UNIVERSITY OF GRONINGEN

MASTER'S THESIS

Can Prediction Explanations Be Trusted?

On the Evaluation of Interpretable Machine Learning Methods

Author:
Denny DIEPGROND

Internal supervisor:
prof. dr. Bart VERHEIJ

External supervisor:
dr. Evert HAASDIJK

A thesis submitted in fulfillment of the requirements
for the degree of Master of Science in Artificial Intelligence

in the

Multi-Agent Systems Group
Department of Artificial Intelligence

May 18, 2020

university of groningen

faculty of science and engineering

"What is vital is to make anything about AI explainable, fair, secure and with lineage, meaning that anyone could very simply see how any application of AI developed and why."

Ginni Rometty—CEO of IBM during her opening address of CES 2019

UNIVERSITY OF GRONINGEN

Abstract

Faculty of Science and Engineering

Department of Artificial Intelligence

Master of Science in Artificial Intelligence

Can Prediction Explanations Be Trusted?
On the Evaluation of Interpretable Machine Learning Methods

by Denny DIEPGROND

The growing complexity and opaqueness of machine learning algorithms have raised interest in methods that combine their good performance with some form of transparency. Explanations of machine learning models try to help a user decide whether to trust its predictions. While performance metrics for machine learning models have been well established and are important to today's AI successes, performance metrics for explanations of model predictions are as yet less well investigated. The question we want to answer in this work goes a step further than trusting predictions: Can we trust the explanations of machine learning predictions?

The contribution of this work is twofold. First a theoretical framework to evaluate methods for interpretable machine learning is established based on regulatory requirements and social explanation theory. Post-hoc interpretation methods that use feature importance indications have appeared as a promising approach to establish interpretability while preserving model performance. Model-agnostic variants of these post-hoc explanation methods accept any black box model as input, making a general framework.

Secondly the framework is applied (in part quantitatively and in part qualitatively) to evaluate two state-of-the-art explanation methods (LIME and Kernel SHAP) on synthetic datasets with known explanatory structure. The predefined data distributions have served as a ground truth that made objective evaluation of the explanations possible. Moreover, an intuitive assessment has been proposed that serves as a first step towards a general evaluation of explanation models that are in production. Our results suggest that evaluating explanations of model predictions should become as integrated in the field of machine learning as evaluating performance of the models themselves. The analysis and methods in this work are a step in that direction.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	**A**rtificial **I**ntelligence
XAI	e**X**plainable **A**rtificial **I**ntelligence
LR	**L**ogistic **R**egression
DT	**D**ecision **T**ree
NN	**N**eural **N**etwork
RF	**R**andom **F**orest
GDPR	**G**eneral **D**ata **P**rotection **R**egulation
MLP	**M**ulti**L**ayer **P**erceptron

Chapter 1

Introduction

Artificial intelligence has been adopted by mass culture with the technology now being deployed in numerous industries. Decision-making is being automated in order to generalize, speed up and improve the effectiveness of processes. Personalized pricing, job recruitment tools and credit scoring models are applications of artificial intelligence (AI) that demonstrate its omnipresence and influence on the lives of many individuals. Now that banks and governmental organizations use risk assessment- and automated decision-making tools, the challenges these systems pose to their users in terms of fairness and transparency are in the spotlight of public attention. Data-driven algorithms can be sensitive to discrimination and biases, perhaps injected by human prejudice, which exemplifies that caution is warranted when considering the output of such systems.

An example of an algorithm that sparked controversy is the judicial risk assessment scoring method that is used in US courtrooms to predict recidivism in different stages of the criminal justice system. A popularized investigation by ProPublica (Angwin et al., 2016) concluded that the COMPAS [1] tool used by courts in the United States (US) to inform judges during sentencing, was biased against African American defendants. Even though race was not a feature used for classification, employment status and zip codes could act as proxies for race due to high correlation with minority groups. Despite these critics, the algorithm surpassed a major legal challenge when it was ruled admissible by the Wisconsin Supreme Court in 2016 (Kirkpatrick, 2017). However, the court specified that an algorithmic risk score cannot be the determinant factor in legal rulings. The analysis of ProPublica endured criticism from scientists for the fairness criteria it had used (Chouldechova, 2017). The debate this case caused is relevant and underlines the lack of transparency in the decision-making process of certain AI models.

Controversial algorithmic decision-making is not limited to the US only. A network of private investigators published a report in which they mapped out the changing landscape of automated-decision making in different countries in the European Union (AlgorithmWatch, 2019). In the Netherlands, the Ministry of Social Affairs and Employment uses the System Risk Indication system, a big data analysis system intended to detect welfare fraud by linking government data sources. The indicators in the risk model are unknown to the public and the Ministry refuses to make them public in order to prevent the system from being manipulated by criminals. This has led to questions by members of the Dutch House of Parliament to the Minister

[1] The Correctional Offender Management Profiling for Alternative Sanctions tool claims to predict the risk of a defendant committing another crime. It is an algorithm that uses the answers to a 137-item questionnaire.

of Legal Protection. A group of civil right initiatives even started legal proceedings against its use (Huisman, 2019).

A statement on the social process of black-boxing by philosopher Bruno Latour from 1999 reflects the achievements in the field of AI from the last decades. According to him, black-boxing is "the way scientific and technical work is made invisible by its own success. When a machine runs efficiently, when a matter of fact is settled, one need focus only on its inputs and outputs and not on its internal complexity. Thus, paradoxically, the more science and technology succeed, the more opaque and obscure they become." (Latour, 1999). However, even though AI models can improve the productivity and objectivity of consequential decisions dramatically, the opaqueness of the steps between input and output can limit the practical use of AI systems. Their black-box nature prompts complex ethical and scientific questions that stand in need of answering.

1.1 Artificial Intelligence

In the early days of AI, the picture of algorithmic problem solving that arose was a general-purpose search procedure (McCorduck, 2004). In this paradigm, an AI agent would usually follow the same basic algorithm to achieve its goals. Actions would be produced step by step, backtracking whenever reaching a deadlock. Early AI systems were mostly implemented in made-up micro worlds and the way they strung together rudimentary reasoning steps to find solutions did hardly scale up to complex environments. Besides that, the algorithms knew nothing about their environments or the objects they dealt with. They reached their goals only by exploiting basic syntactic manipulations.

In response to this first surge of AI, domain-specific knowledge and rules were integrated in AI systems. This injection of expert knowledge led to useful applications in specific real-world domains. Researchers started to believe that intelligence might be grounded in how one deals with different types of knowledge. Knowledge engineering and representation became major points of focus within the field of AI. Knowledge-based AI systems—built on logic and taxonomic hierarchies—are well-defined which facilitates understanding of their internal reasoning. The key limitation of these rule-based systems however, is inflexibility regarding tasks that involve uncertainty. Rules alone do not suffice for defining complex non-linear tasks like visual object detection.

The limitations of this second surge of AI led to a paradigm shift towards data-driven statistical AI models (Russell and Norvig, 2009). Methods from statistics and probability theory facilitated the return of neural networks and the invention of other data-driven methods like random forests (Breiman, 2001a). These systems have become commercially successful as they have been applied to newly available digitized information for applications like speech recognition and credit card default prediction. This caused a reorganization of machine learning as a separate field within AI. With the availability of large datasets and increase of computational power came a more practical approach to AI. In this paradigm, given enough data for learning methods to extract useful information, explicitly expressing all the knowledge a system needs is no longer needed (Halevy, Norvig, and Pereira, 2009).

The main drawback of the increased complexity of modern statistical AI methods is the opacity of their reasoning. While the reasoning of classical rule-based AI methods is well-defined, statistical methods like deep neural networks and tree ensembles (random forests) do not exhibit comparable transparency. This problem is disclosed in studies that use adversarial attacks in which minor perturbations of input instances that are imperceptible to humans, lead to strong deviations in the output of a model (Papernot et al., 2017). Accordingly, as AI systems have become more complex and opaque, interest in methods that combine the performance of state-of-the-art machine learning with some form of transparency has raised.

1.2 Right to Explanation

The interest in more transparent AI systems cannot be seen in isolation from increased regulatory pressure. The spike in data breach complaints in the first months after the introduction of the European General Data Protection Regulation (GDPR)[2], demonstrates the current public interest in the increased possibilities for contesting data abuse (Hern, 2018). The scope is international and since the GDPR applies to controllers and processors that use data of citizens in the European Union (EU), regardless of whether the processing takes place in the EU or not, the regulatory pressure pertains to the global majority of data processing organizations. Article 22 is the key section of the GDPR when it comes to AI since it includes provisions on automated individual decision-making[3]. Decisions that fall under the protection of this Article are based solely on automated processing and should produce legal effects or similarly significantly affect an individual, as stated in Article 22(1). In Recital 71 of the GDPR [4], automatic refusal of an online credit application and e-recruiting practices without any human intervention are given as examples of such decisions.

Although not undisputed, Goodman and Flaxman (2017) argue that the new legislation tries to tackle the lack of transparency in automated decision-making with a so called right to explanation as implicitly described in Article 22 of the GDPR. The acknowledgement of a right to explanation received quite some academic backlash as researchers claimed by means of restrictive interpretation, that the GDPR merely implies a right to be informed (Wachter, Mittelstadt, and Floridi, 2017). This right to be informed would only inquire insights into the functionality of the system before a decision is being made, as opposed to an explanation of the specifics of an individual decision after it has been made. Important to this discussion is Article 22(3), as this provision refers to the requirement for the data controller to "implement suitable measures to safeguard the rights, freedoms and legitimate interests of the data subject, at least the right to obtain human intervention on the part of the controller, to express his or her point of view and to contest the decision". What these mentioned measures are and when they are deemed suitable is what sparks most legal and academic debates. An added layer of understanding for the Article is provided by GDPR Recital 71. This is the only place in the GDPR where a right to explanation

[2] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC [2016] OJ L 119

[3] GDPR Article 22 on the official website of EU Law (visited on 16-05-2020): https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1532348683434&uri=CELEX:02016R0679-20160504

[4] https://gdpr-info.eu/recitals/no-71/ (visited on 16-05-2020)

is mentioned. It declares that "in any case, such processing should be subject to suitable safeguards, which should include specific information to the data subject and the right to obtain an explanation of the decision reached after such assessment and to challenge the decision". Currently, the majority of academic researchers accept the existence of a right to some form of explanation and the discussion has moved on towards deciding what the form of these explanations should be.

Further guidelines for interpreting the right to an explanation are provided by the Article 29 Data Protection Working Party (WP29), now known as the European Data Protection Board (EDPB)[5]. This is an independent European advisory body with a representative from the data protection authority of each EU Member State. Their guidelines on automated individual decision-making and profiling provide a clarification of the relevant provisions in the GDPR, which helps assessing the weight of the alleged right to explanation. The WP29 recognizes the complexity of explaining intricate model decisions in comprehensible terms, as pointed out by Edwards and Veale (2017). Despite that, the WP29 guidelines assure that this complexity can never be used as an excuse to provide information to the users that is inadequate for them to invoke their rights. According to the WP29, a complete report of the algorithm is not needed. An adequate explanation supplies the user with enough information to understand the reasons behind the decision. Additionally, the WP29 states that explanations should allow a user to act upon the decision—which is to say spot errors and contest the decision. Goodman and Flaxman (2017) and the WP29 align in their argumentation that these requirements are met by explanations that provide information about the features that are taken into account for the decision and about "their respective weight on an aggregate level".

The GDPR is not unique in enforcing this kind of transparency upon data-processors. A practical example is given by a Dutch case in which a black box model for property valuation was brought to court. In the Netherlands, the property value (WOZ-value in Dutch) is essential for determining property tax and next to that, it also serves as a basis for several other municipal taxes and charges. Even though the property values are publicly available for perusal, the calculation of the value was—until recently—based on an opaque model. After a seven-year procedure, The Dutch High Council has stated that in case of valuation, municipalities should provide insights into the relevant factors behind the procedure of property value determination[6]. In a broader sense, the High Council has stated that under the Dutch General Administrative Law Act (Awb), algorithmic decisions should be accompanied by information that makes it possible to verify them. This court decision aligns with the WP29 interpretation of the GDPR and the general trend in the regulatory landscape.

Although the ambiguity and lack of explicitness in some of its articles make the legal existence and feasibility of some rights mentioned in the GDPR concerning automated decision-making controversial (Wachter, Mittelstadt, and Floridi, 2017), the perspective of (Goodman and Flaxman, 2017) will be taken in the remainder of this thesis. They argue that even though the legislation poses big challenges to the industry, it should first and foremost be perceived as an opportunity and motivation for computer scientists to take the lead in creating transparent and interpretable systems that serve to prevent biases and discrimination by enabling interpretability.

[5] WP29 - Guidelines on Automated individual decision-making and Profiling for the purposes of Regulation 2016/679 (As last revised and adopted on 6 February 2018)

[6] Hoge Raad 17 August 2018, ECLI:NL:HR:2018:1316

1.3 Explainable Artificial Intelligence

As part of the AI strategy, the High-Level Expert Group on Artificial Intelligence—an independent expert group set up by the European Commission— prepared a document containing Ethics Guidelines for Trustworthy Artificial Intelligence (AI)[7]. This document is intended to set out a common goal when it comes to the development of trustworthy AI systems. The common goal of creating AI systems whose actions and solutions can be understood by humans led to the development of a complete sub-field of AI called Explainable Artificial Intelligence (XAI) (Gunning, 2017). XAI systems aim not only to optimize a task in terms of efficiency and accuracy but additionally, provide explanations as to why it made a decision. Since real-world deployment often differs from controlled training settings, current metrics that validate model function before deployment might not be indicative of the final goal of a model. Users or in some cases human experts should be able to use model explanations to grasp the rationale behind a model's predictions.

With the ambition of creating explainable AI, came the advent of a popular class of models that focuses on interpretability. Machine learning research on interpretability is defined as the discipline of facilitating users with cues to comprehend what a model did and why it did so. These cues can take many forms, depending on the type of data and machine learning model one is working with. Examples of interpretable models that have shown promising results include visual cues to reveal what deep learning models are focusing on in an image and surrogate models that simplify the internal process of opaque systems.

1.3.1 Merits of Explainability

Decision-making models based on data-driven learning methods are associated with opaque internal reasoning processes. The problem associated with this paradigm is that users are inclined to distrust predictions that are not accompanied by any explanation (Edwards and Veale, 2017). When no insight is given into the internal reasoning process, the possibility of the reasoning being biased cannot be ruled out. With that being the case, the trust of users and model owners in these systems can be restored by and built upon explanations accompanying predictions.

Throughout the history of computing science, the algorithm has been the key focus of study. Recent work in AI however suggests that for a lot of problems, it makes more sense to be concerned with the data that is fed to the algorithms instead of which algorithm to use (Russell and Norvig, 2009). The main cause of this shift is the increasing availability of very large datasets. The problem is also known as 'data fundamentalism' (Crawford, 2013), which covers the notion that big datasets are storehouses that yield valid and objective truths, if only we can draw them out using machine learning algorithms.

Next to gaining user trust, explanations can help to give a more nuanced view on this notion and act as a safeguard for algorithmic fairness and for identifying biases in large datasets. Ethical and societal issues regarding data-driven decision-making have been the ground for recent regulatory pressure that serves as an important

[7] Ethics Guidelines for Trustworthy AI on the website of the European Commission (visited on 15-05-2020): `https://ec.europa.eu/futurium/en/ai-alliance-consultation/guidelines`

force behind the push for explainability. Regulatory pressure on its own is not always enough for ensuring fairness and an ethical foundation. The problem called *ethics washings* demonstrates this. Companies establish ethics boards and have principles in place, that actually mean nothing other than getting them out of regulatory trouble (Sharkey, 2019). As Sharkey (2019) points out, it is a very high level thing to 'be fair' or 'be just' so it needs to come down to the application. Without systems in place that allow us to evaluate the fairness of model decisions, it is hard to move from these principles to practice. It is not the goal of interpretable models to ensure fairness, as it is an ethical concept rather than a statistical one, but the move towards increased interpretability does provide stakeholders with arguments that can aid these societal discussions.

Most regulatory pressure aims to provide interpretability from the user perspective. Important to note though, is that interpretable models also have great benefits for model owners and designers. In automated processes like credit and insurance risk assessments, instances of interest are flagged, after which a human controller has to validate the assessment. Predictions that are accompanied by an explanation can give insight into whether the model is functioning as intended, which would speed up the validation process.

To summarize shortly, interpretability in the decision-making process of models has four main motivations:

Trust It fosters trust of users and regulators and thereby promotes model adoption;

Learning The rationale behind a decision allows us to discover new patterns in the data and learn more about the problem domain;

Ethics It helps in making ethical assessments of models, also in terms of regulatory compliance; and

Validation It makes debugging and validating models easier.

1.3.2 Challenges for Explainable Artificial Intelligence

For black box machine learning models it is not always clear why they arrived at a certain prediction or classification. The problem is not that model engineers do not know what their model consists of. They should know exactly the amount of layers their model comprises, parameters that are tuned, how the error is propagated back through the model during training and what activation function it uses. The problem here is that for simple models, it is evident how each input contributes to the output. For more complex models, no matter from which model family, this input contribution is less clear because of numerous re-combinations of the input variables. Evidently, the bigger and more complex models get, the more challenging it becomes to bring back these non-linear interactions to comprehensible terms.

Another problem is what Breiman (2001b) calls the multiplicity of data models. With this he describes the phenomenon that given a particular dataset, there are multiple good models with different internal architectures that map the input variables to their corresponding predictions. If the error surface of a given problem does not have an obvious global minimum, it is very possible that similar models with close predictions would lead to different explanations because they both generalize from a somewhat different underlying model. From an interpretable model perspective, this could lead to explanations of questionable quality and consistency. The influence this has on the robustness and causality of interpretable models should be made clear for these methods to become widely accepted and trusted.

Interpretation involves a trade-off between being truthful to the underlying model (fidelity) and the explanation being comprehensible to the user. This arises from the fact that the complexity of model decisions is hard to capture in comprehensible terms. If the goal is solely to gain user trust, problems of putting implicit human cognitive bias in the interpretable model arise (Herman, 2017). This is something the XAI community should account for by providing objective measures of interpretable model requisites.

The explanations a forensic analyst needs in order to validate a fraud recognition model are very different from the explanations that should accompany a model that classifies CT-scans, for a physical therapist to make an informed decision. This kaleidoscopic character of interpretable models fragments the field of XAI. Put differently, there might be as much approaches needed to interpretability as there are to machine learning. The combination of the problems mentioned in this section reflects the approximate nature of interpretable models. Objectively measuring and testing for interpretable model requisites would be the next step in the direction of reaching the goals of XAI and interpretable models specifically.

1.4 Goal of this Study

While performance metrics for machine learning models have been established, the same cannot be said for performance metrics for explanations of model predictions. Explanations of machine learning models try to help a user decide whether to trust their predictions. Before we start relying on the explanations accompanying automated decisions, the quality and trustworthiness of explanations have to be assessed and established. The question we want to answer in this study therefore goes a step further than trusting predictions: Can we trust the explanations of machine learning predictions?

An important goal of this study is to give an overview of what has been done in the field of XAI. We will establish basic terminology and use this to structure the work that has been done along a number of key dimensions. This work sets out to combine the requirements of automated decision-making models found in the regulatory landscape sketched in Section 1.2 with cognitive explanation theory and the work that has been done in the field of XAI. We believe that this combination adds a layer of practical applicability to the field. Important criteria for evaluating explanations will be established using this framework. These criteria result in different metrics that we will be used to evaluate state-of-the-art explanations methods. This study will establish the first steps at querying and testing these methods with respect to the most important criteria of the explanations to the original models. Since human evaluation of explanations is not always feasible, the focus will be on computational methods for evaluating explanations, both quantitatively and qualitatively.

1.5 Outline

In the introductory chapter of this thesis, it has been outlined that the problems in AI have led to regulatory pressure to add a layer of transparency to machine learning models. The combination of these two factors resulted in the development of the field of explainable AI. In Part I of this thesis, the machine learning methods that will be used to make predictions in the remainder of the study will be introduced. In the next chapter (3), interpretable machine learning attempts will be taxonomized to give an overview of the field and to categorize the different approaches. That chapter will be started of with establishing basic terminology and reviewing cognitive theory of explanation. Additionally, the legislative requirements and possibilities provided by state-of-the-art interpretable models will be combined to establish requisites for the evaluation of explanations that will be used later on in the analysis. Part II of the thesis provides details on how the evaluation of model explanations will be addressed and which methods will be employed to answer the questions posed in the introductory chapters. Chapter 4 describes in turn the details of the state-of-the-art explanation methods that will be used in this study and how they propose to solve interpretability problems in machine learning. The last chapter of this part (5) will describe how the synthetic datasets with known explanatory structure have been generated and the methodology used to apply the evaluation framework. In Part III, the results of the analyses will be described and structured before we will discuss them in Part IV.

Summary Chapter 1

1.1 As AI systems have become more complex and opaque, interest in methods that combine the performance of state-of-the-art machine learning with some form of transparency has raised

1.2 Regulators have started to actively call for more transparency in automated decision-making models. Key requirements that adequate decision explanations should meet for GDPR compliance are that the user should be able to understand the reasons behind an automated decision and that the user should be able to act upon the decision—which is to say spot errors or contest the decision.

1.3 The problems in AI resulted in regulatory pressure to add a layer of transparency to machine learning models. This has led to the development of the field of explainable AI.

1.4 While performance metrics for machine learning models have been well established and are important to today's AI successes, performance metrics for explanations of model predictions are as yet less well investigated.

1.4 An evaluation framework for prediction explanations will be drafted based on a combination of regulatory requirements, explanation theory and research on interpretable machine learning. This framework will be tested using datasets with predefined distributions and a known explanatory structure.

Part I

Theoretical Framework

Chapter 2

Background

The lack of transparency in state-of-the-art AI applications and the ensuing call for explainability, stem from multiple sources. The development of explainable artificial intelligence (XAI) and the direction in which the field is moving, are guided by the regulatory landscape that forces data processing instances to be compliant with legislation on the one hand. On the other hand, practical constraints and limitations—that sometimes conflict with regulatory requirements—are given by the mathematical framework upon with machine learning is built. The learning algorithms that are the basis of machine learning will be discussed in this chapter as they provide necessary background information for the remainder of the study.

2.1 Machine Learning and Statistical Premise

Machine learning is a subfield of AI that focuses on computer programs that use statistical models to perform tasks without explicit specifying how (Samuel, 1959). Fundamentally, machine learning models use example data or past experience to optimize a specified performance criterion by using patterns and inference. One of the problems that machine learning can solve is to perform a task for which there is no human expertise or for which humans cannot explain how they solve the task—like recognizing a particular word in speech. Automating such a task is challenging partly because we can not explicitly provide the system with all the knowledge it needs. By providing a model with many sample words however, it can learn to recognize the patterns in the sounds that make up particular words.

The range of machine learning algorithms can broadly be dissected into three different approaches: Supervised learning, unsupervised learning and reinforcement learning (Alpaydin, 2009). Supervised learning methods learn from a dataset with instances consisting of features that describe its attributes. The goal of the system is to learn a generalized mapping that can predict the correct output value for an unseen instance, given only the features of the instance as input. When the algorithms learns this mapping, the target outputs acts as supervisors that guide the learning of the mapping. When target output variables are not available in the dataset, unsupervised learning methods can be used to find regularities in the input. This form of self-organization can be very useful to discover patterns in an unlabeled dataset. When the output of a system is a sequence of actions and a single action is not what is most important, reinforcement learning algorithms can be used to assess the quality of an action sequence. They do this by learning—through trial and error—which sequences increase the chance of reaching the goal. This study focuses on supervised learning, since most problems related to the lack of transparency in machine learning models stem from biases in collected labeled training data.

2.1.1 Supervised Learning

The learning algorithms in the supervised learning paradigm, use patterns in historical data to create a mapping from input to output. When the task of the model is to estimate the relationship between the features and a continuous output variable, the model will perform a regression analysis. For classification problems on the other hand, the target output is a categorical label and the task consists of deciding to which category the instance belongs based on its feature values. The latter will be the focus of this study. More specifically, binary classification problems will be used in which the task consists of classifying an instance as either one of two binary classes. From the definition of supervised learning, it becomes apparent that the data that is used for training a model is vital to its predictive capabilities. It is believed that there are patterns in observed data, however we do not know the particular process that generated these patterns and use the learning algorithms to extract them. Essentially, the supervised learning algorithm is a computer program that optimizes the parameters of a model using training data without using explicit knowledge. This makes this family of methods very susceptible to having data biases encountered in the learning phase, resonate through its decision-making process.

Computer science and statistics play a role in machine learning by providing efficient algorithms to solve the parameter optimization problem. A wide range of models and training algorithms is available, varying in complexity. Algorithms can be categorized along multiple axis, in accordance with different defining characteristics. The first of which is the trade-off between bias and variance. The bias of a model describes the simplifying assumptions it makes about the target function it is trying to learn. Models with high bias generally learn faster and are more interpretable than models with low bias. They are however, less flexible and the simplifying assumptions hinder the learning of complex functions. Variance on the other hand describes the degree to which a model is dependent on a specific set of training data. Since models are trained on only a sample of the data, they should not be too dependent on the specifics of the selected set of training data. Ideally, the model learns the underlying distribution of the complete dataset independent of which part of the data is used for training. The parameters of a model with high variance and the function it learns are therefore influenced more strongly by the specific training set than for a model with low variance. This sensitivity to overfitting of the mapping function is especially harmful when the training data is not a representative sample of the underlying distribution and when there is a lot of noise in the data. In machine learning, there is always a trade-off between the bias and variance since increasing the bias of a model will decrease the variance and vice versa.

The complexity of a dataset can be characterized by the amount of interactions and non-linear relationships between features. Models that can accurately learn more complex mapping functions tend to use a more opaque reasoning process. This is what sparks the accuracy-explainability discussion in the field of AI. For this study, comparisons will be made between models of different levels of complexity with respect to the target functions they can estimate. One way to visualize the complexity of a model is to look at the decision boundary it draws for a binary classification problem. Since a binary classifier has to always label an instance as either one of the possible categories, there is a decision boundary that separates both classes. The decision boundary is the part of the problem space where classification is an ambiguous decision. In this study, four different supervised machine learning models will be trained on datasets with different underlying distributions for a binary classification task. These models are: Logistic regression, decision trees, random forest models and neural networks and they represent different levels of opacity and complexity. In Figure 2.1, the decision boundaries of all four models for a binary classification task are visualized. The dataset contains instances drawn from a standard normal distribution with $\mu = 0$ and $\sigma = 1$ that are labeled using the following evaluation: $y > {x_0}^2 - x_1$. In this dataset, an instance is thus of class 1 (grey dots) when ${x_0}^2 - x_1 > 0$ and of class 0 (green dots) when ${x_0}^2 - x_1 <= 0$ with 1% noise. The red and blue contours make clear for which input values the model predictions are of class 0 and 1 respectively. It is shown that the logistic regression model is not able to learn this mapping whilst the other models do relatively well, with the remark that the decision tree model seems more sensitive to noise in the training data.

2.2 Learning Algorithms

Most methods that are used in machine learning nowadays, have a statistical foundation that transforms an algorithm into a prediction model, using data in a supervised learning context. This section serves as a general introduction of the range of algorithms that are going to be used in the remainder of this thesis to develop prediction models. The range of algorithms consists of varying complexity as well as a broad array of possible decision boundaries.

2.2.1 Logistic Regression

Logistic regression is a machine learning model that uses a logistic function to model the relation between one or more independent variables and a binary dependent variable. It is a linear method, but the predictions are transformed using the logistic function. In this function, the logarithm of the odds (log-odds) for the instance belonging to one of the classes is a linear combination of the independent variables, which can on their own both be binary or continuous variables. Even though the log-odds is a linear combination of the input variables, logistic regression is different from linear regression. Logistic regression is used over linear regression when the output variable it is trying to predict is of categorical nature, which corresponds to the classification problems used in this study. The probability of the instance belonging to one of the classes is a value between zero and one since the function that converts the log-odds to a probability value is the logistic function:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

Because a linear relationship is assumed between the input variables and the log-odds of instances belonging to class 1 in logistic regression classification, the log-odds of the instance being part of class $y = 1$ with two input variables x_0 and x_1 are

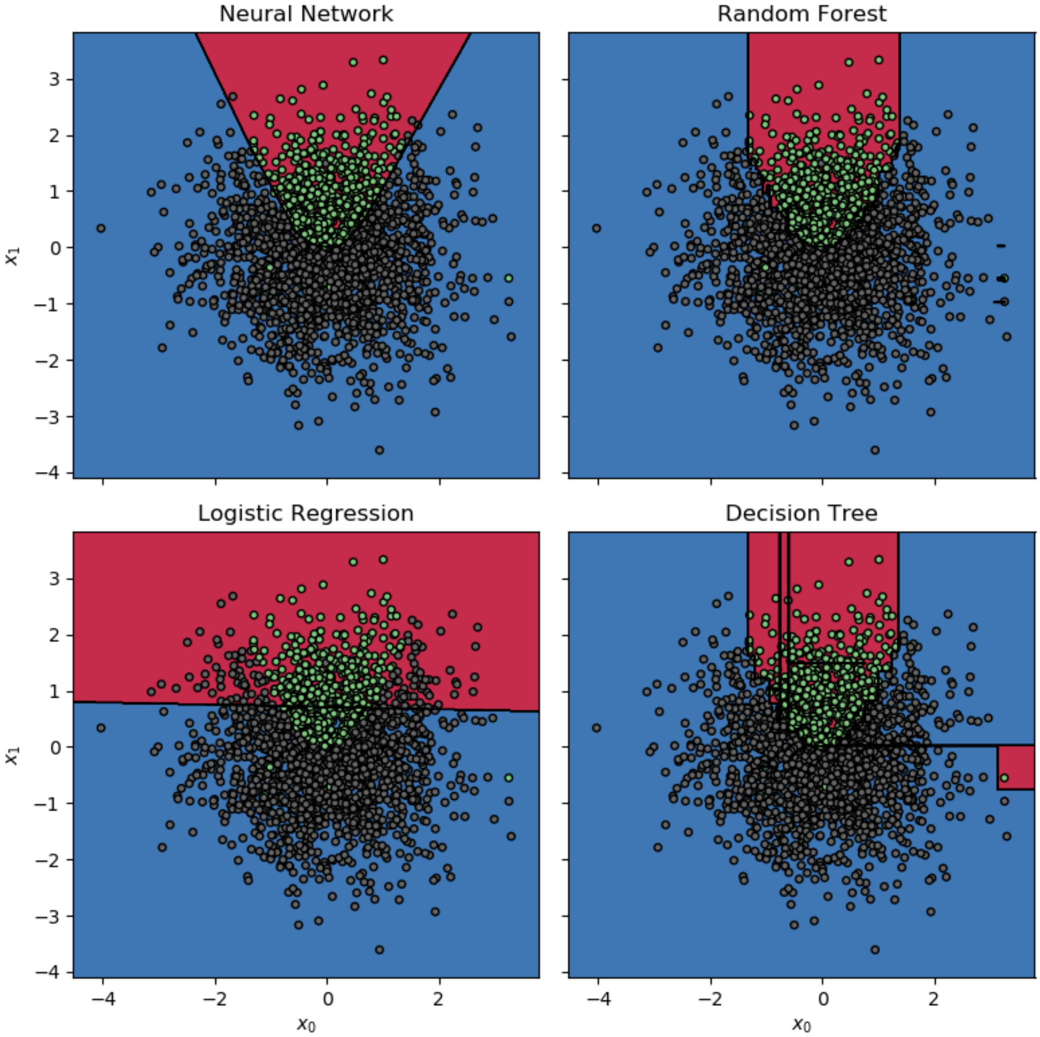


FIGURE 2.1: The learned decision boundaries of the models trained on a binary dataset labeled using the following function: $y = {x_0}^2 - x_1$, where an instance is of class 1 (grey) when $y > 0$ and of class 0 (green) when $y <= 0$ with 1% noise. Red and blue contours denote areas for which the model predictions are of class 0 and 1 respectively.

described by the following equation:

$$\log \frac{p}{1-p} = \alpha_b + \alpha_0 x_0 + \alpha_1 x_1 \tag{2.2}$$

Since the natural logarithm is used in the standard implementation of the model, the odds can be described by:

$$\frac{p}{1-p} = e^{\alpha_b + \alpha_0 x_0 + \alpha_1 x_1} \tag{2.3}$$

and the probability p of the instance belonging to class $y = 1$ is then:

$$p = \frac{1}{1 + e^{-(\alpha_b + \alpha_0 x_0 + \alpha_1 x_1)}} \tag{2.4}$$

The coefficients x_i of the linear combination can be fitted by optimization methods using the labeled training data. The solver used in this study is maximum likelihood estimation using L2 regularization (Malouf, 2002), also known as ridge regression. This learning algorithm tries to find values for the coefficients that minimize the error in the probabilities predicted by the model compared to those in the labeled data. Because of the linear assumptions made by the logistic regression model, it has high bias and low variance which makes it more sensitive to underfitting complex distributions. In the bottom-left part of Figure 2.1, it is shown that the linear assumptions hinder the logistic regression model from learning the non-linear relationship in the $y > {x_0}^2 - x_1$ dataset. However, the resulting linear decision model is very comprehensible for human users. Roughly, users only need to evaluate whether the value of x_1 is above 0.9. In this synthetic situation this might not mean much but let us assume the model is used for credit approval and x_1 represents a gender-related attribute. In that case, the fairness of the automated decision-making tool is highly questionable and the comprehensible nature of the machine learning model led to this realization.

2.2.2 Decision Trees

Decision tree learning comprises of generating a tree structure that can be used to predict the class output of a presented instance (Breiman et al., 1984). The tree is constructed by using labeled training data for learning the supposed relations between the input features and class labels. Classification trees are directed tree structures much like flow-charts starting from a root node with branches representing the outcome of a test on a feature of the instance. Leaf nodes at the bottom of the tree represent class labels and internal nodes between the root node and the leaf nodes represent further tests on features of the instance. Paths from the root node to a leaf node thus can be seen as combinations of features that lead to an instance being classified as part of a specific binary class. The graph-like structure and the step-by-step tests on the features makes the resulting model interpretable for most humans.

The algorithms for creating a decision tree usually work top-down, starting with the root node using the whole dataset intended for training. The data is splitted into subsets by the value of a feature that best splits the data. The best split is usually defined as the split that leads to the maximum level of similarity in terms of classes within the different subsets. The split will lead to two new nodes and both subsets are then splitted in the same way. This splitting is done recursively until a node contains only one class or until the level of similarity in class presence cannot be increased. In both cases, the node becomes a leaf node. This greedy top-down inference of decision trees is the most used method for generating decision trees from a set of labeled training data. The level of similarity used for splitting is based on a predefined metric. In this study, Gini impurity will be used to define the quality of a split. The decision tree algorithm makes few assumptions about the target function and can therefore be seen as a model with low bias and high variance. These quantities can be controlled with model options like a maximum tree depth but these will not be used in this study. Because decision trees map all the instances of the training data to the tree structure, they are not always good at generalizing and therefore prone to overfitting. This can be seen in the bottom-right part of Figure 2.1. The decision tree model is sensitive to the noise in the training data, which leads to problems when new unseen instances will be presented to the model.

Gini Impurity

When splitting a node, the Gini impurity metric provides information on how often a random instance from the set of training data would be incorrectly classified if it was randomly classified according to the distribution of labels in the subset. For a binary classification problem it is calculated by the following formula:

$$G = \sum_{i=0}^{1} p(i) * (1 - p(i)) \tag{2.5}$$

where $p(i)$ is the probability of randomly selecting an instance of class i in a set. Gini impurity is weighted for the subsets, according to their size. This weighted impurity is subtracted from the impurity of the node to be splitted which results in Gini gain value for a specific split. Maximizing this gain value corresponds to choosing the best split for a node.

2.2.3 Random Forest

Random forests are a more complex model type that is built on decision tree learning. Models of this type are meta-estimators that fit multiple decision tree classifiers on subsets of the data over which it averages to increase predictive capabilities and to control overfitting (Breiman, 2001a). The effectiveness of this approach is shown in the top-right part of Figure 2.1 when compared to the bottom-right part. The random forest model seems to be less sensitive to noise in the training data. Random forests are based on the concept of bootstrap aggregated decision trees—better known as bagging. Bagging is an ensemble method that repeatedly selects a random sample of fixed size from the training data with replacement and fits decision trees to these samples as described in Section 2.2.2. For classification, the individual decision trees in the tree ensemble all vote for a class and the final classification is based on the majority vote principle. The intuition behind bagging is to decrease the variance of the model without increasing the bias too much. A single decision tree can be very sensitive to noise and overfitting but the average of many trees, given they are not correlated, should limit this. Sampling by bootstrapping is performed to avoid too much correlation between multiple trees, as opposed to creating multiple decision trees on the same training dataset.

Random forests consist of one additional characteristic compared to this general bagging scheme. During the learning phase, every split is based on a random subset of the features. The addition of feature bagging—which is a form of random subspace method—to the standard sample bagging, further reduces correlation between the different trees in the ensemble. After all, if a couple of features are strong predictors for the output class, they will be present in many trees. Reducing this correlation leads to stronger predictive capabilities under many circumstances (Ho, 2002). In this study, random forest models consisting of 100 decision tree estimators will be used for classification. These individual trees use Gini impurity metrics to fit the training samples (see Section 2.2.2).

2.2.4 Neural Networks

Neural networks make up a class of supervised learning algorithms that is loosely based on the human brain. The analogy drawn, is of an information network that consists of several nodes connected by weighted edges, similar to how signals are transmitted from neuron to neuron via synapses in the brain. Over the years, the focus of the field of neural networks deviated from replicating biology to performing specific machine learning tasks. Therefore, the directed and weighted graphs that are called neural networks, should no longer be seen as an attempt to remain true to their biological counterparts. Artificial neural networks are famously successful at mapping complex non-linear relationships between features and are used in almost any type of task in machine learning (Alpaydin, 2009). In the top-left part of Figure 2.1, the smooth decision boundary of a neural network model trained on a dataset with a non-linear feature relation is visualized.

Artificial neurons

The basic unit of computation in neural networks is a single node or neuron. Such a single node receives input information from another node or from an external source and outputs a signal that can be picked up by other connected nodes. A visual representation of a single node can be found in Figure 2.2. In implementations of neural networks, each of the input signals x_i that come from other nodes or external sources are real numbers with an assigned weight w_i, reflecting the importance with respect to other inputs. The weighted inputs are summed and the activation of the node is calculated based on feeding this weighted sum of inputs to an activation function f, which will be explained later in this section. One of the inputs to the node is a bias input x_b with value one. This bias node allows for a trainable constant value, permitting the activation function to be linearly shifted.

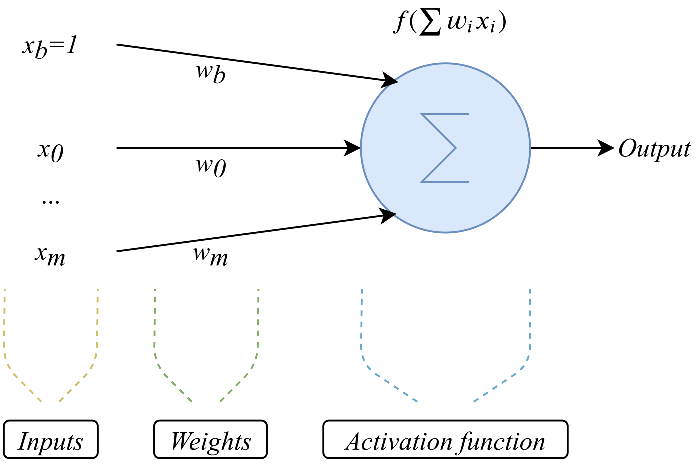


FIGURE 2.2: Visual representation of a single node, the basic unit of computation in a neural network. The output is calculated by feeding the weighted sum of inputs to an activation function.

Multi-Layer Perceptron

The most basic binary classification algorithm that uses the concept of a single neuron with a linear activation function is the perceptron (Rosenblatt, 1958). Single nodes can however be linked together in different network configurations, leading to various learning behaviors. In a neural network, multiple nodes are usually ordered in layers. The first layer is the input layer that consists of vectors providing information to the network. In this study, the input information will consist of numerical vectors with tabular data yet neural networks also accept encoded visual and textual data as input. The last layer is an output layer that translates network computations to real-life variables, actions or decisions. In between these two layers, the network can consist of one or more hidden layers in multiple different configurations that do computations based on the activation function.

A standard type of neural network with at least one hidden layer is the multi-layer perceptron (MLP). A visual representation of this type of network can be found in Figure 2.3. MLP networks are examples of feedforward neural network, in which the information flows only in one direction—from the input- to the output layer. Recurrent neural networks (RNN) on the other hand, have connections in both directions making it possible for the signals to traverse the hidden layers multiple times, allowing for temporal dynamic behavior. In this study however, a relatively straightforward MLP with one hidden layer containing 100 nodes will be used for the classification tasks. Even in this configuration with only hidden layer, MLP models are universal function approximators (Cybenko, 1989).

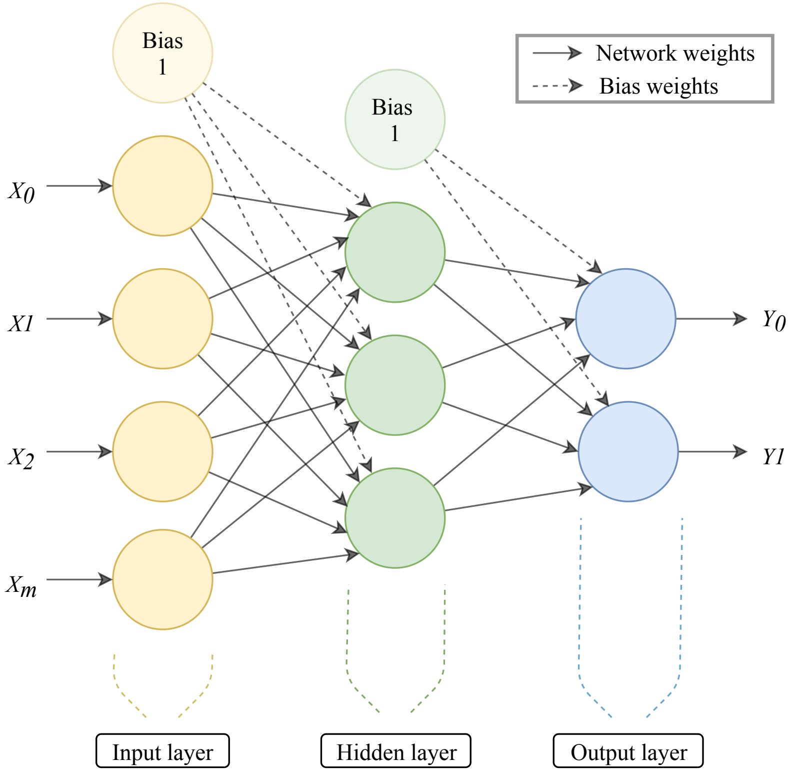


FIGURE 2.3: Visual representation of a multi-layer perceptron with one hidden layer in between the input layer and the output layer.

Activation Functions

An important factor in the ability to approximate non-linear functions is the activation function that the nodes in the network use. The activation function of a node is an abstraction of the action potential in the cell body of biological neurons. As can be seen in Figure 2.2, the activation function f computes an activation values based on the weighted sum u of all the inputs to a node:

$$u = \sum_{i=1}^{m} w_i x_i \tag{2.6}$$

When initial attempts at creating neural networks used step functions to compute node activation, it became clear that only linear classification problems could be solved by networks of a reasonable size. Linear activation functions lead to similar problems with the additional issue of unstable convergence caused by nodes that are on a favorable path in the network, as the activation values in linear functions are not bounded nor normalizable. These problems can be solved by using non-linear activation functions, which allow networks to map non-linear problems using a reasonable amounts of nodes. This ability to learn non-linear relationships is what makes networks of these nodes useful for problems that involve complex relations between features in the data.

Sigmoid A widely used activation function is the sigmoid logistic function. This asymptotic function translates the weighted sum of inputs u to a value in the range between zero and one. The function looks as follows:

$$f(u) = \frac{1}{1 + e^{-u}} \tag{2.7}$$

This activation function represents a smooth transition between a neuron being relatively inactive ($f(u) \approx 0$) and a neuron firing ($f(u) \approx 1$) with the additional property of it being differentiable, which is a requirement for most popular neural network learning methods. Because of its asymptotic character, the function always returns a non-zero value. This results in dense representations of the activation, which can be problematic especially for large networks of nodes. Additionally, it is not computationally efficient to depend on expensive exponential operations when learning the optimal weight values.

ReLU An alternative is the Rectified Linear Unit function (ReLU), which is currently the most popular activation function (Ramachandran, Zoph, and Le, 2017):

$$f(u) = \max(0, u) \tag{2.8}$$

This function leads to inactive nodes when the weighted sum u of inputs to a node is lower than zero, which results in more sparse representation compared to networks using the sigmoid activation fuction. The derivative of the ReLU function is a constant value, zero for $u <= 0$ and one for $u > 0$, which facilitates efficient computation. The disadvantage of these inactive nodes in combination with a zero gradient is that they can sometimes get stuck in this inactive state (a problem known as dying ReLU) which prohibits these nodes from being involved in learning. It has been shown though, that networks using the ReLU activation function, allow for faster training and better convergence performance (Krizhevsky, Sutskever, and

Hinton, 2012). In this study, all neural networks used for classification use the ReLU activation function.

Training the Network

Another key element in the success of neural networks is the learning method that optimizes the weight values for a specific task. The most popular supervised learning algorithm to train MLP models is gradient descent optimization using backpropagation. Gradient descent is an algorithm to find the minimum of a function that reflects the error or loss of a model. For supervised learning, this function is an error measure between model predictions and target output from the labeled training data. In this study, a log-loss function—the most popular loss function for classification tasks (Janocha and Czarnecki, 2017)—will be used to represent the classification error of the model:

$$L(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \tag{2.9}$$

where y is the binary label of instance i and $p(y_i)$ is the predicted probability of the model that the instance belongs to class one. The function adds the predicted log probability that the instance is of class one to the loss when the correct label is indeed one and it adds the predicted log probability of the instance being of class zero ($\log(1 - p(y_i))$) to the loss when the correct label y_i is zero. This leads to low loss values when the model predictions are accurate and higher loss values when the predictions are worse.

Gradient descent is based on the idea of finding the minimum of this error function by changing the weights of the network proportional to the negative of the gradient of the function at a given point. This is why it is important for the activation function to be differentiable. Backpropagation is used to compute these gradients (Rumelhart, Hinton, and Williams, 1985). Although the error is calculated for the activation value of the output node, the other nodes in the network also influenced the classification. Mathematically, backpropagation distributes the error over the network with respect to all the different weights by computing partial derivatives of the error function.

After the weights in the network are initialized randomly, the output can be computed by a forward propagation step using the random weights and the activation functions of the nodes. Following that, the loss of the model is calculated as the average error between the target output and the predicted output using equation 2.9. This error is propagated back through the network using the backpropagation algorithm and the weights are updated by adding the partial derivates to the weights, controlled by a learning rate that defines the step size when minimizing the error. Standard batch gradient descent goes through all the samples in the training set before the weights are updated, which can be unfeasible computationally for large datasets. Stochastic gradient descent tries to tackle this problem by updating the weights after every input-output pair that is presented to the model. Mini-batch gradient descent is a mix of both batch- and stochastic gradient descent as it updates the weights using backpropagation after presenting a batch of training samples of a predefined size somewhere between one and the size of the training set. This effectively means changing the value of N in equation 2.9. In the learning phase, an epoch is defined as an iteration in which all samples from the training set have been presented to the model once. In this study, the Adam weight optimization algorithm will be used with a batch size of 200 and an initial learning rate of 0.001. This is a computationally efficient implementation of mini-batch gradient descent that uses the average of previous gradients along with an adaptive learning rate. The learning will stop when the loss has not significantly ($\Delta 0.0001$) decreased for ten consecutive iterations or when the maximum number of epochs (200) has been reached.

Summary Chapter 2

2.1 Practical constraints and limitations that sometimes conflict with regulatory requirements are given by the mathematical framework upon with machine learning is built.

2.1 This study focuses on supervised learning algorithms, since most problems related to the lack of transparency in machine learning models stem from biases in collected labeled training data.

2.2 For the analyses of this study, algorithms with a varying degree of transparency and complexity have been selected—*Logistic Regression*, *Decision Trees*, *Random Forests* and *Neural Networks*. These learning algorithms will be used to create predictive machine learning models.

Chapter 3

Interpretable Machine Learning

In the preceding chapters of this thesis, the field of XAI has been introduced by providing background information about its origins, the factors that make it challenging and the main drivers behind its recent developments. Interpretable machine learning, which is the focus of this study, has been introduced as a subfield of XAI similar to how machine learning is a subfield of AI. The fundamental concept behind interpretable machine learning is to understand how the models make decisions. Understanding provides ground for assessing these models in terms of fairness and accountability. This gives regulators and users confidence in case of real-world deployment in situations that affect personal lives, businesses and society.

Now that we have established the upshots of interpretable models and have explored where the demand comes from, we will progress towards more formal definitions of what interpretability is and how it can be both achieved and evaluated. In this chapter we will define what makes models interpretable, taxonomize the methods that have been devised to do so and outline how they can be evaluated.

3.1 Defining Interpretability by Explanations

There are many similar expressions that are used in the literature related to XAI. In this section, terminology will be established in order to systematically position the work. We accept interpretability as the denomination for comparing machine learning models on how transparent their internal reasoning is. Following that line of thought, by providing interpretability, models become more transparent and comprehensible for users. Similar to other studies (Miller, 2018), interpretability and explainability are sometimes used interchangeably in this work. Interpretability will be mostly used in the context of machine learning models while explainability will be used more often when a broader AI perspective is taken. A key notion on which the analysis in this chapter is built, is that explanations are the means to achieve interpretability. These explanations can take multiple forms, depending on factors such as the context in which a model is deployed and the type of data that is used by the model. For an overview of the different interpretability methods that generate explanations, please refer to Section 3.2.

The problem of making machine learning models interpretable has been approached from many different angles by communities from different scientific fields. This multitude of perspectives has led to different variations of even basic definitions for interpretability and explanations. Therefore, in this section, the basic terminology that will be used in the remainder of the thesis will be established.

3.1.1 Interpreting Machine Learning Models

For assessing interpretable machine learning methods, a basic terminology that the field can build upon is needed. One key term that lacks an agreed upon definition is *interpretability*. In the dictionary one can find that interpreting means to explain or tell the meaning of or present in understandable terms [1]. Whether terms are understandable, naturally depends on the context and on the intended user of an interpretable model.

We want to use interpretability as a comparative measure and incomplete specification of the problem prevents this. When evaluating models, one wants to be able to say that a model or machine learning technique is more interpretable than another (Lipton, 2018). A general definition will therefore be built on the two most dominant specifications found in the literature. From a social science perspective, Miller (2018) defines how people generate, present and evaluate explanations and uses this research to establish that a model is interpretable if it provides explanations for its predictions in a form humans can understand. He further describes interpretability as "the degree to which an observer can understand the cause of a decision". In the context of machine learning systems, Doshi-Velez and Kim (2017) define interpretability as "the ability to explain or to present in understandable terms to a human". From the combination of these two we can abstract two important insights that will be used in this thesis:

1. Model interpretability is established by providing explanations for decisions.
2. The given explanations should be understandable for a human.

In this study, interpretability refers to the degree of interpretability of the model itself and the decisions it facilitates. We stress that there is a distinction between this notion and the interpretability of the learning algorithm or parts thereof (see Section 2.1), on which the model is based. While the algorithm and mathematical functions behind a random forest model for example, might be interpretable to some users, these users might not be able to interpret the resulting model and decision-making process of a large tree ensemble as it would be too much to comprehend at once. It works the other way around as well. While the mathematical concepts and model fitting algorithm behind a logistic regression model can be too complex to grasp for some user without a mathematical background, the resulting model and the decision-making process (e.g. feature x_1 has a value higher than β) can still be interpretable.

3.1.2 Elements of Explanations

According to the first insight in Section 3.1.1, explanations are the mode to establish interpretable models. It is important to make a distinction however, between explanation and justification. The misapprehension that these words have identical meanings in the context of automated decisions is understandable, as explanations are often used to justify a decision. Yet in this study, explanations do not necessarily justify a decision. Explanations are merely tools to make a decision model interpretable, independent of the moral assessment of the decision being just or unjust.

[1] `https://www.merriam-webster.com/dictionary/interpret` - visited on 16-05-2020

It also works the other way around since decisions can be justified, explaining why they are right, without explaining the process that led to the decision.

In psychological theories about explanations, the difference is being made between *everyday* explanations and *scientific* explanations (Miller, 2018). The former refers to the explanations of why certain decisions occurred, as opposed to the latter, which reflects the explanations of more general relationships. The focus of many studies on everyday explanations is substantiated by research on human-machine interaction which observes that users lose trust when they do not understand automated decisions, instead of seeing it as an incentive to form more general theories about the decision-making process (Stubbs, Hinds, and Wettergreen, 2007). For automated decision-making in regulated industries, it is the balance between the two types of explanations that is important. Instead of focusing on the understanding between two humans in arguments, we are seeking explanations that hold generally in the form of regulation.

In an influential book on software development, Cooper (2004) uses the metaphor of *inmates running the asylum* to describe why computer programs are often poorly designed, from the perspective of lay-users. Software developers that are in charge of creating the technical aspects behind a program, are often also in charge of designing the user interface. This results in computer programs that are very intuitive from their perspective, but not so much for the target audience. Think of explanations as the interface between machine learning models and users and it becomes clear that looking at model interpretation only from an engineering perspective can lead to the same fallacy. This is why the metaphor of Cooper is used by Miller, Howe, and Sonenberg (2017) in the context of XAI. They emphasize that it is important to integrate scientific knowledge and a strong understanding of how humans use explanations into the field of interpretable machine learning because the engineers that understand the models might not be the right people to assess the adequacy of explanations for users that see their models as black boxes. Furthermore, De Graaf and Malle (2017) argue that because people attribute human traits and intentions to artificial agents—which is a known tendency in psychology—they expect model explanations to use the same conceptual framework as humans do when explaining their decisions. This strengthens the case for model explanations based on psychological and philosophical insights.

Causes

The traditional way of looking at explanations is to see them as the answer to a why-question. The essential step in this causal framework of explanation is the attribution of causes and events to ultimate decisions (Miller, Howe, and Sonenberg, 2017). In this study, the main goal of an explanation is to present the causal relation between model variables and the decision in such a way that the user can understand the causal relation between inputs and output. They should then be able to—solely based on this explanation—decide whether they accept or want to contest this causation. Explanations are thus presented as an assignment of causal responsibility, another well-known concept from psychology (Josephson and Josephson, 1996). Theories of explanation-based decision-making describe how people make important decisions in law, politics and everyday life from a psychological perspective (Hastie and Pennington, 2000). The distinctive assumption these explanation-based theories make, is that people construct an intermediate representation which is the basis of the final decision, instead of the complete set of evidence. This corresponds with the idea of insufficient cognitive load in the human brain for processing complete explanations (Miller, 2018). This is why we have become adept at appointing a small number of causes from a list of many, to be the explanation. We use several cognitive biases to do this. For automated decision-making, this selection should be made by regulated and generalized procedures. Attempts at creating these procedures will be outlined in Section 3.2. Miller (2018) argues that the high levels of abstraction on which machine learning models operate, make the chains of causes smaller and less cognitively demanding, especially when they can be represented visually. This has become a very common feature among the methods that aim to make machine learning models interpretable, as will be outlined in the remainder of this chapter.

Achieving an understanding of the causal relation that underlined the decision is not the only determinant of a successful explanation. The coherence of the explanation with prior beliefs and the strength of alternatives also seem to be major determinants of perceptions of strength of explanations and of confidence in the decision, when it comes to human decision-making (Pennington and Hastie, 1988). For the purpose of this study, we do not see coherence with prior beliefs as a determinant of strength for an explanation since the evaluation of explanations accompanying automated decisions should be as objective as possible. Considering alternative explanations and decision however, is a methodology that starts to receive more and more attention in the field of XAI.

Contrasts

An alternative method to prevent the cognitive load of complete explanations is to present an explanation in the form of a contrastive (or counterfactual) example. An explanation of this form, stating why decision y_0 was made instead of decision y_1, is similar to explanations that people use in everyday situations (Miller, 2018; Wachter, Mittelstadt, and Russell, 2017). Setting the decision off to an alternative can act as a shortcut for finding the most important causes among the set of possible explanatory causes. Proponents of contrastive explanations argue that the method is warranted by the contextual nature of explanations. Instead of depending on associations and causal relations, a user might only care about a subset of the possible explanations that is relevant in a specific context. We should also be careful though, with copying human decision-making and evaluation as the objective nature of automated decisions is perceived as one of its strengths.

3.2 Taxonomy of Interpretability Methods

Despite the attention interpretable machine learning models receive from different scientific communities, the results are still quite scattered and unorganized. In this section, a range of approaches, techniques and design paradigms will be taxonomized in order to provide some structure and to place the work performed in this study within the broader perspective of XAI. Interpretability methods are defined as algorithms or techniques that generate or provide explanations in order to increase the interpretability of a machine learning model. These methods can be classified along the following three main dimensions, delineating the structure of the remainder of this section:

1. **Approach to interpretability** - The technique that is used for the generation of the explanation.
2. **Type of explanation** - The chosen interpretable representation of the decision-making process.
3. **Scope of explanation** - Whether the explanation represents the decision-making process behind a single decision or of the complete model.

3.2.1 Approaches to Interpretability

Machine learning methods were being deployed long before the term XAI was coined and methods to get a better understanding of the models and the data they used, have existed for an equivalent amount of time. Exploratory techniques like clustering and visualization provide insights into the key features and relations in the dataset before a model is built upon them. Traditional model performance metrics on the other hand, enable model engineers to compare different models after they have been built on the data. The problem with traditional performance metrics is that there is often a mismatch between the formal objectives in supervised learning, which is predictive performance on a test set, and real world costs. It is also still a big step to go from classical visualization methods to rationale discovery. This can be attributed to the fact that even though visualization helps with the understanding of the data, it does not provide information about the causal chains that lead to a decision.

Traditional performance metrics might be a method that machine learning engineers are familiar with to obtain model information, but their theoretical and mathematical nature does not align with the goal of explanations to be understandable for lay-users. This is why a lot of research has been done on approaches to generate human-interpretable representations of model decisions. From the literature, we can distinguish the three different methods that are outlined in Figure 3.1. When approaching model interpretability, the position of interpretability in the model design process should be considered first. In essence, model engineers have two options. On the one hand, they can create an inherently interpretable model that provides transparency by design and on the other hand they can train a complex yet opaque model and try to explain it afterwards, which is known as post-hoc interpretability.

Inherently Interpretable Models

Machine learning models that use interpretable representations, can serve as an explanation on their own when they are used as the model behind an automated decision-making tool (visualized on the left side of Figure 3.1). Information on the interpretations that are usually considered interpretable can be found in Section 3.2.2. The design choice of avoiding black box models in the first place, usually means sacrificing predictive performance, especially when the data contains non-linear patterns. This has been discussed extensively in Section 2.1 when we compared different machine learning models and the decision boundaries they can learn in the supervised learning paradigm.

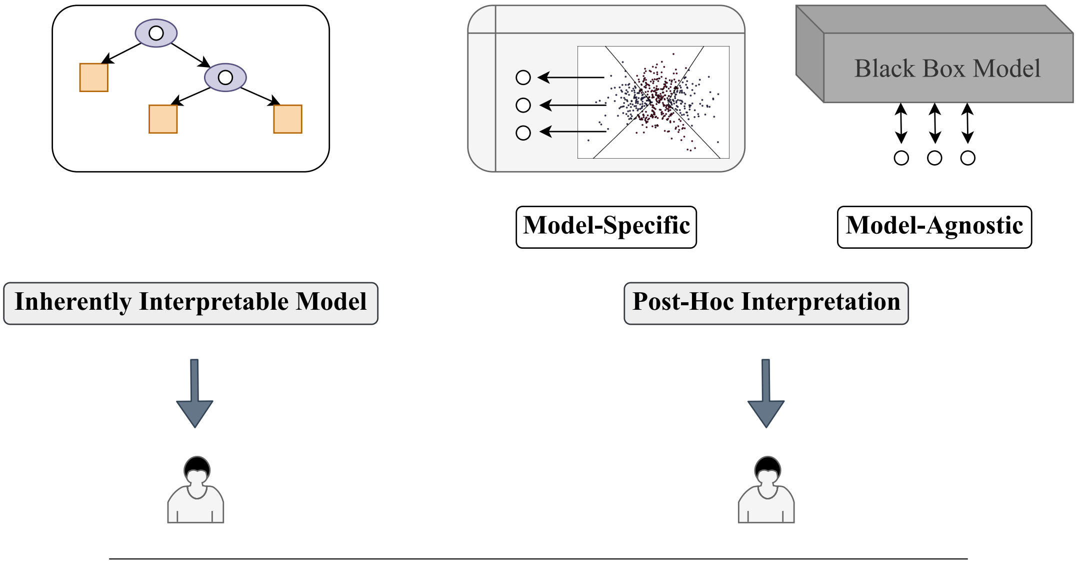


FIGURE 3.1: The different approaches for achieving model interpretability. The transparent decision-making process of inherently interpretable models can serve as an explanation on its own. When models are more complex, post-hoc interpretation methods are needed. Model-specific variants use parts of the model as an explanation while model-agnostic methods treat the model as a black box, querying the model to obtain information about the decision-making process.

Lipton (2018) argues that there are three aspects to transparency. On the first aspect, algorithmic transparency, decision trees and linear models score significantly better than neural networks and tree ensembles because we understand better what happens at the level of the learning algorithm. When interpretable models are adapted to reach high performance, they give up on another aspect of transparency, simulatability. For deep decision trees and high-dimensional linear models, it is not possible to go through every calculation within a reasonable time using the input data and the model parameters. The same holds for the third aspect, decomposability. Not all parts of an interpretable model admit an intuitive explanation since for more complex interpretable models, features are usually heavily engineered and pre-processed while neural networks often operate on raw or slightly processed features. For this reason, most techniques employ a form of post-hoc interpretation in order to preserve performance.

Post-Hoc Interpretation

The advantage of post-hoc interpretation is that it does not sacrifice model performance for the purpose of transparency. The disadvantage is that the explanation is to some extent always of an approximate nature. It can be argued that post-hoc interpretability is the approach to interpretability that is most similar to human explanations as the effects are related to the causes after the event. Many post-hoc interpretation techniques are designed to explain the decisions of a specific model type. Another option is to treat the model as a black box and query it to obtain information about the decision-making process. This model-agnostic approach works not only for one specific model type but for a wide range of machine learning models.

Model-Specific Interpretation Model-specific interpretation methods are tailored to explain the predictions of a specific model or a group of models. In general they possess decompositional characteristics (Guidotti et al., 2018) as they disintegrate a model into separate parts. These separate parts can on their own provide an explanation of the underlying decision-making process. This can be either on the parameter level or on a more granular level of aggregated inputs and outputs. Examples include visualizing what neural networks learn by looking at the activation in different layers of the network. A resulting explanation can then be represented in any of the ways that will be mentioned in Section 3.2.2. In the example of neural networks, the representation can be a mask over an input image but just as well a prototype representing the average member of an object class. The variable importance values that can be given for tree ensembles, based on an aggregated gain over all its individual trees, is also an example of a decompositional explanation. From these examples, it becomes evident that different machine learning models require different ways of decomposing them. This model-specific characteristic makes decompositional interpretation less flexible than model-agnostic interpretability methods.

Although requiring access to the model makes decompositional methods sensitive to disclosing trade secrets or intellectual property, it also gives them the advantage of using interpretable parts of the model for the explanation instead of approximating the complete model. The most popular branch of decompositional methods focuses on neural networks with many layers. Understanding and visualizing what these networks learn by decomposing the different layers and connections is also known as AI neuroscience (Samek, Wiegand, and Müller, 2017). For explanations of the complete model, decompositional methods often work with prototypes (Simonyan, Vedaldi, and Zisserman, 2013). For individual predictions, decompositional model inspection methods provide ways to visualize and inspect model internals without needing to understand the decision-making process as a whole, which is mainly helpful for validation purposes. Next to getting a better understanding of how a black box model works, these decompositional methods for individual predictions can also be used to reason about why the model made a certain prediction.

Model-Agnostic Interpretation Model-agnostic interpretability methods consider the input and output of a model while using the black box model as an oracle. The main benefit of this approach is that it works for any type of black box model. This gives them the added potential of being generalizable and scalable to new model types. In the literature, the approach of treating the model as an oracle is sometimes referred to as pedagogical interpretation (Guidotti et al., 2018). Many pedagogical methods employ a form of model extraction as they not only try to extract important features but a complete interpretable model from the opaque original model. These surrogate models usually require many queries to the black box model in order to extract informative characteristics, which can be disadvantageous when this is an expensive operation. Though usually, post-hoc interpretation will only be requested in cases of unexpected model decisions, which prevents the need for multiple real-time queries.

Model-agnostic interpretability methods come with a couple of important advantages. Since pedagogical methods do not require access to the inner workings of a model, they are resistant to the development of new model types. This lack of access also gives them the edge over decompositional methods when it comes to confidentiality. When the cognitive abilities of a user group are matched with a type of explanation and interpretable representation, this same explanation framework can be used for different models, independent of the underlying machine learning algorithms.

3.2.2 Types of Explanations

The main restriction that we have put on model explanations is given by the second insight of Section 3.1.1, which states that the explanation should be understandable for a human. This leaves room for different representations of explanations. Interpretability is a subjective phenomenon so the ability to be flexible regarding the representation of an explanation can be perceived as an advantage. Three dominant formats for explanations become apparent from the literature, all corresponding—to a varying degree—with the ideas of what constitutes a good explanation (see Section 3.1.2):

- Rule-based models and decision trees
- Feature importance and saliency maps
- Example-based explanations

Many representations are built on models that are acknowledged to be inherently interpretable. In the literature these models are also referred to as comprehensible classification models (Freitas, 2014; Huysmans et al., 2011) and the model itself or a slightly modified version of it can be used as an explanation. Decision trees, rule-based models and linear models fall under this category of comprehensible classification models. The other types of explanations are not classification models themselves but rather clever representational formats that map the decision-making process to a form that is easily comprehensible for a human user. In general, this representation is closely related to the input space as this is the level of abstraction human users are used to work with.

The permitted complexity of an explanation depends on the level of expertise of the user for which the explanation is intended. A system analyst would most likely understand intricate plots better than a lay user would. Thus, the explanation provided by an interpretability method should always consider the goal and intended user. Even for representations we consider inherently interpretable, the complexity of the representational format should be tailored to the specific user group.

Rule-Based Models and Decision Trees

Rule-based models and decision trees as a specific class of rule-based models, come as a very natural representation of a decision-making process to human users (Russell and Norvig, 2009). A simple decision tree for example is comprehensible in the sense that it provides logical explanations regarding the path that led to a single decision. Decision trees also give insight into which variables provide the highest information gain and are thus important in the decision making process in general (Quinlan, 1986). Decision trees can be distilled into a set of decision rules following an if-then structure. This structure closely relates to the causal structure humans aim for when explaining a decision (see Section 3.1.2). Exploring alternative paths downs the decision tree allows for the contrastive style explanations discussed in that same section.

An interpretable classification algorithm that uses the rule-based format is known as classification rule mining (Agrawal, Imieliński, and Swami, 1993). Learning association rules is a method intended to uncover relations between variables in a dataset. These methods possess the benefit over decision trees in that they do not impose mutually exclusive rules, leading to shorter explanations. The textual explanation provided by rules however, does not show the hierarchical structure in the same way the graphical explanation of decision trees does.

Feature Importance and Saliency Maps

Another array of models that provides inherent interpretability is the family of regression models (Nelder and Wedderburn, 1972). The magnitude and sign of the coefficients identify relevant features. Naturally, a coefficient with a positive sign results in an increase in model output while a negative sign leads to a decrease in model output. The magnitude of the attribute reflects the strength of this change in model output. In essence, the interpretable representation of regression models is a feature importance-based explanation. Feature importance explanations give a ranked overview of which features are important or contribute most to either an individual decision or the global model decision-making process.

In order to be understandable for users, the representation of the feature must be in a cognitive format that people can work with. Examples include individual words for classification with textual data, the name of a feature in tabular data or a group of pixels in a visual classification task. The last example is often used in image classification task with neural networks. These so called saliency maps aim to visualize which parts of the input image were most influential in the decision-making process of the underlying model. Different methods exist for obtaining saliency maps but they all have in common the representation of an overlay on the input image, which is interpretable for users as it is defined in terms of parts of the image. In order for the explanation to be useful, the highlighted parts of the input should have a strong causal relation with the model output.

Example-Based Explanations

The last interpretable representation consists of actual model inputs. Depending on the classification task, an instance of the dataset can be presented to the user as an explanation. Prototype methods use an instance that is representative for a specific class as an explanation either by contrasting it with the current input that needs an explanation or by comparing it with the current input. In order to generate contrastive explanations, the representations of example-based explanations and feature importance-based explanations can be combined. To establish an understanding of why an input is being classified as belonging to a certain class instead of another class, an instance of the other class can be provided in combination with a feature-based explanation of how they are different. The advantage of an example-based explanation is that they are very intuitive and give give a practical image of why or why not a certain prediction was made. The biggest disadvantage of these methods is that the input examples should be comprehensible themselves as they are used as explanations.

3.2.3 Scope of Explanation

The scope of interpretation is the third key feature of any interpretability method, and it comes in two flavours. The distinction can be made between local explanations and global explanations. A local explanation considers the outcome of only a single prediction while a global explanation considers the complete logic and reasoning behind a model. A global explanation therefore enables following the reasoning to all outcomes. Understandably, it is very challenging to translate the complete logic of a complex classifier with all its non-linearities into something that is comprehensible for a human user. This is the trade-off that model engineers needs to make when considering the scope of interpretability of their model.

Global Explanations

Global model explanations try to create a proxy model that represents in a simplified way the complete logic and reasoning of a black box model. Especially for model validation, it can be very useful to verify this logic in the format of an interpretable model. Global surrogate models are trained on the predictions of a black box model instead of on the original data. This surrogate model technique is often used in engineering when modeling exact processes is too computationally expensive (Gorissen et al., 2010). Even though global surrogate models for prediction explanations can give an indication of the general decision flow within a model, they are generally too sensitive to overfitting to be used in regulated industries where fidelity is a key requirement.

Global feature contribution methods show the aggregated contribution of the input features on the model decisions. This can give insights into the dataset but might not be representative for every individual decision. Attempts have been made to make a broad range of black box models more interpretable by providing the feature contributions along with their shape functions, which works only for tabular data (Caruana et al., 2015). Even if it is possible to approximate an intricate black box model with a recognized interpretable alternative, the simplification comes with reduced predictive performance. This problem is much the same as the problems associated with the approach of achieving interpretability by using inherently interpretable models (see Section 3.2.1). The interpretability-accuracy trade-off that is obeserved in these cases, is similar to the bias-variance trade-off discussed in Section 2.1. In modern machine learning it is non-trivial to have an intelligible model that does not have to give in with regards to functionality. Part of the XAI community tries to solve this problem by trying to explain individual decisions rather than the complete model.

Local Explanations

Local explanations concern only the causal chain that led to a specific prediction. The idea of explaining individual predictions resonates with the requirements of the GDPR. Every user that is subject to an automated decision, should in an ideal scenario be able to request an explanation that provides an understanding of how the model came to the specific decision. In this paradigm, it is not necessary to have a full understanding of the global model. Mathematically, the neighborhood around

TABLE 3.1: The three main dimensions along which methods to achieve interpretability in machine learning can be classified. For all three dimensions—the approach to interpretability, the type of explanation and the scope of the explanation—the options as discussed in Section 3.2 are listed.

Approach		Type	Scope
Inherent		Rule-based models and decision trees	Local
		Feature importance and saliency maps	
Post-Hoc[1]	Model-Agnostic Model-Specific	Example-based explanations	Global

[1] There are two variants of the post-hoc approach to interpretation. Model-specific variants use parts of the model as an explanation while model-agnostic methods treat the model as a black box, querying the model to obtain information about the decision-making process.

the instance is being explained while the other parts of the model might be disregarded for the specific case. This simplifies the explanation process with the goal of increasing interpretability without sacrificing predictive power. The combination of multiple local explanations can give users or model engineers an idea of how the model functions on a broader scale. This aligns also with the idea of interactive exploration, which serves to give an understanding of the workings of the model without disclosing trade secrets or intellectual property (Edwards and Veale, 2017). The challenge for local explanation methods is to create a model on a local scale with high fidelity to the original model.

In this section, methods for achieving interpretability have been outlined and taxonomized. An overview of the three dimensions along which this has been done can be found in Table 3.1. The categories discussed in this section for each dimension are listed vertically under the corresponding dimension. In principal, there are no restrictive dependencies between categories of the different dimensions. However, the overview literature does suggest a relatively uneven distribution over the different categories (Guidotti et al., 2018). The biggest chunk of work has aimed at providing global model explanations although focus has shifted more towards providing local prediction explanations recently. An approach we see quite often is the model-specific post-hoc explanation of (deep) neural networks. This might be rooted in their omnipresence caused by their great practical achievements in the last decades. The popularity of inherently interpretable appears to be persistent, as research using this approach is being done constantly over the years. Model-agnostic methods are on the rise, with many papers taking up this approach in the last three years. Rule-based models and decision trees are used a lot as explanation, mainly for their intuitive contrastive clarification of decisions. For models that work with image data—often in the deep learning paradigm—example-based explanations are common but for other models this explanation type is not often used. Feature importance explanations are receiving more attention lately for their simplicity and flexibility regarding the representation of the importance values.

3.3 Evaluation of Interpretable Models

In the previous sections, the concepts and terminology behind interpretability and explanations have been introduced. A broad range of methods that tries to incorporate these concepts in machine learning have been outlined subsequently. The analysis of the literature led to a promising portrayal of what has been done in the field of XAI already. Despite this groundwork, XAI is still very much a developing field. Guidotti et al. (2018) defined two important open problems in the field that followed from their literature review. One of them concerns the lack of agreement on what an explanation is. This problem can be subdivided into an implicit and an explicit requirement. The implicit requirement is a general formalism that states what an explanation should facilitate and what it should represent in an understandable manner. This problem has been tackled in Section 3.1.2 along with a similar approach for the term interpretability in Section 3.1.1. The explicit requirement is the definition of what exactly an explanation should look like. We believe the form in which an explanation should be provided, is dependent on the context and user. Therefore we consider the general formalism we have sketched out in Section 3.1.2 to be adequate as a definition of what an explanation should be. Irrespective of the explanation taking the form of a decision tree, a set of rules or an input example.

The other problem defined by Guidotti et al. (2018) is the identification of the properties that an explanation should guarantee and the criteria that follow from those properties. They argue that quantifying these properties is of fundamental importance but no work has seriously addressed this problem. Measuring these properties can be difficult since for example comprehensibility is very subjective to a specific user or specific circumstance in which an explanation is required. Still, the goal of this study to assess the quality of explanations aligns with the notion of Guidotti et al. (2018) that the definition of a formalism for measuring different properties of explanations would improve the practical applicability of the methods discussed in Section 3.2.

In this study, we focus on automated decision-making systems for two main reasons. The demand for interpretability is high in this domain as the decisions can have great effects on people and society. This is what distinguishes them from most other optimization problems for which machine learning is used. Another motive for the focus is the regulation around automated decision-making. Most regulation that has been discussed in Section 1.2 is particularly aimed at those systems. Therefore the criteria that line up with the requirements as discussed in that section provide a logical starting point for the formalization of explanation evaluations. Since the relevant legal texts refer to automated individual decision-making, we will further narrow down the focus towards local interpretability (see Section 3.2.3). Meaning that explanations in this study will describe individual predictions of machine learning models.

3.3.1 Evaluating Local Explanations

Before society starts relying on the explanations that come with automated decisions, the quality and trustworthiness of explanations has to be assessed and established. This will improve the practical applicability of interpretable models. In the remainder of this section, an overview of different types of evaluation will be provided. The legislative requirements and possibilities provided by state-of-the-art interpretable models will be combined to formalize the requirements of explanations and corresponding evaluation criteria for the analysis of this study.

Types of evaluation

The subjective nature of interpretability makes evaluating interpretable models challenging. Therefore, a form of benchmarking is needed to measure varying degrees of interpretability. Some studies use evaluation by human participants to do this, which emphasizes the social and psychological aspects of explanations. The advantage of this approach is that the evaluation is independent of the interpretable model that is used. However, experiments involving human participants are relatively expensive to conduct. Furthermore, human evaluation is easily affected by cognitive biases which might hinder objectivity when assessing the quality of explanations (Miller, 2018). One example is the focus of participants on explanatory coherence (Read and Marcus-Newhall, 1993). The fact that participants are looking for coherence with respect to prior beliefs is deemed undesirable when evaluating an explanation objectively. Besides this, participants might not be able to detach the performance of the machine learning model itself from the explanation quality of the interpretability component. Which makes sense for real-life applications, but not for an evaluation framework that focuses only on the quality of the explanations.

In the application domain of this thesis, objective evaluation is critical. Human explanation, which has a large social aspect to it, focuses not only on a truthful transfer of knowledge and understanding. It has a social context to it, which also holds for human evaluation. The influence of this social context on the assessment of explanations, is the main reason for excluding this type of evaluation from our framework. Next to evaluation by human participants, the interpretability of explanations can also be measured by quantitative proxies (Doshi-Velez and Kim, 2017). The objectivity and low cost of these proxies are regarded as advantages, especially within the domain and context of this study. On the other hand, one can argue that evaluation by predefined metrics is too oversimplified to measure an complicated abstraction like interpretability. This is why we stress the importance of aligning the evaluation criteria with the requirements of explanation that have been established in previous chapters.

Requirements for Explanations

Key findings from the analysis of the regulatory landscape (Section 1.2) with regards to automated individual decisions were the requirements that an adequate explanation should meet. An explanation should supply the user with enough information to understand the reasons behind the decision and it should allow a user to act upon the decision—which is to say spot errors and contest the decision.

Explanations are not only products of these requirement but also a process that includes a cognitive and a social aspect. Apart from undesirable components of social explanations like persuasion and bias, it is sensible to include elements of human explanations in interpretable models. As discussed before, this drives humans to accept and adopt explanations more easily since they attribute human traits and intentions to the explanatory agent (De Graaf and Malle, 2017). The cognitive elements of explanations we have discussed in Section 3.1.2 are causes and contrasts. Sociological research on the principal components of human explanations identifies abductive reasoning as an important factor behind explanation (Miller, Howe, and Sonenberg, 2017). A subset of causes is given as explanation instead of all the causes that can be attributed to a decision. The fact that not all causes are used for explaining has to do with cognitive load and efficiency. When it comes to regulatory compliance, there is a thin line between completeness of the explanation and comprehensibility for human users.

Both the insights from the regulatory landscape and social science ask for distinct elements within an explanation. When considering the taxonomy of Section 3.2, multiple options prevail within the domain and context of this study. As we are seeking a general explanation framework that does not capitulate on model performance, post-hoc interpretation methods—particularly model-agnostic variants (Section 3.2.1)—are very suitable. The ideas of using causation to make a user understand the reasons behind a decision aligns with both rule-based explanations as well as with feature importance-based explanations. Both these explanation types visualize the root causes behind a decision. For the purpose of generality, feature-importance based explanations are more convenient since the features can be in any cognitive format that matches the input data.

Evaluation Criteria

The desired characteristics of explanations that allow users to exercise the rights defined in the regulatory landscape in Section 1.2 are in general very common sense. For understanding a decision, the explanation has to correctly and reliably represent the decision of the prediction model. Two aspects that are important to this can be distilled. The first of which is faithfulness to the original model. In the case of local explanations, this property is defined as *fidelity* of the local model to the original model. The second is stability as we would like the explanation to always be faithful to the original model, not only in some adventitious case. This is especially important within the legal context of automated decision-making. Stability itself has two aspects to it. We want every explanation to be *reliable*, the quality of the explanation should be good enough for every individual prediction. Besides that we would like explanations to adhere to the criterion of *continuity*, meaning that a prediction should result in similar explanations when it is requested multiple times. This is especially important considering the approximate nature of the generated explanations in the post-hoc interpretation framework, influenced by factors of chance caused by for example random sampling.

Desirable properties as defined in the literature for methods that generate local explanations, correspond with the characteristics we described above. Lundberg and Lee (2017) define local accuracy and missingness as essential properties to any local explanation. Local accuracy is defined as the agreement between the output of the original model and the output of the local model. Missingness is described as features that are missing in the local model, to not have an impact in the original model. The combination of these properties is covered by what we refer to as *fidelity*. Ribeiro, Singh, and Guestrin (2016) specifically refer to local fidelity as an essential criterion. They underline that the explanation of a single prediction should be faithful to the original model only locally. This means to say that an explanation that is faithful to the original model on a local scale does not imply global fidelity.

Besides the criteria that specifically explanations of automated decisions should follow, there are also characteristics that we would like every explanation to possess. The *efficiency* of the explanation method in terms of processing capacity and computational performance can also be an important factor. Especially in most real-life scenarios, methods for generating explanations should continuously perform well for high numbers of features and data instances to make the method useful in practice. Naturally, we would like the explanation method to not hinder the *accuracy* of the machine learning model. By design, the post-hoc interpretation methods used in this study preserve model accuracy. We would also like explanations to be *robust* to different input- and model types. The model-agnostic explanation methods that will be used in this study, innately accept every type of model as input to the framework.

To summarize, the following evaluation criteria have been identified:

Fidelity The explanation should be truthful to the original model, at least on a local scale.

Continuity Identical inputs should lead to identical explanations.

Reliability The quality of the explanation should be good enough for every individual prediction.

Accuracy The additional layer of transparency should not hinder the performance of the machine learning prediction model.

Robustness The explanation method should be robust with regards to the machine learning model that is being used and the types of data that the model using.

Efficiency The method for generating explanations should work well for high numbers of features and input instances.

This study sets out to evaluate local explanations along the criterion of **fidelity**. For local explanation methods it is essential that the local explanation correctly reflects the original model that it is based on. The formalization of this evaluation is a first step into the direction of a general framework of explanation evaluation on all the criteria.

Summary Chapter 3

3.1.1 Our use of the term *interpretability* is composed of a combination of terminology from social science and human-machine communication research. Model interpretability is established by providing explanations for decisions where the given explanations should be understandable for a human.

3.1.2 *Explanations* are the means to increase model interpretability. The main components of an explanation are causes and contrasts, these follow from psychological and philosophical insights.

3.2 Research on interpretable machine learning can be classified along three dimensions: The scope of the explanations, the approach to interpretability and the type of explanation.

3.3 Automated decision-making models can best be explained on a local scale by post-hoc explanation methods. The requirements that follow from the regulatory landscape and the explanation theory, are satisfied by explanations that use feature importance indications. Therefore they will be employed in this study. In order to create a general evaluation framework, we will focus on model-agnostic explanation methods only.

3.3.1 The rights that users of automated decisions have—understanding and contesting decisions—have been combined with the components of a useful social explanation—causes and contrasts. This resulted in the formulation of an evaluation framework that consists of several criteria; fidelity, continuity, reliability, accuracy, robustness and efficiency.

3.3.1 This study sets out to evaluate local explanations along the criterion of fidelity. For local explanation methods it is essential that the local explanation correctly reflects the original model that it is based on.

Part II

Methods

Chapter 4

Generating Explanations

When black box models are put into practice to automate decision-making, explanation methods can be applied to clarify the decision-making process after the fact. This post-hoc form of interpretability preserves the accuracy of modern machine learning methods while adding a layer of transparency. As explained in Section 3.3.1, explanations in this study will be given on a local scale which is to say on the level of individual predictions. By separating the explanation method from the machine learning model, a flexible framework arises that allows for generalization and easy comparison between models in terms of interpretability and operation. The methods we discuss in this chapter do this by treating the machine learning model as a black-box, preventing the need to inspect internal model parameters.

A popular variant of these model-agnostic interpretability methods uses feature importance indications as prediction explanations. The idea of attributing importance values to individual features combines the requirements for automated decision-making that have been outlined in Section 3.3.1. Feature importance-based explanations give an indication of the influence of the different features in terms of the statistical contribution to the prediction of the underlying model. Most traditional machine learning model engineering processes involve feature engineering, which transforms raw data into predictor variables that match the automated task at hand. Engineered features generally posses a form of interpretability, which makes feature importance-based explanations intuitive and easy to integrate in most machine learning pipelines.

Feature importance-based explanation methods essentially distill an explanation model $\hat{f}$ from the original model f. For data with interpretable features, a single prediction $f(x)$ based on a set of features x is approximated by $\hat{f}(x')$ where x' is a simplified subset of the original features containing features that are considered to be important for the individual prediction by the explanation method. Algorithms that create feature importance explanations thus try to establish a local model for which holds that $f(x) \approx \hat{f}(x')$. The local models follow the function:

$$\hat{f}(x') = \alpha_0 + \sum_{i=1}^{N} \alpha_i^{FI} x_i' \tag{4.1}$$

in which a feature importance value α_i^{FI} represents the effect of feature x_i' on the prediction $\hat{f}(x')$. What differentiates feature importance explanation methods is the approach they take for calculating the feature importance values α_i^{FI}. Different methods with the goal of attributing importance values to individual features have been designed. The two most popular approaches in the literature, that both will be central to the analysis in the remainder of this study, are *local surrogate models* and *computing feature attributions*. Both methods for calculating feature importance explanations will be outlined in the coming sections. One popular and well-documented proponent of each will be evaluated within our evaluation framework; LIME for local surrogate models and kernel SHAP for computing feature attributions. Both algorithms and their implementations will be discussed in detail in this chapter.

4.1 Local Surrogate Models

The main intuition behind local surrogate models is that the non-linear decision boundary of a complex model, and therefore the black-box decision, can be approximated by a more transparent model on a local scale. A visual representation of this assumption can be found in Figure 4.1. In this figure, the binary decision boundary of a random forest model trained on a numerical dataset representing the parabolic function $y = {x_0}^2 - x_1$ is visualized. Instances of the test set are represented as colored dots with black borders. Except for a few instances close to the decision boundary, the model is able to predict the labels—indicated by the color of the dots—correctly. It would be hard to capture the complexity of the global decision boundary of this model in terms of an interpretable model. Instead, local surrogate models can be created for individual instances. These individual instances are depicted as white dots and the local surrogate models—in this case LASSO regression models (Tibshirani, 1996)—are represented as white lines.

In order to construct a local surrogate model, access to the training data is not necessary. Only oracle access to the model is needed, allowing it to be probed as much as needed to get an understanding of the decision boundary around the instance that requires explaining. The local dataset is labeled with the predictions given by the probed original black box model. In Figure 4.1, this local dataset consists of colored dots with white borders. Optionally, this local dataset can be weighted, for example based on the distance of the generated points to the original data point that is being explained. In the figure, this weight is visualized by the size of of the dots. Instances of the local dataset that are closer to the white dot are larger, which represents a larger weight. An interpretable model of choice can then be trained on the local dataset to fit the predictions of the original model. The characteristics of the local interpretable model can then be used as an explanation. The most common explanation model that is fitted on local datasets is the family of linear regression models yet the type of explanation model is interchangeable.

4.1.1 LIME - Local Interpretable Model-Agnostic Explanations

The most popular proponent of the explanation methods that employ local surrogate models is LIME (**L**ocal **I**nterpretable **M**odel-Agnostic **E**xplanations) (Ribeiro, Singh, and Guestrin, 2016). Its concrete implementation of the local surrogate model method involves the following steps:

- Select an instance x' for which an explanation is needed.
- Generate samples around x' by drawing nonzero elements of x' uniformly at random to create local dataset Z.

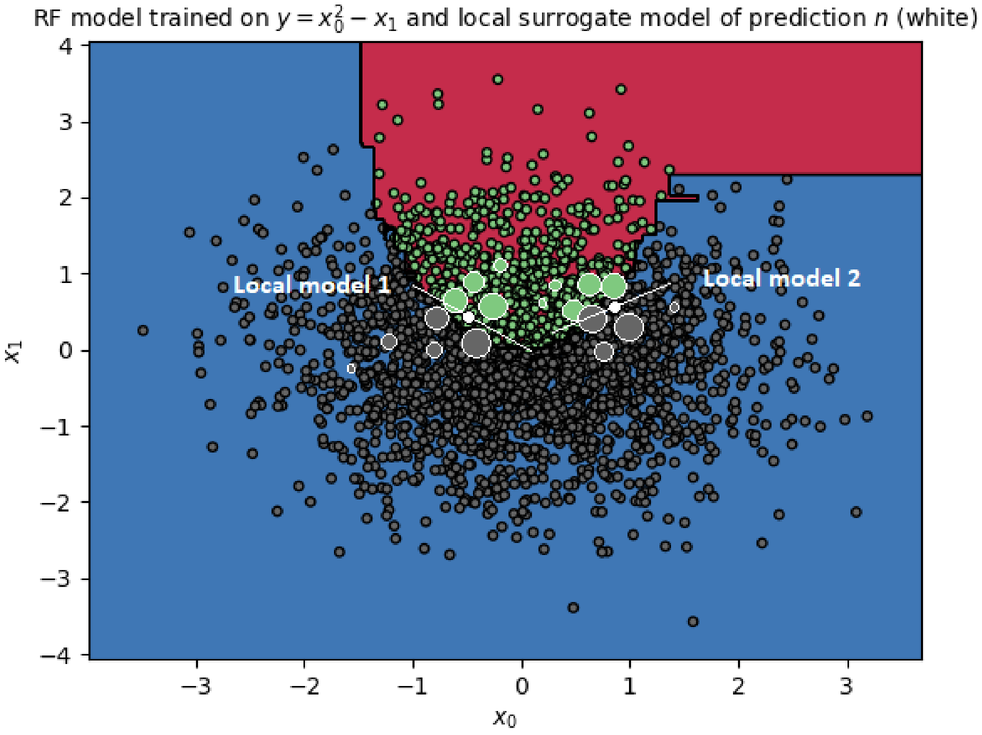


FIGURE 4.1: Local surrogate models (white lines) that explain the predictions of two instances of a random forest model (white dots). Local datasets (colored dots with white borders) are created by random sampling around the original instances. The local datasets are weighted according to the distance of the points to the original input (visualized by dots of varying size). The local model in this case is a LASSO regression model fitted on the local dataset after labeling it by querying the black box model.

- Compute labels for Z by evaluating the instances with the original model f.
- Give a weight to the instances in Z based on the distance to x'.
- Fit a weighted interpretable model $\hat{f}$ to Z.
- Explain x' based on components of the local model $\hat{f}$.

From this list, a few factors that define the quality of the local surrogate model can be distilled. The selected method for weighing the instances of the local dataset, the way perturbed data points are generated for the local dataset and the chosen family and parameters of the interpretable model have a significant effect on the resulting surrogate model $\hat{f}$ and hence on the explanation of x'.

Formal Model Definition

The main goal of LIME is to optimize the trade-off between fidelity and interpretability. This trade-off also becomes apparent from the formal definition of the LIME explanation (as defined by Ribeiro, Singh, and Guestrin, 2016):

$$\xi(x') = \underset{\hat{f} \in F}{\operatorname{argmin}} \, \mathcal{L}(f, \hat{f}, \pi_x) + \Omega(\hat{f}) \tag{4.2}$$

The explanation $\xi(x')$ of instance x' is obtained by selecting a local model that minimizes the combination of two terms. The first term $\mathcal{L}(f, \hat{f}, \pi_x)$ is defined as the locality-aware loss impacted by the original model f, the interpretable model $\hat{f}$ and the distance measure $\pi(z)$. The second term $\Omega(\hat{f})$ represents the complexity of the local model. In practice, the user defines the complexity measure $\Omega(\hat{f})$ by choosing the number of coefficients in the local model and LIME only optimizes local fidelity by minimizing the loss $\mathcal{L}(f, \hat{f}, \pi_x)$. In the LIME implementation, the class of linear regression models is used for training interpretable local models $\hat{f}(z') = \alpha \cdot z'$ where the coefficients α indicate the relative importance of the features. The default is ridge regression (Hoerl and Kennard, 1970), but in theory, the explanation model $\hat{f}$ can be any model from a family of interpretable models F. For the default ridge regression model, the chosen complexity measure $\Omega(\hat{f})$ is the number of non-zero weights in the equation which is to say the number of features K in the local model.

To guarantee the interpretability of the explanation, LIME makes the distinction between an interpretable representation and the original feature space that the model uses. The representation has to be understandable to humans, so its dimension is not always the same as the dimension of the original feature space. It might be for example, that an visual classification model represents the input as a pixel tensor. In such cases, LIME always transfers the input from this feature space to an interpretable representation (e.g. a pixel patch). The tabular data used in the analysis of this study however, does not need mapping to another interpretable representation as the model uses features in a format that is already understandable to users.

The main evaluation criterion for local post-hoc explanations as defined in Section 3.3.1, is fidelity to the original model. In order to comply with this requirement, the interpretable surrogate model should represent the original model on a local scale correctly. The two aspects of LIME that are important to this requirement are the locality of the generated dataset and the accuracy of the interpretable model. Locality around the instance x' is defined by the distance measure $\pi(z)$ that penalizes distance between x' and the point $z \in Z$. The distance of the drawn data points to the original instance is calculated using a Euclidean distance function. Before computing this distance function for tabular data, the LIME implementation encodes categorical features as Boolean values, based on whether they are equal to the particular variable of the original instance. Continuous variables are discretized into quartiles by default in the LIME implementation, as the values might be in different ranges. In this study however, the feature values of the datasets we use are all drawn from the same distribution (see Section 5.1), which makes discretization of the values unnecessary. The distance value between x' and z that follows from the Euclidean distance function D is converted by an exponential kernel of a predefined width σ and included in the loss calculation. The distance measure is therefore formally expressed by:

$$\pi_x(z) = \exp(-D(x', z)^2 / \sigma^2) \tag{4.3}$$

with the default value of kernel width being $\frac{3}{4}\sqrt{n}$ where n is the number of features in the dataset.

The local dataset is generated based on the training data, for which the LIME explanation module computes statistics about the distribution of each variable during initialization. For the generation of data points, the feature distributions are then used to draw samples. A normal distribution with the same mean and standard deviation as the features of the training data is used for this, independent of the individual instance x' that is being explained. This means that the weighing of the generated dataset is the only factor that defines locality of the interpretable model for an individual prediction, making this a fairly important aspect of the explanation. This distance measure is used by the locally weighted square loss function to optimize the fidelity of the local model:

$$\mathcal{L}(f, \hat{f}, \pi_x) = \sum_{z,z' \in Z} \pi_x(z)(f(z) - f'(z'))^2 \tag{4.4}$$

Training a local model with the predefined number of features K that minimizes Equation 4.4 consists of two steps. The first step is selecting the K features that the local model will consist of. In this study, Lasso regularization (Efron et al., 2004) is used for selecting the features. The K features that are least prone to shrinkage based on the regularization path of a Lasso fit of the original model prediction are chosen. The second step consists of learning the weights, which is done via least squares. In the original LIME paper, this method was named K-LASSO (Ribeiro, Singh, and Guestrin, 2016).

4.2 Computing Feature Attributions

Similar to local surrogate models, methods that compute feature attributions define an interpretable approximation of the original model for individual predictions. The approximations of these methods are based on coalitional game theory and conditional expectations and the resulting explanations are presented as relative contributions of individual features. A well-studied technique from game theory that lends itself to assign a unique distribution of pay-off among contributors is the Shapley value (Shapley, 1953). The original version considers coalitional games and strives for fair division of the pay-off among players according to their marginal contribution within a coalition of players. For a specific player, the Shapley value is defined as the average contribution over all possible permutations of coalitions.

Shapley values in a coalitional game with a set of players N and a function $v(S)$ denoting the pay-off for a certain coalition S are computed as follows:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!}(v(S \cup \{i\}) - v(S)) \tag{4.5}$$

Here the sum extends over all possible subsets S of players N excluding player i. The term $(v(S \cup \{i\}) - v(S))$ represents the marginal contribution of player i to the coalition S since it is the pay-off of the coalition S subtracted from the pay-off of the same coalition including i. The marginal contribution is computed over all possible different permutations in which the coalition S can be formed. This is done by calculating the product of the number of permutations of the players in S, which is $|S|!$, with the number of permutations of the players without subsets that include player

i, which is $(|N| - |S| - 1)!$. Ultimately, the average value of this marginal contribution over the possible permutations is computed by dividing by $|N|!$, which is the number of different permutations of N.

The analogy with features as players, model predictions as pay-off and Shapley values as their calculated attribution was first drawn by Lipovetsky and Conklin, 2001. They used Shapley value estimation to compute comparative importance of predictors in regressor models in the presence of multicollinearity. The method requires retraining the model on all possible subsets of all features. To compute the effect of a feature, two models are trained—one including the feature and one excluding the feature. The predictions of both models on the instance that needs explaining are computed. The dependence on other features when withholding a feature, is compensated for by computing the differences for all subsets of features. The Shapley values are then calculated as weighted average of all possible differences and used as feature importance values. Since retraining the model is computationally very expensive, different sampling approximations and integration over samples have been proposed to make the calculation of Shapley values more efficient (Datta, Sen, and Zick, 2016; Štrumbelj and Kononenko, 2014).

4.2.1 Kernel SHAP - Shapley Additive Explanations

The most prominent framework for calculating feature attributions based on Shapley values is SHAP (**SH**apley **A**dditive Ex**P**lanations) (Lundberg and Lee, 2017). The framework consists of multiple model-specific optimizations for calculating feature attributions, such as for tree-based models (Lundberg, Erion, and Lee, 2018). In this study however, only the model-agnostic local explanations provided by Kernel SHAP will be used as we do not want to give in on model flexibility.

For practical datasets with more than just a few features, computing the exact Shapley value of a feature x_i in reasonable time becomes problematic. All possible subsets of features have to be evaluated with and without feature x_i. The number of possible subsets increases exponentially when more features are added to the data. If an instance contains N features then the list of all possible subsets of features will have 2^N elements. Therefore the Shapley values are approximated in the Kernel SHAP implementation by the following steps:

- Select an instance x' for which an explanation is needed
- Generate K random binary vectors k of the same length M as the instance to be explained: $k \in \{0, 1\}^M$.
- Create a dataset Z with samples by mapping the binary vectors to the instance x'. A value k_i in a binary vector represents whether feature i of the instance to be explained will be present (1) or absent (0) in a sample z
- For every 1 in the binary vector, the corresponding value from the instance x' is mapped to z. For every 0 in the binary vector, the value of the corresponding feature of another instance that we sample from the data is mapped to z. This essentially means that *absent* features are replaced by a random feature value from the dataset
- Get prediction for each z by querying the original model f: $f(z)$
- Use the SHAP kernel to compute weights for the instances z of the dataset Z

- Fit a weighted linear model $\hat{f}$ to dataset Z, using the predictions of the original model f
- Use the estimated Shapley values $\alpha_i(z')$, the coefficients from the weighted linear model, as explanation for $f(x')$

In this algorithm, the Shapley value approximations are used as an additive feature attribution technique in a linear model, which connects the method to the LIME approach as described in Section 4.1.1. The main differences between both methods are the sampling approach that is used for creating the local dataset and the kernel that is used to give weights to the samples. Both concepts relate directly to the fidelity criterion as they define the locality of the sample dataset.

Formal Model Definition

From the algorithm above it follows that when creating the sample dataset Z, the kernel SHAP method ignores the dependence structure between present and absent features in the samples z. Statistically, that entails sampling absent features from the marginal distribution instead of the conditional distribution. This is needed in order to guarantee that the resulting values will be Shapley values. The downside is that instances that are unrealistic in practice, might end up in the sample dataset as no conditions are enforced on the relation between the present and the replaced features.

The most important aspect of the kernel SHAP algorithm is the kernel that is used for enforcing locality. In Section 4.1.1, the instances in the local dataset were weighted according to the proximity with respect to the instances that needs explaining. In terms of the binary sample vectors in the SHAP algorithm, this means that the more ones in the binary vector, the larger the weight of the resulting sample. For kernel SHAP however, samples generated using binary vectors with many zeros would also lead to large weights. The intuition behind this weighting scheme is that more can be learned about features if their effects are evaluated in isolation. On the one hand a sample that has many features in common with the instance to be explained, can tell us more about the effect of omitting that feature from the instance. On the other hand a sample with few features in common with the instance to be explained, can tell us more about the effect of those few features on the prediction. A sample that has around half of its features in common with the instance to be explained is not very informative in terms of the effect of individual features as there are many possible permutations with half of the features present. This effect of the number of permutations is also visible in the original Shapley value estimation from Equation 4.5.

In the SHAP paper (Lundberg and Lee, 2017), these ideas are captured in the SHAP kernel:

$$\pi_x(z) = \frac{(M-1)}{\binom{M}{|z|}|z|(M-|z|)} \tag{4.6}$$

In this formula, M is the amount of features that the instance consists of and $|z|$ is the amount of features that are mapped from x onto the sample z. In the SHAP library, if not specified otherwise by the user, the number of samples will be $K = 2 \cdot M \cdot 2048$. Instead of creating completely random binary vectors, the SHAP library uses a heuristic to tactically generate samples. Based on the value of K, as much samples as possible that would get a large weight are generated by starting with all permutations that have 1 and M-1 features in common with the instance to be explained. The next step would be to include all permutations having 2 and M-2 features in common with the instance to be explained and so on. The linear model $\hat{f}$ that is fitted on the sample dataset is then trained using the same locally weighted square loss function we saw in the LIME implementation (Equation 4.4).

Summary Chapter 4

4 Feature importance-based explanations essentially distill an explanation model from the original model to give an indication of the influence of the different features in terms of the statistical contribution to individual predictions. The idea of attributing importance values to individual features combines the requirements for automated decision-making as posed in Section 3.3.1.

4 The two main methods for generating local feature importance explanations are *local surrogate models* and *computing feature attributions*. The proponents of both methods that will be used in this study are *LIME* for local surrogate models and *kernel SHAP* for computing feature attributions.

4.1 Local surrogate models randomly generate a weighted dataset around the instance to be explained. Based on the predictions of the original model on this local dataset, an interpretable model is fitted that can be used to explain the original prediction.

4.1.1 LIME heuristically approximates the decision boundary of a machine learning model on a local scale. Local dataset generation, the similarity metric that is used for weighting and the chosen family and parameters of the interpretable model have a significant effect on the result.

4.2 The approximation methods that compute feature attributions are based on coalitional game theory and conditional expectations (e.g. Shapley values) and the resulting explanations are presented as relative contributions of individual features.

4.2.1 Kernel SHAP uses the intuition behind Shapley values and approximates them by generating independent samples around the original instance and weighting these samples using the SHAP kernel. The estimates are used to attribute a contribution value to individual features with a linear model, which connects the method to LIME.

4.2.1 The main differences between LIME and kernel SHAP are the sampling procedure they employ and the kernel they use for weighting the samples in the local dataset when fitting the linear explanation model.

Chapter 5

Methods

This chapter outlines how the explanation framework from previous chapters will be applied. The goal is to perform a replicable study and align the methodology with the objectives of this thesis as stated in Chapter 1. The data generation process and the intuitions behind the synthetic datasets will be explained in Section 5.1. The qualitative and quantitative methods to evaluate the explanation methods from Chapter 4 on the criteria from Section 3.3.1 will be explained in Section 5.2.

5.1 Synthetic Datasets

Evaluating model predictions in the supervised learning paradigm is based on a provided label which serves as a ground truth. This ground truth can be compared to the actual prediction of the model to calculate an accuracy term. This term can then be used to quantitatively compare different prediction models. The quantitative evaluation of prediction explanations is more challenging since a ground truth is not easily defined for real world data. An evident ground truth would imply knowing beforehand what the decisive features should be for a decision. If this were to be the case, the added value of using a complex and opaque model for learning would be questionable.

The strength of complex and opaque models lies in the fact that they can map complex relations between variables that are not immediately obvious to lay users. A method for quantitative evaluation of the quality of explanations that is proposed in this work, uses synthetic datasets with distributions known beforehand. With this ground truth, explanations can be validated easily. Especially for evaluating the fidelity criterion, this is an essential step. The use of synthetic datasets also allows for freedom in creating multiple types of distributions that can be compared.

The synthetic datasets are constructed with a predefined amount of features and samples. The functions that define the distribution of the data are selected to cover a wide range of possible feature relationships. More information on the composition of the synthetic datasets can be found in Table 5.1. In line with the goals of this study, we stick to the use of tabular data as it is the most used data type for automated decision-making. Concurrently, working with tabular data is computationally more efficient and it is straightforward to define informative features for this type of data. Extensions towards other data types and different distributions are left for future work.

Based on computational considerations, the standard size of the datasets is 2500 instances. The models are trained on 80% partitions of the datasets, after which the

TABLE 5.1: The synthetic datasets used in this study. The numerical and Boolean functions that are used for the distributions can be found in the leftmost column.

			Features			
Function	**Data Type**	**Samples**[1]	Informative[2]	Random[3]	Total[1]	**Explanations**[1]
$x_0 + x_1$	Numerical	2500	2	8	10	500
${x_0}^2 - x_1$	Numerical		2	8		
$x_0 * x_1$	Numerical		2	8		
ds_4 [4]	Numerical		6	4		
x_0	Boolean		1	9		
x_0 & x_1	Boolean		2	8		
$(x_0 \mid x_1)$ & $\neg(x_0$ & $x_1)$	Boolean		2	8		
ds_8 [5]	Boolean		6	4		

[1] Limited based on available computational power
[2] Number of features that are used for evaluating and labeling the instances
[3] Number of random features that are non-informative for computing instance labels
[4] $3 * x_0 + 2 * x_1 + x_2 - 3 * x_3 - 2 * x_4 - x_5$
[5] $x_0 \mid (x_1$ & $x_2) \mid (x_3$ & x_4 & $x_5)$

remaining 20% (500 samples) can be used for testing and generating explanations. The explanation methods will be tested for models trained on both numerical and Boolean data. Numerical datasets are labeled based on the evaluation of a polynomial expression while the Boolean datasets use logical expressions. The different functions are selected based on the diversity with respect to the decision boundaries they generate. For both data types, this means that for the diversity of the decision boundaries, spatial differences are ignored. An example includes the decision boundaries of the two polynomial functions $x_0 + x_1$ and $x_0 - x_1$, which are mirrored variants of each other (see Figure 5.1). The same holds for functions that lead to decision boundary solely divided by a rotational factor or a projection, as is common for logical functions (e.g. $x_0 \wedge x_1$ and $x_0 \vee x_1$). The visualizations of the distributions in the dataset like in Figure 5.1 are two-dimensional for plotting purposes only, as they are intended to demonstrate the relationship between the informative features. In reality, the datasets are multidimensional as non-informative features that do not play a role in the labeling process are also present in the data.

5.1.1 Numerical Data

All the data points in the numerical datasets are drawn randomly from a standard normal distribution with $\mu = 0$ and $\sigma = 1$ that is described by the following probability density function:

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \tag{5.1}$$

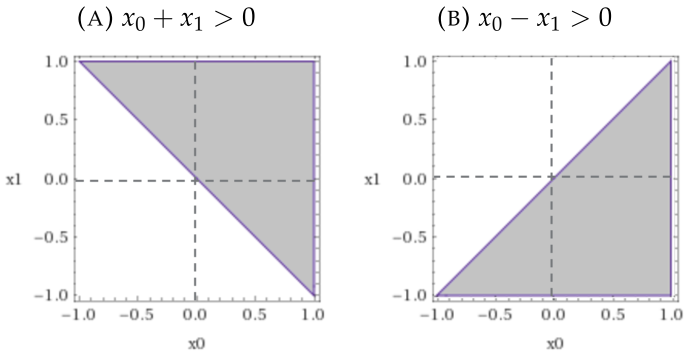


FIGURE 5.1: An example of spatial differences that are ignored for synthetic data generation. Both functions (A) and (B) lead to decision boundaries that are mirrored variants of each other.

The output label of a single instance is then calculated based on the generated values for specific features (in most cases x_0 and x_1) using a polynomial expression. In general, the majority of features does not have any influence on the calculation of the label. Only for ds_4 and ds_8, there are more than two informative features. The datasets are created for a binary classification task so the labels can either have a value of zero or a value of one. When the evaluation of the synthetic function (as displayed in the first column of Table 5.1) using the generated values for the informative features, leads to a value greater than zero, the assigned binary label will be one. If the evaluation leads to a value equal to or below zero, the binary label will be zero.

Visualizations of the true decision boundaries of the numerical functions with two informative features can be found in Figure 5.2. These plots demonstrate the differences between the several functions and the decision boundaries they produce. The filled contours of the plots represent the parts of the data with a binary label one—there the value of the function evaluation surpasses the threshold of zero.

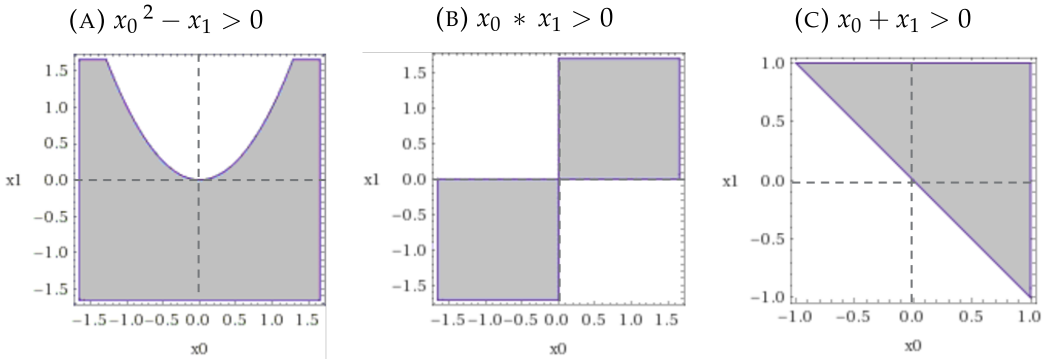
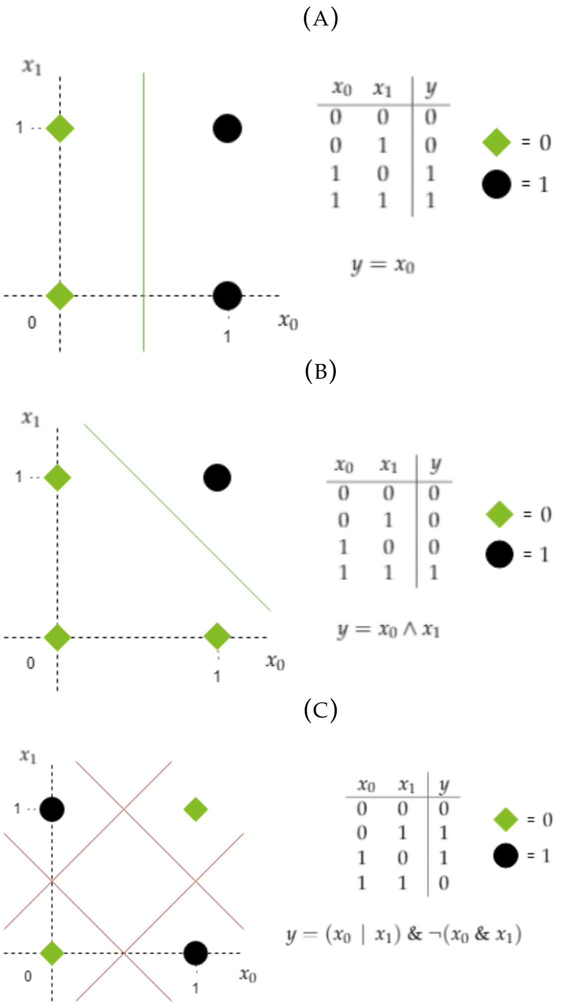


FIGURE 5.2: Visualizations of the true decision boundaries of the numerical datasets with two informative features.

5.1.2 Boolean Data

The data points in the Boolean datasets are assigned a value of zero or one, both with a probability of 50%. Logical expressions given in the function column of Table 5.1 are evaluated with the generated Boolean values of the informative features (in most cases x_0 and x_1), leading to a binary value of one or zero if the expression is *false* or *true* respectively. For three of the Boolean datasets, the function can be visualized on a 2D-grid. These can be found in Figure 5.3. Similar to the numerical data, rotations and other spatial variants are excluded. The logical relations that are used and displayed in Figure 5.3 are logical projection (x_0), conjunction (x_0 & x_1) and the exclusive-or (XOR) relation (($x_0 \mid x_1$) & $\neg$(x_0 & x_1)).


(A)

x_0	x_1	y
0	0	0
0	1	0
1	0	1
1	1	1

(B)

x_0	x_1	y
0	0	0
0	1	0
1	0	0
1	1	1

(C)

x_0	x_1	y
0	0	0
0	1	1
1	0	1
1	1	0

FIGURE 5.3: Visualizations of the decision boundaries in the boolean datasets. (A): Logical projection with a possible decision boundary (green line). (B): Logical conjunction with a possible decision boundary (green line). (C): XOR classification problem. The (red) solid lines demonstrate that the problem space is linearly inseparable.

The XOR-relation is notoriously unsolvable with linear classification models. This is illustrated in Figure 5.3c where a single linear model can not separate the two different classes. Note that the dataset based on the polynomial function $x_0 * x_1$ represents the numerical variant of the XOR problem (Figure 5.2b).

5.2 Evaluation Metrics

The synthetic datasets will be used to evaluate the local explanations generated by LIME and kernel SHAP. A broad overview of the experimental set-up can be found in Table 5.2. Before the LIME and SHAP implementations can be applied to the model predictions, these models have to learn to accurately predict the label of any instance by feeding them the training portion of the different datasets. Applying the four different learning algorithms from Section 2.1 to 80% partitions of all the different datasets, resulted in 32 different machine learning models trained to predict the label of a given input instance. These models were used to predict the labels of unseen instances that were in the remainder of the datasets (20% test partitions). Two metrics for quantifying the quality of predictions are computed; accuracy and F1-score. The F1-score is reported to prevent relying on only accuracy for datasets with class imbalances as it represents the harmonic mean of the precision and the recall of the models on the test set.

Datasets (n=8)
$x_0 + x_1$
$x_0 * x_1$
${x_0}^2 - x_1$
$3 * x_0 + 2 * x_1 + x_2 - 3 * x_3 - 2 * x_4 - x_5$
x_0
$x_0 \ \& \ x_1$
$(x_0 \mid x_1) \ \& \ \neg(x_0 \ \& \ x_1)$
$x_0 \mid (x_1 \ \& \ x_2) \mid (x_3 \ \& \ x_4 \ \& \ x_5)$

Models (n=4)
Random Forest
Neural Network
Decision Tree
Logistic Regression

Explanation Methods (n=2)
LIME
Kernel SHAP

TABLE 5.2: Overview of variables in experimental set-up

LIME and Kernel SHAP have surfaced as the most popular model-agnostic feature importance explanation methods. Part of their prevalence over other methods can be attributed to the open-source libraries that are regularly updated for different programming languages. For this study, we use the Python libraries[1] [2] of both methods. We use the absolute values of the feature importance coefficients provided by LIME and SHAP to simplify the explanation output.

Prediction explanations that use indications of feature importance make a simplified model on a local scale. The main criterion that will be evaluated, as outlined in Section 3.3.1, is the fidelity of the local models to the original machine learning models on a local scale. The evaluation framework for this criterion has been designed along two dimensions. The first is a ground truth assessment, both quantitatively and qualitatively. We know from the predefined underlying distributions that the influence of certain features should be significant. The other features, which are not

[1] LIME: https://github.com/marcotcr/lime - visited on 18-05-2020

[2] SHAP: https://github.com/slundberg/shap - visited on 18-05-2020

used for determining the label, should not be identified as being influential to the prediction by the explanation methods.

For this ground truth assessment, the distribution of the dataset therefore has to be known beforehand. Since this is not the case for almost any real-life dataset, another dimension of evaluation is added to the framework. This step is based on the intuition that altering the value of the most important feature for a particular decision, should result in the highest probability of actually changing the decision. Ideally, this probability should be higher than the probability of changing the decision by altering the values of less important features. Evaluating explanation in this way, serves as a first step towards objective evaluation of generated explanations for any prediction model; a trademark that would greatly benefit the field of machine learning.

From a processing perspective, the performance of the explanation methods will be evaluated by looking at the time differences for explanation generation between both methods. The experiments were performed on a laptop with Intel Core i5 processor with 8GB RAM. This can be seen as a common day-to-day platform that should be able to facilitate computations on datasets of this size within a reasonable amount of time. Especially in domains where large datasets with many features are common, the computational performance can be a decisive component of an explanation method. The synthetic datasets and models that are used in this study do not require state-of-the-art processing power. This does not hold for most automated decision-making models that are used in practice. As both explanation methods require querying (or partly retraining) the model, this might be problematic for datasets containing hundreds of features of millions of users.

5.2.1 Comparison with Ground Truth

The main advantage of the synthetic datasets we have created, is that the we now have a ground truth to which we can compare the generated explanations. This will be done quantitatively, by comparing the rank correlation coefficients of the generated explanations and the ground truth feature importance orders. Since we use the absolute values of the LIME and SHAP explanations, the ground truth orders can be expressed with only positive values. For all the synthetic datasets, the ranking vectors that have been created can be found in Table 5.3.

There are four datasets for which x_0 and x_1 are the only important features with them both having equal importance. Because of the dataset distributions, the importance of x_1 in the ground truth is slightly higher than the importance of x_0 for ${x_0}^2 - x_1$. For that we refer back to the standard normal distribution from which the data points are drawn (see Section 5.1). With most values being between zero and one, the quadratic element of ${x_0}^2$ makes the influence of that feature x_0 smaller than the influence of feature x_1 when evaluating the equation ${x_0}^2 - x_1 > 0$. The datasets with cascading influence (ds_4 and ds_8) are the only datasets with three levels of importance in the ground truth ranking vectors. The ground truth vectors will be used in a quantitative comparison for both explanation methods and they will also be used as reference for the qualitative assessment of the generated feature importance values of both methods.

TABLE 5.3: The synthetic datasets (leftmost column) with their ground truth feature importance vectors (n=10).

	Ground truth									
Dataset	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9
$x_0 + x_1$ $x_0 * x_1$ x_0 & x_1 XOR [i]	1	1	0	0	0	0	0	0	0	0
${x_0}^2 - x_1$	1	2	0	0	0	0	0	0	0	0
ds_4 [ii]	3	2	1	3	2	1	0	0	0	0
x_0	1	0	0	0	0	0	0	0	0	0
ds_8 [iii]	3	2	2	1	1	1	0	0	0	0

[i] $(x_0 \mid x_1)$ & $\neg(x_0$ & $x_1)$

[ii] $3 * x_0 + 2 * x_1 + x_2 - 3 * x_3 - 2 * x_4 - x_5$

[iii] $x_0 \mid (x_1$ & $x_2) \mid (x_3$ & x_4 & $x_5)$

Kendall's Tau-b Rank Correlation Coefficient

For the different models and datasets, we will compute the Kendall's Tau-b rank correlation coefficients (Kendall, 1948). This coefficient reflects the correlation with respect to the ranks within two non-parametric data samples. Values of Kendall's Tau-b rank correlation coefficients range from -1 (strongest negative correlation) to 1 (strongest positive correlation). A value of zero indicates the absence of rank correlation. What distinguished tau-b from other variants of rank correlation metrics is that it makes adjustments for ties. Since we use many features with similar (ground truth) importance values, this is an important addition. Kendall's Tau-b rank correlation coefficient τ_b is computed as follows:

$$\tau_b = \frac{n_c - n_d}{\sqrt{(\frac{n(n-1)}{2} - n_g)(\frac{n(n-1)}{2} - n_e)}} \tag{5.2}$$

where n is the amount of values in both samples (10 in our case), n_c is the number of concordant pairs, n_d is the number of discordant pairs and n_g and n_e is the number of possible pairings with a tie in the ground truth and explanation, respectively. The denominator actually calculates the number of possible ways of selecting distinct pairs, with a correction for the number of ties (Haasdijk and Heinerman, 2018). For the 500 LIME and SHAP explanations of every model and dataset, the coefficient will be computed. Per model and dataset, the mean correlation values of LIME and SHAP with the ground truth will be tested for significant difference. As high positive correlation values indicate strong correlation, they also indicate high quality explanations that are very similar to the ground truth in terms of rank order.

Qualitative Assessment of Normalized Feature Values

The output of LIME and SHAP consists of an importance value for each feature. The mean values will be visualized in a bar plot for every dataset and model. By normalizing the feature importance values of both explanation methods, we can assess the explanations on a common scale, without distorting differences in the ranges of the values. A common normalization method called the z-transformation, will be used for this. For the z-transformation, the mean is subtracted from the sample and then the sample is divided by the standard deviation. The result will have mean $\mu = 0$ and standard deviation $\sigma = 1$. Z-scores become comparable by measuring the observations in multiples of the standard deviation of that sample. Formally, the normalization follows the function:

$$z_i = \frac{x_i - \mu}{\sigma}$$

with z_i the z-transformed sample observations, x_i the original values of the sample, μ the sample mean and σ the standard deviation of the sample

5.2.2 Intuitive Assessment

Next to comparing the feature importance values with the ground truth by means of the synthetic datasets, we propose an intuitive method for evaluating an explanation that is grounded in both elements of explanations that have been defined in Section 3.1.2. For rank-based feature importance explanations, it is intuitive to think that one might have the highest change of flipping a prediction to a desired outcome by changing the most influential features according to the feature importance explanation. This intuition will be used as a proxy for assessing the quality of an explanation. Similar to ground truth evaluation, this method will also consist of both a quantitative metric and a qualitative assessment.

By changing the most important features—according to an explanation method—for multiple predictions , we calculate how big the chance is that the prediction flips. In order to do this, the absolute feature importance values for both methods are ranked based on their magnitude. The feature with the highest magnitude—and hence the biggest influence on the prediction according to that explanation method—will be changed first. If the prediction does not change when changing that one feature, the next feature in the ranked explanation value list is changed. The value that will be saved and reported is the rank of the feature that made the prediction change. We define the rank of the feature that changes the model output as the recourse value, as it symbolizes the recourse of a user that wants to change the prediction of an automated decision-making system. A recourse value of zero is returned when the prediction did not change at all. Feature values are not being changed back to their original value when the next feature in the ranked list is changed.

For both data types, the recourse value for flipping the prediction is computed in a slightly different way. For models trained an tested on Boolean data, flipping a feature value is done by just reversing the Boolean value—0 becomes 1 and 1 becomes 0. For numerical data, this is a bit harder since simply flipping the sign of a value close to zero would not have the same impact on the instance compared to flipping the sign of a value far away from zero. Inspection of the standard normal distribution revealed 99.8% of the data points to be less than 3 times the standard deviation σ away from the mean μ. Therefore every numerical feature value is changed in both directions by adding and subtracting 6σ and checking whether the prediction changes because of either one of these operations.

Skewness

For an effective explanation in terms of giving a user the ability to change an automated decision, the desired recourse value for an explanation should be low. The quantitative metric we will use for evaluating this, is based on the skewness of a distribution. Pearson's coefficient of skewness (Joanes and Gill, 1998) is a metric that defines symmetry. Negative skewness values indicate that the mean of the data values is less than the median, meaning the data distribution is left-skewed. Positive values of skewness indicate that the mean of the data values is larger than the median, meaning that the data distribution is right-skewed. As low recourse values are related to effective explanations, it is desirable that the mass of the distribution is concentrated on the left. This would indicate a mean that is larger than the median, which makes large positive skewness values desirable in our analysis. The formula for Pearson's coefficient of skewness is:

$$g_1 = \mu_3 / \mu_2^{3/2} \tag{5.3}$$

where μ_2 and μ_3 are the second and third central moments. The r-th central moment μ_r is defined as $\Sigma_i (x_i - \mu)^r / n$ where n is the number of values and μ is the mean value.

Qualitative Assessment of Recourse Values

The qualitative assessment of recourse values is based on the frequency of recourse value combinations. Since the high quality of the generated explanations and the relative simplicity of the datasets makes the two explanations concur very often, comparing the means is less useful. The recourse values for all the accurate models (n=30) will be visualized on a 2D-grid with both explanation methods along the different axes. In this way, we can create a heatmap that allows us to look at the conditional distributions of the recourse values more closely.

Summary Chapter 5

5.1 Eight synthetic datasets containing Boolean and numerical data have been created using polynomial and logical functions selected for the diversity in terms of true decision boundaries they represent.

5.2 The methodology to assess the evaluation framework for the criterion of fidelity of the local models to the original machine learning models has been designed along two dimensions. The first is a ground truth assessment that uses the known distributions of the synthetic data for evaluation. The second is a generalizable method that uses the intuition that altering the values of the important features should give a user a high chance of changing the decision. Both methods will be used quantitatively and qualitatively, as this combination yields the most complete evaluation of the framework.

5.2 The ground truth assessment will be performed quantitatively by looking at the rank correlation coefficients of the explanation value vectors and a ground truth vector. Because the data contains many tied ranks, Kendall's Tau-b rank coefficient will be used to compute correlation.

5.2 The intuitive assessment based on recourse will be quantified by calculating the Pearson's coefficient of skewness for both recourse value distributions. The intuition behind this is that low recourse values indicate effective explanations and positively skewed distributions suggest low recourse.

Part III

Results and Analysis

Chapter 6

Evaluating Feature Importance Explanations

Based on the learning algorithms discussed in Section 2.2, machine learning models have been trained and tested on the synthetic datasets described in Section 5.1. In the first section of this chapter, the classification performance of these models will be reported. For the predictions of the models, explanations are generated using the explanation methods from Chapter 4. The generated explanations will be visualized and evaluated on the criteria from Section 3.3.1 using the metrics introduced in Section 5.2.

6.1 Model Performance on the Datasets

Four different type of models have been trained on 80% partitions of eight synthetic datasets containing 2500 instances with ten features and a corresponding binary label (see Table 5.1). In Table 6.1, performance of all 32 models on the 20% test partitions of the datasets is reported. In the *Prior* column of Table 6.1, it is demonstrated that some datasets are slightly imbalanced, as prior probabilities for the binary class 1 varies between 0.250 and 0.721. This is expected due to the evaluation functions that are used for labeling the synthetic data. However, F1-score and accuracy correspond very strongly for every combination of model and dataset so class imbalance does not seem to be a problem. Next to the two performance metrics, the mean time in seconds it took both explanation methods to generate the feature importance values, is also reported.

We observe an accuracy on the test set of around 50% for two Logistic Regression models (colored red in Table 6.1). One trained on the Boolean XOR-dataset and the other on the numerical XOR-dataset ($x_0 * x_1$). The XOR-problem is notoriously unsolvable with linear classification (see Figure 5.3c), which explains the bad performance of the logistic regression model on these datasets. The models have not been able to learn the underlying distribution of the data.

Three other models have an accuracy on the test set under 95% (colored orange in Table 6.1). The numerical dataset that is labeled using the ${x_0}^2 - x_1$ function is characterized by a parabolic decision boundary. This explains why the logistic regression model that is trained on this dataset does not achieve competitive performance. In Figure 6.1, this is visualized clearly. The other two models are both tree-based models trained and tested on the numerical dataset with cascading influence (ds_4).

TABLE 6.1: Classification performance of the predictions of the different model types on 20% test partitions of all the datasets. The accuracy of a model represents the percentage of correctly classified instances in the test set. The F1-score represents the harmonic mean of the precision and the recall of the model on the test set.

			Model Performance[i]			
Datasets	**Prior** [ii]	**Metric**	RF	NN	LR	DT
Synthetic numerical data						
$x_0 + x_1$	0.515	Accuracy	97.0%	99.0%	100%	96.6%
		F1-score	0.968	0.990	1	0.965
		LIME time (s)	0.041	0.032	0.027	0.028
		SHAP time (s)	18.92	18.71	16.25	16.43
${x_0}^2 - x_1$	0.721	Accuracy	96.0%	98.6%	83.8%	98.0%
		F1-score	0.971	0.990	0.885	0.986
		LIME time (s)	0.040	0.032	0.027	0.027
		SHAP time (s)	18.80	18.71	16.25	16.44
$x_0 * x_1$	0.504	Accuracy	94.0%	98.4%	49.0%	99.6%
		F1-score	0.938	0.984	0.569	0.996
		LIME time (s)	0.043	0.033	0.027	0.028
		SHAP time (s)	19.08	18.53	16.11	18.72
ds_4 [iii]	0.517	Accuracy	90.2%	99.0%	99.6%	84.0%
		F1-score	0.900	0.990	0.996	0.837
		LIME time (s)	0.041	0.032	0.028	0.028
		SHAP time (s)	19.00	18.60	16.18	16.38
Synthetic Boolean data						
x_0	0.494	Accuracy	100%	100%	100%	100%
		F1-score	1	1	1	1
		LIME time (s)	0.042	0.030	0.023	0.023
		SHAP time (s)	12.18	10.04	7.743	7.700
x_0 & x_1	0.250	Accuracy	100%	100%	100%	100%
		F1-score	1	1	1	1
		LIME time (s)	0.031	0.031	0.023	0.023
		SHAP time (s)	9.093	10.13	7.703	7.720
XOR [iv]	0.482	Accuracy	99.8%	100%	46.8%	100%
		F1-score	0.998	1	0.502	1
		LIME time (s)	0.035	0.031	0.023	0.023
		SHAP time (s)	9.858	10.14	7.625	7.707
ds_8 [v]	0.677	Accuracy	100%	100%	97.6%	100%
		F1-score	1	1	0.983	1
		LIME time (s)	0.035	0.035	0.026	0.026
		SHAP time (s)	10.32	11.52	8.586	8.640

[i] Random Forest (RF); Neural Network (NN); Logistic Regression (LR); Decision Tree (DT)

[ii] Prior probability of the binary instance label being 1

[iii] $3 * x_0 + 2 * x_1 + x_2 - 3 * x_3 - 2 * x_4 - x_5$

[iv] $(x_0 \mid x_1)$ & $\neg(x_0$ & $x_1)$

[v] $x_0 \mid (x_1$ & $x_2) \mid (x_3$ & x_4 & $x_5)$

Interestingly enough, the other two model types do perform well on this dataset with features of varying importance.

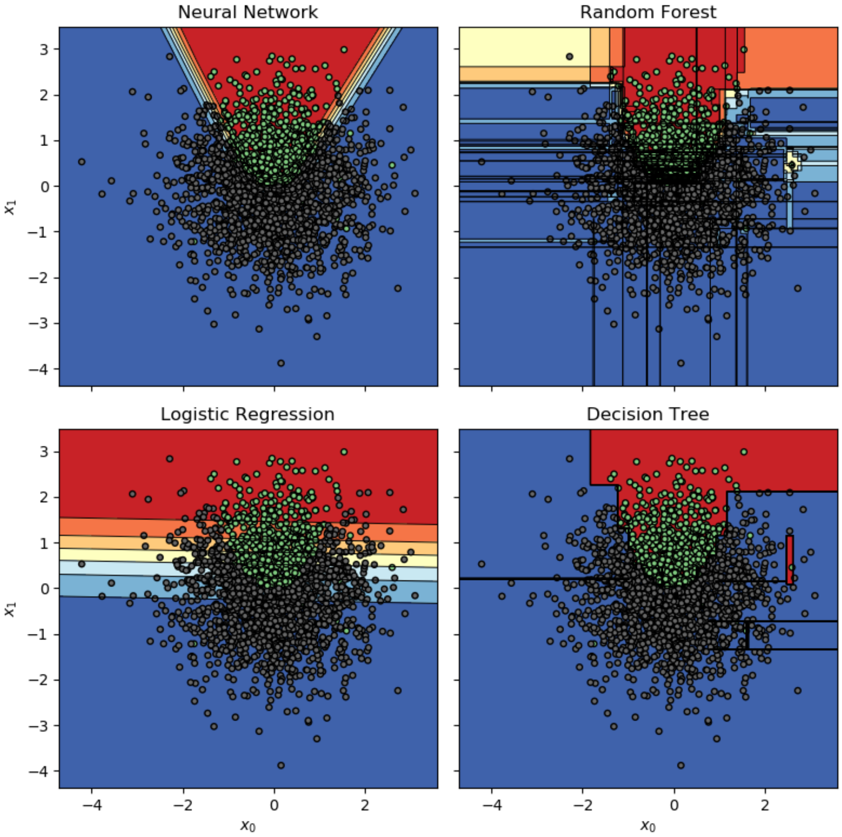


FIGURE 6.1: Learned decision boundaries for different models on ${x_0}^2 - x_1$ data. The class of a data point is indicated by its color (green for 0 and grey for 1). The classification of the model is indicated in red (0) and blue (1). For the models that calculate probabilities, these are visualized by colors ranging from red to blue.

The last trend that can be observed by looking at the accuracy scores, is that it seems to be easier for the models to learn underlying distributions for Boolean data than for numerical data. This can be explained by the variance of the distribution from which the numerical data is drawn. When looking at Figure 6.2 (numerical XOR) and 6.3 (logical XOR), this difference is visualized clearly. The sets in the vector space are divided similarly but the distribution of the data points is not. For the other datasets that have one or two informative features, the data distribution of the first two features and the decision boundaries of the different models have also been visualized. The remaining figures can all be found in Appendix A.

In Table 6.1, we observe significant differences between LIME and SHAP when it comes to the time it takes to generate an explanation. This can be attributed to the efficiency of the underlying method of feature importance calculation as discussed in Chapter 4. There are some small differences between the explanation times for the different model types. The general trend seems to be that explaining predictions for the logistic regression and decision tree models takes slightly less time. Since the only difference in generating explanations between the different models lies in the

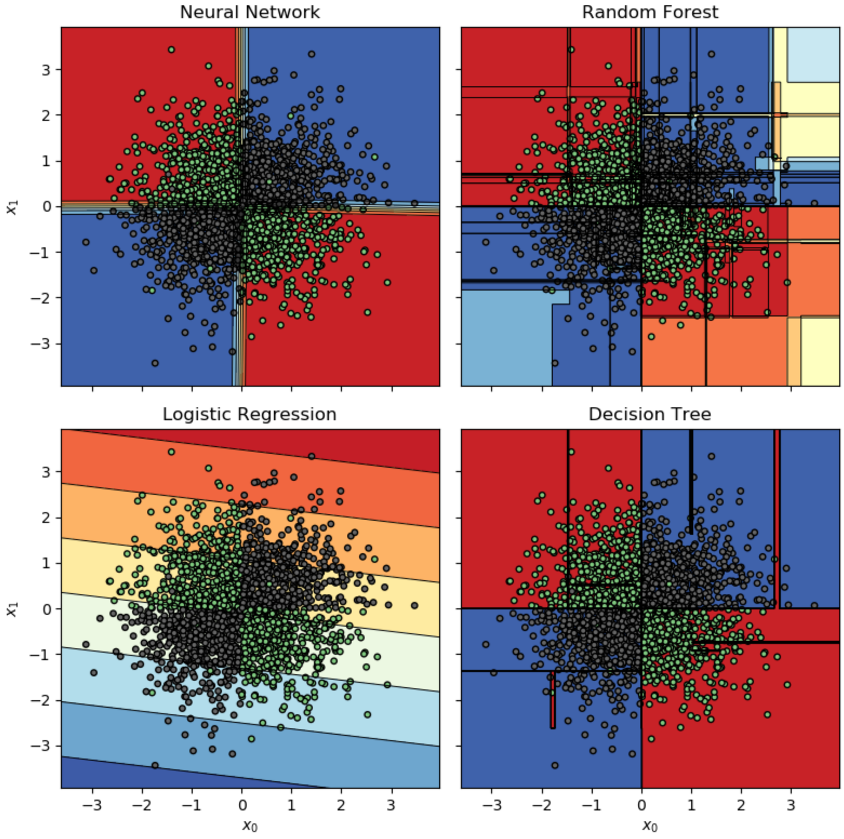


FIGURE 6.2: Learned decision boundaries for different models on $x_0 * x_1$ data. The class of a data point is indicated by its color (green for 0 and grey for 1). The classification of the model is indicated in red (0) and blue (1). For the models that calculate probabilities, these are visualized by colors ranging from red to blue.

fact that a different model is queried, the two aforementioned models seem to be a little more lightweight and efficient.

In line with the poor model performance that could be observed in Table 6.1, both explanation methods could not distinguish between informative and non-informative features for predictions of the logistic regression model trained and tested on both XOR problems. In Table 6.1 we saw that the predictions of the logistic regression models on the numerical XOR and the Boolean XOR data (colored red in the table) were approximately random as there is a 50% chance of guessing the correct label. These models will be excluded from the analyses. In Figures 6.2 and 6.3 we can see that the logistic regression models have not been able to learn the underlying distribution of the data and therefore the explanations will only add noise to the results that follow from our analysis.

6.2 Evaluation by Ground Truth Comparison

Both Kernel SHAP and LIME have been applied to explain 500 predictions for every of the 30 accurate models. The generated explanations of the different models have

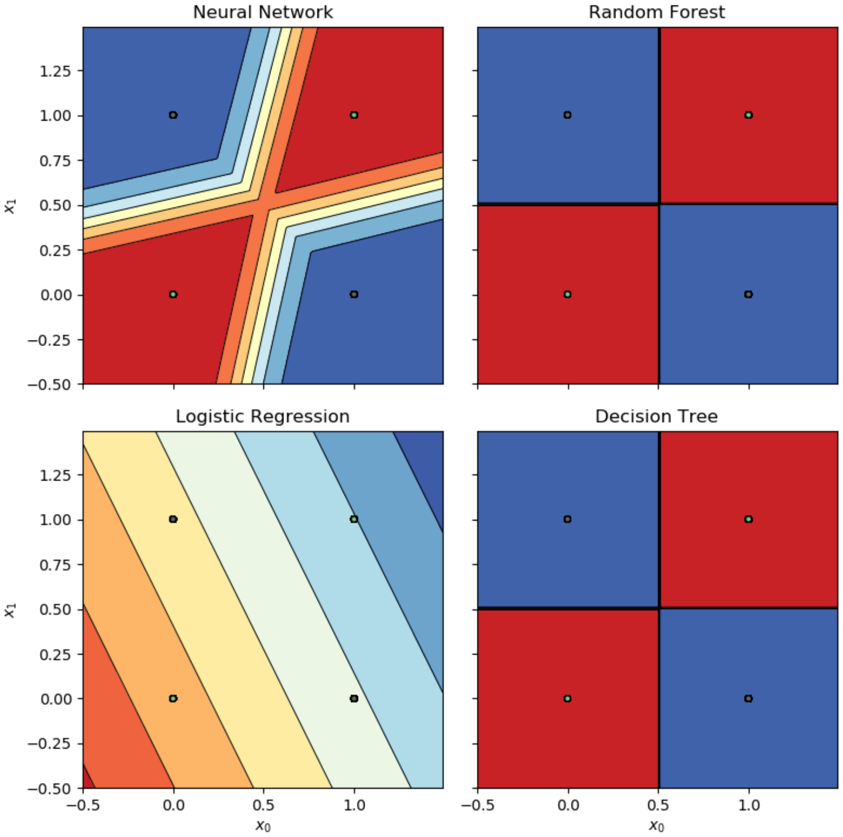


FIGURE 6.3: Learned decision boundaries for different models on $(x_0 \mid x_1)$ & $\neg(x_0$ & $x_1)$ data. The class of a data point is indicated by its color (green for 0 and grey for 1). The classification of the model is indicated in red (0) and blue (1). For the models that calculate probabilities, these are visualized by colors ranging from red to blue.

been compared to the ground truth ranks for the specific dataset they have been trained on.

6.2.1 Kendall's Tau-b Rank Correlation Coefficient

For the different models and datasets, we have computed the Kendall's Tau-b rank correlation coefficients (Kendall, 1948). This coefficient reflects the correlation with respect to the ranks within the explanation and the ground truth rank. The mean rank correlation coefficient (n=500) between the generated explanations and the ground truth ranks are listed in Table 6.2. Whether the difference between the mean Kendall's tau correlation coefficients of LIME and SHAP is significant, has been tested using a paired t-test. The p-value of this test is provided in the last column of the table.

In Table 6.2 we see that for all models and dataset, the correlation value is positive for both explanation methods, indicating strong correlation and therefore explanations that are similar to the ground truth in terms of rank order. A few interesting significant differences between the correlation values of LIME and SHAP can be observed. For the ${x_0}^2 - x_1$ and $x_0 * x_1$ datasets, the correlation value for SHAP is larger than

TABLE 6.2: The mean (n=500) Kendall's tau-b rank correlation coefficient (and corresponding standard errors) between the generated explanations and the ground truth orders. The ground truth vectors are defined in Table 5.3. The p-value in the last column indicates whether the mean correlation for LIME and for SHAP is statistically different. P-values that indicate no significant difference are colored red.

		LIME	**SHAP**	
Datasets	**Model**	Mean (se)	Mean (se)	p-value
Numerical data				
$x_0 + x_1$	NN	0.393 (0.002)	0.396 (0.003)	0.295
	LR	0.387 (0.002)	0.399 (0.002)	< 0.001
	DT	0.440 (0.001)	0.413 (0.002)	< 0.001
	RF	0.413 (0.002)	0.389 (0.004)	< 0.001
${x_0}^2 - x_1$	NN	0.550 (0.006)	0.590 (0.003)	< 0.001
	LR	0.200 (0.006)	0.224 (0.006)	0.004
	DT	0.555 (0.006)	0.624 (0.002)	< 0.001
	RF	0.534 (0.007)	0.587 (0.004)	< 0.001
$x_0 * x_1$	NN	0.323 (0.009)	0.389 (0.003)	< 0.001
	DT	0.320 (0.009)	0.643 (0.004)	< 0.001
	RF	0.220 (0.012)	0.380 (0.004)	< 0.001
ds$_4$ [i]	NN	0.894 (0.000)	0.708 (0.005)	< 0.001
	LR	0.894 (0.000)	0.720 (0.005)	< 0.001
	DT	0.891 (0.001)	0.668 (0.006)	< 0.001
	RF	0.894 (0.000)	0.701 (0.006)	< 0.001
Boolean data				
x_0	NN	0.308 (0.008)	0.315 (0.006)	0.454
	LR	0.291 (0.008)	0.286 (0.004)	0.573
	DT	0.294 (0.009)	0.667 (0.000)	< 0.001
	RF	0.293 (0.008)	0.317 (0.008)	0.039
x_0 & x_1	NN	0.407 (0.002)	0.394 (0.002)	< 0.001
	LR	0.402 (0.002)	0.400 (0.002)	0.447
	DT	0.394 (0.002)	0.651 (0.004)	< 0.001
	RF	0.396 (0.002)	0.392 (0.002)	0.183
XOR [ii]	NN	0.395 (0.002)	0.399 (0.002)	0.230
	DT	0.396 (0.002)	0.399 (0.002)	0.281
	RF	0.400 (0.002)	0.398 (0.002)	0.613
ds$_8$ [iii]	NN	0.882 (0.000)	0.825 (0.004)	< 0.001
	LR	0.882 (0.000)	0.876 (0.001)	< 0.001
	DT	0.882 (0.000)	0.885 (0.004)	0.497
	RF	0.881 (0.000)	0.807 (0.004)	< 0.001

[i] $3 * x_0 + 2 * x_1 + x_2 - 3 * x_3 - 2 * x_4 - x_5$

[ii] $(x_0 \mid x_1)\ \&\ \neg(x_0\ \&\ x_1)$

[iii] $x_0 \mid (x_1\ \&\ x_2) \mid (x_3\ \&\ x_4\ \&\ x_5)$

the value for LIME for all model types. For the ds_4 dataset however, the LIME explanation is more strongly correlated with the ground truth. For the Boolean datasets, we observe much less correlation values that are significantly different. For the decision tree models trained on datasets x_0 and x_0 & x_1, the difference between LIME and SHAP is relatively large when compared to the other model types (in favor of SHAP). For the dataset with cascading influence (ds$_8$), the difference is in favor of LIME, albeit not as strong as for the numerical dataset with cascading influence ds$_4$.

6.2.2 Qualitative Assessment of Normalized Feature Values

To establish fair comparison between the values generated by both methods, we applied a z-transform to the mean feature importance values grouped per model, dataset and explanation method. The normalized FI plots can be found in Figure 6.4 for the numerical datasets and Figure 6.5 for the Boolean datasets respectively.

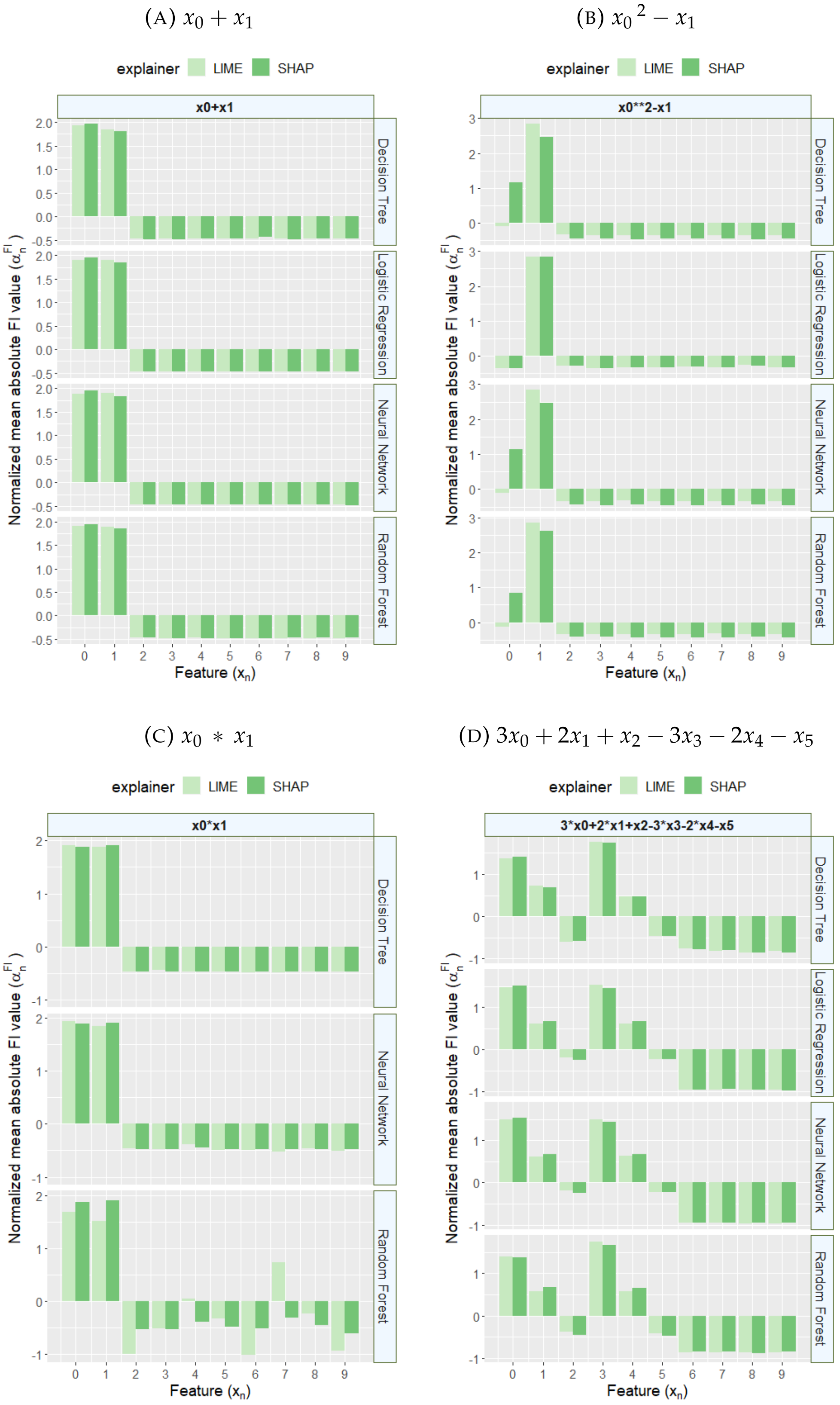
Numerical datasets

In Figure 6.4a, we see that both methods can quite clearly distinguish between the informative (x_0 and x_1) and non-informative features for every model trained to represent the linearly separable plane of the $x_0 + x_1$ function. In Figure 6.4b, the FI values give an indication of why the performance of the Logistic Regression model trained to learn the parabolic plane of function ${x_0}^2 - x_1$, is worse than for the other model types. It makes sense that the importance value of ${x_0}^2$ is smaller than for x_1 when we refer back to the standard normal distribution from which the data points have been drawn (see Section 5.1), with most values being between zero and one. But the linear model seems not to be able to distinguish the quadratic feature x_0 from the non-informative features. This makes the assessment of the explanation methods that x_0 is non-informative for that specific model type, compatible with what we know of the model from the performance metrics.

The models trained on the dataset ds_4 with cascading influence, seem to pick up on the informative features (see Figure 6.4d). The slightly lower performance for the tree-based models on this dataset that has been observed in Table 6.1, seems to be reflected in the feature importance values as well. Both tree-based models seem to have slightly lower feature importance values for the informative features, especially for x_2 and x_5 which are the informative features that are supposed to have the smallest influence on the prediction. Another interesting observation from the plots is the difference between the LIME and SHAP values for some of the models. For Figure 6.4b and to a lesser extent for Figure 6.4c, the SHAP value seems to be larger than the value in the LIME explanations in the case where they differ. SHAP seems to be unique in picking up on feature x_0 being important to the decision-making of the model. For the $x_0 * x_1$ dataset (see Figure 6.4c), LIME seems to be less stable when it comes to determining which features are non-informative for that particular underlying distribution only for the Random Forest model.

Boolean datasets

In Figure 6.5a we see that according to the FI values given by the LIME and SHAP explanations, the models are all able to learn that feature x_0 is the only informative



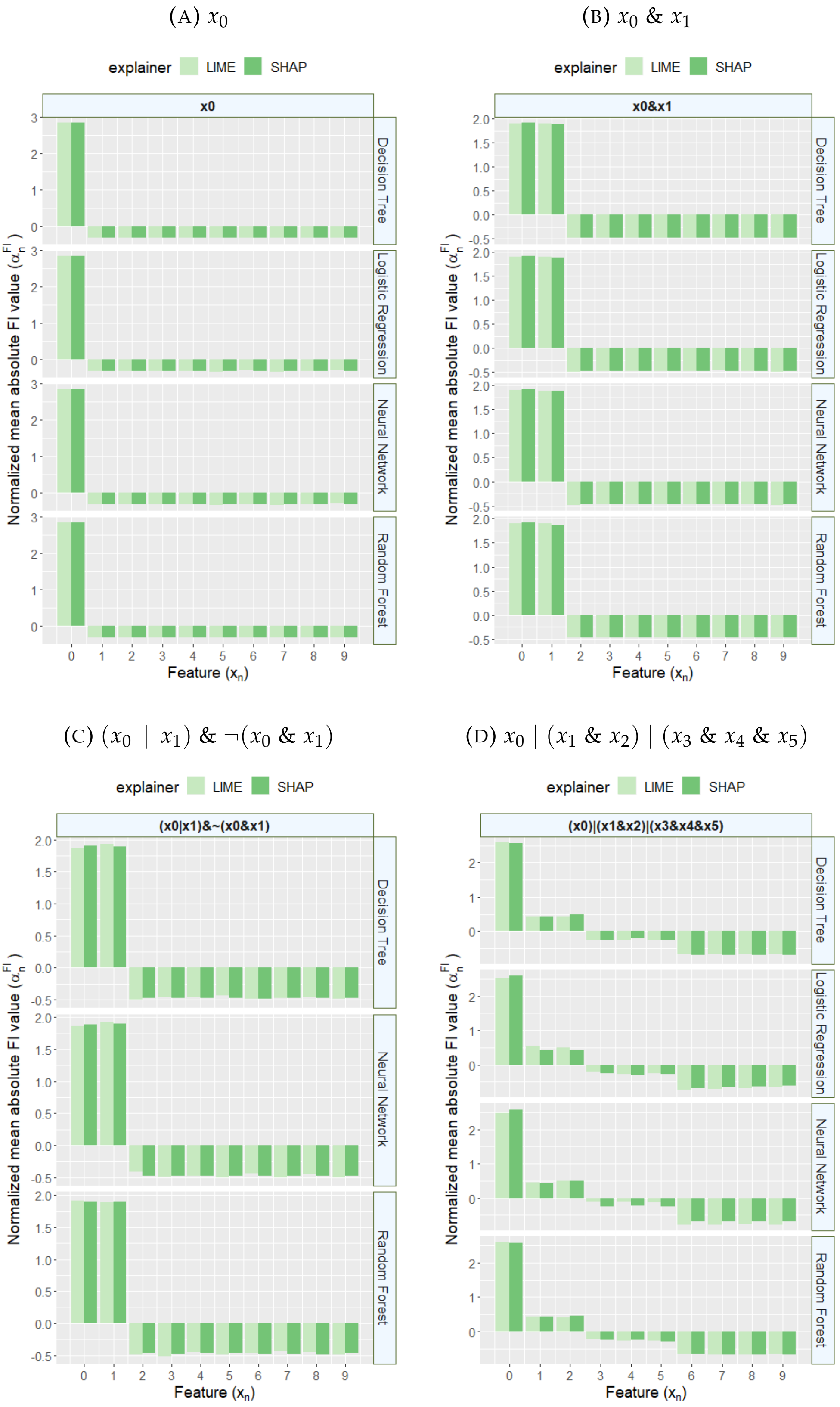
(A) $x_0 + x_1$

(B) $x_0{}^2 - x_1$

(C) $x_0 \, * \, x_1$

(D) $3x_0 + 2x_1 + x_2 - 3x_3 - 2x_4 - x_5$

FIGURE 6.4: Normalized mean (n=500) absolute feature importance explanation values per model, dataset and explanation method for the different models trained and tested on numerical data.



(A) x_0

(B) x_0 & x_1

(C) $(x_0 \mid x_1)$ & $\neg(x_0$ & $x_1)$

(D) $x_0 \mid (x_1$ & $x_2) \mid (x_3$ & x_4 & $x_5)$

FIGURE 6.5: Normalized mean (n=500) absolute feature importance explanation values per model, dataset and explanation method for the different models trained and tested on Boolean data.

feature in the dataset labeled by the projection function. Figure 6.5b shows similar results for the dataset based on the logical conjunction function. For the Boolean XOR data (Figure 6.5c, LIME and SHAP also picked up on the two important features for all three accurate models. Similar to the numerical equivalents, the models trained on the cascading influence data with Boolean features seem to be able to distinguish between informative and non-informative feature as well as between varying magnitudes of influence (Figure 6.5d).

In general, for the models trained on Boolean data, we hardly see any differences between the two explanation methods after normalization. This might be attributed to the technique that is being used for normalization and setting a common scale for both explanation methods. For the Boolean datasets, the variation in importance among the different features in the Boolean datasets is not very large. Therefore, not many explanation values are expected to end up somewhere in the middle of this common scale. For the numerical features, this observation holds to a lesser extent.

For both the numerical and the Boolean datasets, plots of the mean absolute FI values before normalization α_n^{FI} for every feature x_n and every model as computed by LIME and SHAP can be found in Appendix B for further reference, in Figure B.1 and Figure B.2 respectively.

6.3 Intuitive Assessment

The frequencies of the recourse values in Table 6.3 demonstrate that in general, the intuition that changing the most important feature will result in changing the prediction, holds in most cases for both explanation methods. For LIME, this is true for 13130 (87.5%) of the predictions while for SHAP this number is a little higher, with 13568 (90.5%) of the predictions flipping after changing the most important feature.

Recourse value	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
LIME (n=15000)	1045	13130	646	66	85	9	2	7	7	3	0
SHAP (n=15000)	1045	13568	336	29	17	3	0	0	0	2	0

TABLE 6.3: Frequency table of the recourse values over all the different models and datasets per explanation method. A recourse value of zero means the prediction did not change at all and a value of one means the prediction changed after changing the *most important feature* according to the explanation method.

For both explanation methods, the prediction of 1045 instances did not flip after changing all the features. One possible cause for this is the inter-dependency of the different features in the underlying distributions for some of the synthetic datasets. This suspicion is supported by the fact that most of the predictions that did not change, were predictions of instances of Boolean datasets that contain a conjunction (&) in the underlying distribution. This was the case for 888 (85.0%) of the predictions with recourse value zero.

6.3.1 Skewness

From Table 6.3, we have seen that according to our intuitive assessment, LIME and SHAP provide useful explanations in terms of recourse. As both explanations perform well, they also concur with respect to the explanation they provide very often. This leads to the exact same recourse value for those instances in many cases. For 14019 (93.5%) of the predictions made by the accurate models, the recourse value for the explanations generated by LIME and SHAP were the same. The 981 explanations for which the recourse values of LIME and SHAP disagreed are therefore of particular interest for investigating the differences between the explanation methods in the context of our intuitive assessment.

In order to quantitatively assess this difference, the skewness of this distribution of recourse values for all instances where the values did not match has been evaluated for both explanation methods. The Pearson's coefficients of skewness (Joanes and Gill, 1998) (see Equation 5.3) for the distributions were 3.63 for SHAP and 2.55 for LIME respectively. This means that the distribution for the recourse values of SHAP has a more extreme positive skew to the right, meaning that in general the recourse value is lower when we change features according the rank provided by the SHAP explanation.

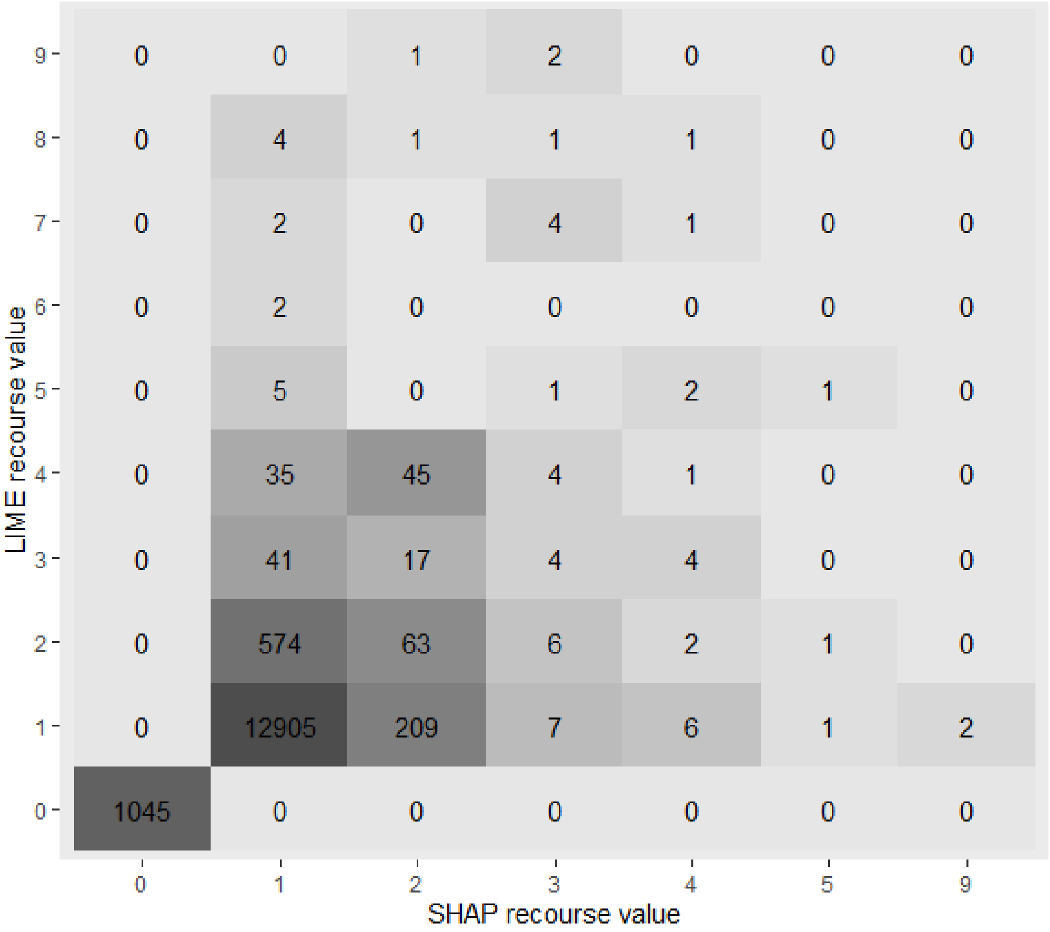
6.3.2 Qualitative Assessment of Recourse Values

In order to qualitatively compare the recourse values of both explanation methods, the frequency of combinations of recourse values for all instances has been visualized on a grid in the categorical heatmap from Figure 6.6. Recourse value combinations that occur more often are depicted by darker tiles in the grid. In this heatmap we observe that for the majority of instances both recourse values are one, which demonstrates the general high quality of the explanations. The second observation is that relatively high recourse values occur more often for LIME explanations. A high recourse value gives an indication of a strong predictive feature not being ranked as (one of) the most important feature by an explanation method. This is depicted by the vertical spread of the dark area, which in this case covers the LIME axis and not so much the SHAP axis.

To visually verify the results that follow from the skewness metric, a histogram of the recourse values for the 981 explanations for which the recourse values did not match has been drawn up in Figure 6.7.

In this histogram, the recourse values of SHAP has a more extreme positive skew to the right, meaning that in general the recourse value is lower when we change features according the rank provided by the SHAP explanation. This aligns with our observation of a more positive skew for the SHAP recourse values from the Pearson's coefficients of skewness. In lign with the heatmap from Figure 6.6, it again becomes clear that relatively high recourse values occur more often for LIME explanations. Important to keep in mind though, is that high recourse values rarely occur and that the histogram represents only a small portion of the explanations. It only serves to zoom in on the rare cases in which the recourse values of the explanation methods do not match.

Note that in contrast with previous chapters, this chapter does not contain a summary. The results from this chapter will be discussed and interpreted in Section 7.1.



LIME recourse value	0	1	2	3	4	5	9
9	0	0	1	2	0	0	0
8	0	4	1	1	1	0	0
7	0	2	0	4	1	0	0
6	0	2	0	0	0	0	0
5	0	5	0	1	2	1	0
4	0	35	45	4	1	0	0
3	0	41	17	4	4	0	0
2	0	574	63	6	2	1	0
1	0	12905	209	7	6	1	2
0	1045	0	0	0	0	0	0

FIGURE 6.6: Categorical heatmap of the recourse value combinations for all instances. The recourse value represents the rank of the feature that made the prediction flip when changing features in order of importance—with this order provided by the explanation method. Recourse value combinations that occur more often are depicted by darker tiles in the grid.

An overview of the main findings from the analyses can be found at the beginning of Section 7.3 as part of the conclusive section.

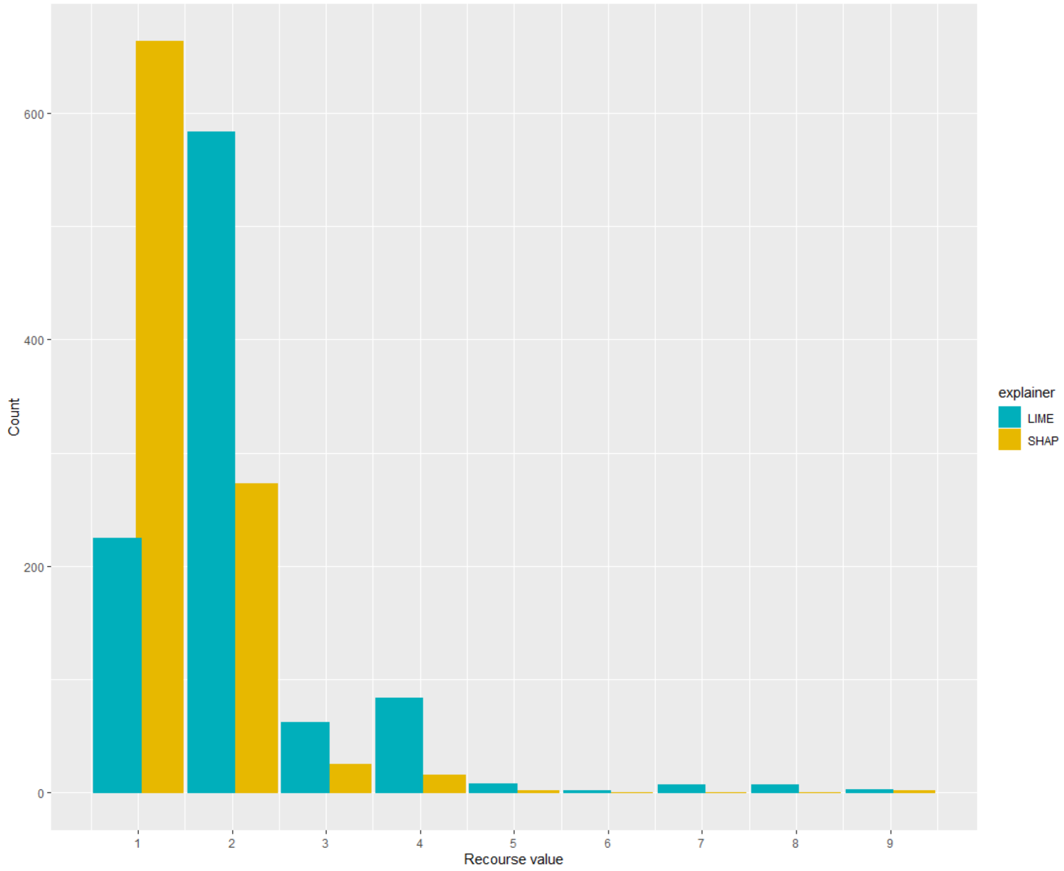


FIGURE 6.7: Histogram of the recourse values per explanation method for all instances of which the recourse values of LIME and SHAP did not match. The recourse value represents the rank of the feature that made the prediction change when changing features in order of importance—with this order provided by an explanation.

Part IV

Discussion and Conclusion

Chapter 7

Discussion and Conclusion

An evaluation framework has been drafted based on the combination of the requirements that follow from the right to explanation (Section 1.2), the definitions of interpretability and explanations from Section 3.1 and the taxonomy of interpretable machine learning (Section 3.2). In this conclusive chapter of the thesis, the added value and relevance of the design and implementation of our explanation framework will be discussed and connected to the literature. Ultimately, possible future work will be outlined, the main findings of the current work will be listed and a conclusion will be drawn.

7.1 Discussion

The framework we built has been tested using datasets with predefined distributions. The evaluation of the explanations using this data with known explanatory structure was twofold. The distributions have served as a ground truth that made objective evaluation of the explanations possible. Moreover, an intuitive assessment of explanations that does not rely on predefined distributions has been proposed and tested. The results of the proposed methodology will be discussed in this section.

7.1.1 Interpretation of the Results

We have been able to match and even clarify regular performance metrics of the different models with the feature importance indications given by both explanation methods. This partly demonstrates the usefulness of the proposed evaluation framework. The results of the ground truth assessment (Section 6.2) indicate that the current methodology helps fulfill the evaluation of the first requirement of automated decisions (see Section 1.2) which is to enable users to understand the reasons—in our case indicated by feature importance—behind a decision. The results of the intuitive assessment (Section 6.3) indicate the usefulness of the current method for evaluating consent to the second requirement of automated decisions (see Section 1.2), which is to enable users to act upon a decision.

For the models that were not able to learn the known explanatory structure, denoted by low accuracy scores in Table 6.1, this decline in model performance can be explained by the explanation values as they align with decision boundary visualization (see Section 6.1). The low performance of the linear Logistic Regression models on the non-linear parabolic dataset and both XOR-datasets was not surprising. The other two models that did not perform well were both tree-based models trained and tested on the numerical dataset with cascading influence (ds_4). Interestingly enough, the other two model types did perform well on this dataset with features of varying importance (as can be seen in Table 6.1). Tree-based algorithms are known to not fit as well on numerical data as they do on Boolean data because they fail to model important information when segmenting the data into different regions needed for the impurity method used for splitting the nodes (Ho, 2002) (see Section 2.2.2). That this is reflected in the numerical dataset with cascading influence was to be expected since this is the dataset with the most variation in terms of the different magnitudes of influence.

Both LIME (Ribeiro, Singh, and Guestrin, 2016) and kernel SHAP (Lundberg and Lee, 2017) are able to distinguish between informative and random features for the models and datasets we used. The explanation values correspond with what we expect from the underlying distributions of the synthetic data. In terms of the ground truth assessment, this is indicated by high rank correlation coefficients between the explanations and the ground truth vectors in Section 6.2.1 (see Table 6.2). The correlation value is positive for both explanation methods for all models, indicating strong correlation and therefore explanations that are in general very similar to the ground truth in terms of rank order. For the ${x_0}^2 - x_1$ and $x_0 * x_1$ datasets, the correlation value for SHAP is larger than the value for LIME for all model types.

The locality of the sampled datasets imposed by the weighting schemes (see Sections 4.1.1 and 4.2.1) has a great influence on the explanations (Laugel et al., 2018). This is probably where the difference between LIME and kernel SHAP is largest, both in terms of fidelity and in terms of the efficiency of the algorithm. The assumption behind LIME that the decision boundary can be approximated by a linear model holds for a infinitely small region around the instance. The user-defined value of the kernel width (see Section 4.1.1) needs to be chosen as to find a balance between locality and efficiency (Ribeiro, Singh, and Guestrin, 2016; Laugel et al., 2018). To a certain extent, the linear assumption of LIME is correct when considering a tiny region of the decision boundary around the instance to be explained. When increasing the size of this region, a linear model might not be powerful enough to accurately approximate the decision boundary of the original model in that region. The SHAP kernel, even though a lot more expensive computationally, circumvents this problem by generating unbiased approximations of the Shapley values.

The results from the rank correlation values are also reflected in the feature importance plots in Section 6.2.2. For the ${x_0}^2 - x_1$ dataset and to a lesser extent for the $x_0 * x_1$ dataset, the SHAP value seems to be more similar to the ground truth than the value in the LIME explanations in the case where they differ. The relatively poor results for LIME on exactly the non-linear datasets, might indicate limited applicability of LIME in scenarios with complex data distributions and models.

For the ds_4 and ds_8 datasets however, the LIME explanation is more strongly correlated with the ground truth. It is however notable that the superior correlation between LIME and the ground truth for the datasets with cascading influence is not reflected in the normalized mean feature importance values. As this is the only case in which both assessment types of the ground truth evaluation do not match, we ought to be wary of interpreting this result, especially since the variance of the LIME explanations is strikingly low (reflected by the standard errors in Table 6.2). The better explanations of LIME for the datasets with cascading influence might be caused by the multi-dimensional (yet still linear) decision boundaries generated by this distribution and the problems we identified with tree-based models on this data.

The results we have discussed above are all also supported by the results (Section 6.3) of intuitive assessment method we proposed in Section 5.2.2. The frequencies of the recourse values demonstrate that in general, the intuition that changing the most important feature will result in changing the prediction, holds in most cases for both explanation methods. For LIME, this is true for 13130 (87.5%) of the predictions while for SHAP this number is a little higher, with 13568 (90.5%) of the predictions flipping after changing the most important feature. The skewness coefficients for the distributions were 3.63 for SHAP and 2.55 for LIME respectively (see Section 6.3.1). This result is also reflected in the categorical heatmap of the conditional distributions in Section 6.3.2. We observe that relatively high recourse values occur more often for LIME explanations than they do for SHAP explanations.

Next to differences in terms of fidelity to the original model, we also observe significant differences between LIME and SHAP when it comes to the time it takes to generate an explanation. For LIME, the mean explanation time is 0.030 seconds while for SHAP the mean explanation time is 13.43 seconds. The difference in efficiency can be attributed to the sampling techniques both methods employ, as explained in Chapter 4.

7.1.2 Connection to the Literature

Automated decision-making models often model human behavior and life trajectories. Even with the use of rich datasets and state-of-the-art machine learning methods however, it has proven to be very difficult to accurately model social patterns (Salganik et al., 2020). Next to suggesting practical limits to the predictability of human behavior in some settings, the work done by Salganik et al. (2020) also justifies the work that has been done in the current study. State-of-the-art machine learning models do not necessarily perform better than simple linear models in this domain. As long as this is the case, it is certainly of importance to understand the reasoning of any automated decision-making system and to be able to reliably evaluate the interpretable machine learning methods that are intended to provide this layer of understanding.

The application of the framework using LIME and kernel SHAP has highlighted some key differences between both methods. As opposed to the current study, the relation between hyper-parameter settings and possible problems with the instability of explanations have been addressed extensively in other work (Zhang et al., 2019; Gosiewska and Biecek, 2019). The general consensus however, matches the patterns that we observed when implementing the framework. Laugel et al. (2018) for example, have already shown that defining the right level of locality defines the quality and relevance of an explanation. Their results have shown that the local models of LIME sometimes approximate the global shape of the black-box decision boundary instead of the local boundary close to the individual instance. Our results tie in with this finding, considering that for linear (global) decision boundaries the variation between the shape and direction of different local decision boundaries is very small compared to complex non-linear decision boundaries. LIME performs well for (partly) linear data distributions as the effect of the global feature influence mitigating the local feature influence is insignificant for those datasets.

Both sampling methods have certain pitfalls when it comes to explaining instances. Kernel SHAP uses the marginal contribution and thereby ignores the dependence between absent and present features in the binary vectors of the samples (see Section 4.2.1), which is needed to ensure that the resulting coefficients are Shapley values (Lundberg and Lee, 2017). If the method was to sample from the conditional distribution, the resulting values would violate the axiom defined by Lundberg and Lee (2017), which says that a feature that does not contribute to the outcome should have a Shapley value of zero. In our study, noise was limited in the synthetic data generation process described in Section 5.1. Yet it has been proved by Gosiewska and Biecek (2019) that this method can suffer from the same problems as other permutation sampling-based approaches since too much weight might be attributed to unlikely instances.

For LIME, the choice of the similarity metric also has an effect on the explanation result. By choosing neighbourhoods of different sizes, the resulting explanations might point in opposite directions. This characteristic threatens the criterion of continuity. For this criterion, Alvarez-Melis and Jaakkola (2018) have defined an assessment method with a qualitative and a quantitative aspect, similar to the methodology for the fidelity criterion used in this current study. The focus of their research however, was to actively develop self-explanatory models for which explainability already plays a role during the learning phase (Melis and Jaakkola, 2018). Zheng, Fernandes, and Prakash (2019) have uncovered that these self-explanatory models are for now still susceptible to adversarial attacks, just like opaque machine learning methods (see Section 1.1).

In this study we were not concerned with developing an interpretable machine learning method ourselves, but merely with establishing and testing a general evaluation framework. The distinctive idea is that evaluations should at least produce expected results in a framework based on synthetic data on which a ground truth assessment can be done. The intention of using synthetic datasets with known explanatory structure was not to accurately represent real-world machine learning problems. The demonstrated usefulness of this approach, is that applying this part of the framework before explanation method deployment can already accurately indicate the performance of the explanation method in terms of fidelity. Synthetic data distributions can be chosen to reflect the expected data distribution in the actual data domain. The intuitive assessment method can then be used after deployment of the explanation model to monitor live evaluation performance, also for real-world machine learning problems. This can be seen as the first step towards evaluating explanations of deployed explanation systems using real-life datasets in an objective manner and adds to the practical applicability of XAI.

7.2 Future Work

Not all of the criteria from the evaluation framework that we drafted in Chapter 3 have been evaluated within the scope of this study. Therefore setting up evaluation procedures for the remaining criteria for local post-hoc explanations (continuity and reliability) would form a natural extension to the current study. One possible methodological starting point for assessing the reliability of explanations—a good explanation should be given for every individual prediction—would be to evaluate the quality of local explanations in relation to the distance of the instance to the original decision boundary. For the criterion of continuity, which says that identical inputs should lead to identical explanations, Alvarez-Melis and Jaakkola (2018) have already defined a quantitative metric—the local Lipschitz continuity value—which can be added to the current methodology. The same holds for adding models from a broader array of machine learning problems, for example based on different data types (e.g. visual and textual) or regression.

The success of the intuitive assessment method, could serve as a stepping stone towards more generic evaluation methods for which predefined distributions are no longer necessary. This move from theoretical evaluation towards a more practical approach would greatly benefit the ad-hoc evaluation of live models and online learning systems. As the intuitive assessment was for a large part based on the contrastive component of an explanation, it is worthwhile to explore counterfactual and example-based explanation methods (Dhurandhar et al., 2018) as a possible addition to the framework of feature importance based explanations.

All the methods that have been used in this study, assume that the features in the data are complete and known beforehand. In practice, systems sometimes also use external reference data and links to other databases to come to accurate decisions. This is not always explicitly communicated with the user. Ideally, models that use these hidden features should also include them in the explanation process. The work by Lakkaraju et al. (2017) uses hidden features in prediction models and can be used as an inspiration for the explanation evaluation of models with hidden features. In addition to the current methodology, the future directions proposed in this section would add aspects to the framework that help increase the completeness of the evaluation of interpretable machine learning methods.

7.3 Conclusion

Automated decision-making models are being deployed in regulated industries such as finance, the judiciary and the government. The use of these models is with good intentions, to speed up and improve the effectiveness of seemingly simple decision-making. However, in this study we have discussed various potentially unwanted side-effects which are hard to monitor because of the opaqueness of the internal reasoning of some machine learning models. Therefore it is important to develop a framework able to not only evaluate the performance of machine learning models themselves but also the explanations that should be provided along with their predictions.

Part I of this study comprised the justification and design of this evaluation framework for automated decision-making models. In order to provide a formal way of assessing interpretable machine learning methods, we combined the extensive research in the social science domain with an analysis of the regulatory requirements posed by the GDPR. Adequate decision explanations should meet the requirements that the user of such a system should be able to understand the reasons behind an automated decision and that the user should be able to act upon the decision—which is to say spot errors or contest the decision (Section 1.2). Interpretable machine learning methods can be used to facilitate this by giving an understanding of how the models make decisions. For this, explanations have been defined as the means to increase model interpretability (Section 3.1.2).

As part of the formation of this framework we taxonomized the research on interpretable machine learning methods along three dimensions: The scope of the explanations, the approach to interpretability and the type of explanation. From this followed a general evaluation framework for automated decision-making explanations that consists of several criteria; fidelity, continuity, reliability, accuracy, robustness and efficiency. The main criterion for local explanations and automated decision-making—and hence the focus of the research done in this study—is fidelity. For explanations of individual predictions, it is essential that the local explanation correctly reflects the original model that it is based on. In Part II, a methodology to assess the framework has been proposed. The framework has been applied by evaluating the two most popular proponents of feature importance based explanations, LIME (Section 4.1.1) and Kernel SHAP (Section 4.2.1), along the criterion of fidelity using synthetic datasets and two complementary assessment types. The combination of both assessments serves as a foundation for evaluating explanations in a general framework. Our results that can be found in Part III of the study, suggest that kernel SHAP is slower but more precise, especially when it comes to non-linear decision boundaries. In cases where the linearity of the decision boundary (also for multi-dimensional data) is well-defined, LIME can be used as a heuristic approximation.

Ultimately, tackling unwanted side-effects of automated decisions calls for a combined effort of machine learning engineers, legislators and psychologists. In the current work, we have addressed a missing piece by creating an objective evaluation framework for interpretable machine learning methods. The combination of legal analysis, cognitive explanation theory and interpretable machine learning proved to be successful for evaluating the explanations of relatively simple machine learning models. It is our hope that the framework and the methodology presented in this study helps facilitate the sustainable integration of automated decision-making models in society.

Appendix A

Decision Boundaries for Datasets with Two Informative Features

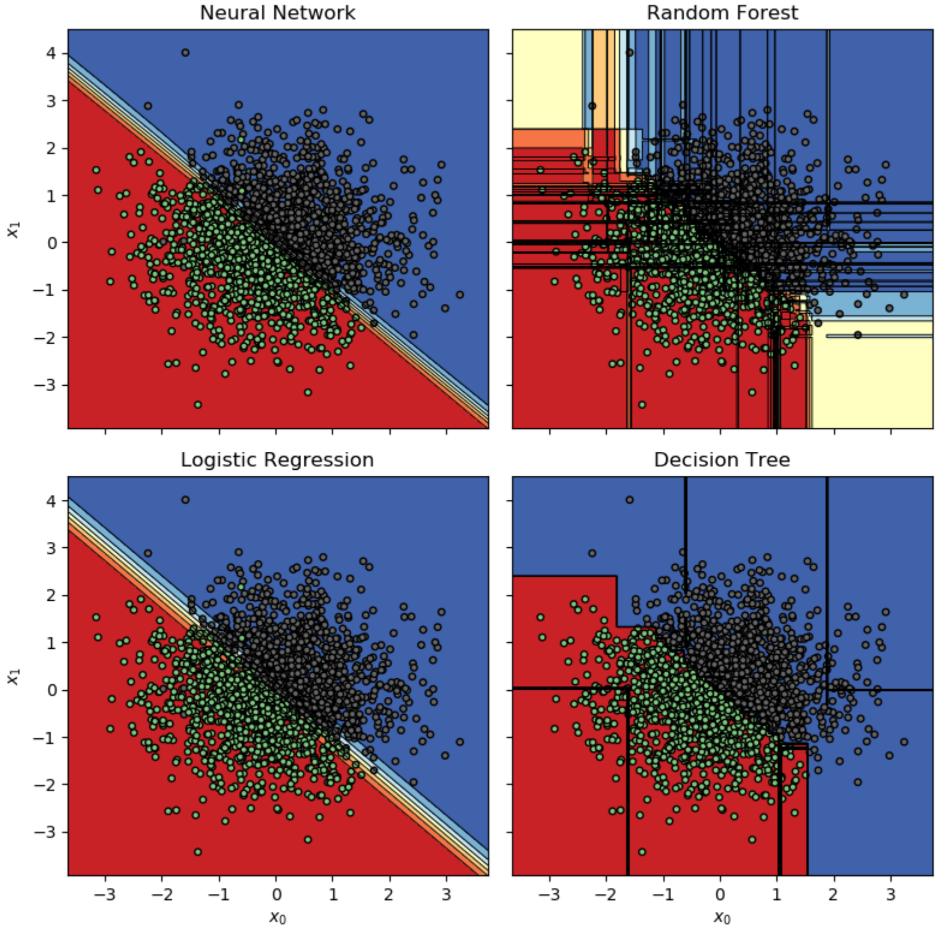


FIGURE A.1: Learned decision boundaries for different models on $x_0 + x_1$ data. The class of a data point is indicated by its color (green for 0 and grey for 1). The classification of the model is indicated in red (0) and blue (1). For the models that calculate probabilities, these are visualized by colors ranging from red to blue.

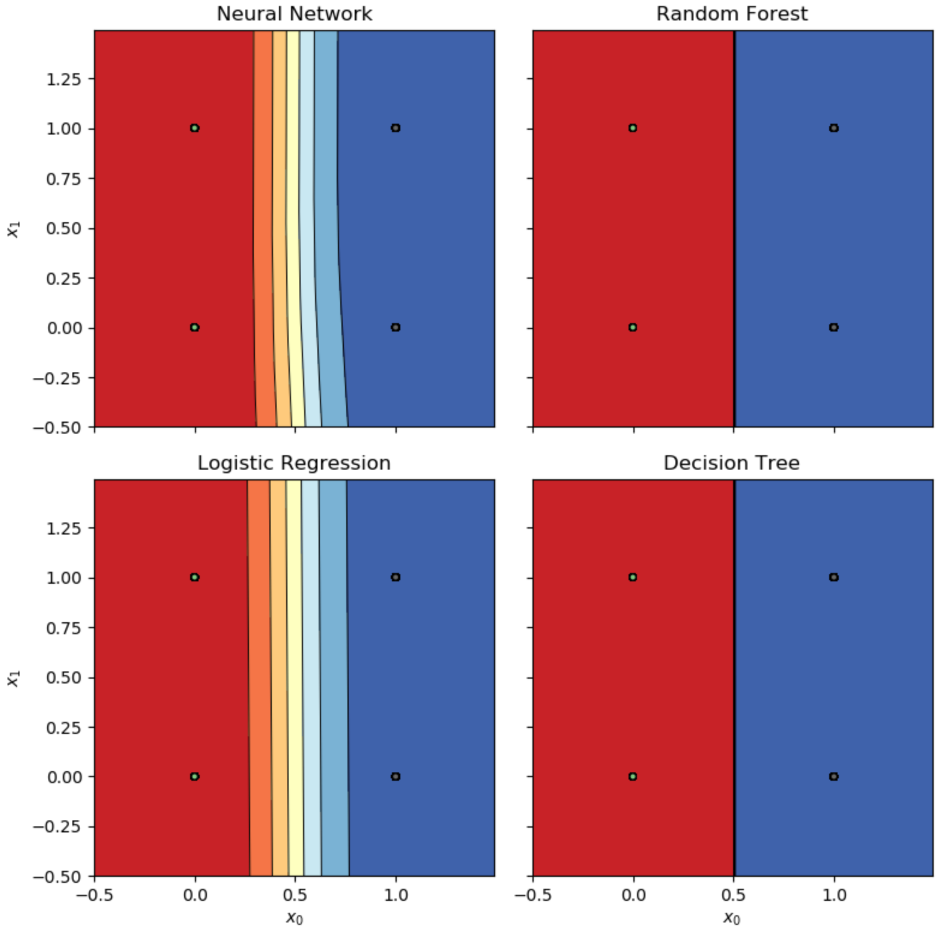


FIGURE A.2: Learned decision boundaries for different models on x_0 data. The class of a data point is indicated by its color (green for 0 and grey for 1). The classification of the model is indicated in red (0) and blue (1). For the models that calculate probabilities, these are visualized by colors ranging from red to blue.

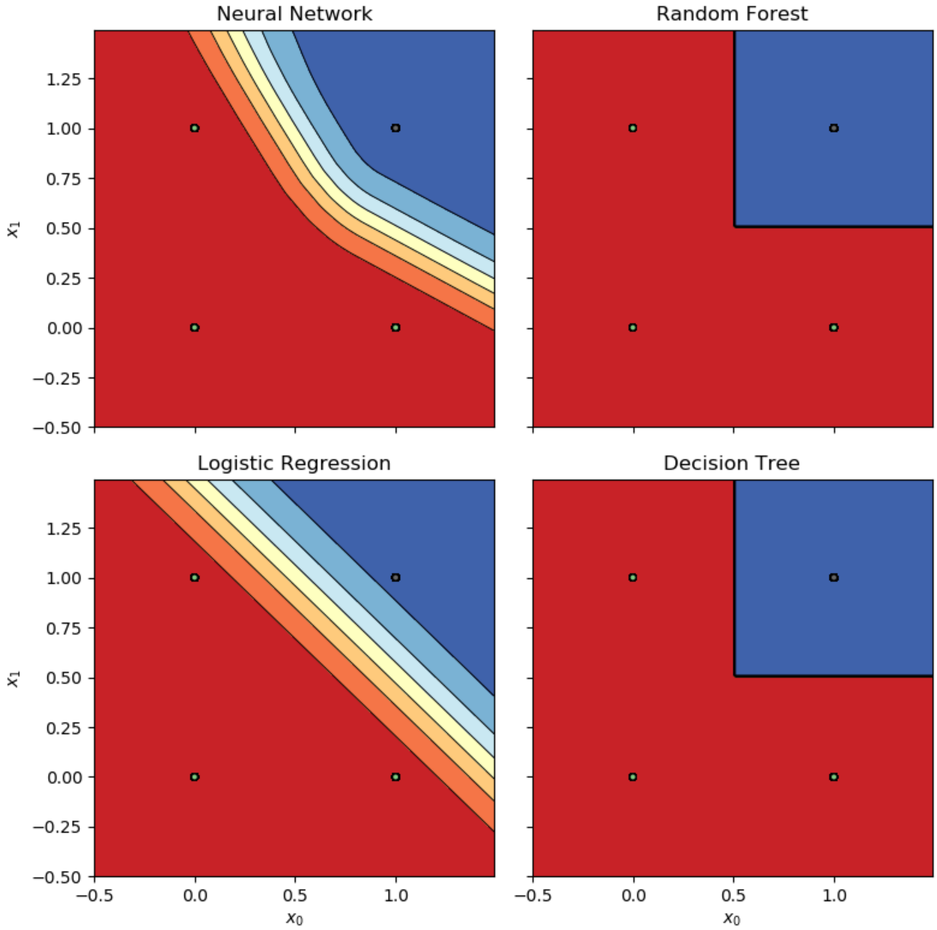


FIGURE A.3: Learned decision boundaries for different models on x_0 & x_1 data. The class of a data point is indicated by its color (green for 0 and grey for 1). The classification of the model is indicated in red (0) and blue (1). For the models that calculate probabilities, these are visualized by colors ranging from red to blue.

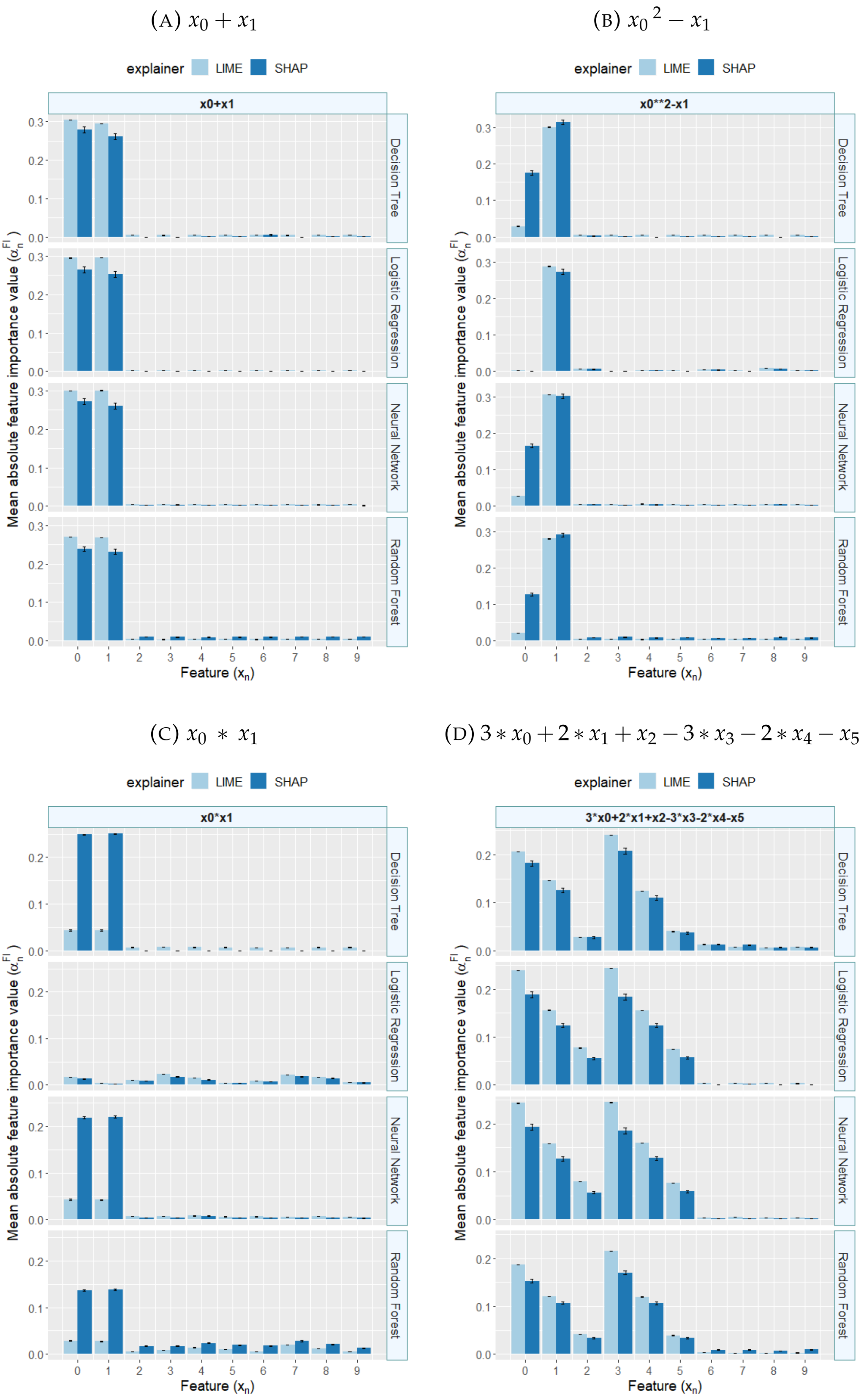
Appendix B

Original Feature Importance Plots for LIME and SHAP on the Synthetic Datasets before Normalization

A visualization of the mean absolute feature importance (FI) values α_n^{FI} for every feature x_n and every model as computed by LIME and SHAP can be found in Figure B.1 and B.2. The vertical bars represent the FI values and error bars denote the standard error of the mean:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}} \tag{B.1}$$

where σ is the standard deviation of the absolute FI values and n is the amount of explanations.

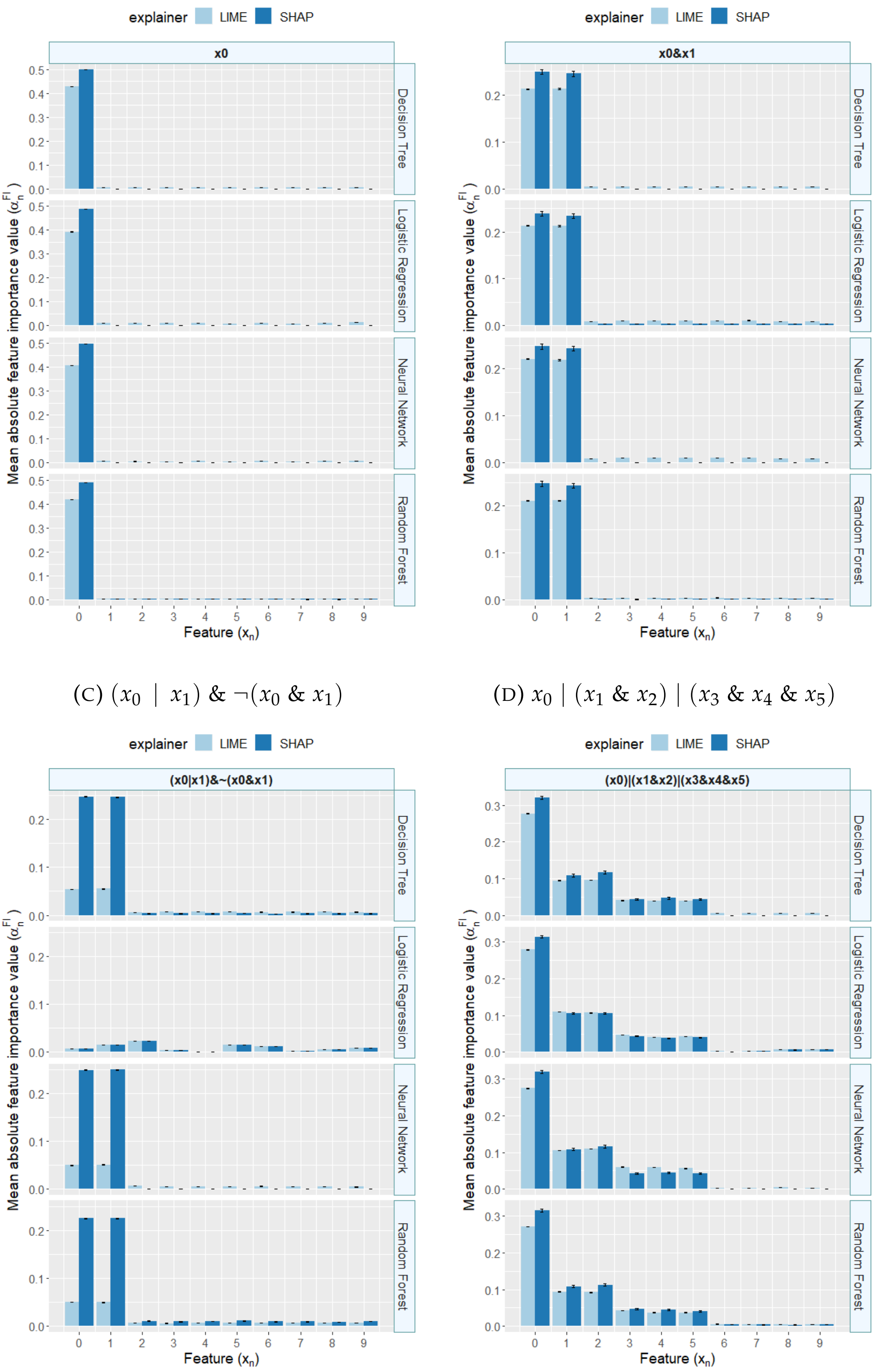

(A) $x_0 + x_1$

(B) ${x_0}^2 - x_1$

(C) $x_0 * x_1$

(D) $3 * x_0 + 2 * x_1 + x_2 - 3 * x_3 - 2 * x_4 - x_5$

FIGURE B.1: Mean absolute feature importance explanation values (n=500) for the different models trained and tested on numerical data.



(A) x_0

(B) x_0 & x_1

(C) $(x_0 \mid x_1)$ & $\neg(x_0$ & $x_1)$

(D) $x_0 \mid (x_1$ & $x_2) \mid (x_3$ & x_4 & $x_5)$

FIGURE B.2: Mean absolute feature importance explanation values (n=500) for the different models trained and tested on Boolean data.

Bibliography

Agrawal, Rakesh, Tomasz Imieliński, and Arun Swami (1993). "Mining association rules between sets of items in large databases". In: *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*. Vol. 22. 2. ACM, pp. 207–216.

AlgorithmWatch (2019). *Automating society: Taking stock of automated decision making in the EU. A report by AlgorithmWatch in cooperation with Bertelsmann Stiftung, supported by the Open Society Foundations*. 1st edition. Available at: https://algorithmwatch.org/wp-content/uploads/2019/02/Automating_Society_Report_2019.pdf. AlgorithmWatch.

Alpaydin, Ethem (2009). *Introduction to machine learning*. MIT press.

Alvarez-Melis, David and Tommi S Jaakkola (2018). "On the robustness of interpretability methods". In: *arXiv preprint arXiv:1806.08049*.

Angwin, Julia et al. (2016). *Machine Bias: there's software used across the country to predict future criminals. And it's biased against blacks, May 2016*. URL: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

Breiman, Leo (2001a). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

– (2001b). "Statistical modeling: The two cultures (with comments and a rejoinder by the author)". In: *Statistical science* 16.3, pp. 199–231.

Breiman, Leo et al. (1984). "Classification and regression trees". In: *Wadsworth Int. Group* 37.15, pp. 237–251.

Caruana, Rich et al. (2015). "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 1721–1730.

Chouldechova, Alexandra (2017). "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments". In: *Big data* 5.2, pp. 153–163.

Cooper, Alan et al. (2004). *The inmates are running the asylum: [Why high-tech products drive us crazy and how to restore the sanity]*. Vol. 2. Sams Indianapolis.

Crawford, Kate (2013). "The hidden biases in big data". In: *Harvard Business Review* 1. URL: http://blogs.hbr.org/2013/04/the-hidden-biases-in-big-data/.

Cybenko, George (1989). "Approximation by superpositions of a sigmoidal function". In: *Mathematics of control, signals and systems* 2.4, pp. 303–314.

Datta, Anupam, Shayak Sen, and Yair Zick (2016). "Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems". In: *2016 IEEE symposium on security and privacy (SP)*. IEEE, pp. 598–617.

De Graaf, Maartje MA and Bertram F Malle (2017). "How people explain action (and autonomous intelligent systems should too)". In: *2017 AAAI Fall Symposium Series*.

Dhurandhar, Amit et al. (2018). "Explanations based on the missing: Towards contrastive explanations with pertinent negatives". In: *Advances in Neural Information Processing Systems*, pp. 592–603.

Doshi-Velez, Finale and Been Kim (2017). “Towards a rigorous science of interpretable machine learning”. In: *arXiv preprint arXiv:1702.08608*.

Edwards, Lilian and Michael Veale (2017). “Slave to the Algorithm: Why a Right to an Explanation Is Probably Not the Remedy You Are Looking for”. In: *Duke L. & Tech. Rev.* 16, p. 18.

Efron, Bradley et al. (2004). “Least angle regression”. In: *The Annals of statistics* 32.2, pp. 407–499.

Freitas, Alex A (2014). “Comprehensible classification models: a position paper”. In: *ACM SIGKDD explorations newsletter* 15.1, pp. 1–10.

Goodman, Bryce and Seth Flaxman (2017). “European Union regulations on algorithmic decision-making and a “right to explanation””. In: *AI magazine* 38.3, pp. 50–57.

Gorissen, Dirk et al. (2010). “A surrogate modeling and adaptive sampling toolbox for computer based design”. In: *Journal of Machine Learning Research* 11.Jul, pp. 2051–2055.

Gosiewska, Alicja and Przemyslaw Biecek (2019). “IBreakDown: Uncertainty of model explanations for non-additive predictive models”. In: *arXiv preprint arXiv:1903.11420*.

Guidotti, Riccardo et al. (2018). “A survey of methods for explaining black box models”. In: *ACM computing surveys (CSUR)* 51.5, p. 93.

Gunning, David (2017). “Explainable artificial intelligence (xai)”. In: *Defense Advanced Research Projects Agency (DARPA), Web* 2nd. URL: https://www.darpa.mil/attachments/XAIProgramUpdate.pdf.

Haasdijk, Evert and Jacqueline Heinerman (2018). “Quantifying selection pressure”. In: *Evolutionary computation* 26.2, pp. 213–235.

Halevy, Alon, Peter Norvig, and Fernando Pereira (2009). “The Unreasonable Effectiveness of Data”. In: *IEEE Intelligent Systems* 24.2, pp. 8–12. ISSN: 1541-1672. DOI: 10.1109/MIS.2009.36. URL: http://dx.doi.org/10.1109/MIS.2009.36.

Hastie, Reid and Nancy Pennington (2000). “13 Explanation-Based Decision Making”. In: *Judgment and decision making: An interdisciplinary reader*, p. 212.

Herman, Bernease (2017). “The promise and peril of human evaluation for model interpretability”. In: *arXiv preprint arXiv:1711.07414*.

Hern, Alex (2018). *European regulators report sharp rise in complaints after GDPR*. URL: https://www.theguardian.com/technology/2018/jun/26/european-regulators\-report-sharp-rise-in-complaints-after-gdpr.

Ho, Tin Kam (2002). “A data complexity analysis of comparative advantages of decision forest constructors”. In: *Pattern Analysis & Applications* 5.2, pp. 102–112.

Hoerl, Arthur E and Robert W Kennard (1970). “Ridge regression: Biased estimation for nonorthogonal problems”. In: *Technometrics* 12.1, pp. 55–67.

Huisman, Charlotte (2019). “Fraudesysteem overheid faalt”. In: *de Volkskrant* 27 juni 2019, pp. 6–7.

Huysmans, Johan et al. (2011). “An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models”. In: *Decision Support Systems* 51.1, pp. 141–154.

Janocha, Katarzyna and Wojciech Marian Czarnecki (2017). “On loss functions for deep neural networks in classification”. In: *arXiv preprint arXiv:1702.05659*.

Joanes, DN and CA Gill (1998). “Comparing measures of sample skewness and kurtosis”. In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 47.1, pp. 183–189.

Josephson, John R and Susan G Josephson (1996). *Abductive inference: Computation, philosophy, technology*. Cambridge University Press.

Kendall, Maurice George (1948). “Rank correlation methods.” In: *Oxford University Press* 5.

Kirkpatrick, Keith (2017). “It’s Not the Algorithm, It’s the Data”. In: *Commun. ACM* 60.2, pp. 21–23. ISSN: 0001-0782. DOI: 10.1145/3022181. URL: http://doi.acm.org/10.1145/3022181.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*, pp. 1097–1105.

Lakkaraju, Himabindu et al. (2017). “The selective labels problem: Evaluating algorithmic predictions in the presence of unobservables”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 275–284.

Latour, Bruno et al. (1999). *Pandora’s hope: essays on the reality of science studies*. Harvard university press.

Laugel, Thibault et al. (2018). “Defining locality for surrogates in post-hoc interpretablity”. In: *arXiv preprint arXiv:1806.07498*.

Lipovetsky, Stan and Michael Conklin (2001). “Analysis of regression in game theory approach”. In: *Applied Stochastic Models in Business and Industry* 17.4, pp. 319–330.

Lipton, Zachary C (2018). “The mythos of model interpretability”. In: *Queue* 16.3, pp. 31–57.

Lundberg, Scott M, Gabriel G Erion, and Su-In Lee (2018). “Consistent individualized feature attribution for tree ensembles”. In: *arXiv preprint arXiv:1802.03888*.

Lundberg, Scott M and Su-In Lee (2017). “A unified approach to interpreting model predictions”. In: *Advances in Neural Information Processing Systems*, pp. 4765–4774.

Malouf, Robert (2002). “A comparison of algorithms for maximum entropy parameter estimation”. In: *proceedings of the 6th conference on Natural language learning-Volume 20*. Association for Computational Linguistics, pp. 1–7.

McCorduck, Pamela (2004). *Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence*. AK Peters Ltd. ISBN: 1568812051.

Melis, David Alvarez and Tommi Jaakkola (2018). “Towards robust interpretability with self-explaining neural networks”. In: *Advances in Neural Information Processing Systems*, pp. 7775–7784.

Miller, Tim (2018). “Explanation in artificial intelligence: Insights from the social sciences”. In: *Artificial Intelligence*.

Miller, Tim, Piers Howe, and Liz Sonenberg (2017). “Explainable AI: Beware of inmates running the asylum or: How I learnt to stop worrying and love the social and behavioural sciences”. In: *arXiv preprint arXiv:1712.00547*.

Nelder, John Ashworth and Robert WM Wedderburn (1972). “Generalized linear models”. In: *Journal of the Royal Statistical Society: Series A (General)* 135.3, pp. 370–384.

Papernot, Nicolas et al. (2017). “Practical black-box attacks against machine learning”. In: *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, pp. 506–519.

Pennington, Nancy and Reid Hastie (1988). “Explanation-based decision making: Effects of memory structure on judgment.” In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 14.3, p. 521.

Quinlan, J. Ross (1986). “Induction of decision trees”. In: *Machine learning* 1.1, pp. 81–106.

Ramachandran, Prajit, Barret Zoph, and Quoc V. Le (2017). “Searching for Activation Functions”. In: *CoRR* abs/1710.05941. arXiv: 1710.05941. URL: http://arxiv.org/abs/1710.05941.

Read, Stephen J and Amy Marcus-Newhall (1993). “Explanatory coherence in social explanations: A parallel distributed processing account.” In: *Journal of Personality and Social Psychology* 65.3, p. 429.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). “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*. ACM, pp. 1135–1144.

Rosenblatt, Frank (1958). “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6, p. 386.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1985). *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science.

Russell, Stuart and Peter Norvig (2009). “The History of Artificial Intelligence”. In: *Artificial Intelligence: A Modern Approach*. 3rd. Upper Saddle River, NJ, USA: Prentice Hall Press. Chap. 1.3, pp. 16–28. ISBN: 0136042597, 9780136042594.

Salganik, Matthew J et al. (2020). “Measuring the predictability of life outcomes with a scientific mass collaboration”. In: *Proceedings of the National Academy of Sciences*.

Samek, Wojciech, Thomas Wiegand, and Klaus-Robert Müller (2017). “Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models”. In: *arXiv preprint arXiv:1708.08296*.

Samuel, A. L. (1959). “Some Studies in Machine Learning Using the Game of Checkers”. In: *IBM Journal of Research and Development* 3.3, pp. 210–229. DOI: 10.1147/rd.33.0210.

Shapley, Lloyd S (1953). “A value for n-person games”. In: *Contributions to the Theory of Games* 2.28, pp. 307–317.

Sharkey, Noel (2019). *‘Not enough control on algorithms’. Video interview Noel Sharkey*. URL: https://www2.deloitte.com/nl/nl/pages/innovatie/artikelen/not-enough-control-on-algorithms.html.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman (2013). “Deep inside convolutional networks: Visualising image classification models and saliency maps”. In: *arXiv preprint arXiv:1312.6034*.

Štrumbelj, Erik and Igor Kononenko (2014). “Explaining prediction models and individual predictions with feature contributions”. In: *Knowledge and information systems* 41.3, pp. 647–665.

Stubbs, Kristen, Pamela J Hinds, and David Wettergreen (2007). “Autonomy and common ground in human-robot interaction: A field study”. In: *IEEE Intelligent Systems* 22.2, pp. 42–50.

Tibshirani, Robert (1996). “Regression shrinkage and selection via the lasso”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1, pp. 267–288.

Wachter, Sandra, Brent Mittelstadt, and Luciano Floridi (2017). “Why a right to explanation of automated decision-making does not exist in the general data protection regulation”. In: *International Data Privacy Law* 7.2, pp. 76–99.

Wachter, Sandra, Brent Mittelstadt, and Chris Russell (2017). “Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GPDR”. In: *Harv. JL & Tech.* 31, p. 841.

Zhang, Yujia et al. (2019). “why should you trust my explanation?” understanding uncertainty in lime explanations”. In: *arXiv preprint arXiv:1904.12991*.

Zheng, Haizhong, Earlence Fernandes, and Atul Prakash (2019). “Analyzing the Interpretability Robustness of Self-Explaining Models”. In: *arXiv preprint arXiv:1905.12429*.