University of Augsburg,
Technical University of Munich,
University of Bayreuth

A Master Thesis (M.Sc. with hon.)
in Finance and Information Management on:

An Analysis of Model-Agnostic Interpretability Machine Learning Techniques

submitted on the 31^{st} of July, 2020

by Philipp Knöpfle
(Matriculation number: 1280130)
Bitschlinstr. 19
86150 Augsburg
knoepflephilipp@gmail.com

Primary advisor: Prof. Dr. Yarema Okhrin
Chair for Statistics
Faculty of Business and Economics
University of Augsburg

Contents

List of Figures

List of Tables

List of Acronyms

AI ..... Artificial intelligence

ALE ..... Accumulated local effect

GDPR ..... General Data Protection Regulation

ICE ..... Individual conditional expectation

LIME ..... Local interpretable model-agnostic explanation

ML ..... Machine learning

PDP ..... Partial dependence plot

St. Dev. ..... Standard deviation

SVM ..... Support vector machine

RSS ..... Residual sum of squares

SSE ..... Squared sum of the errors

SST ..... Squared sum of the total variance

Abstract

The aim of this scientific work is to introduce the reader into the field of interpretable ML. In this thesis we, therefore, discuss interpretability in ML from a holistic perspective and showcase several types of interpretable ML models. Interpretability in ML becomes relevant, when there is an incompleteness in the ML task's problem formalization (Doshi-Velez and Kim, 2017). Since there is no clear consensus on a formal definition of interpretability in ML, we develop our own understanding of the term interpretability and cognates by reviewing the literature. We substantiate this definition by describing several exemplified personas, characteristics, and classification schemes of interpretability in ML. Moreover, we explain two intrinsically interpretive models and five model-agnostic interpretability techniques for regression and classification tasks. For each technique we describe the theoretical background, showcase an empirical example, and discuss its respective advantages and disadvantages. W.r.t. intrinsically interpretive ML models, we discuss linear regression and decision trees for classification. For model-agnostic interpretability techniques we discuss the following methods: partial dependence plots of Friedman (2001), individual conditional expectation curves of Goldstein et al. (2015), Apley's (2016) accumulated local effects, global surrogate models in Molnar (2020), and local interpretable model-agnostic explanations of Ribeiro et al. (2016b). In a discussion section we give recommendation on how to include these interpretable models in a researcher-oriented ML workflow.

"The real risk with artificial intelligence isn't malice but competence."

— Hawking (2018)

1 Introduction

1.1 Research Context

Over the last decades Machine Learning (ML) research has focused on the creation and advancement of efficient and accurate learning algorithms (Alpaydin, 2020; Jordan and Mitchell, 2015; Lipton and Steinhardt, 2019; Mitchell, 2006). Investigations about ML's current impact, its recent accomplishments, as well as its future potential and repercussions permeate an extensive array of scientific fields, e.g. biology, economics, law, medicine, philosophy, psychology, and social sciences.[1] According to the 2019 Stanford AI Index Report, the volume of peer-reviewed AI papers has grown by more than 300% between 1998-2018 accounting for approximately 3% of all peer-reviewed journal and 9% of published conference publications (Raymond et al., 2019). Recently, an ever expanding substantial segment of the ML literature started to analyse the distinct role interpretability has in ML.[2] In particular, its aim is to examine the phenomena of interpretability and explainability in ML, to foster an understanding of interpretability in the scholarship, to create new more interpretable ML algorithms, and to develop interpretability techniques in order to make existing ML models more transparent.

From a practical perspective the amount of ML applications is permeating our lives more and more. Recommender systems tell us which clothes we wear best. Our shopping needs are anticipated by online retail companies. Autonomous cars are already driving on our highways and soon ML might dictate what food is best for our health.[3] In a McKinsey global survey with over $2,000$ executives consulted, 47% report to have embedded at least one ML capability in their standard business processes, while 71% expect ML investments to increase significantly in the next years (McKinsey, 2018a). Additionally, as a further McKinsey report states, AI could potentially increase global economic output by $13 trillion until 2030 making up an annual average global gross domestic product contribution of 1.2% (McKinsey, 2018b).[4] ML truly stands out as the transformation-bolstering technology of our age.

[1] For reviews on biology see Hunter et al. (1993), Tarca et al. (2007); for economics see Athey (2018), Mullainathan and Spiess (2017); for law see Lehr and Ohm (2017), Surden (2014); for medicine see Deo (2015), Obermeyer and Emanuel (2016); for philosophy see Anderson and Anderson (2011), Thagard (1990); for psychology see Dwyer et al. (2018), Harlow and Oswald (2016); and for social sciences see Hindman (2015), Lazer et al. (2009).

[2] For literature reviews on the topic, see among others Adadi and Berrada (2018), Arrieta et al. (2020), Guidotti et al. (2018), Mueller et al. (2019), and Tjoa and Guan (2019).

[3] For an assortment of academic publishings about ML and fashion see Hsiao et al. (2019), Kang et al. (2017), Sanchis-Ojeda et al. (2016), and Thomassey and Zeng (2018). See Lee (2017) for an anticipatory shipping model and see Spiegel et al. (2012) for a patent on ML anticipatory package shipping. See Alonso Raposo et al. (2018) and U.S. Department of Transportation (2020) for regulatory assessments of the current situation of autonomous driving in the European Union and United States respectively. For literature on personalized nutrition via ML see Nunes-Alves (2016), Sonnenburg and Sonnenburg (2015), and von Schwartzenberg and Turnbaugh (2015).

[4] As a comparison, the steam engine's productivity, arguably the principal driving force behind the First Industrial Revolution, reached a peak 0.41% contribution to productivity during the high phase of British industrialization between 1850-1870 (Crafts, 2004). ML could be almost thrice as productivity-enhancing.

1.2 Central Problem

There is, however, one paramount encumbrance when it comes to the dispersion of ML technologies. With more and more AI applications penetrating our daily lives, we slowly start transferring our decision-making authority to ML algorithms (Nokelainen et al., 2018; Varshney, 2016). Hence, we should ask ourselves, if we actually understand ML algorithms and models enough to entrust them with the increasingly important decisions they make for us (Ribeiro et al., 2016b). As stated in the epigraph, when asked about the future of AI, theoretical physicist Stephen Hawking emphasized the central role that our own competence will play in ML's and AI's development. Are we competent enough to understand the ML models which we create today? To what extent can we comprehend why a ML algorithm chose alternative A over B? How can we interpret a ML model, its implicitly learned knowledge, and its decisions? Are there tools which could help us?

A cursory look into the ML literature reveals that many stakeholder groups in ML struggle with the interpretability of ML models and a definition thereof (Preece et al., 2018). Not only the constantly increasing demanded technical knowledge but also the inherent obfuscation of ML model structure motivated by model performance improvements constitute major barriers towards model understanding (Lipton and Steinhardt, 2019). Furthermore, with the recent successes in the fields of deep learning and reinforcement learning, it is only plausible to assume that ML algorithms will get exponentially more complex in the future (Lipton and Steinhardt, 2019).[5] This is especially problematic for if the model's internal logic is hidden to its user base, they cannot validate, interpret, or understand the model's rationale and more importantly its - in many cases influential - decisions. It is becoming increasingly more difficult to acquire a deeper understanding of a ML model's behaviour and in particular how different features affect the model predictions. Only input and output and not the internal workings are observable by the de facto majority of ML application users. The general public and governments express their concerns by frequently portraying ML models as "opaque" and "black boxes" (Castelvecchi, 2016) or even "black art" (Domingos, 2012) systems.

More and more academic research shows that not being able to perceive a system's core functions and fundamental operating principles may lead to a loss of trust in the system, higher levels of suspicion, and lower levels of the system's general acceptance.[6] Multiple studies have shown that the ability to explain decisions is the most desirable feature of decision support systems (Gregor and Benbasat, 1999; Teach and Shortliffe, 1981; Ye and Johnson, 1995). Further empirical studies in the field of recommender systems investigating the importance of explanations to users persistently show that explanations significantly raise user confidence and trust (Bilgic and Mooney, 2005; Herlocker et al., 2000; Pu and Chen, 2007; Sinha and Swearingen, 2002). The innate intransparency of most ML models, thus, poses a major factor of uncertainty propagation

[5] For recent accomplishments in deep learning see Goodfellow et al. (2014), Ha and Eck (2018), LeCun et al. (2015), and Suwajanakorn et al. (2017); and for reinforcement learning see Littman (2015), Neftci and Averbeck (2019), Silver et al. (2018), and Vinyals et al. (2019).

[6] See Braithwaite and Levi (1998), Cook (2001), Kramer and Cook (2004), and Kramer and Tyler (1995). While this work mainly comes from a well established corpus in the fields of organizational psychology and behavioural economics, it seems straightforward to argue that a ML model is just another decision-making entity in a model thereof which replaces agents and with which other agents can interact.

and a paramount obstruction of ML's future development.

What is more, from a ML engineering perspective improving interpretability appears only interesting when it helps improving the understanding of the system such that it can be optimized w.r.t. to a designated metric, e.g. accuracy, running time, etc. (Tomsett et al., 2018). Companies working with proprietary ML applications face the conflict - well known in economics - of disclosing their technological artefact and risking their competitive business advantage versus not disclosing it and suffering economic consequences as well as potentially incurring large societal repercussions in the form of obstructed technological potential (Scotchmer, 1991; Wright, 1983). Most companies simply do not have the incentive to disclose their ML models and the knowledge therein. In Europe this is certainly also due to the inconclusive present situation of the European Patent Law on Artificial Intelligence (AI) and ML which currently fails to protect patentable and disclosable ML inventions.[7]

Indeed, companies profit considerably from the intellectual property contingent on the black box characteristic of their ML algorithms (Rudin, 2019). They only choose to open their ML black box to the public when they experience external pressure, because of notable occurrences such as algorithmic failure to comply with ethics or social norms.[8] Opacity is often viewed as essential in protecting intellectual property, but this rationale is at odds with the requirements of many domains in which ML models are used which include among others public health or safety. ML engineers should not only be interested in the accuracy of their predictions but also in the transparency of their model. They should be invested in the human understandability of the model's decision as well as the full disclosure of the model's decision process.

The impact of the negative consequences on society and the economy caused by the lack of interpretability in ML are a largely uncharted territory in economic and associated research. The very common ML practice of "explorimentation", i.e. fine-tuning a model to see what performs best, may be a pragmatic and appropriate approach for the early stages of development but does not aid in establishing scientific progress in the field of ML and associated areas. Therefore, an increasing voice in the ML community demands more interpretability of ML models, see Adadi and Berrada (2018); Arrieta et al. (2020); Bibal and Frénay (2016); Doshi-Velez and Kim (2017); Guidotti et al. (2018); Lipton (2018); Mueller et al. (2019); Ribeiro et al. (2016a); Tjoa and Guan (2019) among others. For the sake of the research progress and the transparency of scientific discoveries it is essential to promote interpretability in ML and make it accessible to a

[7]For a discussion on this topic see Shemtov (2019) and Yanisky-Ravid and Liu (2017). At present, AI and ML are treated as "mathematical methods" under Art. 52(2a) in the European Patent Convention, which are as such not patentable unless they are tied to the control of a technical system or process, therefore, gaining technical character and moving ML inventions into the domain of a patentable inventions. For example, a new ML algorithm cannot be patented per se, while a training method that causes an existing ML algorithm to converge quicker to its optimum may be recognized as solving a technical problem and, thus, qualify for European patent protection. Even though, a report commissioned by the European Patent Office (Shemtov, 2019) states that a person designing a ML system with the ML system's invention in mind is an inventor and, thus, her work should be legally protected, there is currently no patent legislation which can protect the pre-trained and proprietary knowledge of a ML model initiated by an inventor in any specific area of application in the European Union.

[8]See Obermeyer et al. (2019) where the authors identify systematic discrimination against black patients by a US health care system algorithm and see Datta et al. (2015) for an example of the Google Ads algorithm systematically hiding advertisements of high paying job openings from women.

large audience. Hence, when discussing recent advancements in ML today, one necessarily has to incorporate interpretability, next to performance-related factors, because the former constitutes a major factor of importance. To put it plainly: Interpretability and explanations in ML are not a luxury or curious side feature but are becoming more and more a necessity and crucial factor which has to be considered throughout the entire ML analysis.

1.3 Literature Review

Even though the domain of ML research is a very active and open field, its community has only recently started to explore systems considering non-performance related criteria such as, for instance, fairness (Binns, 2018; Chouldechova and G'Sell, 2017; Corbett-Davies and Goel, 2018; Friedler et al., 2019), reproducability (Gundersen et al., 2018; Hutson, 2018; Patil et al., 2016), providing the right to explanation (Goodman and Flaxman, 2017), security (Barreno et al., 2010; Madry et al., 2017; Otte, 2013), or preventing technical debt (Sculley et al., 2015). The importance of interpretability and explainability in ML has been emphasized in numerous publications over the past decade (Arrieta et al., 2020; Bibal and Frénay, 2016; Doshi-Velez and Kim, 2017; Guidotti et al., 2018; Lipton, 2018; Ribeiro et al., 2016a; Tjoa and Guan, 2019; Van Belle and Lisboa, 2013; Vellido et al., 2012). Today, there is a large consensus between academia and most private company research to make ML algorithms generally more interpretable and more explainable to every stakeholder across the ML application user life cycle. This has resulted in an expanding corpus of academic literature in the interpretability and ML nexus, also labelled as the field of "Explainable Artificial Intelligence" or short "XAI" (Arrieta et al., 2020). Roughly speaking, this field of literature's main objectives can be separated into three categories: 1.) analyse interpretability and explainability in ML from a conceptual perspective, 2.) to create interpretable ML algorithms, and 3.) to develop model-specific and model-agnostic interpretability techniques to make existing ML models more transparent and their knowledge more accessible to a broader user base. Numerous interpretability approaches of the latter emerge almost weekly. According to Du et al. (2019) and Guidotti et al. (2018) these interpretability techniques can be broadly separated into four categories, which we will explain in the following.

First, one can replace the black box with "transparent" box design. That is, one can ex-ante use an intrinsically interpretive ML algorithm, such as linear regression or decision trees, as a substitute model for the original one. Second, one can use model-agnostic interpretability techniques to derive a global explanation which gives an understanding about the entire model's behaviour and knowledge. For instance, one can train a "surrogate" model on the training data observations and corresponding predictions of an initial ML model to approximate the effects in the black box.[9] Third, one can find an explanation for a specific example observation, i.e. a local explanation for a specific data instance. Fourth, one can examine the ML algorithm itself w.r.t. interpretability. This includes finding a local or global inherent explanation by analysing the algorithm and its internal model structure or by finding techniques to make a specific

[9]The idea behind surrogate models to approximate physically and computationally more expensive models with affordable surrogate models comes originally from the field of engineering (Queipo et al., 2005). For an application in the area of ML see Dasari et al. (2019).

algorithm itself more interpretable. While the latter approach requires a significant amount of algorithm-specific knowledge, alternative two and three only assume familiarity with certain model-agnostic interpretability techniques. Method one demands proficiency in intrinsically interpretive ML models.

Since its inception the interpretability and ML nexus has never been static, but has always exhibited a state of flux. With numerous interpretability techniques emerging almost monthly, only a small fraction find traction in the scientific community. According to Doshi-Velez and Kim (2017) and Murdoch et al. (2019), these techniques usually vary in their methodological complexity and their interpretive practicability. Furthermore, the conceptualization of interpretability and associated terms, such as explainability, is still a hardly resolved issue in the literature (Lipton, 2018). Therefore, interpretability techniques also regularly differ in their fundamental understanding of interpretability and explainability. As a consequence of this, we observe a chaotic cluster of the concept of interpretability in ML and, hence, an unstructured complex of interpretability approaches. What is more, since there is no uniform panacea technique to cure model intransparency (Rudin, 2019), it is plausible to assume that not a single interpretability technique can mitigate the black box problem but a set of interpretability techniques can complement each other and, thus, offer a transparent and intelligible look into the black box. Albeit various literature reviews attempting to summarize and consolidate different interpretability techniques and their understanding of interpretability (Adadi and Berrada, 2018; Arrieta et al., 2020; Carvalho et al., 2019; Guidotti et al., 2018; Molnar, 2020; Mueller et al., 2019; Tjoa and Guan, 2019), the practical success and evaluation of such techniques often remains uncharted. In addition, the relationship between techniques continues to be investigated. It is unclear if and how certain interpretability techniques may complement or substitute each other.

In conclusion, it can be stated that an observer who is interested in using interpretability techniques faces a quite challenging task when selecting the appropriate method or model in a specific context, application, or for a particular target audience. Therefore, ML models in many cases remain a black box despite there being a plethora of knowledgeable interpretability techniques. To our knowledge, there is no academic literature which provides a comprehensive academic review and introduction of the concept of interpretable ML and ML interpretability techniques w.r.t. general user practicability and accessibility. We hope to provide remedy to this within this thesis.

1.4 Research Objectives

Set against the above-mentioned background, the aim of this master thesis is to clarify the understanding of interpretability in ML, to provide a comprehensive technical review of recent and popular interpretability techniques, as well as to discuss how to include these techniques in a researcher-oriented ML workflow[10]. Hence, we define the following central research objective:

[10]The term researcher-oriented simply designates a workflow in which other informative criteria than ML accuracy and algorithm efficiency are also considered.

Develop an understanding of interpretability in Machine Learning and identify model-agnostic interpretability techniques in the Machine Learning literature which facilitate the general intelligibility and transparency of Machine Learning algorithms, establish how they function, and determine how to integrate these techniques into a researcher-oriented workflow.

We decompose our central research objective so as to answer the central research questions which it implies. This separation allows us to identify the constituents and subsidiary objectives assisting the resolution of our central research question. We define the subsequent subsidiary objectives.

O1: Define Machine Learning and associated terminology

In this primary subsidiary objective the goal is to derive a clear understanding of ML and relevant terminology which is necessary to facilitate a differentiation between the plethora of existing ML and interpretability techniques. A thorough perception of the nature and goals of ML systems is necessary to delineate the role of interpretability therein. A clear definition of ML is mandatory which is why we first discuss definitions of ML. During this, we want to foster the reader's conceptual understanding of ML which is unfortunately, due to the technical nature of the field, a sometimes neglected perspective (Lipton and Steinhardt, 2019). Yet, it is one we require for discussions vis-à-vis qualitative aspects of ML such as interpretability. Eventually, we introduce necessary ML terminology which is required for conceiving the following interpretability sections when it comes to discussions of how interpretability factors in ML architecture. We assume the reader to be familiar with basic technical ML concepts, such as test-train splits, variance-bias tradeoffs, general model building, evaluation practices, and feature engineering.[11] Objective O1 results in the following two subsidiary objectives.

(a) Define Machine Learning from a conceptual perspective

(b) Define Machine Learning terminology necessary for understanding interpretability discussions

O2: Establish a systematic understanding of interpretability and explainability in Machine Learning

In this subsidiary objective the aim is to unify the at first seemingly contradicting and complex concepts of ML and interpretability. Hence, we motivate and analyse the central phenomenon of interpretability and associated concepts in ML.[12] Interpretability is a demand-side driven aspect of ML (Doshi-Velez and Kim, 2017). Therefore, it is also important to investigate stakeholders of interpretable ML and their requirements profile to derive important interpretability characteristics and desiderata. Thereupon, we explain

[11] See Hastie et al. (2009), James et al. (2013), and Kuhn and Johnson (2013) for an introduction with a focus on statistical learning or the textbooks mentioned in chapter 2.1.

[12] The focus in our analysis is on interpretability and explanations in ML and not the general science behind giving explanations. Brown (2013) provide an extensive review of explanations in social science. See Bunge (1998) for a review of explanations in the philosophy of science.

dimensions of interpretability to work out an understanding of the different types of interpretability techniques. Eventually, we describe the selection process of the interpretability techniques reviewed in this work. The aim of this objective is, thus, to promote an extensive understanding of interpretability in ML but not to present a taxonomy thereof.[13]

(a) Motivate interpretable Machine Learning

(b) Discuss definitions of interpretability in Machine Learning

(c) Describe stakeholders in interpretable Machine Learning

(d) Derive interpretability characteristics in Machine Learning

(e) Establish a terminology and classification of interpretability techniques

(f) Describe selection process of reviewed interpretability techniques

O3: Review and discuss interpretability techniques

With the purpose of working towards an introductory framework of interpretability techniques, in a first step, we examine two classical ML methods (linear regression and decision trees for classification) w.r.t. interpretability. Second, we review recently developed statistical and heuristic techniques which are designed to facilitate interpretability post-hoc the model training. That is, we explain their theoretical properties, illustrate these techniques with an empirical example, and discuss the technique's respective advantages and disadvantages. This subsidiary objectives results in a detailed review of intrinsically interpretive and state-of-the-art interpretability techniques. Our goal is to describe these techniques and showcase how to implement interpret them s.t. they can eventually be used in a researcher-oriented workflow.[14]

(a) Explain intrinsically interpretive Machine Learning techniques

(b) Review theoretical background and explain selected post-hoc model-agnostic interpretability techniques

(c) Showcase interpretability techniques with empirical examples

(d) Discuss the respective main advantages and limitations of each interpretability method

O4: Discuss how to incorporate interpretability methods in a Machine Learning workflow

It is imperative to demonstrate how the distinct interpretability techniques can be integrated into a researcher-oriented workflow. This subsidiary question will result in a set of best-practise interpretability implementation recommendations.

(a) Describe how interpretability can be integrated into a researcher's workflow

(b) Discuss which types of interpretability methods are suited under which circumstance

(c) Discuss limitations of the techniques

[13] To our knowledge there is no taxonomy of interpretability in ML to this date.

[14] We do not evaluate these techniques w.r.t. their general practical success, since the latter is still currently researched and beyond the scope of this thesis. Doshi-Velez and Kim (2018), Gilpin et al. (2018), and Mohseni et al. (2018) provide interesting introductions to the field of interpretability in ML evaluation.

1.5 Research Methodology

As stated in the previous section, one main goal of this thesis is to analyse the concept of interpretability in ML and to showcase relevant interpretability techniques. During this process, we want to interpret and explain a ML model with an interpretability technique as best as possible while maintaining high standards of practicability and accessibility. That is, ideally we identify techniques which can be used for a variety of ML models and facilitate a ML model's interpretability to a larger audience. For these reasons we choose to focus on model-agnostic interpretability techniques for supervised learning tasks. Model-agnostic interpretability techniques are characterized by high general applicability, explanation flexibility, representation flexibility, and the promotion of inter-model comparability (Ribeiro et al., 2016a). Furthermore, they exhibit high levels of generalisability. After learning the model-agnostic technique once, it can be applied to any conventional supervised ML algorithm. We motivate our selection of model-agnostic interpretability techniques in this thesis on the basis of their popularity in the literature. We choose this method since a definition of inclusion criteria of interpretability techniques via quantitative or interpretability properties is hardly possible, because of their heterogeneous and complex nature. Moreover, there is a research gap between interpretability techniques and their application success which is why we cannot justify our selection in this regard.

There are three central components of a ML model which can be interpreted: the input data, the model itself including its structure as well as its parameters, and the output (predictions). The data itself is instrumental in determining interpretability and, hence, an important contributing factor. An extensive understanding of the data can be gained in the data pre-processing phase during explanatory analyses and visualizations which are independent from the actual model training. In this thesis we assume data interpretability as given when we introduce the interpretable models and do not discuss data pre-processing techniques or other techniques to make the data itself more interpretable or discuss shortcomings of techniques when the data is not interpretable. We confine our analysis to the ML model and its output. In order to make the ML model itself more explainable, we identify the following five model-agnostic interpretability techniques in a literature review: partial dependence plots (PDP) (Friedman, 2001), individual conditional expectation (ICE) curves (Goldstein et al., 2015), accumulated local effects (ALE) (Apley, 2016), global surrogate models (Molnar, 2020), and local interpretable model-agnostic explanations (LIME) (Ribeiro et al., 2016b). During this review, we will lay our focus on techniques developed especially for tabular data, i.e. structured and balanced data.[15] Since the latter has a clear data representation, it is the most common form of data and, therefore, particularly suited for interpretability. Text, audio or graphical data demand specific interpretability techniques as they are after all very different in nature and have different requirements w.r.t. interpretability.

[15] While we focus on tabular data only, this does not mean that the techniques introduced cannot be used for non-tabular data as well. Ribeiro et al.'s (2016b) LIME technique, for instance, can be used for text and image data as well.

The main contribution of this thesis is to give an extensive understanding of the concept of interpretability in ML and to review and explain the previously mentioned interpretability techniques. Regarding the latter we will proceed as follows: First, we will explain the theoretical background of each technique. Consequently, we will give examples of the technique on an example data set, after which we will discuss the practical applicability and informative value of the respective technique. This discussion will be aggregated at the end of each subsection into a brief section of advantages and disadvantages for each technique. All findings gathered throughout the evaluation of intrinsically interpretive ML models and post-hoc model-agnostic interpretability techniques will be condensed into a final discussion of the practical application of model-agnostic interpretability techniques in a researcher-oriented ML workflow. A researcher-oriented ML workflow simply describes to us a ML analysis which considers other informative criteria, next to algorithm efficiency and accuracy, as well.

1.6 Structure of this Thesis

This thesis is further structured as follows. Chapter 2 discusses ML definitions from a conceptual perspective and defines the terminology needed throughout this thesis. In chapter 3 we extensively discuss the concepts of interpretability and cognates. During this, we motivate why and in which contexts we need interpretability, discuss definitions of interpretability in ML, describe stakeholders in interpretable ML, derive interpretability characteristics, present the terminology of interpretability in ML in order to work out a classification scheme of interpretability techniques, and motivate the selection of interpretability techniques explained in the following chapters. We introduce the abalone data set which will be used primarily throughout this thesis in chapter 4. We start with the review of intrinsically interpretive ML methods in chapter 5. In this section we discuss linear regression and decision trees for classification in order to introduce intrinsically interpretive techniques. In chapter 6 we analyse post-hoc model-agnostic interpretability techniques. We review and discuss partial dependence plots (Friedman, 2001), individual conditional expectation curves (Goldstein et al., 2015), accumulated local effects (Apley, 2016), global surrogate models (Molnar, 2020), and local interpretable model-agnostic explanations (Ribeiro et al., 2016b) respectively. We will propose and discuss a set of recommendations for the integration of post-hoc model-agnostic interpretability techniques in a researcher-oriented workflow in section 7. The appendix A contains further figures and tables. More detailed explanations for certain techniques can be found in appendix B. All statistical calculations are computed with the statistical programming language *R*. The appendix C contains a description of the different *R* packages which were used to produce the findings of this thesis.

2 Machine Learning

Getting a precise understanding of the scope and objectives of ML is impervious when discussing ML and interpretability with a conceptual lens. Moreover, it also facilitates discussions about interpretability techniques in the later chapters. We discuss ML from a general perspective in the first sub-chapter in order to foster a common basis of understanding of the term. General definitions of ML offer an explanation on the nature and scientific approach of the field. Understanding a machine's learning process is a necessary prerequisite to comprehend what type of knowledge a model acquires during its training and the potential interpretability offers in making this knowledge more transparent. Therefore, we first look at individual ML definitions and highlight individual emphases. In a second step, we compare and summarize these definitions to give an overview over ML definitions.

In the second sub-chapter, we define and discuss important conceptual ML terms in order to develop an accurate apprehension of ML models and their components as well as to delineate the scope of this thesis. This is necessary, since it has been recently observed in ML scholarship that terminological uncertainties pervade the field, as frequently established technical terms are overloaded with new meanings or terms with colloquial connotations get chosen (Lipton and Steinhardt, 2019). The goal of this section is to establish a solid basis and precise terminology for the discussions of ML and interpretability in the subsequent chapters.

2.1 General Definition of Machine Learning

As the field of ML is wide-spun and heavily opinionated (Jordan and Mitchell, 2015), establishing a single definition or unified opinion of ML is an ambitious effort. There are hardly any definitive sources of reference for ML. The majority of journal publications refrains from referencing a definition of ML. Instead of analysing conceptual issues and perspectives which could shine a light on the nature of the field and its historic challenges, literature surveys written in the domain of ML confine themselves rather on a ML application purpose, e.g. financial market prediction (Henrique et al., 2019) and software fault prediction (Malhotra, 2015), or a learning paradigm, e.g. reinforcement learning (Kaelbling et al., 1996) and semi-supervised learning (Zhu, 2005). Furthermore, it is not customary in the ML research community to document recent scientific advances in ML literature sub-strands in a definitive source of reference.[16] Furthermore, to our knowledge, there is no published dedicated taxonomy or ontology of ML to this date. Therefore, we turn to standard ML introductory textbooks for graduates and undergraduates as a main source of information, as they are the default reference work and routinely provide a definition and discussion of the concept of ML in order to build an understanding of the topic prior to explaining ML algorithms.

For instance, Bishop (2006) argues that ML is concerned with pattern recognition in data which, according to him, is an old and fundamental problem statement in science. He defines

[16] Such is common in other sciences, for example, the field of economics where a series of collected, organised and synthesized synopses of recent advances in economic research are published periodically in a compendium-like structure, e.g. the handbook of economic forecasting by Elliott and Timmermann (2013).

pattern recognition as "[...] the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as classifying the data into different categories" (Bishop, 2006, p. 1). Bishop (2006) further reasons that pattern recognition originally stems from the field of engineering, while ML originates from the domain of computer science. To him, however, both disciplines "are two facets of the same field" (Bishop, 2006, p. vii) with similar goals which have undergone considerable development in the past years.

Another example from a standard introductory ML textbook comes from Murphy (2012) who defines ML as "[...] set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty (such as planning how to collect more data!)" (Murphy, 2012, p. 1). Further, he claims that ML is inherently related to statistics and data mining even though all three fields differ in their terminology and emphasis on specific aspects to a small degree. In particular, he mentions that data mining is concerned with the creation of more interpretable models, whereas in ML there is more emphasis on accurate models (Murphy, 2012).

Most definitions in ML emphasize one particular aspect of ML. Bishop (2006) and Murphy (2012), for instance, define ML from an operational point of view. According to them, ML is concerned with the automatic identification of patterns and how to use these for the prediction of out-of-sample data or the, admittedly rather vague, "performing of other kinds of decision-making under uncertainty" (Murphy, 2012, p. 1). To both authors, the concept of pattern recognition is responsible for how a machine's learning process operates. Accordingly, they do not discuss the term "learning" any further. While Bishop (2006) cites algorithms as the main tool of pattern recognition, Murphy (2012) lacks a description of and how pattern recognition is actually done. Both authors do not mention a computational aspect of ML in their definition at all, i.e. the efficiency and accuracy dimension which has become more important with the rise of more sophisticated learning algorithms.

On the other hand, Mohri et al. (2018) define ML "[...] as computational methods using experience to improve performance or to make accurate predictions [...] here, experience refers to the past information available to the learner, which typically takes the form of electronic data collected and made available for analysis" and "[...] consists of designing efficient and accurate prediction algorithms" (Mohri et al., 2018, p. 1). They point out that ML is closely related to data analysis and statistics, as they reason that the ML algorithm's success is highly contingent on the data being used. Mohri et al. (2018) further state that ML techniques are data-based techniques which combine essential concepts from computer science with concepts from statistics, probability theory, and optimization. To them, the main objective of ML is to design efficient and robust algorithms to generate accurate predictions for out-of-sample data.

Mohri et al.'s (2018) definition does emphasize the computational facet of ML compared to Murphy (2012). The main goal of ML, according to Mohri et al. (2018), is the design of algorithms with high efficiency and accuracy to predict unseen or future data. Yet, they do not specify how these algorithms reach this goal. That is, they do not describe the learning process of a ML system. Their definition of ML is rather short and open-ended. ML to them

is a topic which simply cannot be defined in a couple of sentences but rather exhaustively to do justice to the ample scope of the field. Hence, it has to be noted that Mohri et al. (2018) do provide an extensive discussion of learning frameworks and learning tasks in their textbook. These discussions are just not factored into their definition.

Witten et al. (2017) completely refrain from explicitly defining ML, since the latter requires a definition of learning which they like to avoid, because it entails an extensive philosophical discussion. They consider the term "learning" to be a rather imprecise description of what ML techniques actually perform. According to them, learning is a complex concept and to what extent machines actually learn is still subject to extensive investigations. They claim that learning is "[...] to get knowledge of by study, experience, or being taught; to become aware by information or from observation; to commit to memory; to be informed of, ascertain; to receive instruction" (Witten et al., 2017, p. 8). Witten et al. (2017) suggest that "training" is a more suited term as "learning" involves active thinking and a pursued purpose which they deem ML algorithms to not possess. As a matter of fact, the authors rather use the term data mining over ML. Data mining to them is "[...] the process of discovering patterns in data [...] the process must be automatic or (more usually) semiautomatic [...] the patterns discovered must be meaningful in that they lead to some benefit - e.g., an economic advantage" (Witten et al., 2017, p. 6).

While Witten et al. (2017) do not explicitly define ML, their understanding of learning or in their terms "training" probably comes very close to a modern understanding of human learning processes as described, for instance, in Melton (2014). Witten et al.'s (2017) definition of data mining is similar to the ML definition in Murphy (2012) and Bishop (2006). All three emphasize the important role of automatic pattern recognition in ML. What is setting Witten et al.'s (2017) definition apart from the others is the reference of the importance of meaningful patterns, e.g. an economic advantage. Unfortunately, they do not explicitly specify whether meaningfulness refers to the model's real world impact or to the model's knowledge acquisition process. Moreover, ML is a widely applicable tool to Witten et al. (2017) which is why they do not mention a specific purpose of ML in their definition.

Alpaydin (2020) also abstains from explicitly defining ML but rather continues to enumeratively describe ML characteristics. According to him, "machine learning also helps us find solutions to many problems in vision, speech recognition, and robotics [...]", "[...] is programming computers to optimize a performance criterion using example data or past experience [...]", "[...] uses the theory of statistics in building mathematical models, because the core task is making inference from a sample" (Alpaydin, 2020, p. 3). He further substantiates his description of the learning process by claiming that ML algorithms create approximations in the form of patterns or regularities which are used to make predictions about future activity.

The focus in Alpaydin's (2020) definition of ML is rather unique. He is among the few authors to include fields of application and, most notably, instead of discussing computational aspects, he cites making inference from a sample as one of the core tasks of ML in his definition. Even though he mentions ML utilizing statistical theory, this sharply contrasts the old adage in the field of data science stating that ML concentrates on prediction and statistics traditionally

focusses on making inference from data (Breiman et al., 2001; Bzdok et al., 2018). It is likely that Alpaydin (2020) rather refers to inference in the sense of extrapolating a learned pattern to a set of unobserved instances instead of describing causal inference. Yet, his choice of words is remarkably different from other authors.

Mitchell (1997) argues that a machine or computer program "[...] is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" (Mitchell, 1997, p. 2). Mitchell's (1997) definition identifies three key features of a ML algorithm. The task T for which the model is used for, a performance metric P by which means the model is optimized for, as well as the learning experience E which includes the type of ML algorithm, the data used to train the model, and the ML task. As an example, he suggests that a task T corresponds to a computer playing checkers, the performance measure P designates the percentage of games won by the computer program, and the learning experience E represents the amount of games played by the program.

Even though Mitchell (1997) does not include an explanation of the scope of ML, his definition covers a detailed description of the learning process of a ML model. He substantiates his definition by extensively discussing the design of computer learning systems, concept learning, and computational learning theory. Moreover, he discusses several learning paradigms in his book and how they factor in the overall field of ML. According to Mitchell (1997), ML draws on concepts from an extensive array of fields including statistics, AI, philosophy, information theory, biology, cognitive science, computational complexity, and control theory. Mitchell (1997) supplements his definition of ML further in Mitchell (2006) by claiming that a scientific domain is best described by the central question it researches. To him, ML answers the question: "how can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?" (Mitchell, 2006, p. 1). Among all six definitions outlined before, Mitchell (1997) undertakes considerable efforts to provide a precise definition of ML.

While all six definitions mention an important facet of ML, they are far from establishing a unified consensus. It is genuinely challenging to establish a single definition or consensus in a field which gathers a community with such diverse opinions about the field and is as far reaching as ML (Jordan and Mitchell, 2015). This is further substantiated by the fact that ML covers multiple distinct learning paradigms. Moreover, many authors in our "sample" have varying backgrounds and often use domain-specific terminology for similar terms. Therefore, it is not the goal of this section to synthesize the above mentioned six examples into a single definition but rather showcase how different contributors in the field emphasize specific facets of ML. In all six definitions reviewed a similar theme recurs highlighting three important aspects: (1) a machine automatically learns (2) from past experience in the form of electronic data (3) in order to improve its performance at executing a certain objective. The latter may be described as drawing inference (Alpaydin, 2020), optimizing a learning experience at a specific task (Mitchell, 1997), predicting new data (Alpaydin, 2020; Mohri et al., 2018; Murphy, 2012) or the general undertaking of actions (Bishop, 2006).

The learning process occupies a central position in almost all definitions, since it generalizes the fundamental problem task of predicting data by learning a pattern from historical experience. The latter predominantly takes shape in form of electronic data, which is used during the learning process by a ML technique or algorithm to build a ML model. This model is confined to a specific domain or purpose contingent on the data it was derived from in order to perform a certain action, such as prediction or classification. The ability to identify underlying relationships in data and generalize the knowledge a ML model has acquired on new data instances is an essential quality of ML which all six authors collectively emphasize.

However, several unresolved aspects in the definitions remain. Because of its multidisciplinarity and the background heterogeneity of its contributors, the discussion about the field of ML's relationship with other scientific disciplines will hardly ever be resolved. ML is a domain at the intersection of multiple scientific fields and, therefore, there will hardly be an all-purpose definition of ML (Rudin, 2019). Additionally, since the modality of a ML's learning process depends on the learning paradigm, e.g. supervised or unsupervised, and ML encompasses multiple types of learning (Mitchell, 1997), most authors refrain from defining how a machine specifically learns. Witten et al. (2017) even explicitly express concerns when it comes to using the term learning, implying that machines may not actually be able to learn in a way humans do. Perhaps more research in the field of AI will reveal to what extent a machine's learning process differs from those of living organisms.

From an interpretability perspective it is interesting to note that most definitions describe machines as pattern learners and not as gatherers of knowledge.[17] This sharply contrasts recent surveys of interpretability and ML which do claim that ML models capture considerable and valuable contextual knowledge to make increasingly important and complex decisions (Adadi and Berrada, 2018; Arrieta et al., 2020; Carvalho et al., 2019; Guidotti et al., 2018; Mueller et al., 2019; Tjoa and Guan, 2019). Moreover, multiple recent academic sources suggest that interpretability plays a major factor in ML in order to determine the causes behind a ML model's decision (Adadi and Berrada, 2018; Arrieta et al., 2020; Carvalho et al., 2019; Doshi-Velez and Kim, 2017; Guidotti et al., 2018; Kim et al., 2016; Lipton, 2018; Mueller et al., 2019; Ribeiro et al., 2016a; Tjoa and Guan, 2019). This signifies a recent paradigm shift in the conception of ML as a mere pattern recognition and repetition automaton to a more personalized and autonomous learning entity. The increasing ability of ML to function independently on increasingly difficult tasks and the rise of more humanized ML applications may be causes for this. More interaction with ML in our everyday lives, where we encounter ML systems as self-sufficient points of contact inevitably entail a larger interest in their decision-making. Hence, it would seem sensible to update some of the previous ML definitions to raise awareness of qualitative characteristics of ML such as interpretability.[18]

[17] Even though patterns may also constitute knowledge, the emphasis of the authors is usually on the technical notion of patterns instead of learning.

[18] Interestingly enough, Murphy (2012) is the only author to actually mention the term "interpretable" in a brief discussion of his ML definition. He claims, however, that interpretability is more related to the term "data mining", whereas ML is more concerned with building accurate models. This sharply contrasts the general consensus in the recent literature which poses a reconciliation between interpretability and ML.

2.2 Machine Learning Terminology

The most central term in our analysis is the one of a ML model. Awad and Khanna (2015) describe a model as a structure which summarizes a dataset for description or prediction. They identify the learning process as responsible for synthesizing the parameters and corresponding structure of a model from a given data set. In the same vein but in more detail, Kohavi and Provost (1998) define a model as a structure and its associated interpretation which "[...] summarizes or partially summarizes a set of data, for description or prediction [...]" (Kohavi and Provost, 1998, p. 273). They argue that a ML model for regression and classification is produced by applying a ML learning or in their terms "induction" algorithm to a data set. Executing this algorithm then produces a set of parameters following a structure pre-specified by the algorithm, i.e. the architecture of the model. A model, thus, consists of five central components, the input data set, the ML algorithm, the model structure, the parameters, and the output predictions. For instance, a ML model with a backpropagation neural network algorithm comprises: the training data, the backpropagation neural network algorithm, the network architecture (number of the nodes, type of activation function, type of propagation function, learning rate, etc.), the node weights, and the predictions. Models can be generally categorized as parametric, when the model can be described in terms of a finite set of parameters, or non-parametric, when the model and its underlying data distribution cannot be represented by a finite set of parameters (Awad and Khanna, 2015). In this thesis we will focus exclusively on parametrized ML models because of their popularity, even though the techniques introduced in chapter 6 can also be applied to most non-parametric techniques. From now on, we will use the term "model" as a description of ML models if not noted otherwise.[19]

Parameters in this case refer to the parameters of the ML model and should not be mistaken for hyperparameters. The latter are inputs to the learning algorithm and, hence, a part thereof. Consider for example a regularization parameter in a ridge regression model or the learning rate of a neural network. The parameters we refer to are a product of the learning algorithm. A typical example would be the regression coefficients in a linear regression or the weights in a neural network. As a side note, from an interpretability perspective, model-specific interpretability techniques seek to make the ML model's parameters, their corresponding structure induced by the ML algorithm, or derivations thereof more interpretable, whereas model-agnostic interpretability techniques operate detached from a model-structure and -parameter level.

Another important term in our analysis is that of the ML algorithm. An algorithm is defined as a finite series of well-defined instructions which can be solved by a computing machine in order to answer a specific set of computable problems (Math Vault, 2019). A ML algorithm, often referred to as an induction or learning algorithm (Awad and Khanna, 2015; Kohavi and Provost, 1998), is an algorithm which takes specific training instances as input and generates a model which generalizes beyond these initial instances. The ML algorithm is responsible for the synthesis of

[19] Often the term "ML systems" gets equated with ML models. However, we use the former when describing an actually deployed instance of a ML model in a real-world setting with an application purpose and do not use the terms interchangeably.

the model structure and its parameters in order to develop a mapping between the input data and a designated outcome. Hinging on the realized mappings between the training data and the produced output during the training stage, ML algorithms can be largely classified into three categories[20] (Alpaydin, 2020; Murphy, 2012): supervised, non-supervised, and reinforcement learning. We restrict our analysis to supervised learning algorithms, since the majority of ML algorithms falls under this category (Kohavi and Provost, 1998) and it is arguably the most prominent type of learning paradigm in practice (James et al., 2013). When we talk about ML in the following, we refer to supervised learning exclusively.

Supervised learning algorithms build a ML model by inferring underlying relationships between the input variables x and a designated output variable y[21]. As the training data D with sample length N used to train the ML model comprises the training variables x as well as their corresponding target variables y, i.e. $D = \{(x_i, y_i)\} \; \forall \; i = 1, ..., N$, the algorithm is dubbed as learning under supervision. Some supervised learning algorithms are black boxes, that is, the internals of the model are either observable but hardly interpretable to a human observer, e.g. a multi-layer perceptron neural network, or completely unknown, e.g. in a non-parametric ML algorithm. Supervised learning algorithms can be further separated according to their learning task into regression and classification. Regression algorithms produce a mapping or function between input variables and a real-valued output y, where $y \in \mathbb{R}$. A typical example for a ML regression task is the prediction of house prices based on properties of the house, e.g. number of stories, room sizes, construction year, etc. Classification algorithms differ from regression algorithms in that the response variable of them is a categorical or nominal variable from a finite set, i.e. $y \in \{1, ..., C\}$ where C is the number of categories and $1 \leqq C < \infty$. A standard textbook example for a classification algorithm is plant species categorization based on their biological characteristics via decision rules. Classification algorithms may also produce probability estimates instead of discrete mappings. In this case the algorithm estimates for every observation x_i a conditional probability estimate for every class j, where $y = \Pr(y_j \mid x)$. The prediction corresponds then to the class with the highest estimated probability. An example for this is spam detection with logistic regression, where the algorithm has to decide whether an email is spam or ham based on several email characteristics. In our analysis we will cover interpretability techniques applicable to both, regression and classification tasks.

The input data of an algorithm consists of one or more features. One feature is also frequently referred to as a feature vector. Awad and Khanna (2015) describe a feature vector as "an n-dimensional numerical vector of explanatory variables representing an instance of some object

[20] This number usually varies depending on the definitions of types of learning (supervised, unsupervised, and reinforcement), the algorithm's nature of inference (inductive, deductive, and transductive), and the algorithm's learning technique (multi-task, active, online, transfer, and ensemble). For instance, Awad and Khanna (2015) identify six types of learning (supervised, unsupervised, semi-supervised, reinforcement, transductive, and inductive), while Mohri et al. (2018) even mention seven (supervised, unsupervised, semi-supervised, transductive inference, online, reinforcement, and active). However, all definitions separate supervised and unsupervised learning, which is sufficient for the learning paradigm distinction in this thesis.

[21] Because of ML's multidisciplinarity many designations exist for the output y. It may also be referred to as the dependent variable, target (variable), outcome, label, or regressand. From hereinafter we will use these terms interchangeably, since there is no conceptual difference between them.

that facilitates processing and statistical analysis" (Awad and Khanna, 2015, p. 4). Feature vectors are also often times denoted as (independent) variables, predictors, covariates, features, explanatory variables, or regressands. In ML the terms attribute and feature are often times used interchangeably. Yet, strictly speaking, an attribute is a quantity describing an instance (Kohavi and Provost, 1998). Attributes have a domain defined by two available attribute types: categorical and continuous[22], which determine the value assumed by an attribute (Kohavi and Provost, 1998). For example, "weather" is an attribute, whereas "sunny" is a value. A feature would be an attribute and its value combined, i.e. "weather = sunny". In the case of spam classification, features may be the length of the mail, name of the sender, header characteristics, the presence of keywords, etc. Informally, features are often described as the columns in a balanced data set. In this thesis we assume all features to be humanly understandable and naturally interpretable. The complete cross-section of several feature vectors in a data set describes a single data instance or observation. An instance corresponds to a single row in the input data structure. The final output of a supervised ML model may also be referred to as the prediction(s). Predicted variables are denoted throughout this thesis with a hat, such as $\hat{y}$. Our analysis exclusively covers tabular data, where every data instance shares the same set of features and each feature is either numerical, categorical, or dichotomous.

[22]Continuous data is also often referred to as numerical. Categorical data includes dichotomous or boolean variables. Hence, it is also common in the literature to describe three attribute types: numerical, categorical, and dichotomous (boolean/binary).

3 Interpretability

Interpretability is an abstract, multi-dimensional term which is currently subject to various scientific discussions in the interpretability-ML nexus (Adadi and Berrada, 2018; Arrieta et al., 2020; Carvalho et al., 2019; Doshi-Velez and Kim, 2017; Guidotti et al., 2018; Kim et al., 2016; Lipton, 2018; Mueller et al., 2019; Ribeiro et al., 2016a; Tjoa and Guan, 2019). Understanding what interpretability in ML actually describes is not straightforward and requires an extensive discussion of the subject at hand (Lipton, 2018). Therefore, this section devotes itself to present a detailed and holistic description of interpretability and associated terms in ML from multiple perspectives. We explain why and under which application scenarios interpretability in ML systems is required in the first sub-chapter. Consequently, we define interpretability and cognates in ML while explaining the difficulties encountered in the course of this in the next sub-chapter. Since interpretability is a demand-driven concept (Doshi-Velez and Kim, 2017), we describe certain stakeholder profiles in the third sub-chapter in order to foster an understanding of the different requirements for interpretability. An analysis of different conceptual interpretability characteristics or desiderata in the fourth sub-chapter provides insights into the different notions behind the need for interpretable models and further substantiates our definition of interpretability in ML. In the fifth sub-chapter we describe several classification schemes for interpretable models in order to introduce important terminology. Eventually, this chapter concludes with a motivation of the selected interpretability techniques in our review in the following chapters.

3.1 Relevance of Interpretability

Numerous popular press outlets emphasize the importance of interpretability in ML- and AI-topical discussions (Alang, 2017; Bornstein, 2016; Cooper et al., 2004; Core et al., 2006; Harford, 2014; Marcus, 2018; Voosen, 2017). In the same vein, a large and growing body in the ML literature and adjacent fields has called for more interpretability in ML, see Adadi and Berrada (2018); Arrieta et al. (2020); Bibal and Frénay (2016); Carvalho et al. (2019); Doshi-Velez and Kim (2017); Guidotti et al. (2018); Kim et al. (2016); Lipton (2018); Mueller et al. (2019); Ribeiro et al. (2016a); Tjoa and Guan (2019) among others. Furthermore, the 2019 Artificial Intelligence Index Report by Raymond et al. (2019) identifies interpretability and explainability as one of the most frequently mentioned ethical challenges in AI. But why exactly does interpretability in ML matter? The scientific literature offers an abundance of reasons why interpretability in ML today is more relevant than ever.

A seminal study in this area comes from Doshi-Velez and Kim (2017). They trace the need for interpretability back to an incompleteness in the ML task's problem formulation. For some ML models it is not enough to get the prediction (the what) but they also require an explanation of how the model arrived at the prediction (the why). According to them, incompleteness emerges when the real-world costs of a deployed model are not matched by the formal objective of the ML problem statement. For example, an algorithm for credit approval should concurrently optimize the probability of credit repayment while not discriminating based on ethnicity or gender. This

incompleteness creates a fundamental barrier to ML optimization and evaluation as noted by Doshi-Velez and Kim (2017). Incompleteness in this case should not be confused with uncertainty. The former is used to designate quantifiable variance in the estimation process, e.g. estimating parameters from a small-size sample, while incompleteness arises when the ML system produces an unquantifiable bias through circumstantial factors which cannot be accounted for in the ML system, e.g. the inclusion of domain-specific knowledge in a model selection process. Doshi-Velez and Kim (2017) suggest that interpretability is generally not needed when no substantial real-world consequences for improper results of a ML model can be expected and the observer possesses absolute trust in the ML model's decisions, since the problem is well-studied and has been extensively validated in practice. As an example for such a situation they mention the postal code digit recognition for automatic sorting in shipping companies. Doshi-Velez and Kim (2017) describe five scenarios when the effects of gaps in problem formalization are noticeable and interpretability is necessary: gaining scientific understanding, enabling safety, ensuring ethics, learning with mismatched objectives, and highlighting multi-objective trade-offs. We will describe these in the following.

First, the raison d'être of science is the acquisition of knowledge. Since there is no all-encompassing way to specify what knowledge is, the best option to obtain a ML model's information is to interpret or explain it in a humanly understandable way. Opening the ML black box to examine fundamental model blocks via model-agnostic interpretability techniques ensures that researchers understand not only the effects or predictions but also the fundamental causes which are responsible for a model's behaviour (Forde and Paganini, 2019). Many disciplines working with ML algorithms, such as medicine, biology, or socio-economic sciences, do not only require an explanation for the acceptance and validation of the results but also for the purpose of scientific discovery (Guidotti et al., 2018).

Second, when end-to-end systems for some complex ML tasks can only be incompletely tested for an exhaustive list of scenarios, the need for more safety necessitates interpretability. In some cases it might be hardly viable or impossible to receive complete robustness against the uncertainty of the training set not being sampled on the test distribution. From a design perspective the application of interpretable models, features or functions such as model-agnostic interpretability techniques can help identifying and excluding data points or patterns which could cause real-world harm following the model's deployment. This is supported by Varshney (2016) who further claims that inherently safe designed ML systems should include a concept of epistemic uncertainty minimization as part of their optimization function.

Third, when historic data is tinged with stereotypes and past discrimination, a ML algorithm in training may overtake certain biases from the data. After deployment a ML model may engage in automated discriminatory behaviour against ethnicity, religion, or gender notwithstanding the fact that this behaviour might not be desired by the responsible ML engineer. Even if certain bias protection is implemented into a ML system, more interpretability can help capturing those biases which were not considered ex-ante. Hence, incompleteness arises in the case of ethics, because normative constraints are not part of the initial problem statement. Albeit

several studies proposing techniques to identify unduly algorithmic behaviour (Adler et al., 2018; Calmon et al., 2017; Hajian et al., 2016), algorithmic discrimination is not a distant situation in the future but a problem today. This was recently demonstrated by multiple scientific studies: Obermeyer et al. (2019), for instance, found that a US health system algorithm systematically discriminates against black patients by providing them with less financial resources for health care-related treatments even though these patients exhibited the same health needs as - in other characteristics comparable - white patients. Another example comes from Datta et al. (2015) who show that the Google Ads algorithm systematically hid high paying job advertisements from users it recognizes as women. Moreover, Caliskan et al. (2017) provide an in-depth analysis of a natural language processing algorithm implicitly learning gender and racial biases from a standard corpus of text from the world wide web.

Fourth, interpretability may facilitate obtaining the knowledge a ML model has learned even in cases when the ML model's prediction task is orthogonal to the information an observer is interested in. That is, even when a deployed ML system optimizes only a sub target of the main task, i.e. it executes a proxy function of the primary objective, more interpretability in whichever form can help reveal relevant epistemic insights. Consider for instance a ML model which optimizes product recommendations. A product designer might simply be interested in learning about types of customers which is per se unrelated from the optimization of recommendations. However, the model still might learn which items are frequently bought together by which customer giving valuable insight into the behaviour of specific consumers which could be relevant to the product designer.

Fifth, in case of multi-objective trade-offs, interpretability may showcase the exact dynamics of such trade-offs. Doshi-Velez and Kim (2017) mention privacy and prediction quality[23] as an example of the latter. Recently, ML scholarship has started to explore ML systems with non-performance related objectives, such as fairness (Binns, 2018; Chouldechova and G'Sell, 2017; Corbett-Davies and Goel, 2018; Friedler et al., 2019), reproducability (Gundersen et al., 2018; Hutson, 2018; Patil et al., 2016), providing the right to explanation (Goodman and Flaxman, 2017), security (Barreno et al., 2010; Madry et al., 2017; Otte, 2013), or preventing technical debt (Sculley et al., 2015). Such criteria can create diametrically opposed objectives in the ML optimization function. Using interpretability to look into the trade-offs between these objectives can enable an observer to quantify and visualize such complex circumstances in ML systems.

Miller (2019) complements Doshi-Velez and Kim's (2017) analysis by two additional factors from a social learning perspective. One important reason he mentions is the human desire to learn and to discover meaning in ML models and their decisions. Explanations of ML systems can help reconcile contradictions or discrepancies between elements in our knowledge structures induced by the ML model's decision. Understanding how particular model properties or decisions transpire can facilitate learning about a model's behaviour in more detail. Second, Miller (2019) suggests that humans require explanations in order to manage their social interactions with a ML model. Explanations can help unify expectations of a human observer and the model's output,

[23] For a more detailed description of the trade-off see Hardt et al. (2016).

thereby creating a shared understanding of the ML's decision process between the two. This may help humans better manage their expectations towards the model by receiving humanly understandable feedback in the form of explanations in order to improve ML performance.

Other scientists enrich the relevancy discussions of interpretability in ML by arguing that explanatory debugging (Adadi and Berrada, 2018; Guidotti et al., 2018; Kulesza et al., 2015; Molnar, 2020; Ribeiro et al., 2016a) and explanatory auditing (Adler et al., 2018; Tan et al., 2018) are additional arguments for interpretability in ML. Interpretability methods can straightforwardly be used as a debugging tool of faulty predictions in order to identify the cause of the error. Debugging ML algorithms or models is generally troublesome. Most models are trained after extensive pre-processing of the data and after cross-validating a ML model on multiple subsets of the data. It requires extensive efforts to identify simple semantic mistakes or fat-finger errors in the data or model. A famous example for explanatory debugging with interpretability techniques comes from Ribeiro et al. (2016b). They estimate a visual classifier of husky vs. wolf categorization based on example pictures with the dog or wolf in the front and different backgrounds. By using a visual interpretability technique they find that their classifier predicts new instances as a wolf when there is snow in the picture's background and husky when there is not. The classifier learned that background characteristics are more relevant than the head or corpus of the animals. This of course implies that the classifier will perform poorly on new data instances featuring the very likely case of huskies with a snowy background and wolfs without one. Interpretability techniques may be an excellent tool to identify such faulty predictors and help in building more causally related models (Guidotti et al., 2018).

Not only scientists but also regulators are interested in ML interpretability. The General Data Protection Regulation (GDPR), a recent legislation by the European Parliament effectively in force since May 2018, poses a "right to explanation" of algorithmically derived decisions to users, thereby reinforcing the epistemological pressure to apprehend the modus operandi of ML algorithms from a regulatory side (Goodman and Flaxman, 2017). Every European user who is significantly affected by the algorithmic decision making which is based on personal information of this user possesses this right to explanation. If the latter is not ensured, any automatic processing of user data is prohibited.[24]

The United States follow the European regulatory footprints with the introduction of a comparable legislative initiative. The Algorithmic Accountability Act was introduced to US Congress in 2019. If passed, it would direct US federal authorities to develop regulations which require large corporations to regularly conduct impact assessments for high-risk automated decision systems. Such systems include ML models which pose a significant risk towards user data privacy and security. Unfair algorithmic decision making w.r.t. ethnicity, political view, religious belief, gender identity, etc. would, thus, be prohibited.[25] Considering all of this evidence,

[24]See European Parliament and Council of the European Union (2016) for the European GDPR, in particular recital 71. See Goodman and Flaxman (2017), Selbst and Powles (2017), and Wachter et al. (2017) for extensive legal discussions of the GDPR's impact on users and privacy regulations in the EU.

[25]See US Congress (116th) (2019) for the original US congress bill of the Algorithmic Accountability Act. Moreover, Hall (2018) points out that under current US regulation the following acts may also require ML model documentation and explanations when restricting a US citizen's fundamental rights: the Civil Rights Acts of

it seems fair to conclude that the concept of interpretability in ML is more relevant than ever and will only become more relevant in the future.

3.2 The Struggle of Defining Interpretability

Research into interpretability and explanations has a long history in the philosophy of science (Pitt, 1988).[26] According to Mueller et al. (2019), there are four perspectives from which explanations may be analysed: From a logic point of view, as a type of deductive inference; from a mechanistic point of view, as a form of causal reasoning about mechanisms; from a statistical point of view, as a form of statistical inference; and from a relativist/pragmatist view. Unfortunately, the general propositions in this academic work are rather impracticable for application in the field of interpretability and ML. This is mainly, because the interpretability-ML nexus requires explanatory or justificatory notions about a particular incidence and the philosophy of science is generally concerned with the analysis of the causal nature of explanations, such as what is the scientific meaning behind explaining an incidence of events or phenomena (Krishnan, 2019). For instance, in their seminal contribution Hempel and Oppenheim (1948) propose the deductive-nomological model, which states that scientific explanations follow a deductive structure. Giving a scientific explanation involves an argument which deduces the incidence of an event from general laws. Interpretability theorists in ML, however, require explanations which give a specific or intuitive reason or explanation of an algorithm or its decision.

Interpretable systems in ML can generally be described as models where an observer can study and understand how inputs are semantically and mathematically mapped to outputs. That is, a model has to exhibit a certain amount of transparency w.r.t. its model structure and predictions to be considered as interpretable (Guidotti et al., 2018). Interpretability can be, for example, achieved by using an intrinsically interpretive model such as we will describe in chapter 5. Just consider a regression model which can be interpreted by comparing the normalized regression coefficients in order to examine the relative importance of each feature in predicting the target output. Alternatively, one can utilize one of the dedicated interpretability techniques to achieve interpretability. For each respective option a certain understanding of the technical knowledge of the model or the interpretability technique is required. We will present a selection of model-agnostic interpretability techniques based on their popularity in the literature in chapter 6. Yet, we still need a working definition of the term interpretability and cognates such as explainability, explanation, and transparency for our further analyses.

Unfortunately, there is no universally agreed upon verbal or mathematical definition for interpretability in ML to this date.[27] Undeterred by the lack of a more precise definition of

1964 and 1991, the Americans with Disabilities Act, the Genetic Information Nondiscrimination Act, the Health Insurance Portability and Accountability Act, the Equal Credit Opportunity Act, the Fair Credit Reporting Act, the Fair Housing Act, and Federal Reserve SR 11-7.

[26] Notable contributions in this field include among others Hempel and Oppenheim (1948), Salmon et al. (1989), Strevens (2004), and Van Fraassen (1977).

[27] To our knowledge, the only scientific field which defines interpretability formally is meta-mathematics, particularly mathematical logic and model theory. See for instance Visser (1991), and Vuković et al. (1999). Put simply, in this area interpretability describes the idea that a mathematical structure S, e.g. a mathematical set

interpretability, publications frequently make assertions about the interpretability of various ML models or propose their own definition usually as a preliminary discussion and not a main thesis of a paper (Lipton, 2018). Arguably one of the most referenced definitions (Guidotti et al., 2018; Molnar, 2020) comes from Doshi-Velez and Kim (2017). They define interpretability "as the ability to explain or to present in understandable terms to a human" (Doshi-Velez and Kim, 2017, p. 2).[28] This means that in order to make a ML model interpretable it has to be made more understandable from a qualitative and quantitative perspective such that it can be understood by a human audience. Unfortunately, the wording "understandable terms to a human" may be criticised as imprecise, since understanding is a highly subjective term which depends on the interpretive framework of each individual. If we want to formalize and compare notions of interpretability we need a precise formulation. What is more, Doshi-Velez and Kim's definition is fairly restrictive w.r.t a target audience. According to them, interpretable ML models need to be explainable to all humans with different academic backgrounds. This could describe a ML expert or the non-mathematical layperson. Obviously, both personas have a very diverse level of understanding when it comes to the subject matter at hand. The term interpretability does not only have different levels of complexity, but it is also plausible to assume that it targets different audiences (Kirsch, 2017).

Another definition comes from Kim et al. (2016) who propose that "a method is interpretable if a user can correctly and efficiently predict the method's results" (Kim et al., 2016, p. 7). This definition also implies a certain level of human understanding but it restricts interpretable methods to those which can be predicted by a user. Therefore, it reduces interpretability to human predictive power which seems reasonably narrow considering the limited information processing memory allocation of humans (Miller, 1956).[29] Furthermore, it seems sensible to differentiate between the type of interpretability based on the target audience of an interpretability method. For instance, a simple user applying for a credit in a bank with a ML credit scoring algorithm might be more interested in a simple explanation of a prediction along the lines of: "your credit request has been declined, because you have four credit cards with negative account balance and previous customers with the same credit card status have not repayed their loan". On the other hand, a ML expert might be more interested in the actual reasoning process of the model, for example: "if clients have a negative credit card balance on four accounts;, then the expected likelihood of repayment decreases by 28.3% holding all other factors constant". A distinction in the types of interpretability levels, hence, seems sensible. Kim et al.'s (2016) definition, however, does not make such a differentiation. Moreover, they state that methods need to be predicted efficiently but do not elaborate to what criterion efficiency actually refers. It is conceivable that they mean efficiency w.r.t. the ability of an observer to interpret a model in a reasonable

with a collection of finitary operations and relations which are defined on it, may be completely (or partially) formally represented in a different structure T.

[28] Arrieta et al. (2020) define interpretability conceptually in a similar way as "the ability to explain or to provide the meaning in understandable terms to a human" (Arrieta et al., 2020, p. 87). The only difference is that Arrieta et al. (2020) emphasize the provision of meaning, whereas Doshi-Velez and Kim (2017) underline the presentational aspect of interpretation.

[29] Miller (1956) finds that humans have a limited perception budget of 7 ± 2 objects. Research by Cowan (2010) suggests that this number has decreased to 4 ± 1 in recent times.

amount of time. However, it may be difficult for a human to interpret, for instance, a large decision tree with multiple nodes, but that still does not change the fact that decision trees can be interpreted more easily than, for instance, a neural network. Surely, more complexity makes interpretation more complicated, but ML algorithms do not loose their innate status of interpretability because of increased model parameter complexity. This leads one to assume that there is more to interpretability than how accessible a model's coefficients and structure are but also how complex the model is.

Similar to Kim et al. (2016), Biran and Cotton (2017) declare that "[...] systems are interpretable if their operations can be understood by a human, either through introspection or through a produced explanation" (Biran and Cotton, 2017, p. 1). Taking one step back from users making predictions to users understanding the internal representation of a ML model, Biran and Cotton's (2017) definition focusses on the process through which the explanations of decisions, recommendations or actions are undertaken. Furthermore, they specify how a ML model can become more transparent, namely either by providing an additional explanation or by revealing the inner workings of a ML algorithm. Yet, the latter term necessarily entails a discussion about what introspection of an algorithm specifically means. For instance, one could interpret a ML model by looking at its structure and parameters or by explaining a single observation. Also, the same criticism applied to Doshi-Velez and Kim (2017) w.r.t. the ambiguity of the term understanding holds for Biran and Cotton (2017).

All taken together, the presented definitions suggest that interpretability represents a passive model characteristic which refers to the level or ability at which a ML model makes sense to a human observer (Arrieta et al., 2020). Additionally, it seems as if the definitions above do not capture the multi-dimensionality of interpretability vis-á-vis complexity of interpretability, the different types of potential interpretation, and levels of interpretability.

What is more, there is no general consensus what the terms interpretable and explainable specifically mean. Accordingly, some authors use the terms "interpretability" and "explainability" interchangeably (Arya et al., 2019; Koh and Liang, 2017; Silva et al., 2018). This contributes further to conceptual inconsistencies and misunderstood terminology in the field (Arrieta et al., 2020). In their extensive literature meta-review Mueller et al. (2019) note that there is a distinction between interpretability and explainability but they never describe on what terms these two concepts essentially differ. Similarly, Carvalho et al. (2019) acknowledge in their literature survey that there is a conceptual differentiation between the two terms but proceed to use them interchangeably "in the broad general sense of understandability in human terms" (Carvalho et al., 2019, p. 7). Miller (2019) even goes as far as to conceptually equate interpretability and explainability in his analysis of explanations in AI and ML from a social science perspective. All three examples seem contradicting and confusing from an outside perspective w.r.t. their distinction between important conceptual terms.

A contrasting position in the interpretability vs. explainability discussion is taken by Arrieta et al. (2020); Biran and Cotton (2017); Doran et al. (2018); Guidotti et al. (2018); Tjoa and Guan (2019); Tomsett et al. (2018). According to them, the terms interpretability and explainability

refer to two separate concepts. Consequently, there is a distinction between the adjectivized terms interpretable and explainable as well. To Biran and Cotton (2017) explainability is merely one mode in which an observer may make a model more interpretable through verbal or visual tools or techniques. Explainability describes the ability to give an explanation in the respective application and audience context, since this is often a difficult task as most models are not straightforwardly interpretable to observers. Other examples to make ML models more interpretable, according to Biran and Cotton (2017), include model introspection or choosing a model which is inherently more interpretable. Arrieta et al. (2020) associate explainability with the concept of explanations as an interface between a human observer and a decision maker. That is, explainability ensures that the explanation is an accurate proxy of the decision maker and understandable to humans. To them, explainability is an active model characteristic "denoting any action or procedure taken by a model with the intent of clarifying or detailing its internal functions" (Arrieta et al., 2020, p. 87).

As illustrated above and suggested by Krishnan (2019), interpretability and associated terms such as explainability and transparency have not yet been given the kind of definition they would need in order for them to possess a scientifically precise meaning in solving the incompleteness in problem formalization mentioned in the previous sub-chapter. Technical descriptions of interpretability models and objectives which interpretability should solve are usually distinct in their conceptual understanding of the term itself. Their views on the nature and characteristics of interpretability are often conflicting. This suggests that interpretability is not a monolithic concept but rather represents several notions about a single concept. In his work "The mythos of model interpretability" Lipton (2018) supports this notion by providing an in-depth analysis of the special situation of the concept of interpretability. Amongst other things, he argues that interpretability and associated terms such as explainability, understanding, model transparency, etc. are innately ambiguous concepts because of their abstract and imprecise meaning. Krishnan (2019) extends this argument by claiming that definitions of interpretability and cognates rely on fuzzy terms such as "understanding", "comprehending", "intuition", "trust", "fidelity", and "transparency" which are to a large extent subject to interpretation themselves. Krishnan (2019) finds a simple formulation for this: "one interpretability-like word is defined in terms of another interpretability-like word" (Krishnan, 2019, p. 5).

On the other hand, Carvalho et al. (2019) suggest that researchers struggle to define interpretability in ML, because it operates at the intersection of three different fields of science (data science, human science, and human computer interaction). Hence, there cannot be an all-purpose definition. This argument that interpretability is a domain-specific notion used at the intersection of multiple scientific fields which as a result cannot be universally agreed upon in a single definition is also shared by Rudin (2019). The problem of defining interpretability is fundamentally a conceptual one. Interpretability is naturally an opinionated term and covers several individual objectives depending on the application purpose in the respective setting (Lipton, 2018).

Despite these conceptual differences in defining imprecise terms, we follow the separation of

interpretability and explainability according to Arrieta et al. (2020); Biran and Cotton (2017); Guidotti et al. (2018); Tomsett et al. (2018) in our analysis, since it allows us to make a more precise distinction between these two concepts and has been increasingly supported in the literature. Interpretability to us describes the general ability of a ML model to provide information about its internal workings and structure in a humanly understandable way.[30] Interpretability is not about understanding every single bit and number of the model for all data points. It is about knowing enough for a specific goal. Hence, every ML model possesses a natural level of interpretability. Some models such as linear regressions are considered as intrinsically more interpretable, since their coefficients can be accessed directly and suggest a straightforward linear relationship. Other models such as neural networks, provide hardly any information about their inner workings or structure to the outside and are, therefore, considered as less interpretable or not interpretable at all.

On the other hand, explainability describes active actions or procedures with the intent of presenting a cause of a prediction outside the model in a humanly understandable way. While interpretability has its focus on the inner workings of a model, explainability centralizes the identification of causes of observations. Making a model more interpretable relates to raising its overall transparency by explaining its inner workings or predictions. Making a model more explainable means undertaking specific actions to shine light on the cause of a single or set of observations. Explainability is about knowing enough to give a justificatory notion about a certain model aspect.

The relationship between interpretability and explainability is best explained via an example. These two working definitions above imply that a model's level of interpretability can be raised via higher explainability, but higher interpretability does not cause increased explainability. For instance, if someone gives a justificatory explanation of an important observation, this can raise the overall model understanding of an observer. However, if someone provides general information about the internal working of a model, this does not help the observer to directly identify the cause of an observation. A reason for some studies not making the distinction between interpretability and explainability could be, that they are only interested in the overall interpretability of a ML model. Hence, not distinguishing between interpretability and explainability makes things considerably easier from a conceptual perspective, as the concept of interpretability somewhat covers explainability. For instance, technically some interpretability techniques specifically raise the explainability of a model, but since higher explainability positively influences interpretability they are also referred to as interpretability techniques.[31] However, making the distinction between the two terms is scientifically more precise.

Lastly, we specify what an explanation actually means in the interpretability-ML nexus. An

[30] To be fair, it is hardly possible to circumvent using subjective terms, such as understanding, intuition, etc. in discussions of interpretability. We are aware of this issue, but since we want to give the reader a precise as possible understanding there is no way to circumvent using such terms at this current point in time of research. We want to emphasize that our definition does not claim complete scientific validity. It is merely a working definition sufficient for the purposes of this thesis.

[31] Another reason why most techniques which raise model interpretability are designated as interpretability and not explainability techniques is that the term interpretability is simply more popular among interpretability theorists (Preece et al., 2018).

explanation, to us, is a set of statements comprising information provided by a system. These statements are used to describe the reason for a decision or output for a performed task in order to make a model more explainable (Tomsett et al., 2018). Explanations can be used by an observer to form an interpretation or increase the model's interpretability according to his understanding of the term. Explanations may have certain desirable characteristics which determine their success. We will not discuss these, since the topic of explanations in AI is a far-reaching subject in itself and our focus is on interpretability alone.[32]

We want to stress at this point that the arguments mentioned in this paragraph against individual definitions to not deny any legitimacy of the definitions in their application purposes. Each respective definition certainly may fulfil its definitive purpose of the paper it stems from. Yet, each definition taken alone can hardly contribute towards the main goal of this chapter which is to build a comprehensive understanding of the phenomenon interpretability. Simply stated, the above presented definitions hardly do any justice to the complexity and multidimensionality of interpretability in ML. After having established that interpretability can have different objectives for different communities, we first work out important stakeholder groups in ML. Thereupon, we describe several important qualities of interpretability as suggested by the academic literature in a next step to further enrich the understanding of the subject

3.3 Stakeholders of Interpretability

Demands of transparency are increasing from the various stakeholders in AI (Arrieta et al., 2020). Many different roles or stakeholder groups in AI have been proposed in the literature (Arrieta et al., 2020; Preece et al., 2018; Tomsett et al., 2018) which struggle with the interpretation of ML models (Preece et al., 2018). The analysis of such stakeholders groups, while sometimes stereotypical, can provide intelligible insights into the different requirements for interpretability of certain interest representatives. More often than not their interests in interpretability are usually not aligned and sometimes even orthogonal (Weller, 2019).

Tomsett et al. (2018) are among the first to propose a basic role-based model for analysing interpretability in ML ecosystems. In their "interpretable to whom?"-framework they suggest six different roles for human or machine agents from the perspective of explanation recipients. First are creators which have developed the ML system. They either own the intellectual property of the ML system or are directly responsible for its implementation. This can include ML architects, engineers, or programmers. Creators generally want to improve the system's performance. The latter can be measured via quantitative criteria such as predictive accuracy, computational efficiency, and bias minimization or with qualitative factors such as safety. They use interpretability to improve their understanding of the ML system in order to optimize their preferred metric. Second, operators are agents which directly interact with and supervise the ML system. They are responsible for the inputting of correct data and the processing of the

[32] Robnik-Šikonja and Bohanec (2018), specifically their chapter 2, and Carvalho et al. (2019), specifically their chapter 4.5, give overviews of the properties of explanation methods in AI. Mittelstadt et al. (2019) offer a brief introduction of explanations in the philosophy of science with an application-oriented ML focus. Moreover, Miller (2019) provides an overview of explanations in AI from a social science perspective.

output. An operator could be, for instance, a business analyst regularly feeding new input data to a pre-trained ML model. Important and decision-relevant output information gets passed on by the operator to the executor. Operators are, therefore, interested in using interpretability to present all the available information to the executor in an understandable way. In particular circumstances, they may need further explanations from the system to address information-critical topics. Third, executors are agents who use the information by the ML system to make decisions. They are informed by the operators. Interpretability to them is only relevant in the form of information-related explanations. Fourth, decision-subjects are agents which are directly affected by the decision of the executor. They want interpretability in order to understand the information provided by the ML system. Fifth, data-subjects are agents who provide personal data for the training of the ML system. Usually, data- and decision-subjects coincide, there may be cases, however, where they do not. Data-subjects may want to know whether their personal data has led to a fair decision while not having their privacy violated. Sixth, examiners are agents responsible for the audit and inspection of a ML system. They require interpretability insofar as they want to generate explanations from the system to explore its output and internal workings.

Overall, Tomsett et al.'s "interpretable to whom?"-framework offers a first overview of the different roles and relationships in a typical ML system workflow. While relatively rudimentary in the task description of each persona, this framework facilitates an understanding of the individual role requirements w.r.t interpretability. Tomsett et al. (2018) consider their model as a first sketch to assign several notions of interpretability to designated roles in ML practice. However, from a research perspective it would be more informative to examine the interpretability requirements of important groups in ML. In a follow-up paper Tomsett et al. (2018), therefore, extend their first analysis to investigate the different stakeholder communities and their interpretability requirements around interpretable ML in Preece et al. (2018). They describe four central stakeholder communities with different motives and requirements of interpretability in AI:[33] developers, theorists, ethicists, and users.

Developers create ML applications. They coincide predominantly with practitioners applying ML models in their day-to-day work but can also include academics using ML models in their research. Developer activities cover implementing a ML algorithm in a new programming language, code reviewing existing implementations, or simply utilizing a ML model in their everyday work. The primary objective of developers for interpretability is quality assurance. That is, they use interpretability for debugging, evaluation, robustness improvements, and aid system testing of existing or newly implemented ML algorithms and models. Developers correspond the most to the system creators in Tomsett et al. (2018).

Second are theorists. It is the theorists' objective to advance the state of the art ML theory. They are not interested in building practical applications of ML. While theorists mostly tend to be part of academic research, many members of this community come from the industry as well. Theorists use interpretability in order to understand the central characteristics of ML algorithms

[33]Since most current AI systems are based on ML and explainable AI is often used as a synonym of interpretable ML, it is natural to apply their findings to ML.

better and to build ML systems with explainable properties. Developers are often theorists and vice versa. Accordingly, they also correspond to the system creators in Tomsett et al. (2018).

Ethicists are agents concerned with fairness, accountability, and transparency of AI systems. While the other two communities mainly comprise of technically oriented ML experts, this category includes stakeholders from other research domains, such as social scientists, legal experts, journalists, politicians or economists. Ethicists are mainly concerned with interpretability because of qualitative reasons. They require interpretability to determine assurance of fair and unbiased algorithm behaviour, accountability, audibility, legal compliance, and regulatory bodies. Preece et al. (2018) consider all six roles in their "interpretable to whom?"-framework to be somewhat part of the ethicist stakeholder group. They emphasise, however, that the explanation-inquisitive nature of the ethicist community particularly coincides with system examiners, creators as well as data- and decision-subjects.

Last but not least is the user group which describes people who, not unexpectedly, use ML systems. They need interpretability as a provider of explanations in order to help them understand the output/actions of the ML system such that they can decide whether or how to react to the output. The first three stakeholder groups are active contributors of the interpretability-ML nexus, whereas the user group is rather passive. Yet, flexible membership of an agent in the different stakeholder groups is also conceivable for this group sometimes even in relation to the same ML system. That is, a developer may also be a user of the same system she created. In terms of the "interpretable to whom?"-framework users are part of the decision executors, decision-subjects, and operators.

To summarize the findings in Preece et al. (2018). In both of their publications they give a concise overview of the different stakeholder groups in ML and their interests in interpretability. Each stakeholder group seems to exhibit a different interpretability requirement profile. So far, we learned what types of stakeholders in interpretable ML exist and what their main demands or uses for interpretability are. We aim to translate these demands into important characteristics of interpretability in the following section.

3.4 Characteristics of Interpretability

As stated in the previous sub-chapter different tasks cause different requirements for interpretability. Generally speaking, we require interpretability when real-life tasks are not perfectly matched by the ML problem formulations which are intended to solve them. That is, interpretability becomes relevant when predictions and traditional ML metrics do not meet the real-world demands of the model. As a result, the term interpretability is frequently used to capture non-quantitative ML objectives which are deemed important by the observer but cannot be modelled formally (Lipton, 2018). These interpretability characteristics might help us to identify the different notions and purposes behind the need for interpretable models, i.e. intrinsically interpretive models and model-agnostic interpretability techniques. A considerable number of publications summarizes and describes several characteristics or desirable traits of interpretability (Arrieta et al., 2020; Doshi-Velez and Kim, 2017; Guidotti et al., 2018; Lipton, 2018; Mueller

et al., 2019; Robnik-Šikonja and Bohanec, 2018). These may also be denoted in the literature as goals of XAI (Arrieta et al., 2020), auxiliary criteria of interpretability (Doshi-Velez and Kim, 2017), desiderata of interpretability (Doshi-Velez and Kim, 2017; Guidotti et al., 2018), objectives of interpretability (Lipton, 2018), or properties of ML explanations (Mueller et al., 2019; Robnik-Šikonja and Bohanec, 2018). All designations, however, refer to the same concept. These characteristics describe essential qualities falling under the concept of interpretability. To give the reader an overview of characteristics frequently mentioned in the literature, we will summarize the most important qualities as found in Arrieta et al. (2020); Doshi-Velez and Kim (2017); Guidotti et al. (2018); Lipton (2018); Mueller et al. (2019). Sometimes the terms which the authors use to designate interpretability characteristics may be different, but they describe conceptually similar sentiments. When possible we, therefore, try to condense similar notions into a single characteristic to avoid unnecessary overlapping.

Most studies support fairness as a desideratum of interpretable models (Arrieta et al., 2020; Doshi-Velez and Kim, 2017; Guidotti et al., 2018; Lipton, 2018; Mueller et al., 2019). Fairness in ML generally describes the implementation of social and ethical concepts in a model such that its results are independent of variables which may be considered as discriminatory-sensitive from a societal perspective (Verma and Rubin, 2018), e.g. gender, ethnicity, age, etc. The concept of fairness has progressively moved into the focus of ML practitioners as an important criterion and is presently under active research efforts (Chouldechova and Roth, 2018). Multiple successful methods to achieve fairness in ML systems have been proposed.[34] This research changes or replaces existing ML algorithms and techniques to a large extent. However, interpretability can promote fairness by uncovering what knowledge a model has learned whilst leaving it (mostly) untouched. Moreover, Arrieta et al. (2020) suggest that interpretability and explainability can enable attaining and guaranteeing fairness in ML models by visualizing model structure and dependencies to investigate a ML model for biases. Fairness is an extremely important desirable trait of interpretability which is especially important to users affected by model decisions, ethicists, or regulatory bodies (Arrieta et al., 2020).

A related quality of fairness is privacy awareness. Privacy is an extremely delicate matter in data-intensive environments such as ML (Aldeen et al., 2015). Arrieta et al. (2020) and Guidotti et al. (2018) note that not being able to capture the inner working of a ML model can be extremely problematic, since potential privacy violations cannot be checked. On the other hand, the ability to look into a ML model in such a detail as is possible with interpretability techniques can also result in serious privacy breaches (Arrieta et al., 2020). Currently, interpretable ML for privacy preservation is still subject to extensive research efforts (Baron and Musolesi, 2020; Hildebrandt, 2019; Xiang et al., 2019). Privacy will become an increasingly relevant topic especially in consideration of the growing ML applications in data-sensitive areas of everyday life. Regulatory entities, ethicists, and users affected by model decisions are most concerned with privacy from an interpretability perspective (Arrieta et al., 2020).

[34] See for instance Zemel et al. (2013) for a fair classifier. Kamiran and Calders (2012) present data preprocessing techniques for classification without discrimination. Hardt et al. (2016) propose a fairness measure aligning with the goal of ML optimization functions.

Developing trust or trustworthiness in a ML model via more interpretability is seen as another important quality of interpretability (Arrieta et al., 2020; Doshi-Velez and Kim, 2017; Lipton, 2018; Mueller et al., 2019).[35] Trust can be in general defined as a sentiment resulting from prior knowledge and experiences, which generate expectations concerning the reaction of a system or agent (Cahour and Forzy, 2009). In the case of ML, trust may described as "the confidence of whether a model will act as intended when facing a given problem" (Arrieta et al., 2020, p. 90). Thus, trust in this case means that a model always behaves as expected by the observer. Several authors note that trust is not only an important quality of interpretability but an important prerequisite for a ML model (Kim, 2015; Ribeiro et al., 2016b). However, trust is a necessary but not a sufficient condition of interpretability, since some models may be trustworthy but not interpretable. Multiple studies have found that monotonicity can be a determinant of trust in a model (Martens et al., 2011; Verbeke et al., 2011). Unfortunately, trustworthiness is not as easily quantifiable (Hoffman et al., 2018; Siau and Wang, 2018) as other characteristics, such as fairness, which makes it a relatively soft quality. Most interested in the quality of trustworthiness are ML developers and users. (Arrieta et al., 2020).

Another major motive for interpretability is causality, as mentioned in Arrieta et al. (2020); Doshi-Velez and Kim (2017); Lipton (2018); Mueller et al. (2019). More and more research uses ML models as tools for making inference instead of their original purpose: prediction (Grimmer, 2015). An example for this is the work by Kreif and Diaz-Ordaz (2019) who propose using ML models to estimate the counterfactuals in policy treatment evaluation. The correlations found between input and output data do not necessarily reflect cause and effect relationships, since there could always be confounding or omitted variables biasing any estimated causality. Moreover, causal inference from data requires strong assumptions of prior domain-related knowledge (Pearl, 2009). Yet, the thought of a model with a causal relationship seems naturally more straightforward to interpret than a model based purely on correlations to some authors (Arrieta et al., 2020; Doshi-Velez and Kim, 2017; Lipton, 2018; Mueller et al., 2019). Arrieta et al. (2020) assume that interpretability techniques may be used to validate a causal relationship or to provide initiation for a further causal analysis. Moreover, they could be used to generate hypotheses for a new analysis (Oquendo et al., 2012). Causality is an important quality, especially to ML developers and ethically-oriented entities (Arrieta et al., 2020).

Arrieta et al. (2020) and Lipton (2018) emphasize (information) transferability as an additional important quality. Higher transferability of model knowledge can be achieved by determining the boundaries which might affect a model in order to better understand and implement it. Besides, one of the most important reasons to understand the internal representation of a ML model is that it allows users to transfer this knowledge to another problem (Arrieta et al., 2020). Interpretability can facilitate generalizing from one ML model to another, especially in environments where the stakes and deployment costs are high, e.g. medicine or security (Lipton,

[35] From a linguistic stand point, trust and trustworthiness describe different things. While trust is used to designate confidence or reliance on a human or system, trustworthiness describes the state or characteristic of being trustworthy or reliable. In the interpretability-ML nexus both terms refer conceptually to the same quality and are used interchangeably.

2018). For example, in Kumar et al. (2018) the transferability component of interpretable ML models is used to predict new material properties of chemical materials in a cheap and accurate way. Arrieta et al. (2020) suggest that the transferability characteristic is most appreciated by domain experts using ML in their work as well as ML developers.

Moreover, Arrieta et al. (2020) and Lipton (2018) assess informativeness as an important quality of interpretability. Most ML systems are used as decision support systems providing relevant decision information to the observer. What the ML model actually does is, however, to optimize a set of parameters in order to give an output prediction. In black-box ML systems this is the only source of information shared to the outside. Shining light on model knowledge stored inside the model structure can convey additional information and relevant insights for an observer. Interpretations do not necessarily have to explain a model's inner workings in order to be informative (Lipton, 2018) but rather shine light on individual observations. Example-based interpretability techniques can, for instance, provide intuition about a model's rationale by giving an example for a single model prediction. The informativeness criteria is conceptually similar to transferability. Yet, where informativeness designates knowledge acquisition within a ML application or system, transferability allows knowledge extrapolation to a different ML model or even another scientific field. Arrieta et al. (2020) emphasize that informativeness is one of the most found arguments for interpretability in their literature survey. They note that informativeness is relevant to most target audiences in interpretable ML.

Robustness, reliability, and stability are often used to describe the confidence in a ML system which interpretability can raise (Arrieta et al., 2020; Doshi-Velez and Kim, 2017; Guidotti et al., 2018). Robustness in this context describes the ability of a model to exhibit similar levels of performance even against minor perturbations or variations in the input data or parameters (Guidotti et al., 2018). Interpretable models should certainly exhibit robustness in order to guarantee interpretation and explanation consistency. Arrieta et al. (2020) underscore the importance of interpretability stability and even propose that interpretable models should exhibit an information measure about the confidence of its working regime. They mention domain experts, ML developers, and regulatory entities as main benefactors of robustness.

Usability of information (Doshi-Velez and Kim, 2017; Guidotti et al., 2018) or accessibility in Arrieta et al. (2020), is another important component of interpretability. It describes the extent to which end users can get involved in the development process of a ML model. Since interpretability enables an intuitive and non-mathematical understanding of a model, it facilitates the work of non-technical users which otherwise may be overwhelmed by the model complexity at first (Arrieta et al., 2020). Hence, usability can help non-technical observers the most. Interpretability via queryable techniques may be preferred over static explanations, because queryable techniques allow observers to access and use the information according to their intent (Guidotti et al., 2018). Usability of information is a quality which is relevant for all stakeholders in interpretable ML.

Eventually, Guidotti et al. (2018) mention that interpretability techniques with higher generalizability should be preferred as the same technique may be used in different application scenarios and, hence, indirectly make the case for model-agnostic interpretability techniques.

3.5 Terminology of Interpretability Techniques

In this section we explain how interpretability works on a modular level while presenting important interpretability terminology. This facilitates further discussions about interpretability techniques in the later chapters. Interpretability techniques can be distinguished according to the point in time in the model training process when they are used, the dimension which the technique tries to explain, and the scope of their interpretability. Each criterion will be discussed in the following sub-chapters. We will use these categorization schemes to classify the interpretability techniques reviewed in the subsequent chapters.

3.5.1 Interpretability and the Model Building Process

According to Kim and Doshi-Velez (2018), there are three opportunities for the interpretation of a ML model: pre-model (ex-ante), in-model (in tandem), and post-model (post-hoc). Pre-model interpretability techniques are used before the model training and are applied only to the data itself. Hence, they are model-agnostic. Getting a good grasp of the data before building the model is of paramount importance in the ML pipeline not only for reasons of interpretability but also for better model building. A good understanding of the data can be best achieved by selecting purposeful and intuitive features. These may be pre-processed such that they are humanly understandable and easily processable by the algorithm.[36] The selection of a comprehensible and appropriate data basis is one way to make ML models more interpretable before the training process itself. Moreover, simply estimating a feature-sparse model can also help in raising the overall interpretability of a ML model ex-ante.

Kim and Doshi-Velez (2018) state that pre-model interpretability is particularly concerned with exploratory data analysis methods such as can be found in Tukey (1977). Typical exploratory data analysis includes among others: feature engineering, e.g. data transformations for predictors, dealing with missing values, data normalization, the removal/adding/binning of predictors; classic descriptive statistics; feature selection; or principal component analysis (Kuhn and Johnson, 2013). Furthermore, graphical representations of information can help in building a better understanding of the data. Data visualizations for more interpretability can be, for example, bar plots, density plots, scatter plots, box plots, mosaic plots, violin plots, etc. Generally, we assume that data interpretability is given during this thesis.

In-model interpretability can be best accomplished by using intrinsically interpretive models, such as will be explained in chapter 5. Intrinsically interpretive models are models which naturally exhibit high levels of interpretability. For example, a linear regression with five features or a simple decision tree with three nodes can be naturally interpretive to observers. According to Rudin (2019), intrinsic model interpretability is best achieved by using certain quantitative and qualitative model restrictions. She mentions the following important constraints: causality, monotonicity, sparsity, structural (generative), and physical constraints coming from domain knowledge. These constraints are also referred to in the literature as interpretability constraints

[36] In some cases attaining both, data interpretability and algorithm processability, can be difficult as there might be a trade-off between the two (Guidotti et al., 2018).

(Du et al., 2019). In some cases when it might not be straightforward to judge how a feature x can cause a target y, causality constraints facilitate identifying direct cause-effect relationships by restricting the model to only causally well-founded features to find meaningful relationships. On the other hand, monotonicity forces the feature to have a monotonic relationship with the target variable. That is, if a predictor respects the monotonicity constraint, an increase in the numeric value of the predictor either increases or decreases the probability of a classifier in a monotonic way. Sparsity forces a model to use relatively few features such that interpretation is easily manageable and model complexity does not overstrain the mental capacity of an observer. Structural and physical constraints require the model to adapt to the real-world conditions which the model tries to simulate. Du et al. (2019) note that intrinsic interpretability can designate the entire model itself or specific input-prediction mappings. Lipton (2018) comments that intrinsically interpretive models are also sometimes denoted as transparent models.

Post-model interpretability refers to techniques which make a ML model more interpretable after it has been trained. Post-model interpretability, commonly also referred to as post-hoc interpretability, comprises techniques which can analyse most types of models including black boxes and intrinsically interpretive models. Post-hoc interpretability techniques can be used for specific ML algorithms or model-agnostically. We will discuss types of post-hoc model-agnostic techniques in the next subsection in more detail and present selected techniques motivated by their prominence in the literature in section 6.

3.5.2 Types of Interpretability Techniques

Another way to categorize the types of interpretability techniques is to distinguish between model-agnostic and model-specific interpretability techniques. Model-agnostic interpretability techniques can be applied to almost all parametrized and supervised ML models. They are always used after the model training process. Such methods cannot access the model's internal structure and parameters but rely solely on analysing input-prediction mappings. Post-hoc interpretability techniques construct an explanatory more justified model architecture on a complex ML model which explains the model's behaviour (Du et al., 2019). Separating interpretability from the model causes several advantages as described in Ribeiro et al. (2016a): First, the choice of ML algorithm by the model creator is completely flexible, since model-agnostic techniques can be applied to virtually all ML algorithms. This implies, that an observer can choose a model with high accuracy and explain its decision without sacrificing predictive performance at the same time. Second, by separating the model and its explanation an observer has a completely free choice of the type of explanations he prefers to use in his analysis. She can select any model-agnostic technique offering various explanation types without restriction. Third, most model-agnostic interpretability techniques are flexible in their representation. That is, the model-agnostic interpretability technique can use different feature representations compared to the model under interpretation which has a fixed feature form. Thus, feature summaries or explanations of single observations are possible. Fourth, since model-agnostic techniques can be applied to any model, an observer is not being forced to commit to a specific ML model or algorithm but can

deliberately change a ML model after the initial training. Fifth and lastly according to (Ribeiro et al., 2016a), model-agnostic interpretability techniques facilitate the comparison of models and their interpretability. For instance, one can easily compare the explanations of a SVM and a random forest model using the same interpretability techniques and representations. On the other hand, model-specific interpretability techniques limit themselves to a specific type of ML model, because they are designed to allow the interpretation of a specific ML algorithm. They are always used after the model has been trained. One advantage of model-specific ML techniques is that their interpretation can be tailored more closely to the algorithm and can, thus, uncover specific peculiarities of the algorithm (Du et al., 2019).[37] We will only discuss model-agnostic interpretability techniques in this thesis for the reasons stated in this chapter and in Ribeiro et al. (2016a).

Interpretability techniques can be furthermore differentiated w.r.t. the part of the model they aim to explain (Du et al., 2019). A model is said to offer global interpretability, when an observer can understand the entire model by examining the model structure and its parameters. Global interpretability can be achieved by either using an intrinsically interpretive model with interpretability constraints or by using a post-hoc interpretability technique. By explaining a model's internal structure global interpretability can make the entire ML model more transparent. Global interpretability techniques are often associated with the creation of summary statistics for features (Molnar, 2020). They particularly thrive in models where many features or observations require interpretation.

On the other hand, local interpretability means that an observer can inspect a specific individual prediction of a model and is able to understand the model's behaviour w.r.t. that prediction decision. It is achieved by identifying the influence of each feature input contribution towards a specific model output. Local methods are, therefore, also referred to as attribution methods (Du et al., 2019). They are often associated with example based methods or case-based reasoning, as they explain a ML model by focusing on a particular instance (Molnar, 2020). Such techniques are especially useful in sparse models with few instances. In conclusion, it can be stated that global interpretability generally refers to the whole model and is best used to explain multiple features and observations, whereas local interpretability refers to individual input-output mappings or small groups thereof and should be used in low-dimensional and sparse settings.[38]

3.5.3 Scope of Interpretability Techniques

In the interpretability-ML literature there is a clear distinction between intrinsically interpretive models and post-hoc model-agnostic or -specific interpretability techniques applied to ML models.

[37] For the interested reader, Guidotti et al. (2018) provide a mathematical formulation of the model-agnostic and model-specific concepts in their chapter 4.1.

[38] This scale between global and local is, as most concepts in interpretability, relatively soft in its cut-off criteria. For instance, what if a model constitutes of only five observations and we explain three thereof with a local technique. Is the interpretation global or local? Technically, our local interpretation allowed us to understand more than half of the model which certainly helps in establishing global interpretability. Therefore, local interpretability techniques can also contribute towards global interpretability. This does generally not hold for the other way around.

A similar but more granular rationale of how to categorize interpretability comes from Lipton (2018) and is amended by Arrieta et al. (2020). Lipton (2018) differentiates between distinct levels of the scope of interpretability in ML models, namely transparency and post-hoc interpretability. This distinction is very similar to the in-model and post-model distinction of Kim and Doshi-Velez (2018) but it describes the latter two concepts in more detail. It is frequently used to classify interpretable models in ML (Arrieta et al., 2020; Guidotti et al., 2018; Molnar, 2020). We will describe this classification in the following to supplement the classification types of interpretability techniques.

Transparency is responsible for ad-hoc explanations of how a model works in general. Lipton (2018) describes transparency as "the opposite of opacity or blackbox-ness" (Lipton, 2018, p. 12). His characterization of transparency insinuates a sense of directly understanding the entire rationale of a model which coincides with the previously mentioned concept of intrinsically interpretive models. However, Lipton's (2018) understanding of transparency is more detailed as he describes how transparency has three different constituents: simulatability, decomposability, and algorithmic transparency.[39] Simulatability describes the ability of an entire model to be mentally "simulated" or contemplated by an observer. That is, the observer can absorb the information of the model, e.g. its structure and parameters, and comprehend the model step calculations in a reasonable amount of time. Therefore, according to Lipton (2018), sparse linear models are preferable over dense linear models. This implies that complex but low-parametrized models are more simulative than simple but heavily parametrized ones, such as a decision tree with over 100 nodes. This has also been suggested by Tibshirani (1996) before most interpretability discussions in ML started. Lipton (2018) emphasizes the importance of understanding the complex relationship between a model's complexity (including the complexity of the model structure as well as parameter density) and the computational time and effort required by a human observer to understand a model. To him, this simulatability is a major determinant of interpretability.

Second, decomposability describes the extent to which any individual part of the model permits an intuitive interpretation. This refers to input data, parameters, and calculations. An example for an intuitive interpretation of a node in a decision tree might be a plain text description, such as "all individuals taller than two meters take the left branch, all others the right". The fact that every input is interpretable surely entails high requirements on the model. Yet, the concept to be able to directly understand and interpret individual components of a model facilitates overall model interpretability considerably (Lipton, 2018). Models with highly engineered features do not fulfil this characteristic. Lipton (2018) notes, however, that the robustness of parameters and model structure has to be observed for decomposability to be meaningful, as the weights and coefficients for some models may vary considerably with feature selection and preprocessing. Decomposability is especially important for a granular understanding of the model.

Third, algorithmic transparency refers to the extent of transparency of the learning algorithm

[39] Please note, when we refer to transparency in this chapter we refer to Lipton's understanding of the term. Outside of this chapter we mean the general understanding of the term transparency. We will explicitly make it clear when referring to Lipton's understanding after this chapter.

itself. According to Lipton (2018), algorithmic transparency specifically designates the degree of possible exploration by means of mathematical, particularly numerical, analysis of the ML algorithm. That is, to which extent can an observer follow the process after which the ML algorithm produces the ML model and its predictions. While the optimization of linear models is well researched and the structure of the error surfaces can be understood fairly well, most modern ML algorithms require solution approximation via heuristic optimization, e.g. stochastic gradient descent, in which the loss architecture cannot be entirely observed and has to be approximated. Hence, an observer cannot fully grasp how the model will behave in every circumstance. Algorithmic transparency does not directly cause a model to be more interpretable but understanding the optimization landscape can help in building more trust and robustness towards the ML model and, thus, indirectly support interpretability.

Post-hoc interpretability, on the other hand, is different from transparency and the three characteristics thereof mentioned above insofar, as it provides useful insights about the model and its learned knowledge to practitioners and end users of ML after the model training. Post-hoc interpretability methods are particularly useful for the interpretation of black box models but may also be used for intrinsically interpretive ML models. Lipton (2018) notes that this type of interpretability comes close to how we humans rationalize our behaviour. He further suggests that there are four types of explanations which post-hoc interpretability employs: text explanations, visual explanations, local explanations, and explanations by example. These explanations refer to the results of post-hoc interpretability techniques rather than interpretability dimensions as for transparency.

Text explanations are useful for interpretation, since it is natural for humans to explain decisions verbally (Lipton, 2018). This includes a wide range of examples including finding a verbal pendant of a quantitative explanation or even training a ML model to translate a model's state representation into verbal strategy explanations as done in Krening et al. (2016). Text explanations score high on user trust and explanation quality and are, hence, preferred as explanation tools (Gilpin et al., 2018).

Visual explanations of post-hoc explainability are used to graphically capture a model's behaviour. Since the human imagination has its limits in three-dimensional space, most visualization techniques need dimensionality reduction techniques such that they can be visualized in a humanly interpretable manner (Arrieta et al., 2020). Often times visualization techniques are combined with other explanations to reach their full potential. As indicated by high ratings of trust and explanation quality, visual explanations are popular among users as well (Gilpin et al., 2018).

Another common approach is the use of local explanations. The idea behind this type of explanation is to reduce the model space to a smaller less complex solution subspace and consequently give an explanation therein. They correspond to explanations given with the rationale of local interpretability mentioned in section 3.5.2. Local explanations can either explain the behaviour of a single observation-prediction pairing, the behaviour of a group of observation-prediction pairings, or the entire model behaviour.[40]

[40]Consider for instance the case, when an explanation of a single highly influential instance causes the predictions

Explanations by example aim at making a model more understandable by showcasing a single data instance and their predictions as an example. Explanations via this class are usually more successful in conveying model-relevant information when they are representative examples of the model's internal knowledge. Humans frequently use explanations by example to justify their decisions. Therefore, example-based explanations are very popular among ML users and in the literature (Cai et al., 2019; Molnar, 2020).

Lipton (2018) admits that his list of the types of post-hoc interpretability measures does not claim completeness. Consequently, Arrieta et al. (2020) propose to append two additional types of explanations to Lipton's listing: explanations by simplification and feature relevance explanation.

Explanations by simplification comprises methods which estimate a new model based on the model to be explained. The new model tries to be less complex while exhibiting a similar prediction performance and aiming to resemble the model to be explained. Surrogate models which explain a complex model's behaviour by explaining its predictions with an intrinsically interpretive one are an example for this.

Feature relevance explanations include techniques used to illustrate the internal mechanisms of a ML model by calculating or visualizing statistical figures describing the importance of the model's features. These scores describe the effect of a feature on the output of the entire model quantitatively and can be compared to assess the relative importance of each individual feature. Arrieta et al. (2020) state that feature relevance methods try to explain a model indirectly. Most of the techniques explained in chapter 6 such as partial dependence plots, individual conditional expectation curves, and accumulated local effects are an example for this.

Naturally it is conceivable that post-hoc interpretability explanations can be simultaneously part of multiple categories. For instance, a text-based explanation of the single most important instance in a ML model, ticks three boxes: textual, local, and by example.

3.6 Motivation of the Selected Interpretability Techniques

Our goal in this thesis is to develop an understanding of interpretability in ML and make ML models more interpretable. During this, we explain and review current interpretability techniques to introduce the reader to the field and showcase current state-of-the-art interpretability techniques. We explain in the previous section that there are two promising ways in making ML models more interpretable. Either use an intrinsically interpretive model or employ a post-hoc model-agnostic technique. In this subsection we motivate the selection of the reviewed models and techniques in section 5 and 6.

The most apparent way to get a model with high interpretability is to choose an intrinsically interpretive model. Therefore, we start in chapter 5 by explaining two popular naturally transparent model, one for regression, the linear regression, and one for classification, decision trees. We choose the linear regression, because it is the most prominent work horse in statistics and ML. Its interpretability property has been emphasized by multiple textbook authors (Hastie et al., 2009; James et al., 2013; Stock and Watson, 2015). A similar rationale applies to decision

to shift from one sub-space to another.

trees for classification. They offer considerable decision transparency, because of their graphical representation and parameter structure. Moreover, they are easily accessible to non-ML-experts (Bishop, 2006; James et al., 2013). Their straightforward interpretability and their applicability to classification and regression have been pointed out by multiple publications (Arrieta et al., 2020; Guidotti et al., 2018; Molnar, 2020) and standard textbooks alike (Bishop, 2006; Hastie et al., 2009; James et al., 2013). They make incredibly suited examples for transparent models.

We start with transparent models for the following reasons. First, discussing intrinsically interpretive ML techniques represents an illustrative introduction into the topic as most naive models are somewhat familiar to the reader. We will not discuss these techniques in particular detail as most introductory textbooks do a way better job than we could in this brief section.[41] Rather, we will focus on the theoretical background regarding the interpretability aspect of each model. That is, we want to make the reader aware of certain technical model peculiarities such that she is able to use the model with its full potential w.r.t interpretability. Second, more advanced interpretability techniques may be understood in more detail by contrasting them with simpler methods. Eventually, transparent models are an adequate alternative to more complex ones. Third, such simple models may be used as surrogate models for complex black box models, because of their straightforward interpretability as demonstrated in chapter 6.

In chapter 6 we will introduce the reader to global post-hoc model-agnostic interpretability techniques. We chose the latter because of their impressive model, explanation, representation flexibility, their straightforward comparability (Ribeiro et al., 2016a), and for the reasons stated in the previous sub-chapter. During this, we review and discuss partial dependence plots (Friedman, 2001), individual conditional expectation curves (Goldstein et al., 2015), accumulated local effects (Apley, 2016), global surrogate models (Molnar, 2020), and local interpretable model-agnostic explanations (Ribeiro et al., 2016b) respectively. We chose to select these methods based on their popularity in the literature. Table 1 highlights the selected methods and respective contributions mentioning their popularity. We start with PDPs, since they are the most popular and straightforward tools to interpret the effect of a feature by summarising its nature in a visual plot. Since in the PDP aggregation process a lot of information of individual observation may be lost, we also introduce ICE curves to showcase the local pendant to PDPs. ICE curves visualize the individual behaviour of each observation's partial dependence in a straightforward manner. Both PDPs and ICEs assume that features are not correlated which is problematic in most data sets, as inter-feature correlations are high and usually very complex. Thus, the results of PDPs and ICEs have to be generally interpreted with caution. Therefore, we also introduce a model-agnostic interpretability technique which is not prone to such an assumption, namely ALEs. ALEs have recently become a more attractive alternative to PDPs. Through the use of an adept aggregation and effect calculation process, ALEs are the most secure alternative to PDPs and ICEs in that they show the "true" nature of an effect while not being prone to correlation

[41] See for instance James et al. (2013), particularly their chapter 3, for an undergraduate level introduction to linear regression and see Hastie et al. (2009), particularly their chapter 3, for a graduate level introduction of the linear regression. For decision trees see James et al. (2013), particularly their chapter 8.1, and Hastie et al. (2009), particularly their chapter 9.2.

Table 1: Overview of selected literature for post-hoc model-agnostic interpretability techniques

Partial dependence plots (PDPs)	Adadi and Berrada (2018); Carvalho et al. (2019); Guidotti et al. (2018); Hall (2018); Molnar (2020)
Individual conditional expectations (ICE)	Adadi and Berrada (2018); Carvalho et al. (2019); Hall (2018); Molnar (2020)
Average local effects (ALE)	Carvalho et al. (2019); Molnar (2020)
Global surrogate models	Adadi and Berrada (2018); Carvalho et al. (2019); Molnar (2020)
Local interpretable model-agnostic explanations (LIME)	Adadi and Berrada (2018); Carvalho et al. (2019); Hall (2018); Molnar (2020); Preece et al. (2018); Ribeiro et al. (2016a); Robnik-Šikonja and Bohanec (2018); Tjoa and Guan (2019); Tomsett et al. (2018)

Source: Own research.

issues. Eventually, we round up our selection of interpretability techniques by showing how global and local surrogate models can be used to approximate a black box model via substitution of more interpretive models. We use the previously introduced techniques from section 5 as example techniques. With these five techniques we hope to provide a brief but thorough introduction to the most prominent interpretability techniques currently in the literature.

4 Data Set

We will showcase the models and techniques mentioned in the last section empirically. During this, we use the abalone data set.[42] This data set, originally owned by the Department of Primary Industry and Fisheries in Tasmania, was donated by Sam Waugh to the public and is freely available in the UCI Machine Learning Repository (Waugh, 1995). The data set originally comes from a non-ML study (Nash et al., 1994). It has been extensively used in ML research and teaching because of its pedagogical usefulness. The data set comprises observations of abalones (lat.: haliotis) a type of edible, herbivorous sea snail. Abalones have a low, open spiral convex shell structure shaped like an ear. Depending on the species they can grow from $10 - 25$cm in length and up to 7.5cm in depth. Their shells are glistening and iridescent in the inside which is why they are often used in the manufacturing of ornaments, particularly in Maori culture. They can be found in seas worldwide. Figure 1 shows a blacklip abalone from the Tasmanian Sea such as one could look like in the abalone data set.

Figure 1: Blacklip abalone (haliotis rubra)

Source: Southwood (2014), see https://en.wikipedia.org/wiki/Haliotis#/media/File:Haliotis_rubra_P2164176.JPG.
Note: Photograph was taken at Mistaken Cape, Maria Island, Tasmania.

The data set contains categorical and numerical data for eight attributes (*Sex, Length, Diameter, Height, WholeWeight, ShuckedWeight, VisceraWeight,* and *ShellWeight*) described in more detail in table 2. The idea behind the data set is to predict the age of an abalone with its physical measurements. The age of an abalone can be determined by cutting through the shell and staining it such that the number of rings in the shell can be counted with a microscope. This is a troublesome activity and the abalone can be hurt or die in the process. Therefore, a ML algorithm can provide an easier (and more abalone friendly) alternative for estimating the age. The variable to be predicted for regression is the number of rings, *Rings*. The variable is already present in the data set. For classification, we create the variable *AgeCat* which we obtain by

[42] The version of the data set used throughout this thesis can be obtained on https://archive.ics.uci.edu/ml/datasets/Abalone.

splitting the variable *Rings* at its mean into two categories: if $Rings > mean_{Rings}$ then the abalone is considered as old and if $Rings \leq mean_{Rings}$ then it is considered as young. This split has no biological motivation whatsoever. The variable *AgeCat* is rather just an indicator if an abalone is considered as young or old.[43]

Missing values in the data set were removed. The data set comes in already preprocessed form. The ranges of the continuous values have been scaled in the initial data set by dividing all numerical values by 200.[44] However, as we are more interested in the interpretation of the data set, we want the actual non-processed values of the data set. Therefore, we reverse this change by multiplying all numeric values (*Length, Diameter, Height, WholeWeight, ShuckedWeight, VisceraWeight, ShellWeight*) by 200. This is the only preprocessing necessary for our research purposes. In the following chapters we will try and explain some of the predictions made by black box ML models by justifying certain model behaviour with potential biological causes even though we are no expert in the subject of marine biology. The goal is to just showcase how this analysis can trigger certain thought processes w.r.t. the model. We do not aim to make any ground-breaking new insights in the biology of abalones.

Since the main task is to estimate the age of an abalone by predicting the number of rings, we will often use the informal phrase "X rings old" as a synonym to "X years old". While this is not technically true, the actual age of an abalone in years can, however, be obtained by dividing the number of rings by 1.5. For the purposes of our work the formulation "X rings old" will suffice. During the analyses in chapter 5 and 6 we will focus our analysis on the features *WholeWeight*, *ShuckedWeight*, *Length*, and *Sex* in order to have a manageable selection of features which we can in turn compare between techniques. We select these four variables because of their expressive effect behaviours.

[43] We prefer to use a single data set for simplicity reasons and there is unfortunately no other variable which might present a suited candidate for showcasing a classification task. Moreover, since both variables somewhat refer to the same concept of age, we can compare the results of interpretable models for regression and classification tasks.

[44] The data set was initially pre-processed for use with an artificial neural network in 1995. The size of the data set was considered as large at the time, which is why the data was normalized to facilitate the processing of the data with a neural network. Today such pre-processing should not be necessary.

Table 2: Overview of the variables in the abalone data set

Variable	Unit	Data Type	Description
Sex	categorical	m (male), f (female), and i (infant)	Gender of the abalone
Length	numerical	millimetre	Longest shell measurement
Diameter	numerical	millimetre	Perpendicular to length
Height	numerical	millimetre	Weight with meat in shell
WholeWeight	numerical	grams	Whole abalone
ShuckedWeight	numerical	grams	Weight of meat
VisceraWeight	numerical	grams	Gut weight (after bleeding)
ShellWeight	numerical	grams	Weight after being dried
Rings	integer	#	+1.5 rings gives the age in years
AgeCat	categorical	Yes/No	Binary variable (Rings $> mean_{Rings}$ = Yes, Rings $\leq mean_{Rings}$ = No)

Source: Abalone data set.

Table 3 contains descriptive statistics and table 4 a correlation matrix of the abalone data set. As can be seen there are $4,177$ observations in total. Each attribute exhibits a considerable amount of variation as indicated by the standard deviations (St. Dev.). The average abalone in our sample is 10.48 cm long and 8.16 cm wide. Around 66.29% of abalones in our sample are between seven and eleven "rings old". A large fraction of abalones is, therefore, five to seven years old. The youngest abalone in our sample is eight months old, whereas the oldest one is nineteen years and four months old which is quite a considerable age for the average abalone. The split for the variable *AgeCat* almost perfectly separates our sample in half as indicated by a mean of 0.498 of the variable *AgeCat*. There are $2,081$ observations which have more than nine rings and $2,096$ which have less than 9 rings. The feature *Sex* has three different values. Contrary to what one may believe, this does not imply that abalones have three genders. The reason for the category Sex to have three values is the following: Abalones reach sexual maturity at age four to five years old and, therefore, can prior to this only be identified as infants (Prince et al., 1988). Naturally, the correlations among size and weight measures are relatively high. As our analysis is rather explanatory than justificatory we deem the problem of multicollinearity to be rather small from an inferential standpoint. Moreover, multicollinearity is rarely an issue in ML, as it does not change the predictive power of a model. Yet, it does increase coefficient variance which may affect parameter interpretation. Therefore, we have to keep in mind that multicollinearity may be present in all further analyses.

Table 3: Descriptive statistics of the abalone data set.

Feature	N	Mean	St. Dev.	Min.	Pctl(25)	Pctl(75)	Max.	Unit
Length	4,177	104.798	24.019	15	90	123	163	millimetre (mm)
Diameter	4,177	81.576	19.848	11	70	96	130	mm
Height	4,177	27.903	8.365	0	23	33	226	mm
WholeWeight	4,177	165.748	98.078	0.400	88.300	230.600	565.100	grams
ShuckedWeight	4,177	71.873	44.393	0.200	37.200	100.400	297.600	grams
VisceraWeight	4,177	36.119	21.923	0.100	18.700	50.600	152.000	grams
ShellWeight	4,177	47.766	27.841	0	26	65.8	201	grams
Rings	4,177	9.934	3.224	1	8	11	29	#
AgeCat	4,177	0.498	0.500	0	0	1	1	binary

Source: Own computation.
Note: Categorical variable Sex *has been excluded. St. Dev. denotes the standard deviation. Min. denotes the minimum value. Pctl(25) denotes the* 25*th percentile in the sample distribution. Pctl(75) denotes the* 75*th percentile in the sample distribution. Max. denotes the maximum value.*

Table 4: Correlation (Pearson) matrix of the abalone data set.

	Length	Diameter	Height	WholeWeight	ShuckedWeight	VisceraWeight	ShellWeight	Rings	AgeCat
Length	1	0.987	0.828	0.925	0.898	0.903	0.898	0.557	0.515
Diameter	0.987	1	0.834	0.925	0.893	0.900	0.905	0.575	0.530
Height	0.828	0.834	1	0.819	0.775	0.798	0.817	0.557	0.505
WholeWeight	0.925	0.925	0.819	1	0.969	0.966	0.955	0.540	0.529
ShuckedWeight	0.898	0.893	0.775	0.969	1	0.932	0.883	0.421	0.447
VisceraWeight	0.903	0.900	0.798	0.966	0.932	1	0.908	0.504	0.515
ShellWeight	0.898	0.905	0.817	0.955	0.883	0.908	1	0.628	0.574
Rings	0.557	0.575	0.557	0.540	0.421	0.504	0.628	1	0.733
AgeCat	0.515	0.530	0.505	0.529	0.447	0.515	0.574	0.733	1

Source: Own computation.
Note: All statistics are estimated as Pearson's correlation coefficient with $\rho_{x,y} = \frac{Cov(x,y)}{\sigma_x \sigma_y}$*, where* $Cov(x, y)$ *is the covariance of* x *and* y*,* σ_x *is the standard deviation of* x*, and* σ_y *is the standard deviation of* y*. Categorical variable* Sex *has been excluded.*

5 Intrinsically Interpretive Models

In this chapter we introduce and explain two intrinsically interpretive or "transparent" (Lipton, 2018) ML models with a focus on interpretability. In the first sub-chapter we describe linear regression, whereas we devote the second sub-chapter to decision trees for classification. We start each sub-chapter, with a brief description of the theoretical background of each technique, after which we describe how to interpret the latter. Consequently, we showcase the respective technique on an empirical example from the abalone data set. During this, we focus on a particular set of features (*WholeWeight, ShuckedWeight, Length* and *Sex*) to allow the comparison of results between different techniques. Each chapter concludes with a discussion of the respective technique's advantages and disadvantages.

5.1 Linear Regression

5.1.1 The Linear Regression Model

The linear regression is arguably the most prominent model in statistics and ML. It is well-researched and often outperforms complex models in situations with small training sample sizes, low signal-to-noise ratio, or sparse data (Hastie et al., 2009). Moreover, linear regressions provide a simple, linear, and interpretable description of input-output relationships. A linear regression model with the regression function $f(\mathbf{x})$ assumes that it is linear in its input $\mathbf{x}$, where $\mathbf{x}$ denotes the input vector with p-different features and length N. The function $f(\cdot)$ predicts the real-valued output y. Thus, a linear regression estimates a model $f(\mathbf{x})$ to predict in the following form:[45]

$$f(\mathbf{x}) = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_i. \tag{1}$$

Subscript i in x_{ij} refers to observation i, while subscript j refers to the number of variables with length p. The sample length is N. The error term ϵ_i is the prediction error we make when forecasting y. That is, it captures the difference of the model's predicted value $\hat{y}$ and the actual value y_i, i.e. $\epsilon_i = y_i - \hat{y}_i$. It catches other factors which are missing in the model, such as other variables causing variation in y, the true relationship most likely not being linear, and measurement error in the variables. A typical assumption in linear regression is that this error term ϵ_i is independent of y. The observations x_{ij} can be quantitative inputs or transformations thereof (e.g. log, square-root, or higher exponents), dummy/categorical variables, or interaction terms (e.g. $x_j \cdot x_{j+1}$). The β_j's are unknown linear coefficients which have to be estimated numerically. They describe the linear weight of feature j in predicting y and are estimated in the training process. The β_0 is also known as the intercept and gives the expected value of y when all covariates are zero. Since the β_j's are always linear and the input data $\mathbf{x}$ is assumed to be linear, the linear regression model is de facto linear. This property of linearity implies that the linear regression is also always monotonic, which facilitates the model interpretation considerably. Multiple methods exist to estimate the coefficients in a linear regression. By far

[45] Throughout this chapter we follow the notation in Hastie et al. (2009).

the most popular is the ordinary least squares.[46] In OLS the coefficients $\beta = (\beta_0, ..., \beta_1, ..., \beta_p)$ are determined by minimizing the residual sum of squares (RSS):

$$\begin{aligned} RSS(\beta) &= \sum_{i=1}^{N}(y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2. \end{aligned} \tag{2}$$

5.1.2 Ordinary Least Squares Assumptions

As with every model, the linear regression also comes with assumptions. In order for a linear regression model to be interpretable, these assumptions have to hold, even though this is often not the case in reality (Molnar, 2020). Depending on which econometrics or statistics textbook one consults there are different opinions on the number of linear regression assumptions (Stock and Watson, 2015).[47] We will briefly discuss the linear regression assumptions in Hayashi (2000) from an interpretability perspective.

The first assumption states that the relationship between the dependent variable and the independent variables is linear. Since the individual regressor effects are additive, the predicted response is a linear combination of its features. This assumption is certainly one of the reasons why linear regressions are so popular in practice (Molnar, 2020). Linear effects are easily accessible and interpretable compared to non-linear effects. Yet, linearity is also a very restrictive assumption, since in reality most relationships are often more complex.

The second assumption states that the conditional error term distribution ϵ_i given x_i has a mean of zero, i.e. $E[\epsilon \mid x] = 0$. This assumption of strict exogeneity makes sure that there are no other systematic factors contained in the error term ϵ_i which have a biasing influence on the coefficients and ultimately the predictions. This assumption ensures that OLS is an unbiased estimator.[48] This is certainly a necessary condition for causality and a very desirable property for estimators in general.

Third, there is no multicollinearity among the independent variables.[49] If multicollinearity is present in our independent variables, then the estimation process of the regression coefficients becomes unstable and statements on the estimation of the regression coefficients increasingly inaccurate as the standard error of the coefficients are estimated to be smaller than they should

[46] See chapter 3.2 in Hastie et al. (2009) for a more detailed review of least squares optimization.

[47] These assumptions are sometimes also referred to in the context of the Gauss-Markov Theorem, which states that OLS is the best linear unbiased estimator, if the error terms ϵ in the linear regression have an expected value of zero given x, are uncorrelated, and have equal variances (Stock and Watson, 2015).

[48] Unbiasedness of an estimator describes the difference between the estimator's true value and its predicted value. If the difference of the two is zero, the predicted estimator is said to be unbiased. For a more detailed description of the concept of unbiasedness in the estimation of statistical parameters, see Toutenburg and Heumann (2008b), their chapter 6.

[49] Multicollinearity describes the concept in statistics that two or more regressors in a multivariate regression model are strongly correlated. If the correlation between two or more regressors is one then they exhibit perfect multicollinearity and the model cannot be numerically determined.

actually be under no multicollinearity. Inference statistics and parameter estimates should, therefore, not be interpreted and trusted any more. Hence, it is very difficult to make a statement about which of the parameters is actually responsible for the effect. Interpreting coefficients under the effect of high multi-collinearity is, as a result, questionable at best.

Fourth, the error term is homoscedastic if the variance of the conditional distribution of ϵ_i given x_i is constant for all observations in the sample and does not depend on x_i: $Var[\epsilon_i \mid x_i] = \sigma^2 \quad \forall i = 1, ..., N$. If the conditional variance of the error terms depends on the covariates x, the error term is said to be heteroskedastic. While the OLS estimator remains unbiased even under heteroskedasticity, homoskedasticity is an important prerequisite for interpretability. The error terms have to exhibit homoscedasticity, as coefficients under heteroscedasticity are less precise in that heteroscedasticity may lead the OLS estimates of the coefficients' variance to inflate. Less precision in the determination of the coefficients in turn increases the probability that the coefficient estimates are further away from their true sample value.

Fifth, the distribution of the error term ϵ conditional on x is jointly normal, $\epsilon \mid x \sim \mathcal{N}(0, \sigma^2)$. Normality is important for inferential statistics, such as confidence intervals and p-values. Yet, it is by far not the most important assumption, since most inference statistics in large samples are still asymptotically correct even under non-normality.

Each linear regression assumption is important for interpreting the model. Therefore, all of these assumptions should be checked via the respective statistical test to ensure that the interpretation of the coefficients and their corresponding inference statistics are completely valid. Only then can meaningful interpretation take place.

5.1.3 Interpretation

The interpretation of the features in a linear regression is straightforward. Consider the case of a numerical feature x_j, where $j = 1, ..., p$. In a univariate or multivariate linear regression the size of the β-coefficient for each feature j represents the size of the effect that the independent variable x_j has on the dependent variable y. The sign of the coefficient (positive or negative) expresses the direction of the effect. For instance, in a univariate regression with one parameter the β_1-coefficient shows by how much the dependent variable y is expected to increase (if the coefficient is positive) or decrease (if the coefficient is negative), when the independent variable x_1 is marginally increased by one unit. In a multivariate regression with multiple parameters the β_j-coefficient designates by how much the dependent variable y is expected to increase when the independent variable x_j is increased by one unit, while holding all other independent variables constant.[50] This last addition implies that the interpretation of coefficients in linear regression can only take place in isolation of other coefficients.

As stated above, the input variable x_j can consist of different attribute types: numerical variables, dummy/categorical variables, or interaction terms. The interpretation of a weight changes with the respective type. For instance, consider a variable x_{sex} representing a binary

[50]The "holding all other variables constant" is also referred to as the ceteris paribus (c.p.)-assumption in econometrics.

attribute, which is either male or female.[51] The dummy variable x_{sex} can take on two possible values: 0 when an observation is male or 1 when an observation is female. While the mechanics of regression with a binary regressor are the same as for a numerical regressor, the interpretation of the β changes. In this case, an increase of x_{sex} by one unit represents the difference between category 0 (= male) and category 1 (= female). Hence, β_{sex} represents the estimated change in y from the (omitted) reference category (male) to female, while all other features are held constant.[52]

Categorical variables have multiple levels. Their interpretation is similar to binary regressors but slightly more complicated, since it depends on the encoding of the respective variable. The encoding process describes procedures by which a categorical variable is converted into a numerical form which can be fed to an optimization algorithm. A popular method for categorical variable encoding is one-hot-encoding, where each category is assigned its own binary column.[53] That is, if an attribute has three levels d, sunny, cloudy, and rainy, then there are $d-1$ binary columns such that the model is not over-parametrized. If an observation matches the attribute, it gets assigned a value of 1 and 0 otherwise. In this case the interpretation is then similar as for a binary regressor: One category is chosen as a reference category, whereas the other categories are modelled as individual coefficients. Changing x_j from its reference category to the coefficient's category increases the expected value of y by β_j given that all other features are held constant. In the case of the example: if the reference category is sunny, then the β_{cloudy} describes the expected change in y from category sunny to category cloudy and β_{rainy} describes the expected change from category sunny to category rainy. The difference between β_{cloudy} and β_{rainy} can be obtained by the subtraction of the β's.

Interaction terms require a particular modelling and interpretation. An interaction term is formed by the product of two regressors, e.g. $x_j \cdot x_{j+1}$. This allows an observer to model the linear dependence between two variables. The interpretation of interaction terms is rather complex, since it depends on the attribute types of x_j and x_{j+1}. For instance, consider the simple case of two binary variables, where x_{sex} is 0, if a person is male, and 1, if the observation is female. Moreover, x_{race} is 0 if a person is of Caucasian ethnicity and 1 if the observation is of Asian ethnicity. In this example we predict wage. Then the coefficient $\beta_{sex \cdot race}$ in front of the interaction term represents the difference in the expected wage of being Asian versus being Caucasian for women and men.[54]

Moreover, the intercept β_0 deserves an interpretation. The estimated prediction of y is β_0, when all numerical features are 0, there are no interactions between variables, and the binary variables are in their reference category. In most cases the intercept cannot be interpreted,

[51] Please note that the variable in the example is not the variable *Sex* from the abalone data set but just an example.

[52] From a statistical point of view, the regression with a binary regressor is equivalent to performing a difference of means analysis.

[53] For other types of encodings of categorical variables see Toutenburg and Heumann (2008b), their chapter 9.5.

[54] In more detail: If the interaction term's coefficient is statistically significant, this states that there is a statistically different effect in the estimated wage for Asians and Caucasians dependent on the respective gender. For a detailed description of the interpretation of binary-numerical and numerical-numerical interaction terms, see Stock and Watson (2015), their chapter 8.3.

because real-world scenarios do not justify the assumption that all other variables are zero.

Molnar (2020) notes that, next to the linear coefficients, the R^2 is also a relevant measurement for the interpretation of linear regression. The R^2 of the regression is the fraction of the variation in the dependent variable that is accounted for (or predicted by) the model. In a univariate regression the R^2 equals the square of the correlation between the dependent and independent variable. The measure is generally of high concern for making predictions, as it can be an indicator of how well a model has learned from the data. A low R^2 indicates that a model does not explain the data very well. Therefore, an interpretation of the coefficients is not meaningful as the variables in the model do not sufficiently explain the data. A high R^2 indicates the converse, i.e. interpretation may be worthwhile. The R^2 is defined as:

$$R^2 = 1 - \frac{SSE}{SST}. \tag{3}$$

It is the counter fraction of the squared sum of the errors (SSE) ϵ_i^2 which is defined as:

$$SSE = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{4}$$

and the squared sum of the total variance (SST) which is defined as:

$$SST = \sum_{i=1}^{N} (y_i - \overline{y}_i)^2, \tag{5}$$

where $\overline{y}$, refers to the mean of the prediction. The SSE is an indicator of the variance which is not explained by the model and SST captures the entire model's variance. Therefore, the R^2 describes the data variation captured by the model as a fraction of the total variance. It is normalized to the range between 0 to 1.

Molnar (2020) comments that the R^2 does not punish a model for incorporating multiple features, even if their informative value is zero. Hence, he proposes to use the adjusted $\overline{R}^2$ which is defined as:

$$\overline{R}^2 = R^2 - (1 - R^2)\frac{p}{N - p - 1}. \tag{6}$$

A similar rationale w.r.t. interpretability holds for the adjusted $\overline{R}^2$: Higher values indicate a good model fit while correcting for non-informativeness of covariates, whereas low values of the $\overline{R}^2$ could imply a poor model fit or the inclusion of multiple non-informative variables.

Eventually, p-values can help determine whether the relationships that one observes in the sample may also exist in the larger population. The p-value for a variable tests the null hypothesis that the variable has no correlation with the dependent variable. If the null hypothesis can be rejected and there is a statistically significant correlation, then the variable should be interpreted, as there is sufficient statistical evidence to conclude that there is an effect of the variable on the outcome.[55]

[55] For more information on statistical hypothesis testing in linear regressions, see Toutenburg and Heumann

5.1.4 Example

Table 5 shows an example for the results for a simple linear regression with the abalone data set. We regress *Rings* on all eight covariates. In this example, the interpretation for a numerical feature goes as follows: increasing the whole weight of an abalone by one gram increases the expected count of rings by 0.045 ($= \beta_{WholeWeight}$), while holding all other features constant. The interpretation for a categorical feature is: If an abalone is still in its infancy, then the expected number of predicted *Rings* is 0.825 ($= \beta_{SexI}$) lower compared to when it is a female (the reference category), given all other features are held constant. Both example features are significant at the 1% significance level suggesting that there is a statistically significant relationship between the independent and dependent variable. The intercept β_0 states that an abalone is 3.895 rings old, if all other variables are zero, which is of course absurd. The R^2 of 0.538 suggests that the linear model is a moderately good fit to the data. Since there is almost no difference in the R^2 and the adjusted $\overline{R}^2$, we can say that there are hardly any non-informative variables in our regression.[56]

Table 5: Linear regression results for Rings

	Dependent variable:
	Rings
SexI	-0.825^{***}
SexM	0.058
Length	-0.002
Diameter	0.055^{***}
Height	0.054^{***}
WholeWeight	0.045^{***}
ShuckedWeight	-0.099^{***}
VisceraWeight	-0.053^{***}
ShellWeight	0.044^{***}
Constant	3.895^{***}
Observations	4,177
R^2	0.538
Adjusted R^2	0.537

Source: Own computation.
Note: Stars indicate the significance at the X% significance level, where $^{*} = p < 0.1$; $^{**} = p < 0.05$; $^{***} = p < 0.01$.

Interpreting linear regressions via regression output is, however, not the only option. Molnar (2020) recommends visual parameter inspection via coefficient and effect plots. We calculate both plots with the linear regression model of table 5. The results for the coefficient and effect plots

(2008b), their chapter 9.3.

[56] The only exception seems to be the categorical variable *Sex* = male. Intuitively, it is likely to assume that there is no age difference between female and male abalones. A look into the data reveals, that the average for males is 10.7 rings while the average for female abalones is 11.1. Hence, it is plausible to assume that there is no statistical difference between the two.

can be found in figure 2 and 3 respectively. A coefficient or weight plot visualizes the coefficient estimates and their variance. The latter is in this case represented as the 95% confidence interval.[57] Since coefficient plots are best used when visualizing all coefficients on the same scale, we scale the features for the coefficient plot to correct for the different units of measurement of the features. That is, we standardize the coefficients to zero mean and unit standard deviation, calculate the linear regression with the processed data, and then plot the coefficients of this linear model. The coefficient plot indicates that the feature *WholeWeight* has a pretty strong positive effect on the prediction of *Rings*. While the variance of this estimated effect is relatively large, as indicated by the confidence intervals, the effect is always positive. On the other hand, *ShuckedWeight* clearly has a strong negative effect on *Rings*. The weight plot indicates that the category *SexI* has a moderate negative effect. On the other hand, the effect of category *SexM* is negligible. The confidence interval contains the value 0, indicating that there is no effect. The confidence interval of *SexM* is small suggesting that the linear model is relatively confident in its estimation.

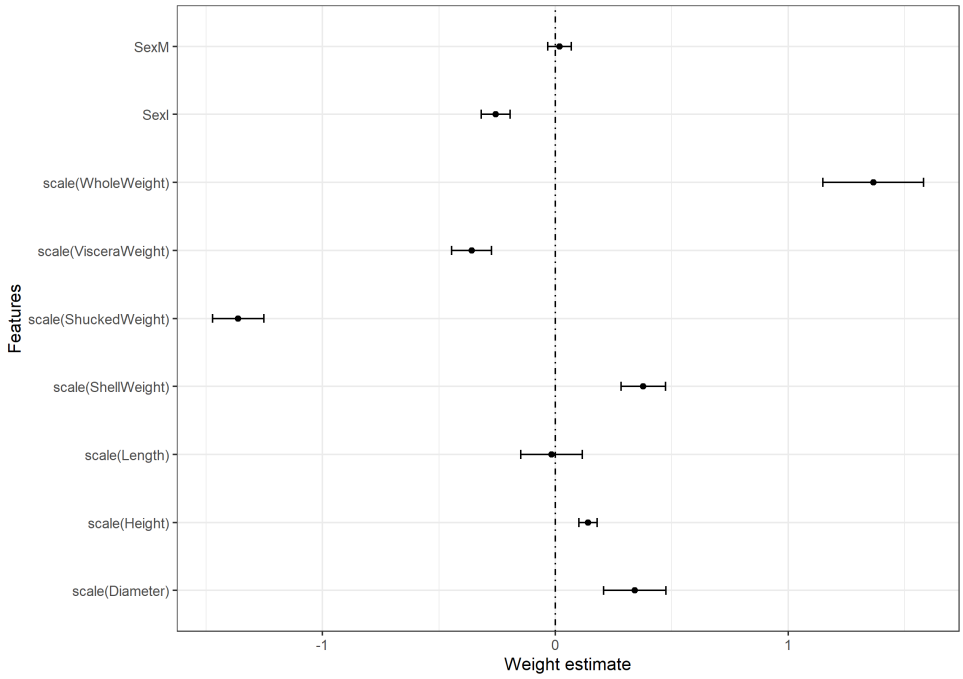


Figure 2: Coefficient plot for linear regression on Rings *with normalized coefficients*

Source: Own computation.
Note: Each dot represents the coefficient's weight. The brackets around each dot designate the 95% *confidence interval. All numerical features have been preprocessed such that their mean is zero and their standard deviation is one.* SexF *is the reference category.*

[57] In statistics, a confidence interval is an interval which specifies the precision of a parameter's estimate. A confidence interval includes the true location of the parameter with a specified probability (the confidence level) in the case of infinite repetitions of an experiment. In this example, a 95%-confidence interval includes the true value of the coefficient with a 95% probability. That is, the interval contains the true value of the coefficient in 95% of all possible randomly drawn samples. Equivalently, it is the set of values of the estimated parameter which cannot be rejected by a 5% two-sided hypothesis test (Toutenburg and Heumann, 2008a).

Another way to make coefficients of different sizes more interpretable is to use an effect plot. In an effect plot the coefficients of the linear model are multiplied by the actual values of the variables. The effect plot helps in understanding how much the combination of a coefficient and an observation's value contributes to the predictions of the linear model. Molnar (2020) defines the effect as: $\text{effect}_{ij} = \beta_j x_{ij}$. That is, in effect plots we visualize the effect for every individual observation i, whereas in weight plots we only plot the feature weights. The effects are visualized as box-plots.[58] Each box describes the respective inner quartile range IQR $(q_{0.75} - q_{0.25})$ of effects. The horizontal line in a box represents the median. The whiskers of the box are graphically represented as $\pm\frac{1.58IQR}{\sqrt{N}}$, whereas the points outside depict the outliers. As supported by the positive coefficients in table 5, the effect plot clearly shows that *WholeWeight, ShellWeight, Height,* and *Diameter* all have a positive effect in predicting the number of *Rings* for most observations. *WholeWeight* exhibits the largest effect. *ShuckedWeight*, *VisceraWeight*, and *Length* have a negative effect on the expected number of rings, whereas results for *Sex* are inconclusive.

An interesting exercise in order to achieve local interpretability with the effect plot is to look at the contribution of each feature for a single prediction. This can be done by choosing a single prediction and calculating the individual effects thereof. The interpretation of a single observation's effect is then best achieved by comparing it with the other feature effects. The red star in figure 3 showcases the effects for a randomly chosen observation, in this case observation 28 in the sample. *WholeWeight, ShellWeight, and Diameter* contribute, for instance, more than the median towards the predicted value. *Sex* has almost no effect on the prediction, which is supported by the insignificant coefficient in table 5. Overall, the linear model predicts a value of 11.40 while the observed value was 12 suggesting that the linear regression for this observation was a moderately close fit.

5.1.5 Discussion of Advantages and Disadvantages

As demonstrated in the previous subsection, a huge advantage of linear regression is their simulatability (Arrieta et al., 2020). Model structure and parameters are easily presentable and understandable for an observer. Moreover, the effects or knowledge the model learns are directly accessible and interpretable in the form of linear parameters. Yet, a dense linear model with complex interaction terms still needs to be decomposed for it to offer the ability to explain each part of the model. Decomposability for linear regressions is, therefore, a matter of the number of predictors and interactions thereof (Arrieta et al., 2020). While OLS is a well-researched method for estimating the parameters in a linear regression, extensive mathematical knowledge and tools are still necessary to understand variables and interactions (Arrieta et al., 2020). Moreover, Molnar (2020) notes that linear regression can only represent non-linearities in a very

[58] A box-plot or box-and-whisker plot is a method which graphically represents different distributional location parameters, such as the position of the median, 25% as well as 75% quartiles, extreme values, and outliers in order to give an impression of the data's concentration and symmetry (Toutenburg and Heumann, 2008a). The distribution characteristics highlighted in box plots may vary according to the respective author's interpretation of what a box-plot constitutes.

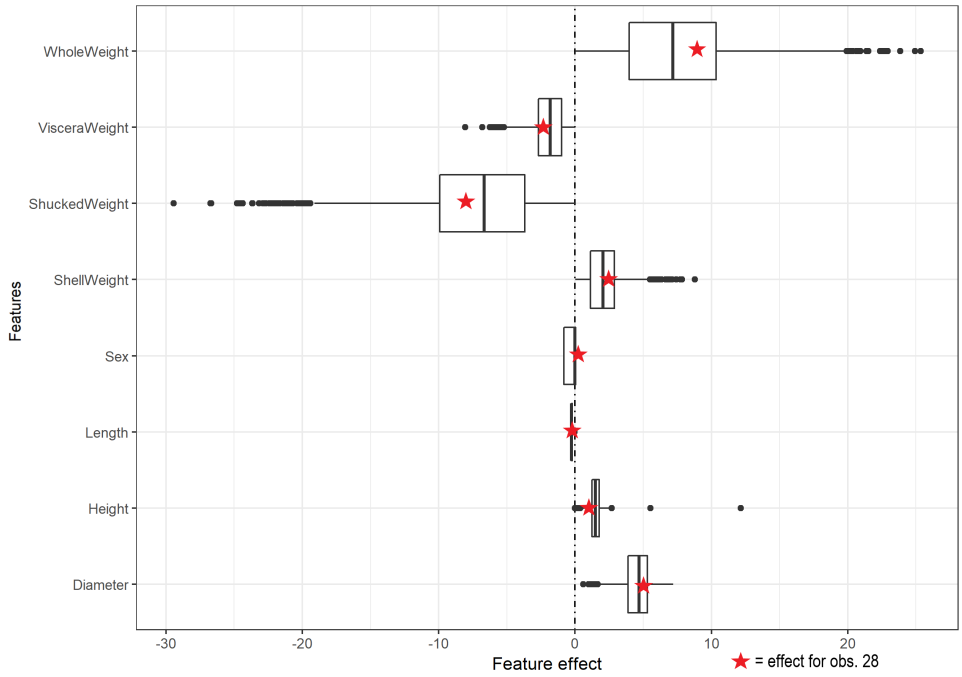


Figure 3: Effect plot for the linear regression on Rings

Source: Own computation.
Note: Each box describes the inner quartile range IQR $(q_{0.75} - q_{0.25})$ *of each effect. The vertical line in each box represents the feature's median value. The box-plot whiskers or horizontal lines correspond to* $\pm\frac{1.58IQR}{\sqrt{N}}$. *All points above these values are considered as outliers.*

complex way. That is, non-linearities such as U-shaped effects or interactions have to be explicitly delivered to the model. He furthermore claims that the interpretation of a linear regression coefficient can be unintuitive, since it always depends on holding the other features constant. While this is neat when one wants to interpret a single effect in isolation, it hardly reflects the complexity of reality. Most data sets exhibit some form of complex correlation which linear regression simply fails to identify and account for. Eventually, Molnar (2020) claims that linear regression often cannot compete with more complex models in performance.

In summary, linear regressions are a useful tool to get a superficial understanding of interpretability. While linear effects can be a powerful instrument and a good approximation of certain relationships, linearity is also a substantial restraining factor when modelling complex relationships. Reality is eventually more complicated and houses non-linear effects and complex correlations between variables which is why more complex ML models are sometimes needed to offer more sophisticated modelling possibilities of effects and a better overall prediction performance compared to linear regression.

5.2 Decision Trees for Classification

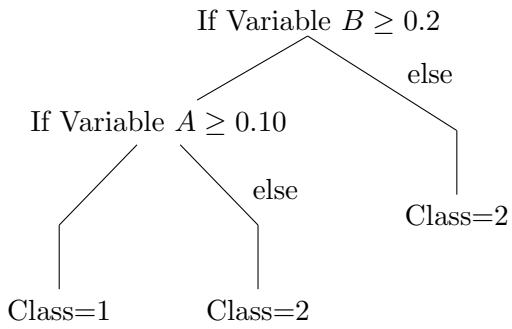
5.2.1 Decision Tree Models

Decision trees for classification are within the family of tree-based models. They consist of nested if-then statements for the predictors which try to partition the predictor space into a set of rectangles. For every partition a constant or in the case of classification trees, a class (e.g. the mode) is predicted. The rules used to separate the data can be graphically represented as a tree structure, which is where the technique derives its name from. For instance, if a simple decision tree for a binary classification tree with two variables (Variable A and Variable B) reflects the following rules:

```
if Variable B >= 0.2 then
    if Variable A >= 0.10 then Class = 1
    else Class = 2
else Class = 2
```

then an example of a graphical representation of the decision tree can be found in figure 4. Every internal node represents a splitting decision. In case the if-statement is true, then one follows the left branch, else one goes right.

Figure 4: Example decision tree



Source: Own illustration.

In this example, two splits of the data separate the predictor space into three regions. The last region of a tree is also called terminal node or leaf in the terminology of tree models. The inner splits or intermediate subsets are also often referred to as internal nodes or split(ting) nodes. In each terminal node the algorithm categorizes an observation either into Class 1 or Class 2. When predicting a new observation one simply follows the if-then-statements defined by the tree with the observations' values until a terminal node is reached. The prediction then corresponds to the class of the terminal node. Decision trees are closely related to decision rules. Decision rules describe a set of if-then conditions which have been collapsed into independent conditions. In fact, transforming decision trees into decision rules and vice-versa is relatively straightforward (Quinlan, 1987a,b). For example, the tree above written as a decision rule results in the following three independent conditions:

```
if Variable A >= 0.1 and Variable B >= 0.2 then Class = 1
if Variable A >= 0.1 and Variable B < 0.2 then Class = 2
if Variable A < 0.1 then Class =  2
```

The more complex a tree's form is, the harder is the transformation from trees to rules. Similarly, complex decision rules require multiple nodes to depict the corresponding decision rules as a tree.

5.2.2 Creating a Decision Tree

To formalize the notion of decision trees: If a sample consists of p variables, a response class c_l with k-maximum classes $l = 1, ..., k$, and N observations, then a decision tree algorithm separates a tree into M regions $R_1, R_2, ..., R_M$ and predicts the responses of the observations that belong to the same region according to a certain criteria, such as the modus, mean, or median.[59] According to Hastie et al. (2009), a decision tree predicts each instance x_i as follows:

$$\hat{y}_i = \hat{f}(x_i) = \sum_{m=1}^{M} c_m I\{x_i \in R_m\}. \tag{7}$$

Each observation x_i is predicted into one of M subsets R_m. The identity function $I\{x_i \in R_m\}$ returns 1 if x_i is in R_m and 0 otherwise. If an instance x_i is predicted into a leaf node R_j, then the prediction $\hat{y}_i$ is class c_j.

The aim of classification trees is to split the data into small and homogeneous groups. In this context, homogeneity describes that the nodes of the split contain a larger proportion of one class in each node, i.e. the nodes of the split are more pure. The concept of purity needs to be measured somehow to make it a quantifiable decision criteria. Purity in classification trees can be defined as minimizing misclassification, which is in turn equivalent to maximizing accuracy.[60] Using accuracy as a measure of purity can be problematic, however, since it focuses on separating the data such that misclassification is minimized rather than focusing on separating the data such that observations are primarily placed in one class (Kuhn and Johnson, 2013). Hastie et al. (2009) propose three different purity measures for splitting nodes and pruning a tree. If a node m represents a region R_m with N_m observations then the proportion $\hat{p}_{mk}$ of class k observations in node m can be written as:

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k). \tag{8}$$

The observations in node m are classified to class $k(m) = \arg\max_k \hat{p}_{mk}$, i.e. the modus class in node m. They further propose three measures of node (im-)purity:

[59] The most frequently chosen distribution parameter for classification trees is usually the modus.

[60] Purity measures are also often referred to as impurity measures in the literature when they designate the impurity of a node. That is, impurity describes the opposite concept of purity. Since maximizing purity is the same task as minimizing impurity, both terms are often used interchangeably.

$$\text{Misclassification error} = \frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk}. \tag{9}$$

$$\text{Gini Index} = \sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}). \tag{10}$$

$$\text{Cross-entropy or deviance} = -\sum_{k=1}^{K} \hat{p}_{mk} log(\hat{p}_{mk}). \tag{11}$$

Both, the cross-entropy and Gini Index, are differentiable and, hence, preferable for numeric optimization. Yet, both measures are more sensitive to node probabilities than to the misclassification error. From an interpretability perspective it is interesting to note that trees are formed by identifying features and split points which result in the largest information gain for a given impurity/purity measure. The actual selection of criteria for growing trees does not provide substantial information for the interpretation process.

In some cases, when trees are grown too large, the technique of tree pruning may help in removing branches of the tree which provide little to no information gain to classify instances.[61] Tree pruning can help reduce the complexity of a tree which in turn facilitates interpretation.

5.2.3 Interpretation

The interpretation of decision trees is relatively straightforward. One starts at the root node and goes down to the next splitting node. At each splitting node an if-else statement tells one at which subset of the data one is looking at. Once you have followed down the tree and there are no splitting nodes, a tree leaf is reached which states the predicted outcome class. The interpretation of a tree is basically a concatenation of multiple if-else statements. There is no difference in interpretation between categorical and numerical variables for decision trees. All features in the junctions are modelled as if-else statements.

5.2.4 Example

Figure 5 contains the graphical representation of a decision classification tree fit on the abalone data set. We predict the variable *AgeCat*, which is 1 if the abalone is older than 9 rings and 0 if it is younger. The decision tree predicts the category that occurs most frequently in the respective final node. The classification and regression tree algorithm in Breiman et al. (1984) was used to create the tree. The Gini index serves as the purity measure. The decision tree has an accuracy of 78.81% on the training data. Each node shows, first, the predicted class ("older than 9 rings" or "younger than 9 rings"), second, the predicted probability of being older or younger, and third, the percentage of all observations in this node. We use observation 28 again in order to exemplify an interpretation of the prediction for this instance. Observation 28 has a *ShellWeight* value of 56. In this case the model interpretation is relatively simple. The first node

[61] For more information on tree pruning see Hastie et al. (2009), their chapter 9.

asks "if *ShellWeight* < 50" go down the left branch of the tree, otherwise go right. Because the *ShellWeight* of observation 28 is higher, we can just follow the right branch and arrive at the final node, where the class prediction is "older than 9 rings". The model is relatively confident in its prediction with a 78% probability of predicting class 1 as indicated by the decimal figure in the box. 46% of all observations in the sample go into this node. If the value of *ShellWeight* was smaller than 50 we would have to go down the left decision tree and use the next junction to decide which branch to follow. Since the observed value of *Rings* for observation 28 is 12 which is larger than 9, we can say that the node predicted correctly for this instance. Table 7 contains the values for observation 28.

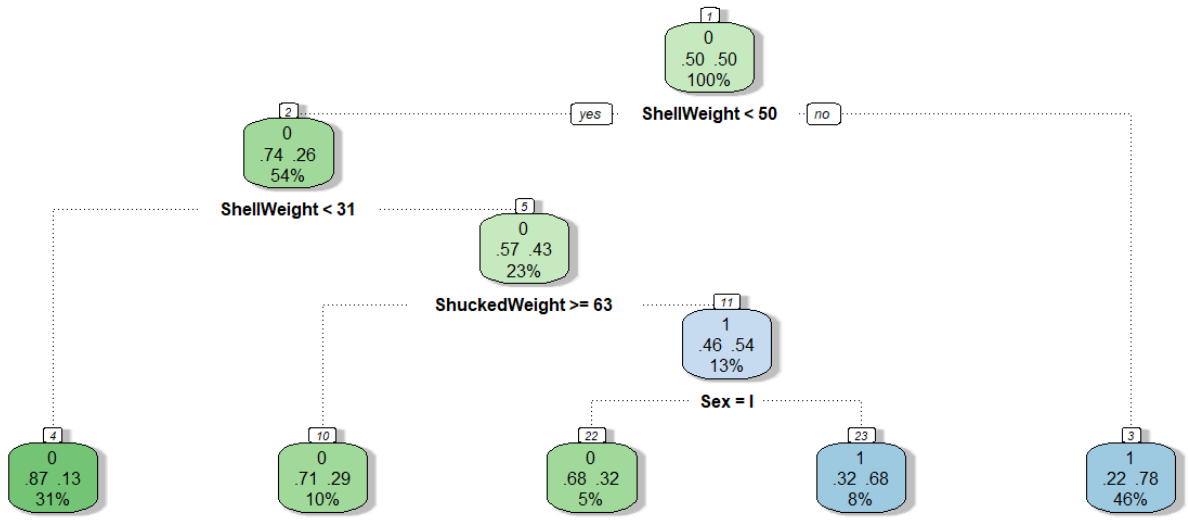


Figure 5: Decision tree for AgeCat

Source: Own computation.
Note: The tree was created with the classification and regression tree algorithm (Breiman et al., 1984). The predicted variable is AgeCat. *The Gini index serves as the impurity measure. Each node shows the predicted class, where a blue box signifies "older than 9 rings", i.e.* AgeCat $= 1$, *and green denotes "younger or equally young as 9 rings", i.e.* AgeCat $= 0$. *The first number in each box shows the predicted class. The predicted probability of being younger or older than the mean is displayed as point decimals in the box. The percentage of observations in the node is described by the last figure in the box.*

5.2.5 Discussion of Advantages and Disadvantages

Decision trees are extremely straightforward to explain and are even considered easier to interpret than linear regression (James et al., 2013). They can be visualized in their natural form, such that they are even interpretable by non-ML experts. Simulatability-wise they score exceptionally high, according to Lipton (2018). This may come from the fact that decision trees realistically mimic human decision-making, since they challenge you to think about single instances in a counterfactual way (Molnar, 2020). For instance, a counterfactual explanation would go like this: "if observation 28 had a value of 39 for *ShellWeight*, then we would have to go down the left decision tree". This makes it straightforward for an observer to process the entire sparse tree at

once. Furthermore, decision trees can also deal with non-linear relationships and interactions pretty well because of their nested structure (James et al., 2013). Molnar (2020) emphasizes additionally the ease of explanation in the prediction form. Because data instances are predicted in groups, they may be easier to understand than, for example, coefficients in a linear regression which represent points on a hyperplane. From a decomposability perspective decision trees seem desirable as well. The model's decision process can be easily read off in a humanly understandable form from the tree.

Yet, trees generally cannot compete with the predictive performance of more complex classification approaches (James et al., 2013). Because of the simplicity in their decision algorithm, they often fail to identify especially linear relationships, as the effect of a feature can only be approximated by splits in the feature space (Molnar, 2020). Splitting a feature into multiple bins to model a linear effect step-wise is very inefficient compared to other modelling approaches. This can be especially critical for input-sensitive feature splits, where marginal changes in the feature x lead to considerable changes in the output y. Moreover, Molnar (2020) states that trees can exhibit non-robust behaviour. That is, smaller changes in the data can cause the internal splitting nodes to completely change, which may create a different tree, and eventually alter predictions substantially. Techniques such as bagging, boosting, and random forests can significantly improve the accuracy of decision trees (Hastie et al., 2009). The consequence of this is, however, that a more sophisticated decision process makes the model's decisions more opaque.

In summation, decision trees offer an incredibly straightforward interpretability potential. Almost no other model comes close to explaining its decisions in such a transparent and humanly comprehensible way. Their biggest weakness is their predictive accuracy and their high sensitivity to small changes in the training data which may alter the tree and its predictions completely.

6 Global Post-Hoc Model-Agnostic Interpretability Techniques

In this section we introduce five model-agnostic interpretability techniques. During this, we will explain and discuss partial dependence plots (Friedman, 2001), individual conditional expectation curves (Goldstein et al., 2015), accumulated local effects (Apley, 2016), global surrogate models (Molnar, 2020), and local interpretable model-agnostic explanations (Ribeiro et al., 2016b). Each sub-chapter starts with a brief explanation of the method's theoretical background after which we will demonstrate how to interpret these techniques on an empirical example from the abalone data set. We focus on a particular set of features (*WholeWeight, ShuckedWeight, Length* and *Sex*) to allow the comparison of results between different techniques. All interpretability techniques introduced in this chapter are primarily used to investigate the model's predictive behaviour.[62] Each subsection concludes with a discussion of advantages and disadvantages of the respective technique.

6.1 Partial Dependence Plots

6.1.1 Friedman's (2001) Partial Dependence Plots

Friedman's (2001) partial dependence plots (PDPs) are a model-agnostic interpretability technique which has proven to be quite successful in visualizing black box prediction methods in multiple disciplines (Adadi and Berrada, 2018; Goldstein et al., 2015). They facilitate the visualization of the relationship between a subset of the features and the predicted response, while accounting for the average effects of other predictors in the model. Partial dependences can illuminate the trajectories of one or multiple feature effects over the entire feature space by computing the marginal effect of a set of feature vectors on the predicted outcome.

A partial dependence, according to Friedman (2001), is defined as the partial dependence of a predictor function $f(\boldsymbol{x})$ on one or more selected variables x_s, after integrating over the marginal distribution of the remaining variables x_c. Both x_s and x_c are subsets of the entire set of variables $\boldsymbol{x}$. Moreover, $x_s \subset \boldsymbol{x}$ and x_c is the complement set of x_s such that $x_c \cup x_s = \boldsymbol{x}$.[63] The partial dependence f_{pd} can then be defined as:

$$f_{pd}(x_s) = E[f(x_s, x_c)] = \int f(x_s, x_c)\, \mathrm{d}P(x_c). \tag{12}$$

Each subset of the predictors x_s has its own partial dependence function which shows the average value of the predicted response for different values of the selected predictor x_s, while the complementary vector x_c is varied over its marginal distribution $\mathrm{d}P(x_c)$. Since neither the true

[62] Identifying to what extent the ML model's features model underlying biological causes is not the primary goal of these techniques. They are merely used to identify what the model learns from these features. It is always conceivable that the underlying ML model cannot deal with the nature of an effect even though there is a clear causal relationship between the feature and the prediction. In this thesis we do not want to illustrate how causal inference can be done with such techniques but rather showcase what possibilities model-agnostic techniques offer and which thought processes they can trigger in an interpretability analysis.

[63] In the following example x_s represents a vector for a single feature. However, as stated above, x_s can also be a matrix when two or more features are under investigation.

model of partial dependence function f_{pd} nor the marginal distribution $\mathrm{d}P(x_c)$ are known, the estimated partial dependence $\widehat{f_{pd}(x_s)}$ is usually derived via Friedman (2001):

$$\widehat{f_{pd}(x_s)} = \frac{1}{N}\sum_{i=1}^{N}\hat{f}(x_s, x_{c_i}). \tag{13}$$

That is, in order to calculate the partial dependence for a specific value of x_s we average over the values $\{x_{c_1}, x_{c_i}, ..., x_{c_N}\}$ which refer to the different values of x_c observed in the training data. If we want to calculate all partial dependences for a feature x_s, we have to perform the calculation in the equation above for all values of x_s.[64] Note that we approximate two different components (Goldstein et al., 2015): The integral over x_c is approximated by taking the arithmetic mean of all N x_c values, while the true model is approximated by $\hat{f}$ which is the ML model to be explained. That is, we approximate the integral of x_c in order to visualize the effect which is itself an approximation of the true effect.[65] During this, we assume the correlation between x_c and x_s to be zero, which can be problematic in certain cases, as we will show in the following.

Partial dependences are best visualized to investigate the effect's behaviour and identify trends or peculiarities thereof. If we calculate the effect $\hat{f}_{s_i}$ at each x_{s_i} in the data, we receive a set of N ordered and paired results $\{(x_{s_i}, \hat{f}_{s_i})\}\ \ \forall\ \ i = 1, ..., N$, where - as a reminder - i is the index for the i^{th} observed instance. Friedman (2001) suggests that the $\hat{f}_{s_i}$ are best plotted as a function of x_s where each value is joined by lines. The resulting graphics are known as PDPs. For regression the $\hat{f}$ represents the value of the predicted outcome for the feature x_s, whereas for classification the partial dependence showcases the estimated probability for a certain class given different values for the x_s. PDPs can be used for any supervised learning model.

6.1.2 Example Interpretation

PDPs make a statement about the effect of a single or multiple independent variables on the predicted outcome. Therefore, they are best explained by using empirical examples. We first calculate a random forest model with the dependent and numeric response *Rings* to create the black box model which we want to explain. This random forest model will serve as a main example for an exemplary regression task throughout this chapter. Thus, we can show for each introduced interpretability technique where the model accounts for a feature effect well, where there is potential for improvement, and which features might not be useful at all. The random forest model has a mean squared error of 1.034 on the training data suggesting a moderately good fit which is sufficient for our analysis.[66] Table 8 in the appendix A shows node purity and standard deviations of the model for the interested reader. After training the random forest model, we compute the partial dependences on the training data and the corresponding predictions of this model. Figure 6 shows the partial dependences of the four variables: *WholeWeight*,

[64] For a more detailed explanation of the calculation of PDPs and ICEs of the subsequent chapter please see appendix B sub-chapter 2.

[65] The calculation of averages in the training data is also known as Monte Carlo method (Molnar, 2020).

[66] The mean squared error is often used to assess forecasting performance. It is calculated as the mean squared difference between the observed values and the forecasted values.

ShuckedWeight, *Length*, and *Sex* for the dependent variable *Rings*. The order goes from top-left to bottom-right. The rug under each chart represents the feature's empirical distribution. That is, the first and second mark indicate the first decile, the third and fourth the second decile and so on. All four partial dependence trajectories exhibit some shakiness, which comes from the fact that the underlying model is a random forest which partitions the sample into specific predictor subspaces via step functions. For smaller values of *WholeWeight* the model predicts a modest amount of *Rings*. This effect, however, almost linearly rises with an increase in *WholeWeight* until it hits a plateau of around 400 grams, after which, an increase in *WholeWeight* only seems to have a negligible effect on the prediction. It has to be noted, however, that, as indicated by the feature distribution at the bottom of the chart, there are hardly any sample observations for this plateau. One could even go as far as saying that values over 400 grams can be considered as outliers. It is debatable whether these extreme values should even be included in the model. The overall positive effect of *WholeWeight* on the predicted number of *Rings* conforms with previous results in the linear regression.

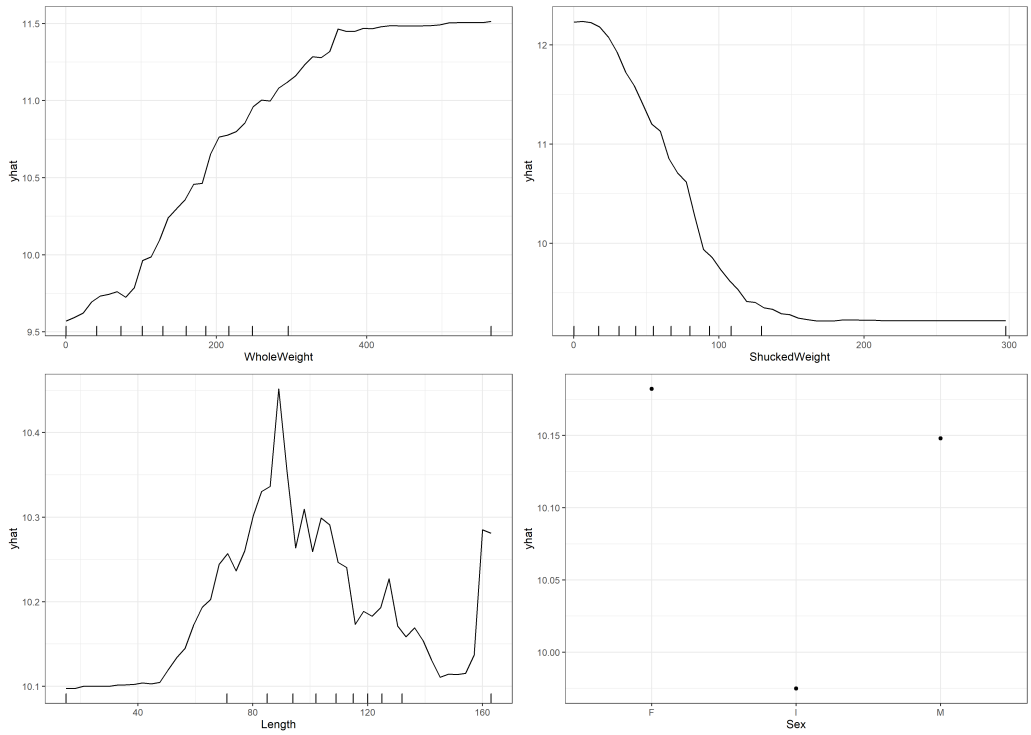


Figure 6: PDPs of WholeWeight, ShuckedWeight, Length, *and* Sex *for* Rings.

Source: Own computation.
Note: From left to right each plot shows the individual partial dependences of the variables WholeWeight, ShuckedWeight, Length, *and* Sex *calculated from a random forest model with the dependent variable* Rings. *Variable* Sex *is categorical with the values female (F), infant (I), and male (M). Each x-axis shows the scale of the respective variable, while the y-axis displays the average value of* $\hat{f}$ *which is numerical. The lines at the x-axis indicate a rug, where two lines represent the feature's decile.*

A contrary picture is drawn by the effect of *ShuckedWeight*. Small values of *ShuckedWeight* seem to evoke predictions of a high number of *Rings*. Biologically speaking, maybe this implies

that older abalones loose meat weight as they grow older, similar to living organisms on land.[67] Higher values of *ShuckedWeight*, lead to lower predictions of the number of *Rings*. This effect is steeply decreasing until around 120 grams of *ShuckedWeight*. There seems to be hardly any change in prediction for values above 120 grams. Once again, there are hardly any values to consider for the calculation of this tail end of the partial dependence, as indicated by the feature's empirical distribution. *ShuckedWeight* also had a negative coefficient in the linear regression mirroring the direction of the partial dependence. Moreover, the decision tree shared a similar rationale by predicting *ShuckedWeight* values above 63 grams to be young, i.e. below 9 rings old (see 5, node 5). So far, all models share the consensus of a negative effect of *ShuckedWeight* on the number of *Rings*. However, we can hardly compare the size of the coefficients .[68] Since *ShuckedWeight* is correlated with *WholeWeight*, the effect may be more complex than it appears.

The effect for *Length* certainly has a more complex form. *Length* seems to exhibit a bell-shaped effect on the prediction. For values from 50 grams to around 90 grams the model predicts an increasing effect of *Length*, whereas this effect decreases again until around 135 grams. Naturally, younger abalones are smaller in *Length* as they grow with age, whereas older abalones seem to shrink with age. Another explanation could be that larger abalones are more prone to be hunted by their natural predators such as fish, rays, or sea otters, as they are more exposed because of their larger size. Alternatively, it is conceivable that *Length* just has no effect at all. Overall, it is hard to clearly identify the shape of the effect, as it appears quite rugged. Turning back to table 5, we find that the coefficient of *Length* in the linear regression was not statistically significant which may explain the non-linear and quite frankly confusing shape of the effect here. Keep in mind that these considerations are only conjectures about the potentially underlying biological causes.

Eventually, the last chart shows the effect for the categorical variable *Sex*. In contrast to the numeric variables, the partial dependence is a point estimator for each category. The predicted age for female abalones is the highest, with males coming in second place. What is true for humans is also true for abalones: women apparently live longer than men. This difference seems, however, negligible, judging by the distance between the two distinct effects in the y-scale. Non-gendered (infant) abalones are naturally predicted to be younger than adult ones. Overall, this supports the previous results from the linear regression and decision tree model, where *Sex* mostly seemed to exhibit very little predictive influence, but there was a difference in the predictions between infants and females/males.

PDPs can also come in a two-dimensional form, where each dimension represents the main effect of a feature. We perform a two-dimensional PDP analysis for the features *WholeWeight*

[67] It is conceivable that next to old abalones, infant abalones may also exhibit a smaller meat weight as they still need to grow. Yet, the Tasmanian government department of Primary Industries, Parks, Water, and Environment explicitly prohibits the fishing of young abalones which is why there are few young abalones in the sample. In fact, this is supported by the distribution of *Rings* which is, with a skewness of 1.11, mildly right-skewed. Thus, we can rule out with certainty that young abalones do have little meat weight.

[68] Interestingly enough, if we roughly estimate the slope of the PDP with a back of the envelope calculation, the coefficient $\beta_{ShuckedWeight}$ of the linear regression and the slope of the PDP are of about the same order of magnitude. $\beta_{ShuckedWeight}$ is -0.099 and the slope coefficient of the PDP is roughly: $\frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} = \frac{9.75 - 12.25}{200 - 0} = 0.0125 \approx -0.099$. This comparison could be purely coincidental and should, therefore, be treated as anecdotal.

and *ShuckedWeight*. That means that $x_s = \{x_{WholeWeight}, x_{ShuckedWeight}\}$. Figure 7 shows the partial dependence of *WholeWeight* on the x-axis and the partial dependence of *ShuckedWeight* on the y-axis. Darker tones of blue indicate a lower predicted age, whereas brighter tones of yellow indicate the opposite. Clearly, there are two polar prediction regions. Lower values of *WholeWeight* and higher values of *ShuckedWeight* result in the prediction of a small number of *Rings* as indicated by the large dark blue area. On the contraire, high values of *WholeWeight* and low values of *ShuckedWeight* lead to the prediction of a large amount of *Rings*, as shown by the bright yellow spot. Maybe this implies that younger abalones have more meat, while their shell still has to grow which is why their overall weight is lower. Older abalones usually weigh more, but most of this is shell weight and not the weight of the meat. PDP analyses for more than one effect can expose the more complex behaviour of variables and potential apparent interactions between two features. It is, thus, plausible to assume that *WholeWeight* and *ShuckedWeight* are correlated or there is some sort of interaction between the two. Figure 7 reveals that the effect of *WholeWeight* and *ShuckedWeight* is more complex than figure 6 suggests. The appendix offers two more two-dimensional plots for the interested reader. The first one showcases the two-dimensional PDP of a numeric and categorical variable (*WholeWeight* and *Sex*, see figure 23). The second two-dimensional plot for *WholeWeight* and *Length* illustrates the step function prediction behaviour of the random forest model very well (see figure 24). That is, since the random forest separates the predictor space into several regions, in the two-dimensional case rectangles, the lines of the decision regions in the PDP coincide with the splitting nodes for *WholeWeight* in the random forest model.

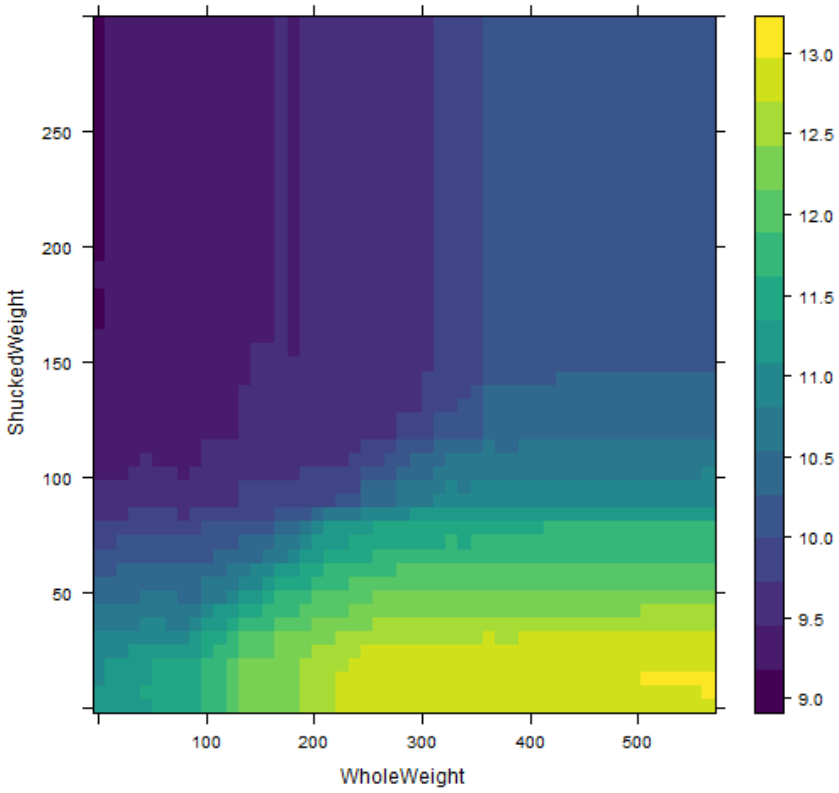


Figure 7: Two-dimensional PDP of WholeWeight *and* ShuckedWeight *for* Rings

Source: Own computation.
Note: Plot shows the partial dependences of WholeWeight *and* ShuckedWeight *calculated from a random forest model with the dependent variable* Rings. *The x-axis shows the scale of* WholeWeight, *whereas the y-axis shows the scale of* ShuckedWeight. *The color scale displays the average value of* $\hat{f}$, *in this case the number of* Rings. *A brighter tone indicates a higher prediction, a darker tone shows a lower prediction.*

Moreover, we present a three-dimensional PDP for the variables *WholeWeight*, *ShuckedWeight*, and *Length* in figure 8. The partial dependence is illustrated as a plane in the three-dimensional space of the variables. Each facet of the plot represents a partial dependence for a subset of the variable *Length*, as indicated by the bar above each chart. Interpreting the plot is relatively difficult, since the illustration is static and differences in the partial dependence surfaces can hardly be observed without moving the plot along its dimensions. One apparent observation, however, is the steep slope along the *ShuckedWeight*-axis, which - size-wise - seems to dominate the shape of the entire surface. Increasing this feature leads to a considerable drop in the predicted age. The effect of *ShuckedWeight* increases for larger values of *WholeWeight* across all four charts. Observing the effect of *WholeWeight* in isolation, reveals nothing new compared to figure 6: Increasing the value for *WholeWeight* increases the predicted age even when varying *ShuckedWeight* and *Length*. With regard to the feature *Length* we can observe that medium to high values of *Length* lead to a higher predicted age, as indicated by the top-left, top-right and bottom-right panel. All effects observed in the three-dimensional plot need to be interpreted while accounting for the other effects at the same time, since it can happen that one remarkable shape of

a feature dominates the error surface and overshadows other intricate effect behaviours. Overall, it can be recommended to perform a partial dependence analysis with three-dimensions only in cases when the features are naturally interpretive and there is a clear relationship between them motivated by their real-world relationship. Only then can minor differences in the predictions be observed in more detail.

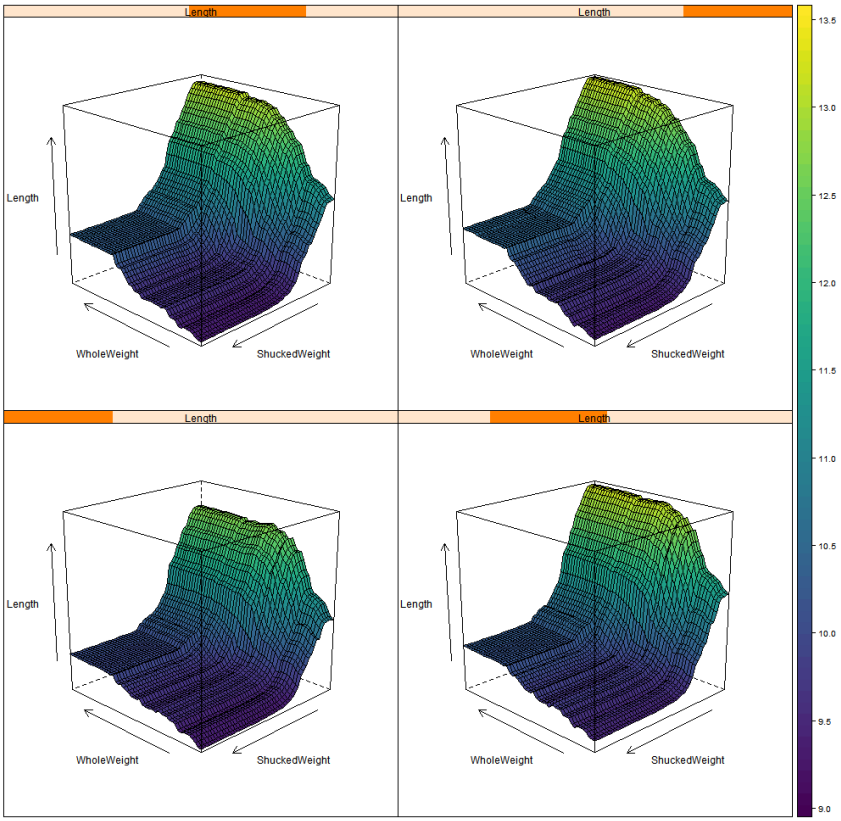


Figure 8: Three-dimensional PDPs of WholeWeight, ShuckedWeight, *and* Length *for* Rings

Source: Own computation.
Note: Plots shows the combined partial dependence surface of WholeWeight, ShuckedWeight, *and* Length. *Each plot represents a subsample of the feature* Length. *Each axis represents one feature dimension. The color scale displays the average value of* $\hat{f}$*, in this case the number of* Rings. *A brighter tone indicates a higher prediction, a darker tone shows a lower prediction.*

Similarly, we can perform a PDP analysis for a classification model. Figure 9 shows four partial dependences for *WholeWeight*, *ShuckedWeight*, *Length*, and *Sex* from the top-left to the bottom-right corner. We fit a support vector machine (SVM) model for the binary response *AgeCat* on all covariates. This model will serve as a main example for an exemplary classification task throughout this chapter. The SVM model has a training accuracy of 80.27%. The rug under a chart represents the ten deciles of the feature's distribution again. Please note that the response variable describes the predicted probability of falling into category 1. As can be seen in the first chart in figure 9, the effect of *WholeWeight* on the probability of being old has an S-shaped trajectory. Small values of *WholeWeight* below roughly 140 grams decrease

the predicted probability of being old. Values above this threshold increase the probability for being old. The effect of *WholeWeight* appreciates in value until around 300 grams after which it converges to around an estimated prediction $\hat{y}$ of 0.4. The section of x_s until 300 grams is where 90% of the *WholeWeight* observations fall, as indicated by the rug. The effect of *ShuckedWeight* on the predicted probability of being old has an inverse bell-shaped form. Values until around 75 grams have a positive effect on the estimated probability, whereas values from there to around 115 grams have a negative influence. The partial dependence rises thenceforth until it hits a plateau at around 200 grams. However, this last section of curvature only represents around 10% of the sample. The effect of *Length* has a similar parabola-shaped effect which has its minimum at around 80 grams but never actually has a negative effect on the prediction.

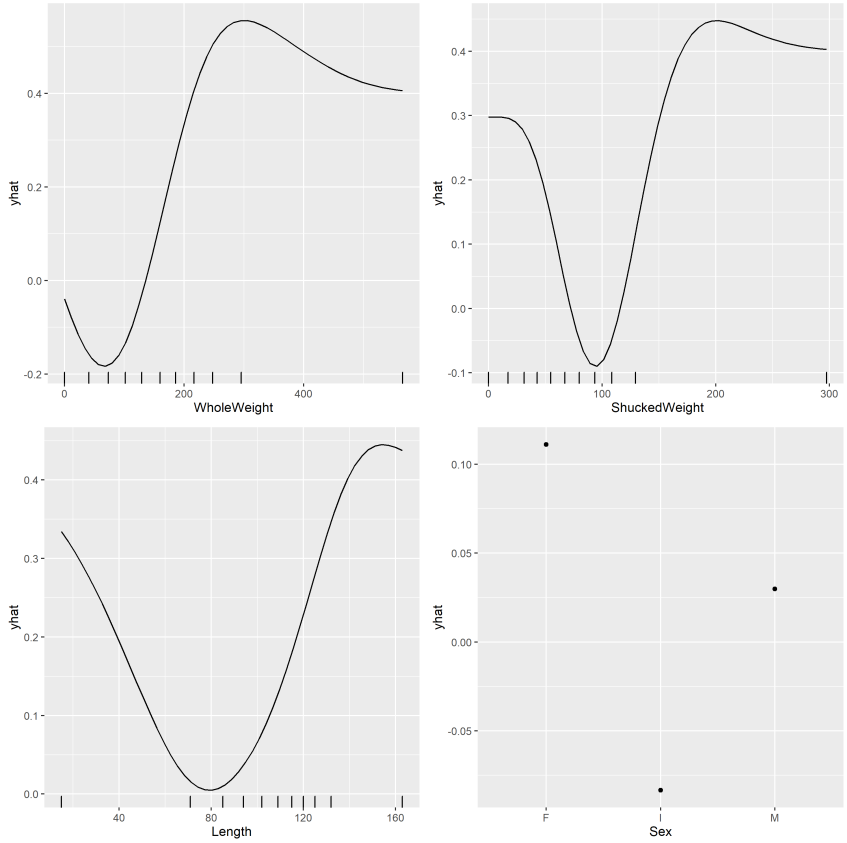


Figure 9: PDPs of WholeWeight, ShuckedWeight, Length, *and* Sex *for* AgeCat

Source: Own computation.
Note: From left to right each plot shows the individual partial dependences of the variables WholeWeight, ShuckedWeight, Length, *and* Sex *calculated from a support vector machine model with the dependent binary variable* AgeCat. *Variable* Sex *is categorical with the values female (F), infant (I), and male (M). Each x-axis shows the scale of the respective variable, while the y-axis displays the average value of* $\hat{f}$ *which represents a probability. The lines at the x-axis indicate a rug, where two lines represent the feature's decile.*

The partial dependence for *Sex* is consistent with previous findings. Females and males are predicted as old, whereas infants are predicted as young.[69]

[69] A proxy measure of how important a feature is in the prediction can be the number of splitting nodes dedicated

For completeness' sake, figure 10 shows a two-dimensional PDP for *WholeWeight* and *ShuckedWeight*. There are two particularly adjacent polar regions. One circular region is centered at around $WholeWeight = 100$ and $ShuckedWeight = 100$. Predictions in and around this region have a small but negative effect on the estimated probability of being old. The other smaller circular region at $WholeWeight = 220$ and $ShuckedWeight = 40$ unites positive effects of the estimated probability. The circular shape could originate from the model under interpretation which is an SVM model with a radial kernel. The interpretation of two- and three-dimensional partial dependences is similar as for PDPs on ML models with numeric response which is why we refrain from going into more detail at this point. The only difference is that the predicted outcome of PDPs of one feature is a probability, whereas the PDPs of multiple features are described by probability surfaces.

In summary, it can be stated that the interpretation of partial dependences for a numerical response is relatively straightforward. Different shapes of the effects can be clearly identified. Variables with large effects or inconclusive effects can easily be recognized as such. Moreover, it is important to always check the size of the change in the predicted outcome by looking at the scale of the y-axis. Thus, the size of the effect can be made clear.[70] When interpreting partial dependences for a classification model it is particularly interesting to note where the zero point of the y-axis is, in order to see which section of the feature distribution has a negative or positive effect on the predicted probability. The y-axis is, hence, more indicative for partial dependences of classification models. Furthermore, it is important to always include the empirical distribution of the feature in a PDP. As shown above, incorporating such information is relevant for the interpretation of PDPs, since some shape at the tails of the PDP may suggest an effect, which only holds for a small fraction of the feature distribution. Moreover, it has to be checked whether parts of the effect are caused by extreme outliers in the data or by a majority of the sample. It is important to assess the behaviour of a model not only at the tails but also for extreme predictions caused by extreme outliers which heavily influence the PDP in the averaging process. One additional interesting thing to note is that certain characteristics of the ML model which the partial dependence tries to explain still are visible in the plots even though the model is treated as a black box. The shaky shape of the trajectory of the random forest model in figure 6 or the radial shape in figure 9 caused by the radial SVM kernel clearly show this.

6.1.3 Discussion of Advantages and Disadvantages

Partial dependences offer an excellent summary of the reduced feature space on a global model scale. They are best interpreted in visualized form. As shown above, analysing PDPs is straightforward and intuitive, since the entire complexity of an effect can be reduced into one

to this feature. Since the effect of *Sex* was small in previous models, we want to investigate if there are actually splitting nodes of *Sex* in the random forest model. 47 of the total $2,751$ splitting nodes of the random forest tree are splitting nodes for the feature *Sex*. That amounts to less than 2%, which seems rather low and confirms the fact that *Sex* does not have a large role in the prediction of *Rings* in the random forest model. A more suited proxy measure would be the calculation of the entropy caused by the nodes of *Sex*.

[70] Another alternative would be to calculate the average prediction and include it as a horizontal line in the chart such that predictions above or below the line can point out whether an effect is more positive or negative.

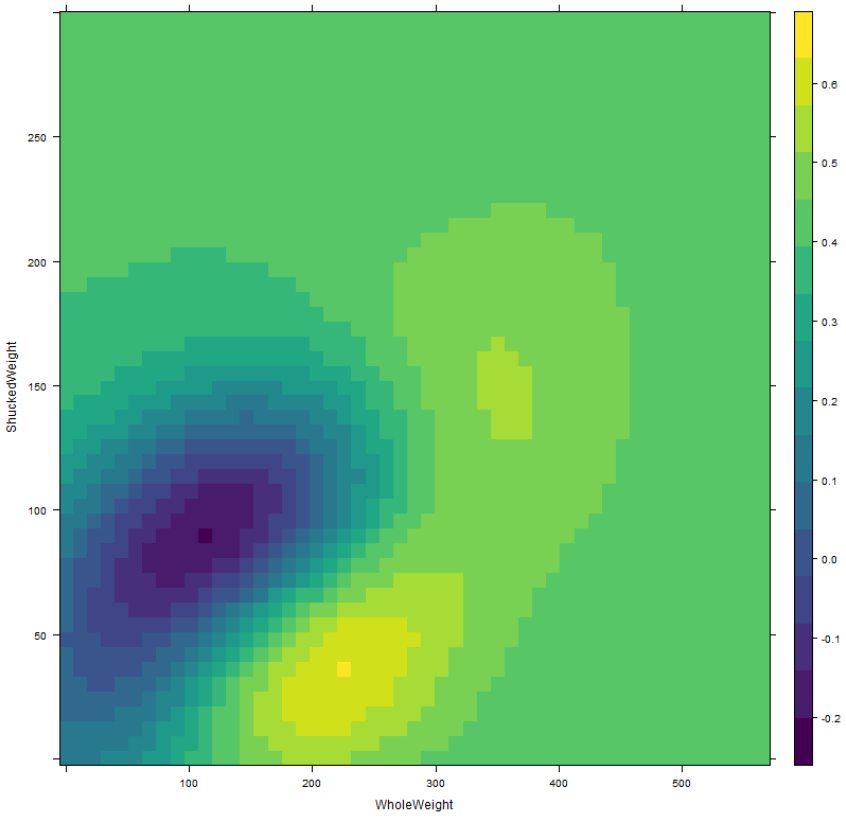


Figure 10: Two-dimensional PDP of WholeWeight *and* ShuckedWeight *for* AgeCat

Source: Own computation.
Note: Plot shows the partial dependences of WholeWeight *and* ShuckedWeight *calculated from a support vector machine model with the dependent binary variable* AgeCat*. The x-axis shows the scale of* WholeWeight*, whereas the y-axis shows the scale of* ShuckedWeight*. The color scale displays the average value of* $\hat{f}$*, in this case the probability of being old or young. A brighter tone indicates a higher prediction, a darker tone shows a lower prediction.*

simple chart. The theoretical background behind PDPs is accessible to most observers with a basic understanding of statistics. Moreover, Molnar (2020) emphasizes the causal aspect when interpreting PDPs. He argues that because a certain feature value displays the average prediction when all data points are forced to take on that feature value, the outcome-feature relationship is explicitly modelled. Therefore, Molnar (2020) thinks of this relationship in the model as causal, which is debatable. Moreover, PDPs are relatively easy to implement, as there is a variety of implemented solutions in the most commonly used programming languages.

A significant disadvantage of PDPs is the need to reduce the feature space for explanation. Interpreting a PDP with one feature is straightforward, understanding the behaviour of two features is more complex, whereas interpreting three features at the same time can only convey understanding about the rough nature of the effects in most cases. Arguably, the biggest disadvantage of PDPs, however, is the assumption of independence between features. Unfortunately, most features in a real world data set exhibit some form of complex correlation. For instance, it is natural to assume that *Length* and *WholeWeight* of an abalone are correlated, because longer

abalones are heavier and vice-versa. If we calculate the PDP for the feature *WholeWeight*, then we aggregate over the feature *Length*. During this, we average, for instance, for a high value of *WholeWeight* over the entire marginal distribution of *Length*. Calculating a *WholeWeight* value for 400 grams for an abalone which is only 10 mm long is, however, physically nonsensical. That is, when features are correlated, certain areas of the feature distribution are overweighted when the probability mass at this location is actually very little (Molnar, 2020). This should be especially problematic in the abalone data set, where correlations between features are generally high as can be seen in the correlation matrix in table 4. An additional disadvantage of PDPs is their computational complexity. Performing the computations for PDPs in a large data set should always be parallelized, because the computational complexity of PDPs rises exponentially. Moreover, since PDPs are summaries of features, a lot of data information is lost during the averaging process. Heterogeneous effects for the feature under investigation disappear in the aggregation process. A remedy for this is to use individual conditional expectation curves, which we will explain in the next sub-chapter.

6.2 Individual Conditional Expectation

6.2.1 Individual Conditional Expectation Plots of Goldstein et al. (2015)

PDPs are used to visualize the dependence of a ML model's prediction on one or more features as a summary of multiple individual effects. During the aggregation of these individual effects a considerable amount of information stored in the data is lost. Therefore, Goldstein et al. (2015) propose individual conditional expectation curves which visualize N-estimated curves, where each curve reflects the individual dependence of the predicted response on a feature x_s, conditional on the other features x_c. That is, instead of averaging the partial effects of x_s on the predicted outcome as in Friedman (2001), Goldstein et al. (2015) estimate and visualize the N-individual conditional expectations curves, where each curve represents the predicted outcome of an observation as a function of x_s conditional on the complementary vector x_c. The following equation describes one individual conditional expectation $f_{ice}(x_s)$:

$$f_{ice_i}(x_s) = E[\hat{f}(x_s, x_{c_i})]. \tag{14}$$

Plotting each of the N-different f_{ice_i} curves, where x_c values are held constant, against x_s results in a graphic also referred to as individual conditional expectation curves plot or short ICE plot. Each curve represents one observation and visualizes the effect of varying x_s of a particular observation on the output prediction, given all other features x_c remain constant. At each x-coordinate x_{s_i} stays fixed, while values of x_c are varied across all observations. A single ICE curve, thus, describes the conditional relationship of x_s and the estimated prediction at fixed values of x_s. Note that averaging the predicted effect across a given value x_{s_i} for all $i = 1, ..., N$ results in the PDP value of feature s at x_{s_i}. That is, Goldstein et al. (2015) propose to visualize the disaggregated individual conditional expectation effects which Friedman (2001) average in the calculation of PDPs.[71] ICE plots are preferable over PDPs in that they are extremely suited for highlighting variations in the fitted values across the range of features. Thereby, they facilitate the identification of heterogeneities and or trends in the predicted outcome.

In some application cases the ICE curves have a large range of predicted values or appear to be stacked on top of each other, because the data set is quite large. From an observational point of view it can, thus, be hard to identify the curvature of the ICEs and to discover certain effects or trends thereof. Goldstein et al. (2015) propose a way to investigate ICEs in cases of a large range of predicted values and heterogeneous behaviours of ICEs: the centered ICE (c-ICE) plot. The ICE curves are centered at a specific point of the feature x_s and, thus, display the difference in the predicted outcome to this point. That is, first one chooses an anchor point x^* in the range of x_s and joins all prediction lines at that point.[72] For each ICE curve $\widehat{f_{ice_i}}$ one calculates the c-ICE $\widehat{f_{c-ice_i}}$ via:

[71] The second sub-chapter in appendix B explains the relationship between PDPs and ICEs in more detail.

[72] Goldstein et al. (2015) found the minimum or maximum value of x_s to be an optimal location for x^*. Anchoring all ICEs at the minimum, for instance, ensures that all curves start at 0 removing the different intercept levels caused by the different values of the x_{c_i}'s.

$$\widehat{f_{c-ice_i}} = \widehat{f_{ice_i}} - \mathbf{1}\widehat{f_{ice_i}}(x^*, x_{c_i}), \tag{15}$$

where $\hat{f}$ is again the fitted model and $\mathbf{1}$ is a vector of 1's of the appropriate dimension.[73] The point $(x^*, \hat{f}(x^*, x_{c_i}))$ serves as a base line for all other curves. By subtracting $\hat{f}(x^*, x_{c_i})$, we perform a monotonous transformation which does not alter the slopes of the ICE curves. Thus, c-ICE plots facilitate the identification of the ICE curves' slope differences.

Goldstein et al. (2015) moreover present a third variation of ICEs, the derivative ICE or short d-ICE. The derivative ICE is useful in identifying interaction effects. The idea behind taking the derivative is that if a feature x_s does not interact with other features x_c, then the prediction function $\hat{f}(x)$ can be written as a sum of two different effects:

$$\hat{f}(x) = \hat{f}(x_s, x_c) = g(x_s) + h(x_c), \quad s.t. \quad \frac{\delta \hat{f}(x)}{\delta x_s} = g'(x_s). \tag{16}$$

If there is no interaction between features, the derivative $g'(x_s)$ should be the same for all observations. Thus, all curves in the d-ICE plot are similar such that the plot should resemble a flat line. If there are interactions between the parameters, however, the d-ICEs will exhibit non-horizontal behaviour. It can also be helpful sometimes to show the standard deviations of $\frac{\delta \hat{f}(x)}{\delta x_s}$ in order to identify conspicuous behaviour indicating interaction effects (Molnar, 2020).

6.2.2 Example Interpretation

Figure 11 shows four ICE plots for our previous random forest model of chapter 6. Each thin black line shows the ICE for a single observation, whereas the red line shows the PDP from figure 6. As can be seen, the scale of the ICE plots has a wider range, since it has to account for multiple ICEs with different intercepts. Generally speaking, if ICE plots show parallel curves then it is likely that there are no interaction effects in the data. However, if there are multiple distinct trends in the ICE curves, it is very plausible that there are different effect groups or there is an interaction between features. For *WholeWeight* the effects seem to be relatively homogeneous. The continuous incline is shared by almost all ICEs. There is some volatile behaviour for values above 100 grams in the area of higher predictions, yet, they still share the same upward trend with most of the ICEs. The picture for *ShuckedWeight* is quite different. While most of the ICEs follow the average steep downward pattern of the PDP, small values of *ShuckedWeight* have an initial minor contrary positive trend, after which the effect runs horizontally. For some observations there is apparently a very distinct effect of *ShuckedWeight*. The effect for *Length* is once again inconclusive and may be a good example for variables which apparently have no effect on the prediction. Most lines match the horizontal direction of the PDP, although there are some outlier movements for values of *Length* above 90 mm for predictions over the PDP. Judging from the vertical movements of the *Length* ICE curves, there seems to be little evidence of any effect. Although it may be hard to identify at first because of the overlaying lines, the

[73] Sub-chapter three in appendix B explains the calculation process of c-ICEs in more detail.

effect for *Sex* is in accordance with previous results. Infants are predicted as younger, female and male abalones as older. Overall, it seems as if the ICE curves give some indication that there are interactions between features which the random forest model does not account for, in particular for *ShuckedWeight*. The PDPs seem to represent the individual effects for *WholeWeight* and *Sex* quite well. This is in accordance with the fact that tree-based ML algorithms generally model feature interactions rather well (Kuhn and Johnson, 2013).

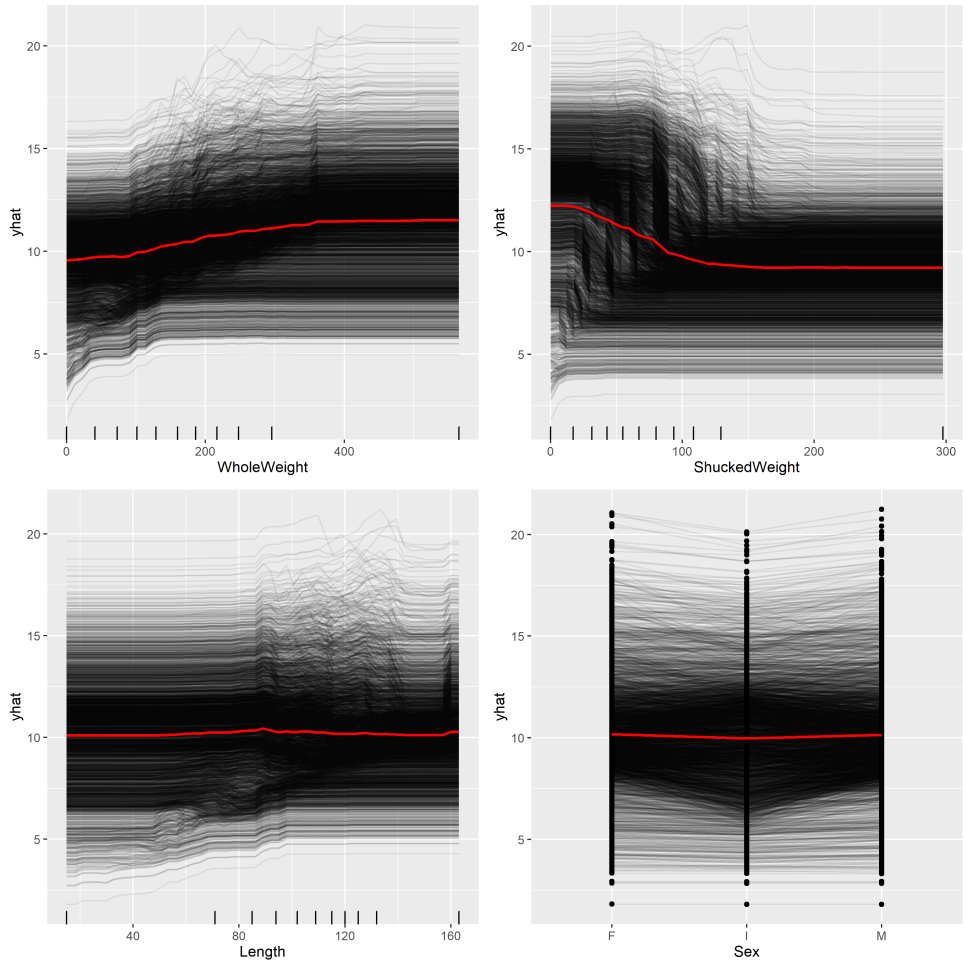


Figure 11: ICE plots of WholeWeight, ShuckedWeight, Length, *and* Sex *for* Rings

Source: Own computation.
Note: From left to right each plot shows the individual conditional expectation curves of the variables WholeWeight, ShuckedWeight, Length, *and* Sex *calculated from a random forest model with the dependent variable* Rings. *Variable* Sex *is categorical with the values female (F), infant (I), and male (M). Each x-axis shows the scale of the respective variable, while the y-axis displays the average value of* $\hat{f}$ *which is numerical. The lines at the x-axis indicate a rug, where two lines represent the feature's decile.*

Results for the ICE plots of the SVM classification model in figure 12 paint a different picture. At first glance, there seem to be two different effect patterns. First, the majority of ICEs follows the S-shaped red partial dependence in direction. Low values of *WholeWeight* lead to a prediction of "young". Increasing the weight, first leads to a "less old" prediction until the majority of

ICEs predicts an abalone to be old at around 140 grams. The second strand of ICEs predicts abalones to be old for almost all values of *WholeWeight* and follows a similar S-shaped effect. This effect increases with *WholeWeight* until it hits a peak for the majority of ICE curves at shortly before 300 grams. Moreover, most notably, there is a moderate fraction of ICEs which has an extremely high and positive prediction for medium values of *WholeWeight* (150 to 200 grams). Clearly, there must be some sort of interaction between variables which the SVM does not capture. Similarly, there are multiple effect trends for *ShuckedWeight*. Even though the PDP helps in identifying the general direction of the effect, it is still hard to discern major feature trends.[74] Graphical overload is a common disadvantage of ICE curves. The only main observation which we can draw from the chart of *Length* is that there are significant interactions which the SVM model did not capture. Everything else is only based on superficial speculations.[75] The potentially largest trend in the effect of *ShuckedWeight* suggests that abalones with moderate meat weight are predicted as young, as reinforced by the PDP. This effect is similar in shape for most ICEs but starts at different intercepts respectively. That is, most observations do have a U-shaped effect. Yet, for the same level of *ShuckedWeight* this effect starts out as positive for some instances and negative for others. In summary, for some ICE plots it is very hard to distinguish distinct effect trends.

[74] Usually in such cases, it is recommended to raise the transparency of the ICE-curves to make largely overlapping ICE curves more visible. We performed this analysis for the ICE curves in figure 11. Yet, it was still as difficult to discern main effects because of the ICE curves' overlap.

[75] There seem to be several different trends for the effect. The first one starts out with a positive prediction for low values of *ShuckedWeight*. The prediction of this effect marginally increases for values of *ShuckedWeight* until 50 grams, then steeply declines until it hits a minimum at around 100 grams, after which it rises again and converges to a slightly positive probability prediction for large values of *ShuckedWeight*. The other main effect starts out with a negative prediction which steeply declines until it hits a minimum, after which it increases again and converges to a slightly positive probability again.

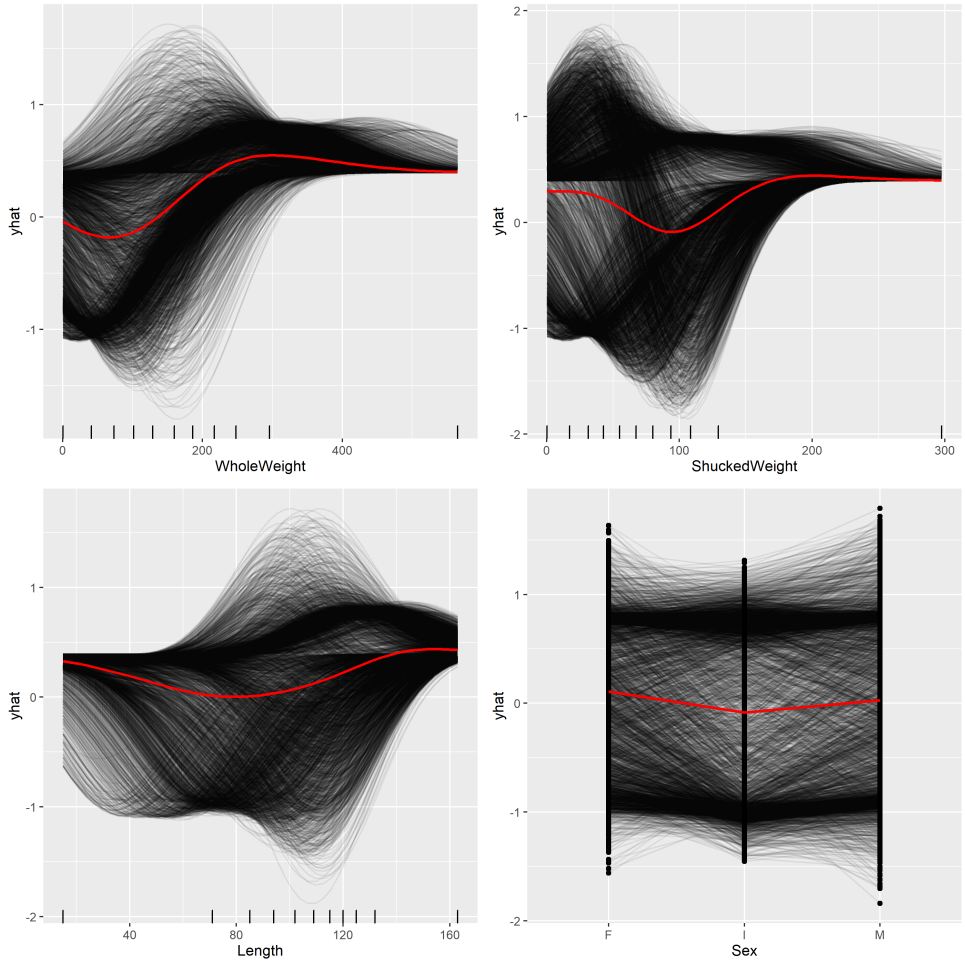


Figure 12: ICE plots of WholeWeight*,* ShuckedWeight*,* Length*, and* Sex *for* AgeCat

Source: Own computation.
Note: From left to right each plot shows the individual conditional expectation curves of the variables WholeWeight*,* ShuckedWeight*,* Length*, and* Sex *calculated from a support vector machine model with the dependent binary variable* AgeCat*. Variable* Sex *is categorical with the values female (F), infant (I), and male (M). Each x-axis shows the scale of the respective variable, while the y-axis displays the average value of* $\hat{f}$ *which represents a probability. The lines at the x-axis indicate a rug, where two lines represent the feature's decile.*

From an observational perspective it is hard to assess patterns in the *Length* plot, since the majority of ICEs cross or lie on top of each other. No clear patterns can be inferred which is very similar to previous findings. For *Sex*, there also seem to be two different types of effects. ICE curves of categorical features are generally harder to interpret, because there is little space to see the movements of the ICE curves. In general, we find that the SVM model did not capture certain interactions in the features. This is in line with the literature suggesting that SVM models often have problems with modelling certain interaction structures between features (Kuhn and Johnson, 2013).

Since the ICE plot for *ShuckedWeight* in figure 11 was not very informative, figure 13 showcases a centered ICE plot for the SVM model predicting *AgeCat*. This makes it easier to compare the

curves of individual observations. In this new c-ICE plot two different effects are more clearly visible. While there is still a fraction of the c-ICEs which predict an abalone to be old for low and medium values of *ShuckedWeight*, the majority of individual instances follows the directions implied by the S-shaped PDP. Yet, there is still a considerable amount of heterogeneity in the ICE plots reinforcing our concerns that there might be interactions between the features.

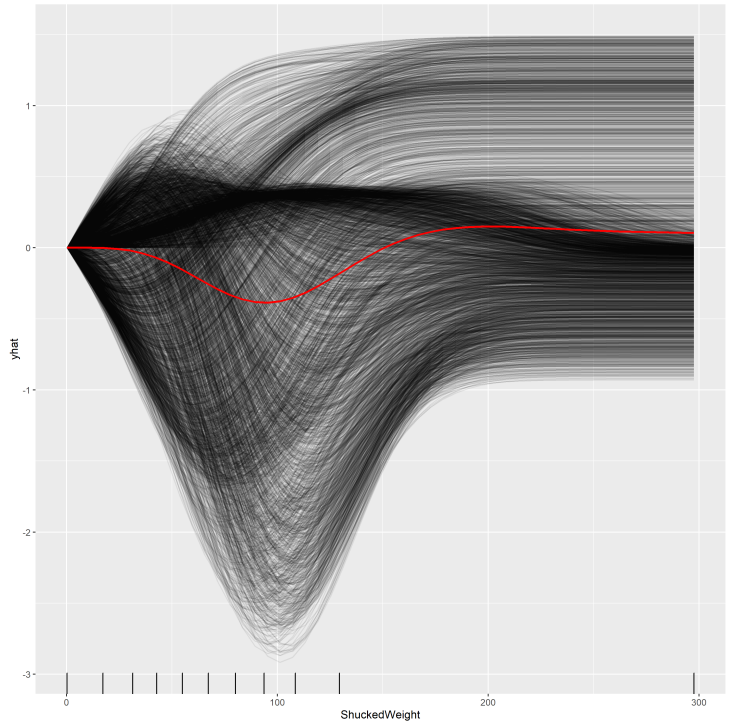


Figure 13: c-ICE plot of ShuckedWeight *for* AgeCat

Source: Own computation.
Note: Each black line represents the centralized individual conditional expectation of the variable ShuckedWeight *anchored at the minimum value of* ShuckedWeight*. The c-ICEs were calculated from a support vector machine model with the dependent binary variable* AgeCat*. The red line marks the PDP from figure 7, top-left chart. The x-axis shows the scale of* ShuckedWeight*, while the y-axis displays the average value of $\hat{f}$ which represents a probability. The lines at the x-axis indicate a rug, where two lines represent the feature's decile.*

Eventually, we showcase a derivative ICE plot in figure 14, calculated for the ICE of *WholeWeight* on *Rings* from figure 11, top-left chart. As you might recall, we found little evidence of interactions for the *WholeWeight* effect. The d-ICE plot shows a similar picture. The estimated derivative of the predictions only shows mild variations, suggesting that $\frac{\delta \hat{f}(x)}{\delta x_{WholeWeight}}$ is marginally different from 0 for values between 0 to 300 grams of *WholeWeight*. Interestingly enough, this is also where we observe the most variation in the ICE curves in figure 11. Moreover, the standard deviation has some notable spikes which imply that there could be some minor feature interactions.

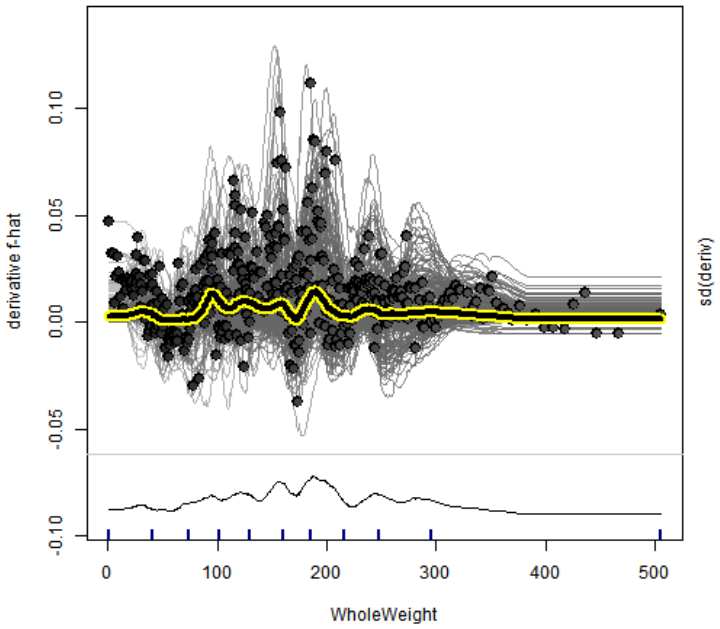


Figure 14: d-ICE plot of WholeWeight *for* Rings

Source: Own computation.
Note: Each black line represents the derivative individual conditional expectation plot of the variable WholeWeight *anchored at the minimum value of* WholeWeight*. The d-ICEs were calculated from a random forest model with the dependent variable* Rings*. The yellow line marks the standard deviation of the d-ICEs. The x-axis shows the scale of* WholeWeight*, while the y-axis displays the average value of* $\hat{f}$ *which is numerical. The bottom half of the panel shows the empirical distribution of* WholeWeight*. The lines at the x-axis indicate a rug, where two lines represent the features decile.*

In summary, interpreting ICEs is done by identifying larger trends of ICEs. Effect trends are identified by finding common or discerning patterns in ICE shapes for different values of x_s. That is, if most ICE curves are parallel, there is little evidence of interaction. If there are different superimposed effect patterns, it is very likely that interactions between the features exist. c-ICE plots can help better visualize overlaying ICE patterns by forcing all intercepts to start at a specific value. If an observer suspects that there is a strong interaction effect in her ML model, she can additionally check this via d-ICE plots, which showcase variations in the individual derivatives of $\hat{f}$ w.r.t. x_s. Goldstein et al. (2015) propose moreover, to use colored plots, where a second feature effect may be visualized as differently colored shades in the ICE line. However, such plots become easily convoluted very quick, and are not as popular as one-dimensional ICEs which is why we do not show them in our analysis.

6.2.3 Discussion of Advantages and Disadvantages

ICEs, c-ICEs and d-ICEs are a straightforward method to summarize a feature's behaviour in a local in particular but also a global manner at the same time. Local trends can be identified by looking at the ICE curves, while global model behaviour can be identified by looking at

ICE trends. Whereas PDPs only provide an aggregate overview of a ML model's internal workings, ICE plots can show feature interactions and differences between individual instances. One significant disadvantage is that ICEs are limited in their visualization capability. Only a single feature with a limited number of observations can be visualized in a meaningful way at once. As demonstrated above, even smaller ICE plots with observations in the thousands' range reach their visual display capability quickly. In some cases it can be helpful to partition the sample for ICE plots to remove plot clutter. Moreover, - equivalent to PDP plots - ICE curves show improbable data points when features are correlated (Molnar, 2020). ICE plots are best interpreted when combined with PDP plots in order to compare individual ICE curves against the average trend. ICE plots are faster to compute than PDPs, since no averaging is needed. Most common programming languages offer different implementations of ICE curves. From a computational perspective it is furthermore interesting to note that d-ICE plots and two-dimensional ICE plots are rather time- and resource-consuming which makes them difficult to use for larger data sets.

6.3 Accumulated Local Effects

6.3.1 Apley's (2016) Accumulated Local Effects Plots

Accumulated local effect plots after Apley (2016) are an alternative to PDPs and ICEs. Equivalently to the latter two techniques, ALEs estimate an effect $f_{s,ALE}(x_s)$ of a feature x_s from a supervised black box model $f(\cdot)$. ALE plots are superior to PDPs in that they are able to account for correlations between features and they are substantially less computationally-expensive. The basic rationale of ALEs is to integrate the partial derivative of the response function after x_s back to x_s in order to obtain an accumulated partial effect of x_s for the target variable. That is, first, the expectation is built over the changes of the predictions $\frac{\delta f(x_1,...,x_d)}{\delta x_j}$ in order to avoid biasing the estimator through not respecting the correlation of the features. Theoretically, the changes of predictions are defined as the gradient. For the empirical computation of the ALEs, the gradient is substituted by calculating the average of the finite differences in predictions for discretised pre-defined intervals of the predictor space x_s. Consequently, we integrate over variable z_j in the entire range of feature x_s in order to accumulate the local effects. For the empirical estimation the outer integral is replaced by the corresponding summation over the same discretised pre-defined intervals used for calculating the finite differences.

Goldstein et al. (2015) propose different types of ALEs. ALEs can be distinguished by their order of effect, i.e. a feature effect can be decomposed into a main and higher-order effects. The most common observed effects are main and second-order effects.[76] Moreover, ALEs can be centered or uncentered. For a centered ALE, a constant *const.* is subtracted to set the average ALE w.r.t. x_j to 0 in order to facilitate interpretation. The following equation shows a centered ALE for a main-order effect of the predictor x_s, where $s \in \{1, ..., p\}$:

$$f_{s,ALE}(x_s) \quad = \quad \int_{z_{0,s}}^{x_s} E[\frac{\delta f(x_1, ..., x_p)}{\delta x_s} \mid x_s = z_s] \, \mathrm{d}z_s - const. \tag{17}$$

The function $f(\cdot)$ can represent any supervised ML model. Once again $x_s \subset \boldsymbol{x}$ is the feature which is to be interpreted and x_c represents the complementary features such that $x_c \cup x_s = \boldsymbol{x}$. The approximate lower bound of x_s is denoted as $z_{0,s}$. The theoretical ALE effect in equation 17 makes the assumption that $f(\cdot)$ is differentiable.[77] Since some ML models, such as for example random forests, cannot be differentiated, Goldstein et al. (2015) propose an estimator of the ALE which approximates the differentiation and integration by taking finite differences and summations. Particularly, finite differences are calculated to approximate the gradients and replace the differentiation, while the integral is substituted by a simple summation over $k_s(x)$

[76] The order refers to the degree of the derivative calculated in the ALE. Technically, ALEs can also be calculated for third and higher-order effects, but since these are difficult to interpret Apley (2016) argue that ALEs are primarily intended for main and second-order effects. A second-order effect in the case of the abalone data set would be calculating the first derivative of the estimated predictions w.r.t. *WholeWeight* and consequently taking the derivative thereof w.r.t. *ShuckedWeight*.

[77] Apley (2016) also provides a slightly changed definition of ALEs for non-differentiable $f(\cdot)$ as well, see their remark 2.

intervals of feature x_s. The estimated main-effect for a centered ALE of a numeric feature is defined as:

$$\widehat{f_{s,ALE}(x)} = \sum_{k=1}^{k_s(x)} \frac{1}{n_s(k)} \sum_{i:x_{i,s} \in N_s(k)} [f(z_{k,s}, x_{i,c}) - f(z_{k-1,s}, x_{i,c})] - \widehat{const}. \tag{18}$$

The index $k_s(x)$ describes the interval into which x falls, that is, $x \in (z_{k_s(x)-1,s}, z_{k_s(x),s}]$. The set $\{N_s(k) = (z_{k-1,s}, z_{k,s}) : k = 1, 2, ..., K\}$ denotes a sufficiently fine partition of the sample range of $\{x_{i,s} : i = 1, 2, ..., n\}$ into K intervals.[78] From $k = 1, ..., K$ the scalar $n_s(k)$ designates the numbers of observations in $\{x_{i,s} : i = 1, ..., n\}$ which fall into the kth interval, such that $\sum_{k=1}^{K} n_s(k) = N$, which refers again to the sample length. The constant $\widehat{const}$ is chosen such that $\frac{1}{N} \sum_{i=1}^{N} \hat{f}_{s,ALE}(x_{i,s}) = 0$. This estimator in equation 18 can be used for differentiable and non-differentiable ML models.

6.3.2 Example Interpretation

ALEs are best visualized for interpretation. Figure 15 shows four centered ALE plots for the same four variables from the previous random forest model. We calculate the ALE with $K = 20$ which should be sufficient, since the number of abalone observations is relatively low with $4,177$ cases and we want to identify smaller changes in the ALE. The rug under each plot visualizes the feature's entire distribution in order to facilitate attributing the K-distinct regions to the sample's feature distribution. We choose to show the entire sample distribution per observation instead of per decile, since it is important when interpreting ALEs to identify more granular parts of the distribution, because of the interval calculations. The top-left chart shows the ALE plot of *WholeWeight*. Note that in centered ALE plots we interpret the relative size of the effect, whereas with PDPs and ICEs we usually interpret the actual estimated effect size. That is, the predicted value at a certain point x_s can be interpreted as the difference to the mean prediction. Colloquially, it is preferable to say "the effect turns negative", when the centered ALE shows an effect below the mean prediction, even though the predicted outcome is not per se negative but just less positive. We will follow this rationale in the coming discussions of the ALE plot. Increasing *WholeWeight* has a positive effect on the predicted number of *Rings*. This effect looks fairly straight and with much imagination almost linear to a certain degree. The effect gets marginally weaker, the larger the value of *WholeWeight* is for values over 300 grams. However, the rug under the chart shows that there are hardly any observations for high values of *WholeWeight*.

[78] K is only used when the predictor is numeric. If a predictor is categorical, then K is equivalent to the number of categories. Goldstein et al. (2015) generally recommend a value of $K = 100$. In case a more granular ALE plot is desired, they recommend using higher values of K as this gives more precise results. In small data sets, smaller values of K may also be appropriate.

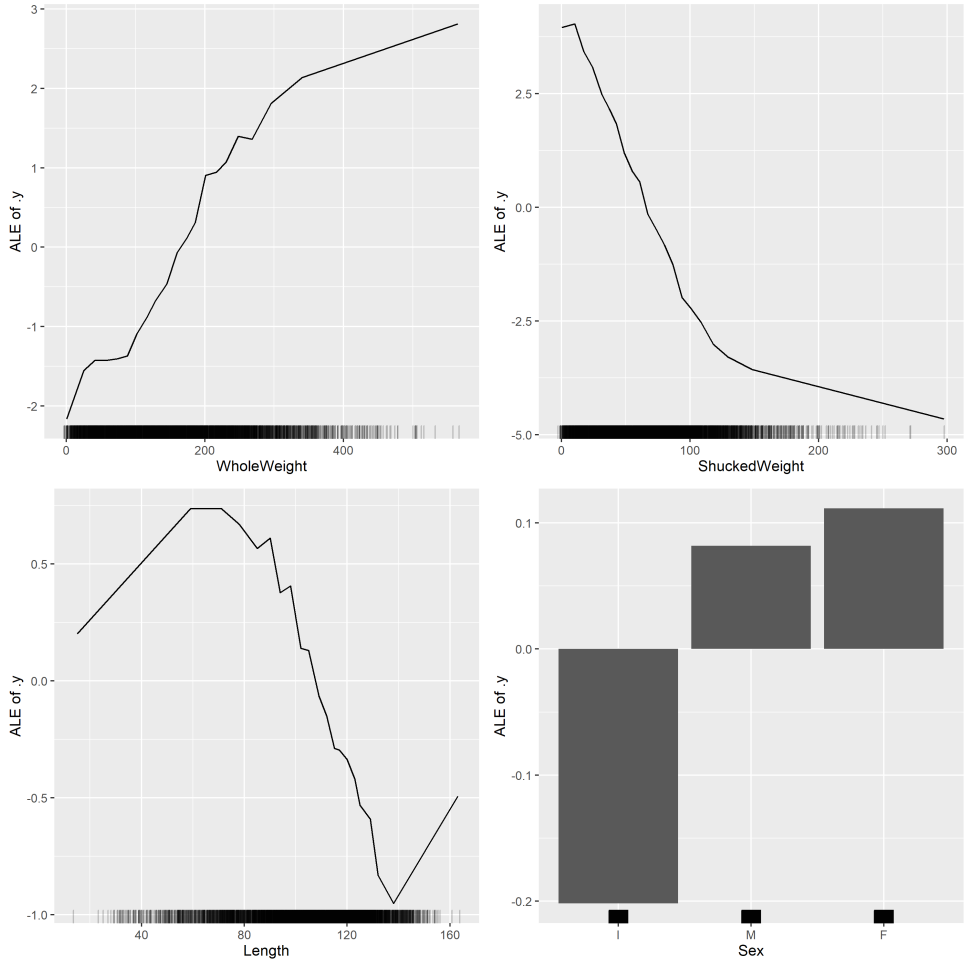


Figure 15: Centered ALE plots of WholeWeight, ShuckedWeight, Length, *and* Sex *for* Rings

Source: Own computation.
Note: From left to right each plot shows the centered accumulated local (main) effect with $K = 20$ *(when applicable) of the variables* WholeWeight, ShuckedWeight, Length, *and* Sex *calculated from a random forest model with the dependent variable* Rings. *Variable* Sex *is categorical with the values infant (I), male (M), and female (F). Each x-axis shows the scale of the respective variable, while the y-axis displays the average value of* $\hat{f}$ *which is numerical. The lines at the x-axis indicate a rug, where a line represents a single observation.*

Comparing the ALE plot of *WholeWeight* to the PDP thereof in figure 6 shows that there is almost no difference in the shape between the estimated effects. The ALE substantiates our presumption that *WholeWeight* has a positive effect on the predicted number of *Rings* and *WholeWeight* exhibits little interaction with other predictors. The ALE chart of *ShuckedWeight* looks similar to the PDP thereof as well. The ALE for *ShuckedWeight* is L-shaped with a kink at around 110 grams *ShuckedWeight*. Large values of *ShuckedWeight* have a strong negative effect on the prediction. Previously demonstrated interpretability techniques showcasing a decreasing predicted number of *Rings* for an increasing number of *ShuckedWeight* share this notion. The bottom-left chart shows the effect of *Length* which seems considerably different from its PD counter-plot. The estimated ALE of *Length* appears inverse-S shaped. For small values of *Length*

the effect is positive and rises, then significantly decreases until it turns negative, after which it becomes slightly "less negative" again. Looking at the scale of the effect shows, however, that *Length* exhibits a smaller effect compared to other features - even when accounting for the predictor's scale. Both, the PDP and ALE of *Length*, unites the shape of an initial incline after which the effect considerably drops. Otherwise, their shapes appear dissimilar. Since the ALE avoids being affected by between-feature correlations, the ALE for *Length* seems "more correct" in that it reflects the correct estimate of the feature's effect. It appears as if the variable *Length* is plagued by feature interactions, which from a biological perspective seems obvious.[79] More testing, such as second-order ALEs would be necessary to work out the true effect of *Length*. The ALE for *Sex* shows three individual effects for the respective category. Being an infant has a strong and negative effect on the mean predicted number of *Rings*, whereas being male or female has a slightly positive effect. Both are, however, relatively similar in size. This matches our previous findings w.r.t. the feature *Sex*.

Figure 16 shows a second-order ALE for *WholeWeight* and *ShuckedWeight*. For low values of *WholeWeight* and high values of *ShuckedWeight* the model predicts a strong positive effect for the predicted number of *Rings*. This seems like quite a strong effect with a centered ALE of around 5. This finding sharply contrasts the nature of the effect in PDP in figure 16 which predicts exactly the opposite behaviour. Therefore, it seems probable that there is some sort of interaction between *WholeWeight* and *ShuckedWeight*, which appears also necessarily true from a biological perspective. Since there are almost no data points which exhibit low *WholeWeight* and high *ShuckedWeight*, this interaction should have a negligible effect on the final predictions, however. Interestingly enough, the second-order ALE looks otherwise relatively even. The only exception occurs for large values of *WholeWeight* and low values of *ShuckedWeight*, when the second-order ALE turns negative. But as indicated by the rug below, there are few observations for *WholeWeight* in this interval. Please note that the second-order plot only describes the second-order effect of *WholeWeight* w.r.t. *ShuckedWeight*. Formally, we first take the derivative of f w.r.t. $x_{wholeweight}$ and then take the second derivative w.r.t. $x_{shuckedweight}$. It may be tempting to interpret the chart in 16 (similar to the two-dimensional PDPs) as a main effect, but it should not. The second-order effect is only interesting for exploring interactions between features and nothing more. It can only be interpreted together with the main effect plots of *WholeWeight* and *ShuckedWeight*.

[79] For instance, *Length* and *Diameter* have to share a correlation, since the *Length* directly influences the size of the *Diameter*.

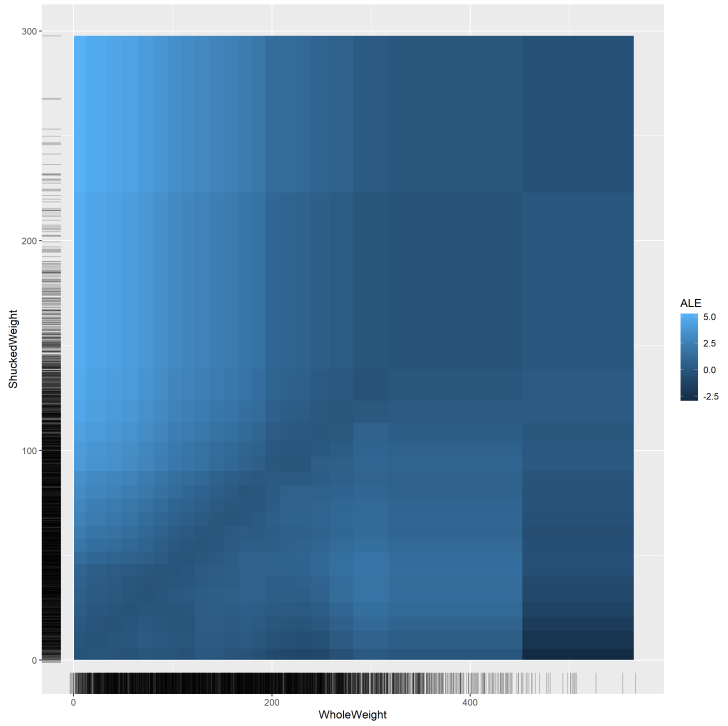


Figure 16: Second-order ALE plot of WholeWeight *for* Rings

Source: Own computation.
Note: Plot shows the second-order accumulated local effect of WholeWeight *and* ShuckedWeight *calculated from a random forest model with the dependent variable* Rings*. The x-axis shows the scale of* WholeWeight*, whereas the y-axis shows the scale of* ShuckedWeight*. The rug at each axis shows the feature's empirical distribution, where a line represents a single observation. The color scale displays the strength of the estimated ALE centered at* 0*. A brighter tone indicates a higher ALE, a darker tone shows a lower ALE.*

We also perform a centered ALE calculation for the SVM model calculated on the binary variable *AgeCat*. Our results for the variables *WholeWeight*, *ShuckedWeight*, *Length*, and *Sex* for *AgeCat* are shown in figure 17. In comparison to PDPs and ICEs, ALEs for classification show two different plots: Each plot shows the effect for falling into a specific category of the predicted variable y. Since we have a binary classification task, the effects are mirrored on the horizontal 0-axis and do not reveal any particular insights in the binary case. Consider, for instance, the top left chart. The left part shows the effect of increasing *WholeWeight* on the estimated probability of falling into class 0, whereas the right chart shows the probability of falling into class 1. Therefore, interpretation of a single chart in the binary case is generally sufficient. In cases where more than two classes are predicted, more charts have to be interpreted in order to get a comprehensive overview of the feature's effect. Let us look at the effect of *WholeWeight* and falling into category 0 now. For values below 100 grams the effect is positive and constant, after which it steeply declines, turns negative at around 175 grams and plateaus at around 220 grams. From thereon the effect in the prediction is constantly around -0.18. That means that even for higher values of *WholeWeight* the prediction remains the same. The PDP estimated for *WholeWeight* is comparable in form but larger in absolute value compared to the

centered ALE. For instance, the PDP estimates an absolute range of around 0.75[80], whereas the centered ALE has a range of around 0.35[81]. Even though the centered ALE shows a relative effect, we can still compare the ranges, since the centered ALE only experiences a shift in its estimated prediction on the y-axis by subtracting a constant but the range remains unchanged. The difference in range probably results from the fact that the ALE specifically tries to minimize the influence of other correlated factors by averaging the local effect. For *ShuckedWeight* we observe a steeply rising positive effect which also seems to hit a plateau at a probability around 0.39. The range of *ShuckedWeight*'s effect is larger in absolute value for the ALE than for the PDP. The effect of *Length* is once again ambiguous in its shape. The estimated probability of being older than 9 *Rings* decreases with *Length* until around 60 mm then hovers more or less around zero. Apparently, the feature *Length* has a positive effect for values of *Length* below 60 mm after which there seems to be almost no effect. The effect for *Sex* has similar directions as the PDP but is way smaller in range than the effects estimated by the PDP.

[80]The PDP in figure 9 has a minimum at around -0.2 and a maximum at around 0.55.
[81]The ALE in figure 17 has a minimum at around -0.2 and a maximum at around 0.15.

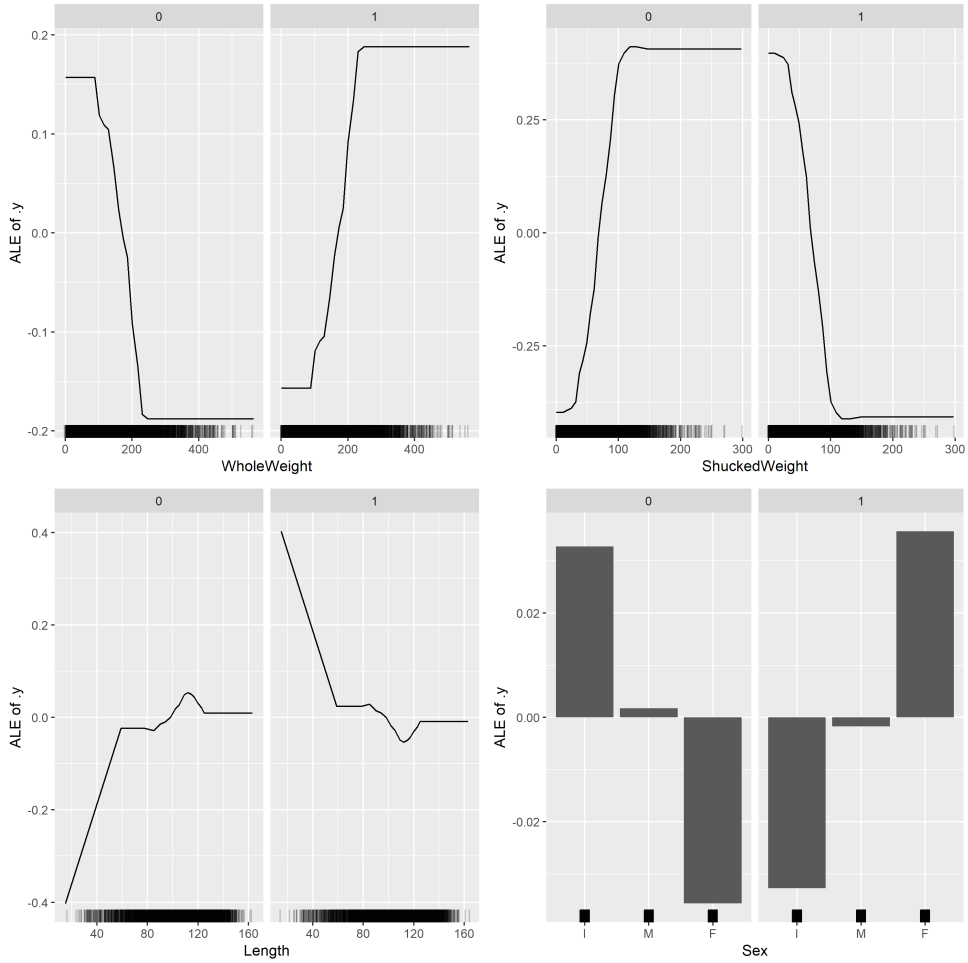


Figure 17: Centered ALE plots of WholeWeight, ShuckedWeight, Length, *and* Sex *for* AgeCat

Source: Own computation.
Note: From left to right each plot shows the centered accumulated local (main) effect with $K = 20$ *(when applicable) of the variables* WholeWeight, ShuckedWeight, Length, *and* Sex *calculated from a support vector machine model with the dependent binary variable* AgeCat. *Each individual chart shows the accumulated local effect for the probability of being young on the left and for being old on the right. Variable* Sex *is categorical with the values infant (I), male (M), and female (F). Each x-axis shows the scale of the respective variable, while the y-axis displays the average value of* $\hat{f}$ *which represents a probability. The lines at the x-axis indicate a rug, where a line represents a single observation.*

To complete the picture for ALEs, we also include a second-order centered ALE plot for a classification model. This plot can be found in figure 18. As with main-effect ALEs for classification, the plot shows two individual charts, one for the estimated effect of *WholeWeight* and *ShuckedWeight* falling into category 0, one for category 1. The plot is more heterogeneous than its random forest regression counterplot. A potential reason for this could be that we estimate a SVM model which partitions the predictor regions into smoother subregions than a random forest model. A notable observation here is the small dark blue region at the bottom right of the right chart, which indicates that there might be an interaction effect for medium to large values of *WholeWeight* and small values of *ShuckedWeight*. Yet, concerns about large prediction interference of this interaction are small, since there are hardly any data points which

fall into the category of low *ShuckedWeight* values and large *WholeWeight* values.

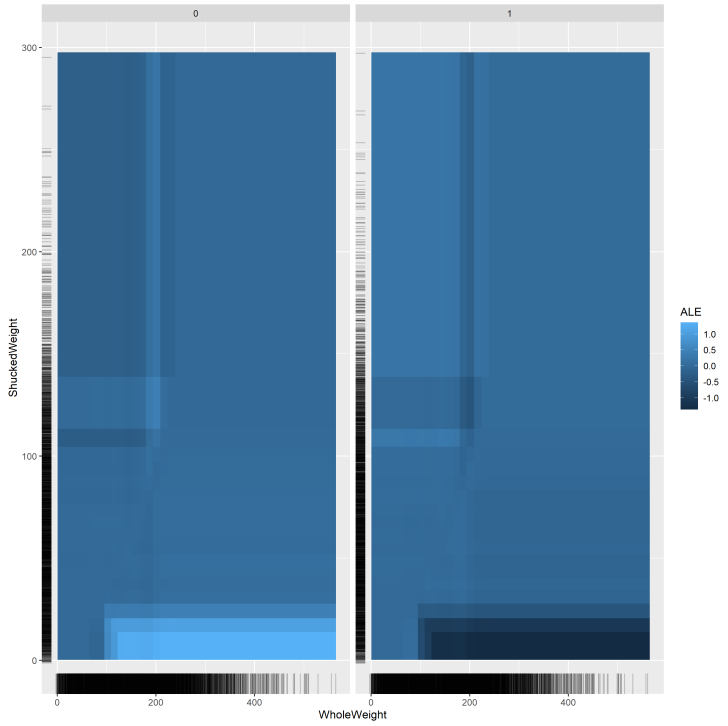


Figure 18: Second-order ALE plot of WholeWeight *for* AgeCat

Source: Own computation.
Note: Plot shows the second-order accumulated local effect of WholeWeight *and* ShuckedWeight *calculated from a random forest model with the dependent variable* AgeCat*. The x-axis shows the scale of* WholeWeight*, whereas the y-axis shows the scale of* ShuckedWeight*. The rug at each axis shows the feature's empirical distribution, where a line represents a single observation. The color scale displays the strength of the estimated ALE centered at* 0*. A brighter tone indicates a higher ALE, a darker tone shows a lower ALE.*

6.3.3 Discussion of Advantages and Disadvantages

ALE plots have recently become a more and more used alternative for PDPs (Molnar, 2020). Their main advantage is that an ALE is an unbiased predictor of the estimated effect (Apley, 2016). That is, even when features are correlated, the ALE shows the "true" estimated effect of the ML model, when other techniques such as PDPs and ICEs fail. Moreover, ALEs are as easy to interpret as PDPs. Therefore, using an ALE over a PDP is always a safer choice. Moreover, Apley (2016) stresses the fact that ALEs are way faster to compute than PDPs.[82] The option to calculate second- and third- order effects also separates ALEs from other techniques. Getting an

[82]The ALE for two features always calculates N-finite differences independent from the intervals K. Its worst-case run time is, thus, always $\mathcal{O}(N)$. For PDPs the number of calculations is N forecasts. That is, the maximum amount of predictions is N ICE curves. A PDP, thus, averages for N data points over N ICE curves. Each additional observation requires the calculation of one additional prediction for each N-ICE curve as well as an additional ICE curve with $N + 1$ observations. Hence, the computational complexity is $\mathcal{O}(N^2)$. If we calculate c-ICE curves, we need to subtract for each ICE curve N times. The computational complexity is then $\mathcal{O}(N^2 + N^2) = \mathcal{O}(2N^2) = \mathcal{O}(N^2)$. The same holds for d-ICE plots where we have to compute N derivatives for N predictions, s.t. $\mathcal{O}(N^2 + N^2) = \mathcal{O}(2N^2) = \mathcal{O}(N^2)$.

understanding of these higher-level effects offers a way to make more complex behaviour of a ML model more transparent even if they are more difficult to interpret. Since ALEs are implemented in most common programming languages, their use is also supported from an implementation availability perspective.

Unfortunately, ALEs can only provide interpretability on a global level. There is no local pendants to what ICEs are to PDPs for ALEs. Therefore, local interpretability has to be pursed with other techniques. Moreover, ALEs can be dependent on the choice of the interval size (Carvalho et al., 2019). The smaller the interval K is chosen, the more precise the effect of a feature is estimated, however, the more wonky the ALE plot can become. Calculating an ALE with varying sizes to see how the trajectory of the ALE changes is always recommended. Moreover, it has been reported that the second-order ALEs sometimes exhibit varying stability across the feature space because of varying accuracies of the local finite difference estimations caused by heterogeneous observation sizes per interval (Molnar, 2020).

6.4 Global Surrogate Models

6.4.1 Global Surrogate Models from the Field of Engineering

The idea behind surrogate models is to approximate a black box ML model by calculating an intrinsically interpretive ML model on the first model's training data and its predictions to mimic the model as close as possible while being more interpretable. The technique was first used in the field of engineering where physically and computationally expensive models are substituted with more affordable simulated models (Queipo et al., 2005). The difference between surrogate models in engineering and interpretable ML is that surrogate models in engineering are primarily used for simulation, whereas in ML the underlying model is a ML model and this model has to be interpretable (Molnar, 2020).[83] Other names for surrogate models include: approximation model, metamodel, response surface model, or emulator (Molnar, 2020). The following steps are required to compute a surrogate model:[84]

1. First, we train a black box ML model which we would like to interpret further. This can include any type of supervised ML algorithm.
2. Second, we select the training data of the initial black box (BB) ML model to be explained x_{BB} as well as the corresponding predictions thereof $\hat{y}_{BB}$ which we will further use to train the surrogate model. This training data can constitute a subset of the data, the whole data set, a differently weighted data set, or specific weighted instances thereof and is ultimately a question of individual preference and the application purpose.
3. Third, we choose an interpretable model, such as one which is introduced in chapter 5. Other intrinsically interpretable ML models we did not discuss in this thesis but can nevertheless be chosen as surrogate models include linear regression with regularization, logistic regression, generalized additive models, naïve Bayes, and k-nearest neighbours.[85] The observer is completely free in her choice and can select any interpretable model of her preference as long as it can fulfil the corresponding ML task, i.e. regression or classification.
4. Fourth, we compute this interpretable surrogate model and use the training data x_{BB} as the training data $x_{surrogate}$ of the surrogate model with the predictions $\hat{y}_{BB}$ as the dependent variable $\hat{y}_{surrogate}$. A global surrogate model can then be defined as:

$$f_{surrogate}(x_{BB}, \hat{y}_{BB}), \tag{19}$$

   where $f(\cdot)$ can be any of the intrinsically interpretable models with learning inputs $x = x_{BB}$ and $y = y_{BB}$.

[83] Using surrogate models in ML for interpretability was not the first time such types of models have been used in ML. Gorissen et al. (2009) propose using surrogate models to emulate a high fidelity approximation model in situations where a simpler approximation of a model is needed to perform sensitivity analyses, visualization, or design space exploration.

[84] During this description, we will loosely follow the procedure elucidated in Molnar (2020).

[85] For an introduction in how to use these methods in an interpretability context, see Molnar (2020).

5. The performance of the surrogate model can be used to asses how well the surrogate model replicates the black box model. As a fifth step we can, thus, calculate certain measures of model fit and check how well the surrogate model mimics the black box model. Naturally, this measure depends on the type of surrogate model which has been used. In the case of linear regression, for instance, we could use the R^2 or the adjusted $\overline{R}^2$. A better fit of the surrogate model manifests itself with a higher R^2. For classification, we can use the accuracy as a measure of fit. In this case, theoretically, we want the accuracy to be as large as possible. However, even if our model objective is to mimic the initial black box, we still have to be aware of overfitting in building surrogate models. That is, in general when building a ML model we do not want to overfit a model, because then the model cannot be generalized on a new set of predictions. For surrogate models the generalization on new instances should not be a problem since we should not use the surrogate model for new predictions. The problem is rather that an overfitted model which has the perfect prediction for each observation has a copious amount of coefficients. Interpreting many coefficients at once can have a negative impact on global interpretability, in particular the simulatability characteristic of Lipton (2018) as discussed in section 3.5.2. Therefore, it is important to note that the measures of fit of the surrogate model usually require a different interpretation than when using them as the initial measure of prediction performance.
6. Eventually, we can interpret the surrogate model or visualize its predictions in order to understand the behaviour of the initial ML model.

The approach of performing a global surrogate model is pretty straightforward. One basically tries to make a ML model more interpretable by using more ML (Molnar, 2020). If a chosen intrinsically interpretive model does not offer the interpretability an observer seeks, one can simply estimate another model to compare the surrogate interpretations. Global surrogate models do not require any specific information about an interpretability technique and can be used by most ML novices, which makes them a suited tool for interpretability to begin with.

6.4.2 Example Interpretation

We compute global surrogate models for both, a regression and classification task. As a first exercise, we, therefore, calculate our working horse for regression tasks, the random forest model, again and use its training data and predictions to compute a linear regression model. Table 6 shows the results for this surrogate model as well as the linear regression model from chapter 5 for comparability reasons. Please note that the comparison between the two analyses serves as a mere rough sanity check for the sign and absolute size of the surrogate model's coefficients. It is important to note that each model predicts something very different. The surrogate model predicts the predictions of the black box model and has never seen the real outcome data. The linear regression from chapter 5, on the other hand, predicts exactly this outcome, i.e. the *Rings* data of the abalone data set.

The coefficients of the surrogate model show the same sign and are relatively similar in size in

comparison to the results of the linear regression on *Rings*. There are no large surprises of the surrogate w.r.t. to the results of other models. *WholeWeight* has a positive effect, *ShuckedWeight* a negative one, and *Length* is inconclusive once again with a coefficient which is almost zero. Interpreting this surrogate model now is basically interpreting a linear regression model. Since we extensively showed how to do this in chapter 5, we refrain from doing so at this point. The only interesting observation we can do, however, is to look at how well the surrogate model approximates the predictions by looking at the R^2 and the adjusted $\overline{R}^2$, which are both relatively high with 0.745. Therefore, the surrogate regression seems to exhibit a proper fit and was a good choice. Naturally, at this point other interpretive regression models could be used to compare the results as well as fit of the model and to obtain additional interpretation opportunities.

As a second exercise we perform a surrogate modelling task for classification. Similar to the previous analysis, figure 19 shows a surrogate decision tree model trained with the Breiman et al. (1984) classification and regression tree algorithm on the SVM model's predictions of *AgeCat*. The surrogate tree looks extremely similar to the one trained in figure 5. Splitting nodes criteria vary marginally and almost all splitting nodes refer to the same features. The only exception are the splitting nodes of the third and fourth level which are interchanged and only responsible for 19% of the sample's observations. Once again, note that the comparison between the two is merely a test of how close the surrogate tree and the initial decision tree are. The surrogate tree does not mimic the decision tree in figure 5 but the SVM model. As we extensively illustrated how to interpret decision trees in chapter 5, we refrain from any further interpretation at this point. For decision trees a good measure of fit is the accuracy. The model in figure 19 has an accuracy of 80.27%, where 1.744 instances were correctly classified as old and $1,609$ were correctly classified as young. Overall, the decision tree seems to provide a suited and more interpretable alternative to the SVM model.

Table 6: Linear regression results for the surrogate regression in comparison to the linear regression from chapter 5

	Dependent variable:	
	$\widehat{y_{Rings}}$ (Surrogate)	Rings (Model from chapter 5)
SexI	-0.823^{***}	-0.825^{***}
	(0.062)	(0.102)
SexM	0.046	0.058
	(0.051)	(0.083)
Length	0.001	-0.002
	(0.006)	(0.009)
Diameter	0.051^{***}	0.055^{***}
	(0.007)	(0.011)
Height	0.057^{***}	0.054^{***}
	(0.005)	(0.008)
WholeWeight	0.030^{***}	0.045^{***}
	(0.002)	(0.004)
ShuckedWeight	-0.080^{***}	-0.099^{***}
	(0.002)	(0.004)
VisceraWeight	-0.038^{***}	-0.053^{***}
	(0.004)	(0.006)
ShellWeight	0.055^{***}	0.044^{***}
	(0.003)	(0.006)
Constant	3.840^{***}	3.895^{***}
	(0.178)	(0.292)
Observations	4,177	4,177
R^2	0.745	0.538
Adjusted R^2	0.745	0.537
Residual Std. Error (df = 4167)	1.337	2.194
F Statistic (df = 9; 4167)	$1{,}355.407^{***}$	538.914^{***}

Source: Own computation.
Note: Stars indicate the significance at the X% significance level, where $^{*} = p < 0.1$; $^{**} = p < 0.05$; $^{***} = p < 0.01$. The first column (1) shows the linear regression results for the surrogate model trained on the random forest predictions. The second column shows the linear regression results on Rings *from table 5.*

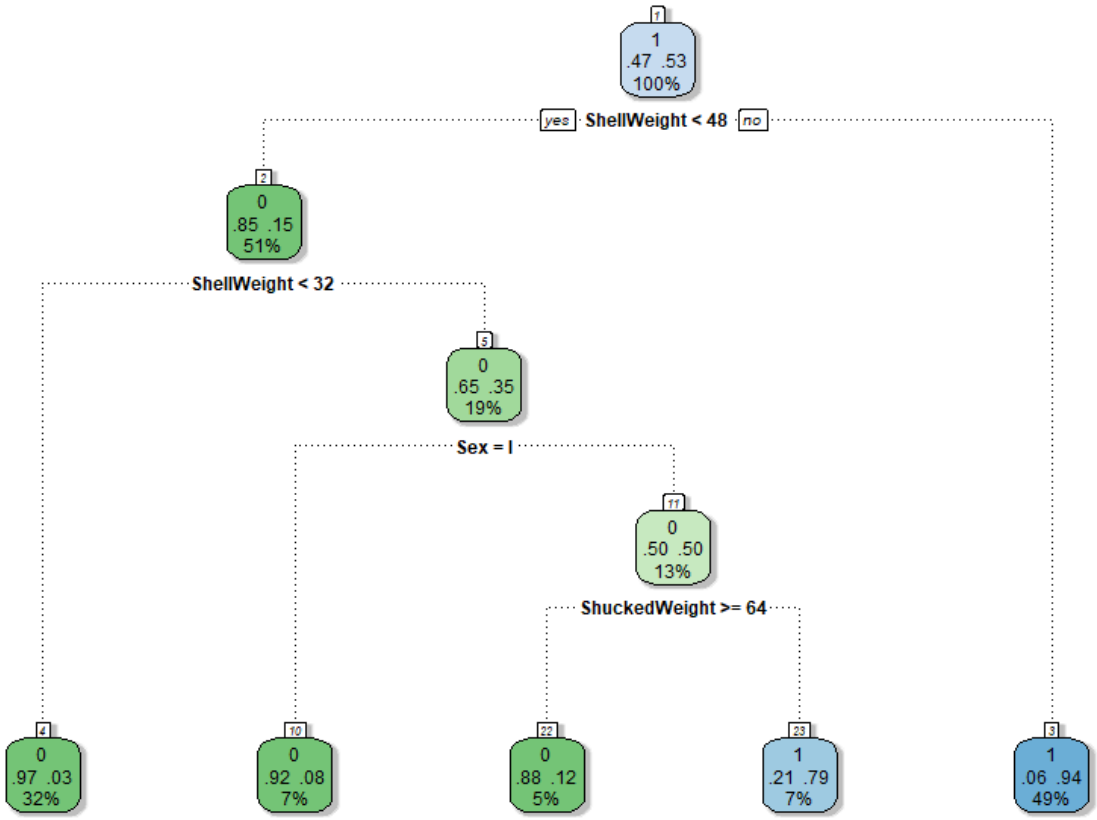


Figure 19: Surrogate decision tree for $\widehat{y_{AgeCat}}$

Source: Own computation.
Note: The tree was created with the classification and regression tree algorithm (Breiman et al., 1984). The predicted variable is $\widehat{y}_{AgeCat}$*, the predictions of the SVM model. The Gini index serves as the impurity measure. Each node shows the predicted class, where a blue box signifies "older than 9 rings", i.e.* $\hat{y}_{BB} = 1$*, and green denotes "younger or equally young as 9 rings", i.e.* $\hat{y}_{BB} = 0$*. The first number in each box shows the predicted class. The predicted probability of being younger or older than the mean is displayed as point decimals in the box. The percentage of observations in the node is described by the last figure in the box.*

6.4.3 Discussion of Advantages and Disadvantages

Global surrogate models are extremely flexible w.r.t. the interpretive model choice. An observer can freely select an interpretive model of his preference and then use it as a surrogate model. This can be especially useful when an observer wants to try multiple interpretive models until he finds one with a particularly good fit. Of course, it is important to choose a suited naturally interpretable model and of course each intrinsically interpretive model has its distinctive advantages and disadvantages. Moreover, global surrogate models are extremely easy to implement (Molnar, 2020). If an observer already knows how to implement a ML model, it is very likely that she already knows how to use another ML model as a surrogate. Most measures of fit are also easily understandable and straightforward to interpret but have to be semantically adapted to the context of surrogacy.

Yet, surrogate models do not come with clear instructions w.r.t. the measure of fit or cut-off criteria thereof. Hence, it becomes difficult to assess how well a surrogate model represents the model's internal behaviour. That is, even if the measure of fit is relatively high, it could be the case that the interpretations provided by the surrogate model do not explain a substantial fraction of the sample. For instance, in the case of the linear regression outlined above, it is possible, that the interpretation produced by the linear coefficients only explain 50% of the effect for *WholeWeight*. That is, since global surrogate models are, as the name suggests, global, it may very well be that they aggregate over individual but important observations and, thus, fail to register local anomalies. Moreover, when choosing an intrinsically interpretive model one necessarily has to accept the interpretive model's disadvantages. In the case of the linear regression example above: As we have shown with PDPs and ICEs, some effects are just very complex and the linear regression certainly does not do justice to them. There is always a good reason why not every deployed ML model in practice is an intrinsically interpretive model. Usually, this is because of accuracy reasons. In summary, it is safe to say that if an observer has no knowledge about other model-agnostic interpretability techniques, surrogate models can be a safe choice for achieving global interpretability. Other more sophisticated tools should still be certainly preferred.

6.5 Local Surrogate Models

6.5.1 Local Interpretable Model-Agnostic Explanations (LIME) of Ribeiro et al. (2016b)

The basic idea of local surrogate models is to create an interpretable ML model over an interpretable data representation which is locally faithful to the ML black box. An interpretable data representation in this case designates a data representation which is understandable to a human without knowledge of the feature. For example, an interpretable explanation of a visual classifier may be a binary vector of 0's and 1's conveying information whether a neighbouring chunk of related pixels is present or not.[86] Local fidelity captures the concept of an explanation to be only meaningful, if the behaviour of the interpretable model coincides with the behaviour of the black box model in the vicinity of the instance under investigation.

Local interpretable model-agnostic explanations by Ribeiro et al. (2016b) are the local surrogate pendant to global surrogate models of the previous chapter. The rationale of LIME is to explain a single prediction of a black box ML model by training an intrinsically interpretable model with certain proximity-weights on a perturbed version of the initial training data set and the corresponding predictions thereof around the observation of interest. The weights of the perturbed initial data set are determined by how close they are to the instance under investigation. A LIME is found by minimizing an "unfidelity" function which depends on the choice of the potentially interpretable model, the initial predictions, and a proximity measure, as well as by minimizing a complexity measure which depends strictly on the choice of the potentially interpretable model.

That is, we first need to identify an instance x_i' of the training data for which we desire an explanation. Second, we perturb the training data set x to receive a sample Z. Third, we need to calculate the predictions $\hat{y}_z$ of the black box model with the sample Z. Fourth, we weight the sample data Z in relation to how close the data points are to the observation x_i'. Eventually, we can train an interpretable weighted model g on the sampled data Z and the corresponding predictions $\hat{y}_z$. This interpretable weighted model constitutes a LIME and can then be interpreted accordingly.

The formal definition goes like this: A black box model is designated as $f(x)$, where $f : \mathbb{R}^d \to \mathbb{R}$. An instance $x \in \mathbb{R}^d$ describes the original representation of an instance under explanation with appropriate length d, whereas $x' \in \{0,1\}^{d'}$ denotes a binary vector for its interpretable representation of appropriate length d'. Ribeiro et al. (2016a) define a single explanation as a model $g \in G$, where G designates the entire space of intrinsically interpretive models such as described in section 5, and where $g \in \{0,1\}^{d'}$. That is, g is 1 if the interpretable component is present and 0 if not. Since not every model $g \in G$ might be good enough to be used as an interpretable model, Ribeiro et al. (2016b) define the complexity measure $\Omega(g)$ which depends on the type of interpretable model which is used. For instance, in the case of a decision tree, $\Omega(g)$

[86] If you recall the example of the husky classifier from section 3.1, the highlighted area of snow influencing the prediction of an image to be classified as a wolf represents an interpretable data representation. In this case, information of the feature was red, blue, and yellow values of the graphical pixels. The interpretable representation is completely detached from this, since information about the red, blue and yellow value of a pixel does not matter for determining if a contiguous chunk of pixels is present or not.

describes the size or depth of the tree, whereas in the case of linear regression, $\Omega(g)$ describes the number of statistically significant parameters. The proximity measure $\pi_x(z)$ describes the distance between x and z. The term $\mathcal{L}(f, g, \pi_x)$ denotes the local unfidelity measure of the LIME model. A LIME can be obtained via minimizing the following equation:

$$\xi(x) = \underset{g \in G}{\arg\min}\, \mathcal{L}(f, g, \pi_x) + \Omega(g). \tag{20}$$

That is, local fidelity is achieved by minimizing the locality-aware loss term $\mathcal{L}(f, g, \pi_x)$ which makes sure that g has an optimal fit to the model f. Lower values of the loss term indicate better local fidelity. Interpretability is guaranteed by ensuring that the complexity term $\Omega(g)$ maintains a certain level of complexity which is not too high in order to make human interpretation possible.

LIMEs are a model-agnostic technique. Hence, the locality-aware loss term is approximated by drawing samples and weighting them with π_x. The motivation behind the sampling is that the more information we can gather about the black box by creating artificial data points through the sampling of more data, the better will be our approximation via the interpretable model.

That is, instances around x' are sampled by drawing non-zero elements of x' uniformly at random to mimic the real data distribution as close as possible. The number of such draws is also uniformly sampled. This perturbed sample $z' \in \{0, 1\}^{d'}$ contains a fraction of the non-zero elements in x' and is in turn used to reconstruct the sample in the "original representation" $z \in \mathbb{R}^d$. Moreover, it is used to calculate the predictions with the trained black box model. This perturbed data set z' and the corresponding predictions thereof are then used to minimize the LIME ξ in equation 20.[87]

6.5.2 Example Interpretation

Once again, we illustrate the interpretability technique for a regression and classification task.[88] Figure 20 shows a LIME example for the regression task for all eight features in the sample. We first calculate the random forest model and consecutively explain it with the LIME model. We choose the familiar observation 28 as the instance of interest. The plot shows the feature weights of a local surrogate linear regression. Red bars show a negative influence, whereas blue bars show a positive influence. Each feature is accompanied by two proximity borders delineating the upper

[87] The actual sampling process in software implementations of LIME is quite different from the theoretical description above. For instance, the *R* implementation of LIME by Pedersen and Benesty (2018) estimates a univariate distribution for each feature and then simply draws out of it. The disadvantage of this is that the information of the estimated covariance between the features is ignored, because the estimated entire feature space is basically a product of multiple univariate distributions. Theoretically, this could lead to problems when unrealistic feature values bias the calculated predictions. The results of the LIME should, therefore, not be trusted, since they do not exhibit local fidelity. Shi et al. (2020) try to remedy this problem by proposing a modified perturbed sampling operation.

[88] Please not that because of implementation issues we cannot use the random forest and SVM model of the previous chapters. The current LIME implementation in *R* unfortunately does not offer an interface for random forest models of the *randomforest* or SVM models of the *e1071* packages which we have used previously. Therefore, we calculate very similar models with the *mlr* package. The difference between the previous random forest and the new model is negligibly small. The same holds for the SVM model. However, comparisons of previous and LIME explanations should be treated carefully since we technically interpret different models even though they are similar.

Table 7: Sample values for observation 28

	Sex	Length	Diameter	Height	WholeWeight	ShuckedWeight	VisceraWeight	ShellWeight	Rings
Obs. 28	M	118	89	28	186.2	71.2	46.8	56	12

Source: Abalone data set.

and lower bounds of the local-fidelity regions. We provide the actual values of observation 28 in table 7 as an anchor of reference. The prediction of the LIME regression is $10.74\bar{3}$. Since the observed value of observation 28 was 12, the prediction was moderately close. The explanation fit of the entire model is relatively low with an R^2 of 0.049. Maybe more complex and non-linear surrogate models would have been a more suited choice in order to achieve a better model fit. Yet, they come with the disadvantage of less interpretability compared to the linear regression.

As can be seen in figure 20 *ShuckedWeight* and *Height* have a strong and negative effect on the prediction of *Rings*. All other factors have a positive effect on the prediction. *ShellWeight* and *WholeWeight* have the largest positive effect. The effect for *Sex = Male* is also considerably higher than it has been in previous techniques. *Length* exhibits an effect of very close to zero. Thus, all found effects for the four variables of interest are in line with the results of the previous chapters. Naturally, some coefficients will be different in sign and size to previous results, since we observe only a small local region of the feature space. This can happen, when the effect for a small region of the feature is different than for the global feature. The possibility to highlight such discrepancies in the feature effect is exactly what makes LIME models so special. The effect for *VisceraWeight*, for instance, was negative in the linear regression in chapter 5 and is now positive in the LIME model for observation 28.[89] Performing a LIME analysis for multiple observations can, thus, help shine more light on a feature effect for a specific region in the feature sub-space. Moreover, they can help in identifying strong or weak prediction influence of several features. We perform a LIME analysis with eight covariates, although it is also common to reduce the predictor space to two to four features of particular interest.

[89] Even though, it has to be noted, that the effect of *VisceraWeight* is relatively small in both cases.

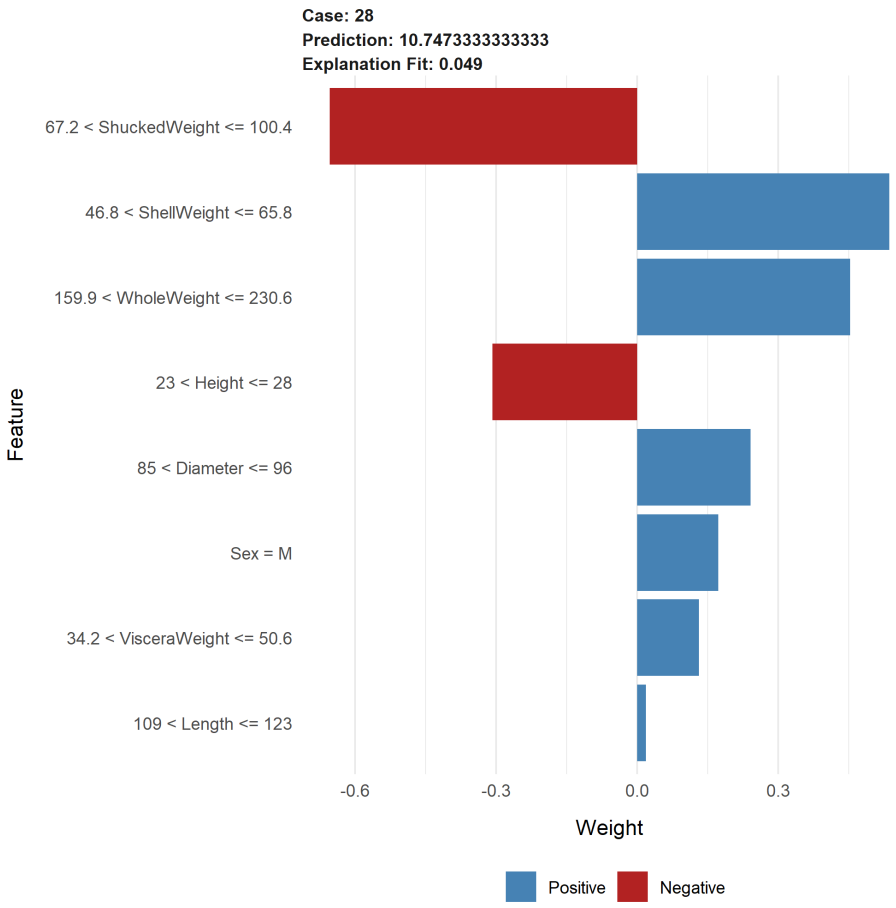


Figure 20: LIME plot for Rings *for observation* 28

Source: Own computation.
Note: This plot shows a LIME model for the random forest model predicting Rings*. The intrinsically interpretive model used is a linear regression. Case identifies the number of the observation in the sample. Prediction gives the predicted response of the LIME model, which is the intercept plus the sum of the coefficients. The Explanation Fit gives the R^2 of the LIME linear regression. Each row corresponds to a feature. Each feature is accompanied by the borders delineating its local proximity. Features exhibiting a positive correlation with the outcome are shown in blue, a negative correlation between feature and outcome is depicted as red.*

Figure 21 shows the classification pendant to figure 20. Classification plots for LIME are very similar to LIME regression plots. First, we compute the SVM model on *AgeCat* and then calculate a LIME model on this black box model. The logistic regression is chosen as a local surrogate model.[90] The plots are very similar in structure. *ShuckedWeight* has, similar to the regression counterpart, a very strong negative influence, while *ShellWeight* has a large positive effect on the prediction. From here on, the plots differ. The effect for *WholeWeight* is positive

[90]We did not choose decision trees for classification as a local surrogate in this case, since the use of decision trees for LIME is generally not recommended because of robustness concerns. It is important that LIMEs are robust to slight variations in the input data and as discussed in section 5.2 this is a property which decision trees lack. Moreover, decision trees are better suited to facilitate global interpretation.

but appears to be less strong than for the regression. Obviously, we have to be careful with the interpretation of the size of the coefficients, since we interpret two distinct x-axes and the absolute size of a logistic regression's coefficients should not be interpreted. *Sex = M* is the only other feature with a positive effect. *VisceraWeight*, *Height*, and *Length* all have a negative but very small effect on the prediction. Categorical features are usually easier to interpret in LIME analysis, as the value of the effect is fixed. It is different to gauge more than relative effect sizes and signs for numerical features, since the effect size is peculiar to the specific proximity region as identified by the LIME model.

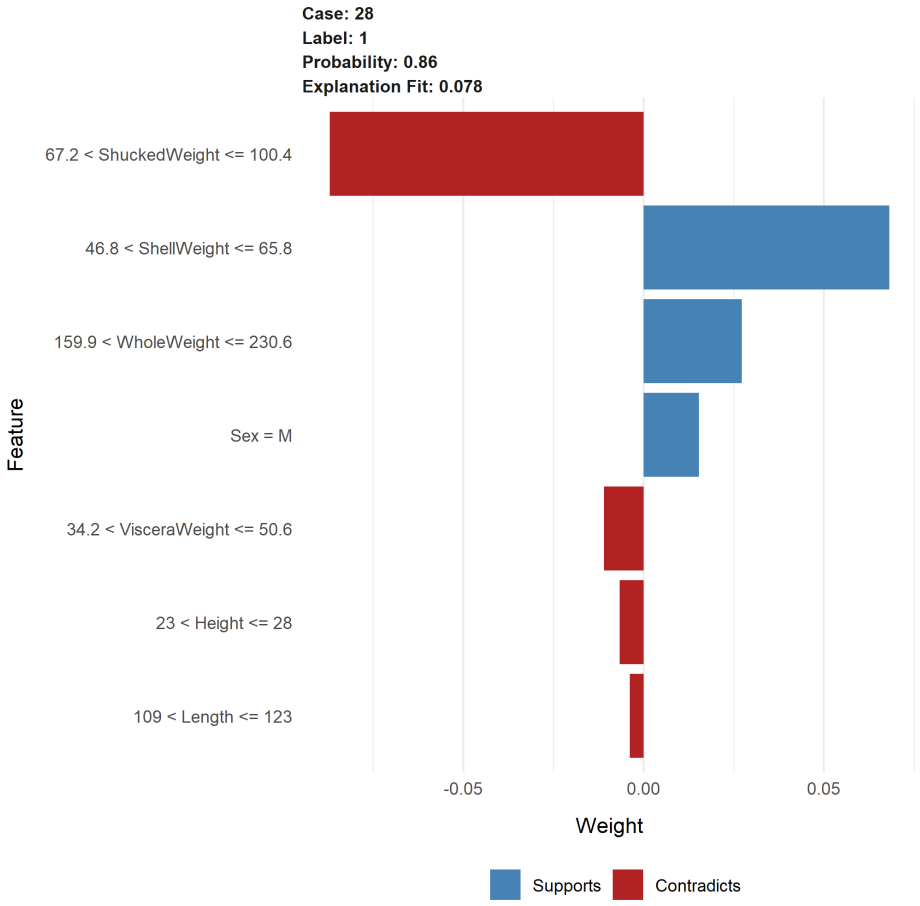


Figure 21: LIME plot for AgeCat *for observation* 28

Source: Own computation.
Note: This plot shows a LIME model for the support vector model predicting AgeCat*. The intrinsically interpretive model used is a logistic regression. Case identifies the number of the observation in the sample. Prediction gives the predicted response of the LIME model, which is the intercept plus the sum of the coefficients. The Explanation Fit gives the* R^2 *of the LIME logistic regression. Each row corresponds to a feature. Each feature is accompanied by the borders delineating its local proximity. Features exhibiting a positive correlation with the outcome are shown in blue, a negative correlation between feature and outcome is depicted as red.*

6.5.3 Discussion of Advantages and Disadvantages

LIMEs are a local interpretability technique which are incredibly popular and have been employed in multiple different fields.[91] They offer a simple and accessible way to get the local interpretation of a specific instance or a set of instances. LIMEs do not require an extensive amount of information about the technique to actually interpret the instance. Only knowledge about LIME and the local substitute model, the intrinsically interpretive model is required for interpretation. Thus, LIME models can be a practicable alternative for local interpretation to ML novices. Moreover, LIME produces human-friendly explanations, as LIME focusses on giving short and preferably contrastive explanations of an observation (Molnar, 2020). Since a LIME - or rather the local intrinsically interpretive model - comes with its particular measure of fit, we can always assess how good the explanation was. From a computational perspective LIME is certainly preferable as most implementations of intrinsically interpretive models (and LIME) are computationally optimized to a high degree. Even though we restrict our analysis to tabular data in this thesis, it is important to note that LIME can be used for multiple types of data. LIMEs are also a very popular method for image- and text-based local explanations.

As stated previously, the sampling process of LIME can be a huge issue. Defining the proximity measure is often an empirical problem formulation, which has not yet been solved. As stated in footnote 86, the sampling processes for LIME currently implemented do not account for the correlation between features. Thus, it can easily be the case that explanations produced by LIME are not faithful to the real model, as these wrongly sampled data points which do not reflect the actual data are used to create a wrong local approximation. It is, thus, always recommended to perform repeated sampling of the observation under investigation to verify that the sampling process is locally fidelitous. Moreover, it has been reported that LIME models often do not produce very robust explanations as they are prone to variations in the sampling process and the selection of proximity measures (Alvarez-Melis and Jaakkola, 2018; Molnar, 2020).

[91] For an application of LIME for music analysis see Mishra et al. (2017); for LIME in graph neural networks see (Huang et al., 2020).

7 Discussion: How to Integrate Post-Hoc Model Agnostic Interpretability Techniques in a Researcher-Oriented Workflow

In this section we discuss how to include the five model-agnostic interpretability techniques introduced in the last chapter into a researcher-oriented ML workflow. The term researcher-oriented simply designates a workflow in which the internal workings of a black-box model are uncovered and other criteria next to ML accuracy and algorithm efficiency are also considered.[92] We structure our discussion in the form of five key recommendations. We make no claim to completeness of the listed inclusion recommendations, they should rather be considered as important guiding principles for including interpretability in a ML workflow.

Integrating Interpretability Techniques in a Researcher-Oriented Workflow

Let us consider the case when a researcher or data scientist wants to interpret her black box ML model. There are multiple things she needs to consider before the analysis. The first and arguably most important prerequisite is to be aware and make a plan of what is actually going to be the subject of interpretability in the analysis. It is crucial to be aware of the subject and level of interpretation before the model training and interpretability analysis. That is, does the observer want to interpret a specific set of features or all of them in order to understand the model better? Which interpretability approach can be taken under the specific circumstance? Can an interpretable model be used or is the model selection fixed because of certain circumstances which in turn requires post-hoc model agnostic interpretability techniques or the use of model-specific interpretability techniques? These questions all imply different interpretability scenarios which require distinct approaches. It is, therefore, difficult to give universal advice about the inclusion of the concept of interpretability in a researcher-oriented workflow when no specific requirements about interpretability are made in advance. Nevertheless, we try to develop a set of inclusion recommendations which facilitate the use of the techniques introduced in this thesis and is as general as possible.

R1: Interpretability starts before the training process.

First, one important requisite for interpretability which has to be determined before the model training process is data interpretability. It is impervious that the data of the underlying model is in an interpretable state to facilitate any further interpretability-related actions. Data interpretability simply describes the concept of having data which is in a humanly understandable and accessible form (Kim and Doshi-Velez, 2018). While text, audio, or visual data is generally harder to transform into an interpretable form, most types of tabular data are naturally interpretable, if they are not extensively pre-processed. Yet, as stated in section 3 there are multiple levels of data interpretability. What suffices as interpretability to one observer may not satisfy another

[92] Consider for example a researcher working in a non-ML scientific domain, e.g. public health, finance or climate change, who uses a ML system in her research. Obtaining qualitative or quantitative insights not directly pertaining to the ML model can be translated into new hypothesis or substantiate existing research.

(Lipton, 2018). For instance, a researcher interested in learning the direct local effect of an observation naturally needs the data to be in an interpretable state s.t. she can make statements about the model's logic based on the underlying data. The abalone data set in our analysis is - after we reversed the pre-processing previously done - in such a state that direct statements about the fundamental biological relationships between features and the output are possible. However, in other analyses an observer might consider the sheer behaviour of an effect to meet his interpretability demands. That is, the state of the data does not matter, as the observer is only interested in the trajectory of the effect. It is, hence, crucial for data interpretability to be aware of the level of interpretability one desires and the level of interpretability the data and model situation permits.

Moreover, for some ML models it might be preferable to transform the data such that it can be more easily processed by the optimization algorithm. For instance, the accuracy of artificial neural networks can be increased by normalization pre-processing techniques which transform the data via min-maxing, using z-scores, or decimal scaling (Nawi et al., 2013). Unfortunately, most normalization techniques change the natural interpretable state of the data in such a matter that the data cannot be interpreted directly. Normalizing all scores from zero to one may make the algorithm run faster, however, the data almost completely looses most of its interpretable information. While all five techniques introduced in this thesis can still show the effect's behaviour of a certain feature, they can hardly be interpreted on a real-world level. Thus, it is important to be also aware of the level of interpretability one desires and the corresponding consequences during the ML workflow.

To summarize: There are multiple important factors concerning interpretability which have to be considered before the analysis: First, the observer needs to define the purpose for the inclusion of interpretability in her ML workflow in order to lay the ground for interpretability requirements in the ML analysis. Usually, the purpose of including interpretability in a ML workflow designates such borders. Yet, it may still be meaningful in some analysis to individually clarify the goal and limits of interpretability before the analysis starts. Second, the purpose of interpretability then has to be translated into a set of demands or restrictions in the ML workflow. For instance, it is extremely important to ensure that the data suffices the interpretability requirements one is interested in. Data interpretability is relevant in most model-building- and interpretability-related steps. In some cases there can even be a trade-off between data interpretability and ML algorithm processability of the data.[93] Being aware of the data granularity before the analysis is important for interpretability. More important is that it is ensured that the interpretability restrictions imposed on the ML workflow do not interfere with the initial purpose of the ML workflow, if this initial purpose is of course everything but achieving more interpretability. Unfortunately, we cannot give more concrete advice at this point. It is difficult to give recommendations on how to minimize the effect of such interpretability restrictions on a ML workflow, since the latter are generally very individual and recommendations have to be adapted to the specific context. For example, in some analyses certain data pre-processing of a particular feature might satisfy

[93] This may be especially true for larger data sets or ML training processes where the parallelization of calculations is not possible.

the algorithm processability as well as interpretability demands. It is, thus, not possible to give specific advice on how to successfully juggle interpretability restrictions and the initial goal of the ML workflow. Therefore, we recommend to plan how to include interpretability before the training process starts to set certain deliverable targets and be aware of the restrictions which interpretability may pose during the ML analysis.

R2: Not every model-agnostic interpretability technique is suited for every application and circumstance.

In section 6 we explain five different model-agnostic interpretability techniques. As with most ML models each interpretability technique is best applied in different contexts. Context in this case describes among others the scale of the model under interpretation. PDPs, ALEs, and global surrogate models visualize a global summary of the feature and are, thus, best used when global interpretability is desired. PDPs and ALEs are great feature visualization techniques, when an observer desires to understand the effect of a feature over its entire distribution. Both techniques can be used to receive an estimated version of the predicted effect for a feature. PDPs and ALEs can be adapted in such a way that each technique shows whether the prediction is over- or under the average prediction, i.e. if the estimated effect has a positive or negative effect on the prediction. Higher-dimensional PDPs can show whether there is a significant interaction for specific regions in the feature sub-space. Similarly, second-order ALEs can highlight more complex effect behaviour, but they always have to be interpreted together with the main-order ALE in order to give the correct interpretation. Both techniques can, hence, be used to gather an extensive amount of information about a feature's effect in the prediction on a global model scale.

On the other hand, ICEs and LIME models are best utilized when specific instances of the model are under interpretation and an observer is interested in local interpretability. LIME, in particular, is most appropriately used when an observer aims to receive a specific explanation for a feature in a particular region of the feature's distribution. Consider for instance the case of a ML credit card application algorithm. If a customer wants to know why her credit card application was denied, LIME provides a simple explanation in the manner of: "Being the owner of multiple credit cards is associated with a negative effect for a specific sample region". LIME can, hence, be used for the showcasing of quantitative local effect sizes as well as for explanations by example which, in the previous case, could be presented to a customer. ICEs, on the other hand, plot each individual predicted effect per observation and, thus, rather provide insights into larger movements or trends in the data set at large. Thereby, individual influential instances or a set thereof can be identified. It is particularly useful to assess whether a model can capture certain heterogeneous groups in the input data. ICEs cannot be utilized as straightforwardly to develop simple explanations by example as LIME. They should be rather used for interpretability-driven model analysis.

When an observer is interested in both, global and local interpretability, it seems most appropriate to use PDPs and ICEs from the five introduced techniques. Both are theoretically related and, thus, most implementations offer to calculate them at the same time, since ICEs

need to be calculated anyway during the computation of PDPs. As stated in the previous chapter there are however certain concerns when it comes to using PDPs and ICEs. Since PDPs and ICEs often fail to highlight feature interactions and to adequately account for feature correlations in most cases, it is advisable to use ALEs when an observer is worried to be in such a situation. ALEs offer the same global feature summary visualization as PDPs but do lack a local pendant similar to what PDPs share with ICEs. Thus, ALEs and PDPs are from a holistic interpretability perspective not completely interchangeable. In scenarios when global interpretability is desired and feature correlations are high, the use of ALEs is strongly recommended.

Global surrogate models, on the other hand, are a special alternative in the introduced model-agnostic interpretability techniques, since the disadvantages and advantages thereof depend on the ML algorithm which is used as a surrogate and the specific context they are applied in. It is, therefore, hard to make a universally applicable statement about the contextual suitability of global surrogate models. Nevertheless, global surrogate models have their niche. They are, arguably, best used as a first overview of the general direction and size of coefficients, if an observer simply wants to get a quick outline of her black box model. Most interpretable models, however, cannot show the feature's effect across its entire distribution and are, thus, generally a worse alternative to PDPs and ALEs. Especially, as most black-box models are designed to account for increasingly complex effect behaviour, most types of intrinsically interpretive surrogate alternatives are inferior to more sophisticated model-agnostic interpretability techniques. Using a linear regression to achieve full "global interpretability" of a ML model is simply perfunctory. But since a surrogate regression model can be calculated with relative ease, they offer a quick interpretability alternative, when an observer just wants to get a superficial notion about the size and direction of an effect.

R3: Multiple model-agnostic interpretability techniques achieve the best results.

As may have become clear from the previous paragraphs, the best recommendation is to use several techniques in practise, since most interpretability techniques somewhat complement and not substitute each other. A potential way, how the five model-agnostic interpretability techniques can be used in a workflow could look like this.[94] First, the observer has to decide if she wants to use a black box or intrinsically interpretive model. If she chooses the latter, interpretability is in a sense naturally achieved. If she chooses the former then model-agnostic interpretability techniques come into play. First, we recommend computing PDPs for each feature to give a first overview of the features' trajectories. If the observer suspects that there is an interaction between two or more variables she can calculate a two-dimensional PDP for the features under suspicion to check for particularly noticeable behaviour. In a next step, ICEs can be computed to supplement PDPs from a local perspective. If the observer notices conspicuous behaviour of an effect, she can try to reconcile the PDP with the ICE in order to discover whether the conspicuous behaviour has causes rooted in local trends. We recommend using

[94]Evidently, there are different reasons of interest for interpretability in a workflow which demand different interpretability approaches as suggested in R1. In this example, we discuss the case of an observer who is interest in raising the general interpretability level of a ML model.

PDPs and ICEs first, since the combination of the two can rapidly foster an initial impression and understanding of the model's behaviour. As a third step, we recommend the calculation of ALE plots in order to verify the results of the PDPs. Since ALEs can account for the correlation between the features, they provide information whether the results of the PDPs were biased or not. If a PDP, for instance, shows a linear effect whereas the ALE shows a non-linear effect, it seems sensible to rather trust the ALE than the PDP. A calculation of the correlation between features may provide more information for this type of analysis. Generally, in cases of higher correlations or a more complex correlation behaviour between features, we advise the use of ALEs. Eventually, a LIME model can be calculated if the observer is interested in the interpretation of single instances or a small prediction region. Global surrogate models can be used if the observer favours an additional global explanation method and can be utilized as necessary. They do not have any particular disadvantages over PDPs and ALEs.

R4: Interpretability is more than just a step in the model building process.

In section 3 we extensively discuss the different perceptions, requirements, and characteristics of interpretable ML. What hopefully became apparent in this chapter is that interpretability is an individual notion which requires specific adjustments when used in a ML analysis. Giving specific recommendations of when to include interpretability in a ML workflow or ML data analysis is problematic, as every ML workflow and analysis is different and interpretability is not just one additional step in a sequence of several data preparation and ML training steps. Imagine, for instance, if interpretability is simply considered as a step after the ML model training to sanity check the behaviour of the effects in the model. What if, the interpretability analysis reveals that a specific feature has no considerable effect in the model and requires special processing for the model to be able to account for the feature. Then the observer would have to go back to the data processing phase and change the feature in such a manner that the model can account for it. Consider another case, when an observer wants to use post-hoc model agnostic ML techniques to gather causal information about a certain relationship between a set of features and an outcome. Naturally, the observer can use post-hoc model agnostic techniques on her data set, but it would make more sense in such an analysis to start with collecting interpretable data in the first place. It is, therefore, not possible to give straight-up advice in the form of: "Include interpretability after the ML training process in a workflow". Interpretability should not be used as an additional step in the model building process. Or as Lipton (2018) states, model-agnostic interpretability techniques should not be blindly used in order to "placate subjective demands" (Lipton, 2018, p. 21-22). It seems more sensible to sharpen the observer's perception of interpretability by giving recommendations on what to consider in the ML workflow, which is what we aim for in this discussion section.

Purposeful interpretability in ML requires extensive considerations before, during, and after the ML model training. It requires the observer to think of the ML model in more terms than just accuracy and efficiency. For example, before the data is collected and the analysis starts, it is important that one actively specifies why and what type of interpretability is desired. Interpretability requirements usually differ in their context and application, which is

why it is important to specify the purpose of interpretability. For instance, there are different interpretability requirements for a ML cancer detection system in a hospital or the ML credit scoring system of a bank. For medical diagnosis it might be interesting to get global interpretability of the knowledge a ML model has learned in order to learn something new about cancer diagnosis, whereas for a ML credit scoring system it might be especially important to give justificatory notions to lone applicants which rather requires local explanations. Interpretability is thus an extremely important factor which has to be considered along the entire ML workflow. With the rise of regulatory interest in uncovering ML black boxes this should also likely hold for ML system owners which have no direct interest in making their black box more transparent.

Before the data is preprocessed, it is crucial to ensure that the data is in an interpretable state. Thus, interpretability generally starts in the data acquisition phase. That is, features which are meaningful to interpretability should be collected to enable real-world interpretability in applications where it is desired. During the pre-processing of the data it is important that the interpretability of the data is assured and the data is eventually in an interpretable and processable state and the interpretability restrictions w.r.t. the data are aligned with data's processability.

In this thesis we propose two avenues for interpretability during the model selection process. First, the intrinsically interpretive models from chapter 5 can be used to build a naturally interpretable ML model. Second, the selection and tuning of a specific ML model can proceed detached from most interpretability considerations, if the observer chooses one of the model-agnostic interpretability techniques from chapter 6. Both alternatives have different consequences and potential for interpretability. The latter is also a large part of the model selection process. In particular, there might be a trade-off between a model's natural interpretability and its predictive accuracy as suggested in Molnar (2020) and Ribeiro et al. (2016b). The adage goes that intrinsically interpretive models perform naturally worse w.r.t. predictive accuracy than black box ML models. Of course, this trade-off is not universally valid, since it depends on the purpose of the application and the respective data situation. Yet, in practise it seems plausible to assume that, for instance, a linear regression model performs worse at a certain complex task compared to an artificial neural network. Therefore, in some scenarios where a certain level of predictive performance has to be achieved it might be advisable to use a black box ML model to achieve a certain level of predictive accuracy. Consecutively, one can use a post-hoc model-agnostic interpretability technique to explain the predictions and the behaviour of the black box. The results of the five model-agnostic interpretability techniques and the intrinsically interpretable models can then be used to provide explanations tailored to the specific context, area of application, and target audience. Alternatively, the results can motivate going back to earlier steps in a ML workflow. To summarize: It is extremely important to first define what type of interpretability one requires in a ML analysis and then to consider interpretability during all steps of a ML workflow.

R5: Model-agnostic interpretability techniques are a powerful and promising tool.

Model-agnostic interpretability techniques are a powerful tool for achieving interpretability in a black box model. Not only because of their wide applicability and flexibility w.r.t. model explanation, and representation (Ribeiro et al., 2016a) but also because how well they can visualize global and local effects of the model under investigation. We highlighted in section 6 to what extent the introduced interpretability techniques can show the prediction behaviour of not only the underlying ML model but also characteristics of the underlying ML algorithm. Being able to show such ML algorithm-individual prediction specific behaviour questions the necessity of model-specific interpretability techniques. Moreover, being able to showcase in what way the predictions of two different models disagree can be a very important factor during model selection, when the observer can see in more detail how well a certain model fits the data.

The potential for interpretability techniques, especially model-agnostic ones, is vast (Ribeiro et al., 2016a). Information or knowledge about the model can be inferred and transferred to other scientific research areas. As we showcase during the demonstration of the empirical example for each technique in chapter 6, the results of the interpretability techniques can provoke interesting food for thought for domain-related and -unrelated scientific research. Issues of fairness, ethical behaviour, and privacy awareness of a model can thus be assessed and verified to a large extent by looking at how the model decided for individual observations with local interpretability techniques. In some cases - when the underlying data situation allows it - even causal analysis can be done by assessing and comparing the results of multiple interpretability techniques. This could enable the generation of new hypothesis or the analysis of previously unsolved scientific research questions.

These are just a few suggestions for further research. More alternatives are always conceivable. However, it is important to note that interpretability techniques are not only a simple tool but solve a fundamental problem in ML research. To come back to the deductive-nomological model of Hempel and Oppenheim (1948) in chapter 3, which states that science is the search for explanations of the incidence of events to find general laws which govern this world, then interpretability techniques allow a deductive-nomological analysis for explaining computerized ML-based systems to a great deal. That is, interpretability techniques facilitate the analysis of the decision-making process and the knowledge acquisition process of ML systems. It may now seem trivial to find out in an analysis that the weight of an abalone has a positive effect in the prediction of its age, but in more complex application cases in the future the ability to observe the internal workings of a ML system in order to trace back and comprehend the decision-making process behind a prediction can constitute a remarkable scientific finding. Even most humans cannot fully explain their motivation or reason behind a specific decision. With interpretability techniques, however, we can exactly identify the reason why a machine made a certain decision. Hence, in the future we may be better at explaining the decisions of machines rather than our own decisions. Interpretability techniques are thus indeed a very powerful tool and a research field which we argue will develop considerably in the next few years and receive a prominence in the domain of ML.

8 Conclusion

This thesis introduces the reader into the field of interpretability and ML. We start by discussing several ML definitions from a conceptual perspective and explaining of important ML terminology in order to lay the theoretical groundwork for further analyses. The conceptual perspective of ML is often a neglected perspective, yet, one which is necessary, since interpretability is a rather qualitative term.

Next, we discuss interpretability in ML from multiple perspective to give a thorough introduction into the topic. Interpretable systems in ML can be described as models where an observer can study and understand how inputs are mathematically mapped to outputs. Unfortunately, it is very difficult to define terms such as interpretability and cognates in ML for multiple reasons: Terms used in ML definitions are highly subjective (Krishnan, 2019; Lipton, 2018) and since ML operates at the intersection of multiple fields, interpretability is a domain-specific notion on which there will hardly be a consensus between fields (Carvalho et al., 2019; Rudin, 2019). Despite these conceptual differences we develop a working definition of interpretability in ML. Interpretability represents a passive model characteristic which describes the general ability of a ML model to provide information about its internal workings and structure in a humanly understandable way.

Since interpretability is a demand-driven concept (Doshi-Velez and Kim, 2017), we explore several exemplified groups with different requirement profiles in order to work out descriptions for interpretability characteristics in ML. The most important four groups are, according to Preece et al. (2018): developers, theorists, ethicists, and users. The term interpretability is frequently used to capture non-quantitative ML objectives which are deemed important by the observer but cannot be modelled formally (Lipton, 2018). In a literature review we identify the following important characteristics or desiderata of interpretability which an interpretable ML model should possess: fairness, privacy awareness, trustworthiness, causality, information transferability, informativeness, robustness, and usability of information.

In order to round up the chapter of interpretability, we discuss several different classification schemes of interpretability techniques as there is no unified taxonomy of interpretability techniques in ML. First, Kim and Doshi-Velez (2018) describe three points in the model building process, when interpretability can be achieved: pre-model, in-model, and post-model training. Whereas pre-model interpretability mainly covers data selection and exploratory data analysis, in- and post-model interpretability are actually concerned with the ML model's interpretability. Interpretability in a ML model can be either achieved by using intrinsically interpretive (or transparent) models or post-hoc interpretability techniques. Post-hoc interpretability techniques are either model-agnostic or model-specific. Model-agnostic techniques have several advantages concerning model flexibility and applicability over model-specific techniques (Ribeiro et al., 2016a). On a model level interpretability can be distinguished in local, referring to specific input-prediction mappings, and global, designating the whole model and its parameter structure. Lipton (2018) supplements these classifications by proposing that intrinsically interpretable models can be judged by their simulatability, decomposability, and algorithmic transparency.

He and Arrieta et al. (2020) identify several types of explanations usable for making ML more humanly understandable: text explanations, visual explanations, local explanation, explanation by example, explanation by simplification, and feature relevance explanations.

As the goal of this thesis is to give a thorough introduction into the field of interpretability in ML, we also showcase how interpretability can be implemented on a model level. Therefore, we explain two intrinsically interpretive models as well as five model-agnostic interpretabillity techniques. We discuss linear regression and decision trees for classification because of their popularity and their accessibility. During this, we explain the theoretical background of each technique with a focus on interpretability and consequently demonstrate how to interpret each technique on an empirical example. While linear regression is a useful tool which facilitates easy and accessible interpretation, its greatest advantage is also its greatest drawback: The linearity assumption allows an observer to straightforwardly access and interpret the coefficients, yet, it also forces every relationship in the model to be linear which hardly reflects the complex feature-output relationships we can observe in most ML data sets. Decision trees on the other hand offer excellent interpretability while they are also able to model more complex relationships in the data. Their largest drawback is their susceptibility to slight variations in the training data which can lead to considerable variations in the grown decision tree.

After demonstrating two intrinsically interpretive ML models, we introduce five popular interpretability techniques: PDPs (Friedman, 2001), ICE curves (Goldstein et al., 2015), ALEs (Apley, 2016), global surrogate models (Molnar, 2020), and local interpretable model-agnostic explanations (Ribeiro et al., 2016b). We first explain the theoretical background of each technique, after which we demonstrate the technique empirically for a regression and classification task. Eventually, a discussion of the technique's advantages and disadvantages rounds up each chapter for the respective interpretability technique. PDPs straightforwardly offer a global summary of a feature. They visualize the main effect of a feature on the predicted outcome for the entire feature's distribution. A significant disadvantage of PDPs is, however, that a large fraction of the feature's information is aggregated into one trajectory. ICE curves can provide remedy to this. They provide a feature's estimated effect on the prediction for each observation and, thus, offer individual local explanations for the entire sample. Aggregating the information of all ICEs for a specific feature results in the feature's PDP. Both techniques are thus quantitatively closely related and can be used together. Because most ICE plots often contain a considerable amount of visual information, it is often difficult to assess individual trends in the data. A remedy for this comes in the form of c-ICE plots which are a centered version of ICE plots where all ICE curves are forced to begin on a specific intercept. This facilitates the differentiation of ICE curves' slopes and makes it easier to identify smaller trends.

One problem with ICEs and PDPs is, however, that both fail to correctly estimate a feature's effect, when there are correlations or interactions between features in the data. ALEs are an excellent alternative for a global feature summary to PDPs in such situations, as they are able to account for between-feature correlations. ALEs can be used to calculate main and second-order effects of a feature and, thus, give an overview about a feature's main estimated effect as well

as interactions between features. In situations of high correlation ALEs are a more robust alternative to PDPs. Unfortunately, there is no local pendant of ALEs.

Global surrogate models can also be used to create global feature summaries by explaining the predictions of a black box ML model with an intrinsically interpretive model. The observer can freely choose which interpretable model she wants to use for interpretation. Each interpretable model has its own advantages and disadvantages which have to be considered before the analysis by the observer. Global surrogate models, unfortunately, do not come with clear criteria for measuring their fit. It is, therefore, often difficult to assess how well the surrogate model actually fits the data. LIME models are a local pendant to global surrogate models. The idea behind local surrogate models is to train an interpretable ML model on an interpretable data representation which adequately replicates the black box ML in the vicinity of the local region which is under interpretation. By using a local sampling of the region under interpretation LIME create a detailed approximation of the effects for the observations under investigation. LIME are great for achieving local interpretability. There are, however, problems w.r.t. the sampling process of LIME implementations which may create non-fidelitous estimators. Thus, the local fidelity of a LIME should always be checked via the respective measure of fit.

In a discussion section we give several recommendations on how to include model-agnostic interpretability in a researcher-oriented ML workflow. The five model-agnostic interpretability techniques which were introduced in this thesis are a powerful and suited tool for making a black box ML model more interpretable. Interpretability always starts before the actual training process, since it requires active considerations about the extent and level of interpretability measures taken during the analysis. Not every model-agnostic interpretability technique is suited for every application and circumstance. Therefore, the respective context needs to be considered before choosing an interpretability technique. Moreover, there is no specific technique which solves all interpretability problems, but in most scenarios a combination of multiple model-agnostic interpretability techniques can achieve better interpretability results.

To conclude this thesis and concur with the epigraph quoting Hawking (2018) in the introduction. Yes, our own competence matters in understanding ML (and AI) to a large extent. Interpretability is a promising avenue with which we can regain the competence in understanding most black box ML applications. The concept of interpretability in ML is currently still subject to intensive research efforts. The research potential in this field is huge and we expect the field to grow considerably in the next few years. We hope that the discourse on interpretability as well as the five introduced model-agnostic interpretability techniques and recommendations on how to include the latter in a researcher-oriented workflow provide an extensive and interesting introduction in this highly complex and ever-expanding field. Future research should be invested in the unification and formalization of interpretability notions and requirements in ML in order to promote more descriptive discourses on how to include interpretability in a researcher-oriented ML workflow.

Appendix A Further Figures and Tables

Table 8: Purity measures and variance thereof for the random forest model

Feature	% Increase in MSE	Std. Error	Increase in Node Purity
Sex	0.61367151	0.01662086	1236.53082
Length	3.43737054	0.20209444	3610.16675
Diameter	3.08930171	0.16520857	4614.55196
Height	2.2662344	0.09990291	5545.84689
WholeWeight	5.13933234	0.25123909	5636.16018
ShuckedWeight	7.24117184	0.1760082	5480.47234
VisceraWeight	3.2514002	0.15014625	5142.39083
ShellWeight	7.07928733	0.22790976	9664.58895

Source: Own computation.
Note: The first column refers to the increase in the mean square error (MSE). The second column contains the standard error thereof. The third column describes the increase in node purity.

Table 8 shows the increase in the mean-square error (MSE), i.e. the decrease in accuracy, the standard error thereof, and the increase in node purity measured as the residual sum of squares. *ShuckedWeight*, *ShellWeight*, and *WholeWeight* seem by far as the most important variables. *Sex* has very little influence.

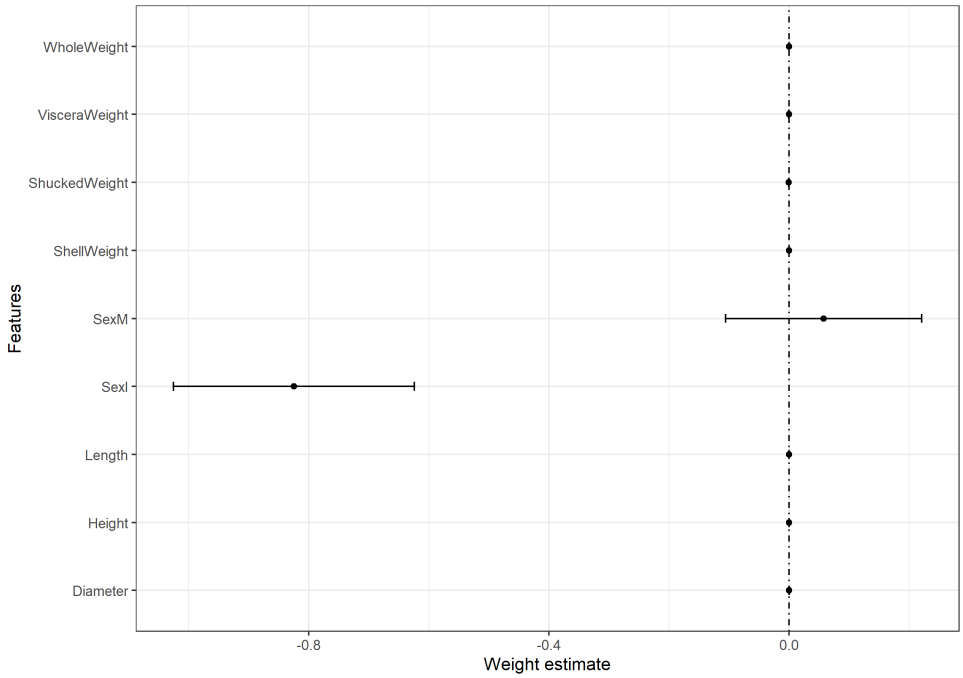


Figure 22: Coefficient plot for linear regression on Rings *with normalized coefficients*

Source: Own computation.
Note: Each dot represents the coefficient's weight from table 5. The brackets around each dot represent the 95% *confidence interval.* SexF *is the reference category.*

Figure 22 shows a coefficient plot with non-processed features. As one can observe, there is not much to observe. Since weight plots visualize all coefficients on the same scale, the effects of smaller coefficients are hardly visible. For example, the coefficient for *WholeWeight* represents the expected change in *Rings* for an increase in one gram of *WholeWeight* and is relatively small in size with 0.045 (see table 5). Effect-wise, it can hardly be compared to the coefficient of *SexI* which represents a change in gender and is almost 19-times the absolute value of $\beta_{WholeWeight}$ with -0.825. Most confidence intervals and parameter estimates, therefore, vary visibly around zero not because they have no effect, but because their effect is quite small as a result of their unit of measure. Preprocessing the weights to zero mean and a standard deviation of one can eliminate this problem. Coefficient sizes are, thus, more accessible to an observer.

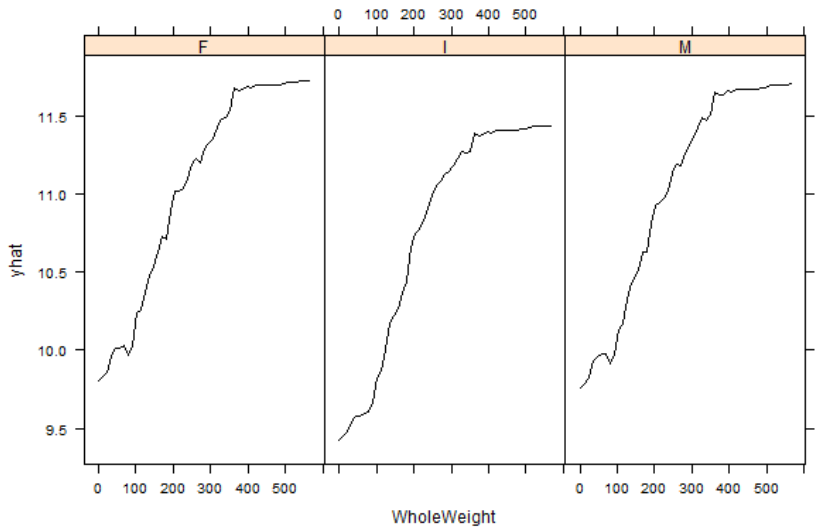


Figure 23: Two-dimensional PDP of WholeWeight *and* Sex *for* Rings

Source: Own computation.
Note: Plot shows the partial dependences of WholeWeight *and* Sex *calculated from a random forest model with the dependent variable* Rings*. The x-axis shows the scale of* WholeWeight*, whereas the y-axis displays the average value of* $\hat{f}$*, in this case the number of* Rings*. Each plot represents a different category of variable* Sex*: female (F), infant (I), and male (M).*

Figure 23 shows the two-dimensional PDP plot for *WholeWeight* and *Sex* predicting *Rings*. Naturally, when plotting a PDP for a numerical and a categorical feature, the different scales must be taken into account. The plot, therefore, shows three individual charts for the three categories *SexF*, *SexI*, and *SexM*. While the effect is similar in shape for all three categories, category *SexI* is shifted downward along the y-axis suggesting that the effect for infant abalones is smaller than for male and female abalones. In accordance with previous results, the trajectory for male and female abalones looks remarkably similar.

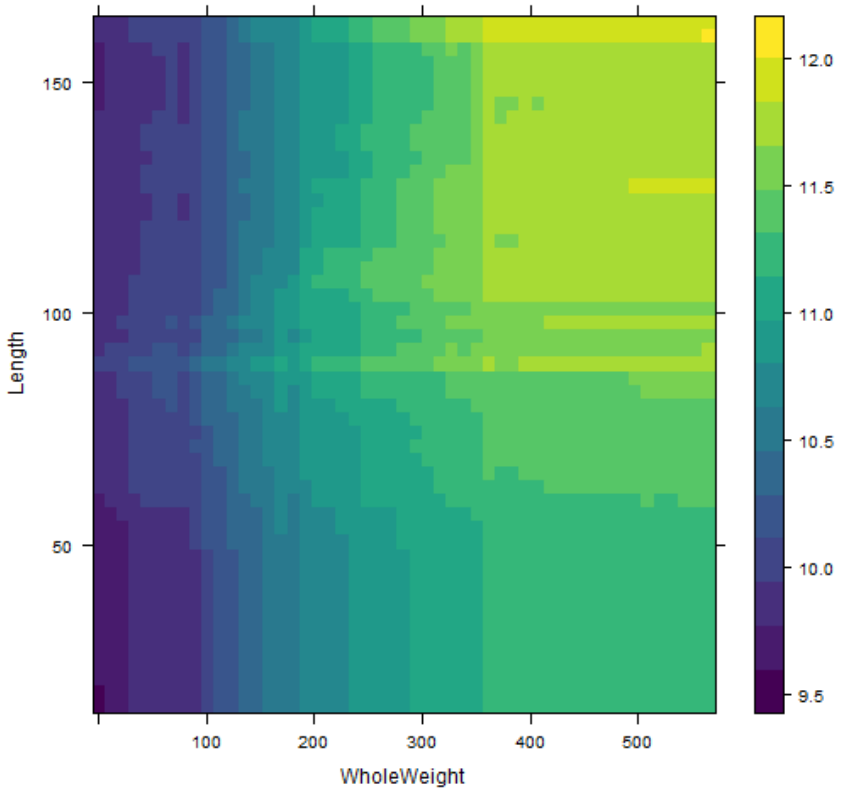


Figure 24: Two-dimensional PDP of WholeWeight *and* Length *for* Rings

Source: Own computation.
Note: Plot shows the partial dependences of WholeWeight *and* Length *calculated from a random forest model with the dependent variable* Rings*. The x-axis shows the scale of* WholeWeight*, whereas the y-axis shows the scale of* Length*. The color scale displays the average value of* $\hat{f}$*, in this case the number of* Rings*. A brighter tone indicates a higher prediction, a darker tone shows a lower prediction.*

Figure 24 shows the two dimensional PDP for *WholeWeight* and *Length*. This plot clearly highlights the step function prediction behaviour of the random forest model. The vertical lines in the plot almost resemble a contour plot of a geographical profile of a mountain range. The different shades of green and blue indicate the decision region of *WholeWeight* which exactly correspond to the splitting nodes in the random forest model. Random forest models can model linear effects only very inefficiently. This becomes very clear in figure 24. *WholeWeight* seems to have a linear or monotonously increasing effect which the random forest models via step functions. This insight could be translated into pre-processing *WholeWeight* such that the random forest processes the feature more easily. Or, if other features experience similar issues, one could change the ML model to one which can model linear relationships more straightforward.

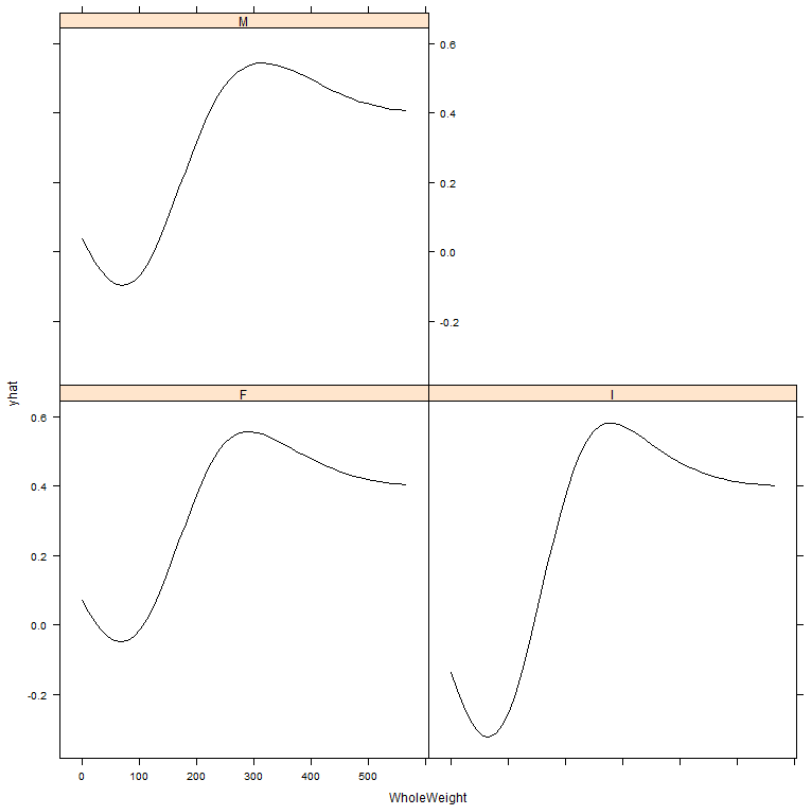


Figure 25: Two-dimensional PDP of WholeWeight *and* Sex *for* AgeCat

Source: Own computation.
Note: Plot shows the partial dependences of WholeWeight *and* Sex *calculated from a support vector machine model with the dependent binary variable* AgeCat*. The x-axis shows the scale of* WholeWeight*, whereas the y-axis displays the average value of* $\hat{f}$*, in this case the probability of being old or young. Each plot represents a different category of variable* Sex*: male (M), female (F), and infant (I).*

Figure 25 shows the two-dimensional PDP plot for *WholeWeight* and *Sex* predicting *AgeCat*. As in figure 23, when plotting a PDP for a numerical and a categorical feature, the different scales must be taken into account. Hence, the PDP is again divided into three individual charts for the three categories *SexF*, *SexI*, and *SexM*. There are no large differences in the shape of the effect. However, the PDP trajectory for *SexI* is more steep than for the other two categories. Once again, the trajectory for male and female abalones looks remarkably similar.

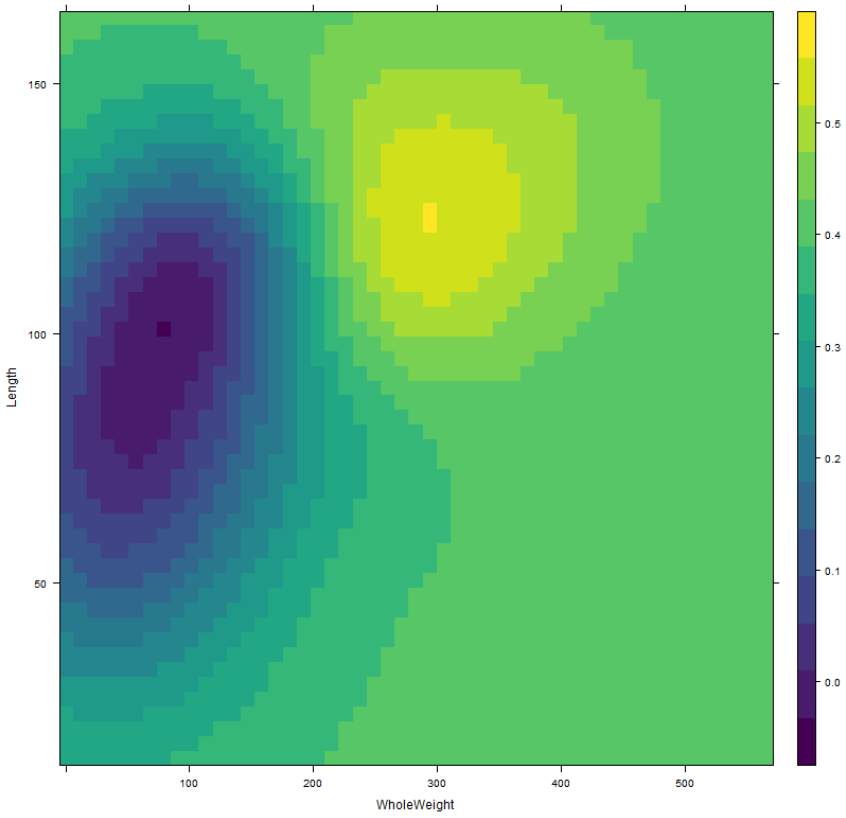


Figure 26: Two-dimensional PDP of WholeWeight *and* Length *for* AgeCat

Source: Own computation.
Note: Plot shows the partial dependences of WholeWeight *and* Length *calculated from a support vector machine model with the dependent binary variable* AgeCat*. The x-axis shows the scale of* WholeWeight*, whereas the y-axis shows the scale of* Length*. The color scale displays the average value of* $\hat{f}$*, in this case the probability of being old or young. A brighter tone indicates a higher prediction, a darker tone shows a lower prediction.*

Figure 26 shows the two-dimensional PDP for *WholeWeight* and *Length*. The yellow and blue kernels show how the radial kernel of the SVM model affects the effect structure of *WholeWeight* and *Length*. Clearly, there are two centers of effects which exhibit a circular shape. The SVM model with a radial kernel estimates radial decision boundaries for the different features. Interestingly enough, there is a sizeable fraction of the sample, the large green space, where predictions do not change.

Appendix B Further Explanations

Questioning the Independence Assumption of Partial Dependence Plots

The crucial assumption in PDPs is the assumption of independence between features. To showcase how this affects PDPs when the assumpton is violated, we simulate a simple toy example. We model an interaction term in a simulated data set and show that the PDP fails to adequately account for this. Thereby, we simulate the following relationship between y and the interaction term between x_1 and x_2:

$$y = x_1^2 - 100x_1x_2 + \epsilon, \tag{21}$$

where ϵ is an error term which is distributed after $\epsilon \overset{iid}{\sim} \mathcal{N}(0, 0.3)$ and x_1 as well as x_2 are the variables of interest which are distributed as $x_1, x_2 \overset{iid}{\sim} U[-1, 1]$. We simulate each observation $10,000$-times. Figure 27 shows a scatter plot of x_2 against y on the left-hand side. There is a strong interaction and pattern in the data. After this, we calculate a simple decision tree predicting y with the two covariates x_1 and x_2. Consequently, we compute a PDP for this decision tree model. The right-hand side displays this PDP of the decision tree model with two covariates x_1, x_2 predicting our simulated y. The decision tree was created with the classification and regression tree algorithm (Breiman et al., 1984).

The PDP shows that the effect of x_2 on y, hovers around zero. However, since we modelled the relationship, we know that x_2 does have an effect. This effect, however, is an interaction term which is not captured by the PDP. Thus we have shown, that PDPs often fail to account for interactions between features and give a wrong picture of the estimated effect.

The spikes of the PDP signify the individual decision regions of the decision tree.

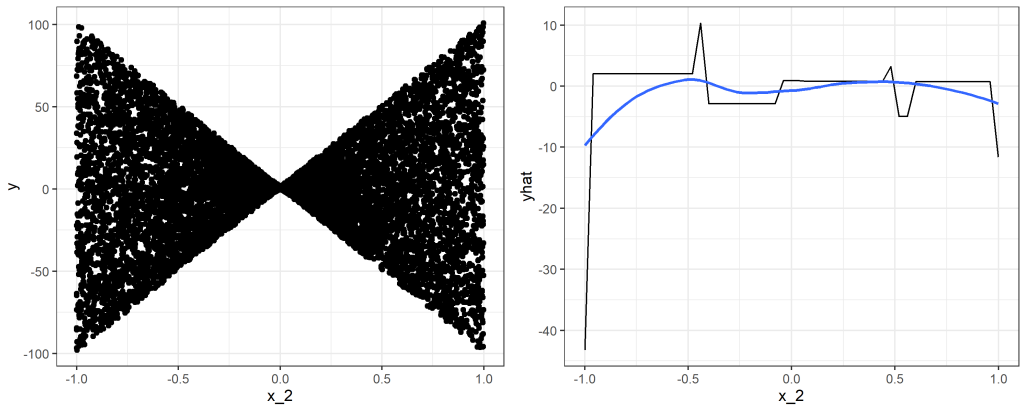


Figure 27: Distribution of x_2-y and PDP of WholeWeight *and* Length *for* AgeCat

Source: Own computation.
Note: The left plot shows the distribution of y for x_2. The right plot shows a partial dependence calculated for a decision tree model. The blue line shows a smooth version of the PDP.

The Calculation of PDPs and ICEs Demonstrated with the Effect Matrix

The calculation of PDPs and ICEs is best visualized as a matrix. This matrix is also used in most software implementation of PDPs and ICEs for calculation purposes.[95] To our knowledge this matrix has no specific designation which is why we refer to it as the effect matrix, since it contains the different effects for variables of interest. The following matrix illustrates an example calculation for a feature s. The complementary feature vector is $x_c = \{x_o, x_p, x_q\}$, such that our feature set is described by $x = \{x_s, x_o, x_p, x_q\}$. The index i refers to the ith value of an observation. Our sample has a length of four observations. The following matrix shows how PDPs and ICEs are calculated for one feature s. We obtain the matrix by simply calculating the average estimated predictions for different values of the feature under investigation x_s and the complementary vector x_c as explained in chapter 6. That is, for the first cell, we input x_{s_1} and o_1, p_1, q_1 into the trained ML model and receive a prediction which is the value of the first cell in the first column. We repeat this for the different values in each row and column. Each row represents the calculated effect for a different observation of s and each column represents fixing the complementary vector at observation 1, 2, 3, and 4 respectively.

$$
\begin{array}{c}
 \\
x_{s_1}(s = s_1) \\
x_{s_2}(s = s_2) \\
x_{s_3}(s = s_3) \\
x_{s_4}(s = s_4)
\end{array}
\begin{array}{c}
\begin{array}{cccc}
x_{c_1} & x_{c_2} & x_{c_3} & x_{c_4}
\end{array} \\
\begin{pmatrix}
\hat{f}(s_1, o_1, p_1, q_1) & \hat{f}(s_1, o_2, p_2, q_2) & \hat{f}(s_1, o_3, p_3, q_3) & \hat{f}(s_1, o_4, p_4, q_4) \\
\hat{f}(s_2, o_1, p_1, q_1) & \hat{f}(s_2, o_2, p_2, q_2) & \hat{f}(s_2, o_3, p_3, q_3) & \hat{f}(s_2, o_4, p_4, q_4) \\
\hat{f}(s_3, o_1, p_1, q_1) & \hat{f}(s_3, o_2, p_2, q_2) & \hat{f}(s_3, o_3, p_3, q_3) & \hat{f}(s_3, o_4, p_4, q_4) \\
\hat{f}(s_4, o_1, p_1, q_1) & \hat{f}(s_4, o_2, p_2, q_2) & \hat{f}(s_4, o_3, p_3, q_3) & \hat{f}(s_4, o_4, p_4, q_4)
\end{pmatrix}
\end{array}
$$

The matrix contains the calculated predicted effects varying by the feature under investigation and the complementary vector. In order to receive the PDP, we need to do one additional calculation step. Each row contains the different supplementary vector values for one value x_{s_i}. That is, the first row contains the calculated four predicted effect values for x_{s_1} when varying over the complementary vector x_c. Thus, if we take the average of a single row i we get the PDP at value x_{s_i}. For example, the calculation for row one goes: $\frac{1}{4}\sum_{i=1}^{4} \hat{f}(x_{s_1}, x_{c_i}) = PDP_{x_{s_1}}$. If we perform this calculation for all four rows and plot the resulting values against their respective value x_{s_i}, we receive a PDP curve for feature x_s.

Individual conditional expectation curves are even easier to obtain from this effect matrix. The first column contains the estimated effects for different values x_{s_i}, when the index of the observations in the complementary vector is one. The second column contains the estimated effects for different values x_{s_i}, when the index of the complementary vector is two and so forth. Taking the first column vector gives us the set of effects $\hat{f}(x_{s_i}, x_{c_1})$ for $i \in \{1, 2, 3, 4\}$, which is nothing but the individual conditional expectation curve for observation one. That is, the first column represents the first ICE, the second column the second ICE, and so on. Plotting all ICE curves for this example is, thus, simply plotting the data points in the columns of the matrix against their respective values $x_{s_1}, x_{s_2}, x_{s_3}$, and x_{s_4}.

[95] The *R* implementation of Greenwell (2017) is an example for this.

The Specification of the Centering Matrix for c-ICE Plots

We calculate c-ICE plots via the following equation:

$$\widehat{f_{c-ice_i}} = \widehat{f_{ice_i}} - \mathbf{1}\widehat{f_{ice_i}}(x^*, x_{c_i}). \tag{22}$$

That is, we first choose an anchor point x^* for the feature s and center all data points at the anchor point. We can achieve this by subtracting each column of the effect matrix of the last sub-chapter by the column of the anchor point x^*. The appropriate dimension of the vector $\mathbf{1}$, hence, refers to the number of the column which was chosen as the anchor point. That is, the column of the anchor point receives the value 1 and all other values are 0. For instance, if we turn back to the effect matrix of the previous sub-chapter and the anchor column is column two, then the vector equals $\mathbf{1} = (0\ 1\ 0\ 0)$. The matrix on the following side illustrates this procedure in a general manner. In this case the star $*$ refers to the number of the anchor column. If, for instance, the anchor point of interest lies in column two, then we subtract column two from column two. Thus, all values for this column are zero and all data points are centered at column two.

$$
\begin{array}{c}
 \\
x_{s_1}(s=s_1) \\
x_{s_2}(s=s_2) \\
x_{s_3}(s=s_3) \\
x_{s_4}(s=s_4)
\end{array}
\begin{array}{c}
x_{c_1} \\
\begin{pmatrix}
\hat{f}(s_1,o_1,p_1,q_1)-\hat{f}(s_1,o^*,p^*,q^*) \\
\hat{f}(s_2,o_1,p_1,q_1)-\hat{f}(s_2,o^*,p^*,q^*) \\
\hat{f}(s_3,o_1,p_1,q_1)-\hat{f}(s_3,o^*,p^*,q^*) \\
\hat{f}(s_4,o_1,p_1,q_1)-\hat{f}(s_4,o^*,p^*,q^*)
\end{pmatrix}
\end{array}
\begin{array}{c}
x_{c_2} \\
\begin{pmatrix}
\hat{f}(s_1,o_2,p_2,q_2)-\hat{f}(s_1,o^*,p^*,q^*) \\
\hat{f}(s_2,o_2,p_2,q_2)-\hat{f}(s_2,o^*,p^*,q^*) \\
\hat{f}(s_3,o_2,p_2,q_2)-\hat{f}(s_3,o^*,p^*,q^*) \\
\hat{f}(s_4,o_2,p_2,q_2)-\hat{f}(s_4,o^*,p^*,q^*)
\end{pmatrix}
\end{array}
\begin{array}{c}
x_c \\
\begin{pmatrix}
\hat{f}(s_1,o_3,p_3,q_3)-\hat{f}(s_1,o^*,p^*,q^*) \\
\hat{f}(s_2,o_3,p_3,q_3)-\hat{f}(s_2,o^*,p^*,q^*) \\
\hat{f}(s_3,o_3,p_3,q_3)-\hat{f}(s_3,o^*,p^*,q^*) \\
\hat{f}(s_4,o_3,p_3,q_3)-\hat{f}(s_4,o^*,p^*,q^*)
\end{pmatrix}
\end{array}
\begin{array}{c}
x_{c_4} \\
\begin{pmatrix}
\hat{f}(s_1,o_4,p_4,q_4)-\hat{f}(s_1,o^*,p^*,q^*) \\
\hat{f}(s_2,o_4,p_4,q_4)-\hat{f}(s_2,o^*,p^*,q^*) \\
\hat{f}(s_3,o_4,p_4,q_4)-\hat{f}(s_3,o^*,p^*,q^*) \\
\hat{f}(s_4,o_4,p_4,q_4)-\hat{f}(s_4,o^*,p^*,q^*)
\end{pmatrix}
\end{array}
$$

Appendix C Software used Throughout this Thesis

All text processing used to create this thesis has been done in LaTeX. All statistical computations were done in the statistical programming language *R*. In the following, we reference all *R* packages which were used throughout this thesis. All statistical results can be reproduced with the *R* code attached to this thesis.

ALEPlot Dan Apley (2018). ALEPlot: Accumulated Local Effects (ALE) Plots and Partial Dependence (PD) Plots. R package version 1.1. https://cran.r-project.org/web/packages/ALEPlot/index.html.

doParallel Hong Ooi, Steve Weston, and Dan Tenenbaum (2019). doParallel: Foreach Parallel Adaptor for the 'parallel' Package. R package version 1.0.15. https://cran.r-project.org/web/packages/doParallel/index.html.

dplyr Hadley Wickham, Romain François, Lionel Henry, and Kirill Müller (2018). dplyr: A Grammar of Data Manipulation. R package version 1.0.0. https://CRAN.R-project.org/package=dplyr.

e1071 David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, Friedrich Leisch, Chih-Chung Chang, and Chih-Chen Lin (2019). e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7.3. https://cran.r-project.org/web/packages/e1071/index.html.

ggplot2 Hadley Wickham (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. ISBN 978-3-319-24277-4. R package version 3.3.1. https://ggplot2.tidyverse.org.

gridExtra Baptiste Auguie (2017). gridExtra: Miscellaneous Functions for "Grid" Graphics. Package version 2.3. https://cran.r-project.org/web/packages/gridExtra/index.html.

ICEbox Alex Goldstein, Adam Kapelner, and Justin Bleich (2017). ICEbox: Individual Conditional Expectation Plot Toolbox. R package version 1.1.2. https://cran.r-project.org/web/packages/ICEbox/index.html.

iml Christoph Molnar and Patrick Scharatz (2020). iml: Interpretable Machine Learning. R package version 0.10.0. https://cran.r-project.org/web/packages/iml/index.html.

lattice Deepayan Sarkar (2008). Lattice: Multivariate Data Visualization with R. Springer, New York. ISBN 978-0-387-75968-5. R package version 0.20.41. https://cran.r-project.org/web/packages/lattice/citation.html.

mlr Bernd Bischl, Michel Lang, Lars Rotthoff, Julia Schiffner, Jakob Richter, Erich Studerus, Giuseppe Casalicchio, and Zachary M. Jones (2016). "mlr: Machine Learning in R." Journal of Machine Learning Research, 17(170), 1-5. R package version 2.17.1. https://cran.r-project.org/web/packages/mlr/citation.html.

pdp Brandon Greenwell (2018). pdp: Partial Dependence Plots. R package version 0.7.7. https://cran.r-project.org/web/packages/pdp/index.html.

randomForest Fortran original by Leo Breiman and Adele Cutler, R port by Andy Liaw and Matthew Wiener (2018). randomForest: Breiman and Cutler's Random Forests for Classification and Regression. R package version 4.6.14. https://cran.r-project.org/web/packages/randomForest/index.html.

rattle Williams GJ (2011). Data Mining with Rattle and R: The art of excavating data for knowledge discovery, series Use R! Springer. R package version 5.4.0. https://cran.r-project.org/web/packages/rattle/index.html.

RColorBrewer Erich Neuwirth (2014). RColorBrewer: ColorBrewer Palettes. R package version 1.1.2. https://cran.r-project.org/web/packages/RColorBrewer/index.html.

rpart Terry Therneau, Beth Atkinson, and Brian Ripley (2017). r part: Recursive Partitioning and Regression Trees. R package version 4.1.15. https://cran.r-project.org/web/packages/rpart/index.html.

rpart.plot Stephen Milborrow (2019). rpart.plot: Plot 'rpart' Models: An Enhanced Version of 'plot.rpart'. R package version 3.0.8. https://cran.r-project.org/web/packages/rpart.plot/index.html.

Stargazer Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables. R package version 5.2.2. https://CRAN.R-project.org/package=stargazer

References

Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160.

Adler, P., Falk, C., Friedler, S. A., Nix, T., Rybeck, G., Scheidegger, C., Smith, B., and Venkatasubramanian, S. (2018). Auditing black-box models for indirect influence. *Knowledge and Information Systems*, 54(1):95–122.

Alang, N. (2017). Turns out algorithms are racist. *The New Republic at https://newrepublic.com/article/144644/turns-algorithms-racist?utm˙content=buffer7f3ea*.

Aldeen, Y. A. A. S., Salleh, M., and Razzaque, M. A. (2015). A comprehensive review on privacy preserving data mining. *SpringerPlus*, 4(694).

Alonso Raposo, M., Grosso, M., Després, J., Fernández Macías, E., Galassi, C., Krasenbrink, A., Krause, J., Levati, L., Mourtzouchou, A., Saveyn, B., Thiel, C., and Ciuffo, B. (2018). An analysis of possible socio-economic effects of a cooperative, connected and automated mobility (CCAM) in Europe-effects of automated driving on the economy, employment and skills. Technical report, Publications Office of the European Union.

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

Alvarez-Melis, D. and Jaakkola, T. S. (2018). On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*.

Anderson, M. and Anderson, S. L. (2011). *Machine ethics*. Cambridge University Press.

Apley, D. W. (2016). Visualizing the effects of predictor variables in black box supervised learning models. *arXiv preprint arXiv:1612.08468*.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. (2020). Explainable Artificial Intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115.

Arya, V., Bellamy, R. K., Chen, P.-Y., Dhurandhar, A., Hind, M., Hoffman, S. C., Houde, S., Liao, Q. V., Luss, R., Mojsilović, A., et al. (2019). One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. *arXiv preprint arXiv:1909.03012*.

Athey, S. (2018). The impact of machine learning on economics. In *The economics of artificial intelligence: An agenda*, pages 507–547. University of Chicago Press.

Awad, M. and Khanna, R. (2015). Machine learning and knowledge discovery. In *Efficient Learning Machines*, pages 19–38. Springer Nature.

Baron, B. and Musolesi, M. (2020). Interpretable machine learning for privacy-preserving pervasive systems. *IEEE Pervasive Computing*, 19(1):73–82.

Barreno, M., Nelson, B., Joseph, A. D., and Tygar, J. D. (2010). The security of machine learning. *Machine Learning*, 81(2):121–148.

Bibal, A. and Frénay, B. (2016). Interpretability of machine learning models and representations: an introduction. In *European Symposium on Artificial Neural Networks*.

Bilgic, M. and Mooney, R. J. (2005). Explaining recommendations: Satisfaction vs. promotion. In *Beyond Personalization Workshop, IUI*, volume 5, page 153.

Binns, R. (2018). Fairness in machine learning: Lessons from political philosophy. In *Conference on Fairness, Accountability and Transparency*, pages 149–159.

Biran, O. and Cotton, C. (2017). Explanation and justification in machine learning: A survey. In *IJCAI-17 workshop on explainable AI (XAI)*, volume 8, pages 8–13.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bornstein, A. M. (2016). Is artificial intelligence permanently inscrutable? *Nautilus at http://nautil.us/issue/40/learning/is-artificial-intelligence-permanently-inscrutable*.

Braithwaite, V. and Levi, M. (1998). *Trust and governance*. Russell Sage Foundation.

Breiman, L. et al. (2001). Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–231.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Brown, R. (2013). *Explanation in social science*. Routledge.

Bunge, M. (1998). *Philosophy of science: From explanation to justification*, volume 2. Transaction Publishers.

Bzdok, D., Altman, N., and Krzywinski, M. (2018). Points of significance: Statistics versus machine learning. *Nature: methods*, 15:233–234.

Cahour, B. and Forzy, J.-F. (2009). Does projection into use improve trust and exploration? An example with a cruise control system. *Safety Science*, 47(9):1260–1270.

Cai, C. J., Jongejan, J., and Holbrook, J. (2019). The effects of example-based explanations in a machine learning interface. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 258–262.

Caliskan, A., Bryson, J. J., and Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Calmon, F., Wei, D., Vinzamuri, B., Ramamurthy, K. N., and Varshney, K. R. (2017). Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*, pages 3992–4001.

Carvalho, D. V., Pereira, E. M., and Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832.

Castelvecchi, D. (2016). Can we open the black box of AI? *Nature News*, 538(7623):20.

Chouldechova, A. and G'Sell, M. (2017). Fairer and more accurate, but for whom? *arXiv preprint arXiv:1707.00046*.

Chouldechova, A. and Roth, A. (2018). The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*.

Cook, K. (2001). *Trust in society*. Russell Sage Foundation.

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

Corbett-Davies, S. and Goel, S. (2018). The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*.

Core, M. G., Lane, H. C., Van Lent, M., Gomboc, D., Solomon, S., and Rosenberg, M. (2006). Building explainable artificial intelligence systems. In *AAAI*, pages 1766–1773.

Cowan, N. (2010). The magical mystery four: How is working memory capacity limited, and why? *Current directions in psychological science*, 19(1):51–57.

Crafts, N. (2004). Steam as a general purpose technology: A growth accounting perspective. *The Economic Journal*, 114(495):338–351.

Dasari, S. K., Cheddad, A., and Andersson, P. (2019). Random forest surrogate models to support design space exploration in aerospace use-case. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 532–544. Springer.

Datta, A., Tschantz, M. C., and Datta, A. (2015). Automated experiments on ad privacy settings. *Proceedings on privacy enhancing technologies*, 2015(1):92–112.

Deo, R. C. (2015). Machine learning in medicine. *Circulation*, 132(20):1920–1930.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87.

Doran, D., Schulz, S., and Besold, T. (2018). What does explainable AI really mean? A new conceptualization of perspectives. In *CEUR Workshop Proceedings*, volume 2071. CEUR.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Doshi-Velez, F. and Kim, B. (2018). Considerations for evaluation and generalization in interpretable machine learning. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pages 3–17. Springer.

Du, M., Liu, N., and Hu, X. (2019). Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77.

Dwyer, D. B., Falkai, P., and Koutsouleris, N. (2018). Machine learning approaches for clinical psychology and psychiatry. *Annual review of clinical psychology*, 14:91–118.

Elliott, G. and Timmermann, A. (2013). *Handbook of economic forecasting*. Elsevier.

European Parliament and Council of the European Union (2016). Regulation (EU) 2016/679.

Forde, J. Z. and Paganini, M. (2019). The scientific method in the science of machine learning. *arXiv preprint arXiv:1904.10922*.

Friedler, S. A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., and Roth, D. (2019). A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 329–338.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE.

Goldstein, A., Kapelner, A., Bleich, J., and Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Goodman, B. and Flaxman, S. (2017). European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57.

Gorissen, D., Dhaene, T., and De Turck, F. (2009). Evolutionary model type selection for global surrogate modeling. *Journal of Machine Learning Research*, 10:2039–2078.

Greenwell, B. M. (2017). pdp: An r package for constructing partial dependence plots. *R J.*, 9(1):421.

Gregor, S. and Benbasat, I. (1999). Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS Quarterly*, pages 497–530.

Grimmer, J. (2015). We are all social scientists now: How big data, machine learning, and causal inference work together. *PS: Political Science & Politics*, 48(1):80–83.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42.

Gundersen, O. E., Gil, Y., and Aha, D. W. (2018). On reproducible AI: Towards reproducible research, open science, and digital scholarship in AI publications. *AI Magazine*, 39(3):56–68.

Ha, D. and Eck, D. (2018). A neural representation of sketch drawings. In *International Conference on Learning Representations*.

Hajian, S., Bonchi, F., and Castillo, C. (2016). Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2125–2126.

Hall, P. (2018). On the art and science of machine learning explanations. *arXiv preprint arXiv:1810.02909*.

Hardt, M., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323.

Harford, T. (2014). Big data: A big mistake? *Significance*, 11(5):14–19.

Harlow, L. L. and Oswald, F. L. (2016). Big data in psychology: Introduction to the special issue. *Psychological Methods*, 21(4):447.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hawking, S. (2018). *Brief Answers to the Big Questions*. Random House Publishing Group.

Hayashi, F. (2000). *Econometrics*. Princeton University Press.

Hempel, C. G. and Oppenheim, P. (1948). Studies in the logic of explanation. *Philosophy of science*, 15(2):135–175.

Henrique, B. M., Sobreiro, V. A., and Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, 124:226–251.

Herlocker, J. L., Konstan, J. A., and Riedl, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250.

Hildebrandt, M. (2019). Privacy as protection of the incomputable self: From agnostic to agonistic machine learning. *Theoretical Inquiries in Law*, 20(1):83–121.

Hindman, M. (2015). Building better models: Prediction, replication, and machine learning in the social sciences. *The ANNALS of the American Academy of Political and Social Science*, 659(1):48–62.

Hoffman, R. R., Mueller, S. T., Klein, G., and Litman, J. (2018). Metrics for explainable AI: Challenges and prospects. *arXiv preprint arXiv:1812.04608*.

Hsiao, W.-L., Katsman, I., Wu, C.-Y., Parikh, D., and Grauman, K. (2019). Fashion++: Minimal edits for outfit improvement. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5047–5056.

Huang, Q., Yamada, M., Tian, Y., Singh, D., Yin, D., and Chang, Y. (2020). Graphlime: Local interpretable model explanations for graph neural networks. *arXiv preprint arXiv:2001.06216*.

Hunter, L. et al. (1993). *Artificial intelligence and molecular biology*, volume 445. AAAI Press Menlo Park.

Hutson, M. (2018). Artificial intelligence faces reproducibility crisis. *Science*, 359(6377):725–726.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning*, volume 112. Springer.

Jordan, M. I. and Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260.

Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.

Kamiran, F. and Calders, T. (2012). Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33.

Kang, W.-C., Fang, C., Wang, Z., and McAuley, J. (2017). Visually-aware fashion recommendation and design with generative image models. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 207–216. IEEE.

Kim, B. (2015). *Interactive and interpretable machine learning models for human machine collaboration*. PhD thesis, Massachusetts Institute of Technology.

Kim, B. and Doshi-Velez, F. (2018). Introduction to interpretable machine learning. In *Proceedings of the CVPR 2018 Tutorial on Interpretable Machine Learning for Computer Vision, Salt Lake City, UT, USA*.

Kim, B., Khanna, R., and Koyejo, O. O. (2016). Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in neural information processing systems*, pages 2280–2288.

Kirsch, A. (2017). Explain to whom? putting the user in the center of explainable AI. In *Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML 2017*.

Koh, P. W. and Liang, P. (2017). Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894.

Kohavi, R. and Provost, F. (1998). Glossary of terms. *Journal of Machine Learning*, 30(2-3):271–274.

Kramer, R. M. and Cook, K. S. (2004). *Trust and distrust in organizations: Dilemmas and approaches*. Russell Sage Foundation.

Kramer, R. M. and Tyler, T. R. (1995). *Trust in organizations: Frontiers of theory and research*. Sage Publications.

Kreif, N. and Diaz-Ordaz, K. (2019). Machine learning in policy evaluation: New tools for causal inference. *arXiv preprint arXiv:1903.00402*.

Krening, S., Harrison, B., Feigh, K. M., Isbell, C. L., Riedl, M., and Thomaz, A. (2016). Learning from explanations using sentiment and advice in RL. *IEEE Transactions on Cognitive and Developmental Systems*, 9(1):44–55.

Krishnan, M. (2019). Against interpretability: A critical examination of the interpretability problem in machine learning. *Philosophy and Technology*, pages 1–16.

Kuhn, M. and Johnson, K. (2013). *Applied predictive modeling*, volume 26. Springer.

Kulesza, T., Burnett, M., Wong, W.-K., and Stumpf, S. (2015). Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th international conference on intelligent user interfaces*, pages 126–137.

Kumar, N., Rajagopalan, P., Pankajakshan, P., Bhattacharyya, A., Sanyal, S., Balachandran, J., and Waghmare, U. V. (2018). Machine learning constrained with dimensional analysis and scaling laws: simple, transferable, and interpretable models of materials from small datasets. *Chemistry of Materials*, 31(2):314–321.

Lazer, D., Pentland, A., Adamic, L., Aral, S., Barabási, A.-L., Brewer, D., Christakis, N., Contractor, N., Fowler, J., Gutmann, M., et al. (2009). Computational social science. *Science*, 323(5915):721–723.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436.

Lee, C. (2017). A GA-based optimisation model for big data analytics supporting anticipatory shipping in Retail 4.0. *International Journal of Production Research*, 55(2):593–605.

Lehr, D. and Ohm, P. (2017). Playing with the data: What legal scholars should learn about machine learning. *University of California Davis Law Review*, 51:653.

Lipton, Z. C. (2018). The mythos of model interpretability. *Queue*, 16(3):31–57.

Lipton, Z. C. and Steinhardt, J. (2019). Troubling trends in machine learning scholarship. *Queue*, 17(1):45–77.

Littman, M. L. (2015). Reinforcement learning improves behaviour from evaluative feedback. *Nature*, 521(7553):445.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Malhotra, R. (2015). A systematic review of machine learning techniques for software fault prediction. *Applied Soft Computing*, 27:504–518.

Marcus, G. (2018). Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*.

Martens, D., Vanthienen, J., Verbeke, W., and Baesens, B. (2011). Performance of classification models from a user perspective. *Decision Support Systems*, 51(4):782–793.

Math Vault (2019). The definitive glossary of higher mathematical jargon - algorithm. *On https://mathvault.ca/math-glossary/#algo*.

McKinsey (2018a). Notes from AI Frontier: AI Adoption Advances, but Foundational Barriers Remain. Discussion paper, McKinsey Global Institute.

McKinsey (2018b). Notes from AI Frontier: Modeling the Impact of AI on the World Economy. Discussion paper, McKinsey Global Institute.

Melton, A. W. (2014). *Categories of human learning*. Academic Press.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.

Mishra, S., Sturm, B. L., and Dixon, S. (2017). Local interpretable model-agnostic explanations for music content analysis. In *ISMIR*, pages 537–543.

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill Science.

Mitchell, T. M. (2006). *The discipline of machine learning*, volume 9. Carnegie Mellon University, School of Computer Science, Machine Learning.

Mittelstadt, B., Russell, C., and Wachter, S. (2019). Explaining explanations in AI. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 279–288.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.

Mohseni, S., Zarei, N., and Ragan, E. D. (2018). A survey of evaluation methods and measures for interpretable machine learning. *arXiv preprint arXiv:1811.11839*.

Molnar, C. (2020). *Interpretable machine learning*. Leanpub.

Mueller, S. T., Hoffman, R. R., Clancey, W., Emrey, A., and Klein, G. (2019). Explanation in human-AI systems: A literature meta-review, synopsis of key ideas and publications, and bibliography for explainable AI. *arXiv preprint arXiv:1902.01876*.

Mullainathan, S. and Spiess, J. (2017). Machine learning: An applied econometric approach. *Journal of Economic Perspectives*, 31(2):87–106.

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT press.

Nash, W. J., Sellers, T. L., Talbot, S. R., Cawthorn, A. J., and Ford, W. B. (1994). The population biology of abalone (haliotis species) in tasmania. i. blacklip abalone (h. rubra) from the north coast and islands of bass strait. *Sea Fisheries Division, Technical Report*, 48.

Nawi, N. M., Atomi, W. H., and Rehman, M. (2013). The effect of data pre-processing on optimized training of artificial neural networks. *Procedia Technology*, 11:32–39.

Neftci, E. O. and Averbeck, B. B. (2019). Reinforcement learning in artificial and biological systems. *Nature Machine Intelligence*, 1(3):133–143.

Nokelainen, P., Nevalainen, T., and Niemi, K. (2018). Mind or machine? Opportunities and limits of automation. In *The impact of digitalization in the workplace*, pages 13–24. Springer.

Nunes-Alves, C. (2016). Microbiome: microbiota-based nutrition plans. *Nature reviews. Microbiology*, 14(1):1.

Obermeyer, Z. and Emanuel, E. J. (2016). Predicting the future - big data, machine learning, and clinical medicine. *The New England journal of medicine*, 375(13):1216.

Obermeyer, Z., Powers, B., Vogeli, C., and Mullainathan, S. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453.

Oquendo, M., Baca-Garcia, E., Artes-Rodriguez, A., Perez-Cruz, F., Galfalvy, H., Blasco-Fontecilla, H., Madigan, D., and Duan, N. (2012). Machine learning and data mining: Strategies for hypothesis generation. *Molecular psychiatry*, 17(10):956–959.

Otte, C. (2013). Safe and interpretable machine learning: A methodological review. In *Computational intelligence in intelligent data analysis*, pages 111–122. Springer.

Patil, P., Peng, R. D., and Leek, J. (2016). A statistical definition for reproducibility and replicability. *BioRxiv*, page 066803.

Pearl, J. (2009). *Causality*. Cambridge university press.

Pedersen, T. L. and Benesty, M. (2018). lime: Local interpretable model-agnostic explanations. *R Package version 0.4*, 1.

Pitt, J. C. (1988). *Theories of explanation*. Oxford University Press.

Preece, A., Harborne, D., Braines, D., Tomsett, R., and Chakraborty, S. (2018). Stakeholders in explainable AI. *arXiv preprint arXiv:1810.00184*.

Prince, J., Sellers, T., Ford, W., and Talbot, S. (1988). Recruitment, growth, mortality and population structure in a southern australian population of haliotis rubra (mollusca: Gastropoda). *Marine biology*, 100(1):75–82.

Pu, P. and Chen, L. (2007). Trust-inspiring explanation interfaces for recommender systems. *Knowledge-Based Systems*, 20(6):542–556.

Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K. (2005). Surrogate-based analysis and optimization. *Progress in aerospace sciences*, 41(1):1–28.

Quinlan, J. R. (1987a). Generating production rules from decision trees. In *International Joint Conferences on Artificial Intelligence Organization: Proceedings of the 10th International joint conference on Artificial Intelligence - Volume 1*, volume 87, pages 304–307. Citeseer.

Quinlan, J. R. (1987b). Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234.

Raymond, P., Shoham, Y., Brynjolfsson, E., Clark, J., Etchemendy, J., Grosz, B., Lyons, T., Mynika, J., Mishra, S., and Niebles, J. (2019). The AI index 2019 annual report. Technical report, Human-Centered AI Institute: AI Index Steering Dommittee.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016a). Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016b). "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*, pages 1135–1144.

Robnik-Šikonja, M. and Bohanec, M. (2018). Perturbation-based explanations of prediction models. In *Human and machine learning*, pages 159–175. Springer.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.

Salmon, W. C. et al. (1989). Four decades of scientific explanation. *Scientific explanation*, 13:3–219.

Sanchis-Ojeda, R., Sibley, D., and Massimi, P. (2016). Detection of fashion trends and seasonal cycles through the analysis of implicit and explicit client feedback. In *KDD Fashion Workshop*.

Scotchmer, S. (1991). Standing on the shoulders of giants: Cumulative research and the patent law. *Journal of economic perspectives*, 5(1):29–41.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., and Dennison, D. (2015). Hidden technical debt in machine learning systems. In *Advances in neural information processing systems*, pages 2503–2511.

Selbst, A. D. and Powles, J. (2017). Meaningful information and the right to explanation. *International Data Privacy Law*, 7(4):233–242.

Shemtov, N. (2019). A study on inventorship in inventions involving AI activity. Technical report, European Patent Office.

Shi, S., Zhang, X., and Fan, W. (2020). A modified perturbed sampling method for local interpretable model-agnostic explanation. *arXiv preprint arXiv:2002.07434*.

Siau, K. and Wang, W. (2018). Building trust in artificial intelligence, machine learning, and robotics. *Cutter Business Technology Journal*, 31(2):47–53.

Silva, W., Fernandes, K., Cardoso, M. J., and Cardoso, J. S. (2018). Towards complementary explanations using deep neural networks. In *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*, pages 133–140. Springer.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.

Sinha, R. and Swearingen, K. (2002). The role of transparency in recommender systems. In *CHI'02 extended abstracts on Human factors in computing systems*, pages 830–831.

Sonnenburg, E. D. and Sonnenburg, J. L. (2015). Nutrition: A personal forecast. *Nature*, 528(7583):484.

Southwood, P. (2014). Blacklip abalone (haliotis rubra) at Mistaken Cape, Maria Island, Tasmania. *On https://en.wikipedia.org/wiki/Haliotis#/media/File:Haliotis_rubra_P2164176.JPG*.

Spiegel, J. R., McKenna, M. T., Lakshman, G. S., and Nordstrom, P. G. (2012). Method and system for anticipatory package shipping. Google Patents. US 8,615,473 B2.

Stock, J. H. and Watson, M. W. (2015). *Introduction to econometrics*.

Strevens, M. (2004). The causal and unification approaches to explanation unified—causally. *Noûs*, 38(1):154–176.

Surden, H. (2014). Machine learning and law. *Washington Law Review*, 89:87–115.

Suwajanakorn, S., Seitz, S. M., and Kemelmacher-Shlizerman, I. (2017). Synthesizing Obama: Learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):95.

Tan, S., Caruana, R., Hooker, G., and Lou, Y. (2018). Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 303–310.

Tarca, A. L., Carey, V. J., Chen, X.-w., Romero, R., and Drăghici, S. (2007). Machine learning and its applications to biology. *PLoS computational biology*, 3(6).

Teach, R. L. and Shortliffe, E. H. (1981). An analysis of physician attitudes regarding computer-based clinical consultation systems. *Computers and Biomedical Research*, 14(6):542–558.

Thagard, P. (1990). Philosophy and machine learning. *Canadian Journal of Philosophy*, 20(2):261–276.

Thomassey, S. and Zeng, X. (2018). *Artificial Intelligence for Fashion Industry in the Big Data Era*. Springer.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Tjoa, E. and Guan, C. (2019). A survey on explainable artificial intelligence (XAI): Towards medical XAI. *arXiv preprint arXiv:1907.07374*.

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

Toutenburg, H. and Heumann, C. (2008a). *Deskriptive Statistik: eine Einführung in Methoden und Anwendungen mit R und SPSS*. Springer-Verlag.

Toutenburg, H. and Heumann, C. (2008b). *Induktive Statistik: eine Einführung mit R und SPSS*. Springer-Verlag.

Tukey, J. W. (1977). *Exploratory data analysis*, volume 2. Reading, Mass.

US Congress (116th) (2019). Algorithmic accountability act of 2019 (s. 1108).

U.S. Department of Transportation (2020). Ensuring american leadership in automated vehicle technologies - automated vehicles 4.0. Technical report, National Science & Technology Council and the United States Department of Transportation.

Van Belle, V. and Lisboa, P. (2013). Research directions in interpretable machine learning models. In *European Symposium on Artificial Neural Networks*.

Van Fraassen, B. C. (1977). The pragmatics of explanation. *American Philosophical Quarterly*, 14(2):143–150.

Varshney, K. R. (2016). Engineering safety in machine learning. In *2016 Information Theory and Applications Workshop (ITA)*, pages 1–5. IEEE.

Vellido, A., Martín-Guerrero, J. D., and Lisboa, P. J. (2012). Making machine learning models interpretable. In *European Symposium on Artificial Neural Networks*, volume 12, pages 163–172. Citeseer.

Verbeke, W., Martens, D., Mues, C., and Baesens, B. (2011). Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert systems with applications*, 38(3):2354–2364.

Verma, S. and Rubin, J. (2018). Fairness definitions explained. In *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, pages 1–7. IEEE.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.

Visser, A. (1991). The formalization of interpretability. *Studia Logica*, pages 81–105.

von Schwartzenberg, R. J. and Turnbaugh, P. J. (2015). Siri, what should i eat? *Cell*, 163(5):1051–1052.

Voosen, P. (2017). How AI detectives are craking open the black box of deep learning. *Science*.

Vuković, M. et al. (1999). The principles of interpretability. *Notre Dame Journal of Formal Logic*, 40(2):227–235.

Wachter, S., Mittelstadt, B., and Floridi, L. (2017). Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, 7(2):76–99.

Waugh, S. (1995). Abalone Dataset UCI Machine Learning Repository https://archive.ics.uci.edu/ml/datasets/abalone. Irvine. *CA: University of California, School of Information and Computer Science [Dataset]*.

Weller, A. (2019). Transparency: Motivations and challenges. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 23–40. Springer.

Witten, I. H., Frank, E., and Hall, M. A. (2017). *Practical machine learning tools and techniques*. Morgan Kaufmann, 4th edition edition.

Wright, B. D. (1983). The economics of invention incentives: Patents, prizes, and research contracts. *The American Economic Review*, 73(4):691–707.

Xiang, L., Ma, H., Zhang, H., Zhang, Y., Ren, J., and Zhang, Q. (2019). Interpretable complex-valued neural networks for privacy protection. *arXiv preprint arXiv:1901.09546*.

Yanisky-Ravid, S. and Liu, X. J. (2017). When artificial intelligence systems produce inventions - the 3a era and an alternative model for patent law. *Cardozo Law Review*, 39:2215–2263.

Ye, L. R. and Johnson, P. E. (1995). The impact of explanation facilities on user acceptance of expert systems advice. *MIS Quarterly*, pages 157–172.

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. (2013). Learning fair representations. In *International Conference on Machine Learning*, pages 325–333.

Zhu, X. J. (2005). Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Ehrenwörtliche Erklärung

Ich, Philipp Knöpfle, versichere, dass ich die Masterarbeit selbständig und ohne unzulässige fremde Hilfe (siehe folgende Seite) angefertigt und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Ich bin damit einverstanden, dass die elektronische Version meiner Arbeit mit Hilfe einer Software zur Plagiatserkennung überprüft wird. Ich bin einverstanden, dass zum Zweck der Überprüfung von Prüfungstexten auf Plagiate mein übermittelter Text dauerhaft in der von Turnitin Netherlands BV ausschließlich für die Wirtschaftswissenschaftliche Fakultät der Universität Augsburg geführten Datenbank gespeichert wird.

Ich bestätige, dass ich die folgenden Richtlinien für den Umgang mit Literaturquellen und Daten verstanden und eingehalten habe:

"Die benutzte Literatur und sonstige Hilfsquellen sind vollständig anzugeben; wörtlich, nahezu wörtlich oder sinngemäß dem Schrifttum entnommene Stellen sind kenntlich zu machen. Es muss durchgängig und unmissverständlich erkennbar sein, was an fremdem geistigem Eigentum übernommen wurde. Zitierten Autoren dürfen keine Aussagen zugeschrieben werden, die diese nicht oder nicht in der wiedergegebenen Form gemacht haben. Internetquellen sind mit vollständiger Adresse und dem Tag des Zugriffs zu versehen. Zugrunde liegende Daten müssen inüberprüfbarer Weise dokumentiert werden. Bei gemeinschaftlichen Arbeiten ist der eigene Anteil des jeweiligen Autors deutlich zu machen. Darüber hinaus gelten die Vorgaben des vom betreuenden Lehrstuhl herausgegebenen 'Leitfaden zur Anfertigung von wissenschaftlichen Arbeiten'."

Ich habe den 'Leitfaden zur Anfertigung von wissenschaftlichen Arbeiten' vom betreuenden Lehrstuhl erhalten, ihn gelesen und verstanden.

Ich nehme zur Kenntnis, dass die Verletzung der hier genannten Richtlinien als versuchte Täuschung bzw. als Plagiat gewertet und mit Maßnahmen bis hin zur Exmatrikulation geahndet werden kann.

Ich versichere, dass die in Papierform abgegebene Arbeit identisch ist mit der elektronischen Fassung, die ich in die Plagiatserkennungssoftware hochgeladen bzw. meinem Betreuer übermittelt habe.

Augsburg, 31^{st} of July, 2020

_____

(Unterschrift)

Erläuterungen

Als unzulässige fremde Hilfe im Sinne dieser Plagiatserklärung gilt jede fremde Hilfe, die über die folgenden Hilfestellungen hinausgeht:

1. Beratung und Unterstützung durch den zuständigen Mitarbeiter am betreuenden Lehrstuhl.
2. Diskussion der Arbeit im Rahmen von Seminarveranstaltungen des betreuenden Lehrstuhls.
3. Ein Probelesen durch Kommilitonen oder andere Personen, die dem Autor ein mündliches Feedback geben und seine Thesen mit ihm diskutieren.
4. Ein Korrekturlesen durch Kommilitonen oder andere Personen oder auch Software, das sich auf Orthographie, Interpunktion und Grammatik beschränkt.
5. Hilfe und Unterstützung bei der technischen Erstellung der Arbeit (Satz, Layout, Druck und Bindung der Arbeit).