



FOM Hochschule für Oekonomie & Management

Hochschulzentrum Düsseldorf

Master-Thesis

im Studiengang IT Management

zur Erlangung des Grades eines

Master of Science (M.Sc.)

über das Thema

**Untersuchung des Einsatzes von Explainable Artificial Intelligence zur
Unterstützung von Machine-Learning-Prognosen in der Immobilienbranche**

von

Emir Begovac

Erstgutachter	Prof. Dr. Adem Alparslan
Matrikelnummer	404883
Abgabedatum	2023-03-04

Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Tabellenverzeichnis.....	VIII
Formelverzeichnis	IX
Abkürzungsverzeichnis.....	X
Symbolverzeichnis.....	XI
1 Einleitung	1
1.1 Problemstellung	1
1.2 Zielstellung und Forschungsfragen.....	2
1.3 Methodisches Vorgehen	3
1.4 Aufbau der Arbeit	6
2 Theoretische Grundlage.....	8
2.1 Künstliche Intelligenz.....	8
2.2 Machine Learning	9
2.2.1 Überwachtes Lernen	10
2.2.2 Regressionsmethoden	11
2.2.3 Künstliche neuronale Netze	11
2.2.4 K-nearest Neighbor.....	14
2.2.5 Extreme Gradient Boosting und Extrem Boosting Machine	15
2.3 Erklärende künstliche Intelligenz	18
2.3.1 Shapley Additive Explanations	21
2.3.2 Local Interpretable Modell-Agnostic Explanations	22
2.3.3 Interpreter ML	23
2.4 Leistungsindikatoren.....	24
2.5 Cross industry standard Process for Data Mining	26
2.5.1 Business Understanding	27
2.5.2 Data Understanding	27
2.5.3 Data Preparation	28
2.5.4 Modeling.....	28
2.5.5 Evaluation.....	29

2.5.6 Deployment	29
2.6 Definition von Analysefeldern und Erklärungseigenschaften zur Unterstützung der Untersuchung	29
3 Einordnung der aktuellen wissenschaftlichen Erkenntnisse zum Immobilienmarkt	33
3.1 Preisentwicklung der Immobilien.....	33
3.2 State of the Art der Immobilienprognosen	40
4 Entwicklung eines Prognosemodells	44
4.1 Data Understanding	44
4.2 Data Preparation	50
4.3 Modeling.....	54
4.4 Evaluation	56
5 Untersuchung des Einsatzes von XAI-Methoden	59
5.1 Untersuchung der XAI-Methoden im Rahmen von EBM	59
5.2 Untersuchung der XAI-Methoden im Rahmen von KNN	72
5.3 Untersuchung der XAI-Methoden im Rahmen von XGBoost.....	85
5.4 Untersuchung der XAI-Methoden im Rahmen von ANN	100
5.5 Vergleich der XAI-Methoden im Kontext der angewendeten Algorithmen ...	108
6 Diskussion der Forschungsergebnisse	129
6.1 Beurteilung der Ergebnisse.....	129
6.2 Implikationen für die Praxis	146
6.3 Reflexion.....	148
6.4 Limitierung	149
7 Schlussfolgerung	152
7.1 Zusammenfassung	152
7.2 Ausblick und Forschungsbedarf	153
7.3 Fazit	154
Literaturverzeichnis.....	VIII
Anhang	XXVI

Abbildungsverzeichnis

Abbildung 1: Vorgehen Literaturrecherche vom Brocke (2009)	4
Abbildung 2: Vorgehen Prototyping (Religia et al., 2020, S.62).....	6
Abbildung 3: KI mit den Teilbereichen (Mockenhaupt,2021, S.130)	9
Abbildung 4: Mathematisches Modell eines künstlichen Neurons (Müller & Lenz, 2013, S.81-83).....	12
Abbildung 5: Performance gegen Interpretierbarkeit (Barredo Arrieta et al., 2020, S.31, Nori et al., 2019, S. 2-3).....	20
Abbildung 6: Phasen des CRISP-DM Referenzmodells (Chapman et al., 2000, S.12)..	27
Abbildung 7: Durchschnittlicher Verkaufspreis von Häusern in den USA (FRED, 2022d, S.1)	34
Abbildung 8: Durchschnittlicher Verkaufspreis der Region Mitte-West in den USA (FRED, 2022c ,S.1).....	35
Abbildung 9: Durchschnittlicher Verkaufspreisentwicklung auf Landes- Bundesstaats- und Stadtbene (FRED, 2022a, 2022b, 2022c).....	36
Abbildung 10: Durschnittliches Einkommen in den USA in \$ von 1990 bis 2021 (US Census Bureau, 2022, S.1)	38
Abbildung 11: Zusammenfassung der Ergebnisse des Kapitels 3.1	40
Abbildung 12: Zusammenfassung der Ergebnisse des Kapitels 3.2	43
Abbildung 13: Fehlende Werte Datensatz	45
Abbildung 14: Verteilung Zielvariable	45
Abbildung 15: Ausschnitt der Verteilung der unabhängigen Variablen mittels Balkendiagramme	46
Abbildung 16: Ausschnitt der Verteilung Sale Type und Sale Condition	47
Abbildung 17: Graphische Korrelationsanalyse der metrischen Variablen.....	48
Abbildung 18: Graphische Korrelationsanalyse mittels Boxplots	49
Abbildung 19: Höhe des Verkaufspreises im Bezug zur Bauart und Gesamtqualität	49
Abbildung 20: Verlustfunktion des ANN-Modells	55
Abbildung 21: SHAP Global Balken-Diagramm EBM-Modell	60
Abbildung 22: SHAP Global Beeswarm-Diagramm EBM-Modell.....	61
Abbildung 23: SHAP Global Beeswarm-Diagramm Variable Paved_Drive	62
Abbildung 24: SHAP Lokal Wasserfall-Diagramm EBM-Modell Beobachtung 1.....	63

Abbildung 25: SHAP Lokal Wasserfall-Diagramm EBM-Modell Beobachtung 2	64
Abbildung 26: LIME unbehandelte kategoriale Variablen lokal EBM-Modell	65
Abbildung 27: LIME Global Balken-Diagramm EBM-Modell.....	66
Abbildung 28: LIME Lokal Wasserfall-Diagramm EBM-Modell Beobachtung 1	67
Abbildung 29: InterpretML Global Balken-Diagramm EBM-Modell.....	69
Abbildung 30: InterpretML Global PDP-Analyse TotalSF und Gr Liv Area.....	70
Abbildung 31: InterpretML Global PDP-Analyse Bath und Overall Qual.....	70
Abbildung 32: InterpretML Lokal Wasserfall-Diagramm EBM-Modell Beobachtung 1+2	72
Abbildung 33: SHAP Global Balken-Diagramm KNN-Modell	73
Abbildung 34: SHAP Global Beeswarm-Diagramm KNN-Modell.....	74
Abbildung 35: SHAP Lokal Wasserfall-Diagramm KNN-Modell Beobachtung 1	75
Abbildung 36: SHAP Lokal Wasserfall-Diagramm KNN-Modell Beobachtung 2	76
Abbildung 37: LIME Global nicht standardisierte Daten KNN-Modell Beobachtung 177	
Abbildung 38: Überprüfung nicht standardisierter Daten mittels SHAP und InterpretML	77
Abbildung 39: LIME Global Balken-Diagramm KNN-Modell.....	78
Abbildung 40: LIME Lokal Wasserfall-Diagramm KNN-Modell Beobachtung 1+ 2 ...	80
Abbildung 41: InterpretML Global Balken-Diagramm KNN-Modell.....	81
Abbildung 42: InterpretML Global PDP-Analyse PoolQC und Overall Qual	82
Abbildung 43: InterpretML Global PDP-Analyse TotalSF und Total Bsmt SF	82
Abbildung 44: InterpretML Lokal Wasserfall-Diagramm KNN-Modell Beobachtung 1+2	84
Abbildung 45: InterpretML LIME Lokal Wasserfall-Diagramm KNN-Modell Beobachtung 1+2	85
Abbildung 46: SHAP Global Balken-Diagramm XGB-Modell	87
Abbildung 47: SHAP Global Beeswarm-Diagramm XGB-Modell.....	88
Abbildung 48: Histogramm Verkaufspreises im Bezug zur Bauart und Gesamtqualität	89
Abbildung 49: SHAP Lokal Wasserfall-Diagramm XGB-Modell Beobachtung 1	91
Abbildung 50: SHAP Lokal Force-Plot XGB-Modell Beobachtung 1	91
Abbildung 51: SHAP Lokal Wasserfall-Diagramm XGB-Modell Beobachtung 2	92
Abbildung 52: LIME Global Balken-Diagramm XGB-Modell.....	93

Abbildung 53: LIME Lokal Wasserfall-Diagramm KNN-Modell Beobachtung 1+2	95
Abbildung 54: InterpretML Global Balken-Diagramm XGB-Modell.....	96
Abbildung 55: InterpretML Global PDP-Analyse TotalSF	97
Abbildung 56: InterpretML Global PDP-Analyse Overall Qual	97
Abbildung 57: InterpretML Global PDP-Analyse 1st Flr SF	98
Abbildung 58: InterpretML LIME Lokal Wasserfall-Diagramm XGB-Modell Beobachtung 1+2	99
Abbildung 59: InterpretML Lokal Wasserfall-Diagramm XGB-Modell Beobachtung 1+2	100
Abbildung 60: SHAP Global Balken-Diagramm ANN-Modell	102
Abbildung 61: SHAP Global Beeswarm-Diagramm ANN-Modell.....	103
Abbildung 62: SHAP Lokal Wasserfall-Diagramm ANN-Modell Beobachtung 1	104
Abbildung 63: SHAP Lokal Force-Plot ANN-Modell Beobachtung 1	105
Abbildung 64: SHAP Lokal Wasserfall-Diagramm ANN-Modell Beobachtung 2.....	106
Abbildung 65: LIME Global Balken-Diagramm ANN-Modell.....	107
Abbildung 66: LIME Lokal Wasserfall-Diagramm ANN-Modell Beobachtung 1+2 ..	108
Abbildung 67: InterpretML Dashboard für XGB- und EBM-Modell	119
Abbildung 68: Überprüfung der Stabilität der Erklärungen des XGB und KNN-Modells	121
Abbildung 69: Überprüfung der Konsistenz der Erklärungen des XGB-Modells	122
Abbildung 70: Überprüfung der Erklärungsähnlichkeit der Erklärungen des XGB- Modells.....	124
Abbildung 71: Überprüfung der Erklärungsähnlichkeit der Erklärungen des XGB- und KNN-Modells.....	125
Abbildung 72: Zusammenfassung der Beurteilung - Wichtigkeit von Variablen.....	131
Abbildung 73: Zusammenfassung der Beurteilung – Optimierung und Debugging	133
Abbildung 74: Zusammenfassung der Beurteilung – Fairness, ethische Korrektheit, Risikokontrolle	135
Abbildung 75: Zusammenfassung der Beurteilung - Transparenz	138
Abbildung 76: Zusammenfassung der Beurteilung – Stabilität, Transparenz, Trennbarkeit	141
Abbildung 77: Zusammenfassung der Beurteilung - Erklärungsähnlichkeit.....	144

Abbildung 78: Zusammenfassung der Beurteilung – Weitere Erkenntnisse 146

Tabellenverzeichnis

Tabelle 1: Verwendete Datenbanken	4
Tabelle 2: Verwendete Suchparameter	5
Tabelle 3: Leistungsmetriken zur Bewertung von Machine Learning Algorithmen	26
Tabelle 4: Analysefelder der Erklärbarkeit nach XAI	31
Tabelle 5: Erklärungseigenschaften der XAI-Methoden	32
Tabelle 6: Ergebnisse der Korrelationsanalyse nach Pearson und Spearman	51
Tabelle 7: Ergebnisse der Korrelationsanalyse nach ANOVA	52
Tabelle 8: Ergebnisse der Korrelationsanalyse nach der Integration	53
Tabelle 9: Hyperparameter der Modelle	55
Tabelle 10: Ergebnisse der Trainings- und Testdaten	57
Tabelle 11: Ergebnisse aller Modelle und XAI-Verfahren Global	113
Tabelle 12: Ergebnisse Lokal Beobachtung 1	116
Tabelle 13: Ergebnisse Lokal Beobachtung 2	117
Tabelle 14: Erklärungseigenschaften je Erklärungsmethode	120
Tabelle 15: Änderungswerte der Variablen	123
Tabelle 16: Erklärungsähnlichkeit Global	127
Tabelle 17: Erklärungsähnlichkeit Lokal	128

Formelverzeichnis

Formel 1: Minimierung der Verlustfunktion im ML (Andrychowicz et al., 2016, S.3988)	10
.....
Formel 2: Mathematische Funktionl eines künstlichen Neurons (Agatonovic-Kustrin & Beresford, 2000, S.719-720)	12
Formel 3: Regression anhand von KNN (Song et al., 2017, S.5)	14
Formel 4: Distanz Berechnung Nearest Neighbor (Hu et al., 2016, S.3).....	15
Formel 5: XBM Objekt Funktion (Budholiya et al., 2022, S.4516)	16
Formel 6: XBM Objekt Funktion Final (Budholiya et al., 2022, S.4517)	17
Formel 7: XBM Algorithmus Modellergebnis (Zhang et al., 2022, S.3-4)	17
Formel 8: EBM-Algorithmus (Nori et al., 2019, S. 2-3).....	17
Formel 9: Interaktionseffekt EBM-Algorithmus (Nori et al., 2019, S. 2-3)	18
Formel 10: Shapley Wert Formel (Ghorbani & Zou, 2019, S.4053-4054).....	21
Formel 11: Shapley Wert Ermittlung anhand eines Modells (Nowak & Radzik, 1994. S.43-44).....	21
Formel 12: LIME-Algorithmus (Ribeiro et al., 2016,S.97-99).....	22

Abkürzungsverzeichnis

ANN	künstliches neuronales Netz
CSV	Comma-separated Values
DM	Data Mining
DSGVO	Datenschutz-Grundverordnung
EBM	Extreme Boosting Maschine
EU	europäischen Union
KI	künstliche Intelligenz
KNN	K-nächster Nachbar
LIME	Local Interpretable Modell-Agnostic Explanations
MAE	Mean Absolute Error
MAPE	Mean absolute percentage Error
ML	Machine Learning
NA	nicht verfügbar
PDP	Partial Dependence Plot
RMSE	Root Mean Squared Error
SHAP	Shapley additive explanations
SLR	Systematische Literaturrecherche
XAI	erklärende künstliche Intelligenz
XGB	Extremes Gradienten Boosting

Symbolverzeichnis

\$	Dollar
%	Prozent
⊗	erfüllt
o	teilweise erfüllt
x	nicht erfüllt

1 Einleitung

1.1 Problemstellung

Im Zeitalter der Digitalisierung erfährt die Nutzung von künstlicher Intelligenz (KI) innerhalb von Unternehmen einen zunehmend hohen Stellenwert. Laut einer Studie von Deloitte von 2017 bis 2022 geben 94 % der weltweit befragten Unternehmen an, dass sie den Einsatz von KI als wesentlich für den Erfolg einstufen und 76 % investieren bereits signifikant in KI. Dies entspricht einem Anstieg von 17 % der Investitionen im Vergleich zu den Befragungsergebnissen des Jahres 2021 (Deloitte, 2022, S. 5-6). Dies bestätigt die gemeinsam durchgeführte Studie von Ernst&Young und dem zentralen Immobilien Ausschuss ebenfalls für die Immobilienbranche in Deutschland. Die befragten Immobilienunternehmen geben an, dass die Digitalisierungsbemühungen fortwährend steigen. Bereits 35 % der befragten Unternehmen investieren mehr als 5 % ihres Jahresumsatzes in Digitalisierungsmaßnahmen, im Vergleich zu 18 % im Jahr 2021 (ZIA & EY Real Estate, 2022, S.10). Gerade in der Disziplin des Machine Learning (ML) lassen sich bedeutende Erfolge innerhalb der letzten Jahre für viele Bereiche beobachten. Hier spielt ML insbesondere in der Vorhersagemodellierung von Verkaufspreisen in der Immobilienbranche eine wichtige Rolle, um Unternehmen einen wertschöpfenden Geschäftsprozess und den damit verbundenen Wettbewerbsvorteil zu ermöglichen (Zhou, 2020, S.1-2). Die Implementierung von ML stellt jedoch viele Unternehmen vor eine Herausforderung. Viele Nutzer empfinden ML-Algorithmen als nicht verständliche Blackbox, bei denen Entscheidungen und Ergebnisse unklar und undurchsichtig sind. Das Blinde Vertrauen, in die nicht transparente Ergebnisfindung von ML-Modellen, kann die Anwendbarkeit und Legitimität des Einsatzes stark schmälern. Daher nimmt verstärkt die Nachfrage nach verständlichen ML-Modellen signifikant zu. Im Kontext der Immobilienbranche impliziert dies die Herausforderung, die nötigen Treiber des Vorhersagepreises zu identifizieren, das ML-Modell auf Basis einer transparenten Darstellung zu verstehen oder die Funktionsweise des Modells nachvollziehen zu können. Ein ausbleibender Einsatz von ML aufgrund von Unverständlichkeit, nicht Nachvollziehbarkeit und Intransparenz, beschränkt im Umkehrschluss den möglichen Nutzen, der durch ML-Prognosen erzielt werden kann (Adadi & Berrada, 2018, S.52138; Cajias, 2021, S.17). Gartner (2022) prognostiziert, dass 85 % der KI-Projekte in Unternehmen in ungenauen und intransparenten

Ergebnissen resultieren. Der Grund hierfür ist das nicht vorhandenen Wissen in Unternehmen hinsichtlich der Ergebnisfindung der KI. Die erklärende künstliche Intelligenz (XAI) kann hier Abhilfe schaffen, indem die Ergebnisfindung von ML-Modellen interpretierbarer, transparenter und verständlicher dargestellt werden. Der gezielte Einsatz von XAI soll gleichzeitig dabei helfen, das Vertrauen von Menschen zu Maschine zu bestärken. Dies erhöht nicht nur das Vertrauen in die KI, sondern ermöglicht auch einen transparenten und kontrollierten Einsatz in wertschöpfenden Geschäftsprozessen. Die rationale und überlegte Anwendung von XAI-Methoden soll Immobilienunternehmen ebenso die Möglichkeit verschaffen, neue Perspektiven und neues Wissen im Rahmen der Vorhersagemodellierung von Immobilienpreisen zu erschließen. Der damit eingehende Wettbewerbsvorteil ebnet den Unternehmen eine fortschrittliche Arbeitsweise und fördert simultan die Akzeptanz und das Vertrauen des Einsatzes von KI und ML innerhalb der Unternehmenssphäre (Gunning et al., 2019, S.2-3; Zolanvari et al., 2021, S.1-2).

1.2 Zielstellung und Forschungsfragen

Die vorliegende Ausarbeitung hat das Ziel zu untersuchen, inwieweit der Einsatz von erklärbarer künstlicher Intelligenz die Akzeptanz und das Vertrauen in ML-Prognosen optimiert und simultan das Verständnis zwischen Menschen und Maschinen stärken kann. Dies geschieht im Kontext der Immobilienbranche mittels der Vorhersagemodellierung von Immobilienpreisen. Hierfür wird als Datenbasis der *Housing prices Ames, Iowa* Datensatz verwendet, der die Immobilienpreise der Stadt Ames im Bundesstaat Iowa abbildet. Nachfolgend soll auf Basis der Ergebnisse eine kritische Begutachtung hinsichtlich des Einsatzes von XAI-Methoden innerhalb von ML-Prognosen durchgeführt werden. Zur Erreichung der Zielsetzung, werden vier Forschungsfragen definiert. Die Formulierung folgt dabei dem Vorgehen von Kitchenham (2009) und werden konsekutiv beantwortet. Die Ergebnisse werden in Kapitel 6 im Hinblick auf die Problemstellung und Zielsetzung gegenübergestellt und kritisch bewertet.

RQ1: „Welche Methoden der erklärenden künstlichen Intelligenz existieren?“ (RQ1).

RQ2: „Welcher Forschungsstand existiert innerhalb der Preisbestimmung und Preisprognosen von Immobilien?“ (RQ2).

RQ3: „Inwieweit können die genutzten XAI-Methoden die Modellergebnisse interpretierbar und transparent darstellen?“ (RQ3).

RQ4: „Bestehen Unterschiede in der Erklärbarkeit der Modellergebnisse die mittels XAI ausgegeben werden?“ (RQ4).

1.3 Methodisches Vorgehen

Als methodische Vorgehensweise zur konsekutiven Beantwortung der vorangegangenen Forschungsfragen und Zielerreichung der Ausarbeitung, werden die Literaturrecherche und das Prototyping ausgewählt. Über die systematische Literaturrecherche wird verfügbares Wissen konsolidiert, strukturiert und bewertet. Hierbei ist es von essenzieller Bedeutung, dass in der Arbeit ein systematisches Vorgehen eingehalten wird, um ein reproduzierbares Forschungsdesign für die Leserschaft zu erstellen. Dies beinhaltet auch die Beschreibung des Suchvorgangs (Kitchenham et al., 2009, S.793).

Dabei folgt die systematische Literaturrecherche dem Ansatz von vom Brocke (2009) und gliedert die sich in fünf Phasen, die der Abbildung 1 zu entnehmen sind. Durch die Dokumentation der Recherche und Selektion der Quellen wird eine spätere Rekonstruierung der Ergebnisse ermöglicht. Der Suchprozess umfasst eine sequenzielle Suche in den Bereichen Abstract, Titel und Thema. Für das Kapitel 3 wird der Suchprozess auf Journalartikel, Tagungsberichten und Reports imitiert. Diesbezüglich wird es als wegweisend betrachtet, lediglich Journalartikel und Tagungsberichten als relevant einzustufen, sobald diese einen höheren Hirsch-Index als 30 aufweisen oder über ein Q-Ranking im aktuellen Jahr von Q1 oder Q2 verfügen. Reports werden als signifikant eingestuft, sofern diese aus Institutionen mit staatlichem Bezug entspringen. Diese Limitationen sollen die Forschungsfragen anhand qualitativ hochwertiger Literatur beantworten und der Zielsetzung entsprechen. Die Backward- und Forward-Suche nach Webster & Watson (2002) komplettiert den Suchprozess, um bereits identifizierte wissenschaftliche Beiträge auf weitere Literatur zu untersuchen.

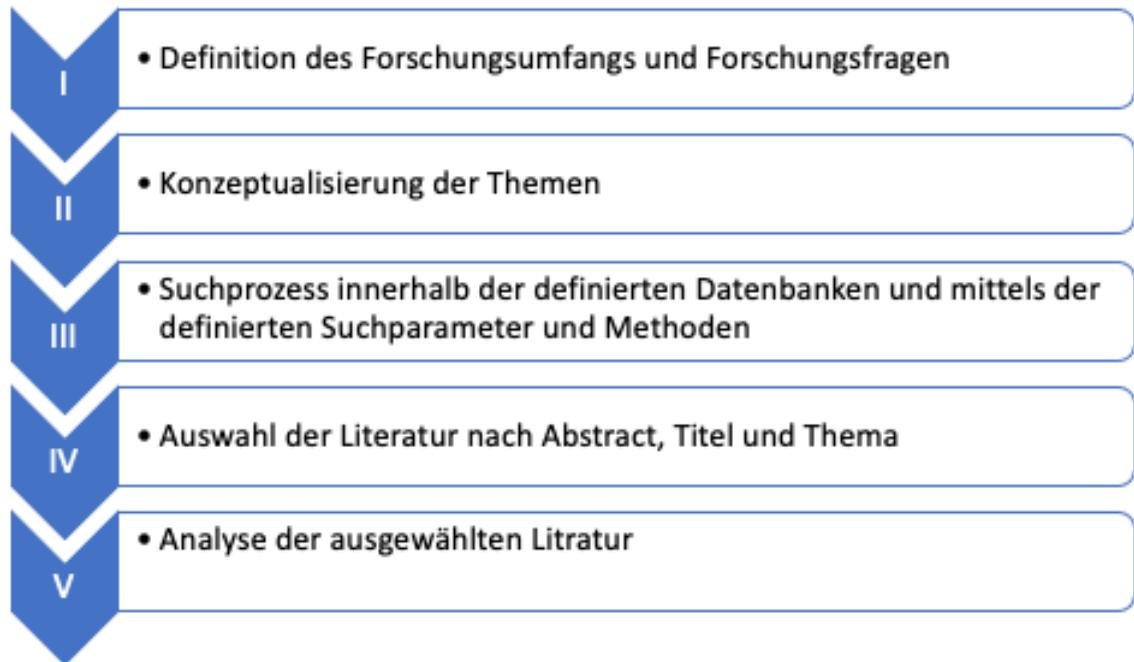


Abbildung 1: Vorgehen Literaturrecherche vom Brocke (2009)

Die Literaturrecherche wird an der FOM Düsseldorf und der Heinrich-Heine-Universität anhand der aufgelisteten Datenbanken in der Tabelle 1 durchgeführt. Die verwendeten Suchbegriffe sind der Tabelle 2 zu entnehmen. Dabei unterscheiden sich die Suchbegriffe je nach Kapitel und der zu untersuchenden Thematik.

Tabelle 1: Verwendete Datenbanken

Datenbank	Website
Google Scholar	http://scholar.google.de
Web of Science	https://www.webofscience.com
Elsevier Science Direct	http://www.sciencedirect.com
IEEE Xplore	http://ieeexplore.ieee.org/xplore/home.jsp
Springer Link	http://link.springer.com
arXiv	https://arxiv.org/
Emerald Insight	http://www.emeraldinsight.com/

Tabelle 2: Verwendete Suchparameter

Kapitel	Suchparameter
2	artificial+intelligence OR kuenstliche+intelligenz
2	explainable+ artificial+intelligence OR erklarende+ kuenstliche+intelligenz
2	machine+learning OR maschinelles+lernen
2	machine+learning+interpretability OR maschinelles+lernen+interpretierbarkeit
2	supervised+learning+regression OR ueberwachtes+lernen
2	Blackbox+algorithm OR Blackbox+Algorithmus
3	real+estate+machine+learning OR Immobilien+ maschinelles+lernen
3	real+estate+price+forecasting OR Immobilien+ Preisvorhersage
3	real+estate+ artificial+intelligence OR Immobilien+ künstliche+intelligenz
3	real+estate+pricing+models OR Immobilien+ Preismodelle
3	real+estate+hedonic+models OR Immobilien+ hedonische+Modelle
3	real+estate+prices OR Immobilien+ Preise
3	real+estate+pricing+factors OR Immobilien+ Preisgestaltung+Faktoren

Charakteristisch für das Prototyping ist es, dass der Ansatz auf einer evolutionären Sicht der Softwareentwicklung basiert und Einfluss auf den gesamten Entwicklungsprozess hat. Es werden frühe Arbeitsversionen (Prototypen) des zukünftigen Anwendungssystems erstellt und damit experimentiert. Ein Prototyp ist ein Betriebsmodell des Anwendungssystems und implementiert bestimmte Aspekte des zukünftigen Systems. Die Modelle, die erstellt werden, nutzen historische Daten, um Immobilienpreise zu prognostizieren. Dabei werden Methoden aus dem Bereich der XAI verwendet, um die Erklärbarkeit der Preisprognosen darzustellen. Der Entwicklungsprozess wird anhand eines ausgewählten Referenzmodells durchgeführt. Die Daten werden im Rahmen von Transformationsaktivitäten selektiert, vorverarbeitet und bereinigt. Die Datenanforderung der Modelle ist, dass ausschließlich quantitative Daten verwendet werden. Weiterführend beginnt das Modelltraining und es werden die Leistungen anhand von definierten Metriken gemessen. Zur Überprüfung der Generalisierbarkeit der Modelle wird der Datensatz in Trainings- und Testdaten aufgeteilt. Der Algorithmus lernt anhand des Trainingsdatensatzes und wendet sein Wissen auf die Testdaten an, um eine Prognose zu erstellen. Nach dem Training und der

Validierung der Modelle wird der Einsatz der XAI-Verfahren untersucht. In Abbildung 2 wird die Vorgehensweise in einem Ablaufmodell dargestellt.

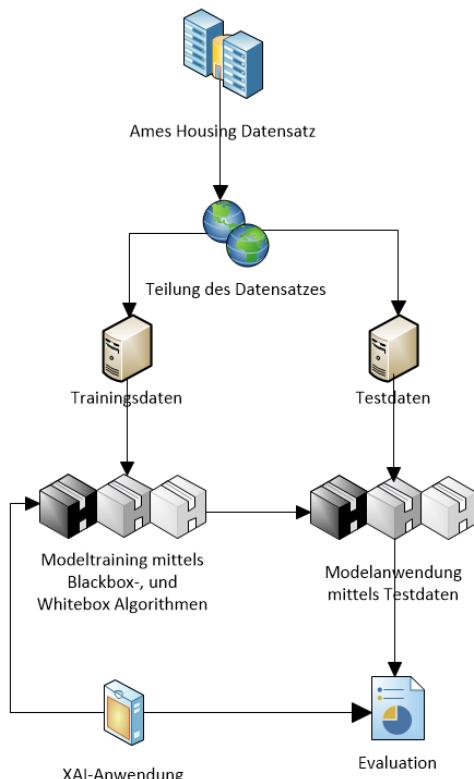


Abbildung 2: Vorgehen Prototyping (Religia et al., 2020, S.62)

Letztlich sind im Fließtext stets Personen männlichen und weiblichen Geschlechtsgleichmaßen angesprochen. Aus Gründen der besseren Lesbarkeit wird das generische Maskulinum verwendet (z. B. „Mitarbeiter“). Damit sind andere Gender eingeschlossen.

1.4 Aufbau der Arbeit

Die Arbeit gliedert sich in insgesamt sieben Kapitel. Die Einleitung beinhaltet die Problemstellung, die Forschungsfragen und die Zielsetzung der Arbeit sowie die Methodik und schließlich den Aufbau. Das zweite Kapitel stellt die theoretischen Grundlagen dar, die zum weiteren Verständnis der Ausarbeitung benötigt werden. Im dritten Kapitel werden die Erkenntnisse aus der Literaturrecherche zum Immobilienmarkt und zur Preisentwicklung erläutert. Das vierte Kapitel befasst sich mit der Entwicklung und Evaluation der Prognosemodelle mithilfe des cross industry standard process for data mining (CRISP-DM) Prozessablaufs. Die Ergebnisse der Untersuchung zum Einsatz von XAI-

Methoden werden im fünften Kapitel präsentiert. Die Ergebnisse werden im sechsten Kapitel unter Beachtung der Zielsetzung kritisch gegenübergestellt und beurteilt. Anschließend folgt eine aus der Diskussion resultierende Handlungsempfehlung für die Praxis, welche mit einer Reflexion des Vorgehens und den Limitierungen fortgeführt wird. Abschließend rundet das zusammenfassende Fazit und der Ausblick, inklusive des erkannten Forschungsbedarfs, die Ausarbeitung ab.

2 Theoretische Grundlage

2.1 Künstliche Intelligenz

Innerhalb der Psychologie wird die Intelligenz als ein hypothetisches Konstrukt definiert, welches die Fähigkeit aufweist, Kognition und Wissen miteinander zu kombinieren (Maier, 2018, S.1). Im Jahr 1950 hat bereits Alan Turing den Versuch unternommen, menschliche Intelligenz nachzubilden. Anhand des Turing Tests soll der Computer feststellen, ob dieser einen Spieler als Maschine von einem Mensch unterscheiden kann (Oppy & Dowe, 2021, S.1-2).

Dabei ist der Erfolg der Technologie bei der Nachbildung der Funktionen des menschlichen Gehirns limitiert. Der KI fehlt zur Informationsgewinnung ein hochkomplexes Bewusstsein und Sinnessensoren. Dies ist technisch derzeit (noch) nicht realisierbar (Mockenhaupt, 2021,S.48). Der gegenwärtige Einsatz von KI findet in verschiedenen Branchen diverse Anwendungsmöglichkeiten. Dabei ermöglicht der Einsatz von KI beispielsweise intelligente Spracherkennung, datengetriebene Problemlösungen oder mathematische Vorhersagemodellierungen (Mittal & Sharma, 2021, S.79-80). Rückwirkend betrachtet lässt sich festhalten, dass die Entwicklung von KI drei wesentliche Phasen durchlief. Zunächst sind intelligente Systeme durch das Kombinieren von Computer und Sensoren Technik geschaffen worden. Dabei ermöglicht die Sensoren Technik das Beschaffen von Daten, die der Computer anschließend verarbeitet. Die zweite Phase umfasst die Kombination von Big Data und KI, um eine Interaktion zwischen den immensen Datenmengen und der KI zu ermöglichen. Die letzte Phase entspringt den vorherigen zwei Phasen, indem durch das Sammeln und Bereitstellen von Daten die KI beginnt zu lernen. Daraus entstehen Handlungen seitens der KI, die beispielsweise in Form von automatischen Produktionsprozessen oder Robotik in Erscheinung treten (Liu, 2022, S.2). KI ermöglicht einen zuverlässigen Abgleich großer Datenmengen, was durch das Verarbeiten von Daten durch Algorithmen möglich wird. Dies führt zur Anwendung mathematischer Methoden und statistischer Verfahren, was die Mustererkennung in Daten ermöglicht und in der Disziplin des ML mündet, welche nachfolgend beschrieben wird (Mockenhaupt, 2021,S.50). Das Deep Learning (DL) bildet ebenso einen Teilbereich der KI und nutzt neuronale Netze, um riesige Datenbestände zu analysieren. Dabei sind Deep-Learning Modelle in der Lage eigenständig zu lernen, indem bereits erlernte Inhalte mit neuen

Erkenntnissen verknüpft werden (Lecun et al., 2015, S.436). Die Abbildung 3 zeigt zusammenfassend nach Mockenhaupt die KI als übergeordnete Disziplin für die Teilbereiche des ML und des Deep Learning, wobei letzteres keinen Schwerpunkt der Ausarbeitung darstellt und demnach nicht näher erläutert wird.

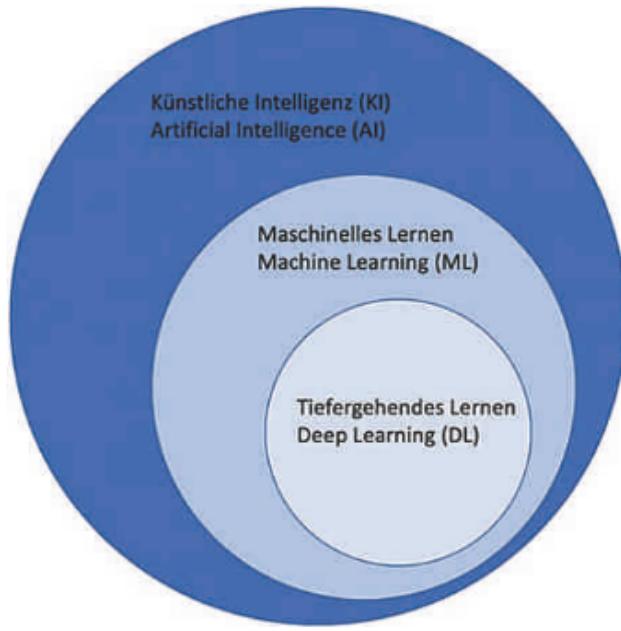


Abbildung 3: KI mit den Teilbereichen (Mockenhaupt, 2021, S.130)

2.2 Machine Learning

Bereits im Jahr 1959 beschrieb Arthur Samuel die Disziplin des ML als ein Fachgebiet, das Computern die Fertigkeiten des Lernens verleiht, ohne eine explizite Programmierung zu erfahren. Mitchell beschrieb im Jahr 1997 ML aus einer technischen Sicht als „ein Computerprogramm [...] aus Erfahrungen E in Bezug auf eine Aufgabe T und ein Maß für die Leistung P lernt, wenn seine durch P gemessene Leistung bei T mit der Erfahrung E anwächst“ (Geron, 2018, S.4). Im Rahmen des ML werden Algorithmen genutzt, deren Aufgabe es ist, Muster und Regelmäßigkeiten in Datensätzen zu erkennen. Die daraus entwickelten Lösungen befähigt den Algorithmus aus den erschlossenen Erkenntnissen Wissen zu generieren. Dies ermöglicht im Umkehrschluss nicht nur die Bewältigung von Wenn-Dann-Ereignissen, sondern die Lösung von Umstrukturierung Problemen (Mockenhaupt, 2021, S.133-136). Generell kann mathematisch festgehalten werden, dass die Aufgaben im ML der Optimierung einer Funktion $f(\theta)$ in einem festgelegten Bereich $\theta \in \Theta$ dient. Ziel kann es dabei sein die Funktion i der Formel 1 soweit es geht zu minimieren (Andrychowicz et al., 2016, S.3988).

$$\theta^* = \operatorname{argmin}_{\theta \in \Theta} f(\theta)$$

Formel 1: Minimierung der Verlustfunktion im ML (Andrychowicz et al., 2016, S.3988)

Der Einsatzbereich des ML kann dabei in diverse Aufgabenbereiche unterteilt werden. Die wesentliche Nutzung umfasst einfache Aufgaben, die einen hohen manuellen Aufwand erfordern und einer gewissen Anzahl Regeln obliegen. Der Einsatz von maschinellen Lernalgorithmen ermöglicht ebenso die Verringerung des Volumens an Kodierung und bietet für schwierige Aufgaben, die mit üblichen Methoden schwer zu lösen und bewältigen sind, eine richtige Herangehensweise. Außerdem kann ein ML-System sich neuen Daten anpassen und ermöglicht die Handhabung durch fluktuierende Umgebungen. Letztendlich bieten die Algorithmen die Möglichkeit eines Erkenntnisgewinns, der anhand komplexer Aufgabenstellungen und großen Datenmengen zu erzielen ist. Dabei lassen sich die ML-Systeme durch die Anzahl und Art der Überwachung beim Training einordnen (Geron, 2018, S. 6–7). ML weist ein breites Methodenspektrum auf. Daher konzentriert sich diese Arbeit im Rahmen der Regression auf die Besonderheiten und Anforderungen des überwachten Lernens, das Algorithmen wie Extreme Bosting Machine (EBM), K-nearest Neighbor (KNN), Extreme Gradient Boosting (XGB) und künstliche neuronale Netze (ANN) umfasst. Auf weitere Eigenschaften und Methoden wie dem unüberwachten Lernen oder dem bestärkenden Lernen kann aus Gründen des Umfangs nicht eingegangen werden. Die bestehenden Disziplinen der Klassifikation oder des Clusterings werden aufgrund des beschränkten Umfangs nicht näher erläutert, sind jedoch im Rahmen des ML ebenso zu beachten. Nachfolgend werden die für Ausarbeitung relevanten Algorithmen und Eigenschaften des ML näher erläutert.

2.2.1 Überwachtes Lernen

Das überwachte Lernen bildet neben dem unüberwachten- halbüberwachten und bestärkenden Lernen, eine wesentliche Kategorie im Rahmen der ML-Systeme dar. Im Rahmen des überwachten Lernens enthalten die Trainingsdaten bereits die gewünschte Lösung, in Form einer Zielvariable. Der Algorithmus kann anhand unbekannter Daten eine Vorhersage der Zielvariable anhand der gelernten Muster zur Verfügung stellen. Im Rahmen einer Regression kann dies beispielsweise die Vorhersage eines Immobilienpreises

bedeuten. Demnach werden die Daten durch den Lernalgorithmus gelabelt. Außerdem lassen sich Regressionsalgorithmen, wie die logistische Regression, zur Klassifikation nutzen, um die Wahrscheinlichkeit einer Zugehörigkeit zu bestimmen (Biamonte et al., 2017, S.198-199; Geron, 2018, S.9).

2.2.2 Regressionsmethoden

Die Regression dient der mathematischen Beschreibung einer funktionalen Abhängigkeit der Zielvariablen von einer oder mehreren unabhängigen Variablen (Beckman & Weisberg, 1987, S.1). Dabei ermöglichen die gewonnenen Erkenntnisse der Regression Wirkungsprognosen, die sich in Form von prädiktiven Modellen widerspiegeln. Die genutzten Regressionsmethoden werden durch mathematische Funktion definiert, die eine Beschreibung der Datenbasis ermöglichen und die Form, Art und Stärke des Zusammenhangs zwischen den betrachteten Variablen aufzeigt. Methodisch verfolgen Regressionsanalysen das Ziel, die Abweichung zwischen den tatsächlich beobachteten Werten und den vorhergesagten Werten zu minimieren. Dabei wird die Abweichung auch als Fehler oder als Residuen bezeichnet. Zur Verringerung der Abweichung gibt es zwei Optionen, nämlich das Durchführen von mathematischen Optimierungen oder die schrittweise Annäherung an die maximale Reduktion der Residuen mittels des Gradientenabstiegsverfahrens. Dabei werden die Parameter iterativ so lange angepasst, bis keine Minderung der Residuen möglich ist oder eine maximale Schrittzahl erreicht ist. Um das aus der angewandten Regressionsmethode resultierende Modell zu bewerten und zu evaluieren, können Leistungsindikatoren helfen, die im Kapitel 2.4 vorgestellt werden (Baars & Kemper, 2021, S.202). Im Rahmen der Ausarbeitung wird der Preis der Immobilien als Regressand festgelegt und die Werte der übrigen Variablen als Regressoren. Derzeit existiert eine Vielzahl an Algorithmen, die für die vorliegende Problemstellung genutzt werden können. Nachfolgend werden die Algorithmen vorgestellt, die in der Regression als Methoden zur Vorhersagemodellierung genutzt werden und unter Anwendung für die Anwendung der XAI-Methoden relevant sind.

2.2.3 Künstliche neuronale Netze

Künstliche neuronale Netze (ANN) stellen die Struktur des menschlichen Gehirns nach, indem ein Netzwerk aus unabhängigen Neuronen aufgebaut ist und diese sich in Schichten einordnen. Die Schichten sind miteinander verbunden, wobei eine Eingabe- und

Ausgabeschicht das Anfang und Ende bilden. Dabei beschreibt die Formel 2 das Mathematische Modell des Neurons und die Abbildung 4 die Ausgabefunktion eines Neurons. Das Neuron ist eine Nachbildung einer Schwellenwert-Einheit und empfängt Inputwerte von vorausgehenden Einheiten oder externen Quellen. Durch die Verschaltung nimmt jedes einzelne Neuron neben den Inputwerten auch einen fixen Wert b in Form eines Bias entgegen. Der Inputwert x_j wird in Form einer Multiplikation durch w_j gewichtet und nachfolgend wird alles gesamtheitlich summiert. Das Ergebnis wird im Ausgabeneuron $F(x)$ durch die Aktivierungsfunktion a verarbeitet (Agatonovic-Kustrin & Beresford, 2000, S.719-720; Müller & Lenz, 2013,S.81).

$$F(x) = a (w_j x_j + b)$$

Formel 2: Mathematische Funktionl eines künstlichen Neurons (Agatonovic-Kustrin & Beresford, 2000, S.719-720)

Im Gegensatz zur Klassifikation ist es bei einer Regression nicht notwendig, das gewichtete Ergebnis zu konvertieren. Die Aktivierungsfunktion spiegelt die tatsächliche Identitätsfunktion wider und gibt das gesamte Ergebnis als Outputwert weiter. Der entstandene Wert wird sukzessiv an die nächste neuronale Schicht weitergegeben, bis alle Schichten durchlaufen sind. Dabei wird der Ausgabewert der vorherigen Schicht als Inputwert verarbeitet. Das entstandene Ergebnis des ANN profitiert dabei durch die selbstständige Anpassung der Gewichtung in den Neuronen. Dies ermöglicht einen selbstlernenden Prozess des ANN (Krogh & Vedelsby, 1995, S.1; Müller & Lenz, 2013, S.81-83).

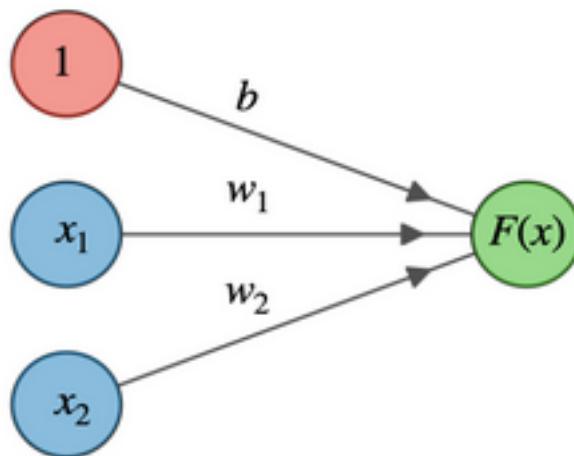


Abbildung 4: Mathematisches Modell eines künstlichen Neurons (Müller & Lenz, 2013, S.81-83)

Im Rahmen der Ausarbeitung wird die innerhalb des ML gängigste Variante eines ANN genutzt. Das Feed-Forward-Netz ist ein Lernalgorithmus, bei dem jedes Neuron den Outputwert an die Neuronen in der nächsten Schicht weitergibt. Demnach besteht keine Verbindung zwischen den Neuronen innerhalb einer Schicht. Zusätzlich begünstigt die Feed-Forward Architektur die Implementation des Backpropagation-Algorithmus. Zunächst werden im Rahmen des Forward-Passes Daten eingespielt. Durch die Gewichtung der Inputwerte ermöglicht die Aktivierungsfunktion die Berechnung der Ausgabewerte für die nächste Schicht. Anhand einer vorab festgelegten Fehlerfunktion wird mit den Ergebnissen ein Fehlerwert berechnet, welcher die Gesamtheit aller Datensätze inkludiert. Das Ergebnis wird gemittelt, mit dem Ziel, dass der Algorithmus die optimale Gewichtung bestimmt, was im Umkehrschluss den mittleren Fehler mindert. (Mao, 1996, S.39; Palanisamy et al., 2008, S.37-38).

Daraufhin erfolgt die eigentliche Backpropagation, nämlich die Anpassung der Gewichtung, damit eine Entwicklung in Richtung der größten Fehlerreduktion stattfindet. Dies impliziert die Ableitung des Fehlers eines Vektors nach allen Gewichten und meint den Gradient des Fehlers. Das Gradientenabstiegsverfahren wird dabei maßgeblich durch die Lernrate bestimmt, indem diese durch den Nutzer gesetzt wird. Die Lernrate beeinflusst die Lerngeschwindigkeit und das Lernverhalten des Algorithmus. Das Feedforward-basierte ANN ermöglicht es, die Gewichte schichtweise rückwärts zu berechnen und anzupassen. Es ist wichtig, die Lernrate so zu wählen, dass der Gradientenabstieg kein globales Minimum erreicht. Bei steigender Zahl an Schichten kann sich die Größe der Gewichtsanpassungen reduzieren oder gegen Null laufen, was zu unbrauchbaren Ergebnissen führen kann (Problem der "vanishing gradients"). Die Anpassung der Gewichtung resultiert in der Wiederholung des Forward-Pass und der Backpropagation. Der Durchlauf wird als Epoche benannt und stellt einen weiteren Hyperparameter dar, neben der Lernrate, der Aktivierungsfunktion und der Anzahl an Neuronen und Schichten. Jede Epoche verfolgt das Ziel den mittleren Fehler zu reduzieren, bis dieser hinreichend minimiert ist oder eine definierte Epochenzahl erreicht wird. Zusätzlich sorgen sogenannte Optimierungsfunktionen für die Verbesserung der Trainingsdauer und des Trainingsergebnisses, die je nach Anwendungsfeld des ANN sorgfältig gewählt werden müssen. Generell ist die sorgfältige Testung und Auswahl von diversen Hyperparametern zielführend, um einem Over- oder Underfitting entgegenzuwirken. (Andrychowicz et al., 2016,

S.3988-3989; Baars & Kemper, 2021, S.228). Für das ANN-Model wird die Python Bibliothek Keras genutzt, welche auf Tensorflow 2 aufbaut. Tensorflow stellt ebenfalls eine Open Source Bibliothek dar und ermöglicht das Entwickeln und Trainieren von ML-Modellen. Die Backpropagation wird seitens der genutzten Bibliotheken automatisch inkludiert und bedarf somit im Prototyping keine gesonderte Definition (Keras, 2022, S.1 ff.; Tensorflow, 2022, S.1 ff.)

2.2.4 K-nearest Neighbor

Der k-Nächste-Nachbar Algorithmus kann für Klassifikations- und Regressionsprobleme genutzt werden und vergleicht die Testinstanzen anhand Ihrer Similarität mit dem Trainingsdatensatz. Demnach wird der Algorithmus der Kategorie der Lazy-Learning Modelle zugeordnet. Dies impliziert, dass nur ein Trainingsdatensatz gespeichert wird und der KNN-Algorithmus keine Trainingsphase durchläuft (Baars & Kemper, 2021, S.177). Dabei wird der Trainingsdatensatz als $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ definiert, bei der jedes x_i eine unabhängige Variable darstellt, die durch Ihre Merkmalsausprägungen m zur Vorhersage von y_i führen kann. Weiterführend stellt N die Gesamtanzahl Instanzen dar. Bei der Hinzugabe einer Testinstanz x, wird jede Distanz d zwischen x und x_i aus T berechnet. Anschließend werden die ermittelten Berechnungen von d anhand Ihrer Werte sortiert und in die jeweilige korrespondierende Instanz $N_i(x)$ zugeordnet. Das Ergebnis wird als $y_i(x)$ vermerkt und ermöglicht die Vorhersage von \hat{y} der Testinstanz x in der Formel 3. Die Berechnung von \hat{y} ergibt sich durch die Berechnung der arithmetischen Durchschnitte der Ergebnisse von $y_i(x)$ und der ausgewählten Anzahl k. Die Auswahl von k impliziert die Anzahl von Nachbarn, die in die Betrachtung miteinfließen. Eine solide Anzahl an k kann im Rahmen einer Kreuzvalidierung ermittelt werden, basierend auf der Wurzel des mittleren quadratischen Fehlers (Imandoust & Bolandraftar, 2013, S. 607-608; Song et al., 2017, S.5).

$$\hat{y} = \frac{1}{k} \sum_{i=1}^k y_i(x)$$

Formel 3: Regression anhand von KNN (Song et al., 2017, S.5)

Dabei ist es von wesentlicher Bedeutung, dass der Nutzer sich für ein Verfahren zur Ermittlung der Distanz d entscheidet. Die gängigsten Abstandsmaße sind die Manhattan-Metrik und der Euklidische Abstand, wobei letzteres die genutzte Methode der Ausarbeitung widerspiegelt. Über die Formel der Formel 4 wird der Abstand zwischen dem Abfragepunkt und dem nächstliegenden Punkt durch eine gerade Linie, in Form eines Vektors, gemessen. Letztendlich gibt der KNN-Algorithmus dem Nutzer die Freiheit, die Anzahl der betrachteten Nachbarn k und das gewählte Abstandsmaß d zu bestimmen, sowie ob die tatsächlichen Distanzen in der Entscheidungsfindung als Gewichtung berücksichtigt werden sollen. (Hu et al., 2016, S.3).

$$d = \sqrt{\sum_{i=1}^n (x - x_i)}$$

Formel 4: Distanz Berechnung Nearest Neighbor (Hu et al., 2016, S.3)

2.2.5 Extreme Gradient Boosting und Extrem Boosting Machine

Das Boosting ist ein Ensemble-Lernverfahren, bei dem schlecht lernende Modelle durch Kombination zu einem besseren Gesamtmodell zusammengeführt werden. Hierbei werden die Fehler der einzelnen Modelle analysiert und bei der Zusammenstellung berücksichtigt. Üblicherweise werden Entscheidungsbäume als Basismodelle verwendet. Der Algorithmus Gradient-Boosting (GB) nutzt neben der Boosting Methode ebenso das Gradientenabstiegsverfahren, um die Verlustfunktion zu minimieren. Demnach sind GB auch dem Bereich der Ensemble-Lernverfahren zugeordnet (Natekin & Knoll, 2013, S.1). Dabei lässt sich der extreme Gradienten Boosting (XGB) Algorithmus als skalierbares ML-System für Entscheidungsbaum Algorithmen definieren. Zusätzlich stellt das XGB eine effiziente und skalierbare Implementation des GB Frameworks dar, welches von Friedman (2001) bereits im Jahre 2001 beschrieben wurde. Der XGB-Algorithmus unterscheidet sich von dem traditionellen Gradient Boosting durch seine Fähigkeit zur parallelisierten und schnelleren Verarbeitung, sowie durch sein verteiltes Training. Darüber hinaus beinhaltet der XGB-Algorithmus einen zusätzlichen Randomisationsparameter, der zur Minderung der Korrelation zwischen Entscheidungsbäumen genutzt werden kann.

Zusätzlich nutzt der Algorithmus Lasso- und Ridge-Regularisierung, um die Generalisierung zu verbessern und ein Overfitting zu vermeiden. Ein maßgeblicher Unterschied lässt sich dabei durch die Arbeitsweise des XGB erkennen, da dieser die Newton-Boosting Methode verwendet, welches dem Newton-Raphson-Verfahren unterliegt. Ergänzend nutzt das XGB entgegen der ersten partiellen Ableitung, die Gradienten zweiter Ordnung der Verlustfunktion nach dem Taylor Approximation. Dies ermöglicht dem Modell Informationsreicher das tatsächliche Minimum der Verlustfunktion zu erreichen, indem die Richtung und Größe des Abstiegs besser vollzogen wird. Dabei wird das Newtonverfahren genutzt, um Näherungswerte der Nullwerte der Modellgleichung zu finden. Die Taylorformel nutzt die Approximation der zweiten Ordnung in der Verlustfunktion, um eine Verbindung zum Newtonverfahren herzustellen (Chen & Guestrin, 2016,S.785-786; Zhang et al., 2022,S.2-3). Nachfolgend ergibt sich die Mathematische Darstellung nach Chen und Guestrin (2016) und Budholiya (2022). Die Zusammenführung der Verlustfunktion und des Regulierungsterms ergibt die Objektfunktion des XGBoost in der Formel 5. Dabei stellt L die Verlustfunktion dar, die den Vergleich zwischen der Vorhergesagten \hat{y}_i und dem tatsächlichen Wert y_i darstellt. R stellt demnach die Regulierungsfunktion dar, die die Komplexität der Funktionen der Entscheidungsbäume f_i senkt, die es zu trainieren gilt.

$$Obj = \sum_{i=1}^n L(\hat{y}_i, y_i) + \sum_{i=1}^k R(f_i)$$

Formel 5: XBM Objekt Funktion (Budholiya et al., 2022, S.4516)

Aufgrund des begrenzten Umfanges der Ausarbeitung wird die finale Objektfunktion des XGB-Algorithmus kumuliert dargestellt. Die Herleitung der Funktion wird den Rahmen der Theoretische Grundlage überspannen. Letztlich wird in der Formel 6 die gesamtheitliche resultierende Objektfunktion dargestellt, die für jeden Durchlauf t genutzt wird. Zusammenfassend lässt sich konstatieren, dass der XGB-Algorithmus das Ziel anstrebt, durch den Gradientenabstieg eine additive Funktion zu berechnen, die die Objekt Funktion minimiert (Nguyen et al., 2021, S.6).

$$Obj = \sum_{i=1}^n [g_i f_i(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \gamma T + \alpha \sum_{j=1}^T w_j + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$$

Formel 6: XBM Objekt Funktion Final (Budholiya et al., 2022, S.4517)

Die Vorhersage der abhängigen Variablen kann mithilfe der Formel 7 berechnet werden, die die Lernrate sowie eine bestimmte Anzahl von schwachen Lernmodellen berücksichtigt. Hierfür stellt $\hat{f}_{(m)}(x)$ den vorhergesagten Wert dar, nachdem alle x_i Werte durch alle m Anzahl an Bäumen durchlaufen sind für die Funktion F. \hat{y}_i stellt dabei den finalen Vorhergesagten Wert aller x_i dar (Zhang et al., 2022, S.3-4).

$$\hat{y}_i = \sum_{m=0}^M \hat{f}_{(m)}(x_i), f_m \in F$$

Formel 7: XBM Algorithmus Modellergebnis (Zhang et al., 2022, S.3-4)

Der Explainable Boosting Machine (EBM) Algorithmus ist ein Glass-Box Modell, welches, ein generalisierendes additives Modell (GAM) darstellt und den Boosting Algorithmen zuzuordnen ist. Die Formel 8 zeigt auf, dass g als Linkfunktion dabei das GAM innerhalb der Regression oder Klassifikation adaptiert. Im Vergleich zu traditionellen Ansätzen der GAMs besitzt der EBM-Algorithmus die Fähigkeit, jede Variablenfunktion f_j durch die Anwendung von Techniken wie Bagging oder Gradient Boosting zu erlernen. Dabei wird während des Boostings durch eine vorab niedrig definierte Lernrate lediglich eine Variable zeitgleich trainiert. Die Reihenfolge der Variablen spielt während des Boostings keine Rolle, da alle gleichwertig in einer festgelegten Abfolge trainiert werden. Dies ermöglicht die Verminderung einer Kollinearität und die Ermittlung der erfolgversprechendsten Funktion f_j , im Hinblick auf den Beitrag, die die Variablen zur Vorhersagequalität des Modells und zur Problemlösung leisten (Nori et al., 2019, S. 2-3).

$$g(E[y]) = \beta_0 + \sum f_j(x_j)$$

Formel 8: EBM-Algorithmus (Nori et al., 2019, S. 2-3)

Letztlich erkennt und inkludiert der EBM-Algorithmus automatisch Interaktionseffekte, welche durch den mathematischen Term in Formel 9 beschrieben ist. Dadurch kann eine höhere Vorhersagegenauigkeit erzielt werden, ohne dass die Verständlichkeit des Algorithmus sinkt (Nori et al., 2019, S. 3).

$$g(E[y]) = \beta_0 + \sum f_j(x_j) + \sum f_{ij}(x_i, x_j)$$

Formel 9: Interaktionseffekt EBM-Algorithmus (Nori et al., 2019, S. 2-3)

2.3 Erklärende künstliche Intelligenz

Die erklärende künstliche Intelligenz stellt eine Möglichkeit dar, die Ergebnisfindung von KI für Menschen nachvollziehbar und damit interpretierbar darzustellen. Ziel der verschiedenen XAI-Methoden ist es, die Gründe einer Entscheidung, die durch die Modelle getroffen wurden, transparent darzulegen. Die Erklärbarkeit ermöglicht eine bessere Überprüfung des Entscheidungsprozesses und trägt dazu bei, die Modellqualität zu optimieren und das Ergebnis interpretierbar zu machen (Barredo Arrieta et al., 2020, S.2 ; Gunning et al., 2019, S.3). Dabei besteht zunächst die Schwierigkeit Interpretierbarkeit und Erklärbarkeit mathematisch zu definieren. Nicht mathematische Definition beschreiben die Erklärbarkeit und Interpretierbarkeit als die Fähigkeit des menschlichen Verständnisses die Ursachen einer Entscheidung zu verstehen. Andererseits wird die Interpretierbarkeit und Erklärbarkeit mit der Fähigkeit des menschlichen Verständnisses gleichgesetzt, Ergebnisse von Modellen auf Basis der bestehenden Daten Vorhersagen zu können (Kim et al., 2016, S.7; Miller, 2019, S.14).

Die diversen bestehende XAI-Methoden weisen einen unterschiedlichen Umfang an Erklärung auf, die dem Nutzer geboten werden (Das & Rad, 2020, S.1). Hierbei wird im Rahmen der Anwendung von KI bereits Anfangs zwischen interpretierbaren, spezifischen und agnostischen Modellen unterschieden. Letzteres stellt einen Ansatz dar, der die Erklärung von dem ML-Modell separat und getrennt vom Algorithmus ausführt. Der Einsatz von Agnostischen Methoden findet überwiegend nach dem eigentlichem Modell Training statt oder im Rahmen der Evaluierung. Generell birgt der Einsatz von

Agnostischen Modellen den Vorteil, dass eine flexible Auswahl von ML-Algorithmen stattfinden kann und Erklärungen diverser ML-Modelle miteinander verglichen werden können. Spezifische Methoden können nur auf einen bestimmten Typen von Modellen angewendet werden, wie beispielsweise auf Baumartige- oder Deeplearning Algorithmen. Die Anwendung von spezifischen Modellen findet ebenso nach dem Training oder innerhalb der Evaluierung statt. (Ribeiro et al., 2016, S.95). Beide Ansätze finden im Rahmen von sogenannten Blackbox-Modellen Ihre Anwendung, deren Funktionen vom mathematischen Standpunkt und menschlichen Verständnis aus schwerer zu beschreiben und nachvollziehen sind. Dies inkludiert beispielsweise Neuronale Netze, extreme Gradient Boosting Machines oder support-vector machines. Interpretierbare Modelle werden auch als Whitebox-Modelle beschrieben und bilden das komplementär zu den Blackbox-Modellen. Diese agieren für den Menschen nach verständlichen Mustern, Regeln und Entscheidungen. Diese umfassen lineare Regressionen, Entscheidungsbäume oder auch generalisierende additive Modelle wie der EBM-Algorithmus (Loyola-Gonzalez, 2019, S.1).

Im Kontext von ML-Algorithmen ist es weit verbreitet, dass Blackbox-Modelle eine hohe Vorhersagegenauigkeit aufweisen, die Interpretierbarkeit der Modelle jedoch darunter leidet. Bei Whitebox-Modellen wird ein gegenteiliger Trend seitens diverser Autoren beschrieben. Für die nachfolgende Ausarbeitung ist es elementar von Bedeutung zu verstehen, dass nicht, wie in vielen Beiträgen fälschlicherweise von einer höheren Genauigkeit der Blackbox-Modelle gesprochen wird, sondern die Performance der Modelle damit gemeint ist. Demnach erzielen Blackbox-Modelle nach den Autoren eine bessere Leistung als durchsichtige Whitebox-Modelle (Hall & Gill, 2018, S.1-2; Saeed & Omlin, 2021, S.2; Sahakyan et al., 2021, S. 135394; Z. Zhang et al., 2022, S. 93114). Die Genauigkeit kann in Rahmen der Klassifikation benannt werden, jedoch nicht in den übrigen Disziplinen des ML. Demnach wird nachfolgend von der Performance und Interpretierbarkeit der Modelle gesprochen. Die Einordnung der verwendeten Modelle innerhalb der Performance- und Interpretierbarkeitsspektrum ist der Abbildung 5 zu entnehmen. Die Einordnung der Modelle erfolgt dabei nach den Erkenntnissen von Barredo et al. (2020). Zusätzlich wird die Einordnung des EBM-Algorithmus nach den Erkenntnissen von Nori et al. (2019) durchgeführt.

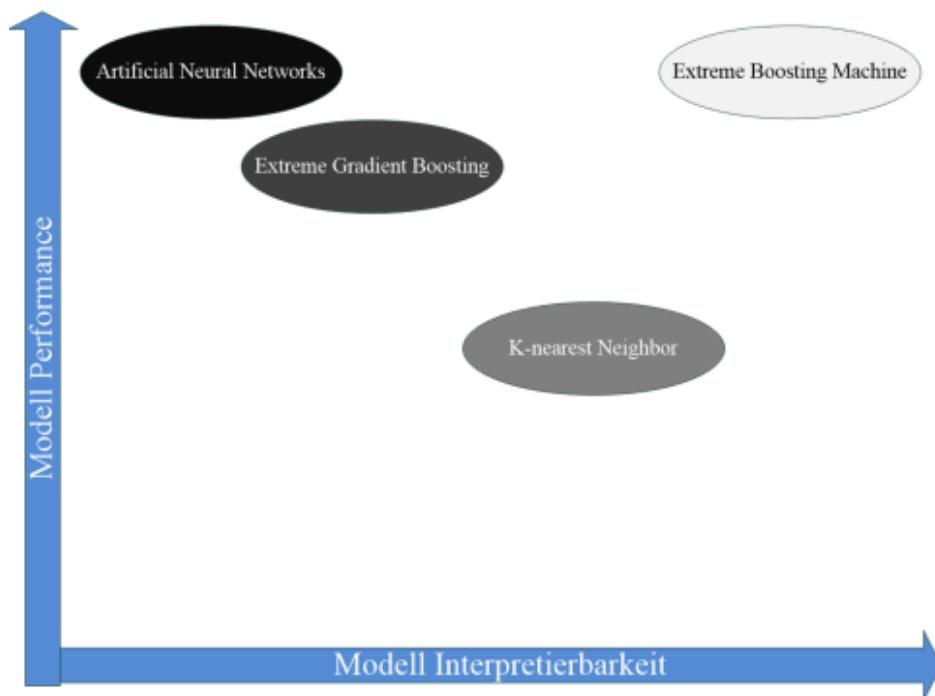


Abbildung 5: Performance gegen Interpretierbarkeit (Barredo Arrieta et al., 2020, S.31, Nori et al., 2019, S. 2-3)

Darüber hinaus sind die XAI-Methoden in lokale und globale Methoden unterteilt. Globale Methoden fördern das Verständnis, in welcher gesamtheitlichen Beziehung die unabhängigen Variablen mit dem Modellergebnis und demnach der Zielvariable stehen. Die gesamtheitliche Betrachtung der Globalen Methoden birgt jedoch das Risiko, dass es sich um Annäherungswerte handelt, die auf diverse einzelne Beobachtungen nicht zutreffen. Hierfür können lokale Interpretationen eingesetzt werden, die das Vorhersageergebnis für eine einzelne oder ähnliche gruppierte Beobachtungen interpretierbar gestalten. Die Untersuchung einer einzelnen Instanz unterstützt dabei die Förderung des Vertrauens hinsichtlich der Möglichkeit für Nutzer, Modellergebnisse auf Faktoren wie ethische Korrektheit, Fairness, Stabilität, Abhängigkeiten und Plausibilität zu untersuchen. Die Gemeinsame Betrachtung der Ergebnisse von lokalen und globalen Methoden ermöglicht die vielversprechendsten Chancen, interpretierbare und verständliche Modellergebnisse zu erzielen (Barredo Arrieta et al., 2020, S.13). Es gibt bereits eine große Anzahl an XAI-Verfahren, deren Zahl weiterhin rasant wächst. Die nachfolgenden drei vorgestellten XAI-Methoden werden in der Ausarbeitung sowohl für die globale als auch für die lokale Betrachtung der Modellergebnisse verwendet.

2.3.1 Shapley Additive Explanations

Die Shapley Additive Explanations (SHAP) Methode ist erstmals durch Lundberg und Lee im Jahr 2017 als spieltheoretischer Ansatz eingeführt worden. Dabei werden die aus der Spieltheorie abgeleitete Shapley Werte nach Lloyd Shapley (1953) genutzt, um Ergebnisse von ML-Modellen zu erklären. Der Shapley Wert einer unabhängigen Variable erschließt sich durch die Gesamtbetrachtung aller möglichen Kombinationen der Variablen, im Hinblick auf die Gewichtung und den durchschnittlichen Beitrag den der Wert der unabhängigen Variable zur Vorhersage leistet (Ghorbani & Zou, 2019, S.4053-4054). Die vorausgegangene Beschreibung lässt sich mathematisch aus der Formel 10 entnehmen.

$$\phi_j(val) = \sum_{S \subseteq \{1, \dots, p\}} \frac{|S|! (p - |S| - 1)!}{p!} (val(S \cup \{j\}) - val(S))$$

Formel 10: Shapley Wert Formel (Ghorbani & Zou, 2019, S.4053-4054)

Die Formel 11 weist auf die mathematische Darstellung der Shapley Werte innerhalb der Modelbetrachtung hin, wobei S für eine Teilmenge der Variablen im Modell steht und x für den Vektor der Werte der unabhängigen Variablen, die es zu erklären gilt. Dabei umfasst p die Anzahl der Variablen und $val_z(S)$ ist die Vorhersage der marginalisierten Werte in S (Nowak & Radzik, 1994. S.43-44).

$$val_z(S) = \int \hat{f}(x_1, \dots, x_p) d\mathbb{P}_{x \notin S} - E_X(\hat{f}(X))$$

Formel 11: Shapley Wert Ermittlung anhand eines Modells (Nowak & Radzik, 1994. S.43-44)

Außerdem lässt sich festhalten, dass Shapley Werte die Charakteristika der Effizienz, Symmetrie, Additivität und die Dummy-Spieler-Eigenschaft erfüllen. Die Zusammenfassung der vier Axiome kann innerhalb der Spieltheorie als eine faire Auszahlungsfunktion definiert werden (Winter, 2002, S. 2027-2078). Der Einsatz von SHAP wird anhand der von Lundberg (2018) entwickelten SHAP-Bibliothek vollzogen.

2.3.2 Local Interpretable Modell-Agnostic Explanations

Die XAI-Methode Local interpretable modelagnostic explanations (LIME) ist durch Marco Tulio Ribeiro, Sameer Singh und Carlos Guestrin (2016) im Jahre 2016 erstmalig entwickelt worden. Der LIME-Algorithmus generiert Vorhersagen auf Basis verschiedener Variationen und Kombinationen der Variablen des Datensatzes. Anhand des neu geschaffenen Datensatzes trainiert LIME ein interpretierbares Modell. Beispielsweise werden bei Textdaten diverse Versionen des Ursprungstextes generiert, wobei eine zufällige Anzahl von Wörtern entfernt wird. Anschließend erfolgt die Klassifizierung der neuen Texte in Kategorien, die auf die An- oder Abwesenheit spezifischer Schlüsselwörter untersucht werden, um somit den Einfluss der Wörter die Klassifizierung zu erkennen. Das interpretierbare Modell der Instanz x wird anhand des Modells g ermittelt, welches die Verlustfunktion L minimieren soll. Dabei misst L wie nah die Verlustfunktion des erklärbaren Modells an der Verlustfunktion des Ausgangsmodells f ist, um somit Vorhersagekraft beider Modelle zu vergleichen. Dabei sollte die Modelkomplexität $\Omega(g)$ niedrige gehalten werden, beispielsweise durch die Nutzung von einer geringeren Anzahl von abhängigen Variablen. Das Proximitätsmaß π_x bestimmt das Distanz- und Ähnlichkeitsmaß der Instanz x , dass für die Erklärung in Betracht gezogen wird. Generell eignet sich der LIME-Algorithmus für eine lokale Betrachtung und optimiert nur entlang der Verlustfunktion (Ribeiro et al., 2016, S.97-99; Wentura & Pospeschill, 2015, S.165-166). Die Mathematische Betrachtung lässt sich in der Formel 12 nachvollziehen.

$$\text{explanation}(x) = \arg \min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

Formel 12: LIME-Algorithmus (Ribeiro et al., 2016, S.97-99)

Eine globale Betrachtung wird seitens der LIME-Bibliothek nicht ermöglicht. Die Vorgehensweise nach O’Sullivan (2022) ermöglicht jedoch eine globale Erkundung, indem die Gewichtungen jeder Variable für jede lokalen vorhersage aggregiert wird. Demnach wird für jede Variable der absolute Mittelwert entlang der lokalen Beobachtungen berechnet. Hierbei kann die Anzahl an Beobachtungen ausgewählt, die für die globale Analyse des Modells genutzt werden können (O’Sullivan, 2022, S.1 ff.).

2.3.3 Interpreter ML

Das open-source Paket InterpretML vereint diverse XAI-Methoden innerhalb einer Bibliothek und ist entlang des bereits beschriebenen EBM-Modells entwickelt worden. Neben den definierten Methoden SHAP und LIME für die lokale Analyse, umfasst InterpretML weitere erklärende Verfahren und Methoden wie die Morris-Sensitivity-Analyse oder das Partial Dependence Plot (PDP). Dabei unterscheidet das von Microsoft entwickelte Paket ebenso zwischen sogenannten Glass-Box- und Black-Box Modellen. Eine Besonderheit bietet dabei die Möglichkeit des Vergleichs von interpretierenden Algorithmen anhand von integrierten Dashboards und interaktiven Visualisierungen (Bird et al., 2020, S3-4).

Für die globale Analyse wird seitens InterpretML die Morris-Sensitivity-Analysis verwendet, die auch unter der Bezeichnung Elementary Effects Method oder Elementary Effects Screening bekannt ist. Dies ist eine Methode zur quantitativen Beurteilung der Sensitivität von Modellausgaben gegenüber einzelnen Modellparametern. Sie verwendet eine statistische Überprüfung, um den Einfluss jedes Parameters auf die Ausgabe zu quantifizieren und die Unsicherheit im Modell zu reduzieren. Die Methode wurde erstmals von Morris (1991) beschrieben und wird für diverse Betrachtungen von Modellen herangezogen. Dabei besteht die Sensitivitätsanalyse bereits seit Jahrzehnten für die lokale Betrachtung und wurde seitens Morris überarbeitet, um eine globale Betrachtung gewährleisten zu können. Dabei existiert keine einheitliche Formel, die die Morris-Sensitivity-Analysis beschreibt, da es sich um einen Ansatz und keine exakte Methode handelt (Iooss & Lemaître, 2015, S.5-6; Morris, 1991, S.161 ff.).

Außerdem wird ein PDP-Diagramm im Rahmen der globalen Analyse seitens InterpretML zur Nutzung bereitgestellt. Ein PDP-Diagramm zeigt dabei die Abhängigkeit einer Modellausgabe von einem einzelnen Eingabeparameter auf, wobei die Werte aller anderen Parameter konstant gehalten werden. Somit wird ermöglicht, den Einfluss eines bestimmten Parameters auf die Vorhersage zu visualisieren. Dies ermöglicht beispielsweise bei quantitativen Variablen zu erkennen, ab welchen tatsächlichen Variablenwert die Variable das prädiktive Ergebnis positiv, negativ oder ohne jeglichen Effekt beeinflusst. Die Ursprünge der PDP-Diagramme lassen sich nicht auf eine einzelne Person zurückzuführen. In den 1990er Jahren sind erste P von Forschern im Bereich ML entwickelt worden, um die Vorhersage von komplexen ML zu visualisieren und zu interpretieren.

Einer der frühen Verfasser von PDPs ist Friedman (2001), wobei die Autoren Tibshirani, Hastie, Narasimhan und Chu (2002) und Goldstein, Kapelner, Bleich, Pitkin und Rice (2015). PDPs haben sich seitdem zu einem weit verbreiteten Werkzeug im Bereich interpretierbares ML entwickelt (Goldstein et al., 2015, S.1-2; Sahakyan et al., 2021, S.135404).

2.4 Leistungsindikatoren

Um im Rahmen der Vorhersagemodellierung die diversen Algorithmen bewerten zu können, ist es von essenzieller Bedeutung klar definierte Leistungsindikatoren zu benennen. Leistungsindikatoren, nachfolgend auch Leistungsmetriken genannt, ermöglichen es die Ergebnisse der verschiedenen ML-Algorithmen sichtbar und vergleichbar darzustellen. Dies ist notwendig, um im späteren Verlauf die XAI-Methoden an generalisierbaren Modellen zu untersuchen und somit eine aussagekräftige Beurteilung hinsichtlich der Transparenz, Nachvollziehbarkeit und Interpretierbarkeit gewährleisten zu können. Die Leistungsmetriken dienen zur Validierung der Modelle, indem die Ergebnisse des Trainingsdatensatzes mit den Werten der Testdaten verglichen werden (Carvalho et al., 2019, S.1-2). In der Tabelle 3 werden die für die Ausarbeitung herangezogenen Leistungsmetriken aufgeführt, die die Bewertung der Modellgüte von diversen Standpunkten aus ermöglichen soll.

Der Mean Absolute Error (MAE) beschreibt den mittleren absoluten Fehler der Vorhersage, in Form der Abweichungshöhe des vorhergesagten Wertes \hat{y}_i zum tatsächlichen Wert y_i . n bildet die Gesamtzahl der Vorhersagewerte ab, die derselben Anzahl an Beobachtungswerten entsprechen muss. Dabei definiert der MAE, dass je kleiner der mittlere absolute Fehler ist, desto geringer ist die Anzahl an Fehlern in der Vorhersage, womit die Modellgüte steigt. Zu beachten ist dabei, dass eine positive oder negative Abweichungsrichtung nicht seitens des MAE angegeben wird. Der Term Absolute umfasst eine mathematische Funktion, die einen positiven Ausgabewert zur Folge hat und somit das Ablesen einer Abweichungsrichtung nulliert. Außerdem eignet sich MAE nicht zur Bestimmung der relativen Größe des Fehlers, ist jedoch stabiler gegenüber Ausreißern. Für die genannten Herausforderungen kann jedoch die Kombination mit anderen Leistungsmetriken Abhilfe schaffen (Qi et al., 2020, S.3413; Willmott & Matsuura, 2005, S. 79-81).

Root Mean Squared Error (RMSE) als Leistungsmetrik ist strenggenommen eine Erweiterung des Mean squared Error (MSE), die die Quadratwurzel des mittleren quadratischen Fehlerwertes darstellt. Dabei neutralisiert die Wurzel den quadratischen Term, um das Ergebnis in derselben Einheit wie Vorhersage- und Beobachtungswert darzustellen. Dies ermöglicht ein leichteres Verständnis Ergebnisses, welches durch einen numerischen Wert nahe der 0 eine höherwertige Modellgüte definiert. Dies muss jedoch im Kontext der vorherzusagenden Zielvariable gesetzt werden. Da die Zielvariable den Kaufpreis von Immobilien darstellt und diese sich in fünf- bis sechsstelligen numerischen Bereichen bewegen, wird dementsprechend je nach Modellgüte der RMSE-Wert höchstwahrscheinlich weit von einem Nullwert entfernt liegen. Nützlich ist die Nutzung von RMSE falls die Streuung von Bedeutung ist und Extremwerte normalisiert werden müssen (Chai & Draxler, 2014. S.1248).

Der Mean absolute percentage Error (MAPE) definiert die mittlere absolute prozentuale Abweichung der Fehlerwerte und drückt die Abweichung der Vorhersage als prozentuales Verhältnis zum wahren Wert aus. Am effektivsten lässt sich ein aussagekräftiger MAPE-Wert erzielen, sobald keine Extrem- oder Nullwerte vorhanden sind. Dabei wird erneut die Anzahl an den vorhergesagten Werten \hat{y}_i von den vorhergesagten Werten y_i subtrahiert, wobei die Differenz durch y_i dividiert wird. Dabei gilt, dass je niedriger der MAPE-Wert ist, desto wertiger das Modell. Prozentual interpretiert, kann ein Modell als sehr gut betrachtet werden, falls der MAPE Wert $< 10\%$ entspricht und als gut falls der Wert zwischen $10\% - 20\%$ liegt (de Myttenaere et al., 2016, S. 2-4).

Letztendlich wird der adjusted R^2 Score als letzte Leistungsmetrik gewählt und bestimmt welcher Teil der Varianz in den Daten N durch die Regression beschrieben wird. Es ist eine modifizierte Version von R^2 und berücksichtigt die Anzahl p der unabhängigen Variablen im Modell. Je mehr unabhängige Variablen im Modell hinzugefügt werden, desto höher wird R^2 tendenziell, unabhängig davon, ob diese Variablen wirklich einen Einfluss auf die abhängige Variable haben. Der adjusted R^2 korrigiert entsprechend und stellt sicher, dass es nur für tatsächlich erklärende Variablen einen Zuwachs gibt. Dieser kann als prozentualer Wert interpretiert werden und impliziert, dass ein adjusted R^2 Score mit dem Ergebnis 1 alle beobachteten Werte zu 100 % vom Prognosemodell erklärt werden, wobei ein Score von adjusted $R^2 = 0$ dem Modell keinen prognostischen Wert zuordnet. Ein wichtiges Kriterium ist hierbei, dass das Bestimmtheitsmaß *adjusted R²* nur bei

linearen Regressionen gültig ist. Demnach wird die Benutzung des adjusted R^2 Scores aufgrund dessen mit Bedacht als Leistungsmetrik verwendet (Chicco et al., 2021, S.5-8).

Tabelle 3: Leistungsmetriken zur Bewertung von Machine Learning Algorithmen

Leistungsmetrik	Formel	Quelle
Mean absolute Error (MAE)	$MAE = \frac{1}{N} \sum_{i=1}^N y_i - \hat{y}_i $	(Qi et al., 2020, S.3413; Willmott & Matsuura, 2005, S. 79-81).
Root mean squared Error (RMSE)	$RMSE = \sqrt{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}}$	(Chai & Draxler, 2014, S.1248)
MAPE	$MAPE = \frac{1}{N} \sum_{i=1}^N \left \frac{y_i - \hat{y}_i}{y_i} \right $	(de Myttenaere et al., 2016, S. 2)
adjusted R^2	$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{(y_i - \bar{y})^2}$ $1 - \frac{(1 - R^2)x(N - 1)}{N - p - 1}$	(Chicco et al., 2021, S.5-8)

2.5 Cross industry standard Process for Data Mining

Das CRISP-DM Modell bildet nur eine von vielen Vorgehensweisen ab, um ML-Projekte im Data Mining (DM) Prozess transparent darzustellen und zu strukturieren. Dabei wird der DM- Prozess nach CRISP-DM in sechs sequenzielle Phasen aufgeteilt und unterstützt dadurch die Implementierung der Aufgaben von ML und Data Mining Prozessen (Shafique & Qaiser, 2014, S.217). Zur Strukturierung des ML-Projektes und der Aufgaben, benötigt es nach dem CRISP-DM Modell die Definition der Aufgaben und der beteiligten Rollen. Dies etabliert einen strukturierten Prozess, der einem Projektteam ein klar definiertes Ziel vermittelt und somit ein kollektives Verständnis der Geschäftsanforderungen schafft (Chapman et al., 2000, S. 12-15).

Innerhalb dieser Ausarbeitung wird eine Problem- und Zieldefinition in der Einleitung formuliert und nicht wie im CRISP-DM Referenzmodells im Hauptteil platziert. In der Praxis ist eine detaillierte Formulierung der Vorgehensweise und Aufstellung des Projektplans in der Phase Business Understanding von elementarer Wichtigkeit. In dieser Arbeit wird auf eine vertiefte Granularität dieser Phase verzichtet. Ebenso wird die Deployment Phase im Sinne der Vollständigkeit im Grundlagenteil behandelt, jedoch in

der Ausarbeitung nicht berücksichtigt, da kein realer Praxisbezug zu einem Unternehmen hergestellt werden kann. Die Phasen des CRISP-DM Modells werden iterativ durchlaufen und sind einzeln in der Abbildung 6 dargestellt. Nachfolgend werden die einzelnen Phasen vorgestellt.

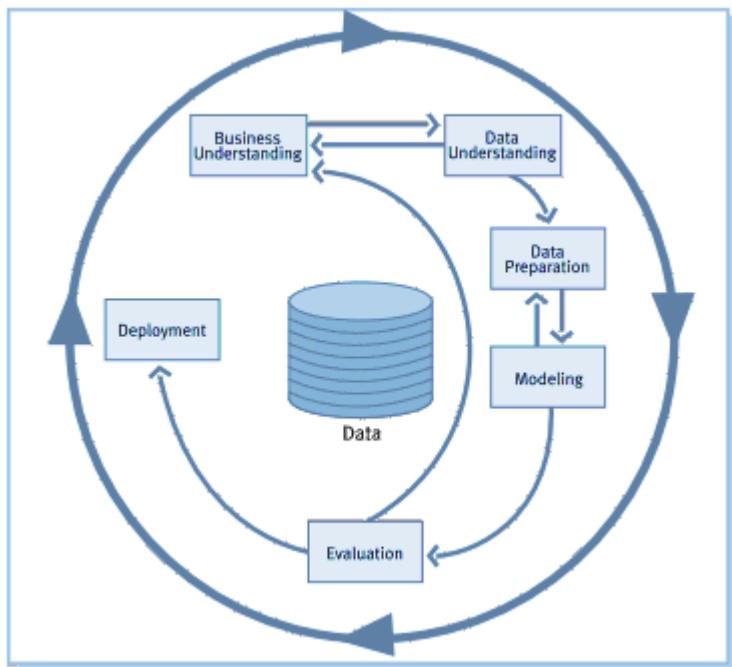


Abbildung 6: Phasen des CRISP-DM Referenzmodells (Chapman et al., 2000, S.12)

2.5.1 Business Understanding

Die Business Understanding Phase spiegelt die erste Phase wider und fokussiert sich auf die Projektziele. Dabei geht es hauptsächlich darum, die Anforderungen zur Erreichung der Ziele aus einer geschäftlichen Perspektive zu betrachten und die gewonnenen Erkenntnisse in einer Problemdefinition widerzugeben. Darauf aufbauend wird der Plan zur Erreichung der Ziele erstellt, indem konkret formuliert wird, welche Maßnahmen zur Zielerreichung benötigt werden. Zusätzlich umfasst die Business Understanding Phase das Festlegen von Ressourcen (Projektteam, Budget zeitliche Kapazitäten), die Entwicklungsumgebung und die Datenquelle (Chapman et al., 2000, S. 12).

2.5.2 Data Understanding

Die zweite Phase des CRISP-DM Modells umfasst die Aneignung des benötigten Datenverständnisses. Dies impliziert in der ersten Instanz die Durchführung von Aktivitäten, die einen ersten Einblick in das Date gewähren. Dies umfasst sowohl visuelle als auch

deskriptive Eigenschaften der Daten. Dabei wird dem Projektteam der erste Eindruck der Daten gewährt, mit dem Ziel Herausforderungen der Datenqualität zu identifizieren und erste Hypothesen zu nicht ersichtlichen Informationen zu generieren. Gerade die Visualisierung ermöglicht bereits früh die Erkennung von Tendenzen hinsichtlich möglicher Ausreißer und Korrelationen (Chapman et al., 2000, S. 20-21).

2.5.3 Data Preparation

Nachdem das Verständnis und die explorative Phase der Daten abgeschlossen ist, werden die Daten hinsichtlich der Modellentwicklung aufbereitet. Von essenzieller Bedeutung ist es im Rahmen der dritten Phase, einen Überblick der Prozessdaten zu erhalten und eine zielführende Datenstruktur für die Modellierung zu selektieren, die im Sinne der Problemstellung ist. Hierfür ist die Auswahl der Daten von den Aufgaben und der Art des Modells abhängig. Ist das Ziel des Modells die Vorhersage von Immobilienpreisen, so ist es notwendig innerhalb der Data Preparation Phase unabhängige Variablen, die eine hohe Korrelation mit der Zielvariable aufweisen zu identifizieren. Zusammenfassend lässt sich die Datenaufbereitung als grundlegender Schritt für das maschinelle Lernen der Prozessdaten festhalten (Ge et al., 2017, S.20591-20592). Zusätzlich bedarf es die Bereinigung der Daten, indem beispielsweise Ausreißer, Fehlerwerte, inkonsistente oder fehlende Werte angepasst werden und in eine verwertbare Form des Lernalgorithmus überführt werden. Generell beansprucht die Data Preparation und Data Understanding Phase die meisten zeitlichen Ressourcen innerhalb der DM-Prozesse, weshalb es notwendig ist diese ausreichend genug im Voraus zu planen. Zusätzlich sind die Präparationsaktivitäten nicht oder nur zu einem Teil automatisierbar, weshalb aufwendige Extract-Transform-Load Prozesse ebenso mit eingeplant werden müssen (Chapman et al., 2000, S. 22-25).

2.5.4 Modeling

Die Modeling Phase ist geprägt durch die Anwendung diverser Modellierungstechniken. Dafür werden für die Problemstellung unterschiedliche Modelle, mit unterschiedlichen Parametereinstellungen getestet und entwickelt. In dieser Phase wird das Modell mit Hilfe der verwendeten Algorithmen anhand der vorab erarbeiteten und transformierten Daten trainiert (Chapman et al., 2000, S. 219-220).

2.5.5 Evaluation

Die vorletzte Phase des CRISP-DM Modells umfasst die Bewertung der Modelle und die Verwendbarkeit der Modellergebnisse. Dabei sind die Interpretation und Reflektion des Modells vom jeweilig genutzten Algorithmus abhängig. Dies impliziert die Bewertung der Modelle anhand von vorab definierten Leistungsmetriken, die eine Überprüfung der Zielerreichung ermöglichen. Sollten die Ergebnisse nicht der Zieldefinition entsprechen, so bedarf es der Modellanpassung und erneuten Evaluation des Modells (Chapman et al., 2000, S. 28-29). Unterstützt wird die Phase der Evaluation durch die Verwendung diverser XAI-Methoden, hinsichtlich der Zielstellung. Die Evaluation der XAI-Methoden erfolgt jedoch in einem gesonderten Kapitel, da dies den Schwerpunkt der Ausarbeitung darstellt.

2.5.6 Deployment

Zuletzt bezieht sich die Deployment Phase auf den möglichen produktiven Einsatz und die Verwendung des gewonnenen Wissens, welches aus den der Modellergebnisse resultiert. Dabei steht das gewonnene Wissen im Rahmen der Organisation, der Berichterstattung und der Präsentation im Mittelpunkt (Chapman et al., 2000, S. 219-220).

2.6 Definition von Analysefeldern und Erklärungseigenschaften zur Unterstützung der Untersuchung

Nach dem aktuellen Stand existieren keine einheitlich definierten Metriken, die eine Messung der Qualität der Erklärbarkeit von XAI-Methoden ermöglichen. Dies erschwert folglich die Evaluierung hinsichtlich der Erklärungsqualität. Hierfür werden nachfolgend auf Basis der Erkenntnisse von Gerlings et al. (2021), Sisk et al. (2022), Hall und Gill (2018), Mohseni et al. (2021), Rosenfeld et al. (2018) und Barredo et al. (2020) Analysefelder definiert, anhand deren die Erklärbarkeit von XAI-Methoden untersucht wird. Die Analysefelder, die innerhalb der Diskussion die Einordnung der Erklärbarkeit entlang der bestehenden wissenschaftlichen Literatur unterstützen, sind in der Tabelle 4 definiert und gelten für die lokalen und globalen Analysen.

Die Gewichtung oder auch Wichtigkeit von Variablen beschreibt den Beitrag, den die einzelnen Variablen zur Ergebnisfindung beitragen. Hierbei ermöglicht die Beitragshöhe einer Variable die hierarchische Einordnung, um eine Aussage darüber tätigen zu können, welchen Einfluss die Variable auf die Ergebnisfindung des Modells ausübt. Außerdem

ermöglicht die Darstellung der Beitragshöhe die Messung des Einflusses der Variable. Innerhalb dieser Ausarbeitung kann die Beitragshöhe sowohl positive als auch negative numerische Werte annehmen (Hall & Gill, 2018, S.32.; Rosenfeld, 2021, S.48). Das Analysefeld der Modell Optimierung und des Debuggings, soll im Zuge der Erklärbarkeit potenzielle Verzerrungen in den Daten aufweisen oder dem Menschen anhand der Erklärbarkeit potenziale erkennen lassen, deren Erkenntnisse beispielsweise nochmals im Feature Engineering umgesetzt werden kann. Die Funktionsweise des Modells soll durch die Erklärbarkeit der Verfahren transparent dargestellt werden, die einen Rückschluss auf mögliche Debugging- oder Optimierungsmaßnahmen erlauben (Mohseni et al., 2021; Sisk et al., 2022). Weiterführend ist anhand der Wichtigkeit der Variablen die Untersuchung der Erklärbarkeit hinsichtlich der Fairness, ethischen Korrektheit und Risikokontrolle möglich. Durch die Identifikation von Variablen mit hoher Einflussnahme, lassen sich diese auf die zuvor genannten Eigenschaften untersuchen. Dabei können nicht nur Modelle hinsichtlich der moralischen Verpflichtung angepasst werden sondern ebenso entlang der Unternehmens Compliance und des gewährten rechtlichen Rahmens (Barredo Arrieta et al., 2020, S. 5 ff.; Gerlings et al., 2021, S. 1287-1288). Abschließend kumuliert das Analysefeld der Transparenz die gesamtheitlichen Ergebnisse der Anwendung von XAI-Verfahren, im Sinne einer transparenten Entscheidungsfindung. Dies kann als Zersetzbarkeit des Modells verstanden werden, indem deutlich wird welche Variablen, mit welchen Werten und mit welchen Hyperparametereinstellungen zu der Ergebnisfindung führt. Hierbei ist zu beachten, dass eine mathematische Nachvollziehbarkeit eine algorithmische Transparenz inkludiert, die für ein lineares Regressionsmodell eher gegeben ist als für eine künstliches neuronales Netz (Barredo Arrieta et al., 2020, S.10-11; Rosenfeld, 2021, S. 47-48). Die definierten Analysefelder dienen nicht der Messung und Bewertung der Erklärbarkeitsqualität von ML-Modellen. Vielmehr sollen sie dazu dienen, zu untersuchen, in welchen Bereichen der praktische Einsatz von XAI-Methoden die Erklärbarkeit von ML-Modellen gewährleisten kann.

Tabelle 4: Analysefelder der Erklärbarkeit nach XAI

Analysefeld	Beschreibung
Gewichtung / Wichtigkeit von Variablen	Die lokalen und globalen Erklärungen schlüsseln die Gewichtung von Variablen auf und weisen dabei auf welche Variablen zu der Ergebnisfindung beigetragen haben
Modell Optimierung und Debugging	Die lokalen und globalen Erklärungen ermöglichen es Optimierungen des Modells vorzunehmen und Unerwartete Ergebnisse zu identifizieren
Modell Fairness, Ethische Korrektheit und Risikokontrolle	Ethisch unkorrekte Variablen lassen sich durch lokale und globale Erklärungen ausfindig machen, genauso wie kritische Beiträge von Variablen, die eine Gefahr darstellen können
Transparenz	Lokale und globale Erklärungen legen transparent dar, wie das Modell zur Ergebnisfindung kommt

Da wie bereits konstatiert keine einheitlichen Metriken zur Messung der Erklärbarkeit definiert sind, werden zusätzlich die XAI-Methoden hinsichtlich der Eigenschaften der Stabilität, Trennbarkeit, Konsistenz und Erklärungsähnlichkeit auf Basis der Ergebnisse des Kapitels 5 untersucht. Dies soll eine Gegenüberstellung der eingesetzten XAI-Methoden ermöglichen und in Kombination mit den Analysefeldern als Grundlage zur Gesamtheitlichen Evaluation dienen. Die definierten Eigenschaften beruhen auf den Erkenntnissen diverser Autoren und sind der Tabelle 5 zu entnehmen.

Die Stabilität beschreibt dabei den wiederholten Einsatz der XAI-Verfahren für denselben Datensatz oder Datenpunkt. Hierbei ist eine Stabilität gewährleistet, sobald beispielsweise für das gleiche Modell und denselben Datenpunkt unter Anwendung des gleichen XAI-Verfahrens die Erklärbarkeit der Ergebnisfindung keine unterschiedlichen Ergebnisse hinsichtlich der Gewichtung oder Beitragshöhe produziert (Molnar, 2019, S.47; Molnar et al., 2022, S.27; Sokol & Flach, 2020, S.62). Die Trennbarkeit zeichnet sich dadurch aus, dass unterschiedliche Datenpunkte, unter Annahme des gleichen Modells und XAI-Verfahrens, zu unterschiedlichen Ergebnissen führen. Sollten gravierende Unterschiede beobachtet werden, die anhand einer minimalen Änderung der Daten oder minimal unterschiedliche Dateninstanzen in Erscheinung tritt, so ist die Erklärungseigenschaft der Konsistenz verletzt. Demnach werden nach dem Konsistenzkriterium ähnliche Erklärungen für stark ähnliche Dateninstanzen berechnet (Molnar, 2019, S.26-27; Sokol & Flach, 2020, S.62). Schlussendlich definiert die Erklärungsähnlichkeit, dass für

dieselbe Aufgabe unterschiedliche Modelle unter Anwendung der gleichen Daten ähnliche Erklärungen generiert werden. Gerade unter dem Aspekt des Rashomon-Effektes, wird der Erklärungsähnlichkeit ein besonderer Stellenwert zugerechnet und in Kapitel 7 diskutiert. Der Rashomon-Effekt bezieht sich auf die Situation, in der mehrere Modelle, die mit den gleichen Daten trainiert werden, gleichgute Vorhersageergebnisse liefern. Dies zeigt die Möglichkeit mehrerer guter Modelle an und wird als Multiplizität von guten Modellen beschrieben. Gleichzeitig kann es vorkommen, dass gleichgute Modelle unterschiedliche Erklärungen für ihre Vorhersage liefern, was eine Unsicherheit bei der Interpretation der Vorhersageergebnisse und Vorhersageerklärungen hervorrufen kann. So mit kann ebenso eine Multiplizität von guten Erklärungen erfolgen. Unter beiden Aspekten wird die Erklärungsähnlichkeit näher untersucht (Molnar et al., 2022, S.27; Semenova et al., 2022,S.1-2; Sokol & Flach, 2020, S.62). Die Einordnung der Ergebnisse im Bezug zur Stabilität, Trennbarkeit, Konsistenz findet nur in der lokalen Betrachtung statt. Die Erklärungsähnlichkeit wird sowohl global und lokal beschrieben. Zusätzlich wird die Berechnungszeit der einzelnen XAI-Modelle je nach Algorithmus gemessen und im Kapitel 5 erfasst.

Tabelle 5: Erklärungseigenschaften der XAI-Methoden

Eigenschaft	Beschreibung
Stabilität	Für den gleichen Datenpunkt wird stets die gleiche Erklärung generiert
Trennbarkeit	Unterschiedliche Datenpunkte führen zu unterschiedlichen Erklärungen
Konsistenz	Ähnliche Erklärungen werden für minimal unterschiedliche Dateninstanzen generiert
Erklärungsähnlichkeit	Ähnliche Erklärungen mithilfe unterschiedlicher XAI-Verfahren
Berechnungszeit	Dauer der Berechnungszeit der XAI-Methoden

3 Einordnung der aktuellen wissenschaftlichen Erkenntnisse zum Immobilienmarkt

3.1 Preisentwicklung der Immobilien

Die seit Anbeginn der 90er Jahre weltweit stark steigenden Immobilienpreise entwickeln den Immobilienmarkt zu einem der bedeutendsten Wirtschaftsbereiche. Dabei ist die in diversen Beiträgen seit Jahrzehnten bestehende Diskussion, ob die steigenden Immobilienpreise in einer platzenden Spekulationsblase ihr Ende finden oder weiter als Wertanlage bestehen bleiben, bis heute relevant (Brzezicka, 2021, S.335-336; Hott & Monnin, 2008, S.228). In den USA war die Immobilienbranche im Jahr 2021 für fast 4 Milliarden Dollar Umsatz verantwortlich, was etwa 17 % des Bruttoinlandsprodukts ausmacht. So konnten circa 6,8 Millionen Immobilien verkauft werden, davon 6,1 Millionen Bestehende und die restlichen 700.000 Neubauten. Laut dem amerikanischen Immobilien-Händlersverband *National Association of Realtors* war das Jahr 2021 für Bestandsimmobilien das erfolgreichste Verkaufsjahr seit dem Immobilienboom 2007 und 2006 (National Association of Realtors, 2021, S.1; National Association of Realtors, 2022, S. 8-9). Mikroökonomische Herausforderungen stellen jedoch eine kontinuierliche Gefahr für alle wirtschaftlichen Bereiche dar und können in makroökonomische Herausforderungen resultieren. Die Anfänge der Finanzkrise im Jahr 2008 entstand zum Teil aus dem wahllosen Gewähren und nicht bedienen von Immobilienkrediten in den USA. Dies hinterließ deutliche Spuren auf der gesamten Welt, indem diverse wirtschaftliche Bereiche kollabierten, Menschen folglich ihren Job verloren und letztlich die Immobilienpreise stark sanken. Der Abfall von durchschnittlich 30 % der Immobilienpreise konnte vorerst wieder im Jahr 2012 auf den Stand von vor 2007 erreicht werden. Dabei unterschied sich die prozentuale Preisminderung von Land zu Land und in den USA von Bundesstaat zu Bundesstaat (Malpezzi, 2017, S.6). Der Preisabfall ging einher mit einer Erhöhung der Zinsen für Immobilienkredite, welches die Chancen für den Erwerb und Verkauf von Häusern und Wohnungen erneut erschwerte (Lin, J. Y., Treichel, 2012, S.40). Durch die neue Zinspolitik der Federal Reserve System (FED) erholten sich die Vereinigten Staaten wirtschaftlich von der Krise. Dies inkludierte die Senkung der Arbeitslosenquote. Dabei trugen die niedrigen Zinsen wesentlich dazu bei, dass die Nachfrage an Immobilien und die Preise simultan zu der Senkung der Zinsen stiegen (Belke & Keil, 2018, S.28).

Anhand der Abbildung 7 lässt sich festhalten, dass der mittelwertige Preis für verkauft Häuser in den USA kontinuierlich seit 1970 ansteigt. Demnach ist für das dritte Quartal 2022 ein mittlerer Verkaufspreis von 468.000 \$ zu verzeichnen, wodurch dies den höchsten mittleren Verkaufspreis jemals darstellt. Für das nachfolgende Quartal ist ein Rückgang des durchschnittlichen Verkaufspreises zu erkennen. Der Verkaufspreis für das vierte Quartal 2022 liegt bei 467.700 \$ (FRED, 2022d, S.1).

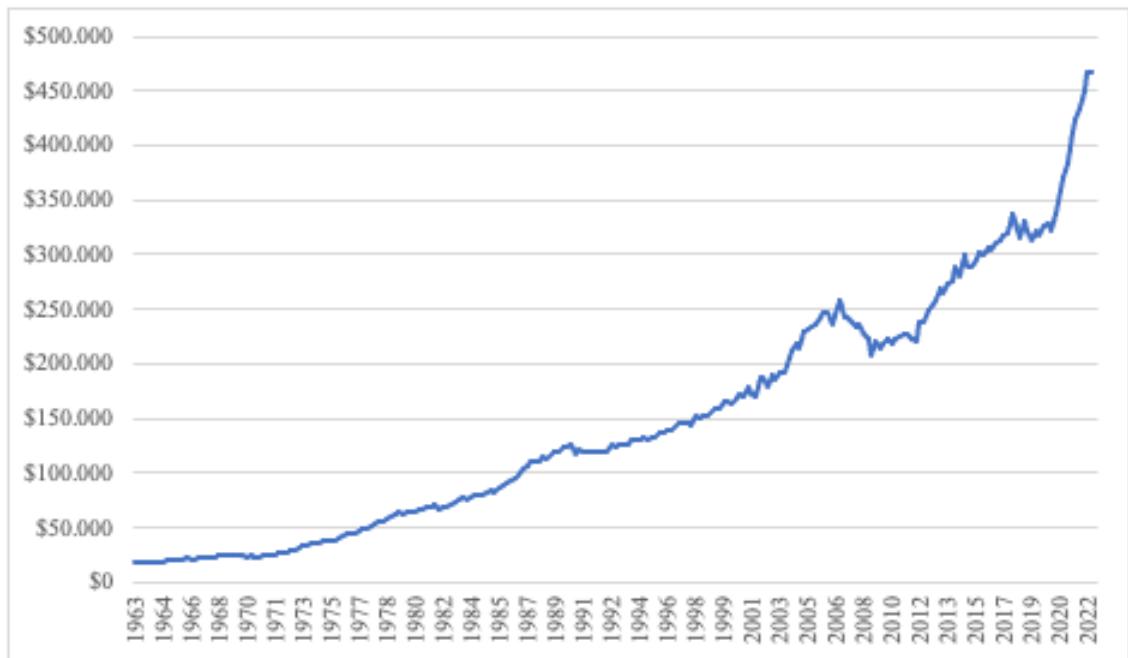


Abbildung 7: Durchschnittlicher Verkaufspreis von Häusern in den USA (FRED, 2022d, S.1)

Diese Entwicklung ist ebenso in Abbildung 8 für die Region Mitte-West der USA zu beobachten, in der Iowa, inklusive der Stadt Ames, liegt. Der Mittelwert des Verkaufspreises für das dritte Quartal 2022 beträgt in dieser Region 409.900 \$. Das ist eine Differenz in Höhe von circa 58.000 \$ zum Mittelwert des gesamten Landes. Vergleicht man die Differenz vom zweiten Quartal 2022 der Landes- und Regionsebene, so belief sich der Unterschied auf lediglich 18.00 \$. Dabei deutet sich ein leichter Rückgang des Verkaufspreises an, der sich im vierten Quartal fortführt. So beträgt der mittelwertige Preis 356.600 \$ und verzeichnet im Vergleich zum dritten Quartal einen Rückgang von 53.300 \$ (FRED, 2022c, S.1).

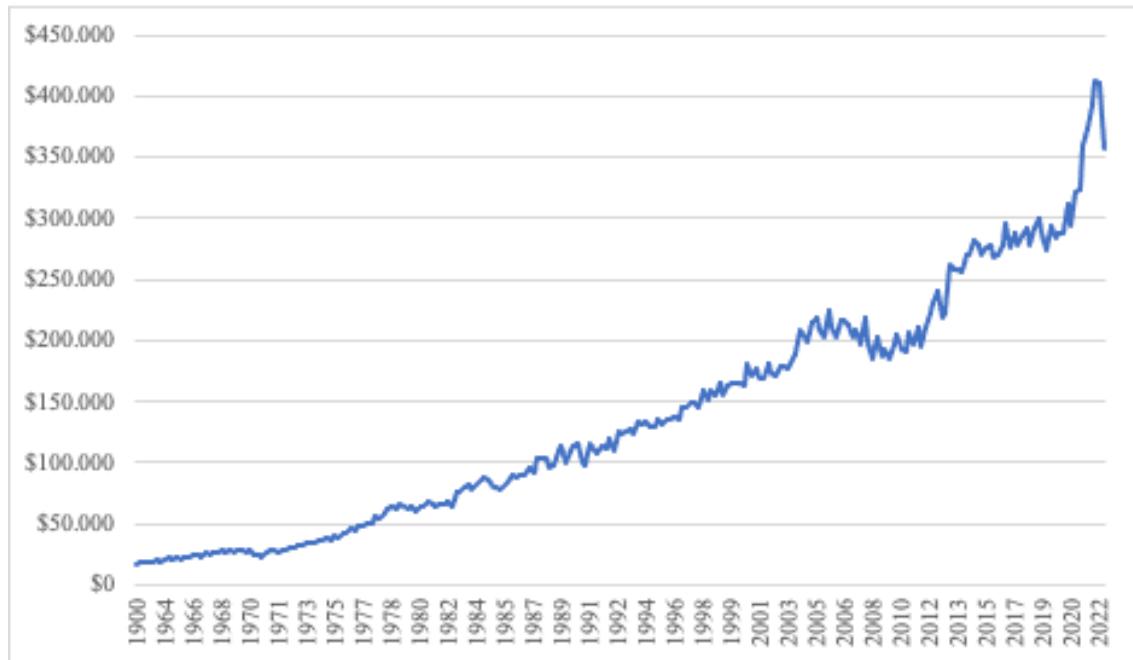


Abbildung 8: Durchschnittlicher Verkaufspreis der Region Mitte-West in den USA (FRED, 2022c, S.1)

Auf der Ebene des Bundesstaates, beträgt der durchschnittliche Verkaufspreis im Staat Iowa im vierten Quartal 2022 396.670 \$. In Bezug auf die Stadt Ames lässt sich dabei ein durchschnittlicher Verkaufspreis von 274.160 \$ Dollar festhalten. Dieser liegt deutlich unter dem Durchschnitt der Landes- und Bundesebene. Für Iowa und Ames lässt sich eine minimale Minderung des durchschnittlichen Verkaufspreises erkennen, die prozentual gesehen jedoch niedriger ist als die Höhe der Reduktion der Landes- und Regionsebene. Die kumulierte Preisentwicklung für die Landes- Regional- Bundesstaats- und Stadtebene ist der Abbildung 9 zu entnehmen. Die Immobilienpreise für Iowa sind erst ab 1973 verfügbar und für Ames ab 1986. In der Grafik ist deutlich zu erkennen, dass für alle Ebenen ein Anstieg des Preises zu beobachten ist. Lediglich der Bundesstaat Iowa und die Stadt Ames verzeichnen im Zeitraum der Finanzkrise keinen Rückgang des Preises, sondern eine Seitwärtsbewegung. Ebenso wird deutlich, dass die Immobilien in Ames einen niedrigeren durchschnittlichen Preis aufweisen als die der übrigen Beobachtungen. Nichtsdestotrotz ist der Anstieg des Verkaufspreises nach 2020 innerhalb der COVID-19 Pandemie hervorzuheben für alle Bereiche (FRED, 2022b, 2022a, 2002c, 2022d S.1).

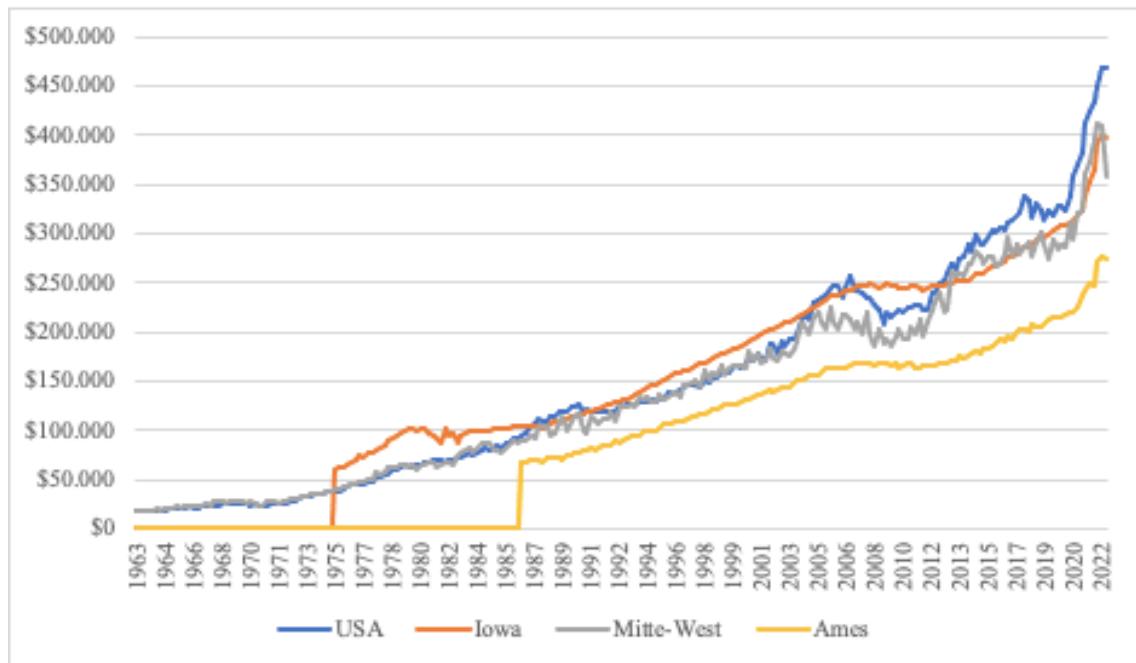


Abbildung 9: Durchschnittlicher Verkaufspreisentwicklung auf Landes- Bundesstaats- und Stadtbene (FRED, 2022a, 2022b, 2022c)

Hierbei stellt sich die Frage, welche Faktoren, neben der bereits benannten Finanzkrise, die Preisentwicklung maßgeblich beeinflusst haben. Demnach wurde die Literatur gleichermassen auf die Benennung sowie die Untersuchung von einflussreichen Faktoren analysiert, die eine Auswirkung auf die Preisentwicklung und den Immobilienpreis haben. Dies soll im Rahmen der Ergebnisdiskussion die Identifizierung von Unterscheidungs- und Überschneidungspunkten zu den Ergebnissen der XAI-Methode ermöglichen. Eine Bewertung auf Basis der Literatur und der Untersuchungsergebnisse ist folglich nicht möglich, da die verwendeten Methoden, Datengrundlagen sowie unterschiedlichen Faktoren sich als stark unterschiedlich und komplex darstellen. Eine vorgenommene Bewertung hätte höchstwahrscheinlich die Formulierung von inkorrekten Schlüssen und Zusammenhängen zur Folge. Da dies nicht der Zielsetzung der Ausarbeitung entspricht, wird lediglich ein wertfreier Abgleich der Erkenntnisse angestrebt. Im Bezug zum Immobilienmarkt der Vereinigten Staaten bleibt diesbezüglich die qualitative Forschungsliteratur begrenzt. So kommen Jafari und Akhavian (2019) mittels eines hedonischen Modells in Verbindung mit einer Korrelationsanalyse zu dem Ergebnis, dass die Quadratfußgröße der Immobilie, das Bau- und Modernisierungsjahr, die Anzahl an Zimmern und die Lage als stark korrelierend in Erscheinung treten. Hierbei werden seitens der Autoren nur

Untersuchungen anhand der Eigenschaften der Immobilie unternommen. Die stärkste positive Korrelation wird bei der Erhöhung der Zimmeranzahl beobachtet. Als Datengrundlage dient hierbei die *American Housing Survey* aus dem Jahr 2016 verwendet und bildet die jährliche Befragung von Immobilieneigentümern in den gesamten Vereinigten Staaten ab (Jafari & Akhavian, 2019, S. 527-528). Am Beispiel des vielverwendeten Boston Housing Datensatzes untersuchten Lee et al. (2021) den Einfluss der unabhängigen Variablen auf den Verkaufspreis mittels einer PDP-Analyse. Dabei sind die wesentlichen Erkenntnisse, dass mit zunehmender Fläche und Zimmeranzahl der Preis im Vergleich zu den übrigen Variablen stärker ansteigt. Darüber hinaus kann beobachtet werden, dass mit zunehmender Luftverschmutzung der Verkaufspreis der Immobilie fallen kann (Zhao & Hastie, 2021, S.275). Aktuelle Untersuchungen seitens Miller und Pinter (2022) zeigen auf, dass die Immobilienpreise in den Vereinigten Staaten an Orten mit einer höheren Wahrscheinlichkeit an Überflutung über die letzten Jahre stagniert sind. Perspektivisch ist sogar eine Preisminderung zu erwarten(R. G. Miller & Pinter, 2022, S. 9-11). Auf dem internationalen Immobilienmarkt sind die Ergebnisse der Untersuchungen ähnlich zu den, die in den Vereinigten Staaten beobachtet werden. Li et al. (2021) sind mittels eines hedonischen Preismodells zu dem Ergebnis gekommen, dass sowohl bauliche Strukturen als auch die Lage der Immobilie einen Einfluss auf deren Preise ausüben. Außerdem beobachten die Autoren eine hohe Korrelation zwischen dem Immobilienpreis und der Entfernung zu verschiedenen Infrastrukturen wie Verkehrsanbindungen oder öffentlichen Einrichtungen (S. Li et al., 2021, S. 10-12). Ebenfalls anhand eines hedonischen Preismodells untersuchten Cui et al. (2018) 16475 Immobilien in China auf die wesentlichen Treiber des Immobilienpreises. Sie kommen zu dem Ergebnis, dass die Zimmeranzahl und die Entfernung zu öffentlichen Einrichtungen und Verkehrsmitteln als signifikant für die Preisentwicklung zu verzeichnen sind. Ebenso wird das Einkommen, das verfügbare Einkommen für einen Kredit und die Ausgaben als stark korrelierend seitens Wie et al. (2022) eingeordnet. 127 Großstädte in Deutschland wurden von Belke und Keil (2018) auf die Determinanten von Immobilienpreisen untersucht und sie kamen dabei zu dem Ergebnis, dass für die die Anzahl an möglichen Haushalten, die Nähe zur regionalen Infrastruktur und die demographische Struktur maßgeblich für den Immobilienpreis entscheidet sind. Die Ergebnisse von Sandbhor und Chaphalkar (2016) stützen die vorherigen Erkenntnisse zu den internen Faktoren, indem die Autoren die Flächengröße und das

Baujahr als stark einflussreich identifizierten. Erstmalig wird hierbei jedoch auch die Qualität der Immobilie benannt, gefolgt von der Nachbarschaft und der wirtschaftlichen Stärke der Umgebung (Sandbhor & Chaphalkar, 2016, S.3).

Weiterführend ist es für das vollständige Verständnis der Immobilienpreisentwicklung wichtig weitere Faktoren zu verdeutlichen, die im genutzten Datensatz nicht vertreten sind, jedoch Daten darstellen, die in perspektivischen Ausarbeitungen den Mehrwert von XAI stärken könnten. Dabei stellen die nachfolgenden externen Faktoren in der Realität weitere mögliche Einflussfaktoren dar, die für die Preisbestimmung von Immobilien nach der Literatur als relevant eingestuft sind. Ebenso inkludieren die externen Faktoren die bereits genannte Zinshöhe für die Finanzierung des Immobiliendarlehns. In der Abbildung 10 wird zunächst das mittlere Haushaltseinkommen in den USA von 1990 bis 2021 dargestellt. Der Einsatz des verdienten Geldes ist dabei notwendig, um den Erwerb von Immobilien zu tätigen. Auffällig ist dabei, dass das mittlere Einkommen sich von 1990 zu 2021 mehr als verdoppelt hat. Im Vergleich zum Immobilienpreis, der sich von 1990 bis 2021 nahezu verfünfacht hat, erscheint die Erhöhung des Medians des Haushaltseinkommen minimal. Lediglich um das Jahr 2008 stagnierte und reduziert sich das mittlere Haushalteinkommen, welches als ein Ergebnis der Finanzkrise zu interpretieren ist (Grimes et al., 2021, S.10).

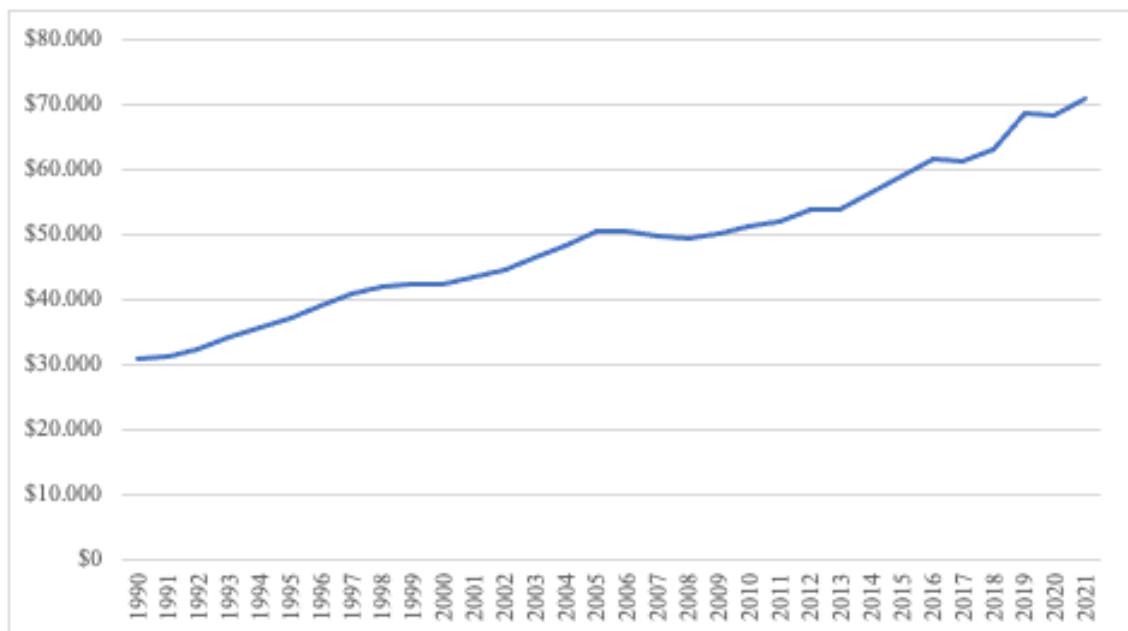


Abbildung 10: Durschnittliches Einkommen in den USA in \$ von 1990 bis 2021 (US Census Bureau, 2022, S.1)

Darüber hinaus existieren etliche Beiträge, die einen Zusammenhang zwischen wirtschaftlichem Wachstum und Immobilienpreisen sehen. Diese Preise steigen demnach nur bei positivem wirtschaftlichem Wachstum. Dabei spielt der globale Aktienmarkt ebenso in Kombination mit dem Immobilienmarkt eine Rolle, um den wirtschaftlichen Wachstum und die Kaufbereitschaften ableiten zu können (Nguyen & Bui, 2019, S.299). Aktuell setzen sich diverse Beiträge damit auseinander, welchen Einfluss der Klimawandel auf die Wirtschaft haben kann. Demnach kann in den USA seitens Baldauf et al. (2020) für die Preisentwicklung der Immobilien festgestellt werden, dass der Verkauf von Immobilien bis zu 7 % teurer ist, falls die Immobilie in einer Gegend mit einer höheren Anzahl an Leugnern des Klimawandels liegt. Das die tatsächlichen wetterbedingten Ereignisse und dynamischen Veränderungen des Klimawandels die zukünftige Preisentwicklung nicht beeinflussen wird, konnte seitens diverser Autoren bisher nicht eindeutig widerlegt werden (Baldauf et al., 2020, S.1290-1291; Giglio et al., 2021, S.3567).

Aktuell versuchen diverse Berichte die Auswirkungen der COVID-19-Pandemie für den Immobilienmarkt zu durchleuchten. Dabei kann kurzfristig festgehalten werden, dass die Preisentwicklung von 2020 bis 2022 vorerst keine Minderung erfahren hat. Lediglich ist laut Milcheva (2022) das Bewusstsein der Immobilienunternehmen für eine mögliche negative Preisentwicklung geschärft, während das Handeln rationaler und bedachter wird (Milcheva, 2022, S.318-319). Langfristige Schätzungen zu Auswirkungen der COVID-19-Pandemie auf die Preisentwicklung sind derzeit begrenzt. Abzuwarten sind ebenso die Folgen der Leitzinserhöhung seitens der FED. Erste Tendenzen zeigen, dass die Immobilienpreise in den USA und Europa zu stagnieren scheinen und dass der Verkauf und Ankauf von Immobilien zurückgegangen ist (Sorkin et al., 2022).

Da die Betrachtung aller möglichen Einflussfaktoren zur Vorhersage des Immobilienpreises den Rahmen der Ausarbeitung überschreiten wird, ein derartiger Datensatz nicht existiert und die Schaffung solch eines Datensatzes ein fortlaufender Prozess ist, beschränkt sich der genutzte Datensatz auf qualitative und quantitative Eigenschaften von Immobilien aus der Stadt Ames im Bundesstaat Iowa. Die Zusammenfassung der Ergebnisse des Kapitels 3.1 sind der nachfolgenden Abbildung 11 zu entnehmen.

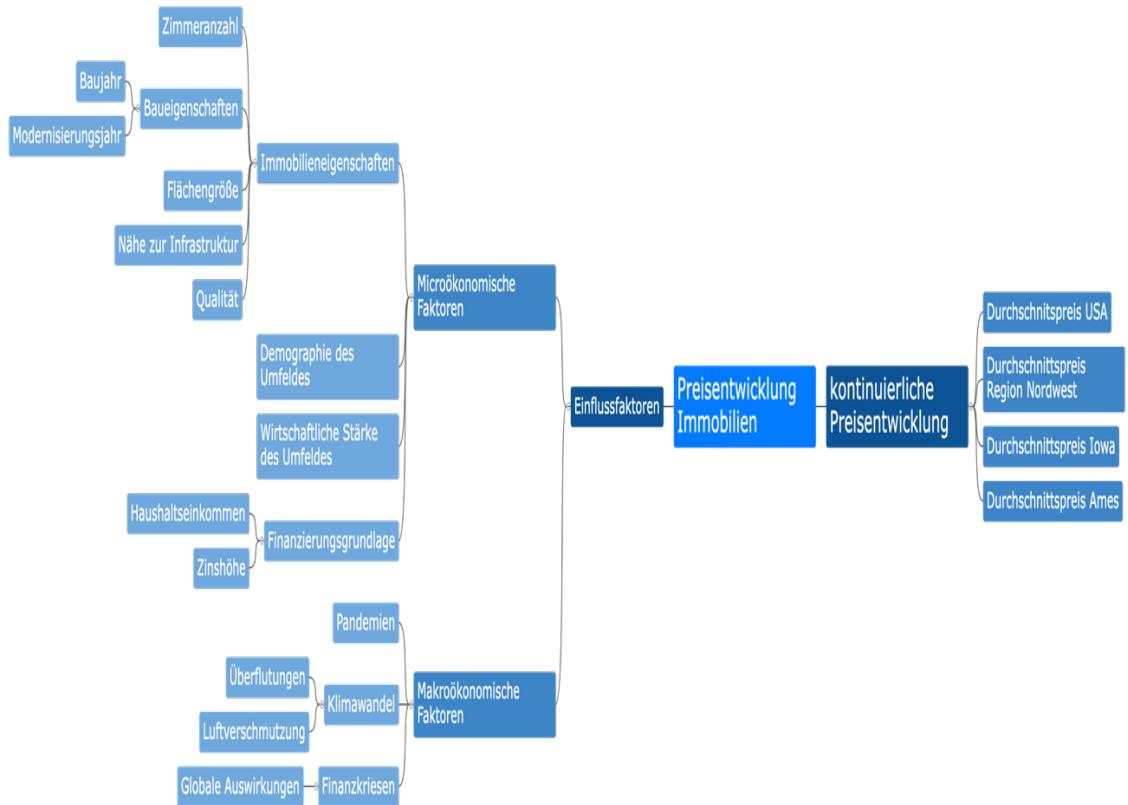


Abbildung 11: Zusammenfassung der Ergebnisse des Kapitels 3.1

3.2 State of the Art der Immobilienprognosen

Neben der Betrachtung einer Auswahl an möglichen Einflussfaktoren, die zur Entwicklung von Immobilienpreisen beigetragen haben, sind Vorhersagen von zukünftigen Preisentwicklungen bis heute hochgradig komplex. Demnach werden beispielsweise in einem hochgepriesenen Beitrag von 1989 die Annahmen getroffen, dass Immobilienpreise aufgrund des demographischen Wandels ab 2007 stagnieren oder sinken werden (Greenwood et al., 2005, S.22-23; Kabaivanov & Markovska, 2021, S. 1-2). Diese Vermutung erweist sich als begrenzt bewiesen, da die resultierende Finanzkrise ein wesentlicher Faktor für den kurzzeitigen Rückgang der Preise ist. Dabei ist die Verwendung von ML zur Prognose von Immobilienpreisen bereits weltweit bei den unterschiedlichen Aspekten gefragt. Hierbei bilden jedoch hedonische Modelle Jahrzehnte lang das bewährte Verfahren der empirischen Bestimmung von Preisen und Mieten von Immobilien. Immobilieneigenschaften bilden dafür die Ausgangslage für die hedonische Preisbestimmung (Alzain et al., 2022, S.2; Lorenz et al., 2022, S.2-3). Der hedonische Ansatz fand seinen ersten Einsatz im Jahr 1939 durch Court (1939) zur Bestimmung von Autopreisen.

Während hedonische Preismodelle in diversen Anwendungsbereichen bereits eingesetzt werden, beschreibt MacLennan (1977) wie der hedonische Ansatz sich im Rahmen der Immobilienbranche als gängige Methode zur Bestimmung von Immobilienpreisen bewährt hat. Es wird jedoch bereits die erste Kritik geäußert, indem beschrieben wird, dass hedonische Preisbestimmungen für Immobilien auf lokaler Ebene Anwendung finden sollen und nicht einen gesamten Markt abbilden können. Hierbei wird die Preisbestimmung zu stark durch die Lage und die Nachfrage bestimmt und umfasst demnach keine weiteren Variablen (MacLennan, 1977, S.64-65).

Nachfolgend werden die ersten Gruppierungen untersucht, die einen Einfluss auf die Preisbestimmungen haben können. Diese wurden in Strukturelle Faktoren sowie Lage- und Nachbarschaftsfaktoren unterteilt. Strukturelle Faktoren definieren dabei Bau- und Maßeigenschaften der Immobilie und Lagefaktoren die geographische Lage, beispielsweise die damit verbundene Entfernung zu anderen Infrastrukturen. Letztlich bilden Nachbarschaftsfaktoren sozioökonomische Eigenschaften ab wie das Einkommen oder die Kriminalitätsrate. Diese Gruppierungen finden bis heute im Rahmen der hedonischen Modelle zur Ermittlung von Immobilienpreisen wiederkehrend ihre Erwähnung und Bedeutung zugesprochen (Can, 1992, S.471; des Rosiers et al., 2011, S.524-525; Haider & Miller, 2000, S.2-3; Lorenz et al., 2022, S.22-23; Stamou et al., 2017, S.280). Dabei liegt ein Forschungsschwerpunkt der letzten Jahre auf den Effekt, den die einzelnen Faktoren aus den drei Gruppierungen auf die Preisvorhersage von Immobilie haben. Hierbei werden einzelne Faktoren isoliert betrachtet, wie beispielsweise der Standort einer Immobilie am Wasser, die Entfernung zu Wäldern oder die Luftverschmutzung den Preis beeinflussen (Conway et al., 2010, S. 150-151; Fernández-Avilés et al., 2012, S.243-244; Łaszkiewicz et al., 2022, S.1 ff.; Rouwendal et al., 2017, S. 829-831). Weitere Forschungsschwerpunkte betreffen die Entfernung von Windrädern, Elektrowerken, infrastrukturelle Besonderheiten wie Autobahnen und der Ausbau des Schienenverkehrs (Chin et al., 2020 S. 887-888; Li, 2020, S. 557-559; Wyman & Mothorpe, 2018, S.121-126). Zusätzlich wird im Rahmen der Immobilienpreisbestimmung ein Zusammenhang zwischen dem Zugang zu Bildungseinrichtungen und Nachbarschaftsstrukturen erforscht (Goodwin et al., 2020, S.321-325; Theisen & Emblem, 2018, S.239). Mit voranschreitender Zeit wird jedoch die Begrenztheit der hedonischen Ansätze bezüglich prädiktiver Aufgaben deutlich, die mit zunehmendem Einsatz und Etablierung von KI an Bedeutung

gewinnt (Pérez-Rave et al., 2019, S.91-93). Besonders erfolgreich zeigt sich dabei die Nutzung von ML zur Bestimmung und Vorhersage von Immobilienpreisen im Vergleich zu konventionellen Ansätzen. Verschiedene ML-Algorithmen werden bereits für unterschiedliche Immobilitytypen und Lagen bezüglich ihrer Vorhersagequalität untersucht. Außerdem werden ML-Methoden genutzt, um Trends in den Immobilienpreisen zu erkennen, die beeinflussenden Eigenschaften der Immobilie zu identifizieren und konventionelle Preisbestimmungen mit den Ergebnissen der ML-Modelle zu vergleichen. (Banerjee & Dutta, 2018, S. 2998-3000; Phan, 2019, S.35ff; Satish et al., 2019, S.717ff). Im Rahmen von Untersuchungen werden bereits mithilfe von ML-Ansätzen die Auswirkungen sozialer, wirtschaftlicher und materieller Metriken auf das Angebot und die Nachfrage nach Immobilienpreisen erforscht. Die Untersuchungen mittels ML-Ansätze beziehen auch externe Faktoren ein, wie zum Beispiel die Auswirkungen der COVID-19-Pandemie auf den Immobilienmarkt im Zuge der Prognosen und Preisbestimmung (Alzain et al., 2022, S.4-5).

Hierbei finden ebenso Vergleiche von hedonischen- und ML-Ansätzen statt, die gegenübergestellt und bewertet werden (Čeh et al., 2018, S.1ff). Hochaktuell wird nicht nur der Vergleich beider Ansätze untersucht, sondern ebenso die Synergieeffekte, die die Kombination beider Methoden mit sich bringen können und wie *Big Data* sich mit dem hedonischen Konzept mittels ML vereinen lässt (Potrawa & Tetereva, 2022, S.50 ff.; Wei et al., 2022, S.1 ff.; Zaki et al., 2022, S.1 ff). Dabei zeigt sich, dass in den letzten Jahren vermehrt auf Algorithmen gesetzt wird, die eine sehr gute Performance bei der Prognose erzielen. Dies inkludiert den Einsatz von artifical neuronal networks, Support Vector Machines, Boosting, multi-layered perceptron, Light Gradient Boosting Machine und Stacked Generalization. Dabei fokussierten sich die diversen Untersuchungen hauptsächlich darauf, die Genauigkeit des ML-Modells sukzessiv zu verbessern (Alfaro-Navarro et al., 2020, S.1ff; Alzain et al., 2022, S.1ff; Bin et al., 2017, S.209ff; Pai & Wang, 2020, S.1ff; Pinter et al., 2020, S.1ff; Quang et al., 2020, S.433ff; Rampini & Re Cecconi, 2021, S.588ff; Zhou, 2020, S.1ff). Gupta et al. (2022) untersuchen in diesem Rahmen eine Synchronisation der Preisvorhersagen für Immobilien in den USA und Nutzen dafür verschiedene ML-Modell. Die hochkomplexe Dynamik, welche die Preisvorhersagen durch die Berücksichtigung von internationalen, nationalen und regionalen Faktoren annehmen kann, wird dabei näher untersucht und erläutert. Jedoch wird seitens

unterschiedlicher Autoren, vor allem mit der Zunahme der Anzahl an Faktoren, das Potenzial der Interpretierbarkeit der Vorhersagen von Immobilienpreisen diskutiert. Dabei wird beobachtet, dass wie zuvor beschrieben, komplexere Algorithmen für eine bessere Performance der Vorhersage genutzt werden, diesbezüglich jedoch die Verständlichkeit und Interpretierbarkeit der Ergebnisfindung gemindert wird (Phan, 2019, S.40; Viriato, 2019, S.52). Hierbei beziehen sich einige wenige Untersuchungen auf die Nutzung von Whitebox-Modellen, die Nutzung von Teilstufen von XAI-Methoden oder die Vorteile der Nutzung von XAI für die Vorhersageprognose von Immobilienpreisen. Darüber hinaus wird seitens der Autoren ein hoher Forschungsbedarf geäußert, dem die Ausarbeitung nachfolgend nachkommen wird (Cajias, 2021; Dimopoulos & Bakas, 2019; Lorenz et al., 2021; Potrawa & Tetereva, 2022). Auch hier fasst die Abbildung 12 die wesentlichen Erkenntnisse des Kapitels zusammen.

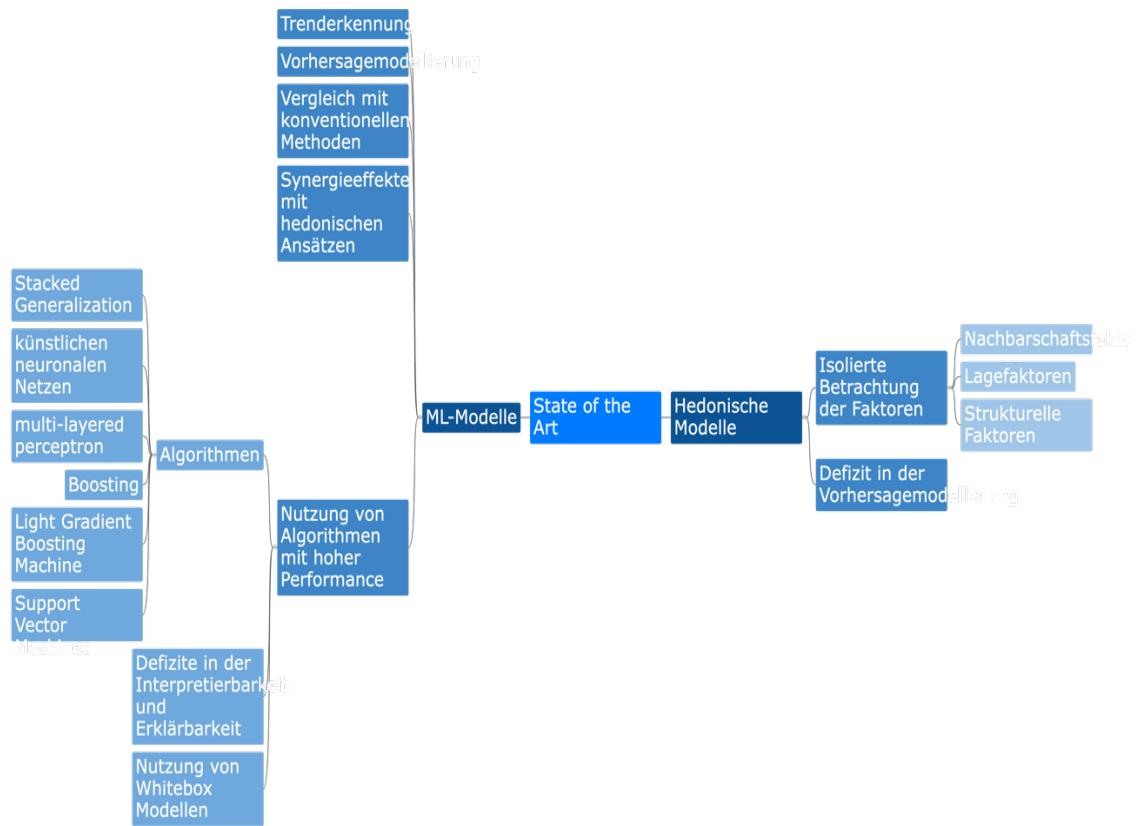


Abbildung 12: Zusammenfassung der Ergebnisse des Kapitels 3.2

4 Entwicklung eines Prognosemodells

4.1 Data Understanding

Als Programmierumgebung wird DataSpell von JetBrains genutzt, welche hauptsächlich für Data Science Aktivitäten entwickelt wurde (JetBrains, 2022). Dabei wird lediglich eine Datenquelle genutzt. Der verwendete Datensatz mit dem Namen *Housing prices Ames, Iowa* wird seitens OpenIntro zur Verfügung gestellt. Die Webseite stellt eine Sammlung von Datensätzen und Weiterbildungsunterlagen im statistischen Kontext zur Verfügung. Dabei kann der Datensatz nicht über die eigentliche Original-Website amstat.org abgerufen werden. Die Daten sind von Dean De Cock (2011) im Jahre 2011 erhoben worden und sollen den Einsatz von ML-Algorithmen anhand von Echtdaten ermöglichen, die nachfolgend empirische Analysen über den Immobilienmarkt der Stadt Ames zulassen (De Cock, 2011, S.1-2). Nennenswert ist hier noch die Information, dass metrische Variablen nach dem Einheitsmaß der Vereinigten Staaten dargestellt sind. So wird beispielsweise die Fläche in dieser Arbeit nachfolgend nicht im europäischen Einheitsmaß des Quadratmeters aufgeführt, sondern als Quadratfuß. Der Datensatz kann als Datei vom Format Comma-separated Values (CSV) heruntergeladen werden und ist unter folgendem Weblink erreichbar:

<https://openintro.org/data/index.php?data=ames>

Zusätzlich wird eine Datei zur Beschreibung der Daten seitens der Website zur Verfügung gestellt, die unter folgendem Link zu erreichen ist:

<https://jse.amstat.org/v19n3/decock/DataDocumentation.txt>

Der Datensatz wird zur Beschreibung zunächst in den Ordner des Projektes der Entwicklungsumgebung abgelegt. Daraufhin werden alle benötigten Bibliotheken für die Ausarbeitung abgerufen. Weiterführend wird der Datensatz per Befehl in tabellarischer Form dargestellt. Durch das Abrufen der Datensatz-Informationen wird ersichtlich, dass der Datensatz 2.930 Beobachtungen und 82 Variablen enthält. Dabei teilen sich die Variablen in 39 metrische- und 43 kategorische Variablen auf. Die Variablen werden anschließend visuell und deskriptiv auf die fehlenden Einträge überprüft, mit der Feststellung, dass Variablen existieren, die eine besonders hohe Anzahl von fehlenden Einträgen aufweisen. Besonders auffällig zeigen sich die Variablen *Alley*, *Fireplace QC*, *Pool QC*, *Fence* und *Misc Feature*.

Die Visualisierung der fehlenden Werte können der grafischen Darstellung in Abbildung 13 entnommen werden. Hierfür ist die Missingno Bibliothek verwendet worden. Dabei wird für jeden fehlenden Eintrag ein weißer Strich in der Spalte für die Variable vermerkt. Je weißer eine Spalte erscheint desto mehr fehlende Einträge sind vorhanden (Bilogur, 2022, S. 1 ff.).

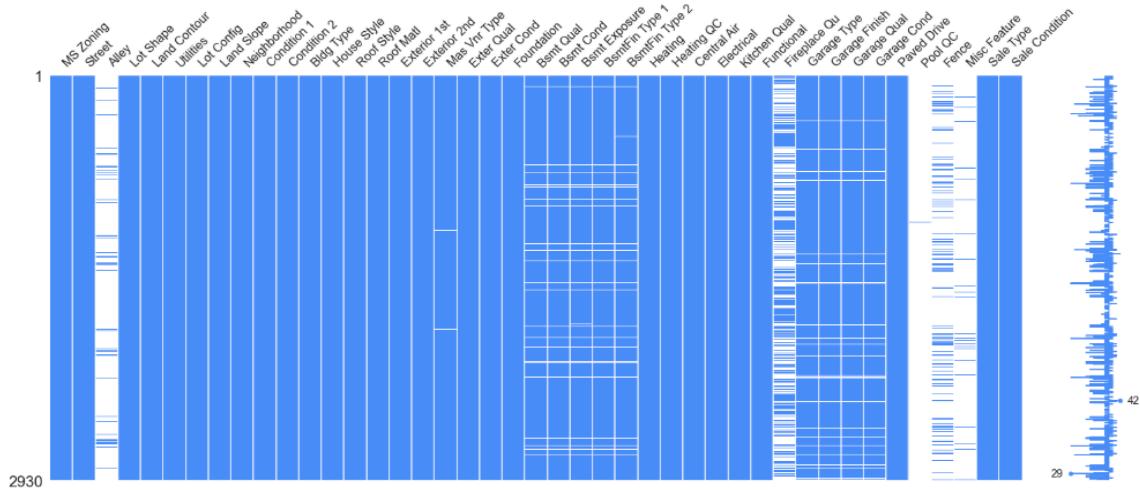


Abbildung 13: Fehlende Werte Datensatz

Weiterführend erfolgt die deskriptive und explorative Datenanalyse der Variablen. Dabei wird zunächst der Verkaufspreis der Immobilien als Zielvariable mittels eines Histogramms und eines Boxplots in der Abbildung 14 näher untersucht. Anhand des Histogramms kann eine Normalverteilung für die Zielvariabel erkannt werden. Das Boxplot-Diagramm ermöglicht es Ausreißer zu identifizieren. Der Umgang mit Ausreißern wird im Kapitel 4.3 näher erläutert.

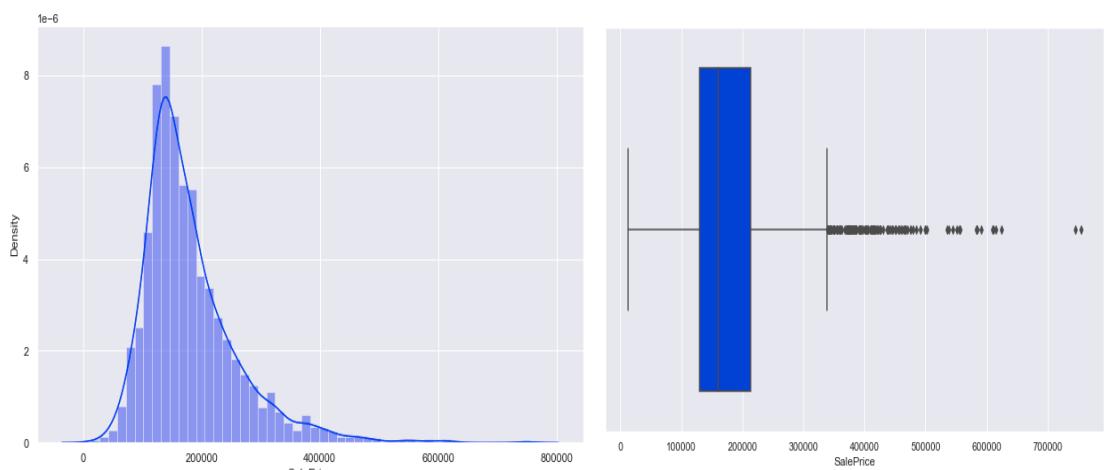


Abbildung 14: Verteilung Zielvariable

Anschließend werden die metrischen Variablen in Abhängigkeit der Betrachtungsmöglichkeit als Boxplots, Histogramme oder Streudiagramme explorativ analysiert. Auch hier ist es das Ziel, die Verteilung zu beobachten und potenzielle Ausreißer zu identifizieren. Zur Unterstützung wird parallel die deskriptive Statistik zur Bewertung der Ausreißer hinzugezogen. Gerade im Hinblick auf die Verteilung der unabhängigen Variablen ist es ebenso von Bedeutung, dass diese im Gleichgewicht sind. So kann eine zu schiefe Verteilung zur Minderung des Lerneffektes der ML-Modelle führen. Dieser Fall tritt nur ein, falls die unabhängigen Variablen in der finalen Auswahl als Teil des Modelltrainings agieren (Branco et al., 2016, S.6-7). Die Abbildung 15 stellt einen Ausschnitt der Histogramme der unabhängigen Variablen dar und lässt erkennen, dass für einige Variablen eine ungleiche Verteilung zu beobachten ist. Nachfolgend findet die Betrachtung der kategorialen Variablen ebenfalls mithilfe von Balkendiagrammen statt. Dabei wird, genau wie zuvor, ebenfalls die Verteilung der Merkmalsausprägungen beobachtet, wobei auch hierbei ungleiche Verteilungen zu berücksichtigen sind.

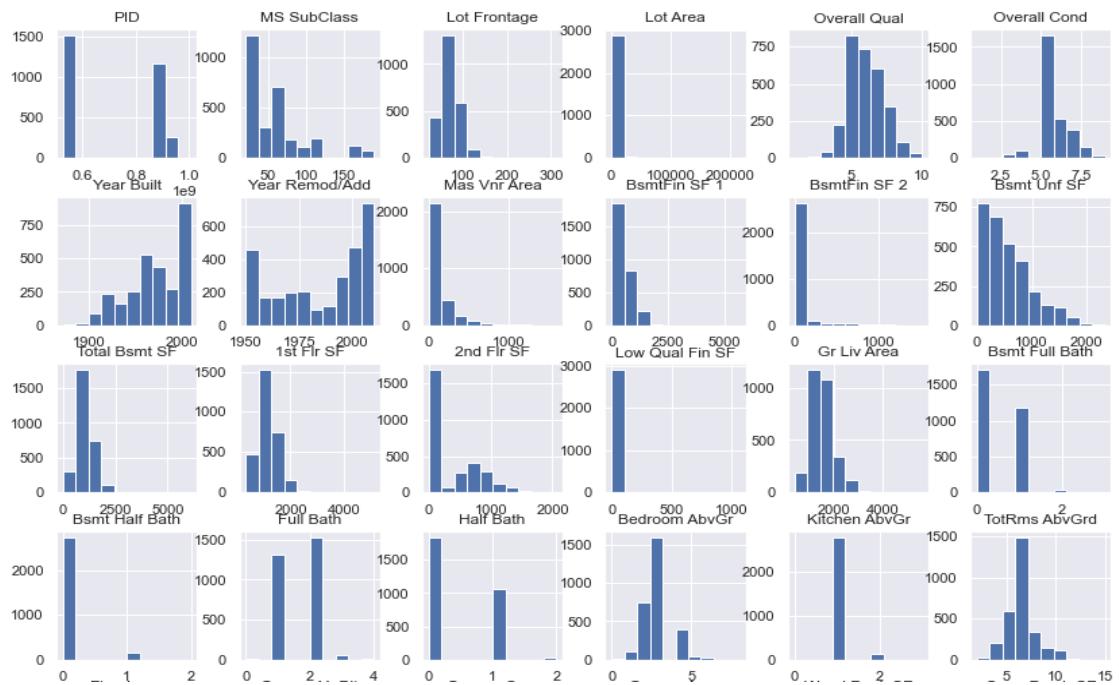


Abbildung 15: Ausschnitt der Verteilung der unabhängigen Variablen mittels Balkendiagramme

So demonstriert die Abbildung 16 beispielsweise den Verkaufstypen und die Verkaufsbedingung als Variablen in dem Datensatz. Abgesehen von der Merkmalsausprägung *WD* für *Sale Type* und *Normal* für *Sale Condition* sind nahezu alle übrigen Merkmalsausprägungen wenig vertreten. Dies birgt die Gefahr, dass die ML-Modelle konsequenterweise

keine ausgewogene Grundlage zum Lernen haben, falls die benannten Variablen verwendet werden.

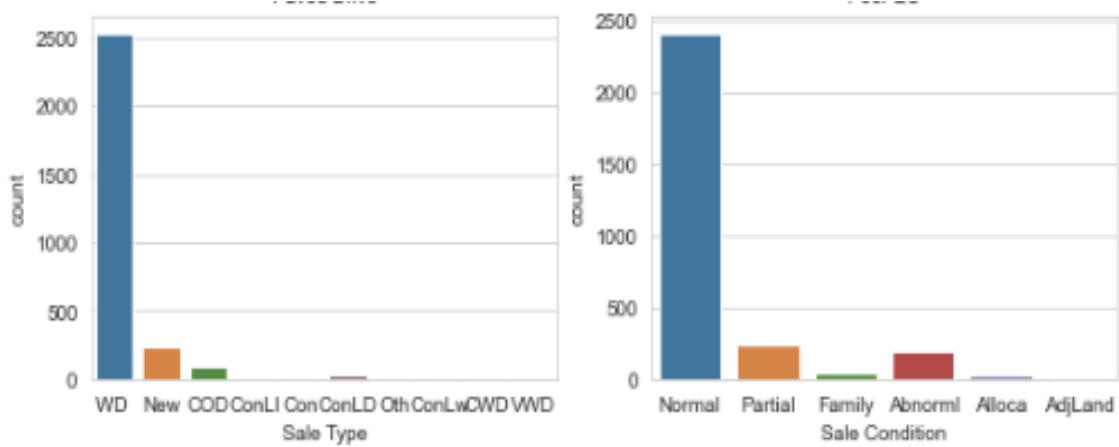


Abbildung 16: Ausschnitt der Verteilung Sale Type und Sale Condition

Von wesentlicher Bedeutung ist es, die Korrelation zwischen den unabhängigen und abhängigen Variablen zu untersuchen. Hierbei erfolgten erste Korrelationsanalysen mittels der visuellen Darstellung in der Abbildung 17. Dabei zeigt die grafische Darstellung die verwendete Darstellungsform für den Vergleich der unabhängigen Variablen *Gr Liv Area* und *Lot Area* und der Zielvariable *Sale Price*. Dies ermöglicht im Zuge der Betrachtung die erste Möglichkeit, das Verhältnis der unabhängigen metrischen Variablen zur Zielvariable zu untersuchen. Hierbei lässt sich ein linearer Anstieg der Datenpunkte erkennen. Ebenso sind dem Diagramm einzelne Ausreißer ab einer Quadratfußgröße von über 4.000 zu entnehmen. Hierbei lässt sich zunächst festhalten, dass nach dem Streudiagramm der Verkaufspreis der Immobilie mit der zunehmender Quadratfußgröße des Lebensbereichs steigt. Sowohl positiv als auch negative lineare Steigungen können auf eine positive oder negative Korrelation hindeuten. Für die Variable *Lot Area* ist kein Muster zu erkennen, das auf eine starke Korrelation hinweist, sodass sie vorerst als nicht stark korrelierend eingestuft wird. Die getätigten Annahmen anhand der visuellen Korrelationsanalysen müssen nachträglich durch statistische Verfahren geprüft werden und sind demnach kein endgültiges Kriterium, um die Variable für die Modellierung auszuschließen (Mukaka, 2012, S.69-71).

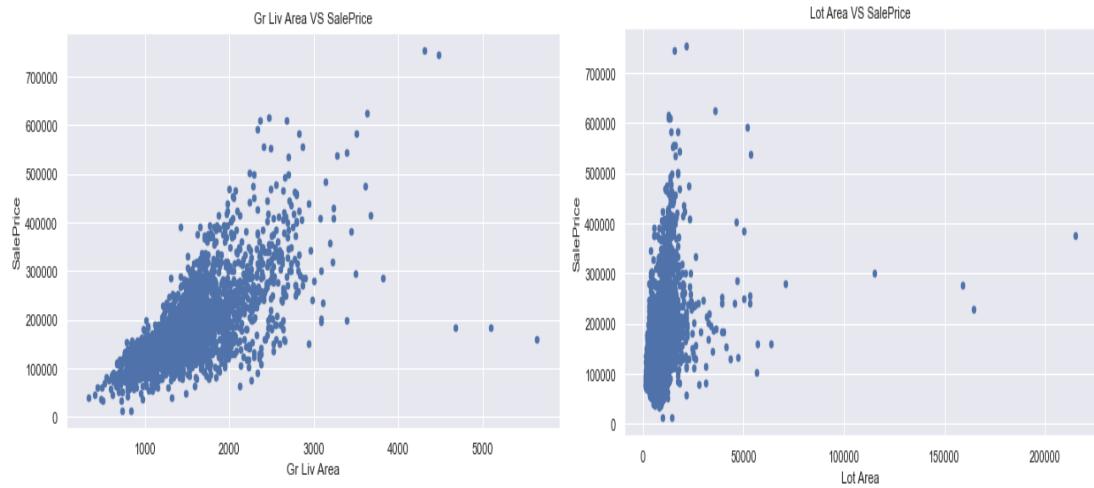


Abbildung 17: Graphische Korrelationsanalyse der metrischen Variablen

Nachfolgend werden für die metrisch stetigen- und kategorialen Variablen Boxplots zur Untersuchung der Korrelation verwendet. Wie die Verteilung der Merkmalsausprägungen für die Gesamtqualität, *Overall Qual*, sich im Verhältnis zum Verkaufspreis bewegt, lässt sich beispielsweise in der Abbildung 18 für das linke Boxplot beobachten. Es ist erkennbar, dass mit einem zunehmenden Verkaufspreis auf der y-Achse ebenso die durchschnittliche Qualität steigt. Die unterschiedlichen durchschnittlichen Werte der Merkmale der Gesamtqualität, Formen und Anordnungen der Boxplots weisen auf eine mögliche Korrelation mit dem Verkaufspreis hin. Eine schwache Korrelation lässt sich für die rechte abgebildete Variablen *Condition I* vermuten. Der Grund für die Annahme, ist die Ähnlichkeit in Bezug auf die Anordnung sowie die Größe der Boxplots je Merkmalsausprägung. Eine mögliche Schlussfolgerung kann demnach lauten, dass mit einer Zunahme der Gesamtqualität auch der Verkaufspreis steigt. Die Richtigkeit der Aussage wird im Rahmen der Anwendung der statistischen Methoden zur Korrelationsanalyse überprüft (A. Li et al., 2016, S.246-247).

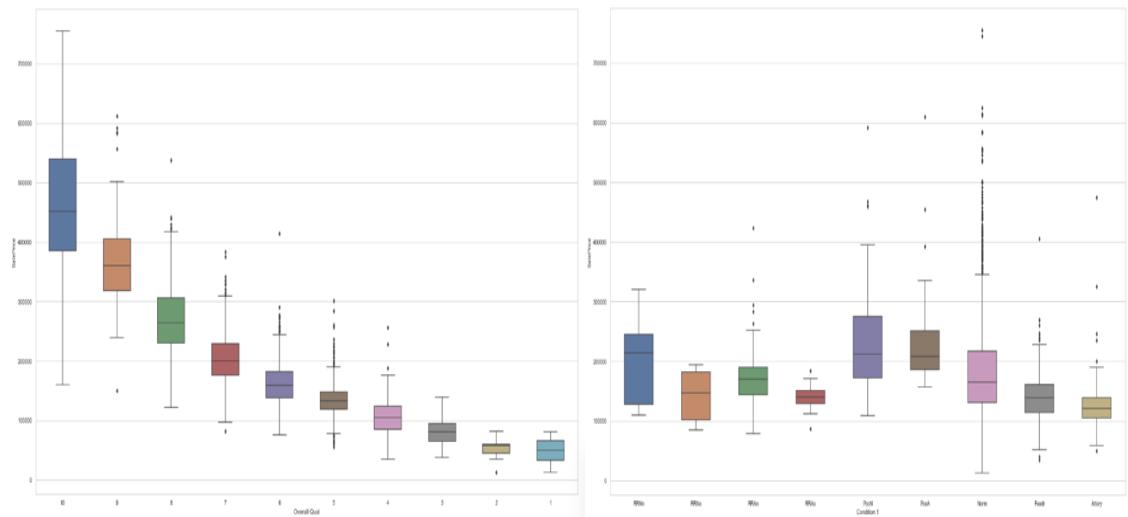


Abbildung 18: Graphische Korrelationsanalyse mittels Boxplots

Zusätzlich werden im Rahmen der explorativen Datenanalyse bereits erste Interaktionseffekte zwischen den Variablen untersucht. Diese sollen zur Erweiterung des Verständnisses für den Datensatz dienen und einen einfacheren und kumulierten Eindruck über die Variablen ermöglichen. Dabei zeigt die Abbildung 19 den Verkaufspreis der Immobilien in Abhängigkeit zur Immobilienart und Gesamtqualität.

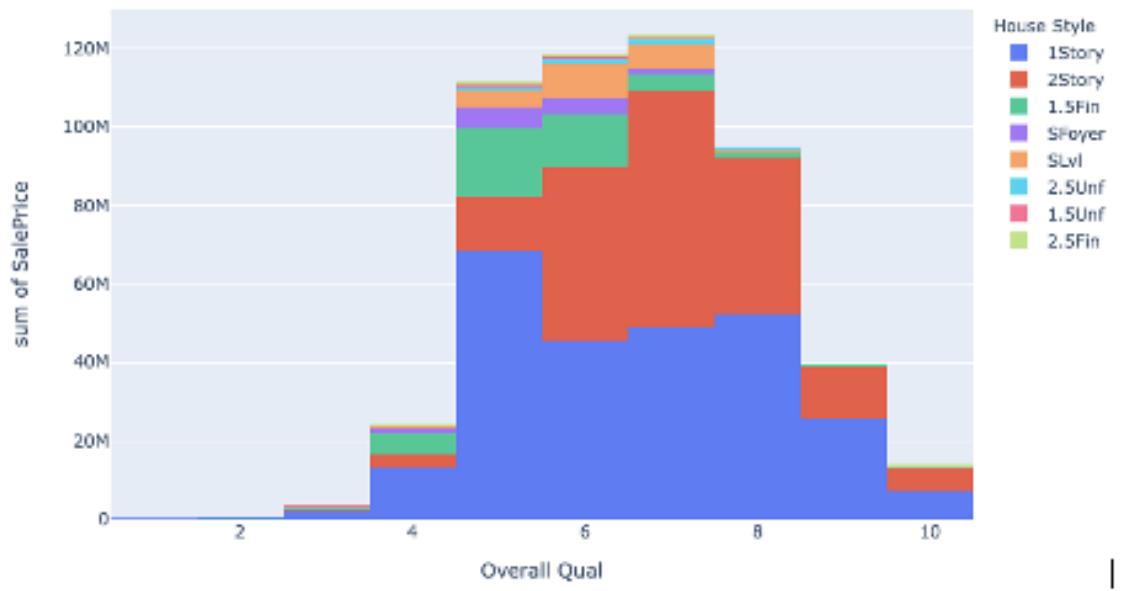


Abbildung 19: Höhe des Verkaufspreises im Bezug zur Bauart und Gesamtqualität

Letztlich wird im Rahmen der Data Understanding Phase die Datenqualität auf Basis der vorab getätigten Schritte überprüft. Die Ausreißer werden mithilfe der deskriptiven Statistik und der graphischen Darstellung der Variablen gegenübergestellt. Alle

identifizierten Ausreißer werden dabei nach mehrmaliger Betrachtung als akzeptabel und als Abbild einer möglichen Realität bewertet. Demnach wird eine Bereinigung als nicht notwendig angesehen. Nachfolgend wird erneut der Datensatz auf fehlende Einträge überprüft. Generell werden Variablen, deren Merkmalsausprägungen mehr als 70 % fehlende Werte aufweisen, kategorisch ausgeschlossen und nicht in die Modellierung mit-einbezogen. Die Überprüfung zeigt dabei, dass für die Variablen *Alley*, *Pool QC*, *Fence* und *Misc Feature* mehr als 70 % der Einträge fehlen. Im Rahmen der Überprüfung der falschen Einträge wird festgestellt, dass die fehlenden Einträge einheitlich als nicht verfügbar (NA) deklariert sind. Nach der Datenbeschreibung fungiert das NA als Platzhalter und impliziert tatsächlich das nicht vorhanden sein eines Wertes. Für *Pool QC* bedeutet dies beispielsweise, dass der eigentlich NA-Wert keine fehlenden Einträge indiziert, sondern einen nicht vorhandenen Pool. Alle identifizierten Variablen, die falsche oder fehlende Einträge enthalten, werden im nachfolgenden Kapitel angepasst.

4.2 Data Preparation

Im Anschluss zum vorherigen Kapitel 4.1 werden zunächst 14 kategoriale Variablen in Bezug auf ihre falschen Einträge durch den Modalwert befüllt. Die metrischen Variablen werden daraufhin auf fehlende Einträge untersucht und anschließend mittels der Median-Werte der Merkmalsausprägungen ergänzt. Die Entscheidung für den Median soll besonders unter Beachtung hoher, stetig metrischen Werte verhindern, dass Werte von Ausreißern stark beeinflusst werden. Nach erfolgter Bereinigung werden die Daten für die Modellierung mithilfe statistischer Verfahren ausgewählt. Gleichzeitig wird der Vorgang durch die visuellen Beobachtungen der Korrelationsanalyse unterstützt. Dafür werden nachfolgend der Korrelationskoeffizient nach Bravais und Pearson und der Spearmans Rangkorrelationskoeffizient für die metrischen Variablen verwendet. Ist der Korrelationskoeffizient größer als 0,2 oder kleiner als -0,2, so wird die Hinzunahme der Variable für das Modellieren beschlossen, da ab diesem Wert von einer gemäßigten Korrelation gesprochen wird (Mukaka, 2012, S.71 ; Ratner, 2009, S.140-141).

Dabei werden die Ergebnisse der Korrelationsanalyse der metrischen Variablen gegenübergestellt. Die Ergebnisse für die zehn am stärksten korrelierenden metrischen Variablen können der Tabelle 6 für beide Verfahren entnommen werden. Die Darstellungen aller korrelierender Werte sind dem Anhang beigelegt. Basierend auf der statistischen und

visuellen Korrelationsanalyse wurden 23 metrische Variablen als korreliert eingestuft und für die Modellierung in Betracht gezogen.

Tabelle 6: Ergebnisse der Korrelationsanalyse nach Pearson und Spearman

Pearsons- Korrelationskoeffizient	
Variable	Korrelationskoeffizient
Overall Qual	0,81
Gr Liv Area	0,71
Garage Cars	0,65
Garage Area	0,64
Total Bsmt SF	0,63
Year Built	0,62
Full Bath	0,56
Year Remod/Add	0,55
Garage Yr Blt	0,53
Mas Vnr Area	0,51
Spearmans Rangkorrelationskoeffizient	
Overall Qual	0,81
Gr Liv Area	0,72
Garage Cars	0,70
Year Built	0,68
Garage Area	0,66
Full Bath	0,63
Garage Yr Blt	0,61
Total Bsmt SF	0,61
Year Remod/Add	0,60
1st Flr SF	0,58

Für die kategorialen Variablen wird die ANOVA-Methode genutzt. Im Rahmen der Ausarbeitung wird eine Variable als statistisch signifikant eingestuft, sobald diese einen p-Wert von 0,05 unterschreitet. Die Ergebnisse der Korrelation zwischen kategorialen Variablen und der Zielvariable zeigen, dass lediglich die Variable *Utilities* als nicht korrelierend eingestuft wird. Demnach werden 41 der 42 verfügbaren kategorialen Variablen für die Modellierung genutzt. Ein Ausschnitt der ANOVA-Ergebnisse sind der nachfolgend dargestellten Tabelle 7 zu entnehmen. Die gesamtheitlichen Ergebnisse sind ebenfalls im Anhang enthalten.

Tabelle 7: Ergebnisse der Korrelationsanalyse nach ANOVA

ANOVA Ergebnisse	
Variable	p-Wert
MS Zoning	1.1004990281237354e-74
Street	0.001267537207464486
Alley	6.239449381200554e-15
Lot Shape	1.0678631356077208e-60
Land Contour	3.7436782290891496e-28
Utilities	0.21290794617934414
Lot Config	1.0412226056008607e-12
Land Slope	0.0010119492021355193
Neighborhood	0.00000013964227889
Condition 1	2.4914066953007e-25

Weiterführend werden Integrationsmöglichkeiten der Variablen geprüft. Hierbei kann die Integration zwei neue Variablen schaffen, welche die Gesamtfläche, *TotalSF*, der Immobilie abbildet und die Anzahl aller Badezimmer als Variable *Bath* ausgibt. Gerade die Anzahl der bestehenden Badezimmer sind in dem Datensatz sehr umständlich dargestellt. Außerdem weisen die Variablen der Badezimmer einzeln eine niedrige Korrelation zur Zielvariable auf. Ziel der Integration ist es, konkludente, stärker korrelierende und wertschöpfende Variablen zu schaffen, die die Vorhersagequalität der Modelle erhöhen können. Die Tabelle 8 zeigt den Korrelationskoeffizienten beiden integrierten Variablen im Vergleich zu den Variablen, die zur Integration genutzt werden. Es bestätigt sich, dass die Integration der Variablen zu einer stärkeren Korrelation mit der Zielvariablen führt. Weiterführend wird im Sinne der Ausarbeitung beschlossen, die verwendeten Variablen zur Integration nicht aus dem Datensatz zu entfernen. Dies soll dem Zweck dienen, im späteren Verlauf die Gewichtung der Variablen zu untersuchen und dabei möglicherweise festzustellen, ob sie nicht nur im Kontext der integrierten Variablen, sondern auch in Bezug auf die globale oder lokale Erklärung einen Beitrag leisten.

Tabelle 8: Ergebnisse der Korrelationsanalyse nach der Integration

Pearsons- Korrelationskoeffizient	
Variable	Korrelationskoeffizient
TotalSF	0,79
Total Bsmt SF	0,63
1st Flr SF	0,62
2nd Flr SF	0,27
Bath	0,58
Bsmt Full Bath	0,28
Bsmt Half Bath	-0,04
Full Bath	0,55
Half Bath	0,29
Spearmans Rangkorrelationskoeffizient	
TotalSF	0,81
Total Bsmt SF	0,61
1st Flr SF	0,58
2nd Flr SF	0,25
Bath	0,64
Bsmt Full Bath	0,26
Bsmt Half Bath	-0,03
Full Bath	0,63
Half Bath	0,33

Anschließend wurden die Zeilen des Datensatzes auf Duplikate überprüft, wobei keine gefunden werden. Schließlich werden die für das Modellieren relevanten Variablen in einem neuen Datensatz gespeichert, bevor mit der Formatierung der Daten begonnen wird. Im letzten Abschnitt der Data Preparation Phase werden die Variablen sukzessiv für das Modeltraining angepasst. Weiterführend werden die unabhängigen Variablen separat von der Zielvariable gespeichert. Dies geschieht ebenso umgekehrt. Für die korrekte Nutzung von LIME, im Sinne der kategorialen Variablen, ist es notwendig die kategorischen und metrischen Variablen separat in einer Liste zu speichern, welche im späteren Verlauf zum Mapping genutzt wird. Die unabhängigen Variablen und die Zielvariable werden mit einer Aufteilungsrate von 70 zu 30 in Test- und Trainingsdatensätze aufgeteilt. Nachfolgend werden die unabhängigen kategorialen Variablen zunächst durch das Encoding transformiert und anschließend werden die numerischen Daten standardisiert. Obwohl es für die Performance des XGBoost-Modells keinen Unterschied macht, ob die Daten standardisiert sind oder nicht, werden die standardisierten Daten als Ausgangslage für alle Modelle genommen, um im Zuge der Datenbasis eine Einheitlichkeit zu

gewährleisten. Der Ablauf, der zunächst die Aufteilung der Daten in Trainings- und Testdatensätze vornimmt und anschließend mit der Standardisierung beginnt, ist einer Vermeidung von Data Leakage geschuldet. Dieser beschreibt die Verwendung von Informationen, die zwischen Test- und Trainingsdaten ausgetauscht werden, um ein Modell zu trainieren. Dies kann dazu führen, dass das Modell ungenaue Vorhersagen trifft, da es auf unwahre Verknüpfungen zwischen den Daten trainiert wird (Kaufman et al., 2012, S.10 ff.). Das durchgeführte Feature Engineering wird im Anschluss überprüft und zeigt, dass alle unabhängigen Variablen eine metrische Merkmalsausprägung aufweisen und für das Modellieren genutzt werden können.

4.3 Modeling

Für die Modellierung der Preisvorhersage werden vier verschiedene Modelle verwendet, die bereits innerhalb des Kapitels 2 benannt und definiert wurden. Wie bereits im Rahmen der Datenformatierung benannt, werden 70 % der Daten zum Trainieren der Modelle genutzt, während 30 % der Daten zur Überprüfung der Generalisierungsleistung der Modelle herangezogen werden. Die Aufteilung wird als sinnvoll für die Validierung betrachtet, da durch eine Aufteilung im Verhältnis 70 zu 30 eine realistische Bewertung der Modelle ermöglicht wird. Der Zufallsstartwert wird für die Modellierung fortlaufend geändert und die verwendete Kreuzvalidierung ist die Hold-Out Methode. Eine weitere Kreuzvalidierung wird mithilfe der GridsearchCV vollzogen, um parallel die geeigneten Parameter der einzelnen Modelle zu finden. Die Ergebnisse der Modelle werden miteinander verglichen und optimiert. Die vielversprechendsten Modelle werden anschließend an den Testdaten validiert. Gleichzeitig werden definierten Leistungsmetriken zur Bewertung der Modelgütigkeit herangezogen. Die verwendeten Hyperparameter aller Modelle sind in der Tabelle 9 enthalten.

Tabelle 9: Hyperparameter der Modelle

Künstliches neuronales Netz			
Anzahl der Schichten	Anzahl Neuronen	Batch Größe	Epochen
Aktivierungsfunktion	Optimizer	Lern Rate	Dropout
K-nearest Neighbor			
Anzahl nächster Nachbarn			
XGBoost			
Lern Rate	Anzahl Entscheidungsbäume	Lernziel	Tiefe des Entscheidungsbaumes
Baum Methode	Art des Entscheidungsbaumes		
EBM			
Zufallsstartwert			

Die einzelnen Modelle werden im nachfolgenden Schritt anhand von zufällig gesetzten Hyperparametern definiert. Die Bewertung der Leistung erfolgt dabei anhand der Leistungsmetriken in numerischer Form. Lediglich für das ANN-Modell wird zusätzlich die graphische Visualisierung der Verlustfunktion hinzugezogen, die für das erfolgversprechendste Modell in der Abbildung 20 dargestellt ist.

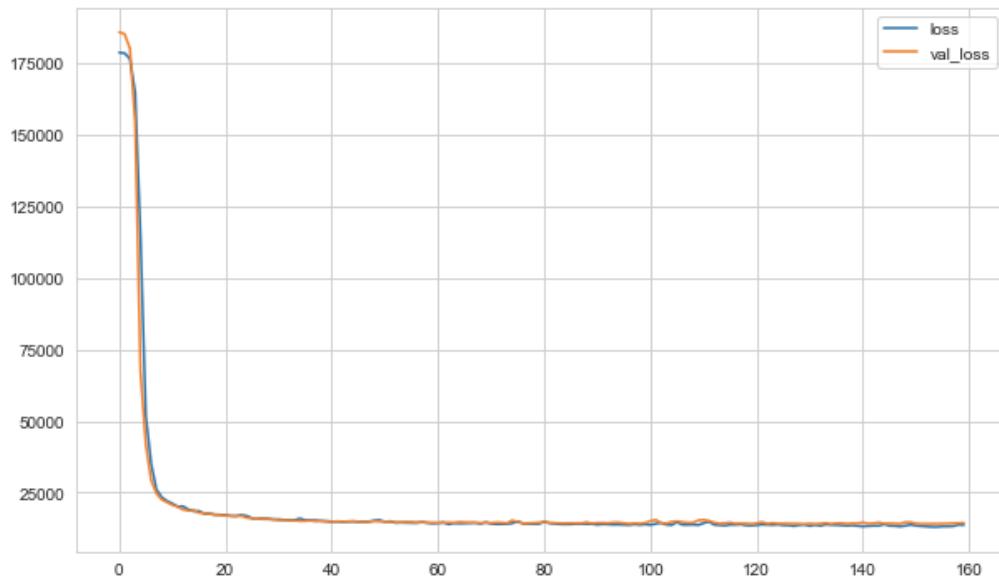


Abbildung 20: Verlustfunktion des ANN-Modells

Die Anpassung der Hyperparameter wird mithilfe der *GridSearchCV* hauptsächlich für das ANN- und XGB-Modell durchgeführt. Neben dem Einsatz der *GridSearchCV*, wird

auch der manuelle Vorgang des Trial-and-Error für die Hyperparameter Epochen, Dropout und Lernrate des ANN-Modells verwendet. Für die Batch-Größe wird zusätzlich die Callback-Funktion verwendet, um automatisch die optimale Batch Größe bestimmen lassen zu können. Ebenso wird der Vorgang des Trial-and-Error ergänzend für das XGB-Modell für die Lernrate, die Tiefe- und die Anzahl an Entscheidungsbäumen angewendet. Die Hyperparameter werden sukzessiv mit jedem Fortschritt angepasst, bis die Qualitätskriterien ausreichend erfüllt werden. Der MAPE-Wert wird als ausreichend beurteilt, sobald dieser niedriger als 30 % ist, was einer guten bis akzeptablen Vorhersage gemäß der Literatur entspricht. (Montaño Moreno et al., 2013, S.501). Ebenfalls sind die Kriterien erfüllt, wenn der Adjusted R2-Wert bei einem höheren Wert als 0,5 liegt, da dies als ein akzeptabler Fit gewertet wird (Harel, 2009, S.1109-1111).

Für den MAE- und den RMSE-Wert können keine eindeutigen Grenzen benannt werden, da die Abweichungen aufgrund der Unterschiedlichkeit der Immobilien je nach angesetztem Preis größer ausfallen können. Eine Abweichung von 30.000 \$ für eine Immobilie mit einem Wert von 500.000 \$ wäre verkraftbar, im Vergleich zu einer Immobilie mit einem Preis von 70.000 \$. Aufgrund der Erkenntnisse aus der Literaturrecherche, dem Umstand, dass hauptsächlich Ein- und Zweifamilienhäuser im Datensatz enthalten sind, dem durchschnittlichen Preis von knapp 180.000 \$ und dem 50 % Quartil bei 160.000 \$, wird eine angemessene Abweichung von 10.000 \$ bis maximal 30.000 \$ als akzeptabel angesehen. Trotzdem wird sukzessiv versucht, die Abweichungen in Richtung von 10.000 \$ oder darunter zu erreichen. Alle Leistungsindikatoren werden nachfolgend auch für die Validierungsdaten verwendet, um den Vergleich zwischen den Trainings- und Testergebnissen aufzuführen. Anschließend werden die besten Modelle gespeichert und endgültig anhand der Trainingsdaten trainiert.

4.4 Evaluation

Über alle Leistungsindikatoren hinweg weisen alle Modelle für die Trainingsdaten gute und akzeptable Ergebnisse auf. Dabei erzielen das ANN- und XGB-Modell die vielversprechendsten Ergebnisse im Rahmen des MAE- und MAPE-Wertes. Im Kontext des RMSE- und adjusted R2-Wertes, performen erneut das XGB- und das EBM-Modell am besten. Insgesamt liefert das XGB-Modell gesamtheitlich die beste Leistung ab.

Nichtsdestotrotz zeigen alle Modelle eine ähnlich gute Leistung und erfüllen die Qualitätskriterien zur weiteren Validierung anhand des Testdatensatzes.

Dabei zeigt sich, dass im Vergleich zu den Ergebnissen aus dem Trainingsdatensatz das EBM-Modell geringste Abweichung für den MAE-, MAPE- und adjusted R2-Wert erzielt. Gleichzeitig erzielt es die besten Ergebnisse für den Testdatensatz für die Leistungsindikatoren RMSE und adjusted R2. Für die Metriken MAE und MAPE verzeichnet das XGB-Modell die erfolgreichsten Werte. Das ANN-Modell weist ebenso wie das EBM-Modell die geringste Abweichung zum MAPE- und RMSE-Wert auf. Alle Ergebnisse sind der Tabelle 10 zu entnehmen.

Tabelle 10: Ergebnisse der Trainings- und Testdaten

Trainingsdaten				
Modell	MAE	MAPE	RMSE	adjusted R2
ANN	12.490,70	0,075	26.969,37	0,87
KNN	15.606,72	0,095	26.360,04	0,88
XGB	12.256,29	0,077	17.033,15	0,95
EBM	14.243,16	0,089	19.968,41	0,93
Testdaten				
Modell	MAE	MAPE	RMSE	adjusted R2
ANN	14.173,09	0,081	26.049,45	0,90
KNN	18.344,66	0,103	29.823,08	0,87
XGB	15.247,09	0,089	25.321,29	0,90
EBM	15.892,23	0,094	23.843,25	0,91

Die Evaluation der Anwendungsergebnisse der XAI-Methoden wird im nachfolgenden Kapitel 5 dargestellt. Generell wird in der Ausarbeitung von einer tiefgründigen Gegenüberstellung der Leistungsindikatoren abgesehen, da die Erstellung der Prognosemodelle die Grundlage für die eigentliche Untersuchung darstellen. Insgesamt lässt sich festhalten, dass die vier Modelle eine ähnlich gute Leistung auf den Trainings- und Testdatensatz zeigen und somit auch ein Maß an Generalisierbarkeit vorweisen. Dies ermöglicht im weiteren Vorgehen die Erklärbarkeit mittels XAI an validierten Modellen zu untersuchen. Die Ergebnisse der Anwendung der drei verschiedenen XAI-Methoden werden im nachfolgenden Kapitel beschrieben und auf Hinblick der RQ3 und RQ4 untersucht. Hierfür dienen die Trainingsdaten als Datenbasis. Dies soll zunächst die Anzahl an Dateninstanzen verringern, da die Berechnungszeit der meisten XAI-Methoden linear mit der Anzahl an mehr Dateninstanzen steigen. Außerdem werden nicht alle Variablen gesamtheitlich

betrachtet, da dies den Umfang der Ausarbeitung überproportional überschreitet. Die Darstellungen werden mindestens die zehn einflussreichsten Variablen aufzeigen und je nach Mehrwert, wird die Darstellung im Sinne Zielsetzung erweitert. Dies trifft auf alle Visualisierungen zu, bis auf die PDP-Diagramme, welche die drei einflussreichsten Variablen im globalen Kontext nochmals vertiefen werden. Die gesamten Ergebnisse der einzelnen Algorithmen werden in Kapitel 6.5 zusammengefasst und gegenübergestellt. Die verwendeten graphischen Darstellungsmöglichkeit der XAI-Methoden werden entlang der Untersuchung beschrieben und erklärt. Dabei wird die Untersuchung mit dem EBM-Modell gestartet und verläuft konsekutiv bis hin zum tiefsten Blackbox-Modell, dem ANN.

5 Untersuchung des Einsatzes von XAI-Methoden

5.1 Untersuchung der XAI-Methoden im Rahmen von EBM

Die Einzelbetrachtung der Algorithmen startet mit dem EBM-Modell. Bevor die lokale und globale Vorhersage des EBM-Modells anhand von SHAP dargestellt werden kann, ist es zunächst notwendig die SHAP-Werte der einzelnen Variablen zu berechnen. Hierfür wird als erklärende Methode der SHAP-Explainer genutzt, der für alle nachfolgenden ML-Modelle verwendet werden kann. Die Berechnungszeit der Shapley-Werte beträgt insgesamt 3 Stunden und 58 Minuten. Bei der Betrachtung von den globalen Ergebnissen der SHAP-Methodik in der Abbildung 21 kann folgendes beobachtet werden. Dies stellt ein Feature-Importance-Diagramm dar und zeigt numerisch absteigend die Variablen, die den größten Einfluss auf die Ergebnisfindung des Modells ausüben. Die Nummern hinter jedem Balkendiagramm sind als möglicher quantitativer Einfluss auf den Immobilienpreis in \$ zu verstehen (Lundberg, 2018, S.1-3).

Auffällig ist dabei, dass die zehn einflussreichsten Variablen nach SHAP, Quantitativen Variablen sind. Dabei können die drei einflussreichsten Variablen *Gr Liv Area*, *TotalSF* und *Overall Qual* eine kumulierte numerische Beeinflussung von $\pm 14.468,48$ \$ ausmachen. Die nachfolgenden Variablen erzielen einen Beitrag zwischen 2.956 \$ bis 3.835 \$ und unterscheiden sich in ihrer Höhe der möglichen Beeinflussung minimal. Dabei sind Qualitätseigenschaften der Immobilie, mit vier Variablen vertreten und Flächenmaße mit zwei. Die Anzahl an verschiedenen Zimmern, Kaminen und die Autokapazität der Garage, stellen die zehn einflussreichsten Variablen des EBM-Modells dar. Einen wesentlichen Einfluss auf die Preisvorhersage üben die zusammengefassten übrigen 232 Variablen mit 729.754,74 \$. aus. Die Beeinflussung der ersten zehn Variablen machen in der Gesamtbetrachtung einen Anteil von 31 % aus. Je nach Variablenkonstellation, können die übrigen Variablen den Einfluss der zehn einflussreichsten Variablen entgegenwirken.

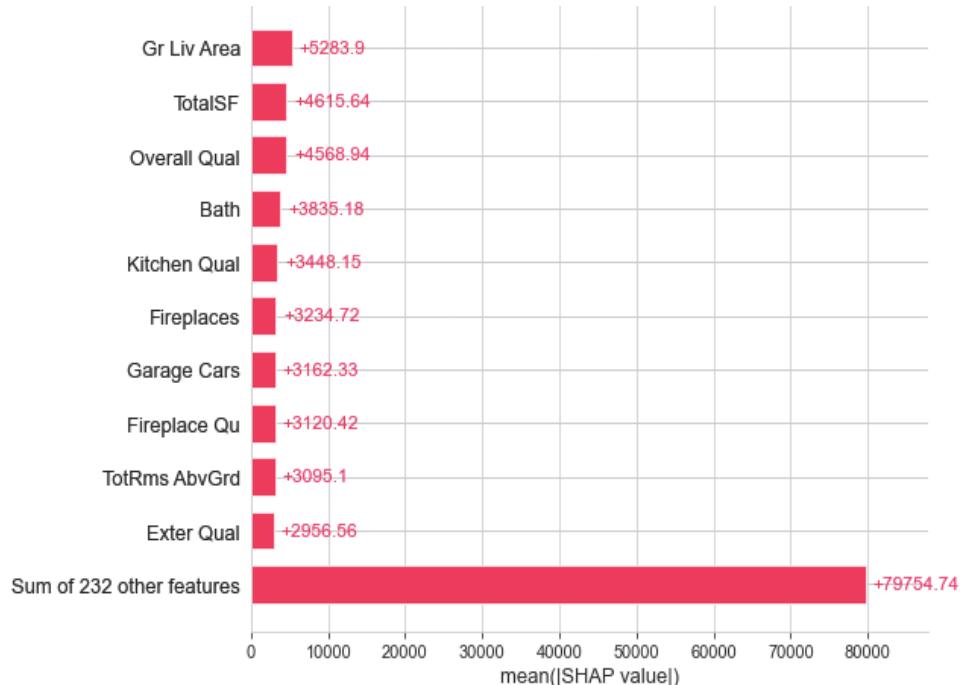


Abbildung 21: SHAP Global Balken-Diagramm EBM-Modell

Die Darstellung des Beeswarm-Diagramms in Abbildung 22 lässt in Verbindung mit dem vorherigen Balkendiagramm neue Beobachtungen zu. Dabei werden die Variablen erneut numerisch absteigend nach Ihrem Einfluss auf das prädiktive Ergebnis dargestellt. Das Beeswarm-Diagramm ermöglicht die Bedeutung jeder einzelnen Ausprägung einer Variablen für die Modellvorhersage darzustellen. Hierbei lässt sich beispielsweise erkennen, dass numerisch größere Werte der Variable *Gr Liv Area* einen positiven Zuwachs für den Preis der Vorhersage des EBMs haben. Dabei steht jeder Punkt für eine Ausprägung der betrachteten Variablen in den 2.051 Beobachtungen. Ebenso spiegelt die farbliche Darstellung der Punkte nicht die Höhe des SHAP-Wertes wider, sondern stellt die tatsächlichen Werte der Variable farblich dar. Je roter ein Punkt erscheint desto höher ist der tatsächliche Wert der Variable (Lundberg, 2018a, S.1-2).

Für die drei einflussreichsten Variablen ist deutlich zu erkennen, dass tatsächlich höhere Werte der Variablen, einem hohen SHAP-Wert entsprechen. Dabei deutet der farbliche Übergang der Variablen darauf hin, dass mit Zunahme des numerischen Wertes, der Beitrag der Variable zum vorhergesagten Preis höher ausfällt. Dies wird ebenso durch den positiven SHAP-Wert indiziert. Dies beeinflusst somit das prädiktive Ergebnis positiv und entspricht ebenso der gegensätzlichen Betrachtung. Dabei ist eine negative Beeinflussung seitens niedriger tatsächlicher Werte für das prädiktive Ergebnis zu erwarten.

Beide Beobachtungen lassen sich ebenso für die übrigen sieben Variablen zusammenfassen, wobei *TotRms AbvGrd* eine Besonderheit darstellt. Für die Anzahl an Räumen oberhalb des Kellers lässt sich beobachten, dass für zwei einzelne hohe tatsächliche Werte, negative SHAP-Werte berechnet werden und demnach zur Senkung der Preisvorhersage führen können. Dies lässt sich für einen Einzelfall der Variable *Fireplaces Qu* ebenfalls erkennen. Gerade der äußerste Wert der Variable *TotRms AbvGrd*, führt nach SHAP zu einer starken negativen Beeinflussung. Insgesamt lässt sich festhalten, dass die meisten Dateninstanzen einen SHAP-Wert von niedrigen negativen Shapley Wert bis hin zum niedrigen positiven Werten vorweisen.

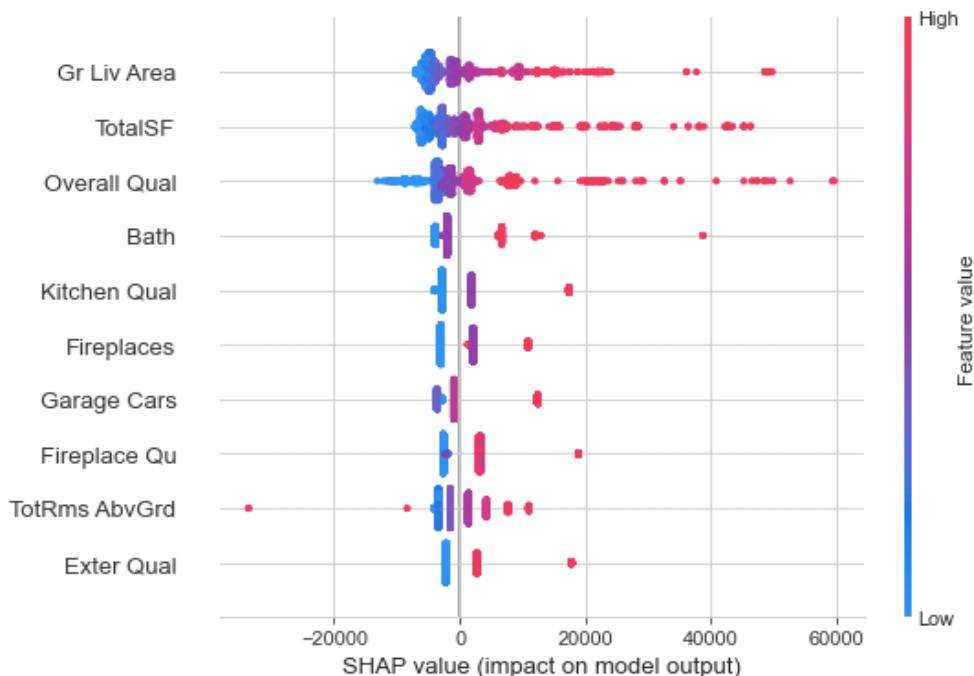


Abbildung 22: SHAP Global Beeswarm-Diagramm EBM-Modell

Im Gegensatz lässt sich aus Abbildung 23 ableiten, dass beispielsweise die Verteilung der Datenpunkte der Variable *Paved Drive_N* darauf schließen lässt, dass eine höhere Zahl, die eine Straße ohne Fahrbelag zur Immobilie indiziert, zur Preisminderung der Vorhersage führt. Zwar ist die genannte Variable nicht innerhalb der zehn einflussreichsten Variablen einzuordnen, sie attestiert jedoch, dass je nach Ausrichtungslage auch hohe numerische Ausprägungen einen negativen SHAP-Wert hervorrufen können und im konkreten Fall den Preis der Immobilie mindern. Da es sich hierbei um eine Variable handelt, die mittels des Dummy-Verfahrens die numerische Ausprägungen 1 oder 0 enthalten kann, bedeutet dies, dass die Ausprägung 1 eine Straße ohne Fahrbelag zur Immobilie

bejaht. Die Ausprägung birgt einen negativer SHAP-Wert mit sich, womit der vorhergesagte Preis im Modell sinken wird. Sollte die Ausprägung eine 0 darstellen, indiziert die Ausprägung keine Straße und kein Fahrbelag zur Immobilie. Dies bedeutet, dass der vorhergesagte Preis des Modells zunimmt. Um den gesamtheitlichen Einfluss einer kategorialen Variable zu erfassen, ist es möglich die SHAP-Werte der Merkmalsausprägungen zu addieren.



Abbildung 23: SHAP Global Beeswarm-Diagramm Variable Paved_Drive

Weiterführend wird der Blick von der globalen Analyse auf die lokale verlagert, indem die erste Vorhersage mittels der Wasserfall-Diagramme in Abbildung 24 dargestellt wird. Dabei weist $f(x)$ auf den prognostizierten Preis der Beobachtung hin. Der Durchschnittswert aller vorhergesagten Preise wird von $E[f(X)]$ dargestellt und als Ausgangswert aller Vorhersagen genutzt. Alle angezeigten Werte sind dabei in \$ zu verstehen. Das Wasserfall-Diagramm gibt durch die unterschiedliche Länge und Farbe der Balken den Einfluss der Variablen wieder (Lundberg, 2018b, S.1).

Die lokale Analyse beginnt mit der Beobachtung 1. Dabei beträgt der prognostizierte Immobilienpreis nach dem EBM-Modell 167.333,27 \$ und geht nach SHAP von einem Basiswert von $E[f(X)] = 175.441,14$ \$ aus. Dabei wird der Basiswert nachfolgend durch den Beitrag der einflussreichsten Variable Garage Cars um 12.477,28 \$ erhöht. Die nachfolgenden drei Variablen *Garage Area*, *Gr Liv Area* und die Zugehörigkeit zur Nachbarschaft *Crawford* führen erneut nach SHAP zur Erhöhung des Basiswertes um kumuliert 34.125,70 \$. Dabei geht von der Garagenfläche ein nahezu ähnlicher Einfluss aus wie von der Garagenkapazität *Garage Cars*. Anschließend senkt der Garagentyp, das Modernisierungsjahr und der Gebäudetyp, der durch zwei Ausprägungen vertreten ist, das prädiktive Ergebnis. Dabei können die zwei Ausprägungen des Gebäudetyps zusammengefasst nach SHAP betrachtet werden und mindern demnach das Ergebnis um 9.605,62 \$. Dazwischen erhöht nochmals die Landschaftseigenschaft, die einem Hügel entspricht, auf dem die Immobilie steht, den Basiswert. Mit einem Anteil von 31 % an der beeinflussten Gesamtsumme, üben die übrigen Variablen einen wesentlichen Einfluss auf die Preisvorhersage aus. Nichtsdestotrotz lässt sich festhalten, dass der negative Einfluss aller Variablen

größer als der positive ausfällt und somit die Preisvorhersage des EBM-Modells vom Basiswert weggedrückt wird.

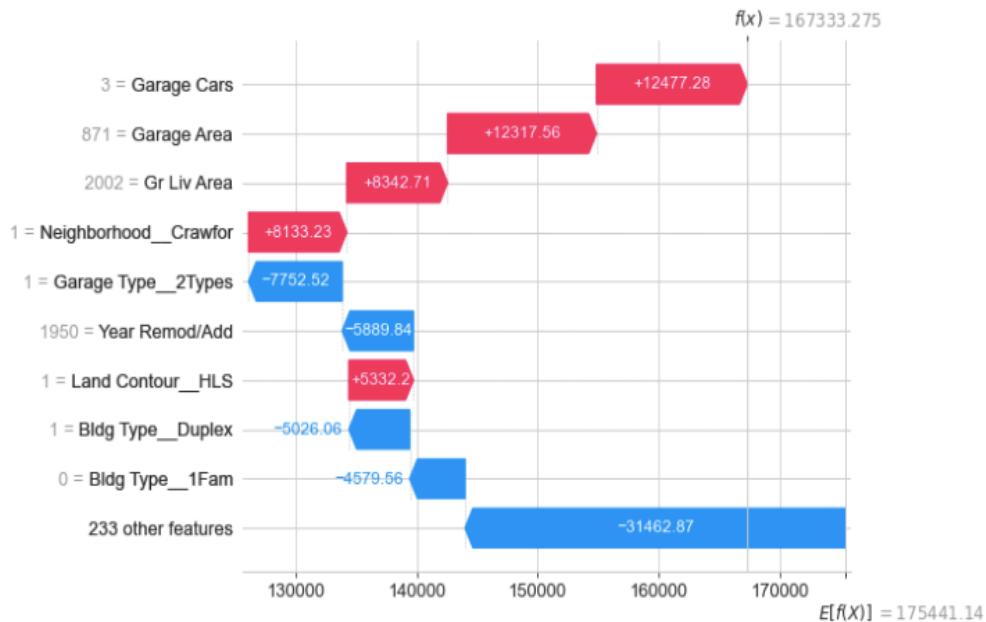


Abbildung 24: SHAP Lokal Wasserfall-Diagramm EBM-Modell Beobachtung 1

Die Beobachtung 2 in der Abbildung 25 liegt mit einem Vorhersagepreis von $f(x)=240.671,15 \$$ quantitativ gesehen 97.671,15 \$ über dem tatsächlichen Verkaufspreis. Die zehn einflussreichsten Variablen haben hierbei einen wesentlichen Anteil zum hohen Vorhersagepreis beigetragen. Signifikant sind dabei die Küchenqualität, die Anzahl an Kaminen und der ausgebauter Keller, der einem Lebensraum gleichkommt. Die drei Variablen beeinflussen die Preisvorhersage mit einem Betrag von 3.6567,13 \$, wobei bereits die Küchenqualität mit einem Beitrag von 17.450 \$ zur Preisvorhersage beiträgt. Die nachfolgenden sieben Variablen üben einen einzelnen Einfluss zwischen 3.800 \$ bis 6.700 \$ aus. Die Gesamtfläche des Lebensraumes oberhalb des Bodens bildet dabei den einzigen negativen Beitrag innerhalb der top zehn. Die übrigen Variablen beeinflussen das prädiktive Ergebnis lediglich um 4.580,38 \$. Demnach lässt sich festhalten, dass nach SHAP die hohe Preisvorhersage des EBM-Modells weitestgehend durch den Beitrag der zehn einflussreichsten Variablen zustande kommt. Dabei beeinflussen die ersten drei Variablen maßgeblich den Vorhersagepreis mit einem Anteil von 50 % des Gesamtbetrages.

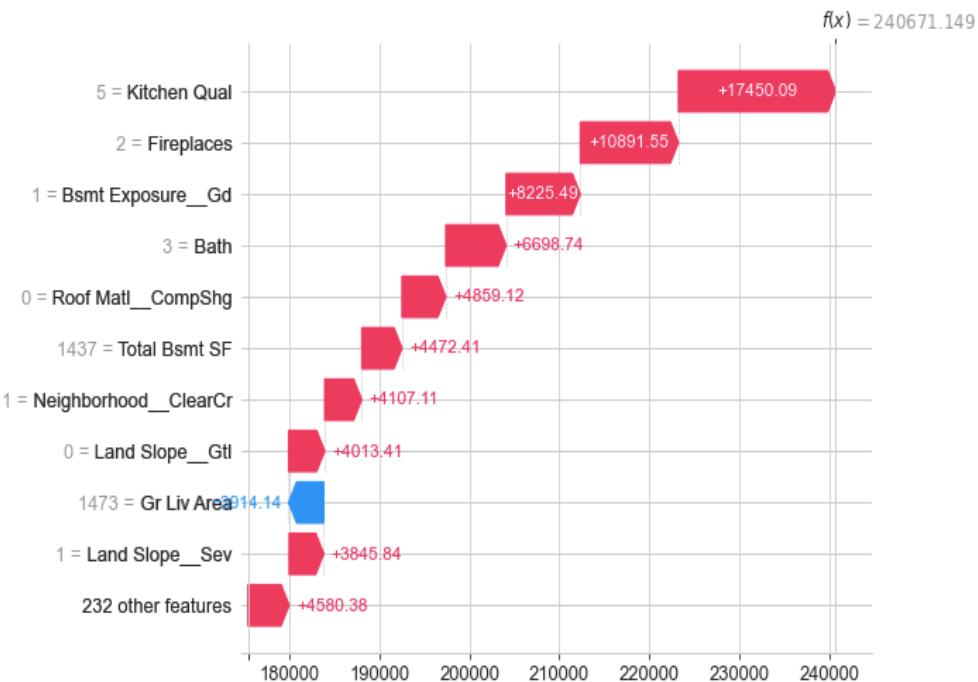


Abbildung 25: SHAP Lokal Wasserfall-Diagramm EBM-Modell Beobachtung 2

Wie bereits in der Erstellung eines Prognosemodells beschrieben, ergeben sich diverse Herausforderungen im Rahmen der Untersuchung von LIME. Hierbei werden verschiedene Ansätze im Feature Engineering getestet und untersucht. Das einheitliche Anpassen der Daten mittels One-Hot Encoding und Standardisierung führt dazu, dass die kategorialen Variablen die Ergebnisfindung der Modelle verzerrt. Die Funktionsweise von LIME ist bereits in Kapitel 2.4.2 beschrieben und impliziert, dass kategoriale Variablen, die mittels des One-Hot Encodings transformiert wurden, zu Beobachtung führen, die kein tatsächlicher Bestandteil des Datensatzes sind. LIME gewichtet jede vorhandene Variable und demnach auch die transformierten kategorialen Variablen. Dies führt zu einer Übergewichtung der kategorialen Variablen, da LIME nicht wissen kann, dass die einzeln aufgeführten Merkmalsausprägungen zu einer kategorialen Variable gehören. Dabei verzerren die transformierten kategorialen Variablen die Erklärbarkeit, sodass Beobachtungen und Konstellationen von Variablen entstehen können, die mit großer Wahrscheinlichkeit nicht in den Daten auftreten. Dies lässt sich beispielhaft der Abbildung 26 für die lokale Beobachtung 1 entnehmen. Hierbei lässt sich die hohe Gewichtung der unbehandelten kategorialen Variablen deutlich erkennen.

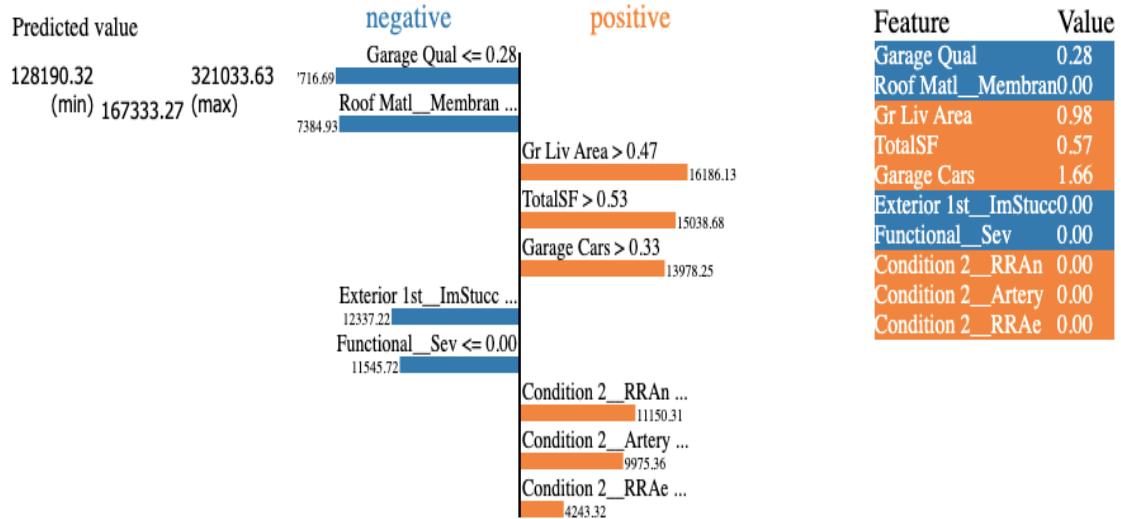


Abbildung 26: LIME un behandelte kategoriale Variablen lokal EBM-Modell

Daraufhin wird eine Feature-Engineering-Methode von Lemagnen (2019) und Ribeiro (2018) speziell für LIME und alle Modelle genutzt. Die beiden Ansätze werden im Kontext der Modelle für die Nutzung von LIME so ergänzt, dass die Interpretierbarkeit durch die Nutzung der Standardisierung steigt, ohne dass die Qualität der Modelle zu stark abnimmt. Dies ist gerade unter Berücksichtigung des Einsatzes des KNN-Modells von Bedeutung, da dieses bevorzugt mit normalisierten Daten arbeitet. Hierfür werden die Beiträge der kategorischen Variablen so angepasst, dass Sie der Gewichtung des linearen Modells gleich sind. Außerdem werden die Merkmalsausprägungen erneut zu Ihrer Ausgangsvariabel zugewiesen, damit eine Gewichtung der Variable korrekt berechnet werden kann (Lemagnen, 2019, S.1 ff.; Raju et al., 2020, S.731-732; Ribeiro, 2018, S.1 ff.).

Nachfolgend werden die globalen Ergebnisse nach LIME anhand eines Balkendiagramms dargestellt. Dabei wird der *LimeTabularExplainer* als Erklärungsmethode genutzt, der für tabellarische Daten entworfen wurde (Ribeiro, 2016, S.1 ff.). Die Berechnungszeit beträgt 14 Minuten. Die Globale Analyse der LIME-Ergebnisse in Abbildung 27 verdeutlicht, dass über alle Beobachtungen hinweg, die Garagenqualität die stärkste Gewichtung verzeichnet. Dabei weichen die durchschnittlich gewichteten Beiträge der Variablen *TotalSF*, *Gr Liv Area*, *Roofmatl* und *Garage Cars* nicht besonders stark von Garage Area ab und können somit als ähnlich einflussreich gewertet werden. Interessant erscheint dabei die Einflussnahme der Straße und der Nachbarschaft der Immobilie. Dabei beziehen beide Variablen die Lage der Immobilie ein und können zusammengefasst eine ähnlich

hohe Gewichtung wie die Garagenqualität ausmachen. Auffällig ist hierbei die hohe Einordnung der Variablen der Garagenqualität. Ebenso auffällig ist die hohe Signifikanz, die LIME dem EBM-Modell für die Beschaffenheit des Daches und der zentralen Lüftung zuschreibt. Insgesamt sind wesentliche Unterschiede zwischen den Globalen Analysen von LIME und SHAP zu beobachten, abgesehen von *TotalSF* und *Bath*, die einen ähnlichen Einfluss in beiden Methoden aufweisen

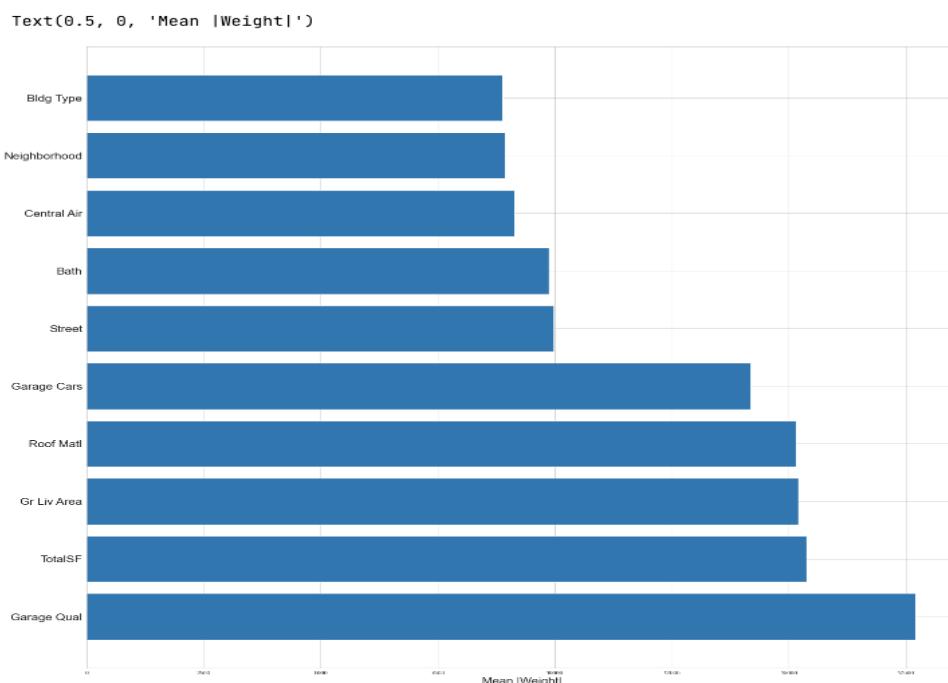


Abbildung 27: LIME Global Balken-Diagramm EBM-Modell

Aus der Abbildungen 28 ist die lokale Betrachtung des EBM-Modells mittels LIME zu entnehmen. Dabei werden seitens LIME ebenfalls Wasserfall-Diagramme verwendet und entsprechen der gleichen Logik wie bei SHAP. Die Berechnungszeit der Beobachtungen beträgt dabei 6 Minuten. Für die Beobachtung 1 lässt sich erkennen, dass ähnlich wie bei SHAP, die Garage als Kategorie einen hohen Einfluss auf das prädiktive Ergebnis ausübt. Dabei spiegelt die Garagenqualität den größten Einfluss in negativer Form wider. Positiven Einfluss auf die Preisvorhersage nehmen die Aufnahmekapazität der Garage und die Fläche der Garage, bevor der Garagentyp die Kategorie Garage durch einen negativen Beitrag komplettiert. Einen größeren Effekt als noch in der SHAP-Analyse haben laut LIME die Variablen *Gr Liv Area* und die Gesamtfläche *TotalSF*. Unter Berücksichtigung aller Gewichtungen überwiegen die negativen Beiträge der Variablen für das

Gesamtergebnis und werden durch das Dachmaterial, die Anzahl der Badezimmer, dem vorherrschenden Gebäudetypen und der nicht vorhandene Zentrallüftung komplettiert. Das Dachmaterial und die nicht vorhandenen Zentrallüftung werden im Vergleich zu SHAP nicht aufgeführt und stellen in der Anwendung von LIME für das EBM-Modell eine Besonderheit dar. Die ersten sechs Variablen weisen eine Beeinflussung von über 10.000 \$ auf und allein die Garagenqualität übt nahezu einen Einfluss in Höhe 20.000 \$ auf die Preisvorhersage aus.

Die Beobachtung 2 beginnt identisch wie die Beobachtung 1 und impliziert die Garagenqualität als einflussreichste Variable, mit einem negativen Beitrag von 20.668 \$. Daraufhin führen die neun nachfolgenden Variablen zu einem Zuwachs der Preisvorhersage. Dabei liegt die numerische Beeinflussung der Küchenqualität, des steilen Grundstückes, der Anzahl an Kaminen, des Gebäudetyps und die Quadratfußanzahl des ersten Stockwerkes über 10.000 \$. Besonders die hohe Einordnung der Küchenqualität und der Kaminanzahl entspricht einer ähnlichen Einschätzung seitens SHAP. LIME ordnet jedoch den steilen Abstieg des Grundstücks als wesentlich einflussreicher ein. Ebenso führt LIME den Gebäudetypen als relevant auf, genauso wie die Quadratfußanzahl des ersten Stockwerkes und die vorhandenen Zentrallüftung. Es wird ersichtlich, dass LIME entgegen SHAP die Garagenqualität als hochgradig einflussreich bewertet. Nichtdestotrotz überwiegt der positive Beitrag der nachfolgenden Variablen, um die Preisvorhersage des Modells über den tatsächlichen Preis zu definieren.

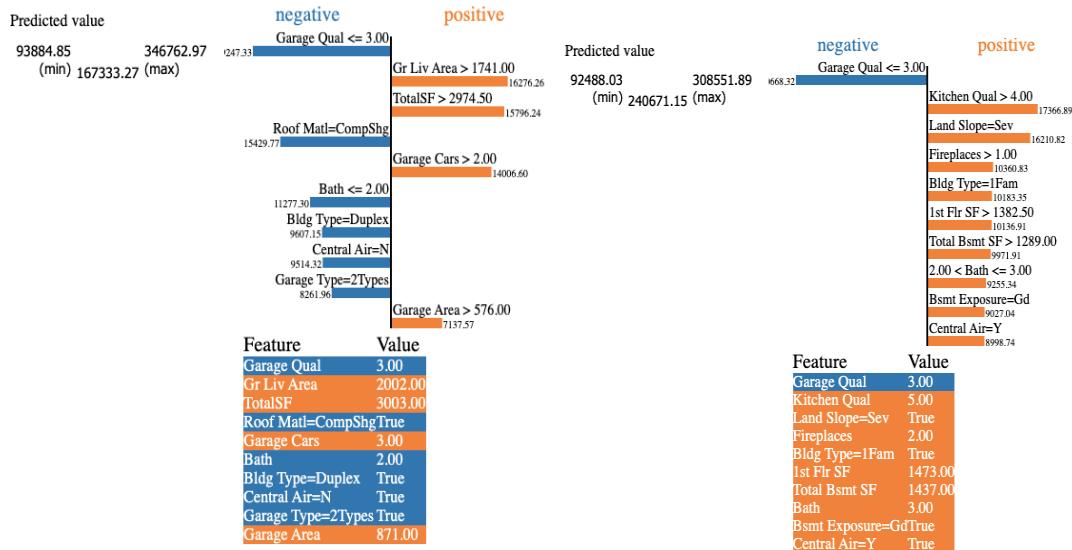


Abbildung 28: LIME Lokal Wasserfall-Diagramm EBM-Modell Beobachtung 1

Die Nutzung des InterpretML-Moduls zur Erklärbarkeit des EBM-Modells findet unter besonderen Bedingungen statt, da der Algorithmus wie bereits beschrieben, entlang des InterpretML als Glassbox-Modell entwickelt wurde. Die Darstellungsart ist ebenfalls wie bei SHAP und LIME ein Balken- und PDP-Diagramm für die Globale Analyse und für die Lokalen Ergebnisse ein Wasserfall-Diagramm. Die Analyse des EBM-Modells wird mit der Globalen Betrachtung des InterpretML Moduls begonnen. Die Berechnungszeit ist dabei mit circa 1 Minute zu verzeichnen. Dabei wird anhand der Abbildung 29 der gesamtheitliche Beitrag, den die Variablen für die Preisvorhersage leisten, in einem Feature Importance-Diagramm dargestellt. Signifikant ist dabei der Anteil der quantitativen Variablen, die den ersten bis zum zehnten Platz belegen. Dabei bilden die Gesamtfläche und Lebensraumfläche oberhalb des Kellers die einflussreichsten Variablen, mit einem jeweiligen Beitrag von durchschnittlich 4.187 \$ und 4.119 \$. Nachfolgend tragen für das prädiktive Ergebnis die Variablen *Bath*, *Overall Qual*, *Kitchen Qual*, *Fireplaces*, *Fireplaces Qual*, *TotRms AbvGrd*, *Garage Cars* und *Exter Qual* einen durchschnittlichen Beitrag zwischen 3000 \$ bis 3700 \$ bei. Dabei lässt sich beobachten, dass vier Variablen den Qualitätseigenschaften zugeordnet werden können und zwei Variablen den Flächenmaßen. Neben der Zimmeranzahl des Badezimmers, oberhalb des Kellers und der Kaminanzahl, wird außerdem die Anzahl des Fassungsvermögens der Garage für Autos als einflussreich eingestuft. Der Kamin stellt eine gesonderte Kategorie nochmals dar, da nicht nur die Anzahl der Kamine als signifikant eingestuft wird, sondern auch Ihre Qualität. Nichtsdestotrotz werden die Flächenmaße als wesentlich einflussreicher eingestuft und können im Vergleich zu der Anzahl an Badezimmern, einen 10 % höheren durchschnittlichen Beitrag zum prädiktiven Ergebnis beitragen.

Overall Importance: Mean Absolute Score

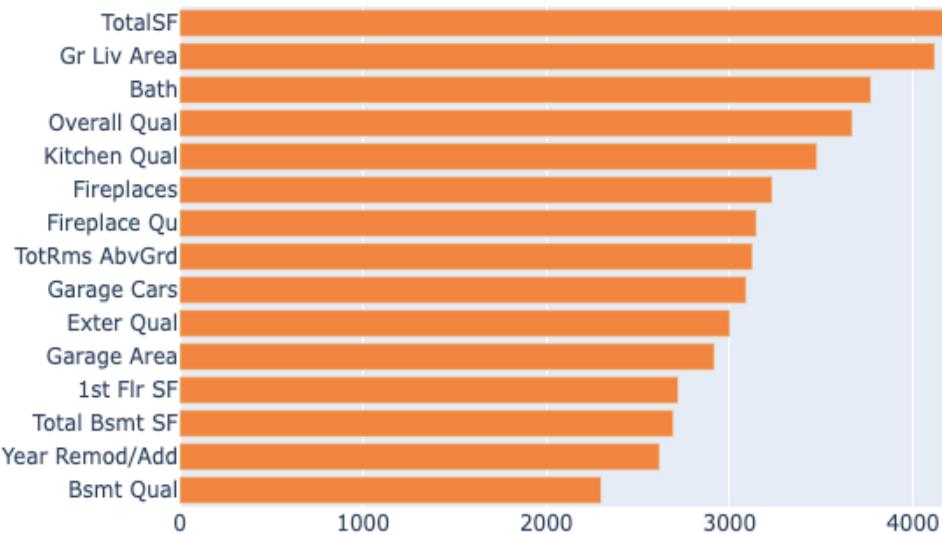
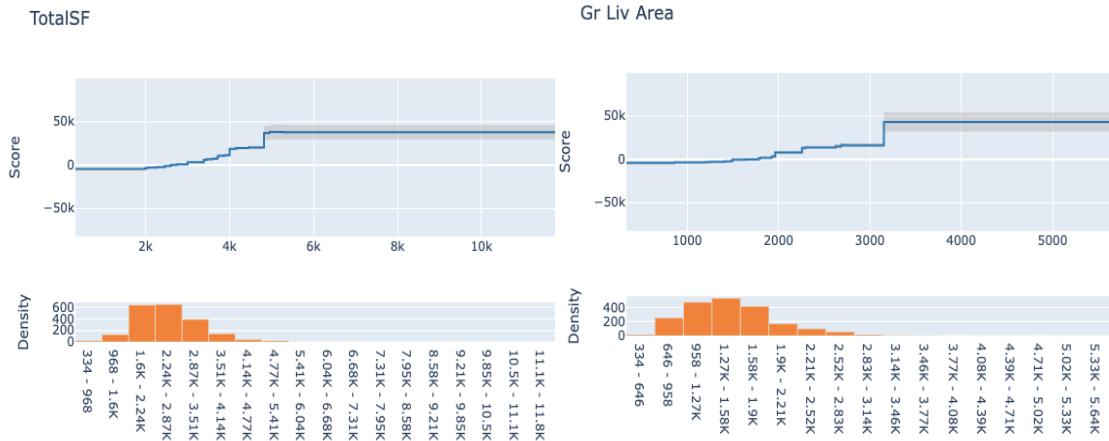


Abbildung 29: InterpretML Global Balken-Diagramm EBM-Modell

Nachfolgend werden die drei einflussreichsten globalen Variablen mittels des PDP-Diagramms genauer analysiert. Dabei lässt sich bei genauerer Betrachtung der Abbildung 30 feststellen, dass für die Gesamtfläche eine Anzahl ab 2.500 Quadratfuß den Beitrag für die Preisvorhersage steigen lässt. Für die Lebensfläche oberhalb des Kellerraumes steigt die Beitragshöhe ab 1.500 Quadratfuß. Dabei lässt sich ebenso feststellen dass der Beitrag beider Variablen sich zunächst aus einem negativen Beitrag ins positive entwickelt und unter den benannten Schwellenwerten einen negativen Einfluss auf das prädiktive Ergebnis ausüben. Anhand der Verteilungsgrafik im unteren Bereich des Diagramms, lässt sich ebenso der Beitrag der Ausreißer für die Preisvorhersage erkennen, wobei diese eine erhöten Beitrag zur Folge haben und nachfolgend konstant bleiben. Diese ist ebenfalls im Hintergrund grau markiert, was ein Indiz dafür ist, dass für den Bereich keinen Merkmalsausprägungen vorhanden sind und somit keine weitere Projektion des Einflusses in Form des Beitrages der Variablen ermöglicht.



Die Variable *Bath* in der Abbildung 31 weist ein ähnliches Muster auf, indem der Beitrag erst ab dem Schwellenwert von 2,5 sich aus einem negativen und neutralen Einfluss sich hin zu einem positiven entwickelt. Ab einem Wert von 3,5 steigt erneut der Beitrag zur Preisvorhersage und bleibt nachfolgend auf einem konstanten Niveau. Zum Vergleich wird der Beitrag der Gesamtqualität erst ab einem Wert von 7,6 als positiv verzeichnet und steigt nachfolgend sprunghaft bis zu einem Wert von 9,4 an.

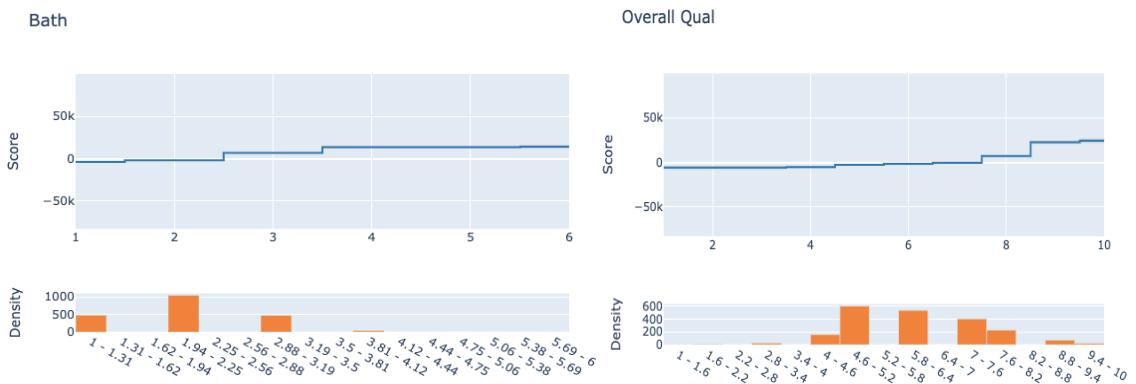


Abbildung 31: InterpretML Global PDP-Analyse Bath und Overall Qual

Nachfolgend wird die Analyse des EBM-Modells anhand der lokalen Betrachtung abgeschlossen. Die Berechnungszeit je Beobachtung liegt dabei bei unter einer Minute. Abbildung 32 zeigt den Beitrag der Variablen für das prädiktive Ergebnis der Beobachtung 1 und 2. Dabei generiert in der Beobachtung 1 die Garage als obigeordnete Kategorie einen wesentlichen Einfluss auf den Vorhersagepreis. Demnach beeinflusst die Garagengröße den Vorhersagepreis am stärksten, gefolgt von drei Beherbergungsplätzen für Autos und zwei bebauten Garagen als Garagentyp. Hierbei beeinflussen die Flächen- und

Beherbergungsgröße den Immobilienpreis positiv mit kumuliert 25.300 \$. Der Garagentyp senkt nachfolgend das prädiktive Ergebnis um 7.762 \$. Weiterführend listet InterpretML die Zugehörigkeit zur Nachbarschaft Crawford als dritt wichtigste Variable auf. Nachfolgend erhöhen die Fläche des Lebensraumes oberhalb des Kellers und die Lage zu einem steilen Hang die Preisvorhersage nochmals, bevor die nachfolgenden vier Variablen entgegen dem positiven Zuwachs wirken. Die vier Variablen umfassen zusammengefasst den Gebäudetypen und die zentrale Klimaanlage und bilden den Abschluss der zehn einflussreichsten Variablen der Beobachtung 1. Auffällig ist dabei, dass der InterpretML anhand der Ausprägungen der kategorialen Variablen das Modell doppelt bestraft. Für die Beobachtung 2 lässt sich erkennen, dass die 10 einflussreichsten Variablen einen positiven Beitrag für das prädiktive Ergebnis erzielen. Hierbei wird zunächst durch die zwei quantitativen Variablen der Küchenqualität und der Anzahl an Kaminen die Preisvorhersage am stärksten durch insgesamt 27.748 \$ beeinflusst. Nachfolgend führen die hochwertige Kellerqualität und die Anzahl an drei Badezimmern zur Erhöhung des Preises. Daraufhin wird das Dachmaterial, welches keinen recycelten Plastik-Dachschindeln entspricht, als einflussreich eingestuft. Anschließend wird die Lage in der Nachbarschaft Clear Creek, die Gesamtfläche des Kellers und das Abfallen des Grundstückes als positiv beitragend für das EBM-Modell eingestuft. Dabei kann erneut eine zusammengefasste Betrachtung für die Variable *Land Slope* stattfinden, da zwei Ausprägungen zur positiven Erhöhung des prädiktiven Ergebnisses führen. Letztlich trägt die Verkaufsbedingung *Partial* zur Erhöhung der Preisvorhersage bei. Zusammenfassend lässt sich beobachten, dass alle Variablen außer *Kitchen Qual* und *Fireplaces* einen Beitrag zwischen 3.000 \$ bis 8.500 \$ für den Vorhersagepreis leisten.

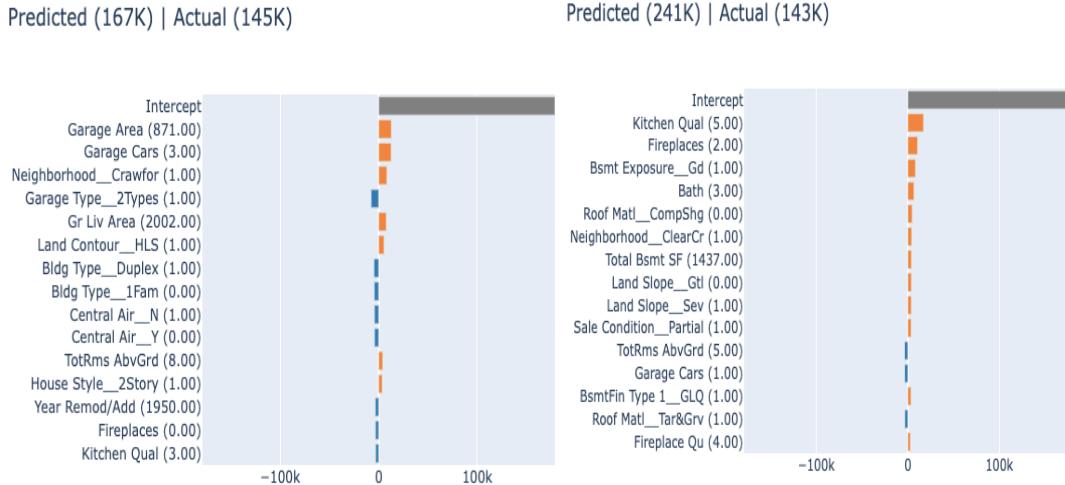


Abbildung 32: InterpretML Lokal Wasserfall-Diagramm EBM-Modell Beobachtung 1+2

5.2 Untersuchung der XAI-Methoden im Rahmen von KNN

Die Untersuchung des KNN-Modells stellt eine besondere Herausforderung dar. Dies unterliegt dem Grund, dass KNN-Algorithmen bevorzugt mit normalisierten Daten arbeiten. Im ersten Durchlauf der Modellierung werden die Daten zunächst zu Überprüfungszwecken für das KNN-Modell normalisiert. Dies führt zu einer minimal besseren Performance des Modells, mündet jedoch in unverständliche und nicht nachvollziehbare Ergebnisse der Anwendung von XAI-Methoden. Nach wiederholter Evaluierung des Modells und der Vorgehensweise, wird wie bereits beschrieben die Untersuchung aller XAI- Methode mittels standardisierter Daten vollzogen. Dabei startet auch für das KNN-Modell die Untersuchung anhand von SHAP. Dabei beträgt die Berechnungszeit der einzelnen SHAP-Werte 37 Minuten. Hierfür wird der Explainer als Erklärungsmethode genutzt. Bei der globalen Betrachtung in der Abbildung 33 fällt auf, dass die mögliche numerisch dargestellte Beeinflussung der einzelnen Variablen keine großen Abstände aufweisen. Demnach üben die ersten zehn Variablen einen ähnlichen Einfluss aus.

Gesamtheitlich üben die Variablen *Gr Liv Area*, *Overall Qual* und *TotalSF* den größten Einfluss auf die Vorhersage des Immobilienpreises aus. Kumuliert liegt die Beitragshöhe der Variablen bei ±14.452,38 \$. Dabei wertet das KNN-Modell den Lebensbereich über den Boden als signifikanter für die Vorhersage ein als die Gesamt-Quadratfußanzahl des Hauses, was den Keller und weitere Räume einschließen kann. Anschließend bewegen sich die nachfolgenden sieben Variablen in einem Bereich von 2.800 \$ bis 3.510 \$. Interessant erscheint dabei die Einflussnahme der Kaminqualität auf das prädiktive Ergebnis,

wobei nicht eindeutig zu erkennen ist, ob ein Kamin vorhanden sein muss oder nicht. Die Einflussnahme von Qualitätseigenschaften ist besonders signifikant, da vier der zehn Variablen verschiedenen qualitativen Aspekte der Immobilie bewerten. Außerdem lassen sich die Variablen *TotRms AbvGrd* und *Bath* einer möglichen Kategorie „Zimmeranzahl“ zuordnen, die neben den Flächenmaßen und Garageneigenschaften, Einfluss auf das prädiktive Ergebnis ausüben. Auffällig zeigt sich der mögliche Einfluss der 232 übrigen Variablen, die eine Auswirkung in Höhe von bis zu $\pm 49.682,88$ \$ erzielen können. Demnach lässt sich für die Globale Betrachtung des KNN-Modells festhalten, dass nach SHAP eine ähnliche Feature Importance für die ersten zehn Variablen zu erkennen ist. Den größten Einfluss üben die kumulierten 232 übrigen Variablen aus und können somit den Einfluss der ersten zehn Variablen insgesamt schmälern oder entgegenwirken.

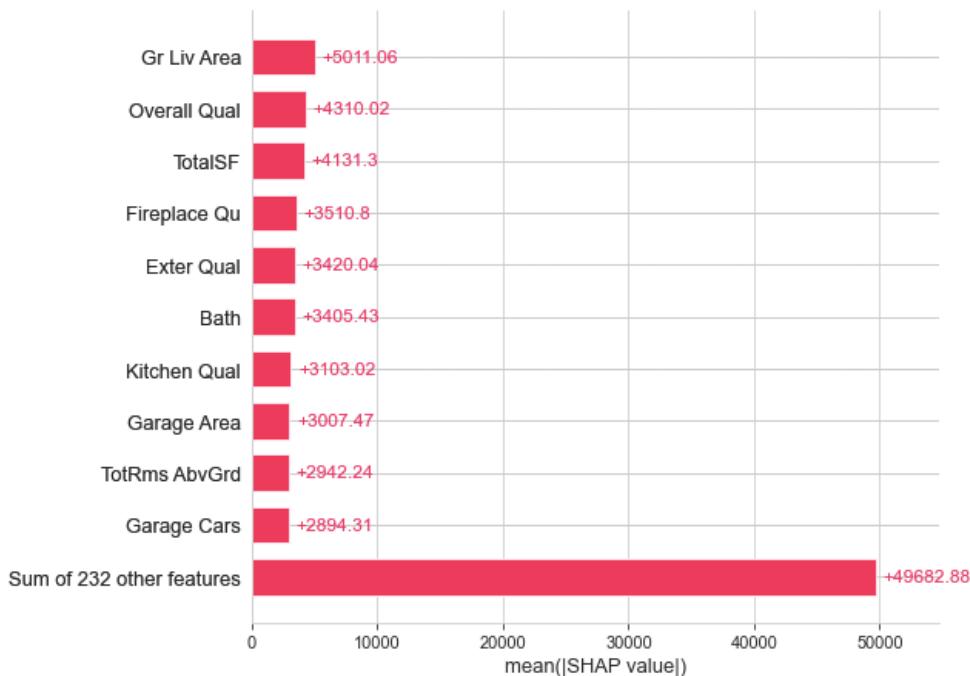


Abbildung 33: SHAP Global Balken-Diagramm KNN-Modell

Nachfolgend findet die Betrachtung des Beeswarm-Diagramms in der Abbildung 34 statt und lässt erkennen, dass für alle zehn einflussreichen Variablen ein höherer tatsächlicher Wert auch zu einem höheren SHAP-Wert führt. Dies impliziert einen höheren positiven Effekt auf die Preisvorhersage. Im Gegensatz führen besonders bei den Variablen *Overall Qual*, *Kitchen Qual*, *TotRms AbvGrd*, *Garage Area* und *GarageCars* auch einige niedrige Werte zu hohen negativen SHAP-Werten, die das prädiktive Ergebnis schmälern. *TotRms*

AbvGrd weist dabei eine weitere Besonderheit auf, bei der ein tatsächlich hoher Wert zur Schmälerung der Preisvorhersage führt. Ein genauerer Blick auf die Variablen *Overall Qual* lässt dabei erkennen, dass der Übergang von einem niedrigen tatsächlichen Wert zu einem hohen tatsächlichen Wert durch einen Schwellenwert getrennt wird, der durch den farblichen Übergang gekennzeichnet ist. Diese Beobachtung lässt sich ebenso für die Variablen *Gr Liv Area*, *TotalSF*, *Kitchen Qual*, *Bath*, *TotRms AbvGrd*, *Garage Area* und *Garage Cars* übertragen. Die Variablen *Fireplace Qual* und *Exter Qual* lassen keinen Übergang nach dem Beeswarm Plot erkennen und das Modell unterscheidet hierbei eindeutig nach hohen und niedrigen tatsächlichen Werten. Zusammenfassend lässt sich für beide Diagramme festhalten, dass höhere tatsächliche Werte der Flächenmaße, Qualitäts-eigenschaften und die Zimmeranzahl der Immobilie, das Modell als stark positiv wertet und die Preisvorhersage global betrachtet erhöhen wird.

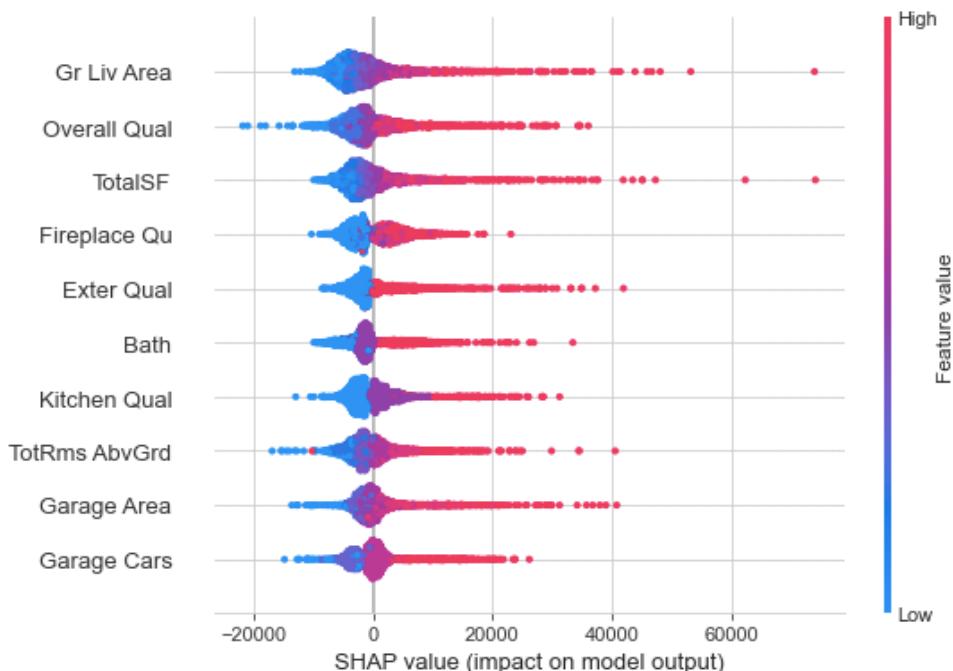


Abbildung 34: SHAP Global Beeswarm-Diagramm KNN-Modell

Nachfolgend findet anhand der Abbildung 35 die Untersuchung der Beobachtung 1 statt. Dabei lässt sich der vorhergesagte Preis anhand von $f(x)$ entnehmen und beträgt 139.833,33 \$. Dabei verfehlt das KNN-Modell den tatsächlichen Preis um knapp 5.000 \$. Der ausgehende Basiswert beträgt 172.968,73 \$ und wird zunächst am stärksten durch die Variablen *Year Remod/Add*, *Garage Yr Built*, *Fireplaces* und *Overall Qual*, um insgesamt 23.964,72 \$ gemindert. Dies umfasst aufgerundet 14 % des Basiswertes und senkt

die Preisvorhersage deutlich. Einzeln betrachtet, senkt das Modell die Preisvorhersage der Beobachtung 1 aufgrund des Modernisierungsjahres und des Baujahres der Garage von 1950, des nicht vorhandenen Kamins und der Gesamtqualität, die bei 5 liegt. Nachfolgend senken ebenfalls die durchschnittliche Kellerqualität und die unfertige Garage den Basiswert. Den einzigen positiven Einfluss üben neben der Aufnahmekapazität der Garage und der Garagenfläche, die Anzahl der Gesamtzimmer aus. Dabei beträgt die Zimmeranzahl acht und dies führt zu einem Anstieg von 2.830,71 \$. Abschließend tragen die übrigen Variablen einen kumulierten negativen Effekt in Höhe von 11.715,39 \$ auf $E[f(x)]$ aus. Dies verdeutlicht, wie die kumulierten negativen Auswirkungen in der Beobachtung 1 den stärksten Einfluss darstellen und konkludent zu dem vom Modell ermittelten Vorhersagepreis führen.

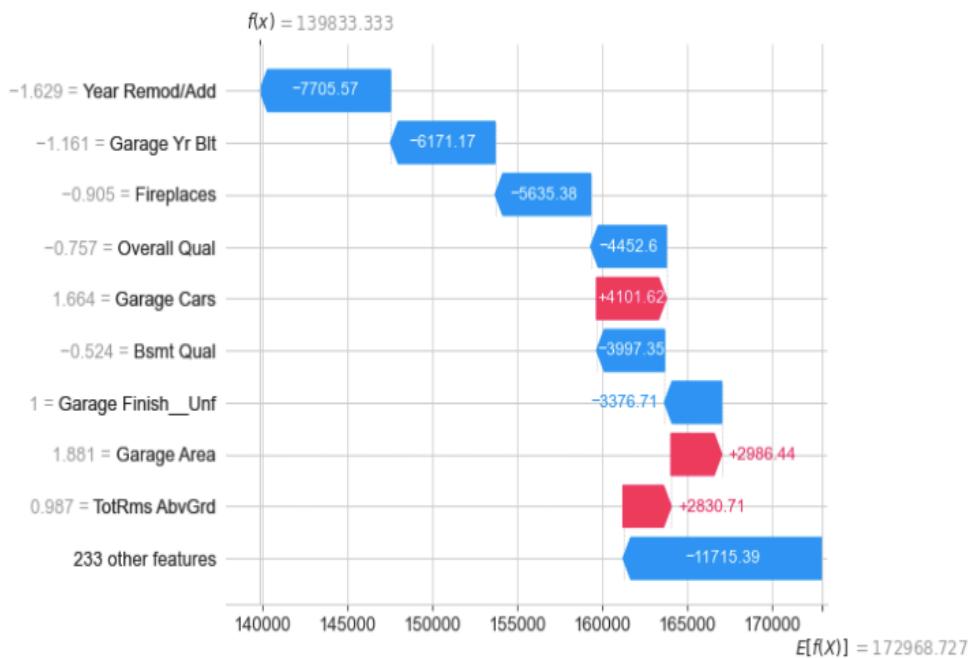


Abbildung 35: SHAP Lokal Wasserfall-Diagramm KNN-Modell Beobachtung 1

Die Beobachtung 2 der Abbildung 36 wird seitens des Modells mit einem Immobilienpreis von 217.150 \$ vorhergesagt und liegt demnach 74.150 \$ über dem tatsächlichen Preis. Auffällig zeigt sich dabei erneut die Variable *Fireplaces*, die nun im Gegensatz zur Beobachtung 1, den Basiswert um 16.486,13 \$ erhöht. Die Küchenqualität, die Anzahl der Badezimmer und die Gesamtfläche erhöhen den Basiswert nochmals um kumuliert 14.573,18 \$. Die Fläche des ersten Stockwerkes ,*Ist Flr SF*, führt erneut zur Erhöhung des prädiktiven Ergebnisses, neben der Erkenntnis, dass die Qualität des Kellers ebenso

mit einem positiven Beitrag zur Erhöhung führt. Zur Senkung der Preisvorhersage führt laut SHAP die Aufnahmekapazität der Garage eines Autos und die Außenqualität der Immobilie. Neben der Außenqualität schmälert auch die Gesamtqualität den Basiswert signifikant.

Die übrigen Variablen verbessern die Preisvorhersage nochmals um 16.191,75 \$. Zusammenfassend betrachtet bestätigen die Beobachtungen 1 und 2 ebenso den Eindruck der globalen SHAP-Analyse in der Hinsicht, dass nicht einzelne einflussreiche Variablen im KNN-Modell die Treiber der Preisvorhersage darstellen, sondern der kumulierte Gesamteinfluss aller Variablen. Die zehn einflussreichsten Variablen geben einen guten Indikator für die Entwicklung der Preisvorhersage ab, jedoch haben die übrigen Variablen einen maßgeblichen Anteil an der finalen Ergebnisfindung des Modells für die Beobachtungen 1 und 2.

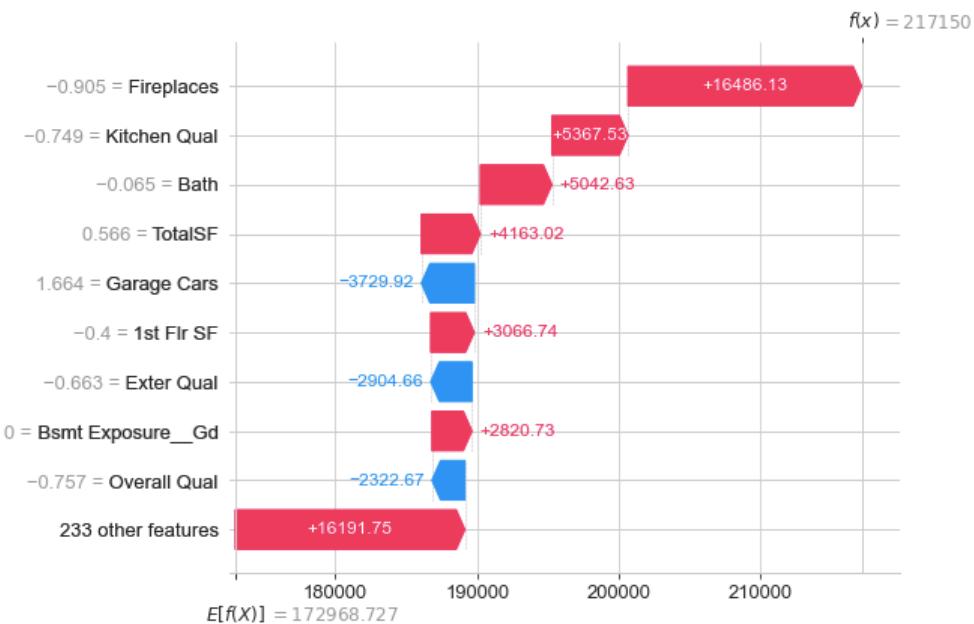


Abbildung 36: SHAP Lokal Wasserfall-Diagramm KNN-Modell Beobachtung 2

Zusätzlich wird das KNN-Modell unter Anwendung von LIME darauf untersucht, wie sich die fehlende Standardisierung von Daten auf die Ergebnisfindung des Modells auswirkt. Der vorhergesagte Preis beträgt 462.308,33 \$ und übertrifft den tatsächlichen Preis deutlich. Bereits in den Ergebnissen der Beobachtung 1 in der Abbildung 37 lässt sich beobachten, dass die numerischen Variablen einen überdurchschnittlich großen Einfluss

auf das Vorhersageergebnis haben. Gerade die Variablen *Garage Area*, *Garage Year Built* und die Qualitätsmetriken tragen am stärksten zum Vorhersagepreis bei.

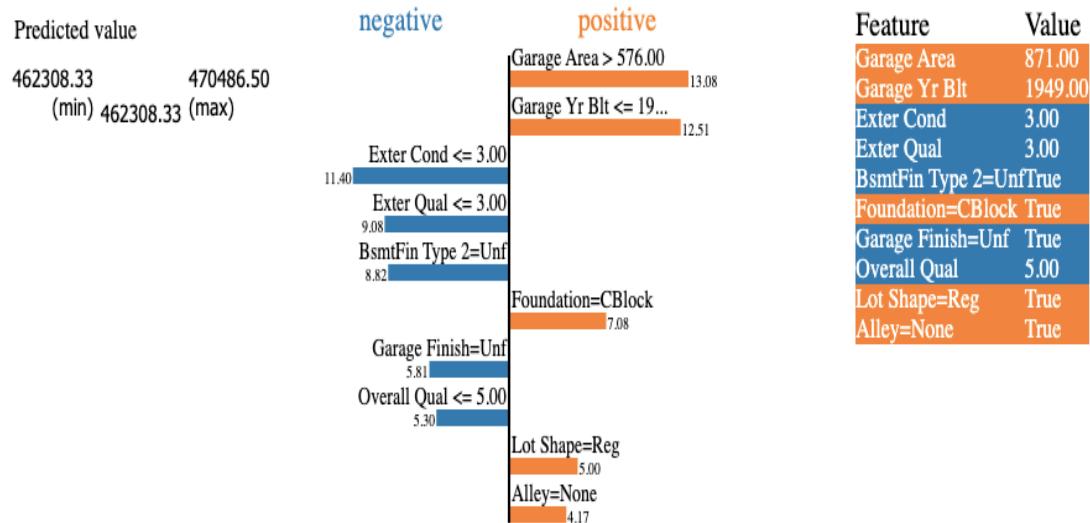


Abbildung 37: LIME Global nicht standardisierte Daten KNN-Modell Beobachtung 1

Die Vermutung liegt nahe, dass die nicht-standardisierten Daten zu diesen verzerrten Ergebnissen führen und der distanzbasierte Ansatz des KNN-Modells, die Verzerrung der Ergebnisse verstärkt. Die Ergebnisverzerrung lässt sich gerade für das KNN beobachten, wenn numerisch große Unterschiede vorherrschen. Zum Abgleich werden die globale SHAP- und InterpretML-Analyse in der Abbildung 38 hinzugezogen. Dabei zeigt sich, dass auch in den beiden anderen Verfahren die quantitativen Variablen die stärkste Gewichtung einnehmen.

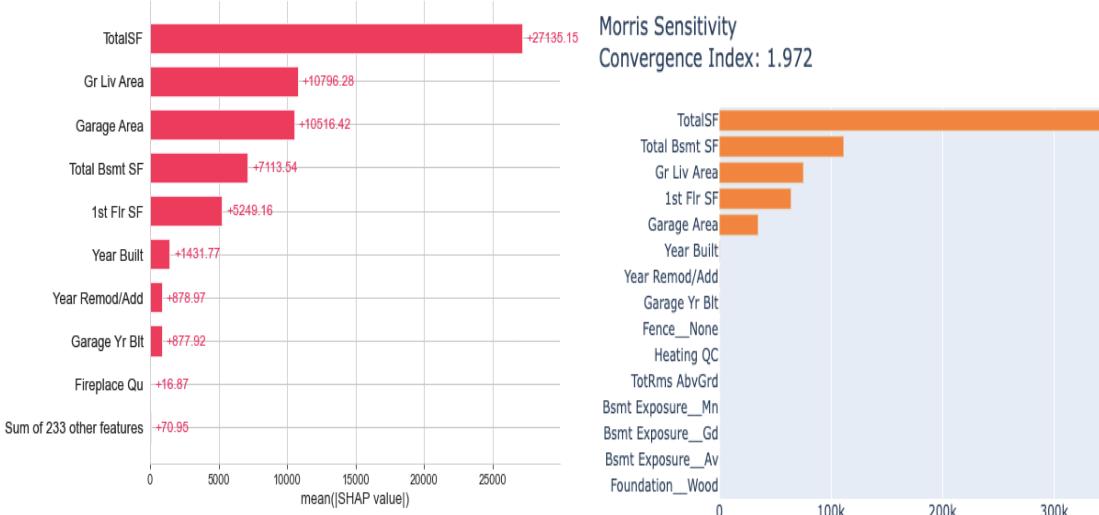


Abbildung 38: Überprüfung nicht standardisierter Daten mittels SHAP und InterpretML

Anschließend werden die Daten gemäß dem beschriebenen Vorgehen in Kapitel 4.3 wieder für die Untersuchung vorbereitet. Im Rahmen der globalen Analyse für das KNN-Modell beträgt die Berechnungszeit 40 Minuten und die Ergebnisse sind in Abbildung 39 dargestellt. Dabei wird eindeutig der Gesamtfläche *TotalSF* die höchste Gewichtung im KNN-Modell zugesprochen. Daraufhin erzielen die Variablen *Gr Liv Area* und *Garage Qual* quantitativ gesehen nur 1/4 des Beitrages von *TotalSF*. Die durchschnittliche Gewichtung der nachfolgenden Variablen weisen keine signifikanten Unterschiede auf. Auffällig ist dabei, wie stark der Beitrag der Variable *TotalSF* nach LIME ausfällt. Dabei werden nach mehrmaligen Wiederholungen der globalen Analyse ähnliche aber nicht identische Ergebnisse seitens LIME angezeigt. Im Vergleich zu SHAP weichen die Ergebnisse beider Methoden stark voneinander ab. LIME weist eindeutig nur einige Variablen eine hohe Einflussnahme zu, während SHAP die numerischen Einflüsse den Variablen ohne gravierenden Beitragsdifferenzen zuordnet. Lediglich die starke Einflussnahme der Variable *TotalSF* wird seitens beider Methoden korrespondierend hoch bewertet.

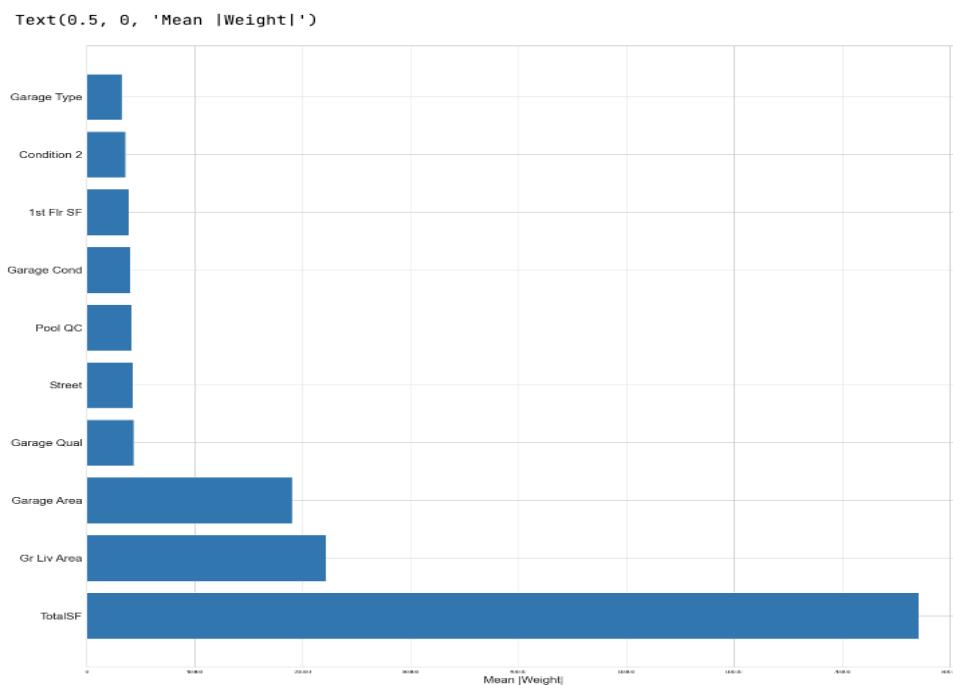


Abbildung 39: LIME Global Balken-Diagramm KNN-Modell

Die Ergebnisse der lokalen Beobachtung können der Abbildung 40 entnommen werden. Die Berechnung einer Beobachtung, inklusive des Zuweisens der transformierten Variablen zu Ihrer Ursprungsform, beträgt insgesamt nur eine Minute. Die Beobachtung 1

wird nach der Berechnung von LIME am stärksten durch die Variable *Pool QC* beeinflusst, die hierbei keinen vorhandenen Pool als Merkmalsausprägung impliziert. Dabei wird das prädiktive Ergebnis um 47.780 \$ gesenkt, während es durch die Flächengröße oberhalb des Kellers um 16.191,91 \$ erhöht wird. Neben der Variable *Gr Liv Area*, die den größten positiven Einfluss ausübt, erhöhen *TotalSF* und *TotRms AbvGrd* die Preisvorhersage. Jedoch lässt sich bereits anhand der ersten zehn Variablen erkennen, dass vermehrt negative Beiträge den Vorhersagepreis mindern. Dabei sind die Qualitätseigenschaften *Overall Qual*, *Kitchen Qual*, *Exer Qual* und *Fireplace Qual*, neben dem Modernisierungsjahr und der Anzahl an Badezimmern, besonders präsent. Auffällig ist dabei der Einfluss, der der Variable *Pool QC* zugeschrieben wird.

Nachfolgend weist LIME der Variable *Pool QC* eine ähnliche Einflussnahme in der Beobachtung 2 zu. Hierbei senkt das Fehlen eines Pools das prädiktive Ergebnis um 46.860,41 \$. Darüber hinaus wird deutlich, dass nach LIME mehr als zwei Badezimmer, eine höhere Küchenqualität als fünf und mehr als einen Kamin mit einer höheren Qualität als 1, als positiv gewertet wird. Die negativen Auswirkungen auf die Preisvorhersage ergeben sich durch die Gesamtqualität, die Außenqualität des Gebäudes, der Aufnahmekapazität an Autos der Garage, die Anzahl an Räumen oberhalb des Kellers und die Kellerqualität. Auffällig dabei ist, dass die zehn einflussreichsten Variablen der Beobachtungen 1 und 2 sich nach der bisherigen Analyse gleichen, unabhängig von ihren positiven oder negativen Beiträgen. Ebenso geht in beiden Beobachtungen der stärkste Einfluss von der Variable *Pool QC* aus. Zur Überprüfung werden weitere Beobachtungen des Trainingsdatensatzes betrachtet. Diese weisen ebenso einen starken Einfluss der Poolqualität aus und stellen einen gravierenden Unterschied zu den globalen Ergebnissen von SHAP dar.

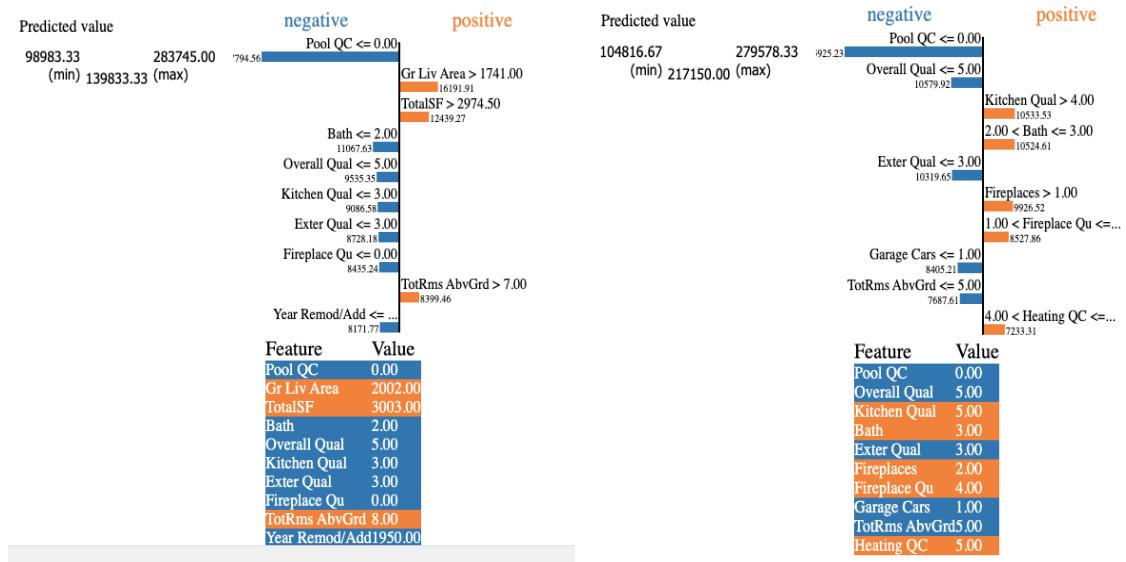


Abbildung 40: LIME Lokal Wasserfall-Diagramm KNN-Modell Beobachtung 1+2

Die globale InterpreML Analyse startet mit dem Morris-Sensitivity-Diagramm in der Abbildung 41. Auffällig dabei erscheint die hohe Einflussnahme der quantitativen Variablen. Die einflussreichste Variable ist die Variable *Pool QC*. Dabei kann die Poolqualität einen Betrag von bis zu 110.880 \$ für die Vorhersage beitragen. Daraufhin folgen die Flächenmaße *TotalSF*, *Total Bsmt SF*, *Gr Liv Area* und *Ist Flr SF*, die das prädiktive Ergebnis in einem Bereich von 58.000 \$ bis zu 90.000 \$ beeinflussen können. Dabei nimmt die Einflussnahme von Variable zu Variable quantitativ gesehen ab. Die Variable *Overall Qual* ist die letzte Variable, die einen Einfluss von über 40.000 \$ auf das prädiktive Ergebnis ausübt, während die nachfolgenden Variablen von Platz sieben bis zehn einen nahezu identischen Einfluss auf die Vorhersageentscheidung des Modells nehmen. Im Vergleich zu SHAP und LIME sind einige Gemeinsamkeiten aber auch Unterschiede zu erkennen. Zwar ordnen alle globalen Methoden *TotalSF* innerhalb der drei einflussreichsten Variablen ein, jedoch lässt sich ebenso erkennen, dass die Verteilung und die Reihenfolge der einflussreichen Variablen eine starke Unterscheidung vorweisen.

Morris Sensitivity
Convergence Index: 2.148

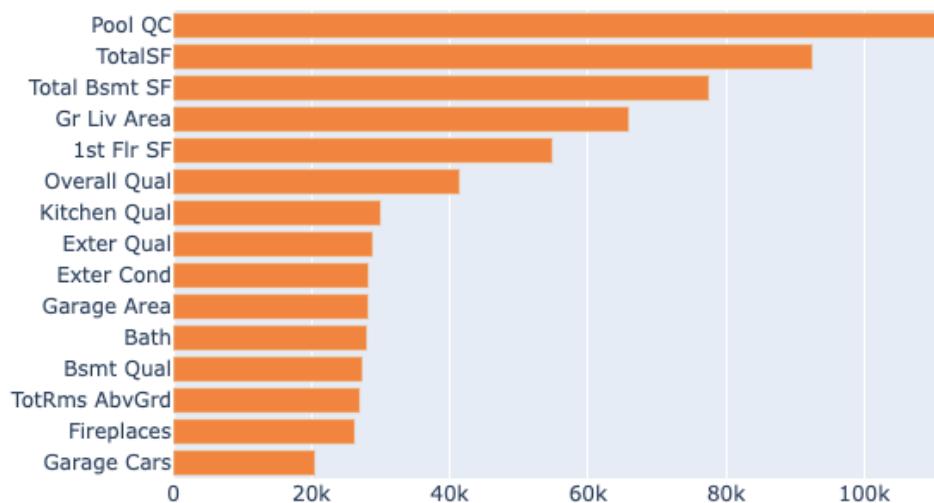


Abbildung 41: InterpretML Global Balken-Diagramm KNN-Modell

Die globale Analyse wird mittels der PDP-Analyse fortgesetzt, indem die drei einflussreichsten Variablen untersucht werden. Die jeweiligen Ausprägungen der Variablen bilden dabei die standardisierten Werte ab. Die Abbildung 42 zeigt dabei die Entwicklung der Einflussnahme der Poolqualität. Auffällig ist dabei, dass die wenigen hoch bewerteten Pools, die durchschnittliche Preisvorhersage so beeinflussen, dass aus einem anfangs parallelen Verlauf des Preises auf der Höhe von 200.000 zur x-Achse, ein stark ansteigender Preisanstieg erfolgt. Hierbei lässt sich festhalten, dass die Preisvorhersage des KNN-Modells stark die Average Response des Preises steigt. Dies lässt sich beobachten, sobald ein Pool vorhanden ist und die Qualität des Pools hoch bewertet wird. Hierbei ist es essenziell zu beachten, dass in dem Datensatz vorhandene Pools als Ausreißer identifiziert sind. Demnach sind in dem Datensatz Immobilien mit Pools als selten zu verstehen. Hierbei wäre es notwendig zu hinterfragen, wie mit der Variable und den Ausprägungen umzugehen ist. Zum Vergleich enthält die Abbildung ebenso die Gesamtqualität, die mit Zunahme der Qualität, eine lineare Steigerung der durchschnittlichen Rückmeldung der Zielvariable darstellt.

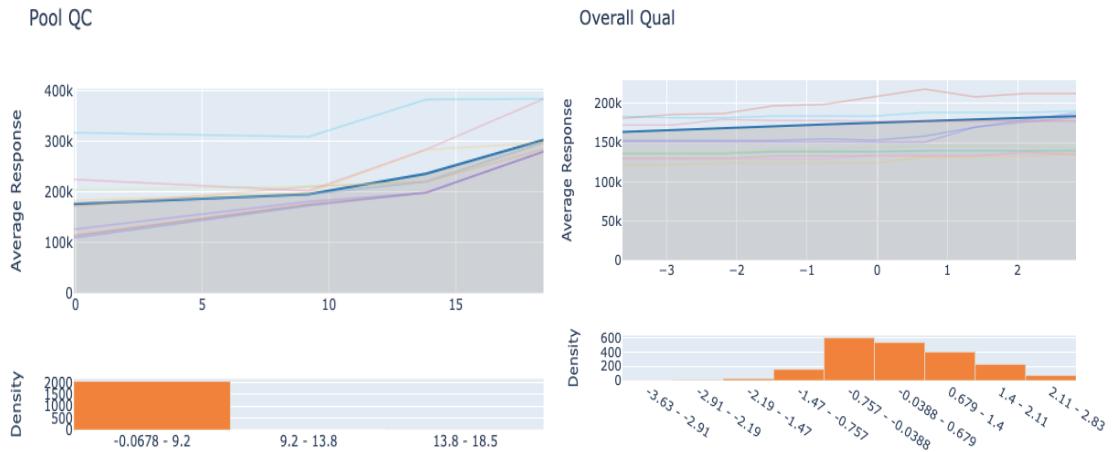


Abbildung 42: InterpretML Global PDP-Analyse PoolQC und Overall Qual

Nachfolgend wird die Abbildung 43 für die weitere PDP-Analyse hinzugezogen. Die Variablen *TotalSF* und *Total Bsmt SF* weisen einen linearen Anstieg auf, wobei die Gesamtfläche ab dem standardisierten Wert 2 einen steileren Anstieg vollzieht und somit die durchschnittliche Rückmeldung der Zielvariable steigert. Demnach beeinflussen beide Variablen den Vorhersagepreise stärker positiv, sobald diese ebenso höhere Werte aufweisen. Dies gilt ebenso für die zuvor beschriebene Variable *Pool QC*. Dies bedeutet, dass nach dem PDP eine lineare Beziehung zu den drei einflussreichsten Variablen besteht und somit höhere Poolqualitäten, größere Gesamtflächen und größere Kellerflächen zu einem höheren Preis führen. Nichtsdestotrotz ist dies eine isolierte Betrachtung des gesamten KNN-Modells, da die übrigen Variablen ebenso einen Einfluss auf die Zielvariable und untereinander in Form einer Multikollinearität ausüben können.

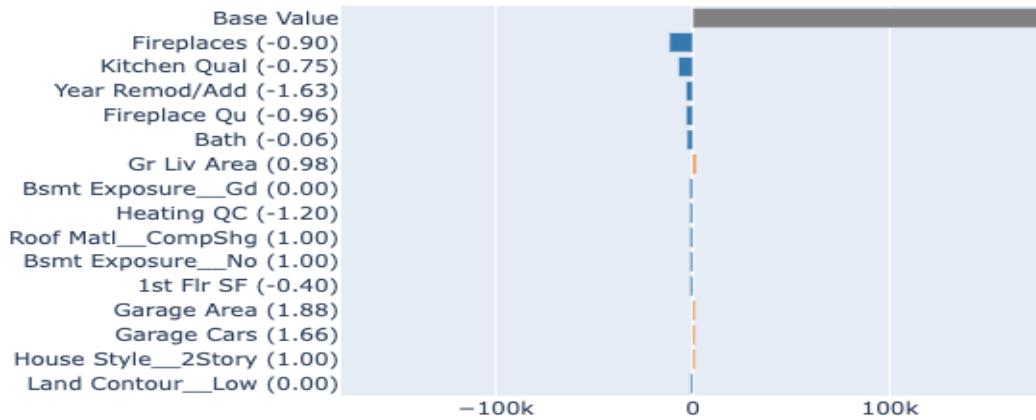


Abbildung 43: InterpretML Global PDP-Analyse TotalSF und Total Bsmt SF

Letztlich beginnt die lokale Betrachtung des KNN-Modells mittels des InterpretML Moduls. Die Wasserfall-Diagramme nutzen den LimeTabular oder KernelExplainer als Erklärmethode. Der KernelExplainer unterscheidet sich in der Form vom normalen SHAP Explainer, dass ein gewichteter Überlagerungsansatz verwendet wird, um regionale Interpretationen zu berechnen. Hierbei werden gewichtete und lokale Interpretationen berechnet, die den Einfluss jeder Eingabe auf die Vorhersage berechnen (Lundberg, 2018a, S.1; Nori et al., 2019, S.2-3).

Zunächst werden die SHAP-Werte beider Beobachtungen anhand des KernelExplainer berechnet. Die Berechnungszeit kann bis zu zehn Minuten beanspruchen. Die Beobachtungen 1 und 2 der Abbildung 44 unterscheiden sich in einigen Punkten zu den Ergebnissen des Explainers von SHAP. Bereits der Basiswert beträgt 17.8491,70 \$ und liegt somit um knapp 6.000 \$ höher als beim SHAP-Explainer. Am signifikantesten ist dabei die Einordnung des KernelExplainers der Variable *Fireplaces*. Der nicht vorhandene Kamin in der Beobachtung 1, wird nach dem KernelExplainers vom Modell als negativer Faktor angesehen und senkt demnach den Basiswert um 12.736 \$. Die zwei vorhandenen Kamine in der Beobachtung 2 erhöhen den Basiswert um 12.507 \$. Hierbei ist die Anzahl an Kaminen als einflussreichste Variable für beide Beobachtungen zu verzeichnen. Dabei ist für beide Dateninstanzen zu beobachten, dass neben dem Kamin, die Gesamtqualität die zweiteinflussreichste Variable darstellt. Diese übt sowohl für die Beobachtung 1 einen negativen Einfluss auf das prädiktive Ergebnis aus als auch einen positiven für die Beobachtung 2. Auffällig ist dabei erneut die starke Gewichtung des Modells von quantitativen Variablen. Dies lässt sich auch im Vergleich zu LIME und den SHAP Explainer Ergebnissen wiedererkennen. Gleichheit besteht ebenso in der Aufführung des Einflusses der Qualitätseigenschaften der Explainer. Hierbei führt der KernelExplainer zusätzlich die Qualität des vorhandenen Heizungstypen für beide Beobachtungen auf. Generell lässt sich festhalten, dass der KernelExplainer die Entscheidungen des KNN-Modells zur Preisvorhersage ähnlich wie der SHAP-Explainer und LIME darstellt, es jedoch für einzelne Variablen Unterschiede in der Ausweisung der Stärke der Einflusswirkung gibt.

Predicted (140K) | Actual (145K)



Predicted (217K) | Actual (143K)

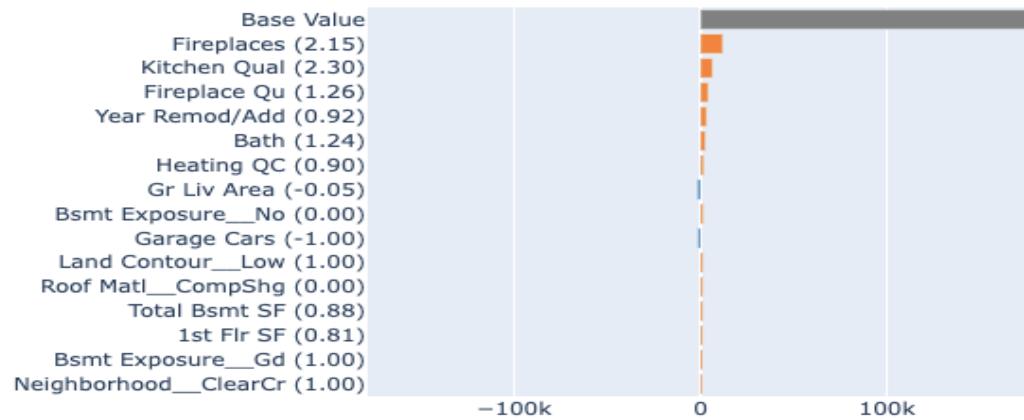


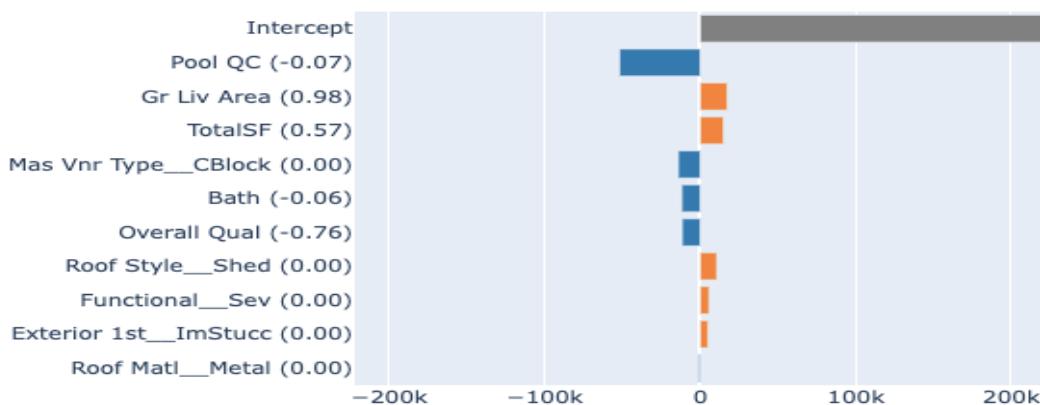
Abbildung 44: InterpretML Lokal Wasserfall-Diagramm KNN-Modell Beobachtung 1+2

Die zweite lokale Beobachtung des InterpretML mittels LIME erzielt verzerrte Ergebnisse wie es bereits im Kapitel vorab beschrieben ist und anhand der Abbildung 45 beobachtet werden kann. Dies resultiert jedoch aus der Gegebenheit, dass das spezielle Feature Encoding der kategorialen Variablen noch nicht für InterpretML adaptiert ist.

Hierbei zeigt sich beispielsweise an der Beobachtung 1, dass selbst verzerrte Ergebnisse nicht immer eindeutig zu erkennen sind. Die drei einflussreichsten Variablen werden im Vergleich zu den anderen Erklärungen, kontextuell gesehen, realistisch gewichtet. Demnach senkt der nicht vorhandene Pool den Vorhersagepreis, während die Fläche oberhalb des Kellers und die Gesamtfläche die Preisvorhersage erhöhen. Ebenso können die aufgezählten kategorialen Variablen in Verbindung mit den quantitativen Variablen in der

aufgeführten Konstellation einen sinnvollen Erklärungsansatz darstellen. Die Betrachtung der Beobachtung 2 zeigt dabei eindeutig, welche Ergebnisse seitens LIME ausgegeben werden, wenn die kategorialen Variablen durch das One-Hot Encoding, unbehandelt als eigene Variablen betrachtet werden. Damit stellen sich die Ergebnisse als unbrauchbar dar und werden demnach nicht weiter beschrieben. Nachfolgend werden die lokalen Beobachtungen für die restlichen Modelle mittels des InterpretML ausschließlich mithilfe des KernelExplainers durchgeführt.

Predicted (140K) | Actual (145K)



Predicted (217K) | Actual (143K)

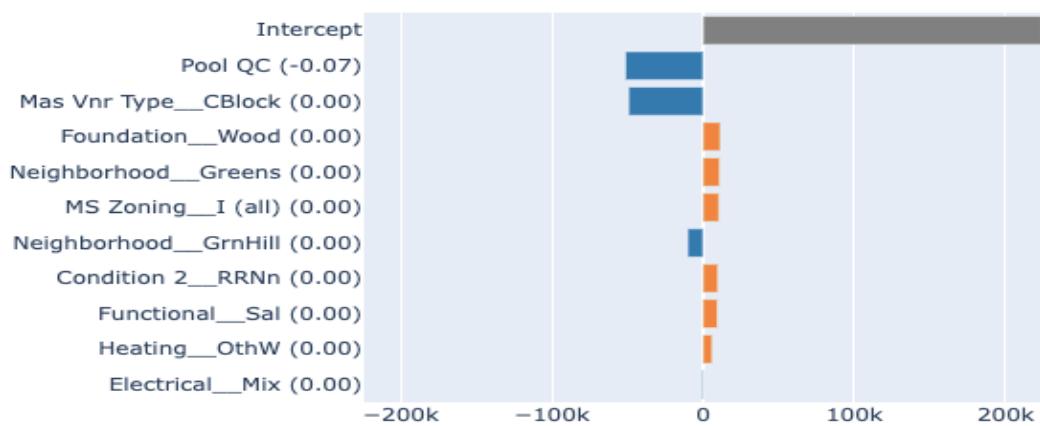


Abbildung 45: InterpretML LIME Lokal Wasserfall-Diagramm KNN-Modell Beobachtung 1+2

5.3 Untersuchung der XAI-Methoden im Rahmen von XGBoost

Im Rahmen der Anwendung von SHAP auf das XGB-Modell wird die TreeExplainer Methode angewendet. Diese wurde speziell für baumartige Algorithmen entwickelt und

basiert auf dem herkömmlichen Explainer, welcher für das ANN- und KNN-Modell verwendet wird. Der Unterschied besteht darin, dass die Komplexität der exakten Shapley Werte reduziert wird. Dadurch nimmt die Berechnungszeit nicht, wie beim Explainer, exponentiell zu, sondern erfolgt innerhalb eines polynomiellen Zeitraums. Demnach ermöglicht der TreeExplainer die Berechnung der exakten SHAP-Werte in einem stark verkürzten Zeitraum, indem die interne baumartige Struktur des Algorithmus genutzt wird (Lundberg et al., 2020, S.61). Die Untersuchung der SHAP-Methode startet mit der Berechnung der Shapley Werte durch den TreeExplainer. Bevor der TreeExplainer vollenfänglich genutzt werden kann, ist es notwendig den XGB-Parameter *tree_method* auf *exact* zu definieren, damit nach jeder Abzweigung des Baumalgorithmus alle Variablen erneut iterativ durchlaufen werden, bevor es erneut zu einer Abzweigung kommt. Dies erhöht zwar die Berechnungszeit des XGB-Modells, ermöglicht jedoch im Nachhinein ein präziseres Modell und die gesamtheitliche Berechnung der Shapley Werte für jede Variable. Die Berechnungszeit für die 2.051 Beobachtungen beträgt 30 Sekunden. Bevor die globale und lokale Analyse der Vorhersage startet, werden vorab mittels des Explanation-Befehls die Basis Werte berechnet, die für die lokale Analyse essenziell sind. Die Untersuchung von SHAP für das XGB-Modell wird ebenfalls zunächst mit der globalen Analyse beginnen. Dabei zeigt sich, dass für die globale Interpretation des XGB-Modells hauptsächlich quantitative Variablen ausschlaggebend für die Ergebnisfindung sind. Dabei können bereits die Variablen *TotalSF* und *Overall Qual* in der Abbildung 46 zusammen das prädiktive Ergebnis um $\pm 34.804,88$ \$ beeinflussen. Gefolgt werden die 2 einflussreichsten Variablen von *Year Built*, *Bath* und *Kitchen Qual*, die sich im Rahmen von ± 4.400 \$ bis ± 4.800 \$ bewegen und zusammengenommen einen Einfluss von circa ± 13.500 \$ auf das prädiktive Ergebnis ausüben. Außerdem fällt besonders der numerische Abstand von *Year Built* zu *Overall Qual* auf. Demnach übt global gesehen die Variable *Overall Qual* fast 4-mal mehr Einfluss auf das prädiktive Ergebnis aus als die Variable *Year Built*. Wird der Abstand von *Year Built*, *Bath* und *Kitchen Qual* hin zu *TotalSF* betrachtet, so fällt der Einfluss nach SHAP sogar fünfmal so groß aus. Nachfolgend sind die nächsten vier einflussreichsten Variablen *Gr Liv Area* mit $\pm 3.227,23$ \$, *Year Remod/Add* mit $\pm 3.040,66$ \$, *Garage Cars* mit $\pm 2.735,37$ \$ und *Fireplaces* mit $\pm 2.250,56$ \$. Letztlich bildet der zehnte Platz die Variable *Bldg Type_1Fam*, diese übt lediglich einen Einfluss von 1.727,41 \$ aus. Auffällig hierbei ist, dass für den XGB-Algorithmus sowohl

das Baujahr der Immobilie als auch das Modernisierungsjahr unter den zehn einflussreichsten Variablen der globalen Betrachtung fällt. Hierbei wäre es für weitere globale Betrachtung interessant, ob das Baujahr und das Modernisierungsjahr auf die Variable *Overall Qual* einen Einfluss haben.

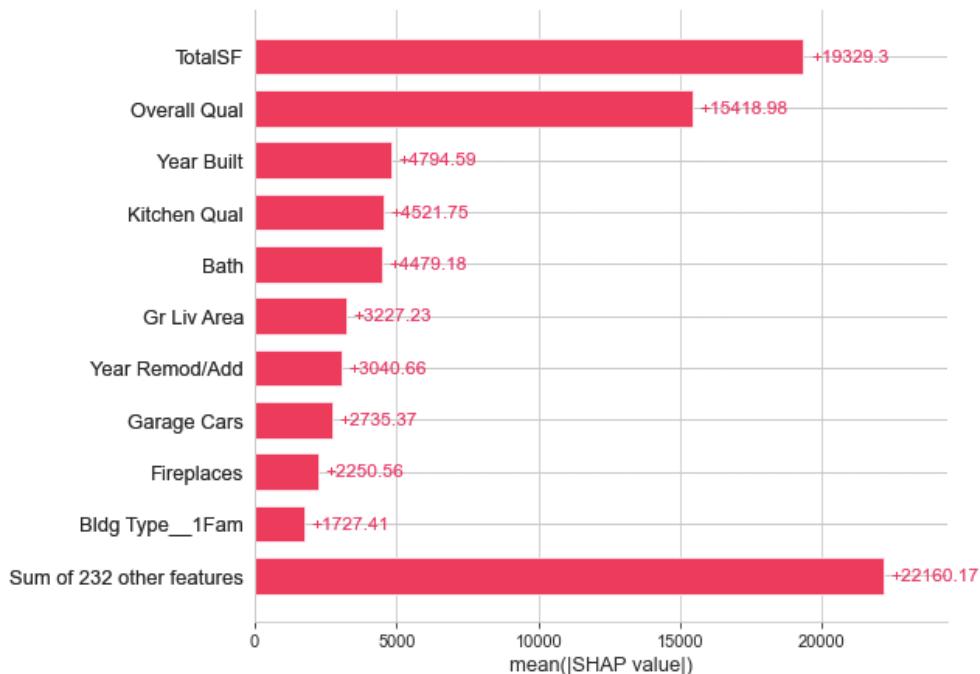


Abbildung 46: SHAP Global Balken-Diagramm XGB-Modell

Nachfolgend wird das Beeswarm-Diagramm in der Abbildung 47 betrachtet. Dabei kann für die zehn einflussreichsten Variablen festgehalten werden, dass die höheren tatsächlichen Werte der Variablen auch einen hohen positiven Shapley-Wert vorweisen. Diese haben somit einen stärkeren positiven Einfluss auf das prädiktive Ergebnis des Modells. Gerade die Entwicklung der Variablen *TotalSF* und *Overall Qual* indizieren eindrucksvoll, wie die Werte der Variablen mit der Zunahme des SHAP-Wertes zunimmt. Auffällig im Diagramm erscheinen dabei die Variablen *Garage Cars*, *Bath* und *Fireplaces*. Die Entwicklung von *Garage Cars* lässt erkennen, dass zwar die höheren Werte auch einen höheren SHAP-Wert aufweisen, jedoch trifft dies nicht für alle Werte zu. Bei genauerem Blick auf der Höhe der 0 der x-Achse lässt sich erkennen, dass auch einige rote Datenpunkte einen niedrigen positiven Shapley Wert aufweisen. Eine Besonderheit stellen dabei die Variablen *Year Built* und *Year Remod/Add* dar. Hierbei zeigt sich, dass je jünger die Immobilie ist, desto höher fällt der positive Shapley Wert aus und beeinflusst das Ergebnis positiv. Gegensätzlich bedeutet dies ebenso, dass je älter die Immobilie ist desto

stärker wirkt sich der negative Effekt auf die Preisvorhersage aus. Dabei fällt ebenso auf, dass sich Datenpunkte um den Shapley Wert 0 stauen. Dies kann implizieren, dass gewisse Jahresspannen keinen positiven oder negativen Effekt auf die Vorhersage ausüben. Ein ähnliches Bild ergibt sich für das Modernisierungsjahr *Year Remod/Add*. Nur die Anzahl an geringen tatsächlichen Werten mindert den negativen Einfluss, den die Variable auf das Ergebnis hat. Gruppiert betrachtet bilden Flächenmaße wie *TotalSF* und *Gr Liv Area* ebenso einen starken Einflussfaktor für die Vorhersage des XGB-Modells, wie die Qualitätseigenschaften *Overall Qual* und *Kitchen Qual*. Letztlich spielt der Gebäudetyp *Bldg Type_1Fam*, der ein Einfamilienhaus impliziert, eine wichtige Rolle für die Vorhersage des XGB-Modells. Isoliert betrachtet, bleibt die Preisvorhersage in Ihrem Wert konstant, wenn der Gebäudetyp ein Einfamilienhaus ist und sinkt, wenn der Gebäudetyp keinem Einfamilienhaus entspricht.

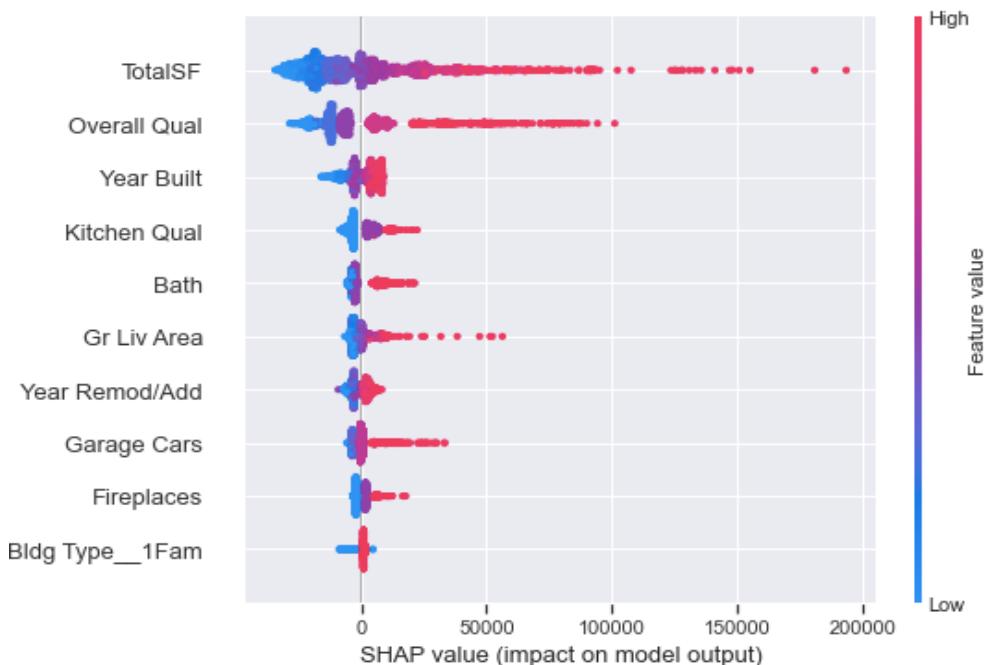


Abbildung 47: SHAP Global Beeswarm-Diagramm XGB-Modell

Eine besondere Möglichkeit des SHAP TreeExplainers stellt dabei das sogenannte Dependence-Diagramm dar. Dieses kann ebenfalls zur globalen Analyse genutzt werden und dient als Alternative zum PDP. Der Unterschied zum PDP zeigt sich, indem SHAP mithilfe der y-Achse die Varianz der Datenpunkte abbildet. Hierbei wird dem Nutzer ermöglicht, zu erkennen, wie sich im Rahmen einer sogenannten Interaktionsbedingung die Betrachtung auf eine Variable auswirken kann (Lundberg, 2018b, S.1-3). Dadurch lässt sich

analysieren, wie sich die Veränderung der tatsächlichen Werte auf die Vorhersage des Modells auswirkt und welche Zusammenhänge zwischen den Variablen bestehen. Dabei wird der Effekt, die die Interaktionsbedingung auf die Wichtigkeit der Variable hat, durch die vertikale Verteilung der Datenpunkte festgehalten. Die Abbildung 48 skizziert das Dependence-Diagramm der Variable *TotalSF* und deutet darauf hin, dass bei einer Größe von knapp 4.000 Quadratfuß, die Erhöhung des Vorhersagepreises für einige Immobilien bei 25.000 \$ liegt und für wiederum andere Immobilien circa 80.000 \$ entspricht. Die y-Achse auf der linken Seite des Diagramms zeigt die tatsächlichen Werte an. Durch die Analyse dieser Achse kann man erkennen, wie sich die Veränderung der tatsächlichen Werte auf die Vorhersage des Modells auswirkt. Die unterschiedliche Höhe des Beitrages unterliegt dem Einfluss, die die übrigen unabhängigen Variablen auf die Gesamtfläche für die Preisvorhersage haben. Zusätzlich wird die Variable *Overall Qual* hinzugezogen, um den Interaktionseffekt zu verdeutlichen. Der Beitrag einer Gesamtfläche von 4.000 Quadratfuß steigt, wenn die Gesamtqualität zunimmt. Dies eröffnet die Möglichkeit, Interaktionseffekte der Variablen mittels SHAP zu analysieren.

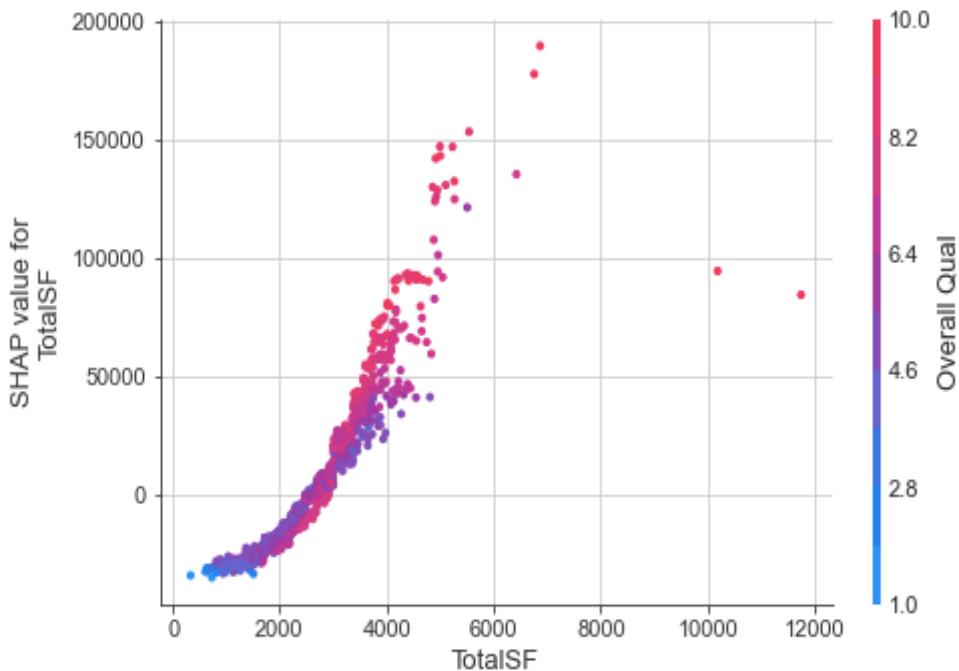


Abbildung 48: Histogramm Verkaufspreises im Bezug zur Bauart und Gesamtqualität

Die lokale Analyse der ersten Beobachtung beginnt mit dem vorhergesagten Preis, der bei $f(x) = 1447.10,92 \$$ liegt und somit den tatsächlichen Preis von 145.000 \$ bisher am nächsten kommt. Dabei wird laut SHAP ein Basis Wert von 176.652,11 \$ berechnet. Eine

Besonderheit spiegelt sich dabei in der Anzeige der tatsächlichen Werte der Variablen wider. Dadurch, dass der XGBoost Algorithmus keine Standardisierung oder Normalisierung der Daten benötigt, können die Werte bereits nach erfolgtem Feature Engineering in Test- und Trainingsdaten aufgeteilt werden. Dies lässt sich der Abbildung 49 abnehmen, bei dem der tatsächliche Wert 5 der Variable *Overall Qual* zur Minderung des Basis Wertes von 15.975,10 \$ führt. Der Durschnitt der Gesamtqualität beträgt dabei 6 und mit der Bewertung 5 die innerhalb des 25 % - 50 % Quartils liegt, ist eine Minderung von 159.85,35 \$ bereits beachtlich. Die Zugehörigkeit der Immobilie zur Nachbarschaft Crawford erhöht die Preisvorhersage um 11.475,60 \$, während gleichzeitig der Basiswert um insgesamt 26.840,17 \$ durch *Year Remod/Add*, *Bldg Type_1Fam*, *Central Air_N* und *Kitchen Qual* gesenkt wird. Dabei wird für den Einfluss der Variable *Year Remod/Add* deutlich, dass das Modernisierungsjahr, welches ein Jahr nach der Erbauung stattfand, zu lange in der Vergangenheit zurück liegt und somit seitens des Modells als negativer Einfluss gewertet wird. Ebenso entspricht die Immobilie keinem Einfamilienhaus und weist keine Zentrallüftung aus, welches das Modell als bauliche negative Eigenschaften einstuft. Ebenso wird die Küchenqualität als negativer Faktor gewertet. Im Vergleich zur Durchschnittsbewertung der Küchenqualität von 3,5, ist es signifikant, dass die preisschmälerung für den Basispreis bei einer Bewertung von 3 in Höhe von 6483,85 \$ ausfällt. Zur Erhöhung der Preisvorhersage führt die Aufnahmekapazität der Garage von Autos, der Lebensbereich von 2002 Quadratfuß und die Gesamtfläche von 3.003 Quadratfuß. Auffällig hierbei ist, dass *TotalSF* nur einen positiven Zuwachs von 4.542,40 \$ für das prädiktive Ergebnis verzeichnet, obwohl die Fläche knapp über dem 75 % Quartil liegt. Bereits die globale Betrachtung deutet darauf hin, wie stark der Einfluss der tatsächlichen hohen Werte der Variable *TotalSF* nach SHAP sein kann, die sich jenseits des 75 % Quartils befinden und es fortwährend zu beachten gilt. Ein ähnlicher Sachverhalt lässt sich für *Gr Liv Area* feststellen. Generell lässt sich für *TotalSF*, *Gr Liv Area* und *Garage Cars* festhalten, dass die tatsächlichen Werte der Variablen überdurchschnittlich sind und im Verhältnis der Gesamtqualität einen kleineren Effekt auf die Preisvorhersage erzielen. Die nachfolgenden Variablen schmälern kumuliert den Basiswert nochmals um 15.860,14 \$ und üben dementsprechend einen ähnlich starken Einfluss auf die Ergebnisfindung des Modells aus, wie die Gesamtqualität *Overall Qual*.

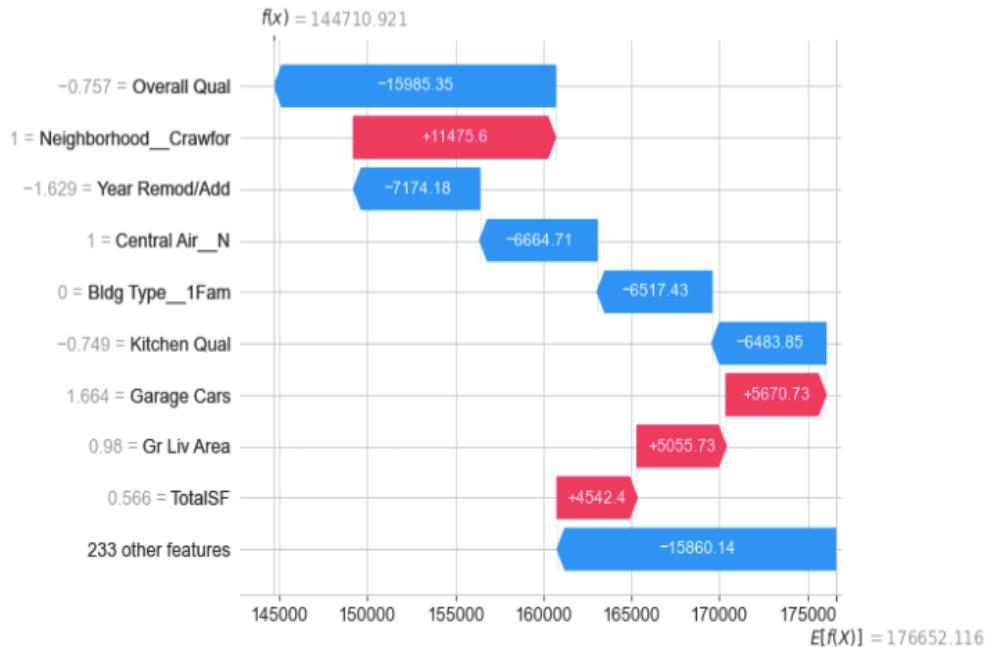


Abbildung 49: SHAP Lokal Wasserfall-Diagramm XGB-Modell Beobachtung 1

Die Verdeutlichung der größeren Einflussnahme der negativen Shapley-Werte auf die Ergebnisfindung der Beobachtung 1 wird durch die Abbildung 50 bestätigt. Das Force Plot Diagramm lässt dabei erkennen, wie stark die negativen Shapley Werte das Ergebnis entgegen des Basiswert verschieben. Gerade die ersten 4 Variablen annullieren den gesamten Effekt der positiven Shapley-Werte verhältnismäßig und drücken den Basiswert in Richtung des prädiktiven Ergebnisses von 144.343,73 \$.



Abbildung 50: SHAP Lokal Force-Plot XGB-Modell Beobachtung 1

Die zweite lokale Beobachtung wird durch das XGB-Modell mit einem Vorhersagewert von $f(x)=191.444$ \$ prognostiziert. Diese verfehlt den tatsächlichen Vorhersagepreis um 48.444 \$. Das Wasserfall-Diagramm der Abbildung 51 stellt die Ergebnisfindung des Modells nachvollziehbar dar. Dabei schmälert zunächst die Gesamtqualität *Overall Qual* den Basiswert, während die Anzahl von zwei Kaminen, die Einstufung des Kellers als

Lebensraum, die Gesamtgröße von 2.910 Quadratfuß und die Badezimmeranzahl die Minderung ausgleicht und zusätzlich $E[f(X)]$ um knapp 9.500 \$ erhöht. Die Variable *Garage Cars* hat im Rahmen der zehn einflussreichsten Variablen der Beobachtung 2 nochmals einen negativen Effekt von 4.658,15 \$, bevor die starke Neigung des Bodens und die höchste Bewertung der Küchenqualität das Ergebnis positiv beeinflussen. Die übrigen Variablen senken kumuliert den Basiswert um 1633,96 \$. Das hinzuziehen der globalen Analyse des Modells deutet bereits darauf hin, dass unter den 6 Variablen, die positive SHAP-Werte in der Beobachtung aufweisen, 5 global stark einflussreiche Variablen vertreten sind. Gerade die Variable *Fireplaces* mit der Ausprägung 2 ist gerade einmal 221-mal im gesamten Datensatz vertreten. Dies bewertet das Modell als starken positiven Einfluss mit einem quantitativen Einfluss von +7.325,87 \$. Diese Beobachtung trifft ebenfalls auf die Variable *Bsmt Exposure_Gd* zu. Hinzukommt, dass die Gesamtfläche über dem Durchschnitt liegt, genauer gesagt 80 Quadratfuß vor dem 75 % Quartil entfernt ist und nach SHAP als positiver Faktor gewertet wird. Letztendlich lassen sich nur Vermutungen anstellen, weshalb der Preisunterschied so groß ausfällt. Die Ergebnisse aller Algorithmen werden im Rahmen des Kapitels 6.5 nochmals verglichen und hinsichtlich der Beitragshöhen und gewichteten Variablen genauer analysiert.

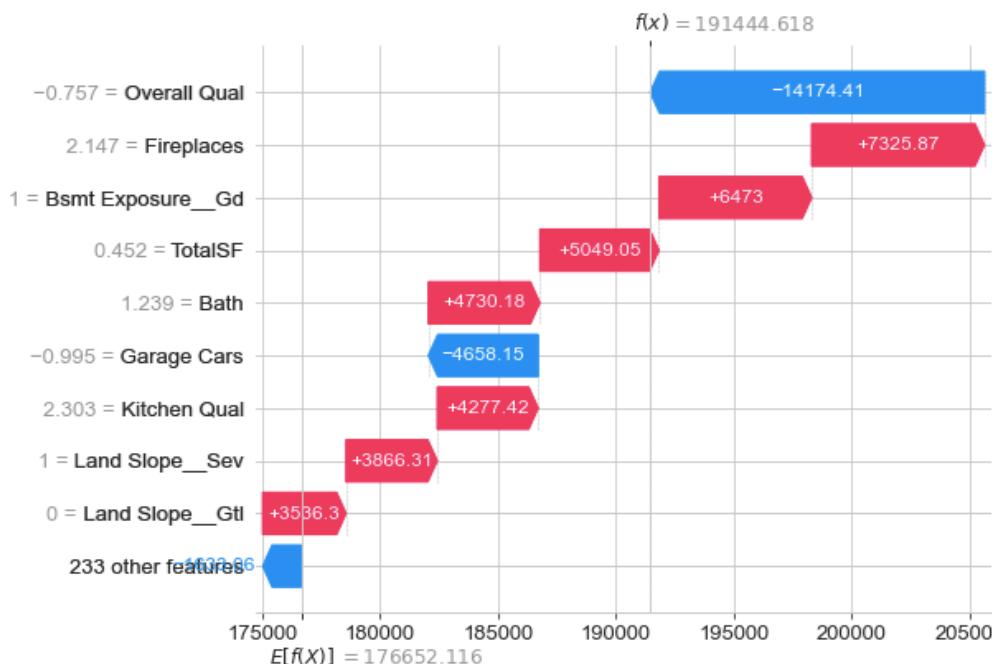


Abbildung 51: SHAP Lokal Wasserfall-Diagramm XGB-Modell Beobachtung 2

Nachfolgend beginnt die Untersuchung anhand von LIME durch die Betrachtung der Abbildung 52. Die Berechnungszeit für das XGB-Modell der globalen LIME-Ergebnisse beträgt 43 Minuten und die Ergebnisse weisen Ähnlichkeiten zu den SHAP-Erkenntnissen auf. Es lässt sich beobachten, dass die zwei einflussreichsten Variablen identisch sind. Dabei wird der Variable *TotalSF* die stärkste Gewichtung für das prädiktive Ergebnis zugesprochen. An zweiter Stelle steht die Gesamtqualität *Overall Qual*, die jedoch nur die Hälfte des numerischen Einflusses der Gesamtfläche nach LIME ausmacht. Die große Beitragshöhe, die *TotalSF* nach LIME attestiert wird, ist ebenso im Rahmen von SHAP zu beobachten. Ein wesentlicher Unterschied zur globalen SHAP-Analyse bilden dabei die Variablen *Neighborhood* und *Pool QC*. Dabei unterscheiden sich die Variablen in ihrer Gewichtung nur minimal und werden trotzdem seitens LIME innerhalb der 10 einflussreichsten globalen Variablen für das XGB-Modell aufgeführt. Die übrigen Variablen weisen eine ähnliche Einflussnahme nach LIME aus, die ebenfalls innerhalb von SHAP beobachtet wird. Der Anteil der Beitragshöhen der einzelnen Variablen weisen ebenso im Vergleich zu SHAP eine Ähnlichkeit auf. Lediglich der Beitrag der Gesamtqualität wird seitens SHAP höher eingestuft als durch LIME. Insgesamt bemisst LIME im Vergleich zu SHAP einen größeren möglichen durchschnittlichen Beitrag der einzelnen Variablen.

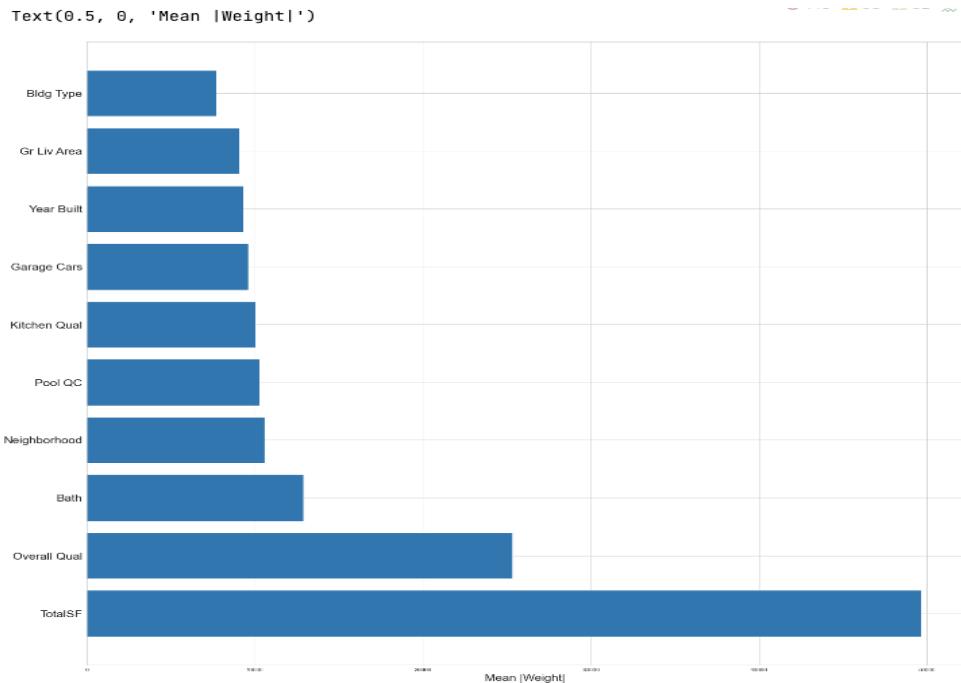


Abbildung 52: LIME Global Balken-Diagramm XGB-Modell

Nachfolgend werden ebenfalls die ersten beiden lokalen Dateninstanzen analysiert. Von essenzieller Bedeutung ist hierbei die korrekte Vorgehensweise des Feature Engineerings, welche bereits in Kapitel 4.3 und im Rahmen der LIME-Anwendung des EBM-Modells beschrieben wird. Die Zeitspanne von der Ausführung bis hin zur Visualisierung der LIME-Ergebnisse beträgt lediglich 1 Minute und 30 Sekunden. Dabei beeinflusst die Variable *TotalSF* die lokale Vorhersage am stärksten. Die Anwendung von LIME lässt in der Abbildung 53 erkennen, dass der tatsächliche Wert der Gesamtfläche größer als 2.974,50 Quadratfuß ist und dies die Preisvorhersage um 50.429 \$ erhöht. Den zweitgrößten Einfluss übt dabei nach LIME die Gesamtqualität *Overall Qual* aus, mit einem tatsächlichen Wert von 5. Dies hat auf das prädiktive Ergebnis einen negativen Effekt von -25.501,43 \$. Zu einem positiven Zuwachs führen unter den 10 einflussreichsten Variablen noch die Zugehörigkeit zur Nachbarschaft Crawford, die Möglichkeit der Beherbergung von drei Autos in der Garage und die Gesamtfläche in Höhe von 2.002 Quadratfuß, die zum Wohnen genutzt werden kann. Dabei impliziert LIME auch die Schwellenwerte, die für die Variable *Garage Cars* bei größer als 2 liegt und für *Gr Liv Area* größer als 1.741 Quadratfuß entspricht. Das Baujahr von 1949 wird seitens LIME als ein weiterer negativer Faktor bewertet und führt zu einer Minderung des prädiktiven Ergebnisses um 10.765,93 \$. Den Abschluss der zehn einflussreichsten Variablen bilden dabei der nicht vorhandene Pool, der Immobilientyp Duplex und die Küchenqualität. Insgesamt lässt sich für die Beobachtung 1 festhalten, dass nach LIME die Flächenmaße und die Lage der Immobilie die größten positiven Treiber der Vorhersage darstellen. Negativen Einfluss üben demnach am stärksten die Gesamtbewertungen, räumliche und bauliche Eigenschaften des Hauses sowie das Baujahr aus.

Die Beobachtung 2 muss ebenso separat nach Anpassungen des Codes berechnet werden und ist mit einer fast identischen Berechnungszeit zur ersten Beobachtung gleichzusetzen. Der Betrag der Preisvorhersage beträgt nach LIME 191444,81 \$. Dabei wird die Variable *Overall Qual* mit der Ausprägung 5 als einflussreichste Variable der Beobachtung 2 angegeben. Dies schmälert das prädiktive Ergebnis 25.216,45 \$. Nachfolgend führen die Anzahl von drei Badezimmern, das steile Grundstück, die höchste Küchenqualität, die normale Qualität der Außenfassade, die zwei vorhandenen Kamine und der Gebäudetyp, der einem Einfamilienhaus entspricht, zur Erhöhung des Vorhersagepreises des Modells. Der Zustand der Garage der mit einer 3 bewertet ist, führt zur Senkung des prädiktiven

Ergebnisses. Die bauliche Qualität der Garage und das Material des Daches, *Tar&Grv*, werden seitens des Modells ebenfalls als negativer Einfluss gewichtet.

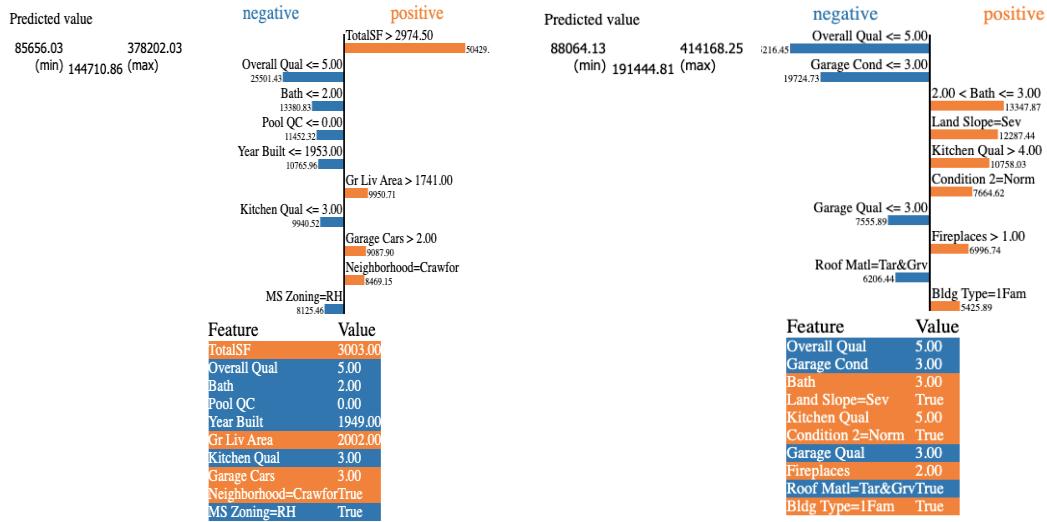


Abbildung 53: LIME Lokal Wasserfall-Diagramm KNN-Modell Beobachtung 1+2

Die Anwendung des InterpretML für den XGB-Algorithmus wird nachfolgend für die lokale und globale Erklärung der Ergebnisse gestartet. Die globale Analyse seitens des InterpretML Modells beginnt erneut mit der Morris-Sensitivity- und PDP-Analyse. Dabei liegt die Berechnungszeit bei lediglich 20 Sekunden. Die Ergebnisse der Morris-Sensitivity sind der Abbildung 54 zu entnehmen. Global gesehen, übt die Variable *TotalSF* den größten Einfluss auf das prädiktive Ergebnis aus und dies mit einer möglichen Beitrags Höhe von bis zu 143.000 \$. Mit einem deutlichen Abstand folgt als zweitgrößter Einfluss *Overall Qual* und darauf die Quadratfußanzahl des ersten Stockwerkes *Ist Flr SF*. Signifikant dabei ist der Abstand der möglichen numerischen Beeinflussung der Variablen. *Overall Qual* wird im XGB-Modell seitens der MS-Analyse ein möglicher Einfluss von bis zu 110.000 \$ zugesprochen. Die Variable *Ist Flr SF*, übt den drittgrößten Einfluss von knapp über 600.00 \$ aus. Die Sensitivitätsanalyse von Morris zeigt, dass die ersten beiden Variablen einen möglichen Einfluss von 253.000 \$ auf das prädiktive Ergebnis ausüben. Die anderen Variablen haben betragsmäßig einen Einfluss von 19.000 \$ bis 58.000 \$. Besonders auffällig ist die Gewichtung der Dachholzschindel *Roof Matl_WdShngl* und der Nachbarschaft *GrnHill*, die die Vorhersage des Preises in einem Bereich von 30.000 \$ bis zu 50.000 \$ beeinflussen können. Beide Variablen werden erstmalig in den globalen Ergebnissen aufgeführt, wobei *GrnHill* in den LIME-Ergebnissen bereits in der Variable

Neighborhood kumuliert, aufgeführt ist. Nichtsdestotrotz wird kategorialen Variablen im XGBoost-Modell nach der Sensitivitätsanalyse vermehrt ein höherer Einfluss eingeräumt.

Morris Sensitivity
Convergence Index: 0.269

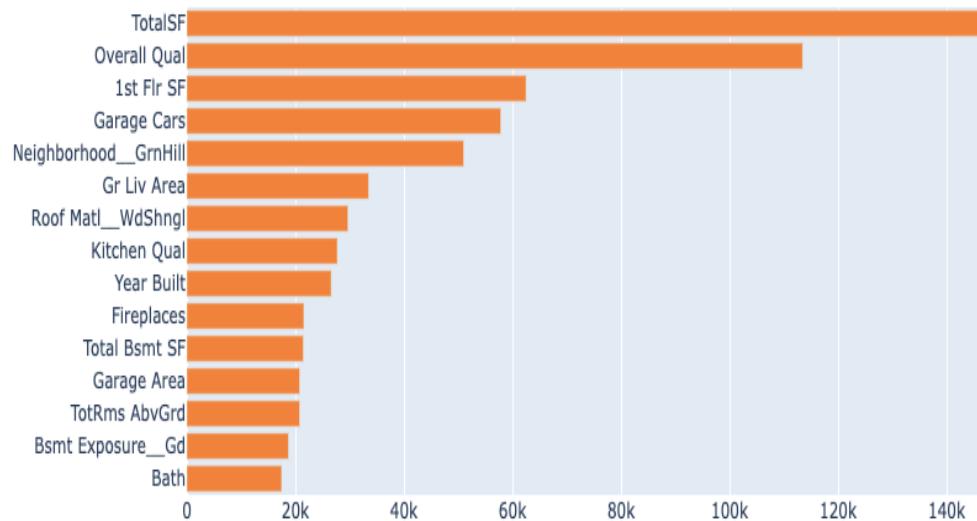


Abbildung 54: InterpretML Global Balken-Diagramm XGB-Modell

Die Nachfolgende PDP-Analyse fokussiert sich auch hierbei auf die drei einflussreichsten Variablen der vorherigen Analyse in den Abbildungen 55 bis 57. Dabei lässt sich für die Variable *TotalSF* festhalten, dass die Einflussnahme auf die Zielvariable *SalePrice* durchschnittlich am stärksten steigt, sobald die Quadratfußanzahl zwischen 2.200 und 3.500 liegt. Daraufhin steigt die Average Response des Verkaufspreises fortgehend leicht linear. Dabei ist zu beachten, dass bereits das darunterliegende Verteilungsdiagramm darauf hinweist, dass die Datenpunkte für die gesamte Quadratfußanzahl nach 3.501 signifikant abnehmen. Jedoch lässt das PDP-Diagramm darauf schließen, dass *TotalSF* einen Einfluss auf die Steigerung des Vorhersagepreises des Modells hat, sobald eine Immobilie mehr als 220 Quadratfuß als Gesamtfläche aufweist.

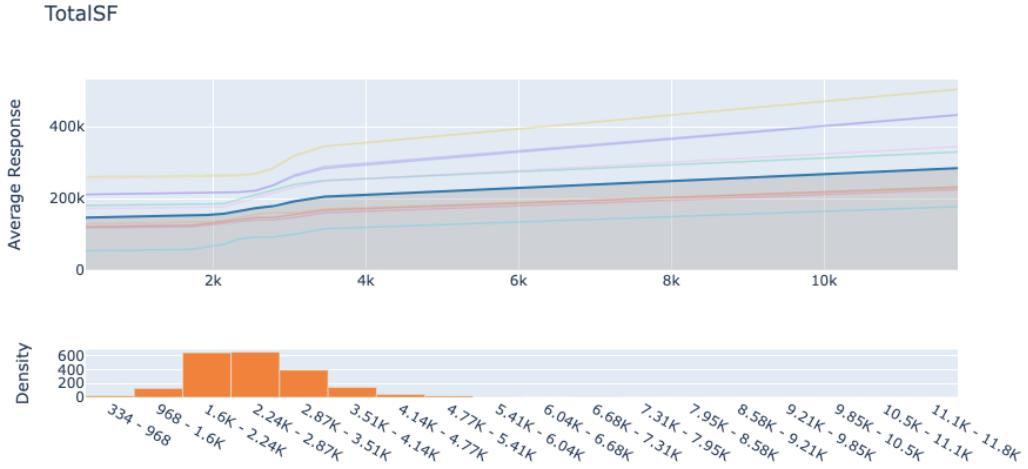


Abbildung 55: InterpretML Global PDP-Analyse TotalSF

Für die Gesamtqualität in der Abbildung 56 lässt sich ein leichter Anstieg ab der Bewertung 5 erkennen. Der stärkste Anstieg der Average Response ist nach der PDP-Analyse für *Overall Qual* von der Bewertung 6 bis 8 zu erkennen, welche auch den Bereich der meisten Datenpunkte abbildet. Demnach hat die Gesamtqualität erst ab einer Bewertung von 6 einen signifikanten Einfluss auf die Preisvorhersage des XGB-Modells. Diese steigt am stärksten im Bereich von 6 bis 8, bis die Steigung ab der Bewertung 8 leicht abnimmt.

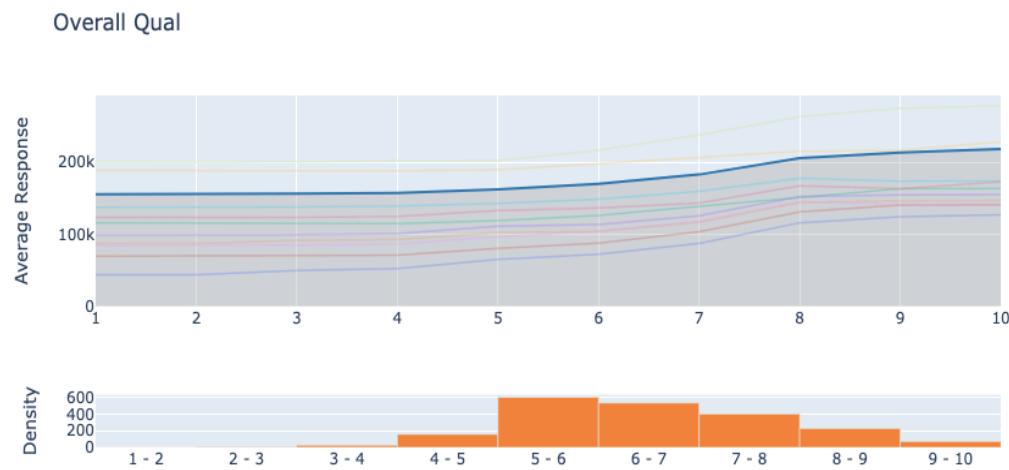


Abbildung 56: InterpretML Global PDP-Analyse Overall Qual

Überraschenderweise zeigt sich für *1st Flr SF* eine durchschnittlich gleichbleibende Average Response, obwohl es laut der globalen InterpretML Analyse den drittgrößten Einfluss ausüben soll. Lediglich die zwei Beobachtungen oberhalb der 200.000 Marke weisen darauf hin, dass ein Abfall des Preises ab einer Quadratfußanzahl von knapp 1.600

bevorsteht. Dies kann jedoch dadurch erklärt werden, dass die letzten Einträge im Bereich von 1.720-2.570 zu wenige Beobachtungen darstellen, um einen tatsächlichen Verlauf der Average Response nach 1.730 Quadratfuß darzustellen.

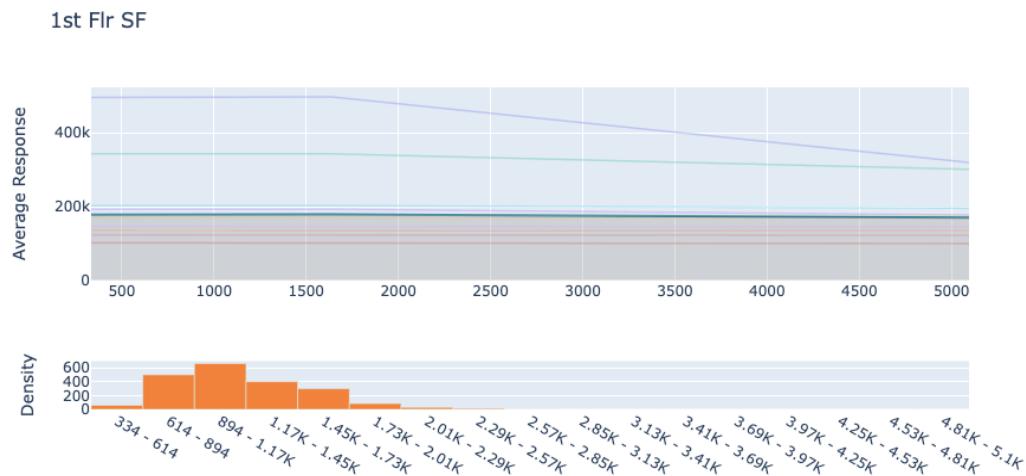
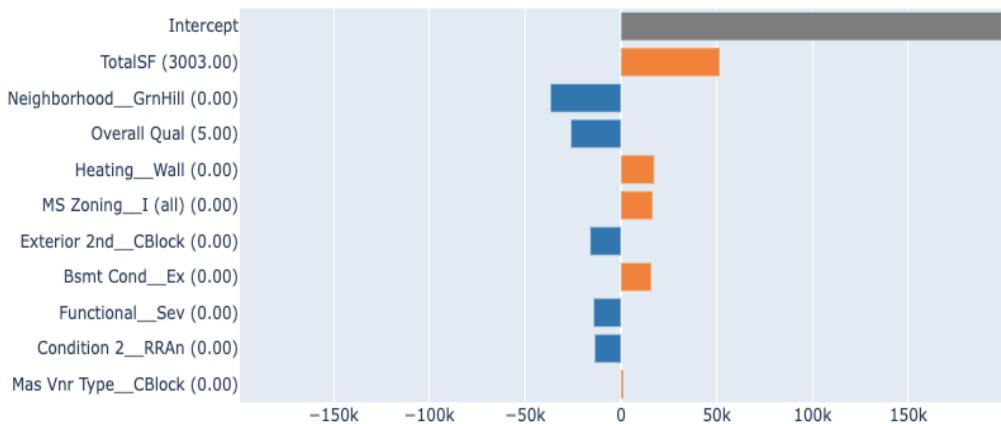


Abbildung 57: InterpretML Global PDP-Analyse 1st Flr SF

Dabei lässt sich auch für das XGBoost-Modell festhalten, dass die Ergebnisse der lokalen Beobachtung unter LIME nach InterpretML derzeitig keine zuverlässigen Ergebnisse zulassen. Dies ergibt sich aus der bereits angesprochenen Problematik zur Handhabung von kategorialen Variablen. Zur Veranschaulichung erfasst die Abbildung 58 dabei die lokale Beeinflussung der Variablen der Beobachtung 1 und 2. Auffällig für die Beobachtung 2 ist erneut die hohe Einflussnahme der kategorialen Variablen, die durch Ihre Ausprägung 0 auf einen nicht vorhandenen Zustand der Merkmalsausprägung hinweisen.

Predicted (146K) | Actual (145K)



Predicted (188K) | Actual (143K)

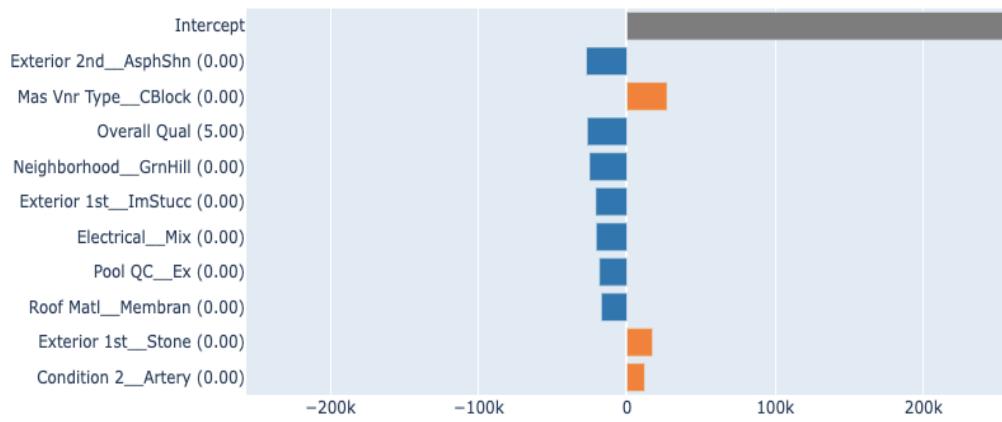
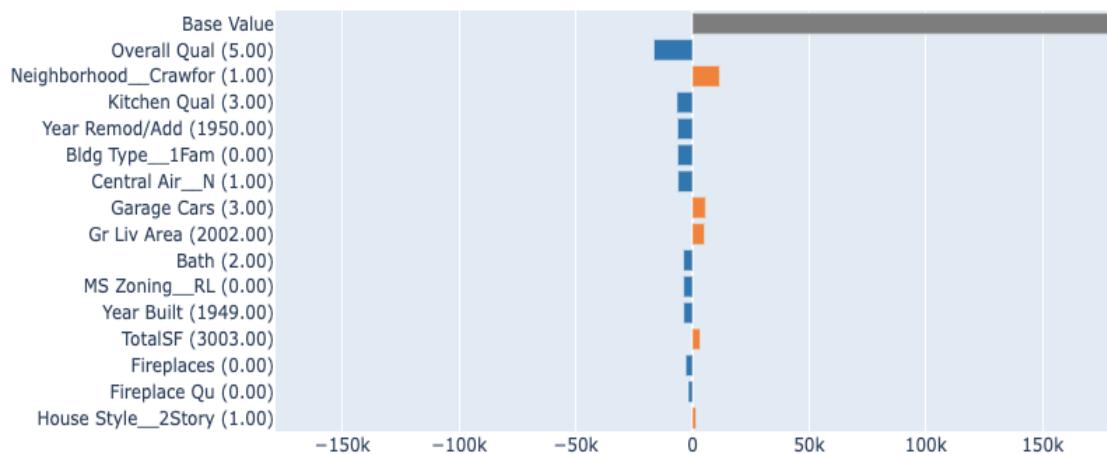


Abbildung 58: InterpretML LIME Lokal Wasserfall-Diagramm XGB-Modell Beobachtung 1+2

Nachfolgend werden in der Abbildung 59 die Ergebnisse der SHAP-Methode im Rahmen des InterpretML-Moduls dargestellt. Die Berechnung von zwei Beobachtungen beträgt hierbei 6 Minuten. Die Berechnung aller Beobachtungen mittels InterpretML nimmt dabei eine deutlich längere Zeit ein als die direkte Berechnung über das SHAP-Modul. Pro Beobachtung betrug die durchschnittliche Berechnungszeit drei Minuten. Demnach hätte eine Berechnung der SHAP-Werte aller 2.051 Beobachtungen einen Zeitaufwand von ungefähr 102 Stunden eingenommen. Dabei ist eindeutig zu erkennen, dass die Ergebnisse identisch zu den SHAP-Ergebnissen sind. Im internen Vergleich fällt dabei auf, dass der KernelExplainer die Variable *TotalSF* sowohl in der ersten Beobachtung als auch in

der Beobachtung 2 als weniger einflussreich einordnet als der TreeExplainer. Dabei wird der Küchenqualität bei der Ergebnisfindung ein höherer Einfluss zugeordnet. Die beiden einflussreichsten Variablen beider Beobachtungen sind im Vergleich zum TreeExplainer identisch und unverändert.

Predicted (145K) | Actual (145K)



Predicted (191K) | Actual (143K)

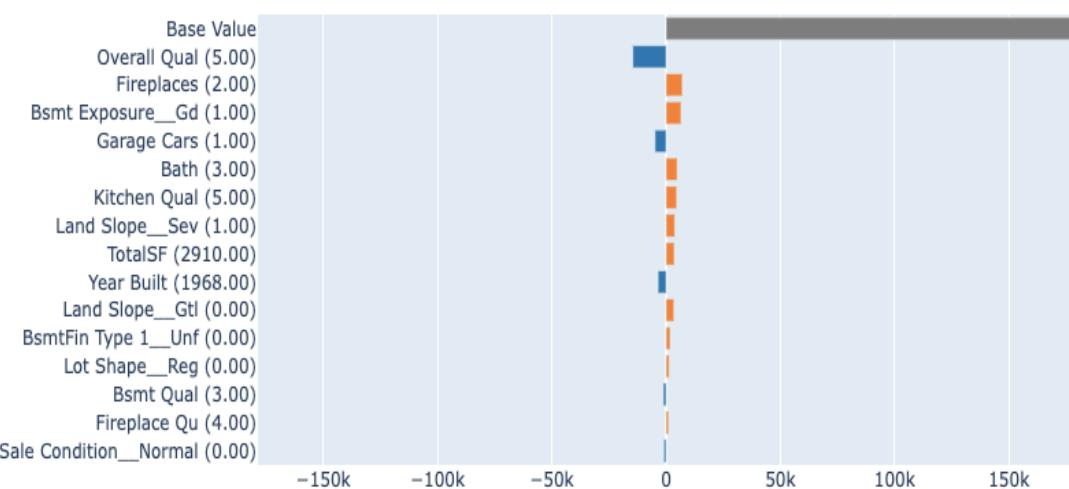


Abbildung 59: InterpretML Lokal Wasserfall-Diagramm XGB-Modell Beobachtung 1+2

5.4 Untersuchung der XAI-Methoden im Rahmen von ANN

Die Betrachtung des letzten Modells beginnt ebenfalls mit der Berechnung der SHAP-Werte. Auffällig hierbei ist die Durchlaufzeit von 1 Stunde und 12 Minuten, für insgesamt 2.051 Beobachtungen. Nachdem die SHAP-Werte erfolgreich berechnet sind, startet die globale Betrachtung der SHAP-Werte in der Abbildung 60. Sieben der zehn Variablen

sind dabei von quantitativer Natur und die Variablen *Bsmt Fin Type 1_GLQ*, *House Style_2Story* und *Garage Type_Attchd* spiegeln kategoriale Variable wider. Kontextuell gesehen beeinflussen der Keller, der einem Lebensraum gleicht, ein Zweifamilienhaus und eine Garage, die an die Immobilie gebaut ist, das prädiktive Ergebnis des Modells um kumuliert $\pm 10.784,49 \$$. Dabei zeigt sich, dass die Variable *Gr Liv Area* den größten Einfluss auf das prädiktive Ergebnis der Vorhersage ausübt und trägt durchschnittlich für jeden vorhergesagten Immobilienpreis einen Betrag von $\pm 11.604,97 \$$ bei. Nachfolgend hat *Overall Qual* mit einem Betrag von $\pm 8136,93 \$$ den zweitgrößten Einfluss auf den Preis. *TotalSF* verzeichnet mit $\pm 7985,08 \$$ den drittgrößten Einfluss. Nach der SHAP-Methode beeinflussen die zehn einflussreichsten Variablen die globale Vorhersage des Modells um insgesamt $\pm 54.147,95 \$$, was einem Anteil von 29 % der gesamten Beeinflussung entspricht. Für das ANN-Modell werden ebenso die Variablen betrachtet, die nach SHAP keinen Einfluss auf die Ergebnisfindung des Modells haben. Dabei machen die Variablen *Exterior 2nd_Other*, *Roof Matl_Roll*, *Pool QC_1*, *Sale Type_VWD*, *Misc Feature_Tenc*, *Exterior 2nd_PreCast* und *Exterior 1st_PreCast* einen Betrag von $\pm 0 \$$ für die globale Betrachtung der Vorhersage aus. Zu beobachten ist, dass die genannten Ausprägungen der Variablen selten im Datensatz vertreten sind und somit keinen Informationsgehalt für die Erklärung der Vorhersage nach SHAP liefern. Außerdem zeigt sich, dass bereits ab der Variable *BsmtFin Type 1_GLQ* der numerisch dargestellte Einfluss im Vergleich zu den drei einflussreichsten Variablen stark abnimmt. Die größte Feature Importance wird jedoch den übrigen 232 Variablen zugeschrieben. Diese wirken sich kumuliert mit $\pm 128.208,61 \$$ auf die Preisvorhersage aus.

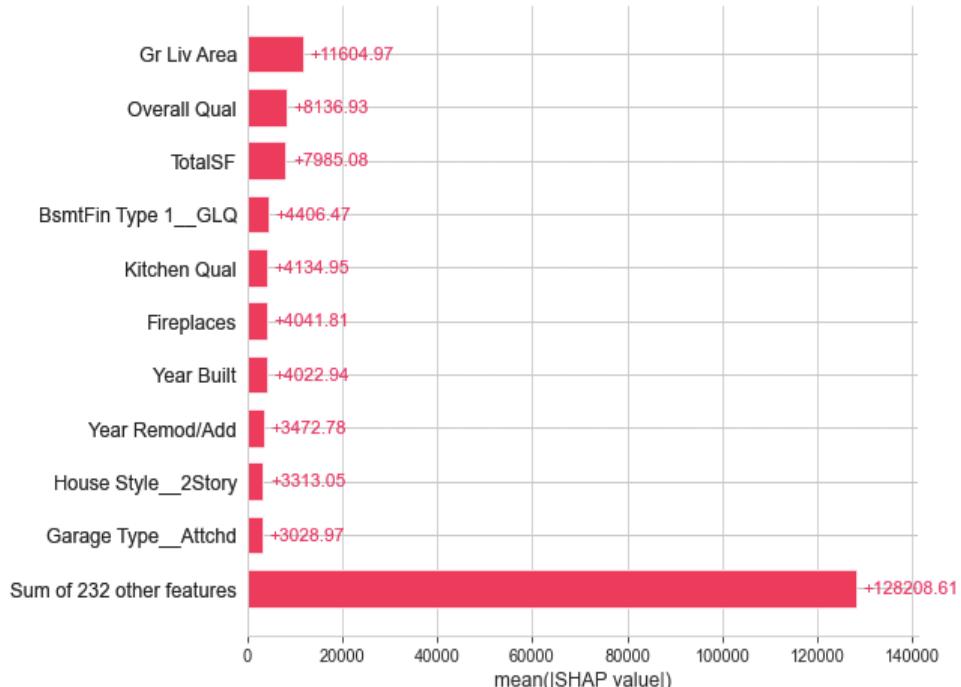


Abbildung 60: SHAP Global Balken-Diagramm ANN-Modell

Nachfolgend wird das Beeswarm-Diagramm die Anwendung von SHAP in der globalen Erklärbarkeit des ANN in der Abbildung 61 dargestellt. Hierbei lässt sich für das ANN-Modell festhalten, dass der Preis einer Immobilie zunimmt, sobald die Gesamtgröße des zu bewohnenden Grundstücks steigt. Dies trifft ebenso für die Variablen *TotalSF* und *Overall Qual* zu, die mit einem Zuwachs des numerischen Wertes auch zur Erhöhung des vorhergesagten Preises beitragen. Generell lässt sich festhalten, dass Flächenmaße und Qualitätseigenschaften, wie *Gr Liv Area*, *TotalSF*, *Overall Qual*, und *Kitchen Qual*, einen hohen Einfluss auf die Ergebnisfindung des Modells ausüben. In die Jahre gekommene Immobilien mit einer lang zurückliegenden Modernisierung haben einen besonders negativen Einfluss auf den vorhergesagten Immobilienwert. Diese Aussage wird durch die gegensätzliche Beobachtung bei Immobilien mit späterem Bau- und Modernisierungsjahr gestützt. Diese üben einen positiven Effekt auf das Ergebnis des ANN-Modells aus, womit im Umkehrschluss der Vorhersagepreis steigt. Außerdem erhöht das Modell die Höhe des ermittelten Preises, sobald die Garage strukturell mit der Immobilie verbunden ist und die Anzahl an Kaminen im Haus zunimmt. Nachfolgend wirken sich laut dem Beeswarm-Diagramm Zweifamilien Häuser positiv auf die Höhe der Preisvorhersage aus und stellen innerhalb der Variable *House Style* eine einflussreiche Ausprägung in der

Ergebnisfindung des Modells dar. Der stärkste Einfluss einer kategorialen Variable geht jedoch von dem Zustand des Kellers aus. Hierbei wird seitens SHAP besonders einem ausgebauten Keller, der die Qualitäten eines Lebensraumes erhöht ein hoher globaler Einfluss auf die Preisvorhersage zugesprochen.

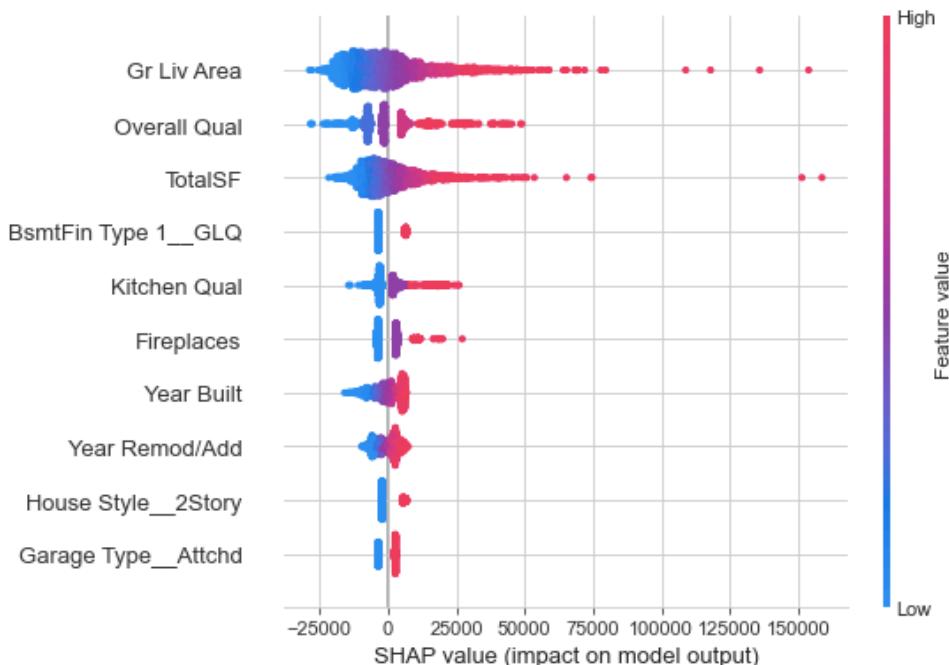


Abbildung 61: SHAP Global Beeswarm-Diagramm ANN-Modell

Nachfolgend wird mit Beendigung der globalen Analyse, die lokale Analyse nach SHAP beschrieben. Dabei weist in der Abbildung 62 $f(x) = 152.899,95$ \$ für die Beobachtung 1 aus. Dieser liegt somit circa 7.400 \$ über dem tatsächlichen Preis. Der Basis Wert für $E[f(X)]$ liegt bei 177.239,91 \$. Die lokale Veränderung des Ausgangswert wird am stärksten durch die Zugehörigkeit zur Nachbarschaft Crawford beeinflusst. Positiv wird vom Modell die Fläche aller Lebensräume oberhalb des Kellers bewertet. Ebenfalls positiv beitragend wird die Variable *MS Zoning_RH* aufgeführt, die die Lage innerhalb eines Bereichs mit einer hohen Wohndichte beschreibt. Auffällig ist, dass die kategoriale Variable *Bld Type* jeweils durch zwei verschiedene Ausprägungen doppelt vertreten ist. Kumuliert senken beide Ausprägungen $E[f(X)]$ um 18.607,29 \$. Das Modell bewertet in dem Fall negativ, dass die Immobilie einem Zweifamilienhaus entspricht. Positiven Einfluss auf die Ergebnisfindung übt der Hausstil *2Story* aus, was einem Zweifamilienhaus mit 2 Etagen entspricht. Interessant ist hierbei, dass ähnliche Eigenschaften der Immobilie, wie es für den Hausstil und den Gebäudetyp der Fall ist, unterschiedliche Auswirkungen auf

die Ergebnisfindung auf das ANN-Modell haben. Die Beobachtung lässt vermuten, dass zwei Ausprägungen, die einen ähnlichen Sachverhalt wiedergeben, zu widersprüchlichen Erklärungen führen können. Dies wird im Rahmen des Kapitels 6.5 nochmals tiefgehender untersucht und beschrieben. Variablen, die ebenfalls den vorhergesagten Preis schmälern sind die Gesamtqualität, die mit einer 5 von 10 bewertet ist, das Modernisierungsjahr, welches ein Jahr nach dem Baujahr erfolgte, die Garagenart, die eine Doppelgarage indiziert und die nicht vorhandene zentrale Lüftung. Zusätzlich schmälern die übrigen Variablen den Basiswert nochmals um 25.540,46 \$. Hierbei wird deutlich, dass die restlichen Variablen zusammengefasst den größten Einfluss auf das Vorhersageergebnis der Beobachtung 1 ausüben. Auffällig ist der hohe Anteil an kategorialen Variablen.

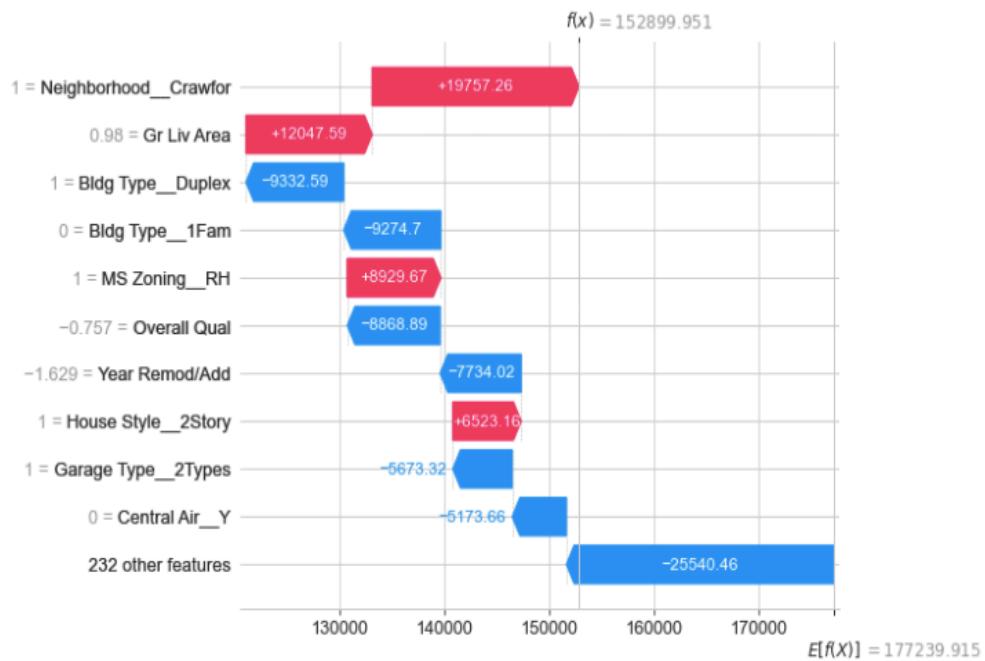


Abbildung 62: SHAP Lokal Wasserfall-Diagramm ANN-Modell Beobachtung 1

Das Force-Diagramm in Abbildung 63 unterstützt die Erkenntnisse darüber, dass sich die vorhandene Konstellation der Variablen überwiegend negativ auf die Preisvorhersage auswirkt. Dabei drücken die negativen Einflüsse in blau die Preisvorhersage deutlich in Richtung 0 entgegen dem Basiswert. Jede Variable steht für einen Streifen im Diagramm und veranschaulicht, welchen Einfluss sie laut SHAP auf die lokale Beobachtung ausübt.

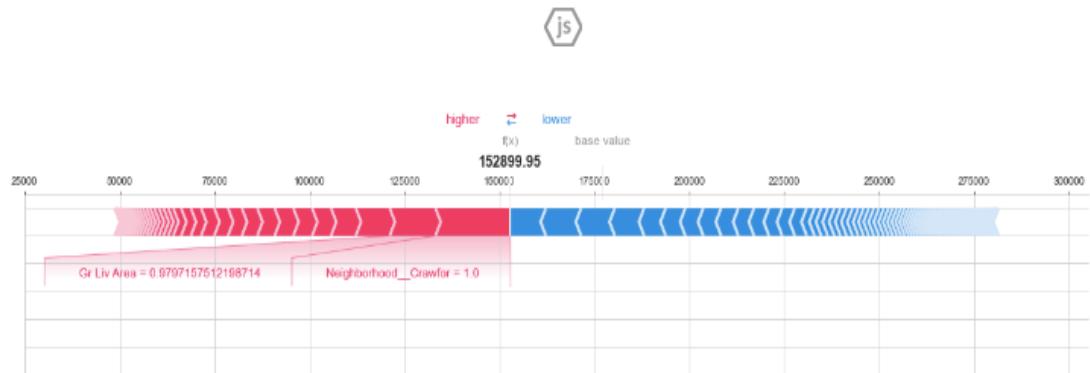


Abbildung 63: SHAP Lokal Force-Plot ANN-Modell Beobachtung 1

Nachfolgend zeichnet sich für die Beobachtung 2 in Abbildung 64 ein wesentlicher Unterschied zur ersten lokalen Beobachtung ab. Der vorhergesagte Preis beträgt demnach 239.515,61 \$ und liegt knapp 96.515 \$ über dem tatsächlichen Preis von 143.000 \$. Zu beobachten ist, dass die drei einflussreichsten Variablen nach SHAP kumuliert den Basiswert um 42568,19 \$ erhöhen. Dies entspricht 24 % der gesamten Beeinflussung des Basiswertes. Dies inkludiert das steile Grundstück, auf dem die Immobilie steht, die Küchenqualität, die einer 4 von 5 entspricht, und der qualitativ gut ausgebauten Keller. Zur Minderung führt lediglich die Gesamtqualität *Overall Qual* und die übrigen kumulierten Variablen, die zusammengefasst zu einer Minderung von 22.398,11 \$ beitragen. Ebenfalls zur Erhöhung tragen die Anzahl von zwei Kaminen bei, die Zugehörigkeit zur Nachbarschaft Clear Creek, Holz als Zweitmaterial für die Fassade, die hohe Qualität des fertiggestellten Kellers, das flache Dach und die Verkaufskonditionen. Die Verkaufskonditionen entsprechen der Gegebenheit, dass das Haus bei Abschluss des Kaufvertrags noch nicht in einem fertigen Zustand war. Zusammengefasst üben die kategorialen Variablen unter den zehn einflussreichsten Variablen den stärksten Einfluss auf das prädiktive Ergebnis der Beobachtung 2 aus.

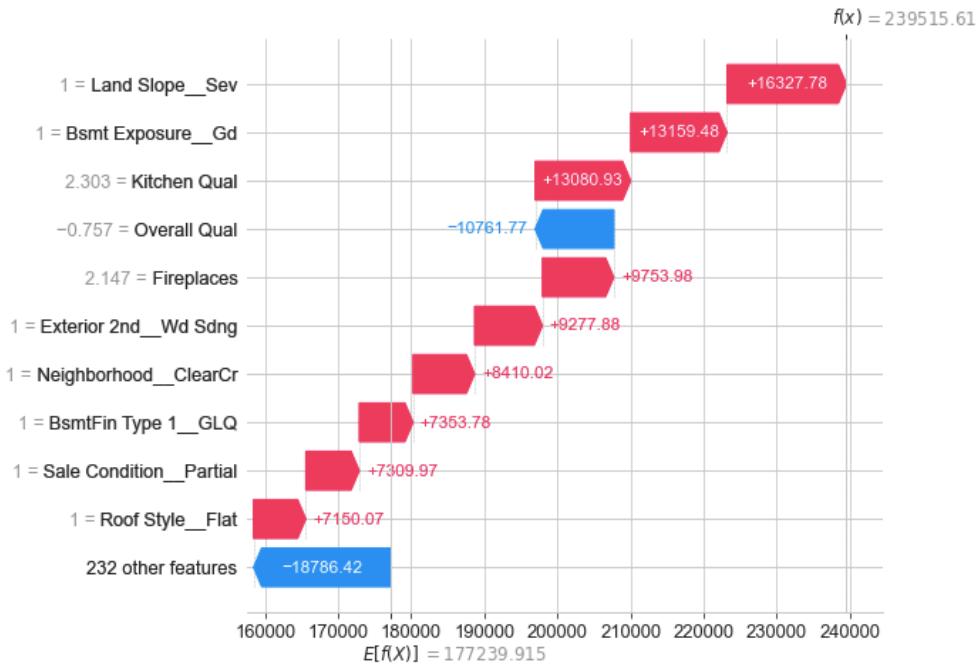


Abbildung 64: SHAP Lokal Wasserfall-Diagramm ANN-Modell Beobachtung 2

Im weiteren Verlauf fokussierte sich die Untersuchung auf die XAI-Methode LIME. Die Durchlaufzeit für die globalen LIME-Ergebnisse für das ANN-Modell beträgt 1 Stunde und 45 Minuten und ist demnach länger als bei SHAP. Hierbei werden in Abbildung 65 die 10 einflussreichsten Variablen für die Vorhersage angezeigt. Wie bereits in der Beobachtung 1 und 2 festgestellt, leistet die Poolqualität laut der globalen Analyse von LIME den größten Beitrag zur Vorhersage des ANN-Modells. Die nachfolgenden Variablen *Gr Liv Area*, *TotalSF* und *Overall Qual* werden seitens LIME als die am stärksten gewichteten Variablen im Modell definiert. Die Ergebnisfindung des Modells weist Ähnlichkeiten zu den SHAP-Ergebnissen auf. Dabei übt der Beitrag der Wohnfläche oberhalb des Kellers einen nahezu identischen Einfluss wie die Poolqualität aus, der sich auf einen durchschnittlichen Beitrag von knapp 30.000 \$ pro Beobachtung beläuft. Auffällig sind nachfolgend die kategorialen Variablen *Neighborhood*, *Bldg Type* und *Garage Type*, die nach LIME die Ränge fünf bis acht belegen. Diese werden seitens SHAP nicht innerhalb der Top zehn aufgeführt. Demnach können die Nachbarschaft und der Gebäudetyp eine durchschnittliche Beeinflussung von circa 17.500 \$ für das prädiktive Ergebnis ausmachen. Diese üben ebenso eine ähnliche Gewichtung aus, wie die Variablen *Overall Qual* und *TotalSF*.

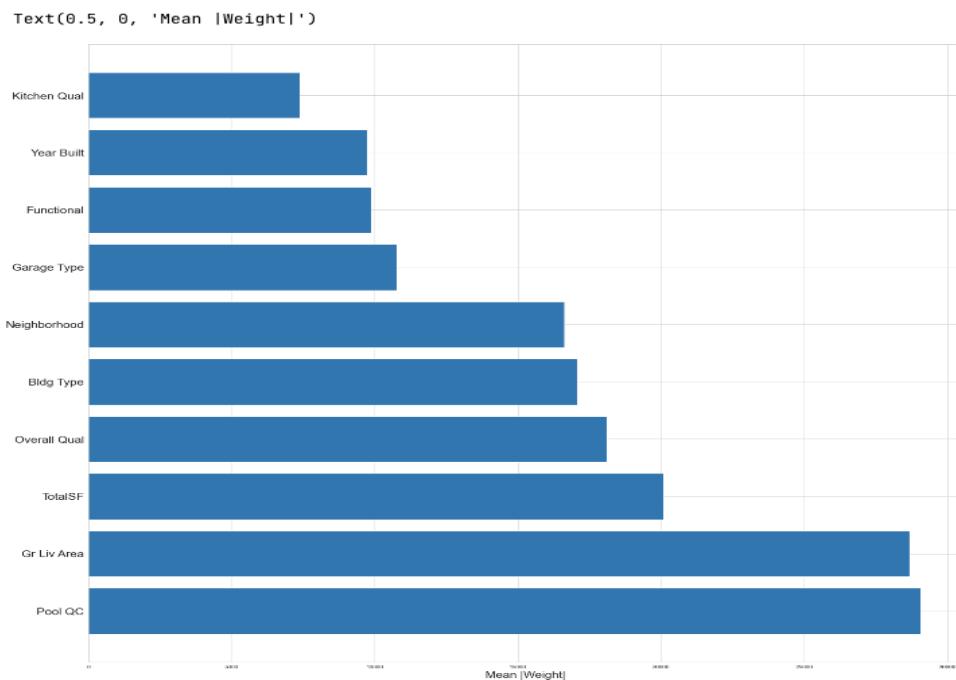


Abbildung 65: LIME Global Balken-Diagramm ANN-Modell

Die Durchlaufzeit beträgt für die erste und zweite Beobachtung zwei Minuten. Für die Beobachtung 1 in der Abbildung 66 wird seitens LIME die Variable *Pool QC* als einflussreichste Variable identifiziert. Diese wird in der SHAP-Betrachtung des ANN-Modells bisher nicht aufgeführt. Dabei senkt der Beitrag der Poolqualität das Vorhersageergebnis um 34.753,34 \$ und stellt somit den größten negativen Beitrag dar. Nachfolgend tragen die Variablen *Gr Liv Area* und *TotalSF* zu Erhöhung der Preisvorhersage bei, genauso wie die Zugehörigkeit zur Nachbarschaft Crawford. Kumuliert senken jedoch der Gebäudetyp, der einem Zweifamilienhaus entspricht, die Gesamtqualität, das Baujahr, die Küchenqualität und der Garagentyp die gesamte Preisvorhersage des Modells um insgesamt 61.689,62 \$. Letztlich trägt die Variable der Hausfunktionalitäten *Functional* einen positiven Beitrag zur Vorhersage des Modells bei und erhöht diese um 7.760,25 \$.

Die Beobachtung 2 wird ebenfalls am stärksten durch die Variable *Pool QC* beeinflusst. Diese führt zur Senkung des Vorhersagepreises, da nach der vorhandenen Ausprägung kein Pool impliziert ist. Neben der Poolqualität beeinflusst die Gesamtqualität die Ergebnisfindung des Modells negativ und mindert somit das prädiktive Ergebnis. Daraufhin führen die acht nachfolgenden Variablen zur sukzessiven Erhöhung des Vorhersagepreises. Auffällig sind dabei die Gemeinsamkeiten, die sich für das ANN-Modell zwischen LIME und SHAP bezüglich der Beobachtung 1 und 2 feststellen lassen. Dabei bestimmen

beide Methoden nahezu alle einflussreichen Variablen gleich, bis auf die Variable *Pool QC*, die in der lokalen LIME-Auswertung einzigartig ist und den stärksten Einfluss ausübt. Ebenso üben nach LIME zunächst negative Einflüsse den stärksten Einfluss auf das prädiktive Ergebnis in der Beobachtung 2 aus. SHAP führt hierfür zunächst drei Variablen auf, die den stärksten positiven Einfluss ausüben. Die Beobachtung 1 stellt ebenfalls minimale Unterschiede dar. Auffallend ist dabei der starke positive Einfluss, der nach LIME von der Variable *TotalSF* ausgeht, die nach SHAP keinen Beitrag innerhalb der ersten zehn einflussreichsten Variablen leistet.

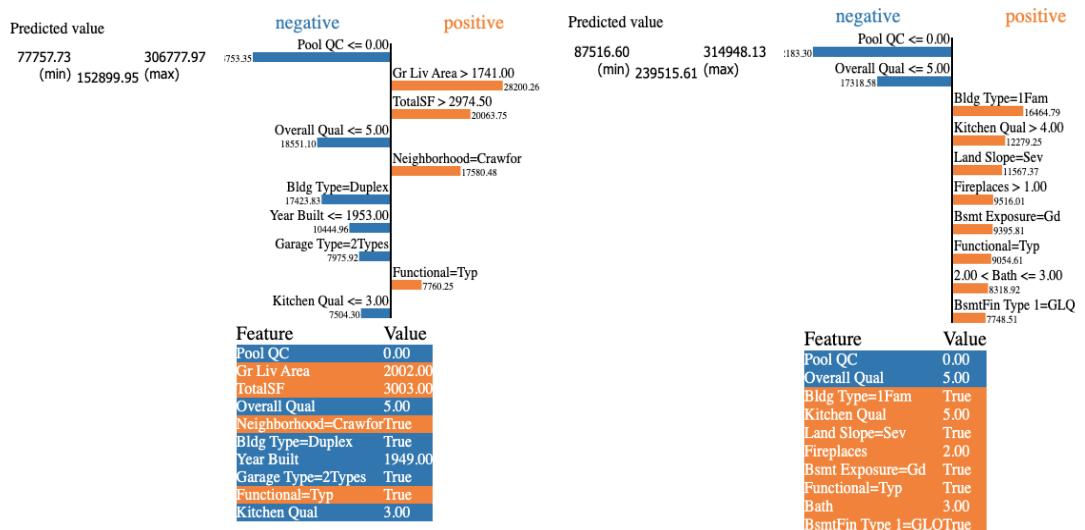


Abbildung 66: LIME Lokal Wasserfall-Diagramm ANN-Modell Beobachtung 1+2

Für die nachfolgende Betrachtung mittels des InterpretML Moduls, ergibt sich die Herausforderung einer technischen Einschränkung. Die InterpretML Bibliothek kann zum derzeitigen Stand der Ausarbeitung keine künstlichen Neuronalen Netze verarbeiten. Die korrespondierende Fehlermeldung ist den Entwicklern der Bibliothek mitgeteilt worden und befindet sich zum jetzigen Stand der Ausarbeitung in Bearbeitung. Demnach entfällt die Untersuchung des ANN-Modells anhand der Nutzung des InterpretML-Moduls.

5.5 Vergleich der XAI-Methoden im Kontext der angewendeten Algorithmen

Der nachfolgende Vergleich der XAI-Methoden im Rahmen der angewendeten Algorithmen und der RQ4 wird anhand der Tabellen 11 bis 13 unterstützt. Dabei enthalten die Tabellen alle Beobachtungen und Ergebnisse in komprimierter Form, die bereits

ausführlich in den vorherigen Kapiteln wiedergegeben sind. Zunächst werden die globalen und lokalen XAI-Ergebnisse der Modelle gegenübergestellt und anhand der Tabellen 11 und 12 beschrieben. Anschließend wird auf Basis des Vergleiches die Einordnung der XAI-Ergebnisse innerhalb der Erklärungseigenschaften vollzogen und durch die Tabellen 14 bis 17 unterstützt.

Zunächst findet der interdisziplinäre Vergleich der globalen Betrachtung der XAI-Methoden statt. Die einzelnen Modelle weisen global gesehen unter der Verwendung von SHAP diverse Gemeinsamkeiten auf. So ist innerhalb der Betrachtung der drei einflussreichsten Variablen zu verzeichnen, dass die Variablen *TotalSF*, *Gr Liv Area* und *Overall Qual* modellübergreifend zu den einflussreichsten Variablen der Ergebnisfindung der Modelle zählen. Lediglich das XGB-Modell weist konträr die Variable *Year Built* auf. Nachfolgend lässt sich die Erkenntnis des Beitrages der Variablen für das KNN- und EBM-Modell ähnlich zusammenfassen. Die Variablen weisen über die Top zehn hinweg marginale Differenzen auf. Dabei nimmt die Beitragshöhe der Einflussstärke der Variablen entsprechend linear ab. Im Gegensatz zeigt das ANN-Modell bereits einen Beitragsunterschied von 3.000 \$ zwischen der einflussreichsten und der zweiteinflussreichsten Variablen auf. Stärkere Differenzen sind für das XGB-Modell zu verzeichnen, bei dem bereits die einflussreichste Variable nach SHAP einen Beitrag von 19.329,30 \$ ausmachen kann. Ebenso leistet *Overall Qual* einen global gesehenen Beitrag von 15.418,98 \$ und liegt als zweiteinflussreichste Variable deutlich über den höchsten Beiträgen aller übrigen Modelle. Hierbei wird nach SHAP deutlich, dass das XGB- und ANN-Modell, einzelnen Variablen einen höheren Beitrag zuspricht. Außerdem wird dies für das XGB-Modell erkenntlich, sobald die übrigen Variablen und deren Beiträge zum prädiktiven Ergebnis näher betrachtet werden. So wird bereits in Kapitel 5.3 festgehalten, dass der Beitrag von *TotalSF* nahezu so hoch ist, wie der kumulierte Einfluss der 232 übrigen Variablen. Das ANN-Modell zeigt eine ähnliche Gewichtung der übrigen Variablen, weshalb der höchste kumulierte Beitrag von 126.208,61\$ ausgewiesen wird. Die Beitragshöhe der übrigen Variablen des KNN- und EBM-Modells ordnen sich zwischen den zwei extremen ein und lassen erkennen, dass das EBM-Modell einzelnen übrigen Variablen einen höheren numerischen Einfluss zuspricht als das KNN-Modell. Das Verhältnis von quantitativen zu qualitativen Variablen zeigt, dass das XGB-Modell im Gegensatz zu den anderen Modellen, quantitative Variablen überdurchschnittlich häufig vertreten sind und

somit eine höhere Gewichtung nach SHAP erfahren. Die ausgeglichenste Verteilung lässt sich im ANN-Modell beobachten, gefolgt vom EBM- und KNN-Modell, welche sechs quantitative und vier kategoriale Variablen aufführen. Wesentliche Unterschiede im Rahmen des Beeswarm-Plots sind nicht zu erkennen. Es lässt sich erkennen, dass alle Algorithmen hohen tatsächlichen Werten der quantitativen Variablen einen höheren Einfluss zusprechen. Ebenso ist zu beobachten, dass die SHAP-Werte der kategorialen Variablen einer ähnlichen Höhe entsprechen und es keine gravierenden Unterschiede zu verzeichnen gibt. Die Einsatzmöglichkeit des Dependence-Diagramms des TreeExplainer signalisiert das Potenzial einer kontextuellen Analyse. Diese Möglichkeit ist im Rahmen des Explainers nicht gegeben. Welche Bedeutung dies für das Verständnis zwischen Menschen und Maschine haben kann, wird im Rahmen der Diskussion näher erläutert.

Im Rahmen der Berechnungszeit sind wesentliche Unterschiede zu erkennen. Der Einsatz des TreeExplainers für das XGB-Modell ermöglicht das Berechnen der globalen und lokalen SHAP Values in lediglich 25 Minuten. Im Vergleich benötigt das ANN-Modell 3-mal länger und das EBM-Modell nahezu zehnmal länger. Der KNN-Algorithmus benötigt 34 Minuten unter Verwendung des Explainers und weist somit die geringste Berechnungszeit nach SHAP auf. Hierbei zeigt sich die Funktionsweise von SHAP, bei dem der TreeExplainer einer zugeschnittenen Version für baumartige Algorithmen entspricht. Die Dauer der Berechnung verläuft nicht exponentiell, sondern polynominal. Ein weiterer Faktor für eine höhere Berechnungszeit ist die Komplexität des Algorithmus. Da die Erklärbarkeit der Ergebnisfindung des KNN-Modells mit einem einfacheren Aufwand für SHAP verbunden ist als für das ANN-Modells, fällt die Berechnungszeit dementsprechend geringer aus (Yang, 2021, S.1-2). Die lange Berechnungszeit für das EBM-Modell kann zum derzeitigen Standpunkt nicht nachvollzogen werden. Lediglich lässt sich ausschließen, dass es nicht an der Entwicklungsumgebung oder an der Leistung des verwendeten Computers liegt, da diese Faktoren für alle XAI-Verfahren gleich gehalten sind.

Die Anwendung von LIME Global lässt zunächst beobachten, dass im Vergleich zu SHAP, die quantitativen Einflüsse der Variablen deutlich höher ausfallen. Nichtsdestotrotz befindet sich modellübergreifend jeweils mindestens eine der Variablen *TotalSF*, *Gr Liv Area* und *Overall Qual* unter den drei Einflussreichsten. Lediglich das ANN-Modell weist im Gegensatz zu den SHAP-Ergebnissen die Poolqualität als einflussreichste Variable auf, welche ebenfalls das Vorhandensein eines Pools indizieren kann. Welchen

Umständen dies geschuldet ist, bedarf einer tiefgründigen Analyse. Auffällig ist dabei die hohe quantitative Gewichtung seitens LIME für das KNN-Modell. Dies ergibt sich aus der Berechnungsweise des globalen LIME-Ansatzes nach O’Sullivan (2022), da bereits die lokalen Beobachtungen höhere Beiträge der Variablen aufweisen und somit der durchschnittliche Beitrag höher ausfällt. Außerdem lässt sich für alle Modelle beobachten, dass LIME ein ausgeglicheneres Verhältnis von quantitativen zu kategorialen Variablen aufweist. Die Berechnungszeit von LIME weist über alle Modelle hinweg keine großen Diskrepanzen auf und beträgt zwischen 14 bis 20 Minuten. Es ist jedoch zu beachten, dass der Programmieraufwand von einer erkenntnisreichen lokalen Beobachtung hin zu einer globalen Deutung nicht zu unterschätzen ist.

Die globale Erklärbarkeit des Vorhersageergebnisses mittels des InterpretML-Moduls wird im Folgenden nicht das ANN-Modell beinhalten, da das Modul derzeit keine Tensorflow-Modelle unterstützt. Die Gesamtbetrachtung vermittelt ein ähnliches Ergebnis wie die beiden vorherigen XAI-Methoden. Erneut sind überwiegend die drei Variablen *TotalSF*, *Gr Liv Area* oder *Overall Qual* vertreten. Auffällig dabei ist, dass die Variable *TotalSF* in allen Modellen als stark beitragend zu verzeichnen ist. Die einzelnen Beitrags-höhen der Variablen entsprechen den Dimensionen des globalen LIME-Ansatzes. Dabei zeichnet sich für das KNN- und XGB-Modell eine große Differenz bezüglich der Abstände der Variablen ab, wohingegen das EBM-Modell den einzelnen Variablen eine ähnlich hohe Gewichtung zuordnet, wie im Rahmen von SHAP. Das XGB-Modell weist nach dem InterpretML-Modul erneut der Variable *TotalSF* die größte Beeinflussung der Ergebnisfindung zu. Die Anteile an quantitativen und kategorialen Variablen in der globalen Betrachtung sind in allen Modellen nahezu gleichmäßig. Die Berechnungsdauer der globalen InterpretML Analyse beträgt für jedes Modell lediglich 30 Sekunden und stellt somit die schnellste Methode dar. Zur globalen Analyse ermöglicht InterpretML die Nutzung des PDP-Diagramms, welches die Basis des Dependence-Plots des TreeExplainers darstellt. Gerade im Hinblick auf die Variable *TotalSF* zeichnen die PDP-Diagramme aller Modelle ein ähnliches Bild ab. Sobald beispielsweise eine Quadratfußanzahl über 2.000 zu verzeichnen ist, steigt ebenfalls der Beitrag der Variable. Dies lässt sich ebenso beim Dependence-Diagramm der SHAP-Methodik erkennen.

Zusammenfassend lässt sich festhalten, dass alle Modelle im Bereich der einflussreichsten Variablen überwiegend Gemeinsamkeiten aufweisen. Unterschiede bestehen in der

berechneten Höhe der Beiträge der Variablen, die nicht nur innerhalb der Methoden, sondern auch übergreifend zu erkennen sind. Es gibt auch Differenzen in der Berechnungszeit der verschiedenen Methoden, wobei selbst innerhalb der SHAP-Methode je nach Algorithmus Unterschiede zu erkennen sind. Eine übermäßige Bevorzugung eines Skalen niveaus kann nicht beobachtet werden. Die graphischen Darstellungsvariationen der verschiedenen Methoden ermöglichen eine globale Analyse bezüglich der Wert- und Signifikanzdeutung einzelner Variablen.

Letzteres ist lediglich für SHAP und InterpretML in Form der Dependence-, Beeswarm- und PDP-Diagramme möglich. Die quantitative Deutung umfassen alle drei Methoden, indem die Feature-Importance mittels des numerischen Einflusses in \$ widergespiegelt wird. Dabei bieten SHAP und LIME die Möglichkeit, die Anzahl an darzustellenden Variablen zu begrenzen und zu erweitern. InterpretML bietet derzeit diese Möglichkeit nicht und schränkt damit zum Teil das globale Verständnis ein. Hierbei wäre eine kumulierte Anzeige der Beiträge der übrigen Variablen, wie Sie bei SHAP vorzufinden ist, zielführend. Insgesamt ermöglicht das Balkendiagramm der globalen SHAP-Analyse einen umfassenden Blick, ohne dabei eine Informationsreduktion zu fördern. Dies kann für die anderen Methoden eine hilfreiche Erweiterung darstellen.

Tabelle 11: Ergebnisse aller Modelle und XAI-Verfahren Global

Algorithmus	SHAP Global			Berechnungsduer
	Top 3 Variablen	Beitragshöhe Top 3	Beitrag übrigen Variablen	
ANN	Gr Liv Area	11.604,97 \$	126.208,61	1 Std 12 Min
	Overall Qual	8.136,93 \$		
	TotalSF	7.985,08 \$		
KNN	Gr Liv Area	5.011,06 \$	4.9682,88	0 Std 25 Min
	Overall Qual	4.310,02 \$		
	TotalSF	4.131,30 \$		
XGB	TotalSF	19.329,30 \$	22.160,17	0 Std 34 Min
	Overall Qual	15.418,98 \$		
	Year Built	4.794,59 \$		
EBM	Gr Liv Area	5.283,90 \$	7.9754,74	3 Std 58Min
	TotalSF	4.615,64 \$		
	Overall Qual	4.568,94 \$		
LIME Global				
ANN	Pool Qc	29.054,13 \$	-	0 Std 20 Min
	Gr Liv Area	28.667,78 \$		
	TotalSF	20.065,33 \$		
KNN	TotalSF	77.080,43 \$	-	0 Std 15 Min
	Gr Liv Area	22.132,35 \$		
	Garage Area	18.985,63 \$		
XGB	TotalSF	50.053,89 \$	-	0 Std 12 Min
	Overall Qual	25.409,99 \$		
	Bath	12.897,75 \$		
EBM	Garage Qual	17.315,90 \$	-	0 Std 14 Min
	TotalSF	15.384,68 \$		
	Gr Liv Area	15.110,86 \$		
InterpretML				
KNN	Pool Qc	110.880,30 \$	-	0 Std 0,5 Min
	TotalSF	92.504,82 \$		
	Total BsmtSF	77.531,36 \$		
XGB	TotalSF	14.0745,40 \$	-	0 Std 0,5 Min
	Overall Qual	120.058,40 \$		
	1st Flr SF	64.592,96 \$		
EBM	TotalSF	4.187,17 \$	-	0 Std 0,5 Min
	Gr Liv Area	4.119,42 \$		
	Bath	3.770,83 \$		

Die Erklärbarkeit der Beobachtung 1 und 2 mittels der XAI-Verfahren weist im Gegensatz zur globalen Beobachtung diverse Unterschiede auf. So lässt sich beobachten, dass die einflussreichsten Variablen für jedes Modell unterschiedlich nach SHAP sind. Die Reihenfolge und der quantitativ ausgewiesene Einfluss der Variablen variieren ebenfalls je nach Modell. Beispielsweise führen das ANN- und das XGB-Modell in der Beobachtung 1 die Variable *Neighborhood_Crawford* unter den zehn einflussreichsten Variablen auf. Dabei lässt sich auch ein Einfluss der Variable im EBM-Modell wiedererkennen, dieser wird jedoch nach SHAP, als nicht stark einflussreich eingeordnet. Weiterführend lässt sich beobachten, dass sich die Erkenntnis der globalen Analyse bezüglich der Beitragshöhe der übrigen Variablen auf die lokalen Beobachtung 1 und 2 übertragen lässt. Dabei bildet die Beitragshöhe des XGB-Modells den geringsten Beitrag und wird maßgeblich durch die zehn einflussreichsten Variablen bestimmt, wohingegen das prädiktive Ergebnis des ANN-Modells maßgeblich durch den Beitrag der übrigen Variablen bestimmt wird. Nachfolgend spiegelt sich für das KNN- XGB- und EBM-Modell eine überwiegend ausgeglichene Verteilung von quantitativen und kategorialen Variablen in den Top 10 wider. Das ANN-Model hingegen weist den kategorialen Variablen eine größere Gewichtung zu. Da die Ausführung von SHAP sowohl die globale als auch lokale Berechnung beinhaltet, ist die Berechnungszeit für beide Analysen identisch. Lediglich die Eingrenzung von definierten Datenspalten für die lokale Analyse ist für SHAP möglich und für große Datensätze nötig.

Die lokale LIME-Betrachtung weist im Vergleich zu SHAP mehr Ähnlichkeiten zur komplementären globalen Beobachtung aus. Die aufgeführten einflussreichen Variablen der Modelle entsprechen zu großen Teilen der globalen Beobachtung und umfassen für die Beobachtung 1 am häufigsten die Variablen *TotalSF* und *Gr Liv Area*. Für die Beobachtung 2 ist dies für die Variable *Overall Qual* zu verzeichnen. Dabei fallen die Gewichtungen der Variablen erneut höher aus als im Vergleich zu SHAP und InterpretML. Der Beitrag der Modelle spiegelt nahezu die globale Verteilung der Modelle wider. Lediglich die einflussreichste Variable des XGB-Models in der Beobachtung 1 übt einen höheren Einfluss als das KNN-Model aus, welches jedoch in den lokalen Beobachtungen in Erscheinung treten kann, da die globalen Analysen den Durchschnitt der Gesamtheit der Menge darstellen. Demnach ist es möglich, dass lokale Beobachtungen von der globalen Analyse abweichen und die lokale Feature Importance sich wesentlich von der globalen

unterscheidet. Eine Gemeinsamkeit zu SHAP spiegelt sich in der Verteilung der Skalenniveaus wider, bei der ebenfalls das ANN-Modell eine höhere Gewichtung auf kategoriale Variablen nach LIME verzeichnet. Die übrigen Modelle weisen erneut eine ausgeglichene Verteilung auf. Die Berechnungsdauer beläuft auf zwischen fünf bis zehn Minuten und ist demnach wesentlich schneller als SHAP. Zu beachten ist jedoch der programmietechnische Mapping Aufwand, der vorab betrieben werden muss, um die gewünschten Ergebnisse zu erzielen. Diese würde die reale Berechnungszeit pro Algorithmus um 5 Minuten erhöhen. LIME bietet in der derzeitig vorhandenen Bibliothek kein Dependence-Plot an, weist jedoch auf Basis der gesamten Daten auf einen Schwellenwert hin, der je nach Auslegung zu einem positiven oder negativen Beitrag der Variable führt. Hierbei lässt sich seitens LIME nachvollziehen, welcher Wert oder welche Ausprägung zum Zuwachs oder zur Minderung des prädiktiven Ergebnisses auf Basis der derzeitig genutzten Datenbasis führt. Dies ermöglicht im Vergleich zu den anderen XAI-Methoden eine lokale Betrachtung der Abhängigkeitsanalyse von Wertigkeiten.

Die Verwendung des lokalen InterpretML-Moduls weist im Rahmen der einflussreichsten Variablen Ähnlichkeiten zu den Ergebnissen der lokalen SHAP-Betrachtung auf. Dies hängt damit zusammen, dass die lokale Methode des InterpretML dem KernelExplainer von SHAP entspricht und somit die Ergebnisse oft identisch sind. Dies wird besonders im Anbetracht des XGB- und EBM-Modells deutlich. Diese Beobachtung trifft ebenso für die Beitragshöhen der Variablen zu. Lediglich das Verhältnis an quantitativen und kategorialen Variablen des EBM-Modells unterscheidet sich im Vergleich zu den Ergebnissen anderer XAI-Methoden. Hierbei wird dem EBM-Modell nach InterpretML eine höhere Gewichtung der kategorialen Variablen zugeschrieben, als es im umgekehrten Fall für das ANN-Modell unter SHAP und LIME zu verzeichnen ist. Insgesamt lässt sich jedoch im Rahmen der Berechnungszeit beobachten, dass das EBM-Modell neben dem XGB- und KNN-Modell am schnellsten performt. Auffällig ist jedoch der Vergleich zur globalen Berechnungszeit, die wesentlich schneller ausfällt als die lokale.

Tabelle 12: Ergebnisse Lokal Beobachtung 1

Algorithmus	SHAP lokal 1			Berechnungsduer
	Top 3 Variablen	Beitragshöhe Top 3	Beitrag übrigen Variablen	
ANN	Neighborhood_Crawford Gr Liv Area Bldg Type Duplex	19.757,26 \$ 12.047,59 \$ -9.332,59 \$	-25.540,46 \$	2 / 8 1 Std 12 Min
KNN	Year Remod/Add Garage Year Build Fireplaces	-7.705,57 \$ -6.171,17 \$ -5.655,38 \$	-11.715,39 \$	5 / 5 0 Std 25 Min
XGB	Overall Qual Neighborhood_Crawford Year Remod/Add	-15.975,10 \$ 11.245,33 \$ -6.953,71 \$	15.860,14 \$	4 / 6 0 Std 34 Min
EBM	Garage Cars Garage Area Gr Liv Area	12.477,28 \$ 12.317,56 \$ 8.342,71 \$	-31.462,87 \$	4 / 6 3 Std 58Min
LIME lokal 1				
ANN	Pool Qc Gr Liv Area TotalSF	-34.753,34 \$ 28.200,26 \$ 20.063,75 \$	-	3 / 7 0 Std 10 Min
KNN	Pool Qc Gr Liv Area TotalSF	-47.780,00 \$ 16.191,91 \$ 12.439,27 \$	-	4 / 6 0 Std 6 Min
XGB	TotalSF Overall Qual Bath	50.429,00 \$ -25.501,43 \$ -13.380,83 \$	-	5 / 5 0 Std 5 Min
EBM	Garage Qual Gr Liv Area TotalSF	-17.240,01 \$ 16.276,26 \$ 15.796,24 \$	-	5 / 5 0 Std 6 Min
InterpretML lokal 1				
KNN	Fireplaces Kitchen Qual Year Remod/Add	-4.243,86 \$ -4.036,13 \$ -3.756,07 \$	-	4 / 6 0 Std 10 Min
XGB	Overall Qual Neighborhood_Crawford Kitchen Qual	-16.519,64 \$ 11.634,05 \$ -6.609,72 \$	-	4 / 6 0 Std 8 Min
EBM	Garage Area Garage Cars Neighborhood Crawford	12838,34 \$ 12573,79 \$ 8244,05 \$	-	3 / 7 0 Std 1 Min

Tabelle 13: Ergebnisse Lokal Beobachtung 2

Algorithmus	SHAP lokal 2			Berechnungszeit
	Top 3 Variablen	Beitragshöhe Top 3	Beitrag übrigen Variablen	
ANN	Land Slope_Sev Bsmt Exposure_Gd Kitchen Qual	16.327,76 \$ 13.159,48 \$ 13.080,93 \$	-18.786,42 \$	1 Std 12 Min
KNN	Fireplaces Kitchen Qual Bath	16.486,13 \$ 5.367,53 \$ 5.042,63 \$	16.191,75 \$	0 Std 25 Min
XGB	Overall Qual Fireplaces Bsmt Exposure_Gd	14.174,41 \$ 7.325,87 \$ 6.473,00 \$	-1.633,86 \$	0 Std 34 Min
EPM	Kitchen Qual Fireplaces Land Slope_Sev	17.450,09 \$ 10.891,55 \$ 8.225,49 \$	4.580,38 \$	3 Std 58 Min
LIME lokal 2				
ANN	Pool Qc Overall Qual Bldg Type Fam	32.183,29 \$ 17.318,58 \$ 16.464,79 \$	-	0 Std 8 Min
KNN	Pool Qc Overall Qual Kitchen Qual	46.860,41 \$ 10.579,92 \$ 10.533,53 \$	-	0 Std 6 Min
XGB	Overall Qual Garage Cond Bath	25.746,40 \$ 19.724,73 \$ 13.347,87 \$	-	0 Std 6 Min
EPM	Garage Qual Kitchen Qual Land Slope_Sev	19.368,48 \$ 17.366,09 \$ 16.210,82 \$	4/6	0 Std 6 Min
InterpretML lokal 2				
KNN	Fireplaces Kitchen Qual Fireplace Qual	9.824,07 \$ 9.329,74 \$ 4.468,13 \$	5/5	0 Std >1 Min
XGB	Overall Qual Fireplaces Bsmt Exposure_Gd	-14.673,06 \$ 7.158,36 \$ 6.532,65 \$	-	0 Std >1 Min
EPM	Kitchen Qual Fireplaces Bsmt Exposure_Gd	17.152,43 \$ 10.596,74 \$ 8.279,41 \$	3/7	0 Std >1 Min

Im Rahmen der grafischen Darstellung lässt sich beobachten, dass SHAP neben dem Wasserfall-Diagramm das Force Plot zu lokaler Analyse bereitstellt, welches nochmals das Kräfteverhältnis von positiven und negativen Beiträgen der Variablen verdeutlichen soll. Unterstützend verdeutlicht der Basiswert den Durchschnittswert aller Preisvorhersagen, ohne den Einfluss jeglicher Variablen (Marcilio & Eler, 2020, S.342). Dies ermöglicht eine tiefergehende kontextuelle Analyse der Beiträge jeder Variablen und die Aussage, wie sie zur Vorhersage beitragen. Weiterführend besteht die Möglichkeit die Anzahl an angezeigten Variablen zu definieren und somit lokale Analysen nach Belieben tiefgehend oder oberflächlich zu bestreiten. Hierbei wird automatisch der kumulierte Wert der ausgeblendeten Variablen seitens SHAP angezeigt. Dadurch kann die Bewertung und Einordnung des lokalen Gesamtergebnisses des Modells ermöglicht werden. Das Wasserfall-Diagramm komplementär von LIME bildet neben den numerischen Gewichtungen der Variablen, einen Schwellenwert ab, der dem Nutzer für quantitative Variablen zu verstehen gibt, ab welchem Wert ein positiver oder negativer Einfluss zu verzeichnen ist. Dabei ermöglicht die integrierte Tabelle einen verständlichen Überblick über alle tatsächlichen Werte der am stärksten beitragenden Variablen. Entgegen einem Basiswert wird seitens LIME eine mögliche Spanne ausgegeben, die in den graphischen Darstellungen einen minimalen und maximalen Preis angibt, indem sich die Preisvorhersage bewegen kann. Die angegebene Spanne fällt jedoch in den betrachteten Beobachtungen groß aus, weshalb diese im Rahmen der Zielsetzung keinen hohen Erkenntniswert bereitzustellen scheint. Für LIME existieren neben dem betrachteten Diagramm weitere Balkendiagramme, die jedoch weniger aussagekräftig sind als die genutzte lokale Betrachtungsweise. InterpretML bietet hauptsächlich die Möglichkeit für Blackbox-Modelle diverse XAI-Methoden unter einem Modul zu verwenden. Diese werden, wie bereits für die lokale Analyse beobachtet, mittels Wasserfall-Diagrammen dargestellt und umfassen den KernelExplainer von SHAP und LIME. Die Diagramme sind dabei interaktiv und ermöglichen es die Darstellung ebenfalls im Bildformat Portable Network Graphic herunterzuladen. Eine große Stärke seitens InterpretML ist die Kompilierung aller genutzten lokalen und globalen Methoden innerhalb eines Dashboards. Die Abbildung 67 zeigt, wie eine Zusammenfassung der lokalen und globalen Ergebnisse des EBM- und XGB-Modells aussehen kann und ermöglicht ebenso die Einblendung des PDP-Diagramms, sobald eine Variable in der Tabelle oberhalb des Diagramms ausgewählt wird. Dabei können

zusätzlich die Datenquelle und die Performance der einzelnen Modelle begutachtet werden. Die hauptsächliche Stärke des InterpretML liegt in der Möglichkeit, lokale und globale Ergebnisse zusammenfassend und interaktiv zu analysieren.

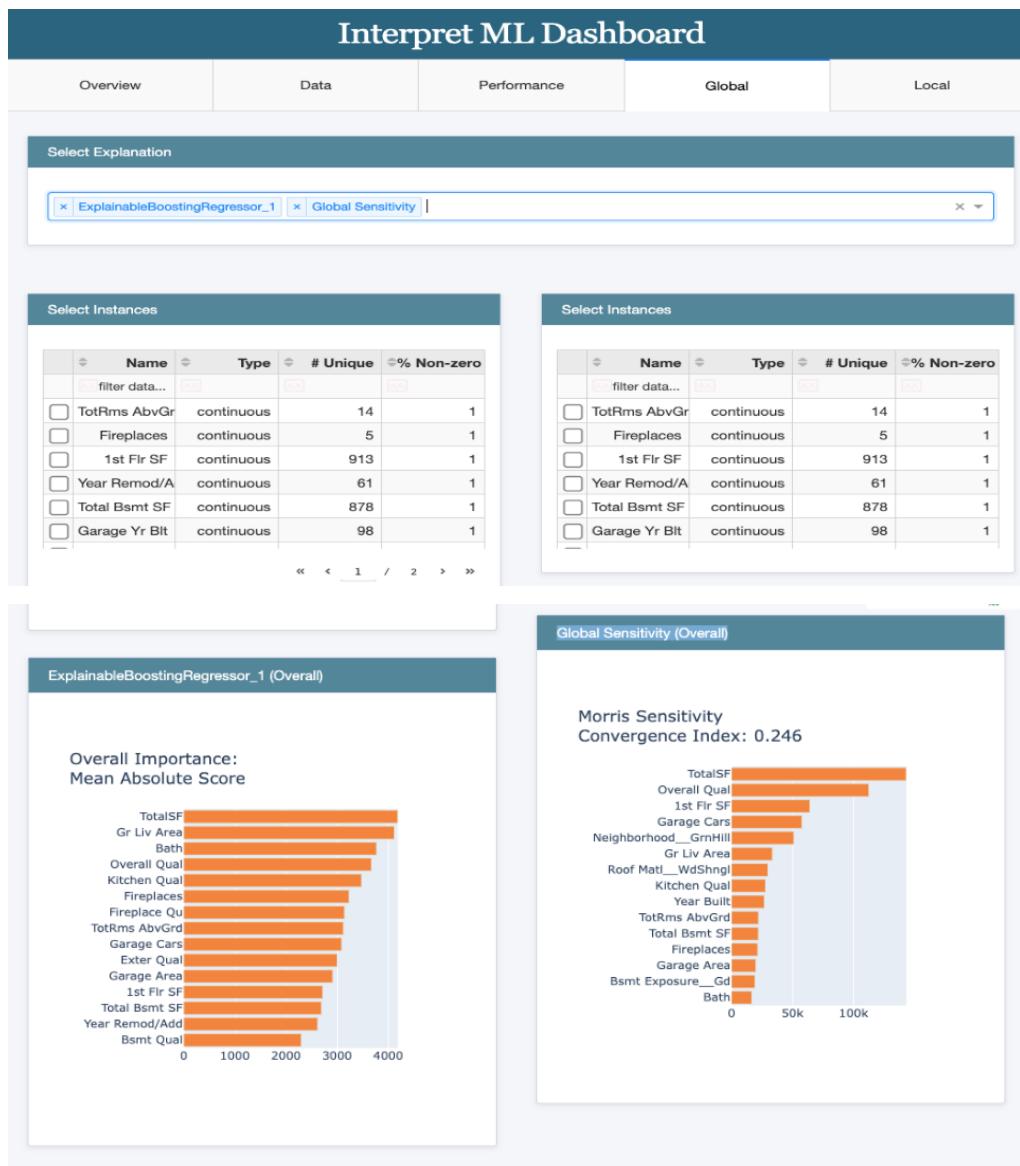


Abbildung 67: InterpretML Dashboard für XGB- und EBM-Modell

Zusammenfassend lässt sich für die Darstellungsweisen der einzelnen Methoden festhalten, dass SHAP die größte Vielfalt an Darstellungsmöglichkeiten im globalen und lokalen Kontext besitzt und mit dem Explainer jedes ML-Modell verwendet werden kann. Dabei resultiert die Gewichtung der kategorialen Variablen seitens SHAP nicht in einer Verzerzung der Beiträge. Die Summierung der einzelnen Ausprägungen ermöglichen die

Erschließung der Gesamtgewichtung. LIME hingegen bietet eine begrenzte Auswahl an Visualisierungsmöglichkeiten und weist einen großen Aufwand an Vorarbeit auf, sobald kategoriale Variablen mittels One-Hot-Encoding Verfahren transformiert werden. Das InterpretML Modul umfasst eine Auswahl an verschiedenen XAI-Methoden für Black-box- und Whitebox-Modellen und ermöglicht es alle getätigten Analysen übersichtlich auf einem Dashboard zu kompilieren.

Nachfolgend werden die Ergebnisse der Erklärungseigenschaften kumuliert und komprimiert in den Tabellen 14 bis 17 dargestellt. Dabei fasst die Tabelle 14 die Ergebnisse der Untersuchung hinsichtlich der Stabilität, Trennbarkeit und der Konsistenz zusammen, die nachfolgend genauer anhand von Beispielen und der Gegenüberstellung der Verfahren dargestellt werden.

Tabelle 14: Erklärungseigenschaften je Erklärungsmethode

XAI-Verfahren	Stabilität	Trennbarkeit	Konsistenz
Explainer	x	⊗	o
TreeExplainer	⊗	⊗	o
LIME	x	⊗	o
InterpretML - KernelExplainer	x	⊗	o
InterpretML – Glassbox Explainer	⊗	⊗	o

Für die Stabilität lässt sich beobachten, dass nahezu alle zur lokalen Analyse angewendeten XAI-Verfahren die Eigenschaft verletzen. Dies trifft jedoch nicht für XGBoost-Modell unter Anwendung des TreeExplainers und das EBM-Modell unter Anwendung des Glasbox Explainers von InterpretML zu. So lässt sich in der Abbildung 68 die Erklärung für das KNN-Modell erkennen, welches in einem erneuten Durchlauf für die Beobachtung 2 anhand von SHAP und LIME berechnet wird. Dabei zeigt sich, dass die Gewichtungen der Variablen Ähnlichkeiten zu dem festgehaltenen Durchlauf in Kapitel 5.2 aufweisen, jedoch keine identische Erklärung für die Beobachtung 2 darstellen. Dieses Verhalten lässt sich ebenso für die übrigen Modelle beobachten und birgt die Herausforderung eines instabilen Modells. Die Folgen für die praktische Implementierung von instabilen Erklärungen werden im Rahmen der Diskussion näher erläutert.

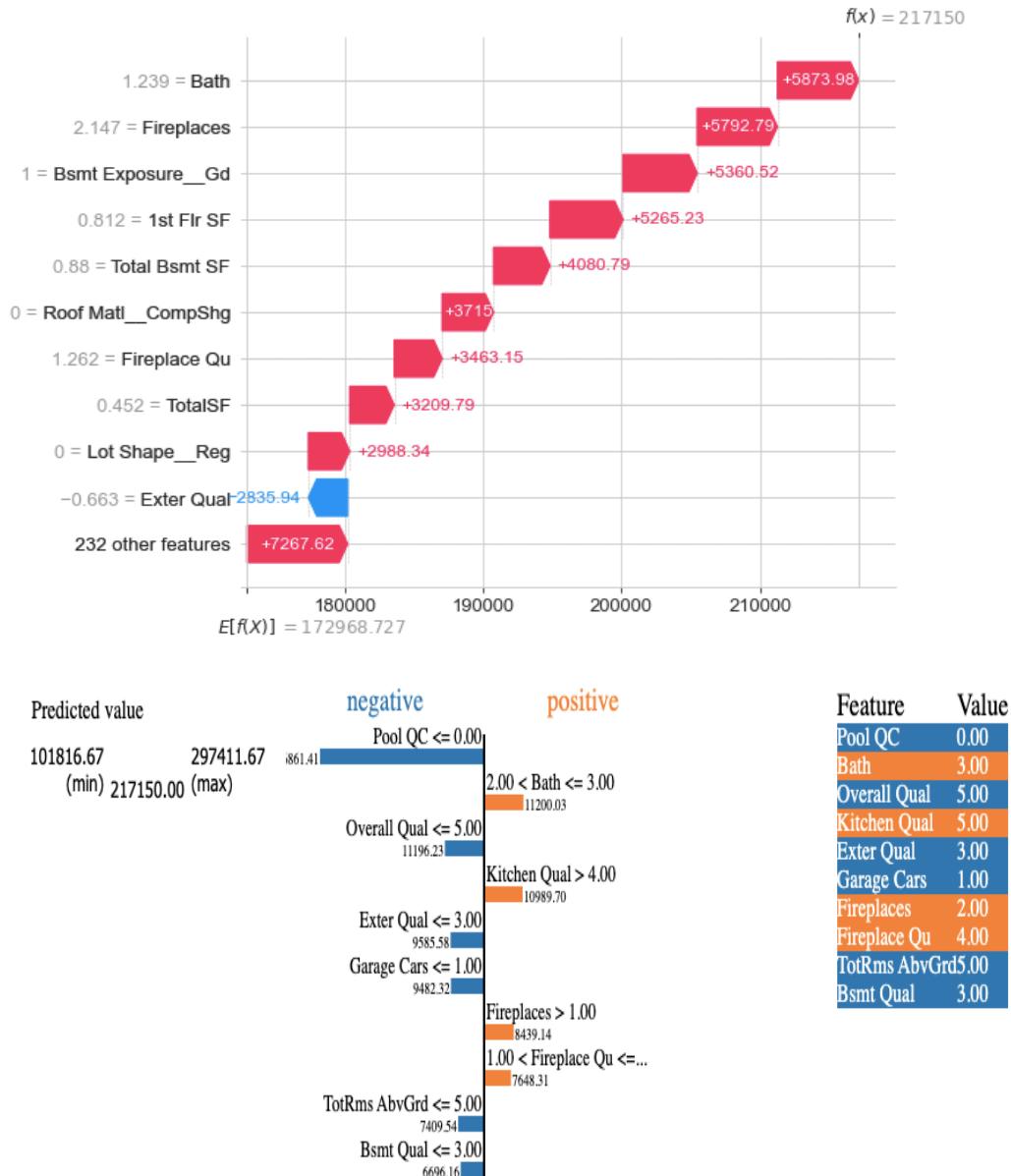


Abbildung 68: Überprüfung der Stabilität der Erklärungen des XGB und KNN-Modells

Im Rahmen der Trennbarkeit ist anhand der lokalen Analysen der Beobachtung 1 und 2 bereits zu erkennen, dass alle XAI-Verfahren unter Anwendung aller Algorithmen das Kriterium erfüllen. Jedoch muss diese Erkenntnis für den XGBoost Algorithmus mit vorsichtig betrachtet werden, da gerade baumartige Algorithmen mehrere Datenpunkte über Baumpfade oder Regeln einordnen und somit eine Trennbarkeit nicht erstrebenswert ist. Weiterführend werden die Ergebnisse hinsichtlich des Konsistenzkriteriums geprüft. Dabei zeigt sich, dass minimale Änderungen der Datenpunkte zu ähnlichen Ergebnissen der Beobachtungen 1 und 2 führen. Essenziell ist dabei welche Variable angepasst wird. So

führen in der lokalen Beobachtung minimale Anpassungen der einflussreichen Variablen *TotalSF*, *Overall Qual*, *Year Built* zu größeren Änderungen in der Erklärung als die Änderung anderer Variablen, deren Beiträge für die Vorhersage nicht so einflussreich ausfallen. Dies lässt sich für alle Modelle und XAI-Anwendungen beobachten. Die Ergebnisse für das XGB-Modell mittels LIME und SHAP sind für die Beobachtung 1 in der Abbildung 69 beispielhaft dargestellt.

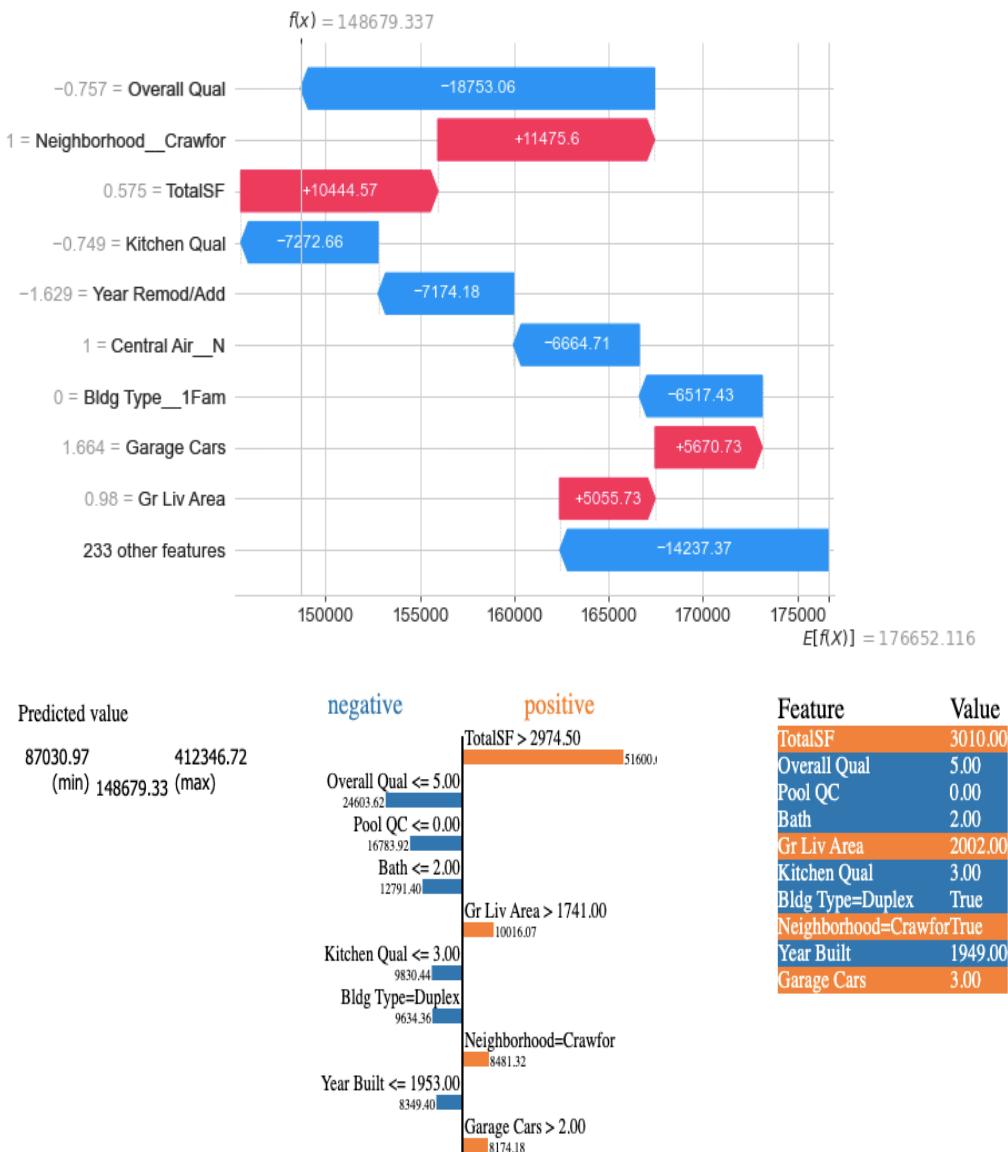


Abbildung 69: Überprüfung der Konsistenz der Erklärungen des XGB-Modells

Hierbei werden für das XGB-Modell minimale Anpassungen für die Variablen *TotalSF*, *Year Built* und *Garage Area* vorgenommen. Die genauen Anpassungen sind in der

Tabelle 15 zu entnehmen und werden für alle Modelle genutzt. Dabei ist zu beachten, dass die Anpassungen ausgeschlossen von möglichen Interaktionseffekten und Multikollinearität untersucht werden und die Erklärbarkeit mittels der XAI-Verfahren im Vordergrund steht.

Tabelle 15: Änderungswerte der Variablen

Variable	Ausgangswert B1	Ausgangswert B2	Anpassung Beobachtung 1	Anpassung Beobachtung 2
TotalSF	3.003	2.910	3.010	2.905
Year Built	1950	1968	1949	1966
Garage Area	871	452	890	445

Letztlich erfolgt die Auswertung der Erklärungsähnlichkeit. Die Ergebnisse lassen sich bereits für die globale und lokale Analyse aller Modelle komprimiert den Tabellen 16 bis 17 entnehmen. Dabei zeigt sich, dass global gesehen alle Modelle anhand der angewandten XAI-Verfahren ähnliche Erklärungen abgeben, die sich punktuell unterscheiden. Lokal betrachtet lässt sich für die Beobachtung 1 und 2 erkennen, dass die Gewichtung der Variablen übergreifend je nach Modell und XAI-Verfahren weniger Ähnlichkeiten aufweisen und überwiegend keine übereinstimmende Erklärung liefern. Demnach unterschiedet sich die Ergebnisfindung der Modelle im übergreifenden Vergleich stark. Die Abbildung 70 zeigt dabei die Gegenüberstellung der Beobachtung 1 für das XGB-Modell. Hierbei wird deutlich, dass für den gleichen Datenpunkt unter Anwendung des gleichen Modells unterschiedliche Gewichtungen und Erklärungen seitens SHAP und LIME ausgegeben werden. Während LIME die Gesamtflächengröße und die Gesamtqualität als die einflussreichsten Variablen darstellt, gibt SHAP die Gesamtqualität und die Lage als wesentlich beeinflussend für das Vorhersageergebnis an. Hierbei lassen sich eindeutig Widersprüche in den Erklärungen erkennen. So wird die Lage innerhalb von SHAP als wesentlich einflussreicher beurteilt, als es von LIME ausgegeben wird. Ebenso lässt sich beobachten, dass die Gesamtfläche *Total SF* seitens SHAP mit einem deutlich geringeren Einfluss und Beitrag dargestellt wird als von LIME. Dies verdeutlicht nochmals stark der Beitrag der Variable *TotalSF*, dervon LIME mit einer Höhe von 50.429 \$ ausgewiesen wird. SHAP im Gegensatz, schreibt *TotalSF* einen 10-mal kleineren numerischen Einfluss zu. Allein anhand des XGB-Modells wird deutlich, mit welchen Schwierigkeiten

die Bewertung einer korrekten Erklärung des gleichen Modells unter Anwendung unterschiedlicher XAI-Methoden einhergehen.

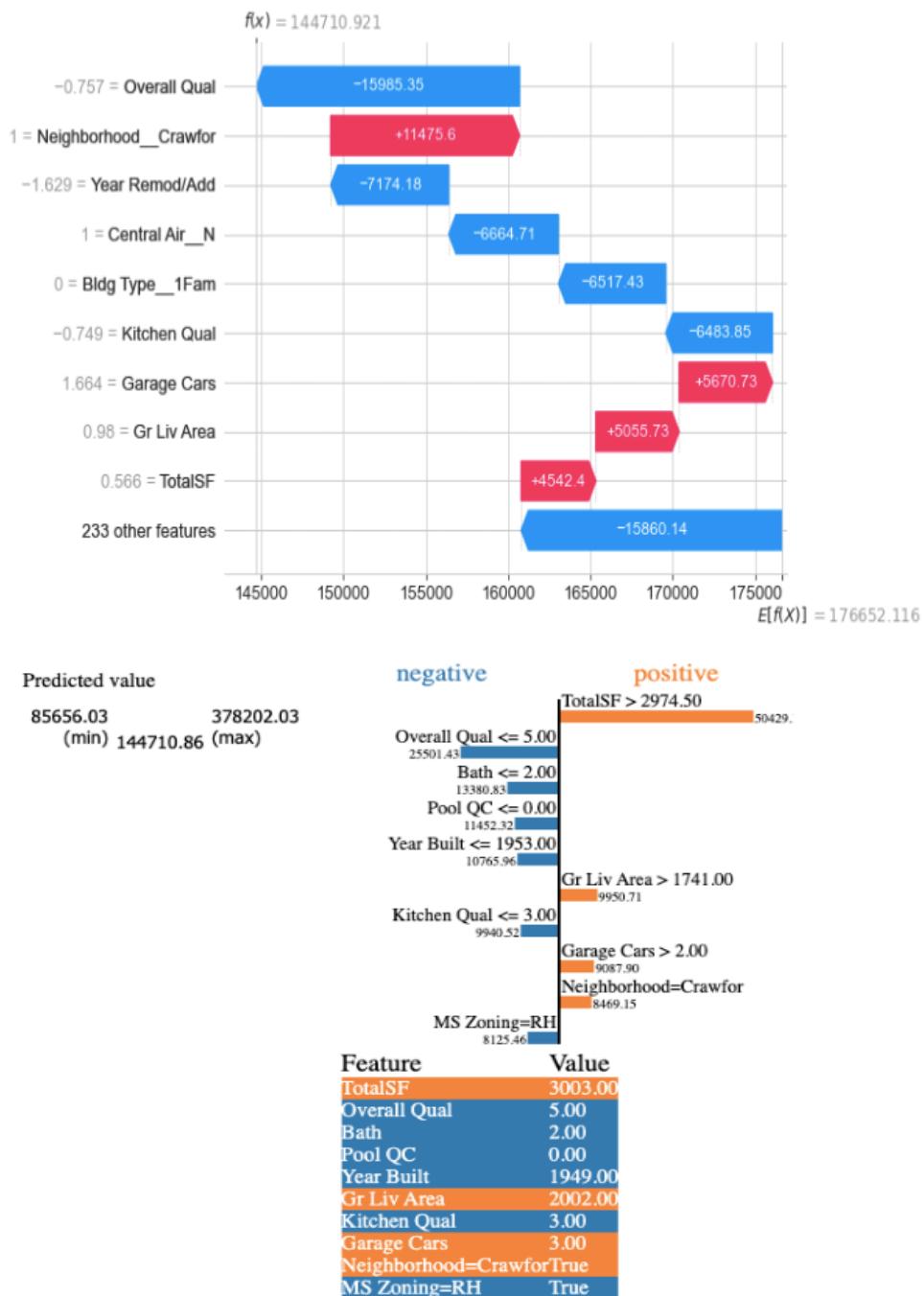


Abbildung 70: Überprüfung der Erklärungsähnlichkeit der Erklärungen des XGB-Modells

Zieht man den Vergleich der Abbildung 71 hinzu, der die Gegenüberstellung von dem KNN- und XGB-Modell betrachtet, so fallen Widersprüche in der Erklärungsähnlichkeit auf. So zeigt sich in der übergreifenden Betrachtung weitergehend, dass unter SHAP das

XGB-Modell der Gesamtqualität und der Lage einen höheren Beitrag für das prädiktive Ergebnis zurechnet. Im KNN-Modell ist wiederum zu beobachten, dass die Lage nicht innerhalb der Top zehn einflussreichsten Variablen vertreten ist und nur einen quantitativen Einfluss von +103 \$ ausübt. Hierbei kristallisiert sich nochmals stärker ein Widerspruch zwischen zwei ähnlich gut performenden Modellen heraus. Dies stellt im Anbetracht der Zielsetzung eine erhebliche Herausforderung dar, die sowohl interdisziplinär als auch modellübergreifend auftritt.

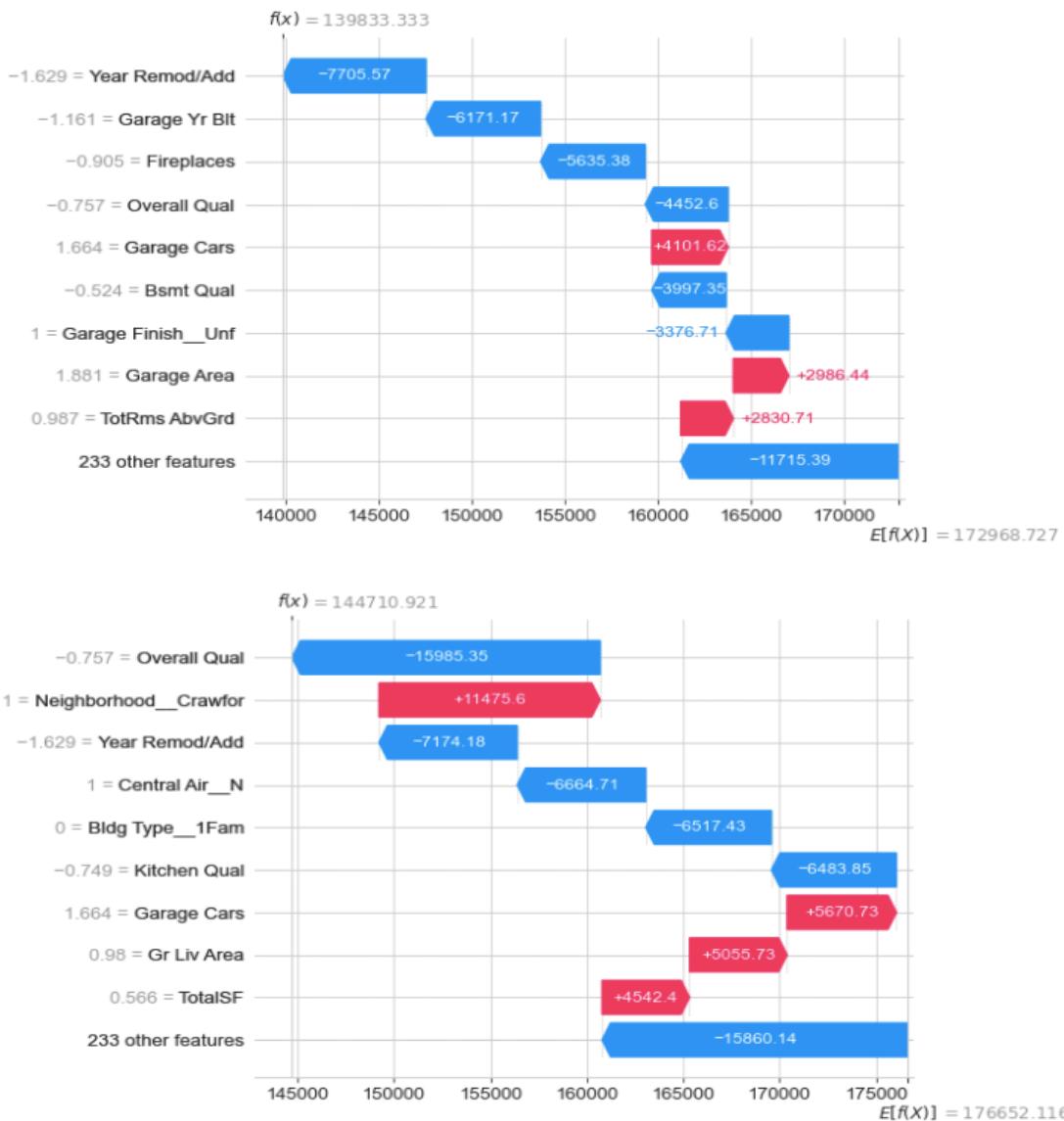


Abbildung 71: Überprüfung der Erklärungsähnlichkeit der Erklärungen des XGB- und KNN-Modells

Im Anbetracht der ähnlich guten Vorhersageergebnisse der multiplen Modelle wird das soeben beschriebene Phänomen als Rashomon-Effekt definiert. Dieser impliziert, dass

verschiedenen statistische- oder ML-Modelle zu ähnlich guten Vorhersageergebnisse für denselben Datensatz führen können. Anhand der beiden intensiv betrachteten Beispiele zeigt sich jedoch, dass die Erklärung zur Ergebnisfindung der Modelle Unterschiede aufweisen können und ebenso dem Rashomon-Effekt unterliegen. Dieser zeichnet sich nicht nur im Bezug der Multiplizität von guten Modellen ab, sondern ebenso innerhalb der Multiplizität von guten Erklärungen (Breiman, 2001, S.200; Leventi-Peetz & Weber, 2023, S.806). Anhand der Beispiele wird die Herausforderung, die der Rashomon-Effekt im Rahmen der Erklärungsähnlichkeit mit sich bringt, deutlich. Der Rashomon-Effekt und die daraus resultierenden Herausforderungen für die Praxis werden näher in der Beurteilung der Ergebnisse diskutiert.

Tabelle 16: Erklärungsähnlichkeit Global

Global												
ANN-SHAP	-	o	-	⊗	⊗	o	⊗	⊗	o	⊗	⊗	⊗
ANN-LIME	o	-	-	⊗	⊗	⊗	o	o	o	⊗	⊗	⊗
ANN-IML	-	-	-	-	-	-	-	-	-	-	-	-
KNN-SHAP	⊗	⊗	-	-	o	⊗	⊗	⊗	⊗	⊗	⊗	⊗
KNN-LIME	⊗	⊗	-	o	-	o	o	o	o	⊗	⊗	⊗
KNN-IML	o	⊗	-	⊗	o	-	o	o	o	o	o	o
XGB-SHAP	⊗	o	-	⊗	o	o	-	⊗	⊗	⊗	o	o
XGB-LIME	⊗	o	-	⊗	o	o	⊗	-	⊗	⊗	o	o
XGB-IML	o	o	-	⊗	o	o	⊗	⊗	-	⊗	⊗	⊗
EBM-SHAP	⊗	⊗	-	⊗	⊗	o	⊗	⊗	o	-	⊗	⊗
EBM-LIME	⊗	⊗	-	⊗	⊗	o	o	o	⊗	-	⊗	⊗
EBM-IML	⊗	⊗	-	⊗	⊗	o	o	o	⊗	⊗	⊗	-

Tabelle 17: Erklärungsähnlichkeit Lokal

Lokal													
ANN-SHAP	-	o	-	x	x	x	x	x	x	⊗	x	⊗	
ANN-LIME	o	-	-	x	⊗	x	x	x	x	x	⊗	x	
ANN-IML	-	-	-	-	-	-	-	-	-	-	-	-	
KNN-SHAP	x	x	-	-	x	⊗	⊗	x	x	x	x	x	x
KNN-LIME	x	⊗	-	x	-	x	x	x	x	x	x	x	x
KNN-IML	x	x	-	⊗	x	-	x	x	x	x	x	x	x
XGB-SHAP	x	x	-	⊗	x	x	-	x	⊗	x	x	o	
XGB-LIME	x	x	-	x	x	x	x	-	x	x	x	x	x
XGB-IML	x	x	-	x	x	x	⊗	x	-	x	x	x	x
EBM-SHAP	⊗	x	-	x	x	x	x	x	x	-	x	⊗	
EBM-LIME	x	⊗	-	x	x	x	x	x	x	x	-	x	
EBM-IML	⊗	x	-	x	x	x	o	x	x	⊗	x	-	

6 Diskussion der Forschungsergebnisse

6.1 Beurteilung der Ergebnisse

In Übereinstimmung mit den Ergebnissen aus den früheren Kapiteln werden die Ergebnisse in Bezug auf die Analysefelder und die Erklärungseigenschaften entlang der Zielsetzung diskutiert. Alle Ergebnisse der Beurteilung sind in den Abbildungen 72 bis 78 komprimiert zusammengefasst und werden nachfolgend ausführlich wiedergegeben und diskutiert.

- Gewichtung / Wichtigkeit von Variablen

Die Wichtigkeit und damit verbundene Gewichtung einzelner Variablen werden von den angewendeten XAI-Methoden entweder auf globaler oder lokaler Ebene beleuchtet. Hierbei zeigt sich, dass ein generelles Verständnis für die Bedeutung von Variablen für die Ergebnisfindung des Modells durch die globale Betrachtung gewonnen werden kann. Demnach kann dem jeweiligen Nutzer der XAI-Methode verdeutlicht werden, welche Variablen in der gesamtheitlichen Betrachtung einen starken Einfluss auf das prädiktive Ergebnis ausüben. Dies ermöglicht es die wesentlichen Treiber für das Modell zu erkennen und zu analysieren. Dabei schaffen gerade das PDP-Diagramm des InterpretML-Moduls und die SHAP-Methode die Möglichkeit, den globalen Eindruck auf einzelne Variablen und ihre Ausprägungen herunterzubrechen. Dies führt nicht nur zur Differenzierung, welche Variablen eine hohe Gewichtung im Modell erfahren, sondern auch zur Klärung, bei welcher Ausprägung eine Variable als bedeutend für die Vorhersage angesehen wird. Dadurch wird simultan die Bedeutung und das Verständnis der Wichtigkeit einzelner Variablen verdeutlicht. Diesbezüglich geht das Dependence-Plot des TreeExplainers einen Schritt weiter und fördert das Verständnis der Abhängigkeit einer Variablen nicht nur zur Zielvariable, sondern ebenso zu einer weiteren unabhängigen Variable. Dies ermöglicht die Funktionsweise des Modells zu erschließen und die Interaktionseffekte zwischen den unabhängigen Variablen in Bezug auf ihre Bedeutung und Multikollinearität zu untersuchen. Obwohl einige Variablen möglicherweise von den XAI-Methoden als weniger bedeutend für die Vorhersage der Zielvariable beurteilt werden, können sie trotzdem andere relevante Variablen beeinflussen, die im Modell stark gewichtet sind.

Ebenso können zum Abgleich die Ergebnisse der Literaturrecherche hinzugezogen werden. Dabei zeigt sich, dass Gemeinsamkeiten in der identifizierten Literatur mit den

Ergebnissen der globalen XAI-Anwendung beobachtet werden können. So wird die Fläche in nahezu allen literarischen Werken als wichtigste Variable benannt, was von den Ergebnissen der Feature Importance Analyse der verschiedenen Methoden bestätigt wird. Unterschiede zeigen sich im Hinblick auf die Auswirkungen der Gesamtqualität, der Lage und der Anbindung an die Infrastruktur. In Bezug auf die Gesamtqualität wird diese nur direkt von Sandbhor und Chaphalkar (2016) angesprochen und indirekt von Jafari und Akhavian (2019) über das Modernisierungsjahr betrachtet. Nach allen Erkenntnissen der XAI hat die Distanz zur öffentlichen Infrastruktur, wie Bildungseinrichtungen, Krankenhäusern und öffentlichen Verkehrsmitteln, keinen starken Einfluss. Hervorzuheben ist dabei, dass die Lage nur in den lokalen Analysen der Beobachtungen und den globalen Analysen durch LIME und InterpretML als relevant eingestuft ist. Diese Unterschiede sind abhängig vom jeweilig gewählten Algorithmus. Gerade im Hinblick der unterschiedlichen Modellergebnisse rückt besonders der Rashomon-Effekt in den Vordergrund. Dieser kann hauptsächlich mittels der Feature Importance Darstellung und Gegenüberstellung aufgezeigt werden und wird nachfolgend im Kontext der Stabilität diskutiert. Gerade für Immobilienunternehmen, die neben der Fläche auch stark die Lage bei der Festlegung von Verkaufspreisen berücksichtigen, werden die Erklärungen von LIME und InterpretML eher als zielführend einstufen als die Ergebnisse von SHAP. Darüber hinaus ermöglichen lokale Analysen die wesentlichen Treiber der Modellentscheidung zu identifizieren. Dabei wird besonders in Beobachtung 2 deutlich, wie relevant die Bestimmung der Wichtigkeit einzelner Variablen ist. Ohne den Einsatz der XAI-Methoden wäre es nicht erkennbar, welche Variablen zu den Unterschieden zwischen den prognostizierten und tatsächlichen Verkaufspreisen beitragen. Hierbei werden möglicherweise Variablen aufgeführt, deren Gewichtung für den Preis seitens der Immobilienexperten bisher über- oder unterschätzt werden.

Weiterführend erlaubt die Quantifizierung der globalen und lokalen Beiträge der Variablen nicht nur die Vorhersage eines komplexen ML-Modells, sondern auch die quantitative Messung der Auswirkungen der Inputs auf das Modellergebnis. Demnach lässt sich den Ergebnissen der Untersuchung entnehmen, dass die Wertigkeit des Beitrages der Variable in Dollar eine Messbarkeit der Wichtigkeit erlaubt. Somit können Vergleiche zwischen den Variablen auf globaler und lokaler Ebene ermöglicht werden. Zusätzlich ermöglicht die Darstellung der Feature Importance in Werten eine quantitative Messung von nicht

linearen und nicht monotonen ML-Funktionen, wie sie beispielsweise in Blackbox-Modellen zu finden sind (Hall et al., 2017, S.4; Owen & Prieur, 2017, S.986-987). Dabei muss beachtet werden, wie die Interpretation der Variablen in der Gewichtung gedeutet wird. Nach der Analyse von Kumar et al. (2020) werden die Gewichtungen der Variablen in der SHAP-Analyse mit dem eigenen mentalen Modell, das auf den Erfahrungswerten basiert, verglichen. Dies impliziert die Gefahr, dass auf Basis des eigenen mentalen Modells entschieden wird, ob ML-Modelle in der Praxis ausgerollt werden. Dies verfehlt den Sinn von XAI-Methoden und kann innerhalb der Gewichtung von Variablenbeiträgen zu fehlerhaften Annahmen führen. Diese Annahme unterstützt auch die Studie von Weerts et al. (2019). Sie kommt zu dem Ergebnis, dass es keine Beweise dafür gibt, dass die Darstellung der Feature Importance mithilfe von SHAP hilfreich ist, um die Korrektheit einer Vorhersage von Menschen zu beurteilen.

Nichtsdestotrotz ist die wertige Darstellung der Variablen ein weiterer Schritt in Richtung der Verständlichkeit für die Zusammensetzung von Immobilienpreisen. Dies ermöglicht es, auf Basis der vorhandenen Daten, ähnliche Objekte miteinander zu vergleichen und Entscheidungen für Käufe und Verkäufe datengetrieben zu treffen. Die Beitragshöhe einzelner Variablen ermöglicht es, Entscheidungen gesamtheitlich auf einzelne Faktoren herunterzubrechen und dadurch verständlicher zu begründen.

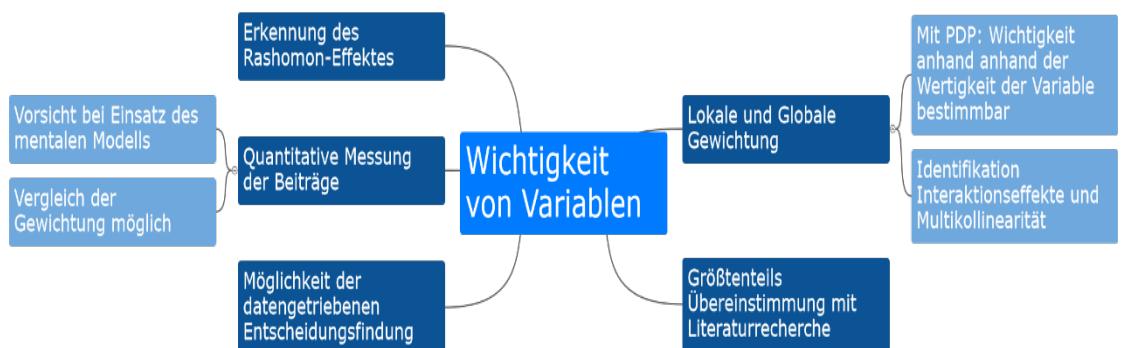


Abbildung 72: Zusammenfassung der Beurteilung - Wichtigkeit von Variablen

- Modell Optimierung / Debugging

Innerhalb der Modell Optimierung und des Debuggings, fördert der Einsatz von XAI-Methoden die Möglichkeit, Optimierungspotenziale auszuschöpfen und Modellfehler zu identifizieren. Hierbei zeigte die Untersuchung, dass unter Anwendung von LIME ein gesonderter Umgang mit kategorialen Variablen erfolgen muss, um unverzerrte

Ergebnisse für die lokale Betrachtung zu erlangen. Die kategorialen Variablen werden unverhältnismäßig seitens LIME gewichtet, da die XAI-Methode jede einzelne Ausprägung separat gewichtet und die dazugehörigen kategorialen Variablen nicht kumuliert. Demnach zeigte sich bereits in einem frühen Stadium der Untersuchung, dass Optimierungen nicht nur im Rahmen der ML-Modelle möglich sind, sondern der übergreifende Einsatz von XAI Methoden Optimierungspotenziale darstellt. Die Ergebnisse der XAI-Methoden können weiterführend für ein erneutes Feature Engineering genutzt werden. Dabei erlaubt gerade die globale Analyse der Methoden die gesamtheitlichen treibenden Variablen zu erkennen und auch die Variablen, die keinen oder nur einen besonders kleinen Einfluss auf das prädiktive Ergebnis ausüben. Dies bietet die Grundlage zur Überdenkung der Variablenanzahl für das Modell Training, um beispielsweise für das ANN-Model, die Anzahl an Dimensionen zu verkleinern und somit entgegen dem Fluch der Dimensionalität zu arbeiten. Hierbei muss jedoch stets bedacht werden, dass mit jeder Variable, die nicht für die Vorhersage verwendet wird, ebenso Informationen und ein Teil der Realität verloren gehen (Fryer et al., 2021, S.144359).

Gleichzeitig können die Variablen, die keinen Einfluss auf das prädiktive Ergebnis ausüben, genauer untersucht werden. So zeigt sich im Rahmen der Untersuchung, dass mit LIME und InterpretML die Variable *PoolQC* mit dem Wert 0 einen besonders starken Einfluss auf das prädiktive Ergebnis hat. Die anderen Ausprägungen zeigen kaum Einfluss und sind auch in ihrer Häufigkeit gering vertreten. Dies gibt Anlass zur Überprüfung, ob die ungleichmäßige Verteilung der Variable zu einer Verzerrung in der Erklärbarkeit des Modells führt. Dies bietet die Grundlage für weitere Analysen bezüglich möglicher struktureller Schwächen im Modell. Damit einhergehend kann dies ein Indiz von unzureichenden Daten sein. Zusätzlich kann der Einsatz von XAI-Methoden auch dazu beitragen, Mängel in Bezug auf Datenqualität und Datenmenge aufzudecken, indem die Ergebnisfindung auf Variablen mit einer ungewöhnlich starken Gewichtung untersucht werden. Darüber hinaus kann XAI wie bereits erwähnt, die Identifikation von verzerrten Modellergebnissen erleichtern. Diese Gefahr besteht beim Einsatz von undurchsichtigen Blackbox-Modellen wie dem ANN kontinuierlich. Gerade die globalen Ansätze können hier Abhilfe schaffen, indem sie ein Verständnis dafür schaffen, welche Variablen oder Ausprägungen seitens der Diskriminierung des Modells betroffen sind (Gerlings et al., 2021, S.1287). Nichtsdestotrotz muss beachtet werden, dass die erlangten

Erkenntnisse aus den Untersuchungen der XAI-Methoden zur Optimierung und zum Debugging von Modellen nicht beständig trivial sind. Die Erkenntnisse sind nützlich, jedoch gleichzeitig komplex und abhängig von der jeweils gewählten XAI-Methode, dem ML-Modell, dem Datensatz und dem vollzogenen Feature Engineering. Dies ist besonders bei der Untersuchung des KNN-Modells im Rahmen des Feature Engineerings, bei der Anwendung eines Normalisierungsverfahrens, deutlich geworden. Wie die diversen Punkte bereits verdeutlichen, kann die Einbindung von XAI-Methoden das Verständnis von ML-Modellen hinsichtlich der Optimierung und Fehlerbehebung fördern. Für Entwickler von ML-Modellen ist dies eine wertvolle Möglichkeit, um die Modellstruktur besser zu verstehen und frühzeitig Fehler und Optimierungspotenzial zu erkennen (Weber et al., 2022, S.170).

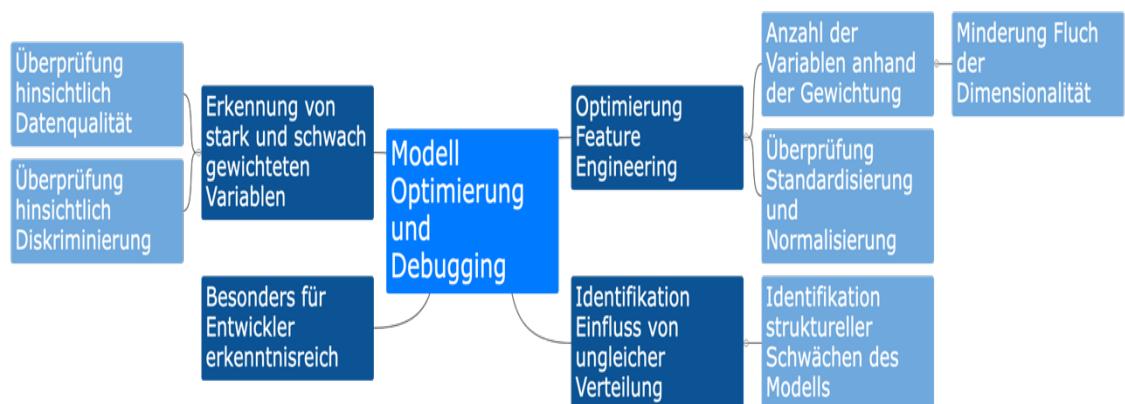


Abbildung 73: Zusammenfassung der Beurteilung – Optimierung und Debugging

- Modell Fairness, Ethische Korrektheit und Risikokontrolle

Die Erkenntnisse der Gewichtung der Variablenbeiträge ermöglichen es, die Funktionsweise und die Ergebnisfindung des Modells hinsichtlich der Fairness, der ethischen Korrektheit und der Risikokontrolle zu prüfen. Gerade dies ist unter Beachtung des Rashomon-Effektes ein notwendiger Schritt, um die Entscheidung für ein Modell nicht nur auf Basis der Performance zu treffen. Demnach bieten die Ergebnisse der XAI-Anwendung die Möglichkeit, die Auditierung der verwendeten Algorithmen zu vollziehen und die Abwägung der Modelle hinsichtlich der Fairness und der ethischen Korrektheit vorzunehmen. Dabei können Variablen entlang der globalen und lokalen Analyse erkannt werden, die im Widerspruch zu den beiden genannten Bereichen stehen. So werden im Rahmen der Untersuchung keine soziologischen Faktoren miteinbezogen. Gerade diese

können seitens des Modells diskriminiert werden, indem bestimmte Gruppen oder Ethnien in den Erklärungen nicht aufgeführt sind oder gegen ethische Grundprinzipien Verstoßen wird (Adadi & Berrada, 2018, S. 52142). Dabei spielt die ethische Korrektheit von KI in der derzeitigen Forschung und Praxis eine große Rolle und wird des Öfteren mit XAI als Lösungsansatz in Verbindung gebracht (Gerlach et al., 2022, S.15 ff; John-Mathews, 2022, S.18; Vale et al., 2022, S.18 ff; G. Zhang, 2022, S.3). Innerhalb der Untersuchung können keine eindeutigen Diskriminierungen oder ethische Unkorrektheiten in den Erklärungsergebnissen festgestellt werden. Die Untersuchung zeigt jedoch, dass die Möglichkeit entlang aller XAI-Methoden besteht, die Variablen im Sinne der Immobilienbranche auf diese Missstände zu prüfen. Die Ergebnisse der XAI-Methoden ermöglichen es, spezifische Beobachtungen mit den lokalen Ansätzen zu prüfen und ein Gesamtverständnis bezüglich der Funktionsweise des Modells mit den globalen Ansätzen zu erlangen.

Des Weiteren sind Risikokontrollen hinsichtlich der Variablengewichtung möglich, die gerade im Rahmen des juristischen Umfelds, der Cyber-Security oder im medizinischen Umfeld unabdingbar sind. Dies zeigt ebenso die Einführung des Datenschutz-Grundverordnung (DSGVO) der europäischen Union (EU) im Jahre 2018. Die DSGVO regelt im Allgemeinen die Datenschutzverordnung hinsichtlich der Sammlung und Verarbeitung von personenbezogenen Daten. Für den Einsatz von Machine Learning ist es demnach zukünftig notwendig, innerhalb der EU die Erklärbarkeit von automatisierten Prozessen oder Vorhersagen sicherzustellen, die personenbezogene Daten enthalten. Dabei ist der derzeitige Umfang der Erklärbarkeit nicht definiert (Rudin, 2019, S. 8-9; Sovrano et al., 2021, S.172-173). Diesbezüglich zeigte sich bereits in der Untersuchung, dass sich der Umfang der Erklärbarkeit je nach Methode unterscheidet. So bieten SHAP und InterpretML tiefgehende Einblicke anhand von verschiedenen grafischen und methodischen Möglichkeiten. LIME hingegen tritt diesbezüglich begrenzt in Erscheinung. Für die Immobilienunternehmen bedeutet dies im Umkehrschluss, dass die Preisprognosen mittels ML transparent und nachvollziehbar zu gestalten sind, was eine Einführung von XAI nahezu unabdingbar erscheinen lässt.

Ebenso ermöglichen die XAI-Methoden die Einhaltung der Compliance Vorschriften in Bezug auf die implementierten ML-Modelle. So kann im Umkehrschluss eine Risikokontrolle im eigenen Immobilienunternehmen hinsichtlich der regulatorischen Grundlagen

vollzogen werden. Hierbei kann das Unternehmen sich für Modelle entscheiden, welche den Voraussetzungen entsprechen. Dies ermöglicht, die Eintrittsbarrieren für die Nutzung von ML-Modellen in systemkritischen Bereichen zu senken, indem XAI-Modelle erklärbare Ansätze schaffen, die risikobehaftete Entscheidungen mit mehr Informationen unterstützen können. So können auch wirtschaftlich kritische Entscheidungen, die in der Vergangenheit allein auf der Erfahrung und Praxiskompetenz von Menschen basieren, datenfundiert unterstützt werden. Die Einbeziehung von XAI im Rahmen von ML fördert gleichzeitig das Vertrauen in die Nutzung von Maschinen und treibt datengetriebene Entscheidungen voran (Gerlings et al., 2021, S.1290; Lauritsen et al., 2020, S.2).

Gerade die Nutzung des InterpretML-Moduls schafft Vorteile zur Förderung des Verständnisses bezüglich der Risikoidentifikation. Wie die Untersuchung aufzeigt, kann ein kumulierter Überblick aller genutzten Modelle geschaffen werden, um alle Modellergebnisse gesamtheitlich gegenüberzustellen und hinsichtlich der Risikokontrolle zu prüfen. Gerade dies ist erneut im Entgegenwirken des Rashomon-Effektes im Rahmen der Multiplizität von guten Modellen von enormer Bedeutung und bietet die Chance eine Validierung der Modelle nicht nur hinsichtlich ihrer Performance zu erzielen, sondern ebenso entlang von ethischen, rechtlichen und risikoreichen Aspekten.

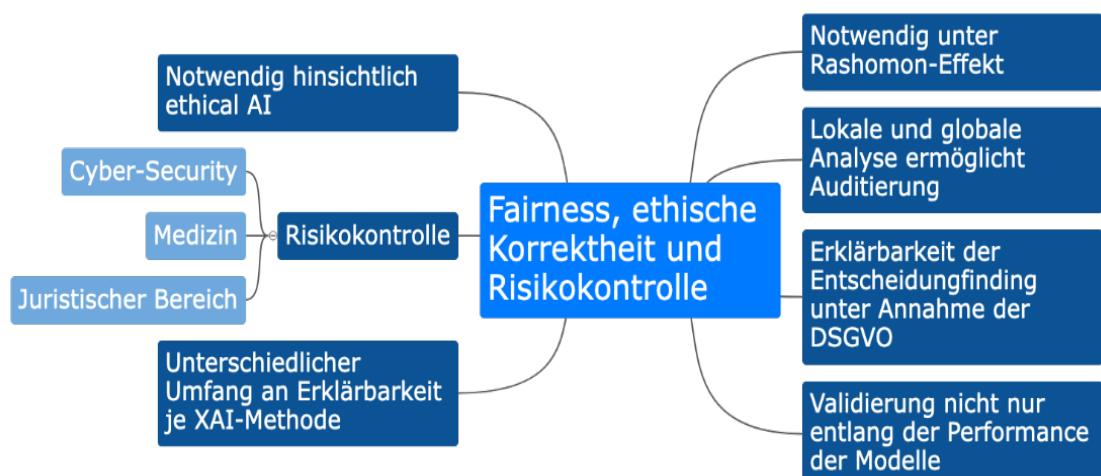


Abbildung 74: Zusammenfassung der Beurteilung – Fairness, ethische Korrektheit, Risikokontrolle

- Transparenz

Hinsichtlich der Transparenz tragen die XAI-Methoden mit der Darstellung diverser Faktoren zum umfassenden Verständnis wesentlich bei. So zeigt sich im Rahmen der Untersuchung, dass die einzelnen lokalen Methoden die preisliche Beitragshöhe der Variablen

unter Beachtung der Zielvariable aufschlüsseln. Dies fördert zunächst die Transparenz hinsichtlich des dadurch entstehenden Relationsvergleichs und ermöglicht es die Beiträge gesamtheitlich anhand der Gewichtung hierarchisch zu ordnen. Dadurch kann die Funktionsweise des Modells sowohl auf lokaler als auch auf globaler Ebene transparenter dargestellt werden. Im weiteren Sinne der Transparenz ermöglicht die Anwendung der XAI-Methoden die transparente Begründung für die Auswahl spezifischer Modelle. So zeigt die Untersuchung, dass die Anwendung der XAI-Methoden auf verschiedene ML-Modelle zu teilweise unterschiedlichen Ergebnissen gelangen. Die transparente Funktionsweise der Modelle schafft die Möglichkeit, das Einsetzen eines bestimmten Modells zu begründen. Gerade im Kontext der Multiplizität von Modellen ist es von enormer Bedeutung, eine fundierte Argumentationsgrundlage auf Basis transparenter Funktionsweisen und damit verbundener Ergebnisse zu haben. Dabei kann ein transparentes Modell hochgradig die Akzeptanz der Nutzer steigern (Páez, 2019, S.452 ff.).

Die verschiedenen grafischen Darstellungen tragen dazu bei, das Verständnis der Funktionsweise des Modells zu fördern. So zeigt die Untersuchung, dass SHAP und InterpretML im Gegensatz zu LIME mehrere grafische Möglichkeiten anbieten, die die Transparenzdichte wesentlich durchdringbarer gestaltet. So kann in der globalen Analyse der globale Effekt einzelner Dateninstanzen gesamtheitlich und transparent anhand des Beeswarm- Dependence- und PDP-Diagramms betrachtet werden. Dies ermöglicht die Funktionsweise des Modells auf einzelne Dateninstanzen transparent darzustellen und zu untersuchen. Beispielsweise lässt sich für das XGB-Modell anhand der Balkendiagramme der XAI-Methode SHAP entnehmen, dass dieses eine global hohe Beitragsgewichtung auf einzelne Variablen legt, wohingegen die globale Gewichtung der Variablen im KNN-Modell überwiegend verteilt ausfällt. Ebenso lässt sich ableiten, dass das XGB-Modell für den vorliegenden Datensatz eher qualitative Variablen hoch gewichtet als kategoriale. Im Vergleich gewichtet das KNN-Modell eher beide Skalenniveaus ausgeglichen. Hier zeigt sich die Möglichkeit der transparenten Ableitung des Modells anhand des vorliegenden Datensatzes. Die lokalen Analysen zeigen, dass spezifische Beobachtungen transparent dargestellt werden, um somit die Entscheidungsunterstützung für diese speziellen Fälle zu ermöglichen. Gerade für hochpreisige Immobilien ist Transparenz von enormer Bedeutung, da fehlgeleitete Entscheidungen infolge einer mangelnden Übersicht zu

erheblichen wirtschaftlichen Verlusten führen und das Immobilienunternehmen in eine Insolvenz treiben können.

Eine in der Untersuchung ebenso aufgeführte Thematik ist die Gegenüberstellung der Whitebox- und Blackbox-Modelle. Dabei zeigt die Untersuchung, dass das EBM-Modell eine mindestens ebenbürtige Leistung im Vergleich zu den Blackbox-Modellen vorweist. Die transparente Struktur des EBM-Modells wirft die Frage in der Forschung auf, ob weiterhin die Notwendigkeit besteht Blackbox-Modelle mithilfe von XAI-Methoden erklärbar zu machen. Wie bereits dargestellt, besteht die Möglichkeit Whitebox-Modelle zu konzipieren, die bereits die Eigenschaft der Transparenz aufweisen und eine ähnlich gute Performance erzielen. Gerade für systemkritische Entscheidungen ist die Transparenz von Modellen unabdingbar, um fundierte Entscheidungen treffen zu können. Befürworter argumentieren, dass die Transparenz bereits im Ansatz des Modells verankert sein sollte, um von Anfang an einen Faktor für die Vertrauensbildung zu schaffen und nicht erst nachträglich die Erklärbarkeit beizusteuern (Rudin, 2019, S.15). Ebenfalls wird auch argumentiert, dass die geschaffene Transparenz bei Blackbox-Modellen nicht ausreicht, um sie für Entscheidungsträger oder Adressaten vollständig vertrauenswürdig zu machen. Die eigene Überzeugung des Empfängers, die Tiefe der Erklärbarkeit der XAI-Methoden und die Relevanz der bereitgestellten Informationen spielen dabei eine wesentliche Rolle (Ananny & Crawford, 2018. S.977 ff; Gerlings et al., 2021, S.1287-1288).

Hierbei kann das InterpretML Modul eine geeignete Lösung darstellen, da bereits die Untersuchung zeigt, dass die Vereinigung von diversen XAI-Ansätzen innerhalb eines Moduls diverse Standpunkte der Erklärbarkeit ermöglicht und somit die Transparenz der Modelle durch mehrere bewährten Ansätze hebt. Letztlich muss jedoch beachtet werden, dass transparente Modelle bezüglich ihrer Funktion und Ergebnisfindung ein Potenzial für Cyberattacken darstellen. Hierbei kann die Anreicherung der Datenquellen auf Basis der Erkenntnisse des transparenten Modells zu manipulierten Ergebnissen führen. Demnach ist es essentiell die XAI-Methoden ebenso zur Modelloptimierung und Debugging der Erklärbarkeit zu nutzen, um eine vertrauenswürdige Transparenz zu gewährleisten (Capuano et al., 2022, S.93594 ; Z. Zhang et al., 2022, S.93127).

Allumfassend lässt sich ebenso anhand der Untersuchung erkennen, dass die verschiedenen Bereiche der Erklärbarkeit und Interpretierbarkeit, unterschiedliche Gruppen von Nutzern der XAI-Methoden ansprechen. Gerade in Bezug auf die Modelloptimierung und

das Debugging ist zu erkennen, dass deren Erklärbarkeit hauptsächlich den Entwicklern von ML-Modellen und Datenwissenschaftlern zugutekommt. Dies inkludiert ebenso die Transparenz des Modells zur besser Nachvollziehbarkeit der Funktionsweise, die mittels der Variablen Gewichtung hohe Einflüsse von unfairen oder ethisch unkorrekten Beiträgen erkennen lässt. Das gesamtheitliche Verständnis wird hierbei durch die globalen Analysen unterstützt und zur Analyse von spezifischen Gegebenheiten können die lokalen Analysen Abhilfe schaffen. Für die Präsentation oder zur Entscheidungsfindung für Manager und Fachexperten, ermöglichen die Risikokontrolle, die Gewichtung der Variablen und die Modeltransparenz eine informationsreichere Einordnung der ML-Ergebnisse. Dabei können anhand der Interpretierbarkeit der Modellergebnisse, Unternehmensrelevante Entscheidungen entlang der Compliance oder des rechtlichen Rahmens getroffen werden. Außerdem können die Erkenntnisse aus den Erklärungen bei der Entscheidungsfindung hinsichtlich wirtschaftlicher Aspekte berücksichtigt werden. Insgesamt wird auch anderen Stakeholdern, die von der Entscheidung von ML-Ergebnissen betroffen sind, ein Zugang zu der Entscheidungsfindung der Modelle ermöglicht, wodurch eine transparente und nachvollziehbare Zusammenarbeit mit den Maschinen gewährleistet wird (Barredo Arrieta et al., 2020, S.7-8; Gerlach et al., 2022, S.1 ff; Páez, 2019, S454 -456).

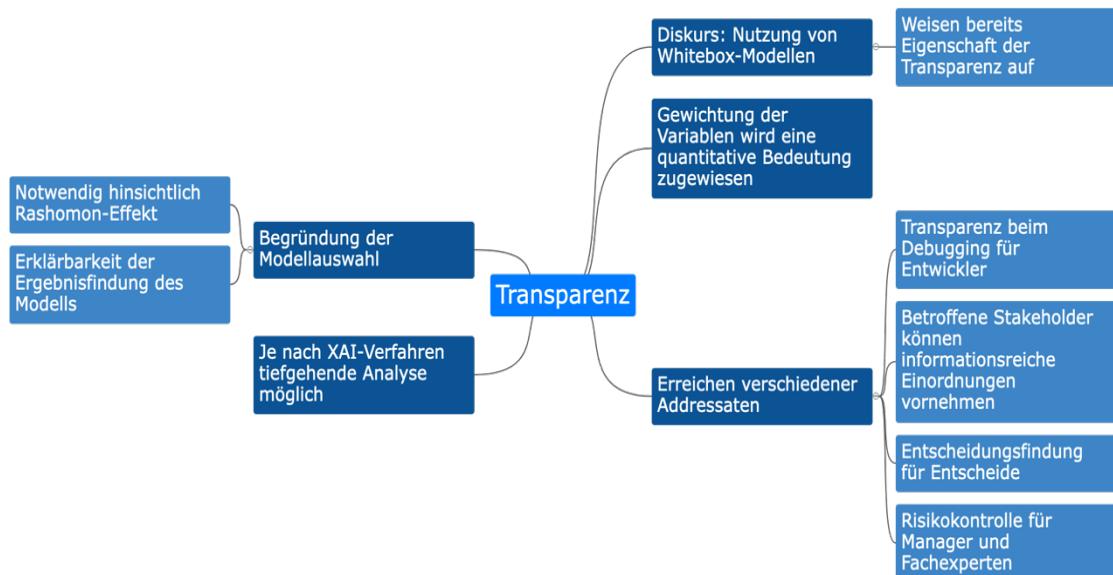


Abbildung 75: Zusammenfassung der Beurteilung - Transparenz

- Stabilität

Eine wesentliche Erkenntnis der Ausarbeitung ist, dass bei erneutem Durchlauf der genutzten XAI-Verfahren die Ergebnisse der lokalen Erklärung von LIME, SHAP und InterpretML für die Modelle variieren. Alvarez-Melis und Jaakkola (2018) unterstützen die Beobachtung, indem ihre Untersuchung zeigt, dass dieselben Erklärungen für zwei nahezu identische Datenpunkte unterschiedliche Ergebnisse produzieren können. Dies ist auf die Unsicherheit der modellagnostischen Methoden zurückzuführen, die aus der Zufälligkeit der Stichproben, der Variation in der Nähe zu den Stichproben und der Variation in der Glaubwürdigkeit des erklärten Modells für verschiedene Datenpunkte resultiert (Y. Zhang et al., 2019, S.3). Die Instabilität der Erklärungen kann demnach das Vertrauen in die XAI-Verfahren mindern, da jede neu generierte Erklärung gleichzeitig die Verlässlichkeit der Ergebnisse anzweifeln lässt. Zwar unterscheiden sich die Erklärungen der XAI-Methoden begrenzt und weisen den Variablen ähnliche Beitragshöhen zu, jedoch können bereits geringe Änderungen zu Unterschieden in den Ergebnissen führen. Demnach sind die Ergebnisse jedes Berechnungsdurchlaufes mit Vorsicht zu beobachten und kritisch zu hinterfragen. Hinzu kommt, dass im Rahmen der Stabilität von Wichtigkeit ist, anhand erneuter Durchläufe die stabilen Variablen zu identifizieren. Es ist jedoch zu beachten, dass zuverlässige und stabile XAI-Ergebnisse ebenfalls durch mögliches vorhandenes und nicht ausreichend bekämpftes Datenrauschen beeinträchtigt werden können (Ribeiro et al., 2016, S.98; Rosenfeld, 2021, S.48).

Im Gegensatz dazu sind die Ergebnisse des TreeExplainers stabiler als die der anderen XAI-Verfahren, da die baumartige Grundstruktur robuster gegenüber Veränderungen im Datensatz ist und nicht auf Annäherungen wie beispielsweise beim SHAP-Explainer oder LIME basieren (Lundberg et al., 2020, S.9). Dies trifft auch auf die lokale Erklärbarkeit des EBM-Modells zu, wenn der Glassbox-Explainer von InterpretML genutzt wird, und hängt ebenso mit der Konzeption von Whitebox-Ansätzen zusammen (Rudin, 2019, S.14 ff.). Insgesamt stellt die Instabilität der Erklärungen eine bedeutende Herausforderung für das Verständnis zwischen Menschen und Maschine dar. Diese Herausforderung kann jedoch durch die Verwendung von baumbasierten und Whitebox-Modellen eingeschränkt werden. Obwohl diese Ansätze ihre eigenen Herausforderungen mit sich bringen, ermöglichen sie es, durch den Einsatz mehrerer XAI-Verfahren und Algorithmen die Stabilität aus verschiedenen Perspektiven zu betrachten.

- Trennbarkeit und Konsistenz

Für die Trennbarkeit zeigt sich, dass alle XAI-Verfahren unter allen Modellen diese in den Ergebnissen vereinbaren. Für die Konsistenz erfüllen die Modelle teilweise die Bedingungen. Wie bereits beschrieben, führen minimale Änderungen zu keinen schwerwiegenden Änderungen in der Erklärung. Dies ist jedoch abhängig von der betrachteten Variablen sowie dem verwendeten Algorithmus. So zeigt sich, dass gerade baum- oder distanzartige Algorithmen wie das XGB- oder KNN-Modell stärker auf Änderungen in den Daten reagieren als beispielsweise das ANN- oder EBM-Modell (Arsov et al., 2019, S.1-3). Die Beachtung der Veränderung der möglichen Gewichtung kann je nach Verfahren und Möglichkeit nochmals durch die Beobachtungen von Interaktionsanalysen intensiviert werden, wie es die Untersuchung mittels des Dependence- oder PDP- Diagramme zeigt. Für das Verständnis zwischen Menschen und Maschine ist die Trennbarkeit als Erklärungseigenschaft unabdingbar und die Konsistenz eine Eigenschaft, die mit den vorhandenen Grundkenntnissen und Werkzeugen der XAI-Verfahren gut zu analysieren ist. Für die Untersuchung lässt sich festhalten, dass minimale Veränderungen im Datenbestand und den Variablen keine großen Auswirkungen auf die Erklärung der Vorhersage haben und somit das Vertrauen nicht beeinträchtigen oder zu Missverständnissen führen. Dies muss jedoch je nach Modell, Datenquelle und angewendetes XAI-Verfahren einzeln und kritisch begutachtet werden. Wie aus der Untersuchung hervorgeht, kann eine geringfügige Änderung, wie die Zunahme der Gesamtfläche in Quadratfuß, zu einem größeren Einfluss auf den Verkaufspreis führen als beispielsweise der Zustand des Kamins. Gerade die Flächenmaße sind wesentlich preisbestimmend und jegliche Änderung kann zu Ergebnissen führen, die innerhalb der Immobilienbranche stets kritisch begutachtet werden müssen.

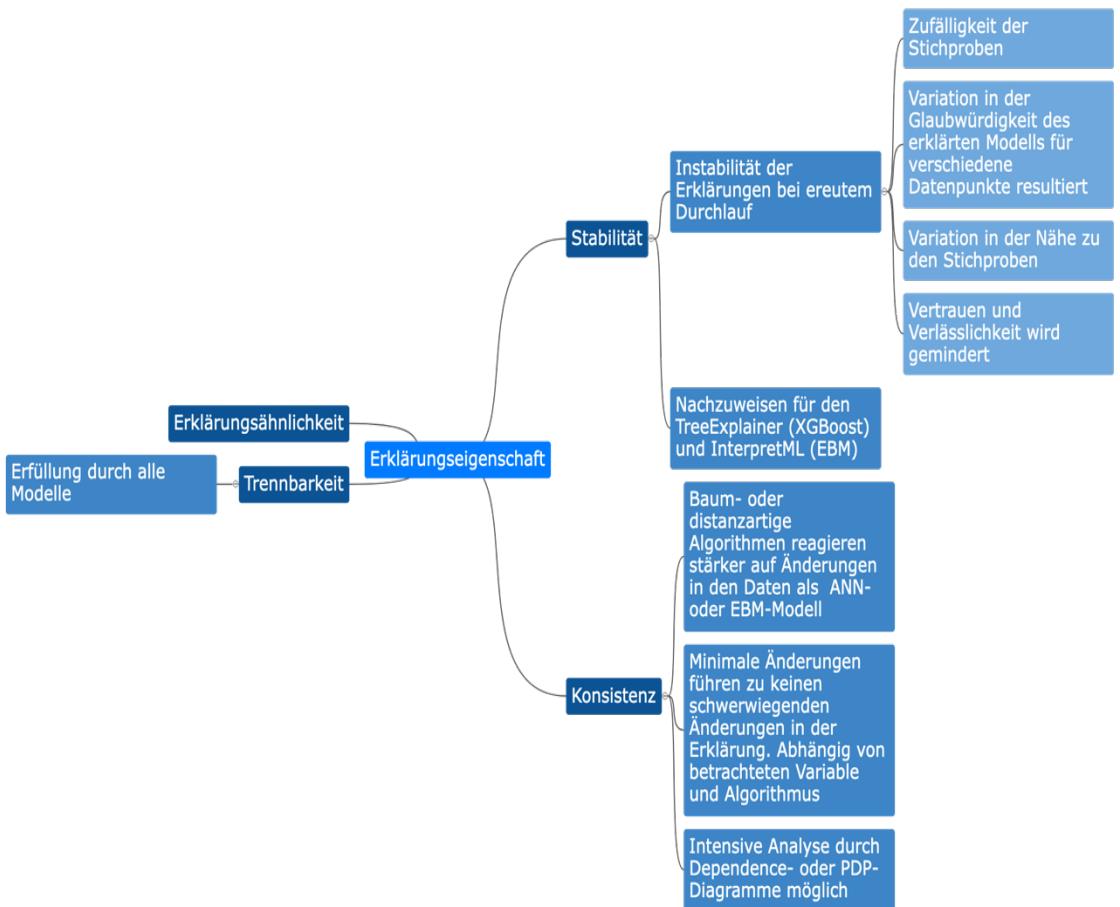


Abbildung 76: Zusammenfassung der Beurteilung – Stabilität, Transparenz, Trennbarkeit

- Erklärungähnlichkeit

Die Ergebnisse der Untersuchung zeigen, dass die Ergebnisse der Erklärungähnlichkeit dem Rashomon-Effekt unterliegen. Dies stellt eine gravierende und offensichtliche Herausforderung im Hinblick auf die Multiplizität von Vorhersagergebnissen und deren Erklärbarkeit dar. Gerade im Bezug zu systemkritischen Entscheidungen und Branchen ist eine transparente und nachvollziehbare Ergebnisfindung von ML-Modellen unabdingbar (Gerlings et al., 2021, S.1284; Sahakyan et al., 2021, S. 135392; Zolanvari et al., 2021, S.1-2). Wie bereits in den vorherigen Analysefeldern dargestellt, hilft XAI bei der Verständnisbildung für die Bedeutung der Variablen und stellt die ethische und rechtliche Richtigkeit auf eine transparente Art dar. In der ersten Instanz verhilft der Einsatz von XAI zu einer Verringerung des Rashomon-Effekts in Bezug auf die Multiplizität von guten Modellen. Wie in der Untersuchung aufgezeigt, geschieht dies durch eine transparente und nachvollziehbare Erklärung der Entscheidungen des Modells mittels der

beschriebenen Analysefelder. Jedoch zeigt gerade die lokale Analyse, dass die Erklärung der XAI-Methoden Widersprüche und Unterschiede aufweisen. Demnach lässt sich beobachten, dass ebenso eine Multiplizität von guten Erklärungen besteht (Leventi-Peetz & Weber, 2023, S.806). Dies schafft zusätzlich eine erhebliche Herausforderung in der praktischen Umsetzung, bezüglich der Akzeptanz, das Vertrauen und das Verständnis von Erklärung der ML-Prognosen. So zeigen die Untersuchungen, dass widersprüchliche Erklärungen je nach Modell und XAI-Verfahren auftreten können. Dies stellt für Entscheidungsträger und systemkritische Entscheidungen einen erheblichen Verlust an Verständlichkeit und Nachvollziehbarkeit dar.

Zusätzlich können Aggregationseffekte oder bestehende Multikollinearität von wesentlichem Einfluss sein, die zusätzlich nochmals die Erklärungsähnlichkeit gravierend beeinflussen. Gerade die Multikollinearität ist im Kontext von SHAP eine wesentliche Herausforderung, die zu missverständlichen Erklärungen der Modellergebnisse führen kann. Wie bereits im Analysefeld der Wichtigkeit von Variablen beschrieben, kann eine Korrelation zwischen niedrig gewichtete Variablen und stark gewichtete Variablen bestehen. Diese kann Probleme verursachen, indem es dem Modell erschwert wird die einzelnen Variablen und ihre spezifischen Beiträge zur Vorhersage zu unterscheiden. Insgesamt kann Multikollinearität die Erklärbarkeit von Vorhersagen beispielsweise mittels SHAP beeinträchtigen, indem das Verständnis der Beiträge jeder Variable zur Vorhersage verwischt wird. Daher ist es wichtig, diese Probleme zu berücksichtigen und gegebenenfalls Maßnahmen zur Reduzierung der Multikollinearität zu ergreifen, bevor mit SHAP gearbeitet werden kann. Ebenso erschwert die Aggregation von mehrerer Entscheidungsfaktoren, bestimmte Entscheidungen verstehen oder erklären zu können. Ein Problem, das sich hierbei ergeben kann, ist, dass die Aggregation von Faktoren einen Verlust an Feinheit und Detailgenauigkeit bei der Erklärung der Ergebnisfindung bedeuten kann. Es kann ebenfalls schwierig sein, eine Übersicht über die Gesamtheit der Entscheidungsfaktoren zu gewinnen. Daher ist es essenziell eine geeignete XAI-Methode zur Aggregation der Entscheidungsfaktoren zu wählen, die ein ausgewogenes Verhältnis zwischen Verständlichkeit und Detailgenauigkeit ermöglicht (Aas et al., 2021, S.22-23; Kenny et al., 2021, S.22). Hierbei wir die Möglichkeit eröffnet, die Hinterfragung des Einflusses der Multikollinearität und der Aggregationseffekt auf die Erklärbarkeit zu intensiveren. Beispielsweise kann die Untersuchung der Variable *Overall Qual* anhand der vorgestellten XAI-

Verfahren vertieft werden. Dabei kann der Frage nachgegangen werden: „Welche Variablen und Faktoren können möglicherweise die Gesamtqualität beeinflussen, wie beeinflusst dies die Erklärbarkeit und wie führt es schließlich zu den Ergebnissen des Modells?“. Da dies nicht das primäre Ziel der Ausarbeitung ist, wird die Diskussion der Erklärungsähnlichkeit fortgeführt.

Weiterführend kommt hinzu, dass das Nichtvorhandensein von einheitlich formulierten und aufgestellten Metriken, die die Güte der Erklärungen der XAI-Verfahren bewertet, zusätzlich die bereits bestehende Erklärungsähnlichkeit erschwert und somit die Leistung intransparent bleibt. Ebenso kann ein Entscheidungsträger nicht definieren welche Erklärung falsch oder korrekt ist. Vielmehr bilden die diversen Erklärungen bereits bestehende Erfahrungswerte ab und sind somit nahbarer als andere Erklärungen. Dies kann sich im Rahmen der Praxis als unkorrekte Entscheidungsgrundlag darstellen und somit in Fehlentscheidungen resultieren. Nichtsdestotrotz besteht weiterhin das Dilemma, dass die Entscheidung für den Einsatz eines Modells nicht anhand der Güte der Erklärung begründet werden kann. In Anbetracht der Stabilität der XAI-Verfahren kann hier zusätzlich nochmals ein Faktor der Unsicherheit auftreten und ebenso die Entscheidung für oder gegen Modelle erschweren. Dabei ist im Rahmen der Erklärungsähnlichkeit zu beachten, dass selbst wenn das Modell gut trainiert und validiert ist, noch immer Unsicherheiten bezüglich der Vorhersagen des Modells bestehen, die unterschiedlich und gegenteilig seitens der XAI-Verfahren interpretiert werden können. Ebenso kann eine mögliche Unsicherheit in den verwendeten Erklärungsmethoden bestehen und somit die Modellergebnisse unterschiedlich interpretiert werden (Saeed & Omlin, 2021, S.10).

Nichtsdestotrotz ist bei der Multiplizität von guten Modellen weiterhin der Ansatz geläufig, die Erklärungen verschiedener XAI-Verfahren zu vergleichen und bei keiner vorzuweisenden Erklärungsähnlichkeit sind kritische Abwägungen notwendig. Der Einsatz von Fachexperten oder die Auswahl des Modells anhand von ethischen, Compliance- oder Sparsamkeitskriterien kann zusätzlich bei der Auswahl eines geeigneten Modells unterstützen (Molnar et al., 2022, S.56). Hierbei wird erneut der Ansatz bezüglich des Einsatzes von primär Whitebox-Modellen seitens diverser Autoren forciert und unterstützt. Geraade im Anbetracht der benannten erheblichen Herausforderungen, die der Rashomon-Effekt mit sich bringt, kann ein transparentes Modell teilweise Abhilfe schaffen (Rudin, 2019, S.15; Semanova et al., 2022, S.14-15). Außerdem mindern widersprüchliche

Erklärungen das Verständnis zwischen Menschen und Maschine und zeigen auf wie dringend Lösungsansätze in Form von Metriken oder Gütekriterien benötigt werden. Der kontrafaktische Erklärungsansatz von Dandl et al. (2020) und Mothilal et al. (2020) könnte dabei eine mögliche Richtung vorgeben, in die sich ein Lösungsansatz entwickeln kann. Zusammenfassend lässt sich festhalten, dass obwohl XAI den Rashomon-Effekt bezüglich der Multiplizität von guten Modellen reduzieren kann, neue Herausforderungen wie die Multiplizität von guten Erklärungen und mögliche Inkonsistenzen zwischen den Erklärungen auftreten können. Generell lässt sich hinsichtlich der Erklärungsähnlichkeit die schwere des Rashomon-Effektes lediglich minimieren und nicht eliminieren (Leventi-Peetz & Weber, 2023, S.806).



Abbildung 77: Zusammenfassung der Beurteilung - Erklärungsähnlichkeit

Darüber hinaus sind weitere Punkte der Ergebnisse zu beachten, die losgelöst von den Analysefeldern und Erklärungseigenschaften in Erscheinung getreten sind und für die Zielsetzung der Ausarbeitung relevant sind. Dabei ist bei allen Modellen und Erklärungen seitens der XAI-Methoden zu beachten, dass diese auf Basis der vorhandenen Daten Erkenntnisse liefern und nicht in jedem Fall die gesamte Realität abbilden. Die Modelle werden anhand der Beobachtungen trainiert und validiert, was nicht simultan zur Schlussfolgerung eines kausalen Modells führt. Demnach bedeutet dies, dass eine Veränderung der Werte aufgrund der Erkenntnisse der Anwendung von XAI nicht zwangsläufig eine Änderung des tatsächlichen Verkaufspreises einer Immobilie bewirkt. Hierbei ist es wichtig, sorgfältig zu überprüfen, ob eine ausreichende Annahme der Kausalitätsbegründung

hinsichtlich der vorliegenden Daten, dem trainierten Modell und der verwendeten Interpretationstechnik vorliegt. Sollten alle Annahmen erfüllt sein, kann eine kausale Interpretation beispielsweise mittels des PDP-Diagramms zwischen abhängigen und einer unabhängigen Variable erfolgen. Diesbezüglich ist im Sinne der Verständnisförderung von enormer Bedeutung, die Möglichkeit der Erklärbarkeit von Blackbox-Modellen nicht mit einer eingehenden Kausalität zu bemessen (Molnar et al., 2022, S.63; Zhao & Hastie, 2021, S. 279). Dies inkludiert ebenso, dass die Erkenntnisse aus der Anwendung der XAI-Methoden allein nicht ausreichen, sondern durch die Anreicherung des Wissens von fachnahen Experten ergänzt werden muss. Dies soll im weiteren Sinne das Verständnis für die Ergebnisse fördern. Zusammenfassend kann durch die Kollaboration der menschlichen Expertise und der maschinellen Erkenntnis, eine Symbiose der Informationsgewinnung für das gesamtheitliche Verständnis ermöglicht werden (Weber et al., 2022, S.170). Es ist weiterhin wichtig zu berücksichtigen, wie sich die Praktikabilität der einzelnen Methoden hinsichtlich ihrer Visualisierbarkeit und benötigten Zeit auf das Verständnis auswirkt. Dabei ist gerade bei den modellagnostischen Interpretationsmethoden der zeitliche Aspekt eine Herausforderung. Je nach Algorithmus und verwendeter XAI-Methode unterscheidet sich die Berechnungszeit. Dabei zeigt die Untersuchung, dass bereits Fortschritte bei modellspezifischen Ansätzen vorzuweisen sind, wie es beim TreeExplainer der Fall ist (Lundberg et al., 2020, S.64).

Letztlich bleibt eine große Herausforderung im Gesamtkontext bestehen, welche die Etablierung eines universellen Konzepts der Erklärbarkeit der XAI-Methoden darstellt. Dabei hat sich im Rahmen der Literaturrecherche herausgestellt, dass derzeit seitens der Literatur und Forschung keine einheitliche und etablierte Methode zur Wertung der Erklärbarkeit von XAI-Systemen existiert. Bereits im Kapitel 2.6 wird darauf verwiesen, dass die Zusammenstellung der Analysefelder und Erklärungseigenschaften auf diversen Erkenntnissen und unterschiedlichen Definition von XAI-Metriken beruhen. Ein nicht existierender wissenschaftlicher Konsens bezüglich einer einheitlichen Methode mit standardisierten Metriken wird perspektivisch die Erfolge der Forschung von XAI negativ beeinflussen. Gerade der Vergleich auf Basis einer Messbarkeit muss gewährleistet werden, um die diversen XAI-Methoden hinsichtlich ihrer Eignung und Performanz für die Erklärbarkeit betrachten zu können. Dies ist von besonderer Bedeutung, um Herausforderungen wie dem Rashomon-Effekt entgegenzuwirken (Barredo Arrieta et al., 2020,

S.32; Weerts et al., 2019, S.6). Dabei kann die Definition von Metriken bereits bestehenden Beispielen wie der Definition von Konfidenzintervallen, die einen Vertrauensbereich für die Erklärbarkeit einräumen, folgen. Denkbar ist auch die Definition von einheitlichen Leistungsmetriken, die angeben, wie vertrauenswürdig die Erklärung unter Annahme des Rashomon-Effektes oder der Multikollinearität ist.

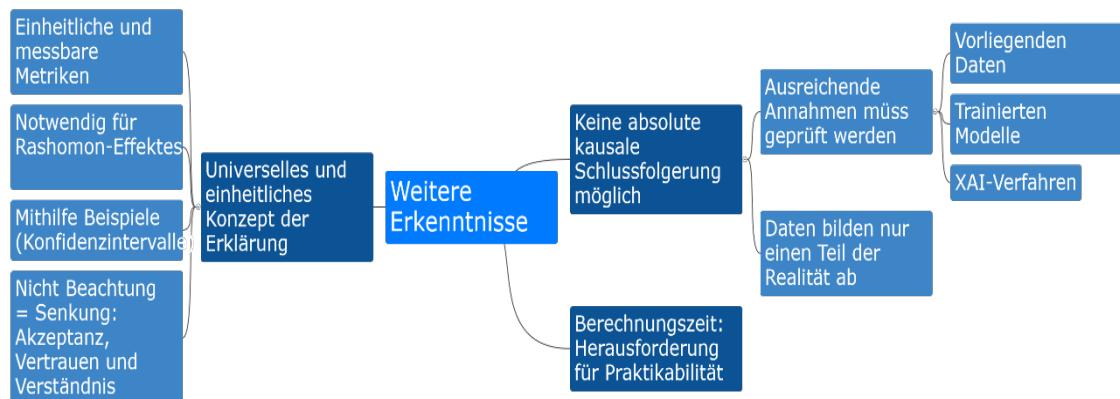


Abbildung 78: Zusammenfassung der Beurteilung – Weitere Erkenntnisse

Nichtdestotrotz lassen sich bereits einige vielversprechende Ansätze identifizieren, die kumuliert eine solide Grundlage bilden und sowohl qualitative als auch quantitative Metriken definieren. Schlussendlich ist sich die Forschung einig, dass keine einheitliche Methodik zur Einordnung von Erklärbarkeit zur Folge hat, dass die Akzeptanz, das Vertrauen und das Verständnis der Nutzer und Empfänger der XAI-Erkenntnisse sukzessiv abnehmen wird (Hoffman et al., 2018, S.4 ff; Hsiao et al., 2021, S.4 ff; Mohseni et al., 2021, S.248 ff; Rosenfeld, 2021, S.45ff; Sisk et al., 2022, S.2 ff.).

6.2 Implikationen für die Praxis

Die Erkenntnisse der Ausarbeitung können in vielfältiger Weise in der Praxis der Immobilienbranche eingesetzt werden.

- Bedeutung für die Wirtschaft

Immobilienunternehmen haben die Möglichkeit bewährte hedonische Ansatz, um die Prognosefähigkeit von ML zu erweitern. Zusätzlich kann ML zu Validierungszwecken genutzt werden. Dies erweitert das Repertoire an Mitteln zur Bewertung des Immobilienpreises, die eingesetzt werden können, wenn eine wirtschaftlich förderliche und verständliche Analyse von Daten und Informationen garantiert werden muss. Gerade die Erklärbarkeit mittels XAI eröffnet der Immobilienbranche eine neue Chance, die

Preisbewertung von Immobilien datengetrieben zu ermitteln, verstehen und begründen. Dabei kann das erlangte Wissen genutzt werden, um marktkonforme und wirtschaftlich sinnvolle Einkaufs- und Verkaufspreise von Immobilien zu erzielen. Ebenso ermöglicht die Anwendung von XAI die Identifizierung von wirtschaftskonformen Modellen, die die juristischen, ethischen und risikobedingten Anforderungen erfüllen und den wirtschaftlichen Erfolg verbessern können. Gleichzeitig kann durch die Interpretierbarkeit und Transparenz von XAI-Modellen die Reduzierung wirtschaftlicher Schäden erreicht werden, indem Missstände frühzeitig erkannt werden. Der Einsatz von XAI in der Immobilienbranche ermöglicht es, wirtschaftliche Erkenntnisse aus den Erklärungen der Modelle zu gewinnen, die mit klassischen Ansätzen oder Blackbox-Modellen verborgen bleiben würden. Dies kann den Prozess der Entscheidungsfindung bei wirtschaftskritischen Entscheidungen durch die Unterstützung und Erweiterung der Erklärbarkeit verbessern.

- Bedeutung Preisbestimmungen und Preisprognosen

Durch den Einsatz der XAI-Verfahren können die Ergebnisse der ML-Modelle für diverse Empfänger greifbarer, transparenter und verständlicher dargestellt werden. Gerade im Zuge der Preisbestimmung können bereits bestehende Praktiken und Erfahrungen mittels ML und XAI verglichen werden. Ebenso kann der Einsatz von bereits bestehenden ML-Modellen in Immobilienunternehmen anhand des Einsatzes von XAI validiert, optimiert und korrigiert werden. Gerade die Fehleranalyse bietet die Möglichkeit, die Faktoren zu identifizieren, die zu einem bestimmten Prognoseergebnis beitragen und eventuelle Variablen zu erkennen, die möglicherweise zu Verzerrungen führen können.

- Verständlichkeit für Stakeholder

Der praktische Einsatz von XAI in Immobilienunternehmen stellt Stakeholdern einen transparenten Zugang zur Entscheidungsfindung des Modells zur Verfügung. Dies ermöglicht im Unternehmen verschiedenen Adressaten, wie Projektplaner, Makler, Entscheidungsträger und Entwickler von ML-Modellen zusammenzubringen und Entscheidungen datenbasiert zu analysieren. Dabei können die globalen und lokalen Erkenntnisse fachübergreifend genutzt werden, um für bisher gängige oder undurchsichtige ML-Implementationen zu sensibilisieren. Die Visualisierung und Wertigkeit von Variablen schafft die Chance, Preisprognosen transparent nachzuvollziehen und gleichzeitig anhand der Wertigkeit die wesentlichen Treiber der Preisprognose zu erkennen. Dabei wird nicht nur eine Verständlichkeit für das Unternehmen selbst geschaffen, sondern ebenso eine

datenfundierte Argumentationsgrundlage für die Bestimmung der weiteren Strategieausrichtung des Immobilienunternehmens. Nichtsdestotrotz muss hervorgehoben werden, dass die Multiplizität an guten Erklärungen erhebliche Herausforderungen hervorruft. Gute Erklärungen diverser Modelle, die widersprüchliche Ergebnisse aufweisen, können signifikant das Vertrauen und die Akzeptanz der Stakeholder beeinträchtigen. Gleichzeitig wird die Verständlichkeit gemindert, da keine einheitlichen und etablierten Metriken existieren, die die Erklärbarkeit evaluieren können. Hierbei ist es von essenzieller Bedeutung für die Praxis die Möglichkeit der XAI-Methoden vollends auszuschöpfen, die diversen Ergebnisse zu vergleichen und entlang der Möglichkeiten Abwägungen vorzunehmen, die eine zufriedenstellende Entscheidung für oder gegen ein Modell ermöglichen.

6.3 Reflexion

- Verwendete Methoden

Mithilfe der Definition von Forschungsfragen nach dem methodischen Vorgehen von Kitchenham (2009) ist eine konsekutive Beantwortung der Zielstellung in einem stringentem Vorgehen möglich. Die systematische Literaturrecherche, die dem Vorbild von Brocke (2009) folgt, bildet eine fundierte Grundlage, um den Stand der aktuellen Forschung und Entwicklung anhand von qualitativ hochwertigen Journalbeiträgen und Konferenzbeiträgen widerzugeben und einzuordnen. Mithilfe der CRISP-DM Methode kann eine gradlinige Arbeitsweise von strukturierten Prozessmodellen transparent und verständlich aufgezeigt werden. Gerade der Einsatz von statistischen Methoden benötigt die Erfüllung von Voraussetzungen, die entlang der CRISP-DM sukzessiv erarbeitet werden. Der Einsatz von ML bildet die Grundlage für den Einsatz von XAI-Verfahren. Die Überprüfung der Generalisierbarkeit mittels Trainings- und Testdaten, die vollzogene Hyperparameteroptimierung und Messung der Leistung anhand der definierten Leistungsmetriken, bilden die notwendigen Schritte, um die Anwendung von XAI-Verfahren anhand von qualifizierten Modellen zu untersuchen. Die aus der systematischen Literaturrecherche entstandenen Analysefelder und Erklärungseigenschaften dienen als wissenschaftlich fundierte Beurteilungsgrundlage für die Untersuchung. Die daraus resultierende Diskussion ermöglichte anhand der Analysefelder und Erklärungseigenschaften die Einordnung der Ergebnisse im Sinne der Zielsetzung.

- Kernelemente der Untersuchung

Das Prototyping bildet in Kombination mit den Analysefeldern den Erklärungseigenschaften und den Einsatz von XAI-Verfahren das Kernelement der Untersuchung. Das Prototyping folgt der CRISP-DM Methode, die eine bewährte Vorgehensweise für ML-Projekte darstellt. Die 4 Modelle, die eine Abstufung im Sinne von White- zu Blackbox-Modellen hin realisieren, werden mittels der XAI-Methoden ausgiebig untersucht, um die Ergebnisse im Kontext der Zielsetzung, der Forschungsfragen und der Immobilienbranche zu ermöglichen. Die Verwendung des Ames Iowa Datensatzes verfügt über ausreichend Beobachtungen und Variablen, die den Einsatz und die Untersuchung von XAI-Verfahren ermöglichen. Gerade die Immobilienbranche ist aufgrund diverser einflussnehmender Faktoren ein hochkomplexes System, welches anhand des Datensatzes für einen Ausschnitt der Realität gut dargestellt und isoliert betrachtet werden kann. Gerade die Visualisierungsmöglichkeiten der XAI-Methoden ermöglichen es die Untersuchungen und Gegenüberstellungen in unterschiedlichen Umfängen zu gewährleisten und somit eine Reduktion der komplexen Ergebnisfindung der Modelle zu veranschaulichen.

6.4 Limitierung

Zur Gewährleistung eines gewünschten Tiefganges der Arbeit, ohne dabei die Güte zu schmälern, werden bewusst Limitierungen vorgenommen, die aufgrund des limitierenden Seitenumfangs unumgänglich sind. Zur Verbesserung der Leserlichkeit werden die bewusst vorgenommenen Limitierungen nachfolgend gruppiert erläutert:

- Entwicklung eines Prognosemodells

Die Beschreibung zur Entwicklung eines Prognosemodells in Kapitel 4 wird bewusst im Umfang limitiert. Die Generierung eines Prognosemodells nimmt allein durch die Data Understanding Phase einen umfassenden Teil des Prototypeings ein, da ein komplettes Verständnis für die Daten geschaffen werden muss. Die Limitierung wird mittels der Darstellung von Tabellen in Grenzen gehalten, die die wesentlichen Ergebnisse festhalten. Nichtsdestotrotz würde die gesamte und detaillierte Beschreibung aller getätigten Tätigkeiten den Umfang der Arbeit maßlos überschreiten. Ebenso wird auf den Einsatz von Methoden zur Dimensionsreduktion, wie der principal component analysis, verzichtet. Eine Untersuchung der Auswirkung der principal component analysis auf die

Erklärbarkeit ist von großem Interesse, wird jedoch aufgrund des verursachten Umfanges nicht mit in die Ausarbeitung miteinbezogen.

- Einsatz XAI-Verfahren

Innerhalb des Einsatzes der XAI-Verfahren werden Limitierungen im Rahmen der untersuchten Variablen vorgenommen. Dabei sind über alle Verfahren hinweg die 10 einflussreichsten Variablen sowohl lokal als auch global dargestellt. Die Darstellung und Beschreibung aller Variablen hätte zur Folge, dass die Untersuchung das vorgegebenen Seitenlimit weit überschreiten würde. Ebenso werden nicht alle graphischen Möglichkeiten der SHAP-Bibliothek verwendet. Die Einbindung von weiteren graphischen Möglichkeiten hätte auch keine weiteren Erkenntnisse erzielt, die nicht schon von den bereits verwendeten Darstellungen abgedeckt sind. Im Zuge von LIME wird der verwendete globale Ansatz nicht von der Bibliothek ermöglicht. Hierbei kann ein Lösungsansatz erfasst werden, der eine mögliche Limitierung aus dem Weg räumt. Letztlich fand eine Limitierung hinsichtlich der verwendeten Daten für die Nutzung der XAI-Methoden statt. Gerade für den Explainer für SHAP unter Anwendung des ANN- und EBM-Modells hätten größere Datenmengen zu einer unverhältnismäßigen Berechnungszeit geführt.

- Datensatz

Der Datensatz bildet einen Teil der Realität des Immobilienmarktes in Ames ab und betrachtet lediglich Eigenschaften der Immobilien, was eine isolierte Betrachtungsweise unter Ausschluss von makroökonomischen Faktoren ermöglicht. Die Entscheidung der Untersuchung des Datensatzes unterliegt der Limitierung, dass nicht alle Variablen, die einen Einfluss auf den Immobilienpreis ausüben können, hinzugezogen werden. Ebenso beschränkt sich die Betrachtung nur auf eine Stadt innerhalb eines Landes und umfasst nicht weitere Städte. Zunächst ist der Datensatz der *American Housing Survey* (2022) aus dem Jahr 2021 für die Ausarbeitung in Betracht gezogen worden. Es stellt sich jedoch heraus, dass die Daten für die gewaltige Größe einen großen Bereinigungsaufwand mit sich bringen. Dies hat zur Folge, dass eine Datenbereinigung einen wesentlich großen Teil der Ausarbeitung einnehmen würde und somit der zeitliche Rahmen nicht für die Untersuchung vorhanden wäre.

- Unbewusste Limitierung

Im Rahmen der Ausarbeitung sind ebenso Limitierungen aufgetreten auf deren Erscheinung kein Einfluss genommen werden kann. So ist derzeitig der Einsatz des InterpretML-

Moduls für das ANN-Modell nicht möglich und verhindert damit die Untersuchung des Modells. Wie bereits in der Ausarbeitung kommuniziert, sind die Entwickler diesbezüglich kontaktiert worden.

7 Schlussfolgerung

7.1 Zusammenfassung

Zusammenfassend lässt sich konstatieren, dass in Anbetracht der Zielsetzung und der unterstützenden Forschungsfragen, die Ausarbeitung diverse Möglichkeiten darlegt wie der Einsatz von XAI-Methoden die Akzeptanz und das Vertrauen in ML-Prognosen fördern kann. Dies impliziert ebenso die simultane Förderung des Verständnisses zwischen Menschen und Maschine. Durch die systematische Literaturrecherche wird eine solide wissenschaftliche Grundlage für die Untersuchung geschaffen, welche es ermöglicht, die aktuellen Erkenntnisse im Kontext der erklärenden künstlichen Intelligenz und der Immobilienbranche angemessen einzuordnen. Weiterführend werden die Ergebnisse diverser Autoren innerhalb von Analysefelder und Erklärungseigenschaften zusammengefasst, die die Basis und Struktur für die Untersuchung darstellen. Mittels des Prototypings werden nach der CRISP-DM Methode ML-Modelle entwickelten und getestet, die weiterführend das Anwenden der XAI-Verfahren ermöglichen. Die Untersuchung der Ergebnisse basiert auf der Anwendung von drei Verfahren zur erklärbaren künstlichen Intelligenz auf vier maschinellen Lernmodellen. Die vorliegende Ausarbeitung ist innerhalb der aktuellen Forschung eine Pionierarbeit, die erstmalig einen interdisziplinären Vergleich von vier ML-Modellen unter Anwendung von drei XAI-Methoden im Kontext der bestehenden Debatte zwischen Performance und Interpretierbarkeit darstellt. Diese Untersuchung umfasst eine Vielzahl von Algorithmen, die die gesamte Bandbreite von transparenten Whitenbox-Modellen bis hin zu undurchsichtigen Blackbox-Modellen abdecken. Hierbei werden sowohl Ähnlichkeiten als auch Unterschiede identifiziert und anhand der definierten Analysefelder und Erklärungseigenschaften diskutiert. Die Ergebnisse spiegeln dabei anhand der Analysefelder und Erklärungseigenschaften wider, wie der Einsatz von XAI die Erklärbarkeit der Ergebnisfindung von ML-Modelle interpretierbarer und nachvollziehbarer darstellen kann. Dabei ermöglichen die diversen Analysefelder verschiedenen Aspekte der Erklärbarkeit zu erfassen, die das Verständnis des Menschen hinsichtlich der Modeltransparenz, des Debuggings, der Variablengewichtung, ethischen Korrektheit, Fairness und Risikokontrolle fördern. Die Erklärungseigenschaften schaffen die Möglichkeit eines interdisziplinären Vergleiches zwischen XAI-Verfahren und ML-Modellen. Dabei lassen die Ergebnisse ebenso gravierende Herausforderungen erkennen. So kann

je nach Modell und Methode die Instabilität der Erklärungen den eigentlichen positiven Effekt des Einsatzes von XAI schmälern. Weiterführend zeigen die Ergebnisse, dass der Einsatz von XAI den Rashomon-Effekt hinsichtlich der Multiplizität von guten Modellen reduzieren kann. Gleichzeitig werden multiple gute Erklärungen erschaffen, die punktuelle Widersprüche in der Ergebnisfindung aufweisen und somit die Entscheidungsfindung für systemkritische Abwägungen maßgeblich behindern kann. Diesbezüglich kann der Zugewinn an Akzeptanz, Verständlichkeit und Transparenz, den Unternehmen durch den Einsatz von XAI-Methoden für ihre ML-Modelle erfahren können, erneut nichtig gemacht werden. Hierfür ist die Formulierung zielführender Lösungsansätzen dringend notwendig, um beispielsweise die Auswirkungen des Rashomon-Effektes reduzieren zu können. Zur Verbesserung des Einsatzes von XAI in der Praxis und Forschung müssen uniforme und wissenschaftlich fundierte Metriken erstellt werden. Diese sollen ermöglichen, die Erklärbarkeit sowohl qualitativ als auch quantitativ messbar zu machen und dadurch die Beurteilung der Erklärbarkeit zu gewährleisten. Die Stagnation des Zustandes wird im Umkehrschluss weiterhin das Verständnis zwischen Menschen und Maschine beeinträchtigen und somit in gravierende Folgen für die praktische Implementierung resultieren.

7.2 Ausblick und Forschungsbedarf

Mit der vorliegenden Ausarbeitung werden vier Forschungsfragen im Kontext der Zielsetzung beantwortet, mit dem Resultat, dass der bestehende Forschungsstand um weitere Erkenntnisse der Ausarbeitung ergänzt werden kann. Außerdem werden Ergebnisse aufzeigt, die weiteren Forschungsbedarf implizieren, welche im Folgenden genauer erläutert werden:

- Einheitliche Metriken

Wie bereits ausgiebig und mehrfach hervorgehoben, ist der Bedarf an einheitlichen und fundierten Metriken von großer Relevanz. Die Entwicklung von Metriken und Methoden zur Überprüfung und Validierung von XAI-Verfahren ist nicht nur für die Verständlichkeit von Relevanz, sondern wird ebenfalls aus ethischer und rechtlicher Sicht als signifikant gedeutet.

- Umgang mit kategorialen Variablen

Die Erklärbarkeit hinsichtlich der Bedeutung von kategorialen Variablen stellt je nach XAI-Verfahren eine besondere Herausforderung dar. Diesbezüglich ist es von Relevanz, die Erklärbarkeit von kategorialen Variablen hinsichtlich der Gewichtung und Transparenz perspektivisch genauer zu untersuchen.

- Inkludierung weiterer Variablen und Untersuchungen

Die Preisprognosen der Immobilien unterliegen einer Auswahl an einflussnehmenden Variablen. Durch die Integration von externen Variablen wie Zinssätzen für die Fremdkapitalaufnahme oder geografischen Lage kann die Verwendung von XAI eine genauere Analyse der Einflussfaktoren ermöglichen. Dabei wird nicht nur um die Hinzunahme weiterer Variablen von Bedeutung sein, sondern impliziert den Bedarf an Untersuchung von weiteren Effekten wie der Multikollinearität oder der Aggregation von Entscheidungsfaktoren.

- Menschenzentrierte Ansätze

Letztendlich ist es ebenso notwendig, menschlich-zentrierte Ansätze für die Interaktion mit XAI-Systemen zu erforschen, indem beispielsweise weitere Befragungen erhoben werden, wie es durch Sisk et al. (2022) bereits getätigter wird. Die Interaktion zwischen Menschen und ML-Modellen wird immer wichtiger, da sie immer häufiger in praktischen Anwendungen eingesetzt werden. Beispielsweise liegt der derzeitige Fokus auf ChatGPT, der einer breiten Nutzerschaft im privaten und beruflichen Umfeld die Möglichkeit bietet, mit KI erstmalig zu interagieren. Besonders im Hinblick auf ethisch fragwürdige oder fehlerhafte Informationen ist der Einsatz von XAI notwendig, um die Transparenz der Ergebnisfindung sicherzustellen und die Antworten für den Menschen verständlicher zu machen. Eine weitere Erforschung dieser Interaktion kann die Akzeptanz von XAI und ML verbessern, indem menschliche Perspektiven einbezogen werden, was schlussendlich zu einer besseren Verständlichkeit auf beiden Seiten führt.

7.3 Fazit

Die Implementierung von ML hatte lange Zeit eine eher untergeordnete Bedeutung in vielen Unternehmen. Gerade die fehlende Transparenz, Erklärbarkeit und Verständlichkeit der Ergebnisfindung von ML führt dazu, dass Immobilienunternehmen weiterhin die bewährten hedonischen Ansätze forcieren und Entscheidungen von subjektiven

Erfahrungswerten abhängig machen. Durch den mittlerweile leistungsfähigen und praxiserprobten Einsatz von ML in der Immobilienbranche hat die Erklärbarkeit der Ergebnisfindung eine stetig wachsende Nachfrage ausgelöst. Der Einsatz von XAI-Verfahren bietet die Möglichkeit, neue Erkenntnisse und neues Wissen innerhalb der Vorhersagemodellierung von Immobilienpreisen zu erschließen. Simultan kann der praxisorientierte Einsatz von ML entlang der XAI-Verfahren transparent, nachvollziehbar und verständlich dargestellt werden. Wie bei jedem technischen Fortschritt bedarf es auch im XAI-Bereich einer wohlüberlegten Nutzung. Dabei ist zunächst grundlegend für Immobilienunternehmen zu klären, wie die Ergebnisse der XAI-Verfahren den aktuellen oder perspektivischen Einsatz von ML-Prognosen unterstützen sollen. Gerade die aufgeführten Herausforderungen erwecken den Anschein, dass der Wettbewerbsvorteil und die Akzeptanz des Menschen vermindert werden kann. Dennoch ist die Nutzung von XAI für fast alle Unternehmen ein Zugewinn und eine Möglichkeit, die transparente Ergebnisfindung der ML-Modelle für kritische Unternehmensentscheidungen miteinzubeziehen. Erklärbares und vorhersagendes Arbeiten mit Hilfe von XAI- und ML-Techniken kann Geschäftsentscheidungen maßgeblich beeinflussen und Immobilienunternehmen am Markt erfolgreich machen. Eine interpretierbare und transparente Ergebnisfindung der ML-Modelle ermöglicht faktenbasierte Erkenntnisse, die auf das Unternehmen abgestimmt sind. Konsolidiert stützt dies wissensfundiert Entscheidungen zu treffen, die im heutigen Wettbewerb unabdingbar sind.

Literaturverzeichnis

- Aas, K., Jullum, M., & Løland, A. (2021). Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, 298, 1–28. <https://doi.org/10.1016/j.artint.2021.103502>
- Adadi, A., & Berrada, M. (2018). Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6, 52138–52160. <https://doi.org/10.1109/ACCESS.2018.2870052>
- Agatonovic-Kustrin, S., & Beresford, R. (2000). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *Journal of Pharmaceutical and Biomedical Analysis*, 22(5), 717–727. [https://doi.org/10.1016/S0731-7085\(99\)00272-1](https://doi.org/10.1016/S0731-7085(99)00272-1)
- Alfaro-Navarro, J. L., Cano, E. L., Alfaro-Cortes, E., Garcia, N., Gamez, M., & Larraz, B. (2020). A Fully Automated Adjustment of Ensemble Methods in Machine Learning for Modeling Complex Real Estate Systems. *Complexity*, 2020. <https://doi.org/10.1155/2020/5287263>
- Alvarez-Melis, D., & Jaakkola, T. S. (2018). *On the Robustness of Interpretability Methods*. Whi. <http://arxiv.org/abs/1806.08049>
- Alzain, E., Alshebami, A. S., & Aldhyani, T. H. H. (2022). Application of Artificial Intelligence for Predicting Real Estate Prices : The Case of Saudi Arabia. *Electronics (Switzerland)*, 11(1), 3448.
- Ananny, M., & Crawford, K. (2018). Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. *New Media and Society*, 20(3), 973–989. <https://doi.org/10.1177/1461444816676645>
- Andrychowicz, M., Denil, M., Colmenarejo, S. G., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., & De Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. *Advances in Neural Information Processing Systems*, 3988–3996.
- Arsov, N., Pavlovski, M., & Kocarev, L. (2019). Stability of decision trees and logistic regression. *Lecture Notes in Computer Science*, 1–13. <http://arxiv.org/abs/1903.00816>
- Baars, H., & Kemper, H.-G. (2021). Business Intelligence & Analytics – Grundlagen und praktische Anwendungen. In *Business Intelligence & Analytics – Grundlagen und*

- praktische Anwendungen.* <https://doi.org/10.1007/978-3-8348-2344-1>
- Baldauf, M., Garlappi, L., & Yannelis, C. (2020). Does climate change affect real estate prices? only if you believe in it. In *Review of Financial Studies* (Vol. 33, Issue 3). <https://doi.org/10.1093/rfs/hhz073>
- Banerjee, D., & Dutta, S. (2018). Predicting the housing price direction using machine learning techniques. *IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, ICPCSI 2017*, 2998–3000. <https://doi.org/10.1109/ICPCSI.2017.8392275>
- Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58, 82–115. <https://doi.org/10.1016/j.inffus.2019.12.012>
- Beckman, R. J., & Weisberg, S. (1987). Applied Linear Regression. In *Technometrics* (Vol. 29, Issue 1). <https://doi.org/10.2307/1269895>
- Belke, A., & Keil, J. (2018). Fundamental Determinants of Real Estate Prices: A Panel Study of German Regions. In *International Advances in Economic Research* (Vol. 24, Issue 1). <https://doi.org/10.1007/s11294-018-9671-2>
- Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. *Nature*, 549(7671), 195–202. <https://doi.org/10.1038/nature23474>
- Bilogur, A. (2022). *Missingno Documentation*. Github. <https://github.com/ResidentMario/missingno>
- Bin, J., Tang, S., Liu, Y., Wang, G., Gardiner, B., Liu, Z., & Li, E. (2017). Regression model for appraisal of real estate using recurrent neural network and boosting tree. *2017 2nd IEEE International Conference on Computational Intelligence and Applications, ICCIA 2017*, 2017-Janua, 209–213. <https://doi.org/10.1109/CIAPP.2017.8167209>
- Bird, S., Sameki, M., & Walker, K. (2020). *InterpretML : A toolkit for understanding machine learning models* *. <https://www.microsoft.com/en-us/research/uploads/prod/2020/05/InterpretML-Whitepaper.pdf>
- Branco, P., Torgo, L., & Ribeiro, R. P. (2016). A survey of predictive modeling on

- imbalanced domains. *ACM Computing Surveys*, 49(2), 1–50. <https://doi.org/10.1145/2907070>
- Breiman, L. (2001). Statistical modeling: The two cultures. *Statistical Science*, 16(3), 199–215. <https://doi.org/10.1214/ss/1009213726>
- Brzezicka, J. (2021). Towards a Typology of Housing Price Bubbles: A Literature Review. *Housing, Theory and Society*, 38(3), 320–342. <https://doi.org/10.1080/14036096.2020.1758204>
- Budholiya, K., Shrivastava, S. K., & Sharma, V. (2022). An optimized XGBoost based diagnostic system for effective prediction of heart disease. *Journal of King Saud University - Computer and Information Sciences*, 34(7), 4514–4523. <https://doi.org/10.1016/j.jksuci.2020.10.013>
- Cajias, M. (2021). Artificial intelligence and real estate - not just an evolution, a real game changer! *Journal of Property Investment and Finance*, 39(1), 15–18. <https://doi.org/10.1108/JPIF-06-2020-0063>
- Can, A. (1992). Specification and estimation of hedonic housing price models. *Regional Science and Urban Economics*, 22(3), 453–474. [https://doi.org/10.1016/0166-0462\(92\)90039-4](https://doi.org/10.1016/0166-0462(92)90039-4)
- Capuano, N., Fenza, G., Loia, V., & Stanzione, C. (2022). Explainable Artificial Intelligence in CyberSecurity: A Survey. *IEEE Access*, 10(August), 93575–93600. <https://doi.org/10.1109/ACCESS.2022.3204171>
- Carvalho, D. V., Pereira, E. M., & Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics (Switzerland)*, 8(8), 1–34. <https://doi.org/10.3390/electronics8080832>
- Čeh, M., Kilibarda, M., Liseč, A., & Bajat, B. (2018). Estimating the Performance of Random Forest versus Multiple Regression for Predicting Prices of the Apartments. *ISPRS International Journal of Geo-Information*, 7(5). <https://doi.org/10.3390/ijgi7050168>
- Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3), 1247–1250. <https://doi.org/10.5194/gmd-7-1247-2014>
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM. *SPSS Inc*, 78, 1–78. <http://www.crisp-dm.org/CRISPWP->

0800.pdf

- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data*, 785–794. <https://doi.org/http://dx.doi.org/10.1145/2939672.2939785>
- Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, 7, 1–24. <https://doi.org/10.7717/PEERJ-CS.623>
- Chin, S., Kahn, M. E., & Moon, H. R. (2020). Estimating the Gains from New Rail Transit Investment: A Machine Learning Tree Approach. *Real Estate Economics*, 48(3), 886–914. <https://doi.org/10.1111/1540-6229.12249>
- Conway, D., Li, C. Q., Wolch, J., Kahle, C., & Jerrett, M. (2010). A spatial autocorrelation approach for examining the effects of urban greenspace on residential property values. *Journal of Real Estate Finance and Economics*, 41(2), 150–169. <https://doi.org/10.1007/s11146-008-9159-6>
- Cui, N., Gu, H., Shen, T., & Feng, C. (2018). The impact of micro-level influencing factors on home value: A housing price-rent comparison. *Sustainability (Switzerland)*, 10(12), 1–23. <https://doi.org/10.3390/su10124343>
- Dandl, S., Molnar, C., Binder, M., & Bischl, B. (2020). Multi-objective counterfactual explanations. *Lecture Notes in Computer Science*, 12269 LNCS(01), 448–469. https://doi.org/10.1007/978-3-030-58112-1_31
- Das, A., & Rad, P. (2020). Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey. *IEEE Access*, 1–24. <http://arxiv.org/abs/2006.11371>
- De Cock, D. (2011). Ames, Iowa: Alternative to the boston housing data as an end of semester regression project. *Journal of Statistics Education*, 19(3). <https://doi.org/10.1080/10691898.2011.11889627>
- de Myttenaere, A., Golden, B., Le Grand, B., & Rossi, F. (2016). Mean Absolute Percentage Error for regression models. *Neurocomputing*, 192, 38–48. <https://doi.org/10.1016/j.neucom.2015.12.114>
- Deloitte. (2022). Fueling the AI transformation: Four key actions powering widespread value from AI, right now. In *Deloitte Insights* (Vol. 5, Issue October).

- des Rosiers, F., Dubé, J., & Thériault, M. (2011). Do peer effects shape property values? *Journal of Property Investment and Finance*, 29(4), 510–528. <https://doi.org/10.1108/14635781111150376>
- Dimopoulos, T., & Bakas, N. (2019). Sensitivity analysis of machine learning models for the mass appraisal of real estate. Case study of residential units in Nicosia, Cyprus. *Remote Sensing*, 11(24). <https://doi.org/10.3390/rs11243047>
- Fernández-Avilés, G., Ming, R., & Montero, J. M. (2012). Geostatistical Air Pollution Indexes spatial hedonic models: The case of Madrid, Spain. *Journal of Real Estate Research*, 34(2), 243–274. <https://doi.org/10.1080/10835547.2012.12091330>
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232. <https://doi.org/10.1214/aos/1013203451>
- Fryer, D., Strumke, I., & Nguyen, H. (2021). Shapley values for feature selection: The good, the bad, and the axioms. *IEEE Access*, 9. <https://doi.org/10.1109/ACCESS.2021.3119110>
- Ge, Z., Song, Z., Ding, S. X., & Huang, B. (2017). Data Mining and Analytics in the Process Industry: The Role of Machine Learning. *IEEE Access*, 5, 20590–20616. <https://doi.org/10.1109/ACCESS.2017.2756872>
- Gerlach, J., Hoppe, P., Jagels, S., Licker, L., & Breitner, M. H. (2022). Decision support for efficient XAI services - A morphological analysis, business model archetypes, and a decision tree. *Electronic Markets*, 0123456789. <https://doi.org/10.1007/s12525-022-00603-6>
- Gerlings, J., Shollo, A., & Constantiou, I. (2021). Reviewing the need for explainable artificial intelligence (XAI). *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2020-Janua, 1284–1293. <https://doi.org/10.24251/hicss.2021.156>
- Geron, A. (2018). *Praxiseinstieg Machine Learning mit Scikit-Learn und TensorFlow: Konzepte, Tools und Techniken für intelligente Systeme*. O'Reilly.
- Ghorbani, A., & Zou, J. (2019). Data shapley: Equitable valuation of data for machine learning. *36th International Conference on Machine Learning, ICML 2019, 2019-June*, 4053–4065.
- Giglio, S., Maggiori, M., Rao, K., Stroebel, J., & Weber, A. (2021). Climate Change and Long-Run Discount Rates: Evidence from Real Estate. *Review of Financial Studies*,

- 34(8), 3527–3571. <https://doi.org/10.1093/rfs/hhab032>
- Goldstein, A., Kapelner, A., Bleich, J., & Pitkin, E. (2015). Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics*, 24(1), 44–65. <https://doi.org/10.1080/10618600.2014.907095>
- Goodwin, K. R., La Roche, C. R., & Waller, B. D. (2020). Restrictions versus amenities: the differential impact of home owners associations on property marketability. *Journal of Property Research*, 37(3), 238–253. <https://doi.org/10.1080/09599916.2020.1740765>
- Greenwood, J., Seshadri, A., & Vandenbroucke, G. (2005). The baby boom and baby bust. *American Economic Review*, 95(1), 183–207. <https://doi.org/10.1257/0002828053828680>
- Grimes, P. W., Rogers, K. E., & Bosshardt, W. D. (2021). Economic Education and Household Financial Outcomes during the Financial Crisis. *Journal of Risk and Financial Management*, 14(7), 316. <https://doi.org/10.3390/jrfm14070316>
- Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., & Yang, G. Z. (2019). XAI-Explainable artificial intelligence. *Science Robotics*, 4(37), 0–1. <https://doi.org/10.1126/scirobotics.aay7120>
- Gupta, R., Marfatia, H. A., Pierdzioch, C., & Salisu, A. A. (2022). Machine Learning Predictions of Housing Market Synchronization across US States: The Role of Uncertainty. *Journal of Real Estate Finance and Economics*, 64(4), 523–545. <https://doi.org/10.1007/s11146-020-09813-1>
- Haider, M., & Miller, E. J. (2000). Effects of transportation infrastructure and location on residential real estate values: Application of spatial autoregressive techniques. *Transportation Research Record*, 1722, 1–8. <https://doi.org/10.3141/1722-01>
- Hall, P., & Gill, N. (2018). *An Introduction to Machine Learning Interpretability*.
- Hall, P., Phan, W., & Ambati, S. S. (2017). *Ideas on interpreting machine learning*. O'Reilly.
- Harel, O. (2009). The Estimation of R-sq and Adjusted R-sq in Incomplete Data Sets Using. *Journal of Applied Statistics*, 36(10), 1109–1118.
- Hoffman, R. R., Mueller, S. T., Klein, G., & Litman, J. (2018). *Metrics for Explainable AI: Challenges and Prospects*. 1–50. <http://arxiv.org/abs/1812.04608>

- Hott, C., & Monnin, P. (2008). Fundamental real estate prices: An empirical estimation with international data. *Journal of Real Estate Finance and Economics*, 36(4), 427–450. <https://doi.org/10.1007/s11146-007-9097-8>
- Hsiao, J. H., Ngai, H. H. T., Qiu, L., Yang, Y., & Cao, C. C. (2021). *Roadmap of Designing Cognitive Metrics for Explainable Artificial Intelligence (XAI)*. <http://arxiv.org/abs/2108.01737>
- Hu, L. Y., Huang, M. W., Ke, S. W., & Tsai, C. F. (2016). The distance function effect on k-nearest neighbor classification for medical datasets. *SpringerPlus*, 5(1). <https://doi.org/10.1186/s40064-016-2941-7>
- Imandoust, S. B., & Bolandraftar, M. (2013). Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events : Theoretical Background. *Int. Journal of Engineering Research and Applications*, 3(5), 605–610.
- Iooss, B., & Lemaître, P. (2015). A review on global sensitivity analysis methods. *Operations Research/ Computer Science Interfaces Series*, 59(around 30), 101–122. https://doi.org/10.1007/978-1-4899-7547-8_5
- Jafari, A., & Akhavian, R. (2019). Driving forces for the US residential housing price: a predictive analysis. *Built Environment Project and Asset Management*, 9(4), 515–529. <https://doi.org/10.1108/BEPAM-07-2018-0100>
- JetBrains. (2022). *DataSpell -Die IDE für professionelle Data Scientists*. <https://www.jetbrains.com/de-de/dataspell/>
- John-Matthews, J. M. (2022). Some critical and ethical perspectives on the empirical turn of AI interpretability. *Technological Forecasting and Social Change*, 174. <https://doi.org/10.1016/j.techfore.2021.121209>
- Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*, 6(4), 1–21. <https://doi.org/10.1145/2382577.2382579>
- Kenny, E. M., Ford, C., Quinn, M., & Keane, M. T. (2021). Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in XAI user studies. *Artificial Intelligence*, 294, 103459. <https://doi.org/10.1016/j.artint.2021.103459>
- Keras. (2022). *Keras Documentation*. <Https://Keras.Io/>. <https://keras.io/>
- Kim, B., Khanna, R., & Koyejo, O. (2016). Examples are not enough, learn to criticize!

- Criticism for interpretability. *Advances in Neural Information Processing Systems, Nips*, 2288–2296.
- Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering - A systematic literature review. *Information and Software Technology*, 51(1), 7–15. <https://doi.org/10.1016/j.infsof.2008.09.009>
- Krogh, A., & Vedelsby, J. (1995). Neural Network Ensembles, Cross Validation, and Active Learning. *Advances in Neural Information Processing Systems* 7, 231–238.
- Kumar, I. E., Venkatasubramanian, S., Scheidegger, C., & Friedler, S. A. (2020). Problems with Shapley-value-based explanations as feature importance measures. *37th International Conference on Machine Learning, ICML 2020, PartF16814*, 5447–5456.
- Lauritsen, S. M., Kristensen, M., Olsen, M. V., Larsen, M. S., Lauritsen, K. M., Jørgensen, M. J., Lange, J., & Thiesson, B. (2020). Explainable artificial intelligence model to predict acute critical illness from electronic health records. *Nature Communications*, 11(1), 1–11. <https://doi.org/10.1038/s41467-020-17431-x>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Lee, Z.-J., He, H.-M., Chen, Y., Xiao, J.-Y., & Chen, X.-Q. (2021). Data Analysis on the Influencing Factors of the Real Estate Price. *Journal of Business & Economic Statistics*, 39(1), 52–66. <https://doi.org/10.37256/aie.222021966>
- Lemagnen, K. (2019). *Model Interpretability: LIME*. Github. https://github.com/klemag/PyconUS_2019-model-interpretability-tutorial/blob/master/02-interpretability_LIME-solution.ipynb
- Leventi-Peetz, A., & Weber, K. (2023). *Rashomon Effect and Consistency in Explainable Artificial Intelligence (XAI)* (Vol. 2).
- Li, A., Feng, M., Li, Y., & Liu, Z. (2016). Application of Outlier Mining in Insider Identification Based on Boxplot Method. *Procedia Computer Science*, 91(May 2010), 245–251. <https://doi.org/10.1016/j.procs.2016.07.069>
- Li, S., Jiang, Y., Ke, S., Nie, K., & Wu, C. (2021). Understanding the Effects of Influential Factors on Housing Prices by Combining Extreme Gradient Boosting and a Hedonic Price Model (XGBoost-HPM). *Land*, 10(5). <https://doi.org/10.3390/land10050533>

- Li, T. (2020). The Value of Access to Rail Transit in a Congested City: Evidence from Housing Prices in Beijing. *Real Estate Economics*, 48(2), 556–598. <https://doi.org/10.1111/1540-6229.12222>
- Lin, J. Y., Treichel, V. (2012). The Unexpected Global Financial Crisis Researching Its Root Cause. *Policy Research Working Paper, January*, 80.
- Liu, Y. (2022). Real Estate Development Strategy Based on Artificial Intelligence. *Scientific Programming*, 2022(1), 6.
- Lorenz, F., Willwersch, J., Cajias, M., & Fuerst, F. (2021). Interpretable Machine Learning for Real Estate Market Analysis. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3835931>
- Lorenz, F., Willwersch, J., Cajias, M., & Fuerst, F. (2022). Interpretable machine learning for real estate market analysis. *Real Estate Economics*, 1–31. <https://doi.org/10.1111/1540-6229.12397>
- Loyola-Gonzalez, O. (2019). Black-box vs. White-Box: Understanding their advantages and weaknesses from a practical point of view. *IEEE Access*, 7, 154096–154113. <https://doi.org/10.1109/ACCESS.2019.2949286>
- Lundberg, S. (2018a). *Documentation Beeswarm Plot*. SHAP Documentation. https://shap.readthedocs.io/en/latest/example_notebooks/api_examples/plots/beeswarm.html
- Lundberg, S. (2018b). *Documentation Waterfall Plot*. SHAP Documentation. https://shap.readthedocs.io/en/latest/example_notebooks/api_examples/plots/waterfall.html
- Lundberg, S. (2018c). *shap.KernelExplainer*. <https://lrbjball.readthedocs.io/en/latest/generated/shap.KernelExplainer.html>
- Lundberg, S. (2018d). *shap.TreeExplainer Documentation*. <https://lrbjball.readthedocs.io/en/latest/generated/shap.TreeExplainer.html>
- Lundberg, S. (2018e). *Welcome to the SHAP documentation — SHAP latest documentation*. <https://shap.readthedocs.io/en/latest/index.html>
- Lundberg, S., Eron, G., Chen, H., DeGrave, A., Prutkin, J., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From Local Explanations to Global Understanding with Explainable AI for Trees. In *Physiology & behavior* (Vol. 2, Issue 1). <https://doi.org/10.1038/s42256-019-0138-9>.From

- MacLennan, D. (1977). Some Thoughts on the Nature and Purpose of House Price Studies. *Urban Studies*, 14(1), 59–71. <https://doi.org/10.1080/00420987720080061>
- Maier, G. (2018). *Definition Intelligenz*. Gabler Wirtschaftslexikon. <https://wirtschaftslexikon.gabler.de/definition/intelligenz-37696>
- Malpezzi, S. (2017). Residential real estate in the U.S. Financial crisis, the great recession, and their aftermath. *Taiwan Economic Review*, 45(1), 5–56. <https://doi.org/10.6277/TER.2017.451.1>
- Mao, J. (1996). Why artificial neural networks - Mao - 1996.pdf. *Computer*, 29(3), 31–44.
- Marcilio, W. E., & Eler, D. M. (2020). From explanations to feature selection: Assessing SHAP values as feature selection mechanism. *Proceedings - 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2020*, 340–347. <https://doi.org/10.1109/SIBGRAPI51738.2020.00053>
- Milcheva, S. (2022). Volatility and the Cross-Section of Real Estate Equity Returns during Covid-19. *Journal of Real Estate Finance and Economics*, 65(2), 293–320. <https://doi.org/10.1007/s11146-021-09840-6>
- Miller, R. G., & Pinter, N. (2022). Flood risk and residential real-estate prices: Evidence from three US counties. *Journal of Flood Risk Management*, 15(2), 1–12. <https://doi.org/10.1111/jfr3.12774>
- Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267, 1–38. <https://doi.org/10.1016/j.artint.2018.07.007>
- Mittal, U., & Sharma, D. M. (2021). Artificial Intelligence and its Application in Different Areas of Indian Economy. *International Journal of Advanced Research in Science, Communication and Technology*, 160–163. <https://doi.org/10.48175/ijarsct-v2-i3-328>
- Mockenhaupt, A. (2021). Digitalisierung und Künstliche Intelligenz in der Produktion. In *Digitalisierung und Künstliche Intelligenz in der Produktion*. <https://doi.org/10.1007/978-3-658-32773-6>
- Mohseni, S., Zarei, N., & Ragan, E. D. (2021). A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems. *ACM Transactions on Interactive Intelligent Systems*, 11(3–4). <https://doi.org/10.1145/3387166>

- Molnar, C. (2019). Interpretable Machine Learning. A Guide for Making Black Box Models Explainable. *Book*, 247. <https://christophm.github.io/interpretable-ml-book>
- Molnar, C., König, G., Herbinger, J., Freiesleben, T., Dandl, S., Scholbeck, C. A., Casalicchio, G., Grosse-Wentrup, M., & Bischl, B. (2022). General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models. *Lecture Notes in Computer Science*, 13200 LNAI(Iml), 39–68. https://doi.org/10.1007/978-3-031-04083-2_4
- Montaño Moreno, J. J., Palmer Pol, A., Sesé Abad, A., & Cajal Blasco, B. (2013). Using the R-MAPE index as a resistant measure of forecast accuracy. *Psicothema*, 25(4), 500–506. <https://doi.org/10.7334/psicothema2013.23>
- Morris, M. D. (1991). Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2), 161–174. <https://doi.org/10.1080/00401706.1991.10484804>
- Mothilal, R. K., Sharma, A., & Tan, C. (2020). Explaining machine learning classifiers through diverse counterfactual explanations. *FAT* 2020 - Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 607–617. <https://doi.org/10.1145/3351095.3372850>
- Mukaka, M. M. (2012). Statistics Corner: A guide to appropriate use of Correlation coefficient in medical research. *Malawi Medical Journal*, 24(September), 69–71.
- Müller, R., & Lenz, J. (2013). *Business Intelligence*. <https://doi.org/https://doi.org/10.1007/978-3-642-35560-8>.
- Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, 7(DEC). <https://doi.org/10.3389/fnbot.2013.00021>
- National Association of Realtors. (2021). *PROFILE OF HOME BUYERS AND SELLERS*. <https://cdn.nar.realtor/sites/default/files/documents/2021-highlights-from-the-profile-of-home-buyers-and-sellers-11-11-2021.pdf>
- National Association of Realtors. (2022). *THE ECONOMIC IMPACT OF A TYPICAL HOME SALE*. <https://cdn.nar.realtor/sites/default/files/documents/2022-state-economic-impact-report-us-04-28-2022.pdf>
- Nguyen, M. L. T., & Bui, T. N. (2019). Stock market, real estate market, and economic growth: An ARDL approach. *Investment Management and Financial Innovations*, 16(4), 290–302. [https://doi.org/10.21511/imfi.16\(4\).2019.25](https://doi.org/10.21511/imfi.16(4).2019.25)

- Nguyen, V. H., Le, T. T., Truong, H. S., Le, M. V., Ngo, V. L., Nguyen, A. T., & Nguyen, H. Q. (2021). Applying Bayesian Optimization for Machine Learning Models in Predicting the Surface Roughness in Single-Point Diamond Turning Polycarbonate. *Mathematical Problems in Engineering*, 2021. <https://doi.org/10.1155/2021/6815802>
- Nori, H., Jenkins, S., Koch, P., & Caruana, R. (2019). *InterpretML: A Unified Framework for Machine Learning Interpretability*. 1–8. <http://arxiv.org/abs/1909.09223>
- Nowak, A. S., & Radzik, T. (1994). A solidarity value for n-person transferable utility games. *International Journal of Game Theory*, 23(1), 43–48. <https://doi.org/10.1007/BF01242845>
- O’Sullivan, C. (2022). *Squeezing More out of LIME with Python - How to create global aggregations of LIME weights. Towards Data Science*. <https://towardsdatascience.com/squeezing-more-out-of-lime-with-python-28f46f74ca8e>
- Oppy, G., & Dowe, D. (2021). *The Turing Test*. Stanford Encyclopedia of Philosophy. <https://plato.stanford.edu/entries/turing-test/>
- Owen, A. B., & Prieur, C. (2017). On shapley value for measuring importance of dependent inputs. *SIAM-ASA Journal on Uncertainty Quantification*, 5(1), 986–1002. <https://doi.org/10.1137/16M1097717>
- Páez, A. (2019). The Pragmatic Turn in Explainable Artificial Intelligence (XAI). *Minds and Machines*, 29(3), 441–459. <https://doi.org/10.1007/s11023-019-09502-w>
- Pai, P. F., & Wang, W. C. (2020). Using machine learning models and actual transaction data for predicting real estate prices. *Applied Sciences (Switzerland)*, 10(17), 1–11. <https://doi.org/10.3390/app10175832>
- Palanisamy, P., Rajendran, I., & Shanmugasundaram, S. (2008). Prediction of tool wear using regression and ANN models in end-milling operation. *International Journal of Advanced Manufacturing Technology*, 37(1–2), 29–41. <https://doi.org/10.1007/s00170-007-0948-5>
- Pérez-Rave, J. I., Correa-Morales, J. C., & González-Echavarría, F. (2019). A machine learning approach to big data regression analysis of real estate prices for inferential and predictive purposes. *Journal of Property Research*, 36(1), 59–96. <https://doi.org/10.1080/09599916.2019.1587489>

- Phan, T. D. (2019). Housing price prediction using machine learning algorithms: The case of Melbourne city, Australia. *Proceedings - International Conference on Machine Learning and Data Engineering, ICMLDE 2018*, 8–13. <https://doi.org/10.1109/iCMLDE.2018.00017>
- Pinter, G., Mosavi, A., & Felde, I. (2020). Artificial intelligence for modeling real estate price using call detail records and hybrid machine learning approach. *Entropy*, 22(12), 1–14. <https://doi.org/10.3390/e22121421>
- Potrawa, T., & Tetereva, A. (2022). How much is the view from the window worth? Machine learning-driven hedonic pricing model of the real estate market. *Journal of Business Research*, 144(January), 50–65. <https://doi.org/10.1016/j.jbusres.2022.01.027>
- Qi, J., Du, J., Siniscalchi, S. M., Ma, X., & Lee, C. H. (2020). On Mean Absolute Error for Deep Neural Network Based Vector-to-Vector Regression. *IEEE Signal Processing Letters*, 27, 1485–1489. <https://doi.org/10.1109/LSP.2020.3016837>
- Quang, T., Minh, N., Hy, D., & Bo, M. (2020). Housing Price Prediction via Improved Machine Learning Techniques. *Procedia Computer Science*, 174(2019), 433–442. <https://doi.org/10.1016/j.procs.2020.06.111>
- Raju, V. N. G., Lakshmi, K. P., Jain, V. M., Kalidindi, A., & Padma, V. (2020). Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification. *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020, Icssit*, 729–735. <https://doi.org/10.1109/ICSSIT48917.2020.9214160>
- Rampini, L., & Re Cecconi, F. (2021). Artificial intelligence algorithms to predict Italian real estate market prices. *Journal of Property Investment and Finance*, 40(6), 588–611. <https://doi.org/10.1108/JPIF-08-2021-0073>
- Ratner, B. (2009). The correlation coefficient: Its values range between 1/1, or do they. *Journal of Targeting, Measurement and Analysis for Marketing*, 17(2), 139–142. <https://doi.org/10.1057/jt.2009.5>
- Religia, Y., Pranoto, G. T., & Santosa, E. D. (2020). South German Credit Data Classification Using Random Forest Algorithm to Predict Bank Credit Receipts. *JISA(Jurnal Informatika Dan Sains)*, 3(2), 62–66. <https://doi.org/10.31326/jisa.v3i2.837>

- Ribeiro, M. T. (2016). *Local Interpretable Model-Agnostic Explanations (lime) — lime 0.1 documentation*. <https://lime-ml.readthedocs.io/en/latest/>
- Ribeiro, M. T. (2018). *Tutorial - continuous and categorical features*. Github. <https://marcotcr.github.io/lime/tutorials/Tutorial - continuous and categorical features.html>
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016a). “Why Should I Trust You?” Explaining the Predictions of Any Classifier. *NAACL-HLT 2016 - 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session*, 97–101. <https://doi.org/10.18653/v1/n16-3020>
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016b). Model-Agnostic Interpretability of Machine Learning. *ICML Workshop on Human Interpretability in Machine Learning*, 91–95. <http://arxiv.org/abs/1606.05386>
- Rosenfeld, A. (2021). Better metrics for evaluating explainable artificial intelligence. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS, 1*, 45–50.
- Rouwendal, J., Levkovich, O., & van Marwijk, R. (2017). Estimating the Value of Proximity to Water, When Ceteris Really Is Paribus. *Real Estate Economics*, 45(4), 829–860. <https://doi.org/10.1111/1540-6229.12143>
- Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), 206–215. <https://doi.org/10.1038/s42256-019-0048-x>
- Saeed, W., & Omelin, C. (2021). Explainable AI (XAI): A Systematic Meta-Survey of Current Challenges and Future Opportunities. *Knowledge-Based Systems*, 263, 110273. <https://doi.org/10.1016/j.knosys.2023.110273>
- Sahakyan, M., Aung, Z., & Rahwan, T. (2021). Explainable Artificial Intelligence for Tabular Data: A Survey. *IEEE Access*, 9, 135392–135422. <https://doi.org/10.1109/ACCESS.2021.3116481>
- Sandbhor, S. S., & Chaphalkar, N. B. (2016). State of art report on variables affecting housing value. *Indian Journal of Science and Technology*, 9(17). <https://doi.org/10.17485/ijst/2016/v9i17/82861>
- Satish, D. G. N., Raghavendran, D. C. V., Rao, M. M. D. S., & Srinivasulu, D. C. (2019).

- House Price Prediction using Machine Learning. *International Journal of Innovative Technology and Exploring Engineering*, 8(9), 717–722.
<https://doi.org/10.35940/ijitee.i7849.078919>
- Semenova, L., Rudin, C., & Parr, R. (2022). On the Existence of Simpler Machine Learning Models. *ACM International Conference Proceeding Series*, 1827–1858.
<https://doi.org/10.1145/3531146.3533232>
- Shafique, U., & Qaiser, H. (2014). A Comparative Study of Data Mining Process Models (KDD , CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, 12(1), 217–222. <http://www.ijisr.issr-journals.org/>
- Shapley, L. (1953). A value for n-person game. *Kuhn HW, Tucker AW (Eds) Contributions to the Theory of Games II*, 307–317.
- Sisk, M., Majlis, M., Page, C., & Yazdinejad, A. (2022). Analyzing XAI Metrics : Summary of the Literature Review Analyzing XAI Metrics : Summary of the Literature Review. *IEEE Access*, 0–6.
<https://doi.org/10.36227/techrxiv.21262041.v1>
- Sokol, K., & Flach, P. (2020). Explainability fact sheets: A framework for systematic assessment of explainable approaches. *FAT* 2020 - Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 56–67.
<https://doi.org/10.1145/3351095.3372870>
- Song, Y., Liang, J., Lu, J., & Zhao, X. (2017). An efficient instance selection algorithm for k nearest neighbor regression. *Neurocomputing*, 251, 26–34.
<https://doi.org/10.1016/j.neucom.2017.04.018>
- Sorkin, A., Giang, V., Gandel, S., Hirsch, L., Livni, E., Warner, B., & Gallagher, D. (2022). *The Fed Sees Housing Trouble Ahead - The New York Times*. New York Times. <https://www.nytimes.com/2022/08/18/business/fed-minutes-housing-interest-rates.html>
- Sovrano, F., Vitali, F., & Palmirani, M. (2021). Making Things Explainable vs Explaining: Requirements and Challenges Under the GDPR. *Lecture Notes in Computer Science*, 13048, 169–182. https://doi.org/10.1007/978-3-030-89811-3_12
- Stamou, M., Mimis, A., & Rovolis, A. (2017). House price determinants in Athens: a spatial econometric approach. *Journal of Property Research*, 34(4), 269–284.
<https://doi.org/10.1080/09599916.2017.1400575>

- Tensorflow. (2022). *Tensorflow Documentation*. [Https://Www.Tensorflow.Org/Learn](https://Www.Tensorflow.Org/Learn).
<https://www.tensorflow.org/>
- Theisen, T., & Emblem, A. W. (2018). House prices and proximity to kindergarten—costs of distance and external effects? *Journal of Property Research*, 35(4), 321–343.
<https://doi.org/10.1080/09599916.2018.1513057>
- Tibshirani, R., Hastie, T., Narasimhan, B., & Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, 99(10), 6567–6572.
<https://doi.org/10.1073/pnas.082099299>
- U.S. Federal Housing Finance Agency. (2023a). *All-Transactions House Price Index for Ames, IA (MSA)*. Fred.Stlouisfed.Org.
- U.S. Federal Housing Finance Agency. (2023b). *All-Transactions House Price Index for Iowa*. Fred.Stlouisfed.Org. <https://fred.stlouisfed.org/series/IASTHPI>
- U.S. Federal Housing Finance Agency. (2023c). *Median Sales Price of Houses Sold for the Midwest Census Region*. Fred.Stlouisfed.Org.
<https://fred.stlouisfed.org/series/MSPMW>
- U.S. Federal Housing Finance Agency. (2023d). *Median Sales Price of Houses Sold for the United States*. Fred.Stlouisfed.Org. <https://fred.stlouisfed.org/series/MSPUS>
- United States Census Bureau. (2022). *American Housing Survey 2021*.
<https://www.census.gov/programs-surveys/ahs/data.html>
- US Census Bureau. (2022). *Median household income in the United States from 1990 to 2021*.
- Vale, D., El-Sharif, A., & Ali, M. (2022). Explainable artificial intelligence (XAI) post-hoc explainability methods: risks and limitations in non-discrimination law. *AI and Ethics*, 2(4), 815–826. <https://doi.org/10.1007/s43681-022-00142-y>
- Viriato, J. C. (2019). AI and machine learning in real estate investment. *Journal of Portfolio Management*, 45(7), 43–54. <https://doi.org/10.3905/jpm.2019.45.7.043>
- Vom Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., & Cleven, A. (2009). Reconstructing the giant: On the importance of rigour in documenting the literature search process. *17th European Conference on Information Systems, ECIS 2009, January*.
- Weber, L., Lapuschkin, S., Binder, A., & Samek, W. (2022). Beyond Explaining:

- Opportunities and Challenges of XAI-Based Model Improvement. *Information Fusion*, 92(March 2022), 154–176. <https://doi.org/10.1016/j.inffus.2022.11.013>
- Webster, J., & Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly*, 26(2), xiii–xxiii. <https://doi.org/10.1.1.104.6570>
- Weerts, H. J. P., van Ipenburg, W., & Pechenizkiy, M. (2019). *A Human-Grounded Evaluation of SHAP for Alert Processing*. <http://arxiv.org/abs/1907.03324>
- Wei, C., Fu, M., Wang, L., Yang, H., Tang, F., & Xiong, Y. (2022). The Research Development of Hedonic Price Model-Based Real Estate Appraisal in the Era of Big Data. *Land*, 11(334), 1–30.
- Wentura, D., & Pospeschill, M. (2015). Multivariate Datenanalyse. In *Multivariate Datenanalyse*. <https://doi.org/10.1007/978-3-531-93435-8>
- Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30(1), 79–82. <https://doi.org/10.3354/cr030079>
- Winter, E. (2002). The Shapley value. *Handbook of Game Theory with Economic Applications*, 3(1), 2025–2054. <https://doi.org/10.1017/cbo9780511794216.019>
- Wyman, D., & Mothorpe, C. (2018). The Pricing of Power Lines: A Geospatial Approach to Measuring Residential Property Values. *Journal of Real Estate Research*, 40(1), 121–154. <https://doi.org/10.1080/10835547.2018.12091490>
- Yang, J. (2021). *Fast TreeSHAP: Accelerating SHAP Value Computation for Trees*. <http://arxiv.org/abs/2109.09847>
- Zaki, J., Nayyar, A., Dalal, S., & Ali, Z. H. (2022). House price prediction using hedonic pricing model and machine learning techniques. *Concurrency and Computation: Practice and Experience*, 34(27), 1–15. <https://doi.org/10.1002/cpe.7342>
- Zhang, G. (2022). Applications of Social Attribution Theory in XAI. *Mdpi Proceedings*, 81, 101. <https://doi.org/10.3390/proceedings2022081101>
- Zhang, P., Jia, Y., & Shang, Y. (2022). Research and application of XGBoost in imbalanced data. *International Journal of Distributed Sensor Networks*, 18(6). <https://doi.org/10.1177/15501329221106935>
- Zhang, Y., Song, K., Sun, Y., Tan, S., & Udell, M. (2019). “*Why Should You Trust My Explanation?*” *Understanding Uncertainty in LIME Explanations*.

<http://arxiv.org/abs/1904.12991>

- Zhang, Z., Hamadi, H. Al, Damiani, E., Yeun, C. Y., & Taher, F. (2022). Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research. *IEEE Access*, 10, 93104–93139. <https://doi.org/10.1109/ACCESS.2022.3204051>
- Zhao, Q., & Hastie, T. (2021). Causal Interpretations of Black-Box Models. *Journal of Business and Economic Statistics*, 39(1), 272–281. <https://doi.org/10.1080/07350015.2019.1624293>
- Zhou, X. (2020). The usage of artificial intelligence in the commodity house price evaluation model. *Journal of Ambient Intelligence and Humanized Computing*, 0123456789. <https://doi.org/10.1007/s12652-019-01616-4>
- ZIA & EY Real Estate. (2022). *Digitalisierungsstudie 2022 : Das digitale Quartier* (Issue September).
- Zolanvari, M., Yang, Z., Khan, K., Jain, R., & Meskin, N. (2021). TRUST XAI: Model-Agnostic Explanations for AI With a Case Study on IIoT Security. *IEEE Internet of Things Journal*, 4662(c), 1–13. <https://doi.org/10.1109/JIOT.2021.3122019>

Anhang

1. Codequellen

Für die Erstellung des Codes und zur Informationsgewinnung wurden neben den bereits im Text genannten Bibliotheken und Quellen auch folgende verwendet:

- Bruin, E. (2018). *House prices: Lasso, XGBoost, and a detailed EDA*. Kaggle. <https://www.kaggle.com/code/erikbruin/house-prices-lasso-xgboost-and-a-detailed-eda>
- DanB. (2018). *Using Categorical Data with One Hot Encoding*. Kaggle. <https://www.kaggle.com/code/dansbecker/using-categorical-data-with-one-hot-encoding>
- Developers, X. (2022). *XGBoost Documentation*. Xgboost.Readthedocs.Io/En/Stable/. <https://xgboost.readthedocs.io/en/stable/>
- Hashmi, F. (2018a). *Boston-housing-price-prediction-case-study-in-python*. Thinking Neuron. <https://thinkingneuron.com/boston-housing-price-prediction-case-study-in-python/>
- Hashmi, F. (2018b). *Diamond-price-prediction-case-study-in-python*. Thinking Neuron. <https://thinkingneuron.com/diamond-price-prediction-case-study-in-python/>
- Hashmi, F. (2019). *How to create an XGboost model for regression in python*. Thinking Neuron. <https://thinkingneuron.com/how-to-create-an-xgboost-model-for-regression-in-python/>
- Helpers. (2022). *Helpers Documentation*. Readthedocs.Org/. <https://readthedocs.org/projects/helper/downloads/pdf/latest/>
- Huneycutt, J. (2018). *Quantitative Data Exploration for Regression in Python (Ames Housing Dataset)*. Medium.Com. <https://medium.com/@hjhuney/quantitative-data-exploration-for-regression-in-python-ames-housing-part-1-25879dd4cc4a>
- Joblib. (2022). *joblib Documentation*. [Https://Joblib.Readthedocs.Io/En/Latest/](https://Joblib.Readthedocs.Io/En/Latest/). <https://joblib.readthedocs.io/en/latest/>
- Matplotlib. (2022). *Matplotlib Documentation*. Matplotlib.Org/. <https://matplotlib.org/>
- Miller, S. (2020). *RegressionHousingPrices*. Github. <https://github.com/sjmiller8182/RegressionHousingPrices>
- Numba. (2022). *No Title*. Numba.Readthedocs.Io/En/Stable/. <https://numba.readthedocs.io/en/stable/>
- Numpy. (2022). *Numpy Documentation*. Numpy.Org/Doc/. <https://numpy.org/doc/>
- Pandas. (2022). *Pandas Documentation*. Pandas.Pydata.Org. <https://pandas.pydata.org/docs/>
- Plotly. (2022). *Documentation Plotly*. Plotly.Com. <https://plotly.com/python/>
- Python. (2022). *Functools Documentation*. <Https://Docs.Python.Org/3/Library/Functools.Html>. <https://docs.python.org/3/library/functools.html>
- Scikit-learn. (2022a). *Scikit-learn Documentation*. Scikit-Learn.Org. <https://scikit-learn.org/stable/>
- Scikit-learn. (2022b). *The Ames housing dataset*. Inria.Githunb.Io. <https://Inria.Githunb.Io>

https://inria.github.io/scikit-learn-mooc/python_scripts/datasets Ames_housing.html
 Scipy.stats. (2022). *scipy.stats Documentation*. Docs.Scipy.Org. <https://docs.scipy.org>
 Seaborn. (2022). *Seaborn Documentation*. [Https://Seaborn.Pydata.Org/](https://Seaborn.Pydata.Org/).
<https://seaborn.pydata.org/>

2.Code

```
#%% md
# Data Analysis

#%%
## Daten Einlesen und erster Eindruck
#%%
import matplotlib.pyplot as plt
import pandas as pd
import csv

Daten = 'AmesHousing(1).csv'

Datensatz = pd.read_csv(Daten,delimiter=',')
Datensatz.head(100)

#%%
#Entfernung der Variable "Order", die die Rangfolge angibt
Datensatz = Datensatz.drop(['Order'],axis=1)
Datensatz.head(100)
#%%
# Ersten Informationen zum Datensatz
Datensatz.info()
#%% md
## Fehlende Werte Ermitteln
#%%
Datensatz.info(verbose=True,show_counts=None)
#%%
#Anzahl der nicht-Null Einträge
Datensatz.isnull().sum(axis=0)
#%%
pip install missingno
#%%
#Nutzung der missingno Bibliothek zur Identifikation von fehlenden
#Einträgen
import missingno as msno
msno.matrix(Datensatz,sparkline=False, figsize=(50,35), fontsize=10,
color=(0.27, 0.52, 1.0))
#%%
# Anzeige nur des Typen "Object"
msno.matrix(Datensatz.select_dtypes(include='object'),color=(0.27,
0.52, 1.0))
#%%
#Anzeige der numerischen Werte
msno.matrix(Datensatz.select_dtypes(include=('int64','float64')),co-
lor=(0.27, 0.52, 1.0))
#%%
```

```

#Deskriptive Statistik der kategorialen und metrischen Variablen
Datensatz.describe(include='all')
#%%
#Deskriptive Statistik nur der numerischen Werte
Datensatz.describe()
#%%
#Visuelle Darstellung der Zielvariabel
Datensatz['SalePrice'].hist()
#%%
#Anzeige Zielvariable mittel Seaborn
import seaborn as sns
sns.set(rc = {"figure.figsize" : (11, 7)})
sns.distplot(Datensatz["SalePrice"], color = "blue");
#%%
#Anzeige Zielvariabel mittels Boxplot
sns.set(rc = {"figure.figsize" : (11, 7)})
sns.boxplot(Datensatz["SalePrice"], color = "blue");
#%%
#Ermittlung des Kurtosis Wertes
print("Kurtosis: ", Datensatz["SalePrice"].kurt())
#Wert höher als 3 idiziert Ausreißer
#%%
#Anzeige der numerischen Variablen des Datentypen "int64"
%matplotlib inline
import matplotlib.pyplot as py

Datensatz.select_dtypes(include='int64').hist(figsize=(18,20),)
#Keine ausgeglichen Verteilung: Lot Area Enclosed Porch, Screen
Porch, 3 Ssn Porch Misc Val, Kitchen AbvGr, Low Qual Fin SF
#%%
md

#%%
## Analyse der metrischen Variablen
#%%
#Speicherung der Variablen in Listen
metrischdiskret=['MS SubClass', 'Overall Qual', 'Overall Cond']

metrischstetig=['Lot Frontage', 'Mas Vnr Area', 'BsmtFin SF 1', 'BsmtFin
SF 2', 'Bsmt Unf SF', 'Total Bsmt SF', 'Bsmt Full Bath', 'Bsmt Half
Bath', 'Garage Yr Blt', 'Garage Cars', 'Garage Area', 'PID', 'Lot Ar-
ea', 'Year Built', 'Year Remod/Add', '1st Flr SF', '2nd Flr SF', 'Low Qual
Fin SF', 'Gr Liv Area', 'Full Bath', 'Half Bath', 'Bedroom AbvGr', 'Kitchen
AbvGr', 'TotRms AbvGrd', 'Fireplaces', 'Wood Deck SF', 'Open Porch
SF', 'Enclosed Porch', '3Ssn Porch', 'Screen Porch', 'Pool Area', 'Misc
Val', 'Mo Sold', 'Yr Sold',]

kategorial=[list(Datensatz.select_dtypes(include='object'))]
#%%
list(Datensatz.select_dtypes(include='int64'))
#%%
# Box Plot Darstellung der metrisch stetigen unabhängigen Variablen
ContinuousColsList=['Year Built', 'Gr Liv Area', 'Overall Qual', 'Lot
Frontage', 'Mas Vnr Area', 'BsmtFin SF 1', 'BsmtFin SF 2', 'Bsmt Unf SF'
, 'Total Bsmt SF', 'Bsmt Full Bath', 'Bsmt Half
Bath', 'Garage Yr Blt', 'Garage Area', 'Lot Area']

import matplotlib.pyplot as plt

```

```

fig, PlotCanvas=plt.subplots(nrows=1, ncols=len(ContinuousColsList),
figsize=(18,5))

for PredictorCol , i in zip(ContinuousColsList, range(len(ContinuousColsList))):
    Datensatz.boxplot(column=PredictorCol, figsize=(5,5), vert=True,
ax=PlotCanvas[i])

#%%
# Box Plot Darstellung der metrisch stetigen unabhängigen Variablen
ContinuousColsList=[ 'Gr Liv Area','Overall Qual']

import matplotlib.pyplot as plt
fig, PlotCanvas=plt.subplots(nrows=1, ncols=len(ContinuousColsList),
figsize=(18,5))

for PredictorCol , i in zip(ContinuousColsList, range(len(ContinuousColsList))):
    Datensatz.boxplot(column=PredictorCol, figsize=(5,5), vert=True,
ax=PlotCanvas[i])

#%%
#Mehrfach Darstellung der metrischen Variablen als Balkendiagramme /
Histogramme
%matplotlib inline
import matplotlib.pyplot as py

Datensatz.select_dtypes(include=['int64','float64']).hist(figsize=(14,17),
#Keine ausgeglichenen Verteilung BsmtFin SF 2, BsmtFin Half Bath
#%%
md
## Analyse der kategorialen Variablen
#%%
import seaborn as sns

#Definition der Darstellung
def data_explorer(df):
    # Speicherung der kategorialen Variablen in einer Liste
    dobj = Datensatz.select_dtypes(include=['object'])

    # Definition der Diagramme inklusive Anzahl der Diagramme pro
    Reihe innerhalb der Liste "dobj"
    fig, axes = plt.subplots(11, 4, figsize=(9, 20))
    for col, ax in zip(dobj.columns, axes.flat):
        sns.countplot(data=dobj[[col]], x=col, ax=ax)
    fig.savefig(f'data_exploration_{"\_".join(dobj.columns)}.pdf')

data_explorer(Datensatz)

#%%
#Genauere Analyse der Variable "Neighborhood" aufgrund der komprimierten
Darstellung der Merkmalsausprägungen
plt.figure(figsize=(25,15))

```

```

sns.countplot(x=Datensatz['Neighborhood'].values, data=Daten-
satz['Neighborhood'])
#Genaueer Blick auf Neighborhood
#%%
#Ausgabe des Verkaufstypen je nach Merkmalszählung
plt.figure(figsize=(25,15))
sns.countplot(x=Datensatz['Sale Type'].values, data=Datensatz['Sale
Type'])
#%%
## Graphische Korrelationsanalyse
#%%
### Kategoriale Variablen
#%%
#Ausgabe List der kategorialen Variablen
list(Datensatz.select_dtypes(include='object'))
#%%
#Speicherung der Variablen in Listen zur besseren Darstellung in den
Diagrammen

metrischdiskret=['MS SubClass','Overall Qual', 'Overall Cond']

metrisch=['Year Built', 'Gr Liv Area','Overall Qual','Lot Front-
age','Mas Vnr Area','BsmtFin SF 1','BsmtFin SF 2','Bsmt Unf SF'
,'Total Bsmt SF''Bsmt Full Bath','Bsmt Half Bath','Garage Yr
Blt','Garage Area','Lot Area','Year Remod/Add','1st Flr SF','2nd Flr
SF'
,'Low Qual Fin SF','Gr Liv Area','Wood Deck SF','Open Porch
SF','Pool Area','Yr Sold']

metrischstetig=['Lot Frontage','Mas Vnr Area','BsmtFin SF 1','BsmtFin
SF 2','Bsmt Unf SF','Total Bsmt SF','Bsmt Full Bath','Bsmt Half
Bath','Garage Yr Blt','Garage Cars','Garage Area','PID','Lot Ar-
ea','Year Built','Year Remod/Add','1st Flr SF','2nd Flr SF','Low Qual
Fin SF','Gr Liv Area','Full Bath','Half Bath','Bedroom AbvGr','Kitchen
AbvGr','TotRms AbvGrd','Fireplaces','Wood Deck SF','Open Porch
SF','Enclosed Porch','3Ssn Porch','Screen Porch','Pool Area','Misc
Val','Mo Sold','Yr Sold',]

kategorial_1=['MS Zoning','Street','Alley','Lot Shape','Land Contour',
'Utilities','Lot Config','Land Slope']

kategorial_2=['Neighborhood','Condition 1','Condition 2','Bldg
Type','House Style','Roof Style','Roof Matl','Exterior 1st','Exterior
2nd', 'Mas Vnr Type']

kategorial_3= ['Exter Qual','Exter Cond','Foundation','Bsmt
Qual','Bsmt Cond','Bsmt Exposure','BsmtFin Type 1', 'BsmtFin Type
2','Heating']

kategorial_4=['Heating QC','Central Air','Electrical','Kitchen
Qual','Functional']

kategorial_5=['Fireplace Qu','Garage Type','Garage Finish','Garage
Qual','Garage Cond','Paved Drive', 'Pool QC','Fence','Misc Feature']
#%%
#Speicherung aller kategorialen Variablen innerhalb einer Liste

```

```

cat_gesamt = ['MS Zoning','Street','Alley','Lot Shape','Land Contour',
'Utilities','Lot Config','Land Slope','Neighborhood','Condition
1','Condition 2','Bldg Type','House Style','Roof Style','Roof
Matl','Exterior 1st','Exterior 2nd', 'Mas Vnr Type',
'Exter Qual','Exter Cond','Foundation','Bsmt Qual','Bsmt
Cond','Bsmt Exposure','BsmtFin Type 1', 'BsmtFin Type 2','Heat-
ing','Exter Qual','Exter Cond','Foundation','Bsmt Qual','Bsmt
Cond','Bsmt Exposure','BsmtFin Type 1', 'BsmtFin Type 2','Heat-
ing','Heating QC','Central Air','Electrical','Kitchen Qual','Function-
al','Fireplace Qu','Garage Type','Garage Finish','Garage Qual','Garage
Cond','Paved Drive', 'Pool QC','Fence','Misc Feature']
#%%
#Graphische Korrelationsanalyse der kategorialen Variablen / Liste 1
import matplotlib.pyplot as plt
fig, PlotCanvas=plt.subplots( ncols=len(kategorial_1), figsize=(40,10))

# Darstellung der in der Liste definierten Variablen entgegen der
Zielvariable innerhalb eines Boxplots
for PredictorCol , i in zip(kategorial_1, range(len(kategorial_1))):
    Datensatz.boxplot(column='SalePrice', by=PredictorCol, figsize=(20,30), vert=True,ax=PlotCanvas[i])
#%%
#Graphische Korrelationsanalyse der kategorialen Variablen / Liste 2
import matplotlib.pyplot as plt
fig, PlotCanvas=plt.subplots( ncols=len(kategorial_2), figsize=(40,10))

# Darstellung der in der Liste definierten Variablen entgegen der
Zielvariable innerhalb eines Boxplots
for PredictorCol , i in zip(kategorial_2, range(len(kategorial_2))):
    Datensatz.boxplot(column='SalePrice', by=PredictorCol, figsize=(20,30), vert=True,ax=PlotCanvas[i])
#%%
#Graphische Korrelationsanalyse der kategorialen Variablen / Liste 3
import matplotlib.pyplot as plt
fig, PlotCanvas=plt.subplots( ncols=len(kategorial_3), figsize=(40,10))

# Darstellung der in der Liste definierten Variablen entgegen der
Zielvariable innerhalb eines Boxplots
for PredictorCol , i in zip(kategorial_3, range(len(kategorial_3))):
    Datensatz.boxplot(column='SalePrice', by=PredictorCol, figsize=(20,30), vert=True,ax=PlotCanvas[i])
#%%
#Graphische Korrelationsanalyse der kategorialen Variablen / Liste 4
import matplotlib.pyplot as plt
fig, PlotCanvas=plt.subplots( ncols=len(kategorial_4), figsize=(40,10))

# Darstellung der in der Liste definierten Variablen entgegen der
Zielvariable innerhalb eines Boxplots
for PredictorCol , i in zip(kategorial_4, range(len(kategorial_4))):
    Datensatz.boxplot(column='SalePrice', by=PredictorCol, figsize=(20,30), vert=True,ax=PlotCanvas[i])
#%%
#Graphische Korrelationsanalyse der kategorialen Variablen / Liste 5

```

```

import matplotlib.pyplot as plt
fig, PlotCanvas=plt.subplots( ncols=len(kategorial_5), figsize=(40,10))

# Darstellung der in der Liste definierten Variablen entgegen der Zielvariable innerhalb eines Boxplots
for PredictorCol , i in zip(kategorial_5, range(len(kategorial_5))): 
    Datensatz.boxplot(column='SalePrice', by=PredictorCol, figsize=(20,30), vert=True,ax=PlotCanvas[i])
#%%
#N here Betrachtung des Geb udetypen
plt.figure(figsize=(25,15))
order = Datensatz.groupby(by=[ "Condition 1"])[ "SalePrice"].mean().iloc[::-1].index
sns.boxplot(x='Condition 1', y='SalePrice',width=0.5,order=order,data=Datensatz)
#%%
### Metrische Variablen
#%%
#N here Betrachtung Condition
ContinuousColsList=[ 'Condition 1', 'Overall Qual', 'Overall Cond']

import matplotlib.pyplot as plt
fig, PlotCanvas=plt.subplots(nrows=1, ncols=len(ContinuousColsList),
figsize=(50,20))

for PredictorCol , i in zip(ContinuousColsList, range(len(ContinuousColsList))): 
    Datensatz.boxplot(column='SalePrice',by=PredictorCol, figsize=(50,20), vert=True, ax=PlotCanvas[i])
#%%
# Darstellung aller metrisch stetigen Variablen entgegen der Zielvariable mittels Streudiagramme
for predictor in metrischstetig:
    Datensatz.plot.scatter(x=predictor, y='SalePrice', figsize=(10,5),
title=predictor+" VS "+ 'SalePrice')
#%%
# Darstellung aller metrisch diskreten Variablen entgegen der Zielvariable mittels Streudiagramme
for predictor in metrischdiskret:
    Datensatz.plot.scatter(x=predictor, y='SalePrice', figsize=(10,5),
title=predictor+" VS "+ 'SalePrice')
#%%
pip install plotly==5.9.0
#%%
#Gegen berstellung der Zielvariabel und der Variablen Gesamtqualit t + Hausst l
import plotly.express as px

fig = px.histogram(Datensatz,
                    x = "Overall Qual",
                    y = "SalePrice",
                    marginal = None,
                    text_auto = True,
                    color = 'House Style',
                    hover_data = Datensatz.columns,
                    height = 500, width = 800)

```

```

fig.show()
#%%
# Data Preparation
#%%
## Qualität der Daten
#%%
#Darstellung aller fehlender Variablen in %
Alle_Fehlenden_Daten = (Datensatz.isnull().sum() / len(Datensatz)) * 100
Alle_Fehlenden_Daten = Alle_Fehlenden_Daten.drop(Alle_Fehlenden_Daten[Alle_Fehlenden_Daten == 0].index).sort_values(ascending=False) [:30]
Fehlende_Daten = pd.DataFrame({'Fehlender Wert in %' :Alle_Fehlenden_Daten})
Fehlende_Daten.head(20)
#%%
#Darstellung aller fehlender Variablen via Balkendiagramme
f, ax = plt.subplots(figsize=(15, 12))
plt.xticks(rotation='90')
sns.barplot(x=Alle_Fehlenden_Daten.index, y=Alle_Fehlenden_Daten)
plt.xlabel('Features', fontsize=15)
plt.ylabel('fehlende Werte in %', fontsize=15)
plt.title('% per Variabel', fontsize=15)
#%%
#Überprüfung vor der Anpassung
Datensatz.isnull().sum(axis=0)
#%%
#Neuer Datensatz für die Behandlung von fehlenden Werten
Datensatz_Neu = Datensatz
#%%
#Befüllung der Werte die NA als eigentlich None im Datensatz enthalten haben - Ableitung anhand der Datenbeschreibung
Datensatz_Neu["Pool QC"] = Datensatz_Neu["Pool QC"].fillna("None")
Datensatz_Neu["Misc Feature"] = Datensatz_Neu["Misc Feature"].fillna("None")
Datensatz_Neu["Alley"] = Datensatz_Neu["Alley"].fillna("None")
Datensatz_Neu["Fence"] = Datensatz_Neu["Fence"].fillna("None")
Datensatz_Neu["Fireplace Qu"] = Datensatz_Neu["Fireplace Qu"].fillna("None")
Datensatz_Neu["Bsmt Qual"] = Datensatz_Neu["Bsmt Qual"].fillna("No Basement")
Datensatz_Neu["Bsmt Cond"] = Datensatz_Neu["Bsmt Cond"].fillna("No Basement")
Datensatz_Neu["Bsmt Exposure"] = Datensatz_Neu["Bsmt Exposure"].fillna("No Basement")
Datensatz_Neu["BsmtFin Type 1"] = Datensatz_Neu["BsmtFin Type 1"].fillna("No Basement")
Datensatz_Neu["BsmtFin Type 2"] = Datensatz_Neu["BsmtFin Type 2"].fillna("No Basement")
Datensatz_Neu["Garage Type"] = Datensatz_Neu["Garage Type"].fillna("No Garage")
Datensatz_Neu["Garage Finish"] = Datensatz_Neu["Garage Finish"].fillna("No Garage")
Datensatz_Neu["Garage Qual"] = Datensatz_Neu["Garage Qual"].fillna("No Garage")
Datensatz_Neu["Garage Cond"] = Datensatz_Neu["Garage Cond"].fillna("No Garage")
#%%

```

```

#Überprüfung nach der Anpassung
Datensatz_Neu.isnull().sum(axis=0)
#%% md
### Änderung der NaN Werte der metrischen Variablen
%%

Datensatz_Neu.select_dtypes(include=['int64','float64'))
#%%
#Ersetzen der NaN Werte durch den Median Wert der diskreten Variablen
import numpy as np

Datensatz_Neu[['Lot Frontage','Mas Vnr Area','BsmtFin SF 1','BsmtFin
SF 2','Total Bsmt SF','Garage Area']] = Datensatz_Neu[['Lot Fronta-
ge','Mas Vnr Area','BsmtFin SF 1','BsmtFin SF 2','Total Bsmt SF','Ga-
rage Area']].replace(np.NaN,Datensatz_Neu[['Lot Frontage','Mas Vnr
Area','BsmtFin SF 1','BsmtFin SF 2','Total Bsmt SF','Garage
Area']].mean())
#%%
#Ersetzen der NaN Werte durch den Median Wert der stetigen Variablen
Datensatz_Neu['Garage Yr Blt'] = Datensatz_Neu['Garage Yr Blt'].re-
place(np.NaN,Datensatz_Neu['Garage Yr Blt'].median())
#%%
#Überprüfung der Daten
Datensatz_Neu.isnull().sum(axis=0)
#%% md
### Änderung der NaN Werte der Kategorischen Variablen
%%

Datensatz_Neu = Datensatz_Neu.fillna(Datensatz_Neu.mode().iloc[0])
#%%
#Überprüfung ob nun alle Werte kein NaN mehr aufweisen
Datensatz_Neu.info()
#%%
#Überprüfung aller Werte auf fehlende Einträge
Datensatz_Neu.isnull().sum(axis=0)
#%% md
## Korrelationsanalyse
#%%
### Metrische Variablen
#%%
# Korrelation nach Pearson
import numpy as np

rs = np.random.RandomState(0)
df = pd.DataFrame(rs.rand(81, 81))
corr = Datensatz_Neu.corr()[['SalePrice']].sort_values('Sal-
ePrice').tail(81)
corr.style.background_gradient(cmap='coolwarm').set_precision(2)
#%%
#Korrelation nach Spearman
import numpy as np

rs = np.random.RandomState(0)
df = pd.DataFrame(rs.rand(81, 81))
corr = Datensatz_Neu.corr(method="spearman")[['SalePrice']].sort_val-
ues('SalePrice').tail(81)
corr.style.background_gradient(cmap='coolwarm').set_precision(2)
#%%
# Multikorrelation Überprüfung nach Pearson

```

```

Datensatz_Neu.corr()
#%%
# Multikorrelation Überprüfung nach Spearman
Datensatz_Neu.corr(method="spearman")
#%%
# Kategoriale Variablen
#%%
# Definition einer Funktion zur Überprüfung der statistische Korrelationsanalyse
def FunctionAnova(inpData, TargetVariable, CategoricalPredictorList):
    from scipy.stats import f_oneway

    # Erstellung einer leeren Liste in der die Variablen gespeichert werden
    SelectedPredictors=[]

    print('##### ANOVA Results ##### \n')
    for predictor in CategoricalPredictorList:
        CategoryGroupLists=inpData.groupby(predictor) [TargetVariable].apply(list)
        AnovaResults = f_oneway(*CategoryGroupLists)

        # Sollte das Signifikanzniveau <0.05, dann wird die Variable / H0 verworfen - Ausgehend von keiner Korrelation
        if (AnovaResults[1] < 0.05):
            print(predictor, 'is correlated with', TargetVariable, '| P-Value:', AnovaResults[1])
            SelectedPredictors.append(predictor)
        else:
            print(predictor, 'is NOT correlated with', TargetVariable, '| P-Value:', AnovaResults[1])

    return(SelectedPredictors)
#%%
# Abruf der Funktion
CategoricalPredictorList=list(Datensatz_Neu.select_dtypes(include=['object']))
FunctionAnova(inpData=Datensatz_Neu,
              TargetVariable='SalePrice',
              CategoricalPredictorList=CategoricalPredictorList)
#%%
#Integration von Variabeln
Datensatz_Neu['TotalSF'] = Datensatz_Neu['Total Bsmt SF'] + Datensatz_Neu['1st Flr SF'] + Datensatz_Neu['2nd Flr SF']

Datensatz_Neu['Bath'] = Datensatz_Neu['Bsmt Full Bath'] + Datensatz_Neu['Bsmt Half Bath']*0.5 + Datensatz_Neu['Full Bath'] + Datensatz_Neu['Half Bath']*0.5

#Überprüfung der Integration
Datensatz_Neu.head(10)
#%%
#Überprüfung der Merkmalsausprägung
Datensatz_Neu['Bath'].value_counts()
#%%
Datensatz_Neu.drop(['Total Bsmt SF', '1st Flr SF', '2nd Flr SF', 'Bsmt Full Bath', 'Bsmt Half Bath', 'Full Bath', 'Bsmt Half Bath'],axis=1)
#%%

```

```

#Überprüfung der Neu geschaffenen Variablen anhand der Pearson Korrelation
import numpy as np

rs = np.random.RandomState(0)
df = pd.DataFrame(rs.rand(81, 81))
corr = Datensatz_Neu.corr()[['SalePrice']].sort_values('SalePrice').tail(81)
corr.style.background_gradient(cmap='coolwarm').set_precision(2)

#%%
#Überprüfung der Neu geschaffenen Variablen anhand der Spearman Korrelation
import numpy as np

rs = np.random.RandomState(0)
df = pd.DataFrame(rs.rand(81, 81))
corr = Datensatz_Neu.corr(method="spearman")[['SalePrice']].sort_values('SalePrice').tail(81)
corr.style.background_gradient(cmap='coolwarm').set_precision(2)

#%%
#Graphische Korrelationsanalyse der Variable TotalSF
Datensatz_Neu.plot.scatter(x='TotalSF', y='SalePrice', figsize=(10,5),
                           title='Total SF'+" VS "+ 'SalePrice')
#%%
md
## Überprüfung von Duplikaten
#%%
#Betrachtung des Datensatzes vor entfernung der Duplikate
print('Shape before deleting duplicate values:', Datensatz_Neu.shape)
# Entfernung von Duplikaten
Datensatz_Neu.drop_duplicates()
print('Shape After deleting duplicate values:', Datensatz_Neu.shape)
#%%
# Erstellung einer Liste mit den final ausgewählten Variablen
FinalenVariabeln=['SalePrice','MS Zoning','Street','Alley','Lot Shape','Land Contour','Lot Config','Land Slope','Neighborhood','Condition 1','Condition 2','Bldg Type','House Style','Roof Style','Roof Matl','Exterior 1st','Exterior 2nd','Mas Vnr Type','Exter Qual','Exter Cond','Foundation','Bsmt Qual','Bsmt Cond','Bsmt Exposure','BsmtFin Type 1','BsmtFin Type 2','Heating','Heating QC','Central Air','Electrical','Kitchen Qual','Functional','Fireplace Qu','Garage Type','Garage Finish','Garage Qual','Garage Cond','Paved Drive','Pool QC','Fence','Misc Feature','Sale Type','Sale Condition','TotRms AbvGrd','Fireplaces','1st Flr SF','Year Remod/Add','Total Bsmt SF','Garage Yr Blt','Bath','Garage Area','Year Built','Garage Cars','Gr Liv Area','Overall Qual','TotalSF' ]

# Erstellung des Datensatzes für das Training der Modelle
DatensatzML=Datensatz_Neu[FinalenVariabeln]

#%%
# Feature Engineering
#%%
# Überprüfung des Datensatzes
DatensatzML.head(1000)
#%%
# Speichern der Finalen Daten

```

```

DatensatzML.to_pickle('DatensatzML.pkl')
#%%
DatensatzML.head(100)
#%%
import pandas as pd

DatensatzML = pd.read_pickle('DatensatzML.pkl')
DatensatzML.head(5)
#%%
### Nur unter Einsatz der Überprüfung der Konsistenz
#DatensatzML['TotalSF'].replace([3003], 3010,inplace=True)
#DatensatzML['TotalSF'].replace([2910], 2905,inplace=True)
#DatensatzML['Year Built'].replace([1950], 1949,inplace=True)
#DatensatzML['Year Built'].replace([1968], 1966,inplace=True)
#DatensatzML['Garage Area'].replace([871], 890,inplace=True)
#DatensatzML['Garage Area'].replace([452], 445,inplace=True)
#%%
DatensatzML['Exter Qual'].replace({'Ex':5, 'Gd':4, 'TA':3,
'Fa':2, 'Po':1 }, inplace=True)
DatensatzML['Exter Cond'].replace({'Ex':5, 'Gd':4, 'TA':3,
'Fa':2, 'Po':1 }, inplace=True)
DatensatzML['Bsmt Qual'].replace({'Ex':5, 'Gd':4, 'TA':3,
'Fa':2, 'Po':1, 'No Basement':0 }, inplace=True)
DatensatzML['Bsmt Cond'].replace({'Ex':5, 'Gd':4, 'TA':3,
'Fa':2, 'Po':1, 'No Basement':0 }, inplace=True)
DatensatzML['Heating QC'].replace({'Ex':5, 'Gd':4, 'TA':3,
'Fa':2, 'Po':1 }, inplace=True)
DatensatzML['Kitchen Qual'].replace({'Ex':5, 'Gd':4, 'TA':3,
'Fa':2, 'Po':1 }, inplace=True)
DatensatzML['Fireplace Qu'].replace({'Ex':5, 'Gd':4, 'TA':3,
'Fa':2, 'Po':1, 'None':0 }, inplace=True)
DatensatzML['Garage Qual'].replace({'Ex':5, 'Gd':4, 'TA':3,
'Fa':2, 'Po':1, 'No Garage':0 }, inplace=True)
DatensatzML['Garage Cond'].replace({'Ex':5, 'Gd':4, 'TA':3,
'Fa':2, 'Po':1, 'No Garage':0 }, inplace=True)
DatensatzML['Pool QC'].replace({'Ex':4, 'Gd':3, 'TA':2, 'Fa':1,
'None':0 }, inplace=True)
#%%
DatensatzML['Exter Qual'] = pd.to_numeric(DatensatzML['Exter Qual'])
DatensatzML['Exter Cond'] = pd.to_numeric(DatensatzML['Exter Cond'])
DatensatzML['Bsmt Qual'] = pd.to_numeric(DatensatzML['Bsmt Qual'])
DatensatzML['Bsmt Cond']= pd.to_numeric(DatensatzML['Bsmt Cond'])
DatensatzML['Heating QC']= pd.to_numeric(DatensatzML['Heating QC'])
DatensatzML['Kitchen Qual']= pd.to_numeric(DatensatzML['Kitchen Qual'])
)
DatensatzML['Fireplace Qu']= pd.to_numeric(DatensatzML['Fireplace Qu'])
DatensatzML['Garage Qual']= pd.to_numeric(DatensatzML['Garage Qual'])
DatensatzML['Garage Cond']= pd.to_numeric(DatensatzML['Garage Cond'])
DatensatzML['Pool QC']= pd.to_numeric(DatensatzML['Pool QC'])
#%%
#Überprüfung der Anpassung
DatensatzML
#%%
# Aufteilung der Zielvariable und der unabhängigen Variable
y = DatensatzML['SalePrice']
X = DatensatzML.drop("SalePrice",axis=1)
#%% md

```

```

#### Anpassung für LIME XGB
#%%
# Erstellung der Listen für das Mapping der kategorialen Variablen
num_Variablen = ['TotRms AbvGrd',
                 'Fireplaces',
                 '1st Flr SF',
                 'Year Remod/Add',
                 'Total Bsmt SF',
                 'Garage Yr Blt',
                 'Bath',
                 'Garage Area',
                 'Year Built',
                 'Garage Cars',
                 'Gr Liv Area',
                 'Overall Qual',
                 'TotalSF', 'Pool QC', 'Garage Qual', 'Garage Cond',
                 'Fireplace Qu', 'Heating QC', 'Kitchen Qual', 'Bsmt
Qual', 'Bsmt Cond', 'Exter Qual', 'Exter Cond']

kat_Variablen = ['MS Zoning', 'Street', 'Alley', 'Lot Shape', 'Land Con-
tour', 'Lot Config', 'Land Slope', 'Neighborhood', 'Condition 1', 'Condi-
tion 2', 'Bldg Type', 'House Style', 'Roof Style', 'Roof Matl', 'Exterior
1st', 'Exterior 2nd', 'Mas Vnr Type', 'Foundation', 'Bsmt Exposure', 'Bsmt-
Fin Type 1', 'BsmtFin Type 2', 'Heating', 'Central Air', 'Electri-
cal', 'Functional', 'Garage Type', 'Garage Finish', 'Paved
Drive', 'Fence', 'Misc Feature', 'Sale Type', 'Sale Condition']
#%%
# ## Trainings-, und Test-Splitting
#%%
#Trainings-, und Test-Split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=.3, random_state=42)
#%%
# ## Standardisierung
#%%
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OrdinalEncoder
from sklearn.preprocessing import StandardScaler
preprocessor = ColumnTransformer([('numerical', StandardScaler(),
num_Variablen),
                                  ("categorical", OneHotEn-
coder(sparse=False, handle_unknown="ignore"), kat_Variablen)])
#%%
# Train preprocessor
preprocessor.fit(X_train)
# Get the list of categories generated by the process
ohe_categories = preprocessor.named_transformers_["categorical"].cate-
gories_
# Create nice names for our one hot encoded features
new_ohe_features = [f"{col}_{val}" for col, vals in zip(kat_Varia-
blen, ohe_categories) for val in vals]
# Create a new list with all names of features
all_features = num_Variablen + new_ohe_features
# Save processed data

```

```

X_train_processed = pd.DataFrame(preprocessor.transform(X_train), columns=all_features)
X_test_processed = pd.DataFrame(preprocessor.transform(X_test), columns=all_features)
# %% md
# Definition Bewertungsmetriken
#%%
def Bewertungsmetriken(true, predicted, train=True):
    mae = metrics.mean_absolute_error(true, predicted)
    mse = metrics.mean_squared_error(true, predicted)
    mape = metrics.mean_absolute_percentage_error(true, predicted)
    rmse = np.sqrt(metrics.mean_squared_error(true, predicted))
    r2_square = metrics.r2_score(true, predicted)
    Adjusted_r2_square = 1 - (1-r2_square)*(len(X)-1)/(len(X) - X.shape[1]-1)
    if train:
        print("=====Ergebnisse Training=====")
        print('MAE: ', mae)
        print('MSE: ', mse)
        print('MAPE: ', mape)
        print('RMSE: ', rmse)
        print('R2 Square: ', r2_square)
        print('Adjusted R2 Square: ', Adjusted_r2_square)
    elif not train:
        print("=====Ergebnisse Test=====")
        print('MAE: ', mae)
        print('MSE: ', mse)
        print('MAPE: ', mape)
        print('RMSE: ', rmse)
        print('R2 Square: ', r2_square)
        print('Adjusted R2 Square: ', Adjusted_r2_square)
    #%%
# %% md
# K-nearest Neighbor
#%%
# K-Nearest Neighbor (KNN)
from sklearn.neighbors import KNeighborsRegressor
from sklearn.decomposition import PCA
from sklearn.pipeline import Pipeline
import numpy as np

KNNReg = KNeighborsRegressor(n_neighbors=6)

# Darstellung aller Parameter des KNN-Algorithmus
print(KNNReg)

# Erstellung des Models anhand der Trainingsdaten
KNN=KNNReg.fit(X_train_processed,y_train)
KNNPredict=KNN.predict(X_train_processed)

from sklearn import metrics
# Anwendung der Bewertungsmetriken
Bewertungsmetriken(y_train,KNNPredict,train=True)
#%%
import joblib

```

```

# Speicherung des besten Modells
joblib.dump(KNN, 'KNN_best_Model.pkl')
#%%
# Laden des besten Modells in die Umgebung
KNN = joblib.load('KNN_best_Model.pkl')
#%%
# Anwendung der Bewertungsmaßnahmen auf den Trainingsdaten
KNNPredict = KNN.predict(X_train_processed)
Bewertungsmaßnahmen(y_train,KNNPredict,train=True)
#%%
# Darstellung der Vorhersage der Trainingsdaten in einem Dataframe
Trainingsdata = pd.DataFrame(data=X_train_processed,columns=all_features)
Trainingsdata['Vorhersage_SalePrice']=KNNPredict
Trainingsdata['Vorhersage_SalePrice'].head(20)
#%%
# Anwendung an den Testdaten
KNNPredict = KNN.predict(X_test_processed)
Bewertungsmaßnahmen(y_test,KNNPredict,train=True)
#%%
# Darstellung der Vorhersage der Testdaten in einem Dataframe
Vorhersage_TestDaten_KNN = KNN.predict(X_test_processed.values)
Bewertungsmaßnahmen(y_test, Vorhersage_TestDaten_KNN, train=False)

# Erstellung eines DataFrames um die Test Daten zu analysieren
TestingData=pd.DataFrame(data=X_test_processed, columns=all_features)
Trainingsdata['Sale Price']=y_train
TestingData['Vorhersage_SalePrice']=Vorhersage_TestDaten_KNN
print(TestingData['Vorhersage_SalePrice'].head(10))
# %% md
## SHAP KNN
#%%
import shap
explainer_KNN = shap.Explainer(KNN.predict,X_train_processed)
# Berechnung der SHAP-Values
shap_values_KNN = explainer_KNN(X_train_processed)
# %% md
### Global
#%%
# Definition des SHAP Objects für die Globale Instanz
shap_object_global_KNN = shap.Explanation(base_values = shap_values_KNN.base_values,
                                             values = shap_values_KNN.values,
                                             feature_names = all_features,
                                             data = shap_values_KNN.data)
#%%
# Balkendiagramm / Feature Importance
shap.plots.bar(shap_object_global_KNN,max_display=11)
#%%
# Beeswarm-Diagramm
shap.summary_plot(shap_object_global_KNN, X_train_processed,feature_names=all_features, max_display=10)
# %% md
### Lokal
#%%
# Definition des SHAP Objects für Lokale Instanzen

```

```

shap_object_KNN = shap.Explanation(base_values = shap_val-
ues_KNN[0][0].base_values,
                                     values = shap_val-
ues_KNN[0].values,
                                     feature_names = all_fea-
tures,
                                     data = shap_val-
ues_KNN[0].data)
#%%
#Force-Diagramm
shap.initjs()
shap.plots.force(shap_object_KNN,matplotlib=True)
#%%
#Wasserfall-Diagramm
shap.plots.waterfall(shap_object_KNN,max_display=20)
#%%
## LIME KNN
#%%
import lime
import lime.lime_tabular
import numpy as np
#%%
#### Definition des LIME Mappings einmalig für alle Modelle
#%%
categorical_names = {}

# Liste aller möglichen Variablen
kat_Werte = preprocessor.named_transformers_["categorical"].catego-
ries_

for col, val in zip(kat_Variablen, kat_Werte):
    categorical_names[X.columns.get_loc(col)] = list(val)
#%%
pip install helpers
#%%
import helpers
#%%
import pandas as pd

def convert_to_lime_format(X, categorical_names, col_names=None, in-
vert=False):
    """Converts data with categorical values as string into the right
format
    for LIME, with categorical values as integers labels.
    It takes categorical_names, the same dictionary that has to be
passed
    to LIME to ensure consistency.
    col_names and invert allow to rebuild the original DataFrame from
a numpy array in LIME format to be passed to a Pipeline or sklearn
OneHotEncoder
    """
    # If the data isn't a dataframe, we need to be able to build it
    if not isinstance(X, pd.DataFrame):
        X_lime = pd.DataFrame(X, columns=col_names)
    else:

```

```

        X_lime = X.copy()

    for k, v in categorical_names.items():
        if not invert:
            label_map = {
                str_label: int_label for int_label, str_label in enumerate(v)
            }

        else:
            label_map = {
                int_label: str_label for int_label, str_label in enumerate(v)
            }

    X_lime.iloc[:, k] = X_lime.iloc[:, k].map(label_map)

    return X_lime
#%%
convert_to_lime_format(X_train, categorical_names).head(10)
#%%
import lime.lime_tabular
explainer = lime.lime_tabular.LimeTabularExplainer(convert_to_lime_format(X_train, categorical_names).values,
                                                    mode="regression",
                                                    feature_names=X_train.columns,
                                                    categorical_names=categorical_names,
                                                    categorical_features=categorical_names.keys(),
                                                    random_state=42)

#%%
i = 0

# First we can check the row in the non-processed dataset
print(X_train.iloc[i])

# Then we extract the observation from the processed dataset
X_observation = X_train_processed.iloc[[i], :]
X_observation
#%%
observation = convert_to_lime_format(X_train.iloc[[i], :], categorical_names).values[0]
observation
#%%
pip install functools
#%%
# Let write a custom predict_proba functions for our models:
from functools import partial

def custom_predict(X, model):
    """Takes X in LIME format, converts back to raw, then processes it and calls predict_proba"""

    X_original = convert_to_lime_format(X, categorical_names,
                                         col_names=X_train.columns, invert=True)
    X_processed = preprocessor.transform(X_original)
    return model.predict(X_processed)

```

```

KNN_predict = partial(custom_predict, model=KNN)
XGB_predict = partial(custom_predict, model=XGB)
ANN_predict = partial(custom_predict, model=r)
EBM_predict = partial(custom_predict, model=ebm_model)
# %% md
### Lokal
#%%
explanation = explainer.explain_instance(observation, KNN_predict,
num_features=10)
explanation.show_in_notebook(show_table=True, show_all=False)
i = 0

# First we can check the row in the non-processed dataset
print(X_train.iloc[i])

# Then we extract the observation from the processed dataset
X_observation = X_train_processed.iloc[[i], :]
X_observation
# %% md
### Global
#%%
def return_weights(exp):

    """Zusammenfassung der Gewichtung der LIME Objekte"""

    exp_list = exp.as_map()[1]
    exp_list = sorted(exp_list, key=lambda x: x[0])
    exp_weight = [x[1] for x in exp_list]

    return exp_weight

weights = []

#Wiederholung durch die Anzahl an Zeilen der Beobachtungen
for x in observation:

    #Abholung der Erklärungen
    exp = explainer.explain_instance(observation, KNN_predict,
                                      num_features=55)

    #Abholung der Gewichtungen
    exp_weight = return_weights(exp)
    weights.append(exp_weight)

#Erstellung DataFrames mit den Gewichtungen von LIME und Darstellung
lime_weights = pd.DataFrame(data=weights, columns=X_train.columns)
lime_weights.head(100)

#Balken Diagramm Feature Importance
abs_mean = lime_weights.abs().mean(axis=0)
abs_mean = pd.DataFrame(data={'feature':abs_mean.index,
'abs_mean':abs_mean}, )

```

```

abs_mean = abs_mean.sort_values('abs_mean', ascending=0) [:10]

#Darstellung des absoluten Durchschnitts
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(25, 25),)

y_ticks = range(len(abs_mean))
y_labels = abs_mean.feature
plt.barh(y=y_ticks, width=abs_mean.abs_mean)

plt.yticks(ticks=y_ticks, labels=y_labels, size= 20)
plt.title('')
plt.ylabel('')
plt.xlabel('Mean |Weight|', size=20)

#Balken Diagramm Feature Importance
abs_mean = lime_weights.abs().mean(axis=0)
abs_mean = pd.DataFrame(data={'feature':abs_mean.index,
 'abs_mean':abs_mean},)
abs_mean = abs_mean.sort_values('abs_mean', ascending=0)
print(abs_mean[:10])
#%%
## InterpretML KNN
#%%
pip install interpret
#%%
## Lokal
#%%
from interpret.blackbox import ShapKernel
from interpret import show

shap_KNN = ShapKernel(predict_fn=KNN.predict, data=X_train_processed,
                      feature_names=all_features)
shap_local_KNN = shap_KNN.explain_local(X_train_processed[:2],
                                         y_train[:2])

show(shap_local_KNN)
#%%
from interpret.blackbox import LimeTabular
from interpret import show

lime = LimeTabular(predict_fn=KNN.predict, data=X_train_processed,
                    random_state=1, feature_names=all_features)

lime_local_KNN = lime.explain_local(X_train_processed[0:2],
                                     y_train[0:2])
show(lime_local_KNN)
#%%
## Global
#%%
from interpret.blackbox import MorrisSensitivity

sensitivity = MorrisSensitivity(predict_fn=KNN.predict,
                                 data=X_train_processed, feature_names=all_features)
KNN_Global = sensitivity.explain_global(name="Global Sensitivity")

show(KNN_Global)
#%%

```

```

from interpret.blackbox import PartialDependence

pdp = PartialDependence(predict_fn=KNN.predict, data=X_train_pro-
cessed)
pdp_global_KNN = pdp.explain_global(name='Partial Dependence')

show(pdp_global_KNN)
# %% md
### Vergleich der Methoden mittels InterpretML Dashboard
# %%
from interpret import set_visualize_provider
from interpret.provider import DashProvider
set_visualize_provider(DashProvider.from_address(('127.0.0.1', 7001)))
# %%
# Vergleich Globaler und Lokaler XAI Methoden anhand InterpretML
show([lime_local_KNN,KNN_Global,pdp_global_KNN,shap_local_KNN])
# %% md
# XGBoost Model Training
# %%
pip install xgboost
# %%
# Xtreme Gradient Boosting Definition Algorithmus (XGBoost)
import xgboost as xgb
from xgboost import XGBRegressor
from xgboost import plot_tree
import matplotlib.pyplot as plt
import numpy as np

RegModel=XGBRegressor(max_depth=3,
                      learning_rate=0.01,
                      n_estimators=1000,
                      objective='reg:squarederror',
                      booster='gbtree',
                      tree_method= 'exact')

# Darstellung aller Parameter
print(RegModel)

# Definition des Models anhand der Trainingsdaten
XGB=RegModel.fit(X_train_processed,y_train)
XGBprediction=XGB.predict(X_train_processed)

from sklearn import metrics
# Messung der Performance des Models
Bewertungsmetriken(y_train,XGBprediction,train=True)
# %%
import joblib

# Speicherung des besten Modells
joblib.dump(XGB, 'XGB_best_Model.pkl')
# %%
import joblib
import numpy as np
# Laden des besten Models
XGB = joblib.load('XGB_best_Model.pkl')
XGBprediction=XGB.predict(X_train_processed)

```

```

from sklearn import metrics
# Messung der Performance des besten Models anhand der Trainingsdaten
Bewertungsmetriken(y_train,XGBprediction,train=True)
#%%
Vorhersage_TrainingsDaten = XGB.predict(X_train_processed)
Bewertungsmetriken(y_train, Vorhersage_TrainingsDaten, train=False)
# Erstellung eines Data Frames um die Trainingsdaten zu analysieren
TrainingData=pd.DataFrame(data=X_train_processed, columns=X_train_processed.columns)
TrainingData['Sale Price']=y_train
TrainingData['Vorhersage_SalePrice']=Vorhersage_TrainingsDaten
TrainingData.head(200)
#%%
from sklearn import metrics
# Messung der Performance anhand der Testdaten
XGBprediction=XGB.predict(X_test_processed)
Bewertungsmetriken(y_test,XGBprediction,train=True)
#%%
Vorhersage_TestDaten = XGB.predict(X_test_processed)
Bewertungsmetriken(y_test, Vorhersage_TestDaten, train=False)

# Erstellung eines Data Frames um die Testdaten zu analysieren
TestingData=pd.DataFrame(data=X_test_processed, columns=X_test_processed.columns)
TestingData['Sale Price']=y_test
TestingData['Vorhersage_SalePrice']=Vorhersage_TestDaten
TestingData.head(20)
#%% md
## SHAP XGB
#%%
import shap
#%%
#Definition der Variablenbezeichnung für die Nutzung von SHAP
cols_when_model_builds = XGB.get_booster().feature_names
X_train_processed = X_train_processed[cols_when_model_builds]
#%%
#Berechnung der SHAP-Values
explainer_XGB = shap.TreeExplainer(XGB,X_train_processed,feature_names=X_train_processed.columns)
shap_values_XGB = explainer_XGB.shap_values(X_train_processed,y=y_train)
#%%
#TreeExplainer hat keinen Masker, daher manuelle berechnung des Base-Wertes
from shap import Explanation
exp = Explanation(shap_values_XGB,explainer_XGB.expected_value,
data=X_train_processed, feature_names=X_train_processed.columns)
#%% md
### Global
#%%
#Darstellung Balken-Diagramm
shap.plots.bar(exp,max_display=11)
#%%
#Darstellung Beeswarm-Diagramm
shap.summary_plot(exp,max_display=10)
#%%
#Darstellung Dependence Diagramm für TotalSF und SalePrice

```

```

shap.dependence_plot('TotalSF', shap_values_XGB, X_train_processed, feature_names=X_train_processed.columns, interaction_index='Overall Qual')
#%%
# Sortierung der einflussreichsten Variablen Absteigend
top_inds = np.argsort(-np.sum(np.abs(shap_values_XGB), 0))

# Darstellung Dependence Diagramme der 3 einflussreichsten Variablen
for i in range(3):
    shap.dependence_plot(top_inds[i], shap_values_XGB, X_train_processed, feature_names=all_features)
#%%
## Lokal
#%%
#Darstellung Wasserfall-Diagramm Beobachtung 1
shap.plots.waterfall(exp[0], max_display=10)
#%%
import matplotlib
#Darstellung Force-Diagramm
shap.force_plot(exp[0],matplotlib=True)
#%%
#Darstellung Wasserfall-Diagramm Beobachtung 2
shap.plots.waterfall(exp[1], max_display=10)
#%%
import matplotlib
#Darstellung Wasserfall-Diagramm Beobachtung 2
shap.force_plot(exp[1],matplotlib=True)
#%%
## LIME XGB
#%%
## Lokal
#%%
# Angabe welche Beobachtung lokal erklärt werden soll
i = 0

# Then we extract the observation from the processed dataset
X_observation = X_train_processed.iloc[[i], :]
X_observation

#Speicherung der Beobachtung zur lokalen Erklärung
observation = convert_to_lime_format(X_train.iloc[[i], :], categorical_names).values[0]
observation
#%%
explanation = explainer.explain_instance(observation,XGB_predict , num_features=10)
explanation.show_in_notebook(show_table=True, show_all=False)
#%%
## Global
#%%
def return_weights(exp):

    """Zusammenfassung der Gewichtung der LIME Objekte"""

    exp_list = exp.as_map()[1]
    exp_list = sorted(exp_list, key=lambda x: x[0])
    exp_weight = [x[1] for x in exp_list]

```

```

    return exp_weight

weights = []

#Wiederholung durch die Anzahl an Zeilen der Beobachtungen
for x in observation:

    #Abholung der Erklärungen
    exp = explainer.explain_instance(observation,XGB_predict,
                                      num_features=55)

    #Abholung der Gewichtungen
    exp_weight = return_weights(exp)
    weights.append(exp_weight)

#Erstellung DataFrames mit den Gewichtungen von LIME und Darstellung
lime_weights = pd.DataFrame(data=weights,columns=X_train.columns)
lime_weights.head(100)

#Balken Diagramm Feature Importance
abs_mean = lime_weights.abs().mean(axis=0)
abs_mean = pd.DataFrame(data={'feature':abs_mean.index,
                             'abs_mean':abs_mean},)
abs_mean = abs_mean.sort_values('abs_mean',ascending=0)[:10]

#Darstellung des absoluten Durchschnitts
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(25,25),)

y_ticks = range(len(abs_mean))
y_labels = abs_mean.feature
plt.barh(y=y_ticks, width=abs_mean.abs_mean)

plt.yticks(ticks=y_ticks, labels=y_labels, size= 20)
plt.title('')
plt.ylabel('')
plt.xlabel('Mean |Weight|', size=20)

#Balken Diagramm Feature Importance
abs_mean = lime_weights.abs().mean(axis=0)
abs_mean = pd.DataFrame(data={'feature':abs_mean.index,
                             'abs_mean':abs_mean},)
abs_mean = abs_mean.sort_values('abs_mean',ascending=0)
print(abs_mean[:10])
#%%
## InterpretML XGB
#%%
## Lokal
#%%
from interpret.blackbox import LimeTabular
from interpret import show

lime = LimeTabular(predict_fn=XGB.predict, data=X_train_processed,
random_state=1, feature_names=all_features)

```

```

XGB_lime_local = lime.explain_local(X_train_processed[:2],y_train[:2])
show(XGB_lime_local)
#%%
from interpret.blackbox import ShapKernel
from interpret import show

shap_XGB = ShapKernel(predict_fn=XGB.predict, data=X_train_pro-
cessed, feature_names=X_train_processed.columns)
XGB_shap_local = shap_XGB.explain_local(X_train_processed[:2],
y_train[:2])

show(XGB_shap_local)
# %% md
### Global
#%%
from interpret.blackbox import MorrisSensitivity

sensitivity = MorrisSensitivity(predict_fn=XGB.predict,
data=X_train_processed, feature_names=all_features)
XGB_Global = sensitivity.explain_global(name="Global Sensitivity")

show(XGB_Global)
#%%
from interpret.blackbox import PartialDependence

pdp = PartialDependence(predict_fn=RegModel.predict, data=X_train_pro-
cessed, feature_names=all_features)
pdp_global_XGB = pdp.explain_global(name='Partial Dependence')

show(pdp_global_XGB)
# %% md
### Vergleich der XAI Methoden
#%%
from interpret import set_visualize_provider
from interpret.provider import DashProvider
set_visualize_provider(DashProvider.from_address('127.0.0.1', 7001))
# %%
# Vergleich Globaler und Lokaler XAI Methoden anhand InterpretML
show([XGB_shap_local,XGB_lime_local,XGB_Global,pdp_global_XGB])
# %% md
# Model Neuronales Netz
#%%
#https://www.nature.com/articles/s41598-022-15618-4.pdf
# %%
# Abrufen der benötigten Bibliotheken
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
from sklearn.datasets import make_classification
from sklearn.model_selection import cross_val_score
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from keras import backend as K
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.optimizers import SGD
from keras.models import load_model
#%%
from sklearn import metrics

```

```

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Input, Dense, Activation, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint

#Definition zur Speicherung des Besten Models
keras_callbacks = [
    EarlyStopping(monitor='val_loss', patience=30, mode='min',
min_delta=0.01)]
mc =ModelCheckpoint('best_model_ANN.h5',monitor='val_loss', mode='min',
verbose=1, save_best_only=True)
#Definition von Hyperparametern
EPOCHS = 500
BATCH_SIZE = 128
DROPOUT =0.01
OPT = Adam(learning_rate=0.0015)
#%%
model = tf.keras.Sequential()

model.add(Dense(X_train_processed.shape[1],activation='relu'))
model.add(Dense(32,activation='relu'))
model.add(Dropout(DROPOUT))

model.add(Dense(64,activation='relu'))
model.add(Dropout(DROPOUT))

model.add(Dense(128,activation='relu'))
model.add(Dropout(DROPOUT))
model.add(Dense(1))

model.compile(optimizer=OPT, loss='mae',metrics=['mse', 'mae', 'mape',])

r = model.fit(X_train_processed, y=y_train,
                validation_data=(X_test_processed,y_test),
                batch_size=BATCH_SIZE,
                epochs=EPOCHS,
                callbacks= [keras_callbacks,mc])
#%%
#Speicherung der besten Trainingsperformance
np.save('best_model_ANN_history.npy',r.history)
#%%
#Laden des besten Models inklusive der Leistungshistorie
from tensorflow.keras.models import load_model
import numpy as np
r = load_model('best_model_ANN.h5')
r_History=np.load('best_model_ANN_history.npy',allow_pickle='TRUE').item()
#%%
#Vorhersage/ des besten Models
r.predict(X_train_processed)
r.evaluate(X_train_processed, y_train)
r.summary()
#%%
ANN_y_train_pred = r.predict(X_train_processed)
Bewertungsmaetriken(y_train, ANN_y_train_pred, train=True)

```

```

Trainingsdata = pd.DataFrame(data=X_train_processed,columns=all_features)
Trainingsdata['Sale Price']=y_train
Trainingsdata['Vorhersage_SalePrice']=ANN_y_train_pred
Trainingsdata.head(5)
#%%
#Darstellung der Verlustfunktion
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))

plt.plot(r_History['loss'], label='loss')
plt.plot(r_History['val_loss'], label='val_loss')
plt.legend()
#%%
#Vorhersage der Testdaten
Vorhersage_TestDaten = r.predict(X_test_processed)
Bewertungsmetriken(y_test, Vorhersage_TestDaten, train=False)

# Erstellung eines Data Frames um die Test Daten zu analysieren
TestingData=pd.DataFrame(data=X_test_processed, columns=all_features)
TestingData['Sale Price']=y_test
TestingData['Vorhersage_SalePrice']=Vorhersage_TestDaten
TestingData.head(20)
#%%
## SHAP ANN
#%%
pip install Numba==0.53
#%%
pip install Numpy==1.21.4
#%%
import numpy
import numba
import shap
#%%
import shap
# Berechnung der SHAP-Werte
explainer = shap.Explainer(r.predict,X_train_processed)
shap_values_ANN = explainer(X_train_processed)
#%%
## Global
#%%
#Definition SHAP für Globale Analyse

shap_object = shap.Explanation(base_values = shap_values_ANN.base_values,
                                 values = shap_values_ANN.values,
                                 feature_names = all_features,
                                 data = shap_values_ANN.data)
#%%
shap.plots.bar(shap_object,max_display= 10)
#%%
shap.summary_plot(shap_values_ANN,X_train_processed,feature_names =
all_features, max_display=10)
#%%
## Lokal
#%%
shap_object_ANN = shap.Explanation(base_values = shap_values_ANN[0][0].base_values,

```

```

values = shap_values_ANN[0].values,
feature_names = all_features,
data = shap_values_ANN[0].data)
#%%
shap.plots.waterfall(shap_object_ANN,max_display=10)
#%%
shap.initjs()
shap.plots.force(shap_object_ANN,matplotlib=True)
#%%
md
## LIME ANN
#%%
pip install lime
#%%
md
### Lokal
#%%
# Angabe welche Beobachtung lokal erklärt werden soll
i = 0

# Then we extract the observation from the processed dataset
X_observation = X_train_processed.iloc[[i], :]
X_observation

#Speicherung der Beobachtung zur lokalen Erklärung
observation = convert_to_lime_format(X_train.iloc[[i], :],categorical_names).values[0]
observation
#%%
explanation = explainer.explain_instance(observation,KNN_predict ,
num_features=10)
explanation.show_in_notebook(show_table=True, show_all=False)
#%%
md
### Global LIME (Ansatz)
#%%
def return_weights(exp):

    """Zusammenfassung der Gewichtung der LIME Objekte"""

    exp_list = exp.as_map()[1]
    exp_list = sorted(exp_list, key=lambda x: x[0])
    exp_weight = [x[1] for x in exp_list]

    return exp_weight

weights = []

#Wiederholung durch die Anzahl an Zeilen der Beobachtungen
for x in observation:

    #Abholung der Erklärungen
    exp = explainer.explain_instance(observation,ANN_predict,
                                      num_features=55)

    #Abholung der Gewichtungen
    exp_weight = return_weights(exp)
    weights.append(exp_weight)

```

```

#Erstellung DataFrames mit den Gewichtungen von LIME und Darstellung
lime_weights = pd.DataFrame(data=weights,columns=X_train.columns)
lime_weights.head(100)

#Balken Diagramm Feature Importance
abs_mean = lime_weights.abs().mean(axis=0)
abs_mean = pd.DataFrame(data={'feature':abs_mean.index,
'abs_mean':abs_mean},)
abs_mean = abs_mean.sort_values('abs_mean',ascending=0)[:10]

#Darstellung des absoluten Durchschnitts
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(25,25),)

y_ticks = range(len(abs_mean))
y_labels = abs_mean.feature
plt.barh(y=y_ticks, width=abs_mean.abs_mean)

plt.yticks(ticks=y_ticks, labels=y_labels, size= 20)
plt.title('')
plt.ylabel('')
plt.xlabel('Mean |Weight|', size=20)

#Balken Diagramm Feature Importance
abs_mean = lime_weights.abs().mean(axis=0)
abs_mean = pd.DataFrame(data={'feature':abs_mean.index,
'abs_mean':abs_mean},)
abs_mean = abs_mean.sort_values('abs_mean',ascending=0)
print(abs_mean[:10])
#%%
## InterpretML ANN
#%%
## Lokal
#%%
from interpret.blackbox import LimeTabular
from interpret import show

lime = LimeTabular(predict_fn=ANN.predict, data=X_train_processed,
random_state=1, feature_names=all_features)

ANN_lime_local = lime.explain_local(X_train_processed[:2], y_train[:2])
show(ANN_lime_local)
#%%
from interpret.blackbox import ShapKernel
from interpret import show

shap_ANN = ShapKernel(predict_fn=XGB.predict, data=X_train_pro-
cessed, feature_names=X_train_processed.columns)
ANN_shap_local = shap_ANN.explain_local(X_train_processed[:2],
y_train[:2])

show(ANN_shap_local)
#%%
## Global
#%%
from interpret.blackbox import MorrisSensitivity

```

```

sensitivity = MorrisSensitivity(predict_fn=model.predict,
data=X_train_processed[100:], feature_names=all_features)
ANN_Global = sensitivity.explain_global(name="Global Sensitivity")

show(ANN_Global)
#%%
from interpret.blackbox import PartialDependence

pdp = PartialDependence(predict_fn=model.predict, data=X_train, fea-
ture_names=Unabhängige_Variablen)
ANN_pdp_global = pdp.explain_global(name='Partial Dependence')

show(ANN_pdp_global)
#%% md
# Explainable Boosting Model
#%%
pip install interpret
#%%
import interpret
from interpret.glassbox import ExplainableBoostingRegressor
from interpret import show
#%%
# Training des Modells
seed=1
ebm = ExplainableBoostingRegressor(random_state=seed,fea-
ture_names=all_features)
ebm_model = ebm.fit(X_train_processed, y_train)
ebmprediction=ebm.predict(X_train_processed)

from sklearn import metrics
# Einsatz Bewertungsmetriken
Bewertungsmetriken(y_train,ebmprediction,train=True)
#%%
import joblib

# Speichern des besten Modells
joblib.dump(ebm_model, 'EBM_best_Model.pkl')
#%%
import joblib
# Laden des besten Modells
ebm_model = joblib.load('EBM_best_Model.pkl')
EBMprediction=ebm_model.predict(X_train_processed)
from sklearn import metrics
# Einsatz Bewertungsmetriken
Bewertungsmetriken(y_train,EBMprediction,train=True)
#%%
#Vorhersage Testdaten
Vorhersage_TestDaten_EBM = ebm_model.predict(X_test_processed)
Bewertungsmetriken(y_test, Vorhersage_TestDaten_EBM, train=False)

# Erstellung eines Data Frames um die Test Daten zu analysieren
TestingData=pd.DataFrame(data=X_test_processed, columns=all_features)
TestingData['Sale Price']=y_test
TestingData['Vorhersage_SalePrice']=Vorhersage_TestDaten_EBM
TestingData.head(20)
#%% md
## SHAP EBM

```

```

# %% md
### Global
# %%
import shap
explainer_EBM = shap.Explainer(ebm_model.predict,X_train_processed)
# Berechnung der SHAP-Werte
shap_values_EBM = explainer_EBM(X_train_processed)
# %%
shap_object_EBM_global = shap.Explanation(base_values = shap_values_EBM.base_values,
                                             values = shap_values_EBM.values,
                                             feature_names = all_features,
                                             data = shap_values_EBM.data)
# %%
shap.summary_plot(shap_values_EBM,X_train_processed,feature_names =
all_features, max_display=10)
# %%
shap.plots.bar(shap_object_EBM_global,max_display=11)
# %% md
### Lokal
# %%
shap_object_EBM = shap.Explanation(base_values = shap_values_EBM[0][0].base_values,
                                             values = shap_values_EBM[0].values,
                                             feature_names = all_features,
                                             data = shap_values_EBM[0].data)
# %%
shap.plots.waterfall(shap_object_EBM,max_display=10)
# %% md
## LIME EBM
# %% md
### Lokal
# %%
# Angabe welche Beobachtung lokal erklärt werden soll
i = 0

# Then we extract the observation from the processed dataset
X_observation = X_train_processed.iloc[[i], :]
X_observation

# Speicherung der Beobachtung zur lokalen Erklärung
observation = convert_to_lime_format(X_train.iloc[[i], :],categorical_names).values[0]
observation
# %%
explanation = explainer.explain_instance(observation,EBM_predict ,
num_features=10)
explanation.show_in_notebook(show_table=True, show_all=False)
# %% md
### Global
# %%
def return_weights(exp):

    """Zusammenfassung der Gewichtung der LIME Objekte"""

    exp_list = exp.as_map()[1]
    exp_list = sorted(exp_list, key=lambda x: x[0])
    exp_weight = [x[1] for x in exp_list]

```

```

return exp_weight

weights = []

#Wiederholung durch die Anzahl an Zeilen der Beobachtungen
for x in observation:

    #Abholung der Erklärungen
    exp = explainer.explain_instance(observation, EBM_predict,
                                      num_features=55)

    #Abholung der Gewichtungen
    exp_weight = return_weights(exp)
    weights.append(exp_weight)

#Erstellung DataFrames mit den Gewichtungen von LIME und Darstellung
lime_weights = pd.DataFrame(data=weights, columns=X_train.columns)
lime_weights.head(100)

#Balken Diagramm Feature Importance
abs_mean = lime_weights.abs().mean(axis=0)
abs_mean = pd.DataFrame(data={'feature':abs_mean.index,
                             'abs_mean':abs_mean},)
abs_mean = abs_mean.sort_values('abs_mean', ascending=0)[:10]

#Darstellung des absoluten Durchschnitts
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(25,25),)

y_ticks = range(len(abs_mean))
y_labels = abs_mean.feature
plt.barh(y=y_ticks, width=abs_mean.abs_mean)

plt.yticks(ticks=y_ticks, labels=y_labels, size= 20)
plt.title('')
plt.ylabel('')
plt.xlabel('Mean |Weight|', size=20)

#Balken Diagramm Feature Importance
abs_mean = lime_weights.abs().mean(axis=0)
abs_mean = pd.DataFrame(data={'feature':abs_mean.index,
                             'abs_mean':abs_mean},)
abs_mean = abs_mean.sort_values('abs_mean', ascending=0)
print(abs_mean[:10])
## md
## InterpretML EBM

##%
import interpret
from interpret import show
## md
### Lokal
##%

```

```

#Lokale Erklärbarkeit
ebm_local = ebm_model.explain_local(X_train_processed[:5],
y_train[:5])
show(ebm_local)
# %% md
### Global
#%%
# Globale Erklärbarkeit
ebm_global = ebm_model.explain_global()
show(ebm_global)
# %% md
### Vergleich der XGB und EBM InterpretML Ergebnisse
#%%
# Vergleich Globaler und Lokaler XAI Methoden anhand InterpretML-Dashboard
show([ebm_global,ebm_local,XGB_shap_local,XGB_Global])
# %% md

```

3.Korrelationsanalyse

ANOVA Ergebnisse	
Variable	p-Wert
MS Zoning	1.1004990281237354e-74
Street	0.001267537207464486
Alley	6.239449381200554e-15
Lot Shape	1.0678631356077208e-60
Land Contour	3.7436782290891496e-28
Utilities	0.21290794617934414
Lot Config	1.0412226056008607e-12
Land Slope	0.0010119492021355193
Neighborhood	0.00000013964227889
Condition 1	2.4914066953007e-25
Condition 2	5.430319153269248e-13
Bldg Type	2.4759233004406355e-21
House Style	3.028709975533312e-47
Roof Style	2.3710441347061137e-49
Roof Matl	4.8475091191785076e-08
Exterior 1st	3.3856088064284634e-106
Exterior 2nd	4.633597591729461e-105
Mas Vnr Type	4.056978182252442e-133
Exter Qual	2.9584562484548742 e-18
Exter Cond	1.611921363209932e-16
Foundation	1.2671240384616744e-205
Bsmt Qual	1.85839456432368886e-25
Bsmt Cond	2.0612829133309308e-31
Bsmt Exposure	1.1888637413284694e-124
BsmtFin Type 1	5.579684221083531e-160
BsmtFin Type 2	1.143398969379218e-17
Heating	2.8022946827171354e-05

Heating QC	4.917676383038957e-152
Central Air	4.2474675635421973e-48
Electrical	3.873698380500518e-37
Kitchen Qual	2.345683835458627e-67
Functional	6.2702858486829475e-09
Fireplace Qu	2.543385614348627e-229
Garage Type	1.6659846883164266e-177
Garage Finish	5.458333772485493e-233
Garage Qual	3.97343740614795e-57
Garage Cond	2.2499541377524902e-52
Paved Drive	2.0346329422075114e-51
Pool QC	1.2389232466266184e-11
Fence	1.1362018580710855e-23
Misc Feature	0.01540320667069519
Sale Type	3.418425304681396e-89
Sale Condition	2.3120857219032514e-91

Pearsons- Korrelationskoeffizient	
Variable	Korrelationskoeffizient
Overall Qual	0,80
TotalSF	0,79
Gr Liv Area	0,71
Garage Cars	0,65
Garage Area	0,64
Total Bsmt SF	0,63
Year Built	0,62
Bath	0,58
Full Bath	0,56
Year Remod/Add	0,55
Garage Yr Blt	0,53
Mas Vnr Area	0,51
TotRms Abvgrd	0,50
Fireplaces	0,47
BsmtFin SF 1	0,43
Lot Frontage	0,34
Wood Deck SF	0,33
Open Porch SF	0,31
Half Bath	0,29
Bsmt Full Bath	0,28
2nd Flr SF	0,27
Lot Area	0,27

Spearmans Rangkorrelationskoeffizient	
TotalSF	0,81
Overall Qual	0,81

Gr Liv Area	0,72
Garage Cars	0,70
Year Built	0,68
Garage Area	0,66
Bath	0,64
Full Bath	0,63
Garage Yr Blt	0,61
Total Bsmt SF	0,61
Year Remod/Add	0,60
1st Flr SF	0,58
Fireplaces	0,53
TotRms Abvgrd	0,50
Open Porch SF	0,48
Mas Vnr Area	0,44
Lot Area	0,43
Lot Frontage	0,37
BsmtFin SF 1	0,43
Wood Deck SF	0,36
Half Bath	0,33
Bsmt Fin SF 1	0,33
Bsmt Full Bath	0,26
2nd Flr SF	0,25
Bedroom AbvGr	0,20

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass die vorliegende Arbeit von mir selbstständig und ohne unerlaubte Hilfe angefertigt worden ist, insbesondere dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen sind, durch Zitate als solche gekennzeichnet habe. Weiterhin erkläre ich, dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde/Prüfungsstelle vorgelegen hat. Ich erkläre mich damit einverstanden, dass die Arbeit der Öffentlichkeit zugänglich gemacht wird. Ich erkläre mich damit einverstanden, dass die Digitalversion dieser Arbeit zwecks Plagiatsprüfung auf die Server externer Anbieter hochgeladen werden darf. Die Plagiatsprüfung stellt keine Zurverfügungstellung für die Öffentlichkeit dar.

Düsseldorf, 04.03.2023

(Ort, Datum)



(Eigenhändige Unterschrift)