universität wien

MAGISTERARBEIT / MASTER'S THESIS

Titel der Magisterarbeit / Title of the Master's Thesis

„Interpretability of Black-Box-Models in Text Mining“

verfasst von / submitted by

Christoph Sattler, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Magister der Sozial- und Wirtschaftswissenschaften (Mag.rer.soc.oec.)

Wien, 2022 / Vienna 2022

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 951
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Magisterstudium Statistik
Betreut von / Supervisor:	Univ.-Prof. i.R. Dr. Wilfried Grossmann

Abstract - English

Text Mining and Sentiment Analysis in particular are increasingly relevant tasks that can be solved using ideas and methods from statistics and machine learning. As these predictive models can become quite complex and opaque, researchers have given increasing attention to explainability and interpretability of black-box models. This thesis combines theory and practice of both tasks. Theory is covered by an overview of text mining, particularly sentiment analysis, as well as by a motivation for discussing interpretability of black-box models and an introduction into various methods to provide either global or local explanations. As for the practical part, three classic machine learning models (Random Forest, XGBoost and Support Vector Machine with linear kernel) get trained to predict the binary sentiment of English-language movie reviews. Afterwards, each model's most noticeable false predictions are investigated further using locally interpretable model-agnostic explanations (LIME), generating explanations for each prediction. Thus, a compelling overview of the interconnectedness of text mining and interpretable black-box models is presented to the reader.

Abstract - Deutsch

Text Mining und das dazugehörige Subthema der Sentiment Analysis (Stimmungserkennung) sind zunehmend relevante Problemstellungen, die mittels Ideen und Methoden aus Statistik und Machine Learning behandelt werden können. Da die dafür verwendeten (statistischen) Prognosemodelle eine komplexe Struktur aufweisen können, wird in der Forschung zunehmende Aufmerksamkeit auf die Erklärbarkeit und Interpretierbarkeit ebenjener Modelle gelegt. Diese Magisterarbeit verbindet Theorie und Praxis aus beiden Themenbereichen. Der theoretische Hintergrund wird mittels eines

Überblicks über Text Mining und Sentiment Analysis sowie einer Einführung in Konzepte und Methoden der Interpretierbarkeit von “Black-Box Modellen” zur Erstellung von globalen oder objektspezifischen Erklärungen dargelegt. Im praktischen Teil werden drei klassische Machine Learning - Modelle trainiert, um die Polarität von englischsprachigen Filmkommentaren zu erkennen und vorherzusagen. Sämtliche Modelle konnten über 80% der Bewertungen korrekt klassifizieren. Die eindeutigsten Fehlklassifizierungen jedes Modells werden anschließend mittels lokal interpretierbarer modell-agnostischer Erklärungen (LIME) analysiert. Mit dieser Magisterarbeit soll ein Überblick über die Verflechtungen von Text Mining - Problemstellungen mit dem Bereich von interpretierbaren Black-Box Modellen präsentiert werden.

Danksagung

Zuallererst möchte ich mich bei meinen Eltern bedanken, die all das hier ermöglicht haben. Von Gutenstein nach Wien kann es ein weiter Weg sein, aber ihr habt mir immer das Gefühl gegeben, dass mir alle Möglichkeiten offen stehen. Mama, Papa, diese Arbeit ist für euch.

Entstanden wäre diese Magisterarbeit auch sicher nicht ohne Prof. Wilfried Grossmann, bei dem ich mich für seine geduldige Betreuung vielmals bedanken möchte.

Vermutlich wäre ich gar nicht bis zur Magisterarbeit gekommen, hätte ich im Laufe des Studiums nicht so viele großartige Menschen kennengelernt. Danke Marlene, Danke Stefan, Lukas, Natalie, Conny, die Liste könnte noch lang fortgesetzt werden. Ihr habt die schönen Stunden noch schöner gemacht, und die mühsamen Stunden sind mit und dank euch wesentlich schneller vorübergezogen.

Contents

1 Introduction

In 2007, Feldman and Sanger designated text mining as a "new and exciting" research field in the introduction of a textbook designated to present an overview of the subject (Feldman and Sanger, 2007). Fifteen years have passed since and it is hard to argue with their verdict of excitement. The rise of the digital age has increased the availability of text data originating from diverse sources, and the new data sources have been met with overwhelming curiosity for academic and commercial use alike.

However, as the use of complex machine learning models increases and reaches many areas, the need for interpretable predictions arises as well. If text mining is used to attribute the authorship of centuries-old documents (Mosteller and Wallace, 1964), this may be viewed as an "academic playground" where wrong predictions do little harm, except perhaps to the academic reputation of those responsible for the model. Used to automatically classify job applications and resumes (Angwin and Larson, 2016), the potential consequences and damages justify thorough inspection of the model. Accordingly, research on "explainable artificial intelligence" has increased in the last 20 years. Even though many techniques can only be applied on tabular data, there are approaches that can be used to interpret predictions on text data. One such approach using local surrogate models will be investigated by means of theory and practice in this thesis.

As its title suggests, the main focus of this thesis is on interpretability of black-box models. There is plenty of in-depth literature on every notion of text mining and detailed coverage of all aspects would be both redundant and out of scope. A concise summary of key aspects from text mining and an introduction into sentiment analysis is given in chapter 2. Chapter 3 contains theoretical background on interpretability in complex predictive models. The motivation and need for explainable predictions is highlighted, along with some basic definitions from both machine learning and social sciences. A major part of the chapter is dedicated to the introduction of various methods available to explain predictions of black-box models. Chapters 4 and 5 are based on practical research: Chapter 4 describes the results of a sentiment analysis using supervised machine learning techniques on a corpus of movie reviews. As these techniques are inherently complex and the "reasoning" behind both true and false predictions

of each model cannot be directly observed, Chapter 5 presents explanations obtained via locally interpretable model-agnostic explanations (LIME). The final chapter 6 provides a conclusion.

2 Text Mining

In the last 20 years text mining emerged as a new field that aims to combine tools and insights from statistics, natural language processing, information retrieval and linguistics (Feldman and Sanger, 2007). Witten (2004) identifies workshops at two conferences during the summer of 1999 as the "origins" of text mining: The *International Machine Learning Conference* and the *International Joint Conference on Artificial Intelligence*. However, researchers have been dealing with processing text, speech or natural language in general for decades. Bledsoe and Browning (1959) implemented a Bayesian system for text recognition, and Mosteller and Wallace (1964) used statistical methods (both Bayesian and non-Bayesian) to attribute authorship to twelve articles belonging to the "Federalist Papers", a series of essays published in 1787 and 1788 propagating the constitution of the United States of America. An extensive overview of further early advancements is presented by Jurafsky and Martin (2014).

The influence of classic natural language processing in text mining is weaker than the influence of techniques rooting in information retrieval (Witten, 2004). Witten's reasoning for this seemingly paradox situation is that natural language processing started off with examining automatic language translation - a complex problem using today's computational resources, let alone those available in the early 1950s. Moving on from these high ambitions, natural language processing researchers retreated into "toy worlds". By doing so, their approaches showed more success in various tasks but failed remarkably when they were given "real text". The separation from "real text" was not possible in other research fields such as compression schemes or practical information retrieval, leading to the development of techniques that are commonly used in text mining. (Witten, 2004)

2.1 Definitions

As text mining is an active and ongoing field of research, various definitions for the term exist. A broad definition of text mining being a *"knowledge-intensive process in which a user interacts with a document collection over time by using a suite of analysis tools"* is given by Feldman and Sanger (2007). According

to Sebastiani (2002), text mining is denoting *"all the tasks that, by analyzing large quantities of text and detecting usage patterns, try to extract probably useful (although only probably correct) information."* A brief definition of the field is given by Wikipedia as *"the process of deriving high-quality information from text"* (Wikipedia contributors, 2022), even though this would in theory also include human expert systems which are regularly not considered to be a part of text mining. The "analysis tools" mentioned in the listed definitions usually root in statistics and machine learning. Pattern detection means that given a large amount of training data that includes the desired information of interest, an algorithm "learns" the connections between input and output and is able to "reuse" the connection scheme on fresh data that does not include the information of interest. In the previously mentioned work of Mosteller and Wallace, the training data are documents with known authorship (this being the information of interest) and the algorithm learns to connect each author with a certain writing style. These connections between writing style and author are then used to assign each article with unknown authorship to its most probable author.

2.2 Text Mining and Data Mining

Text mining and data mining are closely related. In both areas preprocessing techniques are applied to transform given data, rendering it accessible for analytical tools. These tools can be considered as "pattern-discovery algorithms", where a high number of pattern types originates in Data Mining. Data and text mining also share "presentation-layer elements" like visualization tools, fostering comprehension of insights generated by the pattern-discovery algorithms (Feldman and Sanger, 2007).

However, there exist major differences originating in the heterogenous data types that are analyzed. In data mining, the input usually consists of rectangular tables, so preprocessing might consist of methods to standardize columns or to join various tables in order to create a single dataset containing all necessary information. The input is highly structured and consists of explicit features. In contrast, the typical input for text mining consists of a set of documents which are usually unstructured and certainly non-rectangular. Text mining does not operate on the input documents itself but on "feature-based representations" instead (Feldman and Sanger, 2007). Thus a major task in preprocessing text data is the identification and extraction of features that are considered representative for the underlying text. By doing so, the text input gets transformed into an "explicitly structured representation". Typical features represented within a document can include characters, words, terms and concepts. While characters and words can be extracted easily from a given document, terms (i.e. single words or multi-word phrases) and concepts require more complex extraction schemes and computational power. On the other hand, both terms and con-

cepts are able to condense the underlying document's substance at a higher level.

Documents usually contain a large quantity of possible inputs, so considerable effort has to be put towards identifying a (representable) subset of features. Selecting the subset's size is a non-trivial task. Choosing a small size of features might lead to biased characterization of a document, but the inclusion of more features gives rise to computational challenges. Given that the training dataset used to develop a text mining system usually comprises thousands of documents, feature sparsity is a common situation in text datasets. Most features within the dataset appear in a low number of instances (i.e. documents), leading to low support for many usual patterns (Feldman and Sanger, 2007).

A further source of variety is the state of the information which is to be extracted from the input data. In data mining its state is implicit so automatic techniques are needed to harvest the information. Text, however, is usually human-readable (given that it is written in a language comprehensible to the reader) so the information is explicitly available in most use cases. In fact "most authors go to great pains to make sure that they express themselves clearly and unambiguously" (Witten, 2004). Reviewers try their best to convince the reader of their opinion towards the object they evaluated, and neither scientific articles nor news messages hide the topic of their communication. Thus, text mining is deployed because it is often infeasible for humans to read every document in a given document collection (i.e. the input dataset) (Witten, 2004). However, there are use cases for text mining where the information to be extracted is not explicitly given. Automatic detection of "fake news", i.e. news that are "intentionally and verifably false and could mislead readers" (Shu et al., 2017) is an active research topic (Reis et al., 2019). As "fake news" tries to disguise itself as fact-based news, distinguishing is hard when using only the information explicitly available (i.e. the message text itself). Therefore, "auxiliary information" is necessary, making "fake news" detection a highly non-trivial research topic (Shu et al., 2017).

2.3 Sentiment Analysis

Sentiment analysis is a research area analyzing *"people's opinions, sentiments, appraisals, attitudes, and emotions toward entities and their attributes expressed in written text"* (Liu, 2015). Its main research question is to identify sentiment expressions as well as their polarity and strength. Thus, judgments towards specific subjects can be extracted from a given (usually large) collection of documents (Nasukawa and Yi, 2003). The field has originated from computer science, however researchers in social, political and management science as well as economics are engaged in sentiment analysis studies, as all these areas deal with opinions and sentiments by consumers, customers and the general public (Liu, 2015). It is also used in practice to track market opinions in finance or voters'

sentiments in politics (Ribeiro et al., 2016a).

Researchers have noted that the analysis of sentiments is a challenging topic compared to other aspects of text mining such as topic classification, since sentiments are often expressed without using obvious keywords (Pang et al., 2002). To tackle this problem, a multitude of sentiment analysis methods and techniques have been proposed. Mäntylä et al. (2018) count nearly 7000 papers on sentiment analysis as of 2017, with 99% being published after 2004. Ribeiro et al. (2016a) provide an overview over twenty four methods focusing on sentiment detection from reviews, social network postings, etc. The authors did omit most supervised (machine learning) methods requiring labeled training data as the training data is usually not published, preventing them from being a useful "off-the-shelf method". Another group of methods apart from those using machine learning approaches are *"lexical-based methods"* using predefined sentiment dictionaries (Ribeiro et al., 2016a). An extensive overview of current sentiment analysis research was performed by Mäntylä et al. (2018). In their article the twenty most cited papers per year as of 2017 are presented. Those papers focus on applications with online reviews and Twitter data as well as the presentation of newly developed tools and literature reviews. Notably, the authors state a change in research topics. Articles from 2013 and before did focus on product reviews and political scenarios such as the public sentiment before elections. More recent papers (2014 - 2016) frequently use social media platforms as their data sources or apply their analysis in the context of stock market predictions (Mäntylä et al., 2018).

In this thesis, a supervised approach using machine learning methods to classify sentiments on a document-level will be presented. This task is claimed to be the reason for the prominence of sentiment analysis research as well as the *"simplest sentiment analysis task"*, possessing a clear problem definition and similarities to "traditional text classification" (Liu, 2015). In a formal way, Liu (2015) formulates the task as *"Given an opinion document evaluating an entity, determine the overall sentiment of the opinion holder about the entity"*. Unsupervised classification of sentiments on a document-level has also been researched, e.g. using sentiment lexicons as presented by Kennedy and Inkpen (2006). However, this aspect will not be discussed in this thesis.

3 Interpretability

This section starts by defining interpretability in a machine learning context before displaying the need for explainable, interpretable predictive models. Afterwards, multiple methods to generate explanations for black-box models are introduced. Most of these models can be applied exclusively on tabular data, locally interpretable model-agnostic explanations (LIME) and selected others can also be applied to image and textual data.

3.1 Defining explanations and interpretability

A predictive system tries to optimize multiple facets. Contrary to a model's predictive performance, criteria as unbiasedness, causality or safety are hard to formalize and often unquantifiable. Instead, interpretability can be used as a fallback criterion to check these aspects. Interpretability in the machine learning context may be defined as the *"ability to explain or to present in understandable terms to a human*" (Doshi-Velez and Kim, 2017). Therefore, explainable predictions enable human users to verify whether a model fulfills any "auxiliary criteria" (Doshi-Velez and Kim, 2017). Explanations are a vastly researched topic in social sciences. A comprehensive overview of this area and its possible consequences for machine learning is given by Miller (2019). The article reviews sources from philosophy, psychology and cognitive science, promoting their inclusion into research on interpretable machine learning or explainable artificial intelligence. Miller defines explainable AI as a problem at the *"intersection of artificial intelligence, social science, and human-computer interaction"*. Similar to Biran and Cotton (2017), Miller (2019) further defines interpretability of a model as *"the degree to which an observer can understand the cause of a decision"*. The observer's understanding can be facilitated by providing explanations, which are *"assignments of causal responsibility"* (Josephson and Josephson, 1994) that exist within a (social) context depending on the observer or explainee. The context means that given an event the recipient (i.e. the explainee) might care about a subset of possible causes or features. The explainer might further select a smaller subset of reported causes, and lastly explainer and explainee might have any kind of interaction about this explanation (Miller,

2019).

3.2 The case for interpretability

The widespread usage of machine learning models has increased the need for interpretability. Even though pre-known biases can be prevented by explicitly formulating a problem, this does not guarantee fair and unbiased predictions as there might be data patterns (i.e. underlying biases) that only emerge after thorough examination (Doshi-Velez and Kim, 2017). There are numerous examples of "black-box" models including unwanted and unforeseen biases: In 2016 it was demonstrated that COMPAS ("Correctional Offender Management Profiling for Alternative Sanctions"), an algorithm used in several states of the U.S. to predict a defendant's future crime risk was racially biased. The accuracy of predicted recidivism was 61%, however *"blacks are almost twice as likely as whites to be labelled a higher risk but not actually reoffend"*(Angwin and Larson, 2016). Amazon used a machine learning model to review and score job applications and resumes, but decided to abondon it because the model *"was not rating candidates for software developer jobs and other technical posts in a gender-neutral way"* (Dastin, 2018). An overview about these and various other occasions where non-interpretable models have lead to discrimination is given by Zuiderveen Borgesius (2018). Reported examples include online advertising biases where job advertisements were only shown to men, price discrimination, translation tools that reflect and promote gender inequality, discriminatory effects at image search systems and more. In general, blindly trusting a model can lead to various problems (Dzindolet et al., 2003).

Interpretability is an important tool in situations where no appropriate test data is available. Machine learning techniques that are used to build a predictive model usually assume identical distribution of training and test data. A different joint distribution of input and output data between training and test data (*"dataset shift"*) (Quiñonero-Candela et al., 2008) is a further situation highlighting the need for interpretable machine learning. Ribeiro et al. (2016c) present an example of a classifier that was trained on a biased dataset. Potential reasons for a change in data distribution include simple covariate shift (i.e. change in the distribution of input data), prior probability shift (i.e. change in the distribution of output data), sample selection bias, discarding data in a heavily imbalanced setting ("dataset shift by design"), domain shift (i.e. changes in measurement) and source component shift (i.e. changes in strength of contributing variables) (Quiñonero-Candela et al., 2008). Interpretable machine learning methods are often used to understand the inherent process that forms the relationship between input and output data. Therefore it is important to realize if a dataset shift occurs, since the model will not be able to generalize the underlying process from training to test data. Many problems in science are solved by complex, uninterpretable black-box models. In these situations in-

terpretability is necessary to understand the underlying structure as the model replaces the input data as the source of knowledge (Molnar, 2019).

Machine learning models are often used to solve complex tasks, where it is impossible to gather data for all possible situations which a model may encounter "in production" (e.g. autonomous driving systems in cars) (Doshi-Velez and Kim, 2017). Since it is not possible to detect all undesirable outputs, it is crucial to be sure that the underlying structure learned by the model is error-free. Interpretable predictions are a possibility to either confirm the learned structure or to detect flaws or edge cases. Molnar (2019) gives an example of a self-driving car system built to detect cyclists. If an explanation shows that cyclists are recognized by the bicycle's wheels, then one can further test if bicycles with partially covered wheels (e.g. by side bags) are detected as well, so the explanation offers additional insights that support increasing a model's safety and applicability. This is closely related to the topic of reliability or robustness, as minor changes in the input data (e.g. aforementioned side bags) should typically not result in drastic changes in the output.

"Data leakage" is another potential modeling pitfall that can be avoided if the predictions can be interpreted. Leakage means introducing information that "should not be legitimately available" to use when creating a model (Kaufman et al., 2011). The phenomenon usually happens unintentionally and leads to unjustifiably high predictive power on training data. Interpretable predictions clearly support the discovery of leakage, since the impact of each available predictor on a prediction can be traced. Therefore it can be assessed if the model picks up on causal relationships only. Kaufman et al. (2011) review two discussions of a medical dataset that included a patient ID variable with "tremendous and unexpected predictive power". Blindly trusting a black-box model that uses the ID could lead to considerable disappointment after applying the model, as IDs usually do not contain any medical information of interest.

In practice it is common to develop multiple models, assess their predictive performance on a given dataset and then select one model to deploy. Ideally, the model assessment is based on those metrics that are relevant for the use case (e.g. customer engagement). However, it is often impossible or infeasible to compute such metrics. This is another case for interpretable predictions, as domain knowledge about the influence of a model's behaviour on e.g. customer engagement can be used to assess those models whose predictions can be explained. Typically available metrics such as accuracy, sensitivity or specificity can be misleading. They may favor models with high predictive power on a given training dataset but low power on a "real-world" dataset on which it was not trained. An example is shown by Ribeiro et al. (2016c).

If sensitive or personal data is used, having a model that can be interpreted may ensure the protection of this data (Molnar, 2019).

The listed arguments supporting the need for interpretable models have been mostly concerned with the development of models. However, being able to

clearly explain and interpret predictions is also useful to empower the model's usage. As Ribeiro et al. (2016c) state: *"if the users do not trust a model or a prediction, they will not use it"*. Similarly, Dzindolet et al. (2003) note that low or lacking trust can cause disuse of a model. Experiments conducted by Ribeiro et al. (2016c) show that supporting predictions with *"intelligible explanations"* increases the persuasive power of a model. Given that the audience of a machine learning algorithm usually possesses domain knowledge, consensus between knowledge and the model's explanations leads to increased trust in the predictions, differences or unexpected explanations might lead to further examination of the model. Social experiments by Dzindolet et al. (2003) have shown that false predictions lead human operators to distrust a predictive model. However, explaining a model's potential shortcomings or sources of error increased participants' trust regardless of the model's actual accuracy. Ribeiro et al. (2016c) differentiate between trust in a prediction and trust in a model itself. A prediction gets trusted if a user takes some action based on it, whereas a model is trusted if a user "behaves in reasonable ways if (the model gets) deployed". However, both concepts directly relate to users' understanding of a model and its behaviour. Trust in a model can essentially be seen as confidence that a model is consistent on training and real-world data, consistency being measured by any metric of interest. Trusting a prediction is crucial if a model is used to take decisions with relevant consequences such as medical diagnosis or in self-driving cars.

However, there are also situations where models do not need to be interpretable: All of the aforementioned arguments presume some sort of relevance. If a model has no significant real-world impact (e.g. a recommender system for travel destinations or birthday presents that is not tied to a business), it is not necessary to have explainable predictions, since there are no consequences of wrong or biased predictions (Molnar, 2019). If an application is well studied, there is no need to extract further knowledge from a model and the model's suitability to the task is already proven, then a black-box model can be used without delving into the underlying causal structure. Automated address extraction from photographs of envelopes is an example for this scenario (Molnar, 2019).

Furthermore, there are scenarios where a model is used in a situation demanding explainable predictions to check for causality, possible bias, etc. yet these explanations cannot be shown to the model's user. This is the result of a *"mismatch between the goals of the creator and the user of a model"* (Molnar, 2019). An example application is credit scoring - the bank's goal is to identify applicants who are likely to return the loan, the applicant's goal is to receive a loan. If (a) the explanations behind the scoring engine's decision are shown to the applicant and (b) the model's inputs are proxies for but not identical to a causal feature (e.g. use the applicant's number of credit cards instead of his actual debt level), then the user might deceive the system (Molnar, 2019). Even though the explanations cannot be supplied to the user, they are crucial for the model's creator (in the example: a financial institution) to ensure the model's quality with regard to the aforementioned aspects.

3.3 Achieving interpretability in machine learning

If the model itself is interpretable ("white-box model"), no further considerations need to be made towards explaining its predictions. Decision trees, linear and logistic regression are the most commonly used intrinsically interpretable models (Molnar, 2019). Decision rules, Bayesian rule lists and sparse models such as supersparse linear integer models or mind-the-gap models belong to this category, yet the interpretability of a model is directly related to its audience, so not every model architecture is suitable for every usage (Ribeiro et al., 2016c). Intrinsically interpretable models typically have simple structures, as humans need to be able to comprehend the internals in their entirety. Interpretable decision trees have few splits, linear models include a low number of features, etc. (Molnar, 2019). For a comprehensive list of inherently interpretable models see (Biran and Cotton, 2017). To gain further insights, methods analyzing the trained model can be applied ("post hoc interpretation") (Molnar, 2019). Post hoc interpretation methods are also available for non-intrinsically interpretable ("black-box") models.

Non-intrinsically interpretable models can be explained using methods to analyze the trained model ("post hoc") (Molnar, 2019). Machine learning systems/models rise in complexity (Doshi-Velez and Kim, 2017) and the predictions of machine learning models often outperform traditional models (Fernández-Delgado et al., 2014) (Molnar et al., 2020). This promotes the use of post hoc interpretation methods to gain insights into models with high predictive power but no inherent interpretability ("black-box models") (Molnar, 2019). Post hoc explanation methods usually ignore the specifics of the original model and can be used to explain predictions from any model, interpretable or not. Separating explanations from the model ("model agnostic" interpretability) has multiple benefits: For a given task, any model can be used regardless of its complexity (model flexibility). Explanations can be tailored to meet users' needs (explanation flexibility) and even use different (i.e. more interpretable) feature representation as the original model (representation flexibility) (Ribeiro et al., 2016b). If the underlying model is complex, local explanations may not be sufficient to reach a global understanding, local explanations of flexible models may even be seemingly inconsistent (Ribeiro et al., 2016b).

Model agnostic methods for interpretability can be classified according to criteria such as the result of the interpretation method (e.g. feature summary statistic, feature summary visualization, or representative data points) or their scopes (Molnar, 2019). Some interpretation methods explain the entire model ("global"), others restrict themselves to individual predictions ("local"). This taxonomy is used to classify and present selected methods in the following chapters.

3.4 Global model-agnostic interpretation methods

Global methods describe a model's average behaviour. This supports the debugging process of a model as well as the understanding of general mechanisms in the data (Molnar, 2019).

3.4.1 Partial Dependence Plots

Partial dependence plots (PDP) were first introduced by Friedman (2001). Given any model, a PDP visualizes the marginal effect of one or two predictors on the model's prediction. In other words, a PDP graphically displays the "*change in the average predicted value as specified feature(s) vary over their marginal distribution*" (Goldstein et al., 2015). In a regression setting with only numerical features, the partial dependence function of the feature set S is defined as:

$$\begin{aligned} f_{x_S,PDP}(x_S) &= E_{X_C}\left[\hat{f}\left(x_S, X_C\right)\right] \\ &= \int_{X_C} \hat{f}\left(x_S, X_C\right) \mathbb{P}\left(X_C\right) dX_C \end{aligned}$$

The partial dependence function is plotted for features X_S with values x_S, X_C denotes all other features in the underlying prediction model $\hat{f}$. S and C are disjunct sets of features/predictors, S usually contains one or two elements due to constraints in visualization. The output of $\hat{f}$ gets marginalized over the distribution of all variables in C (i.e. X_C) such that the partial dependence function is only dependent on features within set S. Therefore the relationship between the model's predictions and x_S can be calculated and visualized. $\hat{f}_{x_S,PDP}$ can be estimated using the Monte Carlo method, with $x_C^{(i)}$ being the true value of feature C at observation i and n being the number of observations in the dataset:

$$\hat{f}_{x_S,PDP}(x_S) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}\left(x_S, x_C^{(i)}\right)$$

Independence between feature sets C and S is a necessary condition. Models for classification that output class probabilities can be explained by a PDP displaying one line per class. Categorical features can be included by replacing a feature with the desired category on all observations (Molnar, 2019).

Given that they are a graphical procedure, partial dependence plots can show a maximum of two features. A drawback and a reason to be cautious when using PDP is the strong assumption of independence between the features X_S and X_C. This assumption may result in the creation of unlikely datapoints, holding back the entire explanation (Molnar, 2019). Given that a PDP displays only

one line (i.e. the "average marginal effect"), heterogeneous effects within the data might get lost. The line might be horizontal even though there are clear dependencies between features as they might cancel each other out (Molnar, 2019). The individual conditional expectation curves seek to solve this problem and uncover heterogeneity.

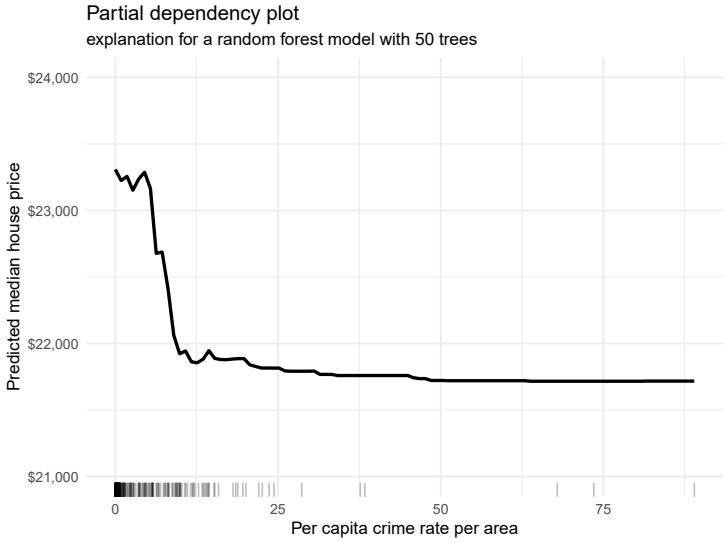

Figure 3.1: Partial Dependency Plot for a numeric predictor

Figure 3.1 shows a PDP explaining the predictions of a random forest model with 50 trees, trained on the Boston housing prices dataset published by Harrison and Rubinfeld (1978). The target variable is the median value of owner-occupied homes per area, towns around Boston, Massachusetts, USA are included in the dataset. The data further contains information on each area's ratio of pupils to teachers, crime rate, proportion of non-retail business acres, average number of rooms per building, nitric oxides contentration, etc. In total there are thirteen numeric predictors available. The PDP visualizes the random forest's predicted median house price based on the area's crime rate. The predictions remain unchanged for very low crime rates, rapidly decrease until a threshold around 10 and then remain practically unchanged again. The PDP line is extended by a rug plot on the horizontal axis used to display the predictor's distribution within the training data, therefore supporting the explanation by showing regions with few training data. In the dataset, the distribution of crime rates per area is clearly skewed. There are few areas with values above 25, making the PDP less reliable for areas with high crime rates .

3.4.2 Marginal Plots

A major drawback of partial dependency plots is their underlying assumption of independent features X_S and X_C (Molnar, 2019). For the calculation of PDP values in the example above, the crime rate is replaced in all observations. Since the crime rate in an area is quite likely not independent of other features, this can lead to the generation of "unreasonable" data points which are used as an input in the model's prediction over which the PDP averages. A possibility to remove unlikely data points is to use the conditional distribution instead of the marginal distribution. In the example this would mean to average over predictions of similar data instances. This procedure is called "M-Plots" or "marginal plots" (Apley and Zhu, 2020). A marginal plot of the set of features S is a plot of the function

$$\begin{aligned} f_{x_S,M}(x_S) &= E_{X_C|X_S}\left[\hat{f}\left(X_S, X_C\right) \mid X_S = x_s\right] \\ &= \int_{X_C} \hat{f}\left(x_S, X_C\right) d\mathbb{P}\left(X_C \mid X_S = x_S\right) \end{aligned}$$

versus x_S. Typically, the set S contains at most two features. The function can be estimated by

$$\hat{f}_{x_S,\,M}(x_S) = \frac{1}{n(x_S)} \sum_{i \in N(x_S)} f\left(x_S, x_C^{(i)}\right)$$

with $N(x_S) \subset \{1, 2, ..., n\}$ being the subset of row indices i for which $x_S^{(i)}$ lie within a small neighbourhood of x_S and $n(x_S)$ being the number of observations within this neighbourhood. While omitting unreasonable data points, marginal plots still suffer from omitted variable bias (Apley and Zhu, 2020). Again in terms of the example introduced in 3.4.1: If the median house price depends on the crime rate and some other predictor, the curve $f_{x_S,\,M}(x_S)$ will reflect both effects. If the crime rate is correlated to the nitric oxides concentration but only the latter has an effect on the predicted median house price, the marginal plot for the crime rate would still display a direct relation between crime rate and predicted median house price. Marginal plots *"mix the effect of a feature with the effects of all correlated features"* (Molnar, 2019).

3.4.3 Accumulated Local Effects (ALE) Plots

Accumulated Local Effects (ALE) plots strive to assess main and interaction effects of (black-box) models, avoiding the problems of PDP and marginal plots (Apley and Zhu, 2020). Like marginal plots their function is based on the conditional distribution of the predictors. However, instead of calculating averages of predictions, ALE plots calculate and visualize their differences.

Effects of features correlated to those of interest are blocked by the use of differences, therefore resulting in a feature's "pure effect". Figure 3.2, which is

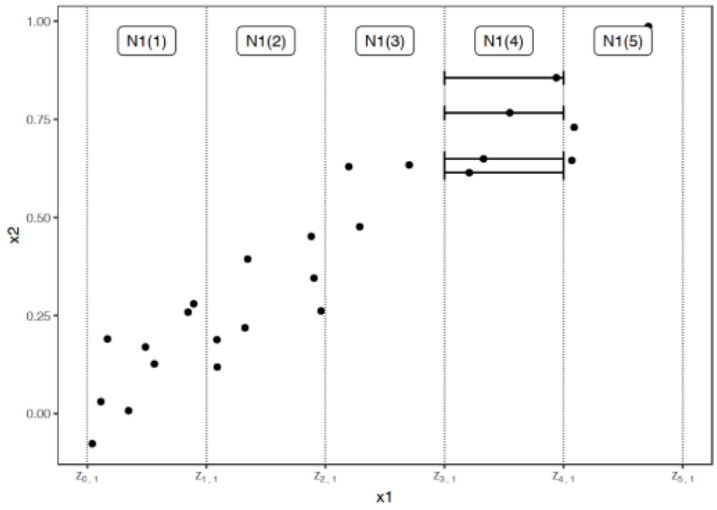

Figure 3.2: Graphical intuition for the calculation of accumulated local effects (ALE) (Molnar, 2019)

presented by Molnar (2019), gives graphical intuition on the calculation of ALE in a two-dimensional setting for feature x_1 that is correlated to feature x_2. The feature space of x_1 is divided into a fixed number of intervals (here: five). The vertical lines represent quantiles of the empirical distribution of x_1 and borders of the intervals. For each data instance within an interval, the underlying model takes two predictions: one with the interval's upper limit replacing the true value of feature x_1, one with the interval's lower limit (horizontal lines). For the ALE curve, the differences between an observation's two "new" predictions are accumulated and centered. The differences can be written as a partial derivative (Apley and Zhu, 2020) :

$$f^S(x_S, x_C) = \frac{\partial f\left(x_S, x_C\right)}{\partial x_S}$$

The ALE main effect of X_S is

$$\begin{aligned} f_{x_S,ALE}\left(x_S\right) &= \int_{z_{0,1}}^{x_S} E_{X_C|X_S}\left[f^S\left(X_s, X_c\right) \mid X_S = z_S\right] dz_S - \text{constant} \\ &= \int_{z_{0,1}}^{x_S} \int_{x_C} f^S\left(z_s, x_c\right) \mathbb{P}\left(x_C \mid z_S\right) dx_C dz_S - \text{constant} \end{aligned}$$

Subtracting a constant sets the average effect across the data to zero. The integral over z is to accumulate the local partial derivatives over the range of features in set S (Molnar, 2019). $z_{0,1}$ is some value near the lower bound of the effective support of $p_S(\cdot)$, which is the marginal density of the features x_S.

Details on the calculation and estimation of Accumulated Local Effects are given by Apley and Zhu (2020).

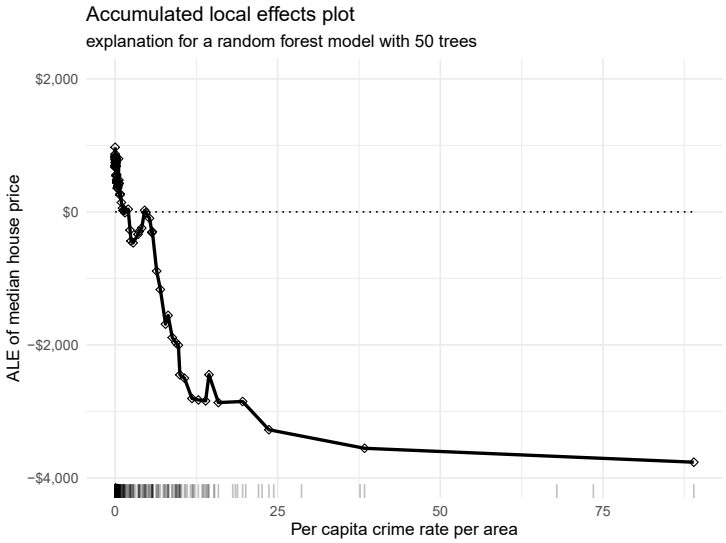

Figure 3.3: Accumulated local effects plot for a numeric predictor

Figure 3.3 shows the ALE plot in the same scenario as figure 3.1. The ALE values are positive only for very low crime rates (which appear very frequently as represented by the rug plot above the x-axis). They can be read as the main effect that a feature has at a certain value when comparing it to the average prediction of the data (Molnar, 2019). The diamonds along the curve signal the crime rate's empirical quantiles. These quantiles were used to define intervals, resulting in a very dense grid for low crime rates. In the scenario of Figure 3.3, the ALE estimate is close to 1000 at crime rate 0, increasing the median house price by 1000$ compared to the average prediction. The random forest's predictions for a per capita crime rate between 1.5 and 2 result in an ALE estimate close to 0, signaling that this might be a "baseline crime rate" that does neither add nor subtract to an area's median house prices.

The interpretation of ALE values for categorical predictors is similar. Figure 3.4 shows another ALE plot, however the feature "relation to river" is analyzed instead of an area's crime rate. The feature has two possible values, an area can either bound a river or not. The result suggests that any area bounding the river has its median house price increased by $950 compared to the average predicted median house price. Note that due to the accumulation applied by the ALE method, features are obligated to have an order. This order is clear for numeric features, however multiple possibilities exist for categorical features. An approach used by Molnar (2019) is to calculate similarities between categories, based on each observation's other features. In the scenario of Figure 3.4 this

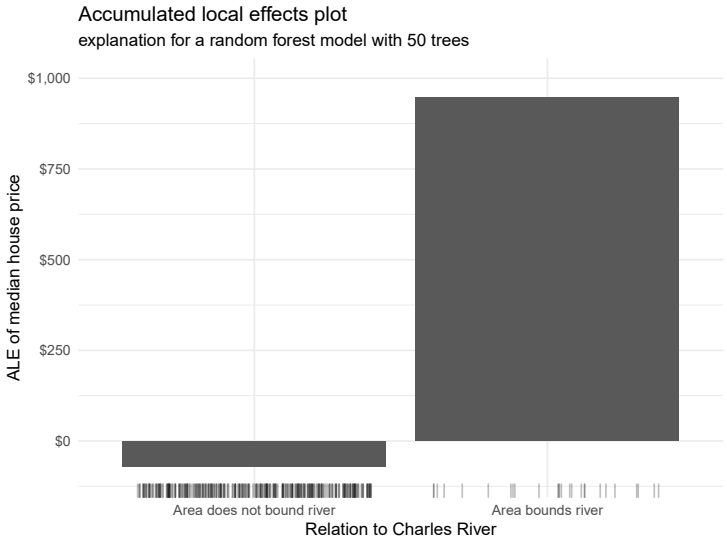

Figure 3.4: ALE plot for a categorical predictor

means comparing those areas bonding a river and those that do not bound a river by their other characteristics such as crime rate, ratio of pupils to teachers, property-tax rate etc. The categories are then ordered by these similarities. Note that different orders result in different ALE values, potentially changing interpretation.

ALE plots can handle correlated features, however any feature having strong correlation should not be analyzed by themselves but combined with other feature, since both usually change together. For ALE plots with two numerical features see (Apley and Zhu, 2020) and (Molnar, 2019).

3.4.4 Further global interpretation methods

The H-statistic (Friedman and Popescu, 2008) is a (computationally expensive) method to gain insights into the interactions of features present in any given model. This method assumes independence among predictors to calculate the strength of interactions. The nature of these interactions can then be assessed via PDP or ICE plots (Molnar, 2019). An approach based on functional decomposition and therefore similar to ALE plots is the generalized functional ANOVA (Hooker, 2007). This method strives to express a prediction function as a sum of individual and interaction effects and can be applied for dependent predictors.

A further way to gain insights into a black-box model is to create a new dataset by permutating the values of a feature and then to measure the model accuracy

on the new dataset. Permutating highly influential predictors should increase the prediction error. The permutation furthermore means that no meaningful interaction terms are present in the new dataset, so interaction is taken into account when calculating permutation feature importance. A model-agnostic way following this idea is to measure the "model reliance" (Fisher et al., 2019) of model f on a predictor X_S by calculating

$$MR_{X_S}(f) = \frac{\text{Expected loss of } f \text{ after permutating } X_S}{\text{Expected loss of } f \text{ without permutating } X_S}$$

Model reliance results in a single parameter per predictor. Since the ratio of errors is used, the exact values of model reliance on a predictor can be compared across different models fitted on the same dataset. The use of the model error for calculating model reliance means that no insights about the "direction" of any effect can be reached, only the mere presence or absence of said effect. If the predictors are correlated, permutation can result in unreasonable new observations similar to the situation for partial dependency plots, therefore adding bias (Molnar, 2019).

A straightforward way to gain insights on a black-box model is to fit a global surrogate model, i.e. an interpretable model that uses the same predictors but has the black-box model's predictions set as its target values. The quality of this explanation method is directly related to the accuracy of the original model. If the black-box model has low predictive power, an interpretable model might be able to explain the original predictions, however this insights have low relevance due to the difference between predictions and true values. The quality of approximation by the interpretable model is usually measured via R-squared, however there is no clear threshold signalling high or (too) low confidence in the approximation. Furthermore, interpretable models might approximate the black-box prediction function well for a subset of data but poor for another subset (Molnar, 2019). An alternative approach to avoid this potential pitfall is to use local surrogate models instead. An explanation method using local surrogate models is discussed in chapter 3.5.3.

Instead of directly explaining the model in its entirety, the concept of "Prototypes" and "Criticisms" (Kim et al., 2016) suggest explanation via selected examples. Prototypes are observations that are representative for the training data, Criticisms (or: criticism samples) are observations poorly represented by said Prototypes. These selected observations can either be used to gain an overview over the dataset, to create an interpretable model or to add interpretability to an existing (black-box) model (Kim et al., 2016)(Molnar, 2019). Observations can be selected via using MMD-critic (Kim et al., 2016) or k-medoids (Kaufman and Rousseeuw, 1987).

3.5 Local model-agnostic interpretation methods

Local interpretation methods explain black-box models by investigating individual predictions (Molnar, 2019). Contrary to global methods, the same value of a given predictor may have different consequences for different observations.

3.5.1 Individual Conditional Expectation (ICE) Plots

The underlying concept of individual conditional expectation (ICE) plots is very similar to the previously introduced partial dependence plots. A PDP is a visualization of the marginal effect of a set of predictors (usually one or two) on a predictor's response. ICE plots disaggregate PDPs and plot a curve for each observation, thus allowing easier identification of heterogenous relationships (Goldstein et al., 2015). Given N observations, an ICE plot shows N individual curves, while the PDP curve is the average of those ICE curves. Using the same notation as in previous chapters, ICE plots display the curve $f^{(i)}_{x_S,\, ICE}$ against $x^{(i)}_S$ for each instance in $\{\left(x^{(i)}_S, x^{(i)}_C\right)\}^N_{i=1}$. The remaining features $x^{(i)}_C$ remain fixed (Molnar, 2019). Heterogeneity can be represented with ICE curves, but some downsides for this visualization remain: If any of the predictors in S is correlated with another predictor, some areas within the visualization may be invalid as there are no data points supporting the perturbed training data. This is a problem that ICE plots share with partial dependence plots. A further downside is that visualizing curves for a high number of observations may lead to visual overload for the user. A potential solution for visual overload is to select a sample of observations and plot their ICE curves only, another solution is to use centered ICE (c-ICE) plots (Goldstein et al., 2015). Using c-ICE plots also adds clarity when the curves' intercepts have high variation that potentially hides heterogeneity. Given an ICE curve $f^{(i)}_{x_S,\, ICE}$ for each observation i, the corresponding c-ICE curve is

$$f^{(i)}_{x_S,\,\text{c-ICE}} = f^{(i)}_{x_S,\, ICE} - \mathbf{1} f\left(x^*, x_{C_i}\right)$$

with x^* being a base point within the range of x_S, usually the minimum or maximum and $\mathbf{1}$ being a vector of 1's of the appropriate dimension. If x^* is chosen to be the minimum value of x_S then all c-ICE curves start at 0 and the combined effect of x_S of f with fixed x_C can be displayed (Goldstein et al., 2015).

Figure 3.5 shows an ICE plot and the PDP curve for the same scenario as figures 3.1 and 3.3. A random forest trained on the Boston Housing dataset was tasked to predict the median house price per area, each area's crime rate per capita was changed in order to estimate the predictor's effect on the total predicted value. The PDP is visualized as an orange line which represents the average of

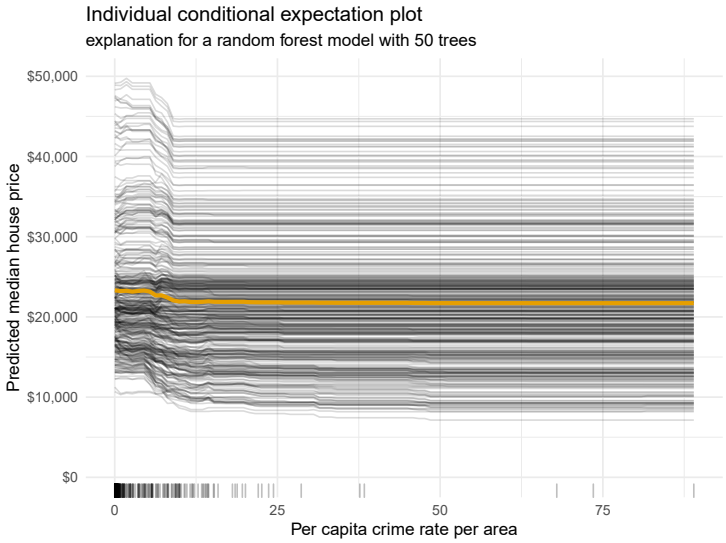

Figure 3.5: ICE plot for a numeric predictor, PDP curve included

all ICE curves. There appears to be a clear trend of decreasing median house prices for increasing crime rates from 0 to 10, higher crime rates do not have any effect. Since the intercepts vary between areas, the curves are stacked and it is difficult to spot any heterogeneity for crime rates between 0 and 10. Since there are few observations (i.e. areas) with crime rates above 25, the predictions in this area have low validity.

Figure 3.6 shows the centered ICE plot for the same scenario as figures 3.1, 3.3 and 3.5. The predicted values were centered at crime rate 0, so each curve shows the difference of its area's predicted median house price at a given crime rate to the predicted median house price at crime rate 0. The orange line represents the centered PDP curve. Centering the curves highlights the presence of heterogeneity for low crime rates, which was not clearly visible in Figure 3.5 (ICE plot) nor Figure 3.1 (PDP). Some areas' predicted values increase when the crime rate increases from 0 to 5, some remain unchanged, others decrease instantly. Furthermore it becomes apparent that the crime rate's effect is not equally strong on all observations. For some areas, the predicted median house price decreases by more than $5,000, while the PDP curve shows an average decline of only $2,000. The prediction for some areas even increases for crime rates higher than 0. Diverging ICE and c-ICE curves (as seen in this scenario) indicate interactions between the displayed feature (i.e. the per capita crime rate per area) and other features. Parallel ICE curves indicate additive effects x_C and x_S. If the remaining predictors x_C have no influence on the model's prediction, the ICE curves are identical (Goldstein et al., 2015). If the variable whose influence on the model gets explained via ICE is categorical, the

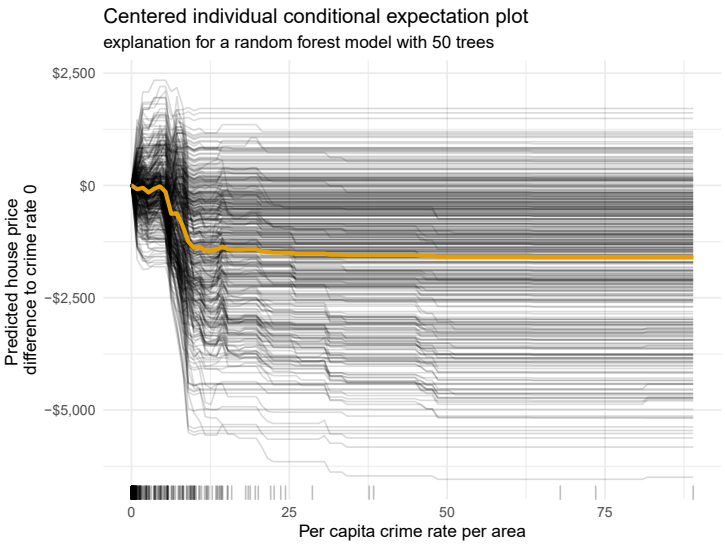

Figure 3.6: Centered ICE plot for a numeric predictor, PDP curve included

proposed visualization are boxplots instead of curves. Figure 3.7 shows such an example. The median house price per area is plotted for the same scenario as in figure 3.4, resulting in a boxplot per category. Even though the median of predicted median house prices is higher when a given area bounds Charles River, there are observations where the state "area does not bound river" results in a very high predicted value. It is difficult to see a more precise effect of the categorical predictor since the predictions belonging to the same observation (in this scenario: the same area) are not grouped as they are when using ICE plots for a numerical predictor (i.e. by displaying a single curve per observation).

ICE plots are mainly used to display the effects of a single predictor, however it is also possible to include information on a second predictor x_k by colouring the curve of prediction $\hat{f}(x^{(i)})$ according to $x_k^{(i)}$. This approach is feasible for both numeric and categorical predictors x_k (Goldstein et al., 2015). However, adding colours to an already crowded visualization can lead to visual overload. A possible solution is to only plot a sample of all observations.

Figure 3.8 shows an ICE plot using colour to include information on the area's location in addition to the effect of the crime rate. The colour coding adds the information that most areas do not bound a river. The effect that a change in crime rate has on predicted median house prices does not seem to vary strongly between areas bounding a river and those that do not.

Interaction effects and heterogeneity can further be assessed via derivative ICE plots ("d-ICE plots") (Goldstein et al., 2015). This procedure plots an estimate of the partial derivative of the estimated response function $\frac{\partial \hat{f}(x)}{\partial x_S}$ against either

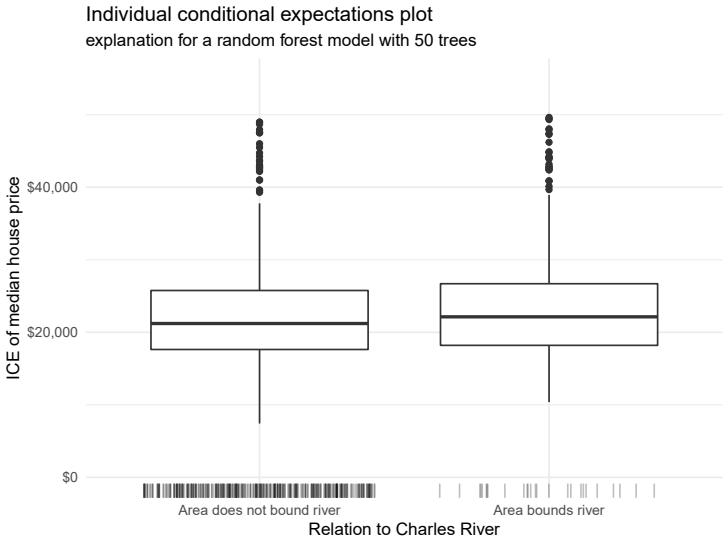

Figure 3.7: ICE plot for a categorical predictor

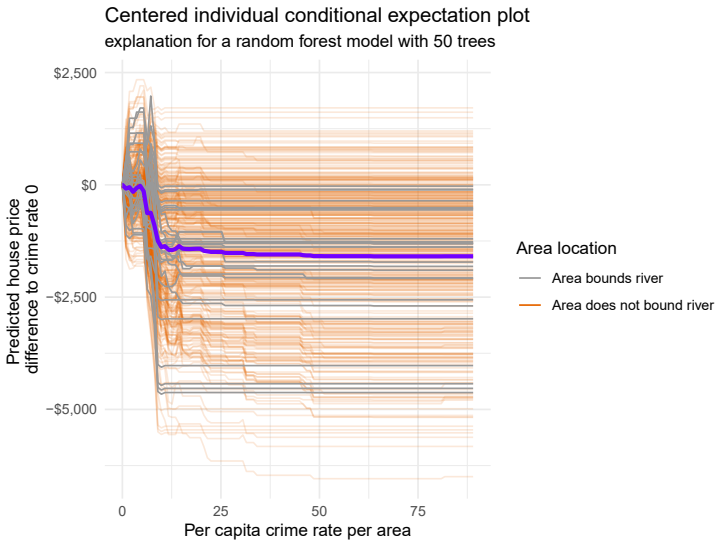

Figure 3.8: Centered ICE plot for a numeric predictor with information on a categoric predictor and PDP

the quantiles of x_S or its actual values.

3.5.2 Shapley Values

Shapley values were first introduced in 1953 (Shapley, 1953), before the research area of machine learning, let alone model-agnostic interpretation methods for machine learning surfaced. The concept stems from game theory and longs to describe a *"method for assigning payouts to players depending on their contribution to the total payout"* (Molnar, 2019). In a machine learning context, the values of a single observation's predictors represent the players, the game is the prediction task for said single instance. The payout or gain represents the difference between the prediction for the single instance (on which the players collaborated) and the average prediction over all observations. Shapley values have desireable properties and advantages that justify discussing the method, even though its downsides prevent it from being fully implemented to interpret and explain a complex machine learning algorithm.

Shapley values do not interpret the actual predicted value but instead the difference between the prediction for a given observation and the average predicted value for all observations. As a local explanation method, each observation x_i has a Shapley value ϕ per feature, so for a given feature j its corresponding $\phi_{i,j}$ may be interpreted as the contribution of the j-th feature to the prediction of instance $x_i, i = 1, ..., n$ compared to the average prediction across all n observations (Molnar, 2019). The Shapley value $\phi_{i,j}$ represents the average contribution of feature j of observation i to the predicted value of observation i across different feature constellations ("coalitions"). Based in game theory, it is the only method satisfying the following four desirable properties or axioms: efficiency (the difference between the predicted value $\hat{f}(x_i)$ on observation x_i and the average prediction $\mathbb{E}\left[\hat{f}(x)\right]$ equals the sum of feature contributions $\phi_{i,j}$, thus guaranteeing fair distribution of the difference among the model's predictors); symmetry ($\phi_{i,j} = \phi_{i,k}$ if features j and k contribute equally to all possible coalitions); dummy (a feature j that never changes the predicted value in any setting has $\phi_{i,j} = 0 \ \forall i = 1, ..., n$) and additivity (if two games with payouts (i.e. prediction differences) are combined, the Shapley values of the new game are the sum of the values of the respective games, so the Shapley values of a random forest are the sum of the values for each tree) (Molnar, 2019).

Slightly tweaking the formula such that the assigned payout is no longer the difference between $\hat{f}(x_i)$ and the mean predicted value but the difference to a prediction of another observation $\hat{f}(x_j)$ or even any value c further allows contrastive explanations (Molnar, 2019), a crucial benefit since social science research indicates that *"people do not explain the causes for an event per se, but explain the cause of an event relative to some other event that did not occur"* (Miller, 2019) and that it for both computational and human explanation is often easier to explain a contrastive question than to provide a full causal attribution (Lipton, 1990)(Miller, 2019).

The original application of Shapley values was in cooperative game theory. In a machine learning context there are multiple implementations differing in their referral to the prediction model, training data and the explanation context. A sample of nine approaches is provided by Sundararajan and Najmi (2020). Differences between the various Shapley value approaches root in two main sources, these being the role of training data in the definition (some procedures rely on training data, others only demand the underlying model's prediction function) and the extension of Shapley values (which are implicitly defined for binary features) to continuous predictors (Sundararajan and Najmi, 2020). Two approaches satisfying linearity, dummy and symmetry are "Baseline Shapley" and "Integrated Gradients" (Sundararajan and Najmi, 2020), yet both do not claim to satisfy the efficiency axiom. A method satisfying all four previously stated axioms is given by (Lundberg and Lee, 2017), but both dummy and linearity can break down for the set function in certain settings (Sundararajan and Najmi, 2020).

Unfortunately, the desireable properties of Shapley value explanations have some unfavorable consequences. The efficiency property guarantees fair payout distribution among all features, but also demands that each predictor within the input data gets reported in the explanation - this may overwhelm recipients of the explanation and also contradicts the human explanation process: Mostly due to cognitive reasons, people select an arbitrary (low) number of available causes as an explanation for a given event (Miller, 2019). As the computation time increases exponentially with the number of predictors, approximative solutions have to be used, which in turn increase the variance of the Shapley values (Molnar, 2019).

In general, Shapley values offer theory-based explanations that fulfill desireable properties, yet their computationally expensive calculation as well as the (for some implementations) necessary access to training data makes exact computation of Shapley values in practice often infeasible.

3.5.3 Locally Interpretable Model-agnostic Explanations (LIME)

In chapter 3.4.4, global surrogate models have been introduced. A potential shortcoming of this approach is that there is no guarantee that the "simple" interpretable model fits the black-box model's complex prediction function across all observations and the complete feature space. However, it is desirable for an observation's explanation to correspond to the model behaviour in the proximity of said observation ("local fidelity") (Ribeiro et al., 2016c), so global accuracy is not always sufficient. This promotes the use of local surrogate models, as they are specifically tailored to explain individual predictions instead of a complex model in total (Molnar, 2019). Surrogate models do not depend on any internal specifics of the complex model but only on its predictions, so they are by definition model-agnostic with all benefits and shortcomings specified in 3.3.

A recently introduced model-explanation method using surrogate models is "Locally Interpretable Model-agnostic Explanations" (LIME) (Ribeiro et al., 2016c). The authors introduce an extensive and flexible framework to explain complex models. In contrast to most methods introduced in 3.4 and 3.5, the authors explicitly introduce the application of their method for text and image data as well as for "classic" tabular data. A key idea to explain predictions on non-tabular data is to use "interpretable data representations" for explanations. Given text data (which is by definition human-readable), a classifier may use word embeddings which are incomprehensible to human users of the model. LIME explanations show interpretable representations that may only indicate the presence or absence of a given word within an observation (usually a document), i.e. a bag of words with limited size. On image data, the authors introduce explanations via the presence or absence of "super-pixels" (i.e. contiguous patches of similar pixels). In mathematical notation, $x \in \mathbb{R}^d$ denotes the original representation of an instance, the binary vector of its interpretable representation is $x' \in \{0,1\}^d$ (Ribeiro et al., 2016c).

The concept of LIME does not directly obligate the human user to use a fixed explanation model (i.e. the surrogate model). Instead, the explanation model is defined as any interpretable model g that can be *"readily presented to the user with visual or textual artifacts"* (Ribeiro et al., 2016c). The complexity of the interpretable explanation model g is measured by $\Omega(g)$. $\Omega(g)$ may be the depth if g is a decision tree, or the number of non-zero weights if g is a linear model. Clearly $\Omega(g)$ cannot grow infinitely, as even for interpretable models such as decision trees, humans can only grasp a certain number of parameters or nodes. As further notation, let $f : \mathbb{R}^d \to \mathbb{R}$ be the (complex) model to be explained. To define the necessary locality of explanations, the authors set $\pi_x(z)$ as a proximity measure between an instance z and the input observation x.

In order to locally approximate f with an interpretable model g around x and its interpretable representation x', non-zero elements of x' are uniformly drawn at random. This results in a perturbed sample $z' \in \{0,1\}^d$. Each z' contains a fraction of the non-zero elements/predictors of the observed x' and gets weighted by π_x. Those generated samples z' that share many elements with x' receive higher weight than those that only share few elements. For each new z' the complex model f calculates $f(z)$ on the original representation $z \in \mathbb{R}^d$. The perturbed samples z and their associated labels (i.e. the model's predictions $f(z)$) form a dataset $\mathcal{Z}$.

To measure the prediction quality of the interpretable model g against the complex model f locally around any observation x, the loss $\mathcal{L}(f, g, \pi_x)$ is introduced. To achieve good locally interpretable explanations with local fidelity, the loss has to be minimized while the complexity $\Omega(g)$ has to remain low enough for the interpretable model $g \in G$ with G being a class of models that are deemed interpretable. LIME produces explanations $\xi(x)$ via optimizing

$$\xi(x) = \underset{g \in G}{argmin}\, \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

on the dataset $\mathcal{Z}$ of perturbed samples within the neighborhood of x. The choice of models that belong to G has to be made with regards to the recipients of the explanation. The authors of the LIME paper focus on using sparse linear models $g(z') = w_g \cdot z'$ for the practical implementation of their concept (Ribeiro et al., 2016c). As for the proximity measure $\pi_x(z)$, the practical implementation suggests an exponential kernel defined on a given distance function D with width σ: $\pi_x(z) = \exp\left(\frac{-D(x,z)^2}{\sigma^2}\right)$. When using text data, D may be the cosine distance. Using this proximity measure, the class of linear models as G, and locally weighted square loss, the loss is

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

For text data, it was already established that the interpretable representation corresponds to a bag of words. To remain interpretable, the size of said bag of words is conditioned to not exceed an integer $K \in \mathbf{N}$ in order to be interpretable, so $\Omega(g) = \infty \mathbb{1}\left[\|w_g\|_0 > K\right]$ with w_g being the weights of the linear model. This formulation of $\Omega(g)$ renders the direct solution of $\xi(x)$ infeasible. However, the solution can be approximated by selecting K features (the authors use LASSO for this task) and using least squares to learn the weights. Algorithm 1 demonstrates the calculation of explanations with LIME. The algorithm produces explanations for any single observation, therefore the computation time depends on the time to produce a prediction from the complex model f and on the number of samples N, but not on the size of the dataset (Ribeiro et al., 2016c).

Require: Classifier f, Number of samples N;
Require: Instance x and its interpretable representation x';
Require: Similarity kernel π_x, Length of explanation $K \in \mathbf{N}$;

$\mathcal{Z} \leftarrow \{\}$;
for $i \in \{1, 2, 3, ..., N\}$ **do**
  $z'_i \leftarrow sample\_around(x')$;
  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
end
$w \leftarrow$ LASSO($\mathcal{Z}$, K) with $f(z)$ as target, maximum K selected features z'_i
Algorithm 1: Sparse Linear Explanations using LIME (Ribeiro et al., 2016c)

LIME produces individual, local explanations for each observation. In order to get a more general understanding about the underlying complex model, explanations for multiple instances have to be assessed. If a user is interested about reasons for a model's failure, it may be beneficial to explain some of those instances that the model predicted wrong. In order to receive a "global understanding" of the model, the "submodular pick" (SP) algorithm is introduced in the same article as LIME. The SP algorithm picks diverse instances with

non-redundant explanations that can represent the model's global behaviour (Ribeiro et al., 2016c). Explanations generated via LIME will be presented in chapter 5.

4 Practical Implementation of Text Mining

As the main focus of this thesis is on interpretable predictions of black-box models in text mining, it is necessary to generate said predictions. The prediction task was to detect a binary (positive or negative) sentiment from movie reviews using black-box algorithms. All computations were made using the software R, version 4.0.5 (R Core Team, 2020).

4.1 Dataset

The prediction task was based on a dataset containing 50,000 movie reviews scraped from the Internet Movie Database (IMDB). This dataset was collected and published by Maas et al. (2011). It is not exactly clear how many distinct movies are included as the data does not contain movie IDs, however the authors state that no movie has more than 30 reviews, so the data includes at least 1667 objects that got rated by users of the website. Each film may be rated by text and a number of stars between one (atrocious) and ten (excellent). Thus, IMDB is an excellent data source for text mining tasks as the awarded stars serve as labels for the text, easily allowing supervised learning. For the binary classification, reviews awarding one to four stars were classified as negative, and those awarding a movie seven stars or more were classified as positive. Balanced reviews with five or six stars were not included in the training or test data.

4.2 Preprocessing the data

Each review was originally entered on a website and is represented as a text file. Due to the web-scraping process, some files further contain HTML snippets that have no relation to the content as well as some characters that were wrongly represented within the encoding. In order to render the data accessible for any supervised machine learning model, extensive preprocessing is necessary. During the preprocessing regime each word is converted to lowercase, all HTML

snippets and most special characters are removed (notable exceptions: apostrophes in contractions and hyphens between characters, allowing the preservation of hyphenated words). Afterwards, the strings get split at each blank. This process of breaking the text into "meaningful constituents" is referred to as "Tokenization" (Feldman and Sanger, 2007). A single document (i.e. a review) is represented by a list of tokens. After having split the text, the tokens can now be recombined into n-grams, combinations of n consecutive linguistical items. 1-grams may be referred to as unigrams (here: single words), 2-grams as bigrams and 3-grams as trigrams (Hvitfeldt and Silge, 2021).

To decrease dimensionality and increase the proportion of words that carry information or sentiment, words that add little informational value ("stop words" such as "the", "a", "by") are removed. Global stop words are detected using the predefined Marimo list from the R-package "stopwords" (Benoit et al., 2021). As all documents within the corpus are movie reviews, movie-related subject specific stop words (e.g. "movie", "film", "actress", "plot") are only included as parts of bi- or trigrams but excluded from appearing as unigrams ("good plot" shows sentiment, "plot" does not).

Most supervised methods cannot process lists of tokens or n-grams directly, so each review has to be processed further. Typically, each document is represented as a vector within the feature space, with each word used within the document collection (i.e. the corpus) being a feature (Feldman and Sanger, 2007). The dimensionality of the feature space may be decreased by removing stop words as well as by only "accepting" those tokens that occur most frequently. The number of words to be included into the feature space is a parameter that can be tuned when training and evaluating supervised models. If the feature space is set to include the 1000 most frequently occuring tokens or n-grams within a corpus, each document within the corpus is represented by a vector of dimension 1000. Those tokens that do not appear in a review are typically represented by 0.

As for representing n-grams present within a document, multiple weighting schemes exist within the classic "bag of words" framework (Witten, 2004)(Feldman and Sanger, 2007). Perhaps the simplest approach is to use binary features only, so each token present within a document is represented by 1. The obvious extension might be to count each n-gram within a document, such that a n-gram is no longer represented by only 0 and 1 but by any natural number. However, doing so puts emphasis on commonly used words, regardless of their significance to a document's class. A commonly used weighting scheme that takes advantage of rare terms carrying more information than frequent ones (Witten, 2004) is to count an n-gram's "term frequency" (tf) and "document frequency" (df). These numbers are then used to compute the "term frequency - inverse document frequency" (tf-idf or tfidf) of an n-gram x within a document d and a corpus of size N. A basic formula is presented by Feldman and Sanger

(2007) as

$$\text{tf-idf}(x, d) = \text{tf}(x, d) \cdot log\left(\frac{N}{\text{df}(x)}\right)$$

with $\text{df}(x)$ representing the number of documents within the corpus that contain the n-gram x and $\text{tf}(x, d)$ counting the n-gram x within document d. This formula may be adapted by standardizing each tf-idf by the length of its corresponding document. It is further common to add 1 to the fraction to avoid calculating the log of zero, which would happen if a term is not appearent in a given document. The default formula to calculate the tf-idf of a term used within the R package "textrecipes" (Hvitfeldt, 2020) is

$$\text{tf-idf}(x, d) = \text{tf}(x, d) \cdot log\left(1 + \frac{N}{\text{df}(x)}\right) \cdot \frac{1}{l_d}$$

using the same notation as above and with l_d being the number of n-grams within document d. This formula is used in all further analysis.

After the preprocessing, the movie reviews have been transformed from a collection of 50,000 labeled text files into a matrix of 50,000 rows. The matrix's first column is the review's label (i.e. positive or negative), the remaining columns contain the tf-idf values. Thus, the previously unstructured textual data can now be analyzed using supervised machine learning algorithms.

4.3 Supervised Learning

The text data has been transformed into a matrix of features suitable for general supervised machine learning algorithms. The purpose of each model is to detect a review's sentiment. As the reviews are labelled as either positive or negative, this corresponds to a binary classification task. Given that LIME, a method for post-hoc interpretability will be applied afterwards, only complex "black-box" model types were selected: random forest, XGBoost and SVM with linear kernel. Every model was trained with different parameters as well as different numbers of n-grams. The parameter constellations are described in table 4.1.

The SVM with linear kernel got trained on less n-grams than the other models due to its enormous computation times. Both the random forest and XGBoost got trained with either 1000, 2000 or 3000 uni- and bigrams as features, whereas the linear SVM only got trained with either 100 or 500 uni- and bigrams. It is worth noting that the majority of features are unigrams with only few bigrams supplementing. Out of the most frequent 1000 n-grams after tokenization and removing stopwords, 973 (97.3%) are unigrams, as are 1872 of the most frequent 2000 (93.6%) and 2729 when including the most frequent 3000 n-grams (91.0%). The SVM sees almost no bigrams at all: 100 of the most frequent 100 and 497 of the most frequent 500 n-grams within the training data are unigrams.

Table 4.1: Parameter constellations for models

Type of model	Number of n-grams used	Additional tuned parameter	Values for tuned parameter
Random Forest	1000, 2000, 3000	Trees	50, 250, 500
XGBoost	1000, 2000, 3000	Learning rate	0.15, 0.30, 0.50, 0.60
linear SVM	100, 500	Cost	0.75, 1

Table 4.2: Random forest evaluated on training data, five-fold crossvalidation

number of n-grams	number of trees	mean accuracy on training data	standard error of accuracy
1000	50	0.82160	0.0019718
1000	250	0.83044	0.0024441
	500	0.83128	0.0022455
2000	50	0.83376	0.0015184
	250	0.83892	0.0027739
	500	0.84148	0.0020086
3000	50	0.83320	0.0022154
	250	0.84560	0.0023281
	500	0.84564	0.0018734

Each model was trained on a balanced subset of 25,000 reviews (12,500 positive and 12,500 negative) using five-fold crossvalidation. As the classification setting is binary and there is no direction of error whose consequences are more severe, the model's accuracy on the test dataset of those remaining 25,000 reviews is used as the metric to select the best parameter constellations per "model type".

4.3.1 Random Forest

For the random forest, the number of trees was chosen as the parameter to be tuned, with either 50, 250 or 500 trees incorporated in the model. Every tree uses $\lfloor\sqrt{m}\rfloor$ variables with m being the number of available predictors (i.e. either 1000, 2000 or 3000), has a minimum node size of 10 reviews, uses the Gini index as splitting criteria and allows only binary splits. Table 4.2 displays the results of all random forest models evaluated on the training data using five-fold crossvalidation. The mean accuracy of the random forests over the datasets created by crossvalidation generally increases both with the number of trees within the model and with the number of uni- and bigrams available as training data. The lowest mean accuracy of 0.8216 is obtained by a random forest using 50 trees and 1000 features, the smallest of all constructed random forest models. In constrast, the maximum mean accuracy of 0.84564 on the training data is obtained by the largest model, containing 500 trees and 3000 features.

Table 4.3: Confusion Matrix - Random forest evaluated on test data

Prediction	Truth	
	negative	positive
negative	0.42704	0.0754
positive	0.07296	0.4246

Table 4.4: XGBoost evaluated on training data, five-fold crossvalidation

number of n-grams	number of trees	mean accuracy on training data	standard error of accuracy
1000	0.15	0.76540	0.0021024
1000	0.30	0.79180	0.0013856
	0.50	0.80104	0.0020942
	0.60	0.80048	0.0018959
2000	0.15	0.76436	0.0026026
	0.30	0.79016	0.0025079
	0.50	0.80208	0.0024320
	0.60	0.80148	0.0023871
3000	0.15	0.76704	0.0040882
	0.30	0.79012	0.0014921
	0.50	0.80208	0.0016169
	0.60	0.80140	0.0014227

Said model with the highest accuracy is then retrained on the full 25,000 reviews of the training dataset before getting evaluated on the test dataset of 25,000 reviews. Identically to the training dataset, the test dataset also contains 50% positive and 50% negative reviews. The final out-of-sample accuracy is 0.85164. Table 4.3 displays the proportional confusion matrix of the random forest evaluated on the test dataset. There is no systematical error as the random forest predicted close to 50% of the comments to be positive or negative as well as correctly identified 42.7% of the reviews as negative and 42.5% as positive reviews.

4.3.2 XGBoost

Introduced by Chen and Guestrin (2016), XGBoost constitutes a computationally efficient implementation of gradient tree boosting. The model allows for flexible parameter constellations, however in this thesis only the learning rate was tuned with possible values of 0.15, 0.30, 0.50 and 0.60. The model consists of fifteen trees, each having binary splits, maximum depth 6 and minimum leaf size one. Contrary to the random forest, each tree uses all available predictors as the trees are built sequentially.

Table 4.5: Confusion Matrix - XGBoost evaluated on test data

Prediction	Truth	
	negative	positive
negative	0.38428	0.07864
positive	0.11572	0.42136

Table 4.4 contains the results of each XGBoost model evaluated on the training data using five-fold crossvalidation. Contrary to the random forest, the mean accuracy does not improve when the number of available features is increased, but lies between 0.76 and 0.80 regardless. Increasing the learning rate to 0.50 yields the highest mean accuracy for all three settings, a learning rate of 0.60 results in a slight decrease on mean accuracy. The worst predictive performance comes from those XGBoost models with a learning rate of 0.15, regardless of the number of features the mean accuracy is around 0.765. The very best XGBoost model on the training data contains 2000 predictors, uses a learning rate of 0.50 and results in a mean accuracy of 0.80208. After being retrained on the full training data, this model results in an accuracy of 0.80564 when being applied on the same balanced test data as the random forest. This is around 4.6 percentage points worse than the best random forest model. Table 4.5 displays the model's confusion matrix on the test data. There appears to be a bias towards positive predictions as the best XGBoost model predicted 53.7% of all reviews to be positive. This results in a larger proportion of negative reviews incorrectly being classified as positive. The performance on those reviews that are factually positive is comparable to the best random forest model, but the XGBoost model is worse on factually negative comments.

4.3.3 Support Vector Machine

For the support vector machine a linear kernel was chosen as it is a popular choice in literature e.g. with Maas et al. (2011) and Go et al. (2009). The cost parameter was selected to be either 0.75 or 1.00, controlling the cost of predicting a sample within or on the wrong side of the SVM's calculated margin. Due to the enormous runtime, the SVM models were only trained on either 100 or 500 predictors, those almost exclusively being unigrams (0 of 100 and 3 of 500 are bigrams).

Table 4.6 displays the mean accuracy of each constructed linear SVM, evaluated on the training data using five-fold crossvalidation. Changing the cost parameter does not result in meaningful differences of the mean accuracy. Predictions improve when the number of predictors is increased to 500. The parameter constellation that yields the highest mean accuracy on training data is to use 500 features and a cost parameter of 0.75, resulting in a mean accuracy of 0.8336. Evaluated on the out-of-sample test data, this model has an accuracy

Table 4.6: linear SVM evaluated on training data, five-fold crossvalidation

number of n-grams	cost	mean accuracy on training data	standard error of accuracy
100	0.75	0.73276	0.0009558
100	1.00	0.73272	0.0010707
500	0.75	0.83360	0.0019829
	1.00	0.83252	0.0013396

Table 4.7: Confusion Matrix - linear SVM evaluated on test data

Prediction	Truth	
	negative	positive
negative	0.41244	0.07572
positive	0.08756	0.42428

of 0.83672. The confusion matrix of this model evaluated on the test data is presented in table 4.7. The linear SVM is slightly biased towards classifying a review as positive, doing so for 51.2% of all predictions. However, this bias is less severe than for the XGBoost model. Therefore, the proportion of negative predictions that are incorrectly classified as positive is slightly higher than the "opposite" error. 41.2% out of 50% are correctly classified as negative and 42.4% out of 50% are correctly classified as positive.

Table 4.8 gives an overview of the results for all models presented. The random forest model has the highest accuracy on both training and test data.

Table 4.8: Results - best parameter constellation per model

model	mean accuracy on training data	accuracy on test data
Random Forest	0.84564	0.85164
XGBoost	0.80208	0.80564
Linear SVM	0.83360	0.83672

5 Generating post-hoc interpretations using LIME

In chapter 4 it was demonstrated that random forest, XGBoost and linear SVM models can be used to differentiate between favourable and unfavourable reviews towards movies. Due to the complex structure of each model, it is unclear what exactly the algorithms have "learned" in order to classify a review as positive or negative. In this chapter, predictions from each model are analyzed using locally interpretable model-agnostic explanations (LIME). Introduced in chapter 3.5.3, LIME provides a framework to explain predictions from any model via a local surrogate model that can be interpreted. Using the implementation provided by Pedersen and Benesty (2021) via the R package "lime", version 0.5.2, said surrogate model is a weighted linear model. As LIME uses "interpretable representations" of the original features (i.e. each n-gram's tf-idf), the linear models are trained using binary features representing the presence or absence of any n-gram which constitutes the basic bag-of-words approach.

In order to generate sufficient trust and gain "global understanding" of a prediction model, Ribeiro et al. (2016c) suggest explaining multiple representative predictions. The authors introduce the "submodular pick" algorithm, a method to select those representative and non-redundant predictions using submodular optimization. However, even though LIME is extensively used, the submodular pick is yet to leave an impact in the literature. Its low usage is also demonstrated by its omission from the "lime" package in R.

In this thesis, the main task for LIME is to explain why the complex models failed to recognize the sentiment of some reviews. Therefore, those ratings that got assigned the highest probability for belonging into the wrong class per model are selected to be explained with LIME. As mentioned in chapter 3.5.3, the size of an explanation given by LIME is flexible. In this scenario, the explanations were constructed to contain the six most relevant uni- or bigrams for each review. Those predictors are selected using LASSO.

5.1 False Negative Predictions

Each model got applied on a balanced dataset of 25,000 observations, containing 12,500 positive and 12,500 negative reviews, as introduced in chapter 4.1. In this section, those two positive reviews which each model predicted to be negative with the highest probability are analyzed and the predictions are explained using LIME. The discussed reviews can be found in section 7 in full length.

5.1.1 Random Forest

The review RFpos1 got classified as negative with probability 0.904. For a human reader, the review clearly contains strongly positive sentiments towards the Danish movie "De største helte":

> This must be one of the funniest Danish movies ever made. Ulrich Thomsen and Thomas Bo Larsen are hilarious, as they drive across Sweden. I don't know how Ulrich Thomsen does it, but somehow he can manage to play insane in a very sane way. BUT if you don't understand Danish (I am not referring to your pastry here) don't waste your time on this Â– I don't think it would work with subtitles.

RFpos1: positive review that the random forest model predicted to be negative with the highest probability

The movie receives praise in the beginning, with positive terms such as "one of the funniest", "hilarious". However, the author warns everybody who is not capable of understanding the movie without subtitles to not watch it at all. The local explanation given by LIME using sparse weighted linear regression suggests that the last part has overwhelming impact on the negative prediction. The explanation can either be presented graphically or in tabular form. Table 5.1 presents the six most important predictors and their weight in the sparse weighted linear regression, i.e. the impact that their presence has on the overall prediction of the underlying complex model. Furthermore, the table contains the predicted label and its probability according to the black box model, and both R-squared and intercept of the surrogate sparse linear model. For the review RFpos1 presented above, the explanation given by LIME suggests that the term "waste" is responsible for the wrong prediction given by the random forest. LIME trained a sparse linear model on perturbed samples of the original review using only six features that were selected via LASSO. As presented in the third column, this simple model fits the predictions of the complex random forest very well, resulting in an R-squared of 0.974. The predictor "waste" has by far the highest positive (and absolute) weight in the linear model, meaning that it contributes most to the predicted label "negative". The terms "time", "ever"

Table 5.1: LIME explanation for the prediction of the random forest model about the sentiment of review RFpos1 using six features

Term	Weight[1]	R-squared[2]	Intercept[3]	Pred. label[4]	Prob.[5]
waste	0.676	0.974	0.107	negative	0.904
time	0.106				
ever	0.034				
hilarious	-0.027				
funniest	-0.019				
somehow	0.009				

[1] weight in local explanation

[2] R-squared of local explanation

[3] Intercept of local explanation

[4] Label predicted by black-box model

[5] Probability of predicted label

and "somehow" also add some weight towards the label "negative", whereas the terms "hilarious" and "funniest" have low negative weights, so LIME realized that these predictors decrease the probability of a review being negative.

Perhaps more aesthetically pleasing, the results of any explanation given by LIME can also be displayed in a plot. Figure 5.1 shows the same explanation as before, using a bar chart. A label and the predicted probability of belonging to that class as well as the local explanation's intercept and R-squared are presented as headers. Using this visualization it becomes clear that the term "waste" has overwhelming responsibility for the negative prediction. The underlying random forest was trained on unigrams and bigrams only. Appearently this resulted in the direct connection from "waste" to negative reviews, and this relation seems to be stronger than the relation between clearly positive terms as "hilarious" or "funniest" and positive reviews.

The full text of the positive review RFpos2 that the random forest predicted to be negative with the second highest probability of 0.903 can be found in the appendix. It starts with the clearly positive phrase *"This movie made me laugh so hard that it hurt"* before using prebuttal. The author states negative aspects (e.g. *"some of you who may think that this movie is nothing more than a waste of film"*) before arguing that the movie is in fact perfectly enjoyable ("*But the thing that most people don't get is that this movie was intended to be bad and cheezy."*). As it was trained only on uni- and bigrams, the random forest was appearently unable to pick up the prebuttal. Table 5.2 contains the results of the local explanation given by LIME.

For this instance, there is no single predictor that has overwhelming weight. Instead, the local explanation's intercept alone lies at 0.577 with no further weight above 0.10. The term "waste" that the explanation identified to strongly

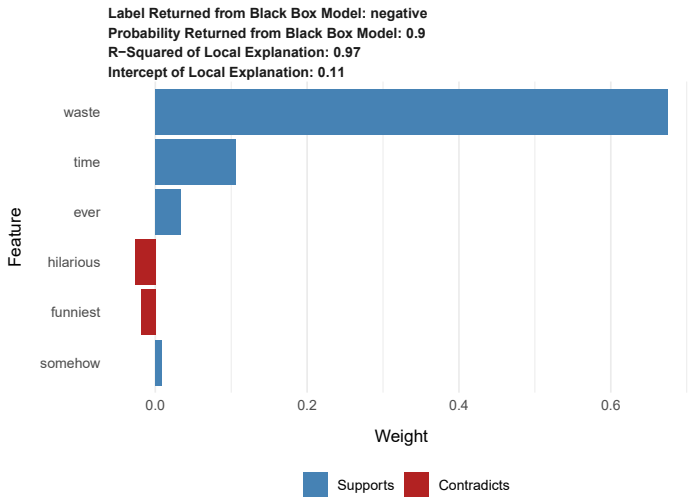

Figure 5.1: LIME explanations for review RFpos1, using six features

Table 5.2: LIME explanation for the prediction of the random forest model about the sentiment of review RFpos2 using six features

Term	Weight	R-squared	Intercept	Pred. label	Prob.
bad	0.096	0.629	0.577	negative	0.904
waste	0.088				
horrible	0.063				
ridiculous	0.057				
nothing	0.041				
instead	0.038				

Table 5.3: LIME explanation for the prediction of the XGBoost model about the sentiment of review XGBpos1 using six features

Term	Weight	R-squared	Intercept	Pred. label	Prob.
bad	0.151	0.793	0.552	negative	0.984
worse	0.086				
laughable	0.076				
unless	0.066				
ridiculous	0.063				
badly	0.062				

influence the classification of review RFpos1 (weight 0.676) is again influential for RFpos2, however it receives lower weight of 0.088. The six predictors selected via LASSO all add positive weight in the local explanation, meaning that they increase the probability of a negative classification when used in a perturbed sample of RFpos2 and applied on the random forest. However, the sparse linear model trained on the perturbed sample could not fit the predictions of the underlying random forest as well as for RFpos1, as is demonstrated by the lower R-squared of 0.629.

5.1.2 XGBoost

Again, LIME is asked to explain those two predictions of positive reviews which the XGBoost model predicted to be negative with the highest probability. Review XGBpos1 describes the movie "The Robot vs. The Aztec Mummy" as *"totally awesome"*, *"hilarious"*, *"countless laughs"* and *"utterly monkeydellic"* (sic!). Furthermore, the author fondly notes the movie's *"laughable"* special effects, *"ridiculous"* acting and *"badly choreographed"* action to underline his positive impression. Any reader of the review is further encouraged to see a dubbed version, having *"even worse actors"*. The XGBoost model was clearly unable to pick up the irony, as it predicted the review XGBpos1 to be negative with probability 0.984.

Table 5.3 gives the local explanations obtained via LIME using six predictors. The local explanation fits the behaviour of the underlying model well, resulting in an R-squared of 0.793. The most relevant predictors are "bad", "worse" and "laughable", all having positive weights that add to the already high intercept of 0.552. This indicates that the XGBoost model (correctly) identified those terms to be negative in their meaning.

The positive review that was predicted to be negative with the second highest probability by the XGBoost model (XGBpos2) turns out to be the same as from the random forest model (RFpos2). However, the XGBoost model was even more confident in its prediction, as the review was classified as negative with probability 0.981, compared to probability 0.904 obtained by the random forest.

Table 5.4: LIME explanation for the prediction of the XGBoost model about the sentiment of review XGBpos2 using six features

Term	Weight	R-squared	Intercept	Pred. label	Prob.
waste	0.122	0.745	0.594	negative	0.981
horrible	0.101				
bad	0.087				
ridiculous	0.059				
nothing	0.047				
instead	0.044				

The local explanation was further able to fit the predictions of the XGBoost model better, as the R-squared increases from 0.629 (underlying model: Random Forest) to 0.745. According to the local explanations presented in tables 5.2 and 5.4, both models agree on the six most influential predictors, suggesting that they "learned" the same underlying structure. Again, the irony and prebuttal was not detected as the preprocesing regime resulted in only unigrams and bigrams without any possibilities to embed further context. However, even if some more complex encoding was used in the preprocessing scheme, LIME would not be able to capitalize on that. This is further discussed in the conclusion.

5.1.3 Support Vector Machine

The support vector machine with linear kernel gets evaluated and "explained" using the same procedure. As mentioned in chapter 4.3.3, the SVM model to be explained was only trained on 500 predictors, a significantly smaller training dataset than used for the best random forest (3000) or XGBoost (2000) models.

As for the review SVMpos1 which got predicted to be negative with probability 0.988, the local explanation fits the behaviour of the SVM only poorly. As presented in table 5.5, the explanation receives an R-squared of only 0.31, which is by far the lowest value of all explanations previously shown. The review is labeled as positive (meaning that the reviewer has rated the movie with at least 7/10 "stars"), even though the text itself is quite critical of the movie. Statements such as *"irrational plot"* and *"poor writing"* reveal negative sentiments, as the author hopes for a *"serious overhaul"*. The explanation indicates a high intercept of 0.802 with the six most influential predictors all adding further positive weight towards a classification as "negative".

LIME's explanation for SVMpos2, the positive review that the linear SVM predicted to be negative with the second highest probability (0.981) has the same shortcomings as the explanation for SVMpos1: the sparse linear model is not able to properly fit the predictions from the complex black box model, as indicated by the low R-squared of 0.535 in table 5.6. This review contains statements as *"my only comment is 'OH WOW' "*, *"pure sexual dynamite"* and

Table 5.5: LIME explanation for the prediction of the linear SVM model about the sentiment of review SVMpos1 using six features

Term	Weight	R-squared	Intercept	Pred. label	Prob.
unfortunately	0.071	0.317	0.802	negative	0.988
poor	0.068				
plot	0.034				
trying	0.028				
making	0.017				
writing	0.014				

Table 5.6: LIME explanation for the prediction of the linear SVM model about the sentiment of review SVMpos2 using six features

Term	Weight	R-squared	Intercept	Pred. label	Prob.
oh	0.159	0.535	0.747	negative	0.981
see	-0.158				
just	0.121				
ok	0.087				
female	0.081				
screen	-0.027				

"I gotta see this movie again". The term that receives the highest weight when fitting a sparse linear model to the SVM predictions is "oh", indicating that this seemingly neutral interjection mostly appears in negative reviews, as do the terms "just", "ok" and "female". On the other hand, the terms "see" and "screen" receive negative weights, contradicting the prediction of the review as being negative.

The inclusion of "female" as a relevant predictor for the SVM's negative prediction may raise some questions. If any "relevant" consequence or decision was to be based on this black-box model, some further analysis may be concluded to potentially detect bias based on gender descriptions, a problem that was also described in section 3.2.

5.2 False Positive Predictions

To further investigate each black-box model, LIME is asked to produce explanations for those reviews with a negative label that each model predicted to be positive with the highest probability. Contrary to chapter 5.1, only one review per model is analyzed. The full reviews can again be found in section 7.

Table 5.7: LIME explanation for the prediction of the random forest model about the sentiment of review RFneg1 using six features

Term	Weight	R-squared	Intercept	Pred. label	Prob.
wonderful	0.102	0.903	0.862	positive	0.909
wants	-0.022				
s	-0.015				
followed	-0.015				
judge	-0.013				
long	-0.006				

5.2.1 Random Forest

The review on which the random forest model seemingly committed an error with the highest probability was in fact falsely labeled, therefore the local explanation for the negative review with the second highest probability to be positive (RFneg1) is presented. The comment does not include any disparaging terms, but describes the movie as *"a long way from the wonderful 'Les enfants du paradis' "*. The local explanation given by LIME is displayed in table 5.7. Having an R-squared of 0.903, the sparse linear model is able to closely model the behaviour of the random forest. According to the explanation, there is no single term responsible for the wrong classification, as the intercept lies at 0.862. The predictor recieving the highest weight adding to the classification as positive is "wonderful", whereas the next five terms selected by LASSO all have negative weights, suggesting that each of those mostly appears in negative reviews.

5.2.2 XGBoost

The review XGBneg1 belongs to the movie "All Dogs Go To Heaven 2", and the author clearly has fond memories of its predecessor, describing it as a *"beautifully animated gem"* before declaring the movie to be *"one of my all-time favorite films"*. However, the same can appearently not be said about the *"charmless, cheesy, uninspired"* sequel with *"tacky animation and an unimaginative plot"*. The high number of positive attributes tricked the XGBoost model into classifying the review as positive. Due to the representation as unigrams and bigrams, the algorithm was unable to detect that said favorable impression belongs to another movie. The local explanation is presented in table 5.8. The six most relevant predictors selected via LASSO all have low positive weights supporting the prediction of the review being favourable to the movie, as does the intercept of 0.621.

Table 5.8: LIME explanation for the prediction of the XGBoost model about the sentiment of review XGBneg1 using six features

Term	Weight	R-squared	Intercept	Pred. label	Prob.
favorite	0.080	0.706	0.621	positive	0.941
gem	0.075				
fun	0.073				
great	0.055				
loved	0.037				
always	0.021				

Table 5.9: LIME explanation for the prediction of the linear SVM model about the sentiment of review SVMneg1 using six features

Term	Weight	R-squared	Intercept	Pred. label	Prob.
entertaining	0.656	0.988	0.301	positive	0.858
getting	-0.071				
eaten	-0.007				
bunch	-0.005				
snakes	-0.004				

5.2.3 Support Vector Machine

The negative review SVMneg1 on which the support vector machine model committed the largest prediction error (positive with probability 0.858) is an unconventional case. The author repeats the statement *"Getting eaten by a bunch of snakes is more entertaining than this film"* multiple times. After preprocessing the string and dropping general and movie-related stopwords, only five terms enter the black-box model and the local explanation, so no further feature selection via LASSO is necessary. According to the explanation obtained by LIME and presented in table 5.9, the presence of the term *"entertaining"* has overwhelming effect on the SVM's prediction. The intercept of the sparse linear model is 0.3, every other term receives negative weight, so every perturbed version of the review that did not include "entertaining" was classified as negative by the SVM (since the scenario is binary and the predicted probability for the review being positive is lower than 0.5). However, the term "entertaining" has positive weight of 0.656, clearly overwhelming all other weights and convincing the linear SVM of classifying the review as positive.

6 Conclusion

This thesis has tried to argue for the importance of explainability of predictive models with a focus on text data. Text mining has become a booming research area, attracting scientific attention from multiple branches without losing its considerable overlap with data mining and data analysis principles. This overlap has been demonstrated by a practical implementation. Three "out-of-the-box" methods from statistical machine learning (random forest, XGBoost, support vector machine with linear kernel) have been trained to solve a classic text mining/sentiment analysis task. After extended standard preprocessing to transform unstructured textual data into structured tabular format with explicit features (uni- and bigrams), the models were trained on 25,000 movie reviews labeled positive or negative. Afterwards, all three models were able to successfully classify the sentiment of more than 80% observations from a test set of 25,000 new movie reviews. Undoubtedly this accuracy could have been increased by using more sophisticated preprocessing steps or more complex model architectures, but this was not the main focus of this thesis.

Instead, the reader's attention should be directed to the notion of explainability and its use in combination with tasks of text analysis and text mining. As there are many practical areas in which models using (textual) data are in use, interpretabily and explainability are not purely academic concepts. If job applications are reviewed and ranked using an algorithm, it is crucial to both the applicant and the company using the model that said algorithm strictly relies on objective criteria without any bias. Inherently interpretable models can provide these explanations by themselves, yet research has shown them to often have lower predictive performance than more complex models. Accuracy, specificity or sensitivity are often used as the only quantifiable measures of a model's "success", even though models need to be evaluated on criteria such as safety or unbiasedness. Model-agnostic post hoc interpretation methods are a feasible way to check for potential problems on those criteria.

After a motivation of explainability and interpretability principles, multiple possibilities to interpret black-box models have been introduced in this thesis. Most of the introduced methods cannot be readily applied to textual data. As demonstrated, partial dependence plots, ICE curves and ALE plots can be applied on both numeric and categorical variables. However, they are not able

to distinguish between the representation of a variable in a model and in the explanation. Text may be transformed into numeric data as input for machine learning models (in chapter 4 tf-idf was used), but any explanation using term frequencies or even tf-idf is hard to interpret for laypeople. As for categorical data, if e.g. an ALE plot is asked to give an explanation based on binary representations of the words, the underlying model has to use the same representation. While the explanation would be easier to understand, the model's predictive power may decrease drastically. LIME allows different representations to be used for predictive and explanatory tasks, harnessing the predictive power of complex preprocessing schemes and models whilst including findings from the social sciences to make explanations as concise as possible.

Having established the need for its deployment on black-box predictions for text data, LIME is not without fault either. The choice of sparse linear models as surrogate models means that, together with weights for each feature selected by LASSO, an intercept is present within the explanation. As presented in chapter 5, the intercept may be a major contributor. The LIME explanation for review RFneg1 presents an intercept of 0.862 whereas the largest weight of an individual predictor (i.e. a word) is 0.102. The interpretation of an intercept with high weight is non-trivial for laypeople, as they might overestimate the impact of those terms explicitly listed within the explanation. In the example above, the largest weight for a term present in the review is 0.102, clearly lower than the intercept. However, the explanation's audience might be tempted to believe that those terms are responsible for the black-box-model's prediction as positive or negative, neglecting the intercept. Furthermore, LIME does not use the same data that was used for creating the black-box model's prediction but relies on "interpretable data representations" instead. While facilitating the presentation of its findings to a non-technical audience, more advanced preprocessing schemes may cause unintuitive explanations. If a model is trained on word embeddings, the sparse linear model used for creating explanations with LIME still relies on a binary representation, resulting in potential loss of information or even misleading explanations. If a word or a phrase is used within a sarcastic context, word embeddings and the complex black-box model might pick up on this finding. Sarcastic praise may well be an indicator for a negative review. Without its context, LIME may present words of praise as explanations for negative classifications, potentially confusing its audience. Presenting explanations together with the phrases in which terms of said explanation appear may be a worthwhile approach.

To conclude, locally interpretable model-agnostic explanations offer a unified framework to explain predictions from black-box models for either tabular, text or even image data (which has not been covered in this thesis). The method does have some shortcomings, but its flexibility and simplicity provide a useful tool to extract and present insights from any black-box model.

7 Appendix - full movie reviews

The dataset containing all 50,000 reviews was published by Maas et al. (2011).

7.1 False negative reviews

7.1.1 Random Forest

Review RFpos1, movie: "De største helte", predicted as negative with probability 0.9039817:

> This must be one of the funniest Danish movies ever made. Ulrich Thomsen and Thomas Bo Larsen are hilarious, as they drive across Sweden. I don't know how Ulrich Thomsen does it, but somehow he can manage to play insane in a very sane way. BUT if you don't understand Danish (I am not referring to your pastry here) don't waste your time on this Â– I don't think it would work with subtitles.

Review RFpos2, movie: "Jack Frost 2", predicted as negative with probability 0.9035611:

> Okay, I'll say it. This movie made me laugh so hard that it hurt. This statement may offend some of you who may think that this movie is nothing more than a waste of film. But the thing that most people don't get is that this movie was intended to be bad and cheezy. I mean, did people actually think that a movie about a killer snowman was intended to be a masterpiece? Just look at the "scary" hologram on the jacket of the movie and you'll find your answer. Instead, like the original Jack Frost (which I thought was just as funny), this movie turned out to be a side-splitting journey into the depths of corny dialogue, bad one liners and horrible special effects. And it's all made to deliver laughter to us viewers. It certainly

> worked for me.For example: Anne Tiler (to her troubled husband): What makes you frown so heavily darling?If that chunk of dialogue doesn't make you laugh, then you have serious issues. Who in their right mind would utter those words in real life? Of course, no one because it was meant to sound ridiculous! Just take one viewing of this movie with an open mind and low expectations, and hopefully you'll see what's so damn funny about Jack Frost 2.

7.1.2 XGBoost

Review XGBpos1, movie: "The Robot vs. The Aztec Mummy", predicted as negative with probability 0.9844366:

> This is a totally awesome movie! If you haven't seen it yet, you damn well should. Sure, the plot is slow to develop, the special effects are laughable, the acting is ridiculous and the action is badly choreographed, but as wrestler DDP would say; That's not a bad thing....that's a good thing! Everything about this movie is hilarious, especially if you get the dubbed version, which has even worse actors. It's countless laughs until you get to the end, yearning for the sequel, where the mummy fights wrestling women. Thus, I give it ten stars. Unless you're one of those 'discriminating' and 'intelligent' people with good taste, who likes only 'high quality' films of the highest calibre, I recommend this utterly monkeydellic movie!

Review XGBpos2, movie: "Jack Frost 2", predicted as negative with probability 0.9808131:

> Okay, I'll say it. This movie made me laugh so hard that it hurt. This statement may offend some of you who may think that this movie is nothing more than a waste of film. But the thing that most people don't get is that this movie was intended to be bad and cheezy. I mean, did people actually think that a movie about a killer snowman was intended to be a masterpiece? Just look at the "scary" hologram on the jacket of the movie and you'll find your answer. Instead, like the original Jack Frost (which I thought was just as funny), this movie turned out to be a side-splitting journey into the depths of corny dialogue, bad one liners and horrible special effects. And it's all made to deliver laughter to us viewers. It certainly worked for me.For example: Anne Tiler (to her troubled husband): What makes you frown so heavily darling?If that chunk of dialogue doesn't make you laugh, then you have serious issues. Who in their right mind would utter those words in real life? Of course, no one because it was meant to sound ridiculous! Just take one viewing of

this movie with an open mind and low expectations, and hopefully you'll see what's so damn funny about Jack Frost 2.

7.1.3 Support Vector Machine

Review SVMpos1, movie: "Hack" (series 1), predicted as negative with probability 0.9879076:

> David Morse and Andre Braugher are very talented actors, which is why I'm trying so hard to support this program. Unfortunately, an irrational plot, and very poor writing is making it difficult for me. I'm hoping that the show gets a serious overhaul, or that the actors find new projects that are worthy of them.

Review SVMpos2, movie: "Let It Be Me", predicted as negative with probability 0.9805253:

> I just finished this movie and my only comment is "OH! WOW!". Jennifer Beals is ok as the fiancee, but Yancy Butler as the female dance instructor is pure sexual dynamite! Having watched her in WITCHBLADE, I was not prepared for the pure unadulterated sensuality and raw sexual excitement she launches onto the screen.I gotta see THIS movie again….if only for Yancy Butler as Corrinne!

7.2 False positive reviews

7.2.1 Random Forest

Review RFneg1, movie: "Les tricheurs", predicted as positive with probability 0.9088183:

> The poet Carne disappears (didnÂ´t he disappeared with PrÃ©vert?) and is followed by the judge Carne. The director wants to give his own vision of a youth that he doesnÂ´t understand and he doesnÂ´t want to. ItÂ´s a long way from the wonderful "Les enfants du paradis"!!!!!!!!!!!!!!!!!!

7.2.2 XGBoost

Review XGBneg1, movie: "All Dogs go to Heaven", predicted as positive with probability 0.9412954:

> How could they take such a beautifully animated gem like Don Bluth's All Dogs go to Heaven and bastardize it with a charmless, cheesy, uninspired sequel. The haunting music and delightful characters are gone, now replaced with tacky animation and an unimaginative plot.The Pros: Charlie Sheen is sometimes fun as Charlie, but he lacks the charming tough guy attitude that brought him to life by Burt Reynolds. I did particularly enjoy the songs "I will always be with you" and "It's too Heavenly here".The Cons: There seems to be no connection between this and the original. In the beginning Charlie is chums with Carface, but wait a minute. Isn't this the same character who was responsible for Charlie's murder and kidnapped the sweet little orphan he loved? I guess that all changes in Heaven but why isn't Anne-Marie even mentioned? If Itchy makes it to Heaven, wouldn't Flo and Killer make it too? What is with Annabelle the whippet's voice? In the original it sounded feminine and charming and in the sequel it sounds like a whiny, bitchy, parrot. The new characters aren't all that great (except Sasha). And the animation is better compared to a generic Saturday morning cartoon. The constant cartoony "humor" is flat and unfunny and the "heart" just doesn't work when compare to the original, which had such a moving unsubtle touch that makes me cry every time.All Dogs go to Heaven is one of my all-time favorite films. How Don Bluth allowed this sequel to be made is beyond me.BOTTOM LINE: Not all sequels go to Heaven.

7.2.3 Support Vector Machine

Review SVMneg1, unknown movie, predicted as positive with probability 0.8579419:

> Getting Eaten By A Bunch Of Snakes Is More Entertaining Than This Film Getting Eaten By A Bunch Of Snakes Is More Entertaining Than This Film Getting Eaten By A Bunch Of Snakes Is More Entertaining Than This Film Getting Eaten By A Bunch Of Snakes Is More Entertaining Than This Film Getting Eaten By A Bunch Of Snakes Is More Entertaining Than This Film Getting Eaten By A Bunch Of Snakes Is More Entertaining Than This Film Getting Eaten By A Bunch Of Snakes Is More Entertaining Than This Film Getting Eaten By A Bunch Of Snakes Is More Entertaining Than This Film Getting Eaten By A Bunch Of Snakes Is More Entertaining Than This Film Getting Eaten By A Bunch Of Snakes Is More Entertaining Than This Film

List of Figures

List of Tables

Bibliography

Angwin, J. and Larson, J. (2016). Machine bias. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing. [accessed 10 February 2021].

Apley, D. W. and Zhu, J. (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 82(4):1059–1086.

Benoit, K., Muhr, D., and Watanabe, K. (2021). *stopwords: Multilingual Stopword Lists*. R package version 2.2.

Biran, O. and Cotton, C. V. (2017). Explanation and justification in machine learning : A survey. In *IJCAI 2017 Workshop on Explanable Artificial Intelligence (XAI)*, pages 8–13.

Bledsoe, W. W. and Browning, I. (1959). Pattern recognition and reading by machine. In *Papers Presented at the December 1-3, 1959, Eastern Joint IRE-AIEE-ACM Computer Conference*, IRE-AIEE-ACM '59 (Eastern), page 225–232, New York, NY. Association for Computing Machinery.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

Dastin, J. (2018). Amazon scraps secret ai recruiting tool that showed bias against women. https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G. [accessed 10 February 2021].

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

Dzindolet, M. T., Peterson, S. A., Pomranky, R. A., Pierce, L. G., and Beck, H. P. (2003). The role of trust in automation reliance. *International journal of human-computer studies*, 58(6):697–718.

Feldman, R. and Sanger, J. (2007). *The Text Mining Handbook - Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, Cambridge.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15(90):3133–3181.

Fisher, A., Rudin, C., and Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of statistics*, 29(5):1189–1232.

Friedman, J. H. and Popescu, B. E. (2008). Predictive learning via rule ensembles. *The annals of applied statistics*, 2(3):916–954.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision.

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.

Harrison, D. and Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102.

Hooker, G. (2007). Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of computational and graphical statistics*, 16(3):709–732.

Hvitfeldt, E. (2020). *textrecipes: Extra 'Recipes' for Text Processing*. R package version 0.4.0.

Hvitfeldt, E. and Silge, J. (2021). *Supervised Machine Learning for Text Analysis in R*. Chapman & Hall/CRC Data Science Series. CRC Press.

Josephson, J. R. and Josephson, S. G. (1994). *Abductive Inference: Computation, Philosophy, Technology*. Cambridge University Press.

Jurafsky, D. and Martin, J. H. (2014). *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Education, Amsterdam.

Kaufman, L. and Rousseeuw, P. J. (1987). Clustering by means of medoids.

Kaufman, S., Rosset, S., and Perlich, C. (2011). Leakage in data mining: formulation, detection, and avoidance. In *Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining*, KDD '11, pages 556–563. ACM.

Kennedy, A. and Inkpen, D. (2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational intelligence*, 22(2):110–125.

Kim, B., Khanna, R., and Koyejo, O. O. (2016). Examples are not enough, learn to criticize! criticism for interpretability. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Lipton, P. (1990). Contrastive explanation. *Royal Institute of Philosophy Supplement*, 27:247–266.

Liu, B. (2015). *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

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

Molnar, C. (2019). *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book.

Molnar, C., König, G., Herbinger, J., Freiesleben, T., Dandl, S., Scholbeck, C. A., Casalicchio, G., Grosse-Wentrup, M., and Bischl, B. (2020). General pitfalls of model-agnostic interpretation methods for machine learning models.

Mosteller, F. and Wallace, D. L. (1964). *Inference and disputed authorship : the Federalist*. Addison-Wesley series in behavioral science : Quantitative methods. Addison-Wesley, Reading, MA.

Mäntylä, M. V., Graziotin, D., and Kuutila, M. (2018). The evolution of sentiment analysis—a review of research topics, venues, and top cited papers. *Computer science review*, 27:16–32.

Nasukawa, T. and Yi, J. (2003). Sentiment analysis: capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on knowledge capture*, K-CAP '03, pages 70–77. ACM.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics.

Pedersen, T. L. and Benesty, M. (2021). *lime: Local Interpretable Model-Agnostic Explanations*. R package version 0.5.2.

Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N. D., Storkey, A., Corfield, D., Hein, M., Hansen, L. K., Ben-David, S., and Kanamori, T. (2008). *Dataset Shift in Machine Learning*. Neural Information Processing series. MIT Press, Cambridge.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Reis, J. C. S., Correia, A., Murai, F., Veloso, A., and Benevenuto, F. (2019). Supervised learning for fake news detection. *IEEE intelligent systems*, 34(2):76–81.

Ribeiro, F. N., Araújo, M., Gonçalves, P., André Gonçalves, M., and Benevenuto, F. (2016a). Sentibench - a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ data science*, 5(1):1–29.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016b). Model-agnostic interpretability of machine learning. In *Proceedings of the 2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 2016)*, pages 91–95.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016c). "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on knowledge discovery and data mining*, KDD '16, pages 1135–1144. ACM.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47.

Shapley, L. S. (1953). A value for n-person games. In Kuhn, H. W. and Tucker, A. W., editors, *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, Princeton.

Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017). Fake news detection on social media: A data mining perspective. *SIGKDD Explorations*, 19(1):22–36.

Sundararajan, M. and Najmi, A. (2020). The many shapley values for model explanation. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9269–9278. PMLR.

Wikipedia contributors (2022). Text mining— Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Text_mining. [accessed 05 March 2022].

Witten, I. H. (2004). Text mining. In Singh, M. P., editor, *The Practical Handbook of Internet Computing*, chapter 14, pages 14–1 – 14–22. Chapman and Hall/CRC Press, Boca Raton, FL.

Zuiderveen Borgesius, F. (2018). *Discrimination, artificial intelligence, and algorithmic decision-making*. Council of Europe, Directorate General of Democracy.