

Fakultät Für Informatik und Mathematik  
Stochastic Engineering in Business and Finance

## Master-Thesis

# Interpretation of linear models with SHAP

Interpretation linearer Modelle mit SHAP

Betreuer: Prof. Dr. Andreas Zielke

Eingereicht von:  
Simon Symhoven, 49651418  
Boschetsriederstraße 59A, D-81379 München  
simon.symhoven@hm.edu

Eingereicht am:  
München, den 17. November 2023



## **Abstract**

Diese Masterarbeit beleuchtet die Interpretation linearer Modelle mit SHAP, welches seine theoretische Basis in den Shapley-Werten der kooperativen Spieltheorie findet. Nach einer historischen Einordnung und Begriffsdefinition werden die Shapley-Werte formal hergeleitet und deren axiomatische Grundlagen beleuchtet. Der Übergang von Shapley-Werten zu SHAP wird zeigen wie Beiträge einzelner Merkmale zur Modellvorhersage beitragen. Am Beispiel einer ausgewählten Modellklasse und unter Verwendung des shap Python-Pakets wird die praktische Anwendbarkeit von SHAP auf einen konkreten Datensatz demonstriert. Die Arbeit schließt mit einer Diskussion über die Grenzen von SHAP und bietet einen Ausblick auf dessen Einsatzmöglichkeiten für transparente und nachvollziehbare Modellentscheidungen in der Datenwissenschaft.



## Inhaltsverzeichnis

1	Einleitung . . . . .	1
2	Historischer Kontext und Begriffsdefinitionen . . . . .	3
2.1	Die Genese der Shapley-Werte in der kooperativen Spieltheorie . . . . .	3
2.2	Shapley-Werte, SHAP, SHAP-Werte und <code>shap</code> . . . . .	4
3	Theorie der Shapley-Werte . . . . .	5
3.1	Wie lässt sich der Gewinn gerecht aufteilen? . . . . .	5
3.2	Formale Definition . . . . .	8
3.3	Axiome . . . . .	10
4	Von Shapley-Werten zu SHAP: Brückenschlag zur Modellinterpretation . . . . .	13
4.1	Berechnung der SHAP-Werte unter Berücksichtigung der zugrundeliegenden Verteilung . . . . .	15
4.2	Axiome . . . . .	20
5	Komplexitätsbewältigung bei der Berechnung von SHAP-Werten . . . . .	23
5.1	Approximation der marginalen Beiträge mittels Monte Carlo Integration . . . . .	24
5.2	Schätzung von Koalitionen . . . . .	25
5.2.1	Linearer SHAP Estimator für lineare Modelle . . . . .	26
6	Praktische Anwendung von SHAP auf lineare Modelle . . . . .	27
6.1	Lineare Modelle als analytische Grundlage . . . . .	27
6.2	Einführung in das <code>shap</code> Python-Paket . . . . .	29
6.3	Einführung in den Datensatz . . . . .	30

6.4	Explorative Datenanalyse & Datenaufbereitung . . . . .	31
6.5	Modellierung der linearen Regression . . . . .	31
7	Ergebnisse . . . . .	33
7.1	Berechnung von SHAP-Werten . . . . .	33
7.2	Interpretation . . . . .	33
7.2.1	Lokale Interpretation . . . . .	33
7.2.2	Globale Interpretation . . . . .	33
8	Fazit & Ausblick . . . . .	37
	Literaturverzeichnis . . . . .	39
	Abbildungsverzeichnis . . . . .	41
	Tabellenverzeichnis . . . . .	43
	Quellcodeverzeichnis . . . . .	45
	Eidesstattliche Erklärung . . . . .	47
	Anhänge . . . . .	49
	Anhang A Quellcode . . . . .	49
	A.1 requirements.txt . . . . .	49
	A.2 charts.py . . . . .	49
	A.3 linreg.py . . . . .	51

## 1. Einleitung

In einer Zeit, in der datengetriebene Ansätze und automatisierte Modelle immer größere Relevanz erlangen, rückt die Notwendigkeit der Erklärbarkeit und Interpretierbarkeit von Modellen in den Vordergrund. Eines der vielversprechendsten Konzepte, das sich dieser Herausforderung annimmt, sind die sogenannten Shapley-Werte. Diese Masterarbeit erkundet die tiefgreifenden Konzepte der Shapley-Werte, ihre Anwendungen im Kontext von Machine Learning-Modellen und ihre praktische Umsetzung auf reale Datensätze.

Die Arbeit beginnt mit einer umfassenden Einführung in die Shapley-Werte und ihre historischen Wurzeln. Dabei wird insbesondere auf die kooperative Spieltheorie als Ursprung dieser Konzepte eingegangen. Anhand ausgewählter Literatur werden die theoretischen Grundlagen erörtert und der Forschungsstand auf diesem Gebiet aufgezeigt.

Im Anschluss daran wird die Brücke zur aktuellen Landschaft des maschinellen Lernens geschlagen. Es wird beleuchtet, wie die Shapley-Werte adaptiert werden können, um Einblicke in die Gewichtung von Merkmalen in komplexen Machine Learning-Modellen zu gewinnen. Dabei wird auf bestehende Methoden und Ansätze Bezug genommen und diskutiert, wie diese auf verschiedene Modelle angewendet werden können.

Ein zentraler Schwerpunkt der Arbeit liegt auf der praktischen Anwendung der Shapley-Werte. Ein realer Datensatz wird vorgestellt und die Methodik wird auf diesen angewendet, um die Wirksamkeit und Aussagekraft der Shapley-Werte in der Praxis zu evaluieren. Dies ermöglicht eine kritische Reflexion über die Stärken und Limitationen dieses Ansatzes im Kontext der Datenerklärung.

Abschließend werden die gewonnenen Erkenntnisse zusammengeführt und ein Ausblick auf zukünftige Entwicklungen und Forschungsrichtungen gegeben. Die Arbeit trägt somit dazu bei, das Verständnis für die Shapley-Werte als Instrument der Erklärbarkeit in komplexen Modellen zu vertiefen und ihre praktische Anwendbarkeit zu beleuchten.





## 2. Historischer Kontext und Begriffsdefinitionen

### 2.1. Die Genese der Shapley-Werte in der kooperativen Spieltheorie

Der Ursprung der Shapley-Werte liegt in der kooperativen Spieltheorie, einem fundamentalen Zweig der Spieltheorie. Dieser Bereich beschäftigt sich mit der Analyse von Situationen, in denen Akteure zusammenarbeiten, um gemeinsame Ziele zu erreichen. Zentrales Anliegen ist dabei die gerechte Verteilung der entstehenden Gewinne unter den Akteuren. Ein Schlüsselkonzept dieser Theorie ist die sogenannte „Charakteristische Funktion“, welche die Bewertung der Gewinnverteilung einer Koalition von Akteuren ermöglicht.

Die Shapley-Werte, entwickelt von Lloyd Shapley in den 1950er Jahren, bieten einen methodischen Ansatz, um den individuellen Beitrag eines jeden Akteurs zur kooperativen Zusammenarbeit gerecht zu bewerten. Dies geschieht durch die Durchschnittsbewertung der Beiträge über sämtliche mögliche Koalitionen hinweg. Diese Methode erweist sich als äußerst nützlich, um eine gerechte und rationale Verteilung von Gewinnen in vielfältigen Szenarien zu ermöglichen, sei es in wirtschaftlichen Verhandlungen oder der Aufteilung von Ressourcen.

Das Verständnis der kooperativen Spieltheorie und ihrer Anwendung in Form der Shapley-Werte ermöglicht es, dieses theoretische Konzept auf den Bereich des maschinellen Lernens zu übertragen. In dieser Arbeit wird der Übergang von abstrakten Spieltheorie-Konzepten zu konkreten Anwendungen in der Welt der datengetriebenen Modelle erforscht.

Zur Erreichung dieses Ziels werden in den kommenden Abschnitten nicht nur die formalen Definitionen und Eigenschaften der Shapley-Werte erläutert, sondern auch ihre Adaption und Anwendung auf Machine Learning-Modelle in Betracht gezogen. Die Anwendbarkeit wird durch die praktische Anwendung auf einen realen Datensatz verdeutlicht.

## 2.2. Shapley-Werte, SHAP, SHAP-Werte und shap

Zur Verdeutlichung und Abgrenzung der verschiedenen, jedoch verwandten Begrifflichkeiten, die im Kontext dieser Arbeit Verwendung finden, ist eine kurze Einordnung essenziell.

Beginnend mit den Shapley-Werten, entstammt dieser Begriff der kooperativen Spieltheorie und beschreibt eine Methode, um den fairen Beitrag eines Spielers zu der Gesamtauszahlung eines kooperativen Spiels zu bestimmen.

SHAP (SHapley Additive exPlanations) ist ein Interpretationsframework, das die Shapley-Werte in den Bereich des maschinellen Lernens überträgt. Der Begriff wurde erstmals von Lundberg und Lee eingeführt [LL17, S. 1].

Die SHAP-Werte sind dann die konkreten quantitativen Beiträge der einzelnen Merkmale zu einer bestimmten Vorhersage, berechnet basierend auf dem SHAP-Framework.

Das Python-Paket **shap** schließlich ist eine Implementierung, die es praktikabel macht, SHAP-Werte in der Anwendung zu berechnen und zu visualisieren. Es stellt eine reiche Auswahl an Werkzeugen zur Verfügung, um diese Werte und ihre Auswirkungen zu interpretieren.

### 3. Theorie der Shapley-Werte

In diesem Kapitel werden die Shapley-Werte als Instrument zur gerechten Aufteilung von Gewinnen in kooperativen Spielen vorgestellt. Durch die Verwendung eines praktischen Beispiels – der Aufteilung eines Preisgeldes aus einem Designwettbewerb unter den Gewinnern – wird zunächst eine intuitive Einführung in das Konzept gegeben. Anschließend wird die formale Definition der Shapley-Werte erläutert, um die theoretischen Grundlagen für ihre Berechnung und Anwendung zu legen.

#### 3.1. Wie lässt sich der Gewinn gerecht aufteilen?

Angenommen, drei Teilnehmer, Anna, Ben und Carla, haben als Team kooperiert und den ersten Platz bei einem Designwettbewerb belegt<sup>1</sup>. Dieser Erfolg führt zu einem Gesamtgewinn von 1000 €. Das Preisgeld für den zweiten Platz beträgt 750 € und 500 € für den dritten Platz. Die Herausforderung besteht nun darin, den Gewinn auf eine Weise zu verteilen, die den individuellen Beitrag jedes Teilnehmers zur Erzielung des ersten Platzes gerecht widerspiegelt.

Die Situation wird komplizierter, wenn man bedenkt, dass jeder Teilnehmer unterschiedlich zu dem Erfolg beigetragen hat und ihre individuellen Leistungen auch zu verschiedenen Ausgängen geführt hätten, wenn sie alleine oder in anderen Teilkonstellationen angetreten wären.

Um eine faire Aufteilung des Preisgeldes zu erreichen, betrachten wir die hypothetischen Gewinne, die Anna, Ben und Carla erzielt hätten, wenn sie in unterschiedlichen Konstellationen am Wettbewerb teilgenommen hätten. Tabelle 1 zeigt die gegebene Gewinnverteilung der verschiedenen Koalitionen. Die Koalition  $\emptyset$  entspricht dabei der leeren Koalition – der Nichtteilnahme an dem Wettbewerb.

Zur Berechnung der Shapley-Werte ist es erforderlich, den marginalen Beitrag jedes Spielers zu erfassen. Marginalbeiträge in der Spieltheorie, und speziell

---

<sup>1</sup>In Anlehnung an das Beispiel aus Kapitel 4.1 „Who’s going to pay for that taxi?“ [Mol23, S.17-20].

Tabelle 1.: Potenzielle Gewinne für verschiedene Teilnehmerkonstellationen im Designwettbewerb.

Koalition	Gewinn	Bemerkung
$\emptyset$	0 €	Keine Teilnahme
{Anna}	500 €	3. Platz als Einzelteilnehmerin
{Ben}	750 €	2. Platz als Einzelteilnehmer
{Carla}	0 €	Kein Gewinn als Einzelteilnehmerin
{Anna, Ben}	750 €	2. Platz als Team ohne Carla
{Anna, Carla}	750 €	2. Platz als Team ohne Ben
{Ben, Carla}	500 €	3. Platz als Team ohne Anna
{Anna, Ben, Carla}	1000 €	1. Platz als Gesamtteam

Quelle: Eigene Darstellung.

im Kontext der Shapley-Werte, sind die zusätzlichen Beiträge, die ein Spieler (Teilnehmer) zum Gesamtgewinn einer Koalition beiträgt, wenn er dieser beitrifft. Die Berechnung des marginalen Beitrags eines Teilnehmers erfolgt, indem man den Wert der Koalition ohne diesen Teilnehmer vom Wert der Koalition mit dem Teilnehmer subtrahiert [Mol23, S. 18].

In diesem Beispiel mit Anna, Ben und Carla, die an einem Designwettbewerb teilnehmen, ist der marginale Beitrag von Anna zur Koalition von {Ben} der zusätzliche Wert, den sie einbringt, wenn sie sich Ben anschließt, ausgehend von Bens individuellem Gewinn.

Tabelle 2.: Marginalbeiträge der einzelnen Teilnehmer zu den möglichen Koalitionen.

Teilnehmer	Zur Koalition	Gewinn vorher	Gewinn nachher	Marginalbeitrag
Anna	$\emptyset$	0 €	500 €	500 €
Anna	{Ben}	750 €	750 €	0 €
Anna	{Carla}	0 €	750 €	750 €
Anna	{Ben, Carla}	500 €	1000 €	500 €
Ben	$\emptyset$	0 €	750 €	750 €
Ben	{Anna}	500 €	750 €	250 €
Ben	{Carla}	0 €	500 €	500 €
Ben	{Anna, Carla}	750 €	1000 €	250 €
Carla	$\emptyset$	0 €	0 €	0 €
Carla	{Anna}	500 €	750 €	250 €
Carla	{Ben}	750 €	500 €	-250 €
Carla	{Anna, Ben}	750 €	1000 €	250 €

Quelle: Eigene Darstellung.

Die Tabelle 2 illustriert den Gewinn jeder möglichen Koalition ohne den betrachteten Spieler und den neuen Gesamtgewinn, sobald dieser Spieler der Koalition beitrifft. Der marginale Beitrag jedes Spielers wird dann als die Differenz zwischen diesen beiden Werten berechnet und gibt Aufschluss über den individuellen Wertbeitrag zum gemeinschaftlichen Erfolg.

Nachdem die marginalen Beiträge jedes Teilnehmers für die verschiedenen Koalitionen festgestellt wurden, ist der nächste Schritt, die Shapley-Werte zu bestimmen, welche eine faire Aufteilung des Gesamtgewinns erlauben. Hierzu wird jede mögliche Reihenfolge (Permutation) betrachtet, in der die Spieler der Koalition beitreten könnten. Jede dieser Permutationen liefert unterschiedliche marginale Beiträge für die Spieler, je nach der Reihenfolge ihres Beitritts [Sha53, S. 307ff].

Im Falle dieses Beispiels mit Anna, Ben und Carla bedeutet dies, dass alle möglichen Reihenfolgen berücksichtigt werden müssen, in denen sie zum ersten Platz beigetragen haben könnten. Die Shapley-Werte werden dann als Durchschnitt der marginalen Beiträge über alle Permutationen berechnet. Dies gewährleistet, dass jeder Spieler einen Anteil des Preisgeldes erhält, der seinem durchschnittlichen Beitrag zum Erfolg entspricht.

Bei drei Teilnehmern existieren  $3! = 3 \cdot 2 \cdot 1 = 6$  Permutationen:

1. Anna, Ben, Carla
2. Anna, Carla, Ben
3. Ben, Anna, Carla
4. Carla, Anna, Ben
5. Ben, Carla, Anna
6. Carla, Ben, Anna

Jede Permutation entspricht einer Koalitionsbildung. Anna wird in zwei Koalitionsbildungen (1. und 2.) einer leeren Koalition hinzugefügt. In weiteren zwei Koalitionsbildungen (5. und 6.) wird Anna der bestehenden Koalition aus Ben und Carla hinzugefügt. In den beiden übrigen Koalitionsbildungen wird Anna einmal der Koalition bestehend aus Ben (3.) und einmal der Koalition bestehend aus Carla (4.) hinzugefügt.

Daraus lässt sich nun der Shapley-Wert mit den gewichteten durchschnittlichen marginalen Beiträgen für Anna berechnen:

$$\frac{1}{6}(\underbrace{2 \cdot 500\text{€}}_{A \rightarrow \{\emptyset\}} + \underbrace{1 \cdot 0\text{€}}_{A \rightarrow \{B\}} + \underbrace{1 \cdot 750\text{€}}_{A \rightarrow \{C\}} + \underbrace{2 \cdot 500\text{€}}_{A \rightarrow \{B, C\}}) \approx 458,34\text{€} \quad (3.1)$$

Analog gilt das für Ben:

$$\frac{1}{6}(\underbrace{2 \cdot 750\text{€}}_{B \rightarrow \{\emptyset\}} + \underbrace{1 \cdot 250\text{€}}_{B \rightarrow \{A\}} + \underbrace{1 \cdot 500\text{€}}_{B \rightarrow \{C\}} + \underbrace{2 \cdot 250\text{€}}_{B \rightarrow \{A, C\}}) \approx 458,34\text{€} \quad (3.2)$$

und Carla:

$$\frac{1}{6}(\underbrace{2 \cdot 0\text{€}}_{C \rightarrow \{\emptyset\}} + \underbrace{1 \cdot 250\text{€}}_{C \rightarrow \{A\}} + \underbrace{1 \cdot (-250)\text{€}}_{C \rightarrow \{B\}} + \underbrace{2 \cdot 250\text{€}}_{C \rightarrow \{A, B\}}) \approx 83,34\text{€} \quad (3.3)$$

Auf Basis der gewichteten durchschnittlichen marginalen Beiträge lässt sich feststellen, dass Anna und Ben jeweils einen Shapley-Wert von ungefähr 458,34 € erhalten, während Carla einen Shapley-Wert von etwa 83,34 € zugewiesen bekommt. Diese Werte spiegeln den fairen Anteil jedes Teilnehmers an der Gesamtprämie wider, basierend auf ihrem individuellen Beitrag zum Erfolg des Teams. Mit dieser konkreten Anwendung der Shapley-Werte auf ein alltagsnahes Beispiel wird nun die zugrunde liegende Theorie und die formale Definition der Shapley-Werte, die diese Berechnungen ermöglichen, detaillierter betrachtet.

### 3.2. Formale Definition

Sei  $\mathcal{N} = \{1, \dots, n\}$  eine endliche Spielermenge mit  $n := |\mathcal{N}|$  Elementen. Sei  $v$  die Koalitionsfunktion, die jeder Teilmenge von  $\mathcal{N}$  eine reelle Zahl zuweist und insbesondere der leeren Koalition den Wert 0 gibt.

$$\begin{aligned} v &: \mathcal{P}(\mathcal{N}) \longrightarrow \mathbb{R} \\ &: v(\emptyset) \mapsto 0 \end{aligned}$$

Eine nicht leere Teilmenge der Spieler  $\mathcal{S} \subseteq \mathcal{N}$  heißt Koalition.  $\mathcal{N}$  selbst bezeichnet die große Koalition. Den Ausdruck  $v(\mathcal{S})$  nennt man den Wert der Koalition  $\mathcal{S}$ . Der Shapley-Wert ordnet nun jedem Spieler aus  $\mathcal{N}$  eine Auszahlung für das Spiel  $v$  zu.

Der marginale Beitrag eines Spieler  $i \in \mathcal{N}$ , also der Wertbeitrag eines Spielers zu einer Koalition  $\mathcal{S} \subseteq \mathcal{N}$ , durch seinen Beitritt, ist

$$v(\mathcal{S} \cup \{i\}) - v(\mathcal{S}). \quad (3.4)$$

Sei  $i = \text{Anna}$  und  $\mathcal{S} = \{\text{Ben}\}$ , dann ist  $v(\{\text{Ben}\} \cup \{\text{Anna}\}) - v(\{\text{Ben}\})$  das zusätzliche Preisgeld, welches gewonnen wird, wenn Anna der Koalition mit Ben beitrifft.

Der Shapley-Wert eines Spielers  $i$  errechnet sich als das gewichtete Mittel der

marginalen Beiträge zu allen möglichen Koalitionen:

$$\varphi_i(\mathcal{N}, v) = \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{i\}} \underbrace{\frac{|\mathcal{S}|! \cdot (n - 1 - |\mathcal{S}|)!}{n!}}_{\text{Gewicht}} \underbrace{v(\mathcal{S} \cup \{i\}) - v(\mathcal{S})}_{\text{marginaler Beitrag von Spieler } i \text{ zur Koalition } \mathcal{S}}. \quad (3.5)$$

Die Summationsnotation  $\sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{i\}}$  erfasst die marginalen Beiträge, die der Spieler  $i$  zu allen Koalitionen leistet, die diesen noch nicht einschließen. Die Verwendung von  $\mathcal{N} \setminus \{i\}$  stellt sicher, dass Spieler  $i$  nur für jene Koalitionen berücksichtigt wird, zu denen er noch beitragen kann. Im Falle von Anna etwa, beziehen sich die Berechnungen auf die Koalitionen bestehend aus der leeren Koalition  $\emptyset$ , aus  $\{\text{Ben}\}$ ,  $\{\text{Carla}\}$ , oder beiden zusammen  $\{\text{Ben}, \text{Carla}\}$  (vgl. Berechnung 3.1).

Die Formel  $\frac{|\mathcal{S}|! \cdot (n-1-|\mathcal{S}|)!}{n!}$  in der Shapley-Wert-Berechnung reflektiert den Gewichtungsfaktor für die marginalen Beiträge eines Spielers. Hierbei gibt  $|\mathcal{S}|!$  die Permutationen der Spieler innerhalb der Koalition  $\mathcal{S}$  an, während  $(n - 1 - |\mathcal{S}|)!$  die Anordnungen der außenstehenden Spieler repräsentiert, nachdem der betrachtete Spieler beigetreten ist. Der Bruchteil  $\frac{1}{n!}$  normalisiert diesen Wert über alle möglichen Koalitionszusammensetzungen, wodurch die Wahrscheinlichkeit der Bildung einer spezifischen Koalition ausgedrückt wird.

Betrachten wir Anna als den Spieler  $i$  und die Koalition  $\mathcal{S} = \{\text{Ben}, \text{Carla}\}$ . Die Formel  $\frac{|\mathcal{S}|! \cdot (n-1-|\mathcal{S}|)!}{n!}$  berechnet den Gewichtungsfaktor für Annas marginalen Beitrag zur Koalition  $\mathcal{S}$ . In diesem Fall ist  $|\mathcal{S}| = 2$  und  $n = 3$ . Somit ergibt sich  $|\mathcal{S}|! = 2!$  und  $n - 1 - |\mathcal{S}| = 0!$ , da nach dem Beitritt von Anna keine weiteren Spieler übrig sind. Der Normalisierungsfaktor ist  $n! = 3! = 6$ . Daraus folgt:

$$\frac{2! \cdot 0!}{3!} = \frac{2 \cdot 1}{6} = \frac{1}{3}. \quad (3.6)$$

Dies bedeutet, dass unter allen möglichen Permutationen der Spielerreihenfolge, Annas Beitritt zu der Koalition  $\{\text{Ben}, \text{Carla}\}$  genau ein Drittel der Zeit am Ende geschieht. Somit wird ihr marginaler Beitrag mit diesem Faktor gewichtet, um den Shapley-Wert zu berechnen (vgl. Berechnung 3.1) [Mol23, S. 21f].

### 3.3. Axiome

Nachdem die Berechnung des Shapley-Werts für das Beispiel konkretisiert wurde, ist es nun von Bedeutung, die zugrundeliegenden Axiome zu betrachten, welche die theoretische Rechtfertigung für die Methode liefern. Der Shapley-Wert wird nicht nur durch seine Berechnungsmethode, sondern auch durch eine Reihe von Axiomen charakterisiert, die seine Fairness und Kohärenz im Kontext kooperativer Spiele sicherstellen. Lloyd Shapley leitete den Shapley-Wert ursprünglich aus diesen Axiomen ab und bewies, dass dieser der einzige ist, der den Axiomen genügt<sup>2</sup>. Diese Axiome sind wesentliche Bestandteile, die die Einzigartigkeit und die wünschenswerten Eigenschaften des Shapley-Werts als Lösungskonzept definieren [Mol23, S. 22].

**Effizienz** Der Wert der großen Koalition wird an die Spieler verteilt:

$$\sum_{i \in \mathcal{N}} \varphi_i(\mathcal{N}, v) = v(\mathcal{N}). \quad (3.7)$$

Dies bedeutet, dass die Summe der Shapley-Werte aller Spieler dem Gesamtwert entspricht, den die Koalition aller Spieler zusammen erreichen kann. Der Gesamtwert, den die große Koalition  $\mathcal{N}$ , bestehend aus Anna, Ben und Carla, generiert, wird komplett unter den Spielern aufgeteilt [Mol23, S. 22]. Unter Vernachlässigung minimaler Rundungsdifferenzen entspricht die Summe der Shapley-Werte, berechnet in den Gleichungen 3.1, 3.2 und 3.3, dem kollektiven Ertrag der großen Koalition:

$$458,34\text{€} + 458,34\text{€} + 83,32\text{€} \approx 1000\text{€} \quad (3.8)$$

**Symmetrie** Zwei Spieler  $i$  und  $j$ , die die gleichen marginalen Beiträgen zu jeder Koalition haben erhalten das Gleiche:

$$v(\mathcal{S} \cup \{i\}) = v(\mathcal{S} \cup \{j\}), \forall \mathcal{S} \subseteq \mathcal{N} \setminus \{i, j\} \Rightarrow \varphi_i(\mathcal{N}, v) = \varphi_j(\mathcal{N}, v). \quad (3.9)$$

Obwohl Anna und Ben den gleichen Shapley-Wert erhalten, ist dies nicht auf

---

<sup>2</sup>Eine detaillierte Darstellung dieser Axiome und des Beweises ihrer Einzigartigkeit findet sich in Shapleys Originalarbeit, deren umfassende Behandlung jedoch den Rahmen dieser Arbeit überschreiten würde [Sha53, S. 307-318].



das Symmetrieaxiom zurückzuführen, da ihre marginalen Beiträge zu den Koalitionen variieren. Zum Beispiel leistet Anna keinen Beitrag zur Koalition, wenn Ben bereits Teil davon ist, während Ben einen positiven Beitrag leistet, wenn Anna bereits zur Koalition gehört (vgl. Tabelle 2). Dies zeigt, dass die Gleichheit ihrer Shapley-Werte ein Ergebnis der spezifischen Zahlenkonstellation in diesem Szenario ist und nicht aus der symmetrischen Interaktion zwischen den beiden Spielern resultiert.

**Null-Spieler-Eigenschaft (Dummy-Spieler-Eigenschaft)** Ein Spieler  $i$  der zu jeder Koalition nichts beiträgt erhält den Wert Null:

$$v(\mathcal{S} \cup \{i\}) = v(\mathcal{S}), \forall \mathcal{S} \subseteq \mathcal{N} \setminus \{i\} \Rightarrow \varphi_i(\mathcal{N}, v) = 0. \quad (3.10)$$

Dies stellt sicher, dass ein Spieler, der keinen Beitrag leistet, auch nicht belohnt wird.

**Additivität** Wenn das Spiel in zwei unabhängige Spiele mit Koalitionswerten  $v$  und  $w$  zerlegt werden kann, dann ist die Auszahlung jedes Spielers im zusammengesetzten Spiel die Summe der Auszahlungen in den aufgeteilten Spielen:

$$\varphi_i(\mathcal{N}, v + w) = \varphi_i(\mathcal{N}, v) + \varphi_i(\mathcal{N}, w). \quad (3.11)$$

Wenn Anna, Ben und Carla neben dem ersten Wettbewerb an einem zweiten, unabhängigen Wettbewerb teilnehmen, besagt das Additivitätsaxiom, dass die Shapley-Werte jedes Spielers aus beiden Wettbewerben einfach die Summe ihrer individuellen Shapley-Werte aus jedem einzelnen Wettbewerb sind. Dies impliziert, dass die faire Aufteilung der Gewinne aus beiden Wettbewerben konsistent bleibt, indem die aus dem ersten Wettbewerb abgeleiteten Prinzipien auf den zweiten Wettbewerb übertragen und dann addiert werden [RWB<sup>+</sup>22, Mol22, S. 5573, S.22f].



## 4. Von Shapley-Werten zu SHAP: Brückenschlag zur Modellinterpretation

Im Rahmen der kooperativen Spieltheorie ermöglichen die Shapley-Werte eine faire Verteilung des kollektiv erwirtschafteten Nutzens auf die beteiligten Akteure. Diese Methodik findet eine analoge Anwendung in der Welt des maschinellen Lernens, um die Beiträge einzelner Merkmale zur Vorhersageleistung eines Modells zu bewerten. Hier wird die Terminologie der Shapley-Werte in den Kontext von Machine Learning Modellen übertragen, wobei jedes Merkmal als „Spieler“ betrachtet wird, dessen Beitrag zur „Auszahlung“ – der Vorhersage des Modells – evaluiert werden soll.

Tabelle 3.: Terminologie der originären Shapley-Werte im Kontext des maschinellen Lernens.

Terminologie Konzept	Terminologie Machine Learning	Ausdruck
Spieler	Merkmal Index	$j$
Anzahl aller Spieler	Anzahl aller Merkmale	$p$
Große Koalition	Menge aller Merkmale	$\mathcal{N} = \{1, \dots, p\}$
Koalition	Menge von Merkmalen	$\mathcal{S} \subseteq \mathcal{N}$
Größe der Koalition	Anzahl der Merkmale in der Koalition $\mathcal{S}$	$ \mathcal{S} $
Spieler, die nicht in der Koalition sind	Merkmale, die nicht in der Koalition enthalten sind	$C : C = \mathcal{N} \setminus \mathcal{S}$
Koalitionsfunktion	Vorhersage für Merkmalswerte in der Koalition $\mathcal{S}$ abzüglich der Vorhersage im Mittel	$v_{f, x^{(i)}}(\mathcal{S})$
Auszahlung	Vorhersage für eine Beobachtung $x^{(i)}$ abzüglich der Vorhersage im Mittel	$f(x^{(i)}) - \mathbb{E}(f(X))$
Shapley-Wert	Beitrag des Merkmals $j$ zur Auszahlung des Modells für eine Beobachtung $x^{(i)}$	$\varphi_j^{(i)}(\mathcal{N}, f)$

Quelle: [Mol23, S. 26].

Die Koalitionsfunktion  $v_{f, x^{(i)}}(\mathcal{S})$  für ein gegebenes Model  $f$  und eine Beobachtung  $x^{(i)}$  ist definiert als:

$$v_{f,x^{(i)}}(\mathcal{S}) = \int_{\mathbb{R}} f(x_{\mathcal{S}}^{(i)} \cup X_C) d\mathbb{P}_{X_C} - \mathbb{E}(f(X)) \quad (4.1)$$

Diese Funktion berechnet den erwarteten Wert der Vorhersage des Modells  $f$ , wenn nur eine Teilmenge  $\mathcal{S}$  der Merkmale genutzt wird, um die Vorhersage für die spezifische Beobachtung  $x^{(i)} \in \mathbb{R}^p$  zu treffen. Das Integral  $\int_{\mathbb{R}}$  repräsentiert die Berechnung dieses erwarteten Wertes über alle möglichen Werte der Merkmale, die nicht in  $\mathcal{S}$  enthalten sind ( $X_C$ ), gewichtet durch deren Wahrscheinlichkeitsverteilung  $\mathbb{P}_{X_C}$ . Die Differenz zum Erwartungswert der Vorhersagen über alle Merkmale  $\mathbb{E}(f(X))$  zeigt, wie viel die spezifische Menge an Merkmalen  $\mathcal{S}$  zur Vorhersage beiträgt [Mol22, Mol23, S. 221, S. 27].

Der marginale Beitrag eines Merkmals  $j$  zu einer Koalition  $\mathcal{S}$  ist dann:

$$\begin{aligned} v_{f,x^{(i)}}(\mathcal{S} \cup \{j\}) - v_{f,x^{(i)}}(\mathcal{S}) &= \int_{\mathbb{R}} f(x_{\mathcal{S} \cup \{j\}}^{(i)} \cup X_{C \setminus \{j\}}) d\mathbb{P}_{X_{C \setminus \{j\}}} - \mathbb{E}(f(X)) \quad (4.2) \\ &\quad - \left( \int_{\mathbb{R}} f(x_{\mathcal{S}}^{(i)} \cup X_C) d\mathbb{P}_{X_C} - \mathbb{E}(f(X)) \right) \\ &= \int_{\mathbb{R}} f(x_{\mathcal{S} \cup \{j\}}^{(i)} \cup X_{C \setminus \{j\}}) d\mathbb{P}_{X_{C \setminus \{j\}}} \\ &\quad - \int_{\mathbb{R}} f(x_{\mathcal{S}}^{(i)} \cup X_C) d\mathbb{P}_{X_C} \end{aligned}$$

Diese Gleichung beschreibt, wie sich der erwartete Wert der Vorhersage ändert, wenn das Merkmal  $j$  zu der Menge der Merkmale  $\mathcal{S}$  hinzugefügt wird [Mol23, S. 29].

Der Beitrag  $\varphi_j^{(i)}(\mathcal{N}, f)$  eines Merkmals  $j$  für eine Beobachtung  $x^{(i)} \in \mathbb{R}^p$  für die Vorhersage  $f(x^{(i)})$  ist gegeben als:

$$\begin{aligned} \varphi_j^{(i)}(\mathcal{N}, f) &= \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{j\}} \frac{|\mathcal{S}|! \cdot (p - 1 - |\mathcal{S}|)!}{p!} \quad (4.3) \\ &\quad \cdot \left( \int_{\mathbb{R}} f(x_{\mathcal{S} \cup \{j\}}^{(i)} \cup X_{C \setminus \{j\}}) d\mathbb{P}_{X_{C \setminus \{j\}}} - \int_{\mathbb{R}} f(x_{\mathcal{S}}^{(i)} \cup X_C) d\mathbb{P}_{X_C} \right) \end{aligned}$$

Diese Formel ist die zentrale Berechnung der SHAP-Werte im maschinellen Lernen. Sie summiert den gewichteten, marginalen Beitrag des Merkmals  $j$  über alle möglichen Kombinationen der anderen Merkmale. Die Gewichtung

berücksichtigt die Anzahl der Merkmale in der Koalition  $\mathcal{S}$  und die Anzahl der verbleibenden Merkmale, die noch hinzugefügt werden können. Dies ergibt den durchschnittlichen Beitrag des Merkmals  $j$  zur Vorhersage für die Beobachtung  $x^{(i)}$  [Mol23, S. 29, 30].

Die Integration in der SHAP-Formel ist ein zentraler Schritt, um den erwarteten Beitrag jedes Merkmals unter Berücksichtigung der gesamten Verteilung der Daten zu ermitteln. In diesem Ansatz werden die Merkmale als Zufallsvariablen behandelt, und die Integration erfolgt über die Wahrscheinlichkeitsverteilungen dieser Zufallsvariablen. Durch das Berechnen der erwarteten Vorhersagewerte mit und ohne des jeweiligen Merkmals, unter Einbeziehung der Verteilung aller anderen Merkmale, ermöglicht SHAP eine präzise und umfassende Einschätzung des Einflusses jedes einzelnen Merkmals. Dieser Prozess der Marginalisierung, bei dem man über die Wahrscheinlichkeitsverteilungen der Merkmale integriert, erlaubt es, den Beitrag eines jeden Merkmals zu isolieren und unabhängig von der spezifischen Zusammensetzung der anderen Merkmale zu bewerten. Dies führt zu einer fairen und ganzheitlichen Bewertung der Beiträge aller Merkmale zur Vorhersage des Modells [Mol23, S. 28].

#### 4.1. Berechnung der SHAP-Werte unter Berücksichtigung der zugrundeliegenden Verteilung

Ein einfaches Beispiel soll helfen, die Anwendung von SHAP-Werten im Kontext des maschinellen Lernens zu illustrieren<sup>1</sup>. Betrachtet wird ein fiktiver Immobilien-Datensatz mit drei Merkmalen: Größe des Hauses in Quadratmetern ( $x_1$ ), Anzahl der Zimmer ( $x_2$ ) und Entfernung zum Stadtzentrum in Kilometern ( $x_3$ ). Es gibt zwei Beobachtungen in diesem Datensatz:

Tabelle 4.: Merkmale von Beobachtungen in einem Immobilien-Datensatz.

	$x_1$ : Größe (in $m^2$ )	$x_2$ : Anzahl Zimmer	$x_3$ : Entfernung zum Zentrum (in km)
$x^{(1)}$	100	3	5
$x^{(2)}$	150	4	10

Quelle: Eigene Darstellung.

Angenommen das Modell  $f(x^{(i)})$  prognostiziert den Preis eines Hauses in Euro als eine lineare Kombination der Merkmale:

$$f(x^{(i)}) = 5x_1^{(i)} + 20x_2^{(i)} - 2x_3^{(i)}. \quad (4.4)$$

<sup>1</sup>In Anlehnung an das Beispiel aus Kapitel 8.5.1 „General Idea“ [Mol22, S.215f].

Die Vorhersagen für die beiden Beobachtungen lauten dann:

$$\begin{aligned} f(x^{(1)}) &= 5x_1^{(1)} + 20x_2^{(1)} - 2x_3^{(1)} \\ &= 5 \cdot 100 + 20 \cdot 3 - 2 \cdot 5 \\ &= 550 \text{ €} \end{aligned} \tag{4.5}$$

und

$$\begin{aligned} f(x^{(2)}) &= 5x_1^{(2)} + 20x_2^{(2)} - 2x_3^{(2)} \\ &= 5 \cdot 150 + 20 \cdot 4 - 2 \cdot 10 \\ &= 810 \text{ €}. \end{aligned} \tag{4.6}$$

Die erwartete Auszahlung des Modells  $\mathbb{E}(f(X))$  wird berechnet als:

$$\begin{aligned} \mathbb{E}(f(X)) &= 5 \cdot \mathbb{E}(X_1) + 20 \cdot \mathbb{E}(X_2) - 2 \cdot \mathbb{E}(X_3) \\ &= 5 \cdot 125 + 20 \cdot 3,5 - 2 \cdot 7,5 \\ &= 680 \text{ €}, \end{aligned} \tag{4.7}$$

mit

$$\mathbb{E}(X_j) = \frac{1}{n} \sum_{i=1}^n x_j^{(i)}. \tag{4.8}$$

Sei  $\mathcal{N} = \{1, 2, 3\}$  die Menge aller Merkmale und die Beobachtung  $x^{(1)} = [100, 3, 5]$ . Der SHAP-Wert für jedes Merkmal  $j \in \mathcal{N}$  wird unter Berücksichtigung der Verteilung der Daten und der Formel 4.3 berechnet:

$$\begin{aligned} \varphi_j^{(1)}(\mathcal{N}, f) &= \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{j\}} \frac{|\mathcal{S}|! \cdot (p - 1 - |\mathcal{S}|)!}{p!} \\ &\quad \cdot \left( \int_{\mathbb{R}} f(x_{\mathcal{S} \cup \{j\}}^{(1)} \cup X_{\mathcal{C} \setminus \{j\}}) d\mathbb{P}_{X_{\mathcal{C} \setminus \{j\}}} - \int_{\mathbb{R}} f(x_{\mathcal{S}}^{(1)} \cup X_{\mathcal{C}}) d\mathbb{P}_{X_{\mathcal{C}}} \right) \end{aligned} \tag{4.9}$$

wobei  $p = |\mathcal{N}| = 3$  die Anzahl der Merkmale ist und  $X_C$  die Menge der Merkmale außerhalb der Koalition  $\mathcal{S}$  repräsentiert. Die Integrale repräsentieren die erwartete Vorhersage des Modells über die Verteilung der nicht in der Koalition enthaltenen Merkmale.

In linearen Modellen, unter der Prämisse, dass die Merkmale unabhängig voneinander und gleichverteilt sind, ist es möglich, die Berechnung der SHAP-Werte zu vereinfachen. Anstelle der komplexen Integration über die Verteilungen aller Merkmale, kann der Fokus auf die Unterschiede in den Modellvorhersagen gelegt werden, die sich aus dem Hinzufügen oder Entfernen einzelner Merkmale ergeben. Hierbei wird anstelle der spezifischen Werte der nicht in der betrachteten Koalition enthaltenen Merkmale Erwartungswerte herangezogen. Diese Vereinfachung ermöglicht es, den Einfluss jedes Merkmals auf die Modellvorhersage auf eine direktere und rechnerisch weniger aufwendige Weise zu erfassen. Diese Vereinfachung ist für lineare Modelle angemessen, da die Auswirkungen jedes Merkmals auf die Vorhersage des Modells additiv und unabhängig sind. Bei komplexeren, nichtlinearen Modellen ist eine detailliertere Berechnung erforderlich, die oft auf numerischen Methoden oder Annäherungen basiert, mehr dazu in Kapitel 5.

Der Beitrag durch das Hinzufügen des Merkmals  $x_1$  zur bestehenden Koalition  $\mathcal{S} = \{x_2\}$  wird nach Formel 4.9 berechnet als:

$$\begin{aligned} \varphi_1^{(1)}(\{x_2\}, f) &= \frac{1! \cdot (3 - 1 - 1)!}{3!} \\ &\quad \cdot \int f(x_1, x_2, X_3) d\mathbb{P}(X_3) - \int f(X_1, x_2, X_3) d\mathbb{P}(X_1, X_3) \end{aligned} \quad (4.10)$$

Da  $X_1$  und  $X_3$  unabhängig und gleichmäßig verteilt sind, können  $X_2$  und  $X_3$  durch ihre Erwartungswerte (Gleichung 4.8) ersetzt werden:

$$\begin{aligned} \varphi_1^{(1)}(\{x_2\}, f) &= \frac{1}{6} \left( f(x_1, x_2, \mathbb{E}(X_3)) - f(\mathbb{E}(X_1), x_2, \mathbb{E}(X_3)) \right) \\ &= \frac{1}{6} \left( f(100, 3, 7.5) - f(125, 3, 7.5) \right) \\ &= \frac{1}{6} \left( (5 \cdot 100 + 20 \cdot 3 - 2 \cdot 7, 5) - (5 \cdot 125 + 20 \cdot 3 - 2 \cdot 7, 5) \right) \\ &= \frac{1}{6} (545 - 670) \\ &= \frac{1}{6} (-125) \end{aligned} \quad (4.11)$$

Die in Tabelle 5 dargestellten Kombinationen illustrieren die marginalen Beiträge und SHAP-Werte für jedes Merkmal in jeder möglichen Koalition von Merkmalen, bezogen auf die Beobachtung  $x^{(1)}$ .

Tabelle 5.: Marginalbeiträge der einzelnen Merkmale zu den möglichen Koalitionen für die Beobachtung  $x^{(1)}$ .

$x_j$	$\mathcal{S}$	$v_{f,x^{(1)}}(\mathcal{S})$	$v_{f,x^{(1)}}(\mathcal{S} \cup \{j\})$	$v_{f,x^{(1)}}(\mathcal{S} \cup \{j\}) - v_{f,x^{(1)}}(\mathcal{S})$	Gewicht	$\varphi_j^{(1)}(\mathcal{S}, f)$
$x_1$	$\emptyset$	680	555	-125	$\frac{1}{3}$	-41,67
$x_1$	$\{x_2\}$	670	545	-125	$\frac{1}{6}$	-20,83
$x_1$	$\{x_3\}$	685	560	-125	$\frac{1}{6}$	-20,83
$x_1$	$\{x_2, x_3\}$	675	550	-125	$\frac{1}{3}$	-41,67
$x_2$	$\emptyset$	680	670	-10	$\frac{1}{3}$	-3,33
$x_2$	$\{x_1\}$	555	545	-10	$\frac{1}{6}$	-1,67
$x_2$	$\{x_3\}$	685	675	-10	$\frac{1}{6}$	-1,67
$x_2$	$\{x_1, x_3\}$	560	550	-10	$\frac{1}{3}$	-3,33
$x_3$	$\emptyset$	680	685	5	$\frac{1}{3}$	1,67
$x_3$	$\{x_1\}$	555	560	5	$\frac{1}{6}$	0,83
$x_3$	$\{x_2\}$	670	675	5	$\frac{1}{6}$	0,83
$x_3$	$\{x_1, x_2\}$	545	550	5	$\frac{1}{3}$	1,67

Quelle: Eigene Darstellung.

Diese Analyse ist ebenso auf die Beobachtung  $x^{(2)}$  anwendbar und erfordert eine analoge Vorgehensweise:

Tabelle 6.: Marginalbeiträge der einzelnen Merkmale zu den möglichen Koalitionen für die Beobachtung  $x^{(2)}$ .

$x_j$	$\mathcal{S}$	$v_{f,x^{(2)}}(\mathcal{S})$	$v_{f,x^{(2)}}(\mathcal{S} \cup \{j\})$	$v_{f,x^{(2)}}(\mathcal{S} \cup \{j\}) - v_{f,x^{(2)}}(\mathcal{S})$	Gewicht	$\varphi_j^{(2)}(\mathcal{S}, f)$
$x_1$	$\emptyset$	680	805	125	$\frac{1}{3}$	41,67
$x_1$	$\{x_2\}$	690	815	125	$\frac{1}{6}$	20,83
$x_1$	$\{x_3\}$	675	800	125	$\frac{1}{6}$	20,83
$x_1$	$\{x_2, x_3\}$	685	810	125	$\frac{1}{3}$	41,67
$x_2$	$\emptyset$	680	690	10	$\frac{1}{3}$	3,33
$x_2$	$\{x_1\}$	805	815	10	$\frac{1}{6}$	1,67
$x_2$	$\{x_3\}$	675	685	10	$\frac{1}{6}$	1,67
$x_2$	$\{x_1, x_3\}$	800	810	10	$\frac{1}{3}$	3,33
$x_3$	$\emptyset$	680	675	-5	$\frac{1}{3}$	-1,67
$x_3$	$\{x_1\}$	805	800	-5	$\frac{1}{6}$	-0,83
$x_3$	$\{x_2\}$	690	685	-5	$\frac{1}{6}$	-0,83
$x_3$	$\{x_1, x_2\}$	815	810	-5	$\frac{1}{3}$	-1,67

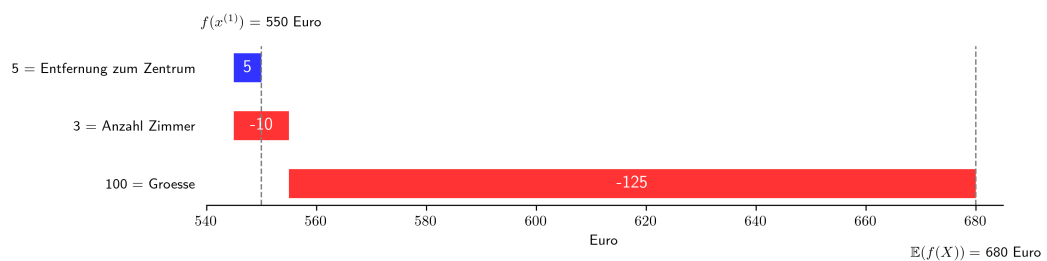
Quelle: Eigene Darstellung.

Die in den Tabellen 5 und 6 dargestellten SHAP-Werte für die Beobachtungen  $x^{(1)}$  und  $x^{(2)}$  zeigen eine interessante Symmetrie. Für jedes Merkmal entspricht der SHAP-Wert für  $x^{(2)}$  genau dem negativen Wert für  $x^{(1)}$ . Die beobachtete Inversion der SHAP-Werte zwischen den Beobachtungen  $x^{(1)}$  und  $x^{(2)}$  lässt sich durch die zentrale Rolle des Erwartungswerts des Modells erklären. Das Modell prognostiziert im Mittel einen Immobilienpreis von 680 €. In einem Szenario,



in dem nur zwei Beobachtungen vorhanden sind, spiegeln die Beobachtungen  $x^{(1)}$  und  $x^{(2)}$  entgegengesetzte Abweichungen vom Erwartungswert wider. Die SHAP-Werte, die den Beitrag jedes Features zur Abweichung der Vorhersage vom Mittelwert messen, zeigen daher für  $x^{(1)}$  und  $x^{(2)}$  genau entgegengesetzte Werte. Dieses Phänomen resultiert aus der Tatsache, dass die Merkmalsbeiträge in Bezug auf den Erwartungswert berechnet werden und unsere beispielhaften Beobachtungen symmetrisch um diesen Mittelwert verteilt sind. Folglich heben sich ihre Beiträge im Kontext unseres linearen Modells exakt auf, was die Konsistenz und Zuverlässigkeit der SHAP-Wert-Berechnung unterstreicht.

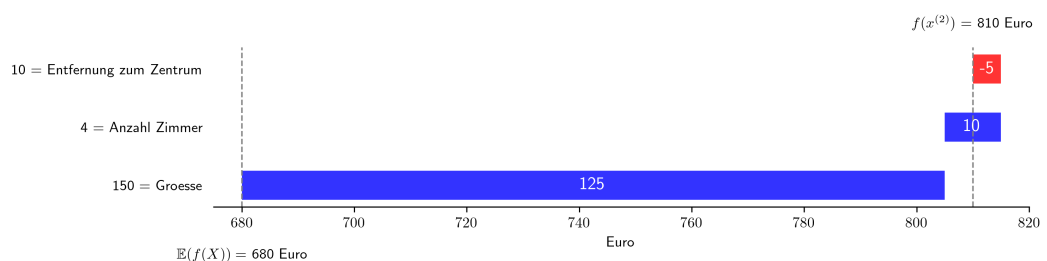
Abbildung 1.: Beitrag der Merkmale  $x_{j \in \{1,2,3\}}$  zur Modellvorhersage  $f(x^{(1)})$ .



Quelle: Eigene Darstellung, A.2.

Das Modell  $f(x^{(i)})$  prognostiziert im Mittel einen Immobilienpreis von 680 €. Im Vergleich zur Verteilung des jeweiligen Merkmals, reduziert die Größe der Wohnung ( $x_1^{(1)}$ ) und die Anzahl der Zimmer ( $x_2^{(1)}$ ) die Prognose des Preises für die Immobilie  $x^{(1)}$  um insgesamt 135 €, während die Entfernung zum Stadtzentrum ( $x_3^{(1)}$ ) den Preis der Wohnung um 5 € erhöht, wie in Abbildung 1 veranschaulicht. Abbildung 2 visualisiert die SHAP-Werte für die Immobilie  $x^{(2)}$ :

Abbildung 2.: Beitrag der Merkmale  $x_{j \in \{1,2,3\}}$  zur Modellvorhersage  $f(x^{(2)})$ .



Quelle: Eigene Darstellung, A.2.

## 4.2. Axiome

Die in Tabellen 5 und 6 präsentierten Ergebnisse bieten eine Grundlage, um die Konformität der SHAP-Werte mit den etablierten Axiomen der Shapley-Werte, wie sie im Kapitel 3.3 diskutiert wurden, zu beurteilen. Die Axiome der SHAP-Werte stellen eine adaptierte und kontextualisierte Anwendung dieser Prinzipien auf die Interpretation von Modellvorhersagen dar [LL17].

**Effizienz** Das Effizienzaxiom besagt, dass die Summe der SHAP-Werte aller Features für eine gegebene Beobachtung  $x^{(i)}$  gleich der Differenz zwischen der Modellvorhersage für diese Beobachtung  $f(x^{(i)})$  und der durchschnittlichen Modellvorhersage  $\mathbb{E}(f(X))$  sein muss:

$$\sum_{j=1}^p \varphi_j^{(i)}(\mathcal{N}, f) = f(x^{(i)}) - \mathbb{E}(f(X)), \quad (4.12)$$

[Mol22, S. 221]. Für die Beobachtung  $x^{(1)}$  aus Kapitel 4.1 und den Berechnungen für  $f(x^{(1)})$  (Gleichung 4.5), sowie  $\mathbb{E}(f(X))$  (Gleichung 4.7) ergibt sich:

$$\begin{aligned} \sum_{j=1}^3 \varphi_j^{(1)}(\mathcal{N}, f) &= -130 \\ f(x^{(1)}) - \mathbb{E}(f(X)) &= 550 - 680 = -130, \end{aligned} \quad (4.13)$$

womit das Effizienzaxiom erfüllt ist. Die Differenz der Vorhersage einer konkreten Beobachtung zur durchschnittlichen Modellvorhersage wird auf alle Merkmale verteilt.

**Symmetrie** Das Symmetrieaxiom fordert, dass zwei Merkmale  $i$  und  $j$ , die in jeder Koalition denselben Beitrag leisten, auch denselben SHAP-Wert erhalten müssen. In dem hier betrachteten Fall der Immobilienpreisprognose würde dies bedeuten, dass wenn zwei Merkmale, beispielsweise die Größe einer Wohnung und die Anzahl der Zimmer, immer den gleichen Einfluss auf den Preis hätten, unabhängig von der Kombination anderer Merkmale, ihre SHAP-Werte identisch sein müssen:

$$v(\mathcal{S} \cup \{i\}) = v(\mathcal{S} \cup \{j\}), \forall \mathcal{S} \subseteq \mathcal{N} \setminus \{i, j\} \Rightarrow \varphi_i(\mathcal{N}, v) = \varphi_j(\mathcal{N}, v), \quad (4.14)$$

[Mol22, S. 221]. Dies wird durch die Tabellen 5 und 6 nicht illustriert, da jedes Merkmal einen unterschiedlichen Beitrag liefert, was die Anwendung dieses Axioms in diesem speziellen Fall ausschließt.

**Null-Spieler-Eigenschaft (Dummy-Spieler-Eigenschaft)** Ein Merkmal  $i$ , das keinen Einfluss auf die Modellvorhersage hat, erhält gemäß der Null-Spieler-Eigenschaft einen SHAP-Wert von Null. Im Kontext des Beispiels würde ein Merkmal, das keine Veränderung in der Vorhersage bewirkt, unabhängig von den anderen Merkmalen, einen SHAP-Wert von Null erhalten:

$$v(\mathcal{S} \cup \{i\}) = v(\mathcal{S}), \forall \mathcal{S} \subseteq \mathcal{N} \setminus \{i\} \Rightarrow \varphi_i(\mathcal{N}, v) = 0, \quad (4.15)$$

[Mol22, S. 222]. In der fiktiven Datenlage der Tabellen 5 und 6 hat jedes Merkmal einen gewissen Einfluss, sodass die Null-Spieler-Eigenschaft hier nicht beobachtet werden kann.

**Additivität** Das Additivitätsaxiom ist ein zentrales Konzept, das die Konsistenz von Shapley-Werten über die Zusammensetzung von Spielen hinweg beschreibt. Es garantiert, dass für zwei separate Spiele oder Modelle  $v$  und  $w$ , die Summe der SHAP-Werte eines Merkmals über beide Spiele seinem SHAP-Wert im kombinierten Spiel entspricht:

$$\varphi_i(\mathcal{N}, v + w) = \varphi_i(\mathcal{N}, v) + \varphi_i(\mathcal{N}, w). \quad (4.16)$$

In Machine Learning-Modellen wie dem Random Forest, besonders bei Ensemble-Modellen, ist die Additivität relevant. Ein Random Forest besteht aus unabhängigen Entscheidungsbäumen, die zusammenarbeiten, um Vorhersagen zu treffen. Die SHAP-Werte der einzelnen Bäume können als additive Beiträge betrachtet werden, die den Gesamteinfluss eines Merkmals auf die Modellvorhersage zusammenfassen. [Mol23, S. 32].



## 5. Komplexitätsbewältigung bei der Berechnung von SHAP-Werten

Die direkte Berechnung von SHAP-Werten kann bei Modellen mit einer großen Anzahl an Merkmalen rechnerisch anspruchsvoll sein. Im Kontext linearer Modelle, wie dem in dieser Arbeit verwendeten Immobilienpreis-Beispiel, ist die vollständige Ermittlung aller möglichen Kombinationen von Merkmalen und deren Beiträgen jedoch nicht notwendig.

In einem linearen Modell sind die Beiträge der einzelnen Merkmale zur Vorhersage additiv und unabhängig voneinander. Dies ermöglicht eine direkte Ableitung der Beiträge jedes Merkmals aus den Modellkoeffizienten. Aufgrund dieser Eigenschaft liefert jedes Merkmal einen konstanten marginalen Beitrag, unabhängig von der Zusammensetzung der restlichen Koalition von Merkmalen. Theoretisch könnte daher für jedes Merkmal nur eine Koalition berechnet werden, um den SHAP-Wert zu bestimmen [Mol23, S. 38]. Dies bestätigen die Tabellen 5 und 6 und die daraus ersichtlichen marginalen Beiträge, die über die jeweiligen Merkmale unabhängig der ihnen beigetretenen Koalition konstant sind. So ist der marginale Beitrag für Merkmal  $x_1^{(1)}$  zu allen möglichen Koalitionen stets  $-125$ .

Die Herausforderung bei der Berechnung von SHAP-Werten in realen Szenarien erwächst aus zwei zentralen Problembereichen: der Anzahl der Merkmale und der Notwendigkeit, über diese Merkmale zu integrieren.

Die Anzahl möglicher Koalitionen von Merkmalen wächst exponentiell mit  $2^p$  an, was bei einer großen Anzahl von Merkmalen  $p$  schnell zu einer nicht handhabbaren Berechnungskomplexität führt. Hinzu kommt die Schwierigkeit, dass für eine exakte Berechnung der SHAP-Werte eine Integration über die Merkmalsverteilungen erforderlich ist, was jedoch voraussetzt, dass die Verteilung der Merkmale bekannt ist. In der Praxis verfügt man meist nur über eine Stichprobe der Daten, ohne genaue Kenntnis der zugrundeliegenden Verteilungen [Mol23, S. 33].

Um diesen Herausforderungen zu begegnen, wird in diesem Kapitel die Anwendung von Monte Carlo Integration zur Schätzung von SHAP-Werten diskutiert.

Diese Methoden ermöglichen es, durch Zufallsstichproben aus den vorhandenen Daten eine Approximation der Verteilungen zu erstellen und dadurch die notwendigen Integrationen näherungsweise auszuführen [Mol23, S. 34]. Darüber hinaus werden Ansätze zur Stichprobenbildung von Koalitionen vorgestellt, die es erlauben, die rechnerische Last zu reduzieren, indem nicht alle Koalitionen betrachtet, sondern repräsentative Stichproben gezogen werden.

### 5.1. Approximation der marginalen Beiträge mittels Monte Carlo Integration

Statt das Integral über eine unbekannte Verteilung zu berechnen, wie in Gleichung 4.1, nähert die Monte Carlo Integration dieses Integral durch den Durchschnitt einer großen Anzahl zufällig ausgewählter Beobachtungen aus dem Eingaberaum an [Mol23, S. 34]. Die Monte Carlo Integration kann als ein erwartungstreuer Schätzer betrachtet werden, wenn die Anzahl der Zufallsstichproben  $n$  hinreichend groß ist. Dies bedeutet, dass mit zunehmendem  $n$  die Schätzung des Integrals immer genauer wird und gegen den wahren Wert des Integrals konvergiert. Dies basiert auf dem Gesetz der großen Zahlen, das besagt, dass der Durchschnitt einer großen Anzahl unabhängiger und identisch verteilter Zufallsvariablen gegen den Erwartungswert der Verteilung konvergiert [RC04, S. 83].

Die Zufallsvariable  $X_C^{(k)}$ , die Merkmale die nicht in der Koalition  $\mathcal{S}$  enthalten sind, entfällt und wird durch konkrete Beobachtungen  $x_C^{(k)}$  aus der Datenbasis ersetzt, was eine wünschenswerte Vereinfachung darstellt. Das Integral  $\int$  wird dadurch zur Summe  $\sum$  und die Verteilung  $\mathbb{P}$  wird durch eine große Anzahl zufällig ausgewählter Beobachtungen ersetzt und das Ergebnis anschließend über alle ausgewählten Beobachtungen  $n$  gemittelt.  $\hat{v}_{x^{(i)},f}(\mathcal{S})$  ist somit ein Schätzer für den Wert der Koalition  $\mathcal{S}$  und definiert als:

$$\hat{v}_{x^{(i)},f}(\mathcal{S}) = \frac{1}{n} \sum_{k=1}^n \left( f(x_{\mathcal{S}}^{(i)} \cup x_C^{(k)}) - f(x_C^{(k)}) \right) \quad (5.1)$$

Analog zu Gleichung 4.2 ist dann zusammen mit Gleichung 5.1 der marginale Beitrag  $\hat{\Delta}_{\mathcal{S},j}$  von Merkmal  $j$  zur Koalition  $\mathcal{S}$  gegeben als:

$$\begin{aligned}\hat{\Delta}_{\mathcal{S},j} &= \hat{v}_{x^{(i)},f}(\mathcal{S} \cup \{j\}) - \hat{v}_{x^{(i)},f}(\mathcal{S}) \\ &= \frac{1}{n} \sum_{k=1}^n \left( f(x_{\mathcal{S} \cup \{j\}}^{(i)} \cup x_{C \setminus \{j\}}^{(k)}) - f(x_{\mathcal{S}}^{(i)} \cup x_C^{(k)}) \right)\end{aligned}\tag{5.2}$$

und die SHAP-Werte  $\hat{\varphi}_j^{(i)}$  über alle möglichen Koalitionen analog zu Gleichung 4.3:

$$\hat{\varphi}_j^{(i)}(\mathcal{N}, f) = \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{j\}} \frac{|\mathcal{S}|! \cdot (p - 1 - |\mathcal{S}|)!}{p!} \hat{\Delta}_{\mathcal{S},j},\tag{5.3}$$

[Mol23, S.36].

## 5.2. Schätzung von Koalitionen

Obwohl die Monte Carlo Integration aus Gleichung 5.3 eine praktische Methode zur Approximation der benötigten marginalen Beiträge in der SHAP-Wertberechnung bietet, bleibt die Herausforderung, alle möglichen Koalitionen der Features zu evaluieren. Die Anzahl der Koalitionen wächst exponentiell mit der Anzahl der Features. Daher sind Ansätze gefragt, die eine effiziente Schätzung der SHAP-Werte erlauben, ohne jede mögliche Koalition explizit zu berücksichtigen.

In diesem Abschnitt werden zwei solcher Ansätze diskutiert: der Linear Estimator und der Kernel Estimator. Beide Verfahren bieten Strategien zur Reduzierung der Berechnungskomplexität, indem sie Annahmen über die Eigenschaften des Modells und der Daten treffen. Weitere Ansätze werden in Kapitel 6.2 angesprochen.

Der Kernel Estimator verwendet Gewichtungen für die verschiedenen Koalitionen, um ihre relative Wichtigkeit bei der Berechnung der SHAP-Werte zu berücksichtigen. Dieser Ansatz ermöglicht es, mit einer kleineren Anzahl von sorgfältig ausgewählten Koalitionen zu arbeiten, indem er diesen Koalitionen mehr Gewicht bei der Schätzung gibt.

Der Linear Estimator hingegen nähert das Verhalten des komplexen Modells in der Nähe einer gegebenen Beobachtung durch ein lineares Modell an. Diese Näherung basiert auf der Annahme, dass die lokalen Beiträge der Features

zur Modellvorhersage durch lineare Beziehungen hinreichend genau beschrieben werden können. Dadurch wird die Berechnung der SHAP-Werte auf die Lösung eines linearen Gleichungssystems reduziert, was deutlich weniger rechenaufwendig ist als die vollständige Kombinatorik.

Beide Methoden stellen Kompromisse zwischen Berechnungsaufwand und Genauigkeit dar und werden im Folgenden näher erläutert.

### 5.2.1. Linearer SHAP Estimator für lineare Modelle

Für lineare Modelle bietet der lineare SHAP Estimator eine direkte und effiziente Methode zur Berechnung der SHAP-Werte. Diese Methode beruht auf der Annahme der Unabhängigkeit der Eingabemerkmale. Unter dieser Voraussetzung können die SHAP-Werte direkt aus den Gewichtungskoeffizienten des linearen Modells abgeleitet werden.

Ein lineares Modell wird typischerweise in der Form  $f(x) = \sum_{j=1}^p w_j x_j + b$  ausgedrückt, wobei  $w_j$  der Gewichtungskoeffizient für das Merkmal  $j$  ist und  $b$  den Achsenabschnitt (Bias) darstellt. Der lineare SHAP Estimator nutzt diese Koeffizienten, um die Beiträge jedes Features zur Modellvorhersage zu bestimmen. Der SHAP-Wert für ein Feature  $j$  wird dann definiert als:

$$\varphi_j(f, x) = w_j(x_j - \mathbb{E}[x_j]), \quad (5.4)$$

wobei  $\mathbb{E}[x_j]$  der Erwartungswert des Merkmals  $j$  über den Datensatz ist. Der SHAP-Wert für den Achsenabschnitt (Bias) ist gleich dem Achsenabschnitt des Modells:  $\varphi_0(f, x) = b$  [LL17, S. 6].

Diese Berechnung der SHAP-Werte basiert auf der Idee, dass die Änderung des Modelloutputs, die durch das Abweichen eines Merkmals von seinem Erwartungswert entsteht, direkt durch den entsprechenden Gewichtungskoeffizienten des linearen Modells beschrieben werden kann. Dieser Ansatz ermöglicht eine schnelle und genaue Berechnung der SHAP-Werte für lineare Modelle und ist besonders nützlich, um die Auswirkungen einzelner Features in einem solchen Modell zu interpretieren.

### 5.2.2. Kernel SHAP Estimator: Modell-agnostischer Ansatz



## 6. Praktische Anwendung von SHAP auf lineare Modelle

In diesem Kapitel wird der Einsatz des SHAP-Frameworks zur Interpretation linearer Modelle im Kontext des maschinellen Lernens untersucht. Lineare Modelle, gekennzeichnet durch ihre Transparenz und einfache Struktur, bilden oft die Basis für das Verständnis komplexerer Algorithmen. Dennoch bleibt die Herausforderung bestehen, die Beiträge individueller Merkmale zur Modellvorhersage zu quantifizieren und zu interpretieren.

Die Anwendung von SHAP-Werten ermöglicht es, diesen Herausforderungen zu begegnen und Einblicke in die Modellvorhersagen zu gewähren, die über traditionelle Methoden hinausgehen. Dieses Kapitel führt in die Grundlagen des `shap`-Pakets ein, demonstriert dessen Anwendung auf einen spezifischen Datensatz und diskutiert die Berechnung sowie Interpretation der resultierenden SHAP-Werte. Die daraus gewonnenen Erkenntnisse leisten einen Beitrag zur Erklärbarkeit von Vorhersagemodellen und unterstützen somit die wissenschaftliche Diskussion um die Verantwortlichkeit und Nachvollziehbarkeit in der maschinellen Lernforschung.

### 6.1. Lineare Modelle als analytische Grundlage

In linearen Regressionsmodellen wird die Zielgröße als eine gewichtete Kombination der Eingangsmerkmale bestimmt. Die einfache lineare Struktur dieser Modelle erleichtert das Verständnis der Beziehungen zwischen den Eingangsdaten und den Vorhersagen.

Lineare Modelle sind ein grundlegendes Werkzeug in der statistischen Modellierung und dienen dazu, das Verhältnis zwischen einer abhängigen Variablen, die üblicherweise mit  $y^{(i)}$  bezeichnet wird, und einem oder mehreren Prädiktoren, den unabhängigen Variablen  $x_i$ , zu erfassen. Diese Beziehungen werden mittels linearer Gleichungen dargestellt, die für jede einzelne Beobachtung  $i$  im Datensatz folgendermaßen formuliert werden können:

$$y^{(i)} = \beta_0 + \sum_{j=1}^p \beta_j x_j^{(i)} + \epsilon^{(i)}, \quad (6.1)$$

wobei das Ergebnis, das von einem linearen Modell für eine gegebene Beobachtung vorhergesagt wird, sich als Summe der mit Gewichten  $\beta_j$  versehenen Merkmale  $p$  ergibt.

Hierbei stellt  $y^{(i)}$  den beobachteten Wert der abhängigen Variablen für die Beobachtungseinheit  $i$  dar. Der Term  $\beta_0$  ist der Achsenabschnitt oder y-Achsenabschnitt des Modells, welcher den erwarteten Wert von  $y$  darstellt, wenn alle unabhängigen Variablen  $x$  null sind. Die Summe  $\sum_{j=1}^p \beta_j x_j^{(i)}$  berechnet sich aus den Produkten der Koeffizienten  $\beta_j$  und den Werten der unabhängigen Variablen  $x_j^{(i)}$  für jede Beobachtungseinheit  $i$  und jeden Prädiktor  $j$ , wobei die Koeffizienten  $\beta_j$  den geschätzten Einfluss der entsprechenden unabhängigen Variablen auf die abhängige Variable beschreiben.

Der Fehlerterm  $\epsilon^{(i)}$  steht für die Residuen, also die Differenzen zwischen den beobachteten und durch das Modell geschätzten Werten von  $y^{(i)}$ . Es wird angenommen, dass diese Fehler normalverteilt sind, was bedeutet, dass Abweichungen in beiden Richtungen um den Mittelwert (hier Null) mit abnehmender Wahrscheinlichkeit für größere Fehler auftreten [Mol22, S. 37].

In einem linearen Modell stellt der Achsenabschnitt die Basislinie dar, an der die Auswirkungen aller anderen Merkmale gemessen werden. Dieser Wert gibt an, was das Modell für die Zielvariable vorhersagen würde, wenn alle anderen Merkmale nicht vorhanden wären – der Ausgangspunkt der Vorhersage für einen Datensatz, in dem alle anderen Variablen auf null gesetzt sind. Es ist wichtig zu erwähnen, dass der Achsenabschnitt für sich genommen nicht immer eine praktische Bedeutung hat, da es selten vorkommt, dass alle Variablen tatsächlich den Wert null annehmen. Die wahre Aussagekraft des Achsenabschnitts tritt zutage, wenn die Daten so standardisiert wurden, dass ihre Mittelwerte bei null und die Standardabweichung bei eins liegen. Unter diesen Umständen repräsentiert der Achsenabschnitt die erwartete Zielvariable für einen hypothetischen Fall, in dem alle Merkmale ihren Durchschnittswert aufweisen.

Bei der Betrachtung einzelner Merkmale innerhalb des Modells sagt das Gewicht  $\beta_j$  eines Merkmals, um wie viel sich die Zielvariable  $y^{(i)}$  ändert, wenn das Merkmal  $x_j^{(i)}$  um eine Einheit erhöht wird – und zwar unter der Annahme, dass alle anderen Merkmale unverändert bleiben. Dies ermöglicht es, den isolierten

Effekt eines jeden Merkmals auf die Vorhersage zu verstehen [Mol22, S. 39].

Die optimalen Gewichte, oder Koeffizienten, eines linearen Regressionsmodells werden üblicherweise durch ein Verfahren bestimmt, das als Methode der kleinsten Quadrate (engl. *Ordinary Least Squares*, OLS) bekannt ist. Diese Methode sucht die Koeffizienten  $\beta_0, \dots, \beta_p$ , welche die Summe der quadrierten Differenzen zwischen den beobachteten Werten der Zielvariablen  $y^{(i)}$  und den von dem Modell vorhergesagten Werten minimieren:

$$\hat{\beta} = \arg \min_{\beta_0, \dots, \beta_p} \sum_{i=1}^n \left( y^{(i)} - \left( \beta_0 + \sum_{j=1}^p \beta_j x_j^{(i)} \right) \right)^2. \quad (6.2)$$

Das Ergebnis der Minimierung,  $\hat{\beta}$  stellt den Vektor der geschätzten Koeffizienten dar [Mol22, S. 37]. In der vorliegenden Arbeit wird das Python-Paket `scikit-learn`<sup>1</sup> verwendet, um die lineare Regression durchzuführen und die Koeffizienten  $\hat{\beta}$  zu bestimmen.

## 6.2. Einführung in das shap Python-Paket

Das Python-Paket `shap`<sup>2</sup> ist eine Open-Source-Bibliothek, die es Nutzern ermöglicht, die Auswirkungen von Merkmalen auf Vorhersagen von maschinellen Lernmodellen zu interpretieren und zu visualisieren. Entwickelt wurde die Bibliothek ursprünglich von Scott Lundberg und weiteren Mitwirkenden im Rahmen der Forschungsarbeit an der University of Washington [LL17]. Das Paket basiert auf dem Konzept der Shapley-Werte aus der kooperativen Spieltheorie und überträgt diese auf den Kontext des maschinellen Lernens, um als Tool für die Interpretierbarkeit und Erklärbarkeit von Modellvorhersagen zu dienen.

Die Kernfunktion des `shap`-Pakets ist die Berechnung von SHAP-Werten, welche die Auswirkung der Einzelmerkmale auf die Modellvorhersage quantifizieren. Jeder SHAP-Wert ist ein Maß dafür, wie viel jedes Merkmal zur Vorhersage beigetragen hat, im Vergleich zu einer durchschnittlichen Vorhersage über den gesamten Datensatz. Diese Werte sind besonders wertvoll, weil sie ein Maß für die Bedeutung jedes Merkmals liefern, das sowohl lokal (für einzelne Vorhersagen) als auch global (über das gesamte Modell) interpretiert werden kann.

---

<sup>1</sup><https://scikit-learn.org>

<sup>2</sup><https://shap.readthedocs.io>

Mit **shap** können Benutzer die Vorhersagen einer Vielzahl von Modellen interpretieren, von linearen Modellen bis hin zu komplexen Konstrukten wie tiefe neuronale Netzwerke. Die Bibliothek bietet eine vielseitige Auswahl an Visualisierungsoptionen, darunter Beeswarm-Plots, Dependence-Plots und Summary-Plots, die es ermöglichen, die SHAP-Werte intuitiv zu verstehen. Diese Visualisierungen erleichtern es, Muster und Beiträge einzelner Merkmale zu erkennen, was nicht nur wertvolle Einblicke in die Leistung des Modells bietet, sondern auch zu faireren und transparenteren Modellentscheidungen führen kann.

TODO: Erklärung EXPLAINER, ESTIMASTOR, Hinweis auf exakte Verechnung für weniger als 10 Features

### 6.3. Einführung in den Datensatz

Der für diese Studie verwendete Datensatz umfasst Informationen aus dem Mietspiegel München des Jahres 2003. Dieser Mietspiegel dient dazu, Mietentscheidungen in der Stadt München zu unterstützen und enthält 2053 Beobachtungen von Wohnungen. Das Ziel dieser Analyse besteht darin, die Nettomiete vorherzusagen, basierend auf den verfügbaren Merkmalen.

Die Variablen im Datensatz umfassen:

- **$y$ : nm (Nettomiete in EUR)**: Die monatliche Nettomiete in Euro, die Mieter für die Wohnung zahlen.
- **$x_1$ : nmqm (Nettomiete pro m<sup>2</sup> in EUR)**: Die monatliche Nettomiete pro Quadratmeter Wohnfläche in Euro.
- **$x_2$ : wfl (Wohnfläche in m<sup>2</sup>)**: Die Gesamtfläche der Wohnung in Quadratmetern.
- **$x_3$ : rooms (Anzahl der Zimmer)**: Die Anzahl der Zimmer in der Wohnung, die die Größe und den Raumbedarf widerspiegelt.
- **$x_4$ : bj (Baujahr der Wohnung)**: Das Baujahr der Wohnung, das auf mögliche Modernisierungen und den Zustand hinweisen kann.
- **$x_5$ : bez (Stadtbezirk)**: Der Stadtbezirk in München, in dem sich die Wohnung befindet.
- **$x_6$ : wohngut (Gute Wohnlage)**: Eine binäre Variable, die angibt, ob die Wohnung in einer guten Wohnlage liegt ( $Ja=1$ ,  $Nein=0$ ).

- $x_7$ : **wohnbest (Beste Wohnlage)**: Eine binäre Variable, die angibt, ob die Wohnung in der besten Wohnlage liegt ( $Ja=1$ ,  $Nein=0$ ).
- $x_8$ : **ww0 (Warmwasserversorgung)**: Eine binäre Variable, die angibt, ob die Warmwasserversorgung vorhanden ist ( $Ja=0$ ,  $Nein=1$ ).
- $x_9$ : **zh0 (Zentralheizung)**: Eine binäre Variable, die angibt, ob eine Zentralheizung vorhanden ist ( $Ja=0$ ,  $Nein=1$ ).
- $x_{10}$ : **badkach0 (Gekacheltes Badezimmer)**: Eine binäre Variable, die angibt, ob das Badezimmer gekachelt ist ( $Ja=0$ ,  $Nein=1$ ).
- $x_{11}$ : **badextra (Besondere Zusatzausstattung im Bad)**: Eine binäre Variable, die angibt, ob besondere Zusatzausstattungen im Badezimmer vorhanden sind ( $Ja=1$ ,  $Nein=0$ ).
- $x_{12}$ : **kueche (Gehobene Küche)**: Eine binäre Variable, die angibt, ob eine gehobene Küchenausstattung vorhanden ist ( $Ja=1$ ,  $Nein=0$ ).

Der Datensatz wurde in Python mithilfe der Bibliothek **pandas** als DataFrame eingelesen. Tabelle 7 zeigt die ersten drei Beobachtungen des Datensatzes:

Tabelle 7.: Auszug aus dem Mietspiegel München (Jahr 2003) mit Wohnungsinformationen.

$y$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	$x_{11}$	$x_{12}$
741.39	10.90	68	2	1918.0	2	1	0	0	0	0	0	0
715.82	11.01	65	2	1995.0	2	1	0	0	0	0	0	0
528.25	8.38	63	3	1918.0	2	1	0	0	0	0	0	0

Quelle: Eigene Darstellung auf Basis der Datengrundlage [Ope10].

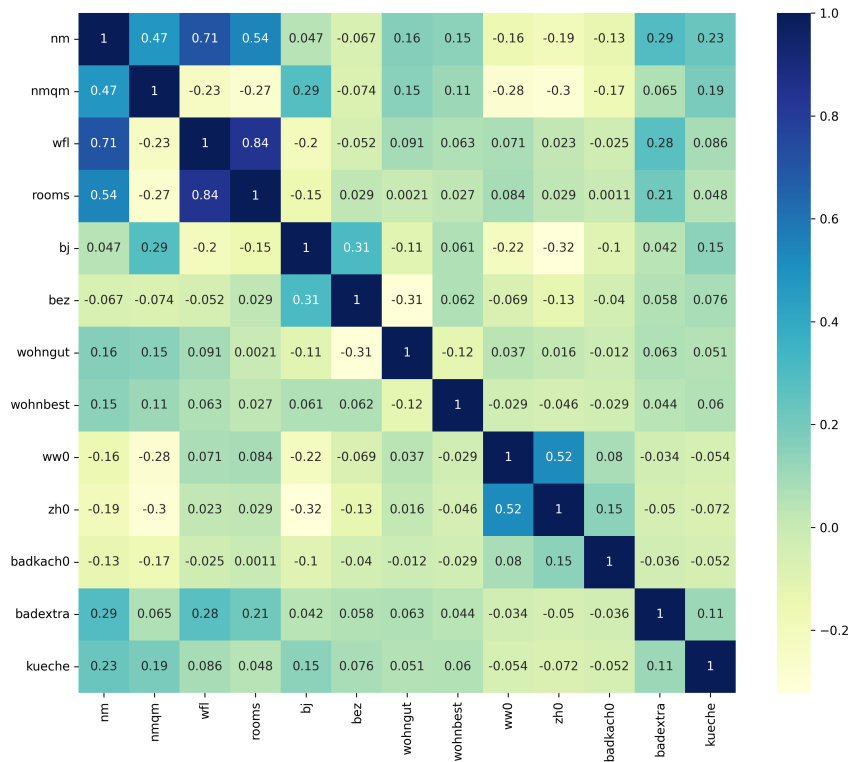
Der vollständige Quellcode für das Einlesen der Daten sowie alle weiteren Analyseschritte ist im Anhang A.3 dieser Arbeit zu finden.

## 6.4. Explorative Datenanalyse & Datenaufbereitung

## 6.5. Modellierung der linearen Regression

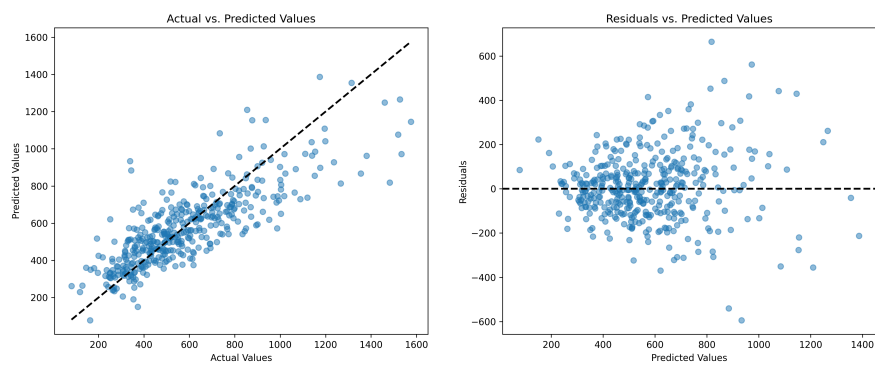
TODO: Modell fitten

Abbildung 3.: Korrelationsmatrix der Merkmale im Datensatz.



Quelle: Eigene Darstellung, A.3.

Abbildung 4.: Residuenanalyse: Beziehung zwischen Vorhersagen und Abweichungen.



Quelle: Eigene Darstellung, A.3.

## 7. Ergebnisse

### 7.1. Berechnung von SHAP-Werten

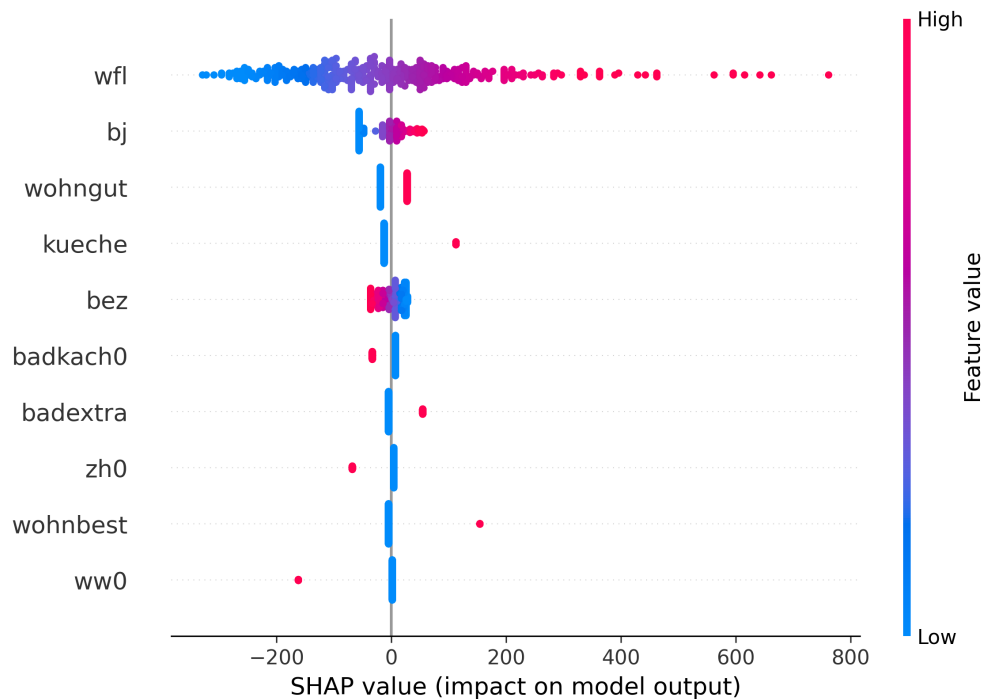
TODO: Berechnung der SHAP-Werte.

### 7.2. Interpretation

TODO: Analyse der Ergebnisse, Interpretation von SHAP-Werten, Vergleich der Koeffizienten mit den **SHAP-Werten!**

#### 7.2.1. Lokale Interpretation

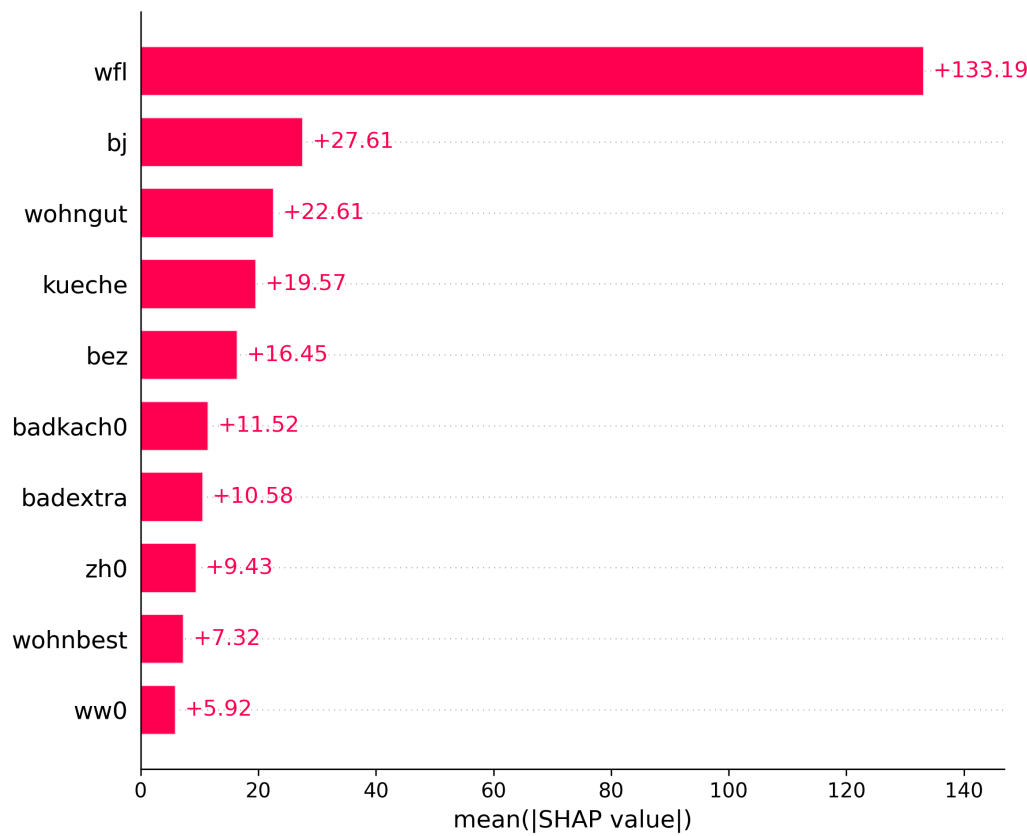
Abbildung 5.: SHAP Summary Plot



Quelle: Eigene Darstellung, A.3.

#### 7.2.2. Globale Interpretation

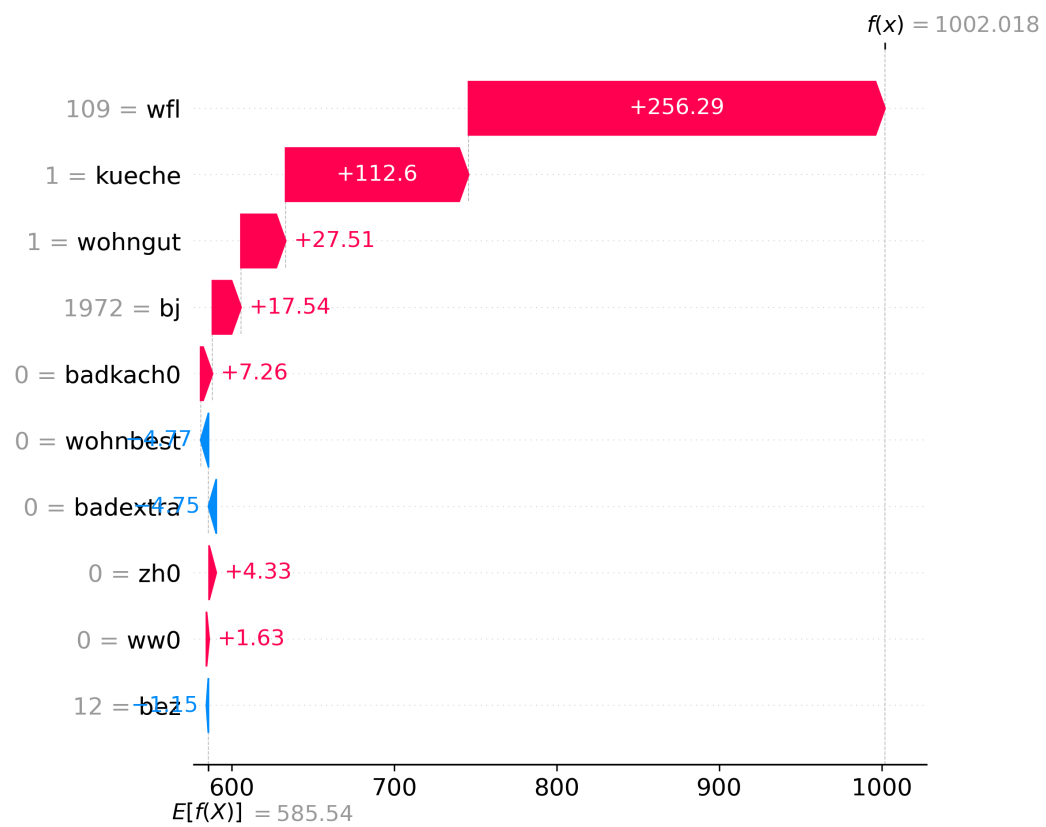
Abbildung 6.: SHAP Bar Plot



Quelle: Eigene Darstellung, A.3.



Abbildung 7.: SHAP Waterfall Plot



Quelle: Eigene Darstellung, A.3.



## 8. Fazit & Ausblick



## Literaturverzeichnis

- [LL17] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [Mol22] Christoph Molnar. *Interpretable machine learning: A guide for making Black Box models explainable*. Chistoph Molnar c/o Mucbook Clubhouse, Heidi Seibold, 2 edition, 2022.
- [Mol23] Christoph Molnar. *Interpreting machine learning models with SAP A guide with python examples and theory on Shapley Values*. Christoph Molnar c/o MUCBOOK, 1 edition, 2023.
- [Ope10] Münchner Mietspiegel 2003, Oct 2010.
- [RC04] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, 2004.
- [RWB<sup>+</sup>22] Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Olivér Kiss, Sebastian Nilsson, and Rik Sarkar. The shapley value in machine learning. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5572–5579. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Survey Track.
- [Sha53] L. S. Shapley. *17. A Value for n-Person Games*, pages 307–318. Princeton University Press, Princeton, 1953.



## Abbildungsverzeichnis

Abbildung 1: Beitrag der Merkmale $x_{j \in \{1,2,3\}}$ zur Modellvorhersage $f(x^{(1)})$ .	19
Abbildung 2: Beitrag der Merkmale $x_{j \in \{1,2,3\}}$ zur Modellvorhersage $f(x^{(2)})$ .	19
Abbildung 3: Korrelationsmatrix der Merkmale im Datensatz. . . . .	32
Abbildung 4: Residuenanalyse: Beziehung zwischen Vorhersagen und Abweichungen. . . . .	32
Abbildung 5: SHAP Summary Plot . . . . .	33
Abbildung 6: SHAP Bar Plot . . . . .	34
Abbildung 7: SHAP Waterfall Plot . . . . .	35





## Tabellenverzeichnis

Tabelle 1: Potenzielle Gewinne für verschiedene Teilnehmerkonstellationen im Designwettbewerb. . . . .	6
Tabelle 2: Marginalbeiträge der einzelnen Teilnehmer zu den möglichen Koalitionen. . . . .	6
Tabelle 3: Terminologie der originären Shapley-Werte im Kontext des maschinellen Lernens. . . . .	13
Tabelle 4: Merkmale von Beobachtungen in einem Immobilien-Datensatz. . . . .	15
Tabelle 5: Marginalbeiträge der einzelnen Merkmale zu den möglichen Koalitionen für die Beobachtung $x^{(1)}$ . . . . .	18
Tabelle 6: Marginalbeiträge der einzelnen Merkmale zu den möglichen Koalitionen für die Beobachtung $x^{(2)}$ . . . . .	18
Tabelle 7: Auszug aus dem Mietspiegel München (Jahr 2003) mit Wohnungsinformationen. . . . .	31



## Quellcodeverzeichnis

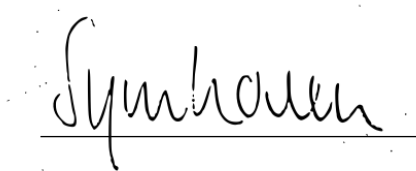


## Eidesstattliche Erklärung

Ich versichere an Eides statt, dass ich die vorstehende Arbeit selbständig und ohne fremde Hilfe angefertigt und mich anderer als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen übernommen wurden, sind als solche kenntlich gemacht. Alle Internetquellen sind der Arbeit beigefügt.

Des Weiteren versichere ich, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und dass die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

München, 17. November 2023

A handwritten signature in black ink, reading 'Symhoven', written over a horizontal line.

SIMON SYMHOVEN



## A. Quellcode

### A.1. requirements.txt

```
1 pandas
2 matplotlib
3 scikit-learn
4 numpy
5 shap
6 seaborn
7 ipywidgets
8 nbconvert
9 jupyter_contrib_nbextensions
10 statsmodels
11 notebook==6.4.12
12 traitlets==5.9.0
```

### A.2. charts.py

```
1 import matplotlib.pyplot as plt
2 from typing import List
3
4 plt.rcParams.update({
5     'text.usetex': True,
6     'text.latex.preamble': r'\usepackage{amsfonts}'
7 })
8
9 def plot_waterfall(expected_value_model: float, observation_values: List[float],
10                   observation_labels: List[str], contributions: List[float],
11                   name: str, index: int):
12     """
13     Creates a waterfall chart plot that visualizes the expected model value,
14     observation values, their contributions, and labels.
15
16     Parameters:
17     - expected_value_model (float): The expected model value.
18     - observation_values (List[float]): A list of observation values.
19     - observation_labels (List[str]): A list of labels associated with
20       the observation values.
21     - contributions (List[float]): A list of contributions representing
22       changes from observation values to the
23       model value.
24     - name (str): The name of the plot and the file for saving.
25     - index (int): The index for the observation.
26     """
27
28     labels = [f'{value} = {label}' for (value, label)
29               in zip(observation_values, observation_labels)]
30     waterfall = [expected_value_model]
31     for contribution in contributions:
```

```

32     waterfall.append(waterfall[-1] + contribution)
33
34     _, ax = plt.subplots(figsize=(10, 3))
35
36     for idx, (contribution, start) in enumerate(
37         zip(contributions, waterfall[:-1])
38     ):
39         ax.barh(idx, contribution, left=start, color='blue'
40                 if contribution >= 0 else 'red', height=0.5, alpha=0.8)
41         text_x = start + contribution / 2
42         ax.text(text_x, idx, contribution, va='center', ha='center', fontsize=12,
43                color='white')
44
45     l1 = r'$\mathbb{E}\{f(X)\} = $' + f' {expected_value_model} Euro'
46     l2 = r'$f(x^{\{{{index}\}}}) = $ {waterfall_value} Euro'.format(index=index,
47                                waterfall_value=waterfall[-1])
48     ax.text(expected_value_model, -1.2, l1, va='center', ha='center')
49     ax.text(waterfall[-1], 2.8, l2, va='center', ha='center')
50
51     ax.set_yticks(range(len(labels)))
52     ax.set_yticklabels(labels)
53     ax.set_xlabel('Euro')
54
55     ax.axvline(x=expected_value_model, color='grey', linestyle='--', linewidth=1)
56     ax.axvline(x=waterfall[-1], color='grey', linestyle='--', linewidth=1)
57
58     ax.set_xlim(min(waterfall)-5, max(waterfall)+5)
59
60     for spine in ['top', 'right', 'left']:
61         ax.spines[spine].set_visible(False)
62
63     plt.tight_layout()
64     plt.tick_params(left = False)
65     plt.savefig(f'images/{name}', dpi=300)
66
67 """
68 Create charts for observation  $x^{\{1\}}$  and  $x^{\{2\}}$ 
69 """
70 # General Model Values
71 observation_labels = ['Groesse', 'Anzahl Zimmer', 'Entfernung zum Zentrum']
72 expected_value_model = 680
73
74 # Observation values and contributions for  $x^{\{1\}}$ 
75 index = 1
76 observation_values_x1 = [100, 3, 5]
77 contributions_x1 = [-125, -10, 5]
78
79 plot_waterfall(expected_value_model=expected_value_model,
80                observation_values=observation_values_x1,
81                observation_labels=observation_labels,
82                contributions=contributions_x1,
83                name="model-output-x1.png", index=index)
84
85 # Observation values and contributions for  $x^{\{2\}}$ 
86 index = 2
87 observation_values_x2 = [150, 4, 10]
88 contributions_x2 = [125, 10, -5]
89
90 plot_waterfall(expected_value_model=expected_value_model,

```



```
89         observation_values=observation_values_x2,  
90         observation_labels=observation_labels,  
91         contributions=contributions_x2,  
92         name="model-output-x2.png", index=index)
```

### A.3. linreg.py

```
1  import pandas as pd  
2  from sklearn.model_selection import train_test_split  
3  import matplotlib  
4  import matplotlib.pyplot as plt  
5  import seaborn as sns  
6  import shap  
7  from sklearn.linear_model import LinearRegression  
8  from sklearn.metrics import mean_absolute_error, mean_squared_error  
9  import numpy as np  
10  
11  matplotlib.use('Agg')  
12  
13  def load_data() -> pd.DataFrame:  
14      """  
15      Loads and returns the dataset from the given URL as a Pandas DataFrame.  
16  
17      Returns:  
18          pd.DataFrame: The loaded dataset.  
19      """  
20      url = "https://data.ub.uni-muenchen.de/2/1/miete03.asc"  
21      df = pd.read_csv(url, sep='\t')  
22      return df  
23  
24  def model(X: pd.DataFrame, y: pd.Series) -> (LinearRegression, pd.DataFrame):  
25      """  
26      Fits a Linear Regression model to the given data.  
27  
28      Args:  
29          X (pd.DataFrame): The feature matrix.  
30          y (pd.Series): The target variable.  
31  
32      Returns:  
33          LinearRegression: The fitted Linear Regression model.  
34          DataFrame: The coefficients as DataFrame.  
35      """  
36      linreg = LinearRegression()  
37      linreg.fit(X, y)  
38  
39      cdf = pd.DataFrame(linreg.coef_.round(5), X.columns, columns=['Coefficients'])  
40      cdf.loc['Intercept'] = linreg.intercept_.round(5)  
41  
42      return linreg, cdf  
43  
44  def plot_residuals(y_test: pd.Series, y_pred: pd.Series) -> None:  
45      """  
46      Creates a residual plots.  
47  
48      Args:  
49          y_test (pd.Series): The actual target values.  
50          y_pred (pd.Series): The predicted target values.  
51      """  
52      residuals = y_test - y_pred
```



```
110
111 linreg, params = model(X=X_train, y=y_train)
112 y_pred = linreg.predict(X_test)
113 explainer = shap.LinearExplainer(linreg, X_train)
114 shap_values = explainer(X_test)
115
116 mae = mean_absolute_error(y_test, y_pred)
117 mse = mean_squared_error(y_test, y_pred)
118 train_score = linreg.score(X_train, y_train)
119 test_score = linreg.score(X_test, y_test)
120
121 print(mae)
122 print(mse)
123 print(train_score)
124 print(test_score)
125
126 print(params)
127
128 plot_corr(X=df)
129 plot_residuals(y_test=y_test, y_pred=y_pred)
130 plot_shap(shap_values=shap_values, idx=148)
```