seven levels.
Figure 3.19 shows the averaged performance with standard errors over 5 runs for the different approaches with varying maximum depth of the explainer for 21 and 26 sampled instances for random, class-stratified and cluster-stratified sampling of instances (Approaches 1-3) and 10 and 8 clusters for structured sampling (Approach 4) of Bank-Marketing and Compas, respectively. For Bank-Marketing, the depth of the explainer does not have much influence on accuracy for all approaches whereas for Compas with increasing depth of the explaining tree, a strong increase in accuracy can be observed for

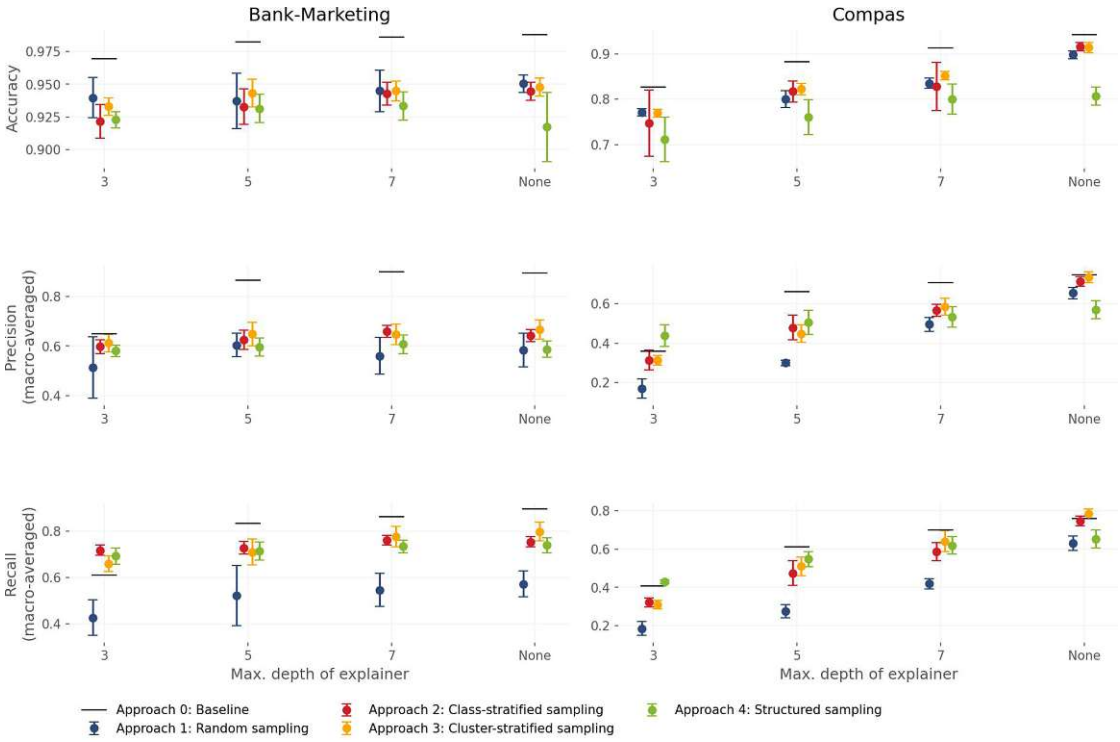


Figure 3.19: Effect of complexity of the explainer

random, class-stratified and cluster-stratified sampling. The increase in performance for increasing depth of the tree is even stronger for macro-averaged precision and recall in case of the Compas dataset. While there is an increase in performance as the depth increases for Bank-marketing, the magnitude of this increase is lower as compared to the Compas dataset. Even for a low maximum depth of the explainer, structured sampling (Approach 4) outperforms the Baseline in terms of macro-averaged recall for Bank-Marketing and additionally in terms of macro-averaged precision for Compas and, thus the structured sampling approach provides easily comprehensible but still the accurate explanations.

Effect of number of sampled instances Figure 3.20 shows the averaged performance with standard errors over 5 runs of accuracy, macro-averaged precision and recall and depth of the resulting explainer for random, class-stratified and cluster-stratified sampling for a varying number of instances of Bank-Marketing and Compas. Overall, there is an increase in performance for increasing complexity of sampling strategies. However, as performance increases, also complexity of the explaining decision tree increases. For each doubling of the number of sampled instances for which a neighborhood is created, mean accuracy increases by 1.8% on average for random sampling (Approach 1) and class-stratified sampling (Approach 2). Doubling the number of instances has a higher effect on precision and recall, increasing mean macro-averaged precision by 11.0% and 8.0% and mean macro-averaged recall by 9.9% and 6.9% for random sampling (Approach

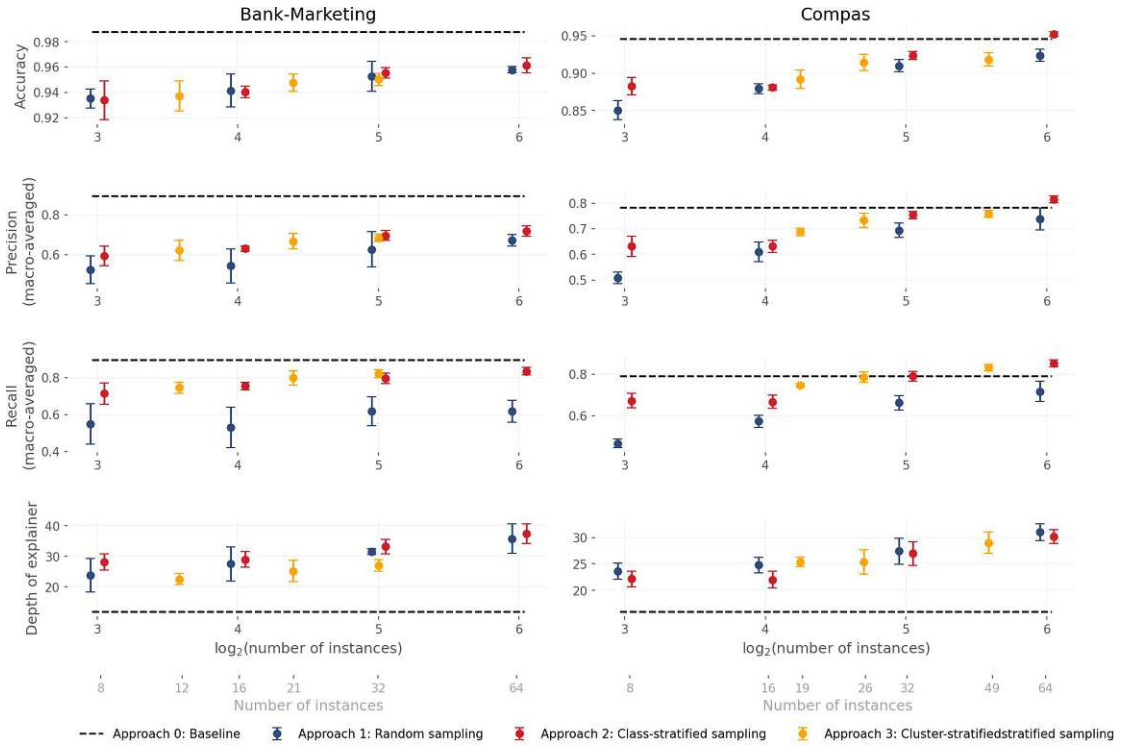


Figure 3.20: Approaches 1.1-1.3: Impact of number of sampled instances on performance

1) and class-stratified sampling (Approach 2), respectively.
The performance of the baseline explainer could not be attained for 64 randomly (Approach 1), class-stratified (Approach 2) nor cluster-stratified (Approach 3) sampled instances of the Bank-Marketing dataset. For Compas, a better performance than baseline was attained using 64 class-stratified sampled instances (Approach 2). Strategic selection of instances (Approach 3: Cluster-stratified sampling) already gives similar performance to the baseline for 49 instances.
Especially for a low number of randomly sampled instances, macro-averaged precision and recall are very low at 50%-55%, indicating that decision differences are mainly missed, and only matching predictions of the two black boxes are predicted by the explainer. Sampling stratified random according to the decision differences tackles this issue and increases performance throughout all scenarios. Consistent with expectations, cluster-based sampling of instances does increase precision and recall even further as compared to stratified random sampling of instances.

Effect of the number of clusters Table 3.10 shows the performance of the structured sampling approach (Approach 4) depending on the number of clusters the data is partitioned into. Although for the lowest number of clusters, this approach yields a fair accuracy of 0.918 (SD: 0.026) for Bank-Marketing and 0.807 (SD: 0.02) for Compas, performance in terms of macro-averaged precision (Bank-Marketing: 0.587 (SD: 0.033),

Dataset	Distance threshold	Clusters	Accuracy (SD)	Precision (SD)	Recall (SD)	Average depth of explainers (SD)
Bank-Marketing	4.5	10	0.918 (0.026)	0.587 (0.033)	0.739 (0.033)	14.22 (0.638)
	4.0	20	0.85 (0.11)	0.572 (0.021)	0.687 (0.048)	11.79 (0.178)
	3.5	39	0.893 (0.059)	0.604 (0.031)	0.766 (0.022)	10.867 (0.526)
	3.0	64	0.89 (0.021)	0.573 (0.023)	0.722 (0.029)	9.168 (0.33)
Compas	3.0	8	0.807 (0.02)	0.57 (0.045)	0.652 (0.048)	13.325 (0.59)
	2.0	20	0.779 (0.052)	0.583 (0.033)	0.665 (0.033)	11.35 (0.514)
	1.5	36	0.757 (0.046)	0.58 (0.017)	0.676 (0.017)	10.05 (0.169)
	1.25	50	0.822 (0.052)	0.603 (0.015)	0.714 (0.017)	10.02 (0.491)

Table 3.10: Structured sampling (Approach 4): Performance depending on the number of clusters

Compas: 0.57 (SD: 0.045)) and recall (Bank-Marketing: 0.739 (SD: 0.033), Compas: 0.652 (SD: 0.048)) is poor.
Increasing the number of clusters and hence decreasing the area for which an explainer is responsible, does not generally increase the performance as it was the case for random, class-stratified and cluster-stratified sampling (Approaches 1-3). For Bank-Marketing, mean accuracy tends to decrease while for mean macro-averaged precision and recall no trend can be observed. On the contrary, in case of Compas, mean accuracy first decreases and then increases whereas mean macro-averaged precision and recall increase.

3.5 Summary

The goal of this part was the derivation of a global explainer by utilization of local explanation generation techniques. We have proposed four bottom-up approaches to obtain a global explanation. First, we used multiple with DiRo2C generated neighborhoods of pre-specified instances and combined them to a global dataset serving as training data for a global explanation model. For this neighborhood combination approach, three strategies to select instances were presented and analyzed. Second, the data structuring approach establishes an ordering of decision differences and provides multiple simple local explainers.
For the continuous running examples, a higher performance was observed for all approaches as compared to the baseline explaining model. Especially cluster-stratified sampling and the subsequent neighborhood combination (Approach 3) clearly outperforms the baseline for all strategies to select instances. A similar high performance was observed for the data structuring approach (Approach 4) which additionally yields a lower mean depth of the explaining decision trees. In case of the benchmark datasets, the proposed approaches overall attain slightly lower performance. However, for Compas, the performance of the baseline could be surpassed by class-stratified sampling of 64 instances. With cluster-stratified sampling, a performance similar to the baseline could already be attained by sampling 49 instances. Overall, strategic and advanced selection

of instances mainly has a positive impact on macro-averaged precision and recall as compared to random sampling of instances.
The structured combination of local explanations did not outperform a single global explanation (Baseline) by means of difference detection accuracy, macro-averaged precision and recall for the benchmark datasets for explainers unrestricted in depth. However, since structured sampling (Approach 4) uses multiple local explainers, the mean depth of the decision trees is lower and even outperforms the Baseline for Compas. For a maximum of three levels of the explaining tree(s), the structured sampling approach even outperforms the Baseline by means of macro-averaged recall for Bank-Marketing and additionally macro-averaged precision for Compas. Thus, this approach provides a sequence of simple and therefore easily comprehensible local explaining models while maintaining good performance.

CHAPTER 4

Communication of Decision Differences in a Multi-Class Setting

While the first part of this thesis focused on the generation of explanations for differences in decisions between two black box classifiers, the second part aims at presenting and communicating these generated explanations in a comprehensible and compact way. The third research question is answered by the description and discussion of the process of analyzing possible decision differences in a multi-class setting. The clustering of data by the structured sampling approach (Approach 4) provides the foundation for this chapter. In the following, the process is described and visualized using the Spiral running example and Compas dataset due to the multi-class setting and hence increased complexity as compared to a binary setting of black box classifier for the Sine running example and Bank-Marketing dataset. The evaluation of RQ 3 is done argumentatively since a user study goes beyond the scope of this thesis.

4.1 Step 1: Existence of Decision Differences

The starting point of the whole process of analyzing possible decision differences is a dataset and two k-class classifiers. The first question arising in this context is about the existence of decision differences. The main goal in this step is to determine if there are any decision differences. Not all $k \times k$ possible combinations of predictions of the black boxes are of interest, but rather the reduction to binary problem: differences and no differences.
The distribution of decision differences across training data of the Spiral running example is visualized in Figure 4.1, showing a simple Scatter plot colored according to presence

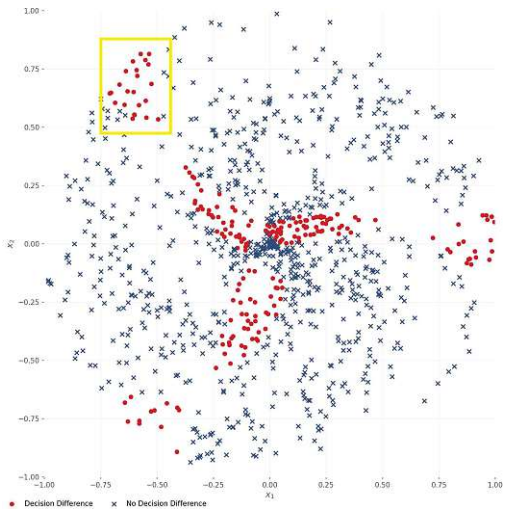


Figure 4.1: Spiral running example colored according to presence of decision differences

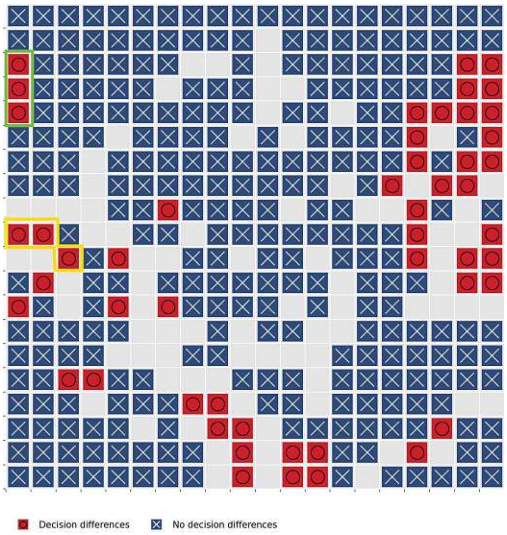


Figure 4.2: SOM-Projection of Compas dataset: Label color coding

of decision differences. It shows the existence of decision differences in form of multiple disjoint areas.
In case of more than two features, dimensionality reduction methods have to be employed. Here we will focus on 2-dimensional SOMs due to the method's topology preserving properties. A SOM projection of the multidimensional Compas training dataset is shown in Figure 4.2. It reveals many individual areas where decision differences are in a majority which are not connected to one another.

4.2 Step 2: Global Image

The next step is to determine **where** decision differences occur. The data structuring of Approach 4 introduced in Section 3.2 provides a sequence of explainer for the dataset. As a starting point, Figure 4.3 for the Spiral running example and Figure 4.4 for Compas show for which partition of the data-space which explainer applies. For Compas, this is done via a SOM projection and a pie-chart visualization that shows the proportion of instances of the training data per cluster falling in the specific neuron. Additionally, the values of each feature can be shown over the whole map, using component planes to provide an idea of the distribution of values as shown in Figure 4.5 for the features of Compas.

Based on the clustered projections and the corresponding position in the component planes, the three nodes of decision differences marked green in Figure 4.2 are assigned to cluster No. 7, and individuals are mostly caucasian men aged about 20 years. They do not have any juvenile convictions, however, they have 5-10 prior convictions and were

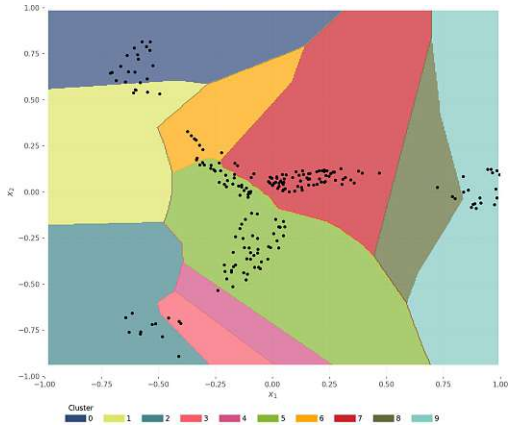


Figure 4.3: Clustered Spiral running example

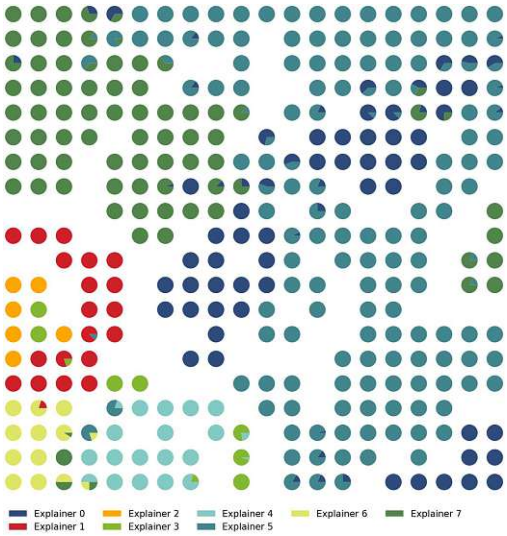


Figure 4.4: Clustered Compas dataset

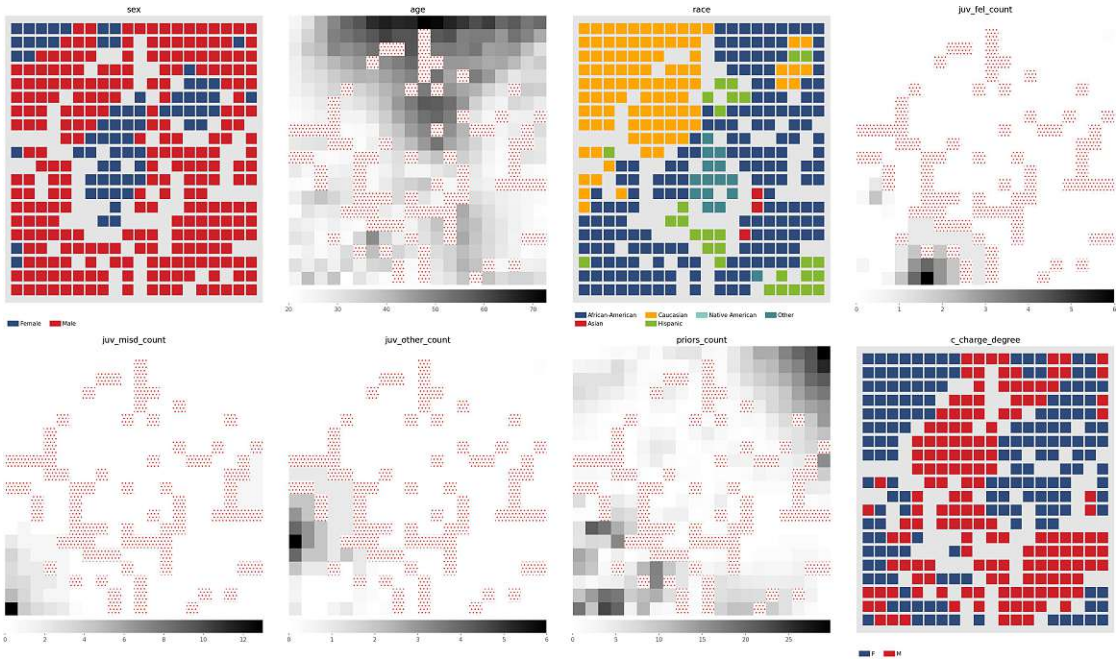
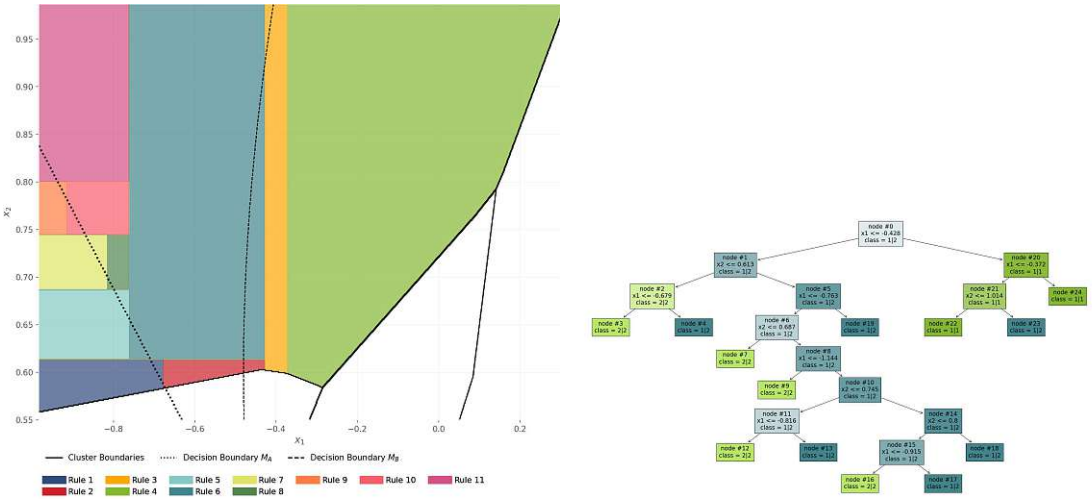


Figure 4.5: Component planes for each feature of Compas



(a) Areas for which rules applies (b) Explainer

Figure 4.6: Spiral running example: Cluster No. 0

charged with felony before. Overall, we see that decision differences are present for a medium number of prior convictions and male Caucasian or African-Americans mostly between 25-35 years.

4.3 Step 3: Single Explanations

For each area of decision differences of Step 1, the corresponding main cluster as introduced in Section 3.2 can be determined, and further the explaining tree of this cluster can be shown to obtain rules applying for this specific subset of data. The explaining decision trees presented in the following were pre-pruned to a maximal depth of 3. Besides the explanation for this cluster, a simple statistical summary of the cluster is added to provide information on the data in this cluster for which the explaining rules apply. The neighborhood generation process of DiRo2C allows for a more thoroughly and precise description of each cluster compared to reliance on training data.

For the Spiral running example, the majority of the instances marked in Figure 4.1 is covered by cluster No. 0 as can be seen in Figure 4.3. The explainer of cluster No. 0 is visualized in Figure 4.6b. The first split of the decision tree rules out the class combination $M_A|M_B : 1|1$ for instances with $x_1 \leq -0.428$. The remaining splits mimic the linear decision boundary between the class combinations $M_A|M_B : 1|2$ and $2|2$ as a step function (see Figure 2.2).

Figure 4.6a shows cluster No. 0 in detail: for each resulting rule from the decision tree (i.e. the path from the root to every leaf), the area for which the rule applies is shown. The explainer predicts decision differences in this cluster according to the following rules. The step function approximation of the linear boundary between the class combinations $M_A|M_B : 1|2$ and $2|2$ can be identified in the rules by alternating features and a similar

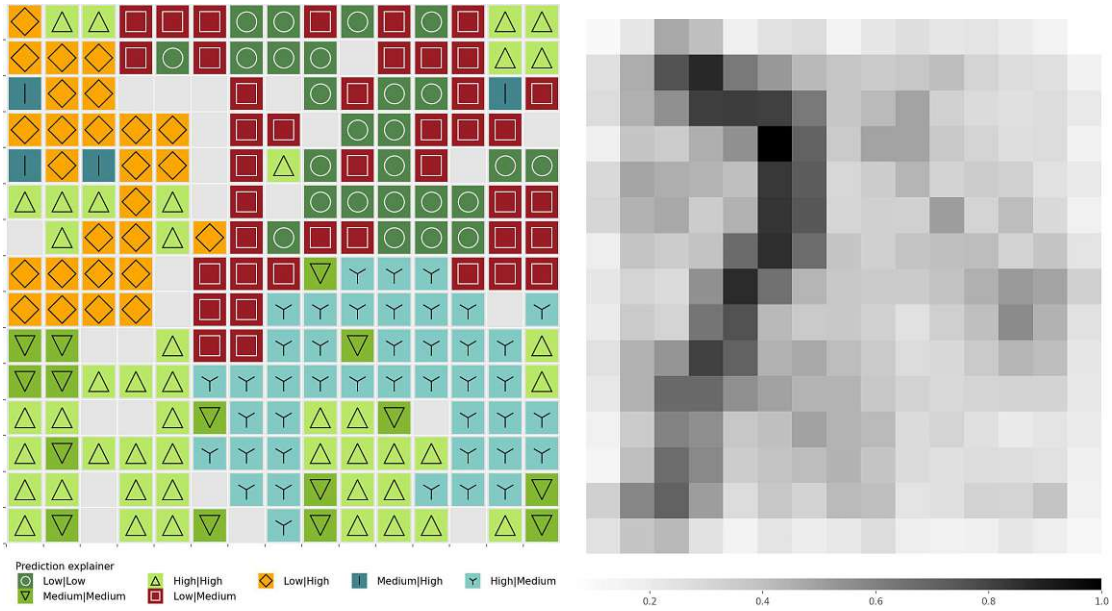
step size per feature.

Rule 1: if $(x1 \leq -0.428)$ and $(x2 \leq 0.613)$ and $(x1 \leq -0.679)$ then class: 2|2
Rule 2: if $(x1 \leq -0.428)$ and $(x2 \leq 0.613)$ and $(x1 > -0.679)$ then class: 1|2
Rule 3: if $(x1 \leq -0.428)$ and $(x2 > 0.613)$ and $(x1 \leq -0.763)$ and $(x2 \leq 0.687)$ then class: 2|2
Rule 4: if $(x1 \leq -0.428)$ and $(x2 > 0.613)$ and $(x1 \leq -0.763)$ and $(x2 > 0.687)$ and $(x1 \leq -1.144)$ then class: 2|2
Rule 5: if $(x1 \leq -0.428)$ and $(x2 > 0.613)$ and $(x1 \leq -0.763)$ and $(x2 > 0.687)$ and $(x1 > -1.144)$ and $(x2 \leq 0.745)$ and $(x1 \leq -0.816)$ then class: 2|2
Rule 6: if $(x1 \leq -0.428)$ and $(x2 > 0.613)$ and $(x1 \leq -0.763)$ and $(x2 > 0.687)$ and $(x1 > -1.144)$ and $(x2 \leq 0.745)$ and $(x1 > -0.816)$ then class: 1|2
Rule 7: if $(x1 \leq -0.428)$ and $(x2 > 0.613)$ and $(x1 \leq -0.763)$ and $(x2 > 0.687)$ and $(x1 > -1.144)$ and $(x2 > 0.745)$ and $(x2 \leq 0.8)$ and $(x1 \leq -0.915)$ then class: 2|2
Rule 8: if $(x1 \leq -0.428)$ and $(x2 > 0.613)$ and $(x1 \leq -0.763)$ and $(x2 > 0.687)$ and $(x1 > -1.144)$ and $(x2 > 0.745)$ and $(x2 \leq 0.8)$ and $(x1 > -0.915)$ then class: 1|2
Rule 9: if $(x1 \leq -0.428)$ and $(x2 > 0.613)$ and $(x1 \leq -0.763)$ and $(x2 > 0.687)$ and $(x1 > -1.144)$ and $(x2 > 0.745)$ and $(x2 > 0.8)$ then class: 1|2
Rule 10: if $(x1 \leq -0.428)$ and $(x2 > 0.613)$ and $(x1 > -0.763)$ then class: 1|2
Rule 11: if $(x1 > -0.428)$ and $(x1 \leq -0.372)$ and $(x2 \leq 1.014)$ then class: 1|1
Rule 12: if $(x1 > -0.428)$ and $(x1 \leq -0.372)$ and $(x2 > 1.014)$ then class: 1|2
Rule 13: if $(x1 > -0.428)$ and $(x1 > -0.372)$ then class: 1|1

For Compas, instances of the area marked yellow in Figure 4.2 are assigned in 93.0% to cluster No. 1 (see Figure 4.4). Cluster No. 1 of Compas is a larger cluster consisting of 26 nodes of the linear SOM and comprises instances of mainly male (59.7%) synthetic individuals with a median age of 25.0. The generated instances are 48.2% Caucasian and 29.9% African-American. 68.8% were charged with misdemeanor before scored by COMPAS. The number of priors ranges from -9.0 to 25 with 4.3 prior convictions on average. The mean number of juvenile misdemeanors and felonies is 0.29 and 0.05, respectively. On average, the individuals in this cluster have 1.48 juvenile convictions that are not misdemeanors or felonies.
Figure 4.8 shows the explainer for this cluster. Based on the basic statistical properties of cluster No. 1, the modifications of the number of prior convictions depending on age and the reduction from the charge degree from felony to misdemeanor for African-Americans with less than two juvenile felonies apply here. For cluster No. 1, we face decision differences for the instances that fulfill:

Rule 3: if $(juv\_other\_count \leq 1.045)$ and $(juv\_other\_count > 0.948)$ and $(priors\_count \leq 4.982)$ then class: Low|Medium
Rule 4: if $(juv\_other\_count \leq 1.045)$ and $(juv\_other\_count > 0.948)$ and $(priors\_count > 4.982)$ then class: High|Medium
Rule 5: if $(juv\_other\_count > 1.045)$ and $(priors\_count \leq 1.014)$ and $(sex \leq 0.5)$ then class: Medium|High
Rule 6: if $(juv\_other\_count > 1.045)$ and $(priors\_count \leq 1.014)$ and $(sex > 0.5)$ then class: Low|High

For instances in this cluster with one juvenile conviction that is not a misdemeanor or a felony (Rule 3 and Rule 4: $juv\_other\_count \leq 1.045$ and $juv\_other\_count > 0.948$) the explainer predicts decision differences of either $M_A|M_B$: Low|Medium for up to 4 prior convictions (Rule 3; $priors\_count \leq 4.982$) or High|Medium for more than four prior convictions (Rule 4; $priors\_count > 4.982$). Additionally, decision differences were detected for individuals with more than one juvenile conviction that are not misdemeanors or felonies ($juv\_other\_count > 1.045$) and at most one prior conviction ($priors\_count \leq 1.014$). In this case, the explainer predicts for male individuals the class combination Medium|High whereas for female individuals Low|High is predicted.
Neither of the detected decision differences in this cluster is directly associated with age smaller than 30 or larger than 60, however, the median age of synthetic individuals in



(a) Label color coding (b) U-Matrix

Figure 4.7: SOM-Projection of synthetic dataset of cluster No.1 of Compas

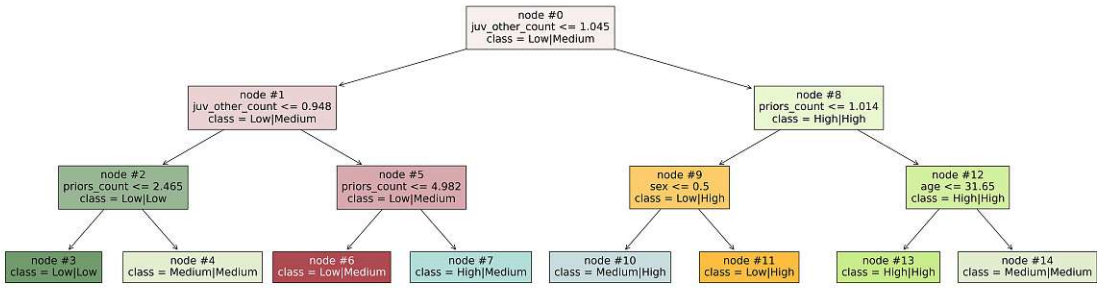


Figure 4.8: Explainer of Compas cluster No.1

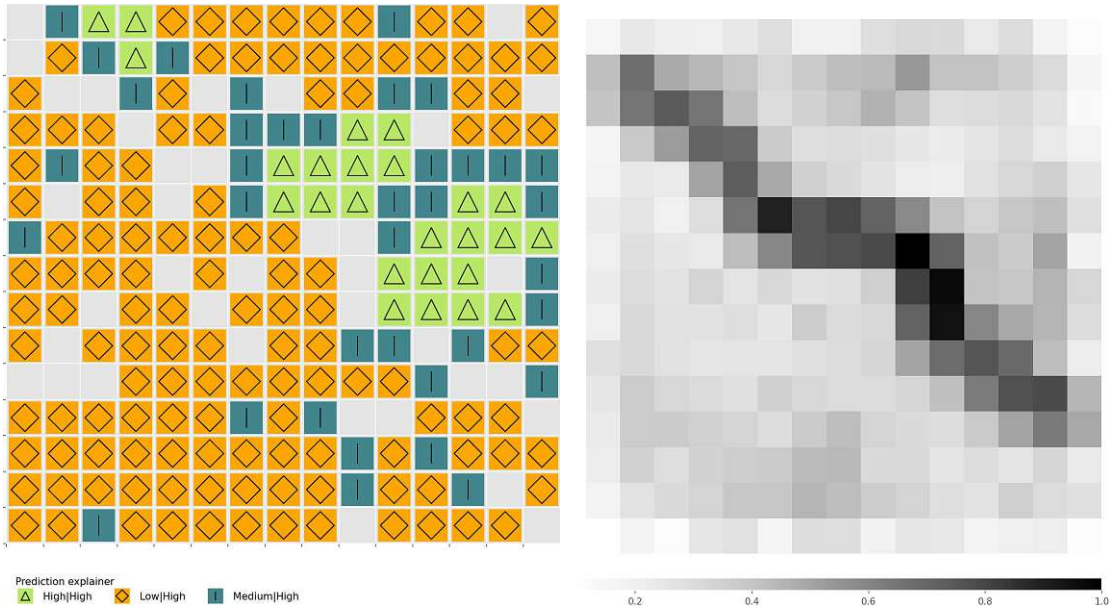
this cluster is 25 with an IQR of 24-28, and thus the second black box is affected by the age-dependent modifications of the number of prior convictions made. Additionally, all synthetic individuals have at most one juvenile felony, hence for 29.9% African-Americans the modifications of the training data have an impact on the second black box.

Figure 4.7 shows the SOM-Projection of the generated synthetic dataset of neighborhoods of cluster No. 2 which is in the SOM ordering adjacent to cluster No. 1. Cluster No. 2 is a small cluster of 8 nodes of the one-dimensional SOM. It also covers mainly male (83.9%) synthetic individuals with a similar median age of 24.0 years. 54.6% of them are Caucasian and 22.5% African-Americans. With an average of 2.2 the number of priors is slightly lower and ranges from -9.5 to 28. In this cluster, the mean number of juvenile misdemeanors and felonies is 0.05 and 0.06, respectively. On average, the individuals in

this cluster have 2.94 juvenile convictions that are not misdemeanors or felonies.
According to the component planes, the main differences between cluster No. 1 and cluster No. 2 are the race distribution and the higher number of juvenile convictions that are not misdemeanors or felonies for cluster No. 2 which can be confirmed based on the statistical properties of this cluster, using the generated synthetic neighborhood. Since the decision differences are ordered by the SOM, the changes from one cluster to the next adjacent cluster are rather smooth.
The explainer for cluster No. 2 is visualized in Figure 4.10. In this cluster, the explainer predicts three classes, all of them having black box B predicting class 'High'. The rules extracted from the explainer from this cluster for decision differences are the following.

Rule 1: if $(age \leq 23.401)$ and $(priors\_count \leq 0.065)$ and $(priors\_count \leq -1.162)$ then class: Low|High
Rule 2: if $(age \leq 23.401)$ and $(priors\_count \leq 0.065)$ and $(priors\_count > -1.162)$ then class: Medium|High
Rule 3: if $(age \leq 23.401)$ and $(priors\_count > 0.065)$ and $(juv\_fel\_count \leq 0.014)$ then class: Medium|High
Rule 5: if $(age > 23.401)$ and $(priors\_count \leq 1.366)$ and $(age \leq 27.646)$ then class: Low|High
Rule 6: if $(age > 23.401)$ and $(priors\_count \leq 1.366)$ and $(age > 27.646)$ then class: Low|High
Rule 7: if $(age > 23.401)$ and $(priors\_count > 1.366)$ and $(age \leq 35.426)$ then class: Medium|High
Rule 8: if $(age > 23.401)$ and $(priors\_count > 1.366)$ and $(age > 35.426)$ then class: Low|High

DiRo2C currently distinguishes only between continuous and categorical features. For the former, normalized euclidean distance is used in the fitness function to quantify the similarity with the specific instance to be explained, whereas for the latter, simple match distance is used. Count variables are therefore treated as continuous features in DiRo2C because of which also the synthetic neighborhood might contain negative continuous values as in the first rule of the explainer for cluster No. 2. In this cluster, we observe a similar age distribution as of cluster No. 1: about 60% of the synthetic individuals are 30 years or younger and thus the second black box model is affected by the imposed modification. For this cluster, we have decision differences for individuals younger than 24 years with no prior convictions (Rule 2: $age \leq 23.401$ and $priors\_count \leq 0.065$) and for individuals younger than 24 years with at least one prior conviction but no juvenile felony (Rule 3: $age \leq 23.401$ and $priors\_count > 0.065$ and $juv\_fel\_count \leq 0.014$). However, for individuals younger than 24 years with at least one prior conviction and in addition at least one juvenile felony both black boxes predict the class 'High'.
For individuals of 24 years or older with at most one prior conviction, the explainer for this cluster predicts the class combination $M_A|M_B$: Low|High (Rule 5 and Rule 6) whereas for individuals of 24-35 years with at least two prior convictions (Rule 7) the class combination Medium|High is predicted. Individuals older than 35 years with at least two prior convictions are predicted by the explainer to be scored Low|High by the black boxes.
Cluster No.7 which is in the 1-dimensional SOM-order further away from cluster No.1 and No.2 consists of 33 nodes. The gender distribution in this cluster is almost equal (53.1% male and 46.9% female). The median age is with 33.0 years higher compared to cluster No.1 and No.2. The number of prior convictions is 3.7 on average, 48.0% are Caucasian and 36.7% Asian. 89.5% were charged with felony before scored by COMPAS. Figure 4.11 shows a SOM-Projection of the generated neighborhoods for cluster No.7 and Figure 4.12 the explainer for this cluster. In contrast to cluster No. 1 and No. 2,



(a) Label color coding (b) U-Matrix

Figure 4.9: SOM-Projection of synthetic dataset of cluster No.2 of Compas

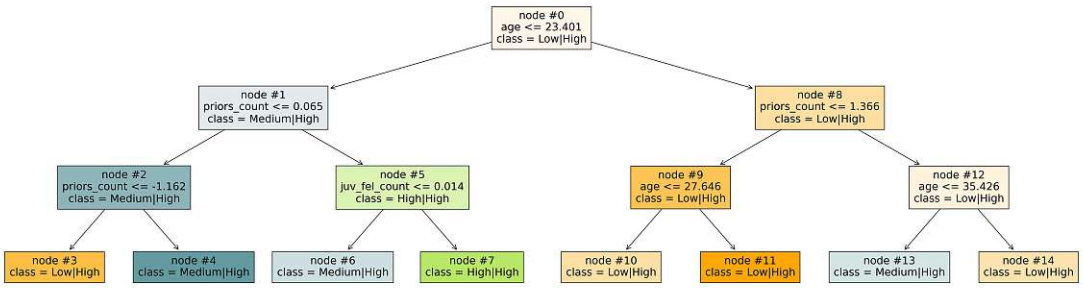


Figure 4.10: Explainer of Compas cluster No.2

the explainer mostly predicts 'Low' and 'Medium' for the second black box. The rules extracted from the explainer from this cluster for decision differences are the following.

Rule 1: if $(juv\_misd\_count \leq 0.659)$ and $(priors\_count \leq 6.805)$ and $(juv\_fel\_count \leq -0.422)$ then class: Low|Medium

Rule 3: if $(juv\_misd\_count \leq 0.659)$ and $(priors\_count > 6.805)$ and $(juv\_fel\_count \leq 0.152)$ then class: Medium|Low

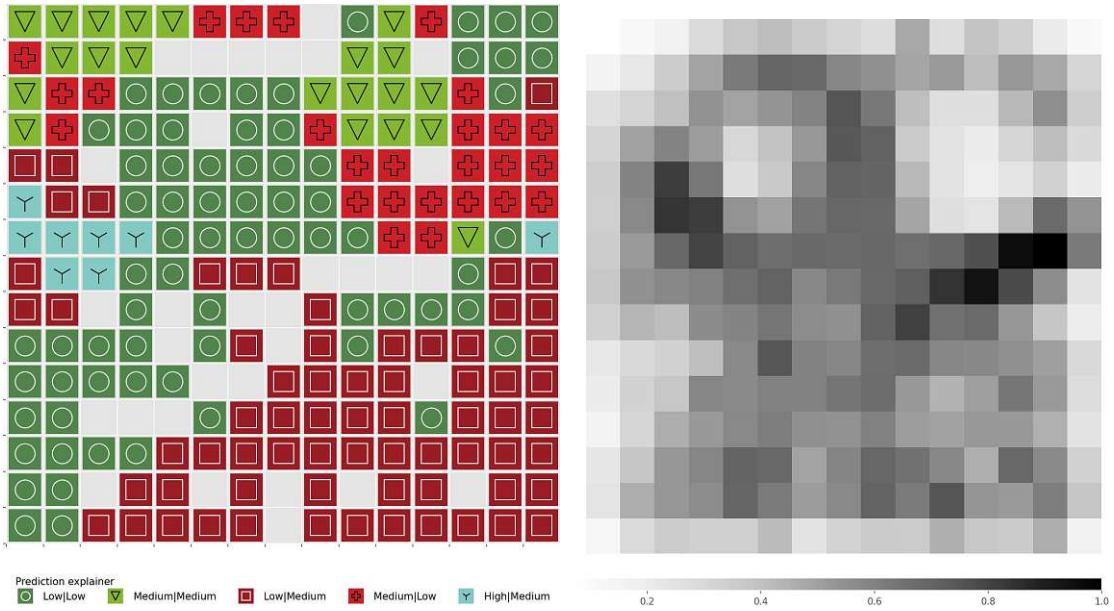
Rule 5: if $(juv\_misd\_count > 0.659)$ and $(priors\_count \leq 1.51)$ and $(age \leq 19.977)$ then class: High|Medium

Rule 6: if $(juv\_misd\_count > 0.659)$ and $(priors\_count \leq 1.51)$ and $(age > 19.977)$ then class: Low|Medium

Rule 7: if $(juv\_misd\_count > 0.659)$ and $(priors\_count > 1.51)$ and $(age \leq 27.164)$ then class: High|Medium

Rule 8: if $(juv\_misd\_count > 0.659)$ and $(priors\_count > 1.51)$ and $(age > 27.164)$ then class: Low|Medium

For synthetic individuals with less more than five prior convictions but no juvenile felonies



(a) Label color coding (b) U-Matrix

Figure 4.11: SOM-Projection of synthetic dataset of cluster No.7 of Compas

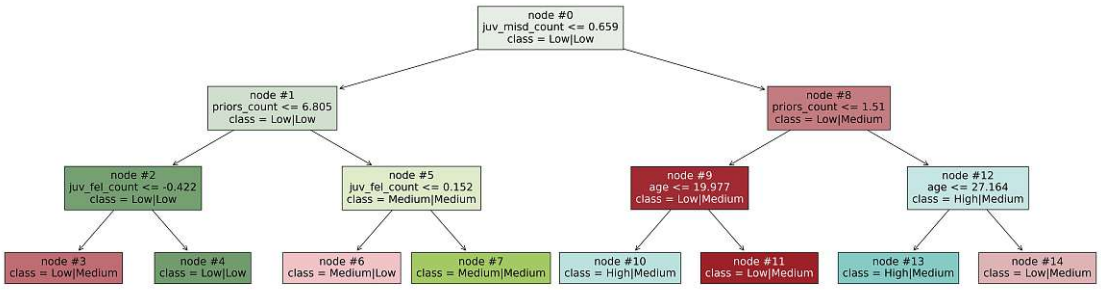


Figure 4.12: Explainer of Compas cluster No.7

($juv\_fel\_count \leq 0.152$) or juvenile misdemeanors ($juv\_misd\_count \leq 0.659$), the explainer predicts the decision difference class $M_A|M_B$: Medium|Low (Rule 3). For individuals with at least one juvenile misdemeanor ($juv\_misd\_count > 0.659$), we only observe decision differences and the specific class combination is dependent on age. For at most one prior conviction ($priors\_count \leq 1.51$), the cutoff at age from between $M_A|M_B$: High|Medium and Low|Medium is at 20, whereas for a higher number of prior convictions, the breakpoint is at age 27. Since the median age in this cluster is slightly higher than 30 and only 3% of the synthetic individuals are African-American, only part of the found and learned decision differences can be traced to the modifications made to train the second black box.

4.4 Discussion

We have proposed a sequence of steps to analyze two black boxes for decision differences by means of a real-world multi-class dataset. DiRo2C's neighborhood allows for a thoroughly and precise description of each cluster with only little dependency on the underlying training data. In combination with component planes of each feature of the projected dataset, an overview of the location of decision differences is provided. Subsequently, the decision differences can be analyzed on a cluster-level to obtain a more accurate and precise location. The steps proposed to communicate decision differences in a multi-class setting therefore allow both for a detailed instance-focused view and an overview.
Issues arising in this context mainly originate from issues of a multi-class setting in general. Since the number of possible decision difference classes increases as a quadratic function of the number of classes k of the black boxes, the number of possible decision difference classes are quickly far too many. This is accompanied by color coding problems and increased complexity of decision trees as explainers resulting in incomprehensible explanations without further actions. A straightforward way to cope with too many combinations of black box class predictions is to hierarchically structure the classes. If one is interested solely in decision differences, the number of classes for the explainer is reduced by $k - 1$ by the combination of classes without decision differences. Additionally, one can analyze decision differences in a one-against-all manner.
The structure and order of the explanations allow for a smooth and continuous analysis of each cluster in the process since the decision differences of two adjacent clusters are more similar to each other than the decision differences of two distant clusters.
The SOM as dimensionality reduction technique for visualization purposes was chosen because of its topology preserving properties and its numerous visualization possibilities [34]. Nevertheless, the SOM can be replaced with any other projection method such as linear Principle Component Analysis or nonlinear Sammon's projection. Siedlecki et al. [30] present a review of such mapping techniques. Regardless of the chosen mapping technique, the target audience should have basic knowledge about the technique used to follow the visualizations.

CHAPTER 5

Summary and Conclusion

We have proposed four bottom-up approaches to derive a global explanation by utilization of local explanation generation techniques. RQ 1 and RQ 2, focusing only on possible improvements of the performance to detect decision differences, were assessed in Chapter 3. The proposed approaches make use of DiRo2C's synthetic neighborhood generation process and were evaluated against a Baseline explainer. Random, class-stratified and cluster-stratified sampling (Approaches 1-3) combine multiple synthetic neighborhoods to a global dataset, serving as training data for a global explanation model and structured sampling (Approach 4) structures data into clusters and a local explanation is provided for each cluster. Experimental evaluations of the proposed approaches have demonstrated a positive effect of advanced and strategic selection of instances on precision and recall, resembling the focus on instances with decision differences between the black boxes as can be seen in Table 3.9 for Benchmark datasets and Table 3.1 for the running examples. The Baseline works without data synthesis strategies and could be outperformed for Compas in macro-averaged recall by 2.5 percentage points with cluster-stratified sampling of instances (see Table 3.9). Additionally, the performance was improved even further by increasing the number of sampled instances such that for Compas class-stratified sampling outperformed the Baseline for all performance metrics by 6 percentage points on average, answering RQ 1.

The structured combination of pre-pruned explainer (Approach 4) outperforms the Baseline in terms of macro-averaged recall for Bank-Marketing and additionally in terms of macro-averaged precision for Compas, and thus the structured sampling approach provides easily comprehensible, but still accurate explanations (see Figure 3.19). Therefore, with respect to RQ 2, the structured combination of local explanations can indeed outperform a single global explanation by up to 11 percentage points for macro-averaged precision and by up to 3.4 percentage points for macro-averaged recall while maintaining low complexity of the explainer. The strength of the proposed approaches lies in the

partial decoupling from training data.

Due to the linear ordering, the data structuring approach (Approach 4) provides valuable insights and can additionally be used to communicate the explanations of decision differences in a comprehensible way. In Chapter 4 with reference to RQ 3, we have proposed a sequence of steps to analyse two black boxes for decision differences. From the first question about the existance of decision differences through a global overview of the location of the decision differences to a detailed explanation on a cluster-level basis, the proposed steps allow both for a detailed instance-focused view as well as an overview.

5.1 Future Work

This section gives an outlook on a number of extensions as future work. An advancement of the proposed approaches to derive global explanations by utilization of local explanation generation techniques is a combination of class-stratified sampling (Approach 2) and structured sampling (Approach 4). A strategic selection of instances within SOM-clusters for which a neighborhood is generated could increase performance since instances are not randomly sampled from the cluster, but it is ensured that at least one instance for every decision difference label is selected from each cluster, and therefore the problem of highly unbalanced decision differences in the data at cluster-level is tackled.
Another extension is a user-case study to empirically evaluate the effectiveness of the presented process of communication of decision differences. Additionally, a solid description of each cluster and the data points for which a certain explanation applies might also be added to better understand the clustering. A first, starting point is the description of clusters using min/max-values of features. Moreover, the feature with the lowest variance within a SOM-cluster can be used as an description.

Acronyms

BMU best-matching unit. 17–19, 28, 31, 32, 39

CV Cross Validation. 30, 31, 36

DiRo2C Difference Recognition of 2 Classifiers. ix, xi, xiii, 1, 2, 5–8, 10–12, 14, 19, 20, 22, 24, 25, 27, 28, 30–32, 43, 48, 51, 54, 55

GLocalX GLObal to loCAL eXplainer. 16

I Indicator function, $I_x := \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{otherwise} \end{cases}$. 10, 18

IQR Interquartile Range. 50

LORE Local Rule-Based Explanations. 1, 7–10

PCA Principle Component Analysis. 54

SD Standard Deviation. 34, 35, 39, 42, 43

SOM Self-organizing map. 2, 16–19, 24, 25, 27, 28, 31, 32, 36–40, 46, 49–54, 56

SVM Support Vector Machine. 6, 30

Bibliography

[1] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018. doi: 10.1109/ACCESS.2018.2870052.

[2] Or Biran and Courtenay Cotton. Explanation and justification in machine learning : A survey. In *IJCAI-17 Workshop on Explainable AI (XAI)*, 2017.

[3] Thomas Bäck, David T. Fogel, and Zbigniew Michalewicz. *Evolutionary Computation 1 (Basic Algorithms and Operators)*. Institute of Physics Publishing, Bristol, 01 2000.

[4] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8), 2019. ISSN 2079-9292. doi: 10.3390/electronics8080832. URL `https://www.mdpi.com/2079-9292/8/8/832`.

[5] Jonathan N. Crook, David B. Edelman, and Lyn C. Thomas. Recent developments in consumer credit risk assessment. *European Journal of Operational Research*, 183 (3):1447–1465, 2007. ISSN 0377-2217.

[6] Sofie Goethals, David Martens, and Theodoros Evgeniou. The non-linear nature of the cost of comprehensibility. *Journal of Big Data*, 9(1), March 2022. doi: 10.1186/s40537-022-00579-2.

[7] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. Local rule-based explanations of black box decision systems. *ArXiv*, abs/1805.10820, 2018.

[8] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5), aug 2018. ISSN 0360-0300. doi: 10.1145/3236009.

[9] Simon S. Haykin. *Neural networks and learning machines*. Pearson Education, Upper Saddle River, NJ, third edition, 2009.

[10] Fei Jiang, Yong Jiang, Hui Zhi, Yi Dong, Hao Li, Sufeng Ma, Yilong Wang, Qiang Dong, Haipeng Shen, and Yongjun Wang. Artificial intelligence in healthcare: past, present and future. *Stroke and Vascular Neurology*, 2(4):230–243, 2017. ISSN 2059-8688.

[11] Cengiz Kahraman. *Computational Intelligence Systems in Industrial Engineering: With Recent Theory and Applications*, volume 6 of *Atlantis Computational Intelligence Systems*. Springer-Verlag, Paris, 2012. ISBN 9491216767.

[12] Teuvo Kohonen. *Self-organization and associative memory*. Springer series in information sciences. Springer, Berlin [u.a.], 3. ed.. edition, 1989. ISBN 3540513876.

[13] Teuvo Kohonen. *Self-Organizing Maps*. Springer Berlin Heidelberg, 2001. doi: 10.1007/978-3-642-56927-2.

[14] Daniel T. Larose and Chantal D. Larose. *Discovering knowledge in data : an introduction to data mining*. Wiley Series on Methods and Applications in Data Mining. IEEE, Hoboken, New Jersey, 2nd ed. edition, 2014. ISBN 1118873572.

[15] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the compas recidivism algorithm. `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`, May 2016. Online; accessed 27 February 2022.

[16] Erwin Loh. Medicine and the rise of the robots: a qualitative review of recent advances of artificial intelligence in health. *BMJ Leader*, 2(2):59–63, 2018.

[17] Oded Maimon and Lior Rokach, editors. *Data Mining and Knowledge Discovery Handbook*. Springer-Verlag, 2005. doi: 10.1007/b107408. URL `https://doi.org/10.1007%2Fb107408`.

[18] Rudolf Mayer, Taha Abdel Aziz, and Andreas Rauber. Visualising class distribution on self-organising maps. In *Artificial Neural Networks – ICANN 2007*, Lecture Notes in Computer Science, pages 359–368, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 3540746935.

[19] George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97, March 1956. doi: 10.1037/h0043158. URL `https://doi.org/10.1037/h0043158`.

[20] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2018.07.007.

[21] Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014. ISSN 0167-9236. doi: https://doi.org/10.1016/j.dss.2014.03.001.

[22] Cecilia Panigutti, Riccardo Guidotti, Anna Monreale, and Dino Pedreschi. *Explaining Multi-label Black-Box Classifiers for Health Applications*, pages 97–110. Springer International Publishing, Cham, 2020. ISBN 978-3-030-24409-5.

[23] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[24] Dino Pedreschi, Fosca Giannotti, Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, and Franco Turini. Meaningful explanations of black box ai decision systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 9780–9784, Jul. 2019. doi: 10.1609/aaai.v33i01.33019780. URL https://ojs.aaai.org/index.php/AAAI/article/view/5050.

[25] Rok Piltaver, Mitja Luštrek, Matjaž Gams, and Sanda Martinčić-Ipšić. What makes classification trees comprehensible? *Expert Systems with Applications*, 62:333–346, 2016. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2016.06.009. URL https://www.sciencedirect.com/science/article/pii/S0957417416302901.

[26] Ravi Ponmalai and Chandrika Kamath. Self-organizing maps and their applications to data analysis. Technical Report LLNL-TR-791165, Lawerence Livermore National Laboratory, 09 2019. URL https://www.osti.gov/servlets/purl/1566795.

[27] Marco T. Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778.

[28] Marco T. Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[29] Mattia Setzu, Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. Glocalx - from local to global explanations of black box ai models. *Artificial Intelligence*, 294:103457, 2021. ISSN 0004-3702. doi: 10.1016/j.artint.2021.103457.

[30] Wojciech Siedlecki, Kinga Siedlecka, and Jack Sklansky. An overview of mapping techniques for exploratory pattern analysis. *Pattern Recognition*, 21(5):411–429, 1988. ISSN 0031-3203. doi: 10.1016/0031-3203(88)90001-5.

[31] Andreas Staufer. Recognition of differences between two binary black-box classifiers to create explanations using model-agnostic methods. Diploma thesis, Technische Universität Wien, 2021.

[32] Alfred Ultsch. U*matrix: a tool to visualize clusters in high dimensional data. Technical Report 36, Department of Computer Science University of Marburg, December 2003.

[33] Alfred Ultsch and H. Peter Siemon. Kohonen's self organizing feature maps for exploratory data analysis. In Bernard Widrow and Bernard Angeniol, editors, *Proceedings of the International Neural Network Conference (INNC-90), Paris, France, July 9–13, 1990 1. Dordrecht, Netherlands*, volume 1, pages 305–308, Dordrecht, Netherlands, 1990. Kluwer Academic Press. URL http://www.uni-marburg.de/fb12/datenbionik/pdf/pubs/1990/UltschSiemon90.

[34] Juha Vesanto. Som-based data visualization methods. *Intelligent Data Analysis*, 3(2):111–126, 1999. ISSN 1088-467X. doi: https://doi.org/10.1016/S1088-467X(99)00013-X. URL https://www.sciencedirect.com/science/article/pii/S1088467X9900013X.

[35] Juha Vesanto and Esa Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 11(3):586–600, 2000. doi: 10.1109/72.846731.

[36] Giuseppe Vettigli. Minisom: minimalistic and numpy-based implementation of the self organizing map, 2018. URL https://github.com/JustGlowing/minisom/.

[37] Geoffrey I. Webb, Roy Hyde, Hong Cao, Hai Long Nguyen, and Francois Petitjean. Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30(4):964–994, Apr 2016. ISSN 1573-756X.