ifes Schriftenreihe

Band 25

Bianca Krol (Hrsg.)

Big Data basierte Analyse des Einflusses traditioneller und neuartiger Faktoren auf Mietpreise in Düsseldorf

~

Dominic Hernes, Frank Lehrbass, Kevin Maucy

FOM Hochschule
ifes
Institut für Empirie & Statistik
der FOM Hochschule
für Oekonomie & Management

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

**Akademie
Verlags- und Druck-
Gesellschaft mbH**

MA Akademie Verlags- und Druck-Gesellschaft mbH
Leimkugelstraße 6, 45141 Essen
info@mav-verlag.de

Dominic Hernes, Frank Lehrbass, Kevin Maucy

Big Data basierte Analyse des Einflusses traditioneller und neuartiger Faktoren auf Mietpreise in Düsseldorf

ifes Institut für Empirie & Statistik
der FOM Hochschule für Oekonomie & Management

ifes Schriftenreihe
Band 25, 2021

ISBN (Print) 978-3-89275-425-1
ISBN (eBook) 978-3-89275-426-8
ISSN (Print) 2191-3366
ISSN (eBook) 2569-5355

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Immer noch wird die Immobilienweisheit „Lage, Lage, Lage“ behauptet (z. B. https://www.immoverkauf24.de/immobilienverkauf/immobilienverkauf-a-z/lage-lage-lage/). Diese Behauptung wird für den Standort Düsseldorf in der vorliegenden Studie wissenschaftlich untersucht.

Dabei zeigt sich, dass es besser „Lebensqualität“ statt „Lage“ heißen sollte. Mit Hilfe von Yelp, Tripadvisor oder Google lässt sich diese Qualität messen. Wir haben dies objektscharf getan.

Die dadurch ermöglichten Vorhersagen brauchen den Vergleich mit verfügbaren Benchmarks nicht zu scheuen.

1.1 Motivation

Wenn es um die Erklärung der Höhe einer Miete oder synonym den Mietpreis oder Mietzins geht, wird unter Praktikern oft die Lage des Mietobjekts herangezogen. Demnach müssten die Koordinaten eines Objekts oder der Stadtteil Faktoren von überragender Bedeutung sein. Dies ist eine Hypothese mit zu klärendem Wahrheitsgehalt.

Die vorliegende empirische Studie untersucht deshalb alle verfügbaren Faktoren auf ihren Einfluss auf den Mietpreis pro Quadratmeter kalt. Gesicherte Erkenntnisse über diese Wirkungszusammenhänge sind von großer Bedeutung für diverse Marktteilnehmer. Wir wollen nur einen Anwendungsbereich exemplarisch hervorheben: Bei der Bewertung von Immobilien spielt im Rahmen des Ertragswertverfahrens der Jahresreinertrag oder synonym Jahresrohertrag eine zentrale Rolle. Dieser benötigt die Kaltmiete als Ausgangsgrösse (Vgl. Brauer, 2001, S. 396). Investoren in Wohnimmobilien könnten von daher ein Interesse an den Faktoren haben, die wirklich relevant sind.

Die neuartigen Faktoren stammen aus Bewertungsportalen und Kartenwerkzeugen im Internet. Da diese neuartigen Variablen aus verschiedenen Datenquellen stammen, nicht zwingend objektive Messwerte darstellen, schnell entstehen und insgesamt eine große Datenmenge darstellen, kann man die betrachteten Faktoren als Big Data bezeichnen.

Den Einfluss dieser neuartigen Variablen, welche auch als nicht-traditionelle Kennwerte bezeichnet werden, wollen wir in dieser Arbeit untersuchen.

1.2 Problemstellung

Um den Einfluss von nicht-traditionellen Faktoren auf die Kaltmiete von Mietobjekten in Düsseldorf zu analysieren, bedarf es eines geeigneten Datensatzes, welcher eine umfassende Menge an Informationen bereitstellt, um signifikante Erkenntnisse gewinnen zu können. Aufgrund der Tatsache, dass ein solcher Datensatz für die Stadt Düsseldorf nach bestem Wissen der Autoren noch nicht existiert, besteht die Herausforderung daher zunächst darin, einen Datensatz mit einer hinreichenden Güte an nicht-traditionellen Daten zu finden bzw. zu erstellen. Zudem stellen wir uns der Herausforderung, eine Benchmark, in Gestalt des offiziellen Mietrichtwerts, zu schlagen.

1.3 Zielsetzung

Das Ziel der hier vorliegenden Studie besteht darin, den Einfluss von traditionellen sowie nicht-traditionellen Daten auf die Kaltmieten (in €/m^2) von Mietobjekten in Düsseldorf zu analysieren. Hierzu werden in den folgenden Kapiteln zunächst die für die Datenbereinigung notwendigen Schritte beschrieben und daraufhin werden die bereinigten Werte mittels explorativer Datenanalyse näher betrachtet. Die Arbeit mündet schließlich in der Erstellung dreier Regressionsmodelle und einem Vergleich der Ergebnisse dieser drei Modelle. Im letzten Kapitel werden die geschätzten/trainierten Modelle schließlich auf eine vorab definierte und vom Training ferngehaltene Stichprobe, bestehend aus fünf Mietobjekten, angewendet. Aus all dem werden schließlich Erkenntnisse über den Einfluss der angesprochenen nicht-traditionellen Daten auf die Zielvariable abgeleitet und diskutiert. Diese Ergebnisdiskussion wird auch auf den Vergleich zur Benchmark in Form des Mietrichtwertes für die Stadt Düsseldorf eingehen.

2 Der Immobilienmarkt in Düsseldorf

Düsseldorf ist ein dynamischer Immobilienstandort mit vielen Facetten. Dies wird nachfolgend verdeutlicht. Im Sinne einer Mensch-Maschine Kooperation werden die Informationen, die durch Nutzer bestimmter Internetseiten wie Yelp, Tripadvisor oder Google generiert werden, mit einbezogen.

2.1 Aktuelle Entwicklungen

Die Landeshauptstadt Düsseldorf mit ihren 617.000 Einwohnern gehört zur zweitgrößten Stadt in Nordrhein-Westfalen neben der Stadt Köln mit über einer Million Einwohnern. Zudem gehört Düsseldorf neben Münster und Köln zu den Top 3 Städten, welche hohe steigende Einwohner- und Haushaltszahlen aufweisen. Das Statistische Landesamt NRW geht von einem Bevölkerungszuwachs von 9,4 Prozent bis zum Jahr 2030 aus und prognostiziert eine Steigerung von 12,7 Prozent bis 2040. Dieser Trend spiegelt sich auch in den Angebotsmieten wider. Hier verzeichnet Düsseldorf nach Dortmund und Bielefeld den dritthöchsten Anstieg bei den Angebotsmieten mit 4,8 Prozent auf durchschnittlich 10,48 €/m², was dem zweiten Platz unter den Top-12-Städten in Nordrhein-Westfalen entspricht. Insbesondere im unteren Segment waren diese Anstiege zu spüren, was auf eine Angleichung an die hohen Angebotsmieten in den anderen Segmenten des Düsseldorfer Wohnungsmarktes hindeutet. (LEG & CBRE, 2019, pp. 18–19)

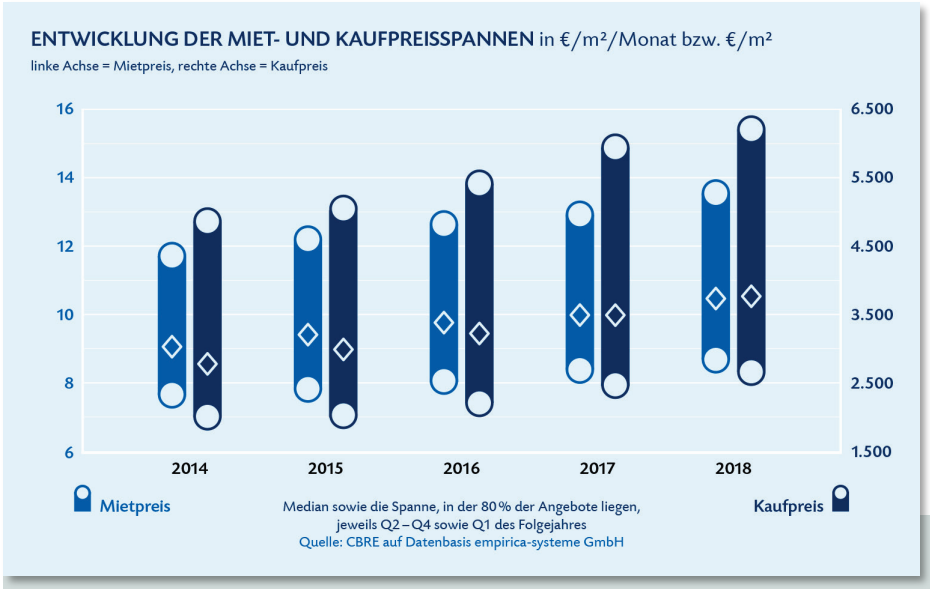

Abbildung 1: Entwicklung der Miet- und Kaufpreisspannen für Düsseldorf, Quelle: (LEG & CBRE, 2019, p. 18)

Diese Trendentwicklung kann man auch in den offiziellen Zahlen der Landeshauptstadt Düsseldorf wiederfinden, welche durch das Amt für Statistik und Wahlen der Stadt Düsseldorf herausgegeben worden sind. Hier verzeichnet das Amt für Statistik und Wahlen einen jährlichen Bevölkerungszuwachs von durchschnittlich 0,9 Prozent pro Jahr sowie einen positiven Haushaltszuwachs von durchschnittlich 0,73 Prozent pro Jahr in den Jahren 2014 bis 2019. Auf Basis dieser positiven Zuwächse stieg auch dementsprechend die durchschnittliche Miete gemäß Mietrichtwert von 7,25 €/m² im Jahr 2014 um insgesamt 0,75 €/m² auf 8 €/m² im Jahr 2018. (Landeshauptstadt Düsseldorf & Amt für Statistik und Wahlen, 2019, pp. 4–5)

Dem gegenüber stehen der steigende Wohnungsbedarf und die dazugehörigen Baufertigstellungen, um diesen Bedarf zu decken. Hier steht die Stadt Düsseldorf vor einer großen Herausforderung. So ermittelte das Stadtplanungsamt der Stadt Düsseldorf in Ihrem Stadtentwicklungskonzept 2020+, welches im Jahr 2011 erschien, dass aufgrund der positiven Entwicklung der Bevölkerungsprognose ein Wohnungsbedarf von 28.269 neuen Wohnungen bis zum Jahr 2020 entsteht. Dies entspricht einer notwendigen jährlichen Bauleistung von 2.200

Wohneinheiten bis zum Jahr 2010 und einer darauffolgenden Bauleistung bis zum Jahr 2020 von 1.700 Wohneinheiten (Landeshauptstadt Düsseldorf & Stadtplanungsamt, 2011, p. 55). Im Jahr 2018 hat die Stadt Düsseldorf erstmals seit 2011 den Bau von über 2.000 Wohneinheiten genehmigt (LEG & CBRE, 2019, p. 19).

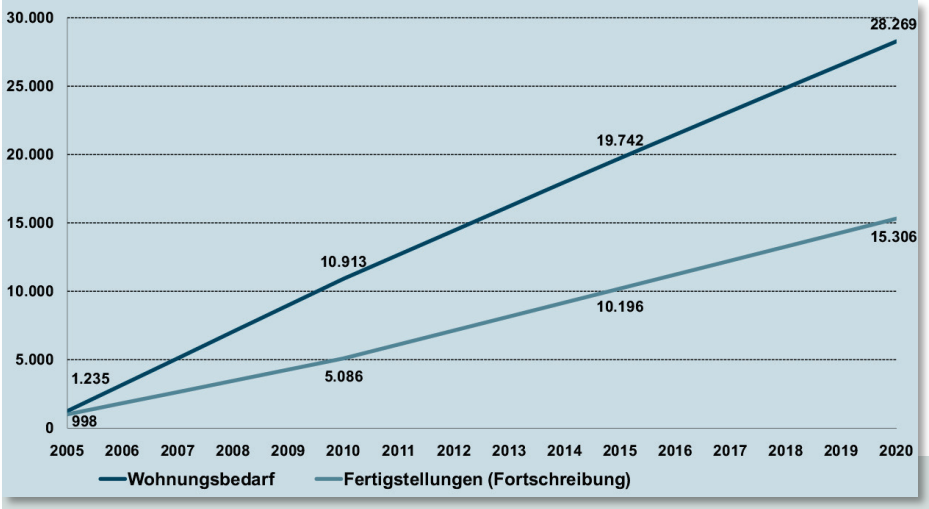

Abbildung 2: Wohnungsbedarf und Baufertigstellungen im Vergleich,
Quelle: (Landeshauptstadt Düsseldorf & Stadtplanungsamt, 2011, p. 55)

Die direkte Gegenüberstellung des Wohnungsbedarfs zu den erwartenden Baufertigstellungen zeigt die erhebliche Bedarfsunterdeckung bis zum Jahr 2020. Würde man den Trend von ca. 1.000 Fertigstellungen, welche im Durschnitt der letzten fünf Jahren ab dem Jahr 2007 beobachtet worden sind, weiter fortschreiben, ergibt dies ein Fehlen von 13.000 Wohnungen im Jahr 2020. So ein Defizit würde den Wohnungsmarkt weiter anspannen und sich zudem negativ auf die Stadtentwicklung auswirken. (Landeshauptstadt Düsseldorf & Stadtplanungsamt, 2011, p. 55–56)

Unter diesen Umständen muss man davon ausgehen, dass die Mieten in den nächsten Jahren weiter ansteigen werden. Zusätzlich kann man davon ausgehen, dass sich die Mietrichtwerte weiter den Angebotsmieten annähern werden.

2.2 Traditionelle und nicht-traditionelle Kennwerte

Die Trennung von nicht-traditionellen und traditionellen Werten erfolgt in Anlehnung an den Report „Getting ahead of the market: How big data is transforming real estate“ der Unternehmensberatungsgesellschaft McKinsey & Company (Asaftei et al., 2018) (im Folgenden: McKinsey), in welchem die Auswirkungen von nicht-traditionellen Kennwerten auf die Prognosefähigkeit von Immobilienwerten anhand des Wohnungsmarktes von Boston untersucht werden.

So hat McKinsey über die Makler- und Informationswebseite Zillow (www.zillow.com) für den Wohnungsmarkt in Boston festgestellt, dass die Werte von Häusern, welche sich innerhalb eines Umkreises einer Viertelmeile von einem Starbucks befinden, in dem Zeitraum von 1997 bis 2014 um 171 Prozent gestiegen sind, 45 Prozent mehr als alle Häuser in der Stadt Boston. Zudem hat man festgestellt, dass Wohngebäude, welche innerhalb eines Umkreises von einer Meile Lebensmittelgeschäfte wie Whole Foods oder Trader Joe’s haben, mehr im Wert zugelegt haben als andere. (Asaftei et al., 2018, p. 2)

Den relativen Effekt, die Aufteilung sowie die Inhalte von nicht-traditionellen sowie traditionellen Kennwerten bei der Prognose wird durch die Abbildung 3 dargestellt.

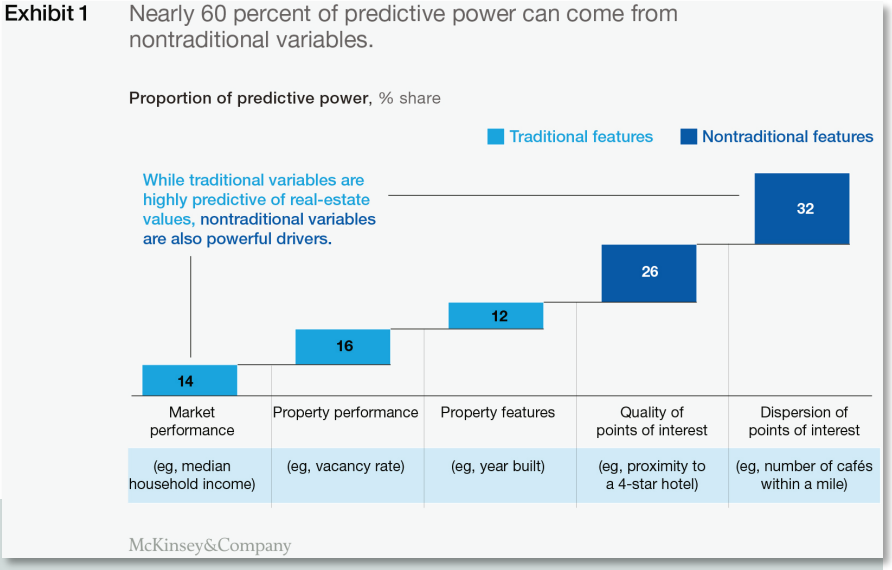

Abbildung 3: Einfluss von nicht-traditionellen und traditionellen Kennwerten, Quelle: (Asaftei et al., 2018 Exhibit 1)

Die Abbildung 3 zeigt uns das relative Verhältnis der verschiedenen Einflüsse sowie die verschiedenen Arten von Kennwerten in Bezug auf deren Vorhersagekraft. Der Anteil der traditionellen Kennwerte ist aufgeteilt in die drei Kategorien:

- Marktentwicklungen
  Darunter fallen Werte, wie bspw. die Stadtquartiere, die demografische Entwicklung oder auch die sozialen Gegebenheiten.
- Entwicklung der Immobilie
  Diese Kategorie bezieht sich auf die Verkehrswerte der Immobilie und deren Entwicklungen.
- Daten der Immobilien
  Diese Kategorie beinhaltet Daten zur Immobilie, wie das Baujahr, die Art der Heizung etc.

In der Arbeit von McKinsey finden hauptsächlich die Kategorien der Marktentwicklung sowie der Daten der Immobilien Anwendung. Dieser Block der traditionellen Kernwerte generiert 42 Prozent der Vorhersagekraft. In unserer Untersuchung wird der Block der traditionellen Kennwerte durch den Immoscout Datensatz abgedeckt, welcher im Kapitel 3.2.1 näher betrachtet wird. Auf der anderen Seite dieser Betrachtung steht der Block der nicht-traditionellen Kennwerte. Dieser Block wird unterteilt in folgende zwei Kategorien:

- Qualität der interessanten Orte (engl. Points of interest, kurz POI)
  Diese Kategorie beschreibt die Objekte anhand ihrer Bewertungen, welche durch die Nutzer der jeweiligen Internetseiten vergeben worden sind.
- Streuung der interessanten Orte
  Diese Kategorie fasst die Anzahl sowie die Entfernung der interessanten Orte innerhalb eines gegebenen Radius zusammen.

Die Generierung solcher Kennwerte erfolgt durch Nutzer bestimmter Internetseiten. Seiten wie Yelp, Tripadvisor oder Google steuern solche Daten bei. Für unseren Datensatz haben wir die Google-Daten genutzt, um nicht-traditionelle Kennwerte zu sammeln. Die Vorgehensweise wird im Kapitel 3.2.2 beschrieben. Gemäß der Abbildung 3 hat der Block der nicht-traditionellen Kennwerte in der Arbeit von McKinsey einen relativen Einfluss von 58 Prozent. Demnach sind die

nicht-traditionellen Kennwerte der größere Treiber für die Prognose von Mietwerten. Ob dies auch für Düsseldorf so ist, wollen wir mit unserer Studie analysieren.

3 Datenlage und -aufbereitung

Die Informationen, die durch Nutzer bestimmter Internetseiten wie Yelp, Tripadvisor oder Google generiert werden, weisen die typischen Merkmale von Big Data auf: Die Datenquellen sind heterogen und damit die Formate, die Datengenerierung erfolgt spontan, schnell und selten qualitätsgesichert.

Für die modellgestützte Analyse werden strukturierte Daten benötigt. Der Weg dorthin wird nachfolgend beschrieben.

3.1 Entwicklungsumgebung

Die Wahl unserer Entwicklungsumgebung fiel auf Anaconda (www.anaconda.com), eine Open-Source Umgebung, welche sich auf das Themengebiet Data Science spezialisiert hat und uns mit der Programmiersprache Python in der Version 3.8.5 einen einfachen Zugang zu den benötigten Bibliotheken für die Datenaufbereitung, Datenreinigung und der Erstellung von Machine Learning Modellen bereitstellt. Im Folgenden werden diejenigen Python Bibliotheken, welche in unserer Studie genutzt worden sind und wesentlichen Anteil hatten, kurz beschrieben.

NumPy

Das NumPy-Paket, welches das NumPy-Array sowie eine Reihe von begleitenden mathematischen Funktionen umfasst, ist in der Wissenschaft, in nationalen Laboren und in der Industrie weit verbreitet. Ein NumPy-Array ist ein mehrdimensionales Array in der Form MxN, welches sich über bis zu zweiunddreißig Dimensionen erstrecken und numerische sowie boolesche Werte aller Art speichern kann, wie zum Beispiel Gleitkomma- oder komplexe Zahlen sowie auch Datumswerte. Dazu bietet uns diese Bibliothek die Möglichkeit über simple Befehle einfache bis komplexe mathematische Operationen auf diesen Arrays auszuführen. (Van Der Walt et al., 2011, p. 22)

Pandas

Pandas bietet mit der Aufbereitung von Daten in DataFrames und den dazugehörigen Funktionen, wie zum Beispiel einer hierarchischen Indexierung oder der automatischen Datenausrichtung, eine enorme interaktive Ergonomie, beim Umgang mit diesen. Die Pandas-Bibliothek, die seit 2008 entwickelt wird, soll die Lücke in der Fülle der verfügbaren Datenanalysewerkzeuge zwischen Py-

thon, einer allgemeinen System- und Wissenschaftssprache, und den zahlreichen domänenspezifischen statistischen Berechnungsplattformen und Datenbanksprachen schließen. (McKinney, 2011, p. 1)

Heute ist Pandas in Verbindung mit NumPy im Bereich des Data Science ein essenzielles Werkzeug zur Bearbeitung von großen Datensätzen.

scikit-learn

Als Antwort auf den wachsenden Bedarf an statistischer Datenanalyse, welche durch die Bereiche Data Analytics sowie Data Science weiter forciert wird, bietet scikit-learn eine große Sammlung hochmoderner Implementierungen von Machine Learning Algorithmen für die Programmiersprache Python. Das Ziel dieser Bibliothek ist es diese Algorithmen näher an Gruppen zu bringen, welche nicht der Computerwissenschaften zugehörig sind, wie zum Beispiel der Physik oder Biologie. Als Basis nutzt scikit-learn die oben beschriebene NumPy Bibliothek. (Pedregosa et al., 2011, p. 1)

googlemaps

Die googlemaps Python-Bibliothek[1] bietet über standardisierte Funktionen Zugriff auf die Google Maps Web Services. Hierin enthalten sind unter anderem die für diese wissenschaftliche Arbeit notwendigen Services „Geocoding" sowie „Places". Der Geocoding Service erlaubt es, Straßennamen oder Ortsbezeichnungen in geografische Koordinaten, bestehend aus Längen- und Breitengraden, zu kodieren. Dies ist notwendig, um im Anschluss mathematische bzw. geografische Aggregationsfunktionen auf die Daten anzuwenden. Der Places Service erlaubt es hingegen Anfragen zu bestimmten geokodierten Koordinaten zu stellen und alle in einem vorher definierten Umkreis befindlichen POI als Ergebnis zu erhalten. Hiermit können die nicht-traditionellen Daten gesammelt werden.

GeoPandas

Die GeoPandas-Bibliothek[2] erweitert die Pandas-Bibliothek um weitere Funktionen, welche beim Umgang mit geografischen Daten unerlässlich sind. Hierzu integriert GeoPandas die auf geografische Methoden spezialisierte shapely-Bibliothek und ergänzt diese um weitere Funktionen, welche die Anwendung von shapely-Methoden auf Pandas DataFrames ermöglichen. In dieser Arbeit wird

1 Siehe https://github.com/googlemaps/google-maps-services-python.
2 Siehe https://geopandas.org/.

GeoPandas im Wesentlichen zur Extraktion von nicht-traditionellen Daten aus geografischen Eigenschaften der in der Nähe eines jeweiligen Mietobjektes befindlichen POI genutzt.

3.2 Datensätze

Die Datenquellen sind divers und werden nachfolgend skizziert.

3.2.1 Der Immoscout Datensatz

Die Basis für die traditionellen Kennwerte beinhaltet Inserate von Mietimmobilien, welche durch die Immobilienseite Immoscout24.de angeboten worden sind. Bereitgestellt wird dieser Datensatz durch die Webseite Kaggle (www.kaggle.com)[3]. Kaggle ist eine Internetseite, welche Wettbewerbe für Data Science Herausforderungen anbietet.

Mithilfe einer Scraping-Methode, also einer Methode womit die Informationen direkt von der Internetseite extrahiert werden, wurden die Daten in den Zeiträumen 22.09.2018, 10.05.2019 sowie 08.10.2019 zusammengetragen und in einer kommaseparierten Datei aufgearbeitet. Diese Datei beinhaltet 268.850 Datensätze mit jeweils neunundvierzig Spalten und erstreckt sich inhaltlich bundesweit. Den größten Anteil der Spalten machen dreiunddreißig Stück aus, welche als Zeichenketten (englisch: Strings) formatiert sind. Zehn Stück sind numerisch formatiert und die letzten sechs Spalten haben boolesche Werte, welche die traditionellen Kennwerte repräsentieren.

Ein hohes Maß an Business Understanding war zur Identifizierung von Ausreißern und zur Erkennung von Unregelmäßigkeiten notwendig, da die Inserate in der Regel durch die Vermieter direkt eingestellt werden und somit ein hohes Fehlerpotential vorhanden ist, bspw. durch Eintragen von Jahreswerten statt von Monatswerten.

3.2.2 Raumbezogene Daten von Google

Das Application Programming Interface (API) für die Google Places Daten (im Folgenden: Google Places API) ist eine Schnittstelle mittels welcher Abfragen an die Google Server gestellt werden können, um geografische Informationen rund

[3] https://www.kaggle.com/corrieaar/apartment-rental-offers-in-germany

um eine Geolokation, bspw. eine Adresse bestehend aus Straße und Hausnummer, zu erhalten. Dieser Dienst wird von Google angeboten und beinhaltet sämtliche Informationen zu einem jeweiligen Ort, welche bspw. ebenfalls in der Navigations-App Google Maps verwendet werden.

Die Daten der Google Places API kommen bereits in zahlreichen anderen wissenschaftlichen Arbeiten zum Einsatz. So werden diese bspw. dazu eingesetzt um Nutzern bei der Planung von Reisen bessere Routen, angepasst an die verschiedenen POI, vorzuschlagen (Vgl. Raji et al., 2020). Des Weiteren werden die Google Daten in Verbindung mit Satellitenbildern von Nachbarschaften gebracht, um so die Auswirkungen von raumbezogenen Daten aus der Nachbarschaft auf Hauspreise zu erforschen (Vgl. Bency et al., 2017). Vor allem letztgenannte Publikation kann beispielhaft einen signifikanten Einfluss von raumbezogenen Daten auf die Preise von Häusern identifizieren. Andere Publikationen (Vgl. bspw. Petkov, 2020) können indes keinen wesentlichen Genauigkeitsgewinn durch die Hinzunahme von raumbezogenen Daten identifizieren.

Es ist an dieser Stelle vor allem auch zu erwähnen, dass es neben der Google Places API noch andere Möglichkeiten gibt, um raumbezogene Daten und Informationen zu erlangen. Als Beispiel seien an dieser Stelle die Daten der Open Street Map (OSM) genannt, welche ebenfalls umfangreiche und aktuelle Daten zu POI beinhalten. Die Daten von OSM werden dabei, im Gegensatz zu den Google Daten, von vielen unterschiedlichen Mitwirkenden gepflegt. Praktisch jeder kann eigene Änderungen in OSM vorschlagen bzw. eintragen, was jedoch vor allem Fragen bzgl. der Qualität und Aktualität der Daten aufkommen lässt.

Der Vorteil der OSM Daten liegt vor allem darin, dass diese Daten jedem Benutzer kostenlos bspw. durch die Geofabrik GmbH[4] zur Verfügung gestellt werden, wohingegen für die Datenabfrage mittels Google Places API je nach Umfang bzw. Menge schnell mehrere tausend Euro zusammenkommen können.

Für die vorliegende Arbeit wird nach gründlicher Abwägung aller Alternativen schließlich die Google Places API verwendet, um raumbezogene Daten in Zusammenhang mit den Immoscout Daten zu bringen und hieraus Erkenntnisgewinne zu generieren. Diese API wurde schließlich zur Geokodierung von Adressen einzelner Mietobjekte (Kapitel 3.2.2.1) sowie zur Abfrage von in der Nähe

[4] Siehe https://download.geofabrik.de/€pe/germany/nordrhein-westfalen/duesseldorf-regbez.html.

der jeweiligen Mietobjekte befindlichen POI (Kapitel 3.2.2.2) genutzt. Diese beiden Vorgehensweisen sollen nun beschrieben werden.

3.2.2.1 Geokodierung von Adressen

Für die Analyse von raumbezogenen Daten ist es notwendig zu wissen, welche POI sich in der Nähe eines jeweiligen Mietobjektes befinden, um so schließlich Auswertungen wie bspw. die durchschnittliche Distanz zu Restaurants zu berechnen. Hierzu ist es notwendig, die individuellen Straßen- und Ortsnamen für jedes Mietobjekt als einzelnen Punkt gemessen in geografischen Koordinaten bestehend aus Längen- und Breitengrad darzustellen. Mittels der Geokodierung von Adressen kann genau dies bewerkstelligt werden.

Hierzu müssen bei einem entsprechenden Service, in diesem Fall bei der Google Geocoding API,[5] Abfragen bestehend aus dem Straßennamen, der Hausnummer sowie der Postleitzahl gestellt werden. Aus dem Immoscout Datensatz wurden für die notwendigen Informationen die Spalten streetplain, housenumber sowie geo_plz zusammengefügt. Als Ergebnis einer Google Geocoding API-Abfrage erhält man die entsprechenden Koordinaten der jeweiligen Adresse in Längen- und Breitengrad zurück. Mithilfe dieses Verfahrens konnten in Summe insgesamt 191.673 Adressen geokodiert werden.

Einige Adressen ließen sich auf diesem Wege aufgrund von teilweise fehlenden Informationen bei den Straßennamen oder Hausnummern oder auch aufgrund von teilweise irreführenden Ergebnissen nicht korrekt kodieren. Diejenigen Mietobjekte, die erfolgreich geokodiert werden konnten, wurden zur späteren Nutzung zwischengespeichert[6]. Somit müssen bei einer erneuten Ausführung des Codes nicht sämtliche Mietobjekte erneut geokodiert werden, was Ressourcen, Zeit und Kosten spart. Mittels der Pandas Funktion read_pickle kann diese Datei anschließend direkt in ein Pandas DataFrame importiert werden.

Mit den so geokodierten Adressen kann der nächste Schritt im Zusammenhang mit der Google Places API durchgeführt werden, welcher im folgenden Kapitel erläutert wird.

[5] Siehe hierzu: https://developers.google.com/maps/documentation/geocoding/overview.
[6] Technisch in der pickle-Datei df_geo.pkl.

3.2.2.2 Abfrage von POI in der Nähe

Nachdem die meisten Adressen korrekt einem Längen- sowie Breitengrad zugeordnet werden konnten, wurden für diese Mietobjekte die in der Nähe befindlichen POI abgefragt. Hierzu wurde eine Datenbank aus sämtlichen Abfragen der Google Places API erstellt. Eine Abfrage beinhaltete hierbei die jeweiligen Koordinaten des Mietobjektes, die Angabe eines Radius von 1.000 Metern in welchem die Google Places API nach entsprechenden POI suchen sollte sowie einen Suchbegriff nach welchem die Ergebnisse gefiltert bzw. ausgewählt werden sollten. Als Suchbegriffe wurden hierbei jeweils die folgenden drei Begriffe übergeben: Restaurant, Bankautomat, Haltestelle.

Dementsprechend wurden sämtliche Restaurants, Bankautomaten sowie Haltestellen in einem Umkreis von 1.000 Metern von einem jeweiligen Mietobjekt abgefragt. In dicht besiedelten Gebieten kann es dazu kommen, dass die Ergebnisse Duplikate aufweisen, wohingegen eher ländliche Gebiete auch keine Ergebnisse aufweisen können. Eventuell vorkommende Duplikate wurden mittels der für jeden POI von Google eindeutig vergebenen place_id identifiziert und schließlich aus dem Datensatz entfernt, sodass lediglich ein Vorkommen im Datensatz vorhanden ist.

Zu erwähnen ist hierbei noch, dass aufgrund des beschränkten Budgets nicht alle Mietobjekte abgefragt werden konnten, weshalb es für einige Gebiete mit nur wenigen Mietobjekten im Immoscout Datensatz keine Google Places Daten gibt. Hieraus ergibt sich, dass bei der Auswertung der raumbezogenen Daten Nullwerte vorhanden sind. Die Lösung dieses Problems wird in Kapitel 3.3.2 beschrieben.

Abschließend lässt sich jedoch festhalten, dass insgesamt 157.252 POI mittels Google Places API-Abfragen für ganz Deutschland gesammelt werden konnten. Eine geografische Abbildung der für diese Arbeit relevanten POI in Düsseldorf kann der Abbildung 20 entnommen werden.

3.2.3 Geodaten Stadtteile und -bezirke in Düsseldorf

Als weiterer geografischer Datensatz wurden die Stadtteil- und Stadtbezirksgrenzen von Düsseldorf verwendet. Diese Daten sind auf der Open-Data Plattform der Landeshauptstadt Düsseldorf[7] öffentlich zugänglich und können dort

[7] Siehe https://opendata.duesseldorf.de/dataset/stadtteilgrenzen-düsseldorf.

kostenfrei bezogen werden. Enthalten sind hier die Polygone der einzelnen Stadtteile bzw. -bezirke zum Zeitpunkt 31. Dezember 2017, womit sich geografische Auswertungen und Zusammenhänge in den Daten visuell darstellen lassen. Im Kapitel 3.4 werden diese Daten genutzt um entsprechende Verteilungen sowie Beziehungen zwischen den Immoscout Daten und den nicht-traditionellen Daten für die Stadt Düsseldorf geografisch und explorativ zu analysieren.

3.2.4 Benchmark

Zum Vergleich unserer €/m² Prognosen für Mietwohnungen in Düsseldorf nehmen wir die herkömmliche Mietrichtwert-Tabelle des Mietervereins Düsseldorf. Diese Tabelle wird in regelmäßigen Intervallen von ein bis drei Jahren durch den Mieterverein Düsseldorf e.V. sowie der Haus & Grund GmbH in Zusammenarbeit fortgeschrieben und ggf. angepasst (Mieterverein Düsseldorf e.V., 2019, p. 6).

Diese dient als eine Orientierungshilfe für Wohnungssuchende nach § 588 ff. BGB. Damit soll eine Möglichkeit geboten werden, im Rahmen ortsüblicher Mieten eigenverantwortlich die Miethöhe für nicht preisgebundene Wohnungen zu vereinbaren. Die Tabellenwerte sind eine Grundlage für die Berechnung der Nettomieten, sprich Kaltmieten, bei Überlassung leeren Wohnraumes einschließlich aller Kosten außer den Betriebskosten nach §§ 1 und 2 BetrKV. (Mieterverein Düsseldorf e.V., 2019, p. 1)

Mietrichtwert-Tabelle Düsseldorf - Stand 1. Dezember 2019
- Werte inkl. Kabelanschluss und Isolierverglasung -

Baujahr	Wohnlage	Ausstattung und Beschaffenheit	
		„A“ ohne zentr. Beheizung mit Bad/Dusche oder mit zentr. Beheizung ohne Bad/Dusche in EURO/m²	„B“ mit zentraler Beheizung und mit Bad/Dusche in EURO/m²
bis 1948	einfache mittlere gute	4,95 - 5,85 5,85 - 6,85 6,85 - 7,95	6,25 - 7,75 6,80 - 9,40 8,25 - 10,05

Abbildung 4: Ausschnitt aus der Mietrichtwert-Tabelle 2019,
Quelle: (Mieterverein Düsseldorf e.V., 2019)

Als Ausgangslage zur Ermittlung eines Mietrichtwertes wird die aktuelle Mietrichtwert-Tabelle hinzugezogen, in welcher die Basis-Kaltmieten mit Bezug zum Baujahr, der Wohnlage sowie der Ausstattung und Beschaffenheit enthalten sind. Die Ausstattung und Beschaffenheit sind in die Kategorien A und B unterteilt. Diese unterscheiden sich in Art der Beheizung sowie ob Sanitärräume vorhanden sind. Die Abbildung 4 zeigt beispielhaft einen Ausschnitt der Mietrichtwert-Tabelle des Jahres 2019. Um einen konkreten Mietrichtwert zu eruieren, werden weitere Parameter verwendet, welche über Aufschläge Einflüsse auf die Basis-Kaltmiete haben. Zu solchen Parametern gehören zum Beispiel die Fragen nach einer umfangreichen Modernisierung oder dem Vorhandensein eines Aufzuges. Aber auch der Standort der Immobilie in einem bestimmten Stadtquartier spielt eine übergeordnete Rolle, welche mit einer Bandbreite von 0 Prozent - 10 Prozent auf die Basis-Kaltmiete aufgeschlagen wird (Mieterverein Düsseldorf e.V., 2019, p. 2).

Eine beispielhafte Berechnung eines Mietrichtwertes für eine 80m² große Immobilie aus dem Jahr 2010 mit mittlerer Lage in Düsseldorf Oberkassel inklusive Aufzug, zentraler Beheizung mit Bad/Dusche und ohne Modernisierung könnte wie folgt aussehen:

Bestandteil	Preiswerte und Aufschläge
Durchschnittliche Basis-Kaltmiete mit Baujahr 2010, mittlere Wohnlage, Ausstattung B	10,20 €/m²
Lage Düsseldorf Oberkassel	Aufschlag 10 Prozent
Aufzug vorhanden	+ 0.25 €/m²
Mietrichtwert	11,47 €/m²

Tabelle 1: Beispielrechnung eines Mietrichtwertes nach eigener Ermittlung

Für unser Modell haben wir die Mietrichtwert-Tabelle in einem Pandas DataFrame digitalisiert. Zudem haben wir diesen um diverse Spalten erweitert, um die möglichen Zuschläge abzubilden und direkt in die Basis-Kaltmieten einzuberechnen. Dies bietet uns die Möglichkeit mithilfe einfacher Befehle einen genauen Vergleichswert zu unserer Prognose aus dem DataFrame zu ziehen.

3.3 Datenbereinigung

Die Datenbereinigung, welches zum „Data Preparation" Schritt des Cross Industry Standard Process for Data Mining, kurz CRISP-DM, gehört, ist einer der zeitaufwändigsten Abschnitte im Aufbau eines Modells für eine computerunterstütze Analyse und Prognose. Auch in unserer Modellierung hat die Datenbereinigung neben der Datenbeschaffung der nicht-traditionellen Faktoren die meiste Zeit benötigt.

Die Datenbereinigung befasst sich primär mit der Erkennung und Beseitigung von Fehlern und Inkonsistenzen im Datensatz, um die Qualität der Daten zu verbessern. Datenqualitätsprobleme treten aufgrund von Rechtschreibfehlern bei der Dateneingabe, fehlenden Informationen oder anderen ungültigen Daten auf (Rahm & Do, 2000, p. 3). Solche Datenqualitätsprobleme haben Auswirkungen auf die Güte der Prognose unserer eingesetzten Modelle. In unserem Immoscout Datensatz trafen wir auf alle gängigen Datenqualitätsprobleme, welche in einem einzelnen Datensatz vorkommen können (Rahm & Do, 2000, p. 5), was auf die manuelle Eingabe durch die Ersteller der Inserate zurückzuführen ist. Zudem besaß der Datensatz alle möglichen Datentypen und benötigte ein

hohes Maß an Business Understanding, welches wir im Kapitel 3.2.1 näher beschrieben haben. Aus diesem Grund war die Datenbereinigung essenziell für unseren Modellbau.

3.3.1 Traditionelle Daten

Die herkömmliche Umsetzung der Datenbereinigung für die traditionellen Daten in unserem Datensatz von Immoscout erfolgte in drei Hauptschritten: Untersuchung von nicht vorhandenen Werten, Bestimmung der Ausreißer und die Festlegung sowie Durchführung von Imputationen, Eliminierung oder Interpolation – alle drei orientiert am Business Understanding.

Für die Spalten, welche einen booleschen Datentyp haben, ist kein spezielles Eingreifen notwendig, da diese problemlos von den eingesetzten Analysemodellen, in unserem Falle der multiplen linearen Regression, des Random Forest und des XGBoost Entscheidungsbaums, genutzt und ausgewertet werden können. Jedoch haben wir weitere kategoriale Spalten im String-Format, welche nicht einfach in boolesche Datentypen umgewandelt werden konnten, da diese mehr als zwei Ausprägungen aufweisen. Hierzu bedienten wir uns, nach der Imputation der nicht vorhanden Werte, einer speziellen Encoding Methode, des One-Hot-Encodings (Cerda et al., 2018), welches wir im Kapitel 3.5.1 näher beleuchten. Diese Methode erlaubt es uns kategoriale Daten so umzugestalten, dass diese durch die Modelle verarbeitet werden können. Die Inhalte in den Spalten mit String-Formaten können nicht einfach imputiert werden, weil die Immobilien beispielhaft in ihrer Ausstattung sehr unterschiedlich sind oder die Art der Heizung aufgrund des Angebotes nicht bestimmbar ist. Aus diesen Gründen haben wir die Null- und nicht vorhandenen Werte durch die Begrifflichkeit „nicht Verfügbar“ gekennzeichnet.

Im Vergleich zu den anderen Spalten, war die Bearbeitung der numerischen Spalten filigran. Hier gab es die Möglichkeiten für gezielte Imputation und der Eruierung von Ausreißern. Im ersten Schritt, bei der Reinigung der zu bearbeiteten Spalte, wurden die Null- und nicht vorhandenen Werte betrachtet. Bei dieser Betrachtung muss man primär Business Understanding einsetzen und gegebenenfalls weiter recherchieren, ob die vorhandenen Nullwerte valide sind und ob die nicht vorhandenen Werte als Nullwerte umgeformt oder doch imputiert werden können. Beispiele für numerischen Spalten mit validen Nullwerten waren die Anzahl der verfügbaren Parkplätze und die Heizkosten. Die Möglichkeit,

dass die Heizkosten Nullwerte aufweisen kann daran liegen, dass diese entweder in die Nebenkosten inkludiert worden sind oder dass diese über die Stromkosten finanziert werden, welche in der Regel direkt über den Mieter selbst getragen werden.

Für die Imputation der fehlenden Werte nutzten wir eine Methode mit der wir die Durchschnittswerte der Zielvariable basierend in Intervallen von 50m² Wohnfläche imputieren können. Das heißt, dass wir im ersten Schritt Klassen (englisch: Bins) mit einer festen Breite von jeweils 50m² Wohnfläche für die Reichweite, welche uns durch den Datensatz zu Verfügung stand, erzeugt haben. Dies waren in unserem Fall Wohnflächen zwischen 5m² und 600m². Danach werden den Bins, die jeweiligen Durchschnittswerte der Zielvariable zugeordnet, sodass wir einen Durchschnittswert der Zielvariable je Bin haben. Mithilfe dieser Bins können wir im letzten Schritt, die Nullwerte anhand der Wohnfläche realistischer abbilden und somit die Nullwerte imputieren. Mit dieser Methode war es nicht mehr notwendig Nullwerte zu eliminieren. Es konnten dadurch so viele Datensätze wie möglich erhalten werden.

Bei der Eruierung von Ausreißen nutzten wir eine Funktion, mit der wir die oberen und unteren Quantile von bis zu einem Prozent oder niedriger gebildet haben. Diese Quantile haben wir für jede einzelne Spalte mit Hilfe von weiteren Bezugsvariablen, wie bspw. der Wohnfläche oder den verschiedenen Kostenarten, intensiv betrachtet. Durch diese intensiven Betrachtungen konnten wir die Quantile genauer anpassen, um für den Datensatz die Grenzwerte den echten Gegebenheiten näher zu bringen. Zudem haben wir die Inhalte der Quantile genauer untersucht, um herauszufinden, inwieweit die Werte auf die fehlerhafte Eingabe der Nutzer zurückzuführen sind und somit als unbrauchbar gekennzeichnet werden konnten. Diese identifizierten Ausreißer werden eliminiert. Im Verlauf unsere Datenbereinigung waren das unter einem Prozent pro Spalte.

3.3.2 Nicht-traditionelle Daten

Die Datenbereinigung der nicht-traditionellen Daten baut direkt auf der Datenbereinigung der traditionellen Daten auf, da die Indizes der nach der Datenbereinigung verbliebenen Mietobjekte zur Zusammenfassung der entsprechenden geokodierten Koordinaten bestehend aus Längen- und Breitengraden genutzt werden. Nachdem die Längen- und Breitengrade dem bereinigten Immoscout

Datensatz mittels der pandas.join Funktion hinzugefügt wurden, werden sämtliche Mietobjekte aus dem Datensatz entfernt, die keine geokodierten Koordinaten, aufgrund der bereits geschilderten Umstände, erhalten haben.

Im nächsten Schritt wird das pandas DataFrame in ein GeoPandas GeoDataFrame konvertiert und die Projektion der geografischen Koordinaten von EPSG:4326 hin zu EPSG:32633[8] transformiert. Es handelt sich hierbei um Standardformate für geografische Projektionen von Koordinaten, welche von der namensgebenden European Petroleum Survey Group entwickelt wurden. (Vgl. Cain, 2013) Die Änderung der Projektion der Längen- und Breitengrade ist notwendig, um im Folgenden die Distanzen zwischen den Mietobjekten und den POI in Metern korrekt berechnen zu können. Einige Tests haben gezeigt, dass die Berechnung der Distanzen bei der Belassung der Projektion auf dem EPSG:4326 Standard nicht zu korrekten Ergebnissen geführt haben. Die mittels Google Places API bezogenen Daten liegen jedoch im EPSG:4326 Standard vor,[9] weshalb die besagte Konvertierung angewendet werden muss.

Den einzelnen Mietobjekten werden im nächsten Schritt die entsprechenden nicht-traditionellen Daten als weitere Spalten im Datensatz hinzugefügt, indem mittels eines Algorithmus für jedes Mietobjekt sämtliche POI in einem Radius von 1.000m aus der Datenbank entnommen und hierauf unterschiedliche Aggregationsmethoden angewendet werden. Eine Übersicht derjenigen Spalten, die so dem Datensatz hinzugefügt werden, kann Tabelle 9 im Anhang entnommen werden.

Da, wie bereits in Kapitel 3.2.2.2 beschrieben, für einige Mietobjekte keine POI mittels der Google Places API extrahiert werden konnten, ergeben sich nach der durchgeführten Extraktion der in Tabelle 9 betitelten Spalten noch Nullwerte, welche bereinigt werden mussten. Aufgrund der Tatsache, dass die extrahierten Daten einen geografischen Bezug zu den entsprechenden Koordinaten eines jeweiligen Mietobjektes haben, können allgemeine Imputationen bspw. über den Mittelwert des ganzen Datensatzes eventuell verfälschend sein. Eine präzisere Herangehensweise besteht daher darin, die Nullwerte eines jeweiligen Mietobjektes mit den Durchschnittswerten des jeweiligen Stadtteils zu imputieren. Sollten nach diesem Schritt weiterhin Nullwerte bestehen, können diese über die nächsthöhere Ebene der Stadtbezirke imputiert werden. Noch präziser wäre es,

[8] Siehe für die Spezifikationen der beiden erwähnten Projektionen: https://epsg.io/4326 und https://epsg.io/32633.

[9] Siehe: https://developers.google.com/maps/documentation/javascript/coordinates?hl=en.

die entsprechenden Mietobjekte in Nachbarschaften oder kleinere Siedlungen zu gruppieren. Aufgrund der gegebenen Datenlage ist diese Herangehensweise jedoch nicht möglich. In zukünftigen Arbeiten würde es sich jedoch anbieten, diesen Ansatz weiter zu verfolgen.

Die nun aufbereiteten, nicht-traditionellen Daten lassen sich anschließend im folgenden Kapitel zusammen mit den traditionellen Daten explorativ analysieren.

3.4 Explorative Datenanalyse

Zur explorativen Analyse der Daten werden im folgenden Abschnitt einige ausgewählte metrische sowie kategoriale Daten in Zusammenhang mit der Zielvariable der Kaltmiete in €/m² gebracht, um hieraus erste Erkenntnisse über die Beziehung zwischen den einzelnen Variablen zu gewinnen.

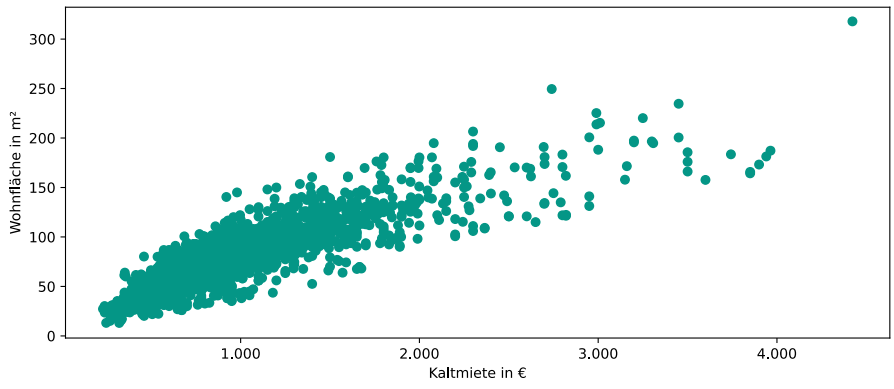

Abbildung 5: Kaltmiete in € im Verhältnis zur Wohnfläche in m²

Zunächst soll mit Abbildung 5 die lineare Beziehung zwischen den beiden Komponenten der Zielvariablen, nämlich der Kaltmiete und der Wohnfläche, dargestellt werden. Die Kaltmiete und die Wohnfläche stehen in einem starken linearen Verhältnis zueinander, was zunächst plausibel ist, da bei einer größeren Wohnfläche von einem höheren Preis für das Mietobjekt ausgegangen werden kann.

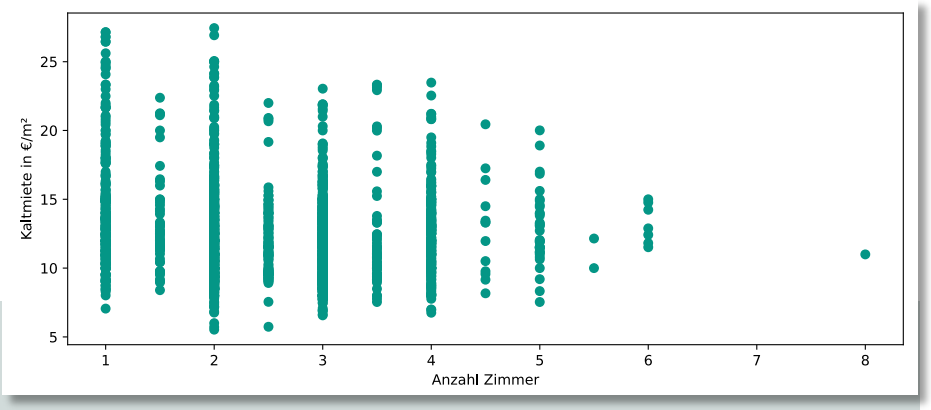

Abbildung 6: Kaltmiete in €/m² im Verhältnis zur Anzahl der Zimmer

Aus Abbildung 6 lässt sich augenscheinlich kein starker Zusammenhang zwischen der Anzahl der Zimmer und der Zielvariablen ausmachen. Was jedoch auffällt ist, dass bei steigender Anzahl der Zimmer die Kaltmiete in €/m² eine geringere Streuung aufweist, was im Wesentlichen damit zu begründen ist, dass mit steigender Anzahl an Zimmern weniger Mietobjekte mit einer hohen Anzahl an Zimmern im Datensatz vorhanden sind.

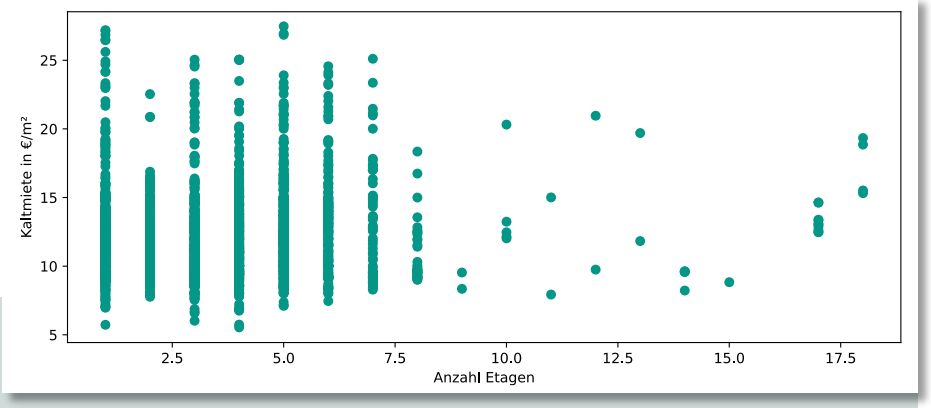

Abbildung 7: Kaltmiete in €/m² im Verhältnis zur Anzahl der Etagen

Ein ähnliches Bild wie bei der Anzahl der Zimmer ergibt sich in Abbildung 7 auch bei Betrachtung der Anzahl der Etagen des Gebäudes, in welchem sich das

jeweilige Mietobjekt befindet. Hierbei ist anzumerken, dass der Großteil des Datensatzes aus Gebäuden mit einer Etagenanzahl zwischen eins (= Erdgeschoss) und sechs besteht.

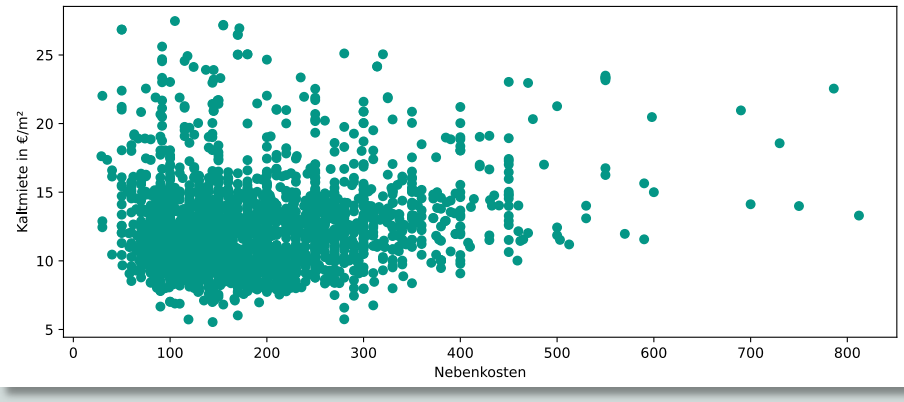

Abbildung 8: Kaltmiete in €/m² im Verhältnis zu den Nebenkosten

Zum Verhältnis der Zielvariablen mit den Nebenkosten lässt sich sagen, dass sich hier ebenfalls kein direkter Zusammenhang erkennen lässt. Es fällt lediglich auf, dass im Datensatz vermehrt Mietobjekte enthalten sind, welche bei einer Kaltmiete in €/m² zwischen ca. 8 €/m² und ca. 16 €/m² für gewöhnlich Nebenkosten zwischen ca. 100 € und 300 € pro Monat mit sich bringen.

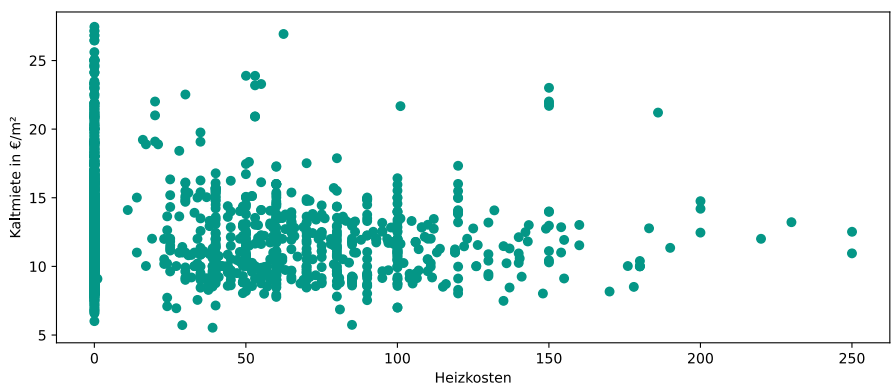

Abbildung 9: Kaltmiete in €/m² im Verhältnis zu den Heizkosten

Wie bereits in Kapitel 3.3.1 erläutert, kommt es bei den Heizkosten vor, dass

diese Nullwerte enthalten können, was sich auch in Abbildung 9 zeigt. Bei den restlichen Mietobjekten mit tatsächlich, gesonderten Heizkosten scheint es ebenfalls keinen direkten Zusammenhang zur Zielvariablen zu geben.

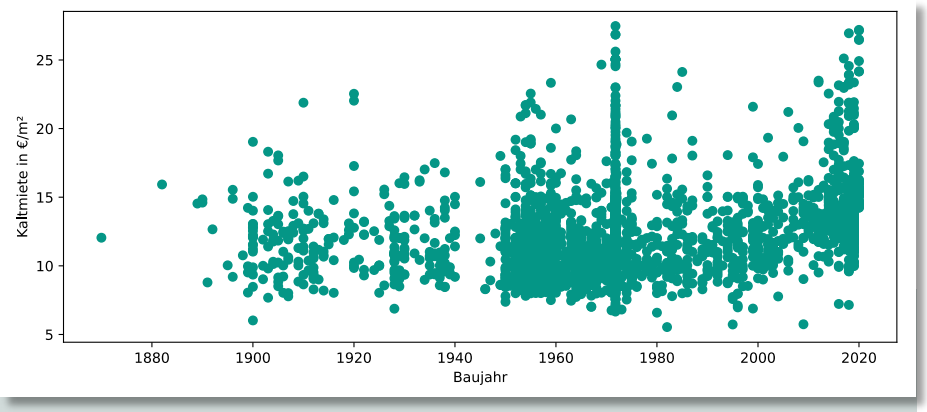

Abbildung 10: Kaltmiete in €/m² im Verhältnis zum Baujahr

Bei der Betrachtung des Baujahres fällt auf, dass ein leichter Aufwärtstrend bei der Zielvariablen für Mietobjekte mit jüngeren Baujahren identifiziert werden kann. Zudem ist die Imputation der Nullwerte durch das durchschnittliche Baujahr ersichtlich[10].

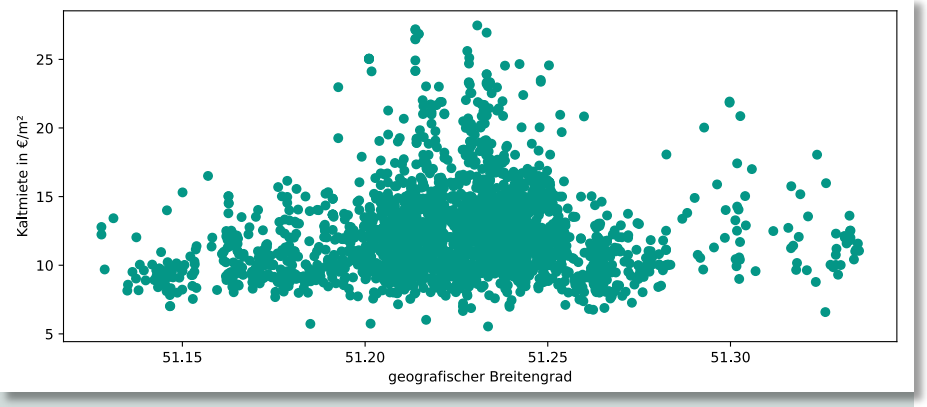

Abbildung 11: Kaltmiete in €/m² im Verhältnis zum Breitengrad

[10] Die auffällig hohen Balken sind gemeint.

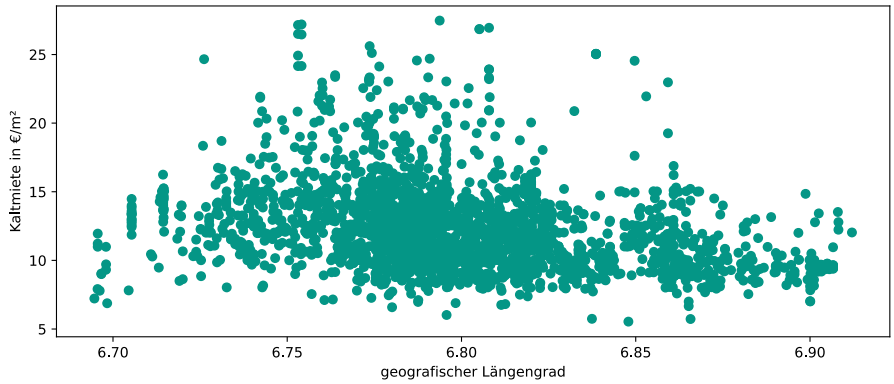

Abbildung 12: Kaltmiete in €/m² im Verhältnis zum Längengrad

Des Weiteren sei noch anzumerken, dass gemessen an den Längen- und Breitengraden eines jeweiligen Mietobjektes ebenfalls kein direkter Zusammenhang zwischen diesen Werten und der Zielvariable hergestellt werden kann, wie Abbildung 11 und Abbildung 12 zeigen. Somit kann ein erstes Faktum festgehalten werden, dass die bloße geografische Lage eines Mietobjektes in Düsseldorf keinen erkennbaren, direkten Einfluss auf die Kaltmiete in €/m² hat[11]. Aus diesem Grund sollen mit dieser Arbeit die nicht-traditionellen Daten in Form von weiteren geografischen Merkmalen, welche einen Einfluss auf die Zielvariable haben könnten, analysiert werden. Die Ergebnisse dieser werden in Kapitel 4.5 gezeigt.

[11] Dies ist für die Debatte unter Praktikern bedeutsam. Manche halten „Lage" für den zentralen Faktor, andere lesen es rückwärts und sagen, dass die Lage per se „egal" ist. Letztere dürfen sich bereits hier bestätigt fühlen. Was wirklich zählt wird im Folgenden deutlich.

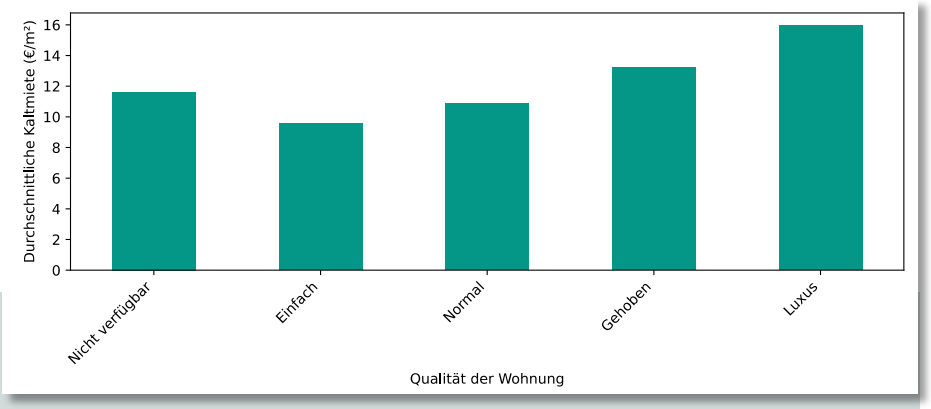

Abbildung 13: Durchschnittliche Kaltmiete in €/m² je Qualitätskategorie

Die Qualität eines Mietobjektes lässt sich vom jeweiligen Inserierenden im Immoscout Portal selbst vergeben, weshalb hier durchaus eine gewisse Verzerrung aufgrund der subjektiven Einschätzung eines Inserierenden enthalten ist. Nichtsdestotrotz zeigt sich, dass die durchschnittliche Kaltmiete je Kategorie durchaus voneinander trennbare Werte annimmt. So haben Luxuswohnungen die im Durchschnitt höchste Kaltmiete pro m² und einfache Wohnungen die niedrigste.

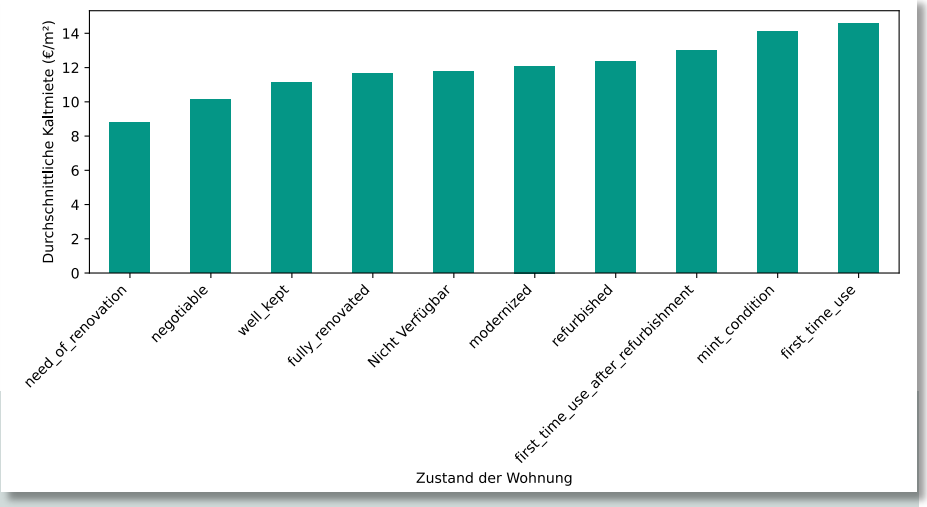

Abbildung 14: Durchschnittliche Kaltmiete in €/m² je Zustandskategorie

Zu den Kategorien des Zustands eines Mietobjektes lässt sich sagen, dass die Beschreibung eines jeweiligen Zustands im Wesentlichen mit den Annahmen über die zu erwartende, durchschnittliche Kaltmiete in €/m² übereinstimmt. Mietobjekte, welche deutlich als renovierungsbedürftig klassifiziert sind, weisen dementsprechend die niedrigste, durchschnittliche Kaltmiete auf, wohingegen Wohnungen zum Erstbezug die höchste, durchschnittliche Kaltmiete in €/m² aufweisen.

Die folgenden Abbildungen und Analysen beziehen sich auf die jeweiligen geografischen Beziehungen der Stadtteile Düsseldorfs zu einigen ausgewählten, metrischen Daten. Der Abbildung 15 lassen sich bspw. die Anzahl der Mietobjekte sowie die Anzahl der im Datensatz befindlichen Mietobjekte pro 1.000 km² Stadtteilfläche je Stadtteil entnehmen.

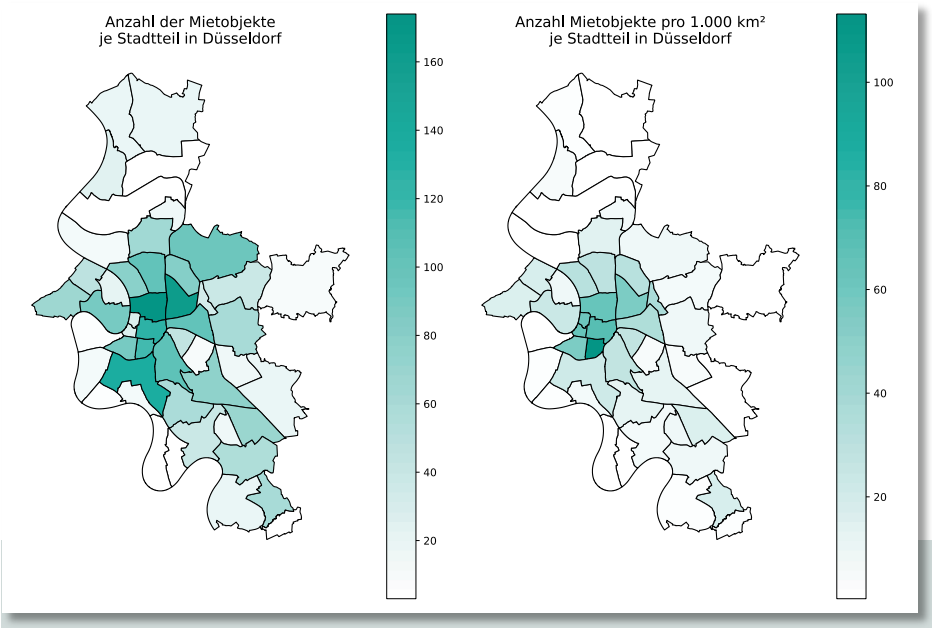

Abbildung 15: Anzahl Mietobjekte und Mietobjekte pro 1.000 km² je Stadtteil

Hieraus ist zu entnehmen, dass sich die meisten Mietobjekte des Datensatzes im Zentrum Düsseldorfs befinden. Dies wird ebenfalls durch die Betrachtung der Dichte der Anzahl der Mietobjekte bezogen auf die räumliche Fläche eines jeweiligen Stadtteils ersichtlich.

Die Ansicht der Kaltmiete in €/m² je Stadtteil in Düsseldorf kann der Abbildung 16 entnommen werden. Hierbei werden die Verteilungen der Kaltmieten in €/m² der einzelnen Stadtteile als Boxplots dem Median nach absteigend sortiert dargestellt. Auffällig ist, dass der Stadtteil Hafen derjenige mit den höchsten und der Stadtteil Hubbelrath derjenige mit den niedrigsten Kaltmieten pro m² ist. Der Stadtteil Hafen sticht hierbei vor allem durch die ebenfalls hohe, minimale Kaltmiete pro m² heraus.

Projiziert man die Daten der durchschnittlichen Kaltmiete pro m² auf eine geografische Ansicht, so ergibt sich die Abbildung 17, in welcher ebenfalls der Stadtteil Hafen direkt heraussticht. Setzt man dagegen die durchschnittliche Wohnfläche des Stadtteils Hafen der Kaltmiete gegenüber wird ersichtlich, dass die Kaltmieten in diesem Stadtteil, im Vergleich zum restlichen Datensatz, sehr hoch ausfallen.

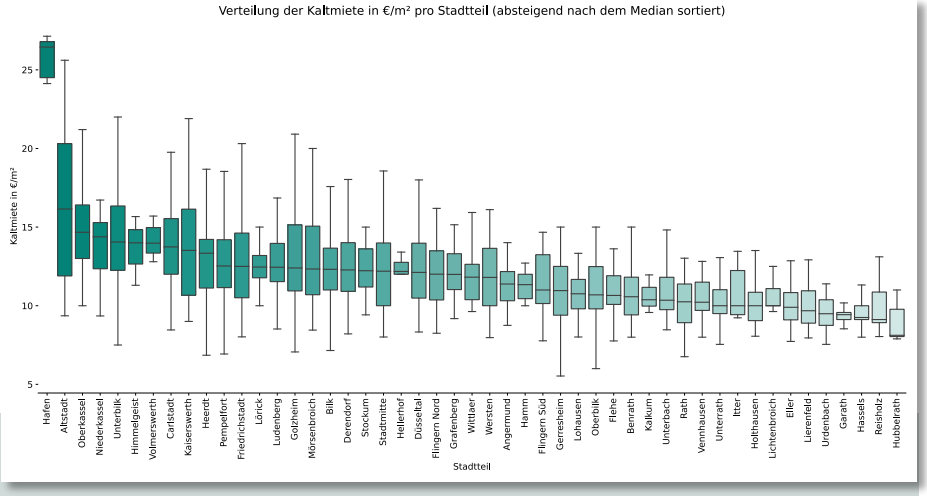

Abbildung 16: Verteilung der Kaltmiete in €/m² je Stadtteil in Düsseldorf

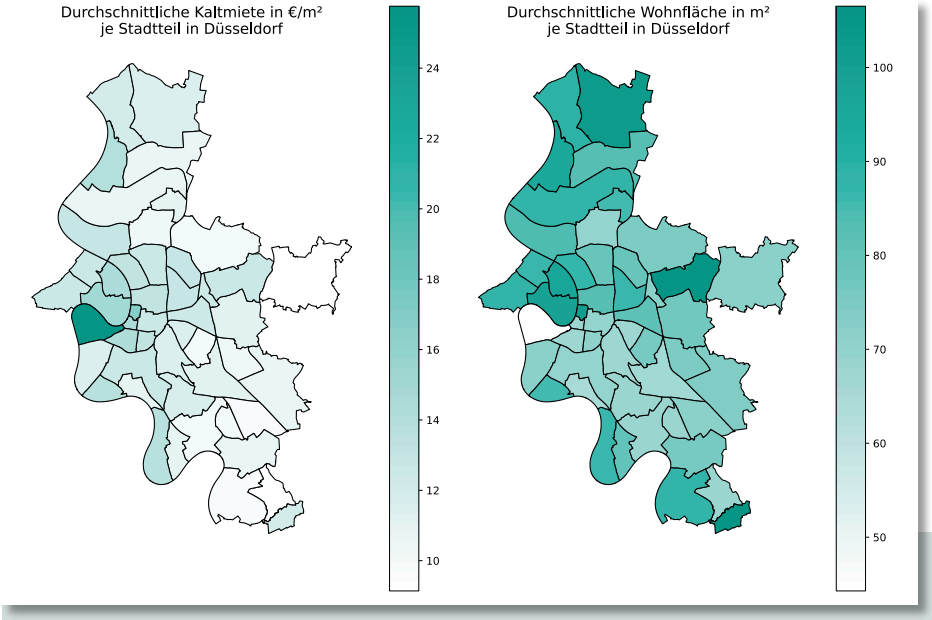

Abbildung 17: Durchschnittliche Kaltmiete pro m^2 und Wohnfläche je Stadtteil

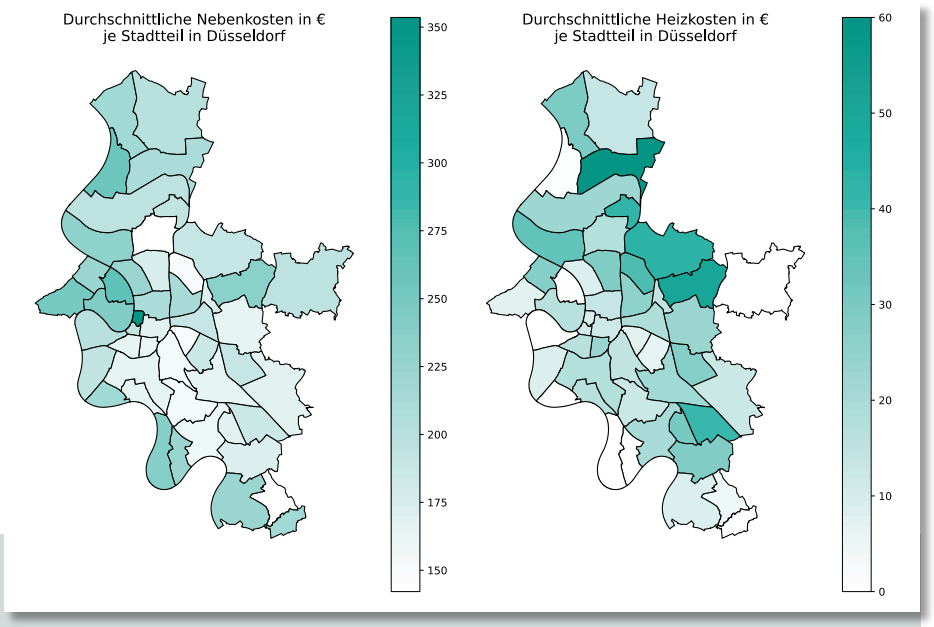

Abbildung 18: Durchschnittliche Nebenkosten und Heizkosten je Stadtteil

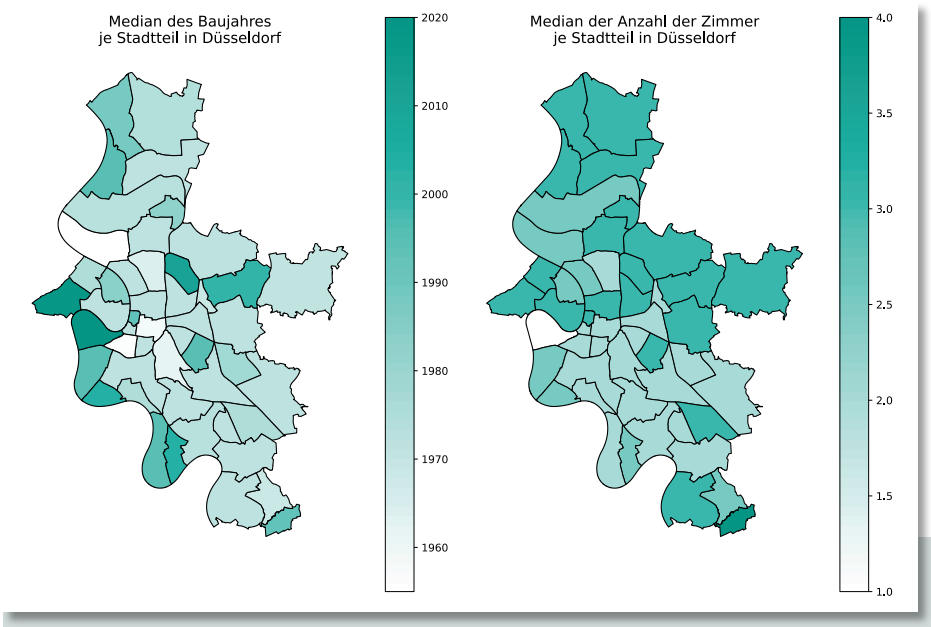

Abbildung 19: Median des Baujahres und Anzahl der Zimmer je Stadtteil

Aus der Abbildung 18 ist ersichtlich, dass sich die durchschnittlichen Nebenkosten bis auf das Maximum im Stadtteil Altstadt nur wenig voneinander unterscheiden. Da die Heizkosten meist auch in den Nebenkosten eines Mietobjektes enthalten sein können, existieren im Immoscout Datensatz viele Mietobjekte mit Nullwerten in dieser Spalte. Aus diesem Grund erscheinen viele Stadtteile in der Abbildung 18 fast ohne entsprechende Einfärbung.

Der Abbildung 19 kann entnommen werden, dass vor allem in den Stadtteilen Hafen und Heerdt Mietobjekte mit jüngeren Baujahren enthalten sind, was ebenfalls ein Indikator für die in diesen Stadtteilen überdurchschnittlichen Kaltmieten sein kann. Zu Median der Anzahl der Zimmer lässt sich feststellen, dass lediglich der Stadtteile Hafen einen Median von einem Zimmer in den entsprechenden Mietobjekten aufweist. Dies erklärt dann wiederum die hohe Kaltmiete der Mietobjekte in diesem Stadtteil (siehe Abbildung 17).

Wie bereits in Kapitel 3.2.2.2 erwähnt wurden insgesamt 157.252 POI in ganz Deutschland mittels der Google Places API abgerufen. Da für die hier stehende Analyse lediglich die POI und Mietobjekte in Düsseldorf interessant sind, wurden

die POI auf einen Bereich rund um Düsseldorf gefiltert. Zudem wurde die Grenze von Düsseldorf um einen Radius von 1.000 Metern erweitert um somit auch POI, welche sich nicht direkt in einem Düsseldorfer Stadtteil befinden aber dennoch im Radius von 1.000 Metern eines Mietobjektes liegen könnten, mit einzubeziehen. Die so entstehende grafische Darstellung all derjenigen POI und Mietobjekte, welche in der Analyse und für das Schätzen der Regressionsmodelle in Kapitel 4 verwendet werden, kann Abbildung 20 entnommen werden.

Zu sehen ist dort, dass, wie bereits bei Abbildung 15 erläutert, die zentralen Stadtteile von Düsseldorf sowohl die meisten Mietobjekte als auch die meisten POI aufweisen. Dadurch könnte es vorkommen, dass die Regressionsmodelle die Kaltmieten der Mietobjekte in den äußeren Regionen Düsseldorfs schlechter prognostizieren können, als es für die Mietobjekte im Zentrum Düsseldorfs der Fall ist.

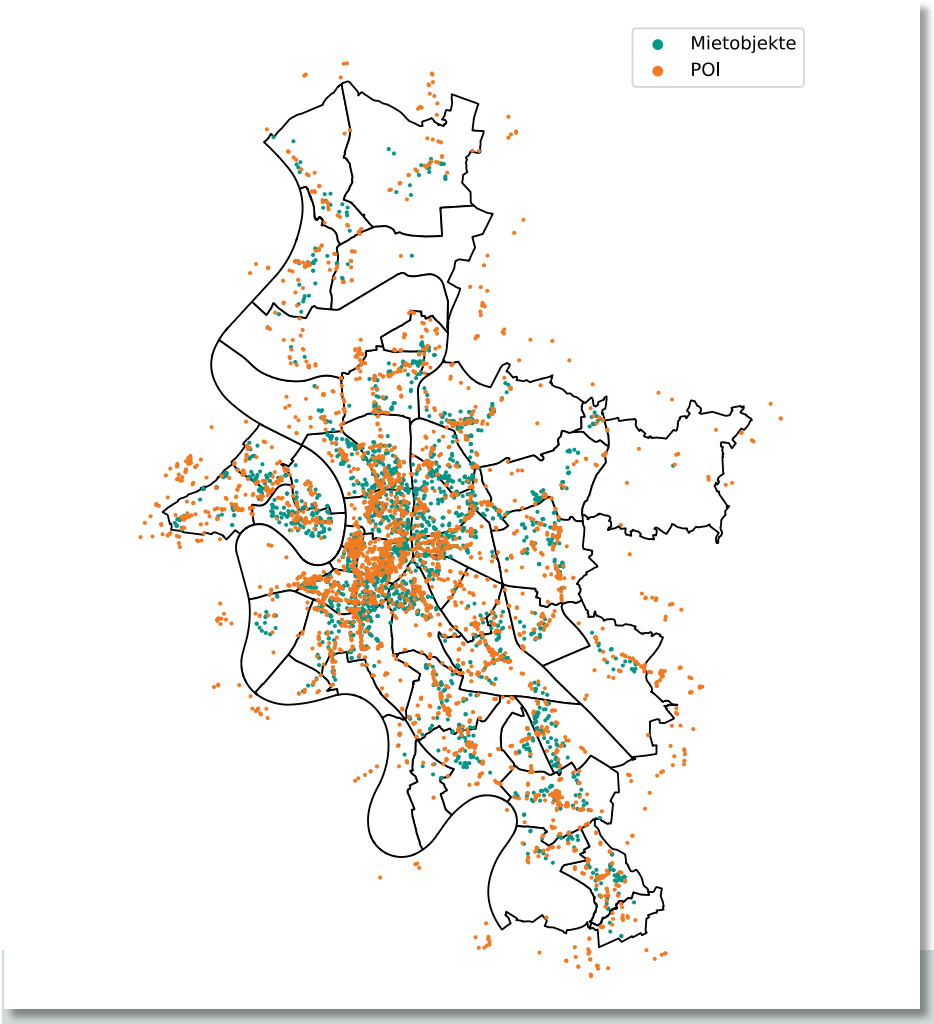

Abbildung 20: Geografische Darstellung der POI und Mietobjekte in Düsseldorf

3.5 Datenvorbereitung zur Analyse

Die modellgestützte Analyse benötigt strukturierte Daten in bestimmten Formaten. In Einzelfällen sind zudem Transformationen hilfreich. Diese vorbereitenden Schritte werden beschrieben.

3.5.1 OneHot - Encoding

Wenn in einem Datensatz kategoriale Daten vorhanden sind, ist es für diverse Analysemethoden notwendig, dass diese in angepasste Merkmalsvektoren (eng. feature vector) in der Regel nominale Werte umgeschrieben werden. Hierfür ist das OneHot - Encoding eine der weitverbreitesten Methoden. Das OneHot - Encoding geht wie folgt vor: Nehmen wir bspw. eine Spalte eines Datensatzes, welches folgende kategoriale Werte aufweist: [Rot, Grün, Blau]. Dann erstellt das OneHot - Encoding daraus einen dreidimensionalen Vektor: [[1, 0, 0], [0, 1, 0], [0, 0, 1]] in welchem jede Dimension eine neue Spalte bedeutet. Im Ergebnis sind alle Spalten orthogonal und haben den gleichen Abstand zueinander (Cerda et al., 2018, p. 1478).

3.5.2 Datentransformationen und Aufteilung für Training und Test

Gemäß (Larose & Larose, 2015, pp. 716–717) funktionieren Regressionen am besten, wenn die Häufigkeitsverteilung der Variablen einer Normalverteilung ähnlich ist, zumindest aber annähernd symmetrisch und unimodal. Darum werden manche numerischen Variablen transformiert.

Zu den möglichen Transformationen gehören die Natural Log (ln) Transformation, die Square-Root Transformation, die Power Transformation und die Box-Cox Transformation (Larose & Larose, 2015, pp. 213–220).

Da wir uns nicht auf eine der vier Transformationen festlegen wollten, haben wir damit begonnen den Effekt je Variable zu untersuchen, bevor diese final angewandt wurde. Abbildung 21 zeigt einen Ausschnitt der Suche nach der optimalen Transformation. Bei diesem Vorgehen werden die jeweiligen metrischen nicht transformierten Spalten den drei transformierten Pendants gegenübergestellt. Zudem befindet sich in der Ecke oben links die Maßzahl Schiefe. Auf Basis dieser Betrachtungen können wir die augenscheinlich optimale Transformation auswählen.

Zudem existieren im Datensatz bspw. bei den Etagen eines Mietobjektes negative Werte, welche uns bei der Transformation Probleme bereiten können. Liegen diese Objekte unter dem Erdgeschoss werden diese in der Regel mit einem Wert von -1 versehen. Hinzu kommen einige Nullwerte in unserem Datensatz. Dies kann zu Fehlern führen, da die Box-Cox Transformation nur positive Werte verarbeiten kann (Box & Cox, 1964, p. 214). Aus diesem Grund entschieden wir, die Box-Cox Transformation aus unserer Untersuchung auszuschließen.

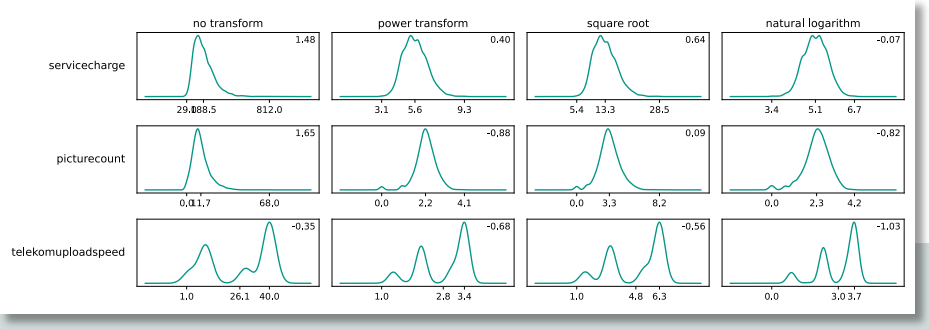

Abbildung 21: Untersuchung der Transformationsarten bei traditionellen Daten

Mit der Betrachtung auf jede einzelne Bezugsvariable und den Betrachtungen der Auswirkungen durch die verschiedenen Transformationen hatten wir die Möglichkeit, die jeweils optimale Variante zu wählen. Bezogen auf unseren Datensatz ergab dies folgendes Bild:

Transformation	Anzahl der Bezugsvariablen
Natural Log (ln)	5
Square-Root	1
Power	3
Keine Transformation	7
Gesamt	16

Tabelle 2: Übersicht der Transformationen traditioneller Daten

Auch die nicht-traditionellen Daten weisen vor der Transformation noch teilweise starke Schiefen in den jeweiligen Verteilungen auf. Aufgrund der hohen Anzahl an Spalten für die nicht-traditionellen Daten (insgesamt 48 Spalten) wäre es zu aufwendig gewesen, die Verteilung jeder einzelnen Spalte im Detail zu betrachten. Daher wurde ein Algorithmus entwickelt, welcher die anzuwendende Transformierungsfunktion auf Basis der geringsten Schiefe aller Transformie-

rungsfunktionen auswählt. Die zur Verfügung stehenden Transformierungsfunktionen sind hierbei dieselben wie diejenigen, die zur Transformierung der traditionellen Daten eingesetzt werden. In Tabelle 10 im Anhang sind die jeweiligen, angewandten Transformierungsfunktionen je Spalte aufgelistet. Nach diesen erfolgten Transformierungen aller einzelnen Spalten ähneln die Verteilungen der nicht-traditionellen Daten im Wesentlichen einer Normalverteilung.

Aufgrund der teilweise hohen, wertmäßigen Varianzen einiger nicht transformierter Spalten, wurden sämtliche Daten in einem letzten Schritt noch mittels Min-Max-Normalisierung (Vgl. Larose & Larose, 2015, p. 30) auf den Wertebereich [0, 1] skaliert. Dies ist bspw. ebenfalls bei der Arbeit mit neuronalen Netzen der Fall, wenn das Outputneuron einen eingeschränkten Wertebereich hat.

Insgesamt liegen für 2.499 Objekte 90 Attribute vor. Eines dieser Attribute ist die vorherzusagende Miete (Regressand), so dass die übrigen 89 die Regressoren sind. Fünf Objekte werden für die spätere Evaluierung im Detail entnommen.

80 Prozent der verbleibenden 2.494 Objekte werden zufällig für den Trainingsdatensatz ausgewählt. Die übrigen 20 Prozent bilden den Testdatensatz, auf dem die Gütemaße gerechnet werden.

4 Anwendung und Evaluation

In den folgenden Unterkapiteln wird auf die drei, in der Analyse der traditionellen sowie nicht-traditionellen Daten verwendeten, Regressionsmodelle eingegangen. Hierbei handelt es sich einerseits um die multiple lineare Regression, den Random Forest und andererseits um den XGBoost Algorithmus. Im Anschluss an die Darstellung der drei Modelle folgt die Evaluation der Ergebnisse anhand einer Evaluationsstichprobe mit n = 5 Objekten, welche im Vorfeld des Trainings aus dem Datensatz entnommen und in Kapitel 4.5 analysiert werden.

4.1 Lineare Regression

Die multiple lineare Regression ist ein etabliertes Verfahren der Aufdeckung von Zusammenhängen in Daten. Eine Variable (Mieten) wird zurückgeführt auf die sie erklärenden Faktoren.

4.1.1 Grundlagen und Arten

Im Kern ist die multiple lineare Regression eine Methode zur Erklärung von numerischen Werten $\hat{y}$, auch Regressanden genannt, gegeben einer Reihe unabhängiger Werte x, auch Regressoren genannt. Dabei wird der Regressand als lineare Funktion der Regressoren angenommen. In der Literatur wird diese Methode auch als „Grundpfeiler der Ökonometrie“ (Verbeek, 2017, p. 6) bezeichnet, wodurch der Stellenwert der linearen Regression in der Wissenschaft ersichtlich wird.

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_n x_{in} + \varepsilon_i \quad (1)$$

Formel 1: Modell der multiplen linearen Regression, Quelle: in Anlehnung an (Verbeek, 2017, p. 13)

Der Schätzalgorithmus zur Bestimmung der Parameter der linearen Funktion besteht in der Minimierung der summierten, quadrierten Abweichungen zwischen Modellprognose $\hat{y}$ und den tatsächlichen, wahren Werten y des Regressanden. Diese Abweichungen werden auch Residuen genannt und mit dem Term ε_i bezeichnet. Formal lautet der Schätzalgorithmus:

$$\hat{\beta} = \arg\min_{\beta} \beta_0 + \sum_{i=1}^{n} (y_i - \beta_i x_i)^2 \quad (2)$$

Formel 2: Minimierungsfunktion der kleinsten Quadrate, Quelle: in Anlehnung an (Verbeek, 2017, p. 7) und (Tibshirani, 1996, p. 268)

Die derart geschätzten Koeffizienten können in Zusammenhang mit neuen Eingabevariablen für die Regressoren x_i bis x_n verwendet werden, um neue $\hat{y}$ Werte zu prognostizieren.

Die beschriebene Methode der kleinsten Quadrate kann um einen Strafterm erweitert werden, durch welchen ein Modell mit geringeren bzw. weniger Koeffizienten einem anderen Modell vorgezogen wird. Als wesentliche Vertreter haben sich der „least absolute shrinkage and selection operator" (LASSO) (Vgl. Tibshirani, 1996) und das Modell der „ridge regression" bzw. Tikhonov-Regularisierung, benannt nach dem Mathematiker Andrey Nikolayevich Tikhonov, (im Folgenden lediglich: Ridge) (Vgl. Hoerl & Kennard, 1970, p. 59) etabliert. Beide Herangehensweisen fügen der oben beschriebenen OLS-Minimierungsgleichung einen Strafterm hinzu, welcher sich an den jeweiligen Koeffizienten $\hat{\beta}_{i\ldots n}$ bemisst. Der Unterschied zwischen den beiden Methoden besteht darin, dass das LASSO-Modell den Strafterm als Summe der absoluten Werte der Koeffizienten und das Ridge-Modell diesen als Summe der quadrierten Werte der Koeffizienten definiert. Beide Modelle multiplizieren diese Summe schließlich noch mit einem vorher definierten Parameter α, welcher vorgibt, wie stark der Strafterm gewichtet werden soll (Vgl. Tibshirani, 1996, p. 268).

In der vorliegenden Arbeit werden die Implementierungen des LASSO- und des Ridge-Modells im scikit-learn Framework genutzt. Beide Modelle unterstützen hierbei die Angabe des Parameters α sowie eines weiteren Parameters tol, welcher die Präzision der Ergebnisse eingrenzt.[12] Die Werte dieser beiden Hyperparameter sind zunächst aufgrund der Tatsache, dass diese theoretisch unendliche Werte annehmen können, ungewiss. Aus diesem Grund werden zunächst zufällige Werte zu Beginn festgelegt und dann einer sogenannten Hyperparameteroptimierung unterzogen, welche im Folgenden näher beschrieben wird.

[12] Siehe: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html?highlight=
ridge#sklearn.linear_model.Ridge.

4.1.2 Hyperparameteroptimierung

Im Rahmen der Hyperparameteroptimierung werden diejenigen Modelle gesucht, welche anhand anpassbarer Hyperparameter die gegebenen Daten am besten abbilden[13]. Da einige Modellparameter praktisch unendlich viele mögliche Ausprägungen haben können, gestaltet es sich schwierig, direkt beim ersten Versuch die optimalen Parameter für ein Modell festzulegen. Aus diesem Grund wurden Verfahren entwickelt, welche automatisiert eine festgelegte Anzahl von Möglichkeiten und Parameterkombinationen durchsuchen, um so schließlich optimale Einstellungen der Parameter zu finden.

Im Rahmen dieser Arbeit wurde die sogenannte Rastersuche (englisch: Grid Search), welche in der scikit-learn Bibliothek als GridSearchCV[14] implementiert ist, angewendet, um das optimale lineare Regressionsmodell zu ermitteln. Hierbei ist anzumerken, dass die gewöhnliche lineare Regression keine einstellbaren Hyperparameter aufgrund der Vorgehensweise mittels OLS bietet. Im Vergleich dazu können für die LASSO- und Ridge-Modelle einstellbare Parameter identifiziert werden, wie sie in Kapitel 4.1.1 bereits erklärt wurden. Die einzelnen Parameter sowie die in der Rastersuche verwendeten Intervalle bzw. Bandbreiten sind Tabelle 3 zu entnehmen.

Parameter	Intervalle
alpha	20 gleichverteilte Werte aus dem Intervall [0,0001; 10]
tol	20 gleichverteilte Werte aus dem Intervall [0,00000001; 2,5]

Tabelle 3: Hyperparameter der linearen Regression

Nach durchgeführter Suche stellt sich heraus, dass ein Ridge Modell mit den Parametern alpha = 3,158 und tol = 0,00000001 die vorhandenen Daten am besten abbilden kann. Dieses Modell wird nun im weiteren Verlauf der Arbeit zur Evaluation der Ergebnisse auf bisher ungesehene Mietobjekte angewendet.

[13] Im Sinne des R^2.

[14] Siehe: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

4.2 XGBoost

In den letzten Jahren hat sich das Tree Boosting, welches zu den Machine Learning Methoden gehört, etabliert, was auf die Erfolge, welche diese Methode in Wettbewerben, wie beispielweise dem Netflix Preis (Bennett & Lanning, 2007) erlangte, zurückzuführen ist. Auf diesem Fundament wurde der Extreme Gradient Boosting (kurz XGBoost) entwickelt. Diese ist eine optimierte verteilte Gradient Boosting Bibliothek, welche auf hohe Effizienz, Flexibilität und Portabilität ausgelegt worden ist und sich als ein end to end tree boosting system versteht.[15] (Chen & Guestrin, 2016, p. 1). Der XGBoost kam auch beim Zillow Prize: Zillow's Home Value Prediction (Zestimate)[16], einem hochdotierten Wettbewerb aus der Immobilienbranche, bei den Top 3 Teams zum Einsatz.

Die wichtigsten Faktoren für den Erfolg von XGBoost sind zum einen die Einführung eines Regularized Learning Objects, welches dem Overfitting des Modells entgegentritt. Es funktioniert analog zur linearen Regression durch Addition eines Strafterms zur zu minimierenden Funktion. Dadurch werden komplexe Bäume bestraft.

Zum anderen bietet die Skalierbarkeit des Modells einen erheblichen Vorteil und macht XGBoost damit zu einem Big Data tauglichen Werkzeug.

Zu den diesbezüglichen Innovationen gehören:

- Weighted Quantile Sketch
- Sparsity-aware Split Finding
- Cache-aware Access
- Blocks for Out-of-core Computation
- Column Block for Parallel Learning

Wir wollen nur die letzten drei skizzieren und verweisen für das Übrige auf die Ursprungsliteratur (Chen & Guestrin, 2016, p. 1).

15 Siehe https://xgboost.readthedocs.io/en/latest/index.html.
16 Siehe https://www.kaggle.com/c/zillow-prize-1.

4.2.1 Technische Innovationen

Das Systemdesign der Spaltenblöcke ist darauf ausgelegt die Bearbeitungszeit des XGBoost Algorithmus zu verkürzen.

Hierbei werden die Spalten der Datenbank in vorsortierte Einheiten verarbeitet, welche im Arbeitsspeicher des Systems gespeichert werden. Die Entwickler haben diese Einheiten als „Blöcke" getauft. Der Vorteil dieser Blöcke liegt darin, dass die Sortierungen der Spalten sowie auch das Layout der Eingabedaten einmalig vor dem Training errechnet werden müssen und im Nachgang über einfache Iterationen wieder benutzt werden können. Zudem kann das Sammeln von Statistiken zu diesen spaltenbasierten Blöcken parallelisiert werden, was wiederum viel Zeit einspart. Gespeichert werden diese Blöcke im compressed column format (kurz CSC). Zu diesem Umgang mit den zu analysierenden Daten kommen zwei weitere Innovationen hinzu, welche das Modell von spaltenbasierten Blöcken weiter unterstützen. Dazu gehören das Cache-aware Access und die Blocks for Out-of-core Computation. (Chen & Guestrin, 2016, p. 5)

Um einen Leistungsverlust durch einen „cache-miss effect", also einen Verlust von Daten, welche während der Berechnung der neuen Outputs pro Baumzweig im L1-Cache des Systems gelagert sind, zu vermeiden, setzten Chen & Guestrin den Cache-aware Access ein. Dabei holt ein Thread die Daten aus einem nicht-kontinuierlichen Speicher in einen kontinuierlichen Puffer und akkumuliert dabei die Gradientenstatistiken im Hauptthread, welche benötigt werden, um die Outputs der Bäume zu berechnen. (Chen & Guestrin, 2016, n. 6)

Das Blocks for Out-of-core Computation dient dazu, große Datenmengen, welche nicht komplett in den Hauptspeicher geladen werden können, schnell und effizient bearbeiten zu können. Dafür unterteilt der XGBoost die Daten in Blöcke auf dem Festplattenspeicher um Out-of-core Computation zu ermöglichen. Diese Blöcke können dann im Pufferspeicher vorgeladen werden und somit bei Bedarf schnell vermittelt werden, sobald die notwendigen Spalten im Datensatz für das Erstellen des Baums gebraucht werden. Zwei Funktionen ermöglichen XGBoost die Out-of-core Computation. Zum einen die Block Compression, welche für die Kompression der Blöcke zuständig ist und bestimmt welche dann unterteilt auf den Festplattenspeicher gespeichert werden und zum anderen das Block Sharding, welches für das Einladen der Daten in den Puffer zuständig ist. (Chen & Guestrin, 2016, p. 8)

4.2.2 Hyperparameteroptimierung via GridSearch

Als Verfahren für die Hyperparameteroptimierung[17] nutzen wir die Rastersuche, welches neben der Zufallssuche (eng. Random Search) ein gängiges Verfahren zur Optimierung der Parameter ist. Der Algorithmus für die Rastersuche trainiert das Modell für jede übergreifende Spezifikation der Hyperparameter im kartesischen Produkt der Wertemenge für jeden einzelnen Hyperparameter. Das Experiment mit dem besten Validierungsdatenfehler wird dann als Optimierung gewählt, in welchem die besten Hyperparameter ermittelt worden sind. Die Wahl der Parameter, welche trainiert werden sollen, wird auf Basis vorheriger Erfahrungen festgelegt und numerisch geordnet in einer Liste dargestellt. Die Ordnung der Elemente erfolgt vom kleinsten bis zum größten Element. (Goodfellow et al., 2016, pp. 420–421)

Unter Berücksichtigung der Argumente, welche uns in der XGBRegressor Methode[18] der scikit-learn API zur Verfügung stehen, haben wir uns für die folgenden Wertemengen an Parametern entschieden:

Argument	Parameter Wertemenge	Beschreibung des Parameters
col-sample_by-tree	[0.1, 0.3, 0.5, 0.7]	Ziel des Parameters ist es die Varianz der Modelle durch Training auf echten Teilmengen der Faktoren zu reduzieren.
learn-ing_rate	[0.01, 0.05, 0.1, 0.15]	Bestimmt die Rate, mit welcher die Größe der Bäume eingeschränkt werden soll.
max_depth	[5, 7, 10, 12]	Bestimmt wie tief die Bäume wachsen dürfen.
alpha	[6, 8, 10, 12]	L1 Regulierung der Gewichte.
n_estimators	[10, 50, 100, 500]	Anzahl der Bäume, die zum Vorhersage-Ensemble gehören.

Tabelle 4: Auswahl der Parameter zur Optimierung

[17] Für die Hyperparameteroptimierung nutzen wir die Funktion sklearn.model_selection.GridSearchCV aus der scikit-learn Bibliothek.

[18] https://xgboost.readthedocs.io/en/latest/python/python_api.html#module-xgboost.sklearn

4.2.3 k-fold Cross Validation

Die k-fache Kreuzvalidierung (eng. k-fold Cross Validation, kurz KCV) ergibt einen weiteren Hyperparameter (Anguita et al., 2012, p. 1).

Bei kleinen Datensätzen kann es bei der Aufteilung in Test und Trainingsdatensätzen für die Validierung zu Problemen kommen, denn eine kleine Testdatenmenge impliziert eine statistische Unsicherheit für den geschätzten durchschnittlichen Testfehler. Die k-fache Kreuzvalidierung bietet eine Alternative, mit der alle Beispiele für die Schätzung des mittleren Testfehlers verwendet werden können. Diese Art der Validierung bildet aus dem Datensatz k Teilmengen ohne Überschneidung. Mithilfe dieser k Teilmengen kann nun der Testfehler durch das Bilden des durchschnittlichen Testfehlers über k Versuche geschätzt werden. Dabei werden die jeweiligen gebildeten Datenmengen als Testdatenmengen verwendet, während der Rest als Trainingsdatensatz verwendet wird (Goodfellow et al., 2016, pp. 118–119).

Für unser Modell haben wir den Datensatz für die Validierung in fünf Teilmengen aufgeteilt.

4.3 Ergebnisse der Modellierungen

Die verwendeten Ansätze der Modellierung sind verschieden. So auch ihre Ergebnisse. Umso interessanter ist die Schnittmenge der Übereinstimmungen.

4.3.1 Lineare Regression

Für das Modell der multiplen linearen Regression[19] ist nach erfolgter Hyperparameteroptimierung und Nutzung eines quadratischen Strafterms ein Modell herausgekommen, welches die vorliegenden Daten gut erklären kann. Die Güte der hier vorliegenden Modelle soll einerseits durch die Wurzel der mittleren Fehlerquadratsumme (englisch: root mean squared error bzw. RMSE) und andererseits durch das Bestimmtheitsmaß R^2 angegeben werden. Beides sind übliche Maßzahlen.

[19] Zu erwähnen ist hierbei, dass sich der RMSE auf die mittels natürlichen Logarithmus skalierten Kaltmieten in €/m² bezieht.

Hierbei beläuft sich der RMSE für das lineare Regressionsmodell, unter Anwendung auf die Teststichprobe i.H.v. 20 Prozent des ursprünglichen Datensatzes, auf 0,103096 und das R^2 auf 52,60 Prozent.

Die so trainierten, wesentlichen Koeffizienten[20] des Modells können Abbildung 22 entnommen werden. Diese Koeffizienten haben demnach den größten, wertmäßigen Einfluss auf die Zielvariable der logarithmierten Kaltmiete in €/m^2.

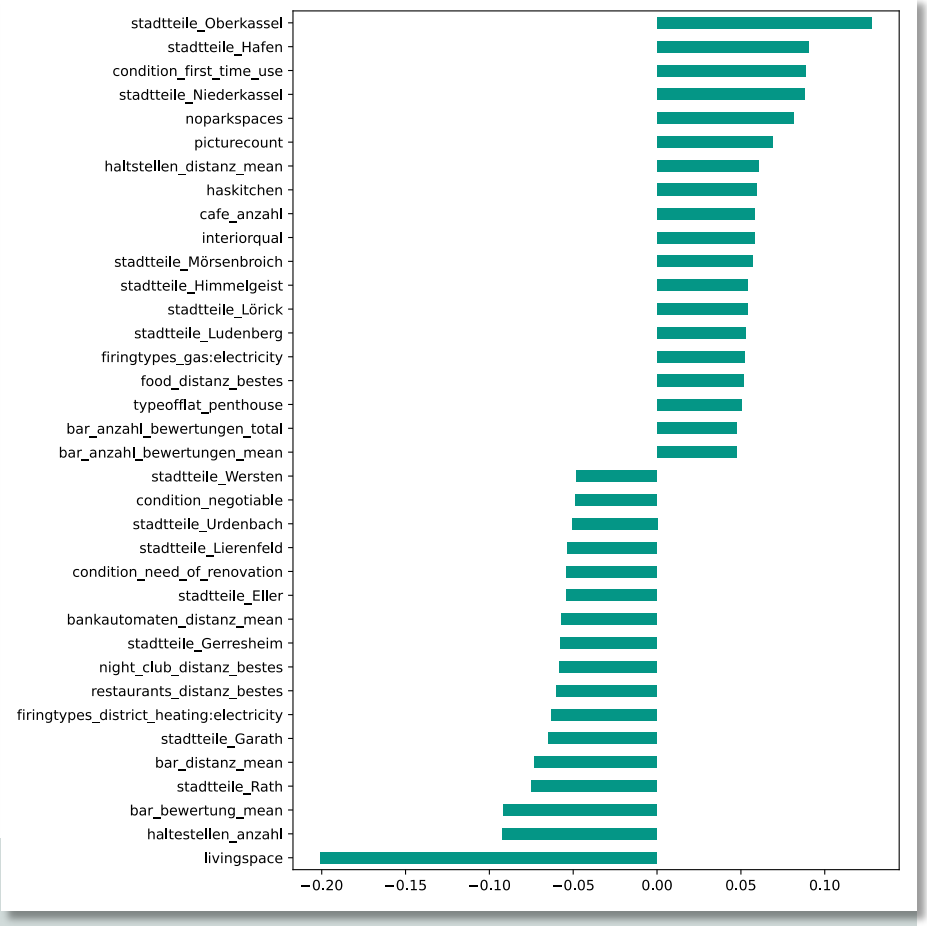

Abbildung 22: Höchste und geringste Koeffizienten der linearen Regression

[20] Durch die Verwendung des Strafterms wird das Modell sparsam gehalten. Die p-Werte der hervorgehobenen Faktoren sind regelmässig kleiner als 5%.

Anhand der Variablen $\hat{\beta}_{bar\_bewertung\_mean}$ lässt sich beispielhaft ein Szenario für den Einfluss dieser Variablen auf die Zielvariable wie folgt skizzieren: Angenommen sei, dass sich die durchschnittliche Bewertung von Bars im Umkreis von 1.000 Metern eines Mietobjektes auf einen Wert von $x_{bar\_bewertung\_mean} = 4{,}2$ beläuft. Hierbei ist anzumerken, dass sich die durchschnittliche Bewertung von Bars im gesamten Datensatz auf ca. 4,2 von 5 möglichen Sternen beläuft.

Da diese Variable auf einen Wertebereich zwischen [0, 1] skalierten wurde, muss diese durchschnittliche Bewertung den ebenso skalierten Wert zur weiteren Berechnung annehmen. Hierbei würde die Variable $\hat{\beta}_{bar\_bewertung\_mean}$ einen skalierten Wert von 0,8209 annehmen. Multipliziert man diesen Wert mit dem Koeffizienten $\hat{\beta}_{bar\_bewertung\_mean} = -0.0915$ ergibt sich somit eine Auswirkung auf die endogene Variable $\hat{y}$ c.p. i.H.v. -0,0751.

Da die Zielvariable jedoch vorab mit dem natürlichen Logarithmus skaliert worden ist, muss diese Transformation nun mithilfe der Exponentialfunktion zur Basis e invertiert werden. Wendet man diese Exponentialfunktion auf den soeben errechneten Wert an, ergibt sich schließlich ein Einfluss i.H.v. 0,9276 auf die Kaltmiete in €/m² dieses beispielhaften Mietobjektes.

Würde man für dieses beispielhafte Mietobjekt lediglich diese eine Variable zur Vorhersage der Kaltmiete in €/m² hinzunehmen, müsste dieser errechnete Wert nun mit dem y-Achsenabschnitt des Modells, nämlich der Variablen $\hat{\beta}_0$, summiert werden. Der y-Achsenabschnitt des Modells beläuft sich auf $\hat{\beta}_0 = 13{,}3965$ weshalb sich nun eine prognostizierte Kaltmiete in €/m² i.H.v. $\hat{y} = 14{,}3241$ ergibt.

Schließlich lässt sich zum Modell der multiplen linearen Regression bei der Betrachtung der Residuen (siehe Abbildung 23), welche auf Basis des Testdatensatzes ermittelt wurden, festhalten, dass das Modell vor allem bei höheren Kaltmieten pro m² stärkere Schwankungen aufweist. Ein Test nach Breusch-Pagan (Vgl. Breusch & Pagan, 1979)[21] bestätigt die Vermutung, dass die Nullhypothese des Tests mit einem p-Wert nahe Null verworfen werden kann und somit die Alternativhypothese, nämlich dass Heteroskedastizität in den Residuen vorliegt, angenommen werden muss.

[21] Siehe zudem die Implementierung hierzu in https://www.statsmodels.org/stable/generated/statsmodels.stats.diagnostic.het_breuschpagan.html#statsmodels.stats.diagnostic.het_breuschpagan.

Möglicherweise könnte eine andere Spezifikation, die die Heteroskedastizität explizit reflektiert, oder eine weitergehende Datenreinigung zu besseren Ergebnissen führen. Dies ist Potential für zukünftige Analysen.

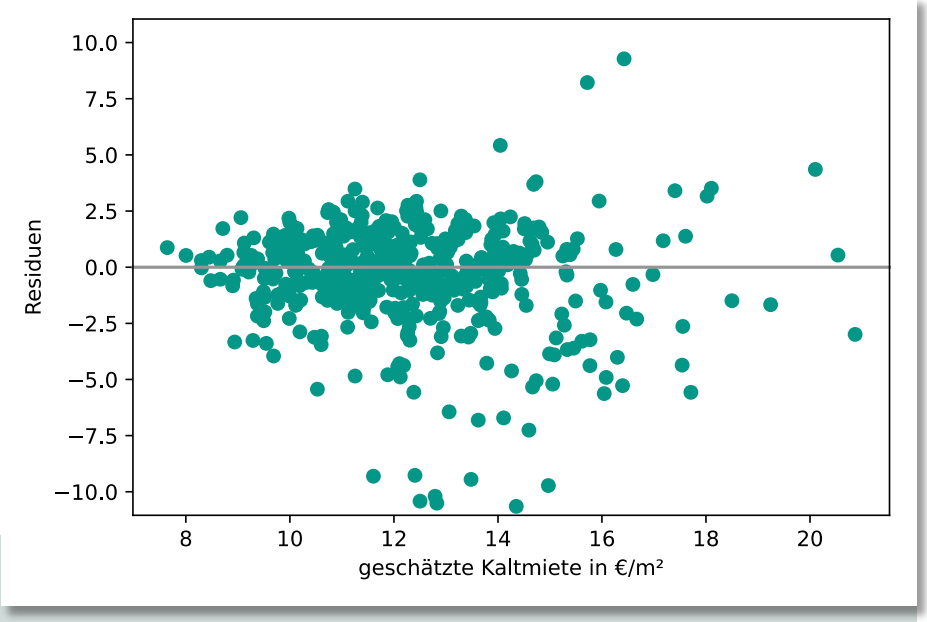

Abbildung 23: Residuen der multiplen, linearen Regression

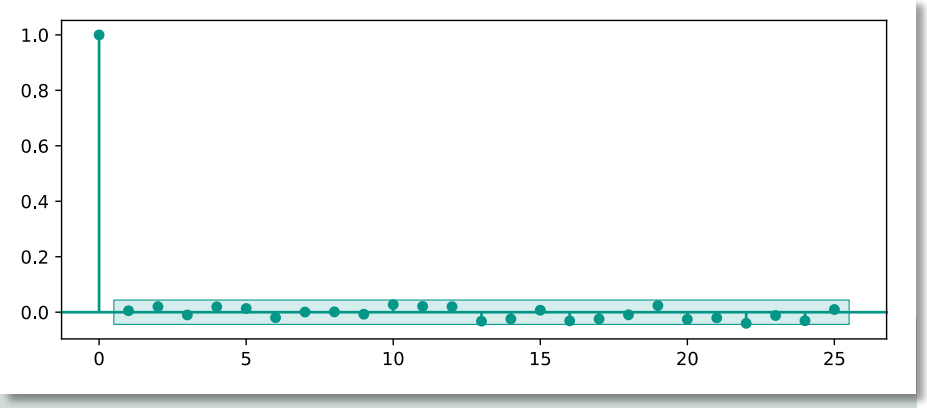

Abbildung 24: Plot der Autokorrelationen im linearen Modell

Die Darstellung der Autokorrelationen der Residuen (siehe Abbildung 24) sowie ein Test nach Breusch-Godfrey (Vgl. Breusch, 1978 und; Godfrey, 1978) legen nahe, dass keine Autokorrelation vorliegt. Der Test nach Breusch-Godfrey gibt einen p-Wert i.H.v. 0,53206 aus und ist somit größer als unser Signifikanzniveau von 5 Prozent (0,05), weshalb die Nullhypothese des Tests, dass keine Autokorrelation vorliegt, nicht verworfen werden kann.

4.3.2 XGBoost

Der GridSearch auf Basis der Parameterwertemengen, welche im Kapitel 4.2.3 erläutert wurden, führte zur folgenden Auswahl der Hyperparameter.

Argument	Parameter
colsample_bytree	0.3
learning_rate	0.1
max_depth	10
alpha	6
n_estimators	500

Tabelle 5: Optimale Parameter für den XGBoost

Die Messung der Güte des Modells erfolgte - genauso wie bei der linearen Regression -über die Maßzahlen RMSE sowie R^2. Die Gütemessungen wurden mithilfe derselben Teststichprobe von 20 Prozent vorgenommen und führten zu einem geringeren RMSE von 0,097654 sowie einem höheren R^2 von 56,01 Prozent. Die zwanzig wichtigsten Faktoren und deren Einfluss auf die Schätzungen werden in der Abbildung 25 gelistet.

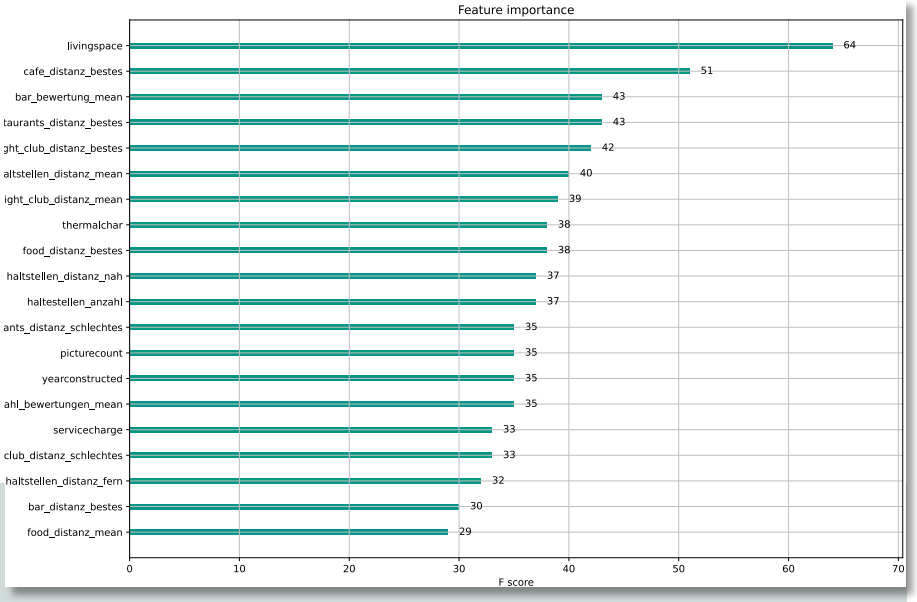

Abbildung 25: Wichtigste Faktoren beim XGBoost

Nach der Analyse von McKinsey[22] spielen nicht-traditionelle Faktoren eine wesentliche Rolle. So auch hier: Es ist auffällig, dass nahezu alle nicht-traditionellen Faktoren, die eine Distanz zu einem POI aufweisen, zu den wichtigsten „Features" gehören. Diese Faktoren repräsentieren bspw. 70 Prozent der Top 10 Faktoren. Bei den traditionellen Faktoren stechen die Anzahl der Bilder[23], welche im Inserat zur Vorstellung des Mietobjekts enthalten sind, gefolgt von der Wohnfläche hervor.

4.3.3 Random Forest

Als dritte Methode zur Vorhersage verwendeten wir den Random Forest (Breiman, 2001). Die Wahl der Hyperparameter erfolgte analog zum XGBoost, so dass wir für den Random Forest die gleiche Größe und Anzahl der Bäume wie im Kapitel 4.3.2 wählten. Unter diesen Voraussetzungen erzielte der Random Forest eine Modelgüte R^2 von 60,57% sowie einem Durchschnittsfehler

22 Vgl. Kapitel 2.1.

23 Gemeint ist die Variable „picturecount".

RSME von 0,092459. Mit diesen Werten schlägt sich der Random Forest mit knappem Abstand an die Spitze der genutzten Prognosemodelle.

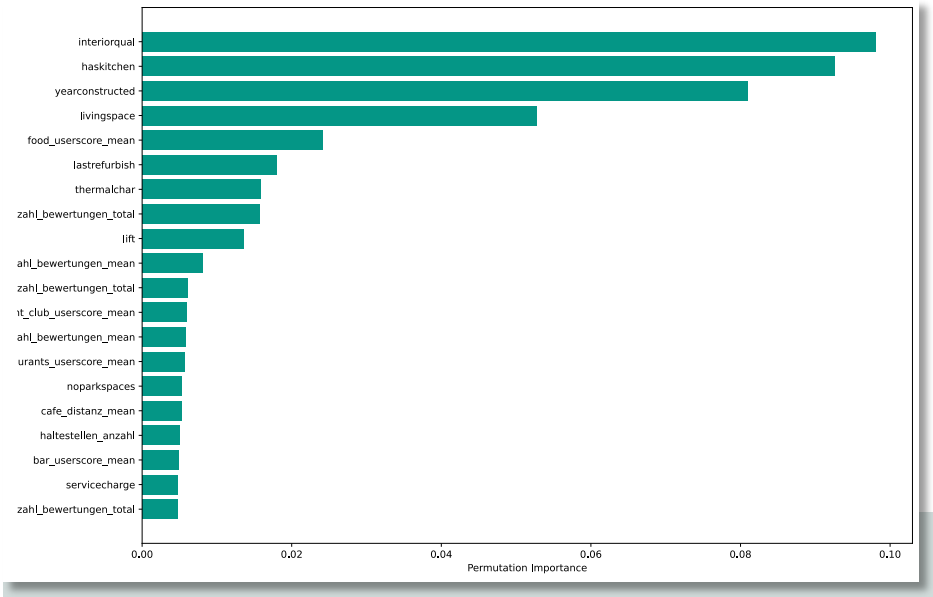

Abbildung 26: Wichtigste Faktoren beim Random Forest

Im Vergleich zu den anderen Modellen nutzt der Random Forest die meisten traditionellen Kennwerte mit einem Anteil von 60% unter den Top 20 Faktoren, was im Schnitt 10% mehr ist als beim XGBoost. Noch auffälliger ist das Verhältnis der traditionellen innerhalb der Top 10 Faktoren. Hier sind die traditionellen Kennwerte im Vergleich zu den anderen beiden Modellen überproportional vertreten mit einem Prozentsatz von 70%.

4.4 Evaluation der Modelle

Zum Zwecke der Evaluation sollen die Ergebnisse nun mit der in Kapitel 3.2.4 vorgestellten Benchmark in Form des Mietspiegels verglichen werden. Hierzu wurde vor dem Training der Modelle eine Stichprobe von fünf Mietobjekten aus dem Datensatz herausgenommen. Zu diesen fünf Mietobjekten sollten die Modelle jeweils die Kaltmieten pro m² vorhersagen. Die Ergebnisse der Benchmark wurden anschließend ebenfalls den fünf Mietobjekten hinzugefügt, sodass sich die in Tabelle 6 dargestellte Ansicht der Gesamtergebnisse inkl.

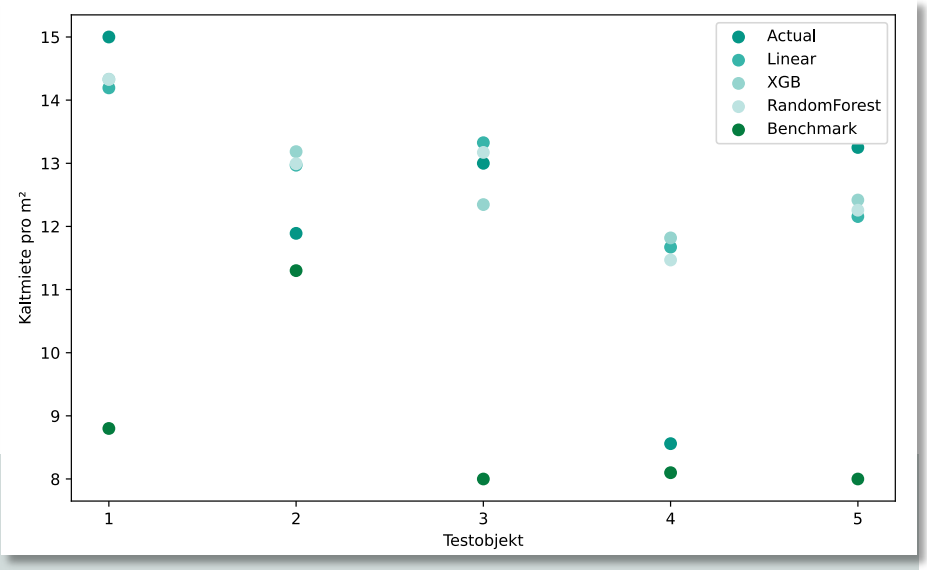

Abbildung 27: Vergleich der Modellergebnisse zur Benchmark

einiger relevanter Variablen ergibt. Abbildung 27 zeigt den Vergleich der Kaltmieten in €/m² der drei Modelle sowie der Benchmarks im Vergleich zu den tatsächlich im Immoscout Datensatz inserierten Kaltmieten pro m².

Variable	Objekt 1	Objekt 2	Objekt 3	Objekt 4	Objekt 5
Stadtteil	Altstadt	Flingern Nord	Oberkassel	Unterbilk	Stadtmitte
Baujahr	1990	2016	1971	1948	1971
Wohnfläche in m²	138,00	95,30	105,00	116,77	117,00
Anzahl Zimmer	4	3	3	3	3
cafe_distanz_bestes	754,66	746,03	125,46	516,92	348,58
bar_bewertung_mean	4,24	3,72	4,50	4,41	4,17
Kaltmiete (Immoscout)	15,00	11,89	13,00	8,56	13,25
Kaltmiete (linear Reg.)	14,19	12,97	13,33	11,67	12,16

Variable	Objekt 1	Objekt 2	Objekt 3	Objekt 4	Objekt 5
Kaltmiete (XGBoost)	14,33	13,18	12,35	11,82	12,42
Kaltmiete (Random Forest)	14,33	13,00	13,17	11,47	12,26
Kaltmiete (Benchmark)	8,80	11,30	8,00	8,10	8,00
Nebenkosten	450,00	305,00	120,00	220,00	260,00
Heizkosten	0,00	0,00	0,00	100,00	0,00
Küche inbegriffen	Ja	Nein	Nein	Nein	Nein
Zustand	Fully renovated	Mint condition	Well kept	Fully renovated	Mint condition
Qualität	Gehoben	Gehoben	Gehoben	Normal	Gehoben

Tabelle 6: Evaluation der Modellergebnisse im Detail

Abbildung 27 und Tabelle 6 kann entnommen werden, dass die Vorhersagen der Modelle eng beieinander liegen. Auffällig ist auch, dass der tatsächlich inserierte Wert bei Objekt 4 sehr nah an der Benchmark liegt, die Modelle jedoch viel höhere Werte vorhersagen. Diese Abweichung kann einerseits dadurch begründet werden, dass im Datensatz, welcher zum Training der Modelle verwendet wurde, lediglich 69 Mietobjekte enthalten sind, welche im Stadtteil Unterbilk liegen. Andererseits ist auch der Einfluss der nicht-traditionellen Werte ausschlaggebend.

Bei den übrigen Objekten fällt auf, dass die Modelle den wahren Wert für die Kaltmiete in €/m² recht gut vorhersagen. In jedem Fall liegen die vorhergesagten Werte stets über den Werten der Benchmark.

Dies lässt darauf schließen, dass die Benchmark in Form des Mietspiegels des Mietervereins Düsseldorf e.V. als erster Anhaltspunkt bei der Bewertung eines Mietobjektes dienen kann. Die vorliegenden Analysen und Auswertungen haben jedoch gezeigt, dass es sich lohnt, weitere Faktoren wie bspw. die erwähnten, nicht-traditionellen Daten zur Preisbestimmung heranzuziehen.

Eine weitere Betrachtung im Sinne der Evaluation ist der Vergleich der wichtigsten Faktoren der Modelle. Die Schnittmenge der Top 20 wird nun folgend abgeleitet. Es sei angemerkt, dass alle genannten Features hochsignifikant sind.

Lineare Regression	XGBoost	Random Forest
stadtteile_Oberkassel	livingspace	interioqual
stadtteile_Niederkassel	cafe_distanz_bestes	haskitchen
Condition_first_time_use	restaurants_distanz_bestes	yearconstructed
stadtteile_Hafen	bar_bewertung_mean	livingspace
haltstellen_distanz_mean	night_club_distanz_bestes	food_userscore_mean
cafe_anzahl	haltstellen_distanz_mean	lastrefurbish
haskitchen	night_club_distanz_mean	food_anzahl_bewertungen_total
stadtteile_Mörsenbroich	food_distanz_bestes	thermalchar
typeoffflat_penthouse	thermalchar	lift
stadtteile_Ludenberg	haltstellen_anzahl	food_anzahl_bewertungen_mean
haltestellen_anzahl	haltestellen_distanz_nah	restaurant_userscore_mean
bar_bewertung_mean	bar_anzahl_bewertung_mean	restaurant_anzahl_bewertungen_total
stadtteile_Rath	yearconstructed	night_club_userscore_mean
bar_distanz_mean	picturecount	haltestellen_anzahl
livingspace	restaurant_distanz_schlechtes	night_club_anzahl_bewertungen_mean
stadtteile_Garath	night_club_distanz_schlechtes	noparkspaces

Lineare Regression	XGBoost	Random Forest
firingtypes_district_ heating:electricity	servicecharge	cafe_distanz_mean
restaurants_distanz_bestes	haltestellen_distanz_fern	bar_userscore_mean
stadtteile_Gerresheim	bar_distanz_ bestens	restaurant_anzahl_ bewertungen_mean
bankautomaten_distanz_ mean	interiorqual	cafe_distanz_schlechtes

Tabelle 7: Top 20 Faktoren aller Regressionsmodelle

Die Schnittmenge repräsentiert die Faktoren, die bei mindestens zwei Modellen gleichermaßen vorkommen und einen großen Einfluss auf die Vorhersage haben.

Schnittmenge traditionelle Faktoren	Schnittmenge nicht-traditionelle Kennwerte
livingspace	haltstellen_distanz_mean
haskitchen	bankautomaten_distanz_mean
yearconstructed	bar_bewertung_mean
	bar_distanz_mean
	restaurants_distanz_bestes

Tabelle 8: Schnittmenge der wichtigsten Faktoren der Modelle

Der überwiegende Anteil besteht aus nicht-traditionellen Faktoren. Der einzige traditionelle Kennwert, welcher bei allen drei Modellen gleichermaßen verwendet wird, ist der Wohnraum. Den anderen Kennwert haskitchen teilen sich die lineare Regression und der Random Forest. Obwohl die Schnittmenge klein ist, kann man sagen, dass bei den beiden Modellen lineare Regression sowie der XGBoost die Distanzen zu den POIs einen hohen Einfluss auf die Vorhersagen

haben. Auffällig ist das der Random Forest im Gegensatz zu den anderen beiden Modellen die Bewertungen der POIs stark bevorzugt.

5 Fazit und Ausblick

Die vorliegende Studie stellt den Zusammenhang zwischen der Kaltmiete in €/m² von Mietobjekten und gängigen, traditionellen sowie nicht-traditionellen Faktoren vor. Es zeigt sich, dass die Einbeziehung von nicht-traditionellen Daten eines jeweiligen Mietobjektes relevant für die Prognose des Mietpreises ist. Die Auseinandersetzung mit nicht-traditionellen Daten ermöglicht eine differenziertere Sicht auf Mietpreise und eine höhere Güte der Vorhersagen als eine alleinige Berufung auf den Mietspiegel.

Der Mietspiegel der Stadt Düsseldorf stellt jedoch einen verlässlichen ersten Ansatzpunkt für die Beurteilung der Kaltmiete in €/m² eines Mietobjektes dar, welcher um weitere traditionelle sowie nichttraditionelle Daten erweitert werden sollte. Die in dieser Studie vorgestellte Herangehensweise zum Umgang mit nicht-traditionellen Daten in Bewertungsmodellen für Mietobjekte stellt einen Weg der Erweiterung dar.

Möglichkeiten der Verbesserung des hier vorgestellten Vorgehens bestehen darin, weitere Datenquellen außerhalb der Google Places API hinzuzuziehen. Zusätzlich zu den in dieser Arbeit vorgestellten, nicht-traditionellen Daten könnte es sich anbieten, weitere unstrukturierte Daten wie bspw. Text-, Bild- oder Videodaten zu analysieren und in die entwickelten Modelle aufzunehmen. So haben Ahmed und Moustafa gezeigt, dass die Einbeziehung von Textdaten, welche die Eigenschaften der Immobilien beschreiben, und Bilddaten, die Güte der Vorhersagen steigern kann (Vgl. Ahmed & Moustafa, 2016).

Anhang

Spalte	Bezeichnung
restaurants_anzahl	Anzahl Restaurants in der Nähe
restaurants_bewertung_mean	Durchschnittliche Bewertung von Restaurants in der Nähe
restaurants_distanz_bestes	Distanz zum am besten bewerteten Restaurant
restaurants_distanz_schlechtes	Distanz zum am schlechtesten bewerteten Restaurant
restaurants_distanz_mean	Durchschnittliche Distanz zu allen Restaurants
restaurants_anzahl_bewertungen_mean	Durchschnittliche Anzahl von Bewertungen
restaurants_anzahl_bewertungen_total	Gesamte Anzahl von Bewertungen
restaurants_userscore_mean	Durchschnittlicher Userscore
cafe_anzahl	Anzahl Cafés in der Nähe
cafe_bewertung_mean	Durchschnittliche Bewertung von Cafés in der Nähe
cafe_distanz_bestes	Distanz zum am besten bewerteten Café
cafe_distanz_schlechtes	Distanz zum am schlechtesten bewerteten Café
cafe_distanz_mean	Durchschnittliche Distanz zu allen Cafés
cafe_anzahl_bewertungen_mean	Durchschnittliche Anzahl von Bewertungen
cafe_anzahl_bewertungen_total	Gesamte Anzahl von Bewertungen
cafe_userscore_mean	Durchschnittlicher Userscore
bar_anzahl	Anzahl Bars in der Nähe
bar_bewertung_mean	Durchschnittliche Bewertung von Bars in der Nähe
bar_distanz_bestes	Distanz zur am besten bewerteten Bar
bar_distanz_schlechtes	Distanz zur am schlechtesten bewerteten Bar

Spalte	Bezeichnung
bar_distanz_mean	Durchschnittliche Distanz zu allen Bars
bar_anzahl_bewertungen_mean	Durchschnittliche Anzahl von Bewertungen
bar_anzahl_bewertungen_total	Gesamte Anzahl von Bewertungen
bar_userscore_mean	Durchschnittlicher Userscore
night_club_anzahl	Anzahl Nachtclubs in der Nähe
night_club_bewertung_mean	Durchschnittliche Bewertung von Nachtclubs in der Nähe
night_club_distanz_bestes	Distanz zum am besten bewerteten Nachtclub
night_club_distanz_schlechtes	Distanz zum am schlechtesten bewerteten Nachtclub
night_club_distanz_mean	Durchschnittliche Distanz zu allen Nachtclubs
night_club_anzahl_bewertungen_mean	Durchschnittliche Anzahl von Bewertungen
night_club_anzahl_bewertungen_total	Gesamte Anzahl von Bewertungen
night_club_userscore_mean	Durchschnittlicher Userscore
bankautomaten_anzahl	Anzahl Bankautomaten in der Nähe
bankautomaten_distanz_mean	Durchschnittliche Distanz zu allen Bankautomaten
bankautomaten_distanz_nah	Distanz zum nächsten Bankautomaten
bankautomaten_distanz_fern	Distanz zum entferntesten Bankautomaten
haltestellen_anzahl	Anzahl Haltestellen in der Nähe
haltstellen_distanz_mean	Durchschnittliche Distanz zu allen Haltestellen
haltstellen_distanz_nah	Distanz zur nächsten Haltestellen
haltstellen_distanz_fern	Distanz zur entferntesten Haltestellen
food_anzahl	Anzahl Essensmöglichkeiten in der Nähe

Spalte	Bezeichnung
food_bewertung_mean	Durchschnittliche Bewertung von Essensmöglichkeiten in der Nähe
food_distanz_bestes	Distanz zur am besten bewerteten Essensmöglichkeit
food_distanz_schlechtes	Distanz zur am schlechtesten bewerteten Essensmöglichkeit
food_distanz_mean	Durchschnittliche Distanz zu allen Essensmöglichkeiten
food_anzahl_bewertungen_mean	Durchschnittliche Anzahl von Bewertungen
food_anzahl_bewertungen_total	Gesamte Anzahl von Bewertungen
food_userscore_mean	Durchschnittlicher Userscore[24]

Tabelle 9: Spalten und Bezeichnungen der nicht-traditionellen Daten

Spalte	Transformierung
restaurants_anzahl	Transformierung mittels dritter Wurzel
restaurants_bewertung_mean	Keine Transformierung
restaurants_distanz_bestes	Keine Transformierung
restaurants_distanz_schlechtes	Keine Transformierung
restaurants_distanz_mean	Keine Transformierung
restaurants_anzahl_bewertungen_mean	Transformierung mittels Quadratwurzel
restaurants_anzahl_bewertungen_total	Transformierung mittels dritter Wurzel

[24] Der „Userscore" eines jeweiligen POI errechnet sich als gewichtete Summe der Spalten user_ratings_total und rating in der POI Datenbank. Die Werte der Spalte user_ratings_total werden dabei mit 0,1 multipliziert und die Werte der Spalte rating mit 0,9. Somit ergibt sich ein Wert, welcher die durchschnittliche Bewertung eines POI höher gewichtet als die Anzahl der gesamten Bewertungen.

Spalte	Transformierung
restaurants_userscore_mean	Transformierung mittels Quadratwurzel
cafe_anzahl	Transformierung mittels Quadratwurzel
cafe_bewertung_mean	Keine Transformierung
cafe_distanz_bestes	Keine Transformierung
cafe_distanz_schlechtes	Keine Transformierung
cafe_distanz_mean	Keine Transformierung
cafe_anzahl_bewertungen_mean	Keine Transformierung
cafe_anzahl_bewertungen_total	Transformierung mittels dritter Wurzel
cafe_userscore_mean	Transformierung mittels dritter Wurzel
bar_anzahl	Transformierung mittels Quadratwurzel
bar_bewertung_mean	Keine Transformierung
bar_distanz_bestes	Keine Transformierung
bar_distanz_schlechtes	Keine Transformierung
bar_distanz_mean	Keine Transformierung
bar_anzahl_bewertungen_mean	Transformierung mittels dritter Wurzel
bar_anzahl_bewertungen_total	Transformierung mittels dritter Wurzel
bar_userscore_mean	Transformierung mittels dritter Wurzel
night_club_anzahl	Transformierung mittels dritter Wurzel
night_club_bewertung_mean	Keine Transformierung
night_club_distanz_bestes	Keine Transformierung
night_club_distanz_schlechtes	Keine Transformierung
night_club_distanz_mean	Keine Transformierung

Spalte	Transformierung
night_club_anzahl_bewertungen_mean	Transformierung mittels Quadratwurzel
night_club_anzahl_bewertungen_total	Transformierung mittels dritter Wurzel
night_club_userscore_mean	Transformierung mittels Quadratwurzel
bankautomaten_anzahl	Transformierung mittels dritter Wurzel
bankautomaten_distanz_mean	Keine Transformierung
bankautomaten_distanz_nah	Transformierung mittels Quadratwurzel
bankautomaten_distanz_fern	Keine Transformierung
haltestellen_anzahl	Keine Transformierung
haltstellen_distanz_mean	Keine Transformierung
haltstellen_distanz_nah	Transformierung mittels Quadratwurzel
haltstellen_distanz_fern	Keine Transformierung
food_anzahl	Transformierung mittels dritter Wurzel
food_bewertung_mean	Keine Transformierung
food_distanz_bestes	Keine Transformierung
food_distanz_schlechtes	Keine Transformierung
food_distanz_mean	Keine Transformierung
food_anzahl_bewertungen_mean	Transformierung mittels Quadratwurzel
food_anzahl_bewertungen_total	Transformierung mittels dritter Wurzel
food_userscore_mean	Transformierung mittels Quadratwurzel

Tabelle 10: Transformierungen der nicht-traditionellen Daten

Literaturverzeichnis

Ahmed, E. H., & Moustafa, M. (2016). House price estimation from visual and textual features. IJCCI 2016 - Proceedings of the 8th International Joint Conference on Computational Intelligence, 3, 62–68. https://doi.org/10.5220/0006040700620068

Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L., & Ridella, S. (2012). The "K" in K-fold Cross Validation. ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. http://www.i6doc.com/en/livre/?GCOI=28001100967420.

Asaftei, G. M., Doshi, S., Means, J., & Sanghvi, A. (2018). Getting ahead of the market: How big data is transforming real estate. In Urban Land.

Bency, A. J., Rallapalli, S., Ganti, R. K., Srivatsa, M., & Manjunath, B. S. (2017). Beyond spatial auto-regressive models: Predicting housing prices with satellite imagery. Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017, 320–329. https://doi.org/10.1109/WACV.2017.42

Bennett, J., & Lanning, S. (2007). The Netflix Prize.

Box, G. E. P., & Cox, D. R. (1964). An Analysis of Transformations. Journal of the Royal Statistical Society: Series B (Methodological), 26(2), 211–243. https://doi.org/10.1111/j.2517-6161.1964.tb00553.x

Brauer, K.-U. (Hrsg.) (2001). Grundlagen der Immobilienwirtschaft. Gabler.

Breusch, T. S. (1978). Testing for autocorrelation in dynamic linear models*. Australian Economic Papers, 17(31), 334–355. https://doi.org/10.1111/j.1467-8454.1978.tb00635.x

Breusch, T. S., & Pagan, A. R. (1979). A Simple Test for Heteroscedasticity and Random Coefficient Variation. Econometrica, 47(5), 1287. https://doi.org/10.2307/1911963

Cerda, P., Varoquaux, G., & Kégl, B. (2018). Similarity encoding for learning with dirty categorical variables. Machine Learning, 107(8–10), 1477–1494. https://doi.org/10.1007/s10994-018-5724-2

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the ACM SIGKDD International Conference on Knowledge

Discovery and Data Mining, 13-17-Augu, 785–794. https://doi.org/10.1145/2939672.2939785

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. In Annals of Statistics (Vol. 28, Issue 2, pp. 337–407). Institute of Mathematical Statistics. https://doi.org/10.1214/aos/1016218223

Godfrey, L. G. (1978). Testing Against General Autoregressive and Moving Average Error Models when the Regressors Include Lagged Dependent Variables. Econometrica, 46(6), 1293. https://doi.org/10.2307/1913829

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. Technometrics, 12(1), 55–67. https://doi.org/10.1080/00401706.1970.10488634

Landeshauptstadt Düsseldorf, & Amt für Statistik und Wahlen. (2019). Datenblatt Wohnungsmarkt Düsseldorf 2019.

Landeshauptstadt Düsseldorf, & Stadtplanungsamt. (2011). Stadtentwicklungskonzept Düsseldorf 2020+.

Larose, D. T., & Larose, C. D. (2015). Data mining and predictive analytics (Second). Wiley.

LEG, & CBRE. (2019). LEG-Wohnungsmarktreport 2019.

McKinney, W. (2011). pandas: a Foundational Python Library for Data Analysis and Statistics. Python for High Performance and Scientific Computing, 1–9. http://pandas.sf.net

Mieterverein Düsseldorf e.V. (2019). Mietrichtwert-Tabelle, Düsseldorf.

Pedregosa, F., Michel, V., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Vanderplas, J., Cournapeau, D., Varoquaux, G., Gramfort, A., Thirion, B., Dubourg, V., Passos, A., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. In Journal of Machine Learning Research (Vol. 12). http://scikit-learn.sourceforge.net.

Petkov, M. (2020). Evaluation of spatial data's impact in mid-term room rent price through application of spatial econometrics and machine learning - Case Study: Lisbon. https://run.unl.pt/handle/10362/93716

Rahm, E., & Do, H. (2000). Data cleaning: Problems and current approaches. IEEE Data Eng. Bull., 23(4), 3–13. http://dbs.uni-leipzig.de

Raji, C. G., Gafoor, A., Ahammed, H., Edavalath, A., & Cijas, P. K. (2020). WeGo: An efficient travel assistant application using android. Proceedings of the 4th International Conference on IoT in Social, Mobile, Analytics and Cloud, ISMAC 2020, 594–598. https://doi.org/10.1109/I-SMAC49090.2020.9243482

Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. Computing in Science and Engineering, 13(2), 22–30. https://doi.org/10.1109/MCSE.2011.37

Verbeek, M. (2017). A guide to modern econometrics (5th editio). John Wiley & Sons.

kostenloser Download
unter **fom-ifes.de**

Lehrbass, F. (2021): Deep Learning Diagnostics – How to Avoid Being Fooled by TensorFlow, PyTorch, or MXNet with the Help of Modern Econometrics, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 24, 2021, ISSN (eBook) 2569-5355, ISBN (eBook) 978-3-89275-424-4

Lehrbass, F. / Wörndl, F. (2021): Was treibt die Renditen von Hedgefonds? Eine empirische Untersuchung ausgewählter Hedgefonds Strategien, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 23, 2021, ISSN (eBook) 2569-5355, ISBN (eBook) 978-3-89275-422-0

Kladroba, A. / Friz, K. / Buchmann, T. / Wolf, P. (2020): Netzwerk- und Outputmessung – Indikatorik für transformative Technologiefelder (NEO-Indikatorik), in: Krol, B. / Kladroba, A. (Hrsg.), ifes Schriftenreihe, Band 22, 2020, ISSN (eBook) 2569-5355, ISBN (eBook) 978-3-89275-420-6

Bähren, T. / Maasjosthusmann, R. / Walter, A. / Lehrbass, F. (2020): Praktische Umsetzung von Business Analytics im Mediensektor: Predictive Analytics im Filmgeschäft, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 21, 2020, ISSN (eBook) 2569-5355, ISBN (eBook) 978-3-89275-418-3

Kladroba, A. (2019): Der Einfluss mathematischer Methoden auf das Ergebnis von Mannschaftswettkämpfen: Eine Simulationsrechnung, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 20, 2019, ISSN (eBook) 2569-5355, ISBN (eBook) 978-3-89275-416-9

Raasch, A. / Lehrbass, F. (2019): Investmentstrategien im Rahmen von Übernahmen börsennotierter Gesellschaften – Merger Arbitrage und Maschinelles Lernen, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 19, 2019, ISSN 2191-3366, ISBN 978-3-89275-413-8

Hagemann, D. / Lehrbass, F. (2018): Prognosemodelle für Länderrisiken: Logit- und Deep Learning-Methoden im Vergleich, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 18, 2018, ISSN 2191-3366, ISBN 978-3-89275-411-4

Graalmann, M.-P. / Lehrbass, F. (2018): Eignung von Varianz-Kovarianz-Ansätzen und Copula-Modellen zur Risikoaggregation in bankaufsichtlichen Risikotragfähigkeitskonzepten, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 17, 2018, ISSN 2191-3366, ISBN 978-3-89275-409-1

Cox, P. / Lehrbass, F. (2018): Determinanten der Replikationsgüte von Exchange Traded Funds, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 16, 2018, ISSN 2191-3366, ISBN 978-3-89275-407-7

Lehrbass, F. / Scheipers, N. (2017): Determinanten der Höhe von Wirtschaftsprüfungshonoraren am Beispiel von gelisteten Unternehmen im Prime Standard, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 15, 2017, ISSN 2191-3366, ISBN 978-3-89275-406-0

Schwarz, J. (2017): Ergebnisse der Analyse von Studienabbrüchen, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 14, 2017, ISSN 2191-3366, ISBN 978-3-89275-405-3

Lehrbass, F. (2016): Risikomessung für den globalen Kohlehandel: Einfache und fortgeschrittene Verfahren nebst Backtesting sowie ein Vergleich mit IFRS 7, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 13, 2016, ISSN 2191-3366, ISBN 978-3-89275-404-6

Godbersen, H. (2016): Die Means-End Theory of Complex Cognitive Structures – Entwicklung eines Modells zur Repräsentation von verhaltensrelevanten und komplexen Kognitionsstrukturen für die Wirtschafts- und Sozialwissenschaften, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 12, 2016, ISSN 2191-3366, ISBN 978-3-89275-403-9

Seng, A. / Landherr, G. (2015): Vielfalt leben und Vielfalt gestalten – Diversity Management in der Lehre, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 11, 2015, ISSN 2191-3366, ISBN 978-3-89275-402-2

Gansser, O. A. / Schutkin, A. (2014): Studie zur Validierung der Persönlichkeitsmerkmale Abenteuerlust und Routineverhalten, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 10, 2014, ISSN 2191-3366, ISBN 978-3-89275-401-5

Gansser, O. A. (2014): Marketingplanung als Instrument zur Krisenbewältigung, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 9, 2014, ISSN 2191-3366, ISBN 978-3-89275-400-8

Runia, P. M. / Wahl, F. / Rüttgers, C. (2013): Das Markenimage von Hersteller- und Handelsmarken: Eine empirische Analyse der Imagekomponenten von Körperpflegemarken auf der Grundlage eines Markenidentitätskonzeptes, in: Krol, B. (Hrsg.), KCS Schriftenreihe, Band 8, 2013, ISSN 2191-3366

Naskrent, J. / Rüttgers, C. (2013): Sportmonitor Essen 2013: Eine empirische Analyse über das Image regionaler Sportvereine und ihre Sponsoring- und Promotionangebote, in: Krol, B. (Hrsg.), KCS Schriftenreihe, Band 7, 2013, ISSN 2191-3366

Seng, A. / Fiesel, L. / Rüttgers, C. (2013): Akzeptanz der Frauenquote, in: Krol, B. (Hrsg.), KCS Schriftenreihe, Band 6, 2013, ISSN 2191-3366

Naskrent, J. / Rüttgers, C. (2012): Wahrnehmung von Werbung mit Sportereignisbezug: Eine empirische Analyse der Einschätzung von Sponsoring und Ambush-Marketing im Rahmen der Fußball-Europameisterschaft und der Olympischen Spiele im Jahr 2012, in: Krol, B. (Hrsg.), KCS Schriftenreihe, Band 5, 2012, ISSN 2191-3366

Seng, A. / Fiesel, L. / Krol, B. (2012): Erfolgreiche Wege der Rekrutierung in Social Networks, in: Krol, B. (Hrsg.), KCS Schriftenreihe, Band 4, 2012, ISSN 2191-3366

Heinemann, S. / Krol, B. (2011): Nachhaltige Nachhaltigkeit: Zur Herausforderung der ernsthaften Integration einer angemessenen Ethik in die Managementausbildung, in: Krol, B. (Hrsg.), KCS Schriftenreihe, Band 2, 2011, ISSN 2191-3366

Hermeier, B. / Rettig, P. / Krol, B. (2010): Marken- und Produktmanagement durch Nutzung von Sportgroßereignissen: Möglichkeiten und Grenzen für Industrie und Handel, in: Krol, B. (Hrsg.), KCS Schriftenreihe, Band 1, 2010, ISSN 2191-3366

ISBN (Print) 978-3-89275-425-1 ISSN (Print) 2191-3366
ISBN (eBook) 978-3-89275-426-8 ISSN (eBook) 2569-5355

Institut für Empirie & Statistik
der FOM Hochschule
für Oekonomie & Management

FOM Hochschule

FOM. Die Hochschule. Für Berufstätige.

Die mit bundesweit über 57.000 Studierenden größte private Hochschule Deutschlands führt seit 1993 Studiengänge für Berufstätige durch, die einen staatlich und international anerkannten Hochschulabschluss (Bachelor/Master) erlangen wollen.

Die FOM ist der anwendungsorientierten Forschung verpflichtet und verfolgt das Ziel, adaptionsfähige Lösungen für betriebliche bzw. wirtschaftsnahe oder gesellschaftliche Problemstellungen zu generieren. Dabei spielt die Verzahnung von Forschung und Lehre eine große Rolle: Kongruent zu den Masterprogrammen sind Institute und KompetenzCentren gegründet worden. Sie geben der Hochschule ein fachliches Profil und eröffnen sowohl Wissenschaftlerinnen und Wissenschaftlern als auch engagierten Studierenden die Gelegenheit, sich aktiv in den Forschungsdiskurs einzubringen.

Weitere Informationen finden Sie unter **fom.de**

ifes

Zunehmende Digitalisierung erfordert und ermöglicht datenbasierten Erkenntnisgewinn und fundiertes unternehmerisches Handeln. Um aus den allgegenwärtigen Daten die richtigen Schlüsse zu ziehen, ist überall eine kritische Methodenkompetenz erforderlich. Der wissenschaftliche Fokus der ifes-Akteure liegt dabei in den Bereichen der empirischen Unternehmens-, Markt- und Konsumentenforschung, der angewandten Statistik, des Data Minings und der Finanzstatistik.

Das ifes verfolgt das Ziel, empirische Kompetenzen an der FOM zu bündeln und die angewandte Forschung im empirischen Bereich der Hochschule weiter voranzutreiben. Damit nimmt das ifes eine zentrale Stellung im Bereich der Entwicklung und Unterstützung der Methodenausbildung in der Lehre der Bachelor- und Masterstudiengänge sowie im Promotionsprogramm der FOM ein.

Weitere Informationen finden Sie unter **fom-ifes.de**

SCIENCE BLOG

Im Forschungsblog werden unter dem Titel „FOM forscht“ Beiträge und Interviews rund um aktuelle Forschungsthemen und -aktivitäten der FOM Hochschule veröffentlicht.

Besuchen Sie den Blog unter fom-blog.de