



Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Informatik

Masterarbeit

Untersuchung von Gentrifizierung am Beispiel Berlin mittels Big Data Analytics

Dennis Helweg

5helweg@informatik.uni-hamburg.de // dennishelweg@gmx.de

Studiengang IT-Management und –Consulting

Matrikelnummer: 6827075

Erstgutachter: Professor Dr. Stefan Voß

Zweitgutachter: Dr. Robert Stahlbock

Abgabe:

Hamburg, 19. September 2018

Denn die einen sind im Dunkeln
Und die andern sind im Licht.
Und man siehet die im Lichte
Die im Dunkeln sieht man nicht.

(Bertolt Brecht, Dreigroschenoper)

Untersuchung von Gentrifizierung am Beispiel Berlin mittels Big Data Analytics

Dennis Helweg

Zusammenfassung

In dieser Arbeit wird die Gentrifizierung in Berlin mittels Big Data Analytics untersucht. Es werden grundlegende Theorien und Definitionen zu Gentrifizierung vorgestellt, sowie der Stand der Forschung zur Untersuchung von Gentrifizierung mit Big Data erörtert. Im Speziellen wird ein zeitlicher Zusammenhang zwischen dem Wandel des sozialen Status und der lokalen Angebotsstruktur (Bars, Restaurants, Cafés, etc.) untersucht. Im Ergebnis wird ein starker Zusammenhang zwischen der lokalen Angebotsstruktur und dem sozialen Status in Berlin nachgewiesen. Dabei korrelieren Restaurants negativ und Fast-Food Shops positiv mit dem Anteil der Transferleistungsempfänger an der Bevölkerung in einem Gebiet. Mit verschiedenen Machine Learning Algorithmen werden Indizien gefunden, die für einen zeitlichen Zusammenhang zwischen dem Wandel des sozialen Status und einer darauf folgenden Änderung der lokalen Angebotsstruktur sprechen. Zum Abschluss wird diskutiert, dass dies ein Anzeichen für die erste Phase des doppelten Invasions-Sukzessions-Zyklus nach Dangschat (1988) darstellen kann, dem eine zweite Phase der Verdrängung einer Mittelschicht folgen kann.

Für diese Untersuchung wird methodisch nach dem CRISP-DM Referenzmodell vorgegangen. Die Arbeit zeigt, wie räumliche Datenquellen ausgewertet werden können, welche typischen Herausforderungen bei räumlichen Daten zu beachten sind, und welche Datenintegrations-schritte für eine Auswertung notwendig sind. Es werden mehrere Datenquellen anhand verschiedener Kriterien, wie Zeitbezug, technische Auswertbarkeit und Lizenzrecht analysiert. Für das Trainieren von verschiedenen Machine Learning Algorithmen werden Daten aus OpenStreetMap mit Daten des Berichts Monitoring Soziale Stadtentwicklung Berlin zusammengeführt. In einem zweistufigen Modellierungs-Evaluations-Zyklus wird die beste Datenkombination für die Modellierung evaluiert, die anschließend für das Trainieren von Modellen für die Bewertung von verschiedenen Gentrifizierungs-Hypothesen genutzt wird.

Measurement of Gentrification in Berlin via Big Data Analytics

Dennis Helweg

Abstract

In this thesis, gentrification in Berlin is examined via big data analytics. Basic theories and definitions of gentrification are presented, as well as the state of research on the study of gentrification with big data are discussed. In particular, a temporal connection between the change in social status and the neighbourhood facilities (bars, restaurants, cafés, etc.) is examined. One result is a correlation between the neighbourhood facilities and social status in Berlin. Restaurants are negatively and fast-food shops positively correlated with the share of welfare recipients in the population in an area. Various machine learning algorithms are used to find signals that indicate a temporal connection between the change in social status and a subsequent change in the local supply structure. Finally, it is discussed that this may be an indication of the first phase of Dangschat's (1988) double invasion-succession-cycle, which may be followed by a second phase of middle-class displacement.

Methodologically, this method is based on the cross-industry standard process for data mining (CRISP-DM). The paper shows how spatial big data sources can be evaluated, which typical spatial data challenges are to be considered, and which data integration steps are necessary for the study. Several data sources are analyzed on the basis of various criteria, such as time relevance, technical evaluability and licensing agreements. To train different machine learning algorithms, data from OpenStreetMap is merged with data from the report Monitoring Social Urban Development Berlin. In a two-stage modeling evaluation cycle, the best data combination for modeling is evaluated, which is then used to train models for the assessment of different gentrification hypotheses.

Inhaltsverzeichnis

Abbildungsverzeichnis	VII
Tabellenverzeichnis.....	XI
Abkürzungsverzeichnis	XIII
1 Einleitung	1
1.1 Ziel und Forschungsfragen	2
1.2 Methodik und Aufbau.....	2
1.3 Open Data und Open Knowledge	5
2 Grundlagen.....	6
2.1 Big Data Analytics – neue Datenquellen und Technologien.....	6
2.2 Spatial Big Data – Besonderheiten raumbezogener Daten.....	8
2.3 Anwendungsfälle für Spatial Big Data Analytics.....	10
3 Business Understanding – Gentrifizierung	13
3.1 Definition.....	13
3.2 Gentrifizierung als Prozess	15
3.3 Gentrifizierung in Berlin.....	16
3.4 Quantitative Analyse der Gentrifizierung in Berlin.....	18
3.5 Big Data Analysen zu Gentrifizierung	19
4 Data Understanding – Analyse potentieller Datenquellen.....	25
4.1 Adressen und Räume	25
4.2 Datenquellen	29
4.3 Prüfung auf Lizenzrecht und Open Data	38
4.4 Auswahl Datenquellen und Betrachtungszeitraum.....	40
4.5 Ableitung domänenspezifischer Forschungsfragen und -hypothesen	42
5 Data Preparation – Systemaufbau und Datenintegration	44
5.1 Big Data System	44
5.2 Datenintegration	47
6 Modeling & Evaluation.....	63

6.1	Tool- und Featureauswahl	63
6.2	Versuchsaufbau.....	64
6.3	Überprüfung der Gentrifizierungs-Hypothesen	71
7	Diskussion	79
7.1	Data Understanding	79
7.2	Data Preparation	80
7.3	Modeling.....	82
7.4	Evaluation – Rückschlüsse auf Gentrifizierung und Business Understanding.....	83
8	Fazit.....	86
9	Anhang	89
9.1	Anwendungsfälle	89
9.2	Gentrifizierung.....	94
9.3	Data Understanding	95
9.4	Data Prepararion – Datentransformationen	111
9.5	Modellbildung und –evaluation.....	156
9.6	Geographische Abbildungen	160
	Literaturverzeichnis.....	168
	Eigenständigkeitserklärung	184

Abbildungsverzeichnis

Abbildung 1-1: Phasen des CRISP-DM Referenzmodells (CRISP-DM consortium, 2000, S. 10)	3
Abbildung 3-1: Doppelter Invasions-Sukzessions-Zyklus (Dangschat, 1988, S. 281)	15
Abbildung 3-2: Entwicklung der Angebotsmietpreise bei ImmobilienScout24 2008-2016 (Eigene Darstellung)	17
Abbildung 3-3: Formel zur Offering Advantage. (Eigene Darstellung nach Venerandi et al., 2015)	20
Abbildung 3-4: Genauigkeit Klassifikation von Verdrängung mittels OSM&Foursquare (Venerandi et al., 2015, S. 260).....	21
Abbildung 3-5: POI-Kategorien und Spearman-Korrelationen zu Verdrängungssindikator IMD (Venerandi et al., 2015, S. 261)	22
Abbildung 3-6: Spearman-Korrelation zwischen Kennzahlen von Geo-Sozialem Netzwerk und Verdrängungssindikator IMD (Hristova et al., 2016)	23
Abbildung 4-1: Weltkarte Geohash (Veness, 2018)	26
Abbildung 4-2: Box-Whisker-Plot der Einwohnerzahlen zum 31.12.2016 in den 443 Planungsräumen	28
Abbildung 4-3: Vergleich Haselhorst und Siemensstadt (links, ImmobilienScout24) mit Bezirksregionen (rechts, eigene Darstellung mit Google MyMaps)	33
Abbildung 4-4: OSM Statistik – Kumulierte Anzahl erstellter Elemente (OpenStreetMap contributors, o.D.)	36
Abbildung 4-5: XML-Code Planet.osm Beispiel Node (Eigene Darstellung nach OpenStreetMap contributors, o.D.).....	37
Abbildung 4-6: Entwicklung der Anzahl der OSM-POI in Berlin nach Tag-Key.	38
Abbildung 5-1: Lambda Architektur (mapr.com, o.D.)	44
Abbildung 5-2: Übersicht Hadoop System Komponenten (In Anlehnung an Hortonworks, o.D.b)	45
Abbildung 5-3: MapReduce vs. TEZ (Hortonworks, 2013)	46
Abbildung 5-4: Hadoop Architektur (Doe, 2018).....	46
Abbildung 5-5: Osmosis-Aufruf zum Extrahieren der POI-Nodes	48

Abbildung 5-6: Hive-Code für die Erzeugung einer Node-Tabelle auf Basis einer OSM-Datei	48
Abbildung 5-7: Hive-Code für das Auslesen der Tags und Bestimmung des Planungsraumes per UDF	49
Abbildung 5-8: Klassendiagramm UDF lormapper	49
Abbildung 5-9: Hive-Code zu osm_poi_changed (vereinfachter Auszug).....	52
Abbildung 5-10: Hive-Code zu osm_poi_state_basis (vereinfachter Auszug).....	53
Abbildung 5-11: Hive-Code zu osm_poi_features_domain_piv (vereinfachter Auszug)	56
Abbildung 5-12: Hive-Code zu lor_own_idx_k11	57
Abbildung 5-13: Formeln zur Einwohnergewichtung der Exportkennzahlen	59
Abbildung 5-14: Formel und Graph des Gewichtungsfaktors	60
Abbildung 5-15: OA Berechnung Typ über alle POIs.....	61
Abbildung 5-16: OA Berechnung Typ innerhalb von Domäne	62
Abbildung 6-1: ROC Beispiel (Seong Ho Park, Jin Mo Goo & Chan-Hee Jo, 2004)	68
Abbildung 6-2: Gemittelte Güte der Algorithmen	71
Abbildung 6-3: Übersicht der Güte der Hypothesen.....	71
Abbildung 6-4: Grafische Aufbereitung des logistischen Regressionsklassifikators für H1/BZR (Eigene Darstellung mit Google MyMaps)	72
Abbildung 7-1: Zeitverlauf der POI-Kennzahlen in Berlin	80
Abbildung 9-1: Karte des Cholera-Ausbruchs in Soho, London (Wikipedia, 2018b)	89
Abbildung 9-2: Screenshot Parkling App (Parkling, o.D.)	92
Abbildung 9-3: Postleitzahlen in Berlin (Eigene Darstellung mit Google MyMaps nach Amt für Statistik Berlin-Brandenburg).....	95
Abbildung 9-4: Vergleich ImmobilienScout24 (oben, ImmobilienScout24) mit Prognoseräumen (unten, Eigene Darstellung mit Google MyMaps nach Amt für Statistik Berlin-Brandenburg).....	96
Abbildung 9-5: Vergleich Kreuzberg ImmobilienScout24 (links, ImmobilienScout24) mit Prognoseräumen (rechts, eigene Darstellung mit Google MyMaps nach Amt für Statistik Berlin-Brandenburg)	97
Abbildung 9-6: XML-Code Planet.osm Beispiele (Eigene Darstellung nach OpenStreetMap contributors, o.D.).....	101

Abbildung 9-7: Screenshot Ambari	107
Abbildung 9-8: Hive-Code zu osm_poi_yyyy	116
Abbildung 9-9: Hive-Code zu osm_poi_type_yyyy	122
Abbildung 9-10: Java Code zum UDF lormapper – Klasse Planungsraum.....	123
Abbildung 9-11: Java Code zum UDF lormapper – Klasse LorArea	124
Abbildung 9-12: Java Code zum UDF lormapper – Klasse AreaFactory.....	125
Abbildung 9-13: Java Code zum UDF lormapper – Klasse PointOfInterest	126
Abbildung 9-14: Hive-Code zu osm_poi_changed.....	129
Abbildung 9-15: Hive-Code zu osm_poi_state_basis.....	135
Abbildung 9-16: Hive-Code zu osm_poi_state_changed_special	135
Abbildung 9-17: Hive-Code zu osm_poi_state_changed_del.....	135
Abbildung 9-18: Hive-Code zu osm_poi_state.....	136
Abbildung 9-19: Hive-Code zu osm_poi_features_type.....	137
Abbildung 9-20: Hive-Code zu lor_ewr_data.....	140
Abbildung 9-21: Hive-Code der Strecke zu lor_own_idx_plr_tb.....	144
Abbildung 9-22: Hive-Code zu lor_mss_idx_bzr_z	145
Abbildung 9-23: Hive-Code zu lor_mss_idx_bzr_idx	146
Abbildung 9-24: Hive-Code zu lor_ewr_einwohnergewichtet	147
Abbildung 9-25: MSS Vergleich Klassengrößen.....	148
Abbildung 9-26: MSS Klassengrenzen	148
Abbildung 9-27: Hive-Code zu result_full_plr (abgekürzt).....	151
Abbildung 9-28: Klassendiagramm Distanzberechnung	151
Abbildung 9-29: Java Code zur Distanzberechnung – Klasse LorCalculator.....	153
Abbildung 9-30: Java Code zur Distanzberechnung – Klasse AreaDistance	155
Abbildung 9-31: Hive-Code zu osm_poi_features_domain_piv_distcalc (Auszug).....	156
Abbildung 9-32: ROC Beispiel	156
Abbildung 9-33: Ausgewählte Algorithmen.....	157
Abbildung 9-34: Übersicht Evaluation der Algorithmen.....	159
Abbildung 9-35: Übersicht Berlin in Bezirksregionen 2016: MSS-Status (Eigene Darstellung mit GoogleMyMaps)	160
Abbildung 9-36: Übersicht Berlin in Bezirksregionen 2016: MSS-Dynamik (Eigene Darstellung mit GoogleMyMaps).....	161

Abbildung 9-37: Übersicht Berlin in Bezirksregionen 2016: Dynamik Index nach (Döring & Ulbricht, 2016) (Eigene Darstellung mit GoogleMyMaps)	162
Abbildung 9-38: Übersicht Berlin in Bezirksregionen 2016: OA Restaurants in Größenklassen (Eigene Darstellung mit GoogleMyMaps)	163
Abbildung 9-39: Übersicht Berlin in Bezirksregionen 2016: OA FastFood in Größenklassen (Eigene Darstellung mit GoogleMyMaps)	164
Abbildung 9-40: Übersicht Berlin in Bezirksregionen 2016: OA Spielotheken in Größenklassen (Eigene Darstellung mit GoogleMyMaps)	165
Abbildung 9-41: Übersicht Berlin in Bezirksregionen 2016: OA Soziales in Größenklassen (Eigene Darstellung mit GoogleMyMaps)	166
Abbildung 9-42: Übersicht Berlin in Bezirksregionen 2016: Geographische Darstellung SimpleLogistic Klassifikator (Eigene Darstellung mit GoogleMyMaps)	167

Tabellenverzeichnis

Tabelle 3-1: Übersicht Big Data Analysen zu Gentrifizierung	24
Tabelle 4-1: Alternative Positionsdarstellungen	25
Tabelle 4-2: Geohash Zellengröße, Beispiel Bundestag (Veness, 2018).....	26
Tabelle 4-3: Kontextindikatoren des MSS nach Handlungsfeld (Eigene Darstellung nach Senatsverwaltung für Stadtentwicklung und Wohnen, 2017).....	31
Tabelle 4-4: Übersicht APIs der kommerziellen Karten- und Empfehlungsdienste	34
Tabelle 4-5: Analyse Terms of Use der POI-Quellen.....	39
Tabelle 4-6: Übersicht Datenquellen	40
Tabelle 4-7: Auswahl des Betrachtungszeitraums	41
Tabelle 5-1: Tabellenaufbau osm_poi_type_yyyy	50
Tabelle 5-2: Beispiele POI in den Zeitscheiben.....	51
Tabelle 5-3: Beispiele Namensänderungen.....	53
Tabelle 5-4: Tabellenaufbau osm_poi_state	54
Tabelle 5-5: Tabellenaufbau osm_poi_features_domain_piv (vereinfachter Auszug)	56
Tabelle 5-6: Tabellenaufbau lor_own_idx_plr_tb (Auszug).....	58
Tabelle 5-7: OA Kennzahlen	62
Tabelle 6-1: Übersicht Daten für Modellbildung.....	64
Tabelle 6-2: Übersicht der Arten von Ensembles	65
Tabelle 6-3: Konfusionsmatrix Beispiel	67
Tabelle 6-4: Kennzahlen für Klasse a	67
Tabelle 6-5: Beispiel Gütekriterien	67
Tabelle 6-6: Raumbezugsgröße - AUC.....	69
Tabelle 6-7: Raumbezugsgröße - F1	69
Tabelle 6-8: Featuregruppe – AUC & F1	69
Tabelle 6-9: Index - AUC	70
Tabelle 6-10: Index - F1.....	70
Tabelle 6-11: H1 – Zusammenhang zwischen Angebotsstruktur und sozialem Status	73
Tabelle 6-12: H1a Korrelationen Cafés und MSS-Status-Index.....	74
Tabelle 6-13: H1b Korrelationen Fast Food und MSS-Status-Index.....	75

Tabelle 6-14: H1c Korrelationen Sport und MSS-Status-Index	75
Tabelle 6-15: Korrelationen H3b - Soziale Dynamik hat Einfluss auf Änderung der Angebotsstruktur.....	77
Tabelle 6-16: Übersicht Bewertung der Hypothesen.....	78
Tabelle 9-1: Liste der ImmobilienScout24 Quarter in Berlin (Immobilien Scout GmbH, o.D.a).....	97
Tabelle 9-2: Durchschnittliche Angebotsmietpreise in EUR (Daten von ImmobilienScout24, Quelle: Rundfunk Berlin-Brandenburg (2016)).....	102
Tabelle 9-3: Veränderung der durchschnittlichen Angebotsmietpreise gegenüber dem Vorjahr (Daten von ImmobilienScout24, Quelle: Rundfunk Berlin-Brandenburg (2016))	103
Tabelle 9-4: Auszug aus „Bestand an Kraftfahrzeugen und Kraftfahrzeuganhängern nach Gemeinden (FZ3)“ (Kraftfahrt- Bundesamt, o.D.)	104
Tabelle 9-5: POI-Hierarchie.....	111
Tabelle 9-6: Tabellenaufbau osmnodes_filtered_yymmdd.....	115
Tabelle 9-7: Tabellenaufbau osm_poi_yyyy.....	116
Tabelle 9-8: Tabellenaufbau osm_poi_changed	129
Tabelle 9-9: Tabellenaufbau osm_poi_features_type	137
Tabelle 9-10: Tabellenaufbau lor_ewr_data	140
Tabelle 9-11: Tabellenaufbau lor_dist_planungsraum.....	155
Tabelle 9-12: Algorithmus AUC & F1	158

Abkürzungsverzeichnis

Kürzel	Bedeutung
API	Programmierschnittstelle (application programming interface)
AUC	Area Under the Curve
BMVI	Bundesministerium für Verkehr und digitale Infrastruktur
BZR	Bezirksregion
CC-BY-3.0	Creative Commons Namensnennung 3.0
CRISP-DM	Cross-industry standard process for data mining
EWR	Einwohnerregister
FN	False Negative
FP	False Positive
GeoZG	Geodatenzugangsgesetz
GIS	Geoinformationssysteme
GPS	Global Positioning System
HDFS	Hadoop Filesystem
HDP	Hortonworks Data Platform
IFG	Informationsfreiheitsgesetz
IMD	Index of Multiple Deprivation
IoT	Internet of Things
IT	Informationstechnologie
KDDM	Knowledge Discovery and Data Mining
LMT	Logistic Model Tree
LOR	Lebensweltlich orientierten Räume
MAUP	Modifiable Area Unit Problem
MSS	Monitoring Soziale Stadtentwicklung
NoSQL	Not only SQL
OA	Offering Advantage
ODbL	Open Data Commons Open Database License
OSM	OpenStreetMaps
PDDL	Open Data Commons Public Domain Dedication and License
PLR	Planungsraum
POI	Point of Interest
PRG	Prognoseraum
ROC	Receiver Operating Characteristic
SQL	Structured Query Language
TN	True Negative
TP	True Positive
UDAF	User-defined Aggregation Function
UDF	User-defined Function
URL	Internetadresse
VM	Virtuelle Maschine
WGS84	World Geodetic System 1984

1 Einleitung

Im April 2018 gingen in Berlin 13.000 Menschen auf die Straßen und demonstrierten „gegen die Verdrängung und Mietwahnsinn“ (ZEIT ONLINE, 2018). Hintergrund waren steigende Mieten für Immobilien. So sind die Mietpreise in Berlin beim Portal ImmobilienScout24 von 2012 bis 2016 jedes Jahr um ca. 5,5% gestiegen (Rundfunk Berlin-Brandenburg, 2016).¹ In Zukunft sind weitere Mietsteigerungen zu erwarten, da die Steigerungen der Mieten deutlich geringer ausfallen als die der Immobilienpreise, welche sich in Berlin innerhalb eines Jahres von 2016 auf 2017 um fast 21% erhöht haben (Knight Frank LLP, 2017). Auch in anderen nationalen und internationalen Großstädten ist dieses Phänomen zu beobachten. Laut Studie von Knight Frank LLP sind unter den internationalen Top-10 der Städte mit der höchsten Steigerung der Immobilienpreise weltweit vier deutsche Städte vertreten. Der globale Spitzenreiter ist Berlin auf dem ersten Platz mit 20,5%, gefolgt von Hamburg (Platz 7 – 14,1%), München (Platz 8 – 13,8%) und Frankfurt am Main (Platz 10 – 13,4%).

Durch die steigenden Mieten findet ein Wandel einer Nachbarschaft statt. Der Teil der einkommensschwachen Bevölkerung, die sich die höheren Mieten nicht leisten kann, wird von einkommensstarken Zuzüglern² verdrängt. Diese Kombination aus Wertsteigerung der Immobilien und Verdrängung von Bewohnern findet sich in gängigen Definitionen von Gentrifizierung wieder (Holm & Schulz, 2016, S. 298–301). Mit dem Gentrifizierungsmodell entwickelten Holm und Schulz ein Messmodell, welches diese beiden Aspekte der Gentrifizierung, die *immobilienwirtschaftliche Aufwertung* und die *verdrängungsinduzierte soziale Aufwertung*, quantifiziert und im Beispiel von Berlin auf Nachbarschaftsebene messbar macht. Als dritte Komponente der Gentrifizierung wird die *Veränderung der lokalen Angebotsstruktur* aufgeführt. Damit ist ein Wandel des Angebots an umliegenden Restaurants, Cafés, Einkaufsmöglichkeiten und weiteren Lokalitäten gemeint. Diese Veränderung wurde im Gentrifizierungsmodell nicht näher betrachtet.

Neben solchen quantitativen Analysen der Gentrifizierung auf Basis von klassischen Datenerhebungen, wurde diese in anderen Forschungsarbeiten mit Big Data Analysen untersucht. Dabei wurden unter anderem Tweets und Restaurantkritiken auf Hinweise zur Gentrifizierung in den USA analysiert (Schaefer, 2014; Zukin, Lindeman & Hurson, 2015). Andere Autoren haben Zusammenhänge zwischen sozialer Verdrängung und der

¹ Durchschnittswert. Minimum bei 4,4%, Maximum bei 6,5%

² Aus Gründen der besseren Lesbarkeit wird in dieser Arbeit jeweils die gebräuchlichere Sprachform gewählt. Sämtliche Personenbezeichnungen sind wertungsfrei und gelten sowohl für das männliche als auch für das weibliche Geschlecht.

Angebotsstruktur in London erforscht. Dazu nutzten sie Daten von Foursquare, einem Empfehlungsdienst für Restaurants (Hristova, Williams, Musolesi, Panzarasa & Mascolo, 2016; Venerandi, Quattrone, Capra, Quercia & Saez-Trumper, 2015).

1.1 Ziel und Forschungsfragen

Ziel dieser Arbeit ist es, basierend auf den Forschungserkenntnissen zu Gentrifizierung und Big Data, sowie den quantifizierbaren Modellen zu dessen Messung, die Gentrifizierung in Berlin mittels Big Data Analytics zu untersuchen. Dabei liegt der Fokus auf der Veränderung der lokalen Angebotsstruktur und ihrem zeitlichen Zusammenhang mit der Veränderung des sozialen Status.

In diesem Zusammenhang sollen die folgenden allgemeinen Forschungsfragen analysiert werden, die später durch domänenspezifische Forschungsfragen und Hypothesen zur Gentrifizierung ergänzt werden:

- Welche Datenquellen sind für die Analyse von Gentrifizierung nutzbar?
- Wie können die Daten integriert werden?
- Welche Features aus den Daten eignen sich zum Aufbau eines Prognosemodells für Gentrifizierung?
- Welche Algorithmen eignen sich zum Aufbau eines Prognosemodells für Gentrifizierung?

1.2 Methodik und Aufbau

In dieser Arbeit werden zunächst in Kapitel 2 die Grundlagen zu Big Data und Spatial Big Data, also Big Data mit Raumbezug, erörtert. Im Anschluss daran werden beispielhafte Anwendungsfälle zu Spatial Big Data vorgestellt.

Anschließend wird in Kapitel 3 auf den Anwendungsfall der Gentrifizierung eingegangen. Es werden die grundlegenden Definitionen und Theorien zu dem Thema vorgestellt. Am Ende von Kapitel 3 wird der aktuelle Stand der Forschung zur Untersuchung von Gentrifizierung mit Big Data vorgestellt. Basierend darauf werden in Kapitel 4 Datenquellen zur Untersuchung von Gentrifizierung analysiert. Aufbauend auf den Erkenntnissen aus diesem und dem vorigen Kapitel, werden am Ende von Kapitel 4 die zu untersuchenden domänenspezifischen Fragestellungen und Hypothesen gebildet. Eine Beschreibung des technischen Systems und der Datenintegration für die anschließende Analyse befindet sich in Kapitel 5. Basierend auf den aufbereiteten Daten werden in Kapitel 6 die aufgestellten Hypothesen überprüft. Dazu werden mit Algorithmen und Methoden des

maschinellen Lernens verschiedene Modelle gebildet. Die Modelle werden am Ende des Kapitels evaluiert und ihre Aussagekraft bezüglich der Hypothesen diskutiert. Die Arbeit schließt mit einer Diskussion der Ergebnisse in Kapitel 7 und einem Fazit in Kapitel 8 ab.

Das beschriebene Vorgehen ist auch als Knowledge Discovery und Data Mining (KDDM) bekannt. Während mit Data Mining das Generieren von Wissen aus gesammelten Daten gemeint ist, ist der Begriff Knowledge Discovery weiter gefasst. Bei Knowledge Discovery geht es um die Suche nach neuem Wissen in einer Anwendungsdomäne, wobei Data Mining ein Schritt davon ist (Kurgan & Musilek, 2006, S. 2). Für den gesamten KDDM-Prozess gibt es verschiedene Referenzmodelle, welche jedoch untereinander eine hohe Ähnlichkeit aufweisen (Kurgan & Musilek, 2006, S. 9). Der Prozess, der nach einer Umfrage von KDnuggets (2014) am häufigsten verwendet wird, ist das CRISP-DM Referenzmodell. Dieses Modell dient auch als methodische Grundlage dieser Arbeit und wird im Folgenden kurz dargestellt. Die Kapitelüberschriften der Kapitel 3 bis 6 orientieren sich an den Phasen des Modells.

Das Referenzmodell CRISP-DM (CRoss Industry Standard Process for Data Mining) wurde um die Jahrtausendwende von einem branchenübergreifendem Konsortium entwickelt und besteht aus den sechs Phasen *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation* und *Deployment* (CRISP-DM consortium, 2000, S. 10). Diese Phasen sind in Abbildung 1-1 als Lebenszyklus dargestellt.

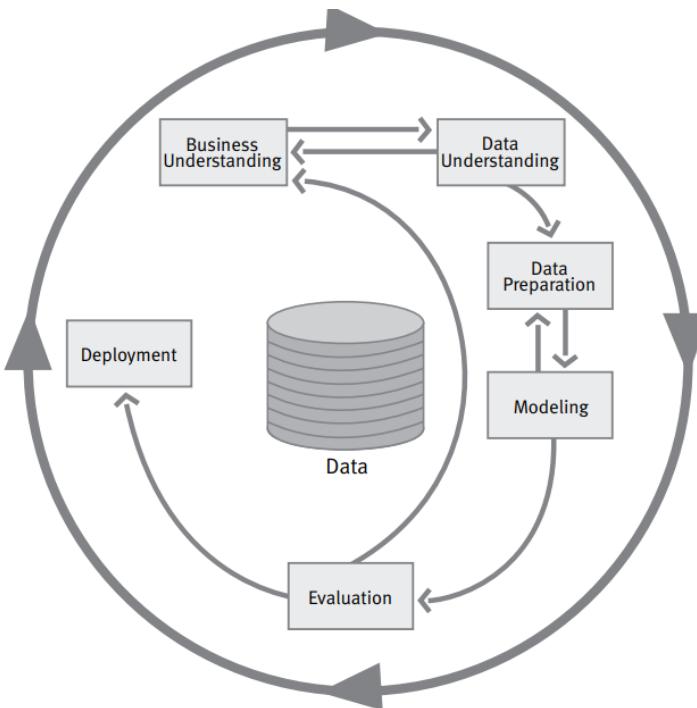


Abbildung 1-1: Phasen des CRISP-DM Referenzmodells (CRISP-DM consortium, 2000, S. 10)

Wirth und Hipp (2000, S. 5–7) haben die einzelnen Phasen des CRISP-DM Referenzmodells wie folgt beschrieben:

Business Understanding – Kapitel 3

In der Phase *Business Understanding* geht es darum die Problemstellung zu verstehen und die Anforderungen aus domänenspezifischer Perspektive zu betrachten. Mit dem gewonnenen Wissen kann dann die Problemdefinition für das Data Mining Projekt gebildet werden.

Data Understanding – Kapitel 4

Darauf folgt die Phase *Data Understanding*, in der eine initiale Datensammlung erfolgt. Die Daten werden explorativ analysiert und beschrieben, zusätzlich wird die Datenqualität geprüft. Durch die explorative Datenanalyse ergeben sich Rückschlüsse auf das Business Understanding. Mit dem Wissen über die verfügbaren Daten und dem Wissen über das Anwendungsfeld können dann erste Hypothesen gebildet werden.

Data Preparation – Kapitel 5

Als nächstes werden in der Phase *Data Preparation* die Daten selektiert, von Datenqualitätsproblemen befreit und integriert. Auf Basis der Datengrundlage können neue Features aus den bestehenden Daten generiert werden. Dieser Teil der Phase wird von anderen Autoren auch als *Feature Engineering* bezeichnet. Nach CRISP-DM steht am Ende der Data Preparation Phase ein fertiger Datensatz, der als Grundlage für die Modellierung genutzt wird.

Modeling – Kapitel 6

In der Phase *Modeling* werden die Machine Learning Algorithmen auf den Daten angewendet. Hierbei werden die Daten in Test- und Trainingsdaten aufgeteilt. Für die Modellierung werden die Algorithmen ausgewählt und deren Parameter eingestellt. Schließlich werden die Modelle gebildet.

Evaluation – Kapitel 6

Anschließend werden in der Phase *Evaluation* die gebildeten Modelle auf Basis des Testdatensatzes überprüft. Aus der Evaluation lässt sich neues Wissen über die Anwendungsdomäne erschließen.

Deployment – (Kapitel 7)

Die Modelle lassen sich dann in der *Deployment* Phase entweder direkt in Systemen einsetzen oder sie vermitteln Wissen zur Anwendungsdomäne, mit dem die Systeme entsprechend angepasst werden können. Je nach Ziel der Analyse kann auch nur ein fachlicher Bericht das Ergebnis eines Data Mining Projektes sein.

Im Data Science Prozess von Microsoft wird die *Featureentwicklung* als erster Schritt der Modellierungsphase betrachtet, gefolgt von der *Featureauswahl*. Danach werden die Modelle gebildet (Microsoft, o.D.). CRISP-DM sieht beides als Teil der Data Preparation an. In dieser Arbeit wird die Featureentwicklung als Teil der Phase Data Preparation betrachtet, während die Featureauswahl als Teil der Modellierungsphase angesehen wird.

1.3 Open Data und Open Knowledge

„Knowledge is open if anyone is free to access, use, modify, and share it — subject, at most, to measures that preserve provenance and openness.“ (Open Knowledge International, o.D.)

Diese Arbeit ist gemäß der Open Definition verfasst worden. Es wurden soweit wie möglich nur kostenlos zugängliche, öffentliche Daten, Formate und Programme genutzt. Die einzigen Einschränkungen bei Open Data dürfen sein, dass die Nutzer der offenen Daten ihre Ergebnisse ebenfalls kostenlos veröffentlichen müssen, oder dass die Quelle der Daten genannt werden muss.

Um dem Anspruch der Offenheit selbst gerecht zu werden, sind die Ergebnisse und der Quellcode aus dieser Arbeit unter www.github.com/dhelweg/masterthesis2018_gentrification unter der GNU General Public License 3³ verfügbar.

Die aufbereiteten Daten sind unter der Open Data Commons Open Database Lizenz 1.0⁴ verfügbar unter www.github.com/dhelweg/masterthesis2018_gentrification/tree/master/data/ready_4_ML.

³ <https://www.gnu.org/licenses/gpl-3.0.de.html>

⁴ <https://opendatacommons.org/licenses/odbl/>

2 Grundlagen

2.1 Big Data Analytics – neue Datenquellen und Technologien

2.1.1 Definition

Big Data wird oft mittels des 3-V-Modells von Gartner (o.D.) beschrieben, indem der Begriff seine Eigenschaften *Volume*, *Variety* und *Velocity* charakterisiert wird. Diese drei Dimensionen gehen auf eine Forschungsarbeit von Laney (2001) zurück, indem dieser die Herausforderungen des Datenwachstums als dreidimensional beschrieben hatte: ansteigendes Volumen (*Volume*), ansteigende Datenvielfalt (*Variety*) und ansteigende Geschwindigkeit (*Velocity*) (Gesellschaft für Informatik [GI], 2013). Bei IBM wurden diese drei Vs durch ein viertes ergänzt, *Veracity*. Damit ist die Unsicherheit gemeint, die bzgl. der Wahrhaftigkeit der Daten aufkommt (Puget, 2013). So können unwahre Daten aus fehlerhaften Sensoren oder von „Fake News“ in sozialen Netzwerken stammen (Bloomberg, 2017).

Ein Treiber dieser neuen heterogenen Datenquellen ist das Internet, welches den Zugriff auf große Datenmengen erlaubt und gleichzeitig selbst eine Vielzahl an Daten produziert. So werden täglich auf YouTube jede Minute im Schnitt 48 Stunden Videomaterial hochgeladen, 1,5 Milliarden Facebook-Nutzer teilen im Monat 30 Milliarden Beiträge, Twitter hat täglich 175 Millionen Tweets von 465 Millionen Nutzer-Accounts (Waterford Technologies, 2017). Neben sozialen Netzwerken gibt es jedoch noch viele andere potentielle Datenquellen, wie Satellitenbilder, Produktinformationen auf Webshops, Presseartikel, (Restaurant-)Kritiken, Open Data Portale und vieles mehr. Auch in Betrieben steigen die Datenmengen, die potentiell ausgewertet werden können. Viele Geschäftsprozesse sind über Software abgebildet, und es können Transaktions- oder Prozessdaten ausgewertet werden. Zudem fallen in den meisten IT-Systemen Log-Dateien an, Sensoren an Produktionsmaschinen sorgen für einen konstanten Datenstrom über die Materialbeschaffung oder den Zustand der Maschine, und per GPS-Chips können Standorte von Betriebsmitteln nachvollzogen werden (Wu, Zhu, Wu & Ding, 2014, S. 99–102). Dies sind nur einige Beispiele, welche einen Eindruck über die Vielfalt der Daten geben sollen.

2.1.2 Technische Implikationen

Mit der Vielfalt der Datenquellen und -formate, der Größe der Daten und dem Tempo indem sich die Daten ändern, können klassische relationale Datenbanksysteme nur unzureichend gut umgehen (Hu, Wen, Chua & Li, 2014). Um mit solchen Daten zu arbeiten,

werden neue Technologien benötigt. Diese Big Data Technologien wurden in einem Artikel der International Data Corporation wie folgt definiert:

“Big data technologies describe a new generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data, by enabling high-velocity capture, discovery, and/or analysis.”

(Gantz & Reinsel, 2011, S. 6)

Dabei sind mehrere Schlüsseltechnologien entwickelt worden, deren Zusammenspiel erst die Verarbeitung von großen Datenmengen möglich macht. Es kann grob in die drei Schichten Infrastruktur, Datenverarbeitung und Anwendung unterteilt werden (Hu et al., 2014).

Infrastruktur

Bei der Infrastruktur sind Fortschritte in Netzwerktechnik, Rechenkapazität und Speichertechnologien zu verzeichnen. Insbesondere die Verwendung von Grafikprozessoren für MapReduce Operationen (s.u.) hat sich als großer Performance Gewinn herausgestellt (Jiang, Chen, Qiao, Weng & Li, 2015). Daneben ergibt sich mit HDFS (s.u.) und MapReduce als Datenverarbeitungsschicht die Möglichkeit, kostengünstige Standardhardware zu nutzen, anstelle von teurer, ausfallsicherer und hochverfügbarer Servertechnologie (White, 2009, S. 42).

Datenverarbeitung

Auf der Seite der Datenverarbeitung sorgen verteilte Dateisysteme und NoSQL-Datenbanken dafür, dass der große Speicherbedarf verteilt, performant und schemafrei abgelegt werden kann. Auf diese Weise können die vielen Daten und verschiedenen Dateiformate in strukturierter und ausfallsicherer Weise gespeichert werden (Hu et al., 2014, S. 653).

Im Zuge der Entwicklung der Suchmaschine Google ist das MapReduce Programmierframework entstanden, welches eine verteilte und fehlertolerante Datenverarbeitung auf verschiedenen Rechnern vereinfacht (Dean & Ghemawat, 2004, S. 1–2). MapReduce wurde anschließend zusammen mit anderen Subprojekten im Apache Hadoop Project gebündelt. Darunter sind unter anderem das verteilte Dateisystem HDFS (Hadoop Filesystem) und Hive, mit dem per SQL-basierter Abfragesprache auf das HDFS zugegriffen werden kann. Die Query-Syntax wird dabei im Hintergrund in MapReduce Jobs übersetzt. Von einigen Autoren wird Hive auch als ein verteiltes Data Warehouse dargestellt (White, 2009, S. 12–13).

Anwendung

Die Anwendungsschicht bezieht sich auf den letzten Teil der Definition nach Gantz und Reinsel, der Analyse der Daten. Hierbei handelt es sich im Kern um Data Mining und

Machine Learning Methoden. Data Mining ist ein Prozess, bei dem es um das Lösen von Problemen durch Analyse von Daten geht. Dabei werden Daten auf strukturelle Muster untersucht, um vorher unbekannte, potentiell nutzbare Informationen zu gewinnen. Machine Learning kann dabei als die technische Basis für Data Mining verstanden werden (Witten, Pal, Frank & Hall, 2017, S. 1–9). Viele der Machine Learning Methoden stammen aus dem 20. Jahrhundert und sind eng verwandt mit statistischen Methoden (Witten et al., 2017, S. 30). Generell gibt es drei Arten, in die sich die Methoden einordnen lassen:

- Zum einen sind es *deskriptive Methoden*, die Zusammenhänge in bestehenden Daten erkennen. Darunter sind Verfahren zum Segmentieren von Daten (engl. clustering) und Analysen zum Finden von Assoziationsregeln. Deskriptive Analysen beschäftigen sich mit der Frage, was geschehen ist (Witten et al., 2017, S. 91–92).
- Zum anderen sind es *prädiktive Methoden* zur Klassifizierung und Regression, also der Bestimmung wie ein unbekannter Zielwert auf Basis von gegebenen Datenfeatures aussieht. Damit beantworten prädiktive Methoden die Frage, was wahrscheinlich passieren wird (Witten et al., 2017, S. 91–92).
- Die dritte Art sind *präskriptive Methoden*. Diese beschäftigen sich damit, welche Aktionen welchen Effekt haben werden. Eine Form von präskriptiven Methoden sind Metaheuristiken, welche bei Optimierungsproblemen eingesetzt werden können (Duarte, Laguna & Marti, 2018, S. 4+13).

Damit ist auf der einen Seite eine Vielzahl von neuen Datenquellen verfügbar, die auf der anderen Seite mit Data Mining Methoden ausgewertet werden können. Diese Methoden können heute aufgrund der neuen Technologien perfomanter genutzt werden als in der Vergangenheit.

2.2 Spatial Big Data – Besonderheiten raumbezogener Daten

Viele der in Kapitel 2.1.1 beschriebenen (Big Data-) Datenquellen liefern Meta-Informationen, mit denen sich die Daten geographisch auswerten lassen. So sind viele Beiträge bei Twitter, Facebook und Instagram mit einem Standort gekennzeichnet. Ebenso sind viele Bilder, die zum Beispiel mit dem Mobiltelefon aufgenommen werden, mit Geo-Tags versehen. Die NASA hat zudem einen großen Bestand an Satellitenbildern von der Erde. Mit ihren Landsat-Satelliten generiert sie von dem gesamten Globus alle 16 Tage Bilder in einer Auflösung von 30 Metern. Daneben können im Zeitalter des Internet of Things (IoT) auch GPS-Sensoren an Betriebsmitteln wie Baumaschinen oder Fahrzeugen

angebracht werden, damit deren Positionen stets für Einsatzplanungen verfügbar sind. All diese Daten lassen sich als Spatial Big Data bezeichnen (Jiang & Shekhar, 2017, S. 3–5). Traditionell werden solche Geodaten mittels Geoinformationssystemen (GIS) analysiert. Aufgrund der großen Datenmengen werden hierzu neuerdings jedoch auch Big Data Systeme verwendet (Bruns & Bernsdorf, 2016; Disy Informationssysteme GmbH, o.D.a). Dabei sollten jedoch Besonderheiten von Geodaten berücksichtigt werden.

Spatial Autocorrelation

Tobler beschreibt in seiner *First Law Of Geography* das Problem der räumlichen Autokorrelation wie folgt: „*Everything is related to everything else, but near things are more related than distant things*“ (Tobler, 1970). Das bringt zum Ausdruck, dass räumliche Daten statistisch nicht unabhängig voneinander sind, und nahgelegene Positionen oft Gemeinsamkeiten haben. Nach Jiang und Shekhar (2017, S. 9) muss dies bei der Datenanalyse berücksichtigt werden, da sonst ungenaue Modelle die Folge sein können. Ein Beispiel hierfür wäre, dass sich georeferenzierte Bilder bei Instagram um Sehenswürdigkeiten wie dem Brandenburger Tor oder dem Eifelturm sammeln.

Spatial Heterogeneity

Das Problem der räumlichen Heterogenität sagt aus, dass Daten in unterschiedlichen Räumen unterschiedlich dargestellt sein können, oder gleiche Darstellungen eine unterschiedliche Bedeutung haben können. So kann bei der Bildanalyse von Wäldern ein Feuchtgebiet ähnlich aussehen wie ein trockener Wald in einer anderen Umgebung. Auch können öffentlich erfasste Daten wie Einwohnerstatistiken in unterschiedlichen Ländern eine andere Bedeutung haben (Jiang & Shekhar, 2017, S. 10). So hat die Aussage des Anteils von Transferleistungsbeziehern an der Bevölkerung in einer Nachbarschaft in den USA eine andere Bedeutung als in Deutschland, da die Gesetze für Sozialleistungen unterschiedlich sind.

Modifiable Area Unit Problem (MAUP)

Eine weitere Herausforderung bei räumlichen Daten sind verschiedene Skalen und Auflösungen, in denen die Daten betrachtet werden können. So haben verschiedene Satelliten verschiedene Bildauflösungen (Jiang & Shekhar, 2017, S. 10–11). Auch aggregierte Daten wie die Alterspyramide in einer Umgebung können eine andere räumliche Auflösung haben als zum Beispiel Daten zu Fahrzeug-Zulassungen. Wenn beide Daten nun kombiniert ausgewertet werden sollen, muss ein gemeinsamer Raumbezug hergestellt werden (Schulz, 2015, S. 30–32). Auch bei der Auswertung sind auf unterschiedlicher Auflösung unterschiedliche Ergebnisse zu erwarten (Jiang & Shekhar, 2017, S. 10).

2.3 Anwendungsfälle für Spatial Big Data Analytics

Für die Analyse von Spatial Big Data gibt es eine Vielzahl von Anwendungsfällen. Auf einige soll nun exemplarisch eingegangen werden.⁵

2.3.1 Standort- und Filialnetzplanung

Für die Entscheidung der Standortauswahl werden klassischerweise verschiedene volks- und betriebswirtschaftliche Standortfaktoren genutzt, die je nach Unternehmen und Geschäftsmodell unterschiedlich sein können. Dabei fließen häufig nur bekannte Standorte in die Untersuchung ein (Neumair & Haas, 2018). Insbesondere bei einer Filialnetzplanung spielen dabei absatzbezogene Standortfaktoren eine große Rolle. Für die Filialnetzplanung von Banken hat das Startup Geospin eine Lösung auf Basis von Spatial Big Data entwickelt, welche die vorhandenen bankinternen Daten mit neuen Datenquellen verbindet. Kombiniert können die externen Daten Auskunft darüber geben, warum eine Filiale profitabler ist als eine andere. Als Datenquelle nutzt Geospin Informationen wie umliegende Geschäfte und Restaurants, deren Öffnungszeiten, Mietpreise, Passanten Frequenz sowie klassische soziodemographische Daten (Wagner, 2016). Auf diese Weise werden Empfehlungen zu optimalen Standorten von Geldautomaten- und Filialstandorten abgeleitet, welche in Zeiten von steigenden Filialschließungen (KfW, 2017) bei Banken zu einem Wettbewerbsvorteil werden können. Durch eine automatisierte Datenerhebung mittels Spatial Big Data können zudem mehr Standorte untersucht werden, als in einer klassischen Untersuchung, bei der nur wenige, oft bereits bekannte Standorte im Detail verglichen werden.

Neben Banken können auch andere Branchen von Big Data Ansätzen bei der Standort- und Filialnetzplanung profitieren. Die Deutsche Apotheker- und Ärztebank bietet zum Beispiel ihren Kunden einen Beratungsdienst zur Standortanalyse bei Existenzgründung an. Dieser basiert derzeit auf soziodemographischen Daten wie Kaufkraft, Beschäftigungsquote, Altersstruktur und einer nach Fachrichtung aufgeschlüsselten Ärztedichte (apoBank, 2015; chance-praxis.de, 2015). Daten wie Mietpreise, umliegende Geschäfte und Passanten Frequenz könnten auch hier neue Erkenntnisse und verbesserte Standortempfehlungen zur Folge haben. Ein weiteres Beispiel für Standortplanung mit Spatial Big Data ist eine Lösung von Geospin zur Planung von Ladestationen für Elektroautos im urbanen Raum (Geospin, 2016).

⁵ Weitere Anwendungsfälle sind im Anhang unter Kapitel 9.1 zu finden.

2.3.2 Wohnortsuche und Immobilieninvestitionen

Ein weiteres Beispiel für die Anwendung von Spatial Big Data ist die Suche nach einem geeigneten Wohnort. So haben unterschiedliche Personen und Haushalte unterschiedliche Bedürfnisse an ihre Nachbarschaft. Während für viele Familien eine Nähe zu Kitas und Schulen wichtig sei können, so können für Studenten wiederum eher Nähe zur Universität, Supermärkten und Bars interessant sein. Diese Fragestellung wird aktuell auch in einem Projekt des Bundesministerium für Verkehr und digitale Infrastruktur (BMVI) (o.D.) beleuchtet. Im WEKOVI-Projekt untersucht das BMVI zusammen mit dem Dienstleister Disy Informationssysteme GmbH wie Orte bezüglich komplexer abstrakter Eigenschaften wie Familieneignung bewertet werden können. Hierzu werden die Eigenschaften in Teilespekte, wie die Nähe zum Kinderarzt o.ä., heruntergebrochen und anschließend in einem Vergleichsindex aufbereitet. Ziel des 2017 gestarteten, zweijährigen Projektes ist es, eine prototypische, offene Software-Plattform zu schaffen, mit der Anwender mit Domänenexpertise und ohne IT-Fachkenntnisse arbeiten können. Hierzu setzt das Projekt auf eine moderne technische Architektur. Diese besteht aus Komponenten wie Talend, einer auf Hadoop basierenden Datenintegrationsplattform, Apache Flink, zur Verarbeitung von verteilten Datenströmen, und ElasticSearch, für eine verteilte Datenhaltung und Suchfunktion (Disy Informationssysteme GmbH, o.D.a). Als Datenquellen sollen laut Projektbeschreibung ausschließlich offene Geodaten verwendet werden (Bundesministerium für Verkehr und digitale Infrastruktur, o.D.).

Diese Informationen könnten für die Suche nach dem passenden Wohnort verwendet werden. Sie könnten jedoch auch für Investitionsentscheidungen in Immobilien verwendet werden. So besagt eine Immobilienweisheit „*Bekanntlich gibt es nur drei wirklich wichtige Kriterien, um den Wert und die Wertsteigerung einer Immobilie zu bestimmen: 1. die Lage, 2. die Lage und 3. die Lage*“ (immoverkauf24.de, o.D.). Da sich aus Spatial Big Data zum einen Erkenntnisse über die aktuelle Lage ergeben, und zum anderen ggf. auch Trends sichtbar gemacht werden können, spricht dies für den Einsatz von Spatial Big Data als Entscheidungshilfe bei Immobilieninvestitionen. Je früher ein zukünftiges Szene-Gebiet zuverlässig erkannt wird, desto günstiger die Investition und so höher die Rendite.

2.3.3 Stadtentwicklung

Ein weiterer Anwendungsfall für Spatial Data Analytics ist die Stadtentwicklung. Hier werden klassischer Weise viele Daten raumbezogen erhoben, unter anderem über den Zensus, bei dem Daten über die Bewohner von Nachbarschaften gesammelt werden. Es

gibt jedoch auch Potential für Big Data Analysen in der Stadtentwicklung. In einer Studie des Berliner Referats für Stadtentwicklungsplanung von 2017 wurde das Potential von Big Data für die Stadtentwicklungsplanung untersucht (Referat Stadtentwicklungsplanung & EBP, 2017). In der Studie wurden verschiedene Datenquellen analysiert und mit der Aussagekraft zu den Themen der Stadtentwicklung geprüft. Ein Thema der Stadtentwicklung ist dabei die Zentrumsausstattung mit Nahversorgung, Einzelhandel, (öffentliche) Dienstleistungen und Gastronomie; Berlin setzt dabei auf ein polyzentrales Zentren-System (Senatsverwaltung für Stadtentwicklung, 2011). Das heißt, es sollen mehrere Stadtzentren existieren, anstelle von einem Zentrum in der Stadtmitte, wie bspw. in Hamburg. Relevante Faktoren sind dabei auch die Bevölkerungsstruktur, sowie die Nachfrage und Kaufkraft von Einwohnern und Besuchern der Stadt. Im Ergebnis der Analyse wurden nutzergenerierte Daten aus Rating-, Foto- und Kartenportalen als aussagekräftige Datenquellen identifiziert um Informationen über (touristische) Nachfrage und Zentrumsausstattung abzuleiten. Über transaktionsgenerierte Daten aus Mobilitätsplattformen, sowie sensorgenerierte Daten über Mobilfunkdaten könnten Informationen über Einzugsbereiche und Nachfrage generiert werden (Referat Stadtentwicklungsplanung & EBP, 2017).

3 Business Understanding – Gentrifizierung

Ein Teil des Anwendungsfalls Stadtentwicklung ist die Gentrifizierung, die sowohl die Änderung der Bevölkerungsstruktur, als auch die Veränderung der lokalen Angebotsstruktur beinhaltet. Diese soll in dieser Arbeit mit Spatial Big Data Analysen untersucht werden. Hierbei soll das *Business Understanding* im Sinne des CRISP-DM Vorgehensmodells erarbeitet werden. Zunächst wird der Begriff der Gentrifizierung in seinen verschiedenen Facetten definiert und am Beispiel Berlin veranschaulicht. Dann soll auf den Stand der Gentrifizierungsforschung mit Big Data eingegangen werden.

3.1 Definition

Der Begriff der Gentrifizierung wurde von Glass (1964) eingeführt. In der Literatur herrscht Uneinigkeit über eine einheitliche Definition. Zook, Shelton und Poorthuis (2017, S. 2) definieren sie grob als Wandel einer Nachbarschaft durch den Austausch von einkommensschwacher durch einkommensstarker Bevölkerung.

Das Deutsche Institut für Urbanistik definiert Gentrifizierung in einem Online-Artikel wie folgt: „*[Gentrifizierung] beschreibt den Wechsel von einer statusniedrigeren zu einer statushöheren (finanzkräftigeren) Bewohnerschaft, der oft mit einer baulichen Aufwertung, Veränderungen der Eigentümerstruktur und steigenden Mietpreisen einhergeht.*“ (Deutsches Institut für Urbanistik, 2011)

Davidson und Lees (2005, S. 1174–1185) definieren Gentrifizierung im Kontext der Aufwertungsprozesse an der Themse in London als Kombination verschiedener Aspekte: Investition von Kapital, Soziale Aufwertung, Umwandlung der Landschaft, sowie Verdrängung von ärmeren Haushalten. Diese Definition deckt sich mit anderen, die nach Glatter (2006) als holistische Definition bezeichnet werden kann. Im Folgenden wird auf die einzelnen Aspekte eingegangen, wobei sich die Begriffe an Holm und Schulz (2016) orientieren. Verdrängung und soziale Aufwertung wurden in einem Aspekt zusammengefasst.

Immobilienwirtschaftliche Aufwertung

Bei Davidson und Lees (2005, S. 1174–1175) ist hiermit eine (überproportional hohe) Investition von Kapital in einem Gebiet gemeint. Erklärbar sind diese Investitionen durch die *Rent-Gap-Theorie* von Smith (1979). Sie besagt, dass wenn die tatsächliche Rente eines Grundstücks die potentielle Rente durch Umnutzung unterschreitet, eine Investition in Grund und Boden eine immobilienwirtschaftliche Aufwertung und somit Gentrifizierung bestärken kann (Helbrecht, 1996, S. 7–10). Weiterentwickelt wurde diese Theorie durch die *Value-Gap-Theorie* von Hamnett und Randolph (Hamnett, 1991; 1983), die den

Wert der Mieteinnahmen der kommenden Jahre mit dem des Verkaufswertes im unvermieteten Zustand vergleicht. Wenn der Verkauf der (modernisierten) unvermieteten Eigentumswohnung mehr Rendite verspricht als die Mieteinnahmen, so würde eine Investition in das Gebäude den Gentrifizierungsprozess bestärken (Kecske & Friedrichs, 2004, S. 27). Schulz (2015, S. 11) gibt zu bedenken, dass politische Maßnahmen großen Einfluss auf diese Ertragslücken (Rent-Gap und Value-Gap) haben können. Neben Ertragslücken sind jedoch auch Spekulationen auf Wertsteigerungen von Immobilien oder Grundstücken ein möglicher Treiber für Investitionen in Immobilien und Grundstücken (Schulz, 2015, S. 11).

Verdrängungsinduzierte soziale Aufwertung

Nach Davidson und Lees (2005, S. 1176–1185) werden bei der Investition hochwertigere Miet- und Kaufobjekte gebaut, die sich nur einkommensstarke Bewohner leisten können. Durch die steigenden Mieten werden also die bisherigen statusschwächeren Einwohner verdrängt und statushöhere Bewohner ziehen hinzu.

Laut Schulz (2015, S. 11) erfordert eine immobilienwirtschaftliche Aufwertung eine soziale Aufwertung, um die Lücke zwischen Investitionskosten und Mieteinnahmen zu schließen. Es würde also nur investiert werden, wenn zu erwarten ist, dass Mieter die höhere Miete oder den Kaufpreis der Eigentumswohnung tragen können. Die höheren Mieten würden laut Holm (2014, S. 284) wiederum die soziale Aufwertung treiben, da nur noch einkommensstarke Haushalte es sich leisten könnten dort zu wohnen. In einer späteren Arbeit definieren beide Gentrifizierung „*als eine Konjunktion von sozialer und immobilienwirtschaftlicher Aufwertung*“ (Holm & Schulz, 2016, S. 300); laut ihnen treten beide Aufwertungseffekte gleichzeitig auf.

Veränderung lokaler Angebotsstrukturen

Als weiteren Punkt nennen Davidson und Lees (2005, S. 1181–1183) ein erneuertes Investment in das kulturelle Kapital einer Nachbarschaft. Sie begründen dies damit, dass vielen Zuziehenden die Kultur und Geschichte einer Nachbarschaft wichtig seien. Immobilieninvestoren würden diese Informationen zudem für Marketingmaterialien nutzen. In beispielhaften Auszügen von Werbematerialien wird auf die Geschichte der Gegend, die Verkehrsanbindungen, die Distanz zur Innenstadt, das Flair und die Kultur der Nachbarschaft, das gastronomische Angebot, sowie auf die umliegende Natur eingegangen. Laut Davidson und Lees (2005, S. 1183) ist das soziale Leben in der Nachbarschaft jedoch nur für 18% der zuziehenden Gentrifizierer ein Grund für den Zuzug. Die meisten nutzen lediglich lokale Restaurants (83%), sowie private Fitnessräume (57%). Andere Angebote wie öffentliche Sportstätten, Büchereien, Nachbarschafts- und Freizeitzentren finden laut

Davidson und Lees nur wenig Anklang (<5%). Die Gentrifizierer (s. u.) würden sich laut ihnen in die Kultur einkaufen und diese lediglich konsumieren, nicht jedoch Teil des sozialen Lebens in der Nachbarschaft sein.

Der Aspekt der Veränderung der Angebotsstrukturen findet sich primär in holistischen Definitionen wieder (Glatter, 2006). Häufig wird entweder nur auf die verdrängungsinduzierte soziale Aufwertung (Döring & Ulbricht, 2016; Zook, 2017), oder auf einen Dualismus aus verdrängungsinduzierte soziale Aufwertung und immobilienwirtschaftlicher Aufwertung reduziert (Holm & Schulz, 2016; Koch, Kortus, Schierbaum & Schramm, 2016).

In dieser Arbeit soll Gentrifizierung holistisch als Dreiklang dieser angesehen werden, also der immobilienwirtschaftlichen Aufwertung, der verdrängungsinduzierten sozialen Aufwertung und der Veränderung lokaler Angebotsstrukturen.

3.2 Gentrifizierung als Prozess

Gentrifizierung wird von vielen Autoren als ein Prozess angesehen.⁶ Eines der bekanntesten Modelle hierzu ist das Phasenmodell von Dangschat (1988). In diesem Modell sind die zuvor angesprochenen Gentrifizierer die zweite Gruppe der Hinzuziehenden. Sein Modell des doppelten Invasions-Sukzessions-Zyklus ist in Abbildung 3-1 dargestellt.

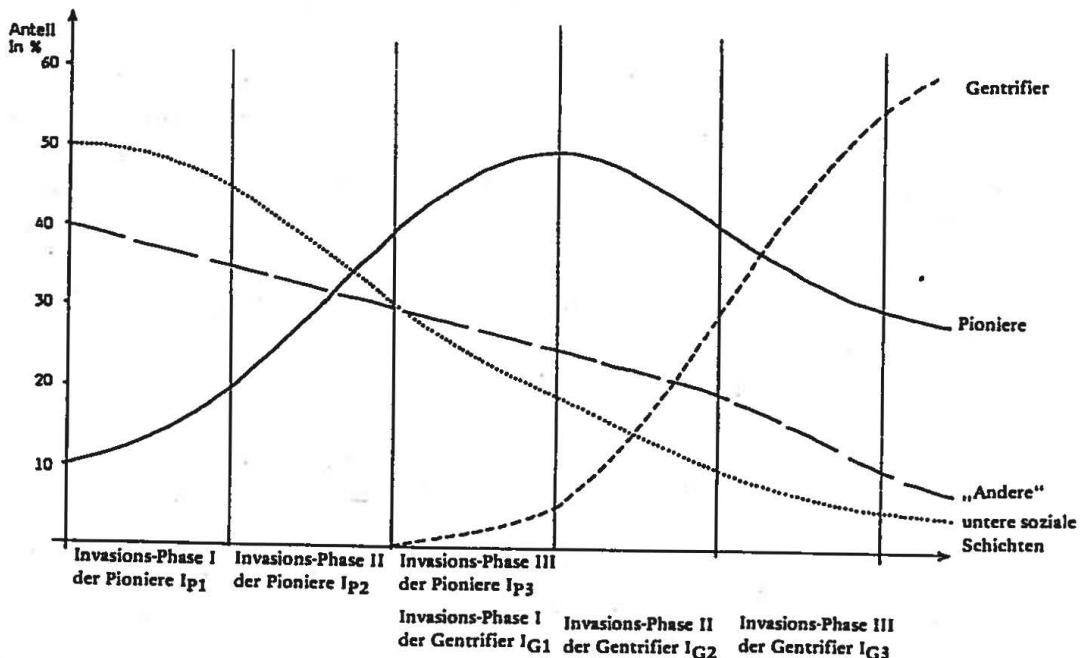


Abbildung 3-1: Doppelter Invasions-Sukzessions-Zyklus (Dangschat, 1988, S. 281)

⁶ Im Anhang unter Kapitel 9.2.1 - Phasenmodelle wird ein alternatives Modell vorgestellt.

Dieses Modell beschreibt zwei Zuwanderungs- und Verdrängungsphasen. Kecskes (1994, S. 28–34) beschreibt diesen Prozess in seiner Arbeit. Zunächst ziehen junge Pioniere in die Nachbarschaft, gemeint sind Künstler und Studenten mit hohem Bildungsgrad und geringem Einkommen, die oft in Wohngemeinschaften leben. Diese nehmen niedere Wohnumstände in Kauf und verbessern ihre Wohnsituation selbst. Vermieter würden zu diesem Zeitpunkt noch keine Investitionen tätigen. Mit dem sukzessiven Zuzug der Pioniere beginnt die Verdrängung der unteren sozialen Schichten wie Arbeitern, Ausländern und alten Bewohnern. Durch die neuen Bewohner verändere sich dann die lokale Angebotsstruktur. Es bilde „*sich eine auf die Bedürfnisse der neuen Bewohner bezogene Szene‘ mit Läden, Kneipen und Restaurants*“ (Kecskes, 1994, S. 28). Durch die Veränderung der Nachbarschaft würden dann in der zweiten Phase die Gentrifizierer auf die Nachbarschaft aufmerksam werden. Diese seien älter und deutlich einkommensstärker als die Pioniere und würden in Ein- bis Zweipersonenhaushalten leben. Durch die erhöhte Nachfrage am Markt würden dann Miet- und Kaufpreise steigen, womit sich Investitionen der Vermieter wieder lohnen würden. Durch die steigenden Mieten würden nun neben den unteren sozialen Schichten auch die Pioniere verdrängt werden.

Ein Kritikpunkt an dem Modell des doppelten Invasions-Sukzessions-Zyklus ist nach Friedrichs (1996, S. 16–17), dass das Modell eine Verdrängung von Pionieren beschreibt, jedoch Pioniere im Laufe der Zeit selbst zu Gentrifizierern werden könnten. So haben Studenten oftmals nach dem Studium ein überdurchschnittliches Einkommen und würden somit zu unterschiedlichen Betrachtungszeitpunkten einmal als Pionier und einmal als Gentrifizierer gelten.

3.3 Gentrifizierung in Berlin

Im Zuge der Finanzkrise und der darauf folgenden Niedrigzinsphase mit günstigen Darlehen für Immobilienkredite hat der Erwerb von Eigentumswohnungen stark zugenommen. So ist die Anzahl der Umwandlungen von Miet- in Eigentumswohnungen in Berlin von 2013 auf 2015 um ca. 90% gestiegen (dpa, 2016). Dieser Anstieg kann durch die Value-Gap-Theorie erklärt werden, auf die in der Definition der immobilienwirtschaftlichen Aufwertung eingegangen wurde.⁷ Der hohe Anstieg der Immobilienpreise um fast 21% von 2016 auf 2017 (Knight Frank LLP, 2017) könnte zudem auf Spekulationseffekte im Markt hindeuten.

⁷ Vgl. Kapitel 3.1

In Abbildung 3-2 ist der Verlauf der Entwicklung der Mietpreise auf dem Portal ImmobilienScout24 dargestellt, welche im Artikel von Rundfunk Berlin-Brandenburg (2016) zur Verfügung stehen. Es sind deutliche Unterschiede zwischen den Bezirken zu erkennen. Während in Berlin Mitte die Mieten zwischen 2012 und 2016 jedes Jahr im Schnitt über 6,6% anstiegen, war es in Marzahn-Hellersdorf, kommend von generell geringeren Mietpreisen, ein durchschnittlicher Anstieg um 4,3% (Rundfunk Berlin-Brandenburg, 2016).

Angebotsmietpreise in EUR	Q2_2008 (Top 3)	Ø jährlicher Anstieg 2008-2012 (Differenz zu gesamt Berlin)	Q2_2012 (Top 3)	Ø jährlicher Anstieg 2012-2016 (Differenz zu gesamt Berlin)	Q2_2016 (Top 3)
Bezirk Treptow-Köpenick	5,49	-0,8%	6,48	-0,7%	7,73
Bezirk Marzahn-Hellersdorf	4,73	-1,7%	5,40	-1,2%	6,31
Bezirk Pankow	6,05	0,5%	7,50	0,1%	9,29
Bezirk Neukölln	5,18	1,4%	6,67	1,3%	8,56
Bezirk Charlottenburg-Wilmersdorf	6,78	-0,5%	8,11	-0,4%	9,80
Bezirk Steglitz-Zehlendorf	6,42	-0,9%	7,55	-0,9%	8,96
Bezirk Lichtenberg	5,29	0,3%	6,51	0,1%	7,99
Bezirk Tempelhof-Schöneberg	5,92	0,1%	7,23	-0,1%	8,75
Bezirk Friedrichshain-Kreuzberg	6,11	1,2%	7,80	1,5%	10,14
Bezirk Reinickendorf	5,42	-0,5%	6,47	-0,7%	7,70
Bezirk Spandau	5,22	-1,0%	6,12	-0,8%	7,29
Bezirk Mitte	5,82	1,5%	7,51	1,1%	9,70
Berlin	5,70	5,1%	6,94	5,5%	8,52

Abbildung 3-2: Entwicklung der Angebotsmietpreise bei ImmobilienScout24 2008-2016 (Eigene Darstellung)

Auffällig ist, dass zwei Top 3-Bezirke von 2008 über die Jahre hinweg einen unterdurchschnittlich starken Anstieg der Mietpreise hatten. So ist in 2008 Steglitz-Zehlendorf noch ca. 24% teurer als Neukölln, in 2016 sind es nur noch knapp 5%. Dies deckt sich mit dem Phasenmodell nach Birch (1971)⁸, wonach am Ende der Phase der Vollentwicklung die Herabstufung folgt (Friedrichs, 1996, S. 18)⁹. Des Weiteren fallen die drei Bezirke Neukölln, Friedrichshain-Kreuzberg, sowie Berlin Mitte auf, sie haben deutlich höhere jährliche Anstiege in den Angebotsmietpreisen als Berlin insgesamt. Dies könnte als Beleg der immobilienwirtschaftlichen Aufwertung¹⁰, als ein Teil der Gentrifizierung angesehen werden (Holm, 2014, S. 282).

Laut Schulz (2015, S. 11) haben Gesetze wie die Mietpreisbremse (§ 556d BGB, 2015) starken Einfluss auf die Größe der Ertragslücken. Die Mietpreisbremse legt fest, dass in

⁸ Vgl. Kapitel 9.2.1

⁹ Vgl. Kapitel 3.2

¹⁰ Vgl. Kapitel 3.1

angespannten Wohnungsmärkten eine Mieterhöhung auf maximal 10% über der ortsüblichen Vergleichsmiete erlaubt ist. Hierbei sind jedoch durch weitere Investitionen neue Ertragslücken erschließbar. So kann mit Investitionen in energetische Modernisierung die Mietpreisbremse umgangen werden oder auch Bestandsmieten erhöht werden (Berliner Mietverein, 2015; FOCUS Online, 2017). In den Zahlen von ImmobilienScout24 ist kein großer Unterschied zwischen dem Anstieg für Berlin insgesamt von Q2_2014 auf Q2_2015 im Vergleich zu Q2_2015 auf Q2_2016 zu erkennen, obwohl die Mietpreisbremse im Jahr 2015 eingeführt wurde (Rundfunk Berlin-Brandenburg, 2016).¹¹

3.4 Quantitative Analyse der Gentrifizierung in Berlin

Laut Holm (2014, S. 277) ist es in einem gesamtstädtischen Aufwärtstrend, wie er in Berlin zu beobachten ist, immer schwieriger Gentrifizierungsprozesse zu identifizieren. So hätten Stadtverwaltungen das Problem, dass sie nicht wissen auf welche Gebiete sie Instrumente zur Vermeidung von unerwünschten Verdrängungsprozessen richten sollen. Insbesondere Prognosen der Dynamik der Gentrifizierung seien schwieriger geworden. So sei nicht klar, in welchen Gebieten der Gentrifizierungsprozess bereits abgeschlossen ist, und welche Gebiete als nächstes betroffen sein könnten. Auch Hammel und Wyly (1996, S. 248) geben zu bedenken, dass die Gentrifizierung wegen des komplexen dahinter liegenden Prozesses nur schwer messbar ist. Gerade die Multidimensionalität der Gentrifizierung mache die Reduktion auf nur eine Variable nahezu unmöglich (Bostic & Martin, 2003, S. 2431).

Als häufig verwendete Indikatoren der Gentrifizierung geben Zook et al. (2017, S. 3) die *Veränderung des Haushaltseinkommens*, des *Bildungsniveaus*, der *Immobilienpreise* und der *Mieten* an. Holm (2014, S. 282) bezeichnet die *Angebotsmieten* als gute Messgröße für die Ertragserwartungen der Investoren. Diese rechnen damit solvante Mieter zu finden, die sich das höhere Mietniveau leisten können. Friedrichs (1996) stellt fest, dass die meisten Autoren für die Bestimmung des Fortschritts des Gentrifizierungsprozesses die folgenden Indikatoren verwenden: „*Alter der Gebäude, durchschnittliche Miete, Anteil der Wohnungseigentümer, Anteil der Minorität(en) im Wohngebiet, durchschnittliches Einkommen der Bewohner, Alter der Bewohner*“ (Friedrichs, 1996, S. 21).

In Berlin wurden in jüngerer Vergangenheit verschiedene quantitative Analysen zur Messung von Gentrifizierung durchgeführt. Zu nennen sind die Autoren Döring und Ulbricht (2016, S. 21–23), die eine gesamtstädtische Analyse auf Basis von 60 Prognoseräumen

¹¹ Detaillierte Daten zu dem Verlauf der Angebotsmietpreise in Berlin befinden sich im Anhang unter Tabelle 9-2 und Tabelle 9-3

durchgeführt haben. Dabei wurden zunächst mittels verschiedener Indikatoren drei Indizes ermittelt: *Mobilität, Veränderung der Bevölkerungsstruktur, sowie Wohnungswirtschaft*. Mittels einer Methode zur Punktbildung wurden für jeden Prognoseraum und Indikator Punkte in Abhängigkeit zum Unterschied zum Mittelwert von gesamt Berlin ermittelt. Ein Punkt entspricht dabei jeweils einer Distanz vom Mittelwert um eine halbe Standardabweichung. Der Index entspricht dann der Summe der Punkte. Der erste Index zur Mobilität basiert auf den Indikatoren des *durchschnittlichen Wanderungsvolumens*, sowie dem *Anteil der Bewohner mit mindestens fünfjähriger Wohndauer*. Der zweite Index zur Veränderung der Bevölkerungsstruktur basiert auf der *Entwicklung der Altersgruppen 18-35 und 35-45*, der *Entwicklung der Zahl der Langzeitarbeitslosen*, sowie der *Entwicklung des Ausländeranteils* und der *Kaufkraft*. Der dritte Index betrachtet die Immobilienwirtschaftliche Aufwertung und nutzt die Indikatoren *Entwicklung der Kaltmiete bei Neuvermietung*, sowie die *Anzahl an Umwandlungen von Miet- in Eigentumswohnungen*.

In einer anderen Arbeit haben Holm und Schulz (2016, S. 298–309) mit dem GentiMap-Projekt ein Messmodell für Gentrifizierung entwickelt, welches diese beiden Aspekte der Gentrifizierung, die „*immobilienwirtschaftliche Aufwertung*“ und die „*verdrängungsinduzierte soziale Aufwertung*“, quantifiziert und im Beispiel von Berlin auf Nachbarschaftsebene messbar macht. Als dritte Komponente der Gentrifizierung wird die Veränderung der lokalen Angebotsstruktur aufgeführt, welche in dem GentiMap Modell jedoch keine Operationalisierung findet. Die Gentrifmap Methode sieht einen fünfschrittigen Ablauf zur Bestimmung eines Gentrifizierungsindexes vor: Wahl der Indikatoren, Berechnung der Trendabweichung, Standardisierung, Bildung von Sozial- und Immoindex, Kombination der beiden Indizes zum Gentrifizierungsindex. Für Berlin haben sie ebenfalls die 60 Planungsräume als Bezugsgröße verwendet. Als Indikatoren verwendeten sie für den Sozialindex die *Anzahl der Transferleistungsempfänger*. Für den Immoindex wurden *durchschnittliche Angebotsmieten, Eigentumswohnungspreise, Anzahl der angebotenen Mietwohnungen*, sowie *Anzahl der zum Kauf angebotenen Wohnungen* verwendet.

3.5 Big Data Analysen zu Gentrifizierung

Neben den oben genannten quantitativen Analysen auf Basis von Zensusdaten und ähnlich strukturierten Daten, gibt es einige Forschungsarbeiten zur Analyse von Gentrifizierung mit Big Data. Im Buchbeitrag „Big Data and the City“ haben Zook et al. (2017) eine aktuelle Übersicht des Forschungsstandes zur Analyse von Gentrifizierung mittels Big

Data gegeben. Dabei sind sie zu dem Schluss gekommen, dass Big Data kein Allheilmittel für die empirische Analyse von Gentrifizierung ist, jedoch einen Beitrag dazu leisten kann, diesen komplexen soziogeographischen Prozess zu verstehen. Im Folgenden soll auf die von ihnen untersuchten Arbeiten eingegangen werden.

Beekmans (2011) hat in seiner Masterarbeit Foursquare Venues (Points of Interest), sowie deren Check-Ins in einem von Gentrifizierung betroffenem Stadtgebiet in Amsterdam visualisiert. Dabei wurde festgestellt, dass in einigen Nachbarschaften, die als von Gentrifizierung betroffen galten, eine erhöhte Dichte an Foursquare-Aktivität sichtbar war. Eine statistische Analyse fand jedoch nicht statt.

In einer anderen Masterarbeit hat Schaefer (2014) georeferenzierte Tweets in Los Angeles ausgewertet. Dabei hat er die Tweets auf Basis einer Stichwort-Liste gefiltert, welche er für Anzeichen von Gentrifizierung hielt. Durch das Filtern wurden 1929 Tweets gefunden, wobei 56% davon vom Stichwort *Starbucks* gefiltert wurden. Weitere Stichworte waren *neighborhood*, *rent* und *hipster* – jeweils mit 8% der Tweets; die restlichen 20% verteilen sich auf 13 anderen Stichworten (Schaefer, 2014, S. 33). Das Ergebnis wurde visuell dargestellt und diskutiert. Ein Zusammenhang zwischen den gefilterten Tweets und als gentrifiziert geltenden Gebieten ergab sich aber nicht (Schaefer, 2014, S. 65).

Venerandi et al. (2015) haben eine Korrelationsanalyse zwischen Verdrängung und lokaler Angebotsstruktur in London, Manchester und West Midlands (Region um Birmingham) durchgeführt. Als Messgröße für die Verdrängung wurde der *Index of Multiple Deprivation* (IMD) genommen, der im Vereinigten Königreich 2011 auf Basis von einem Mikrozensus erhoben wurde und aus mehreren Effekten besteht: *Einkommen*, *Arbeitslosigkeit*, *Gesundheit*, *Bildung*, *Kriminalität*, *Lebensqualität* und *Wohnkosten*. Die IMD-Daten sind normalisiert auf Raumgrößen, die etwa 1500 Einwohner umfassen. Für die Analyse war den Autoren diese Raumgröße jedoch zu klein, sodass sie die IMD-Daten auf die Größe der Wahlbezirke hochrechneten. Die lokale Angebotsstruktur besteht aus Points of Interests (POI) von Foursquare und OpenStreetMap (OSM). Für die Untersuchung wurde nicht die Anzahl, sondern die darauf basierende *Offering Advantage* (OA) der POI-Kategorien berechnet:

$$OA(c_i, n_k) = \frac{count(c_i, n_k)}{\sum_{j=1}^N count(c_j, n_k)} \cdot \frac{\sum_{j=1}^N count(c_j)}{count(c_i)}$$

Abbildung 3-3: Formel zur Offering Advantage. (Eigene Darstellung nach Venerandi et al., 2015)

Die OA basiert auf der Berechnung des Indexwertes *Revealed Comparative Advantage*, mit welchem die Exportstärke von Ländern für spezifische Güter im internationalen Vergleich gemessen werden kann (Balassa, 1965). Die OA setzt dagegen die Anzahl einer POI-Kategorie wie z.B. italienische Restaurants in einer Nachbarschaft mit der Anzahl derselben POI-Kategorie im Stadtgebiet in Verbindung. So werden die charakteristischen Eigenschaften der Nachbarschaften stärker in den Vordergrund gestellt. In der Formel aus Abbildung 3-3 ist c_i die POI-Kategorie, n_k ist die Nachbarschaft, $\text{count}(c_i, n_k)$ gibt die Anzahl der POIs der Kategorie i in der Nachbarschaft K an. N ist die Anzahl der POI-Kategorien und $\text{count}(c_i)$ gibt die Anzahl der POIs von Kategorie c_i im gesamten Stadtgebiet an. Auf Basis der IMD-Werte wurden zwei Kategorien gebildet, welche die Gebiete je zu einer Hälfte beinhalteten. Mittels *Naive Bayes* konnte folgendes Klassifizierungsmodell erstellt werden:

<i>Precision</i>	<i>Recall</i>	<i>F-Measure</i>	<i>Class</i>
0.763 (+41%)	0.692 (+17%)	0.726 (+28%)	50% more deprived
0.713 (+37%)	0.780 (+39%)	0.745 (+38%)	50% less deprived

Abbildung 3-4: Genauigkeit Klassifikation von Verdrängung mittels OSM&Foursquare
(Venerandi et al., 2015, S. 260)

Die meisten POIs wurden aus Foursquare entnommen, da im Analysegebiet für OSM primär Haltestellen und andere Verkehrselemente verfügbar waren. Die POI-Kategorien wurden mittels Spearman-Rangkorrelation mit ihrem Zusammenhang zum IMD-Wert gemessen. In Abbildung 3-5 ist das Ergebnis dargestellt. Alle Korrelationen sind dabei statistisch signifikant. Auffällig ist, dass insbesondere internationale Restaurants eine hohe Korrelation zum Verdrängungswert aufweisen, die jedoch stark abhängig von der Art des Restaurants ist. So sind einige Restaurant-Arten positiv und andere negativ korreliert.

Themes	Category	Greater London	Greater Manchester	West Midlands
<i>Foursquare</i>				
Health harmful food	Fried Chicken Fast Food Wings	0.31 0.22 0.11	0.15 0.22 0.31	0.19
Faith	Mosque Church	0.27 -0.18	0.22 -0.15	
Non-local cuisines	African Caribbean Asian Italian Indian Spanish Chinese	0.32 0.37 0.23 -0.26 -0.27 -0.20 -0.22		0.25 0.21 0.23 -0.36 -0.17 -0.20 -0.25
Beauty & aesthetics	Dentist's Office Nail Salon Salon Barbershop	-0.22 -0.15	-0.21 -0.17 -0.35	-0.15 -0.19
Sports	Golf Course Cricket Tennis Court	-0.24 -0.13	-0.28 -0.42 -0.23	-0.23
Open spaces	Other Outdoors Lake Campground Field Playground Trail Outdoors and Recreation	-0.15 -0.12 -0.15	-0.24 -0.13 -0.22 -0.23 -0.22 -0.21 -0.14	-0.25 -0.23
Bus service	Bus Bus Station Bus Stop	0.15 0.28 0.18		0.23 0.32 0.24
<i>OSM</i>				
Road system elements	traffic signals crossing mini roundabout	0.29 0.25	0.25	0.24

Abbildung 3-5: POI-Kategorien und Spearman-Korrelationen zu Verdrängungsindikator IMD (Venerandi et al., 2015, S. 261)

Hristova et al. (2016) haben in einer weiteren Studie für London ein Geo-Soziales Netzwerk aufgebaut, um die soziale Vielfalt zu messen. Hierzu nutzten sie Foursquare Check-Ins, kombiniert mit Twitter-Accounts. Dabei wurden 432.929 Verbindungen zwischen 36.926 Usern kombiniert mit 549.797 Check-Ins in 42.080 POIs. Die POIs wurden typisiert und in neun Kategorien zusammengefasst: *Arts, Study, Food, Nightlife, Outdoors, Professional, Residences, Shops* und *Travel*. Der eine Teil der Analyse bestand darin, die Top-Bridging und Top-Bonding POI-Typen je Kategorie zu analysieren. Damit sollte untersucht werden, welche Orte potentiell zum Kennenlernen von neuen Personen geeignet sind, und zu welchen Orten Freunde zusammen hingehen. Hierzu wurden vier Kennzahlen aufgestellt:

- Brokerage – Das Potential, neue Personen in einem POI kennenzulernen
- Entropy – Gibt an, wie wahrscheinlich Personen ein POI betreten
- Homogeneity – Gibt an, wie ähnlich sich die Personen, die das POI besuchen, sind
- Serendipity – Die Chance, dass sich zwei Personen zufällig in einem POI treffen

In einem zweiten Schritt wurden dann mittels Spearman-Rangkorrelation der Zusammenhang der generierten Kennzahlen mit dem IMD, sowie dessen Unterkennzahlen untersucht. In Abbildung 3-6 sind die statistisch signifikanten Korrelationen dargestellt. Insbesondere der Brokerage-Wert weist eine hohe Korrelation zum IMD-Wert auf. Die Autoren gehen davon aus, dass die hohe Anzahl von sich unbekannten Besuchern ein Zeichen dafür ist, dass sich ein Stadtteil wandelt und neue Personen hinzuziehen.



Abbildung 3-6: Spearman-Korrelation zwischen Kennzahlen von Geo-Sozialem Netzwerk und Verdrängungsindikator IMD (Hristova et al., 2016)

In einer Studie von Zukin et al. (2015) wurde der Zusammenhang zwischen Restaurant-Reviews auf dem Empfehlungsportal Yelp mit Gentrifizierung untersucht. Die Studie hat eine manuelle, qualitative Analyse der Rezensionen in zwei Nachbarschaften in Brooklyn, New York vorgenommen. Beide Nachbarschaften werden als „up and coming“ bezeichnet und gelten als Gebiete der Gentrifizierung. Untersucht wurde der Unterschied zwischen den Rezensionen in den Nachbarschaften. Dabei wurde ein Unterschied zwischen dem eher afroamerikanisch geprägtem Bed-Stuy und dem „weißen“ Greenpoint festgestellt. In den Rezensionen wurde nach Hinweisen über die Nachbarschaft der Restaurants gesucht. Dabei wurden die Reviews der Top-10 der meist rezensierten, sowie die Top-10 der „traditionellen“ Restaurants, untersucht. Als traditionell galten solche Restaurants, welche die typische Küche für die afroamerikanischen bzw. weißen Einwohner anbieten. Im Ergebnis hatten 8% der Kritiken im „weißen“ Greenpoint und 24% der Kritiken im „schwarzen“ Bed-Stuy einen Bezug auf die Gentrifizierung der Nachbarschaft. Beide Gebiete hatten Aussagen zur Gentrifizierung, wobei Greenpoint eher gelobt wurde für seine europäisch, polnische Kultur. Bed-Stuy dagegen wurde dafür gelobt, dass es sich ändert und nicht mehr als so gefährlich wahrgenommen wird wie früher (Zukin et al., 2015, S. 469–472).

Tabelle 3-1: Übersicht Big Data Analysen zu Gentrifizierung

Autor	Ort	Bestimmung Gentrifizierung	Big Data
Beekmans (2011)	Amsterdam	qualitative Bestimmung	Foursquare POIs und Checkins
Schaefer (2014)	Los Angeles	qualitative Bestimmung	Twitter Tweets (08.2013-01.2014)
Venerandi et al. (2015)	London, Birmingham, Manchester	IMD Status (2011)	Foursquare POIs und OSM POIs (04.2014 / 05.2014)
Hristova et al. (2016)	London	IMD Verbesserung (2010 - 2015)	Geosoziales Netzwerk basierend auf Foursquare Restaurant-POIs und Checkins, sowie Twitter Tweets (12.2010-09.2011)
Zukin et al. (2015)	Zwei Teile von Brooklyn, New York	qualitative Bestimmung	Yelp Restaurantreviews, manuelle/qualitative Analyse (05.2014)

In Tabelle 3-1 sind die bisherigen Arbeiten noch einmal kurz dargestellt. Die Arbeit von Zukin et al. ist im eigentlichen Sinne kein Spatial Big Data Analytics, da eine manuelle Analyse der Rezensionen vorgenommen wurde. Es fällt auf, dass nur zwei der Autoren bei der Bestimmung der Gentrifizierung auf Zensus oder ähnlichen Daten basieren. Somit ist bei den übrigen die Aussagekraft deutlich geringer. Insbesondere die Arbeiten von Beekmans und Schaefer wurden nur visuell ausgewertet, es wurden keine statistischen oder Data Mining Verfahren verwendet. Die Arbeiten von Hristova et al. und Venerandi et al. basieren dagegen auf bewährten öffentlich erhobenen Daten und nutzen statistische und Machine Learning Verfahren bei der Analyse.

Im Vergleich zu den Arbeiten zur Messung von Gentrifizierung fällt auf, dass keine der Arbeiten die Dimension der immobilienwirtschaftlichen Aufwertung in die Analyse einbezogen hat. Lediglich in der Arbeit von Hristova et al. (2016) wurde ein zeitlicher Bezug hergestellt, indem die Verbesserung des IMD-Wertes anstelle des Status genutzt wurde. Es wurde jedoch kein Bezug im Sinne des Phasenmodells nach Dangschat (1988) hergestellt. Für diese Arbeit soll die Studie von Venerandi et al. (2015) als Vorbild dienen. Das Vorgehen soll um die Untersuchung eines zeitlichen Zusammenhangs erweitert werden, und weitere Machine Learning Methoden nutzen.

4 Data Understanding – Analyse potentieller Datenquellen

Dieses Kapitel soll dem CRISP-DM Schritt *Data Understanding* entsprechen und die erste Forschungsfrage „*Welche Datenquellen sind für die Analyse von Gentrifizierung nutzbar?*“ beantworten.

4.1 Adressen und Räume

Um Analysen mit raumbezogenen Daten durchführen zu können, müssen diese in räumlichen Objekten vorliegen. Dies können zum einen Punkte, wie die Adresse eines Restaurants oder dessen Lage in einem Koordinatensystem sein. Ein weiteres räumliches Objekt sind Flächen, wie zum Beispiel Postleitzahlengebiete. Dabei liegen Punkte jeweils innerhalb einer Fläche, wie beispielsweise ein Restaurant, das innerhalb eines Postleitzahlengebietes liegt.

Neben Adressen wie sie in Deutschland bekannt sind, können Positionen auf Karten auch über Koordinaten repräsentiert werden. Eine Möglichkeit diese darzustellen sind geographische Koordinaten nach dem World Geodetic System 1984 (WGS84), wie sie auch bei GPS verwendet werden. Dabei wird die Position über Längen- und Breitengrad angegeben. Die Koordinaten können auf verschiedene Weisen dargestellt werden, entweder klassisch über Grad, Minute, Sekunde oder über Dezimalgrade, wobei der Breitengrad auch als *Latitude* und der Längengrad als *Longitude* bezeichnet werden (www.c-dev.ch, 2012). Die Dezimalwerte werden typischerweise in IT-Systemen verwendet.

Tabelle 4-1: Alternative Positionsdarstellungen

Adresse	Platz der Republik 1 11011 Berlin
WGS84-Koordinaten (Grad, Dezimalgrad)	52.518332, 13.375706
WGS84-Koordinaten (Grad, Dezimalminute)	N52° 31.1 E13° 22.54237
WGS84-Koordinaten (Grad, Minuten, Sekunden)	N52° 31' 06.0" E13° 22' 32.5"
Google Plus Code	9F4MG99G+87
Google Plus Code alternativ	G99G+87, Berlin
Geohash (12-stellig)	u33db2evkw9m

Für die Dezimalkoordinaten gibt es wiederum verschiedene Hashwert-Abbildungen. Eine Übersicht findet sich in Tabelle 4-1; hier wurde die Position des Berliner Bundestages in verschiedenen Formen dargestellt. Mit dem Geohash und dem von Google entwickelten Plus Code sollen nun zwei Hashverfahren der Dezimalkoordinaten beschrieben werden.

Der Geohash gibt eigentlich nur ein Gebiet an. Ein zwölfstelliger Geohash hat jedoch eine Genauigkeit von ca. sieben Quadratzentimetern (Veness, 2018), weshalb auch Adressen dargestellt werden können. Mit dem Weglassen der hinteren Werte wird der Bereich jeweils größer. Diesen Effekt macht sich unter anderem Elasticsearch zu Nutze, indem über diesen Hashwert aggregiert wird (elastic.co, o.D.). In Tabelle 4-2 ist der Hashwert des Bundestages und der umgebenden Raumgrößen dargestellt. In Abbildung 4-1 ist der Einstieg in die Weltkarte mit Geohash-Raster zu sehen.

Tabelle 4-2: Geohash Zellengröße, Beispiel Bundestag (Veness, 2018)

Teil des Bun- destags - Geo- hashs	Geo- hash lengt h	Cell width	Cell height
u	1	$\leq 5,000\text{km}$	5,000km
3	2	$\leq 1,250\text{km}$	625km
3	3	$\leq 156\text{km}$	156km
d	4	$\leq 39.1\text{km}$	19.5km
b	5	$\leq 4.89\text{km}$	4.89km
2	6	$\leq 1.22\text{km}$	0.61km
e	7	$\leq 153\text{m}$	153m
v	8	$\leq 38.2\text{m}$	19.1m
k	9	$\leq 4.77\text{m}$	4.77m
w	10	$\leq 1.19\text{m}$	0.596m
9	11	$\leq 149\text{mm}$	149mm
m	12	$\leq 37.2\text{mm}$	18.6mm



Abbildung 4-1: Weltkarte Geohash (Veness, 2018)

Google wiederum hat mit dem *Plus Code* das Ziel verfolgt eine alternative Adressform zu entwickeln, um Menschen eine Adresse zu geben, die sonst keine Adresse hätten, da es zum Beispiel keine Straße gibt. Der Plus Code teilt sich in zwei Bereiche: Die ersten vier Stellen ergeben den *Area Code*, welcher ein Gebiet von $1.000 * 1.000$ Kilometern definiert. Die nächsten sechs Stellen geben dann den *Local Code* mit einem Gebiet von ca. $14 * 14$ Metern. Zusammen ergeben Area und Local Code den *Global Code*. Der Area Code wäre zum Beispiel für Taxifahrten oder ähnliche lokale Adressen unwichtig. Ähnlich wie beim Geohash kann die Genauigkeit mit weiteren hinten angefügten Stellen vergrößert werden, beide Systeme arbeiten mit in sich verschachtelten Rastern. In jedem Feld ist das Raster in gedrehter Form erneut enthalten (Google, 2018; plus.codes, o.D.). Das Geohash Raster entspricht einer 4x8 Matrix, das vom Plus Code einer 4x5 Matrix. Alle zwei Stellen wird also die Auflösung um den Faktor 32 bei Geohash und 20 bei Plus Code geringer.

Für die maschinelle Verarbeitung der Daten müssen diese in Zeilen und Spalten vorhanden sein. Im Falle der Untersuchung von Gentrifizierung müssen also die Daten über soziale Kennzahlen wie Arbeitslosigkeit oder Kinderarmut, Wanderungsbewegungen und weitere Datenfeatures auf den gleichen Raum bezogen sein; der Raum bildet folglich einen Teil des Primärschlüssels der Datenzeile. Eine Möglichkeit wäre, die oben genannte Form der Aggregation über den Plus Code oder Geohash zu nutzen. Verschiedene Datenquellen sind jedoch häufig in unterschiedlichen Raumbezugsgrößen aufgelöst. Dieses Problem wird auch als *Modifiable Area Unit Problem* (MAUP) bezeichnet, worauf bereits in Kapitel 2.2 eingegangen wurde. Da neben Positionsdaten keine Daten auf einfache Weise in das Geohash-Format gebracht werden können, soll im Folgenden nun auf verschiedene mögliche nationale und lokale Raumbezugsgrößen eingegangen werden, die für eine Analyse für Gentrifizierung in Berlin möglich wären.

Postleitzahlen (PLZ) sind fest im Alltag verwurzelt und jedem bekannt. Mit einem Blick auf die Postleitzahlengebiete fallen in Berlin jedoch teilweise kurios erscheinende Flächen auf. Einige Postleitzahlen überschreiten Bezirksgrenzen, wie die PLZ 10119, andere dringen tief in Nachbar-Gebiete ein, wie 10179 in 10178 oder 101437 in 10439 (vgl. Abbildung 9-3: Postleitzahlen in Berlin im Anhang).¹²

Bis 2006 wurde die Raumeinheit der Verkehrszellen und Teilverkehrszellen als statistische Raumbezugsgröße für sozialräumliche Planung genutzt (Senatsverwaltung für Umwelt, Verkehr und Klimaschutz, o.D.). Heute werden laut Ideation & Prototyping Lab der Technologiestiftung Berlin (2018) die Verkehrszellen nur noch für Förderprojekte genutzt, da sich diese bereits in europäischen Prozessen etabliert haben. Für die meisten Analysen werden jedoch die 2006 neu entwickelten lebensweltlich orientierten Räume (LOR) genutzt.

Die LOR wurden im Jahr 2006 als „*neue räumliche Grundlage für Planung, Prognose und Beobachtung demografischer und sozialer Entwicklungen in Berlin festgelegt*“ (Senatsverwaltung für Stadtentwicklung und Wohnen Berlin, o.D.). Dabei wurde eine Hierarchie bestehend aus 447 *Planungsräumen* (PLR), 138 *Bezirksregionen* (BZR) und 60 *Prognoseräumen* (PRG) gebildet, welche sich in die Grenzen der zwölf Berliner Bezirke gliedern. Diese Hierarchie ist auch in der ID eines jeden Raums enthalten. Als Beispiel hat der Planungsraum Wrangelkiez die PLR-ID 02030402:

¹² Die PLZ-Daten stammen von Ideation & Prototyping Lab der Technologiestiftung Berlin (2018), Urheber der Daten ist das Amt für Statistik Berlin-Brandenburg.

Bezirk	02	Friedrichshain-Kreuzberg
Prognoseraum	03	Kreuzberg Ost
Bezirksregion	04	südliche Luisenstadt
Planungsraum	02	Wrangelkiez

Werden nun die zwei hinteren Ziffern weggelassen, ergibt sich die räumliche Auflösung der Bezirksregion. Die LOR-Hierarchie verbindet also für Berlin die Aggregations-Vorteile des Geohashes mit denen einer fachlich erzeugten Raumgröße, auf deren Basis auch statistische Informationen bereitgestellt werden. So wurden die LOR unter Berücksichtigung von einheitlichen Baustrukturen, natürlichen Barrieren wie Gewässern und großen Straßen aufgebaut. Dabei wurde versucht, vergleichbare Einwohnerzahlen in den Planungsräumen zu erzielen (Senatsverwaltung für Stadtentwicklung und Wohnen Berlin, o.D.). In Abbildung 4-2 ist ein Box-Whisker-Plot mit den Einwohnerzahlen je Planungsraum zu sehen. Darin ist zu erkennen, dass zwar 50% der Planungsräume Einwohnerzahlen zwischen 4.283 und 10.867 haben, die Verteilung der Einwohnerzahlen jedoch durchaus heterogen ist.

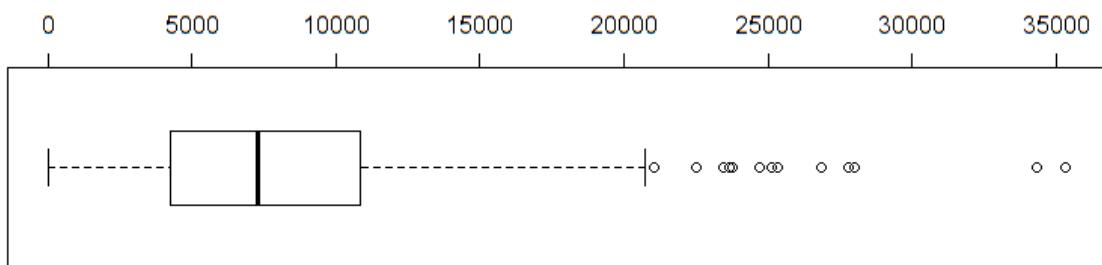


Abbildung 4-2: Box-Whisker-Plot der Einwohnerzahlen zum 31.12.2016 in den 443 Planungsräumen

Für diese Arbeit wird die LOR-Hierarchie verwendet, da die öffentlichen Datenquellen auf dieser Ebene statistische Daten veröffentlichen. Die Geometriedaten hierzu stammen vom Amt für Statistik Berlin-Brandenburg und sind verfügbar bei der Senatsverwaltung für Stadtentwicklung und Wohnen Berlin oder beim Ideation & Prototyping Lab der Technologiestiftung Berlin (2018).

4.2 Datenquellen

Im Folgenden werden Datenquellen analysiert, mit denen sich die dreidimensionale Definition von Gentrifizierung abbilden lässt.¹³ Diese besteht aus immobilienwirtschaftlicher Aufwertung, verdrängungsinduzierter sozialer Aufwertung und der Veränderung der lokalen Angebotsstrukturen.

4.2.1 Öffentliche Datenquellen zur sozialen Aufwertung

Die öffentliche Hand besitzt viele Daten, mit denen sich die soziale Aufwertung ablesen lassen könnte. Mit dem *Geodatenzugangsgesetz* (GeoZG) hat die Bundesregierung im Jahr 2009 die EU-Richtlinie *Infrastructure for Spatial Information in the European Community* befolgt und ein *Gesetz über den Zugang zu digitalen Geodaten* erlassen (Wikipe-dia, 2018a). Das Gesetz sieht vor, dass alle Geodaten der öffentlichen Hand veröffentlicht werden, unter anderem Daten zur Demographie, Bodennutzung, Geologie, Umweltüber-wachung und vieles mehr (§ 4 GeoZG, 2009). Für Geodaten der Länder und Kommunen wurden jeweils ähnliche Gesetze erlassen.

Das GeoZG steht neben dem *Informationsfreiheitsgesetz* (IFG), welches 2005 erlassen wurde. Das IFG regelt nur den Zugang zu Informationen des Bundes. Einige Länder ha-ben ähnliche Gesetze, andere gar keine. Der Bund, wie auch einige der Länder, erhebt Gebühren für die Auskunft im Rahmen des IFG (§ 16 IFG / *Landesnorm Berlin*, 1999; § 10 IFG, 2005). Im Falle von Geodaten, sowie Metadaten mit Georeferenz, müssen die Daten explizit „für die kommerzielle und nicht kommerzielle Nutzung geldleistungsfrei zur Verfügung [...]“ (§ 11 GeoZG, 2009) gestellt werden.

Berlin stellt seine Daten über das eigene Open Data Portal www.daten.berlin.de zur Ver-fügung (*Offene Daten Berlin*, o.D.). Die Daten sind jedoch auch im Datenportal für Deutschland www.govdata.de verlinkt (*GovData*, o.D.). Alle untersuchten Daten sind über eine offene Namensnennungslizenz wie die Creative Commons Namensnennung 3.0 (CC-BY-3.0)¹⁴ veröffentlicht worden.

4.2.1.1 Mikrozensus und Außenwanderung

Berlin hat die Daten des jährlich durchgeführten Mikrozensus in einem web-basierten multidimensionalen Abfragetool unter www.statistik-berlin-brandenburg.de/webapi be-reitgestellt (Amt für Statistik Berlin-Brandenburg, o.D.a). Hier können alle Daten des

¹³ Vgl. Kapitel 3.1

¹⁴ <https://creativecommons.org/licenses/by/3.0/de/>

Mikrozensus abgefragt werden. Insbesondere Daten zu Bildungsabschlüssen, Berufsständen und Einkommen, von Personen sowie Haushalten, wären für die Analyse von Gentrifizierung in dieser Arbeit von Interesse.

Die Daten sind allerdings nur auf der räumlichen Detailebene der zwölf Berliner Bezirke verfügbar. Da dies zu grob für eine kleinräumige Analyse ist, können die Daten aus dem Mikrozensus nicht verwendet werden. Gleiches gilt für die Wanderungsdaten, welche auch im oben genannten Portal verfügbar sind.

4.2.1.2 Einwohnerregister

Aus dem Einwohnerregister (EWR) von Berlin werden jährlich verschiedene Daten auf PLR-Ebene als CSV-Datei veröffentlicht. In der Veröffentlichung *Melderechtlich registrierte Einwohner mit Hauptwohnsitz* wird je PLR die Anzahl aller Einwohner veröffentlicht. Neben einer Aufteilung in die Geschlechter männlich und weiblich findet zudem eine Aufteilung nach Alter in 32 Altersgruppen statt, zum Beispiel 0-1 Jahr oder 75-80 Jahre. Zudem gibt es eine vorgefertigte Aggregation in 9 Altersklassen. Die Datei *Melderechtlich registrierte Einwohner mit Hauptwohnsitz - Ausländer insgesamt* ist analog aufgebaut, enthält jedoch nur die Daten der ausländischen Einwohner. Als Ausländer gelten „Personen mit ausschließlich ausländischer oder ungeklärter Staatsangehörigkeit und Staatenlose“ (Amt für Statistik Berlin-Brandenburg, 2012, S. 4). Gleiches gilt für die Datei *Melderechtlich registrierte Einwohner mit Hauptwohnsitz - Einwohner mit Migrationshintergrund insgesamt (E)*, welche die Daten aller Einwohner mit Migrationshintergrund enthält. Darin sind alle Ausländer enthalten, sowie alle Deutschen mit Migrationshintergrund. Die Datei *Melderechtlich registrierte Einwohner mit Hauptwohnsitz - Einwohner mit Migrationshintergrund nach Herkunftsgebiet (H)* enthält eine Aufgliederung der Einwohner mit Migrationshintergrund nach Herkunftsgebieten (Amt für Statistik Berlin-Brandenburg, 2014). Neben diesen Informationen gibt es zusätzliche Veröffentlichungen zur *Wohndauer* und *Wohnlage*. Bei der Wohndauer wird angegeben wie viele Bewohner schon mindestens 5 bzw. 10 Jahre an der gleichen Anschrift wohnen. In der Veröffentlichung zur Wohnlage wird die Anzahl der Einwohner nach guter, mittlerer und schlechter Wohnlage, jeweils mit und ohne Lärm aufgeteilt (Amt für Statistik Berlin-Brandenburg, 2011).

4.2.1.3 Monitoring Soziale Stadtentwicklung

Der Bericht Monitoring Soziale Stadtentwicklung (MSS) der Stadt Berlin erscheint alle zwei Jahre und zeigt die „Veränderungen der sozistrukturellen Entwicklung“ auf Ebene von Planungsräumen (Senatsverwaltung für Stadtentwicklung und Wohnen, 2017). In

dem Bericht werden ein Status- und ein Dynamik-Index über die vier Kennzahlen *Arbeitslose*, *Langzeitarbeitslose*, *Transferbezieher* und *Transferbezieher unter 15 Jahre* gebildet. Der Dynamik-Index basiert jeweils auf der Veränderung der letzten zwei Jahre. Neben den Indexwerten werden im gleichen Zuge auch sogenannte Kontextindikatoren veröffentlicht. Diese behandeln die Handlungsfelder *Besondere, von Armut bedrohte Zielgruppen*, *Integration*, sowie *Wohnen und Stabilität der Wohnbevölkerung*. In Tabelle 4-3 sind die einzelnen Kontextindikatoren aufgeführt. Besonders interessant sind dabei die Informationen des Wanderungssaldos, welcher sonst nur in stark aggregierter Form auf Bezirksebene vorliegt.

Tabelle 4-3: Kontextindikatoren des MSS nach Handlungsfeld (Eigene Darstellung nach Senatsverwaltung für Stadtentwicklung und Wohnen, 2017)

Armut	K 01	Jugendarbeitslosigkeit
	K 02	Alleinerziehende Haushalte
	K 03	Altersarmut
Integration	K 04	Kinder und Jugendliche mit Migrationshintergrund
	K 05	Einwohnerinnen und Einwohner mit Migrationshintergrund
	K 16	Ausländerinnen und Ausländer
	K 06	Veränderung Ausländeranteil
	K 17	Nicht-EU-Ausländerinnen und Nicht-EU-Ausländer
	K 07	Ausländische Transferbeziehende
Wohnen	K 08	Städtische Wohnungen (landeseigene Wohnungen / alle Wohnungen)
	K 14	Wohnräume (Wohnräume / Einwohner)
	K 15	Wohnfläche (Wohnfläche in m ² / Einwohner)
	K 09	Einfache Wohnlage (einschl. Lärmbelastung)
	K 10	Wohndauer über 5 Jahre
	K 11	Wanderungsvolumen Zuzüge plus Fortzüge pro 100 Einwohnerinnen und Einwohner)
	K 12	Wanderungssaldo (Zuzüge minus Fortzüge je 100 Einwohnerinnen und Einwohner)
	K 13	Wanderungssaldo von Kindern unter 6 Jahren

Der MSS ähnelt dem Sozialindex von Holm und Schulz (2016), welcher sich auf dem Bestand an Transferbeziehern beschränkt. Der MSS kann somit selbst als vereinfachter Wert der Gentrifizierung genommen werden, der jedoch nur die soziale Aufwertung betrachtet. Dabei ist sowohl der Status-Index, als auch der Dynamik-Index von Interesse. Mit den öffentlichen Informationen aus dem MSS-Indexindikatoren, den MSS-Kontext-

indikatoren und den EWR-Daten lassen sich die Indexwerte zur Mobilität und zur Veränderung der Bevölkerungsstruktur nach Döring und Ulbricht (2016) nachbilden, es fehlen lediglich Informationen zur Kaufkraft.¹⁵ Damit ließe sich die Gentrifizierungs-Dimension der verdrängungsinduzierten sozialen Aufwertung messen.¹⁶

4.2.1.4 Bestand an Kraftfahrzeugen je Gemeinde

Um die Kaufkraft der Einwohner zu bestimmen, könnten Angaben zu den Kraftfahrzeugen der Einwohner ggf. einen Einblick geben. Das Kraftfahrt Bundesamt veröffentlicht jährlich den Bericht *Bestand an Kraftfahrzeugen und Kraftfahrzeughängern nach Gemeinden (FZ 3)*. Darin ist der Bestand an Kraftfahrzeugen unterteilt nach Typ (Krafträder, Personen-, Lastkraftwagen, ...) je Gemeinde ausgeführt. Eine Gemeinde ist dabei zum Teil unterhalb der Postleitzahlregion ausgewiesen. Allerdings ist für einige Großstädte, zu denen auch Berlin gehört, nur ein aggregierter Wert über das gesamte Stadtgebiet verfügbar (Kraftfahrt-Bundesamt, o.D.).¹⁷

Da für Berlin die räumliche Auflösung zu grob ist und für die Rückschlüsse auf die Kaufkraft mehr Informationen wie Marke und Baujahr benötigt werden würden, scheidet diese Datenquelle aus.

4.2.2 Immobilienwirtschaftliche Daten von ImmobilienScout24

Eine häufig empfohlene Möglichkeit die immobilienwirtschaftliche Aufwertung zu messen besteht darin, Angebotsmieten zu nutzen.¹⁸ Hierfür könnten Daten aus online Immobilienportalen genutzt werden. ImmobilienScout24 ist laut immobilienportale.com (o.D.) Marktführer unter den Immobilienportalen und wurde bereits in der Arbeit von Schulz (2015) als Datenquelle genutzt. ImmobilienScout24 bietet eine Reihe verschiedener Programmierschnittstellen (APIs) an, welche für raumbezogene Analyse Systeme als Datenquellen genutzt werden könnten. Für die Analyse von Gentrifizierung würde sich dabei die *Pricehistory API* anbieten, welche durchschnittliche Preise im Zeitverlauf darstellt. Dabei können je nach API-Aufruf die Preise pro Quadratmeter von Käufen oder Mieten für Wohnungen oder Häuser angezeigt werden. Des Weiteren kann zwischen Bestand und Neubau unterschieden werden. Die Daten sind für jedes Quartal seit 2007 verfügbar (Immobilien Scout GmbH, o.D.b.).

¹⁵ Vgl. Kapitel 3.4

¹⁶ Vgl. Kapitel 3.1

¹⁷ Vgl. im Anhang Tabelle 9-4: Auszug aus „Bestand an Kraftfahrzeugen und Kraftfahrzeughängern nach Gemeinden (FZ3)“ (Kraftfahrt-Bundesamt, o.D.)

¹⁸ Vgl. Kapitel 3.1

Die Raumbezugsgröße ist über die *Geohierarchy API* abfragbar. Diese gibt Auskunft über die Hierarchie der Regions-Dimension bei ImmobilienScout24. Die Hierarchie hat fünf Ebenen: *Continent, Country, Region, City* und *Quarter* (Immobilien Scout GmbH, o.D.a). Wenn zum Beispiel die Quarter für Berlin abgefragt werden, erhält man die IDs und Namen aller Quarter in Berlin. Das Quarter Berlin Mitte hätte beispielsweise den Aufbau:

continent/1/country/276/region/3/city/1/quarter/46

Die API gibt jedoch keine Information darüber, welche Form die einzelnen Quarter haben. Für Berlin sind 81 Quarter¹⁹ definiert, die so auch in der Suchmaske für den Anwender verfügbar sind. Beim Vergleich mit den LOR-Gebieten fällt auf, dass einige Räume gleich sind, was für eine einfache Datenintegration sprechen würde.²⁰ Für das Beispiel Kreuzberg ist ersichtlich, dass die drei Prognoseräume 0201, 0202 und 0203 dem Quarter Kreuzberg entsprechen.²¹ Bei anderen Gebieten ist im Detail eine Unschärfe zu erkennen. So entsprechen beispielsweise die Quarter Haselhorst und Siemensstadt zwar beinahe den Bezirksregionen 050307 bzw. 050308, sind jedoch in Abbildung 4-3 erkennbar unterschiedlich.

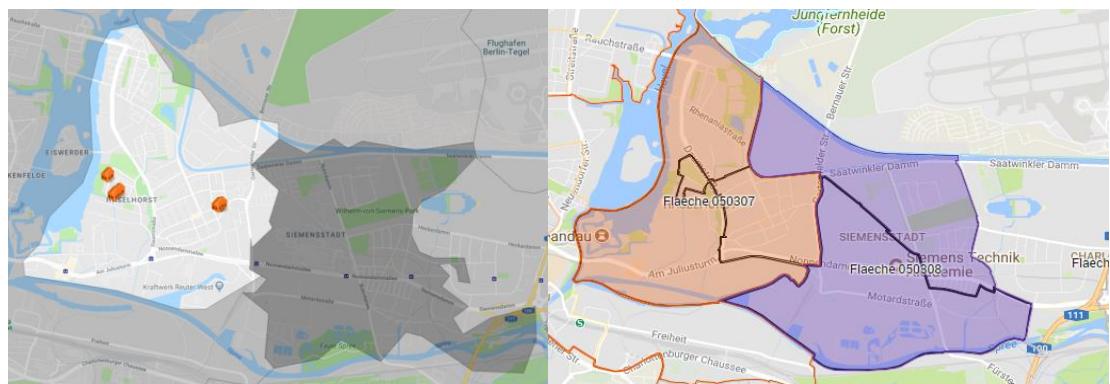


Abbildung 4-3: Vergleich Haselhorst und Siemensstadt (links, ImmobilienScout24) mit Bezirksregionen (rechts, eigene Darstellung mit Google MyMaps)

Auch eine weitere Zergliederung auf Ebene der Prognoseräume (markiert mit schwarzen Gebietsgrenzen) gäbe hier keine genaue Überdeckung. Des Weiteren fällt auf, dass die Quarter unterschiedlich groß sind. So ist Kreuzberg sehr großflächig und enthält drei Prognoseräume, beziehungsweise vier Bezirksregionen oder auch 20 Planungsräume. Bei

¹⁹ Die vollständige Liste der Quarter befindet sich im Anhang unter Tabelle 9-1: Liste der ImmobilienScout24 Quarter in Berlin

²⁰ Vgl. im Anhang Abbildung 9-4: Vergleich ImmobilienScout24 (oben, ImmobilienScout24) mit Prognoseräumen (unten, Eigene Darstellung mit Google MyMaps nach Amt für Statistik Berlin-Brandenburg)

²¹ Vgl. im Anhang Abbildung 9-5: Vergleich Kreuzberg ImmobilienScout24 (links, ImmobilienScout24) mit Prognoseräumen (rechts, eigene Darstellung mit Google MyMaps nach Amt für Statistik Berlin-Brandenburg)

ImmobilienScout24 entspricht Kreuzberg jedoch nur einem Quarter. Dagegen sind Haselhorst und Siemensstadt ähnlich groß wie eine Bezirksregion und enthalten nur jeweils zwei Planungsräume.

Erschwerend kommt hinzu, dass ImmobilienScout24 keine Raumdaten zur Verfügung stellt, mit denen sich die Quater in geometrischen Polygonen abbilden lassen. Dies wäre jedoch nötig, um geographische Berechnungen auszuführen. Eine Integration von Daten dieser unterschiedlichen Raumbezugsgrößen bei fehlenden Polygon-Daten würde einen hohen Datenaufbereitungsaufwand bedeuten. Hierzu müsste zunächst manuell ein Mapping von räumlichen Einheiten auf die 81 Quarter geschehen. Da die Gebiete jedoch nicht immer direkt überführbar sind, müsste ein Wert aus den überlagernden Flächen errechnet werden.

4.2.3 Datenquellen zur Angebotsstruktur

Für die Analyse der Angebotsstruktur bieten sich insbesondere Karten- und Empfehlungsdienste an, da diese viele Daten über Restaurants und andere POIs haben.

4.2.3.1 Kommerzielle Karten- und Empfehlungsdienste

Die Dienstleister Google (Google Developers, 2018), Foursquare (Foursquare, o.D.), Tripadvisor (TripAdvisor Developer Portal, o.D.) und Yelp (Yelp Fusion API, o.D.b) bieten APIs an, um ihre POIs und Rezensionen abzufragen.²²

Tabelle 4-4: Übersicht APIs der kommerziellen Karten- und Empfehlungsdienste

Anbieter	POI Umkreisssuche (maximale Anzahl an POIs in Response)	POI-Details	POI-Katego- rien	POI- Hierar- chie
Google	60	Position, Stammdaten, Preis- segment, Userrating	90	Nein
Four- square	50	Position, Stammdaten, Preis- segment, Userrating, Erfas- sungsdatum	925	Ja
Trip- advisor	0 (Nur Suche nach einzelnen POIs möglich)	Position, Stammdaten, Preis- segment, Userrating	473	(Ja)
Yelp	1.000 (nur POIs mit Reviews)	Position, Stammdaten, Preis- segment, Userrating	1.587	Ja

²² Eine detaillierte Analyse der kommerziellen Datenquellen befindet sich im Anhang unter Kapitel 9.3.1

Wie in Tabelle 4-4 zu sehen ist, sind bei allen APIs POI-Informationen wie die genaue Position, Stammdaten (Name, Öffnungszeiten, Adress- und Kontaktdaten, ...), sowie Informationen zu Preissegmenten und durchschnittlichen Nutzerbewertungen verfügbar. Die Yelp-API bietet die Möglichkeit POIs per Umkreissuche systematisch abzufragen. Allerdings gibt es ein Limit von 1.000 Treffern je Aufruf. Durch die Limitierung von 60 bzw. 50 POIs pro Abfrage ist eine systematische Datenerfassung über die Google API bzw. aus Foursquare sehr aufwendig. Ohne die Möglichkeit per API nach mehreren POIs zu suchen, kann die Tripadvisor API nicht für die Analyse der lokalen Angebotsstrukturen genutzt werden.

Bei allen kommerziellen Anbietern ist es möglich einzelne Rezensionen für die POIs abzufragen. Diese wurden bereits von Zukin et al. (2015) manuell ausgewertet. Diese Auswertungen sind auch über sogenannte Sentiment Analysen automatisiert machbar (Hu & Liu, 2004). Sentiment Analysen sind ein Teilgebiet des Text Mining. Dabei werden die Rezensionen in Sätze und diese in Wörter zerlegt. Die Worte werden dann anhand eines vorher definierten Wörterbuchs positiv, negativ oder neutral bewertet. Die Summe der Bewertungen sagt dann aus, ob ein Satz oder die ganze Rezension eher etwas positives oder eher etwas negatives aussagt (Witten et al., 2017, S. 313–314). Für die Analyse auf Hinweise zur Gentrifizierung müsste jedoch ein spezifisches Wörterbuch entwickelt werden, da die meisten Wörterbücher nur die positive oder negative Haltung zu dem POI ermitteln würden. Diese Information könnte auch aus den vorhandenen Wertungen²³ entnommen werden. Die Arbeit von Schaefer (2014) hat gezeigt, dass eine Suche nach Stichworten wie „Gentrifizierung“ oder „Hipster“ eher wenig erfolgversprechend ist.²⁴ Da der Aufbau und das Validieren eines Wörterbuchs für die Analyse von Gentrifizierung den Rahmen dieser Arbeit übersteigen würden, wird auf diese Datenquelle verzichtet.

4.2.3.2 OpenStreetMap

OpenStreetMap ist ein Projekt, bei dem es darum geht, frei verfügbare geographische Daten zu erzeugen und zu teilen. Die Non-Profit-Organisation OpenStreetMap Foundation betreibt die Internetseite www.openstreetmap.org, auf der der aktuellste Bestand der OSM-Daten verfügbar ist. Die Datenerfassung und -pflege der OSM Daten wird über die OSM Community betrieben (OpenStreetMap Foundation, o.D.). Seit Projektbeginn im Jahr 2005 waren bei OSM eine Million Nutzer aktiv an der Weiterentwicklung des Kartenmaterials beteiligt. Dabei werden im Schnitt drei Millionen Änderungen des Kartenmaterials am Tag vorgenommen.²⁵ Da OSM als Open Data lizenziert ist, wird der OSM

²³ Zum Beispiel 1-5 Sterne bei Google Maps

²⁴ Vgl. Kapitel 3.5

²⁵ Stand 08.11.2017 <https://wiki.openstreetmap.org/wiki/Stats>

Datenbestand auch von weiteren Servern aus verteilt. Diese sogenannten *Mirror* spiegeln die Version des Quellservers in regelmäßigen Update-Zyklen. Einige der Mirror extrahieren nur die Daten für Teilregionen der Welt, wie Europa oder Deutschland (OpenStreetMap contributors, o.D.). Einer dieser Anbieter ist die Geofabrik GmbH, welche die OSM-Daten täglich auf Ebene der Bundesländer bereitstellt. Zusätzlich werden auch die historischen Datenbestände seit dem 01.01.2014 jährlich bereitgestellt (Geofabrik GmbH & OpenStreetMap contributors, o.D.).

Die gesamte Welt ist bei OSM in der Datei Planet.osm enthalten, welche zum 01.07.2018 eine Größe von 940 GB hatte. Diese Datei enthält alle Daten, aus denen die Karte besteht. Dabei ist die Datei im XML-Format aufgebaut und hat nur drei feste Datentypen. Ein *Node* ist eine Position mit Koordinate. Ein *Way* ist ein Weg, der als Polylinie abgebildet wird. Ein Way kann jedoch auch als eine geschlossene Polylinie, also als Polygon, eine Fläche kennzeichnen. Dabei besteht ein Way aus einer geordneten Liste zwischen 2 und 2.000 Nodes, welche die Polyline bzw. das Polygon bilden. Eine *Relation* ist eine Verbindung zwischen anderen Elementen, wie zum Beispiel eine Buslinie, die über mehrere Haltestellen (Nodes) und Straßen (Ways) führt (OpenStreetMap contributors, o.D.). In Abbildung 4-4 ist zu erkennen, wie sich die erfassten OSM-Daten im Laufe der Zeit entwickelt haben. So sind Ende 2017 fast 500 Millionen Ways und 4,35 Milliarden Nodes erfasst. Da jeder Way mindestens zwei Nodes enthält, ist die Anzahl an Nodes logischerweise am höchsten.

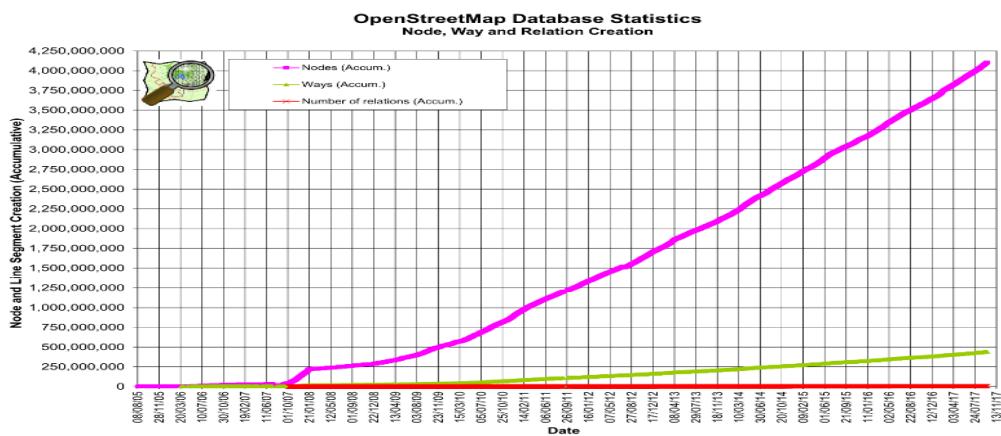


Abbildung 4-4: OSM Statistik – Kumulierte Anzahl erstellter Elemente (OpenStreetMap contributors, o.D.)

Alle Datentypen sind in einer Key-Value Liste, den *Tags*, frei erweiterbar. Damit kann ein Node potentiell ein POI sein, je nachdem welche Tags es enthält. Ein Beispiel für ein

POI-Node ist in Abbildung 4-5 enthalten. In dem Auszug aus der Planet.osm ist ein italienisches Restaurant in Berlin mit Namen, Öffnungszeiten, Adresse, sowie Hinweisen auf Barrierefreiheit für Rollstuhlfahrer dargestellt. Alle Details, bis auf die GPS-Koordinaten (lat, lon) und Versionsinformationen, sind in Tags enthalten. Das zeigt, wie wichtig die Tags für die Auswertung der OSM-Daten sind.

```
<node id="282415700" visible="true" version="15" changeset="60270787"
timestamp="2018-06-29T10:17:12Z" user="wheelmap_visitor" uid="290680"
lat="52.5091398" lon="13.4598832">
  <tag k="addr:city" v="Berlin"/>
  <tag k="addr:country" v="DE"/>
  <tag k="addr:housenumber" v="29a"/>
  <tag k="addr:postcode" v="10245"/>
  <tag k="addr:street" v="Gärtnerstraße"/>
  <tag k="addr:suburb" v="Friedrichshain"/>
  <tag k="amenity" v="restaurant"/>
  <tag k="cuisine" v="italian"/>
  <tag k="name" v="Hannibal"/>
  <tag k="opening_hours" v="Su-Th 09:00-02:00, Fr,Sa 09:00-04:00"/>
  <tag k="smoking" v="outside"/>
  <tag k="toilets:wheelchair" v="yes"/>
  <tag k="wheelchair" v="yes"/>
</node>
```

Abbildung 4-5: XML-Code Planet.osm Beispiel Node (Eigene Darstellung nach OpenStreetMap contributors, o.D.)

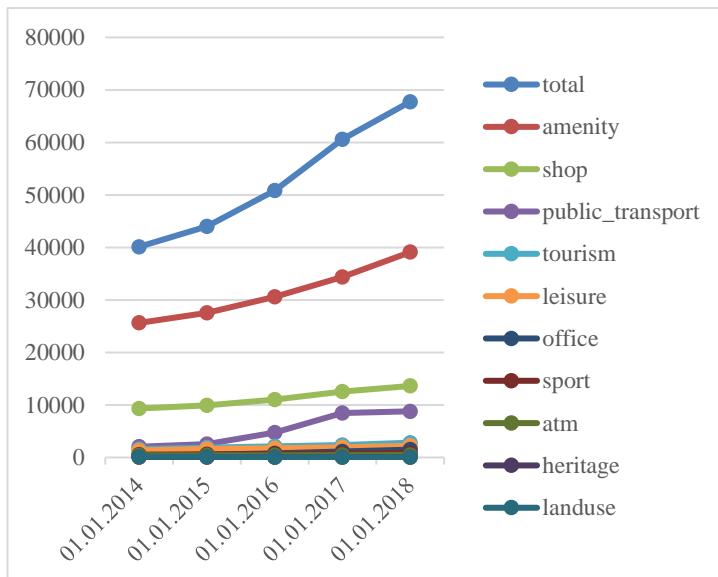
Da die Tags von den Mappern, wie die Erfasser bei OSM auch genannt werden, unterschiedlich gepflegt werden könnten, muss von einer heterogenen Datenqualität ausgegangen werden. Als Hilfe für die Community wurde mit www.taginfo.openstreetmap.org ein Portal geschaffen, auf denen geläufige Key-Value Kombinationen recherchiert werden können, um eine möglichst einheitliche Kategorisierung zu ermöglichen (OpenStreetMap Taginfo, o.D.).

Weil die Daten bereits für das Bundesland Berlin vorliegen, müssen bei OSM keine API-Aufrufe zum Sammeln der Daten erfolgen, sondern ein Selektieren der relevanten POIs aus dem gesamten Datenbestand. Und da die Daten von OSM in mehreren Zeitscheiben vorhanden sind, können die in OSM enthaltenen POIs gut für die Analyse genutzt werden. Allerdings muss die POI-Typen Klassifizierung über die Tags erfolgen.

In dem Projekt OpenPoiMap www.openpoimap.org wurden hierzu bereits viele Tags klassifiziert (*OpenPoiMap*, o.D.). In Kombination mit dem Taginfo Portal wurde damit ein erstes Mapping für die OSM-POIs entworfen. Das Mapping wurde anschließend auf Basis der Datengrundlage in den OSM-Daten verfeinert. So wurden keine Unterkategorien mit verhältnismäßig wenigen POIs im Vergleich zur Oberkategorie gebildet. Im Ergebnis wurde die Tabelle 9-5 aus dem Anhang für die Kategorisierung in drei Ebenen entwickelt. Dabei wurden auf Basis der zehn Keys *tourism*, *amenity*, *heritage*, *sport*,

leisure, office, atm, landuse, public_transport und shop in Summe 199 POI-Typen gebil-det.²⁶

Wertet man die Anzahl POIs wie in Abbildung 4-6 auf Ebene der Keys aus, so fällt schnell auf, dass die Keys *amenity* und *shop* den größten Anteil an den POIs ausmachen. Amenity wird als eine „*Markierung nützlicher und wichtiger Einrichtungen für Besucher und An-*



wohner“ (OpenStreetMap Taginfo, o.D.) beschrieben, shop kennzeichnet ein Geschäft. Zudem fällt auf, dass die Zahl der erfassten Nodes jährlich steigt. Eine Erklärung hierfür könnte die laufende Weiterentwicklung des Kartennmaterials durch die Community sein.

Abbildung 4-6: Entwicklung der Anzahl der OSM-POI in Berlin nach Tag-Key

4.3 Prüfung auf Lizenzrecht und Open Data

Neben der technischen und inhaltlichen Sichtung der Daten und Schnittstellen soll nun auch ein Blick auf die Lizenzen und Nutzungsbedingungen (engl. terms of use) geworfen werden.

In Tabelle 4-5 sind die Bedingungen für das Erfassen und Auswerten der Daten je Datenquelle dargestellt. ImmobilienScout24 besteht auf eine Löschfrist von einem Tag (ImmobilienScout24, 2015, 6.2), was für die Analyse im Rahmen dieser Arbeit ungünstig ist. Google schließt explizit die Speicherung und Auswertung der Daten aus (Google Cloud, 2018, 3.2.4 (a) & (b)). Tripadvisor schließt ebenfalls statistische Analysen auf Basis ihrer Daten aus, zudem muss zwischengespeicherter Inhalt alle 24 Stunden aktualisiert werden, was die Analyse und die Nachvollziehbarkeit der Ergebnisse erschwert (TripAdvisor Developer Portal, 2017, 4.4 (iii) & 4.5). Auch Foursquare besteht auf eine Aktualisierung alle 24 Stunden, zudem ist die Verwendung für kommerzielle oder wissenschaftliche Zwecke nur mit einer Enterprise-Lizenz erlaubt (Foursquare, 2018, I & IV).

²⁶ Das technische Mapping der POI-Typen zu den OSM Tags kann unter https://github.com/dhelweg/masterthesis2018/blob/master/data/OSM_POI_MAPPING.xlsx eingesehen werden

Tabelle 4-5: Analyse Terms of Use der POI-Quellen

Urheber	Daten sammeln			Analytics						
	Nicht-Kommerziell?	Kommerziell?	Wissenschaftlich?	Nicht-Kommerziell?	Kommerziell?	Wissenschaftlich?				
Immobilien-Scout24	24h Löschfrist			keine Angabe						
Yelp	y	24h	y	y	n	y				
Foursquare	24h Löschfrist, maximal 5000 POI- Details pro Stunde	24h Löschfrist, nur mit Enterprise- Lizenz		keine An- gabe	nur mit Enterprise- Lizenz					
Tripadvisor	24h Löschfrist, maximal 10.000 API-Aufrufe am Tag			explizites Verbot						
Google Places API	explizites Verbot			explizites Verbot						
Geofabrik GmbH / OpenStreetMap	y			y						

Yelp hat von den gewerblichen Anbietern die moderatesten Vorschriften, so besteht Yelp nur bei kommerziellen Nutzern auf eine Löschfrist von 24 Stunden. Nicht-kommerziellen Nutzern ist das Sammeln und Auswerten von Yelp-Daten erlaubt (Yelp for Developers, 2018, 6 a.). Genauer heißt es zur nicht kommerziellen Datenanalyse bei Yelp: „*Allowable non-commercial use of the Yelp Content. Notwithstanding the foregoing, you may use the Yelp Content to perform certain analysis for non-commercial uses only, such as creating rich visualizations or exploring trends and correlations over time, so long as the underlying Yelp Content is only displayed in the aggregate as an analytical output, and not individually*“ (Yelp for Developers, 2018, 6 (end)). Das erlaubt es die Daten in Data Mining und Machine Learning Modellen zu nutzen, sofern keine kommerzielle Anwendung vorliegt.

OpenStreetMap ist über die *Open Data Commons Open Database License* (ODbL)²⁷ lizenziert, welche Nutzung, Änderungen und Veröffentlichung der Datenbank erlaubt. Die Lizenzbedingungen dabei sind, dass die Quelle genannt wird, und die aus den OSM-

²⁷ <https://opendatacommons.org/licenses/odbl/summary/>

Daten erzeugte Datenbank ebenfalls per ODbL veröffentlicht wird. Damit sind auch kommerzielle Anwendungsfälle eingeschlossen, solange die Ergebnisse wiederum veröffentlicht werden.

4.4 Auswahl Datenquellen und Betrachtungszeitraum

Tabelle 4-6: Übersicht Datenquellen

Art	Bezeichnung	Histo- rie?	Kom- mer- ziell?	Raum- bezug	Daten- herkunft	Technische Details
Soziale Aufw.	Mirkozensus	ja	nein	Bezirke	Öffentlich per Stichprobe	Webinterface mit CSV-Export
Soziale Aufw.	Außenwan- derung	ja	nein	Bezirke	Meldedaten	Webinterface mit CSV-Export
Soziale Aufw.	Einwohnerre- gister	ja	nein	Pla- nungs- räume	Meldedaten	csv Down- load
Soziale Aufw.	Monitoring Soziale Stadt- entwicklung	ja	nein	Pla- nungs- räume	Melde- und Sozialdaten	Excel Down- load
Soziale Aufw.	Bestand KFZ	ja	nein	Gesamt Berlin	Anmelde- daten KFZ	PDF Down- load
Immobi- lienw. Aufw.	Immobilien- Scout24	ja	ja	IS24 Räume	Angebote auf Platt- form	per API, keine Raum- polygone
Ange- bots- struktur	Google Places API	nein	ja	Koordi- naten	Unterneh- mer / User	per API, 20-60 Treffer pro Aufruf
Ange- bots- struktur	Foursquare	ja (Create date)	ja	Koordi- naten	Unterneh- mer / User	per API, 50 Treffer pro Aufruf
Ange- bots- struktur	Tripadvisor	nein	ja	Koordi- naten	Unterneh- mer / User	per API, keine Suche möglich
Ange- bots- struktur	Yelp	nein	ja	Koordi- naten	Unterneh- mer / User	per API, 1.000 Treffer pro Aufruf
Ange- bots- struktur	OpenStreet- Map	ja (Create date)	nein	Koordi- naten	Mapper- Community	Datenbank Download

Tabelle 4-6 zeigt eine Übersicht über die in diesem Kapitel diskutierten Datenquellen. Die Daten des Mikrozensus, der Außenwanderung und der KFZ Anmeldungen können wegen der zu groben räumlichen Auflösung nicht verwendet werden.

Von den kommerziellen POI-Anbietern bietet nur Yelp mit dem Rückgabewert von 1.000 Treffern bei der POI-Suche überhaupt die Möglichkeit an, alle POIs des Stadtgebiets zu erfassen. Die 50 bzw. 60 Treffer bei Google und Foursquare würden zu sehr vielen kleinräumigen Abfragen führen, wobei jede Abfrage weniger Datensätze als das Maximum zurückgeben muss, da sonst die Gefahr besteht, dass POIs übersehen werden. Dies ist mit einem Maximalwert von 1.000 Treffern einfacher sicherzustellen, als bei 50 bzw. 60 Treffern bei Foursquare bzw. Google. Dieser Vielzahl an API Aufrufen stünden dann jedoch wiederum Lizenzregeln, wie die maximale Anzahl an Anfragen pro Tag, entgegen. Ein Projekt zur Erfassung aller Daten aus diesen Systemen würde also auf der einen Seite einen erheblichen Entwicklungsaufwand bedeuten, und auf der anderen Seite Lizenzkosten verursachen. Da dann auch die Nutzung der Daten oft nicht für eine Machine Learning Verarbeitung freigegeben ist, spricht dies gegen die Daten aller kommerziellen Anbieter. Zudem war es ein Ziel dieser Arbeit, möglichst nur offene Daten zu verwenden. Deshalb sollen sich der Aufbau des Systems und die Datenverarbeitung auf die öffentlichen Datenquellen EWR und MSS, sowie die OSM-Daten beschränken.

Für die Analyse stehen verschiedene Zeitscheiben der verschiedenen Datenquellen zur Verfügung. In Tabelle 4-7 sind die Datenbestände von OSM, EWR und MSS gegeneinandergehalten. Für die verschiedenen Analysen der im nächsten Kapitel vorgestellten Hypothesen werden verschiedene Daten benötigt.

Tabelle 4-7: Auswahl des Betrachtungszeitraums

EWR	OSM	MSS 2013	MSS 2015	MSS 2017	Hypothese
201012		2013			
201112					
201212		2013	2015		H3
201312	201401				H3
201412	201501		2015	2017	H2, H3
201512	201601				H2, H3
201612	201701			2017	H1, H2, H3
	201801				

4.5 Ableitung domänenspezifischer Forschungsfragen und -hypothesen

Die Forschungsfragen aus der Einleitung werden nun auf Basis der Erkenntnisse aus den Phasen *Business Understanding* und *Data Understanding* um fachliche Fragestellungen und Hypothesen zur Gentrifizierung erweitert. Diese Fragestellungen ergänzen die Forschungsfragen „*Welche Features aus den Daten eignen sich zum Aufbau eines Prognosemodells für Gentrifizierung?*“ und „*Welche Algorithmen eignen sich zum Aufbau eines Prognosemodells für Gentrifizierung?*“.

Zunächst soll entsprechend der Studie von Venerandi et al. (2015) der Zusammenhang zwischen dem Ist-Zustand der Angebotsstrukturen und dem Ist-Zustand des sozialen Status untersucht werden.

- F1** Haben lokale Angebotsstrukturen einen Zusammenhang mit dem sozialen Status einer Umgebung?
- H1** Der soziale Status einer Nachbarschaft hängt mit den lokalen Angebotsstrukturen zusammen.

Im Speziellen können Erkenntnisse und Indizien aus anderen Studien im Rahmen der Analyse für Berlin überprüft werden. Das ist zum einen eine Studie von Pearce, Blakely, Witten und Bartie (2007), in der eine signifikante negative Korrelation zwischen dem Abstand vom nächsten Fast Food Geschäft und der Verdrängung für die USA festgestellt wurde. Zum anderen wurde in einer Studie von Papachristos, Smith, Scherer und Fugiero (2011) ein Zusammenhang zwischen der Anzahl neuer Cafés, Kriminalität und Gentrifizierung in Chicago aufgezeigt. In einer weiteren Studie haben Powell, Slater, Chaloupka und Harper (2006) einen Zusammenhang zwischen Sportmöglichkeiten und Haushalteinkommen in den USA feststellen können.

- H1a** Umgebungen mit vielen Cafés haben einen vergleichsweise hohen sozialen Status. (Papachristos et al., 2011)
- H1b** Umgebungen mit vielen Fast Food Geschäften haben einen vergleichsweise niedrigeren sozialen Status. (Pearce et al., 2007)
- H1c** Umgebungen mit vielen Sportmöglichkeiten haben einen vergleichsweise hohen sozialen Status. (Powell et al., 2006)

Anschließend soll der zeitliche Kontext des Gentrifizierungsprozesses untersucht werden. So gibt es nach dem Modell des doppelten Invasions-Sukzessions-Zyklus nach Dangschat (1988) verschiedene Phasen der Gentrifizierung. Zunächst soll die Aussagekraft der Vergangenheits-Angebotsstruktur auf die Veränderung des sozialen Status geprüft werden.

- F2** Gibt es einen Zusammenhang zwischen lokalen Angebotsstrukturen und der Veränderung des sozialen Status?
- H2** Die aktuellen lokalen Angebotsstrukturen haben einen Einfluss auf die zukünftige Veränderung des sozialen Status.

Danach soll ein möglicher Zusammenhang zwischen der Veränderung der Angebotsstruktur mit der Veränderung des sozialen Status untersucht werden.

- F3** Gibt es einen zeitlichen Zusammenhang zwischen der Veränderung der lokalen Angebotsstrukturen und der Veränderung des sozialen Status?
- H3a** Die Veränderung des sozialen Status folgt der Veränderung der lokalen Angebotsstruktur, ist diesem also zeitlich nachgelagert.
- H3b** Die Veränderung des sozialen Status geht der Veränderung der lokalen Angebotsstruktur voraus.
- H3c** Die Veränderung des sozialen Status geschieht zeitgleich mit der Veränderung der lokalen Angebotsstruktur.

5 Data Preparation – Systemaufbau und Datenintegration

In diesem Kapitel soll die zweite Forschungsfrage „*Wie können die Daten integriert werden?*“ beantwortet werden. Damit entspricht das Kapitel dem CRISP-DM Schritt *Data Preparation*.

5.1 Big Data System

Die Grundlage für die Datenintegration in dieser Arbeit wird durch ein modernes Big Data System bereitgestellt. Marz und Warren entwickelten 2013 die Lambda Architektur für Big Data Systeme (Marz & Warren, 2015), welche in Abbildung 5-1 dargestellt ist. Diese Architektur kann auf der einen Seite verwendet werden, um große, statische Datenmengen (data at rest) aus bestehenden Datenbeständen über Batch-Prozesse zu verarbeiten. Auf der anderen Seite können laufend Datenströme (data in motion) von Sensoren, Transaktionen oder Systemlogs analysiert werden.

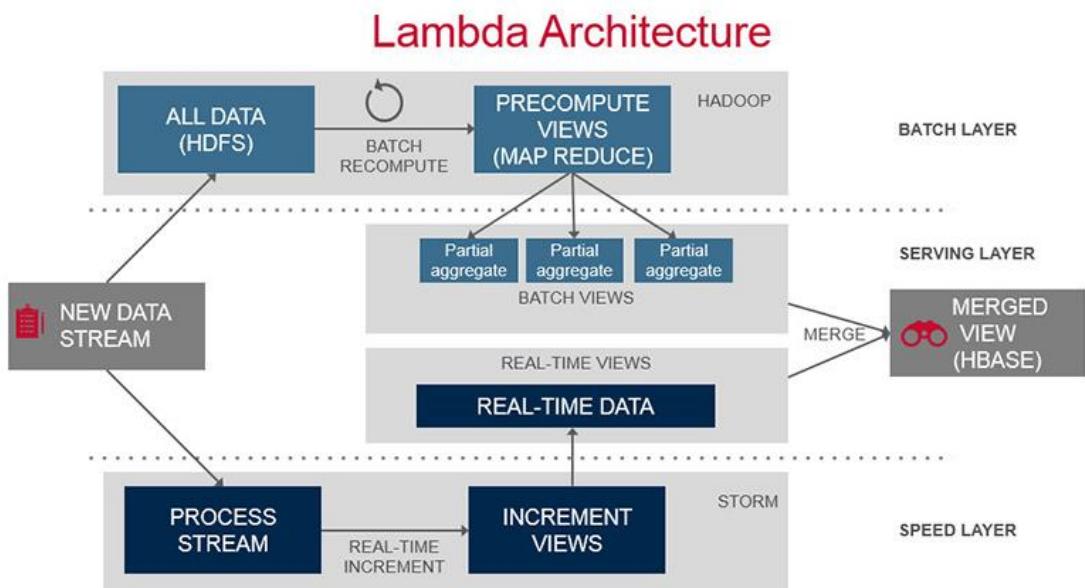


Abbildung 5-1: Lambda Architektur (mapr.com, o.D.)

Neue Daten gehen zum einen in den gesamten Datenpool im Hadoop-basierten *Batch Layer* ein, zum anderen werden sie im *Speed Layer* in Echtzeit verarbeitet. Da die Auswertung der Batchverarbeitung einige Zeit in Anspruch nehmen kann, steht mit der Verknüpfung des vorberechneten Ergebnisses aus dem Batch Layer mit den aktuellen Daten des Speed Layers immer der gesamte Wert im *Serving Layer* für Analysen bereit (Kumar, 2017). Als Beispielanwendung hat Kumar (2017) ein System zur Echtzeitanalyse von Tweets mit einer Lambda Architektur entworfen. Die historischen Tweets sind im Batch

Layer verarbeitet, während die neu eingehenden Tweets im Speed Layer verarbeitet werden und zeitgleich die Datengrundlage für den nächsten Batchlauf erweitern. Technologisch hilft dabei zum Beispiel Apache Kafka, womit Datenströme erfasst und abgespeichert werden können. In der Weiterverarbeitung unterscheidet sich der Batch Layer vom Speed Layer. Während der Batch Layer auf Apache Hadoop Komponenten basiert, nutzt der Speed Layer Komponenten wie Apache Storm für die Echtzeit-Verarbeitung von Datenströmen. Ein Beispiel für den Speed Layer stellt die quelloffene Echtzeit Datenstrom Analyse Plattform *Hortonworks DataFlow* (HDF) dar (Hortonworks, o.D.a).

Die Daten, die im Rahmen dieser Arbeit verwendet werden, sind eher statisch. Daher reicht in diesem Fall das Batch Layer der Lambda Architektur aus. Die quelloffene *Hortonworks Data Platform* (HDP) bietet ein in sich konsistentes Ensemble aus Hadoop-Komponenten für eine solche Batch-Verarbeitung von data at rest (Hortonworks, o.D.b). Für diese Arbeit wurde die HDP-Sandbox in der Version 2.6.3 verwendet. In dem Paket enthalten sind unter anderem die Apache-Komponenten Hadoop, MapReduce, YARN, Tez, Hive, Ambari und ZooKeeper. In Abbildung 5-2 ist das Zusammenspiel dieser Komponenten skizziert.

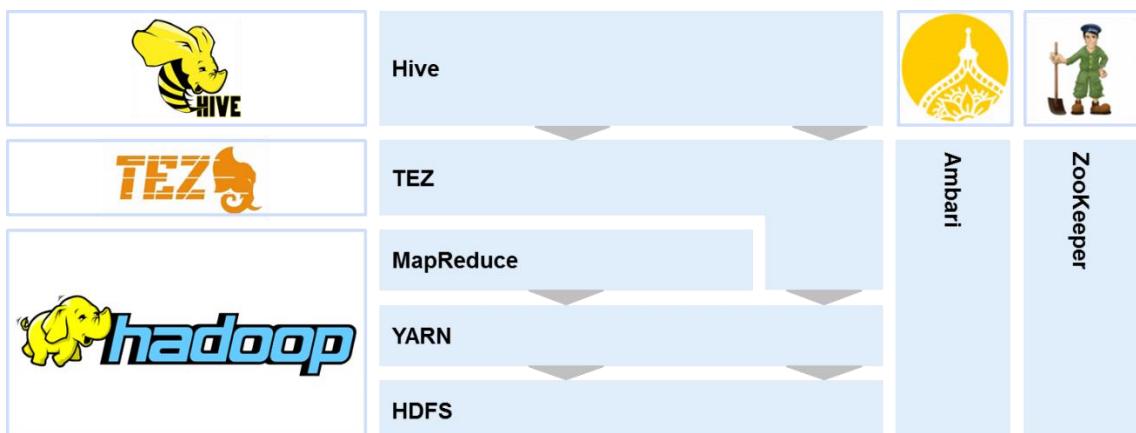


Abbildung 5-2: Übersicht Hadoop System Komponenten (In Anlehnung an Hortonworks, o.D.b)

Apache Hadoop besteht aus den Komponenten HDFS und MapReduce (vgl. Kapitel 2.1). Seit der Apache Hadoop Version 2.0 ist YARN (Yet Another Resource Negotiator) für das Ressourcenmanagement und die Jobverwaltung hinzugekommen. So stellt YARN sicher, dass trotz laufender Batchverarbeitung noch andere Arbeiten auf den Daten ausgeführt werden können. In der nächsten Ebene wird YARN von MapReduce und TEZ genutzt. TEZ ist eine Weiterentwicklung von MapReduce, die auf eine Persistierung der

Zwischenergebnisse verzichtet, und somit schnellere Berechnungen ermöglicht (Hortonworks, 2013). In Abbildung 5-3 ist dargestellt, wie MapReduce nach jedem Reduce-Schritt eine Persistierung vornimmt, was im Vergleich zu TEZ mehr Zeit kostet.

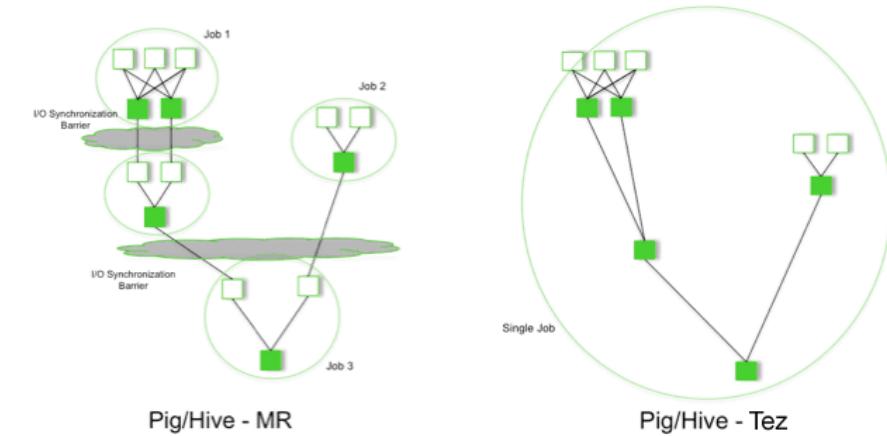


Abbildung 5-3: MapReduce vs. TEZ (Hortonworks, 2013)

Hive bietet die Möglichkeit mit einer SQL-nahen Sprache die Daten im HDFS auszulesen. Hive kann theoretisch auf MapReduce oder auf TEZ aufbauen, im HDP-System ist es mit TEZ verbunden. Ambari und ZooKeeper übernehmen Querschnittsaufgaben in der Infrastruktur. Ambari dient als Managing und Monitoring Oberfläche für Apache Hadoop Cluster²⁸, während ZooKeeper einen zentralen Service für die Konfiguration des Hadoop Systems zur Verfügung stellt.

Ein Hadoop Cluster besteht oft aus einem MasterNode und mehreren SlaveNodes. Der MasterNode enthält den NameNode und den JobTracker. Wie in der Architektur in Abbildung 5-4 ersichtlich, können auch weitere MasterNodes mit SecondaryNameNodes existieren. Diese kommunizieren mit den SlaveNodes, welche DataNodes und TaskTra-

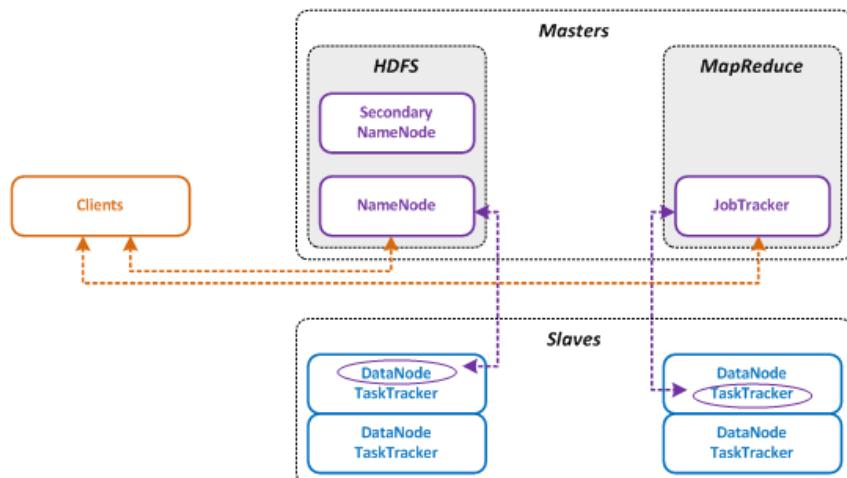


Abbildung 5-4: Hadoop Architektur (Doe, 2018)

cker enthalten. Im DataNode werden die Daten verteilt gehalten und im TaskTracker prozessiert. Der MasterNode kann ebenfalls DataNode und TaskTracker enthalten.

²⁸ Siehe Abbildung 9-7: Screenshot Ambari

Hive kann neben dem Abfragen von Daten in Hadoop auch Tabellen im eigenen sogenannten Warehouse halten. Es wird zwischen internen und externen Tabellen unterschieden. Externe Tabellen referenzieren auf die Ursprungsdatei, was bei sich ändernden Dateien wie zum Beispiel den Tweets der letzten zwei Wochen dafür sorgt, dass die gesamte Verarbeitungsstrecke auf die geänderten Daten direkt reagiert. Werden dagegen interne Tabellen als Kopie im Hive-Warehouse angelegt, müssen alle Persistierungen in der Strecke erneut erfolgen, damit die Auswertung auf den neuen Daten basiert. Zwischenpersistierungen haben jedoch den Vorteil, dass zur Entwicklungszeit bei Änderungen in höheren Auswertungsebenen die Laufzeit teils deutlich besser ist, da nicht alle Berechnungsschritte erneut ausgeführt werden müssen. In dieser Arbeit werden sich die Ursprungsdateien nicht ändern, weshalb aus Performance-Gründen interne Tabellen verwendet werden.

Hive hat neben der Unterstützung der SQL-Syntax eine Vielzahl an eingebauten Funktionen. Zudem gibt es die Möglichkeit, eigene Funktionen zu entwickeln und sie innerhalb der Hive Aufrufe zu nutzen. Die Funktionen nennen sich *User-Defined Functions* (UDF) und können in Java entwickelt werden.

Für diese Arbeit wurde anfangs auf einer Virtuellen Maschine eine HDP-Sandbox installiert. In der Bearbeitung wurde jedoch deutlich, dass die Performance für die Datenintegration nicht ausreichend war. Deshalb wurde die Weiterentwicklung frühzeitig auf einem bestehenden Hadoop-Cluster fortgesetzt, welches aus fünf DataNodes bestand.

5.2 Datenintegration

5.2.1 POI-Daten aus OpenStreetMap erzeugen

Zunächst wurden die OSM-Daten von Berlin von <http://download.geofabrik.de/europe/germany/berlin.html> jeweils zum 01.01. von den Jahren 2014 bis 2018 heruntergeladen. Da die Daten im PBF-Format stark komprimiert vorliegen, können nur spezielle Programme diese Daten auslesen. Das PBF-Format ist ca. 70% kompakter als das XML-basierte OSM-Format und doppelt so komprimiert wie vergleichbare ZIP-Dateien (*DE:PBF Format – OpenStreetMap Wiki*, o.D.). Ein Programm zum Auslesen von PBF-Daten ist das quellöffentige Java Programm *Osmosis*, welches per Kommandozeile bedient wird. Damit ist es möglich, die Daten im PBF-Format zu lesen und direkt in andere Formate zu konvertieren, zum Beispiel in das OSM-XML-Format (*Osmosis – OpenStreetMap Wiki*, o.D.). Beim Lesen der Daten können Filter -und Sortierparameter übergeben werden. So ist das Filtern von Daten innerhalb von vorgegebenen Polygonen möglich, womit zum Beispiel die Daten für den Raum Berlin aus dem Planet.osm extrahiert werden

können. Zudem kann auf die Datentypen Node oder Way, sowie auf Key-Value Tags gefiltert werden. Mit dem Filtern auf die Nodes und den zehn Keys, die in Kapitel 4.2.3.2 vorgestellt wurden, wurde dann eine Datei erzeugt. In Abbildung 5-5 ist der Aufruf für Osmosis dargestellt. Im Ergebnis entsteht pro Jahresscheibe eine unkomprimierte OSM-Datei mit XML-Einträgen wie in Abbildung 4-5.

```
osmosis
--read-pbf "C:\dev\master\files Berlin\berlin-140101.osm.pbf"
--node-key keyList="tourism,amenity,heritage,sport,leisure,of-
fice,atm,landuse,public_transport,shop"
--write-xml "C:\dev\master\files Berlin\result\filtered_140101.osm"
```

Abbildung 5-5: Osmosis-Aufruf zum Extrahieren der POI-Nodes

Nun sollten die Daten im OSM-XML-Format für Auswertungen in Hive verfügbar gemacht werden. Zuerst wurden die Dateien in Hadoop über den Filebrowser in Ambari hochgeladen. Das Code-Beispiel in Abbildung 5-6 zeigt, wie zunächst eine Tabelle mit den Daten der OSM-Datei erzeugt wurde. Danach wurde mit einer UDF von OSM2Hive eine Node-Tabelle erzeugt (Pavie, 2015). Dieses Vorgehen wurde für jede Jahresscheibe wiederholt. Das Ergebnis ist beispielhaft in Tabelle 9-6 im Anhang dargestellt.

```
/* 1. Create Table based on osm file in HDFS */
CREATE TABLE osmdata(osm content STRING) STORED AS TEXTFILE;
LOAD DATA INPATH 'hdfs://itfin105.it.zeb.de:8280/user/admin/dhel-
weg/filtered_140101.osm' OVERWRITE INTO TABLE osmdata;

/* 2. Import OSM2Hive and create node-table */
add jar hdfs://itfin105.it.zeb.de:8280/user/admin/dhel-
weg/OSM2Hive.jar;
CREATE TEMPORARY FUNCTION OSMImportNodes AS 'in-fo.pavie.osm2hive.con-
troller.HiveNodeImporter';
CREATE TABLE osmnodes_filtered_140101 AS
SELECT OSMImportNodes(osm_content) FROM osmdata;
```

Abbildung 5-6: Hive-Code für die Erzeugung einer Node-Tabelle auf Basis einer OSM-Datei

Im nächsten Schritt wurden aus der Key-Value Liste im Feld Tag eigene Datenfelder gemacht. In Hive können Datenfelder den Typ *map* haben und sind somit einfach auswertbar. Zeitgleich wurde per selbst entwickelter UDF der Planungsraum auf Basis der Koordinaten bestimmt. Der Aufruf ist abgekürzt in Abbildung 5-7 dargestellt. Der gesamte Quelltext ist in Abbildung 9-8 im Anhang zu finden.

```

add jar hdfs://itfin105.it.zeb.de:8280/user/admin/dhelweg/lormap-
per_runlc.jar;
CREATE TEMPORARY FUNCTION planungsraum AS 'de.zeb.hive.udf.lormap-
per.Planungsraum';

select
[...]
planungsraum(concat(latitude, " , ",longitude)) as planungsraum,
tags["name"] as name,
[...]

```

Abbildung 5-7: Hive-Code für das Auslesen der Tags und Bestimmung des Planungsraumes per UDF

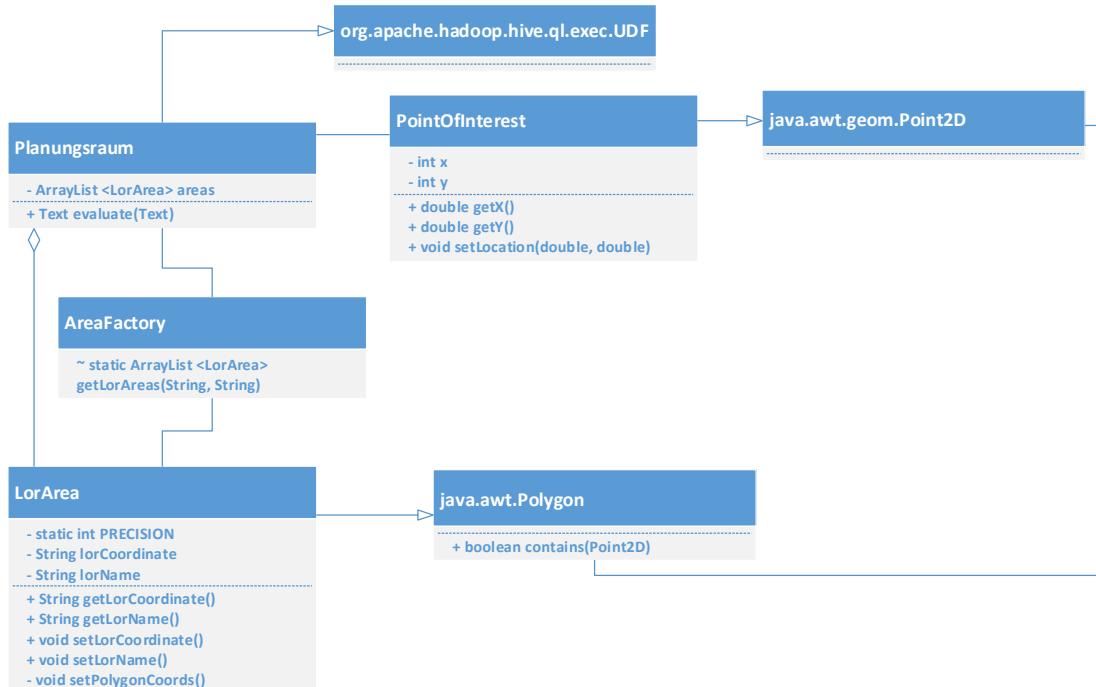


Abbildung 5-8: Klassendiagramm UDF lormapper

Die UDF *lormapper* besteht im Kern aus vier Klassen, welche im Klassendiagramm in Abbildung 5-8 dargestellt sind. Die Klasse Planungsraum erbt von der Klasse UDF aus dem Hive-Paket. Auf diese Weise kann sie innerhalb von Hive als UDF genutzt werden. In der Klasse wird die ArrayList „areas“ ähnlich wie beim Singleton-Pattern nur beim ersten Aufruf initialisiert.²⁹ Für das Initialisieren der Liste an Planungsräumen wird die Klasse AreaFactory verwendet, welche die Planungsraum-Daten vom Amt für Statistik Berlin-Brandenburg aus der KML-Datei ausliest und LorArea-Objekte erzeugt. Beim

²⁹ Die Initialisierung findet nur einmalig statt, um die Laufzeit zu verbessern. Bei ersten Tests ohne die Singleton-ähnliche Initialisierung der ArrayList „areas“ kam es zu sehr hohen Laufzeiten, da bei jeder Zeile in Hive erneut über die UDF lesend auf die 3MB große KML-Datei zugegriffen werden musste, um danach die 443 Polygone der Planungsräume zu erzeugen.

Aufruf der UDF wird als Text eine Geoposition übergeben (lat, lon). Mit dieser wird ein PointOfInterest-Objekt erzeugt. Die Klassen PointOfInterest und LorArea erben jeweils von Objekten aus Java-AWT. Java-AWT ist eigentlich ein Toolkit für grafische Oberflächen, enthält aber auch Logik für zweidimensionale Koordinatensysteme. Dank der Vererbung ist es möglich zu prüfen, ob ein PointOfInterest in einem LorArea liegt. Durch das Iterieren durch alle LorAreas in der ArrayList wird so zu den POIs in Berlin der passende Planungsraum gefunden. Dieses Vorgehen ließe sich auch mit wenig Aufwand für beliebige andere KML-Dateien anwenden. Die detaillierten Java-Sourcen befinden sich im Anhang in Kapitel 9.4.2.

Im Ergebnis wurde das Tag-Feld in mehrere Felder aufgeteilt, je Key ein Feld. Neben der Position als Koordinate enthält jetzt jeder Datensatz auch eine Planungsraum-ID. Im Anhang ist das Ergebnis unter Tabelle 9-7 zu finden.

Im letzten Schritt der POI-Erzeugung aus den OSM-Nodes soll das Mapping auf einen der POI-Typen erfolgen. Hierzu wurde mit Hilfe des Mappings³⁰ die Query aus Abbildung 9-9 generiert. Daraus ergibt sich ein Aufbau wie in Tabelle 5-1 dargestellt, in dem das Feld poi_type hinzugefügt wurde.

Tabelle 5-1: Tabellenaufbau osm_poi_type_yyyy

timeslice	201501
coords	52.5073388, 13.3207848
planungsraum	4030828
node_id	N26735763
last_modification_userid	1260280
last_modification_time	201409
last_modification_version	18
last_modification_changesetid	25170126
name	Sakana
description	null
addr_city	Berlin
addr_country	DE
addr_housenumber	106
addr_postcode	10625
addr_street	Pestalozzistra?e
addr_suburb	Charlottenburg
poi_type	Restaurant Sushi

³⁰ Das technische Mapping der POI-Typen zu den OSM Tags kann unter https://github.com/dhelweg/masterthesis2018_gentrification/blob/master/data/OSM_POI_MAPPING.xlsx eingesehen werden

5.2.2 OSM-POI-Daten in Raumdaten aggregieren

Durch die bisherige Aufbereitung der OSM-Daten ist ein Format geschaffen worden, dass dem der Schnittstellen von Foursquare, Yelp und anderen ähnelt. Zusätzlich sind die Daten jedoch als Zeitscheiben vorhanden und bereits in einen LOR-Raumbezug gebracht worden. Dieser Raumbezug wäre mit den anderen POI-Datenquellen auf die gleiche Weise zu berechnen. Ein Zeitbezug fehlt jedoch bei den Datenquellen – mit Ausnahme von Foursquare, wo das Erfassungsdatum mit angegeben wird.

5.2.2.1 POI-Statuserkennung im Zeitverlauf: Neu, Geschlossen, Bestand

Auf Basis der Zeitscheiben können neu erfasste, gelöschte und geänderte POIs analysiert werden. Auf diese Weise ließen sich die Änderungen der lokalen Angebotsstruktur (Kapitel 3.1) berechnen.

Tabelle 5-2: Beispiele POI in den Zeitscheiben

2014_name	2015_name	2016_name	2017_name	2018_name
China Box	China Box	null	null	null
null	Shiso Burger	Shiso Burger	Shiso Burger	Shiso Burger
Sakana	Sakana	Sakana	Sakana	Lemongrass
Berliner Spar-kasse	Berliner Spar-kasse	Berliner Spar-kasse	Berliner Spar-kasse	Sparkasse

Die Änderung der lokalen Angebotsstruktur soll auf Basis neu eröffneter und geschlossener Geschäfte erfasst werden. In Tabelle 5-2 sind vier Beispiele von verschiedenen POIs in Berlin dargestellt. Im ersten Fall „China Box“ ist das POI bereits im Jahr 2014 enthalten. Damit wird es als bereits vorhanden angesehen und soll in 2014 nur in der Bestandskennzahl enthalten sein.³¹ Dieses POI ist nur in den Daten bis 2015 enthalten. In den Daten von 2016 soll es also als geschlossen erfasst werden. 2017 soll es nicht mehr enthalten sein. Der zweite Fall „Shiso Burger“ ist 2015 hinzugekommen und soll in dem Jahr als neu erfasst werden. Danach soll er im Bestand angezeigt werden. Im dritten Fall „Sakana“ gibt es einen Wechsel des Feldes Name. Dies wird in dieser Arbeit so interpretiert, dass das vorherige Geschäft geschlossen wurde und ein Neues eröffnet wurde. Das

³¹ Eine Ausnahme stellen POIs dar, bei denen das Feld last_modification_version =1 und last_modification_time >= 201301 ist. Diese werden für 2014 als neu erfasst. Die Erkennung, ob ein POI zuvor nicht vorhanden war oder nicht mehr vorhanden ist, erfolgt nicht über den Namen, sondern über das technische Feld last_modification_version. Ist dies für ein Jahr null, so existiert das Node nicht.

POI „Sakana“ wird also in 2018 als geschlossen und das POI „Lemongrass“ als neu erfasst. Anders verhält es sich bei dem Namenswechsel des POIs „Berliner Sparkasse“. Hier ist die Änderung auf „Sparkasse“ erkennbar kein Wechsel des Geschäftes.

Für diese Änderungen musste ein Algorithmus entwickelt werden, der signifikante Änderungen des Namens von Schönheitskorrekturen durch die Mapper-Community unterscheidet.³² Hierzu wurde die in Hive vorhandene Funktion zur Levenshtein-Distanz genutzt, welche den Unterschied zwischen zwei Zeichenketten misst (Levenshtein, 1966).

Für die Erkennung von signifikanten Änderungen wird die Levenshtein-Distanz um die Differenz der Länge der beiden Zeichenketten verringert. So wird das Hinzufügen oder Entfernen von Teilen des Namens abgefangen, wie zum Beispiel bei „Berliner Sparkasse“ und „Sparkasse“. Die Levenshtein-Distanz ist für dieses Beispiel genau die Differenz der Länge der Zeichenketten. Im zweiten Schritt wird geprüft, ob dieser Wert größer als ein Zehntel der Länge der kleineren Zeichenkette ist. Wenn ja, so hat sich der Name signifikant geändert. Dies ist notwendig, um Änderungen, wie einen Bindestrich der hinzugefügt oder weggenommen wurde, nicht als signifikante Änderung zu werten. Gleichzeitig muss jedoch bei einer deutlichen Verkürzung oder Verlängerung des Namens wie von „DM“ auf „Rossmann“ dennoch eine signifikante Änderung gewertet werden. Groß- und Kleinschreibung werden nicht beachtet. Abbildung 5-9 zeigt eine vereinfachte Version des Algorithmus.

```
[...]
case
when
    --levenshtein
    levenshtein(UPPER(j2015.name),UPPER(j2014.name))
    --delta_length
    -(      greatest(length(j2015.name),length(j2014.name))
          - least(length(j2015.name),length(j2014.name)))
    --min_length
    >= 0.1* least(length(j2015.name),length(j2014.name))
then 1
else 0
end as changed_2014_to_2015
[...]
```

Abbildung 5-9: Hive-Code zu osm_poi_changed (vereinfachter Auszug)

In Tabelle 5-3 sind Beispiele zur Anwendung des Algorithmus dargestellt. Diese Methode hat sich bei der Analyse der Daten als relativ treffsicher gezeigt. Es wurden jedoch nicht

³² Alternativ dazu kann auch der Diff Algorithmus nach Myers (1986) verwendet werden. Dazu müsste ein weiteres UDF entwickelt werden. Ein Python Beispiel dazu gibt es unter <http://blog.robertelder.org/diff-algorithm/> Elder (2017)

alle Fälle einzeln untersucht. Theoretisch stellen auch Änderungen wie „Berliner Sparkasse“ zu „Sparkasse Berlin“ eine nicht signifikante Änderung dar. Mit der aktuellen Logik kann dies jedoch nicht erfasst werden, und würde als signifikant geändert gelten.

Tabelle 5-3: Beispiele Namensänderungen

name (pre)	name (new)	leven-shtuin	delta_length	min_length / 10	has_changed
Berliner Sparkasse	Sparkasse	9	9	0,9	0
Sakana	Lemongrass	8	4	0,6	1
Shiso Burger	Shiso-Burger	1	0	1,2	0
DM	Rossmann	7	6	0,2	1
ALDI	LIDL	3	0	0,4	1
Berliner Sparkasse	Sparkasse Berlin	16	2	1,6	1

Der Hive-Code des oben erläuterten Vorgehens ist im Anhang unter Abbildung 9-14 zu finden. Das Ergebnis ist in Tabelle 9-8 dargestellt. Dort ist der Zeitbezug für die Analyse aufgehoben worden und muss nun erneut hergestellt werden. Hier galt es zu beachten, auch die richtigen Felder zum jeweiligen Jahr zu nutzen. Zusätzlich sollte der aktuellste Datenbestand für das jeweilige POI genutzt werden, zum Beispiel um korrigierte Tags für die Typisierung zu nutzen. Der Code hierzu ist in Abbildung 5-10 dargestellt.

```
[...]
--für 2014
case
    when o.changed_2014_to_2015 = 0
        and o.changed_2015_to_2016 = 0
        and o.changed_2016_to_2017 = 0
        and o.changed_2017_to_2018 = 0
        and o.2018_poi_type is not null
    then o.2018_poi_type
    [...]
    when o.changed_2014_to_2015 = 0
        and o.2015_poi_type is not null
    then o.2015_poi_type
    else o.2014_poi_type
end as poi_type,
[...]
```

Abbildung 5-10: Hive-Code zu osm_poi_state_basis (vereinfachter Auszug)

Im gleichen Schritt wird ebenfalls für signifikant geänderte POIs Datensätze jeweils ein zusätzlicher Datensatz generiert, der das geschlossene POI darstellt. Der Hive-Code für die Bestimmung des POI-Status befindet sich im Anhang, in Abbildung 9-15 bis Abbildung 9-18. Das Ergebnis ist in Tabelle 5-4 zu sehen.

Tabelle 5-4: Tabellenaufbau osm_poi_state

node_id	timeslice	planungs- raum	poi_type	name	poi_state
N1000710165	201501	4030931	fast_food Sonstiges	China Box	steady
N1000710165	201601	4030931	fast_food Sonstiges	China Box	deleted
N1000710165	201701	4030931	null	null	null
N1005311925	201701	3030406	Bankfiliale	Spar-kasse	steady
N1005311925	201801	3030406	Bankfiliale	Spar-kasse	steady
N1552074537	201401	1011302	null	null	null
N1552074537	201501	1011302	Restaurant Steakhouse	Shiso Burger	new
N1552074537	201601	1011302	Restaurant Steakhouse	Shiso Burger	steady
N26735763	201701	4030828	Restaurant Sushi	Sakana	steady
N26735763	201801	4030828	Restaurant Asiatisch	Lemon-grass	changed
N26735763	201801	4030828	Restaurant Sushi	Sakana	changed _deleted

5.2.2.2 OSM-Featureentwicklung

Basierend auf dem POI-Typen und dem POI-Status können nun Datenfeatures für die Planungsräume erzeugt werden, indem die Daten je Raumeinheit aggregiert werden. Dazu werden für jedes Element einer jeden Ebene der POI- Hierarchie (Typ, Kategorie und Domäne) fünf Kennzahlen gebildet.³³ Diese Kennzahlen sind:

- *_new* Die Anzahl der neuen POIs.
(new + changed)
- *_deleted* Die Anzahl der geschlossenen POIs.
(deleted + changed_deleted)
- *_steady* Die Anzahl der bereits im Vorjahr vorhandenen POIs, die weder gelöscht noch geändert wurden.
(steady)
- *_ytd* Die Anzahl der neuen abzgl. der Anzahl der geschlossenen POIs.
(*_new - _deleted*)

³³ Code und Ergebnistabelle im Anhang unter Abbildung 9-19: Hive-Code zu osm_poi_features_type und Tabelle 9-9: Tabellenaufbau osm_poi_features_type

- $_stock$ Die Anzahl der in diesem Jahr vorhandenen POIs.
 $(_steady + _new)$

Durch die Aggregation von POI-Typ und POI-Status würde eine Tabelle entstehen, bei der der Kennzahlenname in einer Spalte steht, und der Wert als Anzahl in einer anderen Spalte. Damit die Machine Learning Algorithmen die Daten verarbeiten können, müssen die Datenfeatures jedoch als einzelne Spalte vorhanden sein. Für diese Pivotierung einer Feldausprägung in eine Kennzahl gibt es in Hive bisher keine Standardfunktion. Deshalb musste hier auf ein UDF zur Aggregation (auch UDAF abgekürzt) zurückgegriffen werden. Mit dem CollectUDAF können Werte aus zwei Feldern als Key-Value Map aggregiert werden (klout, 2016). Da in POI-Status und POI-Typ aggregiert werden, wird dieser Schritt zweimal ausgeführt. Abbildung 5-11 zeigt dies beispielhaft für den $_stock$ -Wert der POI-Domäne Gastronomie. Die ursprüngliche Kennzahl „anz“ wird dabei jedoch nicht summiert, sondern zusammen als Value mit dem POI-Status als Key in der neu erzeugten Map gespeichert. Für die innere Aggregation des POI-Status hat das Aggregierte Feld dann die Form $\{"ytd":6, "stock":130, "deleted":24, "new":30, "steady":100\}$. Da die Kennzahl nicht per Summierung aggregiert wird, musste für jede Hierarchie-Stufe (domain, category, type) eine eigene Teilstrecke erstellt werden.

```
add jar hdfs://itfin105.it.zeb.de:8280/user/admin/dhelweg/brickhouse-
0.7.1-SNAPSHOT.jar;

CREATE TEMPORARY FUNCTION collect AS 'brickhouse.udf.collect.CollectUDAF';

DROP table osm_poi_features_domain_piv;
CREATE table osm_poi_features_domain_piv as
select
    planungsraum,
    timeslice,
    [...]
    group_map_stock ['Gastronomie'] as d_gastronomie_stock
    [...]
from (
    select
        planungsraum,
        timeslice,
        [...]
        collect(domain, stock) as group_map_stock
        from (
            select
                planungsraum,
                timeslice,
                domain,
                [...]
                group_map['stock'] as stock
            from (
                select planungsraum,
                    timeslice,
                    domain,
```

```

    collect(poi_state, anz) as group_map
  from osm_poi_features_domain
  group by planungsraum,
    timeslice,
    domain
  )
  m
  )
  a
group by planungsraum,
  timeslice
)
d
;

```

Abbildung 5-11: Hive-Code zu osm_poi_features_domain_piv (vereinfachter Auszug)

Dies wird analog für alle Kennzahlen und für alle Ausprägungen der POI-Domänen (Oberkategorie), -Kategorien und -Typen (Unterkategorie) der POI-Mapping-Hierarchie³⁴ vorgenommen. Die Kennzahlen haben dann jeweils die Form *(d/c/t)_ELEMENT_(new/deleted/ytd/steady/stock)*, der erste Buchstabe gibt die Stufe in der POI-Mapping-Hierarchie an. Tabelle 5-5 zeigt dies in einem Auszug für die Domänen-Teilstrecke. Damit ist die Aggregation auf Ebene des Planungsraumes abgeschlossen.

Tabelle 5-5: Tabellenaufbau osm_poi_features_domain_piv (vereinfachter Auszug)

planungsraum	timesslice	d_gastro-no-mie_new	d_gastro-no-mie_deleted	d_gastro-no-mie_ytd	d_gastro-no-mie_steady	d_gastro-no-mie_stoc k
1011302	201401	4	null	4	120	124
1011302	201501	30	24	6	100	130
1011302	201601	20	21	-1	109	129
1011302	201701	24	18	6	111	135
1011302	201801	22	9	13	126	148

5.2.3 Öffentliche Datenquellen (EWR und MSS)

Aus den Daten des Einwohnerregisters werden die fünf- und zehnjährige Wohndauer, die Herkunftsländer der Migranten, sowie jeweils für alle Einwohner, für Ausländer und für Migranten die Felder Anzahl, Anzahl_Männlich, Anzahl_Weiblich und die Altersklassen 0-18, 18-27, 27-45, 18-35, 35-45, 45-55, 55-65 und 65-110 gebildet. Zusätzlich wird jeweils ein ungefähres Durchschnittsalter berechnet.³⁵ Damit ist das Alter eine Durchschnittsgröße, während die anderen Werte die Anzahl der Personen zählt, die in die Kategorie der Kennzahl fallen. Die MSS-Daten sind bereits auf Planungsraum-Ebene vorhanden und können einfach der Auswertungsebene hinzugefügt werden.

³⁴ Vgl. Tabelle 9-5: POI-Hierarchie

³⁵ Durchschnittsalter = Anzahl der Menschen in einer Altergruppe * Klassenmitte der Altersgruppe. Also zum Beispiel 19.5 * E_E18_21 bei der Altersgruppe 18-21

Neben den MSS-Werten, die dem Sozialindex von Holm und Schulz (2016) nahekommen, soll auch ein Index nach Vorbild von Döring und Ulbricht (2016) erstellt werden.³⁶ Nach diesem Vorgehen werden drei Indizes aufgebaut: *Mobilität*, *Veränderung der Bevölkerungsstruktur*, sowie *Wohnungswirtschaft*. Da für Wohnungswirtschaft die Datengrundlage fehlt, werden nur die anderen beiden Indizes aufgebaut. Dazu wird je Kennzahl die in einem Index verwendet wird der Abstand der Kennzahl vom durchschnittlichen Wert der Kennzahl gemessen. Anschließend wird diese Differenz durch die Standardabweichung geteilt. Im Ergebnis werden die Kennzahlen je nachdem ob sie inhaltlich verstärkend für oder gegen Gentrifizierung sprechen aggregiert. Für die Kennzahl K11 aus den Kontextdaten des MSS ist diese Berechnung in Abbildung 5-12 beispielhaft dargestellt. Der Wert k11_msr ist durch diese Formel standardisiert worden. Das Vorgehen ist der Berechnung des MSS nachempfunden.

```
drop view lor_own_idx_k11;
create view lor_own_idx_k11 as
select
    s.id as raum_id,
    s.name as raum_desc,
    concat(s.jahr-1,'12') as zeit,
    s.k11,
    a.k11_stddev,
    a.k11_avg,
    (s.k11-a.k11_avg) / a.k11_stddev as k11_msr
from zeitreihe_mss_k11 s
join
(
    select
        jahr
        ,stddev_pop(k11) as k11_stddev
        ,avg(k11) as k11_avg
    from zeitreihe_mss_k11
    where k11 > 0
    group by jahr
) a on a.jahr = s.jahr
where s.k11 > 0;
```

Abbildung 5-12: Hive-Code zu lor_own_idx_k11

Für den Index nach Döring und Ulbricht sollen für den Mobilitätsindex die Kennzahlen *Veränderung des Anteils der Bevölkerung >5 Jahre Wohndauer*, *Veränderung des Anteils der Bevölkerung >10 Jahre Wohndauer* und *Wanderungsvolumen* zwischen dem Zeitraum von Ende 2014 bis Ende 2016 einbezogen werden. Wobei jedoch die Erhöhung der Einwohner mit hoher Wohndauer den Index negativ beeinflusst, da dies gegen eine Gentrifizierung sprechen würde. Das Wanderungsvolumen stammt aus den Daten der Kennzahl K11 aus den MSS-Kontextindikatoren, die Wohndauer stammt aus den EWR-Daten.

³⁶ Vgl. Kapitel 3.4

Für den Index zur Veränderung der Bevölkerungsstruktur fließen die Kennzahlen *Veränderung des Anteils der Langzeitarbeitslosen*, *Veränderung des Anteils der Ausländer* und die *Veränderung des Anteils der Menschen mit Migrationshintergrund* negativ ein, höhere Arbeitslosenzahlen deuten also an, dass keine Gentrifizierung vorliegt. Sinkt der jeweilige Anteil, spricht dies für eine Gentrifizierung. Zusätzlich geht die *Veränderung des Anteils der Altersgruppen 18-35*, sowie *35-45* positiv in den Index ein. Die Arbeitslosenzahlen entstammen den MSS-Daten (Kennzahl D2), die anderen Daten kommen aus den EWR-Veröffentlichungen.

Ein Ausschnitt des Ergebnisses der Index-Bildung nach Döring und Ulbricht (2016) ist in Tabelle 5-6 dargestellt.³⁷ In der Tabelle sind die beiden nach dem Index am stärksten und am wenigsten gentrifizierten Planungsräumen enthalten. Auf diese wird im Folgenden kurz eingegangen. Der Planungsraum „Werkstraße“ liegt in Spandau am äußersten Rand von Berlin. Aus dem Gebiet sind verhältnismäßig viele Bewohner mit einer Wohndauer von über 10 Jahren hinzugekommen. Das Gebiet „Eldenaer Straße“ liegt in Pankow und gehört zum Prenzlauer Berg. Es besteht im Unterschied zur Nachbarschaft aus Neubauten (Holm, 2014, S. 280). In den zwei Jahren der Betrachtung haben verhältnismäßig viele Bewohner die Schwelle zu 5 Jahren Wohndauer überschritten. Die Änderung des Anteils der Bewohner mit hoher Wohndauer ist jedoch ein Indikator gegen eine aktuelle Gentrifizierung. Der Planungsraum „Adlershof West“ hat ein sehr hohes Wanderungsvolumen, was durch einen Anstieg der Einwohnerzahlen von 508 auf über 1.259 innerhalb der zwei Betrachtungsjahren zurückzuführen ist. Zur gleichen Zeit sind viele langjährige Einwohner fortgezogen. Über 600 der neuen Einwohner stammen zudem aus den Altersgruppen 18-35 und 35-45. Ebenfalls ist der Anteil der Einwohner die Transferleistungen empfangen stark gesunken. Damit ergeben sich hohe Werte für den Mobilitäts- und Bevölkerungsstruktur-Index. Für das Gebiet „Gleisdreieck“ gilt ähnliches, wobei hier kein Wegzug, sondern nur ein massiver Zuzug in das Entwicklungsgebiet im Betrachtungszeitraum stattfand.

Tabelle 5-6: Tabellenaufbau lor_own_idx_plr_tb (Auszug)

raum_id	raum_desc	sum_idx_points	idx_m_points	idx_b_points
9020701	Adlershof West	75	39	36
2020201	Gleisdreieck/Entwicklungsgebiet	60	41	19
...
3061441	Eldenaer Straße	-12	-8	-4
5020420	Werkstraße	-13	-6	-7

³⁷ Siehe auch Abbildung 9-21: Hive-Code der Strecke zu lor_own_idx_plr_tb

5.2.4 Erzeugung der Ergebnistabelle für die Modellbildung

Damit die Daten der verschiedenen Planungsräume vergleichbar werden, wird im letzten Schritt jeder Wert über die Anzahl der Einwohner normalisiert. Für die POI-Daten wird die Logik von Kennzahl_POI aus Abbildung 5-13³⁸ angewendet, bei den EWR-Daten wird die Berechnungslogik von Kennzahl_EWR³⁹ genutzt. Die Index-Daten werden nicht über die Einwohner normalisiert, da sie bereits mit prozentualen Verhältnissen errechnet wurden.

$$\text{Kennzahl_POI} = \frac{\text{Kennzahl_POI} \cdot 1000}{\text{Anzahl_Einwohner}} \quad \text{Kennzahl_EWR} = \frac{\text{Datenfeld}}{\text{Anzahl_Einwohner}}$$

Abbildung 5-13: Formeln zur Einwohnergewichtung der Exportkennzahlen

Diejenigen Planungsräume, welche als Ausreißer nicht in den MSS-Daten berechnet wurden, werden auch aus der Ergebnistabelle für die Modellbildung ausgeschlossen. Die Planungsräume mit weniger als 300 Einwohnern wurden ausgeschlossen, da sie das Ergebnis verzerren.⁴⁰ Die Planungsräume *Gewerbegebiet Bitterfelder Straße* und *Messegelände* wurden als weitere Ausreißer ausgeschlossen.

In den originalen Klassengrenzen der MSS-Daten fallen die meisten Planungsräume in eine Klasse.⁴¹ Dadurch würden jedoch die Klassifizierungsalgorithmen dazu verleitet, alles in diese Klasse zu klassifizieren, da dann z.B. bei der Dynamik über 75% richtig klassifiziert sein würden. Ein ähnliches Problem hatten auch Venerandi et al. (2015, S. 260), weshalb sie die Klassifizierung auf zwei Klassen begrenzt haben. Für diese Arbeit wurden zwei neue Grenzen generiert, einmal eine Einteilung in zwei, und einmal eine Einteilung in drei Klassen. Die neuen Klassengrenzen werden auf den metrischen Index-Werten angewendet und sind sichtbar gleichmäßiger verteilt.⁴² Für den neu berechneten Index zur Gentrifizierung nach Döring und Ulbricht (2016) wurden ebenfalls Klassengrenzen gebildet, die möglichst gleichverteilt große Klassen bilden. Einmal in zwei Klassen (*negativ < 0 <= positiv*), sowie einmal mit Mittelklasse von (*negativ < -1 <= mittel <= 1 < positiv*).

Für die Erstellung der Ergebnistabelle werden alle *_stock*, *_ytd* und *_new*-Kennzahlen ausgeleitet. Auf *_steady* und *_deleted* wird verzichtet. Für die *_ytd* und *_new*-Kennzahlen

³⁸ Siehe Abbildung 9-27: Hive-Code zu result_full_plr (abgekürzt)

³⁹ Siehe Abbildung 9-24: Hive-Code zu lor_ewr_einwohnergewichtet

⁴⁰ Betroffen sind die Planungsräume *Großer Tiergarten*, *Westhafen*, *Lietzengraben*, *Olympiagelände*, *Güterbahnhof Grunewald*, *Stadion Wilmersdorf*, *Forst Grunewald*, *Am Treptower Park Nord* und *Tegeler Forst*

⁴¹ Siehe Abbildung 9-25: MSS Vergleich Klassengrößen

⁴² Siehe Abbildung 9-26: MSS Klassengrenzen

werden die Werte der Jahre 2016, sowie 2017 aggregiert. Für stock wird der Wert zu Beginn des Jahres 2017 genommen. Der Hive-Code für die Erzeugung der Ergebnistabelle ist abgekürzt im Anhang 9.4.6 unter Abbildung 9-27 enthalten, das Beispiel der POI-Domäne „Gastronomie“ ist für alle weiteren POI-Domänen, -Kategorien und -Typen analog.

5.2.5 Nachträgliche Erweiterungen der Datenaufbereitung

5.2.5.1 Distanzgewichtete Aggregation umliegender Features

Im Rahmen der Modellierungsphase wurden Verbesserungspotentiale bei der Datenaufbereitung festgestellt. So sind die Planungsräume relativ klein und berücksichtigen nicht die umliegenden POIs der benachbarten PLR. Unter der Annahme, dass auch POIs aus nahen Planungsräumen einen Einfluss auf den betrachteten PLR haben können, wurde ein Gewichtungsfaktor mit in die Berechnung aufgenommen. Der Faktor aggregiert die Kennzahlen in Abhängigkeit zur Distanz zu umliegenden PLR. Dieser Gewichtungsfaktor wurde über eine exponentiell regressive Funktion basierend auf dem Abstand der Polygontzentren der PLR ermittelt. Abbildung 5-14 zeigt die Formel zur Berechnung und stellt den Gewichtungsfaktor grafisch dar. Die Formel sorgt dafür, dass nahe PLR stärker einfließen als weiter entfernte. Der PLR hat zu sich selbst eine Distanz von 0, weshalb er zu 100% gewichtet wird. Bei einer Distanz von einem halben Kilometer zwischen zwei PLR liegt der Gewichtungsfaktor bei 53%, bei einem Kilometer beträgt er nur noch 23%. Ab einer Distanz von ca. 1,64 Kilometern und darüber ist die Gewichtung 0%. Der Faktor wird in das Innere der Pivotierungsschicht in *osm_poi_features_domain_piv* am Ende von Kapitel 5.2.2 integriert. Die technische Dokumentation der Distanzberechnung, der Gewichtungskalkulation und der Integration in die Pivotierungsschicht befindet sich im Anhang in Kapitel 9.4.7. Der Vergleich zwischen PLR mit und ohne Gewichtungsfaktor folgt in Kapitel 6.

$$\text{Gewichtungsfaktor} = \max(3^{-\text{distanzInKm}} - 0.1 \cdot \text{distanzInKm}, 0)$$

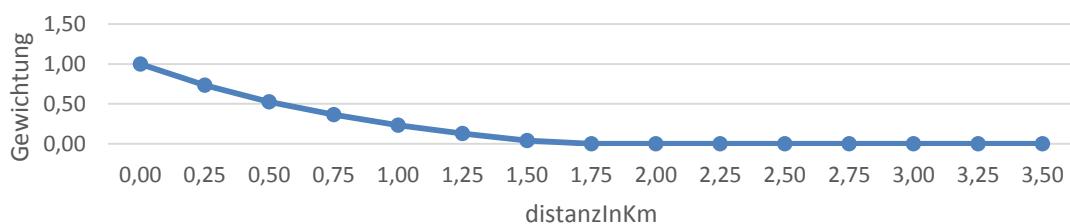


Abbildung 5-14: Formel und Graph des Gewichtungsfaktors

5.2.5.2 Raumbezugsgröße Bezirksregion

Als weitere alternative Betrachtung der Daten soll die Auflösung der Raumbezugsgröße geändert werden. Die nächst gröbere Auflösung in der LOR-Hierarchie ist die der Bezirksregion. Auf Basis der Bezirksregionen können einige Werte direkt aggregiert werden, andere benötigen jedoch eine erneute Berechnung. So können die OSM und EWR-Daten einfach aufsummiert werden und danach mit der Anzahl der Einwohner wieder normalisiert werden. Für die Indexwerte muss jedoch eine erneute z-Transformation erfolgen. Hierzu wurde der MSS-Wert, sowie der Index nach Döring und Ulbricht (2016) auf Ebene der Bezirksregion nachgerechnet.⁴³ Für die Bezirksregionen wurde kein Gewichtungsfaktor berechnet. Die 447 Planungsräume wurden in 138 Bezirksregionen aggregiert. Die Bezirksregion Forst Grunewald (040617) wurde von der Auswertung ausgeschlossen, da sie weniger als 300 Einwohner hat.

5.2.5.3 Offering Advantage

Zusätzlich zu den POI-Kennzahlen *_ytd*, *_new* und *_stock* wurde dann noch eine Abwandlung der *Offering Advantage*, die bei Venerandi et al. (2015, S. 257) genutzt wurde, berechnet.⁴⁴ Diese setzt die Anzahl eines POI-Typen wie z.B. italienische Restaurants in einer Nachbarschaft mit der Anzahl derselben POI-Kategorie im Stadtgebiet in Verbindung. So werden die charakteristischen Eigenschaften der Nachbarschaften stärker in den Vordergrund gestellt. Allerdings würde nach dem Vorgehen von Venerandi et al. (2015, S. 257), wie in Abbildung 5-15 dargestellt, die OA einer POI-Kategorie auf Basis von allen POIs ermittelt werden. Damit würden auch Geldautomaten mit Restaurants verglichen werden.

$$\frac{\text{count}(\text{RestaurantItaliener}, \text{Wrangelkiez})}{\text{count}(all, \text{Wrangelkiez})} \cdot \frac{\text{count}(all)}{\text{count}(\text{RestaurantItaliener})}$$

Abbildung 5-15: OA Berechnung Typ über alle POIs

Analog dazu soll die OA in dieser Arbeit für alle POI-Domänen ermittelt werden, basierend auf der Anzahl aller POIs. Daneben soll aber die OA auch innerhalb von einigen Domänen berechnet werden. Die obere Formel aus Abbildung 5-16 ermittelt, wie dominant eine Domäne in einem LOR ist. In der unteren Formel wird die OA mit der Anzahl eines einzelnen Restaurant-Typen mit der Anzahl aller Gastronomie-POIs berechnet.

⁴³ Siehe Abbildung 9-22: Hive-Code zu lor_mss_idx_bzr_z und Abbildung 9-23: Hive-Code zu lor_mss_idx_bzr_idx – own_idx analog Abbildung 9-21: Hive-Code der Strecke zu lor_own_idx_plr_tb

⁴⁴ Vgl. Abbildung 3-3: Formel zur Offering Advantage. (Eigene Darstellung nach Venerandi et al., 2015)

$$\frac{\text{count}(\text{Gastronomie}, \text{Wrangelkiez})}{\text{count}(\text{all}, \text{Wrangelkiez})} \cdot \frac{\text{count}(\text{all})}{\text{count}(\text{Gastronomie})}$$

$$\frac{\text{count}(\text{RestaurantItaliener}, \text{Wrangelkiez})}{\text{count}(\text{Gastronomie}, \text{Wrangelkiez})} \cdot \frac{\text{count}(\text{Gastronomie})}{\text{count}(\text{RestaurantItaliener})}$$

Abbildung 5-16: OA Berechnung Typ innerhalb von Domäne

Auf diese Weise wurden 85 OA Kennzahlen berechnet, innerhalb der Domänen wurde zum Teil Kennzahlen auf Basis der darunter liegenden Kategorien gebildet, zum Teil wurde auch die unterste Hierarchiestufe der Typen für die OA Kennzahlen genutzt. Eine Übersicht über die neuen Kennzahlen ist in Tabelle 5-7 zu finden. Die gesamte Hierarchie ist im Anhang unter Tabelle 9-5: POI-Hierarchie zu finden.

Tabelle 5-7: OA Kennzahlen

Bezugsgröße	Bezugsgröße Hierarchiestufe	OA Kennzahlen Hierarchiestufe	Resultierende Anzahl OA Kennzahlen
alle POIs	(alle)	Domäne	13
Gastronomie	Domäne	Kategorie	3
Gastronomie	Domäne	Typ	19
Dienstleistung	Domäne	Kategorie	6
Public Service	Domäne	Kategorie	6
Waren	Domäne	Kategorie	12
Vergnügen	Domäne	Typ	12
Sport	Domäne	Typ	14

6 Modeling & Evaluation

In diesem Kapitel sollen die Forschungsfragen „*Welche Features aus den Daten eignen sich zum Aufbau eines Prognosemodells für Gentrifizierung?*“ und „*Welche Algorithmen eignen sich zum Aufbau eines Prognosemodells für Gentrifizierung?*“ beantwortet werden. Damit entspricht dieses Kapitel den CRISP-DM Phasen *Modeling* und *Evaluation*.

6.1 Tool- und Featureauswahl

Für die Modellierung wurde das Data Mining Programm Weka in der Version 3.8.2, sowie die statistische Programmiersprache R verwendet. Weka wurde von der Machine Learning Group der Universität von Waikato in Neuseeland entwickelt und enthält eine Vielzahl von Machine Learning Algorithmen (Witten, Pal, Frank & Hall, 2016).

Für die Modellierung stehen drei Tabellen je Raumbezugsgröße (PLR, PLR mit Distanzgewichtung und BZR) zur Verfügung, die Tabellen enthalten jeweils 1.722 Spalten. Die hohe Anzahl an Spalten ergibt sich aus:

- 2 Betrachtungszeiträumen – aktuell (201612) und vorher (201412)
- * (722 Datenfeatures (= 3 Kennzahlen – _stock, _ytd, _new
 - * 229 POI-Klassen – Domänen, Kategorie, Typen
 - + 85 OA Kennzahlen)
- + 89 Zusatzinformationen (Indexwerte, Daten zum Raum, Zensusdaten))

Die drei erzeugten Dateien enthalten für die Planungsräume jeweils 436 Zeilen⁴⁵, bzw. 137 Zeilen für die Bezirksregionen. Für die einzelnen Modelle werden jedoch immer nur ein Teil der Spalten bzw. der Datenfeatures benötigt. Deshalb wurden für jede Hypothese aus Kapitel 4.5 drei Dateien erzeugt (PLR, PLR mit Distanzgewichtung und BZR). In Tabelle 6-1 sind die Hypothesen mit den benötigten Daten dargestellt.

⁴⁵ Die Datenbasis für die Untersuchung der Hypothese H3b basiert auf den MSS-Daten von 2014. Zu diesem Zeitpunkt war der Planungsraum „Gleisdreieck/Entwicklungsgebiet“ von dem MSS als zusätzlicher Ausreißer ausgeschlossen. Deshalb wurde dieser Planungsraum nur für die Untersuchung dieser Hypothese ebenfalls ausgeschlossen.

Tabelle 6-1: Übersicht Daten für Modellbildung

Hypothesen	Indexwert			Featuregruppe			
	MSS-Status	MSS-Dynamik	Döring & Ulbricht (2016)	OSM_stock OA	OSM_stock	OSM_new	OSM_ytd
H1 (a-c)	201612			201701			
H2		201612 (Änderung seit 201412)		201501			
H3a		201612 (Änderung seit 201412)				201401 + 201501	
H3b		201412 (Änderung seit 201212)				201601 + 201701	
H3c		201612 (Änderung seit 201412)				201601 + 201701	

6.2 Versuchsaufbau

Die Modelle werden jeweils auf Basis der Index-Kategorien trainiert. Als Klassifizierungsalgorithmen sollen zunächst die drei Algorithmen *RandomTree*, *RandomForest* und *LMT* genutzt werden, um indikativ die Güte der Datensätze, Featuregruppen und Indexwerte zu ermitteln. Basierend darauf wird eine Testreihe mit verschiedenen Algorithmen auf den besten Datenkombinationen aufgesetzt. In der Überprüfung der Hypothesen sollen dann ausgewählte Algorithmen untersucht werden.

6.2.1 Klassifizierungsalgorithmen

6.2.1.1 RandomTree

Der RandomTree-Algorithmus baut einen Entscheidungsbaum auf, indem bei jedem Knoten K^{46} zufällig ausgewählte Datenfeatures zur Auswahl stehen. Der Algorithmus wählt jeweils das Feature aus, das am besten die *Entropie*⁴⁷ der Daten reduziert. Eine minimale Entropie würde ein genaues Aufteilen der Datensätze in die Klassen bedeuten. Der Algorithmus versucht also, eine möglichst saubere Trennung der Klassen zu erreichen. Dies wird so lange wiederholt, bis alle Blätter des Baumes genau eine Klasse ent-

⁴⁶ Dabei wird das K entweder übergeben, oder auf Basis von $\log_2(\text{anzahlDatenFeatures})+1$ ermittelt. Siehe Witten, Pal, Frank und Hall (2016, S. 36).

⁴⁷ Die Entropie von einem Blatt mit 45 Instanzen der Klasse A und 5 Instanzen der Klasse B wäre ca. 0,47 = - $\text{anzA} / (\text{anzA}+\text{anzB}) * \log_2(\text{anzA}/(\text{anzA}+\text{anzB})) - \text{anzB} / (\text{anzA}+\text{anzB}) * \log_2(\text{anzB}/(\text{anzA}+\text{anzB}))$. Siehe Witten, Pal, Frank und Hall (2017, S. 110)

halten (Witten et al., 2017, S. 105–113). Bei der Weka-Implementierung des Baumes findet kein *Pruning*⁴⁸ des Baumes statt, es kann jedoch eine maximale Baumtiefe angegeben werden.

6.2.1.2 RandomForest und weitere Ensemble Klassifikatoren

Der RandomForest (Breiman, 2001) ist ein homogener *Ensemble* Klassifikator, der auf den oben genannten RandomTrees basiert. Ensemble Klassifikatoren kombinieren verschiedene andere Modelle und nutzen diese dann um ein Gesamturteil über die Klassifikation zu fällen. Die verbreitetsten Formen von Ensembles sind *Voting*, *Bagging*, *Randomizing*, *Boosting* und *Stacking* (Witten et al., 2017, S. 480).

Tabelle 6-2: Übersicht der Arten von Ensembles

Art des Ensembles	Klassifikations-Algorithmen	Entscheidungsfindung	Beschreibung
Bagging	1	Gleiches Stimmrecht	Gleicher Algorithmus wird mehrfach mit zufälliger Teilmenge der Daten trainiert.
Randomizing	1	Gleiches Stimmrecht	Gleicher zufallsbasierter Algorithmus wird mehrfach mit den gleichen Daten trainiert, aber anderer RandomSeed.
Boosting	1	Gewichtetes Stimmrecht	Mehrere Iterationen mit neuer Kosten-Gewichtung der bisher falsch klassifizierten Instanzen (FP/FN). Stimmrechtsgewichtung auf Basis der Güte
Voting	N	Gleiches Stimmrecht	Mehrheitsentscheidung der Klassifikationsalgorithmen
Stacking	N + 1 Meta-Algorithmus	Meta-Algorithmus	Meta-Algorithmus lernt auf Basis der Vorhersagen der Klassifikationsalgorithmen.

Tabelle 6-2 stellt die verschiedenen Ensembles einander gegenüber. Beim Bagging, Voting und Randomizing haben alle im Ensemble verwendeten Modelle gleiches „Stimmrecht“, während beim Boosting die Klassifikatoren ein Stimmrecht in Abhängigkeit ihrer Klassifikationsgüte erhalten. Beim Stacking dagegen basiert auf zwei Ebenen, in der unteren Ebene errechnen mehrere Klassifikatoren ihr Ergebnis, während in der oberen Ebene ein Meta-Klassifikator diese Ergebnisse für eine endgültige Klassifikation verwendet.

⁴⁸ Beim Pruning werden nachträglich Äste des Baumes entfernt, die nur einen geringen Informationsgewinn haben (*information gain*). Der Informationsgewinn berechnet sich durch die Differenz der aktuellen Entropie abzüglich der Summe der Entropien der beiden neuen Blätter. Siehe Witten et al. (2017, S. 110)

det. (Witten et al., 2017, S. 480–500). RandomForest ist ein Randomizing Ensemble bestehend aus vielen RandomTrees, welchen jeweils eine andere Zufallszahl für die Auswahl der zufälligen Datenfeatures übergeben wird.

6.2.1.3 Logistic Model Tree und LogitBoost Algorithmus

Der *Logistic Model Tree* (LMT) (Landwehr, Hall & Frank, 2005) ist eine Ensemble-Spezialform. Der Klassifikator besteht aus einem Entscheidungsbaum mit *logistischer Regression* in den Blättern. Logistische Regression ist eine Form der Klassifizierung über Regression. Dabei werden die nominalen Klassenwerte in numerische Werte umgewandelt. Darauf basierend wird eine logistische Regressionsberechnung durchgeführt, mit welcher der Klassenwert auf Basis der Datenfeatures ermittelt werden soll. Die Berechnung erfolgt mit dem *LogitBoost* Algorithmus (Landwehr et al., 2005, S. 164–167). „*LogitBoost performs additive logistic regression.*“ (Witten et al., 2017, S. 496) In jeder Iteration wird das nächste jeweils beste Feature für die logistische Regression hinzugefügt. Der Algorithmus stoppt, sobald der Klassifikationsfehler nicht weiter sinkt (Frank, 2014). An diesem Punkt wird die Datenbasis aufgeteilt und es entstehen zwei Äste. In den Blättern wird dann wiederum der LogitBoost Algorithmus ausgeführt (Witten et al., 2017, S. 497).

6.2.2 Kreuzvalidierung und Kennzahlen der Modellevaluation

Für das Trainieren von Klassifizierungsmodellen werden die Daten in Test- und Trainingsdaten aufgeteilt. Bei großen Datenmengen und wenigen Klassen wird hierbei oft ein einfaches Aufteilen in 2/3 Trainingsdaten und 1/3 Testdaten angewendet. Mit den Trainingsdaten wird das Modell erstellt und anschließend mit den Testdaten evaluiert (Zeng & Martinez, 2000, S. 3–4). Beim Erzeugen von Modellen auf Basis kleiner Datensätze, geht bei einem Aufteilen der Daten in Trainings- und Testdaten viel Potential für das Trainieren der Modelle verloren. Daher ist bei kleinen Datenmengen das Verfahren der Kreuzvalidierung von Vorteil. Dabei wird der gesamte Datensatz in n Teile aufgeteilt, in dieser Arbeit wird mit 10-facher Kreuzvalidierung gearbeitet. Das Modell wird dann 10-mal mit jeweils mit 90% der Daten trainiert, und mit 10% der Daten getestet. Die Ergebnisse der Tests werden dann für die gesamte Statistik gemittelt (Witten et al., 2017, S. 167–168). Das finale Modell wird durch einen elften Durchlauf auf Basis von allen Daten trainiert. Damit stehen alle Daten für Training zur Verfügung, ohne das dies die Testergebnisse verfälscht. Weka nutzt für Klassifizierungen automatisch eine *Stratified Cross-Validation*, bei der jedes der n Teile der Daten eine möglichst gute Repräsentierung

aller Klassen enthält (Bouckaert et al., 2018). Auf diese Weise wird eine mögliche Verzerrung (*bias*) durch eine Unter- oder Überrepräsentation von einer Klasse in Trainings- und Testdaten vermieden, die durch das zufällige Aufteilen der Daten entstehen könnte. Die erstellten Modelle werden anhand der Testdaten evaluiert. Dabei wird mit dem erzeugten Modell für die Testdaten eine prognostizierte Klasse ermittelt und mit der tatsächlichen Klasse verglichen. Die Ergebnisse können in einer *Konfusionsmatrix* (engl. *Confusion Matrix*) dargestellt werden:

Tabelle 6-3: Konfusionsmatrix Beispiel

a	b	c	<-- classified as
28	12	8	a = mittel
11	36	1	b = hoch
9	5	27	c = niedrig

Tabelle 6-4: Kennzahlen für Klasse a

a	b	c	<-- classified as
TP	FN	FN	a = mittel
FP	TN	TN	b = hoch
FP	TN	TN	c = niedrig

Auf Basis der Konfusionsmatrix können eine Vielzahl von Gütekriterien ermittelt werden. Als Grundlage dient die Bestimmung der folgenden Basiskennzahlen:

- True Positive (TP) – Richtig klassifiziert als in der Klasse
- True Negative (TN) – Richtig klassifiziert als nicht in der Klasse
- False Positive (FP) – Falsch klassifiziert als in der Klasse
- False Negative (FN) – Falsch klassifiziert als nicht in der Klasse

Diese Basiskennzahlen können dann in diverse Kennzahlen überführt werden. In Tabelle 6-5 sind diese Kennzahlen zu der Konfusionsmatrix aus Tabelle 6-3 berechnet worden:

- TP Rate (= Recall, Sensitivity) – $TP / (TP + FN)$
- FP Rate – $FP / (FP + TN)$
- Precision – $TP / (TP + FP)$
- F1-Measure – $2 * Recall * Precision / (Recall + Precision)$

Tabelle 6-5: Beispiel Gütekriterien

TP Rate	FP Rate	Precision	F1-Measure	ROC Area (= AUC)	Class
0,583	0,225	0,583	0,583	0,676	a = mittel
0,750	0,191	0,679	0,713	0,889	b = hoch
0,659	0,094	0,750	0,701	0,878	c = niedrig
0,664	0,174	0,667	0,664	0,811	(weighted avg)

Um die Güte eines Modells an einer Kennzahl abzulesen, gibt es viele verschiedene Kennzahlen und Messgrößen. Es gibt keine Messgröße, die auf jeden Kontext passt (Powers, 2011, S. 2). Mit der *F1-Measure* und der *Area Under the Curve* (AUC) sollen zwei typische Gütekriterien in dieser Arbeit für den Modellvergleich genutzt werden.

Die F1-Measure berechnet sich durch das harmonische Mittel aus Recall und Precision. Ist F1 genau eins, so ist das Modell perfekt, es gäbe keinen FP und keinen FN. Ein kleinerer F1 Wert muss jedoch nicht in jedem Fall bedeuten, dass ein Modell schlecht ist. Es müssen die Kosten für FP- und FN-Klassifizierungen beachtet werden (Renuka, 2016). So wäre es zum Beispiel bei einer Werbekampagne wichtiger, jeden potentiell interessierten Kunden zu erkennen, als die Anzahl derer zu maximieren, die richtigerweise nicht angeschrieben werden sollen; FN hätte also höhere Kosten als FP. In einem solchen Fall wäre die Maximierung des Recalls für die Klasse der potentiell interessierten Kunden im Fokus. In unserem Beispiel sollen die Modelle über ihre gesamte Prognosegüte über alle Klassen verglichen werden. Hierzu berechnet Weka ein gewichtetes Mittel⁴⁹ des F1-Wertes. Die folgende Skala soll als Interpretationshilfe dienen:

- 0,8-1,0 = A
- 0,7-0,8 = B
- 0,6-0,7 = C
- 0,5-0,6 = D
- <0,5 = F

Der AUC eignet sich laut Witten et al. (2017, S. 191–192) für den Vergleich von Modellen bei unbekannten Kosten der Klassifizierung, also falls nicht bekannt ist ob ein FP oder ein FN teurer ist. Der AUC stellt die Fläche unter der ROC-Kurve in Abbildung 6-1 dar.

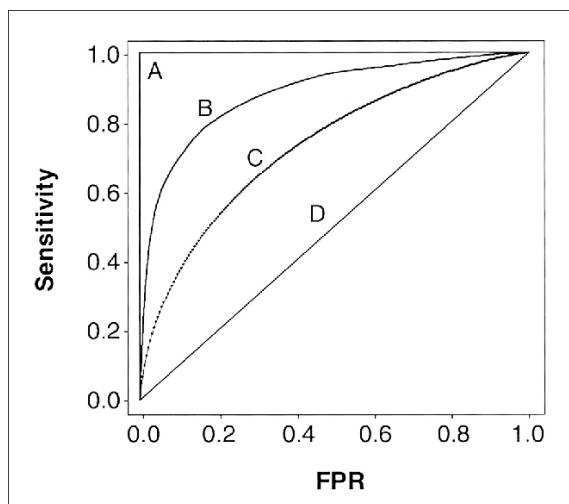


Abbildung 6-1: ROC Beispiel (Seong Ho Park, Jin Mo Goo & Chan-Hee Jo, 2004)

Die Kurve des *Receiver Operating Characteristic* (ROC) besteht aus der TP- und der FP-Rate. Je schneller sich die Funktion einer TP-Rate von 1,0 annähert, desto besser ist das Modell. Der AUC-Maximalwert liegt bei 1,0 – ein Wert von 0,5 würde ein nutzloses Modell bedeuten (Fawcett, 2004). Auch der AUC-Wert wird von Weka gewichtet für das gesamte Modell ausgegeben. Tape (2015) hat für den AUC die folgende Interpretationsskala aufgestellt:

- 0,9-1,0 = ausgezeichnet (A)
- 0,8-0,9 = gut (B)
- 0,7-0,8 = mittelmäßig (C)
- 0,6-0,7 = schwach (D)
- 0,5-0,6 = mangelhaft (F)

⁴⁹ Gewichtung über den prozentualen Anteil der Klasse an allen Datensätzen.

6.2.3 Evaluation der Datenbasis

Um die Güte der Raumbezugsgrößen, Featuregruppen zu Indexwerte zu vergleichen, wurden 486 Experimente systematisch durchgeführt. Jeder Algorithmus wurde mit Standardparametern und zehnfacher Kreuzvalidierung ausgeführt. Die gewichteten Gütekennzahlen wurden dokumentiert und sollen nun eine Indikation der Güte der verschiedenen Auswahlmöglichkeiten geben.

Mit Blick auf Tabelle 6-6 und Tabelle 6-7 wird ersichtlich, dass beim Raumbezug die Bezirksregion deutlich bessere Ergebnisse erzielt als der Planungsraum. Der Planungsraum mit der Distanzgewichtung erhält jedoch ebenfalls gute Modelle. Da die Anzahl der Planungsräume deutlich größer ist, als die der Bezirksregionen, sollen für die Evaluation der Algorithmen beide Varianten genutzt werden.

Tabelle 6-6: Raumbezugsgröße - AUC

Raumbezug	A	B	C	D	F
bzr		5	18	31	108
plr			10	21	131
plr_distcalc		2	19	44	97
Summe	7	47	96	336	

Tabelle 6-7: Raumbezugsgröße - F1

Raumbezug	A	B	C	D	F
bzr	1	6	28	39	88
plr		4	10	63	85
plr_distcalc		5	27	68	62
Summe	1	15	65	170	235

Bei den Featuregruppen muss zwischen Bestands- und Dynamikkennzahlen unterschieden werden. In Tabelle 6-8 fällt auf, dass die Bestandskennzahlen bessere Modelle ergeben, als die Dynamikkennzahlen. Bei den Bestandskennzahlen ist zu sehen, dass das Hinzufügen der OA Kennzahlen die Modelle verbessert hat. Insbesondere beim AUC gibt es mehr gute Modelle im Vergleich zum Datenbestand ohne OA Kennzahlen. Für die Evaluation der Algorithmen sollen nur OA Kennzahlen einfließen. Bei der Dynamik sollen die _new Kennzahlen genutzt werden, da diese die besten Ergebnisse liefern.

Tabelle 6-8: Featuregruppe – AUC & F1

Featuregruppe		AUC					F1				
		A	B	C	D	F	A	B	C	D	F
Bestand H1 + H2	all		3	9	10	32	1	5	8	19	21
	no OA		1	11	12	30		4	10	19	21
	OA only		3	10	12	29		5	11	15	23
	Summe	7	30	34	91	1	14	29	53	65	
Dynamik H3 (a-c)	all			5	20	83			11	39	58
	new only			7	23	78		1	14	40	53
	ytd only			5	19	84			11	38	59
	Summe			17	62	245		1	36	117	170

Die Indexwerte dienen als Zielwert der Klassifizierung. Aus Tabelle 6-9 und Tabelle 6-10 können zwei Erkenntnisse geschlossen werden. Erstens, der Dynamik-Index nach Döring und Ulbricht (2016) ist mit den Daten, die in den Modellen vorhanden sind, aktuell nicht mit einem Modell vorherzusagen. Zweitens, die binäre Klassifizierung erzielt immer bessere Gütekennzahlen als die Klassifizierung mit Mittelkategorie. Den gleichen Effekt haben bereits Venerandi et al. (2015, S. 260) festgestellt. Für die Modellierung der Dynamik soll deshalb der binäre Dynamikindex des MSS verwendet werden. Die Modellierung des Status soll analog auf dem binären Statusindex des MSS basieren.

Tabelle 6-9: Index - AUC

Index	A	B	C	D	F
mss_dyn_prj			3	36	69
mss_dyn_prj_bi			19	34	55
doe_ulb				10	98
doe_ulb_bi				5	103
mss_stat_prj		1	14	3	9
mss_stat_prj_bi		6	11	8	2

Tabelle 6-10: Index - F1

Index	A	B	C	D	F
mss_dyn_prj				1	19
mss_dyn_prj_bi			1	46	53
doe_ulb					108
doe_ulb_bi				2	85
mss_stat_prj				6	11
mss_stat_prj_bi	1	14	10	2	

6.2.4 Evaluation der Algorithmen

Auf Basis der Evaluation der Datengrundlage wurden zehn Dateien für einen systematischen Vergleich verschiedener Algorithmen für die Hypothesen erstellt. Je Hypothese (H1, H2, H3a, H3b, H3c) wurden zwei Dateien erstellt, einmal auf Basis der Planungsräume mit distanzgewichteten Daten, einmal auf Basis der Bezirksregionen. Für den Vergleich wurden die folgenden Algorithmen gewählt:

- 3x RandomForest (1x Standardparameter, 1x maximale Baumtiefe = 5, 1x maximale Baumtiefe = 3)
- 2x LMT (1x Standardparameter, 1x fixe Anzahl Boosting-Iterationen für LogitBoost = 3)
- 1x LogitBoost (Standardparameter)
- 2x SimpleLogistic⁵⁰ (1x Standardparameter, 1x fixe Anzahl Boosting-Iterationen für LogitBoost = 3)
- 1x BayesNet (Standardparameter)
- 1x NaiveBayes (Standardparameter)
- 1x J48⁵¹(Standardparameter)
- 1x AdaBoostM1 – Boosting Ensemble (Algorithmus: J48)
- 1x Stacking – Ensemble (Meta-Algorithmus: J48, Algorithmen: 3x RandomForest (s.o), 2x LMT (s.o.), 2x SimpleLogistic (s.o.))
- 1x Bagging – Ensemble (Algorithmus: SimpleLogistic mit Standardparametern)
- 1x Voting – Ensemble (Algorithmen: 3x RandomForest (s.o), 2x LMT (s.o.), 2x SimpleLogistic (s.o.))
- 1x MultilayerPerceptron – Neuronales Netzwerk (20 hidden layer)

⁵⁰ Logistische Regression, basierend auf LogitBoost

⁵¹ Optimierender Entscheidungsbaum mit Pruning

Algorithmus	AUC	F1
02 - RandomForest	● 0,74	● 0,67
03 - RandomForest_5	● 0,73	● 0,67
04 - RandomForest_3	● 0,72	● 0,66
05 - LMT	● 0,68	● 0,64
06 - LMT_3	● 0,69	● 0,64
08 - LogitBoost	● 0,69	● 0,64
09 - SimpleLogistic	● 0,67	● 0,63
10 - SimpleLogistic_3	● 0,68	● 0,62
11 - BayesNet	● 0,68	● 0,63
12 - NaiveBayes	○ 0,61	○ 0,54
13 - J48	○ 0,59	● 0,58
14 - AdaBoostM1	● 0,65	● 0,61
15 - Stacking	● 0,65	● 0,65
16 - Bagging	● 0,70	● 0,63
18 - Vote	● 0,73	● 0,66
19 - MultilayerPerceptron	● 0,64	● 0,58

Abbildung 6-2: Gemittelte Güte der Algorithmen

Die Übersichtsgrafik der Ergebnisse (Abbildung 9-34: Übersicht Evaluation der Algorithmen) befindet sich im Anhang in Kapitel 9.5. Darin sind jeweils die zwei besten Algorithmen je Hypothese und Betrachtungsraum markiert. In Abbildung 6-2 sind die gemittelten Gütekennzahlen aus der oben genannten Gesamtübersicht je Algorithmus dargestellt. Damit kann zwar keine Aussage getroffen werden, welche Algorithmen sich generell für die Untersuchung von Gentrifizierung eignen, es fällt jedoch auf, dass bestimmte Algorithmen mit dem Datenbestand in dieser Arbeit besser umgehen können als andere. Insbesondere der RandomForest Algorithmus

scheint sehr gute Modelle zu generieren. Dies stimmt mit der Beobachtung einiger Autoren überein, dass der RandomForest Algorithmus oft sehr gute Modelle erzeugt, die nur schwer zu überbieten sind (Deeb, 2015; Donges, 2018). Ebenfalls zeigen sich die Modelle, die auf logistischer Regression basieren, als relativ stark. Das Neuronale Netz, NaiveBayes und der Entscheidungsbaum J48 dagegen erzeugen deutlich schlechtere Ergebnisse. In den Ensembles werden RandomForests und logistische Regressionsmodelle genutzt, weshalb diese ebenfalls ähnlich stark ausfallen.

6.3 Überprüfung der Gentrifizierungs-Hypothesen

Die in Kapitel 4.5 aufgestellten Hypothesen werden nun anhand der Modelle überprüft. Außerdem werden Korrelationen in den Daten analysiert. Mit Blick auf die Hypothesen

Hypothese Raumbezug	Durchschnitt		Bestes Modell	
	AUC	F1	AUC	F1
h1_bzr	● 0,80	● 0,73	● 0,88	● 0,79
h1_plr-dc	● 0,76	● 0,70	● 0,83	● 0,75
h2_bzr	○ 0,62	○ 0,58	● 0,77	● 0,65
h2_plr-dc	● 0,67	● 0,63	● 0,73	● 0,69
h3a_bzr	○ 0,54	○ 0,51	○ 0,61	○ 0,58
h3a_plr-dc	● 0,61	● 0,58	● 0,72	● 0,67
h3b_bzr	● 0,71	● 0,66	● 0,81	● 0,72
h3b_plr-dc	● 0,66	○ 0,60	● 0,74	● 0,68
h3c_bzr	● 0,68	● 0,64	● 0,75	● 0,71
h3c_plr-dc	● 0,67	● 0,61	● 0,74	● 0,70

lässt sich in der Übersicht in Abbildung 6-3 bereits ableiten, welche Hypothesen basierend auf den erhobenen Daten eher zutreffend sind, und welche nicht. Im Folgenden soll dies anhand ausgewählter Modelle untersucht werden.

Abbildung 6-3: Übersicht der Güte der Hypothesen

6.3.1 H1 – Zusammenhang zwischen Angebotsstruktur und sozialem Status

Die Modelle für die Hypothese H1 haben durchweg gute Werte, was für einen Zusammenhang zwischen lokaler Angebotsstruktur und sozialem Status spricht. Das beste Modell für H1 ist das Voting-Ensemble, welches auf vielen ebenfalls guten RandomForests und logistischen Regressionsmodellen beruht. Der zweitbeste Klassifikator für die Bezirksregion ist ein sehr einfaches logistisches Regressionsmodell:

Class niedrig :

$$\begin{aligned} & 0.9 + \\ & [\text{r.oa_gastro_c_restaurant_stock}] * -1.53 + \\ & [\text{r.oa_pubserv_c_sozial_stock}] * 0.3 + \\ & [\text{r.oa_vergnuegung_t_spielothek_stock}] * 0.15 \end{aligned}$$

Class hoch :

$$\begin{aligned} & -0.9 + \\ & [\text{r.oa_gastro_c_restaurant_stock}] * 1.53 + \\ & [\text{r.oa_pubserv_c_sozial_stock}] * -0.3 + \\ & [\text{r.oa_vergnuegung_t_spielothek_stock}] * -0.15 \end{aligned}$$

Mit diesem einfachen Modell können fast 80% der Bezirksräume richtig in niedrigem oder hohem sozialen Status eingeordnet werden. In der Kategorie Sozial sind Nachbarschaftszentren und Sozialeinrichtungen enthalten.

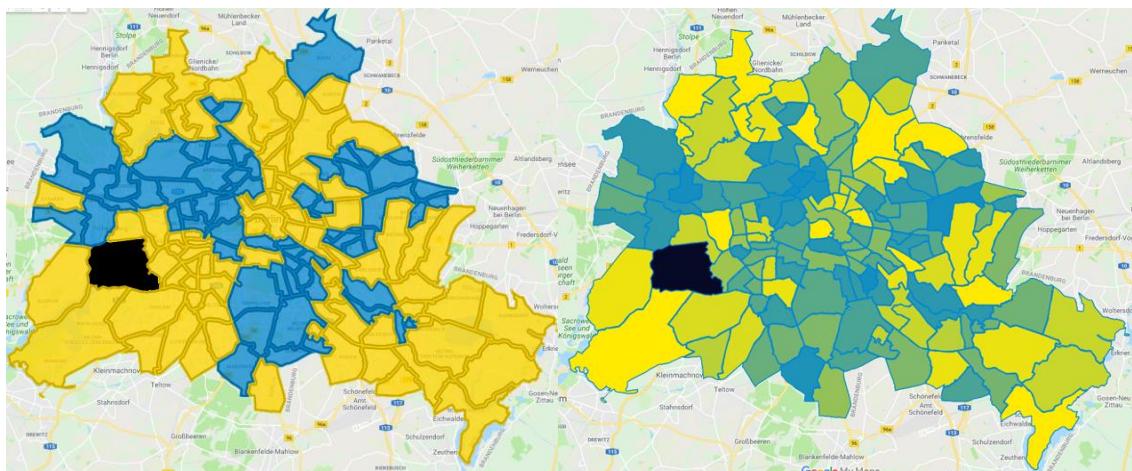


Abbildung 6-4: Grafische Aufbereitung des logistischen Regressionsklassifikators für H1/BZR (Eigene Darstellung mit Google MyMaps)

Dieses Modell konnte auch grafisch dargestellt werden. Abbildung 6-4 zeigt links die MSS-Einteilung in hohem (gelb) und niedrigen (blau) Status. Auf der rechten Seite sind eine gelbe Basiskarte und drei überlagerte Ebenen mit transparentem Blau dargestellt, je Kennzahl der Regression eine. Die Karte ist stark vereinfacht, da die Transparenz nur näherungsweise über die Quartile der Kennzahlen erstellt werden konnte. Jede Ebene, sowie die beiden Karten sind im Großformat im Anhangs Kapitel 9.6 verfügbar. Dennoch

ist eine Ähnlichkeit zu erkennen. Insgesamt lässt sich für die Hypothese H1 „*Der soziale Status einer Nachbarschaft hängt mit den lokalen Angebotsstrukturen zusammen.*“ feststellen, dass es einen starken Zusammenhang zwischen lokaler Angebotsstruktur und sozialem Status gibt.

Ergänzend zu H1 wurden die Hypothesen H1a – H1c basierend auf vorherigen Forschungsergebnissen aufgestellt. Diese können als Spezialformen der Grundhypothese angesehen werden und beziehen sich auf spezielle POI-Kategorien – *Cafés*, *Fast Food* und *Sport*. Die Hypothesen wurden zunächst mit dem statistischen Programm R untersucht. Hierzu wurden jeweils die einwohnergewichteten Bestandswerte (_stock) und deren OA auf Korrelation mit dem metrischen Index-Wert zum sozialen Status untersucht. Als statistische Methode wurde sowohl die Pearson-Korrelation, als auch die Spearman-Rangkorrelation verwendet. Letztere ist robust gegen Ausreißer (Universität Zürich, o.D.), wie sie z.B. durch relativ niedrige Einwohnerzahlen in einem PLR vorkommen können. Neben den Hypothesen wurden die Top-Korrelationen aller Hypothesen ausgewertet. Dabei handelt es sich ausschließlich um Pearson-Korrelationen zwischen dem Indexwert aus dem MSS (bei H1 Status, sonst Dynamik) und allen verfügbaren Datenfeatures.⁵²

Tabelle 6-11: H1 – Zusammenhang zwischen Angebotsstruktur und sozialem Status

cor	Feature
-0,45	r.oa_gastro_c_restaurant_stock
0,44	r.oa_gastro_c_fast_food_stock
-0,43	r.t_briefkasten_stock
-0,43	r.c_briefe_und_pakete_stock
0,41	r.oa_vergnuegung_t_spielothek_stock
-0,41	r.t_haltestelle_stock
-0,40	r.c_oepnv_stock
0,39	r.t_spielothek_stock
-0,37	r.oa_gastro_t_restaurant_deutsch_stock
0,35	r.c_zwielicht_stock
0,32	r.oa_total_d_vergnuegung_stock
0,31	r.oa_gastro_t_fastfood_kebab_stock
-0,30	r.t_florist_stock

Wie in Tabelle 6-11 zu sehen, gibt es eine Vielzahl an mittleren Korrelationen zwischen dem Bestand an lokaler Angebotsstruktur und dem sozialen Status. Am stärksten fällt auf, dass der OA-Wert für Restaurants eine vergleichsweise hohe Korrelation zu positivem sozialen Status hat. Die OA- Kennzahl für Fast Food ist

fast in gleicher Höhe mit dem negativen sozialen Status korreliert. Im Speziellen sind deutsche Restaurants positiv und Kebap-Shops negativ mit dem sozialen Status korreliert.

⁵² Die Korrelationen wurden nicht auf Signifikanz untersucht. Es werden nur Korrelationen über 0,3 ausgewertet. Die vollständige Liste der Korrelationen befindet sich unter https://github.com/dhelweg/masterthesis2018/blob/master/data/regressionen_bzr_indexwert.xlsx

Daneben hängen Spielotheken negativ, und ÖPNV-Angebote sowie Briefkästen positiv mit dem sozialen Status zusammen.

Für die Hypothesen H1a – H1c wurden zusätzlich die Korrelationen in den verschiedenen Raumbezugsgrößen untersucht.

6.3.1.1 H1a – Cafés & positiver sozialer Status

Es konnte eine sehr geringe negative Korrelation zwischen dem einwohnergewichteten Bestandswert von Cafés und dem Indexwert des sozialen Status nachgewiesen werden. Wie in Tabelle 6-12 zu sehen ist, besteht diese Korrelation jedoch nur auf Ebene des Planungsraums. Ein negativer Indexwert bedeutet eine geringere Arbeitslosigkeit, Kinderarmut, etc. Die Korrelation sagt also aus, dass bei vergleichsweise mehr Angebot an Cafés der soziale Status leicht besser ist. Da die Korrelation sehr gering ist, kann die Hypothese „*Umgebungen mit vielen Cafés haben einen vergleichsweise hohen sozialen Status.*“ weder gestützt noch verworfen werden.

Tabelle 6-12: H1a Korrelationen Cafés und MSS-Status-Index

Daten-satz	Kennzahl	Ergebnis Pearson		Ergebnis Spearman	
		p-value	cor	p-value	rho
bzr	c_cafe_stock	0,14	-0,13	0,33	-0,08
plr	c_cafe_stock	0,00	-0,14	0,25	-0,06
plr_dist-calc	c_cafe_stock	0,13	-0,07	0,13	0,07
bzr	oa_gastro_c_cafe_stock	0,87	0,01	0,72	-0,03
plr	oa_gastro_c_cafe_stock	0,70	0,02	0,61	-0,02
plr_dist-calc	oa_gastro_c_cafe_stock	0,24	0,06	0,06	0,09

6.3.1.2 H1b – Fast Food & negativer sozialer Status

Wie oben erwähnt, konnte zwischen dem OA-Wert von Fast-Food und dem Indexwert des sozialen Status eine mittlere positive Korrelation festgestellt werden. In Tabelle 6-13 ist zu sehen, dass die Korrelation für den OA-Wert sowohl über Pearson, als auch über Spearman messbar ist. Die Korrelation verstärkt sich, wenn die räumliche Betrachtung größer wird, oder Distanzgewichtungen hinzukommen. Es fällt auf, dass für den einwohnergewichteten Bestandswert - also ohne OA - nur eine geringe positive Korrelation messbar ist. Die Hypothese „*Umgebungen mit vielen Fast Food Geschäften haben einen vergleichsweise niedrigeren sozialen Status.*“ kann insgesamt jedoch bestätigt werden.

Tabelle 6-13: H1b Korrelationen Fast Food und MSS-Status-Index

Daten-satz	Kennzahl	Ergebnis Pearson		Ergebnis Spearman	
		p-value	cor	p-value	rho
bzr	c_fast_food_stock	0,57	0,05	0,06	0,16
plr	c_fast_food_stock	0,43	-0,04	0,04	0,10
plr_dist-calc	c_fast_food_stock	0,91	0,01	0,00	0,15
bzr	oa_gastro_c_fast_food_stock	0,00	0,44	0,00	0,49
plr	oa_gastro_c_fast_food_stock	0,00	0,23	0,00	0,27
plr_dist-calc	oa_gastro_c_fast_food_stock	0,00	0,29	0,00	0,34

6.3.1.3 H1c – Sportmöglichkeiten & positiver sozialer Status

Es konnte eine sehr geringe positive Korrelation zwischen dem OA-Wert von Sporteinrichtungen und dem Indexwert des sozialen Status nachgewiesen werden. Diese Korrelation besteht jedoch nur auf Ebene der Planungsräume und ist bei der Distanzgewichtung leicht höher. Auf Bezirksräumen ist die Korrelation nicht mehr vorhanden. Weiter ist in Tabelle 6-14 zu erkennen, dass es auf Ebene BZR eine entgegengesetzte geringe negative Korrelation nach Pearson gibt. Die Hypothese H1c „*Umgebungen mit vielen Sportmöglichkeiten haben einen vergleichsweise hohen sozialen Status*“ kann für Berlin damit insgesamt weder verworfen noch bekräftigt werden. Bei diesem Beispiel fällt ein starker Unterschied zwischen Spearman und Pearson auf. Dies könnte ein Anzeichen für Extremwerte sein.

Tabelle 6-14: H1c Korrelationen Sport und MSS-Status-Index

Daten-satz	Kennzahl	Ergebnis Pearson		Ergebnis Spearman	
		p-value	cor	p-value	rho
bzr	d_sport_und_erholung_stock	0,02	-0,20	0,20	-0,11
plr	d_sport_und_erholung_stock	0,15	-0,07	0,27	-0,05
plr_dist-calc	d_sport_und_erholung_stock	0,37	-0,04	0,80	0,01
bzr	oa_total_d_sport_und_erholung_stock	0,74	-0,03	0,27	0,09
plr	oa_total_d_sport_und_erholung_stock	0,56	0,03	0,03	0,10
plr_dist-calc	oa_total_d_sport_und_erholung_stock	0,90	-0,01	0,01	0,13

6.3.2 H2 und H3 – Hypothesen zur sozialen Dynamik

6.3.2.1 H2 – Einfluss aktueller Angebotsstruktur auf zukünftige soziale Dynamik

Für die Hypothese H2 wurde der Einfluss der „aktuellen lokalen Angebotsstrukturen [...] auf die zukünftige Veränderung des sozialen Status“ untersucht. Dabei ist das beste Modell eher mittelmäßig. Der RandomForest nutzt als wichtigste⁵³ Datenfeatures die Domänen *Tourismus, Mobilität, Vergnügen, Public Service (Bildung, Gesundheit, Sicherheit, Soziales)* und *Gastronomie*. Damit kann die Hypothese H2 nicht bewiesen werden, dennoch kann ein mittelmäßiges Modell mit inhaltlich validen genutzten Datenfeatures ein Indikator für ihre Richtigkeit sein.

Bei der Untersuchung der höchsten Korrelationen war nur eine Korrelation über 0,3 enthalten. Diese war mit genau -0,30 eine schwache negative Korrelation zwischen dem OA-Stock Wert von *Fast Food Geschäften* und der sozialen Dynamik. Dies kann jedoch daran liegen, dass die Orte mit einem eher schlechteren sozialen Status eine bessere Dynamik aufweisen. Und da der soziale Status eine mittlere Korrelation mit dem OA-Wert von Fast Food hat, scheint es sich hierbei um eine Scheinkorrelation zu handeln.

6.3.2.2 H3a – Änderung der Angebotsstruktur hat Einfluss auf soziale Dynamik

Die Hypothese H3 besteht aus mehreren Unterhypothesen, die inhaltlich gegenläufig aufgestellt sind. Die Unterhypothesen postulieren einen zeitlichen Kontext zwischen Änderung der lokalen Angebotsstruktur und der Dynamik des sozialen Status.

Dabei steht H3a für eine Änderung der lokalen Angebotsstruktur vor der Änderung des sozialen Status. Diese Hypothese kann mit den erhobenen Daten auf Ebene der Bezirksregionen nicht modelliert werden. Bei den Planungsräumen dagegen konnte ein mittelmäßiges Modell für H3a erzeugt werden. Dieses Modell wurde durch den LogitBoost Algorithmus mit Features wie *Parkautomaten, Reinigungsservices, Spielplätzen, Kleidungscontainern* und *Tischtennisplatten* aufgestellt. Es ist davon auszugehen, dass dieses mittelmäßige Modell keine inhaltliche Relevanz hat.

Auch in der Korrelationsanalyse ist kein Feature über 0,25 mit der Dynamik korreliert. Die höchsten Korrelationen haben *Zoofachgeschäfte* (0,24) und *Kioske* (-0,23). Das Entstehen von neuen Kiosken in Berlin könnte durchaus ein Indiz für eine eintretende Aufwertung sein, die Korrelation ist jedoch zu schwach, um die Hypothese zu stützen.

⁵³ Beim RandomForest können die genutzten Features mit mittlerem Informationsgewinn und Anzahl der Verwendungen im Forest ausgegeben werden. Wichtig = Informationsgewinn * Anzahl

6.3.2.3 H3b – Soziale Dynamik hat Einfluss auf Änderung der Angebotsstruktur

Bessere Modelle werden für die Hypothese H3b erstellt. Diese besagt, dass die Änderung sozialen Status der Änderung der lokalen Angebotsstruktur vorrausgeht. Sie ist also entgegengesetzt zur Hypothese H3a. Bei den Modellen stechen insbesondere die positiv heraus, die auf logistischer Regression basieren. Das beste Modell ist das Bagging-Ensemble, basierend auf SimpleLogistic. Die einzelnen Modelle verwenden sehr viele Datenfeatures, die mit dem höchsten Einfluss in den Modellen sind neue *Zoofachgeschäfte* (positiv korreliert mit Steigerung des sozialen Status), *Läden für Hörgeräte* (negativ), *Reinigungen* (negativ), *Secondhand Shops* (positiv) und *Kinos* (positiv). Aufgrund des Modells lässt sich eine leichte Tendenz zur Richtigkeit der Hypothese ableiten.

Tabelle 6-15: Korrelationen H3b - Soziale Dynamik hat Einfluss auf Änderung der Angebotsstruktur

cor	Feature
-0,37	r.t_bar_new
-0,36	r.c_gaststaetten_new
-0,34	r.d_vergnuegung_new
-0,32	r.t_fahrrad_new
0,31	r.t_haltestelle_new
-0,30	r.t_bar_ytd
-0,29	r.t_pub_new
0,29	r.c_oepnv_new
-0,27	r.c_sozial_new
-0,27	r.t_restaurant_international_ytd

In Kombination mit den schwachen Korrelationen aus Tabelle 6-15 wird diese Tendenz weiter verstärkt. So ist der Zuwachs an Pubs und Bars leicht positiv mit dem vorhergehenden Anstieg des sozialen Status korreliert. Ein zuvor angestiegener sozialer Status ist positiv mit dem Zuwachs von Fahrradhändlern und negativ mit dem Zuwachs an ÖPNV korreliert.

6.3.2.4 H3c – Gleichzeitige Änderung von sozialem Status und lokaler Angebotsstruktur

Die dritte Hypothese H3c besagt, dass die Änderungen der Angebotsstruktur und des sozialen Status gleichzeitig stattfinden. Für diese Hypothese sind eine Reihe mittelmäßiger Modelle entstanden. Das beste Modell ist das Voting-Ensemble, das aus vielen der anderen mittelmäßigen Klassifikatoren besteht. Es ist jedoch nur geringfügig besser als die verwendeten inneren Klassifikatoren. Das zweitbeste Modell wurde direkt mit Logit-Boost erstellt. Mit leichter Anpassung an das Modell auf ein Limit von zwei Iterationen konnte ein noch etwas besseres Modell erzeugt werden, mit einer F-Measure von 0,77 und einem AUC von 0,71. Das Modell basiert nur auf den zwei Features *neue Bar* und *neue asiatische Restaurants*. Bei den Korrelationen ist die Kennzahl für neue Bars jedoch nur mit -0,22 mit der gleichzeitigen Dynamik des sozialen Status korreliert. Asiatische

Restaurants sogar nur mit -0,03. Dieses Beispiel zeigt anschaulich, dass der LogitBoost-Algorithmus nicht mit den besten Korrelationen arbeitet, sondern mit dem jeweils besten Informationsgewinn.

Bei den übrigen Korrelationen hat nur der Zuwachs an ÖPNV mit 0,37 über einer Korrelation von 0,3.

Tabelle 6-16: Übersicht Bewertung der Hypothesen

Hypothese		Bestes Machine Learning Model				Korrelationsanalyse		Bewertung
ID	Bedeutung	Algo-rithmus	AUC	F1	Fea-tures sinn-voll?	Anzahl Features mit cor > 0,3	Fea-tures sinn-voll?	
H1	istOSM_stock => istStatus	Simple- Logistic _3	0,87	0,79	ja	8	ja	✓
H1a	-cafe positiv							0
H1b	-fastfood negativ							✓
H1c	-sport positiv							0
H2	prevOSM_stock => istDynamik	RandomForest	0,77	0,65	eher nein	1	ja	0
H3a	prevOSM_new => istDynamik	Logi- Boost	0,72	0,67	nein	0	-	X
H3b	istOSM_new => prevDynamik	Bagging _Simp- leLogis- tic	0,81	0,72	eher ja	3	ja	(✓)
H3c	istOSM_new => istDynamik	Logi- Boost _2	0,71	0,77	ja	1	nein	0

In Tabelle 6-16 sind noch einmal alle Ergebnisse der Evaluation zusammengefasst worden. Die Hypothesen H1 und H1b konnten nachgewiesen werden. H1a und H1c haben signifikante Korrelationen, die der Hypothese entsprechen aber sehr gering sind. Die Änderung des sozialen Status scheint vor der Änderung der Angebotsstruktur stattzufinden. H3b ist eher bestätigt, während wenig für die entgegengesetzte Hypothese H3a spricht. Bei H2 und H3c ergibt sich kein klares Bild.

7 Diskussion

In diesem Kapitel sollen das Vorgehen, die Ergebnisse, Limitierungen und Auswirkungen dieser Arbeit diskutiert werden. Ziel der Arbeit war es, zu beantworten, welche Daten für die Analyse von Gentrifizierung nutzbar sind, und wie diese integriert werden können. Weiter sollte betrachtet werden, welche generierten Datenfeatures und Algorithmen sich für den Aufbau eines Prognosemodells für Gentrifizierung eignen. Dazu wurde nach dem CRISP-DM Referenzmodell vorgegangen, das auch die Kapitelstruktur der Arbeit geprägt hat. Aus Sicht des Autors hat dieses Vorgehen zu einer systematischen Bearbeitung der Datenanalyse geführt. Insbesondere der Aufbau des Domänenverständnisses, sowie die Analyse der Datenquellen halfen dabei, die domänenspezifischen Hypothesen aufzustellen und diese zu untersuchen.

7.1 Data Understanding

Die genutzten Datenquellen beschränkten sich auf öffentliche Daten zu Sozialleistungsempfängern und Einwohnermelddaten. Diese wurden zur Bestimmung eines gentrifizierten Gebietes genutzt. Von der dreidimensionalen Gentrifizierungsdefinition, bestehend aus immobilienwirtschaftlicher Aufwertung, verdrängunginduzierter sozialer Aufwertung und dem Wandel der Angebotsstruktur, konnten damit nur Teilespekte untersucht werden. Da der MSS-Bericht den Blick nur auf die sozial Schwachen legt, ist eine weitere Differenzierung der sozialen Schichtung nicht möglich. So geben die Zahlen keinen Hinweis auf die Verdrängung von Familien aus der Mittelschicht in gentrifizierten Nachbarschaften. Es ist jedoch davon auszugehen, dass insbesondere junge Familien mit dem ersten Kind eine Nachbarschaft verlassen müssen, da sie sich eine größere Wohnung dort nicht leisten können.

Aus dem Bereich Big Data beschränkt sich die Arbeit auf Daten von OpenStreetMap, mit denen der Wandel der Angebotsstruktur abgebildet wurde. Unter den Gesichtspunkten der 4Vs hat OpenStreetMap als Ganzes (planet.OSM) ein hohes Datenvolumen (volume). Der Filter auf spezielle POI-Nodes in Berlin sorgt jedoch für eine deutliche Verkleinerung der Datenmenge. Die Geschwindigkeit (velocity), mit der sich die Daten ändern, ist relativ gering. Obgleich täglich neue Versionen der Daten verfügbar sind, würde für den Anwendungsfall der POI-Erhebung ein wöchentlicher oder monatlicher Abzug der Daten ausreichen. Die Datenvielfalt (variety) ist dagegen hoch. So ist zwar eine Struktur der Daten vorgegeben, über die Key-Value Tags ist die tatsächliche Ausgestaltung der Daten jedoch trotz Community-Standards relativ heterogen. Durch die Community ist auch eine

Unsicherheit in der Datenqualität (veracity) gegeben. So ist nicht klar, ob alle erfassten POIs existieren, oder ob die Daten noch aktuell sind. Die Annahme, dass neu erfasste POIs auch automatisch neu eröffneten Geschäften entsprechen, ist eine weitere Limitierung der Arbeit.

Die Anbindung weiterer Datenquellen würde wahrscheinlich weitere Erkenntnisse eröffnen. Laut einer Umfrage in einem Forbes Artikel wird für das Suchen und Aufbereiten von Datenquellen ca. 80% der Arbeitszeit von Data Scientists verwendet (Press, 2016). Auch beim Erstellen dieser Arbeit wurde für die Phasen Data Understanding und Data Preparation deutlich mehr Zeit benötigt als für die Modellierung. Aus Sicht des Autors wäre insbesondere die Anbindung von Datenquellen zur immobilienwirtschaftlichen Aufwertung ein sinnvoller nächster Schritt. Dies könnten zum Beispiel auch Daten über die Verteilung von Angeboten auf Airbnb sein (airbnbvsberlin.de, 2016). Eine Untersuchung im Rahmen dieser Arbeit wäre jedoch zu zeitaufwendig gewesen.

7.2 Data Preparation

Abbildung 7-1 stellt den Zeitverlauf der OSM Daten dar. In der Analyse fällt auf, dass der Anteil der POIs, die als geändert erkannt werden, relativ gering ist. Der Anteil der geänderten POIs an der Kennzahl *_new* beträgt durchschnittlich etwas über sieben Prozent. Allerdings stammt ein Großteil dieser Änderungen aus den POI-Kategorien Restaurants, Fast-Food und Einkaufsmöglichkeiten. Diese haben sich in den Modellen als gute Datenfeatures mit hoher Korrelation erwiesen. Der Autor geht davon aus, dass der Anteil der geänderten POIs ohne den Algorithmus zur Erkennung von signifikanten Änderungen deutlich höher gewesen wäre.

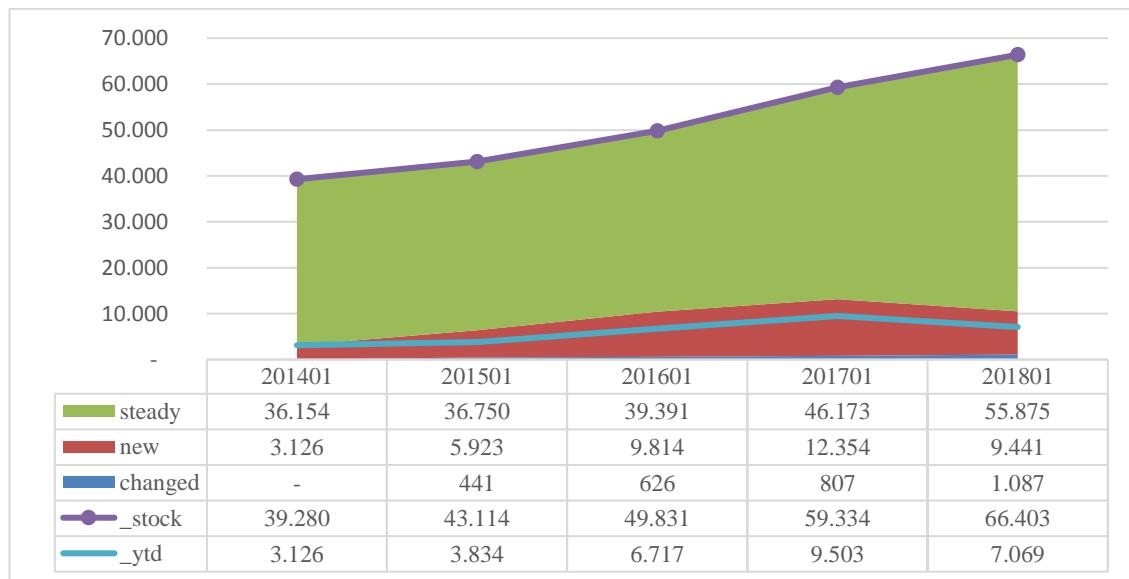


Abbildung 7-1: Zeitverlauf der POI-Kennzahlen in Berlin

Die Verwendung der Offering Advantage, welche die charakteristische Eigenschaft einer Nachbarschaft bzgl. eines POI-Typens hervorhebt, hat die Modelle deutlich verbessert. Dieser Schritt hat die beidseitige Verbindung zwischen den CRISP-DM Phasen Modeling und Data Preparation für den Autoren spürbar gemacht. In einer weiterführenden Arbeit sollten auch die Kennzahlen zur Veränderung der Angebotsstruktur per OA berechnet werden.

In der Auswertung der Modelle hat sich gezeigt, dass die Distanzgewichtung deutlich bessere Ergebnisse geliefert hat, als die reine Planungsraumbetrachtung. Und dies, obwohl die Formel der Distanzgewichtung aktiv eine räumliche Autokorrelation herbeigeführt hat, indem POIs in umliegenden PLRs die Datenfeatures des betrachteten Gebietes erhöhen. Es könnte möglich sein, dass die soziokulturellen Effekte, die der Gentrifizierung zugrunde liegen, ebenfalls über die Raumgrenzen der Planungsräume hinweggehen und auch räumlich autokorreliert sind. Die Auswertungen der Bezirksregionen waren noch genauer als die der distanzgewichteten Planungsräume.

Beim Vergleich von dem berechneten Index nach Döring und Ulbricht (2016) mit der Klassifizierung der MSS-Dynamik ergeben sich deutliche Unterschiede, wie auch in den Kartendarstellungen der Klassifizierungen in Abbildung 9-37 und Abbildung 9-36 zu sehen ist. Ein Grund hierfür ist die Kombination der Mobilitätsraten und der Änderung der Bevölkerungsstruktur, die in den berechneten Index eingeht. Der MSS-Index zur Dynamik fokussiert sich dagegen auf die Änderung des Anteils der Leistungsempfänger. Die Zahl der Ausländer und Migranten, die im Zuge der Flüchtlingskrise zugezogen sind, hatte bei einigen Planungsräumen einen hohen Einfluss auf die Dynamik-Kennzahl nach Döring und Ulbricht (2016). So sind hohe Änderungen des Ausländer- und Migrantanteils in einigen Räumen zu beobachten, was starke Auswirkungen auf beide Teilwerte hatte. Auf der einen Seite ist der Anteil der Migranten und Ausländer gestiegen, was in den Wert der Änderung der Bevölkerungsstruktur einging. Auf der anderen Seite hat die erhöhte Anzahl der Gesamtbevölkerung den einwohnergewichteten Anteil der Bevölkerung mit einer Wohnzeit von mehr als fünf bzw. zehn Jahren beeinflusst. Diese Werte gingen in den Teilwert der Mobilität ein. Zum Beispiel ist im Planungsraum *Freiheit* (05010339) im Jahr 2015 eine Flüchtlingsunterkunft für 500 Personen entstanden ([tagespiegel.de](#), 2015). Damit hat sich die Einwohnerzahl in dem wenig bewohnten Gewerbegebiet von 448 in 2014 auf 974 im Jahr 2016 verdoppelt, wodurch es zu Ausreißern gekommen ist. Im gesamten Stadtbereich wurden Notunterkünfte bereitgestellt, was die Einwohnerzahlen erhöht hat, aber keinen Zusammenhang mit der Gentrifizierung hatte. Durch solche Verzerrungen ist in diesem Zeitraum der Index nach Döring und Ulbricht

ein schwieriger Messwert für Gentrifizierung. Laut Bostic und Martin (2003, S. 2431) ist eine Kombination vieler Effekte in einem Index für die Erkennung von Gentrifizierung zum Scheitern verurteilt. Das hat sich auch in dieser Arbeit gezeigt.

7.3 Modeling

Auf Basis der aufbereiteten Daten wurden deskriptive Korrelationsanalysen und prädiktive Klassifizierungsmodelle erstellt. Dabei konnte für den Zusammenhang zwischen sozialem Status und lokaler Angebotsstruktur ein präziseres Klassifizierungsmodell erstellt werden als bei Venerandi et al. (2015). Während sie in ihrer Studie einen gemittelten F1-Wert von 0,74 für ein Klassifizierungsmodell mit NaiveBayes (Abbildung 3-4) erreicht haben, konnte in dieser Arbeit ein F1-Wert von 0,79 bei einem AUC von 0,89 mit einem logistischen Regressionsmodell erzeugt werden. Allerdings sind die Datengrundlagen beider Arbeiten verschieden. Während die Analyse von Venerandi et al. in London primär auf Foursquare basiert, nutzt diese Arbeit OSM in Berlin. Die Autoren der Londoner Studie gaben an, dass in OSM nur wenige Daten zu POIs vorhanden waren. Im Betrachtungszeitraum dieser Arbeit in Berlin ist dieses Problem nicht aufgefallen. Vielmehr konnten basierend auf den Bestandsdaten sehr gute Modelle erzeugt werden. Abbildung 7-1 zeigt jedoch, dass laufend neue POIs erfasst werden. Für diese wurde vereinfacht angenommen, dass sie neu eröffnet wurden. Es ist jedoch davon auszugehen, dass die Daten in OSM nicht alle tatsächlichen POIs beinhalten, und die enthaltenen teilweise veraltet oder inkorrekt sind. Gleiches ist jedoch auch bei Foursquare und anderen Plattformen nicht auszuschließen. Auch der IMD-Wert, der bei Venerandi et al. als Zielwert für die Klassifizierung genutzt wurde, unterscheidet sich vom MSS-Status. Dennoch zeigen beide Arbeiten, dass es insgesamt einen starken Zusammenhang zwischen Angebotsstruktur und sozialem Status gibt. Im Detail jedoch sind die enthaltenen Korrelationen der POI-Typen sehr unterschiedlich, was für eine räumliche Heterogenität⁵⁴ zwischen London und Berlin spricht. Zum Beispiel haben in Berlin deutsche Restaurants einen positiven und Kebap Fast-Food Shops einen negativen Zusammenhang mit dem sozialen Status. In London dagegen haben afrikanische Restaurants einen positiven und italienische Restaurants einen negativen Zusammenhang mit dem Verdrängungsindikator IMD. So stellen sich unterschiedliche Räume unterschiedlich dar.

Der NaiveBayes-Algorithmus lieferte in dieser Arbeit die schlechtesten Modelle, wohingegen die besten Modelle durch Ensembles und logistische Regression entstanden. Da

⁵⁴ Vgl. Kapitel 2.2

NaiveBayes für jedes Datenfeature eine Wahrscheinlichkeit für eine Klasse bestimmt, nimmt die Prognosegüte bei zunehmender Anzahl an Attributen ab, auch wenn diese einen Informationsgewinn liefern. Dagegen profitieren laut Jakulin (2005, S. 150) insbesondere Modelle mit logistischer Regression monoton von neuen Features, da Features ohne Informationszuwachs vom Modell nicht verwendet werden. Dieser Effekt zeigte sich auch in den Modellen dieser Arbeit.

7.4 Evaluation – Rückschlüsse auf Gentrifizierung und Business Understanding

7.4.1 Gentrifizierung

Mit Hilfe der Modelle und Korrelationen wurden bestehende Hypothesen aus anderen Forschungsarbeiten überprüft. Es konnte ein Zusammenhang zwischen sozialem Status und der lokalen Angebotsstruktur in Berlin nachgewiesen werden. Dabei sind insbesondere Restaurants positiv und Fast-Food Shops negativ mit dem sozialen Status korreliert. Aber auch andere POI-Typen wie Spielotheken haben einen (in diesem Fall negativen) Zusammenhang mit dem sozialen Status.

In den Daten zu einigen Hypothesen gab es Korrelationen zwischen dem Indexwert und POIs der Kategorie ÖPNV, wie zum Beispiel Haltestellen. Im sozialen Status hatte dieser POI-Typ eine mittlere positive Korrelation zu einem hohen sozialen Status. Dieser Effekt könnte entweder einen tatsächlichen Zusammenhang aufzeigen, oder aber etwas mit dem hohen Anstieg des OM-Keys „public_transport“ in den Berliner OSM-Daten zu tun haben.⁵⁵ Die Anzahl der Nodes mit diesem Key ist von ca. 2.500 zu Beginn des Jahres 2015 auf ca. 8.500 zu Beginn des Jahres 2017 gestiegen. Somit liegt diese massive Erfassung im Zeitfenster der Auswertung. Der Autor geht davon aus, dass die mittlere Korrelation der Bestandsdaten (also nicht der Änderungswerte) mit dem sozialen Status daher röhrt, dass die Mapper der OSM-Community sozial eher besser gestellt sind und eher selten Sozialleistungsempfänger sind.

Des Weiteren sind Anzeichen für einen zeitlichen Bezug zwischen der Änderung des sozialen Status und der Änderung der Angebotsstruktur sichtbar geworden. So gibt es durch die Modelle und Korrelationen Indizien dafür, dass in Berlin der Wandel der Bevölkerung vor dem Wandel der Angebotsstruktur erfolgt ist. In Bezug auf doppelten Invasions-Sukzessions-Zyklus nach Dangschat (1988) kann das dafür sprechen, dass hier die frühen Phasen des Gentrifizierungsprozesses erkannt wurden. Der Dynamik-Index bezieht sich

⁵⁵ Vgl. Abbildung 4-6: Entwicklung der Anzahl der OSM-POI in Berlin nach Tag-Key

nur auf die Leistungsempfänger, die nach dem Modell von Dangshat in den frühen Phasen stark von den Pionieren verdrängt werden. Laut Kecske (1994, S. 28) ändern erst die Pioniere die lokale Angebotsstruktur, die dann wiederum die Gentrifizierer anziehen. Nach Davidson und Lees (2005, S. 1183) würden diese zu 83% lokale Restaurants in der Nachbarschaft nutzen. Unter den Korrelationen von H3b befanden sich die POI-Typen Bars, Pubs und internationale Restaurants. Diese POI-Typen sind in der Untersuchung alle leicht positiv mit einer positiven Dynamik korreliert.

Mit diesem Wissen könnte die negative MSS-Dynamik in der Mitte von Berlin (Abbildung 9-36) ggf. ein Anzeichen für die zweite Phase der Gentrifizierung sein, in der die Pioniere von den Gentrifizierern verdrängt werden. Die Dynamik gibt schließlich nur Auskunft darüber, ob ein Gebiet eine bessere oder schlechtere Veränderung des Anteils der Leistungsempfänger hat, als der Berliner Durchschnitt. Wenn ein Großteil der Leistungsempfänger bereits verdrängt wurde, kann sich in gentrifizierten Gebieten ihr Anteil an den Einwohnern eines Raumes nur geringfügig ändern. Eine geringfügige Änderung ist jedoch schlechter als ein Trend zur Verringerung des Anteils der Sozialleistungsempfänger in Berlin insgesamt. Damit hätten gentrifizierte Gebiete in Berlin mit einem hohen sozialen Status eine leicht negative Dynamik, wenn Berlin insgesamt einen Trend zu sinkendem Anteil an Sozialleistungsempfängern an der Bevölkerung hat.

Die laut Holm (2014, S. 277) mangelnden Möglichkeiten zur Erkennung von Gentrifizierung in einem solchen gesamtstädtischen Aufwärtstrend könnten somit in einem anderen Licht betrachtet werden. Die Änderung der lokalen Angebotsstruktur, die einer vorherigen positiven Dynamik folgt, könnte ein Indikator für eine aktuelle oder bevorstehende Invasionsphase der Gentrifizierer nach Dangshat sein. Dies sollte in weiteren Studien qualitativ untersucht werden.

7.4.2 Weitere Anwendungsfälle

Für weitere Anwendungsfälle, wie die Standortplanung, können die aufbereiteten Daten Auskunft über die Angebotsstruktur und deren Entwicklung in einem Gebiet liefern. Je nach Geschäftsstrategie kann es nützlich sein, in der Nähe von Konkurrenz ein Geschäft zu eröffnen; zum Beispiel im Bereich Kleidung und Mode. Für andere Unternehmen könnte eine möglichst attraktive Umgebung ein Ziel sein, um die Mitarbeiter mit einem ansprechenden Standort an das Unternehmen zu binden. Bei Themen der Filialnetzplanung können dagegen wiederum andere Aspekte von Belang sein. So könnten zum Beispiel die Standorte von Geldautomaten anhand von umliegenden Geschäften optimiert werden.

Für die Stadtentwicklung in Berlin ist es anhand der erhobenen Daten möglich, ein Bild der Angebotsstruktur in verschiedenen räumlichen Auflösungen zu erlangen. Insbesondere durch die Kombination der OSM-Daten mit der Offering Advantage können die charakteristischen Merkmale eines Raumes aufgezeigt werden. Bezogen auf die polyzentrale Stadtentwicklungsstrategie von Berlin ist damit eine Untersuchung des Status und der Entwicklung der verschiedenen Zentren möglich.

Bei der Suche nach einem geeigneten Wohnort können die erhobenen Daten dabei helfen, einen Wohnort zu finden, der den individuellen Vorlieben entspricht. Diese Arbeit bietet einen Einblick in die Möglichkeiten der Datenauswertung mit OSM. Damit kann sie auch für das WEKOVI-Projekt des BMVI von Interesse sein, da dieses einen Fokus auf offene Geodaten hat. Der Autor kann sich darüber hinaus vorstellen, dass zum Beispiel Makler, Banken oder Bausparkassen einen Online-Service einrichten, mit dem Kunden Miet- und Kauf-Angebote entsprechend ihrer Präferenzen an ihre Umgebung finden können. Bisher muss der Kunde sich selbst darüber informieren, welche Stadtteile attraktiv für ihn sind. Das ist unter anderem für neu hinzuziehende Personen teils schwer einzuschätzen. Fraglich ist jedoch, ob sich die Entwicklung eines solchen Service lohnen würde. Aufgrund des knappen Wohnungsmarktes in Ballungszentren finden viele Objekte schnell einen Abnehmer. Gegebenenfalls könnten so jedoch Stadtteile außerhalb der allgemein bekannten Trendgebiete an Attraktivität gewinnen.

8 Fazit

In dieser Arbeit wurde die Gentrifizierung in Berlin mittels Big Data Analytics untersucht. Für diese Untersuchung wurde methodisch nach dem CRISP-DM Referenzmodell vorgegangen, welches sich auch in den Kapitelüberschriften wiederspiegelt.

In Kapitel 2 wurden Grundlagen zu Big Data und Big Data mit Raumbezug beschrieben, sowie (teils betriebswirtschaftliche) Anwendungsfälle zu räumlichen Big Data Analysen aufgezeigt.

Anschließend wurde in Kapitel 3 – Business Understanding – das benötigte Domänenwissen zur Gentrifizierung geschaffen. Dabei wurde auch der Stand der Forschung zur Untersuchung von Gentrifizierung mit Big Data erörtert, der von Zook et al. (2017) erarbeitet wurde. Insbesondere die Arbeit von Venerandi et al. (2015) zur Untersuchung von Abhängigkeiten zwischen sozialer Verdrängung und Foursquare-POIs in London, sowie die Arbeiten von Döring und Ulbricht (2016) und Holm und Schulz (2016) zur Messung von Gentrifizierung in Berlin, haben diese Arbeit geprägt.

In Kapitel 4 – Data Understanding – wurden diverse Datenquellen untersucht, mit denen Gentrifizierung in seiner Multidimensionalität aus immobilienwirtschaftlicher Aufwertung, verdrängungsinduzierter sozialer Aufwertung und der Veränderung der lokalen Angebotsstrukturen analysiert werden kann. Zudem wurden verschiedene Ansätze zur technischen Darstellung von Adressen und Räumen diskutiert. Für die Untersuchung in Berlin wurde die LOR-Hierarchie verwendet. Da viele kommerzielle Datenquellen keinen Zeitbezug hatten, und die Möglichkeiten zur systematischen Datenerfassung teils technologisch, teils lizenzerrechtlich beschränkt waren, hat sich der Autor auf die POI-Daten aus OpenStreetMap fokussiert. Auf die Integration immobilienwirtschaftlicher Daten wurde verzichtet. Stattdessen hat sich die Arbeit auf den Zusammenhang zwischen dem Wandel des sozialen Status und der lokalen Angebotsstruktur fokussiert. Hierzu wurden am Ende des Kapitels mehrere domänen spezifische Hypothesen aufgestellt, die in den darauf folgenden Kapiteln untersucht wurden.

In Kapitel 5 – Data Preparation – wurde aufgezeigt, wie die Daten technisch integriert werden können. Dazu wurde mit einem auf dem Hadoop Ökosystem basierenden Big Data System gearbeitet. Der dazu benötigte Programmcode, sowie die erzeugten Datensätze sind öffentlich zugänglich⁵⁶ und können als Grundlagen für weitere Analysen verwendet werden.

⁵⁶ Siehe Kapitel 1.3

Aufbauend auf den Datensätzen wurden in Kapitel 6 – Modeling & Evaluation – Machine Learning Algorithmen trainiert. Hierzu wurden das Data Mining Programm Weka und die statistische Programmiersprache R genutzt. Für die Modellierung wurde in einem zweistufigen Modellierungs-Evaluations-Zyklus zunächst die beste Datenkombination für die Modellierung evaluiert, und anschließend mit einer Vielzahl von Algorithmen versucht, Modelle zu den domänen spezifischen Hypothesen zu entwickeln. Dabei haben sich insbesondere Klassifizierungsmodelle mit Ensembles und logistischer Regression hervorgetan.

In Kapitel 7 wurden abschließend das Vorgehen, die Ergebnisse, Limitierungen und Auswirkungen dieser Arbeit diskutiert. Dabei wurde auch die Verwendung der Daten für andere Anwendungsfälle diskutiert, darunter die betriebliche Standortsplanung und die Idee eines Systems zur Suche eines geeigneten Wohnorts nach individuellen Anforderungen an die Nachbarschaft.

Im Ergebnis der Untersuchung konnte ein starker Zusammenhang zwischen der lokalen Angebotsstruktur und dem sozialen Status auf der Ebene von 137 Bezirksregionen nachgewiesen werden. Die Daten zur lokalen Angebotsstruktur basieren auf POI-Daten des Kartendienstes OpenStreetMap, die in einer Hierarchie aus Typen, Kategorien und Domänen zusammengefasst wurden. Der soziale Status basiert auf dem „Bericht Monitoring Soziale Stadtentwicklung Berlin“, bei dem Zahlen zu Sozialleistungsempfängern vergleichend ausgewertet werden (Senatsverwaltung für Stadtentwicklung und Wohnen, 2017). Es wurde basierend auf den Daten zum Ende des Jahres 2016 festgestellt, dass insbesondere Restaurants positiv mit dem sozialen Status korreliert sind, während Fast-Food Shops negativ korrelieren. Im Speziellen sind deutsche Restaurants mit einer positiven und Kebap Fast-Food Shops mit einer negativen Korrelation aufgefallen.

Auf der Basis von Machine Learning Algorithmen konnten Modelle trainiert werden, welche Indizien dazu liefern, dass die Änderung der lokalen Angebotsstruktur der Änderung des sozialen Status zeitlich folgt. Es wurde diskutiert, dass dies ein Anzeichen für die erste Phase des doppelten Invasions-Sukzessions-Zyklus nach Dangschat (1988) darstellen könnte. In dieser Phase wird eine sozial schwache Bevölkerung von Pionieren (u.a. Studenten und Künstler) verdrängt, welche laut Kecske (1994, S. 28) die Kultur und die Angebotsstruktur einer Nachbarschaft ändern. Darauf folgt in der Theorie dann die zweite Phase der Gentrifizierung, in der wiederum die Pioniere von einer einkommensstarken Bevölkerung verdrängt werden. Diese Arbeit hat Indizien für einen Zusammenhang zwischen der positiven sozialen Dynamik in den Jahren 2012 – 2014 und der Änderung der

Angebotsstruktur von 2014 – 2016 in Berlin aufzeigen können. Die Änderung der lokalen Angebotsstruktur, die einer vorherigen positiven Dynamik folgt, könnte ein Indikator für eine aktuelle oder bevorstehende Invasionsphase der Gentrifizierer nach Dangchat sein. Dies könnte in weiteren Studien qualitativ untersucht werden.

Weiter sollte nach Datenquellen zur Messung der Verdrängung der Mittelschicht gesucht werden. Diese Effekte sind in den MSS-Daten nicht enthalten, da dort lediglich Daten zu Sozialleistungsempfängern erfasst werden.

Die Ergebnisse dieser Arbeit unterliegen der weiteren Limitierung, dass OpenStreetMap im Vergleich zu anderen Portalen weniger erfasste POIs und eine teils heterogene Datenerfassungsqualität hat. Eine Annahme bei der Messung des Wandels der lokalen Angebotsstruktur war, dass neu erfasste POIs in OSM neu eröffneten Geschäften etc. entsprechen. Der konstante Anstieg der POIs spricht jedoch dafür, dass hier Ungenauigkeiten enthalten sein könnten. Aufbauend auf diese Arbeit könnten weitere POI-Datenquellen untersucht werden, um die Ergebnisse zu überprüfen.

Weitere quantitative, Big Data gestützte Untersuchungen der Gentrifizierung könnten andere inhaltliche und zeitliche Zusammenhänge der verschiedenen Dimensionen untersuchen. Holm und Schulz (2016, S. 300) zufolge tritt die immobilienwirtschaftliche und die verdrängungsinduzierte soziale Aufwertung zur gleichen Zeit auf. Dies könnte mit Daten über Immobilien- und Mietpreise, sowie Airbnb Angeboten überprüft werden. Auch könnten inhaltliche und zeitliche Zusammenhänge zwischen der immobilienwirtschaftlichen Aufwertung und der lokalen Angebotsstruktur, sowie deren Wandel, untersucht werden.

Sowohl für Politik und Stadtentwicklung, als auch für Investoren sind Prognosemodelle zur Gentrifizierung von Interesse. Während die Politik mit Gesetzen wie der Mietpreisbremse versucht die Gentrifizierung zu beschränken, werden Investoren mit günstigen Krediten weiter in Immobilien investieren. Dabei ist es ein Wettbewerbsvorteil einen Gentrifizierungstrend, also einen Aufwertungstrend, frühzeitig zu erkennen.

Sollte ein Modell gefunden werden, mit dem sich die Gentrifizierung präzise voraussagen lässt, wäre es fraglich, ob das Modell dabei hilft die Verdrängung zu beschränken, oder ob es die Immobilien- und Mietpreisseigerungen noch weiter beschleunigt.

9 Anhang

9.1 Anwendungsfälle

9.1.1 Epidemiologie und Versorgung

Das Forschungsgebiet der Epidemiologie, das sich mit den Ursachen, der Verbreitung und den Folgen von Krankheiten in einer Population beschäftigt, ist erst durch die Auswertung von raumbezogenen Daten entstanden. So fand John Snow im 19. Jahrhundert in London bei einem Cholera-Ausbruch die Infektionsquelle durch die Kartierung⁵⁷ von Krankheitsfällen und Brunnen. Auf diese Weise gelang es ihm, den von Cholera befallenen Brunnen zu identifizieren (Frerichs, 2016; Wikipedia, 2018b).



Abbildung 9-1: Karte des Cholera-Ausbruchs in Soho, London (Wikipedia, 2018b)

Bezogen auf Big Data hatte Google mit Google Flu Trends versucht Grippewellen über Suchanfragen in geographischen Gebieten vorherzusagen. Einem Beitrag im Science Ma-

⁵⁷ Siehe Abbildung 9-1: Karte des Cholera-Ausbruchs in Soho, London (Wikipedia, 2018b)

gazine von Lazer, Kennedy, King und Vespignani (2014) zufolge waren die Google Mitarbeiter damit jedoch nur bedingt erfolgreich. So wurden auf der einen Seite saisonale Grippewellen über lange Zeit überschätzt, auf der anderen Seite wurde eine nicht-saisonale Grippe übersehen. Das Projekt wurde von Google eingestellt (Google, 2016).

Aktuell gibt wird in diesem Anwendungsgebiet weitere Forschungsaktivitäten. So haben Kraemer et al. (2018) mittels georeferenzierter Tweets die Verbreitung von Dengue-Fieber in Pakistan untersucht. Eine andere Arbeit von Rubrichi, Smoreda und Musolesi (2018) befasst sich mit der Modellierung von Epidemien auf Basis von Bewegungsprofilen, die auf Daten von Mobilfunkzellen basieren. Die Erkenntnisse sind für Gesundheitsministerien, der Weltgesundheitsorganisation oder auch Versicherungen relevant.

Verwandt hiermit ist auch eine Studie des Bundesministeriums für Bildung und Forschung, welche mit dem SMART-MOVE Projekt die Wasserversorgung im Nahen Osten verbessern will. Als ein Teil des Projektes hat das Karlsruher Institut für Technik mit Projektpartnern eine Korrelation zwischen Niederschlagsereignissen und einer Trübung des Wassers feststellen konnten. Mit der Trübung erhöhte sich auch die Leitfähigkeit des Wassers. Darauf folgte zeitlich eine erhöhte Belastung des Wassers mit E.Coli-Bakterien. Auf Basis dieser Untersuchung wurde im Anschluss ein Frühwarnsystem entwickelt, welches Jordanien dabei hilft mit den knappen Wasserressourcen besser haushalten zu können (Disy Informationssysteme GmbH, o.D.b).

9.1.2 Katastrophenmanagement und Versicherungen

Neben Epidemien gibt es viele weitere Katastrophen, die sich geografisch analysieren lassen. Darunter sind nach Jiang und Shekhar (2017, S. 7) zum Beispiel Fluten, Waldbrände, Erdbeben und Erdrutsche. Damit wären zum Beispiel Frühwarnsysteme, die Waldbrände mittels Satellitenaufnahmen und Wetterdaten vorhersagen, ein möglicher Anwendungsfall von Spatial Big Data Analytics. Solche Informationen wären für Katastrophenschutz-Behörden, wie dem deutschen Bundesamt für Zivilschutz, von Belang.

Doch auch für Rückversicherungen sind solche Analysen unerlässlich für das Geschäftsmodell. So decken Rückversicherungen oftmals Großschadensereignisse ab (Swiss Re, 2013, S. 9). Da Versicherungsobjekte wie Immobilien einen geographischen Bezug haben, können bestimmte Schadenswahrscheinlichkeiten wie Flutgefahren mit Spatial Big Data besser berechnet werden (Joseph, 2014).

Allerdings ist neben der Risikokalkulation und Preisgestaltung auch eine möglichst frühe Schadensprognose für Versicherer relevant, denn diese müssen im Schadensfall ausreichend Liquidität haben um die Schäden zu regulieren. Auf der anderen Seite ist es für die

Versicherer in einem normalen Zinsumfeld wichtig, dass möglichst viel Kapital zinstragend angelegt ist, und nicht unnötig viel Kapitel als liquide Mittel in der Kasse zu haben (Swiss Re, 2013, S. 37).

9.1.3 Smart City

Auch im Bereich Smart City gibt es eine Vielzahl an Anwendungsfällen für Spatial Big Data Analytics. Darunter die öffentliche Sicherheit, die Straßenerhaltung sowie das Parkplatzmanagement.

Öffentliche Sicherheit

Für die öffentliche Sicherheit sind Spatial Big Data Analysen relevant, da auf Basis von historischen Verbrechens- oder Unfallmeldungen Hotspots im Straßenverkehr oder der Kriminalität festgestellt werden können (Jiang & Shekhar, 2017, S. 7). So setzt die Polizei in Manchester auf intelligente Streifenfahrten auf Basis von historischen Daten und konnte damit unter anderem die Zahl der Einbrüche um 21% reduzieren (Clark, 2017).

Erkennung von Straßenschäden

In Boston wurde 2012 ein Projekt namens „Street Bump“ ins Leben gerufen, mit dem Straßenschäden per Smartphone erkannt werden sollen. Dabei konnten Einwohner eine App installieren, die bei Aktivierung Erschütterungen über den Beschleunigungssensor des Smartphones misst und diese mit GPS-Position an den Street Bump Server schickt. Bei mehr als drei Anschlägen an der gleichen Stelle wurde die Stelle auf die Wartungsliste aufgenommen. In der Analyse ist aufgefallen, dass die meisten Schlaglöcher abgesunkene Kanaldeckel waren. Mittels der App wurden 1250 Kanaldeckel identifiziert und repariert. Die false positive Rate lag dabei bei unter 10% (boston.gov, 2017; streetbump.org, o.D.). Auf diese Weise kann Straßeninstandhaltung optimiert werden, indem die Arbeit von Straßenbauämtern, die sonst die Erkennung manuell in Streifenfahrten vornehmen müssten, zum Teil durch eine solche Lösung unterstützt wird.

Parkplatzmanagement

Das Berliner Startup Parkling nutzt Spatial Big Data Analysen um frei Parkplätze auf den Straßen vorherzusagen. Dabei werden freie Parkplätze in Straßen einer Stadt per Lasermessung identifiziert. Dieser Vorgang wird durch fahrende Messautos mehrfach vorgenommen. Aus den gewonnenen georeferenzierten Daten werden per Spatial Big Data Analysen Muster erkannt. Der Service ist bisher in Berlin in einer Pilotphase und im Appstore frei verfügbar. In Abbildung 9-2 ist der Raum rund um den Alexanderplatz an einem Sonntagnachmittag zu sehen. Die App zeigt eine Heatmap, je grüner die Straßen desto

höher die Wahrscheinlichkeit einen freien Parkplatz zu finden (digital kompakt, 2018). Nach Berlin wird das System als nächstes in Stockholm ausgerollt (Parkling, o.D.).

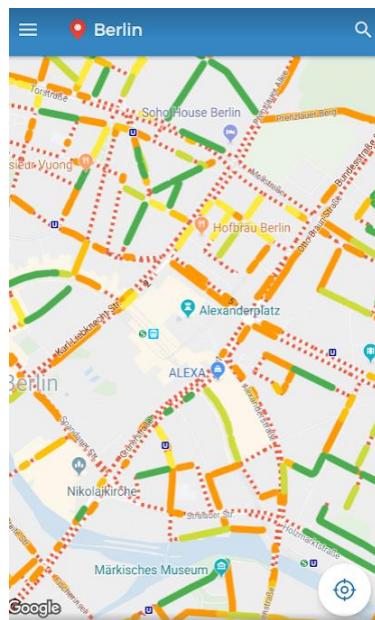


Abbildung 9-2: Screenshot Parkling App (Parkling, o.D.)

9.1.4 Logistik- und Navigationsdienstleister

Logistiker wie Hermes haben täglich viele Zusteller auf den Straßen. Jeder Zusteller hat eine Menge an Paketen, die er ausliefern muss. Die Zusteller selbst sind wiederum auf den öffentlichen Straßen unterwegs und somit wie andere Autofahrer auch von Staus, Baustellen und Straßensperrungen betroffen. Hermes investiert im aktuellen Jahr 2018 nach eigener Aussage mehrere Millionen Euro in eine neue Software für eine intelligente Tourenplanung und Navigation. Dabei setzt Hermes auf die Navigationssoftware NUNAV Courier, welche von dem Hannoveraner Startup Graphmasters entwickelt wird. Die Anwendung integriert dabei neben klassischen Routing weitere Parameter wie kundenindividuelle Zustellzeitfenster, Öffnungszeiten, Pausen- und Arbeitszeiten und optimiert daraufhin die Routen (Hermes Germany, 2018).

UPS hat 2013 ein ähnliches Projekt gestartet, bei dem mit Spatial Big Data Analysen Routen während der Tour laufend optimiert werden. Eine gesparte Meile pro Tag und Fahrer würden für UPS die Benzinkosten um ca. 50 Millionen US-Dollar reduzieren (datafloq.com, 2014). Ein Ergebnis der eine Milliarde US-Dollar schweren Investition in das ORION genannte Projekt (datafloq.com, 2014) ist eine jährliche Ersparnis von 300 bis 400 Millionen US-Dollar. Eine Erkenntnis der Analysen war es, dass Linksabbiegen teuer ist und eine Gefahrenquelle für Unfälle ist. Deshalb wurde die Navigationslogik so angepasst, dass Linksabbiegen vermieden werden soll (CNN, 2017). Die Analyse basierte auf

250 Millionen Datenpunkten von Sensoren, die Werte wie GPS-Position, Geschwindigkeit, Anzahl der Stopps und Verbrauch der Fahrzeuge gemessen haben. So kamen 16 Petabytes an Daten für die Datenauswertung zusammen (Digital Innovation and Transformation, HBS, 2017).

Auch bei der Stauerkennung sind Spatial Big Data Analysen hilfreich. So erkennt Google mit dem in Google Maps integrierten Dienst Google Traffic über Position und Bewegung der Mobilfunkgeräte, auf welchen Straßen viel Verkehr ist. Über die von Google gekaufte App Waze können Benutzer Verkehrsinformationen wie Staus, Baustellen und Straßen-sperrungen melden. Außerdem wertet Waze alternative Routen aus, welche die Nutzer teilen können und verteilt so den Verkehr über mehrere mögliche Routen (heise online, 2011; theguardian.com, 2013). Graphmasters hat mit NUNAV eine „kollaborative Routing-Plattform für Navigationssysteme“ (Graphmasters GmbH, 2018) entwickelt, welcher den Verkehr im Voraus über verschiedene Routen schickt, um so Staus zu vermeiden. Dieses System wurde mehrfach bei Großveranstaltungen im Raum Hannover eingesetzt.

9.1.5 Daten-Broker und Geomarketing

Neben den praktischen Anwendungsfällen gibt es im Zuge von Big Data auch ein das Geschäftsmodell des Daten Brokers. Sie sammeln Daten und stellen diese aufbereitet anderen für deren Analysen zur Verfügung. Ein Beispiel für die das Sammeln und Verkaufen von Geodaten sind Unternehmen wie microm, GfK oder ddsgeo. Sie sammeln sozio-demographische Daten, zum Konsumentenverhalten, Kaufkraft, Dienstleisterstruktur und vieles mehr. Diese Daten werden dann entweder in eigene Produkten wie dem Regio-Graph von GfK (2018) aufbereitet, oder auch als Daten verkauft. Die Unternehmen microm und ddsgeo haben zusammen die fünfstellige Postleitzahl um drei Ziffern erweitert und auf diese Weise homogene Raumgliederung mit ca. 500 Haushalten je Raumeinheit geschaffen. Mit dieser feineren Raumeinheit und Kennzahlen – die auf dieser Ebene aggregiert werden – können zum Beispiel Werbeaktionen mit Flyern zielgenauer und kosteneffizienter durchgeführt werden (ddsgeo, 2013; microm, 2017).

Keiner der Dienste gibt detailliert Auskunft darüber, woher die Daten stammen. Aufgrund aktueller Veröffentlichungen von ddsgeo (2018) und GfK (Bosch, 2016) mit eher zurückhaltenden Aussagen über Big Data ist jedoch davon auszugehen, dass Big Data bei der Datenerhebung bisher noch keine große Rolle spielt. Das Spatial Big Data jedoch auch für Geomarketing relevante Daten bereit hält, zeigt das Anwendungsbeispiel zur Standortplanung von Geospin (Wagner, 2016). Mit dem Einsatz von Spatial Big Data könnte hier auf der einen Seite neue Erkenntnisse gewonnen werden und auf der anderen

Seite ggf. kosteneffizienter gearbeitet werden, da weniger manuelle Datenerhebungen notwendig sein würden.

9.2 Gentrifizierung

9.2.1 Phasenmodelle

Neben dem doppelten Invasions-Sukzessions-Zyklus gibt es auch weitere phasenbasierte Modelle. Friedrichs bezeichnet diese als „Phasenmodelle des Wandels von Nachbarschaften“ (Friedrichs, 1996, S. 17). Unter den vorgestellten Modellen sind jedoch viele, die den Niedergang von Wohnvierteln thematisieren, an dessen Ende dann wiederum ein Gentrifizierungsprozess stehen kann. Beispiele hierzu wären die Modelle von Hoover und Vernon (1959), sowie das von Birch (1971). Beide Modelle haben mehrere Phasen, die nach einander durchschritten werden. Friedrichs (1996, S. 17–21) hat die Modelle in seiner Arbeit überführt und gegenübergestellt. Er stellt fest, dass allen Modellen eine „regelhafte und irreversible Abfolge der Phasen [zugrunde liegt], wobei die Länge der einzelnen Phase offen bleibt“ (Friedrichs, 1996, S. 17). Im Folgenden werden die Phasen nach Birch (1971) kurz dargestellt:

1. Neubaugebiet
Neubaugebiet am Stadtrand mit Einfamilienhäusern
 2. Übergang
Bau von Mehrfamilienhäusern, Erhöhung der Einwohnerdichte
 3. Voll entwickelt
Umwandlung von Einfamilienhäusern in Mehrfamilienhäusern, Hohe einwohnerdichte, Hoher Anstieg von Bodenpreisen und Mieten
 4. Herabstufung
Zuzug von Status schwächeren Gruppen, Gebäude mit veralteter Substanz, Überbelegungen von Wohnungen, hohe Einwohnerdichte, niedrige Mieten
 5. Ausdünnen
Junge Haushalte ziehen fort, Sinkende Zahl der Bewohner, geringe Einwohnerdichte, Leerstand
 6. Erneuerung
Sanierung nach Wechsel der Besitzer, Zuzug von Mittelklasse und Oberklasse, staatliche Sanierung, verbesserte Wohnqualität, Änderung der Nutzung durch Büros oder neue Apartments
- Laut Friedrichs (1996, S. 20) kann in dieser letzten Phase Gentrifizierung auftreten, muss sie jedoch nicht.

9.3 Data Understanding

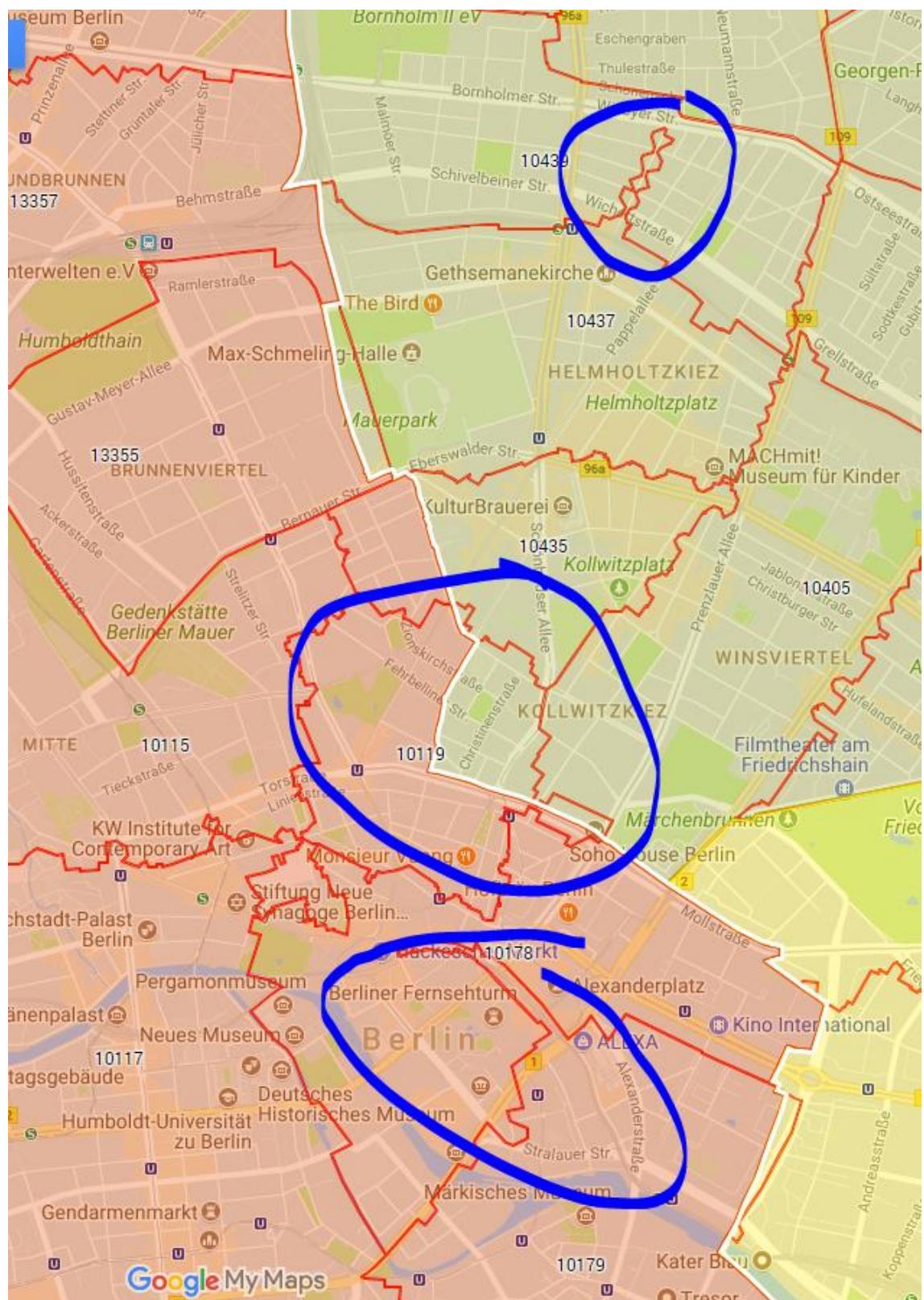


Abbildung 9-3: Postleitzahlen in Berlin (Eigene Darstellung mit Google MyMaps nach Amt für Statistik Berlin-Brandenburg)

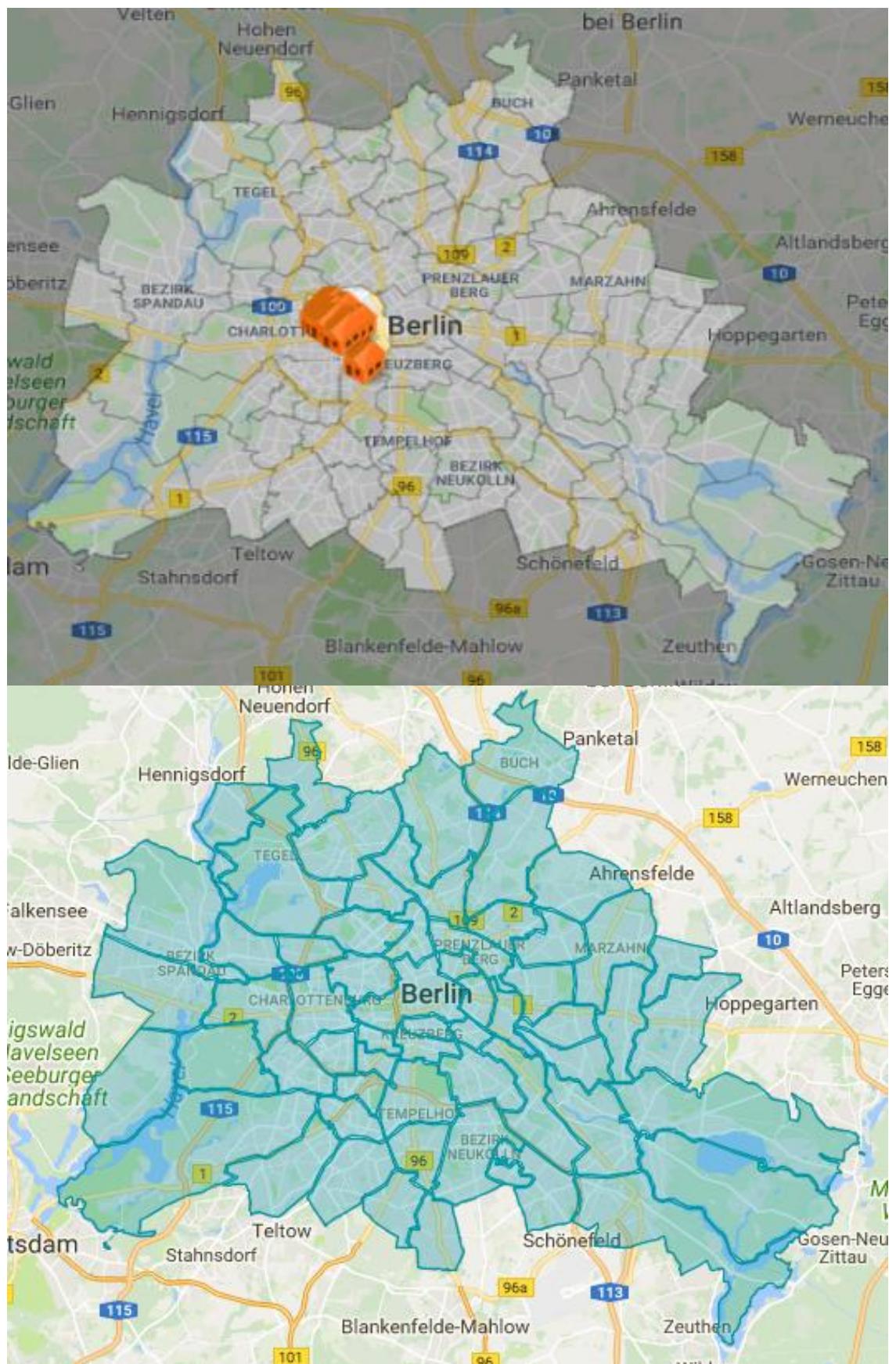


Abbildung 9-4: Vergleich ImmobilienScout24 (oben, ImmobilienScout24) mit Prognoseräumen (unten, Eigene Darstellung mit Google MyMaps nach Amt für Statistik Berlin-Brandenburg)



Abbildung 9-5: Vergleich Kreuzberg ImmobilienScout24 (links, ImmobilienScout24) mit Prognoseräumen (rechts, eigene Darstellung mit Google MyMaps nach Amt für Statistik Berlin-Brandenburg)

Tabelle 9-1: Liste der ImmobilienScout24 Quarter in Berlin (Immobilien Scout GmbH, o.D.a)

Quarter-ID	Quarter-Label
1	Adlershof (Treptow)
2	Altglienicke (Treptow)
3	Biesdorf (Marzahn)
4	Blankenburg (Weißensee)
5	Blankenfelde (Pankow)
6	Bohnsdorf (Treptow)
7	Britz (Neukölln)
8	Buch (Pankow)
9	Französisch Buchholz (Pankow)
10	Buckow (Neukölln)
11	Charlottenburg (Charlottenburg)
12	Dahlem (Zehlendorf)
13	Falkenberg (Hohenschönhausen)
14	Friedenau (Schöneberg)
15	Friedrichsfelde (Lichtenberg)
16	Friedrichshagen (Köpenick)
17	Friedrichshain (Friedrichshain)
18	Frohnau (Reinickendorf)
19	Gatow (Spandau)
20	Grunewald (Wilmersdorf)

21	Grünau (Köpenick)
22	Haselhorst (Spandau)
23	Heiligensee (Reinickendorf)
24	Heinersdorf (Weißensee)
25	Hellersdorf (Hellersdorf)
26	Hermsdorf (Reinickendorf)
27	Alt-Hohenschönhausen (Hohenschönhausen)
28	Johannisthal (Treptow)
29	Karlshorst (Lichtenberg)
30	Karow (Weißensee)
31	Kaulsdorf (Hellersdorf)
32	Kladow (Spandau)
33	Konradshöhe (Reinickendorf)
34	Kreuzberg (Kreuzberg)
35	Köpenick (Köpenick)
36	Lankwitz (Steglitz)
37	Lichtenberg (Lichtenberg)
38	Lichtenrade (Tempelhof)
39	Lichterfelde (Steglitz)
40	Lübars (Reinickendorf)
41	Mahlsdorf (Hellersdorf)
42	Malchow (Hohenschönhausen)
43	Mariendorf (Tempelhof)
44	Marienfelde (Tempelhof)
45	Marzahn (Marzahn)
46	Mitte (Mitte)
47	Müggelheim (Köpenick)
48	Neukölln (Neukölln)
49	Niederschöneweide (Treptow)
50	Niederschönhausen (Pankow)
51	Nikolassee (Zehlendorf)
52	Oberschöneweide (Köpenick)
53	Pankow (Pankow)
54	Prenzlauer Berg (Prenzlauer Berg)

55	Rahnsdorf (Köpenick)
56	Reinickendorf (Reinickendorf)
57	Rosenthal (Pankow)
58	Rudow (Neukölln)
59	Schmargendorf (Wilmersdorf)
60	Schmöckwitz (Köpenick)
61	Schöneberg (Schöneberg)
62	Siemensstadt (Spandau)
63	Spandau (Spandau)
64	Staaken (Spandau)
65	Steglitz (Steglitz)
66	Tegel (Reinickendorf)
67	Tempelhof (Tempelhof)
68	Tiergarten (Tiergarten)
69	Baumschulenweg (Treptow)
70	Waidmannslust (Reinickendorf)
71	Wannsee (Zehlendorf)
73	Wedding (Wedding)
74	Weißensee (Weißensee)
76	Wilmersdorf (Wilmersdorf)
77	Wittenau (Reinickendorf)
78	Zehlendorf (Zehlendorf)
79	Treptow (Treptow)
80	Neu-Hohenschönhausen (Hohenschönhausen)
81	Plänterwald (Treptow)
82	Rummelsburg (Lichtenberg)
83	Wartenberg (Hohenschönhausen)

```

<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="CGImap 0.6.1 (12524 thorn-01.open-
streetmap.org)" copyright="OpenStreetMap and contributors" attribu-
tion="http://www.openstreetmap.org/copyright" li-
cense="http://opendatacommons.org/licenses/odbl/1-0/">
  <bounds minlat="52.5080000" minlon="13.4582900" maxlat="52.5095200"
maxlon="13.4625100"/>
  <node id="282415700" visible="true" version="15" changeset="60270787"
timestamp="2018-06-29T10:17:12Z" user="wheelmap_visitor" uid="290680"
lat="52.5091398" lon="13.4598832">
    <tag k="addr:city" v="Berlin"/>

```

```

<tag k="addr:country" v="DE"/>
<tag k="addr:housenumber" v="29a"/>
<tag k="addr:postcode" v="10245"/>
<tag k="addr:street" v="Gärtnerstraße"/>
<tag k="addr:suburb" v="Friedrichshain"/>
<tag k="amenity" v="restaurant"/>
<tag k="cuisine" v="italian"/>
<tag k="name" v="Hannibal"/>
<tag k="opening_hours" v="Su-Th 09:00-02:00, Fr,Sa 09:00-04:00"/>
<tag k="smoking" v="outside"/>
<tag k="toilets:wheelchair" v="yes"/>
<tag k="wheelchair" v="yes"/>
</node>
<nnode id="5727191929" visible="true" version="2" changeset="60401491"
timestamp="2018-07-04T08:38:31Z" user="Segelpaule" uid="146822"
lat="52.5088869" lon="13.4599333">
  <tag k="crossing" v="traffic_signals"/>
  <tag k="highway" v="crossing"/>
  <tag k="image" v="http://storage8.openstreetcam.org/fi-
les/photo/2018/5/12/lth/1180083_6e80e_5af66af5999e8.jpg"/>
  <tag k="survey:date" v="2018-05-12"/>
  <tag k="wheelchair" v="yes"/>
</node>
<way id="23248761" visible="true" version="12" changeset="60265829"
timestamp="2018-06-29T04:55:03Z" user="kartonage" uid="1497225">
  <nd ref="26960748"/>
  <nd ref="5727191929"/>
  <nd ref="5727191930"/>
  <nd ref="29785855"/>
  <tag k="highway" v="tertiary"/>
  <tag k="maxspeed" v="30"/>
  <tag k="name" v="Gärtnerstraße"/>
  <tag k="postal_code" v="10245"/>
  <tag k="surface" v="asphalt"/>
</way>
<relation id="1843581" visible="true" version="122" chang-
eset="61062280" timestamp="2018-07-25T16:49:54Z" user="kartonage"
uid="1497225">
  <member type="way" ref="35937388" role="" />
  [...]
  <member type="way" ref="265762765" role="" />
  <tag k="name" v="Innerer Parkring"/>
  <tag k="network" v="rwn"/>
  <tag k="operator" v="Senatsverwaltung für Stadtentwicklung und Um-
welt Berlin"/>
  <tag k="osmc:symbol" v="blue:white:blue_bar:18:black"/>
  <tag k="ref" v="18"/>
  <tag k="route" v="hiking"/>
  <tag k="symbol" v="blauer Balken auf weiß mit 18"/>
  <tag k="type" v="route"/>
  <tag k="wikidata" v="Q17353685"/>
  <tag k=".wikipedia" v="de:Innerer Parkring"/>
</relation>
<relation id="5824760" visible="true" version="1" chang-
eset="36343175" timestamp="2016-01-03T17:39:03Z" user="Balgofil"
uid="95702">
  <member type="way" ref="389400777" role="platform"/>
  <member type="node" ref="1822518541" role="stop"/>
  <member type="way" ref="389400776" role="platform"/>
  <member type="node" ref="1137469153" role="stop"/>
  <tag k="name" v="Wühlischstraße/Gärtnerstraße"/>
  <tag k="network" v="Verkehrsverbund Berlin-Brandenburg"/>
  <tag k="network:short" v="VBB"/>
  <tag k="operator" v="Berliner Verkehrsbetriebe"/>

```

```
<tag k="operator:short" v="BVG"/>
<tag k="public_transport" v="stop_area"/>
<tag k="type" v="public_transport"/>
</relation>
</osm>
```

Abbildung 9-6: XML-Code Planet.osm Beispiele (Eigene Darstellung nach OpenStreetMap contributors, o.D.)

Tabelle 9-2: Durchschnittliche Angebotsmietpreise in EUR (Daten von ImmobilienScout24, Quelle: Rundfunk Berlin-Brandenburg (2016))

Durchschn. Angebotsmietpreise in €	Q2_2008	Q2_2009	Q2_2010	Q2_2011	Q2_2012	Q2_2013	Q2_2014	Q2_2015	Q2_2016
Bezirk Treptow-Köpenick	5,49	5,58	5,85	6,12	6,48	6,78	7,14	7,40	7,73
Bezirk Marzahn-Hellersdorf	4,73	4,79	4,96	5,12	5,40	5,64	5,81	6,01	6,31
Bezirk Pankow	6,05	6,22	6,64	7,09	7,50	7,96	8,42	8,85	9,29
Bezirk Neukölln	5,18	5,45	5,78	6,17	6,67	7,22	7,67	8,08	8,56
Bezirk Charlottenburg-Wilmersdorf	6,78	6,96	7,27	7,63	8,11	8,52	8,99	9,31	9,80
Bezirk Steglitz-Zehlendorf	6,42	6,56	6,83	7,17	7,55	7,91	8,30	8,58	8,96
Bezirk Lichtenberg	5,29	5,45	5,78	6,09	6,51	6,88	7,33	7,67	7,99
Bezirk Tempelhof-Schöneberg	5,92	6,08	6,38	6,73	7,23	7,64	8,03	8,44	8,75
Bezirk Friedrichshain-Kreuzberg	6,11	6,37	6,79	7,22	7,80	8,33	8,95	9,48	10,14
Bezirk Reinickendorf	5,42	5,53	5,85	6,09	6,47	6,76	7,09	7,41	7,70
Bezirk Spandau	5,22	5,31	5,54	5,79	6,12	6,40	6,70	6,98	7,29
Bezirk Mitte	5,82	6,00	6,48	7,05	7,51	8,20	8,91	9,27	9,70
Berlin	5,70	5,86	6,18	6,52	6,94	7,35	7,78	8,12	8,52

Tabelle 9-3: Veränderung der durchschnittlichen Angebotsmietpreise gegenüber dem Vorjahr (Daten von ImmobilienScout24, Quelle: Rundfunk Berlin-Brandenburg (2016))

Veränderung der durchschn. Angebotsmietpreise ggü. Vorjahr	Q2_2009	Q2_2010	Q2_2011	Q2_2012	Q2_2013	Q2_2014	Q2_2015	Q2_2016
Bezirk Treptow-Köpenick	1,6%	4,8%	4,6%	5,9%	4,6%	5,4%	3,6%	4,5%
Bezirk Marzahn-Hellersdorf	1,3%	3,4%	3,3%	5,4%	4,5%	3,2%	3,3%	5,0%
Bezirk Pankow	2,9%	6,6%	6,9%	5,7%	6,2%	5,8%	5,0%	5,1%
Bezirk Neukölln	5,1%	6,2%	6,7%	8,0%	8,3%	6,2%	5,3%	6,0%
Bezirk Charlottenburg-Wilmersdorf	2,7%	4,5%	4,9%	6,2%	5,1%	5,5%	3,6%	5,3%
Bezirk Steglitz-Zehlendorf	2,2%	4,1%	5,0%	5,3%	4,7%	5,0%	3,3%	4,5%
Bezirk Lichtenberg	3,0%	6,0%	5,3%	6,9%	5,7%	6,5%	4,5%	4,2%
Bezirk Tempelhof-Schöneberg	2,8%	5,0%	5,5%	7,4%	5,6%	5,1%	5,1%	3,6%
Bezirk Friedrichshain-Kreuzberg	4,3%	6,5%	6,4%	8,0%	6,9%	7,4%	5,9%	7,1%
Bezirk Reinickendorf	2,1%	5,8%	4,0%	6,3%	4,4%	5,0%	4,5%	3,8%
Bezirk Spandau	1,8%	4,3%	4,5%	5,6%	4,6%	4,7%	4,3%	4,4%
Bezirk Mitte	3,1%	7,9%	8,8%	6,5%	9,2%	8,7%	4,1%	4,6%
Berlin	2,8%	5,5%	5,6%	6,5%	5,9%	5,8%	4,4%	4,9%

Tabelle 9-4: Auszug aus „Bestand an Kraftfahrzeugen und Kraftfahrzeughängern nach Gemeinden (FZ3)“ (Kraftfahrt-Bundesamt, o.D.)

Land	Zulas-	PLZ, Gemeinde	Kraft-	Personenkraftwa-		Last-	Zugmaschinen		Sonstige	Kraft-	Kraft-
				gen	insge-		darun-	kraft-	ein-	fahr-	fahr-
bezahl	bezahl		bezahl	bezahl	bezahl	wagen	bezahl	schl.	zeuge	insge-	zeug-
bezahl	bezahl		bezahl	bezahl	bezahl		bezahl		samt	samt	anhän-
				insge-	darun-		land-/		Kraftomni	insge-	nger
				samt	ter ge-		forst-		-busse	samt	
					werbl.		wirt-				
					Halter		schaft-				
NIEDERSACHSEN (03456)	GRAFSCHAFT BENTHEIM	48455 BAD BENTHEIM,ST.	655	9.905	874	580	569	363	59	11.768	2.233
		49824 EMLICHHEIM	433	4.174	313	336	333	263	54	5.330	1.301
		48465 ENGDEN	31	260	8	.	83	71	.	394	74
		49828 ESCHE	.	369	9	48	96	78	.	556	129
		49828 GEORGSDORF	82	804	34	.	102	87	.	1.034	225
		49843 GETELO	25	323	7	15	94	82	-	457	127
		49843 GOELENKAMP	40	390	.	.	168	139	.	631	175
		49843 HALLE	33	388	7	.	124	114	.	576	145

	49846 HOOGSTEDE	206	1.786	68	172	306	259	18	2.488	697
	48465 ISTERBERG	38	414	33	.	147	117	.	638	168
	49847 ITTERBECK	121	1.205	132	157	301	246	16	1.800	509
	49824 LAAR	96	1.259	49	92	375	313	15	1.837	525
	49828 LAGE	78	620	.	.	46	41	.	772	171
	49828 NEUENHAUS,ST.	497	5.861	474	454	283	237	35	7.130	1.663
	48527 NORDHORN,ST.	2.090	30.138	2.865	1.955	1.081	615	248	35.512	5.632
	48465 OHNE	34	370	10	41	88	71	16	549	157
	49828 OSTERWALD	53	751	83	176	278	225	32	1.290	432
	48465 QUENDORF	33	338	8	.	85	73	.	476	115
	49824 RINGE	139	1.324	89	150	212	180	12	1.837	466
	48465 SAMERN	.	444	17	35	84	71	.	596	140
	48465 SCHUETTORF, STADT	626	7.440	676	413	131	86	52	8.662	1.337
	49843 UELSEN	297	3.486	196	260	175	150	17	4.235	944
	49847 WIELEN	35	340	8	.	77	67	.	485	133
	49835 WIETMARSCHEN	616	7.675	631	597	543	441	50	9.481	2.059
	49849 WILSUM	.	1.107	108	102	247	206	.	1.564	419
	ZUSAMMEN	6.429	81.171	6.705	5.805	6.028	4.595	665	100.098	19.976

NIE- DER- SACH SEN INS- GE- SAMT			418.92 2	4.674.05 9	444.75 4	288.419 8	249.15 8	176.034 	39.514 	5.670.07 2	927.59 3
BER- LIN	BER- LIN (11000)	10115 BERLIN ZUSAMMEN	105.08 0	1.202.82 9	157.69 6	96.943 96.943	6.446 6.446	1.634 1.634	10.767 10.767	1.422.06 5	88.808 88.808
BER- LIN INS- GE- SAMT			105.08 0	1.202.82 9	157.69 6	96.943 96.943	6.446 6.446	1.634 1.634	10.767 10.767	1.422.06 5	88.808 88.808

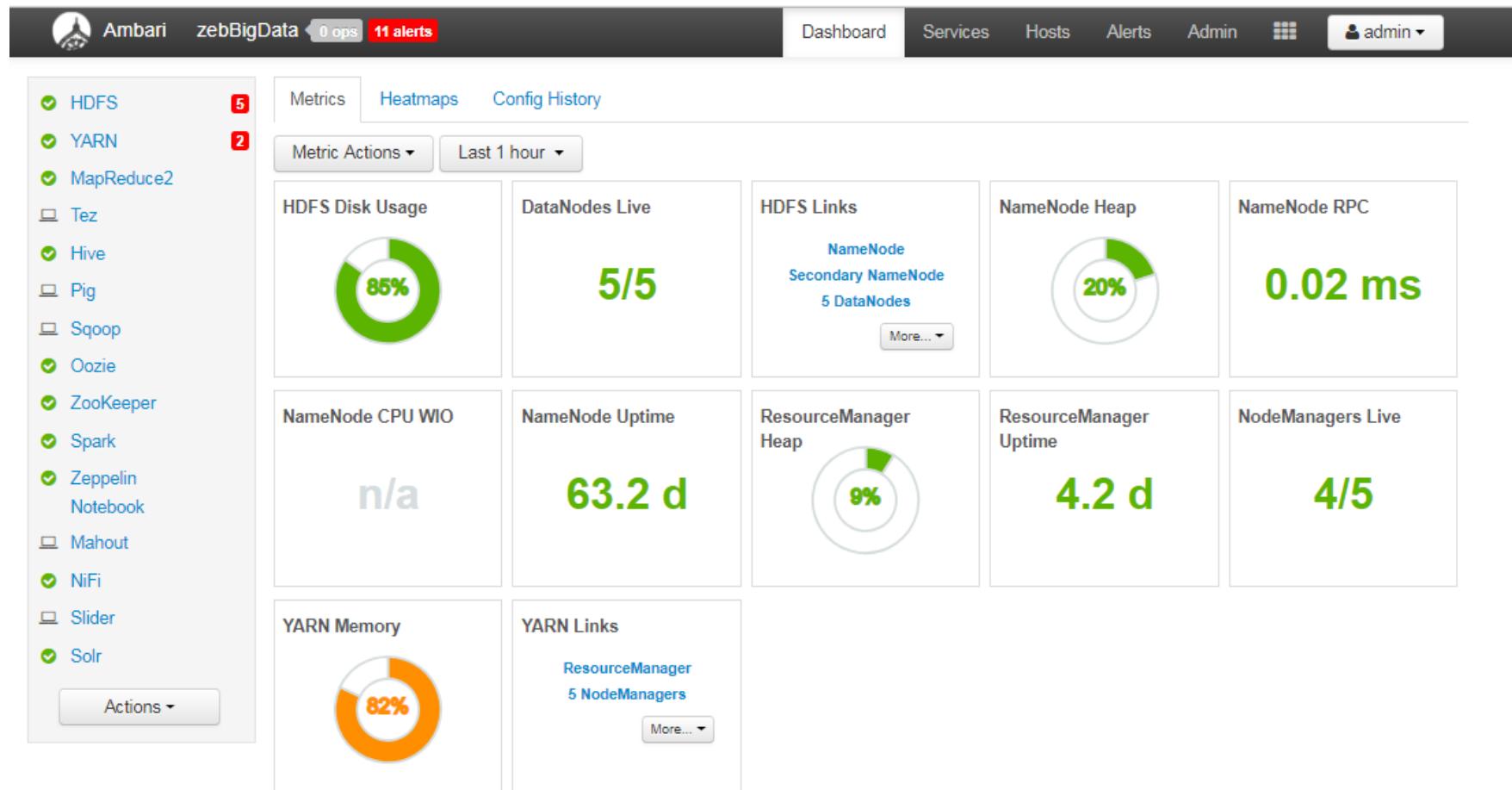


Abbildung 9-7: Screenshot Ambari

9.3.1 Details zu kommerziellen Datenquellen zur Angebotsstruktur

9.3.1.1 Google Places API

Google Maps ist einer der bekanntesten Kartendienste. Für Entwickler bietet Google die Google Places API an, mit der auf der einen Seite nach POIs gesucht werden kann, und auf der anderen Seite Details zu einem spezifischen POI abgefragt werden können. Bei der Suche nach POIs kann innerhalb der *Google Place Search* die *Nearby Search* verwendet werden. Hier kann ein Punkt mit Suchradius von bis zu 50 km angegeben werden. Optional können Filterkriterien angegeben werden, wie *aktuell geöffnet*, *Preissegment* oder den *POI-Typ*. Als Typ hat Google 90 POI-Typen definiert, darunter *restaurant*, *bar*, *bank*, *atm*, *doctor* und viele mehr. Als Ergebnis werden 20 POI zurückgegeben. Die Suche kann auf maximal 60 Treffer erweitert werden. Jeder POI kommt mit den vorhandenen Informationen *Position in WGS84*, *Plus Code*, *Name*, *Öffnungszeiten*, *Place_ID*, *Icon-Link*, *Foto-Links*, *Preissegment* (0- gratis bis 4 sehr teuer), *Rating* (zwischen 1.0 und 5.0, aus User-Reviews), *Liste an passenden POI-Typen zum Geschäft*, sowie ein Kennzeichen ob das Geschäft *permanent geschlossen* ist. Um mehr Informationen zum POI zu bekommen, stehen mittels *Google Place Details* Daten wie *detaillierte Adressdaten*, *Telefonnummern*, *Internetadresse (URL)*, *Zeitzone*, *Preissegment*, *Rating* und einzelne *Reviews* zur Verfügung (Google Developers, 2018).

Die verfügbaren Datenfeatures je POI sind sehr interessant für Analysen. Allerdings ist durch die Beschränkung auf 60 Treffer je Suchanfrage eine Datenerhebung sehr aufwendig.

9.3.1.2 Foursquare

Foursquare ist ein Dienst der Usern Orte zum Essen, Trinken, Einkaufen oder Besichtigen empfiehlt. Nach eigener Angabe sind bei dem Dienst über 105 Millionen Venues erfasst; Venues ist die Bezeichnung die Foursquare für POI wählt. Bei Foursquare können User per Check-In bekannt geben, dass sie in einem Venue sind, wie zum Beispiel in einem Restaurant. Foursquare bietet im Vergleich mit Google eine deutlich detailliertere POI-Typen Liste an, welche zudem in einer Hierarchie vorliegt. So werden zum Beispiel 26 Typen von Italienischen Restaurants unterschieden. Italienische Restaurants sind wiederum Teil der Oberkategorie Essen. Verfügbar sind 923 POI-Typen in den Oberkategorien *Kultur & Unterhaltung* (65 Typen), *Hochschule & Universität* (39 Typen), *Veranstaltung* (11 Typen), *Essen* (342 Typen), *Nachtleben* (25 Typen), *Natur & Freizeit* (104 Typen), *Berufliches & andere Orte* (105 Typen), *Wohnort* (6 Typen), *Geschäft & Dienstleistungen* (172 Typen), sowie *Reise & Verkehr* (54 Typen) (Foursquare, o.D.). Mit dem API-Aufruf */venue/search* können POIs gesucht werden. Der Aufruf kann auf verschiedene

Arten parametrisiert werden, für die Zwecke dieser Analyse wäre die Suche auf *intent=browse* einzustellen. Dabei kann entweder eine Umkreissuche, mit Position und Radius, oder eine Box-Suche, mit Angabe einer Süd-West- sowie Nord-Ost-Koordinate, durchgeführt werden; der maximale Radius beträgt 100 km, die maximale Box Größe beträgt 10.000 km². Es kann auf Basis von ein oder mehrerer POI-Typen gefiltert werden, wenn ein Element mit Kinder-Elementen gewählt wird, so werden auch diese mit ausgegeben. Im Ergebnis wird eine Liste an POIs zurückgegeben, welche *ID*, *Namen*, *Adresse*, *GPS-Position* und eine Liste der passenden POI-Typen enthält. Per API-Aufruf *venues/VENUE_ID* können weitere Details abgefragt werden wie *Anzahl User*, *Anzahl Check-Ins*, *Öffnungszeiten*, *Stoßzeiten*, *Link zum Menü*, *Preiskategorie* (1 günstig – 4 teuer), *Rating* (1 - 10), *Alter* des Eintrags bei Foursquare, *URL*, *Anzahl der Likes*, sowie eine Reihe an weiteren möglichen Featuren wie *Parkmöglichkeiten* (Foursquare, o.D.). Allerdings hat die Foursquare API bei der Suche ein Limit von 50 Ergebnissen.

Foursquare besitzt sehr viele POIs in einer feingliedrigen Hierarchie. Zudem gibt es die Möglichkeit das Erfassungsdatum auszuwerten, sowie eine Suche per Rechteck auszuführen, was eine Stadtweite Erfassung aller POIs vereinfacht. Auch Preis- und Ratinginformationen könnten für die Analyse der lokalen Angebotsstruktur nützlich sein. Durch die Limitierung von 50 POIs pro Abfrage ist jedoch wie bereits bei der Google API eine systematische Datenerfassung aus Foursquare sehr aufwendig.

9.3.1.3 Tripadvisor

Tripadvisor ist ein weiterer bekannter Anbieter von Restaurant- und Hotel- und Aktivitäten-Empfehlungen. Auch Tripadvisor hat eine POI-Typen Hierarchie, welche auf der höchsten Ebene die drei Kategorien *restaurants*, *hotels*, und *attractions* hat. Die Restaurants sind unterteilt in *sit down*, *fast food* und *bakery* haben. Zusätzlich kann ein Restaurant die Information *cuisine* haben, welche unter 157 verschiedenen Küchenarten unterscheidet, wovon mehrere hinterlegt werden können. Die API bietet die Möglichkeit Details eines POIs zu erhalten (TripAdvisor Developer Portal, o.D.). In der Dokumentation ist jedoch kein Hinweis darauf zu finden, ob eine Suche von POIs über die API möglich ist.

Ohne die Möglichkeit per API nach mehreren POIs zu suchen, kann die Tripadvisor API nicht für die Analyse der lokalen Angebotsstrukturen genutzt werden.

9.3.1.4 Yelp

Wie die anderen Empfehlungs- und Bewertungsportal für Restaurants und andere Geschäfte hat auch Yelp eine API entwickelt. Auch bei Yelp gibt es eine Vielzahl an POI-Typen, dabei darf jedes POI maximal drei Typen zugeordnet werden. Die Liste der POI-

Typen ist länderspezifisch und hierarchisch geordnet. Für Deutschland stehen 447 spezielle und 694 länderübergreifende POI-Typen zur Verfügung, insgesamt gibt es 1.587 POI-Typen (Yelp Fusion API, o.D.a). Über den API-Aufruf `/businesses/search` können bis zu 1.000 Suchergebnisse auf Basis von Filterkriterien zurückgegeben werden. Es werden jedoch keine POIs ohne bestehendes Review ausgegeben. Räumlich kann auch hier Mittels GPS-Koordinate und Radius gefiltert werden. Mittels API-Aufruf von `/businesses/{id}` können dann die Details zu den einzelnen POIs abgerufen werden. Auch bei Yelp stehen Informationen wie *Adresse*, *Telefonnummer*, *Öffnungszeiten*, *Preissegment* und *Rating* zur Verfügung (Yelp Fusion API, o.D.b).

Die Yelp-API bietet die Möglichkeit POIs systematisch abzufragen. Allerdings gibt es ein Limit von 1.000 Treffern je Aufruf. Zusätzlich gibt es eine nur die Möglichkeit per Umkreissuche POIs zu suchen, was ein systematisches Abdecken einer ganzen Region erschwert.

9.3.2 Sonstige Datenquellen

9.3.2.1 openSenseMap

Das Institut für Geoinformatik der Universität Münster hat im Jahr 2015 in Zusammenarbeit mit dem Unternehmen senseBox eine Plattform für offene Sensordaten entwickelt. Mittels einer API und einer auf OSM-basierten Karte können die Daten von über 2.000 senseBoxen historisiert ausgelesen werden (senseBox, 2015). Die senseBoxen ermitteln unter anderem Messwerte zu Temperatur, Luftfeuchtigkeit und Feinstaubgehalt. Die Sensordaten sind über die offene Lizenz *Open Data Commons Public Domain Dedication and License (PDDL)*⁵⁸ geschützt, die keinerlei Restriktionen enthält. Diese Daten könnten genutzt werden um den Living Environment Wert des IMD nachzubilden.⁵⁹

Auch wenn die Sensordaten bzgl. der Luftqualität eine interessante Größe wären, so ist ein fachlicher Bezug zur Gentrifizierung schwer herzustellen. Aus diesem Grund wird diese Datenquelle nicht für die Analyse verwendet.

⁵⁸ <https://opendatacommons.org/licenses/pddl/summary/>

⁵⁹ Vgl. Venerandi, Quattrone, Capra, Quercia und Saez-Trumper (2015) und Hristova, Williams, Musolesi, Panzarasa und Mascolo (2016) in Kapitel 3.5

9.4 Data Preparation – Datentransformationen

9.4.1 OSM-POIs erzeugen

Tabelle 9-5: POI-Hierarchie

domain	category	type
Buero	Buero	Buero
Dienstleistung	Beerdigung	Beerdigung
Dienstleistung	Friseur	Friseur
Dienstleistung	Kosmetik und Beauty	Kosmetik und Beauty
Dienstleistung	Massage	Massage
Dienstleistung	Reisen	Reisen
Dienstleistung	Waescherei	Waescherei
Dienstleistung	Waescherei	Reinigung
Gastronomie	Cafe	Eisdiele
Gastronomie	Cafe	Kaffee
Gastronomie	Cafe	Sonstige Cafes
Gastronomie	Fast Food	Fastfood Kebap
Gastronomie	Fast Food	Fastfood Burger
Gastronomie	Fast Food	Fastfood Asiatisch
Gastronomie	Fast Food	Fastfood Pizza
Gastronomie	Fast Food	Fastfood Pommesbude
Gastronomie	Fast Food	fast_food Sonstiges
Gastronomie	Restaurant	Restaurant Italiener
Gastronomie	Restaurant	Restaurant Deutsch
Gastronomie	Restaurant	Restaurant Indisch
Gastronomie	Restaurant	Restaurant Asiatisch
Gastronomie	Restaurant	Restaurant Sushi
Gastronomie	Restaurant	Restaurant Griechisch
Gastronomie	Restaurant	Restaurant Steakhouse
Gastronomie	Restaurant	Restaurant International
Gastronomie	Restaurant	Restaurant Tuerkisch
Gastronomie	Restaurant	restaurant Sonstiges
Leerstand	Leerstand	Leerstand
Mobilitaet	Individual	Tankstelle
Mobilitaet	Individual	Autovermietung
Mobilitaet	Individual	Ladestation
Mobilitaet	Individual	Fahrradverleih
Mobilitaet	Individual	Taxistand
Mobilitaet	Individual	Parkplatz
Mobilitaet	Individual	Fahrrad Parkplatz
Mobilitaet	Individual	Parkticketautomat
Mobilitaet	OEPNV	Fahrscheinautomat
Mobilitaet	OEPNV	Haltestelle
Oeffentlicher Raum	Automaten	Sonstige Automaten

Oeffentlicher Raum	Automaten	Kotbeutelautomat
Oeffentlicher Raum	Automaten	Zigarettenautomat
Oeffentlicher Raum	Automaten	Suessigkeitenautomat
Oeffentlicher Raum	Automaten	Kondomautomat
Oeffentlicher Raum	Automaten	Getraenke und Suessigkeitenautomat
Oeffentlicher Raum	Automaten	Getraenkeautomat
Oeffentlicher Raum	Briefe und Pakete	Briefkasten
Oeffentlicher Raum	Briefe und Pakete	Paketautomat
Oeffentlicher Raum	Briefe und Pakete	Briefmarkenautomat
Oeffentlicher Raum	Parkbank	Parkbank
Oeffentlicher Raum	Post	Post Filiale
Oeffentlicher Raum	Recycling	Muelleimer
Oeffentlicher Raum	Recycling	Glascontainer
Oeffentlicher Raum	Recycling	Glas- und Kleidungscontainer
Oeffentlicher Raum	Recycling	Kleidungscontainer
Oeffentlicher Raum	Recycling	Sonstige Container
Oeffentlicher Raum	Sonstiges	Grillplatz
Oeffentlicher Raum	Sonstiges Gebaeude	Sonstiges Gebaeude
Oeffentlicher Raum	Sonstiges Gelaende	Sonstiges Gelaende
Oeffentlicher Raum	Telefon	Telefon
Oeffentlicher Raum	WC	WC
Public Service	Bank	Bankfiliale
Public Service	Bank	Geldautomat
Public Service	Bildung	Kindergarten
Public Service	Bildung	Universitaet
Public Service	Bildung	KiTa
Public Service	Bildung	Hoehere Schule
Public Service	Bildung	Buecherei
Public Service	Bildung	Schule

Public Service	Gesundheit	Zahnarzt
Public Service	Gesundheit	Tierarzt
Public Service	Gesundheit	Arzt
Public Service	Gesundheit	Apotheke
Public Service	Gesundheit	Krankenhaus
Public Service	Gesundheit	Klinik
Public Service	Sicherheit	Poilzei
Public Service	Sicherheit	Feuerwehr
Public Service	Sonstiges	Wochenmarkt
Public Service	Sonstiges	Fahrschule
Public Service	Sonstiges	Musikschule
Public Service	Sozial	Nachbarschaftszentrum
Public Service	Sozial	Sozialeinrichtung
Religion	Friedhof	Friedhof
Religion	Religioese Gebaeude	Kirche
Religion	Religioese Gebaeude	Moschee
Religion	Religioese Gebaeude	Sonstige Tempel
Sonstiges	Hipster	Coworking-Space
Sonstiges	Sonstiges	Botschaft
Sport und Erholung	Erholung	Sauna
Sport und Erholung	Erholung	Spielplatz
Sport und Erholung	Erholung	Wassersport
Sport und Erholung	Erholung	Sonstiges Erholung
Sport und Erholung	Sport	Schwimmen
Sport und Erholung	Sport	Basketball
Sport und Erholung	Sport	Fussball
Sport und Erholung	Sport	Sportzentrum
Sport und Erholung	Sport	Tischtennis
Sport und Erholung	Sport	Tennis
Sport und Erholung	Sport	Kampfsport
Sport und Erholung	Sport	Fitnesszentrum
Sport und Erholung	Sport	Sonstige Sportarten
Sport und Erholung	Sport	Sonstiges Sport
Tourismus	Info	Info
Tourismus	Sehenswuerdigkeit	Kunstwerk

Tourismus	Sehenswuerdigkeit	Aussichtspunkt
Tourismus	Sehenswuerdigkeit	Baudenkmal
Tourismus	Sehenswuerdigkeit	Sonstiges Denkmal
Tourismus	Sonstiges Tourismus	Sonstiges Tourismus
Tourismus	Uebernachtung	Hostel
Tourismus	Uebernachtung	Hotel
Tourismus	Uebernachtung	Sonstige
Vergnuegung	Ausgehen	Kino
Vergnuegung	Ausgehen	Nachtclub
Vergnuegung	Gaststaetten	Biergarten
Vergnuegung	Gaststaetten	Bar
Vergnuegung	Gaststaetten	Pub
Vergnuegung	Kultur	Kunstzentrum
Vergnuegung	Kultur	Gallerie
Vergnuegung	Kultur	Museum
Vergnuegung	Kultur	Zoo
Vergnuegung	Kultur	Theater
Vergnuegung	Zwielicht	Spielothek
Vergnuegung	Zwielicht	Bordell
Waren	Drogerie	Drogerie
Waren	Essen & Trinken	Baeckerei
Waren	Essen & Trinken	Supermarkt
Waren	Essen & Trinken	Kiosk
Waren	Essen & Trinken	Getraenke
Waren	Essen & Trinken	Feinkost
Waren	Essen & Trinken	Spirituosen
Waren	Essen & Trinken	Schlachter
Waren	Essen & Trinken	Suessigkeiten
Waren	Handwerk	Baumarkt
Waren	Handwerk	Eisenwarenhandlung
Waren	Kleidung	Kleidung
Waren	Kleidung	Schuhe
Waren	Kleidung	Boutique
Waren	Kleidung	Schneider
Waren	Kleidung	Sport
Waren	Kleidung	Second Hand
Waren	Kunst	Foto
Waren	Kunst	Kunst
Waren	Medical	Optiker
Waren	Medical	Hoergeraeete
Waren	Medical	Medical
Waren	Print	Buecher
Waren	Print	Copyshop
Waren	Print	Zeitung
Waren	Sonstige Waren	Florist
Waren	Sonstige Waren	Schmuck

Waren	Sonstige Waren	Moebel
Waren	Sonstige Waren	Postenmarkt
Waren	Sonstige Waren	Computer
Waren	Sonstige Waren	Dekoration
Waren	Sonstige Waren	Zoofachgeschaeft
Waren	Sonstige Waren	Textilgeschaeft
Waren	Sonstiger Shop	Sonstiger Shop
Waren	Spielzeug & Geschenke	Geschenke
Waren	Spielzeug & Geschenke	Spielzeug
Waren	Technik	Mobilfunk
Waren	Technik	Elektronik
Waren	Werkstatt	Fahrrad
Waren	Werkstatt	Autoreperatur
Waren	Werkstatt	Autohaus

Tabelle 9-6: Tabellenaufbau osmnodes_filtered_yymdd

id	N26735763
userid	1260280
ti-mestamp	2014-09-01T23:16:55Z
isvisible	false
version	18
change-setid	25170126
tags	{"addr:suburb":"Charlottenburg","wheelchair":"limited","addr:house-number":"106","amenity":"restaurant","addr:country":"DE","name":"Sakana","cuisine":"japanese","wheelchair:description":"Stufe am Eingang, aber rollstuhlgerechtes WC","addr:street":"Pestalozzistra?e","addr:postcode":"10625","addr:city":"Berlin"}
latitude	52.5073388
lon-gitude	13.3207848

```

add jar hdfs://itfin105.it.zeb.de:8280/user/admin/dhelweg/lormap-
per runlc.jar;

CREATE TEMPORARY FUNCTION planungsraum AS 'de.zeb.hive.udf.lormap-
per.Planungsraum';

CREATE TABLE osm_poi_2014 as
select
"201401" as timeslice,
concat(latitude, " ", longitude) as coords,
planungsraum(concat(latitude, " ", longitude)) as planungsraum,
id as node_id,

```

```

userid as last_modification_userid,
date_format(`timestamp`,'yyyyMM') as last_modification_time,
`version` as last_modification_version,
changesetid as last_modification_changesetid,
tags["name"] as name,
tags["description"] as description,
tags["addr:city"] as addr_city,
tags["addr:country"] as addr_country,
tags["addr:housenumber"] as addr_housenumber,
tags["addr:postcode"] as addr_postcode,
tags["addr:street"] as addr_street,
tags["addr:suburb"] as addr_suburb,
tags["amenity"] as amenity,
tags["cuisine"] as cuisine,
tags["vending"] as vending,
case
    when (tags["recycling:glass_bottles"] = "yes" or tags["recycling:glass"] = "yes")
        and (tags["recycling:clothes"] is null or tags["recycling:clothes"] = "no")
            then 'glass'
    when (tags["recycling:glass_bottles"] is null or tags["recycling:glass_bottles"] = "no")
        and (tags["recycling:glass"] is null or tags["recycling:glass"] = "no")
            and tags["recycling:clothes"] = "yes"
            then 'clothes'
    when (tags["recycling:glass_bottles"] = "yes" or tags["recycling:glass"] = "yes")
        and tags["recycling:clothes"] = "yes"
        then 'glass_clothes'
    when tags["amenity"] = "recycling"
        and (tags["recycling:glass"] is null and tags["recycling:glass_bottles"] is null
            and tags["recycling:clothes"] is null)
        then 'Sonstige Container'
else null end as recycling,
tags["atm"] as atm,
tags["heritage"] as heritage,
tags["religion"] as religion,
tags["tourism"] as tourism,
tags["sport"] as sport,
tags["leisure"] as leisure,
tags["office"] as office,
tags["landuse"] as landuse,
tags["shop"] as shop,
tags["public_transport"] as public_transport,
tags["wheelchair"] as wheelchair,
tags["opening_hours"] as opening_hours
from osmnodes_filtered_140101;

```

Abbildung 9-8: Hive-Code zu osm_poi_yyyy

Tabelle 9-7: Tabellenaufbau osm_poi_yyyy

timeslice	201501
coords	52.5073388, 13.3207848
planungsraum	4030828
node_id	N26735763

last_modification_userid	1260280
last_modification_time	201409
last_modification_version	18
last_modification_change-setid	25170126
name	Sakana
description	null
addr_city	Berlin
addr_country	DE
addr_housenumber	106
addr_postcode	10625
addr_street	Pestaloz-zistra?e
addr_suburb	Charlottenburg
amenity	restaurant
cuisine	japanese
vending	null
recycling	null
atm	null
heritage	null
religion	null
tourism	null
sport	null
leisure	null
office	null
landuse	null
shop	null
public_transport	null
wheelchair	limited
opening_hours	null

```
CREATE OR REPLACE VIEW osm_poi_type_2014 as
select
timeslice,
coords,
planungsraum,
node_id,
last_modification_userid,
last_modification_time,
last_modification_version,
last_modification_changesetId,
name,
description,
addr_city,
addr_country,
addr_housenumber,
addr_postcode,
addr_street,
```

```

addr_suburb,
case
WHEN tourism LIKE 'hostel' THEN 'Hostel'
WHEN tourism LIKE 'hotel' THEN 'Hotel'
WHEN tourism LIKE 'guest_house' THEN 'Sonstige'
WHEN tourism LIKE 'apartment' THEN 'Sonstige'
WHEN tourism LIKE 'motel' THEN 'Sonstige'
WHEN tourism LIKE 'artwork' THEN 'Kunstwerk'
WHEN amenity LIKE 'arts_centre' THEN 'Kunstzentrum'
WHEN tourism LIKE 'gallery' THEN 'Gallerie'
WHEN tourism LIKE 'information' THEN 'Info'
WHEN tourism LIKE 'museum' THEN 'Museum'
WHEN tourism LIKE 'viewpoint' THEN 'Aussichtspunkt'
WHEN tourism LIKE 'zoo' THEN 'Zoo'
WHEN sport LIKE 'swimming' THEN 'Schwimmen'
WHEN sport LIKE 'basketball' THEN 'Basketball'
WHEN sport LIKE 'soccer' THEN 'Fussball'
WHEN leisure = 'sports_centre' AND sport NOT like 'martial_arts' THEN
'Sportzentrum'
WHEN sport LIKE 'table_tennis' THEN 'Tischtennis'
WHEN leisure LIKE 'sauna' THEN 'Sauna'
WHEN sport LIKE 'tennis' THEN 'Tennis'
WHEN sport LIKE 'martial_arts' THEN 'Kampfsport'
WHEN leisure LIKE 'fitness_centre' THEN 'Fitnesszentrum'
WHEN leisure LIKE 'playground' THEN 'Spielplatz'
WHEN leisure LIKE 'marina' THEN 'Wassersport'
WHEN leisure LIKE 'adult_gaming_centre' THEN 'Spielothek'
WHEN sport IS NOT NULL AND sport NOT IN
('swimming',
'basketball',
'soccer',
'table_tennis',
'tennis',
'martial_arts')
THEN 'Sonstige Sportarten'
WHEN amenity LIKE 'casino' THEN 'Spielothek'
WHEN amenity LIKE 'kindergarten' THEN 'Kindergarten'
WHEN office IS NOT NULL THEN 'Buero'
WHEN amenity LIKE 'fuel' THEN 'Tankstelle'
WHEN amenity LIKE 'coworking_space' THEN 'Coworking-Space'
WHEN amenity LIKE 'university' THEN 'Universitaet'
WHEN amenity LIKE 'gambling' THEN 'Spielothek'
WHEN amenity LIKE 'police' THEN 'Poilzei'
WHEN amenity LIKE 'fire_station' THEN 'Feuerwehr'
WHEN amenity LIKE 'childcare' THEN 'KiTa'
WHEN amenity LIKE 'college' THEN 'Hoehere Schule'
WHEN amenity LIKE 'marketplace' THEN 'Wochenmarkt'
WHEN amenity LIKE 'cinema' THEN 'Kino'
WHEN amenity LIKE 'biergarten' THEN 'Biergarten'
WHEN amenity LIKE 'car_rental' THEN 'Autovermietung'
WHEN amenity LIKE 'library' THEN 'Buecherei'
WHEN amenity LIKE 'community_centre' THEN 'Nachbarschaftszentrum'
WHEN amenity LIKE 'theatre' THEN 'Theater'
WHEN amenity LIKE 'ice_cream' THEN 'Eisdiele'
WHEN amenity LIKE 'nightclub' THEN 'Nachtclub'
WHEN amenity = 'place_of_worship' AND religion like 'christian' THEN
'Kirche'
WHEN amenity = 'place_of_worship' AND religion like 'muslim' THEN
'Moschee'
WHEN amenity LIKE 'place_of_worship' AND religion NOT IN
('christian',
'muslim')
THEN 'Sonstige Tempel'
WHEN amenity LIKE 'school' THEN 'Schule'

```

```

WHEN amenity LIKE 'brothel' THEN 'Bordell'
WHEN amenity LIKE 'post_office' THEN 'Post Filiale'
WHEN amenity LIKE 'social_facility' THEN 'Sozialeinrichtung'
WHEN amenity LIKE 'charging_station' THEN 'Ladestation'
WHEN amenity LIKE 'dentist' THEN 'Zahnarzt'
WHEN amenity LIKE 'bicycle_rental' THEN 'Fahrradverleih'
WHEN amenity LIKE 'driving_school' THEN 'Fahrsschule'
WHEN amenity LIKE 'veterinary' THEN 'Tierarzt'
WHEN amenity LIKE 'embassy' THEN 'Botschaft'
WHEN amenity LIKE 'taxi' THEN 'Taxistand'
WHEN amenity LIKE 'toilets' THEN 'WC'
WHEN amenity LIKE 'bank' THEN 'Bankfiliale'
WHEN amenity LIKE 'doctors' THEN 'Arzt'
WHEN amenity LIKE 'parking' THEN 'Parkplatz'
WHEN amenity LIKE 'bar' THEN 'Bar'
WHEN amenity LIKE 'pharmacy' THEN 'Apotheke'
WHEN amenity LIKE 'pub' THEN 'Pub'
WHEN amenity LIKE 'telephone' THEN 'Telefon'
WHEN amenity = 'cafe' AND cuisine like 'ice_cream' THEN 'Eisdiele'
WHEN amenity = 'cafe' AND cuisine like 'coffee_shop' THEN 'Kaffee'
WHEN amenity LIKE 'cafe' AND cuisine NOT IN
('ice_cream',
'coffee_shop')
THEN 'Sonstige Cafes'
WHEN amenity LIKE 'bicycle_parking' THEN 'Fahrrad Parkplatz'
WHEN amenity LIKE 'waste_basket' THEN 'Muelleimer'
WHEN amenity LIKE 'bench' THEN 'Parkbank'
WHEN amenity LIKE 'hospital' THEN 'Krankenhaus'
WHEN amenity LIKE 'clinic' THEN 'Klinik'
WHEN amenity LIKE 'atm' THEN 'Geldautomat'
WHEN atm LIKE 'yes' THEN 'Geldautomat'
WHEN amenity LIKE 'post_box' THEN 'Briefkasten'
WHEN landuse LIKE 'cemetery' THEN 'Friedhof'
WHEN amenity LIKE 'music_school' THEN 'Musikschule'
WHEN amenity = 'vending_machine' AND vending like 'parcel_pickup;par-
cel_mail_in' THEN 'Paketautomat'
WHEN amenity LIKE 'vending_machine' AND vending NOT IN
('parcel_pickup;parcel_mail_in',
'public_transport_tickets',
'parking_tickets',
'excrement_bags',
'cigarettes',
'sweets',
'stamps',
'condoms',
'drinks;sweets',
'drinks')
THEN 'Sonstige Automaten'
WHEN amenity = 'vending_machine' AND vending like 'pub-
lic_transport_tickets' THEN 'Fahrsccheinautomat'
WHEN amenity = 'vending_machine' AND vending like 'parking_tickets'
THEN 'Parkticketautomat'
WHEN amenity = 'vending_machine' AND vending like 'excrement_bags'
THEN 'Kotbeutelautomat'
WHEN amenity = 'vending_machine' AND vending like 'cigarettes' THEN
'Zigarettenautomat'
WHEN amenity = 'vending_machine' AND vending like 'sweets' THEN
'Suessigkeitenautomat'
WHEN amenity = 'vending_machine' AND vending like 'stamps' THEN
'Briefmarkenautomat'
WHEN amenity = 'vending_machine' AND vending like 'condoms' THEN
'Kondomautomat'
WHEN amenity = 'vending_machine' AND vending like 'drinks;sweets'
THEN 'Getraenke und Suessigkeitenautomat'

```

```

WHEN amenity = 'vending_machine' AND vending like 'drinks' THEN 'Get-
raenkeautomat'
WHEN amenity LIKE 'bbq' THEN 'Grillplatz'
WHEN amenity LIKE 'recycling' AND recycling LIKE 'clothes' THEN
'Glascontainer'
WHEN amenity LIKE 'recycling' AND recycling LIKE 'glass' THEN 'Glas-
und Kleidungscontainer'
WHEN amenity LIKE 'recycling' AND recycling LIKE 'glass_clothes' THEN
'Kleidungscontainer'
WHEN amenity LIKE 'recycling' AND recycling not in
('clothes','glass','glass_clothes') THEN 'Sonstige Container'
WHEN amenity = 'restaurant' AND cuisine like 'italian' THEN 'Restau-
rant Italiener'
WHEN amenity = 'restaurant' AND cuisine like 'pizza' THEN 'Restaurant
Italiener'
WHEN amenity = 'restaurant' AND cuisine like 'mediterranean' THEN
'Restaurant Italiener'
WHEN amenity = 'restaurant' AND cuisine like 'german' THEN 'Restau-
rant Deutsch'
WHEN amenity = 'restaurant' AND cuisine like 'regional' THEN 'Restau-
rant Deutsch'
WHEN amenity = 'restaurant' AND cuisine like 'indian' THEN 'Restau-
rant Indisch'
WHEN amenity = 'restaurant' AND cuisine like 'asian' THEN 'Restaurant
Asiatisch'
WHEN amenity = 'restaurant' AND cuisine like 'vietnamese' THEN 'Res-
taurant Asiatisch'
WHEN amenity = 'restaurant' AND cuisine like 'chinese' THEN 'Restau-
rant Asiatisch'
WHEN amenity = 'restaurant' AND cuisine like 'thai' THEN 'Restaurant
Asiatisch'
WHEN amenity = 'restaurant' AND cuisine like 'korean' THEN 'Restau-
rant Asiatisch'
WHEN amenity = 'restaurant' AND cuisine like 'sushi' THEN 'Restaurant
Sushi'
WHEN amenity = 'restaurant' AND cuisine like 'japanese' THEN 'Restau-
rant Sushi'
WHEN amenity = 'restaurant' AND cuisine like 'greek' THEN 'Restaurant
Griechisch'
WHEN amenity = 'restaurant' AND cuisine like 'steak_house' THEN 'Res-
taurant Steakhouse'
WHEN amenity = 'restaurant' AND cuisine like 'mexican' THEN 'Restau-
rant Steakhouse'
WHEN amenity = 'restaurant' AND cuisine like 'burger' THEN 'Restau-
rant Steakhouse'
WHEN amenity = 'restaurant' AND cuisine like 'international' THEN
'Restaurant International'
WHEN amenity = 'restaurant' AND cuisine like 'french' THEN 'Restau-
rant International'
WHEN amenity = 'restaurant' AND cuisine like 'spanish' THEN 'Restau-
rant International'
WHEN amenity = 'restaurant' AND cuisine like 'turkish' THEN 'Restau-
rant Tuerkisch'
WHEN amenity = 'restaurant' AND cuisine like 'kebab' THEN 'Restaurant
Tuerkisch'
WHEN amenity = 'restaurant' AND cuisine like 'arab' THEN 'Restaurant
Tuerkisch'
WHEN amenity = 'restaurant' AND cuisine like 'oriental' THEN 'Restau-
rant Tuerkisch'
WHEN amenity = 'fast_food' AND cuisine like 'kebab' THEN 'Fastfood
Kebap'
WHEN amenity = 'fast_food' AND cuisine like 'turkish' THEN 'Fastfood
Kebap'

```

```

WHEN amenity = 'fast_food' AND cuisine like 'arab' THEN 'Fastfood Ke-
bab'
WHEN amenity = 'fast_food' AND cuisine like 'burger' THEN 'Fastfood
Burger'
WHEN amenity = 'fast_food' AND cuisine like 'asian' THEN 'Fastfood
Asiatisch'
WHEN amenity = 'fast_food' AND cuisine like 'chinese' THEN 'Fastfood
Asiatisch'
WHEN amenity = 'fast_food' AND cuisine like 'vietnamese' THEN
'Fastfood Asiatisch'
WHEN amenity = 'fast_food' AND cuisine like 'thai' THEN 'Fastfood
Asiatisch'
WHEN amenity = 'fast_food' AND cuisine like 'sushi' THEN 'Fastfood
Asiatisch'
WHEN amenity = 'fast_food' AND cuisine like 'pizza' THEN 'Fastfood
Pizza'
WHEN amenity = 'fast_food' AND cuisine like 'italian' THEN 'Fastfood
Pizza'
WHEN amenity = 'fast_food' AND cuisine like 'german' THEN 'Fastfood
Pommesbude'
WHEN amenity = 'fast_food' AND cuisine like 'regional' THEN 'Fastfood
Pommesbude'
WHEN amenity = 'fast_food' AND cuisine like 'sausage' THEN 'Fastfood
Pommesbude'
WHEN amenity LIKE 'fast_food' THEN 'fast_food Sonstiges'
WHEN amenity LIKE 'restaurant' THEN 'restaurant Sonstiges'
WHEN shop LIKE 'hairdresser' THEN 'Friseur'
WHEN shop LIKE 'bakery' THEN 'Baeckerei'
WHEN shop LIKE 'clothes' THEN 'Kleidung'
WHEN shop LIKE 'supermarket' THEN 'Supermarkt'
WHEN shop LIKE 'convenience' THEN 'Kiost'
WHEN shop LIKE 'florist' THEN 'Florist'
WHEN shop LIKE 'kiosk' THEN 'Kiost'
WHEN shop LIKE 'bicycle' THEN 'Fahrrad'
WHEN shop LIKE 'beauty' THEN 'Kosmetik und Beauty'
WHEN shop LIKE 'optician' THEN 'Optiker'
WHEN shop LIKE 'shoes' THEN 'Schuhe'
WHEN shop LIKE 'books' THEN 'Buecher'
WHEN shop LIKE 'massage' THEN 'Massage'
WHEN shop LIKE 'jewelry' THEN 'Schmuck'
WHEN shop LIKE 'car_repair' THEN 'Autoreperatur'
WHEN shop LIKE 'chemist' THEN 'Drogerie'
WHEN shop LIKE 'travel_agency' THEN 'Reisen'
WHEN shop LIKE 'mobile_phone' THEN 'Mobilfunk'
WHEN shop LIKE 'furniture' THEN 'Moebel'
WHEN shop LIKE 'beverages' THEN 'Getraenke'
WHEN shop LIKE 'deli' THEN 'Feinkost'
WHEN shop LIKE 'electronics' THEN 'Elektronik'
WHEN shop LIKE 'boutique' THEN 'Boutique'
WHEN shop LIKE 'car' THEN 'Autohaus'
WHEN shop LIKE 'alcohol' THEN 'Spirituosen'
WHEN shop LIKE 'gift' THEN 'Geschenke'
WHEN shop LIKE 'variety_store' THEN 'Postenmarkt'
WHEN shop LIKE 'butcher' THEN 'Schlachter'
WHEN shop LIKE 'laundry' THEN 'Waescherei'
WHEN shop LIKE 'computer' THEN 'Computer'
WHEN shop LIKE 'toys' THEN 'Spielzeug'
WHEN shop LIKE 'copyshop' THEN 'Copyshop'
WHEN shop LIKE 'tailor' THEN 'Schneider'
WHEN shop LIKE 'photo' THEN 'Foto'
WHEN shop LIKE 'art' THEN 'Kunst'
WHEN shop LIKE 'confectionery' THEN 'Suessigkeiten'
WHEN shop LIKE 'interior_decoration' THEN 'Dekoration'
WHEN shop LIKE 'dry_cleaning' THEN 'Reinigung'

```

```

WHEN shop LIKE 'greengrocer' THEN 'Feinkost'
WHEN shop LIKE 'funeral_directors' THEN 'Beerdigung'
WHEN shop LIKE 'sports' THEN 'Sport'
WHEN shop LIKE 'pet' THEN 'Zoofachgeschaef'
WHEN shop LIKE 'fabric' THEN 'Textilgeschaef'
WHEN shop LIKE 'hearing_aids' THEN 'Hoergeraete'
WHEN shop LIKE 'doityourself' THEN 'Baumarkt'
WHEN shop LIKE 'vacant' THEN 'Leerstand'
WHEN shop LIKE 'medical_supply' THEN 'Medical'
WHEN shop LIKE 'second_hand' THEN 'Second Hand'
WHEN shop LIKE 'newsagent' THEN 'Zeitung'
WHEN shop LIKE 'hardware' THEN 'Eisenwarenhandlung'
WHEN shop LIKE 'cosmetics' THEN 'Drogerie'
WHEN shop LIKE 'department_store' THEN 'Supermarkt'
WHEN shop LIKE 'tattoo' THEN 'Kosmetik und Beauty'
WHEN shop LIKE 'perfumery' THEN 'Drogerie'
WHEN shop IS NOT NULL THEN 'Sonstiger Shop'
WHEN public_transport IS NOT NULL THEN 'Haltestelle'
WHEN amenity IS NOT NULL THEN 'Sonstiges Gebaeude'
WHEN landuse IS NOT NULL THEN 'Sonstiges Gelaende'
WHEN heritage LIKE '4' THEN 'Baudenkmal'
WHEN heritage IS NOT NULL THEN 'Sonstiges Denkmal'
WHEN tourism IS NOT NULL THEN 'Sonstiges Tourismus'
WHEN sport IS NOT NULL THEN 'Sonstiges Sport'
WHEN leisure IS NOT NULL THEN 'Sonstiges Erholung'
end as poi_type
from osm_poi_2014 o;

```

Abbildung 9-9: Hive-Code zu osm_poi_type_yyyy

9.4.2 UDF lormapper

```

package de.zeb.hive.udf.lormapper;

import java.util.ArrayList;

import org.apache.hadoop.hive.ql.exec.UDF;
import org.apache.hadoop.io.Text;
import de.zeb.hive.udf.lormapper.LorArea;

public final class Planungsraum extends UDF {
    private ArrayList<LorArea> areas = null;

    public Text evaluate(final Text s) {
        if (s == null) { return null; }

        Text result = null;
        String result_helper = null;

        String helper = s.toString();
        String[] coord = helper.split(",");
        if(this.areas == null){
            System.out.println("init");
            this.areas = AreaFactory.getLorAreas("planungs-
raum_fl_neu.kml", null);
        }

        PointOfInterest point = new PointOfInterest();

```

```

        point.setLocation(Double.parseDouble(coord[0]), Double.parseDouble(coord[1]));

        for (LorArea la : areas) {
            if (la.contains(point))
                result_helper = la.getLorName();
        }

        if(result_helper == null)
            return null;

        result = new Text(result_helper);

        return result;
    }

}

```

Abbildung 9-10: Java Code zum UDF lormapper – Klasse Planungsraum

```

package de.zeb.hive.udf.lormapper;

import java.awt.Polygon;
import java.util.ArrayList;

class LorArea extends Polygon{

    private static final long serialVersionUID =
651360784392811790L;
    private String lorCoordinate;
    private String lorName;
    public static int PRECISION = 100000;

    public LorArea() {
        super();
    }

    public LorArea(String lorCoordinate, String lorName) {
        this();
        setLorCoordinate(lorCoordinate);
        setLorName(lorName);
    }

    private void setPolygonCoords(String s){
        String[] polypoints = s.split(" ");
        ArrayList<Double> lat_array = new ArrayList<Double>();
        ArrayList<Double> long_array = new ArrayList<Double>();

        for(String point : polypoints){
            String[] c = point.split(",");
            lat_array.add(Double.parseDouble(c[0]));
            long_array.add(Double.parseDouble(c[1]));
        }

        for(int i = 0; i < polypoints.length; i++)
        {
            super.addPoint((int) Math.round(lat_array.get(i)*PRECISION), (int) Math.round(long_array.get(i)*PRECISION));
        }
    }

    public String getLorCoordinate() {

```

```

        return lorCoordinate;
    }

    public void setLorCoordinate(String lorCoordinate) {
        this.lorCoordinate = lorCoordinate;
        setPolygonCoords(lorCoordinate);
    }

    public String getLorName() {
        return lorName;
    }

    public void setLorName(String lorName) {
        lorName = lorName.replaceFirst("Flaeche", "");
        lorName = lorName.replace(" ", "");
        this.lorName = lorName;
    }

}

```

Abbildung 9-11: Java Code zum UDF lormapper – Klasse LorArea

```

package de.zeb.hive.udf.lormapper;

import java.io.InputStream;
import java.util.ArrayList;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

import de.zeb.hive.udf.lormapper.LorArea;

class AreaFactory {

    /**
     * code snippets from
     * http://www.mkyong.com/java/how-to-read-xml-file-in-java-dom-parser/
     */
    static ArrayList<LorArea> getLorAreas(String filename, String filter-
prefix) {

        ArrayList<LorArea> result = new ArrayList<LorArea>();

        String area_name = "";

        try {

            DocumentBuilderFactory dbFactory = DocumentBuilder-
Factory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocument-
Builder();

            ClassLoader classLoader = Thread.cur-
rentThread().getContextClassLoader();
            InputStream stream = classLoader.getResource-
AsStream("resources/" +filename);

```

```

        Document doc = dBuilder.parse(stream);

        // optional, but recommended
        // read this -
        // http://stackoverflow.com/questions/13786607/normalization-in-dom-parsing-with-java-how-does-it-work

        doc.getDocumentElement().normalize();
        NodeList nList = doc.getElementsByTagName("Place-
mark");

        for (int temp = 0; temp < nList.getLength(); temp++)
        {

            Node nNode = nList.item(temp);

            if (nNode.getNodeType() == Node.ELEMENT_NODE) {

                Element eElement = (Element) nNode;

                LorArea area = new LorArea();

                area_name = "";
                area_name = eElement.getEle-
mentsByTagName("name").item(0).getTextContent();

                if (filterprefix == null ||
area_name.startsWith("Flaeche "+filterprefix)){
                    area.setLorName(area_name);

                    // check if area has no defined coordi-
nates
                    Node n = eElement.getEle-
mentsByTagName("coordinates").item(0);
                    if (n != null)
                        area.setLorCoordinate(eEl-
ement.getElementsByTagName("coordinates").item(0).getTextContent());

                    result.add(area);
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println(area_name);
        }

        return result;
    }

}

```

Abbildung 9-12: Java Code zum UDF lormapper – Klasse AreaFactory

```

package de.zeb.hive.udf.lormapper;

import java.awt.geom.Point2D;

class PointOfInterest extends Point2D{

    private int x;
    private int y;
}

```

```

@Override
public double getX() {
    // TODO Auto-generated method stub
    return this.x;
}
@Override
public double getY() {
    // TODO Auto-generated method stub
    return this.y;
}

@Override
public void setLocation(double y, double x) {
    this.x = (int) Math.round(x*LorArea.PRECISION);
    this.y = (int) Math.round(y*LorArea.PRECISION);
}
}

```

Abbildung 9-13: Java Code zum UDF lormapper – Klasse PointOfInterest

9.4.3 OSM POI-Status

```

CREATE TABLE osm_poi_changed AS
SELECT
    COALESCE(j2018.coords, j2017.coords, j2016.coords, j2015.coords, j2014.coords) AS coords,
    COALESCE(j2018.planungsraum, j2017.planungsraum, j2016.planungsraum, j2015.planungsraum, j2014.planungsraum) AS planungsraum,
    COALESCE(j2018.node_id, j2017.node_id, j2016.node_id, j2015.node_id, j2014.node_id) AS node_id,
    COALESCE(j2018.addr_city, j2017.addr_city, j2016.addr_city, j2015.addr_city, j2014.addr_city) AS addr_city,
    COALESCE(j2018.addr_country, j2017.addr_country, j2016.addr_country, j2015.addr_country, j2014.addr_country) AS addr_country,
    COALESCE(j2018.addr_housenumber, j2017.addr_housenumber, j2016.addr_housenumber, j2015.addr_housenumber, j2014.addr_housenumber) AS addr_housenumber,
    COALESCE(j2018.addr_postcode, j2017.addr_postcode, j2016.addr_postcode, j2015.addr_postcode, j2014.addr_postcode) AS addr_postcode,
    COALESCE(j2018.addr_street, j2017.addr_street, j2016.addr_street, j2015.addr_street, j2014.addr_street) AS addr_street,
    COALESCE(j2018.addr_suburb, j2017.addr_suburb, j2016.addr_suburb, j2015.addr_suburb, j2014.addr_suburb) AS addr_suburb,
    j2014.last_modification_time AS 2014_last_modification_time,
    j2015.last_modification_time AS 2015_last_modification_time,
    j2016.last_modification_time AS 2016_last_modification_time,
    j2017.last_modification_time AS 2017_last_modification_time,
    j2018.last_modification_time AS 2018_last_modification_time,
    j2014.last_modification_version AS 2014_last_modification_version,
    j2015.last_modification_version AS 2015_last_modification_version,
    j2016.last_modification_version AS 2016_last_modification_version,
    j2017.last_modification_version AS 2017_last_modification_version,
    j2018.last_modification_version AS 2018_last_modification_version

```

```

,j2014.name           as 2014_name
,j2015.name           as 2015_name
,j2016.name           as 2016_name
,j2017.name           as 2017_name
,j2018.name           as 2018_name
,j2014.poi_type       as 2014_poi_type
,j2015.poi_type       as 2015_poi_type
,j2016.poi_type       as 2016_poi_type
,j2017.poi_type       as 2017_poi_type
,j2018.poi_type       as 2018_poi_type
,case
  when
    levenshtein(UPPER(coalesce(j2015.name,"")),UPPER(co-
lesce(j2014.name,"")))
      -( greatest(length(coalesce(j2015.name,"")),length(UP-
PER(coalesce(j2014.name,""))))
        - least(length(coalesce(j2015.name,"")),length(UP-
PER(coalesce(j2014.name,""))))
        >= 0.1* least(length(coalesce(j2015.name,"")),length(UP-
PER(coalesce(j2014.name,""))))
        and least(length(coalesce(j2015.name,"")),length(UPPER(co-
lesce(j2014.name,"")))) > 0
      then 1
    when
      j2015.name is null and j2014.name is not null
      then 1
    when
      j2015.name is not null and j2014.name is null
      then 1
      else 0
  end as changed_2014_to_2015
,case
  when
    levenshtein(UPPER(coalesce(j2016.name,"")),UPPER(co-
lesce(j2015.name,"")))
      -( greatest(length(coalesce(j2016.name,"")),length(UP-
PER(coalesce(j2015.name,""))))
        - least(length(coalesce(j2016.name,"")),length(UP-
PER(coalesce(j2015.name,""))))
        >= 0.1* least(length(coalesce(j2016.name,"")),length(UP-
PER(coalesce(j2015.name,""))))
        and least(length(coalesce(j2016.name,"")),length(UPPER(co-
lesce(j2015.name,"")))) > 0
      then 1
    when
      j2016.name is null and j2015.name is not null
      then 1
    when
      j2016.name is not null and j2015.name is null
      then 1
      else 0
  end as changed_2015_to_2016
,case
  when
    levenshtein(UPPER(coalesce(j2017.name,"")),UPPER(co-
lesce(j2016.name,"")))
      -( greatest(length(coalesce(j2017.name,"")),length(UP-
PER(coalesce(j2016.name,""))))
        - least(length(coalesce(j2017.name,"")),length(UP-
PER(coalesce(j2016.name,""))))
        >= 0.1* least(length(coalesce(j2017.name,"")),length(UP-
PER(coalesce(j2016.name,""))))
        and least(length(coalesce(j2017.name,"")),length(UPPER(co-
lesce(j2016.name,"")))) > 0
      then 1
    when
      j2017.name is null and j2016.name is not null
      then 1
    when
      j2017.name is not null and j2016.name is null
      then 1
      else 0
  end as changed_2016_to_2017

```

```

    then 1
    when
        j2017.name is null and j2016.name is not null
    then 1
    when
        j2017.name is not null and j2016.name is null
    then 1
    else 0
end as changed_2016_to_2017
,case
    when
        levenshtein(UPPER(coalesce(j2017.name,"")),UPPER(co-
lesce(j2018.name,"")))
            -( greatest(length(coalesce(j2017.name,"")),length(UP-
PER(coalesce(j2018.name,""))))
                - least(length(coalesce(j2017.name,"")),length(UP-
PER(coalesce(j2018.name,"")))))
            >= 0.1* least(length(coalesce(j2017.name,"")),length(UP-
PER(coalesce(j2018.name,""))))
                and least(length(coalesce(j2017.name,"")),length(UPPER(coa-
lesce(j2018.name,"")))) > 0
        then 1
        when
            j2017.name is null and j2018.name is not null
        then 1
        when
            j2017.name is not null and j2018.name is null
        then 1
        else 0
end as changed_2017_to_2018
,case
    when
        levenshtein(UPPER(coalesce(j2015.name,"")),UPPER(coa-
lesce(j2014.name,"")))
            -( greatest(length(coalesce(j2015.name,"")),length(UP-
PER(coalesce(j2014.name,""))))
                - least(length(coalesce(j2015.name,"")),length(UP-
PER(coalesce(j2014.name,"")))))
            >= 0.1* least(length(coalesce(j2015.name,"")),length(UP-
PER(coalesce(j2014.name,""))))
                and least(length(coalesce(j2015.name,"")),length(UPPER(coa-
lesce(j2014.name,"")))) > 0
        then 1
        when
            j2015.name is null and j2014.name is not null
        then 1
        when
            j2015.name is not null and j2014.name is null
        then 1
        when
            levenshtein(UPPER(coalesce(j2015.name,"")),UPPER(coa-
lesce(j2016.name,"")))
                -( greatest(length(coalesce(j2015.name,"")),length(UP-
PER(coalesce(j2016.name,""))))
                    - least(length(coalesce(j2015.name,"")),length(UP-
PER(coalesce(j2016.name,"")))))
                >= 0.1* least(length(coalesce(j2015.name,"")),length(UP-
PER(coalesce(j2016.name,""))))
                    and least(length(coalesce(j2015.name,"")),length(UPPER(coa-
lesce(j2016.name,"")))) > 0
        then 1
        when
            j2015.name is null and j2016.name is not null
        then 1

```

```

when
    j2015.name is not null and j2016.name is null
then 1
when
    levenshtein(UPPER(coalesce(j2017.name,"")),UPPER(co-
lesce(j2016.name,"")))
        - (      greatest(length(coalesce(j2017.name,"")),length(UP-
PER(coalesce(j2016.name,""))))
            - least(length(coalesce(j2017.name,"")),length(UP-
PER(coalesce(j2016.name,"")))))
        >= 0.1* least(length(coalesce(j2017.name,"")),length(UP-
PER(coalesce(j2016.name,""))))
            and least(length(coalesce(j2017.name,"")),length(UPPER(coa-
lesce(j2016.name,"")))) > 0
then 1
when
    j2017.name is null and j2016.name is not null
then 1
when
    j2017.name is not null and j2016.name is null
then 1
when
    levenshtein(UPPER(coalesce(j2017.name,"")),UPPER(coa-
lesce(j2018.name,"")))
        - (      greatest(length(coalesce(j2017.name,"")),length(UP-
PER(coalesce(j2018.name,""))))
            - least(length(coalesce(j2017.name,"")),length(UP-
PER(coalesce(j2018.name,"")))))
        >= 0.1* least(length(coalesce(j2017.name,"")),length(UP-
PER(coalesce(j2018.name,""))))
            and least(length(coalesce(j2017.name,"")),length(UPPER(coa-
lesce(j2018.name,"")))) > 0
then 1
when
    j2017.name is null and j2018.name is not null
then 1
when
    j2017.name is not null and j2018.name is null
then 1
else 0
end as name_changed
from osm_poi_type_2014 j2014
full outer join osm_poi_type_2015 j2015
    on j2014.node_id = j2015.node_id
full outer join osm_poi_type_2016 j2016
    on coalesce(j2015.node_id,j2014.node_id) = j2016.node_id
full outer join osm_poi_type_2017 j2017
    on coalesce(j2016.node_id,j2015.node_id,j2014.node_id) =
j2017.node_id
full outer join osm_poi_type_2018 j2018
    on coa-
lesce(j2017.node_id,j2016.node_id,j2015.node_id,j2014.node_id) =
j2018.node_id
;

```

Abbildung 9-14: Hive-Code zu osm_poi_changed

Tabelle 9-8: Tabellenaufbau osm_poi_changed

coords	52.5057043, 13.3315992	52.61394, 13.4707006	52.5272291, 13.3988136	52.5073388, 13.3207848
--------	---------------------------	-------------------------	---------------------------	---------------------------

planungsraum	4030931	3030406	1011302	4030828
node_id	N1000710165	N1005311925	N1552074537	N26735763
addr_city	Berlin	Berlin	Berlin	Berlin
addr_country	DE	null	DE	DE
addr_housenumber	4	61	29C	106
addr_postcode	10623	13125	10119	10625
addr_street	Joachimsthaler Straße	Bahnhofstraße	Auguststraße	Pestalozzistraße
addr_suburb	Charlottenburg	null	Mitte	Charlottenburg
2014_last_modification_time	201312	201101	null	201306
2015_last_modification_time	201409	201101	201405	201409
2016_last_modification_time	null	201101	201512	201409
2017_last_modification_time	null	201101	201512	201607
2018_last_modification_time	null	201703	201706	201709
2014_last_modification_version	5	2	null	15
2015_last_modification_version	8	2	4	18
2016_last_modification_version	null	2	8	18
2017_last_modification_version	null	2	8	19

2018_last_modification_version	null	4	10	20
2014_name	China Box	Berliner Sparkasse	null	Sakana
2015_name	China Box	Berliner Sparkasse	Shiso Burger	Sakana
2016_name	null	Berliner Sparkasse	Shiso Burger	Sakana
2017_name	null	Berliner Sparkasse	Shiso Burger	Sakana
2018_name	null	Sparkasse	Shiso Burger	Lemongrass
2014_poi_type	fast_food Sonstiges	Bankfiliale	null	Restaurant Sushi
2015_poi_type	fast_food Sonstiges	Bankfiliale	Restaurant Steakhouse	Restaurant Sushi
2016_poi_type	null	Bankfiliale	Restaurant Steakhouse	Restaurant Sushi
2017_poi_type	null	Bankfiliale	Restaurant Steakhouse	Restaurant Sushi
2018_poi_type	null	Bankfiliale	Restaurant Steakhouse	Restaurant Asiatisch
chan-ged_2014_to_2015	0	0	1	0
chan-ged_2015_to_2016	1	0	0	0
chan-ged_2016_to_2017	0	0	0	0
chan-ged_2017_to_2018	0	0	0	1
name_changed	1	0	1	1

```

create view osm_poi_state_basis as
select distinct
o.node_id,
201401 as timeslice,
o.planungsraum,
case
    when o.changed_2014_to_2015 = 0 and o.changed_2015_to_2016 = 0
    and o.changed_2016_to_2017 = 0 and o.changed_2017_to_2018 = 0 and
    o.2018_poi_type is not null

```

```

        then o.2018_poi_type
        when o.changed_2014_to_2015 = 0 and o.changed_2015_to_2016 = 0
and o.changed_2016_to_2017 = 0 and o.2017_poi_type is not null
        then o.2017_poi_type
        when o.changed_2014_to_2015 = 0 and o.changed_2015_to_2016 = 0
and o.2016_poi_type is not null
        then o.2016_poi_type
        when o.changed_2014_to_2015 = 0 and o.2015_poi_type is not null
        then o.2015_poi_type
        else o.2014_poi_type
end as poi_type,
case
        when o.changed_2014_to_2015 = 0 and o.changed_2015_to_2016 = 0
and o.changed_2016_to_2017 = 0 and o.changed_2017_to_2018 = 0 and
o.2018_name is not null
        then o.2018_name
        when o.changed_2014_to_2015 = 0 and o.changed_2015_to_2016 = 0
and o.changed_2016_to_2017 = 0 and o.2017_name is not null
        then o.2017_name
        when o.changed_2014_to_2015 = 0 and o.changed_2015_to_2016 = 0
and o.2016_name is not null
        then o.2016_name
        when o.changed_2014_to_2015 = 0 and o.2015_name is not null
        then o.2015_name
        else o.2014_name
end as name,
case
        when o.2014_last_modification_version is not null and
o.2014_last_modification_time < 201301
        then "steady"
        when o.2014_last_modification_version is not null and
o.2014_last_modification_time >= 201301 and o.2014_last_modifica-
tion_version > 1
        then "steady"
        when o.2014_last_modification_version is not null and
o.2014_last_modification_time >= 201301 and o.2014_last_modifica-
tion_version = 1
        then "new"
        else null
end as poi_state
from osm_poi_changed o
union all
select distinct
o.node_id,
201501 as timeslice,
o.planungsraum,
case
        when o.changed_2015_to_2016 = 0 and o.changed_2016_to_2017 = 0
and o.changed_2017_to_2018 = 0 and o.2018_poi_type is not null
        then o.2018_poi_type
        when o.changed_2015_to_2016 = 0 and o.changed_2016_to_2017 = 0
and o.2017_poi_type is not null
        then o.2017_poi_type
        when o.changed_2015_to_2016 = 0 and o.2016_poi_type is not null
        then o.2016_poi_type
        when o.2015_poi_type is not null
        then o.2015_poi_type
        else o.2014_poi_type
end as poi_type,
case
        when o.changed_2015_to_2016 = 0 and o.changed_2016_to_2017 = 0
and o.changed_2017_to_2018 = 0 and o.2018_name is not null
        then o.2018_name

```

```

        when o.changed_2015_to_2016 = 0 and o.changed_2016_to_2017 = 0
and o.2017_name is not null
        then o.2017_name
        when o.changed_2015_to_2016 = 0 and o.2016_name is not null
        then o.2016_name
        when o.2015_name is not null
        then o.2015_name
        else o.2014_name
end as name,
case
        when o.2014_last_modification_version is null and
o.2015_last_modification_version is not null
        then "new"
        when o.2014_last_modification_version is not null and
o.2015_last_modification_version is null
        then "deleted"
        when o.2014_last_modification_version is not null and
o.2015_last_modification_version is not null and
o.changed_2014_to_2015 = 0
        then "steady"
        when o.2014_last_modification_version is not null and
o.2015_last_modification_version is not null and
o.changed_2014_to_2015 = 1
        then "changed"
        else null
end as poi_state
from osm_poi_changed o
union all
select distinct
o.node_id,
201601 as timeslice,
o.planungsraum,
case
        when o.changed_2016_to_2017 = 0 and o.changed_2017_to_2018 = 0
and o.2018_poi_type is not null
        then o.2018_poi_type
        when o.changed_2016_to_2017 = 0 and o.2017_poi_type is not null
        then o.2017_poi_type
        when o.2016_poi_type is not null
        then o.2016_poi_type
        else o.2015_poi_type
end as poi_type,
case
        when o.changed_2016_to_2017 = 0 and o.changed_2017_to_2018 = 0
and o.2018_name is not null
        then o.2018_name
        when o.changed_2016_to_2017 = 0 and o.2017_name is not null
        then o.2017_name
        when o.2016_name is not null
        then o.2016_name
        else o.2015_name
end as name,
case
        when o.2015_last_modification_version is null and
o.2016_last_modification_version is not null
        then "new"
        when o.2015_last_modification_version is not null and
o.2016_last_modification_version is null
        then "deleted"
        when o.2015_last_modification_version is not null and
o.2016_last_modification_version is not null and
o.changed_2015_to_2016 = 0
        then "steady"

```

```

    when o.2015_last_modification_version is not null and
o.2016_last_modification_version is not null and
o.changed_2015_to_2016 = 1
        then "changed"
        else null
end as poi_state
from osm_poi_changed o
union all
select distinct
o.node_id,
201701 as timeslice,
o.planungsraum,
case
    when o.changed_2017_to_2018 = 0 and o.2018_poi_type is not null
        then o.2018_poi_type
    when o.2017_poi_type is not null
        then o.2017_poi_type
        else o.2016_poi_type
end as poi_type,
case
    when o.changed_2017_to_2018 = 0 and o.2018_name is not null
        then o.2018_name
    when o.2017_name is not null
        then o.2017_name
        else o.2016_name
end as name,
case
    when o.2016_last_modification_version is null and
o.2017_last_modification_version is not null
        then "new"
    when o.2016_last_modification_version is not null and
o.2017_last_modification_version is null
        then "deleted"
    when o.2016_last_modification_version is not null and
o.2017_last_modification_version is not null and
o.changed_2016_to_2017 = 0
        then "steady"
    when o.2016_last_modification_version is not null and
o.2017_last_modification_version is not null and
o.changed_2016_to_2017 = 1
        then "changed"
        else null
end as poi_state
from osm_poi_changed o
union all
select distinct
o.node_id,
201801 as timeslice,
o.planungsraum,
case
    when o.2018_poi_type is not null
        then o.2018_poi_type
        else o.2017_poi_type
end as poi_type,
case
    when o.2018_name is not null
        then o.2018_name
        else o.2017_name
end as name,
case
    when o.2017_last_modification_version is null and
o.2018_last_modification_version is not null
        then "new"

```

```

    when o.2017_last_modification_version is not null and
o.2018_last_modification_version is null
      then "deleted"
    when o.2017_last_modification_version is not null and
o.2018_last_modification_version is not null and
o.changed_2017_to_2018 = 0
      then "steady"
    when o.2017_last_modification_version is not null and
o.2018_last_modification_version is not null and
o.changed_2017_to_2018 = 1
      then "changed"
      else null
end as poi_state
from osm_poi_changed o;

```

Abbildung 9-15: Hive-Code zu osm_poi_state_basis

```

create view osm_poi_state_changed_special as
select
  act.node_id,
  act.timeslice,
  act.planungsraum,
  act.poi_type,
  act.name,
  case
    when act.poi_state = 'changed'
        and prev.name is null
        and act.poi_type = prev.poi_type
      then "steady"
    when act.poi_state = 'changed'
        and prev.name = act.name
        and act.poi_type = prev.poi_type
      then "steady"
    else act.poi_state
  end as poi_state
from osm_poi_state_basis act
  left join osm_poi_state_basis prev
    on act.node_id = prev.node_id
    and act.timeslice - 100 = prev.timeslice;

```

Abbildung 9-16: Hive-Code zu osm_poi_state_changed_special

```

create view osm_poi_state_changed_del as
select
  act.node_id,
  act.timeslice,
  act.planungsraum,
  prev.poi_type,
  prev.name,
  "changed_deleted" as poi_state
from osm_poi_state_changed_special act
  left join osm_poi_state_changed_special prev
    on act.node_id = prev.node_id
    and act.timeslice - 100 = prev.timeslice
where act.poi_state = 'changed';

```

Abbildung 9-17: Hive-Code zu osm_poi_state_changed_del

```

create table osm_poi_state as
select
  act.node_id,
  act.timeslice,
  act.planungsraum,
  act.poi_type,
  act.name,

```

```

act.poi_state
from osm_poi_state_changed_del act
union all
select
  act.node_id,
  act.timeslice,
  act.planungsraum,
  act.poi_type,
  act.name,
  act.poi_state
from osm_poi_state_changed_special act
;

```

Abbildung 9-18: Hive-Code zu osm_poi_state

9.4.4 OSM Featureentwicklung und Aggregation

```

CREATE view osm_poi_features_type as
  select
    o.planungsraum,
    o.timeslice,
    case
      when o.poi_state = 'changed' then 'new'
      when o.poi_state = 'changed_deleted' then 'deleted'
      else o.poi_state
    end as poi_state,
    o.poi_type,
    p.category,
    p.domain,
    count(1) as anz
  from osm_poi_state o
  join osm_poi_mapping p on o.poi_type = p.type
  where o.poi_state is not null
  group by
    o.planungsraum,
    o.timeslice,
    case
      when o.poi_state = 'changed' then 'new'
      when o.poi_state = 'changed_deleted' then 'deleted'
      else o.poi_state
    end,
    o.poi_type,
    p.category,
    p.domain
union all
  select
    o.planungsraum,
    o.timeslice,
    'stock' as poi_state,
    o.poi_type,
    p.category,
    p.domain,
    count(1) as anz
  from osm_poi_state o
  join osm_poi_mapping p on o.poi_type = p.type
  where o.poi_state in ('changed', 'new', 'steady')
  group by
    o.planungsraum,
    o.timeslice,
    o.poi_type,
    p.category,
    p.domain

```

```

union all
  select
    o.planungsraum,
    o.timeslice,
    'ytd' as poi_state,
    o.poi_type,
    p.category,
    p.domain,
    sum(if(poi_state in ('deleted','changed_deleted'), -1,
if(poi_state in ('new','changed'), 1,0))) as anz
  from osm_poi_state o
  join osm_poi_mapping p on o.poi_type = p.type
  group by
    o.planungsraum,
    o.timeslice,
    o.poi_type,
    p.category,
    p.domain
;

```

Abbildung 9-19: Hive-Code zu osm_poi_features_type

Tabelle 9-9: Tabellenaufbau osm_poi_features_type

planungs- raum	times- lice	poi_state	poi_type	category	domain	anz
1011302	201501	deleted	Bar	Gaststaetten	Vergnue-gung	4
1011302	201501	new	Bar	Gaststaetten	Vergnue-gung	2
1011302	201501	steady	Bar	Gaststaetten	Vergnue-gung	24
1011302	201501	stock	Bar	Gaststaetten	Vergnue-gung	26
1011302	201501	ytd	Bar	Gaststaetten	Vergnue-gung	-2
1011302	201501	deleted	Restau-rant Itali-ener	Restaurant	Gastrono-mie	3
1011302	201501	new	Restau-rant Itali-ener	Restaurant	Gastrono-mie	5
1011302	201501	steady	Restau-rant Itali-ener	Restaurant	Gastrono-mie	14
1011302	201501	stock	Restau-rant Itali-ener	Restaurant	Gastrono-mie	19
1011302	201501	ytd	Restau-rant Itali-ener	Restaurant	Gastrono-mie	2

9.4.5 EWR und MSS Daten

```

DROP table lor_ewr_data;
CREATE table lor_ewr_data as
select
ee.zeit as zeit, if(length(ee.raumid)=7,concat(0,ee.raumid),ee.raumid)
as planungsraum
,wd.dau5, wd.dau10
,ee.e_e, ee.e_em, ee.e_ew
,( 0.5 * ee.E_E00_01 +
1.5 * ee.E_E01_02 +
2.5 * ee.E_E02_03 +
4 * ee.E_E03_05 +
5.5 * ee.E_E05_06 +
6.5 * ee.E_E06_07 +
7.5 * ee.E_E07_08 +
9 * ee.E_E08_10 +
11 * ee.E_E10_12 +
13 * ee.E_E12_14 +
14.5 * ee.E_E14_15 +
16.5 * ee.E_E15_18 +
19.5 * ee.E_E18_21 +
23 * ee.E_E21_25 +
26 * ee.E_E25_27 +
28.5 * ee.E_E27_30 +
32.5 * ee.E_E30_35 +
37.5 * ee.E_E35_40 +
42.5 * ee.E_E40_45 +
47.5 * ee.E_E45_50 +
52.5 * ee.E_E50_55 +
57.5 * ee.E_E55_60 +
61.5 * ee.E_E60_63 +
64 * ee.E_E63_65 +
66 * ee.E_E65_67 +
68.5 * ee.E_E67_70 +
72.5 * ee.E_E70_75 +
77.5 * ee.E_E75_80 +
82.5 * ee.E_E80_85 +
87.5 * ee.E_E85_90 +
92.5 * ee.E_E90_95 +
102.5 * ee.E_E95_110 ) / ee.E_E as E_EALTER
,ee.E_U1+ee.E_1U6+ee.E_6U15+ee.E_15U18 as E_EU1U18
,ee.E_E18_21+ee.E_E21_25+ee.E_E25_27 as E_E18U27
,ee.E_E27_30+ee.E_E30_35+ee.E_E35_40+ee.E_E40_45 as E_E27U45
,ee.E_E18_21+ee.E_E21_25+ee.E_E25_27+ee.E_E27_30+ee.E_E30_35 as
E_E18U35
,ee.E_E35_40+ee.E_E40_45 as E_E35U45
,ee.E_E45_50+ee.E_E50_55 as E_E45U55
,ee.E_E55U65 as E_E55U65
,ee.E_E65U80+ee.E_E80U110 as E_E65U110
,ea.e_a, ea.e_am, ea.e_aw
,( 0.5 * ea.E_A00_01 +
1.5 * ea.E_A01_02 +
2.5 * ea.E_A02_03 +
4 * ea.E_A03_05 +
5.5 * ea.E_A05_06 +
6.5 * ea.E_A06_07 +
7.5 * ea.E_A07_08 +
9 * ea.E_A08_10 +
11 * ea.E_A10_12 +
13 * ea.E_A12_14 +
14.5 * ea.E_A14_15 +
16.5 * ea.E_A15_18 +
19.5 * ea.E_A18_21 +

```

```

23 * ea.E_A21_25 +
26 * ea.E_A25_27 +
28.5 * ea.E_A27_30 +
32.5 * ea.E_A30_35 +
37.5 * ea.E_A35_40 +
42.5 * ea.E_A40_45 +
47.5 * ea.E_A45_50 +
52.5 * ea.E_A50_55 +
57.5 * ea.E_A55_60 +
61.5 * ea.E_A60_63 +
64 * ea.E_A63_65 +
66 * ea.E_A65_67 +
68.5 * ea.E_A67_70 +
72.5 * ea.E_A70_75 +
77.5 * ea.E_A75_80 +
82.5 * ea.E_A80_85 +
87.5 * ea.E_A85_90 +
92.5 * ea.E_A90_95 +
102.5 * ea.E_A95_110 ) / ea.e_a as E_AALTER
,ea.E_AU1+ea.E_A1U6+ea.E_A6U15+ea.E_A15U18 as E_AU1U18
,ea.E_A18_21+ea.E_A21_25+ea.E_A25_27 as E_A18U27
,ea.E_A27_30+ea.E_A30_35+ea.E_A35_40+ea.E_A40_45 as E_A27U45
,ea.E_A45_50+ea.E_A50_55 as E_A45U55
,ea.E_A55U65
,ea.E_A65U80+ea.E_A80U110 as E_A65U110
,eme.mh_e ,eme.mh_em, eme.mh_ew
,( 0.5 * eme.MH_E00_01 +
1.5 * eme.MH_E01_02 +
2.5 * eme.MH_E02_03 +
4 * eme.MH_E03_05 +
5.5 * eme.MH_E05_06 +
6.5 * eme.MH_E06_07 +
7.5 * eme.MH_E07_08 +
9 * eme.MH_E08_10 +
11 * eme.MH_E10_12 +
13 * eme.MH_E12_14 +
14.5 * eme.MH_E14_15 +
16.5 * eme.MH_E15_18 +
19.5 * eme.MH_E18_21 +
23 * eme.MH_E21_25 +
26 * eme.MH_E25_27 +
28.5 * eme.MH_E27_30 +
32.5 * eme.MH_E30_35 +
37.5 * eme.MH_E35_40 +
42.5 * eme.MH_E40_45 +
47.5 * eme.MH_E45_50 +
52.5 * eme.MH_E50_55 +
57.5 * eme.MH_E55_60 +
61.5 * eme.MH_E60_63 +
64 * eme.MH_E63_65 +
66 * eme.MH_E65_67 +
68.5 * eme.MH_E67_70 +
72.5 * eme.MH_E70_75 +
77.5 * eme.MH_E75_80 +
82.5 * eme.MH_E80_85 +
87.5 * eme.MH_E85_90 +
92.5 * eme.MH_E90_95 +
102.5 * eme.MH_E95_110 ) / eme.MH_E as E_MHALTER
,eme.MH_U1+eme.MH_1U6+eme.MH_6U15+eme.MH_15U18 as MH_EU1U18
,eme.MH_E18_21+eme.MH_E21_25+eme.MH_E25_27 as MH_E18U27
,eme.MH_E27_30+eme.MH_E30_35+eme.MH_E35_40+eme.MH_E40_45 as MH_E27U45
,eme.MH_E45_50+eme.MH_E50_55 as MH_E45U55
,eme.MH_55U65 as MH_E55U65
,eme.MH_65U80+eme.MH_80U110 as MH_E65U110

```

```

,emh.hk_eu15, emh.hk_eu27, emh.hk_polen, emh.hk_ehejug, emh.hk_ehesu,
emh.hk_turk, emh.hk_arab, emh.hk_sonst, emh.hk_nzord
from lor_ewr_e ee
left join lor_ewr_a ea on 1=1
and ea.raumid = ee.raumid
and substr(ea.zeit,0,4) = substr(ee.zeit,0,4)
left join lor_whndauer wd on 1=1
and wd.raumid = ee.raumid
and substr(wd.zeit,0,4) = substr(ee.zeit,0,4)
left join lor_ewr_migra_e eme on 1=1
and eme.raumid = ee.raumid
and substr(eme.zeit,0,4) = substr(ee.zeit,0,4)
left join lor_ewr_migra_h emh on 1=1
and emh.raumid = ee.raumid
and substr(emh.zeit,0,4) = substr(ee.zeit,0,4)
;

```

Abbildung 9-20: Hive-Code zu lor_ewr_data

Tabelle 9-10: Tabellenaufbau lor_ewr_data

zeit	201012	201112	201212	201312	201412	201512	201612
planungs- raum	101130 2						
dau5	5.491	5.811	6.179	6.285	6.378	6.710	6.971
dau10	2.976	3.100	3.298	3.388	3.486	3.825	4.130
e_e	11.808	11.879	12.078	12.118	12.137	12.281	12.535
e_em	6.022	6.074	6.188	6.166	6.193	6.284	6.457
e_ew	5.786	5.805	5.890	5.952	5.944	5.997	6.078
e_ealter	36,6	36,8	37,0	37,3	37,5	37,7	37,9
e_eu1u18	1.780	1.822	1.850	1.890	1.898	1.919	1.957
e_e18u27	1.352	1.318	1.312	1.259	1.225	1.161	1.191
e_e27u45	5.228	5.124	5.142	4.993	4.965	5.015	5.006
e_e18u35	3.754	3.668	3.679	3.580	3.544	3.547	3.550
e_e35u45	2.826	2.774	2.775	2.672	2.646	2.629	2.647
e_e45u55	1.812	1.948	2.048	2.176	2.187	2.238	2.365
e_e55u65	832	852	897	951	998	1.034	1.062
e_e65u110	804	815	829	849	864	914	954

e_a	2.185	2.313	2.532	2.662	2.803	3.047	3.246
e_am	1.052	1.136	1.227	1.305	1.392	1.541	1.656
e_aw	1.133	1.177	1.305	1.357	1.411	1.506	1.590
e_aalter	34,4	34,7	34,9	35,2	35,3	35,3	35,5
e_au1u18	140	142	148	149	166	198	209
e_a18u27	432	454	486	500	492	489	498
e_a27u45	1.241	1.309	1.421	1.473	1.587	1.769	1.898
e_a45u55	232	252	314	363	369	365	413
e_a55u65	101	97	106	117	113	132	130
e_a65u110	39	59	57	60	76	94	98
mh_e	3.350	3.512	3.791	4.009	4.147	4.439	4.699
mh_em	1.638	1.744	1.857	1.999	2.076	2.250	2.400
mh_ew	1.712	1.768	1.934	2.010	2.071	2.189	2.299
e_mhalter	30,9	31,1	31,3	31,7	31,7	32,0	32,4
mh_eu1u18	687	718	752	793	816	867	867
mh_e18u27	517	526	566	588	577	575	609
mh_e27u45	1.602	1.674	1.795	1.879	2.001	2.176	2.321
mh_e45u55	339	376	443	491	503	527	602
mh_e55u65	133	130	154	175	171	185	181
mh_e65u110	72	88	81	83	79	109	119
hk_eu15	1.112	1.227	1.355	1.413	1.561	1.656	1.702
hk_eu27	1.466	1.586	1.753	1.865	1.989	2.169	2.256
hk_polen	149	144	165	179	180	208	216
hk_ehejug	111	101	120	83	89	95	92
hk_ehesu	280	283	294	338	333	366	385

hk_turk	155	165	160	168	178	187	189
hk_arab	110	86	97	114	148	192	238
hk_sonst	1.130	1.190	1.272	1.340	1.288	1.361	1.461
hk_nzord	98	101	95	101	122	69	78

```

drop view lor_own_idx_k11;
create view lor_own_idx_k11 as
select
s.id as raum_id, s.name as raum_desc, concat(s.jahr-1,'12') as zeit
, s.k11, a.k11_stddev, a.k11_avg, (s.k11-a.k11_avg) / a.k11_stddev as
k11_msr
from zeitreihe_mss_k11 s
join
(
select jahr
, stddev_pop(k11) as k11_stddev
, avg(k11) as k11_avg
from zeitreihe_mss_k11
where k11 > 0
group by jahr
) a on a.jahr = s.jahr
where s.k11 > 0
;

drop view lor_own_idx_d2;
create view lor_own_idx_d2 as
select
s.raum_id, s.raum_desc, s.zeit
, s.d2, a.d2_stddev, a.d2_avg, (s.d2-a.d2_avg) / a.d2_stddev as d2_msr
from lor_mss_idx s
join
(
select zeit
, stddev_pop(d2) as d2_stddev
, avg(d2) as d2_avg
from lor_mss_idx
where s1 > 0 or s2 > 0 or s3 > 0 or s4 > 0
group by zeit
) a on a.zeit = s.zeit
where s1 > 0 or s2 > 0 or s3 > 0 or s4 > 0
;

drop table lor_ewr_calc_base;
create table lor_ewr_calc_base as
select
curr.planungsraum as raum_id,
null as raum_desc,
curr.zeit,
curr.dau5 - prev.dau5 as dau5_ytd,
curr.dau10 - prev.dau10 as dau10_ytd,
curr.ea - prev.ea as ea_ytd,
curr.mh - prev.mh as mh_ytd,
curr.ee_18u35 - prev.ee_18u35 as ee_18u35_ytd,
curr.ee_35u45 - prev.ee_35u45 as ee_35u45_ytd
from lor_ewr_einwohnergewichtet_tb curr
;
```

```

left join lor_ewr_einwohnergewichtet_tb prev
on curr.planungsraum = prev.planungsraum and curr.zeit =
(prev.zeit+200)
where 1=1;

drop view lor_own_idx_ewr;
create view lor_own_idx_ewr as
select
s.raum_id, s.raum_desc, s.zeit
,s.dau5_ytd, a.dau5_ytd_stddev, a.dau5_ytd_avg
,(s.dau5_ytd-a.dau5_ytd_avg) / a.dau5_ytd_stddev as dau5_msr
,s.dau10_ytd, a.dau10_ytd_stddev, a.dau10_ytd_avg
,(s.dau10_ytd-a.dau10_ytd_avg) / a.dau10_ytd_stddev as dau10_msr
,s.ea_ytd, a.ea_ytd_stddev, a.ea_ytd_avg
,(s.ea_ytd-a.ea_ytd_avg) / a.ea_ytd_stddev as ea_msr
,s.mh_ytd, a.mh_ytd_stddev, a.mh_ytd_avg
,(s.mh_ytd-a.mh_ytd_avg) / a.mh_ytd_stddev as mh_msr
,s.ee_18u35_ytd, a.ee_18u35_ytd_stddev, a.ee_18u35_ytd_avg
,(s.ee_18u35_ytd-a.ee_18u35_ytd_avg) / a.ee_18u35_ytd_stddev as
ee_18u35_msr
,s.ee_35u45_ytd, a.ee_35u45_ytd_stddev, a.ee_35u45_ytd_avg
,(s.ee_35u45_ytd-a.ee_35u45_ytd_avg) / a.ee_35u45_ytd_stddev as
ee_35u45_msr
from lor_ewr_calc_base s
join
(
select zeit
, stddev_pop(dau5_ytd) as dau5_ytd_stddev
, avg(dau5_ytd) as dau5_ytd_avg
, stddev_pop(dau10_ytd) as dau10_ytd_stddev
, avg(dau10_ytd) as dau10_ytd_avg
, stddev_pop(ea_ytd) as ea_ytd_stddev
, avg(ea_ytd) as ea_ytd_avg
, stddev_pop(mh_ytd) as mh_ytd_stddev
, avg(mh_ytd) as mh_ytd_avg
, stddev_pop(ee_18u35_ytd) as ee_18u35_ytd_stddev
, avg(ee_18u35_ytd) as ee_18u35_ytd_avg
, stddev_pop(ee_35u45_ytd) as ee_35u45_ytd_stddev
, avg(ee_35u45_ytd) as ee_35u45_ytd_avg
from lor_ewr_calc_base
where raum_id in (select raum_id from lor_mss_idx where s1 > 0 or s2 >
0 or s3 > 0 or s4 > 0)
group by zeit
) a on a.zeit = s.zeit
where s.raum_id in (select raum_id from lor_mss_idx where s1 > 0 or s2 >
0 or s3 > 0 or s4 > 0);

drop view lor_own_idx_plr;
create view lor_own_idx_plr as
select
d2.raum_id, d2.raum_desc, d2.zeit as jahr,
round(k11.k11_msr * 2, 0) as k11_msr_points,
round(ewr.dau5_msr * -1 * 2, 0) as dau5_msr_points,
round(ewr.dau10_msr * -1 * 2, 0) as dau10_msr_points,
round(k11.k11_msr * 2, 0) +
round(ewr.dau5_msr * -1 * 2, 0) +
round(ewr.dau10_msr * -1 * 2, 0) as doering_ulbricht_idx_m_points,
k11.k11_msr +
ewr.dau5_msr * -1 +
ewr.dau10_msr * -1 as doering_ulbricht_idx_m_value,
round(d2.d2_msr * -1 * 2, 0) as d2_msr_points,
round(ewr.ea_msr * -1 * 2, 0) as ea_msr_points,
round(ewr.mh_msr * -1 * 2, 0) as mh_msr_points,

```

```

round(ewr.ee_18u35_msr * 2, 0)    as ee_18u35_msr_points,
round(ewr.ee_35u45_msr * 2, 0)    as ee_35u45_msr_points,
round(d2.d2_msr * -1 * 2, 0)      +
round(ewr.ea_msr * -1 * 2, 0)      +
round(ewr.mh_msr * -1 * 2, 0)      +
round(ewr.ee_18u35_msr * 2, 0)      +
round(ewr.ee_35u45_msr * 2, 0)    as doering_ulbricht_idx_b_points,
d2.d2_msr * -1      +
ewr.ea_msr * -1      +
ewr.mh_msr * -1      +
ewr.ee_18u35_msr      +
ewr.ee_35u45_msr    as doering_ulbricht_idx_b_value,
round(d2.d2_msr * -1 * 2, 0)      +
round(k11.k11_msr * 2, 0)      +
round(ewr.dau5_msr * -1 * 2, 0)  +
round(ewr.dau10_msr * -1 * 2, 0) +
round(ewr.ea_msr * -1 * 2, 0)    +
round(ewr.mh_msr * -1 * 2, 0)    +
round(ewr.ee_18u35_msr * 2, 0)    +
round(ewr.ee_35u45_msr * 2, 0)    as sum_idx_points,
d2.d2_msr * -1      +
k11.k11_msr      +
ewr.dau5_msr * -1  +
ewr.dau10_msr * -1 +
ewr.ea_msr * -1    +
ewr.mh_msr * -1    +
ewr.ee_18u35_msr  +
ewr.ee_35u45_msr  as sum_idx_value
from lor_own_idx_d2 d2
join lor_own_idx_k11 k11
on cast(d2.raum_id as int) = cast(k11.raum_id as int) and d2.zeit =
k11.zeit
left join lor_own_idx_ewr ewr
on cast(d2.raum_id as int) = cast(ewr.raum_id as int) and d2.zeit =
ewr.zeit;

drop table lor_own_idx_plr_tb;
create table lor_own_idx_plr_tb as select * from lor_own_idx_plr;

```

Abbildung 9-21: Hive-Code der Strecke zu lor_own_idx_plr_tb

```

DROP view lor_mss_idx_bzr_z;
CREATE view lor_mss_idx_bzr_z as
select
raum_id
,raum_desc
,s.zeit
,ew
,(s1 - s1_avg) / s1_stddev as zs1
,(s2 - s2_avg) / s2_stddev as zs2
,(s3 - s3_avg) / s3_stddev as zs3
,(s4 - s4_avg) / s4_stddev as zs4
,(
((s1 - s1_avg) / s1_stddev)+
((s2 - s2_avg) / s2_stddev)+
((s3 - s3_avg) / s3_stddev)+
((s4 - s4_avg) / s4_stddev)
) as status_summe
,null as status_index
,null as status_klasse
,(d1 - d1_avg) / d1_stddev as zd1
,(d2 - d2_avg) / d2_stddev as zd2

```

```

,(d3 - d3_avg) / d3_stddev as zd3
,(d4 - d4_avg) / d4_stddev as zd4
,(
((d1 - d1_avg) / d1_stddev)+
((d2 - d2_avg) / d2_stddev)+
((d3 - d3_avg) / d3_stddev)+
((d4 - d4_avg) / d4_stddev)
) as dynamik_summe
,null as dynamik_index
,null as dynamik_klasse
,s1
,s2
,s3
,s4
,d1
,d2
,d3
,d4
from lor_mss_idx_bzr_sum s
join
(
select zeit
, stddev_pop(s1) as s1_stddev
, stddev_pop(s2) as s2_stddev
, stddev_pop(s3) as s3_stddev
, stddev_pop(s4) as s4_stddev
, avg(s1) as s1_avg
, avg(s2) as s2_avg
, avg(s3) as s3_avg
, avg(s4) as s4_avg
, stddev_pop(d1) as d1_stddev
, stddev_pop(d2) as d2_stddev
, stddev_pop(d3) as d3_stddev
, stddev_pop(d4) as d4_stddev
, avg(d1) as d1_avg
, avg(d2) as d2_avg
, avg(d3) as d3_avg
, avg(d4) as d4_avg
from lor_mss_idx_bzr_sum
group by zeit
) a on a.zeit = s.zeit
;

```

Abbildung 9-22: Hive-Code zu lor_mss_idx_bzr_z

```

DROP table lor_mss_idx_bzr_idx;
CREATE table lor_mss_idx_bzr_idx as
select
raum_id
,raum_desc
,s.zeit
,ew
,zs1
,zs2
,zs3
,zs4
,status_summe
,(status_summe - status_avg) / status_stddev as status_index
,case
    when ((status_summe - status_avg) / status_stddev) < -1 then
"hoch"
    when -1 <= ((status_summe - status_avg) / status_stddev) <= 1
then "mittel"

```

```

        when 1 <= ((status_summe - status_avg) / status_stddev) <= 1.5
then "niedrig"
    when ((status_summe - status_avg) / status_stddev) > 1.5 then
"sehr niedrig"
    else null
end as status_klasse
,zd1
,zd2
,zd3
,zd4
,dynamik_summe
,(dynamik_summe - dynamik_avg) / dynamik_stddev as dynamik_index
,case
    when ((dynamik_summe - dynamik_avg) / dynamik_stddev) < -1 then
"positiv"
    when -1 <= ((dynamik_summe - dynamik_avg) / dynamik_stddev) <= 1
then "stabil"
    when ((dynamik_summe - dynamik_avg) / dynamik_stddev) > 1 then
"negativ"
    else null
end as dynamik_klasse
,s1
,s2
,s3
,s4
,d1
,d2
,d3
,d4
from lor_mss_idx_bzr_z s
join
(
select zeit
    ,stddev_pop(status_summe) as status_stddev
    ,stddev_pop(dynamik_summe) as dynamik_stddev
    ,avg(status_summe) as status_avg
    ,avg(dynamik_summe) as dynamik_avg
from lor_mss_idx_bzr_z
group by zeit
) a on a.zeit = s.zeit
)

```

Abbildung 9-23: Hive-Code zu lor_mss_idx_bzr_idx

9.4.6 Ergebnistabelle

```

DROP view lor_ewr_einwohnergewichtet;
CREATE view lor_ewr_einwohnergewichtet as
select
    d.zeit           as zeit
    ,substr(d.planungsraum,0,8)      as planungsraum
    ,sum(d.e_e)       as ee
    ,sum(d.dau5)      ) / sum(d.e_e)      as dau5
    ,sum(d.dau10)     ) / sum(d.e_e)      as dau10
    ,sum(d.e_em)      ) / sum(d.e_e)      as ee_m
    ,sum(d.e_ew)      ) / sum(d.e_e)      as ee_w
    ,sum(d.e_ealter * d.e_e) / sum(d.e_e) as ee_alter
    ,sum(d.e_eulu18)   ) / sum(d.e_e)      as ee_u1u18
    ,sum(d.e_e18u27)   ) / sum(d.e_e)      as ee_18u27
    ,sum(d.e_e27u45)   ) / sum(d.e_e)      as ee_27u45
    ,sum(d.e_e18u35)   ) / sum(d.e_e)      as ee_18u35
    ,sum(d.e_e35u45)   ) / sum(d.e_e)      as ee_35u45
    ,sum(d.e_e45u55)   ) / sum(d.e_e)      as ee_45u55
    ,sum(d.e_e55u65)   ) / sum(d.e_e)      as ee_55u65
)

```

```

,sum(d.e_e65u110 ) / sum(d.e_e)      as ee_65u110
,sum(d.e_a        ) / sum(d.e_e)      as ea
,sum(d.e_am       ) / sum(d.e_e)      as ea_m
,sum(d.e_aw       ) / sum(d.e_e)      as ea_w
,sum(d.e_aalter * d.e_a ) / sum(d.e_a) as ea_alter
,sum(d.e_aulu18   ) / sum(d.e_e)      as ea_uulu18
,sum(d.e_a18u27   ) / sum(d.e_e)      as ea_18u27
,sum(d.e_a27u45   ) / sum(d.e_e)      as ea_27u45
,sum(d.e_a45u55   ) / sum(d.e_e)      as ea_45u55
,sum(d.e_a55u65   ) / sum(d.e_e)      as ea_55u65
,sum(d.e_a65u110  ) / sum(d.e_e)      as ea_65u110
,sum(d.mh_e        ) / sum(d.e_e)      as mh
,sum(d.mh_em       ) / sum(d.e_e)      as mh_m
,sum(d.mh_ew       ) / sum(d.e_e)      as mh_w
,sum(d.e_mhalter * d.mh_e ) / sum(d.mh_e) as mh_alter
,sum(d.mh_eulu18  ) / sum(d.e_e)      as mh_uulu18
,sum(d.mh_e18u27  ) / sum(d.e_e)      as mh_18u27
,sum(d.mh_e27u45  ) / sum(d.e_e)      as mh_27u45
,sum(d.mh_e45u55  ) / sum(d.e_e)      as mh_45u55
,sum(d.mh_e55u65  ) / sum(d.e_e)      as mh_55u65
,sum(d.mh_e65u110 ) / sum(d.e_e)      as mh_65u110
,sum(d.hk_eu15     ) / sum(d.e_e)      as hk_eu15
,sum(d.hk_eu27     ) / sum(d.e_e)      as hk_eu27
,sum(d.hk_polen   ) / sum(d.e_e)      as hk_polen
,sum(d.hk_ehejug  ) / sum(d.e_e)      as hk_ehejug
,sum(d.hk_ehesu   ) / sum(d.e_e)      as hk_ehesu
,sum(d.hk_turk    ) / sum(d.e_e)      as hk_turk
,sum(d.hk_arab    ) / sum(d.e_e)      as hk_arab
,sum(d.hk_sonst   ) / sum(d.e_e)      as hk_sonst
,sum(d.hk_nzord   ) / sum(d.e_e)      as hk_nzord
from lor_ewr_data d
group by d.zeit, substr(d.planungsraum,0,8)
-- substr anpassen für andere Raumauflösung
;

DROP table lor_ewr_einwohnergewichtet_tb;
CREATE table lor_ewr_einwohnergewichtet_tb as
select * from lor_ewr_einwohnergewichtet;

```

Abbildung 9-24: Hive-Code zu lor_ewr_einwohnergewichtet

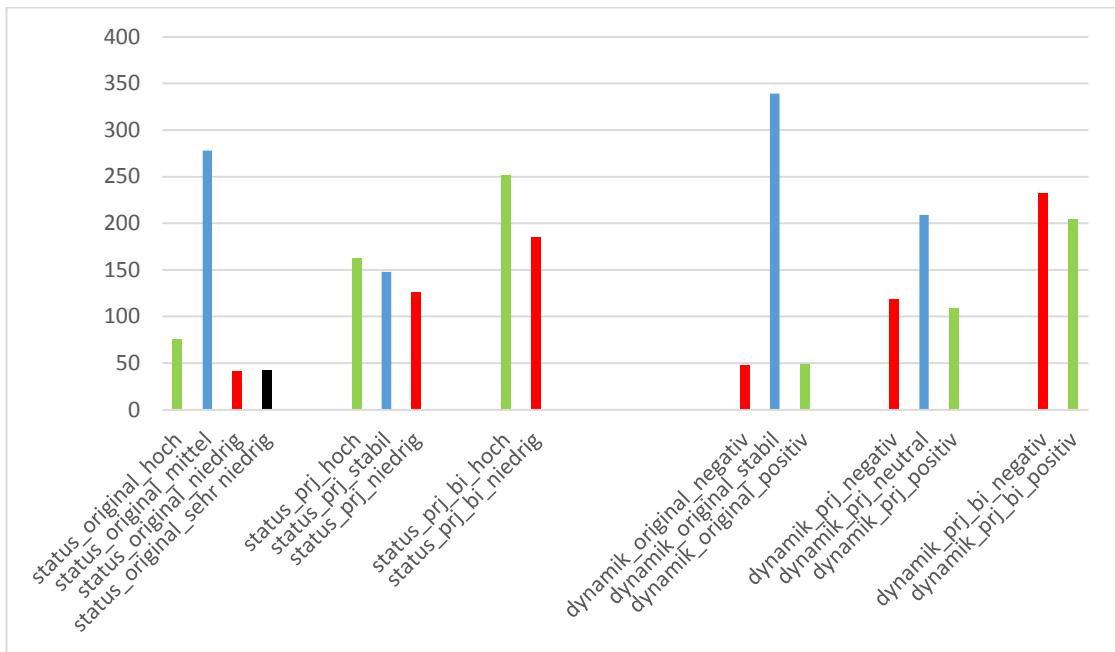


Abbildung 9-25: MSS Vergleich Klassengrößen

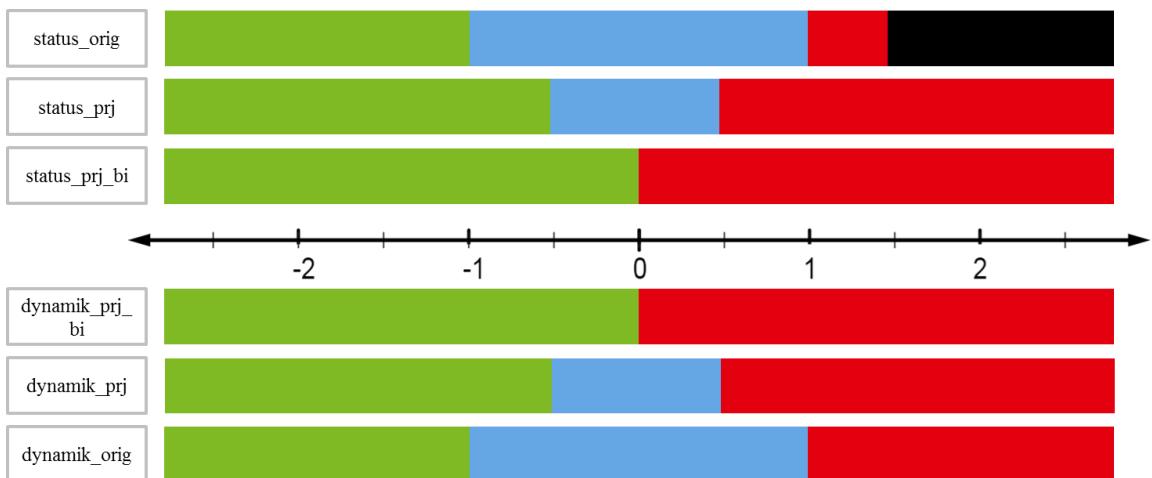


Abbildung 9-26: MSS Klassengrenzen

```

drop table result_full_plr;
create table result_full_plr as

select
case
when mss.status_index > 0.5 then "niedrig"
when mss.status_index < -0.5 then "hoch"
else "stabil"
end as status_klasse_prj,
case
when mss.dynamik_index > 0.5 then "negativ"
when mss.dynamik_index < -0.5 then "positiv"
else "neutral"
end as dynamik_klasse_prj,
case
when mss.status_index > 0 then "niedrig"
when mss.status_index < -0.5 then "hoch"
else "stabil"
end as status_klasse_dynamik,
case
when mss.dynamik_index > 0 then "negativ"
when mss.dynamik_index < -0.5 then "positiv"
else "neutral"
end as dynamik_klasse_dynamik
end

```

```

when mss.status_index <= 0 then "hoch"
else "stabil"
end as status_klasse_prj.bi,
case
when mss.dynamik_index > 0 then "negativ"
when mss.dynamik_index <= 0 then "positiv"
else "neutral"
end as dynamik_klasse_prj.bi,
mss.raum_id
,mss.raum_desc
,mss.zeit
,mss.ew
,mss.zs1
,mss.zs2
,mss.zs3
,mss.zs4
,mss.status_summe
,mss.status_index
,mss.status_klasse
,mss.zd1
,mss.zd2
,mss.zd3
,mss.zd4
,mss.dynamik_summe
,mss.dynamik_index
,mss.dynamik_klasse
,mss.s1
,mss.s2
,mss.s3
,mss.s4
,mss.d1
,mss.d2
,mss.d3
,mss.d4
,own_idx.k11_msr_points
,own_idx.dau5_msr_points
,own_idx.dau10_msr_points
,own_idx.doering_ulbricht_idx_m_points
,own_idx.doering_ulbricht_idx_m_value
,own_idx.d2_msr_points
,own_idx.ea_msr_points
,own_idx.mh_msr_points
,own_idx.ee_18u35_msr_points
,own_idx.ee_35u45_msr_points
,own_idx.doering_ulbricht_idx_b_points
,own_idx.doering_ulbricht_idx_b_value
,own_idx.sum_idx_points
,own_idx.sum_idx_value
,ewr.ee
,ewr.dau5
,ewr.dau10
,ewr.ee_m
,ewr.ee_w
,ewr.ee_alter
,ewr.ee_u1u18
,ewr.ee_18u27
,ewr.ee_27u45
,ewr.ee_18u35
,ewr.ee_35u45
,ewr.ee_45u55
,ewr.ee_55u65
,ewr.ee_65u110
,ewr.ea
,ewr.ea_m

```

```

,ewr.ea_w
,ewr.ea_alter
,ewr.ea_u1u18
,ewr.ea_18u27
,ewr.ea_27u45
,ewr.ea_45u55
,ewr.ea_55u65
,ewr.ea_65u110
,ewr.mh
,ewr.mh_m
,ewr.mh_w
,ewr.mh_alter
,ewr.mh_u1u18
,ewr.mh_18u27
,ewr.mh_27u45
,ewr.mh_45u55
,ewr.mh_55u65
,ewr.mh_65u110
,ewr.hk_eu15
,ewr.hk_eu27
,ewr.hk_polen
,ewr.hk_ehejug
,ewr.hk_ehesu
,ewr.hk_turk
,ewr.hk_arab
,ewr.hk_sonst
,ewr.hk_nzord,
[...]
coalesce((d_gastronomie_stock / (ewr.ee/1000)),0) as d_gastronomie_stock,
[...]
coalesce((d_gastronomie_ytd / (ewr.ee/1000)),0) as d_gastronomie_ytd,
[...]
coalesce((d_gastronomie_new / ewr.ee/1000),0) as d_gastronomie_new,
[...]
from lor_ewr_einwohnergewichtet_tb ewr
join lor_mss_idx mss
  on ewr.zeit = mss.zeit
  and cast(ewr.planungsraum as int) = mss.raum_id
join lor_own_idx_plr_tb own_idx
  on ewr.zeit = own_idx.jahr
  and cast(ewr.planungsraum as int) = cast(own_idx.raum_id as int)
[...]
left join
(select
  planungsraum,
  201612 as timeslice,
[...]
  d_gastronomie_stock,
[...]
  from osm_poi_features_domain_piv
  where timeslice in (201701)
) stock_d
  on ewr.zeit = stock_d.timeslice
  and cast(ewr.planungsraum as int) = cast(stock_d.planungsraum as int)
left join
(select
  planungsraum,
  201612 as timeslice,
[...]
  SUM(d_gastronomie_ytd) as d_gastronomie_ytd,
[...]
  SUM(d_gastronomie_new) as d_gastronomie_new,

```

```

[...]
from osm_poi_features_domain_piv
where timeslice in (201601, 201701)
group by planungsraum
) dyn_d
    on ewr.zeit = dyn_d.timeslice
    and cast(ewr.planungsraum as int) = cast(dyn_d.planungsraum as
int)
where ewr.zeit = 201612
and mss.dynamik_klasse not like "%"
;

```

Abbildung 9-27: Hive-Code zu result_full_plr (abgekürzt)

9.4.7 Distanzberechnung

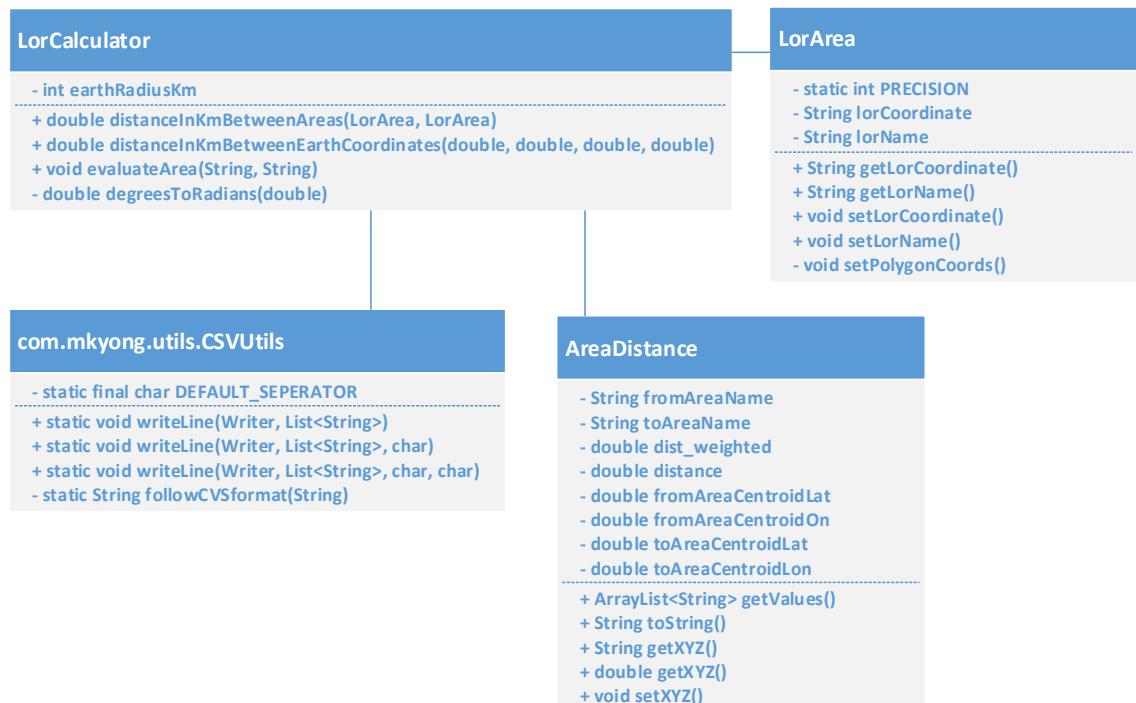


Abbildung 9-28: Klassendiagramm Distanzberechnung

```

package de.zeb.hive.udf.lormapper;

import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;

public class LorCalculator {

    // snippets from
    // https://stackoverflow.com/questions/365826/calculate-distance-between-2-gps-coordinates

    private int earthRadiusKm = 6371;

    private double degreesToRadians(double degrees) {
        return degrees * Math.PI / 180;
    }
}

```

```

    public double distanceInKmBetweenEarthCoordinates(double lat1,
double lon1, double lat2, double lon2) {
    double dLat = degreesToRadians(lat2 - lat1);
    double dLon = degreesToRadians(lon2 - lon1);

    lat1 = degreesToRadians(lat1);
    lat2 = degreesToRadians(lat2);

    double a = Math.sin(dLat / 2) * Math.sin(dLat / 2)
            + Math.sin(dLon / 2) * Math.sin(dLon / 2) *
    Math.cos(lat1) * Math.cos(lat2);
    double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
    return earthRadiusKm * c;
}

public double distanceInKmBetweenAreas(LorArea a, LorArea b) {
    return this.distanceInKmBetweenEarthCoordinates(a.getBounds2D().getCenterX(),
a.getBounds2D().getCenterY(),
b.getBounds2D().getCenterX(), b.getBounds2D().getCenterY());
}

public void evaluateArea(String kmlFileName, String outputFileName) {
    ArrayList<LorArea> areas = null;
    ArrayList<AreaDistance> result = new ArrayList<AreaDistance>();

    String result_helperA = null;
    String result_helperB = null;

    double midAX = 0;
    double midAY = 0;
    double midBX = 0;
    double midBY = 0;

    areas = AreaFactory.getLorAreas(kmlFileName, null);

    for (LorArea la : areas) {

        result_helperA = la.getLorName();
        midAX = la.getBounds2D().getCenterX() / LorArea.PRECISION;
        midAY = la.getBounds2D().getCenterY() / LorArea.PRECISION;
        for (LorArea la2 : areas) {
            result_helperB = la2.getLorName();
            midBX = la2.getBounds2D().getCenterX() / LorArea.PRECISION;
            midBY = la2.getBounds2D().getCenterY() / LorArea.PRECISION;
            double dist = distanceInKmBetweenEarthCoordinates(midAY, midAX, midBY, midBX);
            result.add(new AreaDistance(result_helperA, midAY, midAX, result_helperB, midBY, midBX, dist));
        }
    }

    String csvFile = outputFileName;
    FileWriter writer;
    try {
        writer = new FileWriter(csvFile);

```

```
        CSVUtils.writeLine(writer, Arrays.asList("fromAreaName", "fromAreaCentroidLat", "fromAreaCentroidLon", "toAreaName", "toAreaCentroidLat", "toAreaCentroidLon", "distance", "dist_weighted"));

    for(AreaDistance ad : result){
        CSVUtils.writeLine(writer, ad.getValues());
        System.out.println(ad.toString());
    }

    writer.flush();
    writer.close();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}
```

Abbildung 9-29: Java Code zur Distanzberechnung – Klasse LorCalculator

```
package de.zeb.hive.udf.lormapper;

import java.util.ArrayList;

public class AreaDistance {

    private String fromAreaName;
    private double fromAreaCentroidLat;
    private double fromAreaCentroidLon;
    private String toAreaName;
    private double toAreaCentroidLat;
    private double toAreaCentroidLon;
    private double distance;
    private double dist_weighted;

    public AreaDistance(String fromAreaName, double fromAreaCentroidLat, double fromAreaCentroidLon, String toAreaName, double toAreaCentroidLat, double toAreaCentroidLon, double distance) {
        super();
        this.fromAreaName = fromAreaName;
        this.fromAreaCentroidLat = fromAreaCentroidLat;
        this.fromAreaCentroidLon = fromAreaCentroidLon;
        this.toAreaName = toAreaName;
        this.toAreaCentroidLat = toAreaCentroidLat;
        this.toAreaCentroidLon = toAreaCentroidLon;
        this.distance = distance;
        this.dist_weighted = Math.max(Math.pow(3,distance*-1)-0.1*distance,0);
    }

    @Override
    public String toString() {
        return "AreaDistance [fromAreaName=" + fromAreaName + ", fromAreaCentroidLat=" + fromAreaCentroidLat + ", fromAreaCentroidLon=" + fromAreaCentroidLon + ", toAreaName=" + toAreaName + ", toAreaCentroidLat=" + toAreaCentroidLat + ", toAreaCentroidLon=" + toAreaCentroidLon + ", distance=" + distance + ", dist_weighted=" + dist_weighted + "]";
    }
}
```

```

public ArrayList<String> getValues() {
    ArrayList<String> result = new ArrayList<String>();

    result.add(fromAreaName);
    result.add(fromAreaCentroidLat+"");
    result.add(fromAreaCentroidLon+"");
    result.add(toAreaName);
    result.add(toAreaCentroidLat+"");
    result.add(toAreaCentroidLon+"");
    result.add(distance+"");
    result.add(dist_weighted+"");

    return result;
}

public String getFromAreaName() {
    return fromAreaName;
}

public void setFromAreaName(String fromAreaName) {
    this.fromAreaName = fromAreaName;
}

public double getFromAreaCentroidLat() {
    return fromAreaCentroidLat;
}

public void setFromAreaCentroidLat(double fromAreaCentroidLat) {
    this.fromAreaCentroidLat = fromAreaCentroidLat;
}

public double getFromAreaCentroidLon() {
    return fromAreaCentroidLon;
}

public void setFromAreaCentroidLon(double fromAreaCentroidLon) {
    this.fromAreaCentroidLon = fromAreaCentroidLon;
}

public String getToAreaName() {
    return toAreaName;
}

public void setToAreaName(String toAreaName) {
    this.toAreaName = toAreaName;
}

public double getToAreaCentroidLat() {
    return toAreaCentroidLat;
}

public void setToAreaCentroidLat(double toAreaCentroidLat) {
    this.toAreaCentroidLat = toAreaCentroidLat;
}

public double getToAreaCentroidLon() {
    return toAreaCentroidLon;
}

public void setToAreaCentroidLon(double toAreaCentroidLon) {
    this.toAreaCentroidLon = toAreaCentroidLon;
}

```

```

    public double getDistance() {
        return distance;
    }

    public void setDistance(double distance) {
        this.distance = distance;
    }

    public double getDist_weighted() {
        return dist_weighted;
    }

    public void setDist_weighted(double dist_weighted) {
        this.dist_weighted = dist_weighted;
    }

}

```

Abbildung 9-30: Java Code zur Distanzberechnung – Klasse AreaDistance

Tabelle 9-11: Tabellenaufbau lor_dist_planungsraum

fromareaname	toareaname	distance	dist_weighted
1011201	1011201	-	100,0%
1011201	1011203	0,73	37,8%
1011201	1011204	0,91	27,7%
1011201	1011202	0,93	26,7%
1011201	2010101	0,97	24,9%
1011201	2010102	1,20	14,6%
1011201	1011105	1,34	9,6%
1011201	1011104	1,47	5,2%
1011201	2020201	1,64	0,1%
1011201	2020202	1,75	0,0%
...

```

[...]
select planungsraum,
       timeslice,
       domain,
       collect(poi_state, anz) as group_map
from (
    select
        fromareaname as planungsraum,
        timeslice,
        domain,
        poi_state,
        sum(anz * dist_weighted) as anz
    from osm_poi_features_domain osm
        join lor_dist_planungsraum lor
            on lor.toareaname = osm.planungsraum
            where dist_weighted > 0
    group by fromareaname,
            timeslice,
            domain,
            poi_state
) w

```

```
group by planungsraum,
timeslice,
domain
[...]
```

Abbildung 9-31: Hive-Code zu osm_poi_features_domain_piv_distcalc (Auszug)

9.5 Modellbildung und –evaluation

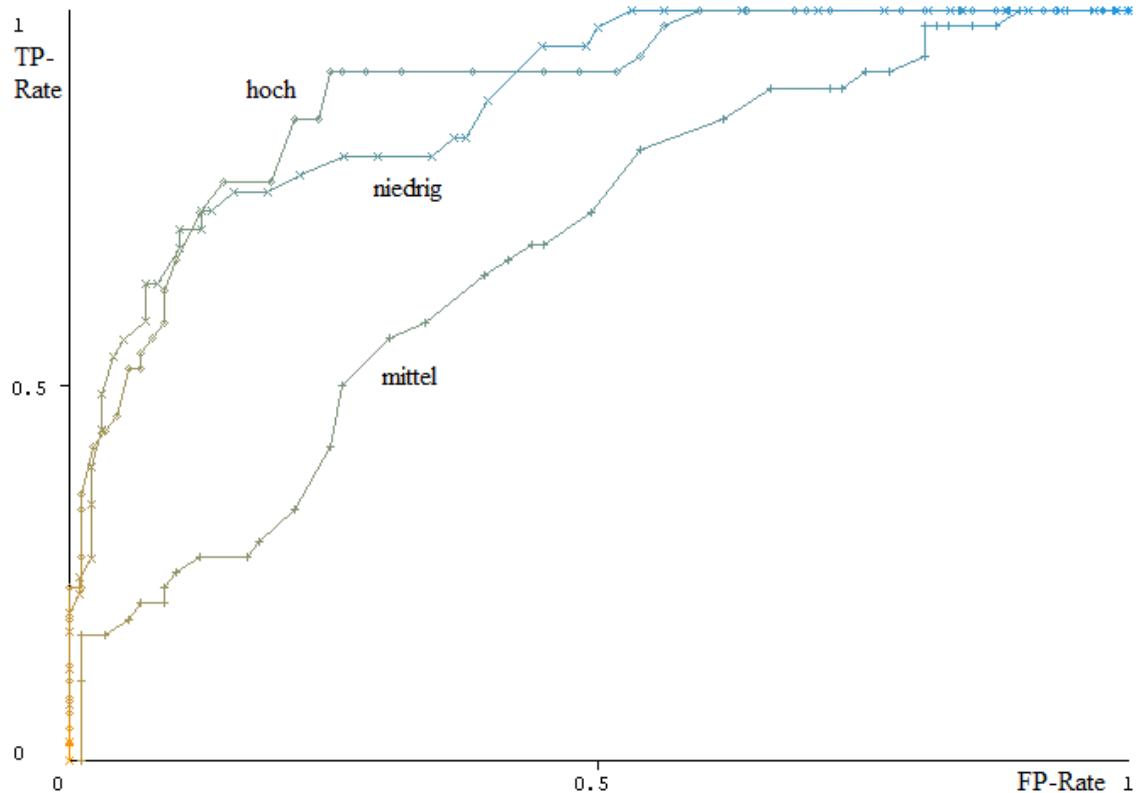


Abbildung 9-32: ROC Beispiel

Ausgewählte Algorithmen

Kalkuliert:

```
[1],'rules.ZeroR
[2],'trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1\
[3],'trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1 -depth 5\
[4],'trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1 -depth 3\
[5],'trees.LMT '-I 1 -M 15 -W 0.0\
[6],'trees.LMT '-I 3 -M 15 -W 0.0'
```

```
[8],'meta.LogitBoost '-P 100 -L -1.7976931348623157E308 -H 1.0 -Z 3.0 -O 1 -E 1 -S 1
-I 10 -W trees.DecisionStump\''
[9],'functions.SimpleLogistic '-I 0 -M 500 -H 50 -W 0.0\''
[10],'functions.SimpleLogistic '-I 3 -M 500 -H 50 -W 0.0\''
[11],'bayes.BayesNet '-D -Q bayes.net.search.local.K2 -- -P 1 -S BAYES -E
bayes.net.estimate.SimpleEstimator -- -A 0.5\''
[12],'bayes.NaiveBayes \''
[13],'trees.J48 '-C 0.25 -M 2\''
[14],'meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.J48 -- -C 0.25 -M 2\''
[15],'meta.Stacking '-X 10 -M \\\"trees.J48 -C 0.25 -M 2\\\" -S 1 -num-slots 1 -B
\\\"trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1\\\" -B
\\\"trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1 -depth 5\\\" -B
\\\"trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1 -depth
3\\\" -B \\\"trees.LMT -I 3 -M 15 -W 0.0\\\" -B \\\"functions.SimpleLogistic -I 0 -M 500
-H 50 -W 0.0\\\" -B \\\"functions.SimpleLogistic -I 3 -M 500 -H 50 -W 0.0\\\" -B
\\\"trees.LMT -I -1 -M 15 -W 0.0\\\"'
[16],'meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W functions.SimpleLogistic -- -I 0 -
M 500 -H 50 -W 0.0\''
[18],'meta.Vote '-S 1 -B \\\"functions.SimpleLogistic -I 3 -M 500 -H 50 -W 0.0\\\" -B
\\\"functions.SimpleLogistic -I 0 -M 500 -H 50 -W 0.0\\\" -B \\\"trees.RandomForest -P
100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1 -depth 3\\\" -B \\\"trees.RandomForest
-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1 -depth 5\\\" -B \\\"trees.LMT -I -1
-M 15 -W 0.0\\\" -B \\\"trees.LMT -I 3 -M 15 -W 0.0\\\" -B \\\"trees.RandomForest -P
100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1\\\" -R AVG\''
[19],'functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H 20\''
```

Nicht kalkuliert wegen Abbrüchen:

```
[7],'meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W functions.SimpleLogistic -- -I 0 -M 500 -H
50 -W 0.0\''
[17],'meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W functions.SimpleLogistic -- -I 3 -
M 500 -H 50 -W 0.0\''
```

Abbildung 9-33: Ausgewählte Algorithmen

Tabelle 9-12: Algorithmus AUC & F1

Algorithmus	AUC				F1			
	B	C	D	F	B	C	D	F
02 - RandomForest	2	7		1	2	7		1
03 - RandomForest_5	2	6	1	1	3	6	1	
04 - RandomForest_3	2	6	1	1	2	7	1	
05 - LMT	2	2	4	2	3	4	3	
06 - LMT_3	1	4	4	1	2	4	4	
08 - LogitBoost		5	4	1	2	6	2	
09 - SimpleLogistic	2	1	5	2	2	4	4	
10 - SimpleLogistic_3	1	4	4	1	1	4	4	1
11 - BayesNet	1	4	3	2	1	7	1	1
12 - NaiveBayes		2	2	6		3	4	3
13 - J48			6	4		4	4	2
14 - AdaBoostM1		2	6	2	1	6	3	
15 - Stacking		2	6	2	2	6	2	
16 - Bagging	2	3	4	1	3	5	1	1
18 - Vote	2	5	2	1	3	5	2	
19 - MultilayerPerceptron		2	6	2	1	5	1	3
Gesamtergebnis	17	55	58	30	28	83	37	12

Measure	Dataset	[2]	[3]	[4]	[5]	[6]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[18]	[19]	MAX	Bewertung	MEAN	Bewertung
Weight.avg AUC	h1_bzr	● 0,86	● 0,85	● 0,84	● 0,86	● 0,86	● 0,78	● 0,86	● 0,87	● 0,82	● 0,70	● 0,63	● 0,76	● 0,76	● 0,82	● 0,88	● 0,73	● 0,88	AUC - B	● 0,80	AUC - B
Weight.avg AUC	h1_plr-dc	● 0,82	● 0,82	● 0,80	● 0,80	● 0,77	● 0,79	● 0,80	● 0,72	● 0,77	● 0,70	● 0,65	● 0,75	● 0,73	● 0,79	● 0,83	● 0,76	● 0,83	AUC - B	● 0,76	AUC - C
Weight.avg AUC	h2_bzr	● 0,77	● 0,73	● 0,71	● 0,62	● 0,61	● 0,60	● 0,60	● 0,63	● 0,56	● 0,59	● 0,47	● 0,56	● 0,57	● 0,70	● 0,67	● 0,67	● 0,77	AUC - C	● 0,62	AUC - D
Weight.avg AUC	h2_plr-dc	● 0,73	● 0,72	● 0,73	● 0,66	● 0,70	● 0,66	● 0,66	● 0,70	● 0,68	● 0,66	● 0,58	● 0,65	● 0,67	● 0,68	● 0,72	● 0,65	● 0,73	AUC - C	● 0,67	AUC - D
Weight.avg AUC	h3a_bzr	● 0,53	● 0,57	● 0,57	● 0,55	● 0,60	● 0,56	● 0,55	● 0,61	● 0,50	● 0,54	● 0,51	● 0,47	● 0,53	● 0,51	● 0,58	● 0,45	● 0,61	AUC - D	● 0,54	AUC - F
Weight.avg AUC	h3a_plr-dc	● 0,70	● 0,68	● 0,67	● 0,57	● 0,56	● 0,72	● 0,55	● 0,50	● 0,65	● 0,53	● 0,58	● 0,66	● 0,64	● 0,62	● 0,64	● 0,61	● 0,72	AUC - C	● 0,61	AUC - D
Weight.avg AUC	h3b_bzr	● 0,74	● 0,75	● 0,75	● 0,76	● 0,73	● 0,69	● 0,74	● 0,72	● 0,69	● 0,67	● 0,61	● 0,66	● 0,62	● 0,81	● 0,78	● 0,68	● 0,81	AUC - B	● 0,71	AUC - C
Weight.avg AUC	h3b_plr-dc	● 0,74	● 0,74	● 0,72	● 0,65	● 0,65	● 0,66	● 0,64	● 0,67	● 0,70	● 0,59	● 0,61	● 0,67	● 0,64	● 0,67	● 0,71	● 0,58	● 0,74	AUC - C	● 0,66	AUC - D
Weight.avg AUC	h3c_bzr	● 0,75	● 0,74	● 0,72	● 0,70	● 0,75	● 0,74	● 0,69	● 0,71	● 0,74	● 0,54	● 0,60	● 0,68	● 0,63	● 0,67	● 0,75	● 0,61	● 0,75	AUC - C	● 0,68	AUC - D
Weight.avg AUC	h3c_plr-dc	● 0,73	● 0,74	● 0,73	● 0,61	● 0,68	● 0,72	● 0,63	● 0,68	● 0,70	● 0,55	● 0,66	● 0,67	● 0,68	● 0,70	● 0,71	● 0,65	● 0,74	AUC - C	● 0,67	AUC - D
Weight.avg F1	h1_bzr	● 0,77	● 0,77	● 0,73	● 0,74	● 0,76	● 0,68	● 0,74	● 0,79	● 0,71	● 0,68	● 0,65	● 0,68	● 0,78	● 0,74	● 0,79	● 0,69	● 0,79	F1 - B	● 0,73	F1 - B
Weight.avg F1	h1_plr-dc	● 0,74	● 0,72	● 0,71	● 0,73	● 0,71	● 0,73	● 0,73	● 0,66	● 0,68	● 0,60	● 0,66	● 0,70	● 0,70	● 0,71	● 0,75	● 0,72	● 0,75	F1 - B	● 0,70	F1 - B
Weight.avg F1	h2_bzr	● 0,65	● 0,63	● 0,64	● 0,63	● 0,56	● 0,57	● 0,61	● 0,59	● 0,50	● 0,55	● 0,49	● 0,50	● 0,57	● 0,62	● 0,59	● 0,65	● 0,65	F1 - C	● 0,58	F1 - D
Weight.avg F1	h2_plr-dc	● 0,69	● 0,68	● 0,67	● 0,64	● 0,66	● 0,62	● 0,64	● 0,66	● 0,62	● 0,59	● 0,57	● 0,59	● 0,67	● 0,64	● 0,67	● 0,62	● 0,69	F1 - C	● 0,63	F1 - C
Weight.avg F1	h3a_bzr	● 0,49	● 0,52	● 0,52	● 0,52	● 0,56	● 0,52	● 0,52	● 0,58	● 0,45	● 0,55	● 0,47	● 0,50	● 0,57	● 0,40	● 0,55	● 0,42	● 0,58	F1 - D	● 0,51	F1 - D
Weight.avg F1	h3a_plr-dc	● 0,64	● 0,63	● 0,63	● 0,57	● 0,58	● 0,67	● 0,55	● 0,44	● 0,62	● 0,49	● 0,57	● 0,61	● 0,64	● 0,58	● 0,61	● 0,49	● 0,67	F1 - C	● 0,58	F1 - D
Weight.avg F1	h3b_bzr	● 0,65	● 0,67	● 0,67	● 0,70	● 0,65	● 0,64	● 0,68	● 0,63	● 0,66	● 0,60	● 0,59	● 0,61	● 0,66	● 0,72	● 0,65	● 0,66	● 0,72	F1 - B	● 0,66	F1 - C
Weight.avg F1	h3b_plr-dc	● 0,68	● 0,67	● 0,65	● 0,62	● 0,57	● 0,60	● 0,59	● 0,59	● 0,65	● 0,55	● 0,60	● 0,62	● 0,64	● 0,63	● 0,64	● 0,41	● 0,68	F1 - C	● 0,60	F1 - D
Weight.avg F1	h3c_bzr	● 0,67	● 0,69	● 0,64	● 0,66	● 0,68	● 0,70	● 0,65	● 0,68	● 0,69	● 0,41	● 0,55	● 0,64	● 0,61	● 0,63	● 0,71	● 0,61	● 0,71	F1 - B	● 0,64	F1 - C
Weight.avg F1	h3c_plr-dc	● 0,67	● 0,70	● 0,69	● 0,58	● 0,65	● 0,66	● 0,59	● 0,59	● 0,67	● 0,42	● 0,63	● 0,62	● 0,67	● 0,65	● 0,52	● 0,70	F1 - B	● 0,61	F1 - C	

Abbildung 9-34: Übersicht Evaluation der Algorithmen

9.6 Geographische Abbildungen

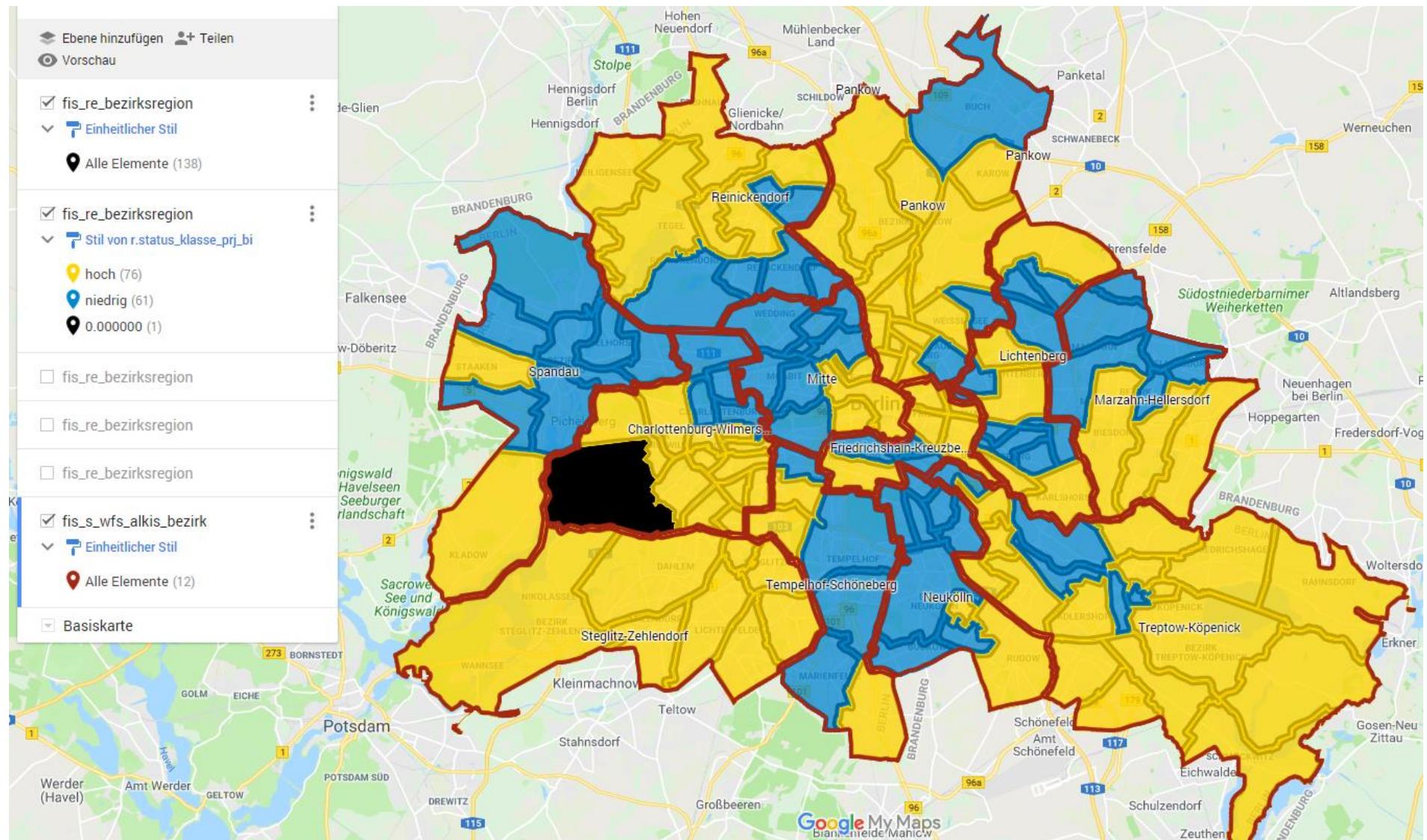


Abbildung 9-35: Übersicht Berlin in Bezirksregionen 2016: MSS-Status (Eigene Darstellung mit GoogleMyMaps)

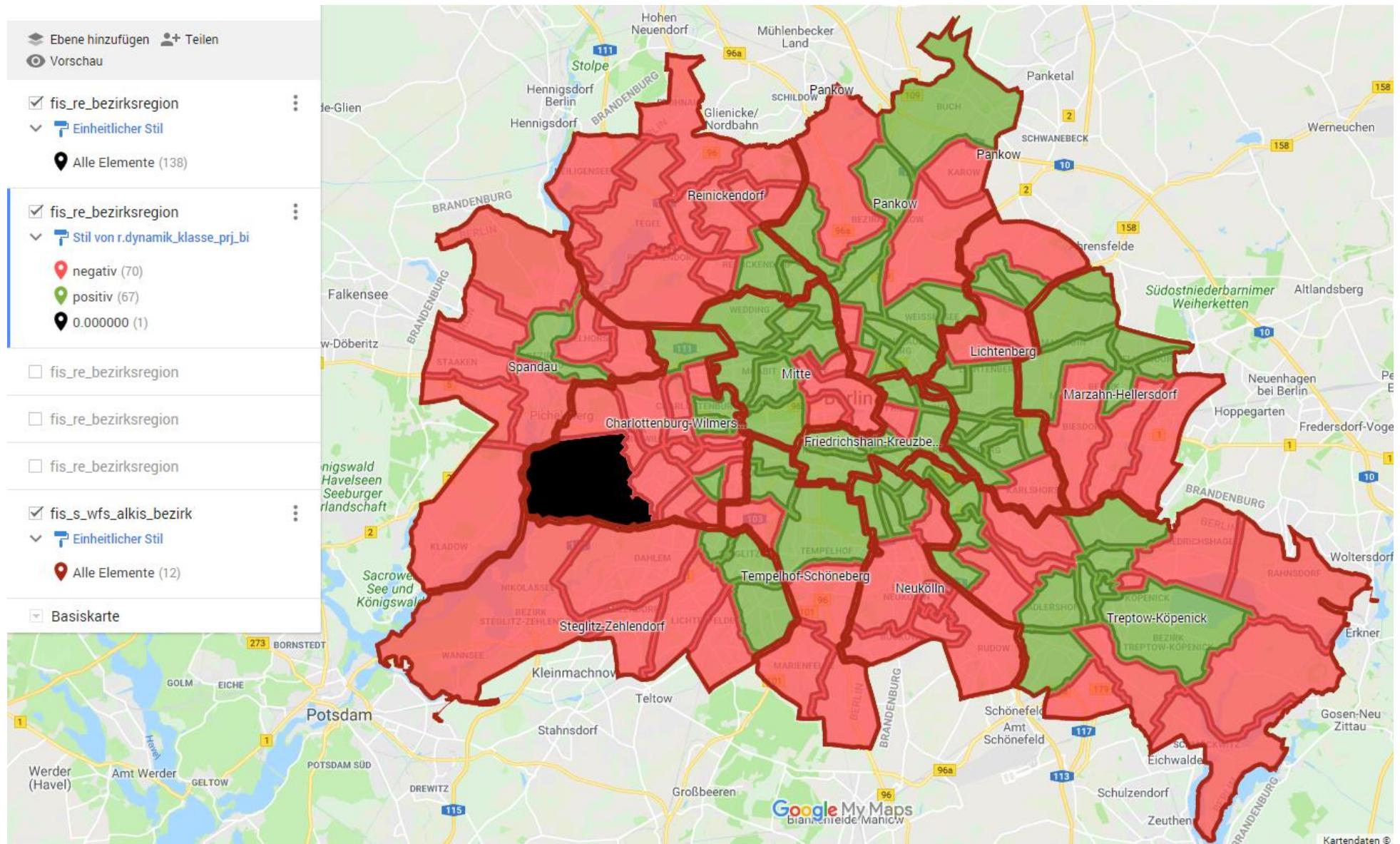


Abbildung 9-36: Übersicht Berlin in Bezirksregionen 2016: MSS-Dynamik (Eigene Darstellung mit GoogleMyMaps)

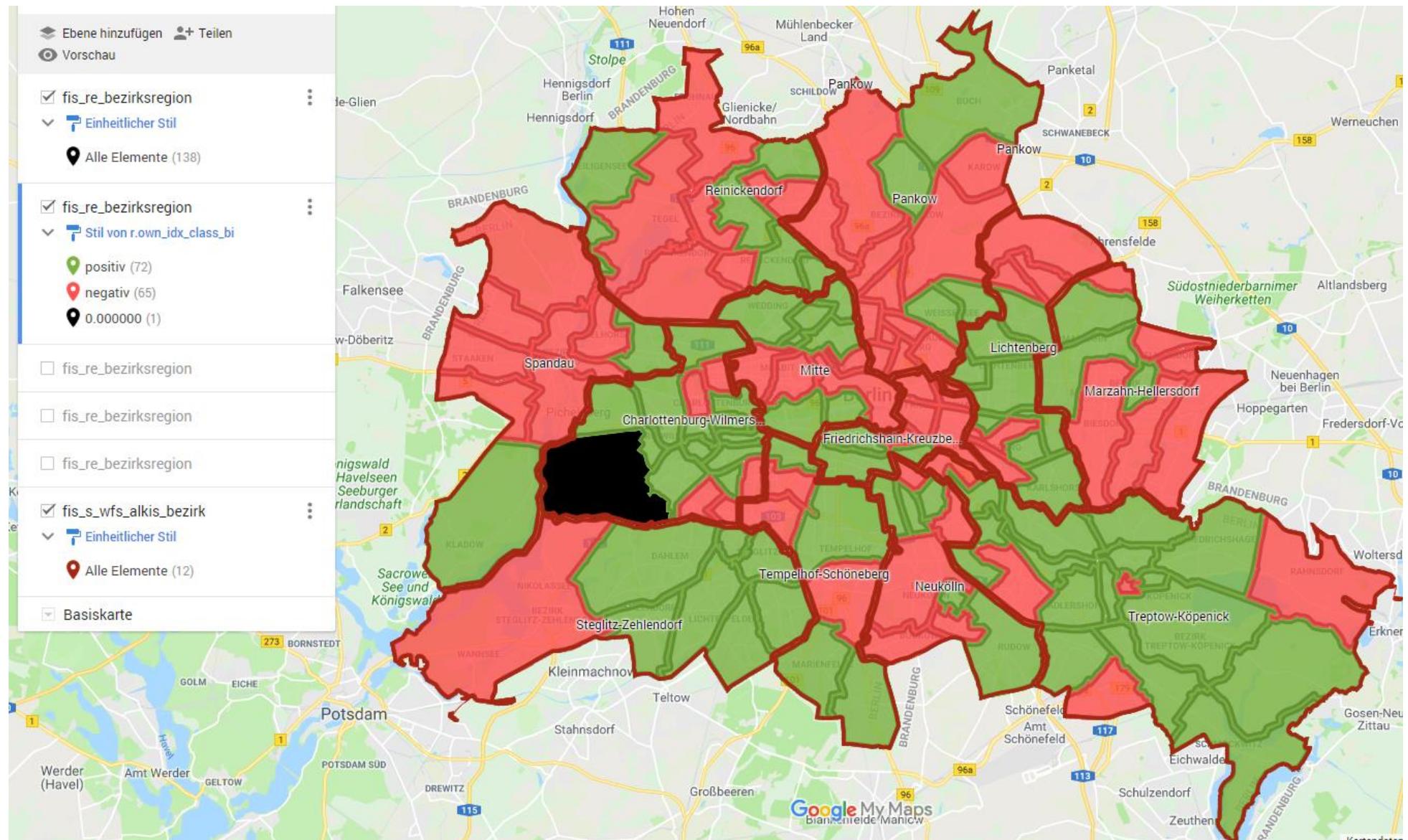


Abbildung 9-37: Übersicht Berlin in Bezirksregionen 2016: Dynamik Index nach (Döring & Ulbricht, 2016) (Eigene Darstellung mit GoogleMyMaps)

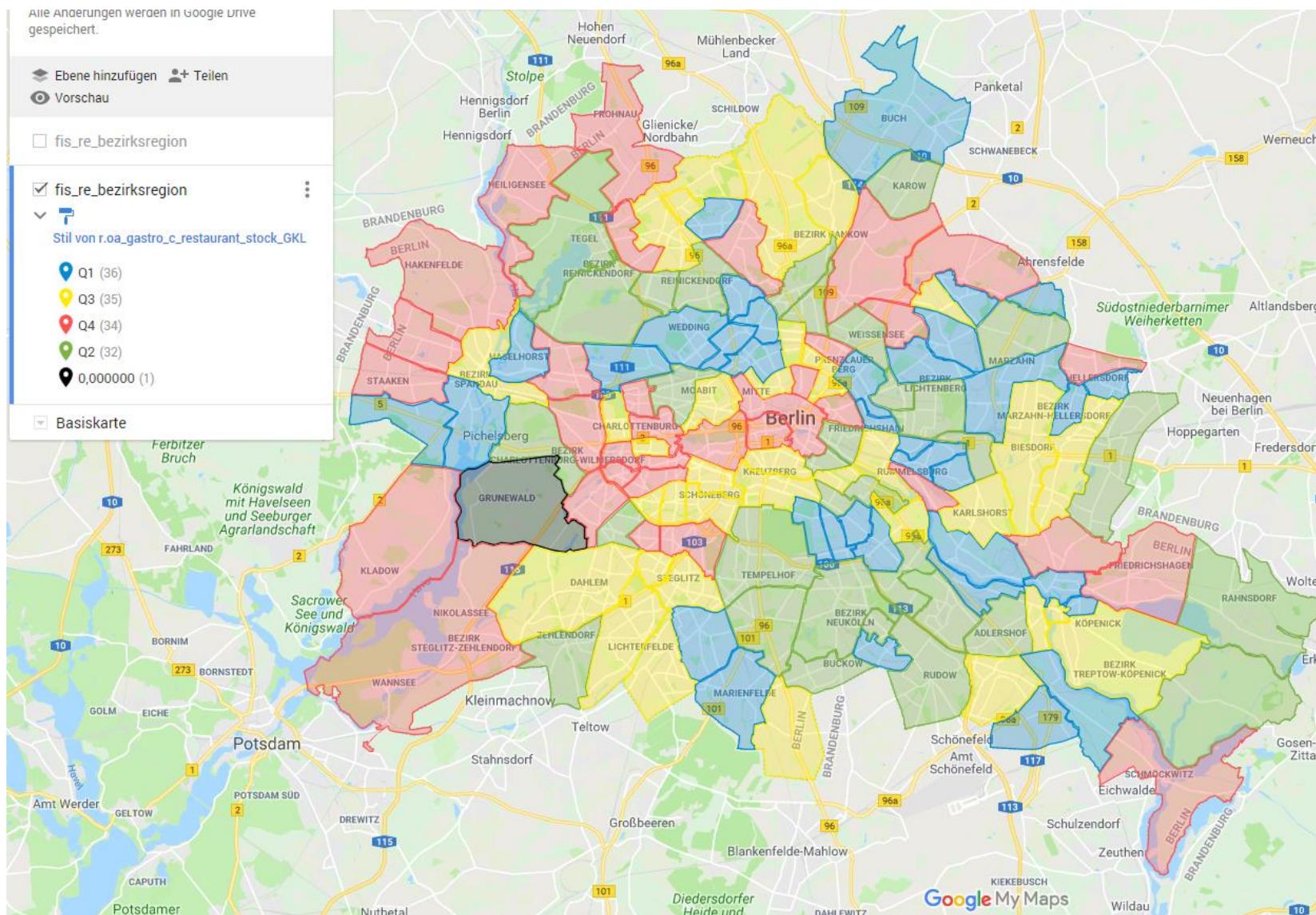


Abbildung 9-38: Übersicht Berlin in Bezirksregionen 2016: OA Restaurants in Größenklassen (Eigene Darstellung mit GoogleMyMaps)

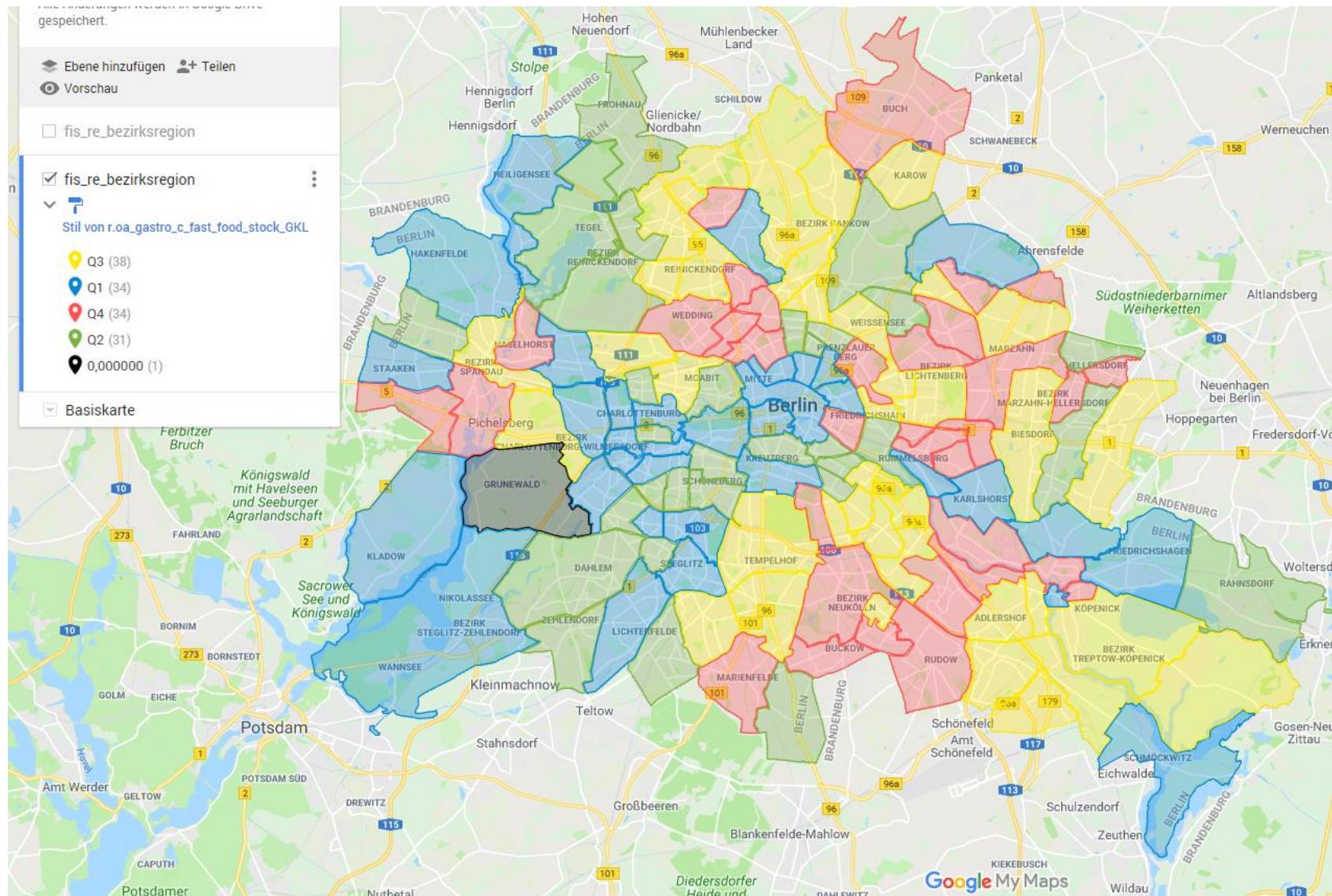


Abbildung 9-39: Übersicht Berlin in Bezirksregionen 2016: OA FastFood in Größenklassen (Eigene Darstellung mit GoogleMyMaps)

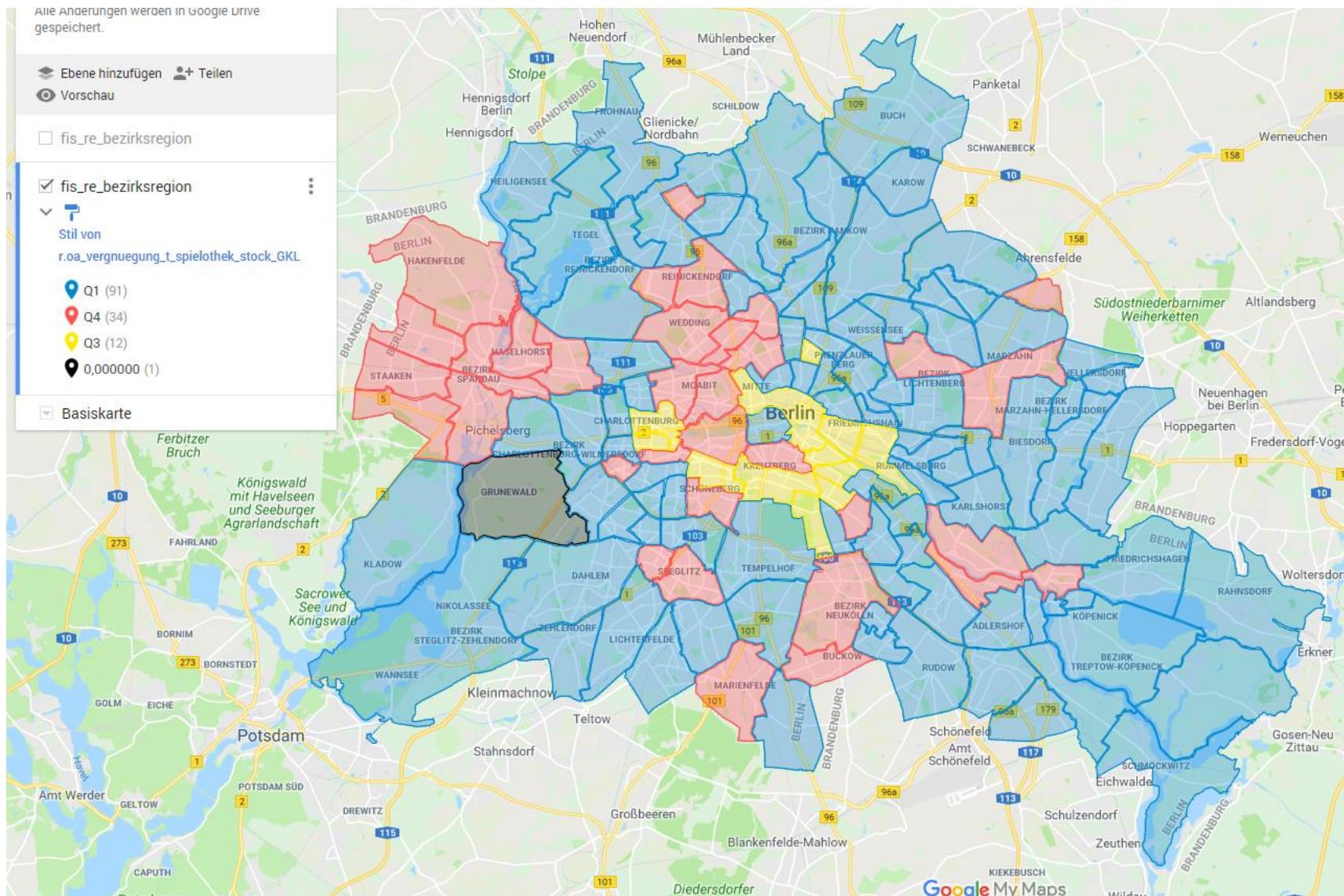


Abbildung 9-40: Übersicht Berlin in Bezirksregionen 2016: OA Spielotheken in Größenklassen (Eigene Darstellung mit GoogleMyMaps)

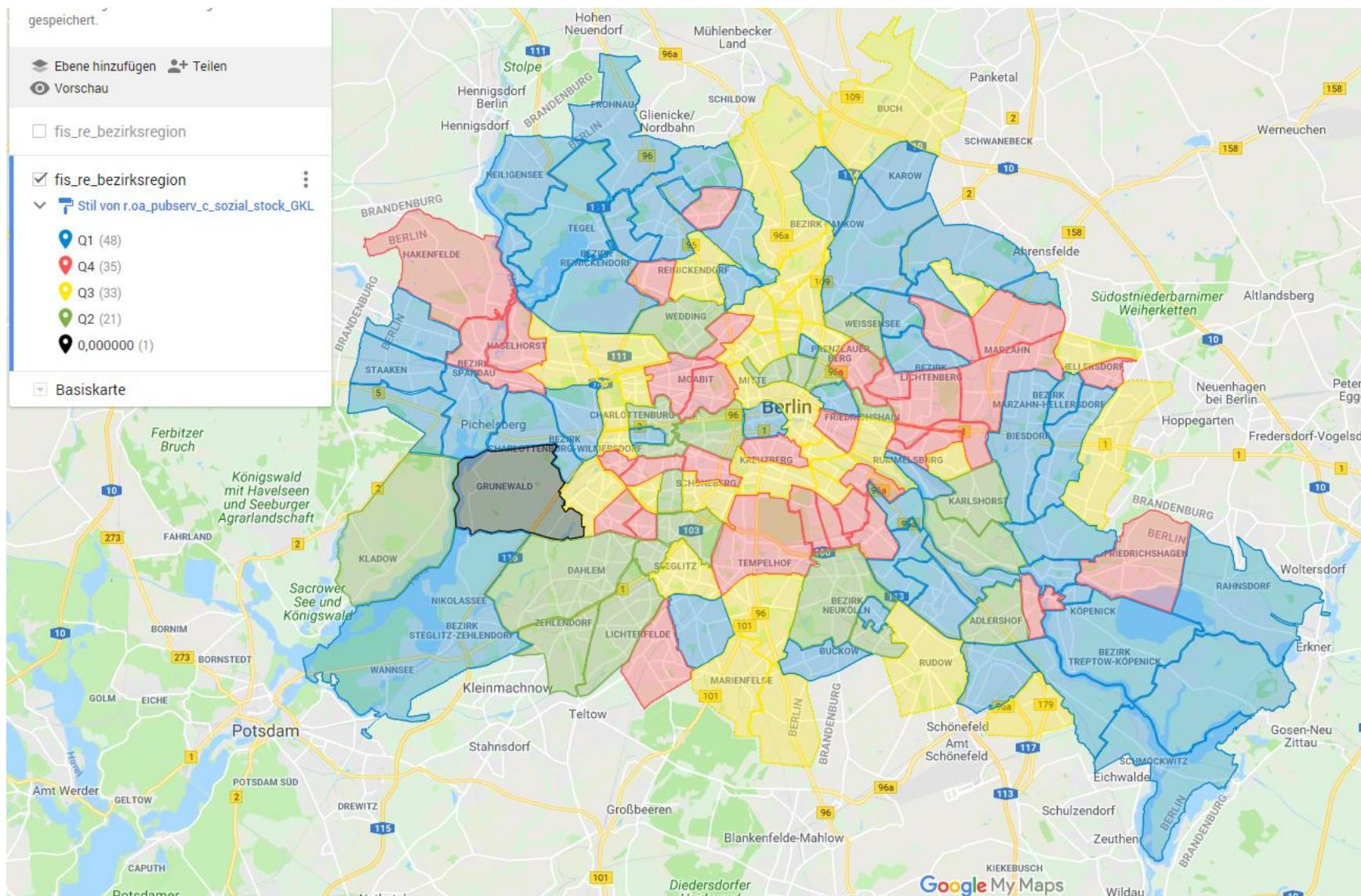


Abbildung 9-41: Übersicht Berlin in Bezirksregionen 2016: OA Soziales in Größenklassen (Eigene Darstellung mit GoogleMyMaps)

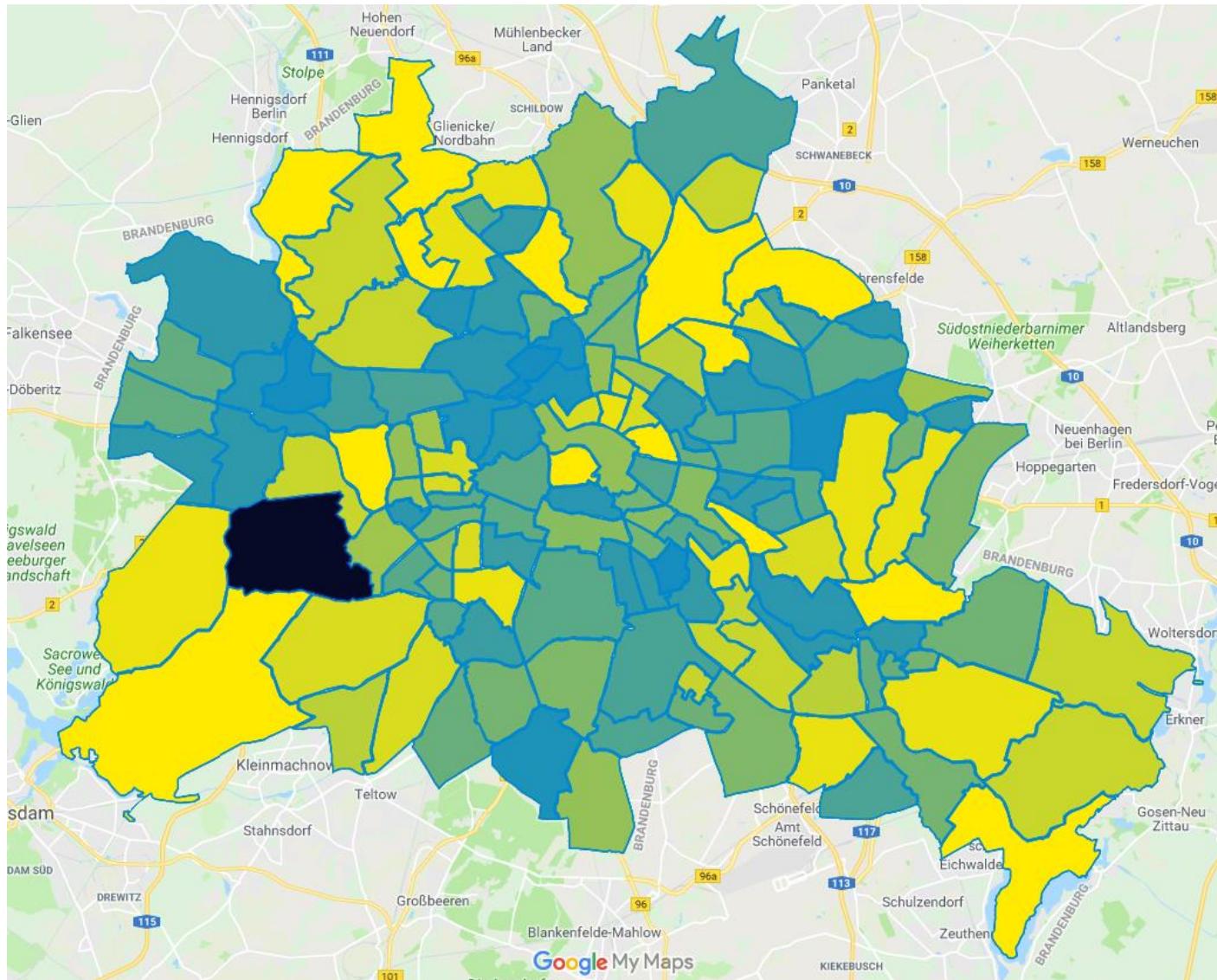


Abbildung 9-42: Übersicht Berlin in Bezirksregionen 2016: Geographische Darstellung SimpleLogistic Klassifikator (Eigene Darstellung mit GoogleMyMaps)

Literaturverzeichnis

- (2005). § 10 *IFG*. Verfügbar unter https://www.gesetze-im-internet.de/ifg/__10.html
- (2009). § 11 *GeoZG*. Verfügbar unter http://www.gesetze-im-internet.de/geozg/__11.html
- (1999). § 16 *IFG / Landesnorm Berlin*. Zugriff am 21.08.2018. Verfügbar unter http://gesetze.berlin.de/jportal/portal/t/ls9/page/bsbeprod.psml?pid=Dokumentanzeige&showdoccase=1&js_peid=Trefferliste&documentnumber=1&numberofresults=1&fromdoctodoc=yes&doc.id=jlr-InfFrGBEpP16#focuspoint
- (2009). § 4 *GeoZG*. Verfügbar unter http://www.gesetze-im-internet.de/geozg/__4.html
- (2015). § 556d *BGB*. *Zulässige Miethöhe bei Mietbeginn; Verordnungsermächtigung - dejure.org*. Zugriff am 26.07.2018. Verfügbar unter <https://dejure.org/gesetze/BGB/556d.html>
- Airbnbvsberlin.de. (2016, 29. April). *Airbnb vs. Berlin? Was sagen die Daten?* Zugriff am 17.09.2018. Verfügbar unter <http://www.airbnbvsberlin.de/#map>
- Amt für Statistik Berlin-Brandenburg. (o.D.a). *AfS StatIS-BBB - Tabellenansicht*, Amt für Statistik Berlin-Brandenburg. Zugriff am 22.08.2018. Verfügbar unter <https://www.statistik-berlin-brandenburg.de/webapi/jsf/tableView/tableView.xhtml>
- Amt für Statistik Berlin-Brandenburg. (o.D.b). *Berliner Raumbezüge*, Amt für Statistik Berlin-Brandenburg. Zugriff am 22.08.2018. Verfügbar unter <https://www.statistik-berlin-brandenburg.de/regionales/rbs/raumberlin.asp?kat=4001>
- Amt für Statistik Berlin-Brandenburg. (10.2011). *Abgestimmter Datenpool Berlin - Einwohnerregisterstatistik Wohndauer und Wohnlage*, Amt für Statistik Berlin-Brandenburg.
- Amt für Statistik Berlin-Brandenburg. (Juni 2012). *Open Data - Einwohnerregisterstatistik Beschreibung*, Amt für Statistik Berlin-Brandenburg.
- Amt für Statistik Berlin-Brandenburg. (02.2014). *Abgestimmter Datenpool Berlin - Einwohnerregisterstatistik Daten zum Migrationshintergrund*, Amt für Statistik Berlin-Brandenburg.
- ApoBank. (2015). *Standortanalyse für Ärzte & Apotheker*, apoBank. Zugriff am 27.08.2018. Verfügbar unter <https://existenzgruendung.apobank.de/existenzgruendungsberatung/standortanalyse.html>
- Balassa, B. (1965). Trade Liberalisation and "Revealed" Comparative Advantage. *The Manchester School*, 33 (2), 99–123. <https://doi.org/10.1111/j.1467-9957.1965.tb00050.x>

- Beekmans, J. (2011). *Check-In Urbanism. Exploring Gentrification through Four-square Activity*. Master Thesis. University of Amsterdam, Amsterdam. Zugriff am 18.08.2018. Verfügbar unter <http://arno.uva.nl/document/228932>
- Berliner Mietverein. (2015). *Info 169: Die Mietpreisbremse bei Wiedervermietung*. Zugriff am 26.07.2018. Verfügbar unter <https://www.berliner-mieterverein.de/recht/in-foblaetter/info-169-die-mietpreisbremse-bei-wiedervermietung.htm>
- Birch, D. L. (1971). Toward a Stage Theory of Urban Growth. *Journal of the American Institute of Planners*, 37 (2), 78–87. <https://doi.org/10.1080/01944367108977361>
- Bloomberg, J. (Forbes, Hrsg.). (2017, 8. Januar). *Fake News? Big Data And Artificial Intelligence To The Rescue*. Zugriff am 27.08.2018. Verfügbar unter <https://www.forbes.com/sites/jasonbloomberg/2017/01/08/fake-news-big-data-and-artificial-intelligence-to-the-rescue/#5b6f759a4a30>
- Bosch, V. (2016). Big Data in der Marktforschung. Warum mehr Daten nicht automatisch bessere Informationen bedeuten. *GfK Forschung*. Zugriff am 27.08.2018. Verfügbar unter https://www.gfk-verein.org/sites/default/files/medien/2327/doku-mente/bosch_gfk_vol_8_no_2_deutsch.pdf
- Bostic, R. W. & Martin, R. W. (2003). Black Home-owners as a Gentrifying Force? Neighbourhood Dynamics in the Context of Minority Home-ownership. *Urban Studies*, 40 (12), 2427–2449. <https://doi.org/10.1080/0042098032000136147>
- Boston.gov (New Urban Mechanics, Hrsg.). (2017, 21. April). *Street Bump*, City of Boston. Zugriff am 20.08.2018. Verfügbar unter <https://www.boston.gov/departments/new-urban-mechanics/street-bump>
- Bouckaert, R. R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A. et al. (2018, 4. September). *WEKA Manual for Version 3-8-3*. Verfügbar unter <file:///C:/Users/dhelweg/Downloads/WekaManual-3-8-3.pdf>
- Breiman, L. (2001). Random Forests. *Machine learning*, 45 (1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- Bruns, J. & Bernsdorf, B. (2016, 12. Mai). BigGIS - Big-Data und Data-Mining im Umfeld städtischer Nutzungskartierung. Zugriff am 27.08.2018. Verfügbar unter https://amazonas.fzi.de/smw/sites/fzi.de.biggis/images/b/ba/20160512_Big-Data_und_Data-Mining.pdf
- Bundesministerium für Verkehr und digitale Infrastruktur. (o.D.). *Werkzeuge für die einfache Erstellung komplexer Vergleichsindizes – WEKOVI*, Bundesministerium für Verkehr und digitale Infrastruktur. Zugriff am 27.08.2018. Verfügbar unter

- <https://www.bmvi.de/SharedDocs/DE/Artikel/DG/mfund-projekte/werkzeuge-erstellung-komplexer-vergleichsindizes-wekovi.html>
- Chance-praxis.de (Hrsg.). (2015, 15. November). *ApoBank: Standortanalysen künftig noch umfangreicher*. Zugriff am 27.08.2018. Verfügbar unter <http://www.chance-praxis.de/praxisgruender/praxisgruendung/apobank-%E2%80%A8standortanalysen-kuenftig-noch-umfangreicher/>
- Clark, J. (IBM, Hrsg.). (2017, 22. August). *Facing the threat: Big Data and crime prevention*. Zugriff am 13.08.2018. Verfügbar unter <https://www.ibm.com/blogs/internet-of-things/big-data-crime-prevention/>
- CNN (Hrsg.). (2017). *Why UPS trucks never turn left*. Zugriff am 17.08.2018. Verfügbar unter <https://edition.cnn.com/2017/02/16/world/ups-trucks-no-left-turns/index.html>
- CRISP-DM consortium, Chapman, P.; Clinton, J.; Kerber, R.; Khabaza, T.; Reinartz, T.; Shearer, C. & Wirth, R. (Mitarbeiter) (NCR Systems Engineering Copenhagen, DaimlerChrysler AG, SPSS Inc. & OHRA Verzekeringen en Bank Groep B.V., Hrsg.). (2000). *CRISP-DM 1.0*, CRISP-DM consortium. Zugriff am 19.08.2018. Verfügbar unter <https://www.the-modeling-agency.com/crisp-dm.pdf>
- Dangschat, J. S. (1988). Gentrification. Der Wandel Innenstadtnaher Wohnviertel. In J. Friedrichs (Hrsg.), *Kölner Zeitschrift für Soziologie und Sozialpsychologie* (S. 272–292). Opladen: Westdt. Verl. https://doi.org/10.1007/978-3-322-83617-5_14
- Datafloq.com (Hrsg.). (2014). *Why UPS spends over \$ 1 Billion on Big Data Annually*. Zugriff am 17.08.2018. Verfügbar unter <https://datafloq.com/read/ups-spends-1-billion-big-data-annually/273>
- Davidson, M. & Lees, L. (2005). New-Build ‘Gentrification’ and London’s Riverside Renaissance. *Environment and Planning A*, 37 (7), 1165–1190. <https://doi.org/10.1068/a3739>
- Ddsgeo. (o.D.). *Geodaten, Demographiedaten, Software, Geoinformationssysteme (GIS)*. Zugriff am 21.08.2018. Verfügbar unter <http://www.ddsgeo.de/produkte.html>
- Ddsgeo. (2013). Flyer PLZ8 Deutschland. Zugriff am 27.08.2018. Verfügbar unter http://www.ddsgeo.de/download/flyer/PLZ8_Flyer.pdf
- Ddsgeo. (2018). Zoom! 01/2018. Zugriff am 27.08.2018. Verfügbar unter http://www.ddsgeo.de/de/zoom/zoom_01_18.pdf
- (o.D.). *DE:PBF Format – OpenStreetMap Wiki*. Zugriff am 29.08.2018. Verfügbar unter https://wiki.openstreetmap.org/wiki/DE:PBF_Format

- Dean, J. & Ghemawat, S. (2004). MapReduce: Simplified Data Processing on Large Clusters. In *Proceedings of the 6th Conference on Symposium on Operating Systems Design & Implementation - Volume 6* (OSDI'04, S. 10). Berkeley, CA, USA: USENIX Association. Verfügbar unter <http://dl.acm.org/citation.cfm?id=1251254.1251264>
- Deeb, A. E. (2015, 18. Juni). *The Unreasonable Effectiveness of Random Forests – Rants on Machine Learning – Medium*. Zugriff am 14.09.2018. Verfügbar unter <https://medium.com/rants-on-machine-learning/the-unreasonable-effectiveness-of-random-forests-f33c3ce28883>
- Deutsches Institut für Urbanistik (Hrsg.). (2011). *Was ist eigentlich Gentrifizierung?* 4. Zugriff am 08.03.2018. Verfügbar unter <https://difu.de/publikationen/difu-berichte-42011/was-ist-eigentlich-gentrifizierung.html>
- (Digital Innovation and Transformation, HBS, Hrsg.). (2017, 5. April). *UPS – Digital Innovation and Transformation*, Havard Business School. Zugriff am 17.08.2018. Verfügbar unter <https://digit.hbs.org/submission/ups/>
- Hohlfeld, J. & Kaczmarek, J. (Mitarbeiter) (digital kompakt, Hrsg.). (2018, 12. Juli). *Parkling – per App zum freien Parkplatz*, digital kompakt. Zugriff am 19.08.2018. Verfügbar unter <https://www.digitalkompakt.de/podcast/parkling-parkplatz-suche-app-podcast/>
- Disy Informationssysteme GmbH (Hrsg.). (o.D.a). *BMVI-Projekt: Wertschöpfung aus Open Data*. Zugriff am 27.08.2018. Verfügbar unter <https://www.disy.net/de/unternehmen/aktuelles/news-2018/bmvi-projekt-wertschoepfung-aus-open-data/>
- Disy Informationssysteme GmbH (Hrsg.). (o.D.b). *Neues Frühwarnsystem für Quellwasserverschmutzung in Jordanien*. Zugriff am 27.08.2018. Verfügbar unter <https://www.disy.net/de/unternehmen/aktuelles/news-2018/neues-fruehwarnsystem-fuer-quellwasserverschmutzung-in-jordanien/>
- Donges, N. (2018, 22. Februar). *The Random Forest Algorithm – Towards Data Science*. Zugriff am 14.09.2018. Verfügbar unter <https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd>
- Döring, C. & Ulbricht, K. (2016). Gentrification-Hotspots und Verdrängungsprozesse in Berlin. Eine quantitative Analyse. In I. Helbrecht (Hrsg.), *Gentrifizierung in Berlin. Verdrängungsprozesse und Bleibestrategien* (Urban studies, S. 17–43). Bielefeld: transcript.

- Dpa. (2016). *In Berlin steigt die Zahl der Eigentumswohnungen massiv an.* Zugriff am 26.07.2018. Verfügbar unter <https://www.morgenpost.de/berlin/article208312211/Deutlich-mehr-Eigentumswohnungen-in-Berlin.html>
- Duarte, A., Laguna, M. & Marti, R. (2018). *Metaheuristics for Business Analytics. A Decision Modeling Approach* (EURO advanced tutorials on operational research). Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-68119-1>
- Elastic.co. (o.D.). *GeoHash grid Aggregation / Elasticsearch Reference [6.3] / Elastic*, elastic.co. Zugriff am 21.08.2018. Verfügbar unter <https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-bucket-geohash-grid-aggregation.html>
- Elder, R. (2017, 7. Juni). *Myers Diff Algorithm - Code & Interactive Visualization*. Zugriff am 14.09.2018. Verfügbar unter <http://blog.robertelder.org/diff-algorithm/>
- Fawcett, T. (2004). ROC graphs: Notes and practical considerations for researchers. *Machine learning*, 31 (1), 1–38. Verfügbar unter <http://bifn.gmu.edu/mmasso/ROC101.pdf>
- FOCUS Online. (2017, 9. August). *Mietpreisexplosion in Berlin: Mieter müssen bis zu 200 Prozent mehr zahlen.* Zugriff am 25.07.2018. Verfügbar unter https://www.focus.de/immobilien/mieten/mietpreisexplosion-mieter-muessen-bis-zu-200-prozent-mehr-zahlen_id_7452852.html
- Foursquare. (o.D.). *Venue Categories - Foursquare Developer*. Zugriff am 24.08.2018. Verfügbar unter <https://developer.foursquare.com/docs/resources/categories>
- Foursquare. (2018, 10. April). *Foursquare-Richtlinien zur API-Plattform und Datennutzung*, Foursquare. Zugriff am 24.08.2018. Verfügbar unter <https://de.foursquare.com/legal/api/platformpolicy>
- Frank, E. (2014, 12. Juni). *Logistic VS Simple Logistic*. Verfügbar unter <http://weka.8497.n7.nabble.com/Logistic-VS-Simple-Logistic-tp31410p31420.html>
- Frerichs, R. R. (2016, 18. April). *Father of Modern Epidemiology*. Zugriff am 13.08.2018. Verfügbar unter <http://www.ph.ucla.edu/epi/snow/fatherofepidemiology.html>
- Friedrichs, J. (1996). Gentrification. Forschungsstand und methodologische Probleme. In J. Friedrichs & R. Kecske (Hrsg.), *Gentrification. Theorie und Forschungsergebnisse* (S. 13–40). Wiesbaden: VS Verlag für Sozialwissenschaften. https://doi.org/10.1007/978-3-322-97354-2_2

- Gantz, J. & Reinsel, D. (EMC, Hrsg.). (2011). *Extracting Value from Chaos*, IDC. Zugriff am 10.08.2018. Verfügbar unter <https://www.emc.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf>
- Gartner. (o.D.). *What Is Big Data? - Gartner IT Glossary - Big Data*, Gartner. Zugriff am 10.08.2018. Verfügbar unter <https://www.gartner.com/it-glossary/big-data>
- Geofabrik GmbH; OpenStreetMap contributors. (o.D.). *Geofabrik Download Server*, Geofabrik GmbH; OpenStreetMap contributors. Zugriff am 25.08.2018. Verfügbar unter <https://download.geofabrik.de/>
- Geospin (Hrsg.). (2016). *Big Data Visualisierung, Analyse und Optimierung*. Zugriff am 16.08.2018. Verfügbar unter <https://www.geospin.de/de/>
- Gesellschaft für Informatik. (2013). *Big Data*. Zugriff am 10.08.2018. Verfügbar unter <https://gi.de/informatiklexikon/big-data/>
- GfK. (2018, 12. Februar). *Geomarketing Software RegioGraph*. Zugriff am 08.03.2018. Verfügbar unter <http://regiograph.gfk.com/de/>
- Glass, R. L. (1964). *London: aspects of change* (Bd. 3): MacGibbon & Kee.
- Glatter, J. (2006). *Gentrification in Ostdeutschland – untersucht am Beispiel der Dresdner Äußeren Neustadt*. Dissertation. Technische Universität Dresden, Dresden. Zugriff am 27.07.2018. Verfügbar unter <http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-205079>
- Google (Hrsg.). (2016, 8. Dezember). *Google Flu Trends*. Zugriff am 13.08.2018. Verfügbar unter <https://www.google.org/flutrends/about/>
- Google (Hrsg.). (2018, 29. Juni). *google/open-location-code*. Zugriff am 21.08.2018. Verfügbar unter https://github.com/google/open-location-code/blob/master/docs/olc_definition.adoc#fig_olc_area
- Google Cloud. (2018, 19. Juli). *Google Maps Platform Terms of Service*, Google Cloud. Zugriff am 24.08.2018. Verfügbar unter <https://cloud.google.com/maps-platform/terms/>
- Google Developers. (2018, 9. Juli). *Places API*, Google Developers. Zugriff am 24.08.2018. Verfügbar unter <https://developers.google.com/places/web-service/details?hl=de>
- (o.D.). *GovData. Datenportal für Deutschland - GovData*. Zugriff am 21.08.2018. Verfügbar unter <https://www.govdata.de/>
- Graphmasters GmbH (Hrsg.). (2018). *Graphmasters digitalisiert den HAJ Hannover Marathon 2018*. Zugriff am 16.08.2018. Verfügbar unter <https://www.graphmasters.net/de/>

- Hammel, D. J. & Wyly, E. K. (1996). A MODEL FOR IDENTIFYING GENTRIFIED AREAS WITH CENSUS DATA. *Urban Geography*, 17 (3), 248–268.
<https://doi.org/10.2747/0272-3638.17.3.248>
- Hamnett, C. (1991). The Blind Men and the Elephant. The Explanation of Gentrification. *Transactions of the Institute of British Geographers*, 16 (2), 173.
<https://doi.org/10.2307/622612>
- Hamnett, C. & Randolph, B. (1983). The Changing Tenure Structure of the Greater London Housing Market, 1961–1981. *The London Journal*, 9 (2), 153–164.
<https://doi.org/10.1179/ldn.1983.9.2.153>
- Simonite, T. (Mitarbeiter) (heise online, Hrsg.). (2011, 7. Februar). *Navigation im Schwarm*, Heise Medien. Zugriff am 16.08.2018. Verfügbar unter
<https://www.heise.de/tr/artikel/Navigation-im-Schwarm-1183752.html>
- Helbrecht, I. (1996). Die Wiederkehr der Innenstädte. Zur Rolle von Kultur, Kapital und Konsum in der Gentrification. *Geographische Zeitschrift*, 84 (1), 1–15. Verfügbar unter <http://www.jstor.org/stable/27818731>
- Hermes Germany (Hrsg.). (2018, 22. Mai). *Paketzustellung in Deutschland: Hermes führt intelligente Tourenplanung ein*. Zugriff am 16.08.2018. Verfügbar unter
<https://newsroom.hermesworld.com/paketzustellung-in-deutschland-hermes-fuehrt-intelligente-tourenplanung-ein-15331/>
- Holm, A. (2014). Gentrifizierung - mittlerweile ein Mainstreamphänomen? In Bundesinstitut für Bau-, Stadt- und Raumforschung (Hrsg.), *Zwischen Erhalt, Aufwertung und Gentrifizierung - Quartiere und Wohnungsbestände im Wandel* (IzR 4.2014). Verfügbar unter https://www.bbsr.bund.de/BBSR/DE/Veroeffentlichungen/IzR/2014/4/Inhalt/DL_Holm.pdf?__blob=publicationFile&v=2
- Holm, A. & Schulz, G. (2016). Gentrimap: Ein Messmodell für Gentrification und Verdrängung. In I. Helbrecht (Hrsg.), *Gentrifizierung in Berlin. Verdrängungsprozesse und Bleibestrategien* (Urban studies, S. 287–318). Bielefeld: transcript.
- Hoover, E. M. & Vernon, R. (1959). Anatomy of a metropolis. The changing distribution of people and jobs within the New York Metropolitan Region.
- Hortonworks. (o.D.a). *Internet of Things and Big Data Real-Time Analytics with Hortonworks DataFlow (HDF)*. Zugriff am 28.08.2018. Verfügbar unter <https://de.hortonworks.com/products/data-platforms/hdf/>
- Hortonworks. (o.D.b). *Manage Data-at-Rest and Deliver Big Data Analytics with Hortonworks Data Platform (HDP)*. Zugriff am 28.08.2018. Verfügbar unter
<https://de.hortonworks.com/products/data-platforms/hdp/>

- Hortonworks, Murthy, A. (Mitarbeiter). (2013, 20. Februar). *Introducing... Tez: Accelerating processing of data stored in HDFS*. Zugriff am 09.09.2018. Verfügbar unter <https://de.hortonworks.com/blog/introducing-tez-faster-hadoop-processing/>
- Hristova, D., Williams, M. J., Musolesi, M., Panzarasa, P. & Mascolo, C. (2016). Measuring Urban Social Diversity Using Interconnected Geo-Social Networks. In J. Bourdeau (Hrsg.), *WWW'16. Proceedings of the 25th International Conference on World Wide Web : May 11-15, 2016, Montreal, Canada* (S. 21–30) [Geneva, Switzerland]: International World Wide Web Conferences Steering Committee.
- Hu, H., Wen, Y., Chua, T.-S. & Li, X. (2014). Toward Scalable Systems for Big Data Analytics. A Technology Tutorial. *IEEE Access*, 2, 652–687.
<https://doi.org/10.1109/ACCESS.2014.2332453>
- Hu, M. & Liu, B. (2004). Mining and summarizing customer reviews. In R. Kohavi, J. Gehrke, W. DuMouchel & J. Ghosh (Hrsg.), *KDD-2004. Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining : August 22-25, 2004, Seattle, Washington, USA* (S. 168). New York, NY: ACM Press.
- Ideation & Prototyping Lab der Technologiestiftung Berlin, Meier, S. (Mitarbeiter). (2018, 14. Februar). *Berlins räumliche Einheiten*, Ideation & Prototyping Lab der Technologiestiftung Berlin. Zugriff am 26.06.2018. Verfügbar unter <https://lab.technologiestiftung-berlin.de/projects/spatial-units/index.html>
- Immobilien Scout GmbH. (o.D.a). *Geohierarchy API*, Immobilien Scout GmbH. Zugriff am 23.08.2018. Verfügbar unter <https://api.immobilienscout24.de/our-apis/gis/geohierarchy.html>
- Immobilien Scout GmbH. (o.D.b). *Pricehistory API*, Immobilien Scout GmbH. Zugriff am 23.08.2018. Verfügbar unter <https://api.immobilienscout24.de/our-apis/valuation/pricehistory-api.html>
- Immobilienportale.com. (o.D.). *Übersicht Immobilienportale - Immobilienportale.com*. Zugriff am 23.08.2018. Verfügbar unter <https://www.immobilienscout24.de/uebersicht-immobilienportale/>
- ImmobilienScout24. (2015, 4. September). *Terms of Use for IS24 REST-API*, ImmobilienScout24. Zugriff am 24.08.2018. Verfügbar unter <https://api.immobilienscout24.de/terms-of-use.html>
- Immoverkauf24.de. (o.D.). *Lage, Lage, Lage: Das bedeutet die alte Immobilienweisheit*. Zugriff am 27.08.2018. Verfügbar unter <https://www.immoverkauf24.de/immobilienverkauf/immobilienverkauf-a-z/lage-lage-lage/>

- Jakulin, A. (2005, 13. Juni). *Machine Learning Based on Attribute Interactions*. PhD Dissertation. University of Ljubljana. Zugriff am 17.09.2018. Verfügbar unter <http://www.stat.columbia.edu/~jakulin/Int/jakulin05phd.pdf>
- Jiang, H., Chen, Y., Qiao, Z., Weng, T.-H. & Li, K.-C. (2015). Scaling up MapReduce-based Big Data Processing on Multi-GPU systems. *Cluster Computing*, 18 (1), 369–383. <https://doi.org/10.1007/s10586-014-0400-1>
- Jiang, Z. & Shekhar, S. (2017). *Spatial Big Data Science. Classification Techniques for Earth Observation Imagery*. Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-60195-3>
- Joseph, B. (2014, 25. Februar). *Can reinsurers ignore big data? PwC's Bryan Joseph on how big data will reshape the industry*. Zugriff am 13.08.2018. Verfügbar unter <https://www.globalreinsurance.com/can-reinsurers-ignore-big-data/1407237.article>
- KDNuggets (Hrsg.). (Oct 2014). *CRISP-DM, still the top methodology for analytics, data mining, or data science projects*. Zugriff am 19.08.2018. Verfügbar unter <https://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html>
- Kecskes, R. (1994). Gentrification: eine Klassifikation von Wohnungsnachfragern auf dem Wohnungsmarkt. *ZA-Information / Zentralarchiv für Empirische Sozialforschung* (35), 27–48. Verfügbar unter <http://nbn-resolving.de/urn:nbn:de:0168-ssoar-201194>
- Kecskes, R. & Friedrichs, J. (2004). *Angewandte Soziologie. [...] Festschrift zum 65. Geburtstag von Jürgen Friedrichs ...]* (1. Aufl.).
- KfW. (2017, 8. Oktober). Deutschlands Banken schalten bei Filialschließungen einen Gang höher - Herkulesaufgabe Digitalisierung. Zugriff am 27.08.2018. Verfügbar unter <https://www.kfw.de/PDF/Download-Center/Konzernthemen/Research/PDF-Dokumente-Fokus-Volkswirtschaft/Fokus-2017/Fokus-Nr.-181-Oktober-2017-Bank-filialen.pdf>
- Klout. (2016, 5. Oktober). *brickhouse*. Zugriff am 03.09.2018. Verfügbar unter <https://github.com/klout/brickhouse>
- Knight Frank LLP. (2017). *GLOBAL RESIDENTIAL CITIES INDEX*. Zugriff am 04.07.2018. Verfügbar unter <https://content.knightfrank.com/research/1026/documents/en/global-residential-cities-index-q4-2017-5413.pdf>
- Koch, S., Kortus, M., Schierbaum, C. & Schramm, S. (2016). Wohin (ver-)drängt es die Kreuzberger_innen? Wohin ziehen die Verdrängten innerhalb eines Gentrification-

- Prozesses? In I. Helbrecht (Hrsg.), *Gentrifizierung in Berlin. Verdrängungsprozesse und Bleibestrategien* (Urban studies, S. 69–106). Bielefeld: transcript.
- Kraemer, M. U. G., Bisanzio, D., Reiner, R. C., Zakar, R., Hawkins, J. B., Freifeld, C. C. et al. (2018). Inferences about spatiotemporal variation in dengue virus transmission are sensitive to assumptions about human mobility. A case study using geolocated tweets from Lahore, Pakistan. *EPJ Data Science*, 7 (1), 107.
<https://doi.org/10.1140/epjds/s13688-018-0144-x>
- Kraftfahrt-Bundesamt. (o.D.). *Bestand nach Gemeinden (FZ 3)*, Kraftfahrt-Bundesamt. Zugriff am 21.08.2018. Verfügbar unter https://www.kba.de/DE/Statistik/Produktkatalog/produkte/Fahrzeuge/fz3_b_uebersicht.html
- Kumar, N. (2017, 31. Januar). *Twitter's tweets analysis using Lambda Architecture*. Zugriff am 28.08.2018. Verfügbar unter <https://blog.knoldus.com/twitter-tweets-analysis-using-lambda-architecture/>
- Kurgan, L. A. & Musilek, P. (2006). A survey of Knowledge Discovery and Data Mining process models. *The Knowledge Engineering Review*, 21 (01), 1.
<https://doi.org/10.1017/S0269888906000737>
- Landwehr, N., Hall, M. & Frank, E. (2005). Logistic model trees. *Machine learning*, 59 (1-2), 161–205. Verfügbar unter <https://link.springer.com/content/pdf/10.1007/s10994-005-0466-3.pdf>
- Laney, D. (2001). 3D Data Management: Controlling Data Volume, Velocity and Variety. In *META Group Research Note 6*.
- Lazer, D., Kennedy, R., King, G. & Vespignani, A. (2014). Big data. The parable of Google Flu. Traps in big data analysis. *Science (New York, N.Y.)*, 343 (6176), 1203–1205. <https://doi.org/10.1126/science.1248506>
- Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10 (8), 707–710. Zugriff am 31.08.2018. Verfügbar unter <https://nymity.ch/sybilhunting/pdf/Levenshtein1966a.pdf>
- Mapr.com. (o.D.). *Lambda Architecture / MapR*, mapr.com. Zugriff am 28.08.2018. Verfügbar unter <https://mapr.com/developercentral/lambda-architecture/>
- Marz, N. & Warren, J. (2015). *Big data. Principles and best practices of scalable real-time data systems*. Shelter Island, NY: Manning.
- Microm (Micromarketing-Systeme und Consult GmbH, Hrsg.). (2017, 11. Januar). *Geodaten auf PLZ8-Ebene für eine homogene Raumgliederung*. Zugriff am 08.03.2018. Verfügbar unter <https://www.microm.de/geodaten/plz8/>

- Microsoft (azure.microsoft.com, Hrsg.). (o.D.). *Nutzen des Team Data Science-Prozesses mit Azure Machine Learning*. Zugriff am 19.08.2018. Verfügbar unter <https://azure.microsoft.com/de-de/documentation/learning-paths/data-science-process/>
- Myers, E. W. (1986). An O(ND) difference algorithm and its variations. *Algorithmica*, 1 (1-4), 251–266. Verfügbar unter <http://www.xmailserver.org/diff2.pdf>
- Neumair, S.-M. & Haas, H.-D. (Gabler Wirtschaftslexikon, Hrsg.). (2018, 19. Februar). *Definition: Standortwahl*. Zugriff am 27.08.2018. Verfügbar unter <https://wirtschaftslexikon.gabler.de/definition/standortwahl-42973/version-266310>
- (o.D.). *Offene Daten Berlin. Offene Daten lesbar für Mensch und Maschine. Das ist das Ziel*. Zugriff am 21.08.2018. Verfügbar unter <https://daten.berlin.de/>
- Open Knowledge International. (o.D.). *Open Definition 2.1 - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge*. Zugriff am 21.08.2018. Verfügbar unter <https://opendefinition.org/od/2.1/en/>
- (o.D.). *OpenPoiMap*. Zugriff am 29.08.2018. Verfügbar unter <http://openpoimap.org/>
- OpenStreetMap contributors. (o.D.). *OpenStreetMap*, OpenStreetMap contributors. Zugriff am 25.08.2018. Verfügbar unter www.openstreetmap.org/copyright
- OpenStreetMap Foundation. (o.D.). *OpenStreetMap*. Zugriff am 25.08.2018. Verfügbar unter https://wiki.osmfoundation.org/wiki/Main_Page
- OpenStreetMap Taginfo. (o.D.). *Über Taginfo*, OpenStreetMap Taginfo. Zugriff am 25.08.2018. Verfügbar unter <https://taginfo.openstreetmap.org/about>
- (o.D.). *Osmosis – OpenStreetMap Wiki*. Zugriff am 29.08.2018. Verfügbar unter <https://wiki.openstreetmap.org/wiki/Osmosis#Windows>
- Papachristos, A. V., Smith, C. M., Scherer, M. L. & Fugiero, M. A. (2011). More Coffee, Less Crime? The Relationship between Gentrification and Neighborhood Crime Rates in Chicago, 1991 to 2005. *City & Community*, 10 (3), 215–240. <https://doi.org/10.1111/j.1540-6040.2011.01371.x>
- Parkling (Hrsg.). (o.D.). *Parkling named winner of Stockholm's traffic administration innovation contest*. Zugriff am 19.08.2018. Verfügbar unter <http://www.parkling.eu/stockholm>
- Pavie, A. (2015, 5. Oktober). *OSM2Hive*. Zugriff am 30.08.2018. Verfügbar unter <https://github.com/PanierAvide/OSM2Hive>
- Pearce, J., Blakely, T., Witten, K. & Bartie, P. (2007). Neighborhood deprivation and access to fast-food retailing. A national study. *American journal of preventive medicine*, 32 (5), 375–382. <https://doi.org/10.1016/j.amepre.2007.01.009>

- Plus.codes. (o.D.). *How it works*. Zugriff am 21.08.2018. Verfügbar unter <https://plus.codes/howitworks>
- Powell, L. M., Slater, S., Chaloupka, F. J. & Harper, D. (2006). Availability of physical activity-related facilities and neighborhood demographic and socioeconomic characteristics. A national study. *American journal of public health*, 96 (9), 1676–1680. <https://doi.org/10.2105/AJPH.2005.065573>
- Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. Verfügbar unter http://www.flinders.edu.au/science_engineering/fms/School-CSEM/publications/tech_reps-research_artfcts/TRRA_2007.pdf
- Press, G. (2016, 23. März). *Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says*. Zugriff am 16.09.2018. Verfügbar unter <https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#f45319c6f637>
- Puget, J. F. (IBM developerworks community, Hrsg.). (2013). *Big Data For Dummies (IT Best Kept Secret Is Optimization)*. Zugriff am 27.08.2018. Verfügbar unter https://www.ibm.com/developerworks/community/blogs/jfp/entry/big_data_for_dummies23?lang=en
- Hebes, P.; Plate, E. & Tonndorf, T. (Mitarbeiter) (Referat Stadtentwicklungsplanung & EBP, Hrsg.). (2017, 27. April). *Big Data und Crowd Data für die Berliner Stadtentwicklungsplanung*, Senatsverwaltung für Stadtentwicklung und Wohnen Berlin. Zugriff am 08.03.2018. Verfügbar unter http://www.stadtentwicklung.berlin.de/planen/basisdaten_stadtentwicklung/big-data/index.shtml
- Renuka, J. (2016, 9. September). *Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog*. Zugriff am 11.09.2018. Verfügbar unter <https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/>
- Rubrichi, S., Smoreda, Z. & Musolesi, M. (2018). A comparison of spatial-based targeted disease mitigation strategies using mobile phone data. *EPJ Data Science*, 7 (1), 355. <https://doi.org/10.1140/epjds/s13688-018-0145-9>
- Rundfunk Berlin-Brandenburg, r. (Hrsg.). (2016). *Suche Mitte, finde Stadtrand. Datenanalyse zum Berliner Mietwohnungsmarkt*. Zugriff am 21.07.2018. Verfügbar unter <https://www.rbb24.de/politik/thema/2016/wohnen/thema-uebersicht.html>
- Schaefer, B. (12.2014). *Social media to locate urban displacement. assessing the risk of displacement using volunteered geographic information in the city of Los Angeles*.

- Master Thesis. University of Southern California, California. Zugriff am 14.12.2017.
 Verfügbar unter <http://digitallibrary.usc.edu/cdm/ref/collection/p15799coll3/id/514559>
- Schulz, G. (2015, 25. August). *Aufwertung und Verdrängung in Berlin. Räumliche Analysen zur Messung von Gentrifizierung*. Masterarbeit. Humboldt-Universität zu Berlin, Berlin. Zugriff am 08.03.2018. Verfügbar unter <https://cloud.freiheitswolke.org/index.php/s/epgnZyBiftoNtoq#pdfviewer>
- Senatsverwaltung für Stadtentwicklung. (2011). Stadtentwicklungsplan Zentren 3. Zugriff am 18.08.2018. Verfügbar unter https://www.stadtentwicklung.berlin.de/planen/stadtentwicklungsplanung/download/zentren/2011-07-31_StEP_Zentren3.pdf
- Senatsverwaltung für Stadtentwicklung und Wohnen. (2017). *Bericht Monitoring Soziale Stadtentwicklung Berlin 2017 / Land Berlin*, Senatsverwaltung für Stadtentwicklung und Wohnen. Zugriff am 23.08.2018. Verfügbar unter https://www.stadtentwicklung.berlin.de/planen/basisdaten_stadtentwicklung/monitoring/de/2017/index.shtml
- Senatsverwaltung für Stadtentwicklung und Wohnen Berlin. (o.D.). *Lebensweltlich orientierte Räume (LOR) in Berlin*, Senatsverwaltung für Stadtentwicklung und Wohnen Berlin. Zugriff am 22.08.2018. Verfügbar unter https://www.stadtentwicklung.berlin.de/planen/basisdaten_stadtentwicklung/lor/
- Senatsverwaltung für Umwelt, Verkehr und Klimaschutz. (o.D.). *Teilverkehrszellen (TVz) und Verkehrszellen (Vz) in Berlin / Land Berlin*, Senatsverwaltung für Umwelt, Verkehr und Klimaschutz. Zugriff am 22.08.2018. Verfügbar unter <https://www.berlin.de/senuvk/verkehr/datengrundlagen/verkehrszellen/de/informationen.shtml>
- SenseBox. (2015). *openSenseMap*, senseBox. Zugriff am 21.08.2018. Verfügbar unter <https://opensensemapper.org/info>
- Smith, N. (1979). Toward a Theory of Gentrification A Back to the City Movement by Capital, not People. *Journal of the American Planning Association*, 45 (4), 538–548. <https://doi.org/10.1080/01944367908977002>
- Streetbump.org (New Urban Mechanics & Connected Bits, Hrsg.). (o.D.). *Street Bump*. Zugriff am 20.08.2018. Verfügbar unter <http://www.streetbump.org/about>
- Swiss Re (Hrsg.). (2013). *Wegweisende Einführung in die Rückversicherung*. Zugriff am 13.08.2018. Verfügbar unter http://www.swissre.com/library/archive/Wegweisende_Einführung_in_die_Rückversicherung.html#inline

- Tagesspiegel.de (Hrsg.). (2015, 8. Oktober). *500 Flüchtlinge ziehen in die Nachbarschaft von Ikea*. Zugriff am 11.09.2018. Verfügbar unter <https://www.tagesspiegel.de/berlin/bezirke/spandau/berlin-spandau-500-fluechtlinge-ziehen-in-die-nachbarschaft-von-ikea/12420302.html>
- Tape, T. (2015, 22. Juni). *The Area Under an ROC Curve*, University of Nebraska. Zugriff am 10.09.2018. Verfügbar unter <http://gim.unmc.edu/dxtests/roc3.htm>
- Theguardian.com (Hrsg.). (2013, 11. Juni). *Google buys Waze map app for \$1.3bn*. Zugriff am 16.08.2018. Verfügbar unter <https://www.theguardian.com/technology/2013/jun/11/google-buys-waze-maps-billion>
- Tobler, W. R. (1970). A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography*, 46, 234. <https://doi.org/10.2307/143141>
- TripAdvisor Developer Portal. (o.D.). *Content API / Locations*, TripAdvisor Developer Portal. Zugriff am 24.08.2018. Verfügbar unter <https://developer-tripadvisor.com/content-api/locations/>
- TripAdvisor Developer Portal. (2017, 2. Juni). *Content API / Terms and Conditions*, TripAdvisor Developer Portal. Zugriff am 24.08.2018. Verfügbar unter <https://developer-tripadvisor.com/content-api/terms-and-conditions/>
- Universität Zürich. (o.D.). *Methodenberatung - Zusammenhänge*, Universität Zürich. Zugriff am 09.09.2018. Verfügbar unter https://www.methodenberatung.uzh.ch/de/datenanalyse_spss/zusammenhaenge.html
- Venerandi, A., Quattrone, G., Capra, L., Quercia, D. & Saez-Trumper, D. (2015). Measuring Urban Deprivation from User Generated Content. In D. Cosley, A. Forte, L. Ciolfi & D. McDonald (Hrsg.), *CSCW '15. Proceedings of the 2015 ACM International Conference on Computer-Supported Cooperative Work and Social Computing : March 14-18, 2015, Vancouver, BC, Canada* (S. 254–264). New York: ACM.
- Veness, C. (2018, 4. März). *Geohash encoding/decoding*. Zugriff am 21.08.2018. Verfügbar unter <https://www.movable-type.co.uk/scripts/geohash.html>
- Wagner, S. (2016, 22. April). *Geo-Big-Data für die Filialnetzplanung von Banken. Mit maschinellem Lernen zur perfekten Filialnetzplanung*, IT Finanzmagazin. Zugriff am 08.03.2018. Verfügbar unter <https://www.it-finanzmagazin.de/geo-big-data-fuer-die-filialnetzplanung-von-banken-mit-maschinellem-lernen-zur-perfekten-filialnetzplanung-29982/>
- Waterford Technologies (Hrsg.). (2017). *Big Data - Interesting Statistics, Facts & Figures*. Zugriff am 10.08.2018. Verfügbar unter <https://www.waterfordtechnologies.com/big-data-interesting-facts/>

- White, T. (2009). *Hadoop. The Definitive Guide* (Safari Books Online). Sebastopol: O'Reilly Media Inc.
- Wikipedia (Hrsg.). (2018a, 7. August). *Geodatenzugangsgesetz*. Zugriff am 21.08.2018. Verfügbar unter <https://de.wikipedia.org/w/index.php?oldid=174359265>
- Wikipedia (Hrsg.). (2018b, 12. August). *Epidemiologie*. Zugriff am 13.08.2018. Verfügbar unter <https://de.wikipedia.org/w/index.php?oldid=179540535>
- Wirth, R. & Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining.
- Witten, I. H., Pal, C. J., Frank, E. & Hall, M. A. (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques". Zugriff am 09.09.2018. Verfügbar unter https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf
- Witten, I. H., Pal, C. J., Frank, E. & Hall, M. A. (2017). *Data mining. Practical machine learning tools and techniques* (Fourth edition). Cambridge, MA: Morgan Kaufmann.
- Wu, X., Zhu, X., Wu, G.-Q. & Ding, W. (2014). Data mining with big data. *IEEE Transactions on Knowledge and Data Engineering*, 26 (1), 97–107. <https://doi.org/10.1109/TKDE.2013.109>
- Www.c-dev.ch. (2012, 26. Oktober). *Koordinatenformate, Koordinaten Darstellung und Schreibweise WGS84 - Schweizer Gitter - c-dev*. Zugriff am 03.09.2018. Verfügbar unter <http://www.c-dev.ch/2012/10/26/koordinatenformate/>
- Yelp for Developers. (2018, 23. März). *API Terms of Use*, Yelp for Developers. Zugriff am 24.08.2018. Verfügbar unter https://www.yelp.com/developers/api_terms
- Yelp Fusion API. (o.D.a). *All Category List*, Yelp Fusion API. Zugriff am 24.08.2018. Verfügbar unter https://www.yelp.de/developers/documentation/v3/all_category_list
- Yelp Fusion API. (o.D.b). *Business Search Endpoint*, Yelp Fusion API. Zugriff am 24.08.2018. Verfügbar unter https://www.yelp.com/developers/documentation/v3/business_search
- ZEIT ONLINE (2018, 14. April). 13.000 protestieren gegen steigende Mieten. Zugriff am 04.07.2018. Verfügbar unter <https://www.zeit.de/gesellschaft/zeitgeschehen/2018-04/berlin-demonstration-mietenwahnsinn-tausende-protestieren-gegen-hohe-mieten>
- Zeng, X. & Martinez, T. R. (2000). Distribution-balanced stratified cross-validation for accuracy estimation. *Journal of Experimental & Theoretical Artificial Intelligence*, 12 (1), 1–12. <https://doi.org/10.1080/095281300146272>

- Zook, M. (2017). Crowd-sourcing the smart city. Using big geosocial media metrics in urban governance. *Big Data & Society*, 4 (1), 205395171769438.
<https://doi.org/10.1177/2053951717694384>
- Zook, M., Shelton, T. & Poorthuis, A. (2017, 10. Mai). Big Data and the City. Zugriff am 14.12.2017. Verfügbar unter https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2966568
- Zukin, S., Lindeman, S. & Hurson, L. (2015). The omnivore's neighborhood? Online restaurant reviews, race, and gentrification. *Journal of Consumer Culture*, 17 (3), 459–479. <https://doi.org/10.1177/1469540515611203>

Eigenständigkeitserklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt und mich anderer als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Dennis Helweg