TECHNISCHE UNIVERSITÄT WIEN

DIPLOMARBEIT

Explainable Artificial Intelligence Methods for Modeling Categorical Responses

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Statistik und Wirtschaftsmathematik

unter der Anleitung von

Univ.-Prof. Dipl.-Ing. Dr.techn. Peter Filzmoser

eingereicht von

Marcus Mayrhofer
Matrikelnummer 01607509
ausgeführt am Institut für Stochastik und Wirtschaftsmathematik
der Fakultät für Mathematik und Geoinformation
an der Technischen Universität Wien

Wien, am 13.09.2021

(Unterschrift Verfasser) (Unterschrift Betreuer)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Abstract

Artificial Intelligence (AI) and Machine Learning (ML) are technologies that are not only among the top fields of research, but their areas of applicability are also rapidly increasing. For instance, complex non-linear or non-parametric ML models are already being applied to support key decisions in sectors like health care, criminal justice, or finance. However, while using ML or AI in critical sectors can be viewed as favorable due to their generally high predictive performance and power, the main drawback for human users is that drawing inference, detecting bias or retracing the basis for the different models' decisions is often hard, if not impossible. The field of Explainable AI addresses exactly this prevalent lack of interpretability of various complex machine learning models – and multiple proposals on how to solve this issue have already been made. In this work we discuss three different methods of Explainable AI, which allow for an interpretation of complex models in a way that is accessible to humans. Moreover, we compare the methods in terms of performance and applicability on a use-case scenario, namely to predict the electricity imbalance price for Austria, using data collected from the Transparency Platform, which is operated by the European Network of Transmission System Operators for Electricity (ENTSO-E). Specifically, the considered methods are Local Interpretable Model-agnostic Explanations (LIME), Shapley values for model explainability, and SHapley Additive exPlanations (SHAP), for all of which the focus is placed on their application in the context of classification tasks. Regarding the classification of the electricity imbalance price, we analyze the time-dependent electricity market data from Austria and explore multiple modeling strategies. While all three methods of Explainable AI discussed in this work provide model-agnostic explanation procedures for individual observations, only the model-specific Tree SHAP algorithm from the SHAP framework offers efficient implementations that enable us to fully explain all predictions of tree-based models. The final model for the classification of the electricity imbalance price is based on an online learning approach utilizing boosted tree ensemble models, for which we employ the methods of Explainable AI to interpret the model. Furthermore, tree-based methods do not only yield the best results for the currently available data, but also possess the highest potential for improvement should the data quality or availability improve in the future.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Kurzfassung

Künstliche Intelligenz (KI, englisch: Artificial Intelligence) und Maschinelles Lernen (ML, englisch: Machine Learning) erfreuen sich seit einigen Jahren an einer noch nie dagewesenen Popularität und einem enormen, stetig wachsenden Interesse, sowohl vonseiten der Wissenschaft als auch seitens des Wirtschaftssektors. Die hohe Nachfrage nach den Methoden von KI und ML zeichnet sich vor allem an dem rasanten Wachstum und an der zunehmenden Relevanz ihrer Anwendungsgebiete ab. Bereits jetzt kommen beispielsweise komplexe nicht-lineare oder nicht-parametrische ML-Modelle oftmals im Gesundheitswesen, Strafjustizsystem oder dem Finanzsektor zum Einsatz, um kritische Entscheidungen zu treffen oder die Richtigkeit dieser zu bekräftigen. Die Anwendung von ML oder KI in solchen systemrelevanten Sektoren bringt sowohl Vor- als auch Nachteile mit sich: Einerseits weisen die erstellten Modelle oft eine sehr hohe Prognosegenauigkeit auf, andererseits muss jedoch auch berücksichtigt werden, dass menschliche Anwender die Entscheidungen der oftmals komplexen Modelle nur schwer bis gar nicht nachvollziehen können. Darüber hinaus kann die fehlende Interpretierbarkeit der Modelle auch die Aufdeckung von etwaigen systematischen Verzerrungen deutlich erschweren. Um Lösungswege für eben jene Problemstellungen zu finden, wurde im Feld der KI die untergeordnete Sparte der erklärbaren künstlichen Intelligenz (englisch: Explainable Artifcial Intellegence) geschaffen, aus welcher bereits mehrere Herangehensweisen und Lösungsansätze hervorgegangen sind. Der Fokus der vorliegenden Arbeit liegt auf drei dieser möglichen Ansätze, welche eine Interpretation komplexer Modelle auf eine für den Menschen zugängliche Weise ermöglichen sollen. Darüber hinaus werden die jeweiligen Methoden anhand eines realen Anwendungsbeispiels in Bezug auf ihre Leistungsfähigkeit und Anwendbarkeitseigenschaften miteinander verglichen und analysiert. Zu diesem Zweck soll auf Basis von erhobenen Energiedaten der Transparancy Platform des Verbands Europäischer Übertragungsnetzbetreiber (englisch: European Network of Transmission System Operators for Electricity, ENTSO-E) der Parameter des Ausgleichsenergiepreises für den österreichischen Strommarkt vorhergesagt werden. Konkret beschäftigen wir uns mit folgenden drei Methoden der erklärbaren KI in Bezug auf Klassifikationsprobleme: Local Interpretable Model-agnostic Explanations (LIME), Shapley values zur Modellinterpretierbarkeit, sowie SHapley Additive exPlanations (SHAP). Alle drei der betrachteten Methoden können zur Erklärung individueller Beobachtungen mittels modellunabhängiger Erklärungsverfahren für Menschen verwendet werden. Jedoch bietet nur der Tree SHAP Algorithmus aus dem SHAP Framework eine effiziente Implementierungsmöglichkeit, welche eine vollständige Erklärung aller Beobachtungen Modellen auf Grundlage von Entscheidungsbäumen (englisch: decision trees) erlaubt. Die Methode zur Klassifizierung des Ausgleichsenergiepreises besteht darin, ein schrittweises Prognoseverfahren zu entwickeln, welches anhand der genannten Methoden der erklärbaren KI analysiert wird. Es stellt sich heraus, dass auf Entscheidungsbäumen basierende Modelle nicht nur die höchste Prognosegenauigkeit für die bisher verfügbaren Daten aufweisen, sondern dass sie darüber hinaus auch das stärkste Verbesserungspotential besitzen, sofern sich die Datenqualität zukünftig verbessern sollte oder die Energiemarktdaten zeitnäher verfügbar werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Acknowledgement

During the writing process of this thesis, I received a lot of support and encouragement.

I would like to thank my supervisor Univ.-Prof. Dipl.-Ing. Dr.techn. Peter Filzmoser, Vienna University of Technology. Your expertise and insights were invaluable and enabled me to bring my work to a higher level.

I also would like to express my gratitude to Mag. Jürgen Zeindl, MBA, voestalpine Stahl GmbH. Thank you for your advice, encouragement and for giving me the opportunity to collaborate with the Applied Statistics Department at voestalpine Stahl GmbH.

Further, I want to acknowledge the pleasant, motivating work environment, created by all colleagues in the Applied Statistics Department at voestalpine Stahl GmbH. I am grateful for the opportunity to intern in this department and want to thank you for your collaboration, support, and sustained interest in my work.

Finally, I would also like to thank my family and friends for their love, council, and their sympathetic ear. I could not have completed this thesis without you. I would particularly like to thank Alena Rottensteiner for her invaluable council, patience, and encouragement.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am 13.09.2021

Marcus Mayrhofer

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Contents

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

1. Introduction

Computational science has become an integral and ever-increasing part of our modern-day life, as it is overtaking more and more tasks from humans because it can not only automate and speed up very basic processes, but also because of its potential for creating more efficient and sophisticated data-driven solutions. Machine Learning (ML) and Artificial Intelligence (AI) are two scientific domains that are specifically designed to meet those demands and their respective areas of application are rapidly expanding. Starting from the advertisements we see on Social Media Sites, an email spam filter, or the weather forecast, over to autonomous cars – ML and AI are not only employed as subsidiary tools but are on the same level or even exceeding the performance of humans for many tasks (Molnar, 2019). The astonishing accomplishments of ML and AI are based on the use of non-linear or non-parametric ML models like random forests, gradient boosting or neural networks – which are oftentimes referred to as black-box models. While those methods frequently outperform traditionally used parametric models like linear or logistic regression, their structure and predictions are comparatively harder to trace or interpret. Since ML models and especially black-box models steadily increase to find applications in critical areas like health care, criminal justice or the financial sector, a high performance is oftentimes not the only desired objective – we also want to be able to comprehend the decisions of those models (Lipton, 2018; Biran and Cotton, 2017). Therefore, a crucial part of Machine Learning is to explain and understand the used models as well as their resulting predictions. This task is commonly referred to as **Explainable Artificial Intelligence** or **Interpretable Machine Learning** (IML) (Molnar et al., 2020). In this work we particularly focus on three methods of Explainable AI: Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016), Shapley values for model explainability (Štrumbelj and Kononenko, 2010, 2014), and SHapley Additive exPlanations (SHAP) Lundberg and Lee (2017), which are discussed in Chapters 3, 4 and 5, respectively.

Before we introduce our use-case, which poses as a feasibility study for employing the methods of Explainable AI on a real-world task, we want to motivate the corresponding topic: Economic prosperity as well as everyday life in the twenty-first century are heavily dependent on the access to electricity. Regarding power generation, the focus is shifting from fossil fuels to sustainable and renewable energy sources. While this transition is of extreme importance, it can induce problems with the stability of the power supply, as outlined in Chapter 6. In that chapter we also describe the datasets based on the Austrian energy market data, which can be accessed via the Transparency Platform operated by European Network of Transmission System Operators for Electricity (ENTSO-E). Moreover, we discuss multiple modeling strategies to predict the energy imbalance price for Austria, based on the previously mentioned datasets in Chapter 7, before utilizing the methods of Explainable AI to interpret the most promising model. Furthermore, we want to note that this work was conducted in cooperation with the applied statistics department of voestalpine Steel Division[1], hereafter referred to as voestalpine. Not only does voestalpine provide the specification of the use-case scenario concerning the Austrian energy market as well as the corresponding data, but they also shared the results of previous research concerning the energy market as well as Explainable AI.

Unless stated otherwise, we use `R` (R Core Team, 2021) for all computational tasks and the `ggplot2` (Wickham, 2016) package for the different visualizations. While we will not include `R` code in this work, it is worth mentioning that all code associated with this thesis is provided by the author for voestalpine. This includes but is not limited to the code that was used to produce the results discussed within this thesis. To directly access and interact with the database from voestalpine with `R`, we use the packages `DBI` (R Special Interest Group on Databases (R-SIG-DB) et al., 2021), `odbc` (Hester and Wickham, 2021) and `dbplyr` (Wickham et al., 2021). For data processing purposes, we mainly utilize the packages included in the `tidyverse` (Wickham et al., 2019), additionally we work with `lubridate` (Grolemund and Wickham, 2011) to handle date and time related data as well as `janitor` (Firke, 2021) for data cleaning. As a general model building framework, we use `caret` (Kuhn, 2021), in addition we work with `glmnet` (Simon et al., 2011) as well as `nnet` (Venables and Ripley, 2002) for multinomial logistic regression. For boosted tree ensemble models, we apply `xgboost` (Chen et al., 2021), and `ranger` (Wright and Ziegler, 2017) for random forests. While it is not a focus of this thesis, we work with `keras` (Allaire and Chollet, 2021) to create neural networks. To allow for a more efficient implementation of computationally intensive tasks, we use the libraries `foreach` (Microsoft and Weston, 2020), `future` and `doFuture` (Bengtsson, 2020) for parallelization, additionally `doRNG` (Gaujoux, 2020) helps us generate reproducible results while using parallel computing. To keep track of the progress of all parallel computation tasks, we use `progressr` (Bengtsson, 2021). The original implementations for most of the Explainable AI concepts discussed in this thesis are implemented in `Python` (Van Rossum and Drake, 2009) and we access those methods directly by using `reticulate` (Ushey et al., 2021) as an interface from `R` to `Python`. In `R` we can for example use the packages `iml` (Molnar et al., 2018) and `lime` (Pedersen and Benesty, 2021) to fit local surrogate models. For a model-agnostic computation of Shapley values we may also use `iml` or `fastshap` (Greenwell, 2020a), and the `xgboost` package includes a model-specific procedure to compute Shapley values. For a more detailed overview of the `R` packages related to Explainable AI, we refer to Maksymiuk et al. (2020).

[1]voestalpine Stahl GmbH, voestalpine-Straße 3, 4020 Linz, Austria, https://www.voestalpine.com/stahl/en

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

2. Methods of Explainable Artificial Intelligence

When dealing with Explainable AI, the first objective should be to familiarize oneself with the terminology used in this context. Concerning this topic, we want to note that at the time this thesis is written, there is a lot of ongoing research in this area, but there does not yet exist a common taxonomy used for Explainable AI, which is a topic that is for instance addressed in Doshi-Velez and Kim (2017); Lipton (2018); Biran and Cotton (2017). Arguably the most important term related to this topic is **interpretability**. To the best of the author's knowledge, there is no formal mathematical definition of interpretability. In this thesis we want to state two definitions which fall in line with the objective of the methods which are discussed afterwards:

- *Interpretability is the degree to which an observer can understand the cause of a decision* (Miller, 2019).
- *Systems are interpretable if their operations can be understood by a human, either through introspection or through a produced explanation* (Biran and Cotton, 2017).

As stated in Molnar et al. (2018), the goal of the methods of IML lies in making the actions and predictions of Machine Learning systems understandable to humans. In this work we focus on post-hoc explanation methods for supervised learning ML models with an emphasis on classification, which aim to generate interpretable explanations for model predictions, after a model is fitted.

Before we introduce those methods, we want to provide some examples and motivation why model interpretability of ML models is important. As mentioned in Doshi-Velez and Kim (2017), interpretability can be used to assess some important characteristics of ML models, some of which are listed below:

- Fairness or Unbiasedness: Those properties imply that there is no implicit or explicit discrimination against protected groups.
- Privacy: This assures that sensitive information in the data is protected.
- Reliability or Robustness: Models adhering those properties are not unduly affected by variations of inputs.
- Causality: Ideally, causal relationships are the main drivers of the model.
- Trust: For humans it is easier to trust in a system that provides explanations for its decisions.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Oftentimes we do not only want to understand *how* a model works, but also *why* it makes certain decisions. For example, we could utilize Explainable AI to investigate whether a credit scoring method is discriminatory against minorities.

In the following sections we first introduce local surrogate models, focusing on Local Interpretable Model-agnostic Explanations (**LIME**), as proposed in Ribeiro et al. (2016). Moreover, we analyze the concept of **Shapley values** from cooperative game theory and relate it to the context of model explanations, following Štrumbelj and Kononenko (2010, 2014). Furthermore, we discuss SHapley Additive exPlanations (**SHAP**), particularly discussing Kernel SHAP, which is a method that unites the ideas of LIME and Shapley values (Lundberg and Lee, 2017). Finally, we examine a computationally efficient method to compute Shapley values for tree-based models, as described in Lundberg et al. (2018, 2020).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

3. Local surrogate models

The idea behind local surrogate models is to explain individual predictions of a Machine Learning model by local approximation with an intrinsically interpretable model. Local Interpretable Model-agnostic Explanations (**LIME**), proposed in Ribeiro et al. (2016), suggest a concrete implementation, which is applicable to tabular, text and image data.

3.1. Introductory example to local surrogate models

Since the focus of the current thesis is on classification tasks, we introduce the concept of local surrogate models with a binary classification example, which is visualized in Figure 3.1. Suppose we have two predictors $x_1, x_2 \in \mathbb{R}$ and want to predict whether an observation $x = (x_1, x_2) \in X \subseteq \mathbb{R}^2$ belongs to class $y \in \{\texttt{A}, \texttt{B}\}$, where class `A` is represented by blue dots and class `B` by red triangles. For this simple example we assume that an observation x is classified according to

$$y = \begin{cases} \texttt{blue} & f(x) \leq 0 \\ \texttt{red} & f(x) > 0 \end{cases} \quad \text{with} \quad f(x) = 2 \cdot \sin(x_1) - \mathbb{1}_{x_1>0}(x_1) - x_2.$$

In contrast to most real-world applications, where the decision boundary is produced by a ML model and is not intrinsically interpretable, the decision boundary is explicitly defined by $f(x)$ in this example. Our objective is to approximate f with an intrinsically interpretable model g.

First, we generate an artificial dataset, containing 20 observations, by sampling x_1 from a normal distribution with mean equal to zero and a standard deviation of three, denoted as $\mathcal{N}(0, 3)$, and x_2 according to $\mathcal{N}(0, 4)$ and classifying those observations using f. Those samples are displayed in the top left plot of Figure 3.1, with the decision boundary f being represented by the thick black line and the classification regions being colored respectively. Proceeding to the top right panel, we choose the observation of interest x, which is represented by the big yellow dot. Additionally, 200 new observations are sampled using univariate normal distributions with the sample means and standard deviations of x_1 and x_2 and can be seen as gray dots in the plot. To get a local explanation for the observation of interest, we fit a weighted logistic regression model to approximate f in the proximity of x. For this purpose, the sampled observations are weighted according to the exponential kernel π_x, given by

$$\pi_x(z) = \exp\left(-\frac{\|x - z\|}{(\sqrt{2} \cdot 0.75)^2}\right),$$

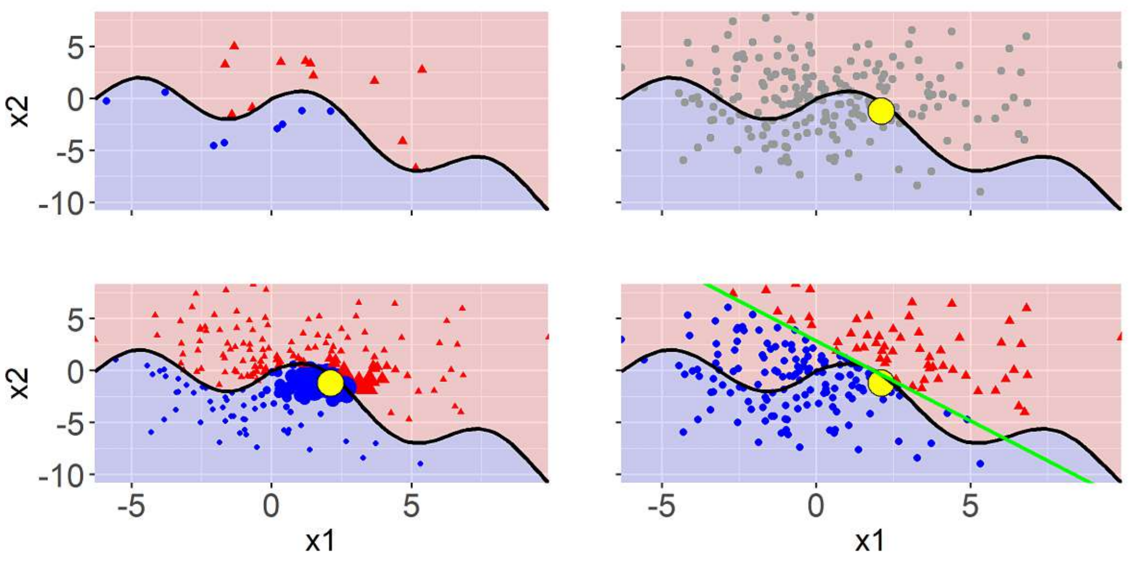


Figure 3.1.: Intuitive visualization of the concept behind local surrogate models and LIME.

which computes the weight for a sampled observation z with respect to x. Those samples are then classified by f, as displayed in the bottom left plot. In the bottom right diagram, we display the decision boundary generated by the weighted logistic regression model g and the sampled observations which are classified according to this model.

3.2. Overview and introduction

As already mentioned, the goal of local surrogate models is to approximate a ML model f in the neighborhood of a single observation x with an interpretable model g. When LIME was originally proposed by Ribeiro et al. (2016), the authors motivated the introduction of their method with three desirable properties of explanations, which we will discuss before introducing the notation and the optimization task connected to LIME.

The first property is **interpretability**, which we already discussed in Chapter 2. An important consideration regarding interpretability is that it always depends on the audience which models, methods or explanations are considered to be interpretable. For example, a person who is not familiar with statistics might consider the visualization of a shallow decision tree interpretable, while the parameters and p-values (Schervish, 1996) of a sparse linear model might be less helpful to them. Likewise, even a professional might struggle to interpret the parameters of a linear model with thousands of significant features. This motivates the inclusion of a complexity constraint to the objective function in Equation (3.1). The second property, **local fidelity**, means that the model should be a good approximation of the ML model f in the locality of the observation of interest. In LIME this is ensured by the structure of the loss function and the resulting minimization problem, given in Equation (3.2). Another desirable characteristic of an explanation method is its appli-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

cability to any kind of Machine Learning model, hence that it is **model agnostic**. Since the local behavior of the ML model f is approximated using a perturbed version of the training dataset, as described in Section 3.3, it is not necessary to make any assumptions for the underlying model.

At this point we want to introduce the formal notation for LIME: First, let $g \in G$ denote the local surrogate model, where G is the class of potentially interpretable models. Since not every model $g \in G$ has to be simple enough to be humanly interpretable, $\Omega(g)$ is introduced as a measure of complexity for the given model $g \in G$. Moreover, the proximity measure (weight function) π_x defines the locality around the observation x. The loss function $\mathcal{L}(f, g, \pi_x)$ is a measure of unfaithfulness of g approximating f in the locality given by π_x.

Considering that we want to get an interpretable model which is locally faithful (local fidelity), $\mathcal{L}(f, g, \pi_x)$ is minimized, while $\Omega(g)$ ensures that $g \in G$ is simple enough to attain humanly interpretable models (interpretability). Therefore, the objective function to obtain an explanation ξ is given by

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \Big\{ \mathcal{L}(f, g, \pi_x) + \Omega(g) \Big\}. \tag{3.1}$$

In practice, the optimization problem in Equation (3.1) can be simplified by limiting G to models with the same complexity, e.g., linear models with the same number of parameters or decision trees with the same depth. By following this approach, $\Omega(g)$ is the same for every model $g \in G$ and can be omitted from the minimization task in Equation (3.1).

An important mathematical detail of this approach is that f and g can be defined on different domains. While the model $f : X \to \mathbb{R}$ operates on the p-dimensional feature space X, the local surrogate model $g : X' \to \mathbb{R}$ is defined on the q-dimensional space of **interpretable representations** X'. As in Lundberg and Lee (2017), let $h_x : X' \to X$ denote the function which relates interpretable feature representations x' with the original features $x = h_x(x')$. Since h_x is specific to the observation of interest x, the transformation works even though x' might contain less information than x.[1]

3.3. Algorithm and specific implementations

In this section we cover the necessary steps to obtain explanations using LIME and compare different existing implementations. First, we present Algorithm 1 for local surrogate models and describe the original implementation (Ribeiro et al., 2016). Subsequently, we perform a more detailed analysis of all the steps in the algorithm, along with a comparison of

[1]When LIME was first introduced, there was no concise description of how the relation between X and X' is defined. In this work, we want to describe this relation using the **simplified input mapping**, introduced in Lundberg and Lee (2017) for the formulation of SHAP as mentioned in Chapter 5. The term simplified input is another name for interpretable representation. More details about the interpretable representations and the simplified input mapping are given in Section 3.3.1.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

the original implementation[2] to the realizations in the `lime` package in `Python` and the procedure for local surrogate models included in the `iml` package in `R` (Molnar et al., 2018).

Algorithm 1: Local surrogate model

```
   Input: Observation to be explained $x$, interpretable representation $x'$
   Input: Machine Learning model $f$, function $h_x$, proximity measure $\pi_x$
   Input: Complexity parameter $K$, sample size $N$
   Output: Explanation $\xi$
1  for $i \leftarrow 1$ to $N$ do
2      $z'_i \leftarrow \text{sample\_around}(x')$
3      $z_i \leftarrow h_x(z'_i)$
4      $y_i \leftarrow f(z_i)$
5      $w_i \leftarrow \pi_x(z_i)$
6  $g \leftarrow \text{weighted\_model}(y, z', w, K)$
7  $\xi \leftarrow \text{interpret}(g)$
8  return $\xi$
```

3.3.1. Interpretable feature representation and sampling

The fact that the local surrogate model g can use different features than the original model f is a big advantage, especially if the original features are non-interpretable. Nevertheless, it is important to note that the interpretable representations X' are derived from the original feature space X. To fit the locally interpretable model g in the space of interpretable representations X' and learn the behavior of the model as the inputs vary, we need a sufficiently large number of observations in the proximity of the observation of interest x. Before new instances are generated, an interpretable feature representation has to be selected. In the original implementation the simplified inputs are given by a binary vector $x' \in X' = \{0,1\}^q$, which is dependent on the input space as described below.

In the case of tabular data, the dimensions of X and X' are equal and the interpretable feature representation of the instance of interest x is given by a vector of ones. A new sample z' is generated by selecting a subset of elements of x', which are changed to zero uniformly at random, resulting in a new instance $z' \in \{0,1\}^p$. Let S be the indices of the non-zero elements of z and $\bar{S} = \{1, ..., p\} \setminus S$ the indices of the zero elements, so we have $z'_S = 1$ and $z'_{\bar{S}} = 0$. At this point the simplified input mapping h_x is used to map z' to the original input space X. For the non-zero elements z'_S we have $h_x(z'_S) = x_S$, while the zero elements $z'_{\bar{S}}$ are drawn depending on the type of the feature. Numerical features are mapped to a random sample drawn from a normal distribution with mean and standard deviation of the training data of the corresponding feature, while categorical features are sampled according to the training distribution.

[2] We want to mention that the `lime` package in `Python` was developed by Ribeiro et al. (2016) and since some details on the sampling procedure are not explicitly explained in the paper, we refer to the documentation of their `Python` implementation.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

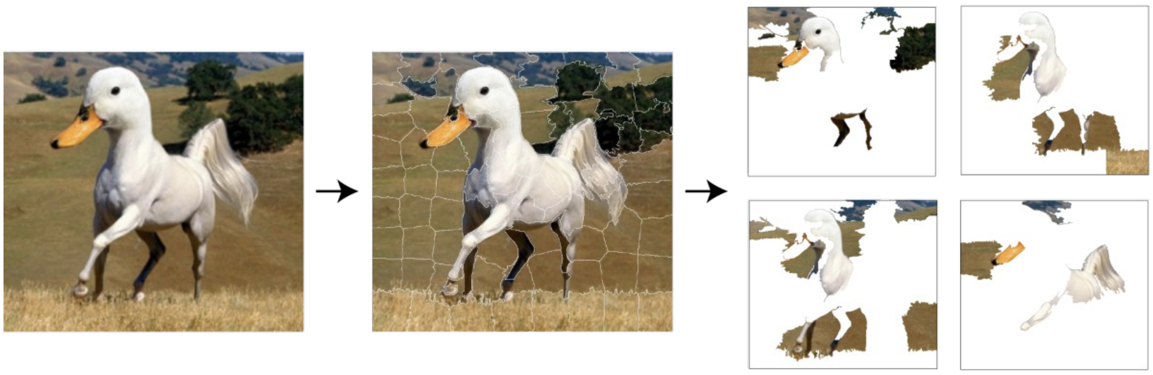


Figure 3.2.: This figure from Biecek and Burzykowski (2021) illustrates the sampling strategy used to create explanations for image data. In the first frame we see the original image, which is then segmented into 100 superpixels. In the last step we see 4 new pictures where only a subset of the 100 superpixels is present.

The default setting in the `Python` implementation of LIME in the `lime` package is to discretize continuous features (for example using quantiles) to categorical variables, followed by binary encoding for each variable. With this setting the permutation strategy for categorical features would be applied to all features, but this behavior can be changed to use the original representation for continuous variables. On the one hand this approach ensures that we obtain a dataset that includes observations which clearly differ from the observation of interest, hence we should be able to produce an explanation. But, on the other hand, it does not account for correlations in the data and can lead to the inclusion of unrealistic data points, with the help of which we then learn the model for the local explanation. To preserve the original data distribution, the `R` version in the `iml` package uses the original data instead of generating a new dataset by permutation. From this we conclude that the best strategy depends on the particular use case, since both approaches have their benefits and drawbacks.

The binary feature representation is especially important and useful for text and image classification. If we think of a text classifier using word embeddings as features, a possible interpretable representation could be a binary vector indicating the presence or absence of a word. For image classifications we could for example consider a picture with a size of 100×100 pixels, which is represented in three color channels, which can be represented by a vector $x \in \mathbb{R}^{30,000}$. As a humanly interpretable alternative we can use a segmentation of the picture into 100 superpixels. The local model can then operate on the binary feature space, that indicates the presence or absence of a superpixel, as illustrated in Figure 3.2. However, since the focus of this thesis is placed on classification of tabular data, we will not go into further details concerning the sampling strategies for image or text data.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

3.3.2. Proximity measure

The proximity measure proposed in Ribeiro et al. (2016) is given by

$$\pi_x(z) = \exp\left(\frac{-D(x,z)^2}{\kappa^2}\right),$$

where D is a distance function and κ is the kernel width. This formulation introduces an additional parameter κ that can be very influential for the final prediction, as visualized in Figure 3.3.

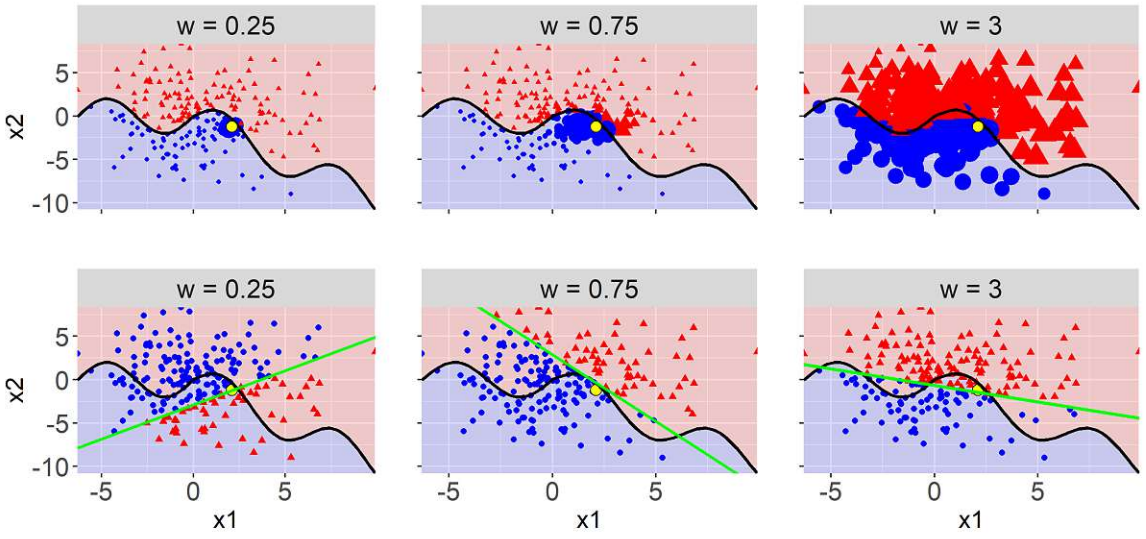


Figure 3.3.: To display the influence of the kernel width $\kappa = \sqrt{p} \cdot w, p = 2$, we use different values of w to derive the weights for the samples as described in the introductory example in Section 3.1. Here we can observe that w and hence κ has a major effect on the approximation. When we choose a too small kernel width, as illustrated in the graphs in the first column, we generate a misleading explanation. While the approximation displayed in the second columns provides a good local estimation, the plots in the last column are biased towards a global estimation. We can easily fix the problem by visualization in this example, but it might lead to unstable explanations for higher dimensional datasets.

In the `Python` implementation this kernel is used by default as proximity measure, with D denoting the Euclidean distance and the kernel width given by $\kappa = \sqrt{p} \cdot 0.75$. To ensure that an explanation created with LIME is actually meaningful and valid, we have to try different kernel widths. A different approach is described in Molnar et al. (2018), where the Gower similarity is used as a proximity measure by default, but it can still be changed to an exponential kernel. The similarity coefficient introduced in Gower (1971) is described in Appendix A. While the Gower similarity has the advantage that it does not introduce an additional parameter, it offers less flexibility and may bias the local model g more towards a global approximation.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

3.3.3. Local surrogate model and explanations

Since our objective is to get interpretable explanations, we want to use a simple local surrogate model $g \in G$ with a limited complexity. Some of the most popular choices are linear or logistic regression models with some sort of variable selection strategy. Another possibility are decision trees with a constrained depth.

In the original implementation G is limited to the class of linear models and feature selection is done by Lasso regression, where the minimization problem is given by

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{3.2}$$

To obtain a simple explanation containing the K most influential parameters, the penalty parameter λ in the Lasso objective function in Equation (3.2) is adjusted, such that only K variables remain. For classification tasks $f : \mathbb{R}^p \to [0, 1]$ models the class probability and in the case of multiple classes $f(x)$ models the probabilities for the respective classes and each class is explained separately. This choice for the local model has a drawback for classification problems since probabilities are estimated with a linear regression model. In the current `Python` implementation of LIME it is possible to distinguish between regression and classification problems and choose appropriate models and variable selection methods, where one of the available options is the procedure described above. In the `R` version implemented in `iml`, the local model is fitted using Lasso and, depending on the task, either a linear or logistic regression model is applied.

In the final step the interpretable explanation ξ is returned to the user. In the case of linear or logistic regression models we have to decide whether we want to return the coefficients or the effects of the local model g. Returning the coefficients as proposed in Ribeiro et al. (2016) can induce problems with interpretation, if we do not use a binary vector as simplified feature representation. This is because a negative coefficient still has a positive influence on the prediction if the feature value is negative as well. It is also worth mentioning that the stability of the explanations created with LIME can be an issue (Alvarez-Melis and Jaakkola, 2018). To reduce the variance of the procedure, we could for example increase the sample size or repeat the procedure multiple times, which comes at the cost of an increased computation time. Moreover, we note that we can intentionally create explanations with LIME to hide biases, as shown in Slack et al. (2020).

3.4. Examples

We will now present two explanations generated by LIME, the first one is for image classification, taken from Ribeiro et al. (2016). In this example the authors trained a classifier to predict whether a Husky or a wolf is displayed in an image. They purposefully biased the training set such that all images containing wolfs have snow in the background.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

In Figure 3.4 we see a Husky, wrongly classified as a wolf along with the explanation created by LIME. This highlights, that the model actually acts as a snow detector and should not be trusted to distinguish between Husky and wolf. The authors even conducted a small study with 27 participants of which ten trusted the wrong classification before seeing the explanation, while only three of them trusted the prediction after seeing the explanation visualized in Figure 3.4.

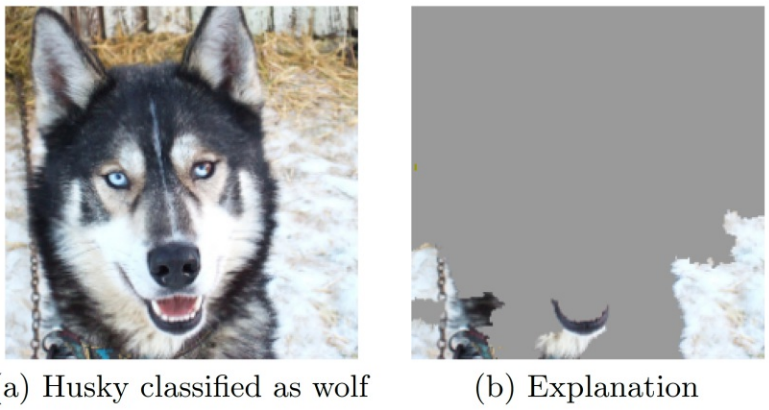


Figure 3.4.: In the left image we see a Husky which is wrongly classified as a wolf and on the right side the explanation produced by LIME.

For the second example, we consider a multinomial classification task based on the iris dataset from Fisher (1936), which consists of 150 observations and five variables named `Sepal.Length` (`S.L`), `Sepal.Width` (`S.W`), `Petal.Length` (`P.L`), `Petal.Width` (`P.W`), and `species` $\in$ {`setosa`, `versicolor`, `virginica`}. We use 105 randomly selected instances as a training set and fit a random forest to predict the parameter `species`, given the remaining four variables. Then we use the `LocalModel` function from the `R` `iml` package to create a local surrogate model to explain one observation from the test set, where we limit the number of features to two. As described in Section 3.3, this approach uses the original data instead of a permuted dataset, as suggested in the original implementation of LIME. The effects are given by the feature value multiplied by the coefficient, which are displayed in Figure 3.5. Since we consider a multinomial classification task in this example, we have to explain each `species` separately. The plot in the left panel shows the effects for the prediction of `species` = `virginica` and in the right panel we display the effects for `species` = `versicolor`. From this comparison we can see that `S.W` = 2.4 has a positive effect on the prediction for `species` = `virginica`, while it has a negative effect for `species` = `versicolor`. Moreover we note that for `species` = `virginica` the most important feature is `P.W` = 1, whereas `S.W` = 2.4 is the most important feature when `species` = `versicolor`.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

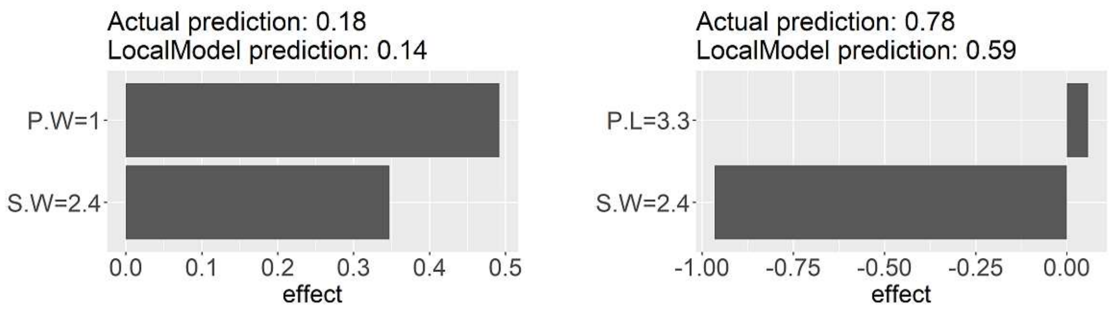


Figure 3.5.: Feature effects of the local surrogate models for `species` = `virginica` (left) and `species` = `versicolor` (right) created with the `R` `iml` package.

3.5. Summary

An outstanding benefit of local surrogate models and LIME is, that the underlying method is intuitive and easy to understand: We approximate a ML model locally with an interpretable model, to gain insights on its behavior, in the proximity of the observation of interest. This should improve our understanding of the ML model. Moreover, explanations created with LIME have the significant advantage that they are straightforward to interpret, even when used by people without a sophisticated background knowledge of statistics or machine learning. Additionally, LIME is a model-agnostic method that can be used to explain not only tabular, but also text and image data. We can also use different features in the local surrogate model than those we used in the original ML model. While the idea of LIME is rather uncomplicated, the question on how we should choose the neighborhood for tabular data remains. We have to be very careful, and we should analyze different settings, since the proximity measure has a major influence on the explanation, as we have seen in the example in Section 3.4. Furthermore, we have to choose an adequate sampling strategy to generate a dataset on which we fit the local model. At this point, problems can occur if we generate unrealistic instances, since those might bias the explanations. Finally, not only the stability of the explanations created with LIME can be an issue (Alvarez-Melis and Jaakkola, 2018), but explanations created with LIME can also be manipulated to hide biases, as shown in Slack et al. (2020).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4. Shapley values

In this section we introduce another model agnostic-method, that allows us to explain predictions for single observations, which can be aggregated to obtain global feature importance measures.[1] This approach relies on Shapley values, a concept which originated in cooperative game theory. There exist several proposals on how to use this method in the context of Explainable AI, however, here we focus on the method introduced in Štrumbelj and Kononenko (2010, 2014). The results and formulations of their work are also used as a foundation in Lundberg and Lee (2017) to introduce SHAP, which is another method based on Shapley values and will be described in Chapter 5.

Intuitively, one might ask at this point: How is cooperative game theory related to explaining model predictions? To answer this question, we first look at how we can compute feature attributions for a single observation in the setting of linear regression, before introducing the theory behind Shapley values from cooperative game theory and ultimately relating them to model agnostic feature importance.

4.1. Notation

For the remainder of this chapter, the set $N = \{1, 2, ..., p\}$ represents the p features present in a ML model $f : \mathbb{R}^n \to \mathbb{R}$. For classification tasks the function $f : \mathbb{R}^n \to [0, 1]$ models the class probability whereas in the case of a multinomial classification problem $f(x)$ models the probabilities for the respective classes. The p-dimensional vector $x = (x_1, ..., x_p)$ represents an instance from the p-dimensional feature space $X = (X_1, ..., X_p)$. Furthermore, $X_S = (X_j : j \in S)$ and $x_S = (x_j : j \in S)$ are subsets of X and x, respectively, where $S \subseteq N$ is a subset of features.

4.2. Feature attributions for linear regression

In the context of linear regression, it is comparatively simple to answer the question of how each feature affects the prediction of a specific instance $x = (x_1, ..., x_p)$, or in other

[1]The global feature importance measures were introduced in Lundberg et al. (2018) and we will describe them in Section 5.5

TU WIEN Bibliothek Your knowledge hub

words how much each feature contributes to the prediction of x. The prediction of a linear regression model for a single instance x can be expressed as

$$f(x) = \beta_0 + \sum_{j=1}^{p} x_j \beta_j,$$

where each feature value is given by x_j and the corresponding weights are given by β_j. The feature effect for the feature value x_j is given by $\beta_j x_j$ and the **feature contribution** $\phi_j(f)$ (Molnar, 2019) for each feature value x_j is given by

$$\phi_j(f) = \beta_0 + \sum_{i=1}^{p} x_i \beta_i - (\beta_0 + \sum_{i=1, i \neq j}^{p} x_i \beta_i + \beta_j \mathbb{E}[X_j]) = \beta_j x_j - \beta_j \mathbb{E}[X_j]. \tag{4.1}$$

We can gather from the definition in Equation (4.1), that the contribution $\phi_j(f)$ of the feature value x_j to the prediction is the difference between the feature effect and the average effect. If we sum over all the feature contributions ϕ_j for the instance x, we obtain

$$\begin{aligned}\sum_{j=1}^{p} \phi_j(f) &= \sum_{j=1}^{p} (\beta_j x_j - \mathbb{E}[\beta_j X_j]) \\ &= (\beta_0 + \sum_{j=1}^{p} \beta_j x_j) - (\beta_0 + \sum_{j=1}^{p} \mathbb{E}[\beta_j X_j]) \\ &= f(x) - \mathbb{E}[f(X)].\end{aligned}$$

Since linear models are additive and therefore features do not interact, they are easy to interpret.

4.3. Feature attributions for general models

In real-world applications, features can often interact and their relation to the response might be non-linear, which prompts the use of more complex black-box prediction models. As mentioned in Štrumbelj and Kononenko (2014), previous approaches like Lemaire et al. (2008); Robnik-Sikonja and Kononenko (2008) used the formulation

$$\phi_j(f) = f(x_1, ..., x_p) - \mathbb{E}[f(x_1, ..., X_j, ..., x_p)] \tag{4.2}$$

to generalize the idea of feature contributions from linear models, as in Equation (4.2), to general models f. For example, if we would compute the feature contributions using Equation (4.2) for the non-linear model $f(x_1, x_2) = \max(x_1, x_2)$ with x_1 and x_2 uniformly distributed on $[0, 1]$, we could run into trouble. Considering the observation $x = (1, 1)$, the model would yield $f((1, 1)) = 1$ and the feature contributions $\phi_1(f)$ and $\phi_2(f)$ would both be zero, which would lead to an unintuitive explanation. To avoid this, we account for interactions by considering all possible subsets of feature combinations. With the aim of

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

generalizing Equation (4.1), the prediction of a model f, where only the subset of features $S \subseteq N$ is known, is defined as

$$f_S(x) := \mathbb{E}[f(X)|X_S = x_S].$$

For the empty set, the definition reduces to $f_\emptyset(X) = E[f(X)]$, which enables us to define the prediction difference $\Delta_S(x)$ for each of the 2^p subsets of feature values $S \subseteq N$ as

$$\Delta_S(x) := f_S(x) - f_\emptyset(x) = \mathbb{E}[f(X)|X_S = x_S] - E[f(X)]. \tag{4.3}$$

Thus $\Delta_S(x)$ is the change in the prediction due to the observation of the feature values x_S. To get p feature contribution values, we have to assign those 2^p prediction differences Δ_S to our features using interactions. Since the definition of $\Delta_S(x)$ in Equation (4.3) still does not account for possible feature interactions, we implicitly introduce interactions by defining that each prediction difference $\Delta_S(x)$ is given by

$$\Delta_S(x) = \sum_{W \subseteq S} I_W(x). \tag{4.4}$$

The definition of the prediction differences in Equation (4.4) allows us to uniquely characterize interactions in this setting as

$$I_S(x) = \Delta_S(x) - \sum_{W \subset S} I_W(x), \tag{4.5}$$

where $I_\emptyset(x) = 0$. Finally, the marginal feature value contribution ϕ_j is defined by assigning it an equal share of all the interactions in which the feature j is participating. Hence, ϕ_j is given by

$$\phi_j(x) = \sum_{W \subseteq N \setminus \{j\}} \frac{I_{W \cup \{j\}}(x)}{|W| + 1} \tag{4.6}$$

and in Theorem 4.3.3 we will further prove that it can be explicitly expressed as

$$\phi_j(x) \overset{!}{=} \sum_{S \subseteq N \setminus \{j\}} \frac{|S|!(p - |S| - 1)!}{p!} \left(\Delta_{S \cup \{j\}}(x) - \Delta_S(x)\right). \tag{4.7}$$

4.3.1. Cooperative game theory

In the following we will introduce some formulations from cooperative game theory, to prove Equation (4.7). The theoretical foundation is based on the book Peters (2008), which can be recommended for a more detailed introduction to the topic. In cooperative game theory players can form coalitions which produce a payoff, and they decide on how the proceeds of their coalitions are distributed.

Definition 4.3.1. *(TU-game)* A coalitional (cooperative) game with transferable utility (N, v) is given by a set of players $N = \{1, 2, ..., p\}$ and the characteristic function v, which assigns the worth $v(S) \in \mathbb{R}$ to each coalition $S \subseteq N$, such that $v(\emptyset) = 0$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

In other words, the function v tells us how much collective payoff a coalition S of players can gain by cooperating. A payoff distribution for the grand coalition N is given by $\varphi(v) = (\varphi_1(v), ..., \varphi_p(v))$, where $\varphi_j(v) \in \mathbb{R}$ is the payoff to player j. There are several proposals on how the payoff should be assigned to the players $j \in N$ to obtain a *fair* distribution. Whilst there are different solution concepts and notions of fairness, we will focus on the ones introduced by Shapley (1953). The **Shapley value** is the only payoff distribution that fulfills Axioms 4.3.1 - 4.3.4, therefore allowing for an axiomatic definition of a fair payoff distribution for the grand coalition N.

Axiom 4.3.1. *(Efficiency)* The payoff to the individual players $\varphi_j(v)$ must sum up to the worth of the grand coalition $v(N)$, hence $\sum_{j=1}^{p} \varphi_j(v) = v(N)$.

Axiom 4.3.2. *(Dummy)* If a player j in a game (N, v) is a null player, meaning $v(S \cup i) - v(S) = 0$ for every coalition $S \subseteq N \setminus \{j\}$, then $\varphi_j(v) = 0$.

Axiom 4.3.3. *(Symmetry)* If $v(S \cup \{j\}) = v(S \cup \{k\})$ holds for all $S \subseteq N \setminus \{j, k\}$ for two players j and k, then $\varphi_j(v) = \varphi_k(v)$.

Axiom 4.3.4. *(Additivity)* For any two games (N, v) and (N, w) it holds that $\varphi(v + w) = \varphi(v) + \varphi(w)$.

Before we define the Shapley value, we first introduce the notion of the predecessors of a player, which is then used to describe the marginal contribution of a player to a TU-game (N, v).

Definition 4.3.2. *(Predecessors of a player)* Let $\sigma : N \to N$ be a permutation of the player-set of a TU-game (N, v). The predecessors of player j in σ are defined as the set of players

$$P_\sigma(j) := \{l \in N | \sigma^{-1}(l) < \sigma^{-1}(j)\}.$$

For example, if the set of players is given by $N = \{1, 2, 3, 4, 5, 6\}$ and $\sigma(N) = \{6, 2, 5, 4, 1, 3\}$, then the set of predecessors of player 1 is given by $P_\sigma(1) = \{6, 2, 5, 4\}$.

Definition 4.3.3. *(Marginal vector)* For a given permutation σ of the player-set of a TU-game (N, v), the components of the marginal vector m^σ are defined as

$$m_j^\sigma = v(P_\sigma(j) \cup \{j\}) - v(P_\sigma(j)). \tag{4.8}$$

This means that every player j receives the marginal contribution, which is created by him joining the coalition.

Definition 4.3.4. *(Shapley value)* The Shapley value $\phi(v) = (\phi_1(v), ..., \phi_p(v))$ of a TU-game (N, v) is the average of all marginal vectors of the game, hence it is given by

$$\phi(v) := \frac{1}{p!} \sum_{\sigma \in \Pi(N)} m^\sigma, \tag{4.9}$$

where $\Pi(N)$ is the set of all ordered permutations of N.

TU Bibliothek Your knowledge hub WIEN
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Following an example from Peters (2008), let (N, v) be a three-person game with $v(1) = v(2) = v(3) = 0$, $v(1,2) = 4$, $v(1,3) = 7$, $v(2,3) = 15$, and $v(1,2,3) = 20$. The marginal vectors of this game are given in Table 4.1 and the Shapley value is equal to $(1/6) \cdot (21, 45, 54)$.

Table 4.1.: Marginal vectors for the Shapley value computation example in Section 4.3.1.

$\sigma(\{1,2,3\})$	m_1^σ	m_2^σ	m_3^σ
(1,2,3)	0	4	16
(1,3,2)	0	13	7
(2,1,3)	4	0	16
(2,3,1)	5	0	15
(3,1,2)	7	13	0
(3,2,1)	5	15	0
$\sum$	21	45	54

Moving on, we state a probabilistic interpretation of the Shapley value as introduced in Definition 4.3.4. Assuming we are drawing from an urn containing all permutations $\Pi(N)$ of the player-set N, a permutation $\sigma \in \Pi(N)$ will be selected with probability $(p!)^{-1}$. Following the order specified by σ, we let the players enter a coalition one by one and assign each one the marginal contribution created by their respective entrance into the coalition. The Shapley value assigns each player their expected payoff in the random procedure described above.

To obtain the formulation of the Shapley value, used in Theorem 4.3.1, we combine equations (4.8) and (4.9) to rewrite the j-th component of the Shapley value as

$$\phi_j(v) = \frac{1}{p!} \sum_{\sigma \in \Pi(N)} v(P_\sigma(j) \cup \{j\}) - v(P_\sigma(j)). \tag{4.10}$$

It should be noted that the terms inside the sum in Equation (4.10) are of the form

$$v(S \cup \{i\}) - v(S),$$

with $S \subseteq N \setminus \{i\}$. Before we can rewrite the sum with respect to the subsets S, we have to determine how many orderings it holds that $P_\sigma(j) = S$. The number of orderings of the set S is given by $|S|!$ and for the set $N \setminus (S \cup \{i\})$ we have $(p - |S| - 1)!$ many, so in total there are $|S|!(p - |S| - 1)!$ orderings where $P_\sigma(j) = S$. This enables us to denote Equation (4.10) as

$$\phi_j(v) = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|!(p - |S| - 1)!}{p!} \left(v(S \cup j) - v(S) \right). \tag{4.11}$$

TU Bibliothek WIEN Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Theorem 4.3.1. *For a TU-game (N, v), there exists a unique payoff distribution $\varphi(v)$ for the grand coalition N, which fulfills the Efficiency, Dummy, Symmetry and Additivity axioms: The Shapley value payoff $\phi(v) = (\phi_1(v), ..., \phi_p(v))$, which is given by*

$$\phi_j(v) = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|!(p - |S| - 1)!}{p!} \left(v(S \cup j) - v(S)\right).$$

Consequently, each player $j \in N$ receives their average marginal contribution across all possible coalitions.

Proof. A detailed proof of Theorem 4.3.1 can be found in Shapley (1953); Peters (2008). ☐

Since the introduction of the Shapley value in Shapley (1953), it has been generalized and reformulated and we state another characterization, introduced in Young (1985), which is based on Axioms 4.3.1, 4.3.3 and 4.3.5.

Axiom 4.3.5. *(Monotonicity)* If for any two games (N, v_1) and (N, v_2) and all $S \subseteq N$ the condition

$$v_1(S \cup \{i\}) - v_1(S) \geq v_2(S \cup \{i\}) - v_2(S),$$

is satisfied, then $\varphi_j(v_1) \geq \varphi_j(v_2)$.

Theorem 4.3.2. *For a TU-game (N, v), the only payoff distribution $\varphi(v)$ which complies with the Efficiency, Symmetry and Monotonicity axioms is the Shapley value.*

Proof. The proof that the Axioms Dummy and Additivity can be replaced by the Axiom Monotonicity can be found in Young (1985). ☐

Finally, yet another generalization of the Shapley value can be obtained by using the Harsanyi dividend (Harsanyi, 1963).

Definition 4.3.5. *(Harsanyi dividend)* Considering a TU-game (N, v), we have that for each coalition $S \subseteq N$ the Harsanyi dividend is recursively defined as

$$\begin{aligned} d_v(\emptyset) &:= 0, \\ d_v(S) &:= v(S) - \sum_{W \subset S} d_v(W), \text{ if } |S| \geq 1. \end{aligned}$$

Using the Möbius transformation or Möbius inverse, $d_v(S)$ can be explicitly expressed as $d_v(S) = \sum_{W \subseteq S} (-1)^{|S|-|W|}$ (Grabisch, 2016). The Shapley value $\phi_j(v)$ for the j-th player is given by the sum of equally distributed dividends over all coalitions in which player j participates:

$$\phi_j(v) = \sum_{S \subseteq N : j \in S} \frac{d_v(S)}{|S|}.$$

To summarize, the Shapley value is a method that tells us how the payoff $\varphi(v)$ of a TU-game (N, v) is assigned to each player j in the grand coalition N, depending on their contribution to the total payout.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4.3.2. Connection to model agnostic feature importance

We will now use the results from cooperative game theory to formulate and proof Theorem 4.3.3 and relate the Shapley value to the context of Explainable AI.

Theorem 4.3.3. *The set of features $N = \{1, 2, ..., p\}$, together with the difference function*

$$\Delta_S(x) = \mathbb{E}[f(X)|X_S = x_S] - E[f(X)], \qquad \text{(4.3, revisited)}$$

define a TU-game $(N, \Delta(x))$ and the marginal feature value contributions $\phi = (\phi_1, ..., \phi_p)$ introduced in Equation (4.7) correspond to this game's Shapley value, where the components are given by

$$\phi_j(x) = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|!(p - |S| - 1)!}{p!} \left(\Delta_{S\cup\{j\}}(x) - \Delta_S(x)\right). \qquad \text{(4.7, revisited)}$$

Proof. We begin by showing that $(N, \Delta(x))$ actually defines a TU-game as stated in Definition 4.3.1: Since $\Delta(x)$ is defined for every subset $S \subseteq N$ and

$$\Delta_\emptyset(x) = f_\emptyset(x) - f_\emptyset(x) = 0,$$

we conclude that $(N, \Delta(x))$ can be treated as a TU-game.

To prove that the marginal feature value contributions $\phi_j(x)$ coincide with the Shapley value of the game $(N, \Delta(x))$, we first rewrite the recursive definition of the interactions $I_S(x)$ given in Equation (4.5) as

$$I_S(x) = \Delta_S(x) - \sum_{W \subset S} I_W(x) = \sum_{W \subseteq S} ((-1)^{|S|-|W|}\Delta_W(x)). \qquad (4.12)$$

The second equality in Equation (4.12) holds since it is equivalent to the conversion from the recursive to the explicit formulation of the Harsanyi dividend. To obtain a non-recursive expression for the contributions, we combine equations (4.6) and (4.12) to obtain

$$\phi_j(x) = \sum_{W \subseteq N \setminus \{j\}} \frac{I_{W\cup\{j\}}(x)}{|W| + 1} \qquad (4.13)$$

$$= \sum_{W \subseteq N \setminus \{j\}} \left(\frac{1}{|W| + 1} \sum_{Q \subseteq (W\cup\{j\})} ((-1)^{|W\cup\{j\}|-|Q|}\Delta_Q(x)) \right). \qquad (4.14)$$

To further simplify the term in Equation (4.14), we count the appearances of $\Delta_Q(x)$ in the double sum and distinguish between the cases of $j \in Q$ and $j \notin Q$. To account for the first case, we denote Q with $j \in Q$ as

$$Q = (S \cup \{j\}) \text{ with } S \subseteq N \setminus \{j\},$$

and analyze how often $\Delta_{S\cup\{j\}}(x)$ occurs. Let $M_{\Delta_{S\cup\{j\}}(x)}$ be the weighted sum of those occurrences an let $k = (p - |S| - 1)$ denote the number of elements in the set $N \setminus (S \cup \{j\})$.

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

The term $\Delta_{S\cup\{j\}}(x)$ occurs only in interactions I_W, where $(S \cup \{j\}) \subseteq W$ and solely appears once in such an interaction. For those W we have that $|W| = |S| + a, a \in \{1, ..., k\}$ and, depending on whether a is even or odd, $\Delta_{S\cup\{j\}}(x)$ occurs with alternating signs. In Equation (4.14) such a set W appears exactly $\binom{k}{a}$ times, because we can combine S with any a element from the remaining k features in $N \setminus (S \cup \{j\})$. Next, we sum up all those terms up to $W = N$ and we have to consider that each interaction I_W is divided by $|W|$, thus we obtain

$$M_{\Delta_{S\cup\{j\}}} = \frac{\binom{k}{0}}{p-k} - \frac{\binom{k}{1}}{p-k+1} + ... \pm \frac{\binom{k}{k}}{p} = \sum_{i=0}^{k}(-1)^i \frac{\binom{k}{i}}{p-k+i} =: V(p,k).$$

The same approach can be applied for the second case where $j \notin Q$. To stay consistent with the notation used for the previous case, we denote Q as $Q = S, S \subseteq N \setminus \{j\}$, count the number of appearances of $\Delta_S(x)$ and obtain $M_{\Delta_S} = -V(p,k)$. The result can be explained by the fact that the size of the set W, where $S \subseteq W$, is $|W| = |S| + 1$ and hence the sign in the summation changes.

In the next step, we examine the series $V(p,k)$ and show that it can be rewritten as a beta function. We start with expanding the binomial $(1-x)^k$ as

$$\begin{aligned}(1-x)^k &= \binom{k}{0} - x\binom{k}{1} + x^2\binom{k}{2} - ... \pm x^k\binom{k}{k} \\ &= \sum_{i=0}^{k}(-1)^i x^i \binom{k}{i},\end{aligned}$$

then we multiply it with x^{p-k-1} and obtain

$$x^{p-k-1}(1-x)^k = \sum_{i=0}^{k}(-1)^i\binom{k}{i}x^{p-k+i-1}. \tag{4.15}$$

By integration over the interval $[0,1]$, we obtain the beta function on the left-hand side of Equation (4.15) while the right-hand side corresponds to the series $V(p,k)$.

$$\begin{aligned}\underbrace{\int_0^1 x^{p-k-1}(1-x)^k dx}_{B(p-k,k+1)} &= \int_0^1 \left(\sum_{i=0}^{k}(-1)^i x^{p-k+i-1}\binom{k}{i}\right) dx \\ &= \sum_{i=0}^{k}(-1)^i \left(\int_0^1 x^{p-k+i-1}dx\right)\binom{k}{i} \\ &= \sum_{i=0}^{k}\frac{(-1)^i\binom{k}{i}}{p-k+i} = V(p,k)\end{aligned}$$

By rewriting the beta function using gamma functions, the series simplifies to

$$V(p,k) = B(p-k,k+1) = \frac{\Gamma(p-k)\Gamma(k+1)}{\Gamma(p+1)} = \frac{(p-k-1)!k!}{p!}. \tag{4.16}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Finally, we can rewrite the marginal feature value contributions as

$$\begin{aligned}\phi_j(x) &= \sum_{S\subseteq N\setminus\{j\}} \big(V(p,k)\Delta_{S\cup\{j\}}(x) - V(p,k)\Delta_S(x)\big)\\ &= \sum_{S\subseteq N\setminus\{j\}} \frac{|S|!(p-|S|-1)!}{p!}\left(\Delta_{S\cup\{j\}}(x) - \Delta_S(x)\right), \end{aligned} \tag{4.17}$$

which corresponds to the Shapley value's j-th coordinate for the TU-game (N, v). The coordinates ϕ_j of the Shapley value can be rewritten in terms of the conditional expectations by combining equations (4.3) and (4.17) to obtain

$$\phi_j(x) = \sum_{S\subseteq N\setminus\{j\}} \frac{|S|!(p-|S|-1)!}{p!}\left(\mathbb{E}[f(X)|X_{S\cup\{j\}} = x_{S\cup\{j\}}] - \mathbb{E}[f(X)|X_S = x_S]\right).$$

□

The Axioms 4.3.1 - 4.3.4 of the Shapley value can now be interpreted as follows:

- **Efficiency:** The marginal feature value contributions add up to the prediction difference between x and the average prediction:

$$\sum_{j=1}^{p} \phi_j(x) = \Delta_N(x) = \mathbb{E}[f(X)|X_N = x_N] - \mathbb{E}[f(X)] = f(x) - \mathbb{E}[f(X)]$$

- **Dummy:** If all coalitions satisfy the condition that a feature has no influence on the prediction when it is added to a coalition, then the feature's Shapley value is zero:

$$\forall S \subseteq N\setminus\{j\} : \Delta_{S\cup\{j\}}(x) = 0 \Longrightarrow \phi_j(x) = 0.$$

- **Symmetry:** If the prediction differences of two feature values are the same for all possible coalitions, then their marginal feature value contributions are identical:

$$\forall S \subseteq N\setminus\{j,k\} : \Delta_{S\cup\{j\}}(x) = \Delta_{S\cup\{k\}}(x) \Longrightarrow \phi_j(v) = \phi_k(v).$$

- **Additivity:** If a model f is a sum of two other models f_1 and f_2, then the Shapley value for f is the sum of the Shapley values of models f_1 and f_2.

The exact computation of the Shapley values as described in Theorem 4.3.1 involves the evaluation of all possible feature value coalitions with and without the j-th feature. Consequently, as the number of features increases, the computation time grows exponentially and solving Equation (4.7) is no longer a feasible approach in practical applications in the presence of high dimensional data. In the following we will discuss two possible model agnostic approaches that can be used to speed up the computation time. While the first method introduced in Štrumbelj and Kononenko (2014) is based on Monte-Carlo sampling, the second method – called SHAP by Lundberg and Lee (2017) – provides a framework

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

that allows us to unify multiple model agnostic feature importance concepts, including LIME and Shapley values. At this point we want to note that the Shapley value depends on how we model the prediction difference (value function of the corresponding TU-game) and for this reason on the sampling strategy. On the one hand, if we use the marginal distribution of the features for sampling, it might lead to unrealistic instances. On the other hand, sampling based on the conditional distribution can solve this problem, but the resulting values are no longer Shapley values, as discussed in Janzing et al. (2020). In the following we describe the approach based on Monte-Carlo sampling, before discussing SHAP framework in Chapter 5.

4.3.3. Approximation of Shapley values

First, we rewrite Equation (4.7) using the alternative formulation of the Shapley value given in Equation (4.10) to obtain

$$\phi_j = \frac{1}{p!} \sum_{\sigma \in \Pi(N)} \left(\Delta_{P_\sigma(j) \cup \{j\}}(x) - \Delta_{P_\sigma(j)}(x) \right). \tag{4.18}$$

Here, $\Pi(N)$ is the set of all ordered permutations of N and $P_\sigma(j)$ is the set of predecessors of the j-th feature in the coalition σ, as introduced in Definition 4.3.2. Since the computation of the Δ-terms in Equation (4.18) poses a task of exponential computational complexity, merely using a sampling approach to simplify Equation (4.18) is not sufficient for achieving an efficient algorithm. In Štrumbelj and Kononenko (2014) they assume that the features are mutually independent, which allows us to estimate the j-th component of the Shapley value ϕ_j via

$$\hat{\phi}_j = \frac{1}{M} \sum_{m=1}^{M} \left(f(\tilde{x}_{j+}) - f(\tilde{x}_{j-}) \right). \tag{4.19}$$

In Equation (4.19), $\tilde{x}_{j+}$ denotes a combination of the instance of interest x and a random data point z, where a random number of feature values of x is replaced by feature values of z, apart from the j-th feature value. The instance $\tilde{x}_{j-}$ is almost identical to $\tilde{x}_{j+}$, the only difference being that the j-th feature value of x is also replaced by the j-th feature value of z. Since we are computing the average of the samples in Equation (4.19), those instances are implicitly weighted by the probability distribution of X. To get an estimate of the Shapley value, we have to repeat this procedure for all p features.

Algorithm 2 summarizes this approach. First, a random observation z is selected from the dataset and a permutation σ is generated. Afterwards, two new instances $\tilde{x}_{j+}$ and $\tilde{x}_{j-}$ are assembled as a combination of the permuted instances x_σ and z_σ.[2] Then we compute $\hat{\phi}_j^m = f(\tilde{x}_{j+}) - f(\tilde{x}_{j-})$ and average them over the M iterations to get the approximation $\hat{\phi}_j = \frac{1}{M}\sum_{m=1}^{M} \hat{\phi}_j^m$.

[2]Assembling instances this way has the drawback that the dependence structure of the data is ignored, thus this sampling strategy might include feature values that do not make sense for this instance.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

Algorithm 2: Approximation of the Shapley value for one feature value.

```
    Input: Observation to be explained $x$, feature of interest $j$
    Input: Machine Learning model $f$, data matrix $X$, Number of iterations M
    Output: Approximation of the Shapley value for the $j$-th feature value
 1  for $m \leftarrow 1$ to $M$ do
 2      $z \leftarrow \text{random\_sample}(X)$
 3      $\sigma \leftarrow \text{random\_permutation}(\{1, ..., p\})$
 4      $x_\sigma \leftarrow \sigma(x) = (x_{(1)}, ..., x_{(j)}, ..., x_{(p)})$
 5      $z_\sigma \leftarrow \sigma(z) = (z_{(1)}, ..., z_{(j)}, ..., z_{(p)})$
 6      $\tilde{x}_{j+} \leftarrow \sigma(z) = (x_{(1)}, ..., x_{(j-1)}, x_{(j)}, z_{(j+1)} ..., z_{(p)})$
 7      $\tilde{x}_{j-} \leftarrow \sigma(z) = (x_{(1)}, ..., x_{(j-1)}, z_{(j)}, z_{(j+1)} ..., z_{(p)})$
 8      $\hat{\phi}_j^m \leftarrow f(\tilde{x}_{j+}) - f(\tilde{x}_{j-})$
 9  $\hat{\phi}_j \leftarrow \frac{1}{M} \sum_{m=1}^{M} \hat{\phi}_j^m$
10  return $\hat{\phi}_j$
```

As an example, we use estimated Shapley values to explain an instance from the iris dataset, which we already described in Section 3.4, as visualized in Figure 4.1. The average predicted class probabilities of the 105 training observations for the three classes `setosa`, `versicolor`, `virginica` from the `species` variable are given by $(0.34, 0.31, 0.35)$ and the prediction of the instance from the test set is $(0.04, 0.78, 0.18)$. The approximated Shapley values that are visualized in Figure 4.1, explain how much each feature contributes to the difference between the actual and the average prediction, which is given by $(-0.30, 0.47, 0.17)$.

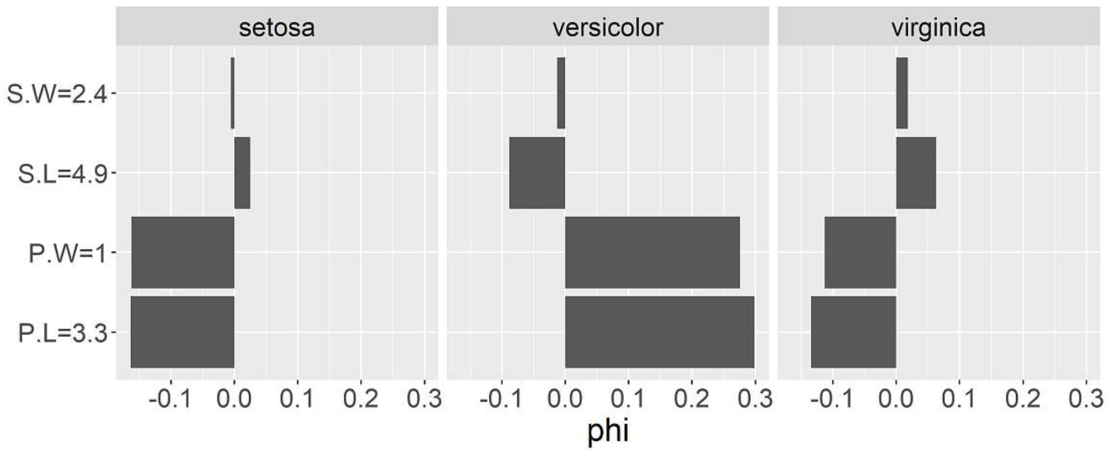


Figure 4.1.: Estimated Shapley values for a single instance from the iris dataset, created with the `Shapley` function from the `R iml` package Molnar et al. (2018), using 10,000 iterations.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4.4. Summary

The concept of explaining ML models using Shapley values yields multiple advantages. Since the Shapley value is based on a solid theory from game theory and we can translate its properties into the context of Explainable AI, we have a solid foundation to create explanations. In particular, the Axiom 4.3.1 (Efficiency) guarantees that we obtain a full explanation of a prediction, which means we can analyze the impact of all features on the difference between the average and the actual prediction. While this is important for gaining a full understanding of the model, using all the information does not provide sparse explanations for high dimensional datasets. In this case we might want to display only a subset of the most influential features in our final explanation to obtain selective results. In comparison to LIME, we do not get a local model, which could be used for predicting similar instances in the neighborhood of the instance of interest. Unfortunately, the biggest drawback of Shapley values is arguably the fact that the computation of the exact Shapley values is very time-consuming, since it constitutes a task of exponential computational complexity. We introduced an approximation method based on Monte-Carlo sampling in Section 4.3.3, which assumes feature independence. Finally, it is worth mentioning that the estimated Shapley values are dependent on the sampling approach (Janzing et al., 2020).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5. SHAP framework

The idea behind SHapley Additive exPlanations (SHAP) is similar to LIME, in the sense that an importance value is assigned to each feature to explain predictions of individual observations. The SHAP framework was introduced in Lundberg and Lee (2017), with one of the novel components of this method being the representation of Shapley values as an additive feature attribution method. This approach combines the ideas of the methods described in the Chapters 3 and 4.

5.1. Additive feature attribution methods

The notation in this chapter is similar to Chapters 3 (Local surrogate models) and 4 (Shapley values). We denote f as the prediction model, g represents the explanation model and x stands for the instance being explained. The simplified input mapping h_x, which relates x' to x, was actually not used in the original paper (Ribeiro et al., 2016), but was instead introduced later in Lundberg and Lee (2017). We chose to introduce it earlier on in this thesis already, since it allows for a clean and more comprehensive notation of the underlying mathematics.

Definition 5.1.1. *(Additive feature attribution method)* An additive feature attribution method is a linear explanation model g with simplified inputs $z' \in \{0,1\}^q$ as features, which can be written as

$$g(z') = \phi_0 + \sum_{j=1}^{q} \phi_j z'_j, \tag{5.1}$$

where $\phi_j \in \mathbb{R}$ is the feature attribution of the j-th feature.

The binary vector of simplified inputs $z' \in \{0,1\}^q$ can be related to the interpretable feature representation in the original formulation of LIME, as described in Section 3.3.1. Moreover, it can also be interpreted as a coalition of features as explained in Chapter 4, where $z'_j = 1$ and $z'_j = 0$ correspond to the presence or absence of the j-th feature in the coalition vector z', respectively.

As in Lundberg and Lee (2017), we introduce Properties 5.1.1 - 5.1.3 that enable us to uniquely determine the feature attributions ϕ_j in Equation (5.1) and relate them to the Shapley values described in Section 4.

Property 5.1.1. *(Local Accuracy)* For the observation of interest $x = h_x(x')$, the explanation model $g(x')$ is equal to the original model $f(x)$:

$$f(x) = g(x') = \phi_0 + \sum_{j=1}^{q} \phi_j x'_j$$

We can now rewrite this property, such that it corresponds to the to the Efficiency axiom of the Shapley value (Axiom 4.3.1):
If the coalition vector is given by x' with $x'_j = 1$ for $j = 1, ..., q$ and we define $\phi_0 = \mathbb{E}[f(x)]$, we obtain

$$f(x) = \phi_0 + \sum_{j=1}^{q} \phi_j x'_j = \mathbb{E}[f(x)] + \sum_{j=1}^{q} \phi'_j.$$

Property 5.1.2. *(Missingness)* Simplified inputs with $x'_j = 0$ are set to have no impact on the prediction:

$$x'_j = 0 \Longrightarrow \phi_j = 0$$

This property is required since the Local Accuracy property is formulated as a linear model and hence x' could have zero entries. If there exist entries in x' with $x'_j = 0$, the Local Accuracy property would hold, regardless of the value of ϕ_j. The Missingness property ensures that features with $x_j = 0$, are assigned a feature attribution of zero. In practice, SHAP does not consider a feature to be missing unless the feature is constant in the whole background dataset (Molnar, 2019).

Property 5.1.3. *(Consistency)* Let $f_x(z') = f(h_x(z'))$ and $z'_{\setminus k}$ indicate that $z'_k = 0$. For any two models f^1 and f^2 that satisfy

$$f_x^2(z') - f_x^2(z'_{\setminus k}) \geq f_x^1(z') - f_x^1(z'_{\setminus k}),$$

for all inputs $z \in \{0, 1\}^q$, it holds that

$$\phi_j(f^2, x) \geq \phi_j(f^1, x).$$

This property can be rewritten, such that it corresponds to the Monotonicity axiom of the Shapley value (Axiom 4.3.5):
Let $Z = \{1, ..., q\}$ be the index set of the simplified features x'. If for two models f^1 and f^2 and all $S \subseteq (Z \setminus \{j\})$ the condition

$$f_x^1(S \cup \{j\}) - f_x^1(S) \geq f_x^2(S \cup \{j\}) - f_x^2(S),$$

holds, then $\phi_j(f^1, x) \geq \phi_j(f^2, x)$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Theorem 5.1.1. *Assuming Feature Anonymity*[1]*, there exists only one explanation model g that follows Definition 5.1.1 and satisfies Properties 5.1.1 - 5.1.3. The feature attributions ϕ_j are given by*[2]

$$\phi_j(f, x) = \sum_{z' \subseteq x'} \frac{(|z'| - 1)!(q - |z'|)!}{q!}(f_x(z') - f_x(z'_{\setminus j})), \tag{5.2}$$

where $|z'|$ is the number of non-zero elements of z' and $z'_{\setminus j}$ means that $z'_j = 0$. Moreover, $z' \subseteq x'$ denotes all binary vectors z' where the non-zero elements are a subset of the non-zero elements of x'.

As we already know from Chapter 4, the values $\phi_j(f, x)$ in Theorem 5.1.1 are known as Shapley values. Before we start with the proof, we note that in Equation (5.2) we are removing the j-th feature, while in Equation (4.7) the j-th feature is added to the feature set, hence the notation of the weights is slightly different. It is worth mentioning that due to this fact, the weights are slightly different in the original publication (Lundberg and Lee, 2017):

$$\frac{|z'|!(q - |z'| - 1)!}{q!} \quad \text{instead of} \quad \frac{(|z'| - 1)!(q - |z'|)!}{q!}.$$

Moreover, Feature Anonymity is implicitly assumed in the proof that is provided in the supplementary material of Lundberg and Lee (2017), so we added this assumption to the statement of Theorem 5.1.1. Both issues, the typo in the weights and the missing Feature Anonymity assumption, are mentioned in the erratum and the latter is extensively discussed in Fisher (2020). Finally, we want to mention that the Anonymity axiom in the context of game theory implies the Symmetry axiom, as stated in Peters (2008).

Proof. We show that the Monotonicity axiom also implies the Symmetry axiom in the setting of models, before demonstrating that Theorem 5.1.1 follows from Theorem 4.3.2. Let us assume that the only difference between f^1 and f^2 is that arbitrary two inputs j and k are interchanged. This implies that for all $S \subseteq (Z \setminus \{j, k\})$ we have

$$f_x^1(S) = f_x^2(S) \quad \text{and} \quad f_x^1(S \cup \{j\}) = f_x^2(S \cup \{k\}). \tag{5.3}$$

In the remainder of the proof, we can ignore the case that S contains k, because in these cases we know that $f_x^1(S \setminus \{k\} \cup \{j, k\}) = f_x^2(S \setminus \{k\} \cup \{j, k\})$. First, we transform Axiom 4.3.5

$$\forall S \subseteq (Z \setminus \{j\}) \quad f_x^2(S \cup \{j\}) - f_x^2(S) \geq f_x^1(S \cup \{j\}) - f_x^1(S) \Longrightarrow \phi_j(f^2, x) \geq \phi_j(f^1, x),$$

using Equation (5.3), to obtain

$$\forall S \subseteq (Z \setminus \{j, k\}) \quad f_x(S \cup \{j\}) \geq f_x^1(S \cup \{k\}) \Longrightarrow \phi_j(f^2, x) \geq \phi_j(f^1, x).$$

[1]Feature Anonymity implies that the feature names do not influence their assigned credit. While this assumption is not mentioned in the original formulation of the theorem in Lundberg and Lee (2017), its added in the erratum.

[2]Here we assume that $|x'| = q$, for a more general formulation we could replace q with $|x'|$ to allow for zeros in x'. An alternative proof that works without the Missingness axiom is given in Fisher (2020).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

When we switch j and k and repeat the same rational, the Symmetry axiom follows if we assume Feature Anonymity:

$$\forall S \subseteq (Z \setminus \{j,k\}) \quad f_x^2(S \cup \{j\}) = f_x^1(S \cup \{k\}) \Longrightarrow \phi_j(f^2, x) = \phi_j(f^1, x).$$

□

According to Theorem 5.1.1 we can uniquely determine the feature attributions ϕ_j for an additive feature attribution method, which satisfies the properties Local Accuracy, Missingness, and Consistency for a given simplified input mapping h_x. This implies that other additive feature attribution methods like LIME are not optimal with respect to those properties. We will now introduce SHapley Additive exPlanation (SHAP), which allow us to unify multiple feature attribution methods, including LIME and Shapley values.

To get SHAP values as described in Lundberg and Lee (2017), we choose the simplified input mapping as $h_x(z') = z_S$, where S is the subset of non-zero indices from z' and z_S has missing values for features that are not contained in S. To account for the fact that most models are not able to handle missing values, $f(z_S)$ is approximated by $\mathbb{E}[f(z)|z_S]$, therefore we get

$$f_x(z') = f(h_x(z')) = \mathbb{E}[f(z)|z_S]. \tag{5.4}$$

Using this setup yields the SHAP values, which satisfy Properties 5.1.1 - 5.1.3, as unique solutions of Equation (5.2) and we can write them as

$$\phi_j(f,x) = \sum_{z' \subseteq x'} \frac{(|z'|-1)!(q-|z'|)!}{q!}(\mathbb{E}[f(z)|z_S] - \mathbb{E}[f(z)|z_{S \setminus j}]). \tag{5.5}$$

We want to mention that Theorem 4.3.3 and Theorem 5.1.1 are both based on the Shapley value and both methods use conditional expectations to model the influence of the features. Due to this fact, the SHAP values in Equation (5.5) and the marginal feature value contributions in Equation (4.7) yield the same solutions. As already discussed for the marginal feature value contributions in Chapter 4, the exact computation of the SHAP values in Equation (5.5) is computationally very expensive. Thus, we will now introduce the model agnostic approximation method proposed in Lundberg and Lee (2017).

For an efficient approximation, the conditional expectation in Equation (5.4) can be simplified when we introduce the following assumptions:

$$\begin{aligned} f_x(z') = f(h_x(z')) &= \mathbb{E}[f(z)|z_S] && \text{simplified input mapping for SHAP} && (5.6)\\ &= \mathbb{E}_{z_{\bar{S}}|z_S}[f(z)] && \text{expectation over } z_{\bar{S}}|z_S && (5.7)\\ &\approx \mathbb{E}_{z_{\bar{S}}}[f(z)] && \text{assuming feature independence} && (5.8)\\ &\approx f((z_S, \mathbb{E}[z_{\bar{S}}])) && \text{assuming model linearity} && (5.9) \end{aligned}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5.2. Kernel SHAP – model agnostic approximation

The approximation procedure proposed in Lundberg and Lee (2017) can be related to LIME and it is called **Kernel SHAP**. Considering additive feature attribution methods, we can uniquely determine the solution of the LIME objective function given in Equation (3.1), under the condition that it adheres to Properties 5.1.1 - 5.1.3, when we use a linear model to create the explanations. To achieve this goal, the loss function $\mathcal{L}$, the proximity measure $\pi_{x'}$ and the regularization term Ω must be chosen according to the description given in Theorem 5.2.1. Here we note that, in contrast to the original introduction of LIME, the proximity measure $\pi_{x'}$ operates on the simplified input x' and not on the original observation x.

Theorem 5.2.1. *Considering the additive feature attribution method LIME with the objective function*

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \Big\{ \mathcal{L}(f, g, \pi_x) + \Omega(g) \Big\}, \qquad \text{(3.1, revisited)}$$

the only choices of π_x, Ω and $\mathcal{L}$ without neglecting the boundaries imposed by Properties 5.1.1 - 5.1.3 are[3]

$$\pi_{x'}(z') = \frac{q-1}{\binom{q}{|z'|} |z'| (q - |z'|)}, \tag{5.10}$$

$$\Omega(g) = 0, \tag{5.11}$$

$$\mathcal{L}(f, g, \pi_{x'}) = \sum_{z' \in Z} \left(f(h_x^{-1}(z')) - g(z') \right)^2 \pi_{x'}(z'), \tag{5.12}$$

where $|z'|$ is the number of non-zero elements in z' and Z are the training data.

Proof. We define the proximity measure (Shapley kernel) $\pi_{x'}$ from Equation (5.10) for a simplified input vector $z' \in \{0,1\}^q$ as

$$\pi_{x'}(z') = k(q,s) = \begin{cases} \frac{q-1}{\binom{q}{s} s(q-s)} & \text{for } s \in \{1, 2, ..., q-1\} \\ c & \text{for } s \in \{0, q\} \end{cases}, \tag{5.13}$$

where $s = \sum_{j=1}^{q} z'_j$ is the number of ones in z', and $c \in \mathbb{R}^+$ is a large constant. The distinction of cases is necessary[4] because the fraction in Equation (5.13) is not well defined for $s \in \{0, q\}$.

In the next step we compute the Shapley values with weighted linear regression using the Shapley kernel. We consider the matrix $X \in \{0,1\}^{2^q \times q}$ of all possible binary vectors $X_{i\cdot} \in \{0,1\}^q$ and let $s_i = \sum_{j=1}^{q} X_{i\cdot} \in \{0, 1, ..., q\}$ denote the number of ones in $X_{i\cdot}$. The

[3] Like in Theorem 5.1.1, we implicitly assume that $|x'| = q$. If we allow zero entries of x', we have to reduce the dimension of $\{0,1\}^q$ to $\{0,1\}^{|x'|}$, which results in omitting the features with zero indices, as is also mentioned in Jia (2020).

[4] From an analytical point of view, we could also avoid this by introducing the constraints $\phi_0 = f_x(\emptyset)$ and $f(x) = \sum_{j=0}^{q} \phi_i$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

diagonal weight matrix W consists of the Shapley kernels for each row of X and is given by

$$W = \text{diag}(k(q, s_1), k(q, s_2), ..., k(q, s_{2^q})) \in \mathbb{R}^{2^q \times 2^q}.$$

The dependent variable y contains the predictions of f for each row of X, so each element is given by $y_i = f_x(X_{i.})$. Let us consider the weighted linear regression problem

$$\underset{\phi}{\text{argmin}} \left\| W^{1/2}(y - X\phi) \right\|^2,$$

for which the parameter estimates are given by

$$\hat{\phi} = (X^T W X)^{-1} X^T W y. \tag{5.14}$$

Moving on, we want to rewrite the terms in Equation (5.14), beginning with analyzing one component of $X^T W$:

$$(X^T W)_{i,j} = \begin{cases} k(q, s_i) & \text{for } X_{i,j} = 1 \\ 0 & \text{otherwise} \end{cases}.$$

This enables us to compute any element of the $q \times q$-dimensional matrix $X^T W X$ as

$$(X^T W X)_{i,j} = (X^T W)_{i.} X_{.,j} = \sum_{l=1}^{2^q} k(q, s_l) X_{l,i} X_{l,j}. \tag{5.15}$$

We note that for all $X_{l.}$ and $X_{k.}$, where $s_l = s_k = s$ it holds that $k(q, s_l) = k(q, s_k) = k(q, s)$. This means the Shapley kernel only depends on the number of non-zero elements of $X_{i.}$ and without loss of generality, we assume that the rows of X are ordered with respect to s_i. This means we can rewrite Equation (5.15) as

$$(X^T W X)_{i,j} = \sum_{s=0}^{q} n_s(i, j) k(q, s), \tag{5.16}$$

where $n_s(i, j)$ is the cardinality of the set $\{X_{l.} | s_i = s, X_{l,i} = X_{l,j} = 1\}$, which is given by

$$n_s(i, j) = \begin{cases} \binom{q-1}{s-1} & \text{for } i = j \\ \binom{q-2}{s-2} & \text{for } i \neq j \end{cases}. \tag{5.17}$$

For the case of $k < 0$ we set $\binom{n}{k} = 0$, in the case that $i = j$ we get $n_0 = 0$ and if $i \neq j$ we obtain $n_0 = n_1 = 0$. Moreover, for a given s, $n_s(i, j)$ does not depend on the specific values of i and j. Combining equations (5.16) and (5.17) yields

$$(X^T W X)_{i,j} = \begin{cases} \sum_{s=1}^{q} \binom{q-1}{s-1} k(q, s) & \text{for } i = j \\ \sum_{s=2}^{q} \binom{q-2}{s-2} k(q, s) & \text{for } i \neq j \end{cases}. \tag{5.18}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

In the case that $1 \leq s < q$ we get that

$$\binom{q-1}{s-1} k(q,s) = \binom{q-1}{s-1} \frac{q-1}{\binom{q}{s} s(q-s)} = \frac{q-1}{q(q-s)}, \tag{5.19}$$
$$\binom{q-2}{s-2} k(q,s) = \binom{q-2}{s-2} \frac{q-1}{\binom{q}{s} s(q-s)} = \frac{s-1}{q(q-s)}, \tag{5.20}$$

and in case of $s = q$ we obtain

$$\binom{q-1}{s-1} k(q,s) = \binom{q-2}{s-2} k(q,s) = c. \tag{5.21}$$

As mentioned previously, the entries of $X^T W X$ given in Equation (5.18) do not depend on the specific values of i and j. Therefore, $X^T W X$ can be written as

$$X^T W X = \alpha_1 I_q + \alpha_2 J_q, \tag{5.22}$$

where $\alpha_1, \alpha_2 \in \mathbb{R}$ are constants, I_q is the q-dimensional identity matrix and J_q is a matrix consisting only of ones with dimension q. To determine α_1 and α_2, we compute the difference between the diagonal and off-diagonal elements of $X^T W X$:

$$\sum_{s=1}^{q} \binom{q-1}{s-1} k(q,s) - \sum_{s=2}^{q} \binom{q-2}{s-2} k(q,s) \tag{5.23}$$
$$= \sum_{s=1}^{q-1} \frac{q-1}{q(q-s)} - \sum_{s=2}^{q-1} \frac{s-1}{q(q-s)} \tag{5.24}$$
$$= \frac{q-1}{q(q-1)} + \sum_{s=2}^{q-1} \left(\frac{q-1}{q(q-s)} - \frac{s-1}{q(q-s)} \right) \tag{5.25}$$
$$= \frac{1}{q} + \sum_{s=2}^{q-1} \frac{1}{q} = \frac{q-1}{q}. \tag{5.26}$$

In Equation (5.24) we only sum up to $q-1$ because the term of the sum for $s = q$ is equal to c in both sums. With the above calculations we obtain the coefficient for the diagonal elements as $\alpha_1 = \frac{q-1}{q}$ and the coefficient for the off-diagonal elements can be written as

$$\alpha_2 = \sum_{s=2}^{q} \binom{q-2}{s-2} k(q,s) = \sum_{s=2}^{q-1} \frac{s-1}{q(q-s)} + c. \tag{5.27}$$

If $c >> 0$, one can see from Equation (5.27) that c is dominating the other terms and we can simplify $X^T W X$ to[5]

$$X^T W X = \frac{q-1}{q} I_q + \alpha_2 J_q. \tag{5.28}$$

[5]In the proof given in the supplementary material of Lundberg and Lee (2017), the factor of for the identity matrix I_p in Equation (5.28) is incorrect, since it is given by $1/(q-1)$. Nevertheless, the asymptotic inverse is computed correctly. While this fact is not addressed in the erratum, it was already mentioned in Jia (2020).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Next, we show that for $q \to \infty$ the inverse of X^TWX approaches $\frac{q}{q-1}I_p - \frac{1}{q-1}J_q$ and does not depend on α_2:

$$\left(\frac{q-1}{q}I_q + \alpha_2 J_q\right)\left(\frac{q}{q-1}I_p - \frac{1}{q-1}J_q\right) \tag{5.29}$$

$$= \frac{(q-1)q}{(q-1)q}I_q - \frac{q-1}{q(q-1)}J_q + \alpha_2\frac{q}{q-1}J_q - \alpha_2\frac{1}{q-1}\underbrace{(J_q)^2}_{=qJ_q} \tag{5.30}$$

$$= I_q - \frac{1}{q}J_q \xrightarrow{q\to\infty} = I_p. \tag{5.31}$$

In the following we simply denote the asymptotic inverse of X^TWX as $(X^TWX)^{-1}$ and it is given by

$$(X^TWX)^{-1} \underset{q\to\infty}{\longrightarrow} \frac{q}{q-1}I_p - \frac{1}{q-1}J_q. \tag{5.32}$$

If we multiply X^TW by $(X^TWX)^{-1}$, we get the matrix $\tilde{W} \in \mathbb{R}^{q\times 2^q}$ containing the weights, which are applied to y for the computation of $\hat{\phi}$:

$$\begin{aligned}
\tilde{W}_{j,i} &= ((X^TWX)^{-1}X^TW)_{j,i} = \left(\frac{q}{q-1}J_p - \frac{1}{q-1}I_q\right)_{j.}(k(q,s_i)X_{i.}) \\
&= k(q,s_i)\left(\frac{q}{q-1}X_{i,j} - \frac{1}{q-1}s_i\right) = \frac{q-1}{\binom{q}{s_i}s_i(q-s_i)}\left(\frac{q}{q-1}X_{i,j} - \frac{1}{q-1}s_i\right) \\
&= \frac{(qX_{i,j}-s_i)(s_i!(q-s_i)!)}{q!s_i(q-s_i)} = \frac{(qX_{i,j}-s_i)((s_i-1)!(q-s_i-1)!)}{q!} \\
&= \begin{cases} -\frac{(s_i)!(q-s_i-1)!}{q!} & \text{for } X_{i,j}=0 \\ \frac{(s_i-1)!(q-s_i)!}{q!} & \text{for } X_{i,j}=1 \end{cases}.
\end{aligned} \tag{5.33}$$

Let us consider the binary vector $X_{i.}$ where $X_{i,j}=1$ and let $(X_{i.})_{\setminus j}$ denote the binary vector that is identical to $X_{i.}$, except that the j-th component is set to zero. Using Equation (5.33), multiplying $\tilde{W}_{j,i}$ with y_i results in

$$\tilde{W}_{j,i}y_i = \frac{(s_i-1)!(q-s_i)!}{q!}f_x(X_{i.}) \quad \text{and} \quad \tilde{W}_{j,i}y_i = -\frac{(s_i-1)!(q-s_i)!}{q!}f_x((X_{i.})_{\setminus j}).$$

To finally compute $\hat{\phi}_j$, we sum up all contributions $\tilde{W}_{j,i}y_i$ to obtain

$$\phi_j = \tilde{W}_{j.}y = \sum_{X_{i.}\in\{0,1\}^q|X_{i,j}=1}\frac{(s_i-1)!(q-s_i)!}{q!}(f_x(X_{i.}) - f_x((X_{i.})_{\setminus j})) \tag{5.34}$$

$$= \sum_{z'\subseteq x'}\frac{(|z'|-1)!(q-|z'|)!}{q!}(f_x(z') - f_x(z'_{\setminus j})). \tag{5.35}$$

The equality of Equation (5.34) and (5.35) holds, since we can include all subsets of x' (and consequently all rows of X) in the sum without affecting the result, because instances with

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

the j-th entry equal to zero have no impact on the sum. The expression in Equation (5.35) corresponds to the classic form of the Shapley value, showing that the coefficients of the weighted linear model approximate the Shapley value.

$\square$

The Kernel SHAP estimates of the Shapley value can be computed using Algorithm 1, with the exception that the proximity measure $\pi_{x'}$ operates on the space of simplified inputs X' and the model complexity Ω is not restricted. The simplified input mapping h_x relates the coalitions z' to the original feature space. In the case of tabular data, instead of sampling numerical features from a normal distribution as with LIME (Section 3.3.1), they are sampled directly from the dataset. As a result, we are sampling from the marginal distribution, which implies that we are ignoring correlations between features. This may lead to the inclusion of unrealistic instances and the results can get unreliable. As mentioned in Molnar (2019), this issue could be avoided by sampling from the conditional distribution. However, such an approach has the disadvantage that the value function of the corresponding game and therefore also the Shapley value is altered. Moreover, it is possible to intentionally hide biases in explanations, when using the SHAP framework to interpret a ML model, as shown in (Slack et al., 2020).

When we compare the proximity functions of LIME and SHAP, we observe weights that are assigned to the sampled instances are inherently different. While the exponential kernel used in LIME weighs instance according to their proximity to the observation of interest, the proximity measure used in Kernel SHAP attributes the weights to instances depending on their interpretable representation. When we use the exponential kernel in LIME, increasing the number of ones in z' leads to a higher weight of z.[6] According to Equation (5.10), coalitions z' containing either almost exclusively or almost no ones get assigned the highest weights. Interestingly, we can use this fact to improve the sampling strategy, as mentioned in Molnar (2019). We should begin our sampling strategy by choosing the coalitions z' with $|z'| \in \{1, q-1\}$, since those result in the highest weights and contribute the most to the approximation of the Shapley value. If the sample size allows for more samples, we continue this approach, sampling coalitions where $|z'| \in \{2, q-2\}$ and so on.

5.3. Linear SHAP

We already introduced a model specific approximation procedure for Shapley values of linear regression models in Section 4.2. Here we relate it to the context of an additive feature attribution method (Definition 5.1.1). To achieve this goal, we set $\phi_0 = \beta_0 + \sum_{j=1}^{p} \beta_j \mathbb{E}[x_j]$ and $\phi_j = \beta_j(x_j - \mathbb{E}[X_j])$, as in Equation (4.1). In this manner the Shapley values are directly approximated using the parameters β_j of the linear model.

[6]For LIME we can not directly connect the interpretable feature representations to the weights, since the weights are calculated on the original feature space. Nevertheless, it is more likely that an observation $z = h_x(z')$ is similar to x if z' contains many ones.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5.4. Tree SHAP

The last approximation procedure we want to discuss in this work can be applied to tree-based methods. We want to mention that when this method was first introduced in Lundberg et al. (2018), it relied on conditional expectations to compute $f_x(z')$, as in Equation (5.6). With this approach, the resulting SHAP values failed to satisfy Property 5.1.2 (Missingness), as described in Janzing et al. (2020). The algorithms have since been updated and currently use the *interventional conditional expectation* (Lundberg et al., 2020; Chen et al., 2018), which corresponds to Equation (5.9). This also shows that the simplifying assumptions for Kernel SHAP lead to unbiased results.

Here we present a naive algorithm, as discussed in Lundberg et al. (2020), to provide an introduction into the calculation of Shapley values for trees. Subsequently, we explain the idea behind a more efficient but also more complex algorithm. An interactive explanation and visualization for tree-based algorithms can be found in the article by Chen et al. (2018).

Disregarding the computational complexity of the task, we can approximate the Shapley values by first estimating $f_x(z') = f_x(S)$ for all 2^p possible feature value combinations and then using Equation (5.2) for the computation. In Algorithm 3 we recap the approximation approach for $f_x(S)$, which is based on a path-dependent perturbation of the features. For this algorithm we need to supply the information of the tree we want to explain, which we can summarize as follows:

- Let v be a vector of node values, where internal nodes of the tree are assigned the value `internal`.
- We use the vectors a and b to denote the left and right indices of all internal nodes.
- The thresholds for each internal node are stored in the vector t.
- In the vector d we store the indices of the features used for splitting.
- Let r represent the vector that stores the information of how many samples fall in a subtree for each node.

In Algorithm 3 we define the recursive procedure $G(j)$, which works as follows: In the first step, we check whether the j-th node is a leaf node or not. Let us consider the case that it is a leaf node, then we return its value v_j and the routine is finished. If it is an internal node of the tree, we follow the decision path of the instance x if the node is contained in S, and we compute the weighted average of both branches if it is not included in S. We estimate $f_x(S)$ by starting this recursive algorithm in the first node of the tree.

The computation time of the procedure summarized in Algorithm 3 depends on the number of leaves l in the tree, which means that in combination with the computation of Equation (5.2) we get a complexity of $\mathcal{O}(lp2^p)$. If we explain a tree ensemble model with T trees and a maximum number of L leaves in any tree, then the complexity is $\mathcal{O}(TLp2^p)$.

The more sophisticated, but also more efficient **Tree SHAP** algorithm reduces the exponential computational complexity of the approach described above to $\mathcal{O}(TLD^2)$, where D is the maximum depth of any tree. The conceptual idea of this approach is to monitor

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Algorithm 3: Estimate $f_x(S)$ for trees.

```
   Input: Observation to be explained $x$, subset of features $S$, Tree = {v,a,b,t,r,d}
   Output: Approximation of $f_x(S)$
 1 Function G($j$):
 2     if $v_j \neq internal$ then
 3         return $v_j$
 4     else
 5         if $d_j \in S \wedge x_{d_j} \leq t_j$ then
 6             return $G(a_j)$
 7         else if $d_j \in S \wedge x_{d_j} > t_j$ then
 8             return $G(b_j)$
 9         else
10             return $\frac{G(b_j)r_{a_j}+G(b_j)r_{a_j}}{r_j}$
11 return $G(1)$
```

what proportion of all possible subsets S arrives at each of the leaves of the tree. For more details on this approach, we refer to the literature (Lundberg et al., 2018, 2020).

5.5. Implementations, visualization and global feature importance

Along with the introduction of the SHAP framework in Lundberg and Lee (2017), the authors also provide a `Python` implementation of their methods in the `shap` package. It does not only contain all the methods we discussed in this chapter, but also includes methods for explaining models for image classification or to cluster observations based on Shapley values, and it provides multiple visualization tools. In `R` we can estimate Shapley values, for example by using the previously mentioned `iml` package (Molnar et al., 2018), the `fastshap` package (Greenwell, 2020b), or the `xgboost` package (Chen and Guestrin, 2016; Chen et al., 2021), the latter of which includes an efficient version of the Tree SHAP algorithm.

In this section we discuss different visualization tools for Shapley values, using the iris dataset, which is described in Section 3.4. We work with the `xgboost` package to build a boosted tree ensemble model to predict the variable `species` ∈ {`setosa`, `versicolor`, `virginica`}, using the other variables as predictors. Moreover, we estimate the Shapley values with the same package. Subsequently, we create plots to either explain the predicted probability for one class using Shapley values, or we aggregate the Shapley values of all classes to analyze the total impact of the explanatory variables on the response. The plots and graphs displayed in this section were created based on the visualizations published in the original papers (Lundberg and Lee, 2017; Lundberg et al., 2018, 2020).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5.5.1. Visualizations for individual observations

Before we describe and analyze the plots in the following paragraphs, let us recap the interpretation of the Shapley value:

For a given set of feature values, the Shapley value is the difference between the actual prediction and the average prediction.

When we use the `R xgboost` package to derive the Shapley values for multinomial classification problems, then they are *on the scale of the untransformed margin* (Chen et al., 2021). Hence, Shapley values do not represent the change in probability in this case. Nevertheless, they still explain how much the actual prediction differs from the average prediction, just not in terms of probabilities. Moreover, we can either explain each class individually, or we can sum up the importance scores of all factor levels to analyze the influence across all classes.

In Figure 5.1 we see the Shapley values of two different observations from the iris dataset. We list the feature values of the instance of interest on the y-axis, and the x-axis denotes the Shapley values. Both plots in Figure 5.1 show an explanation for `species` = `versicolor`. For the first instance (left panel) the differences between the actual class probabilities of the instance of interest and the average probabilities on the training data are given by

$$\begin{pmatrix} \Delta_{\texttt{setosa}} \\ \Delta_{\texttt{versicolor}} \\ \Delta_{\texttt{virginica}} \end{pmatrix} = \begin{pmatrix} 0.6 \\ -0.28 \\ -0.33 \end{pmatrix} = \begin{pmatrix} -0.95 \\ -0.03 \\ 0.02 \end{pmatrix} - \begin{pmatrix} 0.35 \\ 0.31 \\ 0.35 \end{pmatrix},$$

while they are given by

$$\begin{pmatrix} \Delta_{\texttt{setosa}} \\ \Delta_{\texttt{versicolor}} \\ \Delta_{\texttt{virginica}} \end{pmatrix} = \begin{pmatrix} -0.26 \\ 0.51 \\ -0.26 \end{pmatrix} = \begin{pmatrix} 0.09 \\ 0.82 \\ 0.09 \end{pmatrix} - \begin{pmatrix} 0.35 \\ 0.31 \\ 0.35 \end{pmatrix},$$

for the second instance (right panel). For `species` = `versicolor` we have to explain a negative difference of −0.28 for the first observation, whereas for the second instance we have a positive difference of 0.51. While `Petal.Length` = 1.4 has a negative influence on the difference between the predicted and the actual class probability for the first instance and explains almost the entire prediction, we do not only observe a positive influence of `Petal.Length` = 3.5 for the second observation but also see that the other features contribute more to the explanation. We can conclude that for both observations `Petal.Length` is the most influential feature when `species` = `versicolor`.

To get an overview of the influence of the feature values over all three `species`, we present the sum of the absolute Shapley values on the x-axis in Figure 5.2. Here we see that `Petal.Length` yields the contribution to explain the difference between the actual prediction and the mean prediction among the four features. When we compare the plots for both observations, we notice that for the first observation (left panel) the contribution of `Petal.Length` is stronger compared to the influence of the other features, than for the second observation (right panel).

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

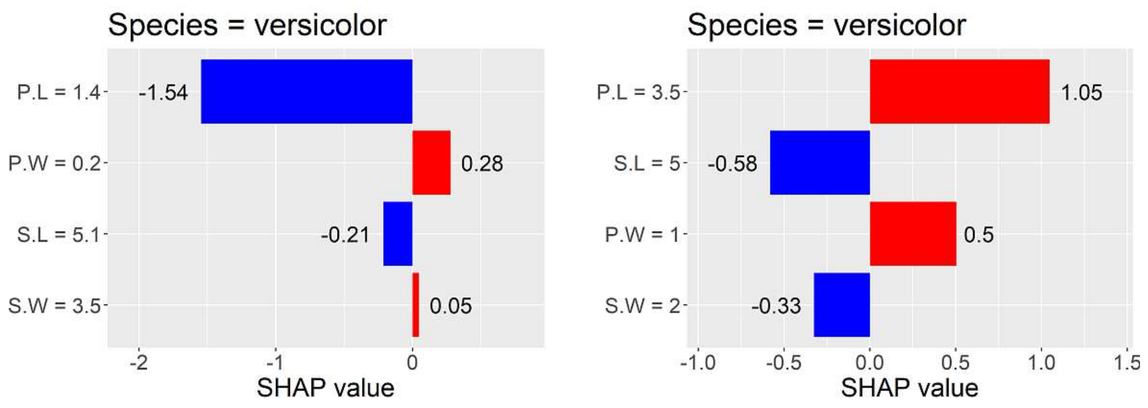


Figure 5.1.: Shapley values for two instances where `species` = `versicolor`.

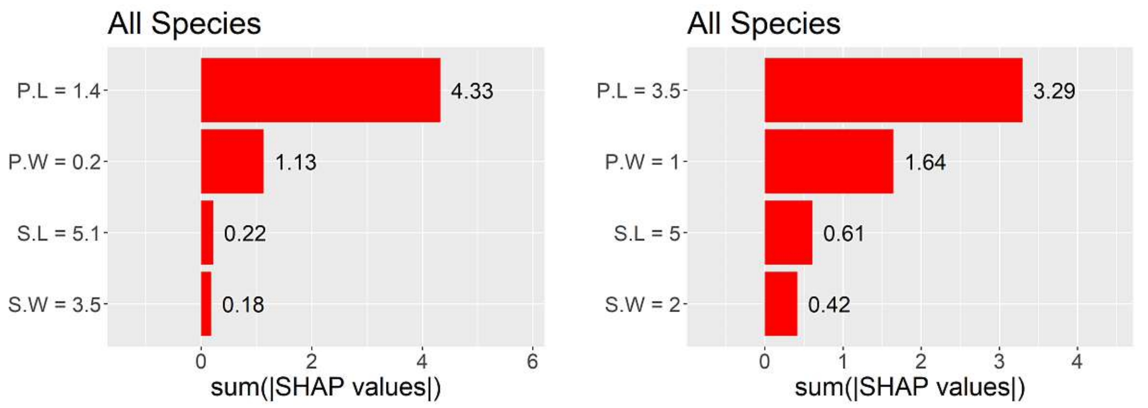


Figure 5.2.: Sum of the absolute Shapley values over all `species` for two observations.

5.5.2. Global feature importance

Having already introduced the theoretical background and the estimation procedures in Chapters 4 and 5, we can explain individual predictions using Shapley values. If we compute the Shapley value for every instance of our dataset, we can aggregate the results to analyze and interpret the behavior of the entire model. The global importance of the j-th feature with respect to the Shapley values is defined as

$$I_j = \sum_{i=1}^{n} \left| \phi_j^{(i)} \right|, \tag{5.36}$$

where $\phi_j^{(i)}$ is the j-th coordinate of the Shapley value $\phi^{(i)}$ from the i-th observation. Since Property 5.1.1 (Local Accuracy) and Axiom 4.3.1 (Efficiency) imply that the magnitude of the components of the Shapley value depends on the model f, we are interested in comparisons of the relative ratios of the Shapley value. Therefore, features with a comparably high absolute Shapley value are considered to be important or influential in the model.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

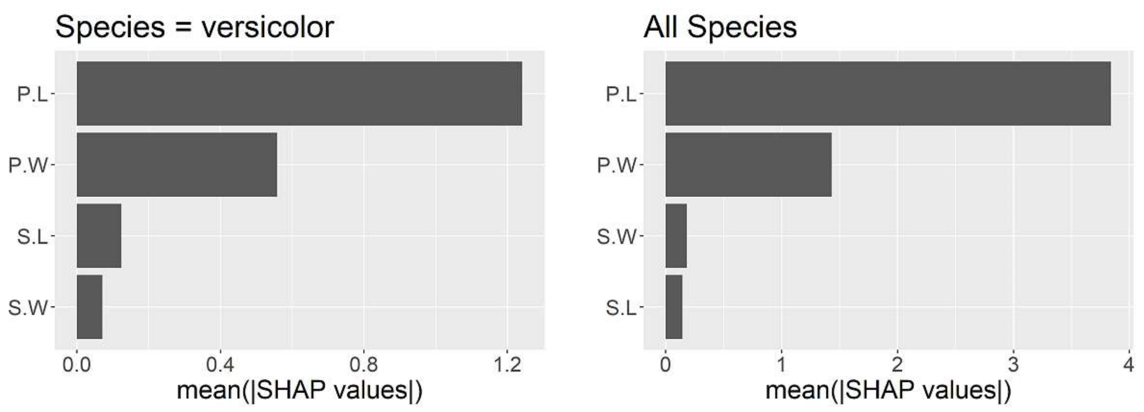


Figure 5.3.: Global feature importance based on the mean absolute Shapley value, for `species` = `versicolor` (left) and summed over all three `species` (right).

The global importance with respect to the Shapley values can be summarized in a bar plot, as displayed in Figure 5.3. In the two plots, the features are listed on the y-axis, sorted according to their global importance I_j. On the x-axis we display the mean absolute Shapley value as a measure of global variable importance. In this example we aggregated over all 45 observations in the test set, and we observe the same results as for the two observations we considered in Section 5.5.1. It is worth noting, that the global feature importance based on Shapley values can serve as an alternative to permutation feature importance for random forests (Hastie et al., 2009), or the model agnostic version called model reliance, introduced in Fisher et al. (2019).

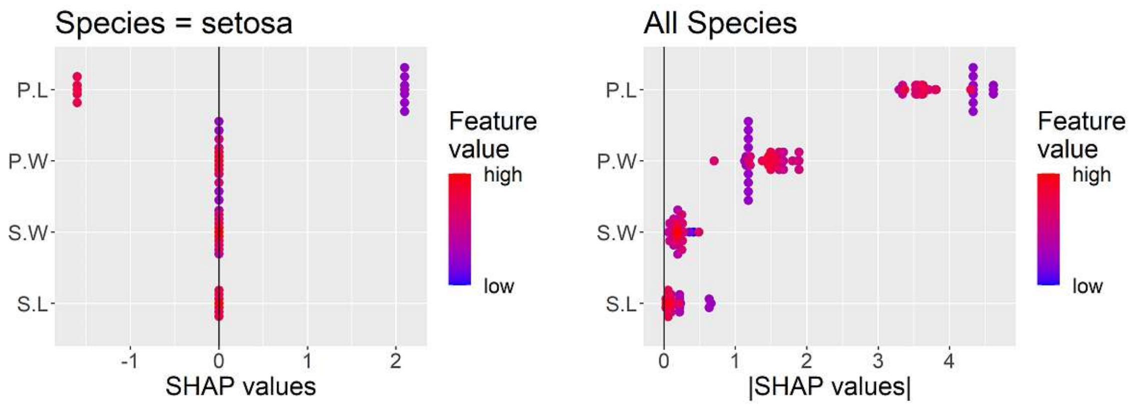


Figure 5.4.: The SHAP Summary Plots allow us to summarize the information of the Shapley values and the corresponding feature values from all the observations into one plot. In the left panel we see the Shapley values for `species` = `setosa` and in the right panel we see the sum of the absolute Shapley values for all `species`.

Since the Shapley values are calculated for each observation, we are not limited to display the global variable importance, but we can add more details to the plot, as shown in the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

SHAP Summary Plot (Lundberg et al., 2018). In Figure 5.4 we combine the feature importance with the feature effect as follows: First, the features are sorted by their global importance I_j and listed on the y-axis accordingly. In the next step, dots that represent the Shapley values $\phi_j^{(i)}$ are plotted on the x-axis and stacked vertically when they overlap, to visualize the distribution of the Shapley values for each feature. Finally, those dots are colored with respect to their standardized feature values, from low (blue) to high (red). In the left panel of Figure 5.4 we can observe, that out of the four features, only `Petal.Length` is influential for `species` = `setosa`. In contrast to this result we see in the right panel that the remaining features are also contributing when we aggregate over all three `species`.

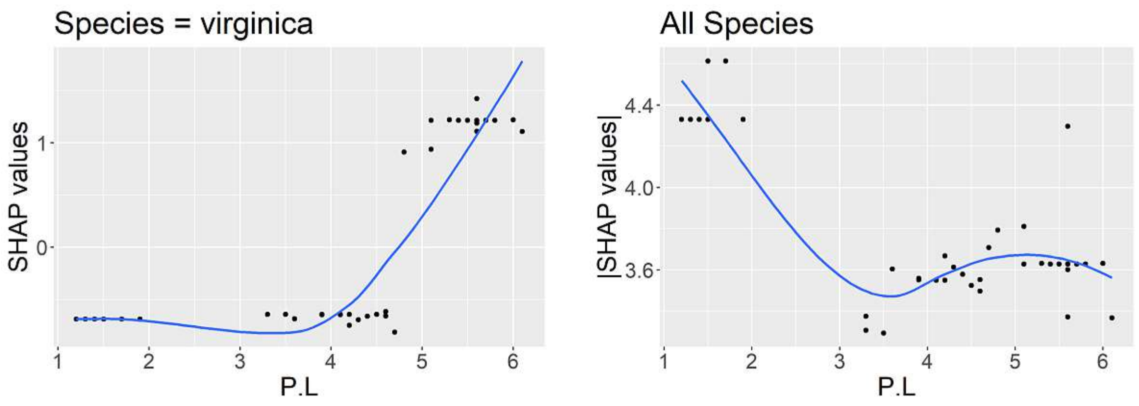


Figure 5.5.: With the SHAP Dependence Plot we can analyze the relationship between the feature values and the Shapley values. The plot on the left-hand side shows this relation for `Petal.Length` when `species` = `virginica`, and on the right-hand side we see the connection between the sum of the absolute Shapley values for all `species` and the feature values of `Petal.Length`.

Finally, the **SHAP Dependence Plot** (Lundberg et al., 2018) allows us to analyze the relationship between the feature values and the corresponding Shapley values for a single feature over all observations, as visualized in Figure 5.5. We are plotting the values of the feature on the x-axis and the associated Shapley values on the y-axis, which allows us to investigate how the influence on the prediction changes as the feature values vary. When we inspect the plot in the left panel of Figure 5.5, where we consider the case that `species` = `virginica`, we see that the Shapley values are negative up to the threshold of `Petal.Length` ≈ 4.7, and for feature values that are larger than this threshold, they are positive. Furthermore, comparing the left and right panel, we observe that as `Petal.Length` is increasing, the magnitude of the absolute Shapley values is also rising when `species` = `virginica`, while larger values of `Petal.Length` lead to a reduction of the absolute Shapley values when we consider the aggregated results over all three `species`, displayed in the right panel.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5.6. Summary

Since feature attributions generated using the SHAP framework correspond to the Shapley values discussed in Chapter 4, they inherit all the advantages of the Shapley values. Moreover, the Kernel SHAP method connects LIME and Shapley values, which helps us to solve the problem of the uncertainty involved in the choice of the neighborhood when we use LIME: Theorem 5.2.1 provides the theoretical background on how we should define the neighborhood of the instance of interest. In addition to the local explanations, we can aggregate Shapley values to gain a global understanding of the model, if we compute them for a large number of instances. However, this might lead to long computation times for the model-agnostic estimation procedures. Fortunately, the Tree SHAP algorithm provides an efficient approximation method for tree-based models. Finally, we want to mention, that the SHAP frameworks also suffers from the drawback that we have to choose a sampling strategy for the computation of the Shapley values and the feature attributions can be manipulated to hide biases (Slack et al., 2020).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

6. Electricity markets and data description

Life as we know it would not be even remotely possible without the access to electricity – and if the current state of the world economy and lifestyle is any indication, our demand will only increase (Lewis and Nocera, 2006). What is more, factors like climate change, price and decreasing access to the remaining fossil fuels (Mcglade and Ekins, 2015) have been the main drivers for the research and development of renewable energy sources regarding our mobility factors (cars, trains etc), which means the energy market is on the brink of receiving an even larger amount of interest.

While transitioning from an energy market that is powered by fossil resources to a sustainable and non-polluting energy generation is of uttermost importance, market integration and maximizing the full potential of renewable energy sources while preserving supply security and the physical grid's reliability presents new problems. Not only are wind and solar parks spatially dispersed and often located in remote areas further away from large consumption centers, but we also have to consider the difficulty of accurately forecasting their electrical output, or other technical obstacles when compared to fossil fuels. As a result, ensuring system stability with high shares of renewable energy sources significantly changes the market, product, process, and coordination schemes (ENTSO-E, 2021a). To lead life as we know it, we depend on a stable power supply that is resistant towards blackouts and system failures. To ensure this, the European energy market is coordinated by the European Network of Transmission System Operators for Electricity (ENTSO-E). It is an organization responsible for the cooperation between the European transmission system operators (TSOs), which currently encompasses 42 members representing 35 countries. According to their website (ENTSO-E, 2021a), the objective of ENTSO-E can be summarized as follows: *Ensuring the security of the interconnected power system in all time frames at pan-European level and the optimal functioning and development of the European interconnected electricity markets, while enabling the integration of electricity generated from renewable energy sources and of emerging technologies.*

In this work we focus on the Austrian energy market, which is a part of the European power grid – the largest interconnected electrical grid in the world (ENTSO-E, 2021a). We want to provide a broad outline of electricity markets, thereby introducing the ENTSO-E Transparency Platform (TP) (ENTSO-E, 2021b), which we use as source for the electricity market data that is analyzed in this thesis. With the collected information we can create various datasets and pursue multiple modeling strategies, as outlined in Section 6.1 and Chapter 7, respectively. Afterwards, we apply and adapt the methods of Explainable AI to the final online modeling approach in Section 7.3.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

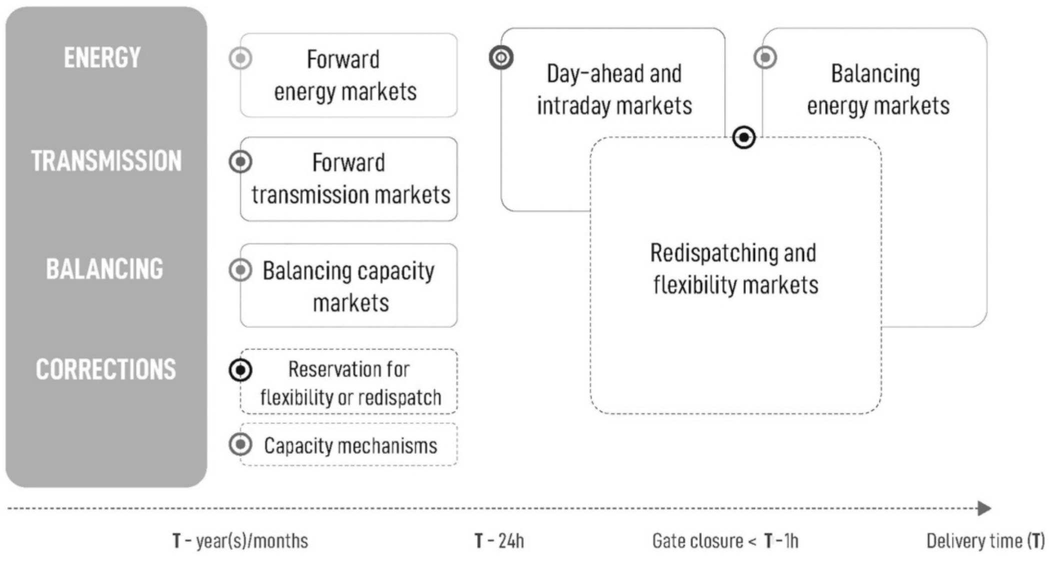


Figure 6.1.: This figure from Meeus (2020) shows a visual representation of the electricity market sequence in Europe.

The ENTSO-E operate the Transparency Platform (ENTSO-E, 2021b), which stores and provides access to the European electricity system and market data. As outlined in Hirth et al. (2018), the purpose of this platform is to support market members, such as energy generators, retailers, and traders, and to provide equal opportunities for all participants, regardless of company size, net worth or economic power of the individual actors. Each TSO is required to provide information about the energy market for their geographic region in the form of data instances, which can be divided into six categories: *Load, Generation, Transmission, Balancing, Outages* and *Congestion Management*. This information can for example be accessed via the website of the TP or via an API (Application Programming Interface) to collect the data instances and store them in a database.

As outlined in Schittekatte et al. (2020), we have to consider the three physical characteristics *time, location, flexibility* of electricity, when discussing electricity markets. To guarantee a stable energy supply and to maintain the utility frequency of the electricity system, supply and demand must be balanced at all times. In comparison to other commodities such as gold or oil, we are currently not able to store electricity in large volumes, which leads to a high variability of the value and price of electricity over time. Moreover, the transmission capacities are limited, and transmission components must be operated with respect to safe limits, hence the cost and value of electricity is also dependent on the location. Ultimately, the flexibility of the electricity market must be considered, as there is not only a high fluctuation in the demand of electricity, but also the possibility of sudden power station failures or other technical difficulties. As a result, the capacity to quickly adjust power generation or consumption is extremely valuable. This motivates the electricity market sequence, which is illustrated in Figure 6.1. From this figure we can observe that

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

the trading of electricity can start years before the actual delivery of the energy. However, in this thesis our focus is on the balancing energy market, which operates close to real time and helps to ensure that energy supply and demand are always balanced. On this market Balancing Service Providers specify the price they wish to receive, to raise or reduce their energy injection into or withdrawal from the power grid in real-time. Specifically, we want to create a predictive model for the `imbalance price`, which has the unit EUR/MWh. In the case of Austria, energy market data are published for timeframes with a validity of 15 minutes on the TP. It is imperative to note, that the data instances are oftentimes updated retrospectively and that such updates are possible until a legally mandated deadline is reached. We will follow up on this aspect of the energy market with an analysis in Section 6.1, once we have discussed the Austrian energy market in general terms. For further information about the ENTSO-E TP and more details on the European energy market, we refer to ENTSO-E (2021b); Hirth et al. (2018); Meeus (2020); Schittekatte et al. (2020).

6.1. Data analysis and feature engineering

At voestalpine, energy market data are collected from the ENTSO-E TP and stored in a database. The data used for this thesis is provided by voestalpine and it concerns a period of nine months, starting in August 2020 and ending with April 2021. In Table 6.1 we exemplary list 20 out of the $2,028,659$ available database entries.

From the column labeled `time` we can see for which time interval t_b, where t_b denotes the beginning of each 15-minute interval, a parameter value p_v is valid. Each parameter corresponds to an identification number p_{id}, which is stored in the column `id` with the respective value stored in column the `value`. The parameter with $p_{id} = 85$ corresponds to the `imbalance price`, which is our factor of interest. This means, that parameter values p_v with $p_{id} = 85$ represent the dependent variable y in our models, which will be further subdivided into into three classes later. The remaining parameters with $p_{id} \neq 85$ represent the explanatory variables, which we denote as $x_{p_{id}}$. In Appendix B in Table B.1 we list the parameter names corresponding to the parameter id p_{id}, for a more detailed description of the parameters we refer to the website of the TP (ENTSO-E, 2021b). The column `query timestamp` contains the timestamp t_q at which the respective entry was stored in the database, which matches the publishing time on the TP up to a negligible number of seconds. Since parameter values can be updated until the legally fixed deadline is reached, we keep track of those updates by storing the update number n_{up} in column `update`. Additionally, we derive the time difference between the time the price became valid and the time it was queried as $\Delta = t_b - t_q$, which we find in column `time difference`. It is worth mentioning that we limit ourselves to the energy market data from Austria in this work, but it would be possible to include information regarding the neighboring countries of Austria, or even from all members of the ENTSO-E, as a further step in subsequent works.

To get a better understanding of the updating procedure, we examine the parameter with $p_{id} = 95$ as an example, the corresponding entries in Table 6.1 are highlighted with a gray background color. We observe three entries for t_b = 2020-08-01 00:15:00. The first and

TU Bibliothek, WIEN Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

the second instance were published before t_b, while the third update arrived 42.18 minutes after t_b.

Table 6.1.: Longitudinal data, as stored in the database along with an additional column for the time difference between t_b and t_q.

`time` t_b	`id` p_{id}	`value` p_v	`query timestamp` t_q	`update` n_{up}	`time difference` $\Delta = t_b - t_q$
2020-08-01 00:15:00	82	5.00	2020-08-01 00:45:12	1	30.20 min
2020-08-01 00:15:00	82	32.00	2020-08-05 07:53:13	2	6218.22 min
2020-08-01 00:15:00	85	42.97	2020-09-24 09:03:12	1	78288.20 min
2020-08-01 00:15:00	86	42.98	2020-07-31 08:03:11	1	-971.82 min
2020-08-01 00:15:00	87	-1497.00	2020-07-31 08:03:11	1	-971.82 min
2020-08-01 00:15:00	88	1889.00	2020-07-31 08:03:11	1	-971.82 min
2020-08-01 00:15:00	89	9.30	2020-07-31 08:03:11	1	-971.82 min
2020-08-01 00:15:00	90	15.00	2020-08-01 00:37:11	1	22.18 min
2020-08-01 00:15:00	91	0.00	2020-08-01 00:37:11	1	22.18 min
2020-08-01 00:15:00	92	0.00	2020-07-30 10:01:12	1	-2293.80 min
2020-08-01 00:15:00	93	176.00	2020-07-30 09:03:11	1	-2351.82 min
2020-08-01 00:15:00	93	196.00	2020-07-31 05:51:12	2	-1103.80 min
2020-08-01 00:15:00	93	184.00	2020-07-31 15:51:11	3	-503.82 min
2020-08-01 00:15:00	94	0.00	2020-07-30 10:01:12	1	-2293.80 min
2020-08-01 00:15:00	**95**	192.00	2020-07-31 22:57:11	1	**-77.82** min
2020-08-01 00:15:00	**95**	160.00	2020-07-31 23:57:12	2	**-17.80** min
2020-08-01 00:15:00	**95**	120.00	2020-08-01 00:57:11	3	**42.18** min
2020-08-01 00:15:00	96	0.00	2020-08-01 03:01:11	1	166.18 min
2020-08-01 00:15:00	97	120.00	2020-08-01 04:01:11	1	226.18 min
2020-08-01 00:15:00	98	0.00	2020-08-01 01:27:11	1	72.18 min

6.1.1. Data availability

To create datasets on which we can build a predictive model for the `imbalance price`, we need to spread the data into a *wide* format, such that each parameter associated with a specific p_{id} has its own column, where the respective parameter value p_v is stored for every time interval t_b. This results in a dataset of the form $(X, y) \in \mathbb{R}^{n \times p+1}$, with n observations, p features and the dependent variable y. The matrix X consists of all parameters with $p_{id} \neq 85$ and the elements of the vector y correspond to $p_{id} = 85$. To reach this objective, we need to filter the available instances, so that we are left with a unique value for each combination of a parameter identification p_{id} and a time interval t_b. We use $\delta(t), t \in N$ to describe the relation between the row number of the dataset and the time interval t_b. A time interval of interest is selected with $t_b = \delta(t)$ and $t_b = \delta(t \pm s)$ denotes a shift of $\pm s$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

times 15-minutes. If we use the notation $t_b = \delta(t_{cur}), t_{cur} \in N$ we assume that there is no information available that is published later than t_b.

Before we construct the datasets, we analyze the time differences between the final[1] entries of the explanatory variables $x_{p_{id}}$ and the dependent variable y:

- For the `imbalance price` we get a minimum time difference of about half an hour, $\Delta_{min} = 26.18$ min, and the median is given by approximately 39 days, $\Delta_{med} = 56620.68$ min. Finally, the update with the highest time difference was published almost two months belatedly with respect to the time interval t_b, $\Delta_{max} = 84217.17$ min. Due to this fact, we avoid modeling y as a time series based on previous observations of y, as in a moving average model for example, but choose a supervised learning approach to model the prediction task.

- Considering the explanatory variables, we analyze all entries with $p_{id} \neq 85$ and observe a minimum time difference of approximately negative two and a half days, $\Delta_{min} = -3643.80$ min, while the maximum is almost two weeks or exactly $\Delta_{max} = 20104.18$ min. This means that, on the one hand, some final parameter entries are published up to two and a half days in advance, for example the solar power production forecast x_{92}. On the other hand, the final entries for some explanatory variables are published with a delay of almost two weeks. This poses a problem when we consider the predictive power of the model built on this information, since our objective is to construct a model that enables us to forecast the imbalance price y given X for the current time period starting at $t_b = \delta(t_{cur})$. Our problem is therefore that we cannot simply construct a matrix of explanatory variables X that consists of the final values, since those values are unknown if $\Delta > 0$. To create a matrix of explanatory variables X, where each parameter has its own column with one entry for each time interval t_b, we outline three different approaches in Sections 6.1.2, 6.1.3 and 6.1.4.

6.1.2. Final values

If we want to describe the relationship between the parameters in their final state, we construct the matrix X_{final} as follows:

1. For every combination of parameter id p_{id}, where $p_{id} \neq 85$, and time interval t_b, we select the entry with the highest number in the `update` column to get the final value.
2. Next, we spread the data into a *wide* format.
3. If this data preparation process leads to parameters with missing values, we replace them with the last available value.

With X_{final} we can draw inference on the relationship between the explanatory variables and the `imbalance price` y, but as already mentioned earlier, we cannot use it for prediction since at $t_b = \delta(t_{cur})$ we do not have the final values available.

[1] Final entry means that for every combination of parameter id p_{id} and time interval t_b, we select the entry with the highest number in the `update` column, which corresponds to the most recent date.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

6.1.3. Most recent values

Another approach of constructing a matrix of explanatory variables that can be used for prediction is based on using the most recent available value for each parameter. This results in the matrix X_{cur}, which is provided by voestalpine and could be replicated as follows:

1. In the first step, we remove all the entries where $\Delta > 0$.
2. For every combination of parameter id p_{id}, where $p_{id} \neq 85$, and time interval t_b, we select the entry with the highest number in the `update` column to get the final value.
3. Next, we spread the data into a *wide* format.
4. This process will introduce missing values; therefore we replace a missing parameter values for given time period t_b with the most recent parameter value p_v that is available at t_b.

In practice, the dataset is not constructed using the filtering approach as described above, but every minute we check if new values are available for any parameter p_{id} for the current time period $t_b = \delta(t_{cur})$ and when this is the case, those parameter values are updated. If there is a time period $t_b = \delta(t)$ where no parameter has changed compared to the previous time period $t_b = \delta(t-1)$, there is no new row created in the matrix X_{cur}. However, this is not a problem, as long as the model always uses the last available row of X_{cur} for prediction. For the data considered in this thesis, this leads to a matrix consisting of 34 columns and $25,849$ rows.

Concerning the modeling approach, on the one hand, we can train a model, using the matrix X_{final} that only consists of the final instances for the explanatory variables, and test the model on X_{cur}. Since some of the parameters are updated up to almost two weeks belatedly, we expect a significant drop in predictive power when we compare the evaluation of the training dataset to the test dataset. On the other hand, we could train and evaluate the model based on X_{cur}, which should lead to comparable results for the training and test datasets. Nevertheless, X_{cur} contains columns that represent explanatory variables that are updated two weeks belatedly, and hence might contain unreliable feature values.

6.1.4. Restricted timespan

The last method to construct a matrix of explanatory variables that we consider, works as follows:

1. We begin by selecting a time threshold $\tilde{t}$.
2. Given this threshold, we only keep entries for which $\Delta < \tilde{t}$.
3. Subsequently, we select the parameters where the average time difference is smaller than $\tilde{t}$.
4. Next, we spread the data into a *wide* format.
5. Finally, missing values get replaced with the last available value.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Analyzing the time difference Δ for each parameter individually, we observe that for the majority of parameters it holds that the median as well as the third quantile of the time difference Δ is smaller than 30 minutes. Following steps one to five from above, we construct a matrix of explanatory variables $X^{30\text{min}}$, with $\tilde{t} = 30$. Since this matrix includes observations that are not available for prediction at $t_b = \delta(t_{cur})$, we create a lagged (shifted) version X_{lag_2}: The t-th row of $X^{30\text{min}}$ corresponds to row $t + 2$ of X_{lag_2}, as illustrated in Table 6.2. With the matrix X_{lag_2} of time-shifted explanatory variables at hand, we can build and evaluate a model using the dataset (X_{lag_2}, y). The matrix of explanatory variables X_{lag_2} consists of 26 features and 26,048 observations, while the matrices X_{final} and X_{cur} are composed of 34 columns.

Table 6.2.: Illustration of how the matrix X_{lag_2} is created out of $X^{30\text{min}}$ and the relation to the dependent variable y, the subscripts denote the rows of $X^{30\text{min}}$ and the elements of y.

$X^{30\text{min}}$	X_{lag_2}	y
$X_1^{30\text{min}}$	–	y_1
$X_2^{30\text{min}}$	–	y_2
$X_3^{30\text{min}}$	$X_1^{30\text{min}}$	y_3
$X_4^{30\text{min}}$	$X_2^{30\text{min}}$	y_4
$X_5^{30\text{min}}$	$X_3^{30\text{min}}$	y_5
$X_6^{30\text{min}}$	$X_4^{30\text{min}}$	y_6
$X_7^{30\text{min}}$	$X_5^{30\text{min}}$	y_7
$\vdots$	$\vdots$	$\vdots$

To implement this approach in a practical application, we can follow a similar procedure as for X_{cur}. For the prediction of the `imbalance price` of the current time period $t_b = \delta(t_{cur})$, we consider parameter values for $t_b = \delta(t_{cur} - 2)$, including all updates of those parameters up to $t_b = \delta(t_{cur})$. Since we only consider parameters where the average time difference Δ is smaller than 30 minutes, unreliable feature values should not constitute the same problem as for X_{cur}. Moreover we can consider the following two scenarios: Firstly, we might include additional lags in the dataset and analyze their influence on the predictive quality of the model, e.g., adding the parameters with a lag of three, meaning that we not only consider the parameters for $t_b = \delta(t_{cur} - 2)$, but also $t_b = \delta(t_{cur} - 3)$ to construct $X_{lag_{2-3}}$. Secondly, we can consider a hypothetical scenario to analyze the influence of a better data availability, e.g., if we would observe that the median time difference is lower than 15 minutes, we could consider the time threshold of $\tilde{t} = 15$, and use a lag of one to construct the matrix of explanatory variables X_{lag_1}, see Section 7.1.1. While the first scenario is already viable for the case at hand, the second approach might become more relevant in the future when it is applicable in practice, since the energy market is constantly evolving, and data quality and availability are improving continuously.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

6.2. Exploratory data analysis

In Section 7.1 we will see that the dataset based on X_{lag_2} yields the most promising results, hence we will only include a more detailed analysis of the dataset (X_{lag_2}, y). As mentioned previously, this dataset consists of 26,048 observations and 26 features and in this section we analyze all observations with descriptive statistics, without further splitting the dataset into separate training and test sets. The motivation for this approach is twofold: Firstly, the validation procedure for model selection in Chapter 7, does not rely on the insights we gain from analyzing all observations in this chapter. Secondly, we neither perform manual feature selection nor statistical tests, but focus on an analysis of the final model with Explainable AI in Section 7.3.

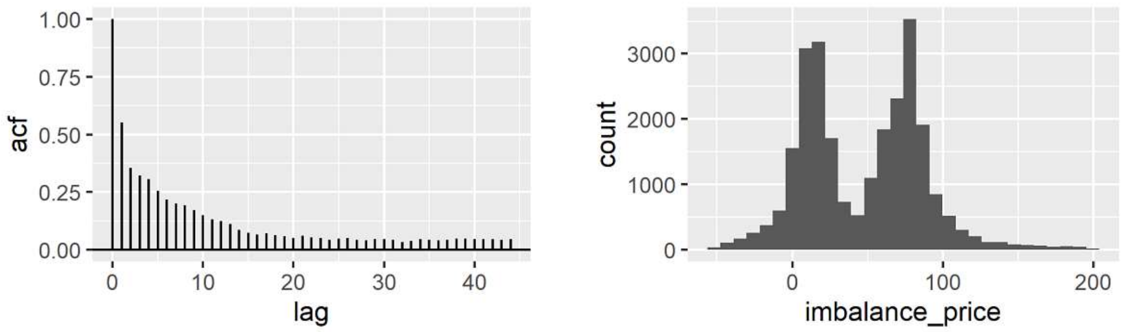


Figure 6.2.: Visualization of the autocorrelation of the `imbalance price` (left panel) and histogram of the `imbalance price`, based on those observations where its value is in between the 0.01 and 0.99 percent sample quantile (right panel).

Before we divide the `imbalance price` into three categories, we visualize the auto correlation function and the histogram in Figure 6.2. While the minimum and maximum of the `imbalance price` are given by $-2,198.04$ and $3,017.78$, respectively, the majority (98 percent) of the observations are contained in the interval $[-52.88, 198.97]$. The histogram of those 98 percent of the observations is visualized in the right panel of Figure 6.2. Moreover, we observe a strong autocorrelation of the `imbalance price`, but we cannot use this information for prediction due to the publishing time delay, as pointed out in Section 6.1.1.

In the next step we analyze the correlation structure in the dataset. The two explanatory variables that have the highest absolute correlation to the `imbalance price` are the `TRADED_AMOUNT` ($p_{id} = 80\_81$) and the `SPOT_PRICE_D_1` ($p_{id} = 84$), with a Pearson correlation coefficient of -0.26 and 0.18, respectively. In Figure 6.3 we observe high correlations between explanatory variables, for example the two parameters related to solar power generation, with $p_{id} = 92$ and $p_{id} = 96$, are highly correlated. The same relation holds true for the parameters which contain the information about wind power generation, with $p_{id} = 93$ and $p_{id} = 97$, although the correlation is less extreme. We want to note, that we analyzed the effect of removing one of the features on the modeling procedure described in Chapter 7 and we did not observe a change in predictive power. Therefore, we included both

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

parameters to analyze the effect of highly correlated variables on the feature importance based on Shapley values.

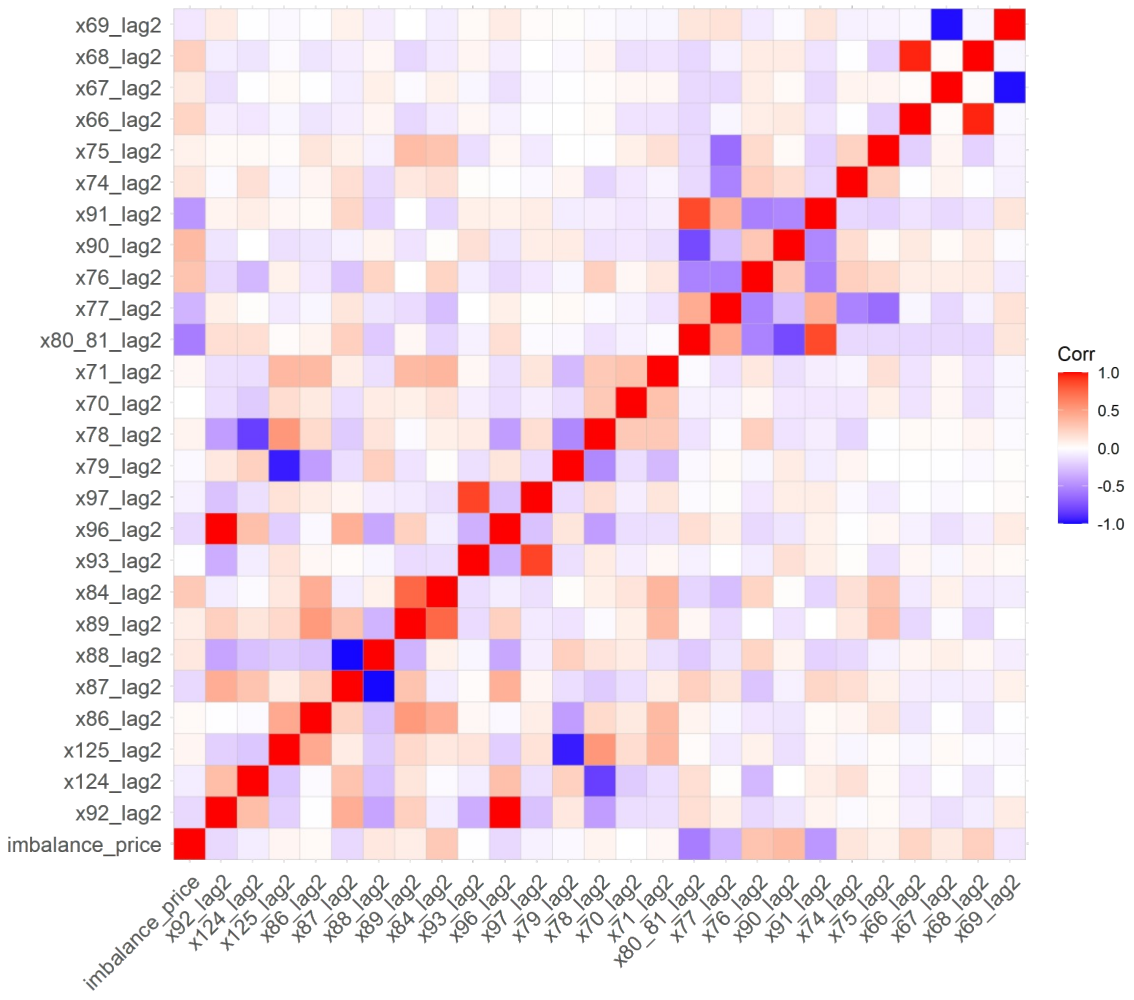
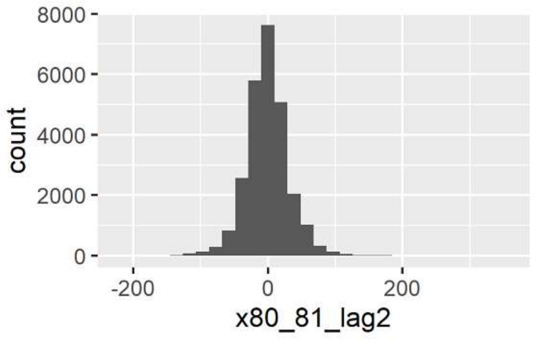


Figure 6.3.: Correlation analysis based on the Pearson correlation coefficient for the `imbalance price` and the 26 explanatory variables.

While we carried out an in-depth analysis of all features in `R`, we only include the histograms of the `TRADED_AMOUNT` (p_{id} = 80_81) and the `SPOT_PRICE_D_1` (p_{id} = 84) as examples in Figure 6.4. The `TRADED_AMOUNT`, displayed in the right panel, is the total aggregated volume of the imbalance for each 15-minute time period and is given in MWh. We observe positive as well as negative feature values, which represent either an energy surplus or deficit. The `imbalance price` for energy deficits is based on the TSO's payment to the balancing service provider. The price for energy surplus refers to the payment made to the TSO by the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.



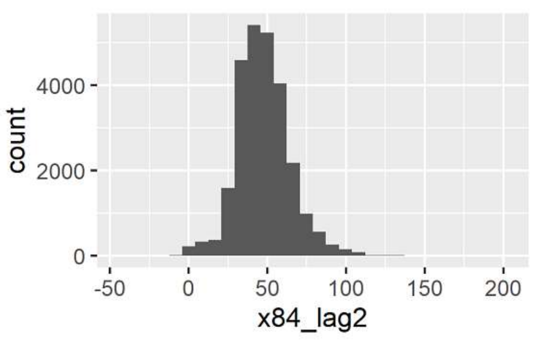


Figure 6.4.: Histograms of the TRADED_AMOUNT (left panel) and the SPOT_PRICE_D_1 (right panel).

balancing service provider. A negative price denotes the acquisition of energy in exchange for a monetary payment (ENTSO-E, 2021b). In the right panel we display the day-ahead spot price (x_{84}), with the unit EUR/MWh, and we observe mostly positive values for this feature.

The objective of the models, discussed in Chapter 7, is to predict the `imbalance price`, which is split into the three categories. The information on how to split the `imbalance price` into classes, was provided by voestalpine. The resulting dependent variable y is defined by

$$y := \begin{cases} \texttt{low} & \text{if } \texttt{imbalance price} \in (-\infty, 10) \\ \texttt{medium} & \text{if } \texttt{imbalance price} \in [10, 70) \\ \texttt{high} & \text{if } \texttt{imbalance price} \in [70, \infty) \end{cases} . \tag{6.1}$$

In Figure 6.5 we summarize the class counts of y in the left panel, where we observe that most observations correspond to class `medium`. Nevertheless, none of the three classes seems to be underrepresented. In the right panel of Figure 6.5 we show the estimated densities of the TRADED_AMOUNT for each class and note that while those are overlapping, we see that this explanatory variable seems to offer some form of separation between the classes.

To emphasize the time dependence of our dataset, we analyze the class counts of y for each week in Figure 6.6. We observe that the class `high` is dominating in the year 2021, while class `medium` is more frequently observed in 2020.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

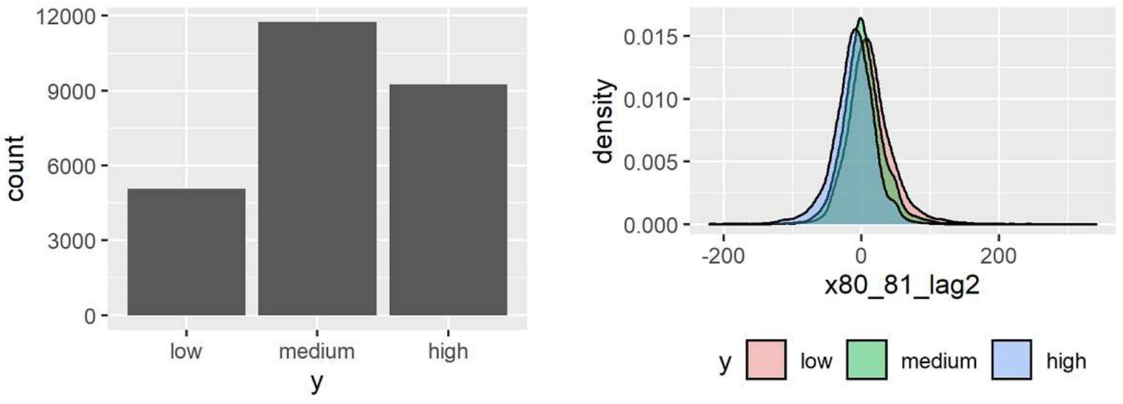


Figure 6.5.: The plot in the left panel summarizes the class counts for of y and in the right panel the densities of the TRADED_AMOUNT are displayed for each class.

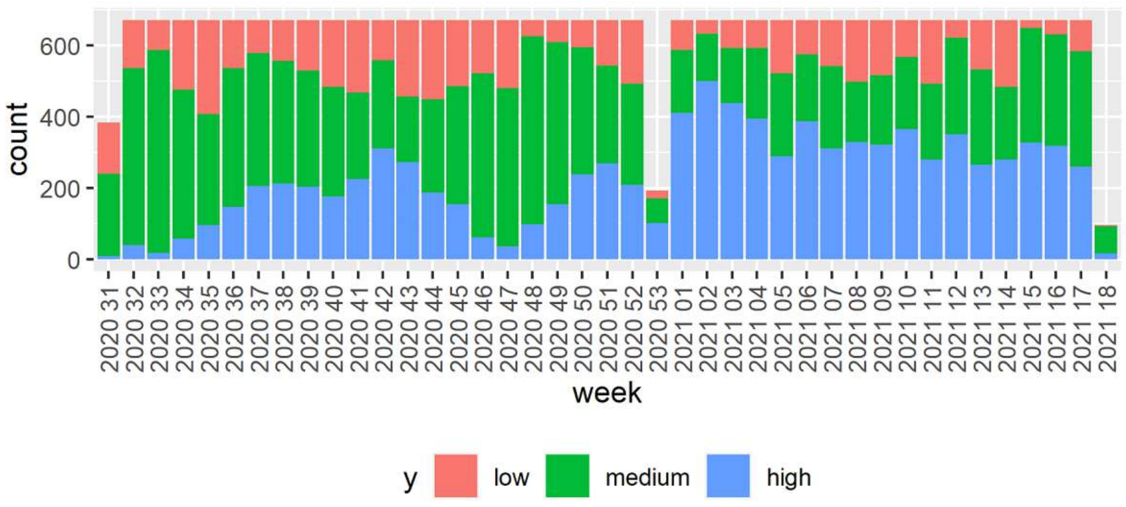


Figure 6.6.: This plot displays the weekly class counts of y.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

7. Model and Explainable AI

Our objective is to create a model to predict the `imbalance price` class, as defined in Equation (6.1), and analyze it with the methods of Explainable AI, introduced in Chapters 3, 4 and 5. Formally speaking, we want to predict y based on the currently available energy market data. The remainder of this chapter is structured as follows: First we compare different modeling approaches in Section 7.1, before we introduce an online learning procedure in Section 7.2 for further model improvement. Last, but certainly not least, we analyze our best-performing model using Explainable AI in Section 7.3.

7.1. Modeling approaches

In this section, we compare four strategies to predict y, using the datasets (X_{final}, y), (X_{cur}, y) and (X_{lag_2}, y) that were introduced in the previous chapter, and compare them to a baseline accuracy. The four approaches we consider are listed below:

Approach 1: Train on (X_{final}, y) and test on (X_{final}, y) – not feasible in practice.

Approach 2: Train on (X_{final}, y) and test on (X_{cur}, y).

Approach 3: Train on (X_{cur}, y) and test on (X_{cur}, y).

Approach 4: Train on (X_{lag_2}, y) and test on (X_{lag_2}, y).

We use the following validation procedure to compare the four approaches: In the first step we use the first $2,000$ observations as training data and the subsequent $1,000$ observations for testing. We do not use random sampling to select those instances due to the fact that the problem has an interwoven time-dependence. Randomly selecting data would mean that we could in fact mix *future* observations into the training dataset, which in turn might lead to unreliable accuracy estimations as training the model on normally not yet existent data is obviously not a realistic use-case situation. In the second step of the validation process we use observations 1 to $3,000$ for training and observations $3,001$ to $4,000$ for testing and every subsequent step follows the same pattern, expanding the training set observations by $1,000$ and predicting the following $1,000$ instances, until the last row of our dataset is reached.

While there is a wide variety of ML models that would be applicable to our problem, we restrict ourselves to the five models listed below. We want to note that we used the standard (hyper)-parameters for most instances, except for the adjustments denoted in the list below[1]:

1. Multinomial logistic regression without variable selection, referred to as **LR**, using the `R` package `nnet` (Venables and Ripley, 2002).
2. Multinomial logistic regression with variable selection, denoted as **LR lasso**, employing the `R` package `glmnet` (Simon et al., 2011) and applying the "one SE" rule to decide on the model used for prediction.
3. Random forest **RF**, utilizing the `R` package `ranger` (Wright and Ziegler, 2017), where we only used 50 trees to limit computational time.
4. Boosted tree ensemble **XGB**, employing the `R` package `xgboost` (Chen et al., 2021) with 10 boosting rounds.
5. Multilayer perceptron **MLP**, using the `R` package `keras` (Allaire and Chollet, 2021). We use two hidden layers with $3p$ and $6p + 3$ neurons, each followed by a dropout layer with a dropout rate of 50 percent, and a softmax activation function.[2]

The results of the models are displayed in Table 7.1 and Figure 7.1. First, we note that all models perform better than the median (0.383) and mean (0.438) of the baseline model, which predicts the majority class of the training data for all test data. Comparing the different datasets, we observe that only Approach 1, which is based on training and evaluating the model on X_{final}, yields a significantly higher accuracy for all models compared to the model performance on the remaining datasets. Unfortunately, this approach is infeasible in practice due to the lack of availability of *future* data points, but it nonetheless serves as a scenario on how the model performance could improve, if we had enhanced data availability and quality at our disposal. Moreover, looking at Table 7.1, we conclude that tree-based models perform best on this dataset.

Moving on, we observe that that we cannot detect a substantial difference in test set accuracy between the models based on Approach 2, Approach 3 and Approach 4. Nevertheless, tree-based models yield a slightly higher and more stable prediction quality on the test data. Additionally, keeping Approach 1 in mind, they have the highest potential for improvement, if we could have better data availability and/or quality at hand in the future.[3] We will focus on the XGB model, since the `xgboost` package comes with an efficient version of the Tree SHAP algorithm and tree-based models are the most promising in this

[1]While other (hyper)-parameter choices might lead to different and possibly better results, we want to mention that we used (hyper)-parameter tuning to explore hundreds of different settings for the models using X_{lag_2}, but we hardly saw any change in accuracy on the test data.

[2]The number of neurons in the hidden layers is motivated by Bölcskei et al. (2017), where this number of hidden layers and neurons is proposed for a regression task. Even tough we are confronted with a multiple classification task, this choice provided good results. We tried various specifications for the MLP, but we did not overall achieve better results with different approaches.

[3]One could also argue that the LR model yields a similar prediction accuracy compared to the XGB model and has the advantage that it is inherently interpretable. However, choosing the tree-based model and using Shapley values does not only fit better to the goal of this thesis, but also possesses a higher potential for improvement.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Table 7.1.: Average model performance in terms of test set accuracy. The standard errors are given in parentheses.

	Average test set accuracy			
Model	Approach 1	Approach 2	Approach 3	Approach 4
LR (lasso)	0.68 (0.022)	0.531 (0.013)	0.525 (0.018)	0.514 (0.019)
MLP	0.727 (0.027)	0.526 (0.015)	0.457 (0.021)	0.488 (0.018)
LR	0.657 (0.023)	0.516 (0.015)	0.513 (0.017)	0.512 (0.02)
RF	0.838 (0.008)	0.548 (0.012)	0.53 (0.016)	0.54 (0.018)
XGB	**0.841** (0.007)	0.547 (0.012)	0.55 (0.015)	**0.559** (0.018)

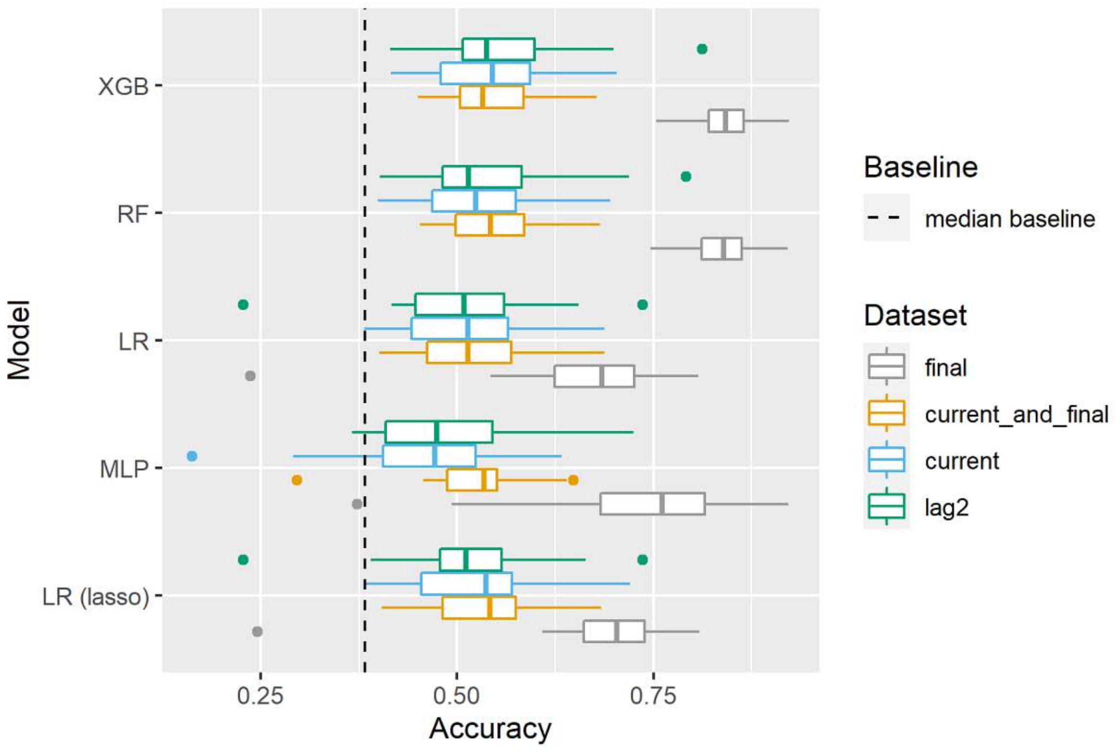


Figure 7.1.: Box plots for each model and dataset, to compare the performance of the different models, based on test set accuracy.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

scenario. Hence, the XGB model does not yield high accuracy, but also allows us to interpret the model using Shapley values. Regarding the choice of the dataset: Approach 2 has the drawback that it uses two different datasets for training and testing the model, which results in a major difference between in-sample and out-of-sample performance (comparing Approach 1 to Approach 2). The decision between Approach 3 and Approach 4 is harder since the performance of the models on both datasets is very similar. However, X_{lag_2} contains fewer parameters ($p = 26$) than X_{cur} ($p = 33$), since we only consider parameters for which the median time difference Δ is lower than 30 minutes, while it yields approximately the same prediction quality. Moreover, it allows us to consider the scenario that we have X_{lag_1} available, as described in Section 7.1.1. Due to those reasons, we choose Approach 4 for further analysis.

One could also model y as an ordinal response, since its definition imposes an inherent hierarchy, as we can see in Equation (6.1). Hence, misclassifying an observation by two classes – class `low` instead of `high` and vice versa, is worse than a classification error by just one class. It is worth noting, that we also investigated ordinal classification models, however, those methods did not match the accuracy, nor did they reduce the misclassifications by two classes, compared to the methods discussed in this work. Moreover, we analyzed the effect of parameter tuning on model performance, and it showed negligible effects on the dataset (X_{lag_2}, y). Since the focus of this thesis is to analyze the methods of Explainable AI for categorical responses, we choose not to include the results of our research into modeling y as an ordinal response here. However, applying an **online learning** approach yields a comparably higher accuracy, as will be discussed further in Section 7.2.

7.1.1. Emulating better data availability

Let us consider the scenario where the median time difference across all features is lower than 15 minutes while maintaining the same data quality. We emulate this case by constructing X_{lag_1} based on $X^{30\text{min}}$ and the results are given in Table 7.2. As expected, tree-based models still yield the best results, reinforcing our choice to use the gradient boosted ensemble models for further analysis.

Table 7.2.: The average model performance in terms of test set accuracy with the standard errors in parentheses, regarding the matrix of explanatory variables X_{lag_1}, which emulates better data availability.

XGB	LR	LR (lasso)	RF	MLP
0.63 (0.014)	0.569 (0.019)	0.564 (0.019)	0.607 (0.017)	0.546 (0.023)

7.2. Online learning

In contrast to a classic supervised learning approach, as illustrated in Figure 7.2, the online learning procedure refits the ML model in every time step t to predict y_{t+1}, as depicted

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

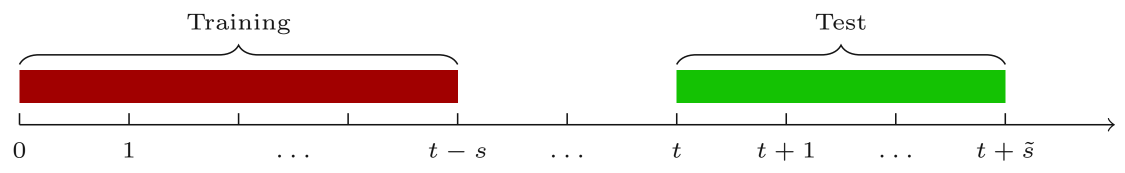


Figure 7.2.: Illustration of the standard supervised learning approach, using a training and test dataset.

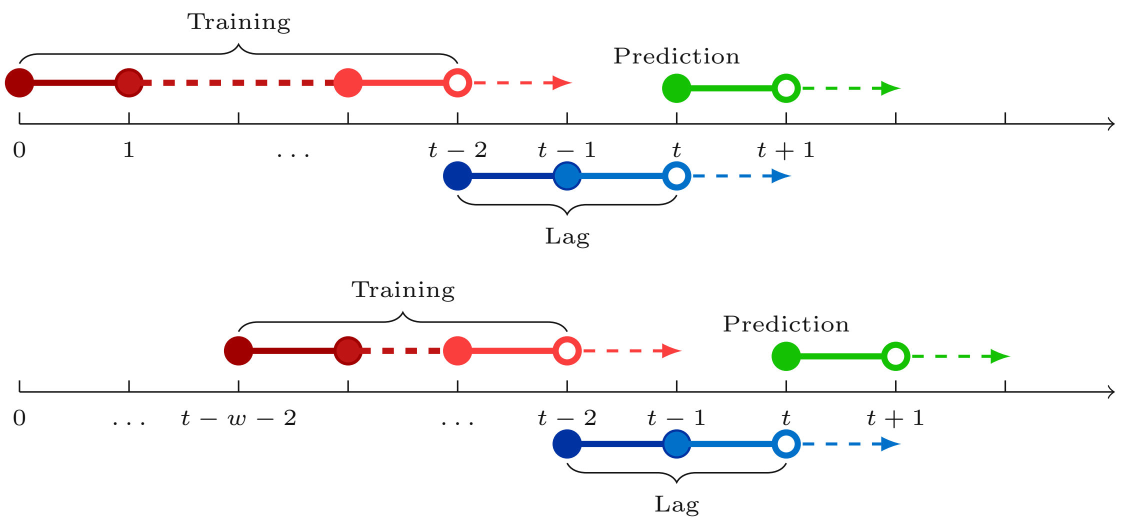


Figure 7.3.: Schematic of the online learning approach based on X_{lag_2}, using all previous observations (top) and using a training set of fixed length w (bottom).

in Figure 7.3. Applying this strategy to the energy market data is possible since the data instances on the TP are published and updated in a sequential procedure. Hence, we can refit the model whenever new data becomes available.

The classical supervised learning approach (Figure 7.2) uses a training set for model building and a test set for model evaluation. We used this procedure to select the model (XGB) and to decide on which dataset the model should be applied (X_{lag_2}), since it is computationally less expensive compared to the online learning approach. This is because we only fit the model once for each training set and not for every prediction.

For the online learning approach, we use the last 24,048 observations of our dataset one step at a time. While the prediction of the 24,048 instances is computationally more expensive following the online learning approach, when compared to the standard supervised learning procedure with test sets consisting of 1,000 observations, fitting the XGB model only takes a few seconds in each step, thanks to the efficient implementation. Therefore, we can still use it in a real-world scenario, where the model is refitted whenever a new data instance becomes available. We can either use all previous observations to fit the model (top panel of Figure 7.3) or limit the training data to a fixed length (bottom panel of Figure 7.3). On

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

the one hand, the latter approach has the advantage that it adapts faster to market changes – the fewer observations in the training set the faster the adjustment. On the other hand, the former approach might be advantageous since it can pick up on seasonal trends, which might be particularly useful when the training set includes observations over a timespan of at least a year.

As in Section 7.1, we apply the same validation approach for the online learning procedure like for the previous model comparisons. For the online learning models with a fixed training set length, we start by selecting an optimal number of observations in the training set. In Figure 7.4 we see a box plot for each training set length, and we conclude that 500 observations yield the highest median accuracy.

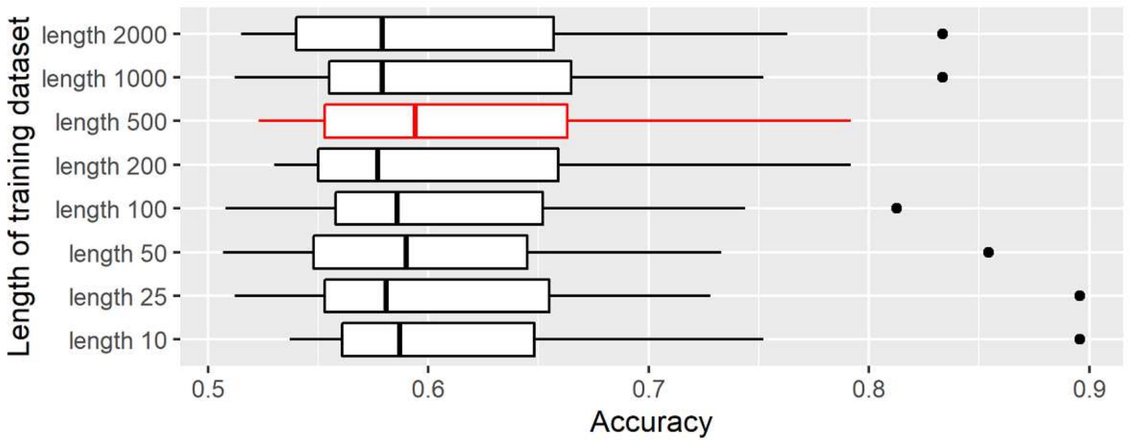


Figure 7.4.: Box plots of the accuracy of the XGB model, using an online learning approach with a limited length of the training set, where the box plot with the highest median is highlighted in red.

In Figure 7.5 and Table 7.3 we compare the optimal limited training set model with the online model using all previous observations and the classic training and test set approach. Comparing all modeling results for the prediction of the `imbalance price`, we conclude that the optimal modeling strategy is based on the X_{lag_2} dataset and the XGB model, employing an online learning approach with 500 observations in each training set. The accuracy of all models visualized in Figure 7.5 is varying greatly over time. We not only observe time periods where we correctly classify about 80 percent of the observations, but we also see periods with an accuracy of about 50 percent. As soon as we have observations of more than a whole year available, we could check if there are seasonal trends and adapt the model accordingly.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

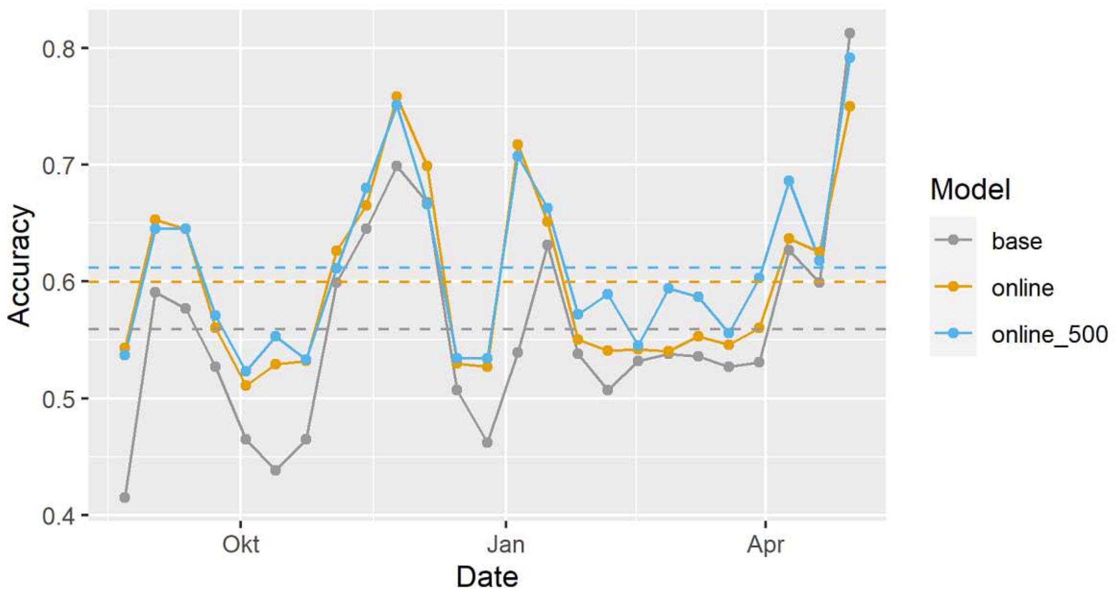


Figure 7.5.: Evolution of the aggregated accuracy for the three modeling approaches.

Table 7.3.: Average accuracy for the three modeling approaches, denoting the average model performance in terms of test set accuracy with the standard errors in parentheses.

Model	Accuracy
Classic	0.559 (0.018)
Online	0.6 (0.015)
Online 500	0.612 (0.015)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

7.3. Explainable AI

Since the online learning approach based on XGB models with 500 training observations resulted in the best overall model performance, we want to analyze this model using Shapley values. We predict 24,048 observations using this approach, meaning we have to compute and analyze the Shapley values of 24,048 XGB models. While we cannot use the default explanation methods to analyze multiple models, we can still calculate a global variable importance based on Shapley values if we aggregate them over all models.

7.3.1. Analysis of all observations

First, we start with a global variable importance plot, where we compare the variable importance based on Shapley values between training and test sets, as visualized Figure 7.6. Analyzing this figure, we observe that there are only marginal differences between both approaches. Those plots help us to detect the most influential features across all models. The four most influential features based on averages across all models, according to both training and test data, are x_{84} (`SPOT_PRICE_D_1`), x_{80_81} (`TRADED_AMOUNT`), x_{93} (`WIND_SHALL_D_1`), and x_{74} (`aFRR_PRICE_UP`). Additionally, the second feature related to wind power generation `WIND_SHALL` is also assigned a high importance. As pointed out in Section 6.2, we purposefully retained the highly correlated variables, which contain the information for wind and solar power generation, to analyze the effect on the global feature importance based on Shapley values. When we repeat the online learning procedure without the variables x_{92} and x_{93}, the accuracy drops from 0.612 to 0.599 and the variables x_{96} (`SOLAR_SHALL`) and x_{97} (`WIND_SHALL`) are assigned a higher importance score. However, the combined effects of x_{92} and x_{96} as well as x_{93} and x_{97} are comparably stronger than the case where only x_{96} as well as x_{97} are included in the model, as we can observe in Figure 7.7. Therefore, we conclude that retaining both variables provide reliable importance scores and yields a better model performance, but in spite of these facts one should be aware of the correlation structures in the dataset when interpreting the global variable importance based on Shapley values.

Moreover, we can monitor the Shapley values over time with the plot displayed in Figure 7.8, where we list the mean absolute Shapley values based on the test observations on the left y-axis and the date on the x-axis. Additionally, we included the model accuracy on the right y-axis. For this plot, we use moving averaged smoothing to obtain a comprehensible visualization of the 24,048 observations. The diagram allows us to monitor changes in variable importance. Such variations could for example be due to seasonal trends or new market policies. Since we are using an online learning approach, trained on the 500 most recent observations, we expect that the model would adapt quickly to new market policies or guidelines, and with the help of the Shapley values we can monitor and detect such changes.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

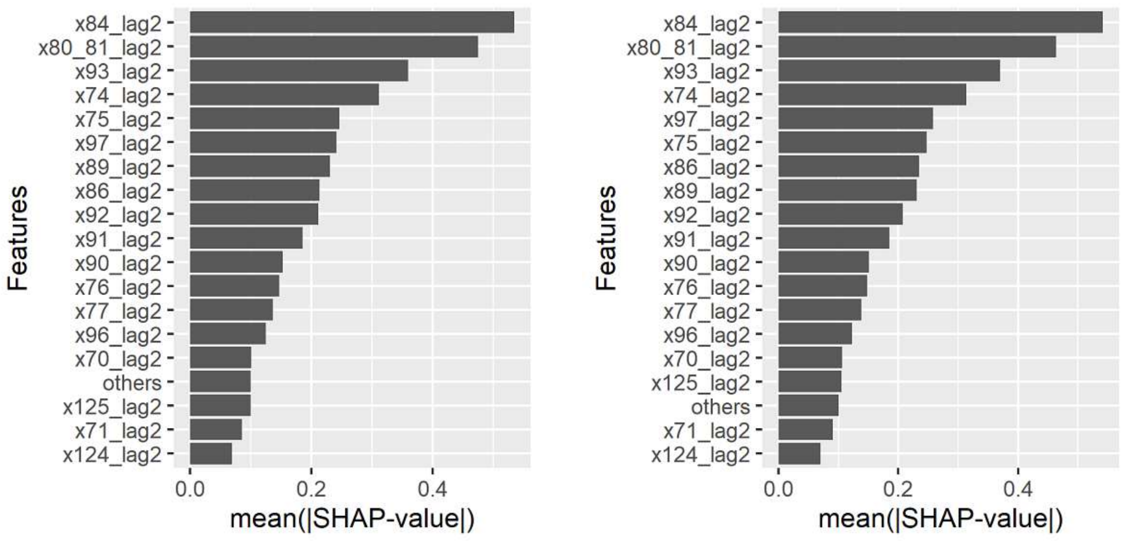


Figure 7.6.: Global feature importance based on the mean absolute Shapley value: The plot on the left is based on the mean over the average absolute Shapley values of the 500 training instances of the XGB model, while we used the test observation for the computation in the right plot.

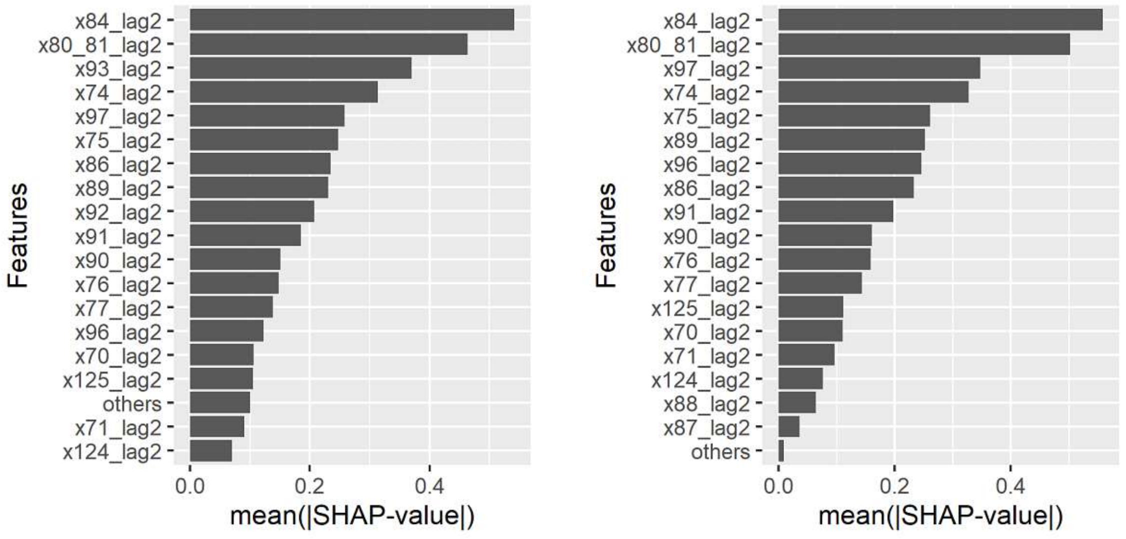


Figure 7.7.: Global feature importance based on the mean absolute Shapley value for the test observation: While all features are considered in the plot in the left panel, the variables x_{92} and x_{93} are removed in the graph in right panel.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

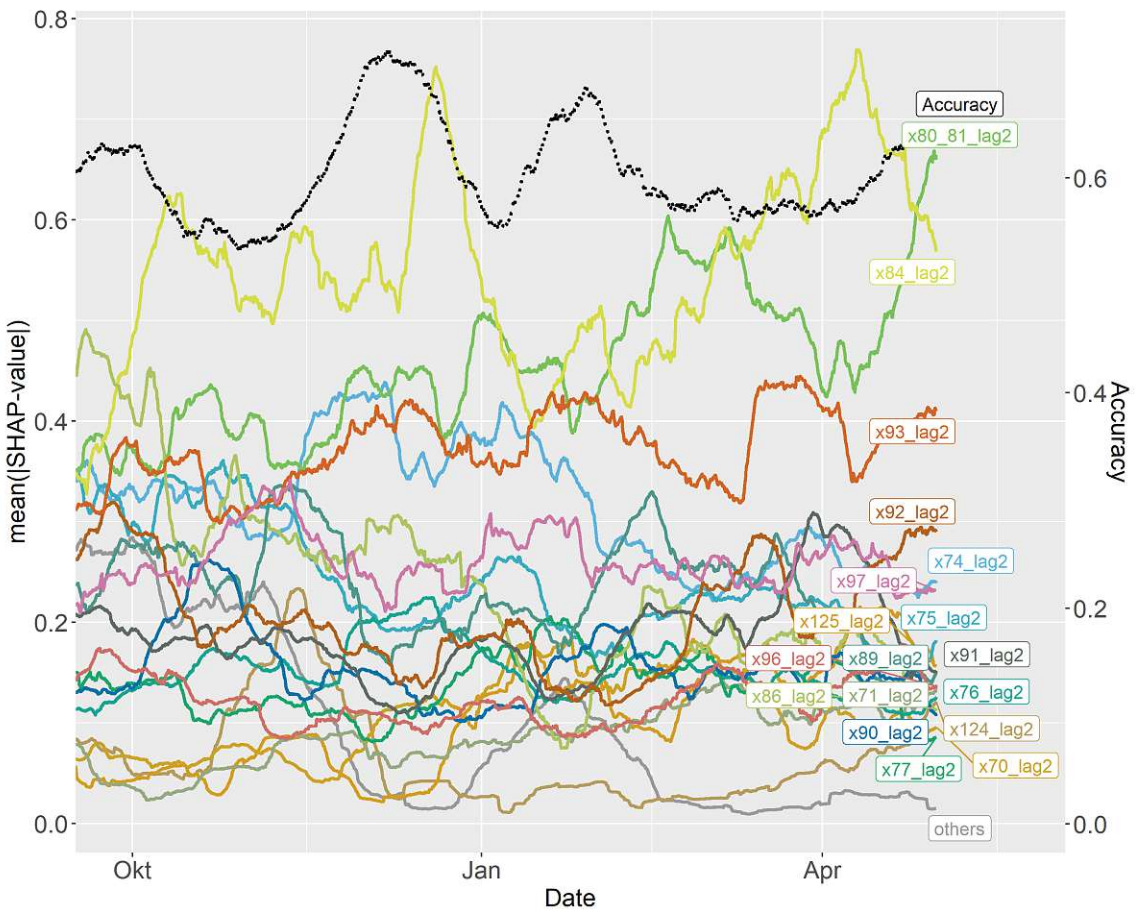


Figure 7.8.: Smoothed mean absolute Shapley values based on the test observations are given on the left y-axis and the date is denoted on the x-axis. The dotted black line illustrates the smoothed model accuracy over time.

7.3.2. Analysis of a single prediction

If we are interested in a single prediction, we can analyze the respective model which was used to predict this instance. We can either compute the Shapley values for the 500 training set observations or the single test observation. As an example, we analyze the model that was used for prediction at 23:45:00 on 2021-02-10, corresponding to the 18, 567-th instance in our dataset. For this instance, the model correctly labeled the observation as `high`, with class probabilities of 0.034 (`low`), 0.074 (`medium`) and 0.892 (`high`). For this single prediction we use Shapley values as well as a local surrogate model for analysis.

Before we analyze this model in more detail, we want to remind ourselves of the interpretation of the Shapley value: For a given set of feature values, the Shapley value is the difference between the actual prediction and the average prediction. Moreover, we keep in mind that the Shapley values calculated with the `xgboost` package do not represent the change in probability but are given *on the scale of untransformed margin* (Chen et al.,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2021). They do, however, indicate how far the actual prediction departs from the average prediction, just not in terms of probability.

In Figure 7.9 we analyze the absolute Shapley values summarized across all tree classes, based on the 500 training instances. In the left panel we display the **SHAP Summary Plot** and in the right panel the **SHAP Dependence Plot** for the variable x_{84} (`SPOT_PRICE_D_1`). Comparing the feature importance from this model to the importance scores across all models, we observe that x_{84} is still the most important variable, while the importance of the remaining features changed noticeably. For example, in comparison to the aggregated feature importance scores across all models, x_{92} (`SOLAR_SHALL_D_1`) is assigned a higher importance than x_{93} (`WIND_SHALL_D_1`). From the SHAP Dependence Plot we see that high feature values of x_{84} pose the highest difference between the actual prediction and the average prediction.

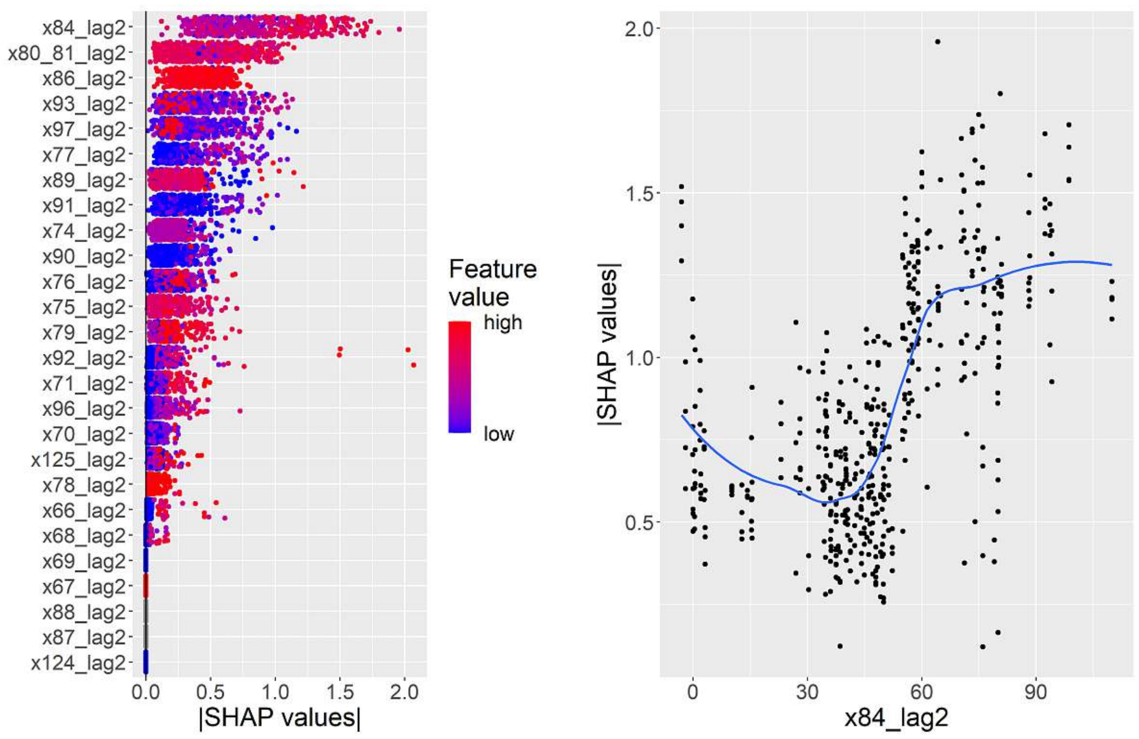


Figure 7.9.: SHAP Summary Plot (left panel) and SHAP Dependence Plot (right panel) visualizing the absolute Shapley values summarized across all three classes, based on the 500 training instances.

Moving on, each of the Figures 7.10, 7.11 and 7.12 is used to analyze the prediction of the single test instance, for class `low`, `medium`, and `high`, respectively. Each figure includes the model effects of a local surrogate model created with the `iml` package in the left panel, and in the right panel we display the Shapley values. Considering the local surrogate models, for each of the three classes a separate sparse logistic regression model with six features is fitted to explain the prediction of the XGB model. For the Shapley values, we explain the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

difference between the actual class probabilities of the test instance and the average class probabilities on the training data:

$$\begin{pmatrix}\Delta_{\texttt{low}}\\ \Delta_{\texttt{medium}}\\ \Delta_{\texttt{high}}\end{pmatrix} = \begin{pmatrix}-0.152\\ -0.185\\ 0.337\end{pmatrix} = \begin{pmatrix}0.034\\ 0.074\\ 0.892\end{pmatrix} - \begin{pmatrix}0.186\\ 0.259\\ 0.555\end{pmatrix}.$$

The local surrogate model provides a quick overview, but we should keep in mind that the prediction of the local model and the actual prediction differ for all three classes, while the Shapley values provide a full explanation.

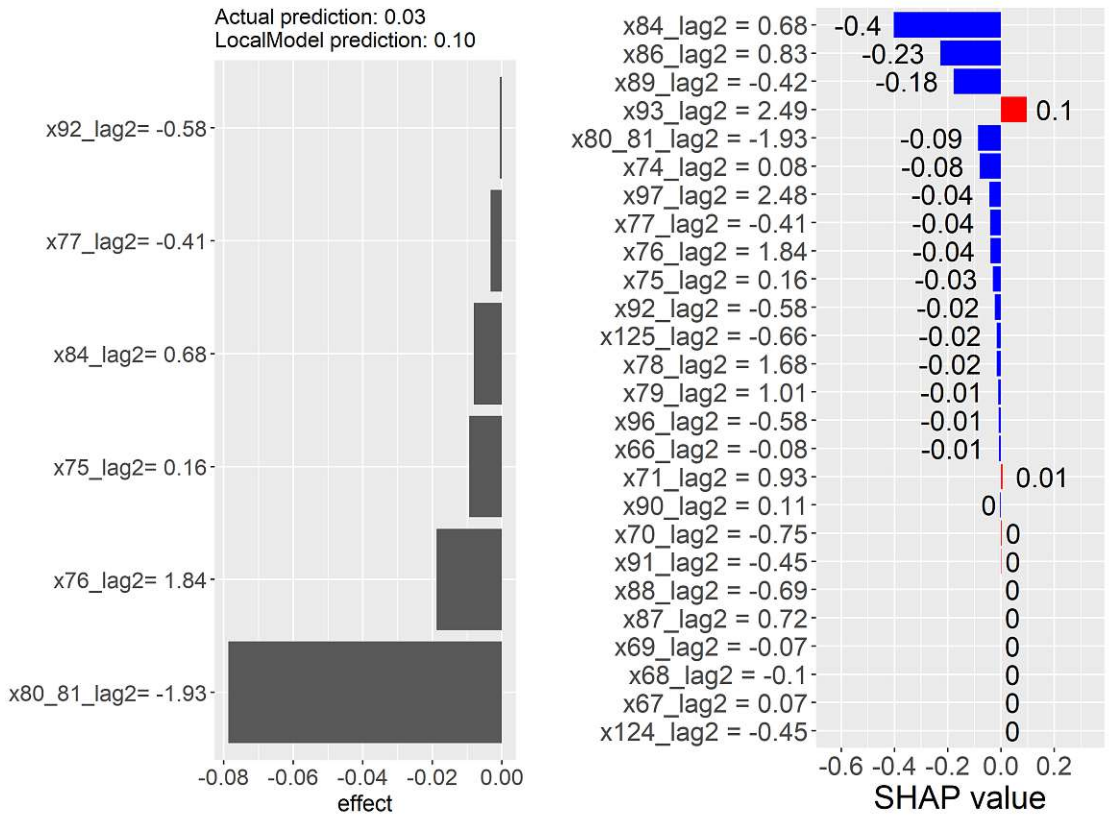


Figure 7.10.: Model analysis for class `low`: The left plot shows a visualization of the effects from the local surrogate model for the test observation and the right plot displays the Shapley values for this instance.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

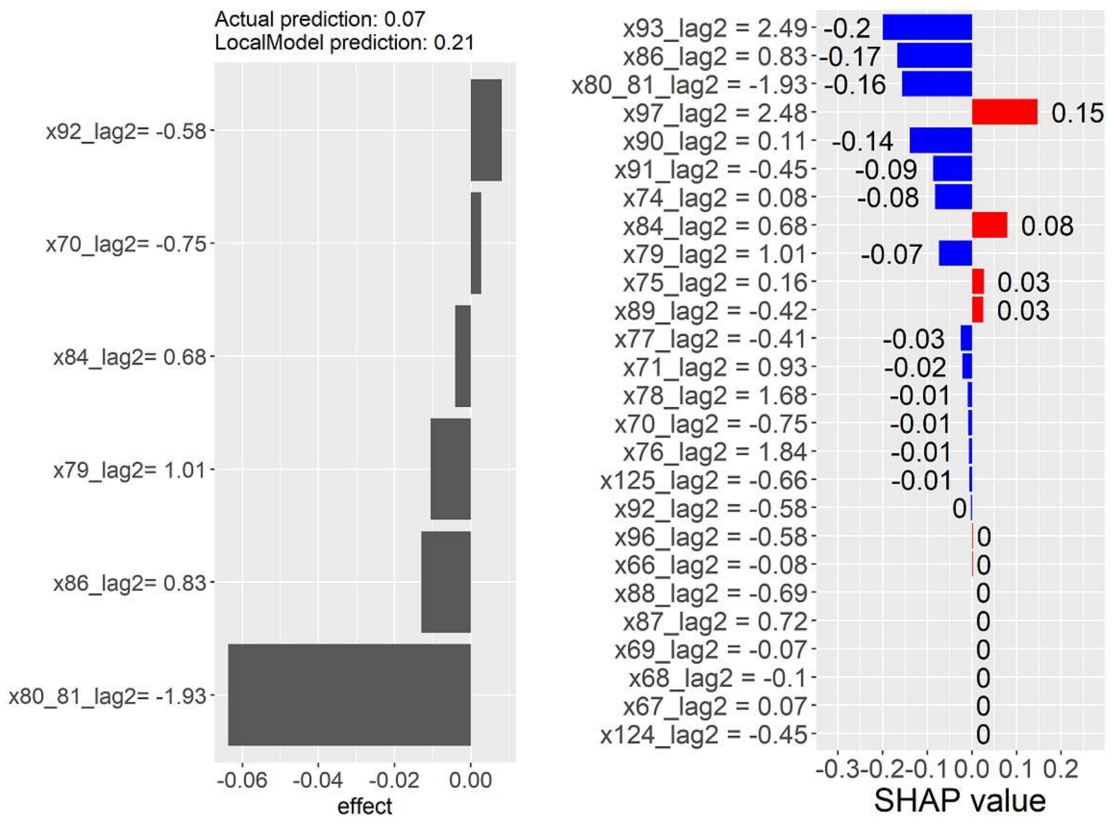


Figure 7.11.: Model analysis for class `medium`: The left plot shows a visualization of the effects from the local surrogate model for the test observation and the right plot displays the Shapley values for this instance.

TU WIEN Bibliothek Your knowledge hub

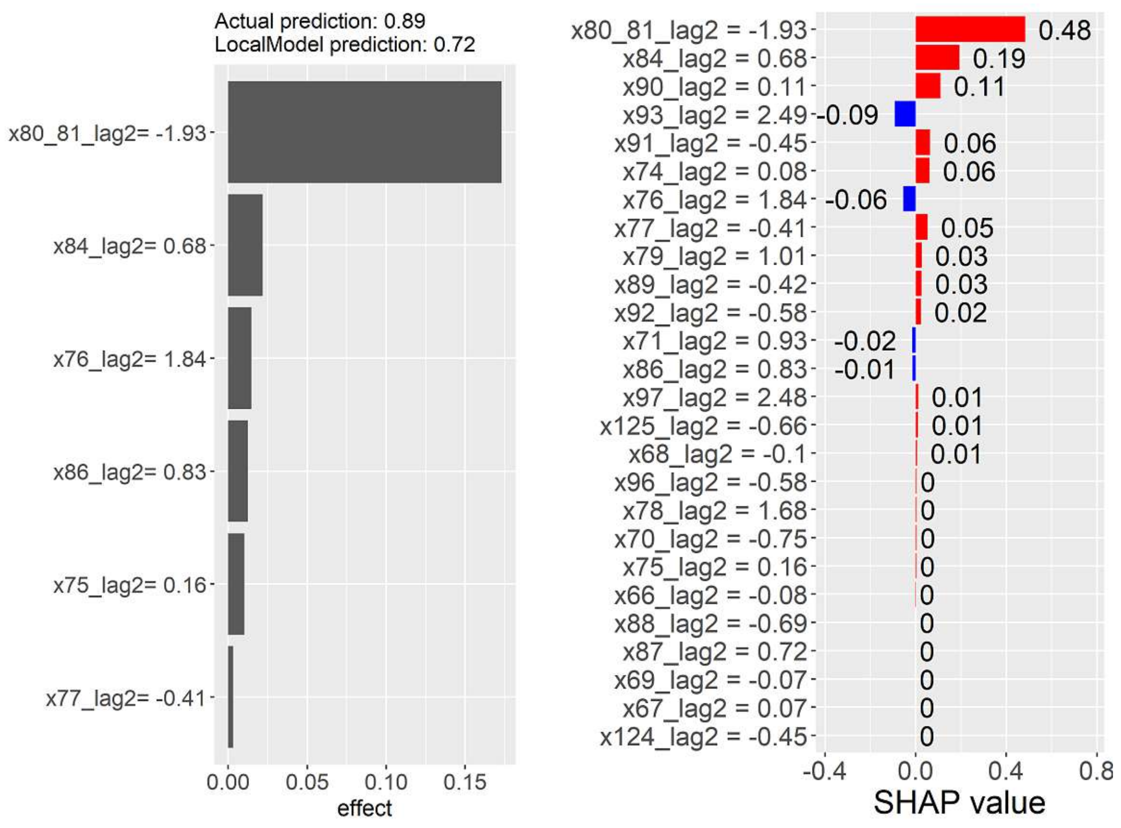


Figure 7.12.: Model analysis for class `high`: The left plot shows a visualization of the effects from the local surrogate model for the test observation and the right plot displays the Shapley values for this instance.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

8. Discussion and conclusions

In the theoretical part of this work we discussed the theory behind Local Interpretable Model-agnostic Explanations (LIME), Shapley values for model explainability, and SHapley Additive exPlanations (SHAP). Since LIME is based on local surrogate models, it enables us to provide concise explanations of individual observations. The methods based on Shapley values have the advantage that they are based on a solid theory and that they do not only provide a full explanation of individual instances, but they can also be aggregated to obtain feature importance scores. While the model-agnostic calculation of the Shapley values is computationally expensive, the Tree SHAP algorithm provides an efficient implementation for tree-based models.

Regarding the prediction of the energy imbalance price of the Austrian energy market, we compared multiple supervised classification techniques and concluded that tree ensembles provide the most promising results. Moreover, we observed that using an online learning approach improves the prediction accuracy. Concerning the feasibility study on the applicability of the methods of Explainable AI on this classification problem, we conclude that we can not only employ those methods to analyze supervised learning tasks, but also adapt them to explain an online learning model. Since voestalpine is participating in the balancing market, those improvements in predictive power and model explainability can be used to optimize decision-making processes and contribute to the goal of voestalpine, to be one step ahead.

In future works one could for example use the energy market data of all members of the European Network of Transmission System Operators for Electricity (ENTSO-E), since this might improve the accuracy of the model as the power grid in Europe is interconnected. Furthermore, one could investigate the effects of under- or oversampling the training data in each step of the online learning procedure.

Bibliography

Allaire, J. and Chollet, F. (2021). *keras: R Interface to 'Keras'*. https://CRAN.R-project.org/package=keras.

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

Bengtsson, H. (2020). A unifying framework for parallel and distributed processing in R using futures. *arXiv preprint arXiv:2008.00553*.

Bengtsson, H. (2021). *progressr: An Inclusive, Unifying API for Progress Updates*. R package version 0.8.0, https://CRAN.R-project.org/package=progressr.

Biecek, P. and Burzykowski, T. (2021). *Explanatory Model Analysis*. Chapman and Hall/CRC, New York.

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

Bölcskei, H., Grohs, P., Kutyniok, G., and Petersen, P. (2017). Optimal approximation with sparsely connected deep neural networks. http://arxiv.org/abs/1705.01714.

Chen, H., Lundberg, S. M., and Lee, S.-I. (2018). Understanding shapley value explanation algorithms for trees. https://hughchen.github.io/its_blog/index.html. (accessed: 10.08.2021).

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.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., and Li, Y. (2021). *xgboost: Extreme Gradient Boosting*. R package version 1.4.1.1.

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

ENTSO-E (2021a). Entso-e mission statement. https://www.entsoe.eu/about/inside-entsoe/objectives/. (accessed: 10.08.2021).

ENTSO-E (2021b). Entso-e transparency platform. https://transparency.entsoe.eu/. (accessed: 10.08.2021).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Firke, S. (2021). *janitor: Simple Tools for Examining and Cleaning Dirty Data*. R package version 2.1.0, https://CRAN.R-project.org/package=janitor.

Fisher, A. (2020). The uniqueness of shap depends on how you handle external information. https://aaronjfisher.github.io/SHAP-Symmetry.html. (accessed: 10.08.2021).

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. *J. Mach. Learn. Res.*, 20(177):1–81.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.

Gaujoux, R. (2020). *doRNG: Generic Reproducible Parallel Backend for 'foreach' Loops*. R package version 1.8.2, https://CRAN.R-project.org/package=doRNG.

Gower, J. C. (1971). "A general coefficient of similarity and some of its properties". *Biometrics*, 27(4):857–871.

Grabisch, M. (2016). *Set Functions, Games and Capacities in Decision Making*. Springer, Berlin.

Greenwell, B. (2020a). *fastshap: Fast Approximate Shapley Values*. R package version 0.0.5, https://CRAN.R-project.org/package=fastshap.

Greenwell, B. (2020b). *fastshap: Fast Approximate Shapley Values*. R package version 0.0.5.

Grolemund, G. and Wickham, H. (2011). Dates and times made easy with lubridate. *Journal of Statistical Software*, 40(3):1–25. https://www.jstatsoft.org/v40/i03/.

Harsanyi, J. C. (1963). A simplified bargaining model for the n-person cooperative game. *International Economic Review*, 4(2):194–220. http://www.jstor.org/stable/2525487.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York.

Hester, J. and Wickham, H. (2021). *odbc: Connect to ODBC Compatible Databases (using the DBI Interface)*. R package version 1.3.2, https://CRAN.R-project.org/package=odbc.

Hirth, L., Mühlenpfordt, J., and Bulkeley, M. (2018). The entso-e transparency platform – a review of europe's most ambitious electricity data platform. *Applied Energy*, 225:1054–1067. https://www.sciencedirect.com/science/article/pii/S0306261918306068.

Janzing, D., Minorics, L., and Blöbaum, P. (2020). Feature relevance quantification in explainable ai: A causal problem. In *International Conference on Artificial Intelligence and Statistics*, pages 2907–2916. PMLR.

Jia, E. (2020). Explaining explanations and perturbing perturbations. Bachelor's thesis, Harvard College. https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37364690.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Kuhn, M. (2021). *caret: Classification and Regression Training*. R package version 6.0-88, https://CRAN.R-project.org/package=caret.

Lemaire, V., Feraud, R., and Voisine, N. (2008). Contact personalization using a score understanding method. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 649–654.

Lewis, N. S. and Nocera, D. G. (2006). Powering the planet: Chemical challenges in solar energy utilization. *Proceedings of the National Academy of Sciences*, 103(43):15729–15735.

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Lundberg, S. M., Erion, G., Chen, H., Degrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.-I., and et al. (2020). From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):56–67.

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

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 30*, pages 4765–4774. Curran Associates, Inc.

Maksymiuk, S., Gosiewska, A., and Biecek, P. (2020). Landscape of R packages for explainable artificial intelligence. *arXiv preprint arXiv:2009.13248*.

Mcglade, C. and Ekins, P. (2015). The geographical distribution of fossil fuels unused when limiting global warming to 2 °C. *Nature*, 517:187–190.

Meeus, L. (2020). *The evolution of electricity markets in Europe*. Edward Elgar Publishing.

Microsoft and Weston, S. (2020). *foreach: Provides Foreach Looping Construct*. R package version 1.5.1, https://CRAN.R-project.org/package=foreach.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38. https://www.sciencedirect.com/science/article/pii/S0004370218305988.

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

Molnar, C., Casalicchio, G., and Bischl, B. (2018). iml: An R package for interpretable machine learning. *Journal of Open Source Software*, 3(26):786.

Molnar, C., Casalicchio, G., and Bischl, B. (2020). Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 417–431. Springer.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

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

Peters, H. (2008). *Game Theory*. Springer, Berlin Heidelberg.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

R Special Interest Group on Databases (R-SIG-DB), Wickham, H., and Müller, K. (2021). *DBI: R Database Interface*. R package version 1.1.1, https://CRAN.R-project.org/package=DBI.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why Should I Trust You?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Robnik-Sikonja, M. and Kononenko, I. (2008). Explaining classifications for individual instances. *Knowledge and Data Engineering, IEEE Transactions on*, 20:589 – 600.

Schervish, M. J. (1996). P values: What they are and what they are not. *The American Statistician*, 50(3):203–206.

Schittekatte, T., Reif, V., and Meeus, L. (2020). The EU electricity network codes (2020 ed.). *European University Institute*.

Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2011). Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13.

Slack, D., Hilgard, S., Jia, E., Singh, S., and Lakkaraju, H. (2020). Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186.

Ushey, K., Allaire, J., and Tang, Y. (2021). *reticulate: Interface to 'Python'*. R package version 1.20, https://CRAN.R-project.org/package=reticulate.

Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer, New York. https://ggplot2.tidyverse.org.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi, K.,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Vaughan, D., Wilke, C., Woo, K., and Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686.

Wickham, H., Girlich, M., and Ruiz, E. (2021). *dbplyr: A 'dplyr' Back End for Databases*. R package version 2.1.1, https://CRAN.R-project.org/package=dbplyr.

Wright, M. N. and Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1):1–17.

Young, H. (1985). Monotonic solutions of cooperative games. *International Journal of Game Theory*, 14:65–72.

Štrumbelj, E. and Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11:1–18.

Štrumbelj, E. and Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41:647–665.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

A. Gower similarity

The calculation of the Gower similarity between the observations $x_i = (x_{i1}, ..., x_{ip})$ and $x_j = (x_{j1}, ..., x_{jp})$ is given by

$$P_{ij} = \frac{\sum_{k=1}^{p} s_{ijk}\delta_{ijk}}{\sum_{k=1}^{p} \delta_{ijk}},$$

where s_{ijk} are the similarity scores for the comparison of observations x_i and x_k considering the k-th variable, and the weights $\delta_{ijk} \in \{0, 1\}$ indicate whether a comparison is possible or not. The definition of the score s_{ijk} depend on which type of variables x_i and x_j are and can be classified as follows:

- Binary variables are considered to be asymmetric, hence

$$s_{ijk} = \begin{cases} 1 & x_{ik} = x_{jk} = 1 \\ 0 & \text{otherwise} \end{cases}.$$

- For categorical variables the scores are given by

$$s_{ijk} = \begin{cases} 1 & x_{ik} = x_{jk} \\ 0 & \text{otherwise} \end{cases}.$$

- Finally, numerical variables are treated as interval-scaled variables and the scores are obtained via

$$s_{ijk} = 1 - \frac{|x_{ik} - x_{jk}|}{R_k},$$

  where R_k is the range of the k-th variable.

The weights $\delta_{ijk} \in \{0, 1\}$ allow us to obtain a similarity measure, even if missing variables (`NA`s) are present in the data set. The weights are only set to zero in two cases: On the one hand, if either x_{ik} or x_{jk} is missing, and on the other hand if the variable is an asymmetric binary where $x_{ik} = x_{jk} = 0$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

B. Parameter names

The parameter names in Table B.1 are based on the following acronyms: manual Frequency Restoration Reserve (mFRR), automatic Frequency Restoration Reserve (aFRR), Cross Border Balancing (CoBA), and “D_1” stands for day ahead. Volumes are measured in terms of MWh and the unit for prices is EUR/MWh.

Table B.1.: Names of the most important parameters.

p_{id}	parameter names
66	mFRR_PRICE_UP
67	mFRR_PRICE_DOWN
68	mFRR_VOLUME_UP
69	mFRR_VOLUME_DOWN
70	mFRR_SUPPLY_VOLUME_UP
71	mFRR_SUPPLY_VOLUME_DOWN
74	aFRR_PRICE_UP
75	aFRR_PRICE_DOWN
76	aFRR_VOLUME_UP
77	aFRR_VOLUME_DOWN
80_81	TRADED_AMOUNT
84	SPOT_PRICE_D_1
86	CoBA_PRICE_MIN_UP
87	CoBA_PRICE_MIN_DOWN
88	CoBA_PRICE_MAX_UP
89	CoBA_PRICE_MAX_DOWN
90	CoBA_VOLUME_UP
91	CoBA_VOLUME_DOWN
92	SOLAR_SHALL_D_1
93	WIND_SHALL_D_1
96	SOLAR_SHALL
97	WIND_SHALL
124	CoBA_VOLUME_POOLED_UP
125	CoBA_VOLUME_POOLED_DOWN