

# Räumliche Vorhersage des urbanen Radverkehrs mit Machine Learning

Maximilian Samuel Weinhold  
Economics, 7. Semester  
505314  
[mweinhol@uni-muenster.de](mailto:mweinhol@uni-muenster.de)

Masterarbeit  
Wintersemester 2022  
Institut für Verkehrswissenschaft  
Prof. Dr. Gernot Sieg

## INHALTSVERZEICHNIS

0.1 Abkürzungsverzeichnis . . . . .	6
1. <i>Einleitung</i> . . . . .	7
2. <i>Literaturüberblick</i> . . . . .	8
2.1 Erforschung des Radverkehrs . . . . .	8
2.1.1 Forschung mit Zählstationen . . . . .	9
2.1.2 Forschung mit Bike Sharing Diensten . . . . .	10
2.1.3 Forschung mit GPS und Handy Daten . . . . .	13
2.2 Faktoren des Radverkehrs . . . . .	16
2.2.1 Faktor Wetter . . . . .	16
2.2.2 Faktor Feinstaubbelastung . . . . .	18
2.2.3 Faktor Corona Maßnahmen . . . . .	19
2.2.4 Faktor der städtischen Unterschiede . . . . .	19
2.2.5 Faktor Radverkehrsunfälle . . . . .	20
2.2.6 Sonstige Faktoren . . . . .	22
2.3 Forschungslücken und Anknüpfungspunkte . . . . .	24
3. <i>Zusammensetzung des Datensatz</i> . . . . .	26
3.1 Fahrradzähler . . . . .	26
3.2 Wetterdaten . . . . .	30
3.3 Demographische und soziale Statistiken . . . . .	30
3.3.1 Statistisches Bundesamt . . . . .	31
3.3.2 ADFC Fahrradindex . . . . .	32
3.4 Corona Daten . . . . .	34
3.5 Open Street Map Daten . . . . .	37
3.5.1 Ausgewählte PoIs . . . . .	38

3.5.2	Ausgestaltung des öffentlichen Verkehrs . . . . .	41
3.5.3	Straßentypen . . . . .	42
3.5.4	Sonstige . . . . .	47
4.	<i>Verwendete Methoden</i> . . . . .	48
4.1	Problem zur Autokorrelation . . . . .	48
4.1.1	Lagged Variablen . . . . .	49
4.1.2	Erklärende Variablen . . . . .	50
4.1.3	Resampling . . . . .	51
4.2	OLS Regression . . . . .	51
4.3	Support Vector Regression . . . . .	52
4.4	Random Forests Regression . . . . .	55
4.5	Neuronale Netze . . . . .	57
4.5.1	Aufbau eines neuronalen Netzes . . . . .	57
4.5.2	Berechnung eines neuronalen Netzes . . . . .	59
4.6	Validation . . . . .	61
4.6.1	Cross Validation . . . . .	62
4.6.2	Conditional Validation Set Building . . . . .	62
4.6.3	Weighted Subset Building . . . . .	64
5.	<i>Ergebnisse</i> . . . . .	66
5.1	OLS Regression . . . . .	66
5.2	Support Vector Regression . . . . .	68
5.3	Random Forest Regression . . . . .	72
5.4	Neuronales Netz . . . . .	76
5.5	Model Projektion . . . . .	77
5.6	Räumliche Korrelation zu Verkehrsunfällen . . . . .	86
6.	<i>Diskussion</i> . . . . .	87
6.1	Fazit . . . . .	87
7.	<i>Anhang</i> . . . . .	88

Literaturverzeichnis . . . . .	98
--------------------------------	----

## ABBILDUNGSVERZEICHNIS

2.1 Anzahl der Leihräder (A) Singapore zwischen 8-9 Uhr am 25. Juni 2017; (B) New Taipei City zwischen 18-19 Uhr am 1. August 2018; (C) Chicago zwischen 16-18 Uhr am 26. August 2019; (D) New York zwischen 17-18 Uhr am 9. Juni 2014. . . . .	12
2.2 Fahrradaufkommen für Glasgow 2017 . . . . .	15
2.3 (a und b). Berechnete Wahrscheinlichkeiten für Radunfälle in Philadelphia . . . . .	21
2.4 Räumliche Verteilung aller Fahrradunfälle (2011–2014) nach Zensus Blöcken in Florida. . . . .	23
3.1 Städte die im Datensatz vertreten sind. . . . .	27
3.2 Verteilung der Beobachtung nach Einwohnergröße der Städte .	28
3.3 Verteilung der Beobachtung nach Jahren . . . . .	29
3.4 Wetterdaten im Überblick . . . . .	30
3.5 Verhältnis von Bevölkerung und Fahrradaufkommen . . . . .	31
3.6 Verteilung des Fahrradaufkommens in Entfernung zum Stadtzentrum . . . . .	33
3.7 Verhältnis von Bevölkerung und Fahrradaufkommen . . . . .	34
3.8 Verlauf der Corona Inzidenz und des Radverkehrs . . . . .	35
3.9 Radverkehr im Lockdown . . . . .	36
3.10 Verteilung von Straßentypen im Datensatz nach der alten Ermittlung . . . . .	43
3.11 Beispiel der Straßentypanalyse . . . . .	45
3.12 Radverkehr nach Straßenbeschaffenheit . . . . .	47
4.1 Hyperebenen (Hyperplanes) in Support Vector Machines . . .	53
4.2 Soft Margins in Support Vector Machines . . . . .	53

4.3	Eindimensionale lineare SVR . . . . .	54
4.4	Ein Entscheidungsbaum basierend auf Daten von Fahrradzählstationen in Mannheim und Daten des DWD 2016 bis 2022. . . . .	56
4.5	Neural network with a single hidden layer . . . . .	58
4.6	Aktivierungsfunktionen . . . . .	59
4.7	Gradient Descent . . . . .	60
5.1	Feature Selection Ergebnis der OLS Regression in 6 Modellen	67
5.2	Support Vector Regression Performance nach Anteil des Datensatzes . . . . .	70
5.3	Support Vector Regression Ergebnis der RF Regression in 6 Modellen . . . . .	71
5.4	Random Forest Performance nach Anteil des Datensatzes . . . . .	73
5.5	Random Forest Performance nach Anzahl der Zufallsbäume . . . . .	74
5.6	Feature Selection Ergebnis der RF Regression in 6 Modellen . . . . .	75
5.7	Räumliche Modell Projektion: Modellvergleich . . . . .	80
5.8	Feature Selection Ergebnis der RF Regression in 6 Modellen . . . . .	82
5.9	Modellprojektionen in Hamburg und Leipzig . . . . .	83
5.10	Modellprojektionen in Mannheim und Oberhausen . . . . .	84
5.11	Modellprojektionen in Dresden . . . . .	86
7.1	Verteilung des Fahrradaufkommens nach Alter . . . . .	88
7.2	Zusammenhang von Fahrradklima und Radverkehr . . . . .	89
7.3	Zusammenhang von Anzahl der Uni-Gebäude in einem 500 M Radius und Radverkehr . . . . .	90
7.4	Zusammenhang von Anzahl der Supermärkte in einem 1 km Radius und Radverkehr . . . . .	91
7.5	Zusammenhang von Anzahl der Kleidungsgeschäften in einem 2 km Radius und Radverkehr . . . . .	92
7.6	Zusammenhang von Anzahl der Busstationen in einem 1 km Radius und Radverkehr . . . . .	93
7.7	Zusammenhang von Anzahl der Ampeln in einem 1 km Radius und Radverkehr . . . . .	94

7.8	Zusammenhang von Anzahl der Straßenbahnstationen in einem 1 km Radius und Radverkehr . . . . .	95
7.9	Zusammenhang des nächsten Bahnhofes und dem Radverkehr	96
7.10	Entfernung zur nächsten Brücke und Radverkehr . . . . .	97

## TABELLENVERZEICHNIS

5.1	Performance des OLS Modells . . . . .	68
5.2	Performance des SVR Modells . . . . .	72
5.3	Performance des RF Modells . . . . .	74
5.4	Performance des neuronalen Netzes . . . . .	77
5.5	Performance des dritten RF Modells . . . . .	82

### *0.1 Abkürzungsverzeichnis*

**ZIV** Zweirad-Industrie-Verband

**OLS** Ordinary Least Square

**bzw.** beziehungsweise

**z.B.** zum Beispiel

**u.a.** unter anderem

## 1. EINLEITUNG

Vielerorts erlebt der Individualverkehr eine Renaissance des Fahrradfahrens. So ist laut Eisenberger (2015) und dem Verband der Zweirad Industrie ZIV (2022) der Bestand an Fahrrädern in Deutschland von 72 Mio. in 2015 auf 81 Mio. in 2021 angestiegen. Der anhaltende Boom hat viele Gründe. Im Vergleich zum Auto ist die bewegungsintensivere Fortbewegung auf dem Fahrrad gesünder, schont die Umwelt und das Portemonnaie.

Viele Kommunen entscheiden sich u.a. aus diesen Gründen dafür, ihre Stadt fahrradfreundlicher zu gestalten. So unterstützt das Bundesministerium für Verkehr und digitale Infrastruktur BMDV (2020) die Länder und Gemeinden beim Ausbau von Radwegen durch eine direkte Hilfe in Höhe von 660 Mio. Euro bis 2023.

Dies, sowie der genaue lokale Bedarf und die Auslastung von Fahrradwegen, muss bei der Planung der Infrastruktur beachtet werden. Motiviert ist diese Arbeit mit dem Wunsch, Antworten auf mögliche Fragen der Infrastrukturplanung bieten zu können. Die Forschungsfrage, die sich daraus ableitet, lautet: Ist es möglich ein räumliches Modell zu entwickeln, das für ein vollständiges Straßennetz oder für ausgewählte flexible Knotenpunkte zu bestimmten Zeiten Vorhersagen zum Fahrradverkehr machen kann und lassen diese auch zusätzlich die gefährlichsten Stellen für Fahrradunfälle erkennen? Ziel dabei ist es, eine Anleitung zu Erstellung solcher Modelle zu geben.

Dazu beginnt diese Masterarbeit mit einem Einblick in die bestehende Literatur, welcher kategorisiert ist nach Daten und Faktoren. Häufig verwendete Methoden werden im Nachgang näher beschrieben. Das erworbene Wissen dient dazu, ein Modell zu entwickeln, dass im Rahmen dieser Abschlussarbeit auch evaluiert wird. Im Anschluss wird analysiert, wie sehr das Radverkehrsaufkommen mit Radverkehrsunfällen räumlich korreliert.

## 2. LITERATURÜBERBLICK

Die Auslastung von Fahrradwegen bzw. das Aufkommen von Fahrrädern beruht im Wesentlichen auf der individuellen Entscheidung eines jeden Fahrers, das Fahrrad einer anderen Transportalternative vorzuziehen. Versteht man die Faktoren, aus denen sich diese Entscheidung zusammensetzt, dann kann man leichter eine solche Entscheidung vorhersagen. Mit einer Zusammenfassung der bisherigen Literatur wollen Heinen et al. (2010) diese Faktoren finden, wobei sie sich auf Pendler beschränken. Zum einen ist einer dieser Faktoren die bauliche Substanz, die nicht nur die Radwege sondern auch Abstellmöglichkeiten und Ampeln wie auch Verkehrsschilder beinhaltet, zum anderem spielt auch das Wetter eine große Rolle. Einen negativen Effekt hat die Rate der Autobesitzer und natürlich auch die Verfügbarkeit anderer Transportmöglichkeiten.

### 2.1 *Erforschung des Radverkehrs*

Seitdem erschienen zahlreiche weitere empirische Studien, die unser Bild konkretisieren. Diese Studien lassen sich nach Datenquellen oder auch Methoden kategorisieren. Die meisten nutzen drei verschiedene Datengrundlagen zum Fahrradverkehr in Verbindung zu Daten mit anderen Faktoren, wie dem Wetter. Die drei Datenquellen zum Fahrradverkehr stammen von Fahrradzählstation, einer Induktionsschleife die darüber fahrende Räder verifiziert, Daten von Bike Sharing Diensten und Daten verschiedener GPS und Handy Applikationen.

### 2.1.1 Forschung mit Zählstationen

In Deutschland findet sich eine weite Verbreitung von Fahrradzählstationen in verschiedenen Städten, deren Daten oft öffentlich einsehbar sind. Deswegen wäre eine Verwendung dieser Datengrundlage überaus praktisch. Studien, die ähnliche Daten verwenden, sind zB Holmgren et al. (2017), Broucke et al. (2019), Wessel (2020) und Goldmann and Wessel (2021). Auf die Arbeit von Wessel (2020) geht diese Arbeit später ein, denn dort liegt ein starker Fokus auf Wetter Daten. Holmgren et al. (2017) verwenden tägliche Daten von 2006 bis 2014 aus Malmö, ebenfalls wie später bei Wessel (2020) verbunden mit Wetterdaten, Feiertagen und Schulferien. Als Methode zur Auswertung dieser Daten und zur Schätzung des Fahrradaufkommens verwendeten sie Zufallsbaum Regressionssysteme, Support Vector Regression, eine lineare Regression und ein Multiy Layer Perceptron also ein neuronales Netz. Im Vergleich dieser Methoden erzielen sie die treffsichersten Resultate mit einem Regressionsbaum und der Support Vector Regression, die auf quadratische und kubische Kernels zurückgreift. Mithilfe des Support Vector Regressionssystems kommen sie auf ein Bestimmtheitsmaß  $R^2$  von 86,9 %.

Um diese Methoden zu vergleichen nutzen Holmgren et al. (2017) die Cross Validation. Ähnlich gehen Broucke et al. (2019) vor, unter Verwendung von Daten von 12 Zählstationen aus Brüssel. Sie verbinden diese Daten mit temporalen, geographischen und metereologischen Daten. Ebenfalls verwenden sie eine Support Vector Regression, die auf einer radialen Basis Funktion aufbaut, um eines ihrer vier Modelle zu berechnen. Für die anderen drei verwenden sie Random Forests Regression, welche auf Zufallsbäume aufbaut, ein gradient Boosting Modell und ein voll verknüpftes neuronales Netzwerk. Letzteres beinhaltet 2 Schichten mit einmal 28 und einmal 14 Knoten, die die ReLU (Rectified Linear Unit) Funktion als Aktivierung verwenden. Von allen Modellen mit Wetterdaten erzielt das neuronale Netzwerk hier den niedrigsten RMSE (Root Mena Squared Error) und schneidet somit am besten ab.

### 2.1.2 Forschung mit Bike Sharing Diensten

Die überragende Mehrheit an Studien zur Schätzung und Vorhersage des Fahrradverkehrs verwendet Daten von Bike Sharing Diensten, dazu zählen Kaltenbrunner et al. (2010), Xu et al. (2013), Li et al. (2015), Mitchell (2018), Colace et al. (2020), Gao and Chen (2022) und Li et al. (2022). Dies ist natürlich immer noch nur eine Auswahl an Studien. Die Literatur zu Bike Sharing Systemen ist sehr ausführlich. Eine weitere Übersicht hierzu findet sich bei Mitchell (2018).

Kaltenbrunner et al. (2010) nutzen z.B. Daten von 400 öffentlichen Rad Ausleihstationen in Barcelona. Ihre Studie hegt die Absicht, die Effizienz des bestehende Verleih Systems in Barcelona, das zu dem Zeitpunkt um die 180000 Abonnenten hatte, zu verbessern. Hier geht es also eher darum, vorherzusagen, wie viele Fahrräder sich in welcher Ausleihstation zu welchem Zeitpunkt befinden, um den Nutzern detaillierte Informationen zu geben.

Eine ähnliche Motivation haben Xu et al. (2013). Diese verwenden Daten ihrer Aussagen zufolge des größten öffentlichen Bike Sharing Systems der Welt in Hangzhou in China. Ihr Datensatz beinhaltet die aufgezeichnete Auslastung der Stationen einiger Tage und Wochen zuvor, sowie Daten zum aktuellen und vergangenem Wetter und Informationen über Feiertage. Diesen Datensatz normalisieren und clustern sie mit der k-means Cluster Methode. Darauf wenden sie Support Vector Machines an, um die Gewichte der beschriebenen Estimatoren zu finden. Dieses hybride Modell hat nur noch eine Fehlerrate von 3,57 % vergleicht man dessen Vorhersagen, mit den tatsächlich eingetretenen Auslastungen.

Li et al. (2015) stellen hier ein Paper zur Verfügung das ähnlich funktioniert. Auch sie wollen eine Ausbalancierung der Fahrradbestände an allen Stationen in New York und Washington DC erleichtern. Und genau wie Xu et al. (2013) unterteilen sie die Stationen in Cluster. Für diese Cluster wollen sie Vorhersagen machen, was zu robusteren Ergebnissen führt, als wenn man für jede einzelne Station Vorhersagen bildet. Zusätzlich verwenden Li et al. (2015) Wetter Beobachtungen in ihrem Datensatz. Darauf wird ein Gradient Boosting Regression Tree angewendet. Gradient Boosting kurzläu-

fige zufällige Entscheidungsbäume, die der Reihe nach auf den Fehlerterm des vorherigen Beruhen und diesen versuchen zu verbessern.

Auf Cluster Level Bike Sharing geht auch Mitchell (2018) ein. Er evaluierst verschiedene Machine Learning Methoden miteinander: Random Forests, Fast Feed Forward Neural Networks, Deep Residual Networks und Recurrent Neural Networks. Dabei schlagen sich die Feed Forward Neural Networks am besten.

Einen besonderen Ansatz verfolgen Colace et al. (2020), die Aufnahmen von Überwachungskameras in ihrer Analyse mit aufnehmen.

Eine besonders aktuelle Analyse stammte von Gao and Chen (2022). Gao and Chen (2022) sind hierbei die Ersten von den bisher genannten, die mehr Informationen als nur Wetterberichte in seiner Analyse aufnehmen. Aufbauend auf Daten zu 2098 Fahrradstationen aus Seoul nehmen sie in ihr Modell z.B. Feinstaubbelastungen mit auf, was Hong J (2022) und Zhao et al. (2018) als relevanten Faktor gezeigt haben. Daneben inkludieren sie auch Verkehrsdaten, Corona Fälle und Sozioökonomische Daten, Verkehrsunfälle und Saisonalität. Darauf wenden sie lineare Regressionen, k-nearest neighbour (knn), Random Forests und Support Vector Machines an. All diese Methoden wurden in R angewendet. Diese Modelle vergleichen sie mit einem Validation-Set-Schnitt von 75 % für das Trainings Set und 25 % für das Test Set. Damit betreiben Gao and Chen (2022) eine Feature Selection, die auf den Boruta Algorithmus zurückgreift. Das Ziel von Feature Selektion ist, am Ende ein Modell zu haben, dass nur auf statistisch relevante Variablen zurückgreift. Dazu erstellt der Boruta Algorithmus nach Kursa and Rudnicki (2010) verschiedene unabhängige Bagging Samples, zieht daraus Klassifikationsbäume und beurteilt die Features nach Verlust an Akkurarität. Die relevantesten Variablen waren das Wetter und die Anzahl der Corona Fälle. Mit einer 10-fold Cross Validation fand man heraus, dass sich die Support Vector Machines und die Random Forests am besten geschlagen haben. So lässt sich mit Random Forests ein Test  $R^2$ -Wert von 93 % erreichen und mit SVM ein Test  $R^2$ -Wert von 90 %, was beides sehr gute Werte sind. Dabei spielten auch die sozioökonomische Daten eine Rolle, die im Gesamten zwar wenig Relevanz hatten, aber Unterschiede zwischen den Docking Stationen

erklären konnten.

Li et al. (2022) möchten nicht nur den Bestand von Rädern an fixen Fahrrad Ausleihstationen vorhersagen, sondern die generelle Nutzung und das Verkehrs Volumen von Leihräder in einem Gebiet berechnen. Dazu verwenden sie Daten von festen Fahrradstationen in Chicago und New York und Daten von Ausleihsystemen mit frei stehenden Fahrrädern die GPS Daten speichern in Singapore und New Taipei City. Jede Stadt zerteilen sie in ein Raster und wenden darauf convolutional neural Networks an, die auch oft für die Bilderkennung verwendet werden. Das Ergebnis dessen ist in der Abbildung 2.1 zu sehen.

Eine Besonderheit dieser Arbeiten, die alle auf Daten von Bike Sharing

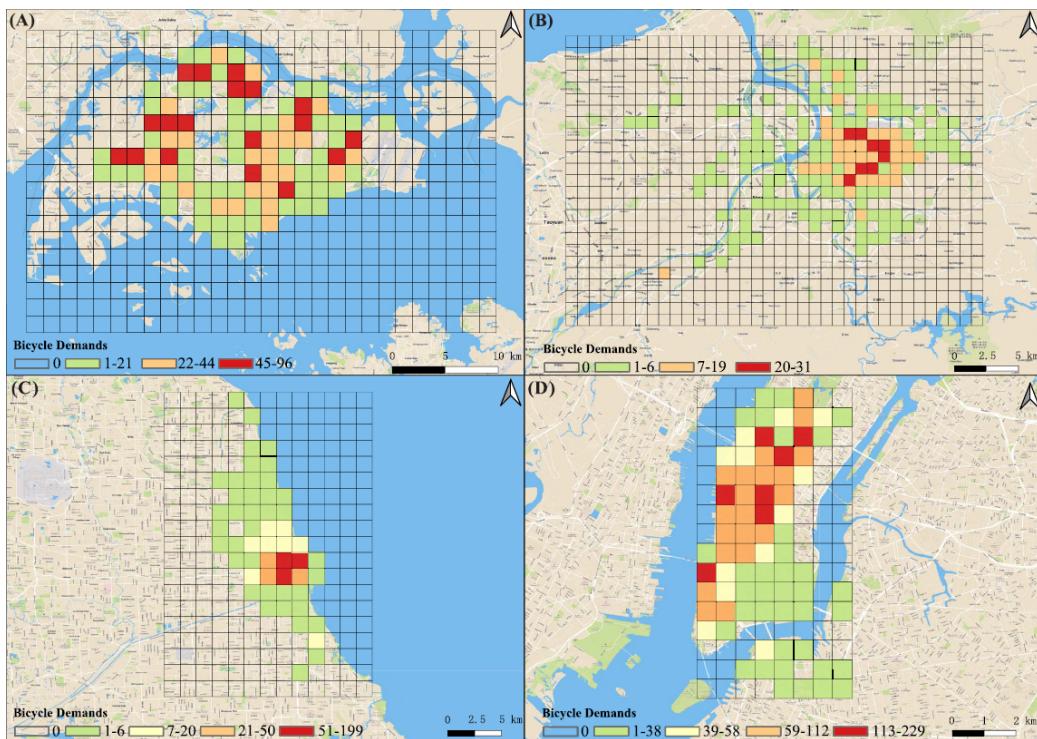


Fig. 2.1: Anzahl der Leihräder (A) Singapore zwischen 8-9 Uhr am 25. Juni 2017; (B) New Taipei City zwischen 18-19 Uhr am 1. August 2018; (C) Chicago zwischen 16-18 Uhr am 26. August 2019; (D) New York zwischen 17-18 Uhr am 9. Juni 2014.

**Quelle:** Li et al. (2022)

Diensten beruhen, ist, dass sie mehr Datenpunkte in einer Staat zur Verfü-

gung haben, da ein Netz von Leihstation eine höhere Dichte haben muss. So können Gao and Chen (2022) z.B. 2098 Stationen in einer Stadt beobachten, während Broucke et al. (2019) z.B. nur 12 Zählstationen in einer Stadt zur Verfügung haben. Deshalb ergibt das Clustering von Bike Sharing Daten Sinn, wie es zB Xu et al. (2013) und Li et al. (2015) vornehmen.

Modelle die auf Zählstationen beruhen, können das gesamte Verkehrs Volumen an bestimmten Punkten messen und schätzen. Die hier kennen gelernten Modelle, die auf Bike Sharing Daten zurückgreifen, können das nicht, sondern ermitteln nur das Verkehrs Volumen, das von diesen Mieträdern kommt, dafür aber immer für das vollständige Mietradssystem. Interessant wäre die Frage, wie Mietradverkehr und Verkehr von Rädern in Privatbesitz miteinander korrelieren. Wer mit dem Fahrrad regulär pendelt, für den ist der Besitz eines Fahrrads langfristig kostengünstiger. So kann es sein, dass Spitzen in beiden Varianten von einander abweichen, weil Mieträder möglicherweise eher touristischen statt utilitaristischen Zwecken dienen. Mietraddaten allein sind also kein ausreichendes Mittel, um den gesamten Fahrrad Verkehr zu modellieren. Möglicherweise gäben aber Handy Daten mehr Aufschluss.

### 2.1.3 Forschung mit GPS und Handy Daten

Schon im vorangegangenen Abschnitt zu den Daten von Bike Sharing Systemen hat Li et al. (2022) GPS Daten von Leihfahrrädern für seine Analyse genutzt. Doch das ist nicht die einzige Möglichkeit um an GPS Daten zu kommen. Ein weiterer Weg sind Handy Applikationen, die beständig GPS Koordinaten aufzeichnen und damit Bewegungsprofile erstellen. Romanillos et al. (2016) stellen solche Studien vor, die auf GPS Daten von Fitness und Leisure Applikationen zurückgreifen.

Eine der ersten Arbeiten wurde von Harvey and Krizek (2007) mit der Absicht erstellt, durch ihr Modell eine Priorisierung von Fahrradinfrastruktur zu ermöglichen. Dazu verwendeten sie noch GPS Logging Ausrüstung bei 51 Teilnehmern in einem Zeitraum von 3 Wochen in South Minneapolis, um favorisierte Radstrecken aufzuzeichnen. Neben der niedrigen Stichprobenmenge war diese Studie noch mit dem Problem des GPS Cleanings konfrontiert, das

zu Positionsabweichungen führen kann. Diese Probleme wurden von Folgestudien wie von Menghini et al. (2010) erstmals behoben, die in Zürich Daten von 2400 Teilnehmern hatten und mit einem verbesserten Detection Algorithmus auch Probleme des GPS Post Processing behoben haben.

Das Prinzip der GPS Aufzeichnung entwickelte sich weiter und Reddy et al. (2010) verwendeten erstmals Handys als GPS Logging Geräte. Dazu verwenden sie Daten der App Biketastic. Das Problem bei Bike Logging Apps ist, dass hier eher Fahrrad Routen aufgezeichnet werden, die der Erholung dienen und nicht unbedingt dem alltäglichen utilitaristischen Stadtverkehr. Die Nutzung dieser Daten gibt also nicht Aufschluss über den gesamten Fahrradverkehr. Bei der hier genutzten Applikation Biketastic ist das jedoch anders, denn Biketastic wird speziell von Pendlern genutzt, um nicht nur GPS Daten aufzuzeichnen, sondern auch Fotos und Audioaufnahmen der Handys, um Straßen mit signifikant hohen Lärm ausfindig zu machen und so lauten Straßenverkehr in das Modell aufzunehmen. Bei der Evaluation stellte man schnell fest, dass Nutzer der App dazu tendierten, diese nur für lange Strecken zu nutzen, und sich nicht die Mühe machen, die App auf dem Handy zu betätigen, wenn man eine Strecke von unter einer Meile zurück legen wollte. Eine weitere Studie stammte von Broach et al. (2012), aber auch hier ist die Anzahl der Studienteilnehmer mit knapp über 164 gering. Weitere Studien in dem Bereich stammen von Musakwa and Selala (2016), Pritchard (2018), Lee and Sener (2020) und Alattar et al. (2021). Eine der aktuelleren Studien ist von Alattar et al. (2021). Sie verwenden Daten, die durch die Fitness App Strava in Glasgow 2017 bis 2018 gewonnen wurden, um den Einfluss des Straßenlayouts auf die Routenwahl der Fahrer zu untersuchen. Dabei bauen sie auf ein räumliches Modell, das an bestimmten Straßenknotenpunkten stündliche durchgehende Fahrradfahrten nach Straße, Abfahrtsort und Zielort betrachtet. Um zu testen, wie gut die Strava Daten den tatsächlichen Radverkehr abbilden, wurde mit insgesamt 36 temporären Zählstationen das Radverkehrsvolumen über zwei Tage stichprobenartig verglichen. Neben den Daten von Strava nutzten die Autoren das Python Paket OSMnx, dass direkten Zugang zu Open Street Map bietet. Open Street Map bietet umfassende Information über Glasgows Straßennetz und so können die Kno-

tenpunkte mit dem Straßennetz verglichen werden. Das daraus resultierende Ergebnis ist in Abbildung 2.2 zu sehen. Auf diese Daten kann nun eine Re-

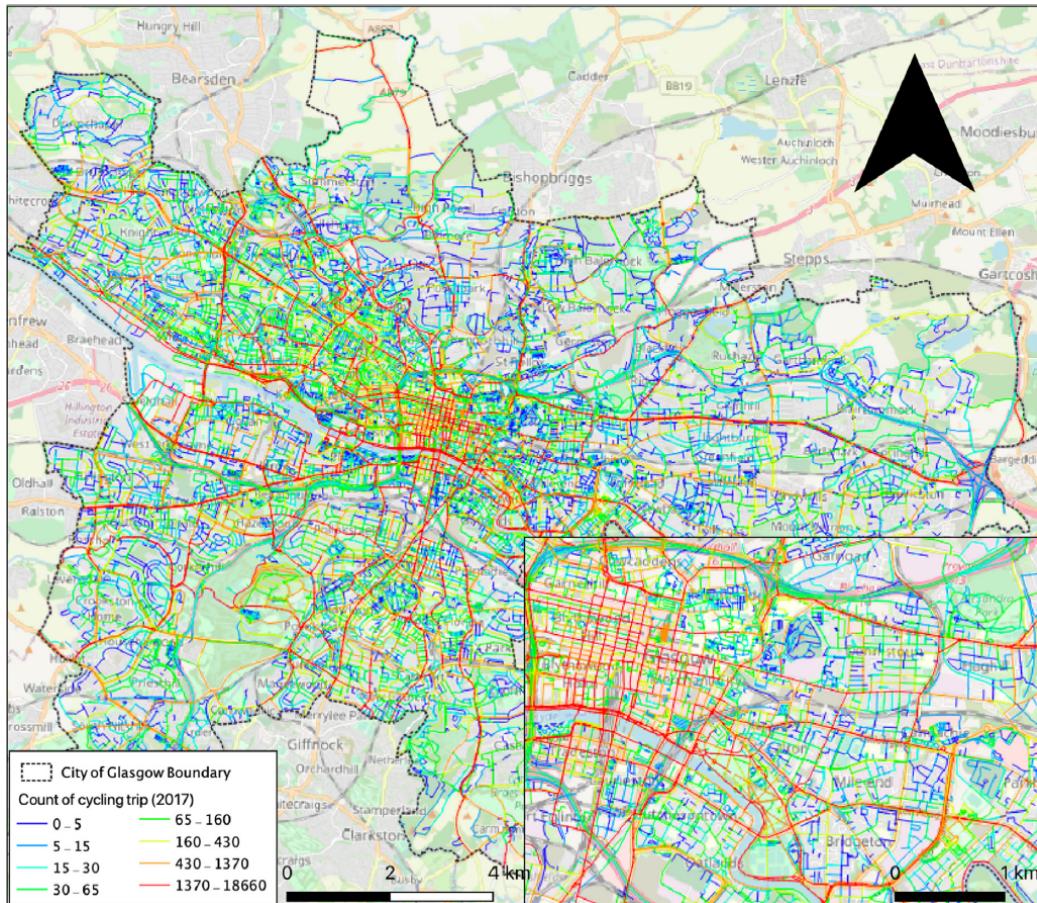


Fig. 2.2: Fahrradaufkommen für Glasgow 2017

**Quelle:** Alattar et al. (2021)

gression angewendet werden, deren erklärende Variablen logarithmerte Indikatoren der Zentralität innerhalb des Knotennetzwerkes sind, die die Anzahl der Fahrradtrips erklären sollen. Bei diesem Regressionsmodell kommt ein Erklärungswert  $R^2$  von 42 % zustande, was im Vergleich zu vorherigen Werten gering ist. Ein interessanter Forschungspunkt den Alattar et al. (2021) hätten hier noch verfolgen können, wäre wie sich die Vorhersagekraft ändert, wenn man Wetter, Feier und Ferientage in das Modell mit aufnimmt. Im Überblick zeigen sich häufige Schwächen von Studien, die Handy Applikationsdaten nutzen, wie den Strava Datensatz. Zunächst unterliegen solche

Apps häufig einer Selbstauswahl der Probanden. So sind z.B. in dem Datensatz von Alattar et al. (2021) Frauen unterrepräsentiert. Häufig dienen diese Apps der sportlichen Betätigung und nicht der Anfahrt zum Arbeitsplatz. Zudem sind ihre Daten schwer zugänglich.

## 2.2 Faktoren des Radverkehrs

Die Prädiktoren eines Modells spielen eine große Rolle, denn mit der Auswahl der richtigen Prädiktoren steht und fällt die Validität des Modells. Deswegen klärt ein gesonderter Literaturüberblick die Tragweite einzelner Variablen, die möglicherweise das Aufkommen an urbanen Fahrradverkehr erklären können. Wichtiges Attribut eines Prädikators ist zudem die Datenverfügbarkeit.

### 2.2.1 Faktor Wetter

Vielmals begegnet in der bisherigen Literatur Betrachtung sind Wetter Daten, so z.B. bei Holmgren et al. (2017), Broucke et al. (2019), Li et al. (2015). Literatur, die sich speziell mit diesem Zusammenhang beschäftigt findet sich bei Wessel (2020). Dabei hat er sich z.B. im Gegensatz zu Nankervis (1999), der sich auch mit dem Zusammenhang von Wetter und Fahrrad Aufkommen beschäftigt, gezielt mit dem Zusammenhang von Wettervorhersagen beschäftigt. Auch Meng et al. (2016) untersucht diesen Zusammenhang.

Jedoch während Meng et al. (2016) Daten einer Umfrage von 553 Fahrradfahrern in Singapore verwendet, stützt sich Wessel (2020) auf umfangreichere Daten von 188 Fahrradzählstationen in 37 deutschen Städten, die stündlich zählen. Daten zum aktuellen Wetter stammen vom Deutschen Wetterdienst. Daten über Wettervorhersagen stammen aus der ARD Mediathek, der abendlichen Tagesschau um acht Uhr. Die Aufzeichnungen der Wettervorhersagen werden dabei manuell ausgewertet, wobei der deutsche Raum in sechs Hemisphäre Nordwest, Nordost, mittlerer Westen und mittlerer Osten so wie Südwest und Südost unterteilt wurde und die Städte in die korrespondierende Hemisphäre eingeteilt worden sind nach folgenden Wetter Klassifikationen: klarer Himmel, leichte Bewölkung, schwere Bewölkung, Regen, Schneefall,

Gewitter und zusätzlich Wind, Rutschgefahr, Vereisungen, Überflutung und generelle Warnungen. Neben dieser manuellen Einschätzung der Bewölkung wurde zusätzlich eine digitale automatisierte Einschätzung verwendet, die sich auf die Dunkelheit der Pixel des Kartenmaterials bezieht.

Aufbauend auf diesen Daten nutzt Wessel (2020) ein log-lineares Regressionsmodell und ein negativ binomiales Regressions Modell. Das letztere Modell geht auf Hausman et al. (1984) zurück und ist im Besonderen für ganzzahlige Regressoren nützlich, wie es hier der Fall ist. Der Bestimmtheitswert dieser verschiedenen Modelle  $R^2$ , die teils abweichende Variablen verwenden, liegt zwischen 75,9 % und 78,5 %. Gerade der Schritt mehrere Städte in ein Modell zu bringen ist interessant. Saha et al. (2018) hat zwar eine Betrachtung für einen ganzen Bundesstaat angefertigt, hat Vorhersagen jedoch nur auf Makro Ebene getroffen. Das Modell von Wessel (2020) findet hingegen auf der Mikro Ebene der einzelnen Zählstationen statt, und zeigt, dass doch trotz Unterschiede in der städtischen Infrastruktur und Fahrkultur präzise Vorhersagen machbar sind, denn der Einfluss des Wetters ist einer der wesentlichen Prädiktoren und gehört auf jeden Fall in ein Modell, dass das Fahrrad Verkehrs Volumen vorhersagen möchte. Außerdem verwendet sein Modell auch Schul- und Semesterferien, sowie Feiertage als Prädikator, die ebenfalls stark ins Gewicht fallen.

Auch wenn das Modell von Wessel (2020) zu guten Vorhersagen das Radverkehrs führt, muss diese Masterarbeit darauf verzichten, Information der Wettervorhersagen händisch in Variablen zu übersetzen, da dies mit einem enormen Arbeitsaufwand verbunden wäre. Es ist bedeutend einfacher Daten des deutschen Wetterdienstes zu nutzen, die sicherlich ähnlich gute Vorhersagen liefern, denn der Unterschied zwischen beiden Varianten ist auch bei Wessel (2020) nicht allzu groß.

Zusätzlich beschäftigen sich Goldmann and Wessel (2021) mit der sogenannten Wetterelastizität des Radverkehrs. Diese beschreibt die Reaktion von Radfaherinnen auf Wetterereignisse und unterscheidet sich in unterschiedlichen Städten. Ist der Radverkehr in einer Stadt inelastisch, so bedeutet das, dass selbst bei Regen noch relativ viele Radfahrer auf den Straßen unterwegs sind, wo in elastischen Städten unter selben Bedingungen bedeutend weniger Rad-

fahrer unterwegs wären. Ein Beispiel für eine wetter inelastische Stadt ist z.B. Münster.

### 2.2.2 Faktor Feinstaubbelastung

Die Feinstaubbelastung ist ein weiterer möglicher Prädikator, das zeigen Zhao et al. (2018), Gao and Chen (2022) und Hong J (2022). Im Speziellen setzt sich Hong J (2022) mit der Nutzung von Bike Sharing Systemen in Seoul unter der Aussetzung von Feinstaubbelastung auseinander. Er argumentiert, dass durch die Corona Krise, mehr Menschen vom öffentlichen Verkehr auf Fahrräder umgestiegen sind, um Menschenmassen in U-Bahnen aus dem Weg zu gehen, dabei aber einer höheren Feinstaubbelastung ausgesetzt waren. Ob Fahrradfahrer der Feinstaubbelastung aus dem Weg gehen untersucht er mit einer linearen Regression, in der das PM<sub>2.5</sub> Level als Maßstab der Feinstaubbelastung herangezogen wird. Daneben berücksichtigt er Wochentage, Jahreszeiten und Wetterdaten wie Wind, Bewölkung und Temperatur. Demnach hat das PM<sub>2.5</sub> Level einen negativen Effekt auf die gesamte Dauer aller Fahrrad Touren auf einem Signifikanz Level von 0.05.

Grundsätzlich sind Daten zur Feinstaubbelastung in Deutschland an vielen Stellen erhältlich durch die Seite <https://openseNSEmap.org> zugänglich, doch ist zu bezweifeln, dass hier der Aufwand im Verhältnis zum Nutzen stünde, denn in den allermeisten deutschen Städten dürften grundsätzlich andere Verhältnisse vorherrschen als in Seoul. So lag nach OECD Daten (<https://data.oecd.org/air/air-pollution-exposure.htm>) die durchschnittliche Feinstaubbelastung PM<sub>2.5</sub> 2019 in Südkorea bei 45.2 und in Deutschland bei 11.9 Mikrogramm per m<sup>3</sup>. Zusätzlich ist zu bedenken, dass ohne öffentliche Smog Warnungen Fahrradfahrer keine Möglichkeit haben, auf gestiegene Feinstaubbelastungen zu reagieren, oder diese bei Auswahl ihrer Fahrradrouten zu berücksichtigen. Demnach dürfte die Feinstaubbelastung als Faktor zur Vorhersage des Fahrrads Verkehrsaufkommen in Deutschland irrelevant sein.

### 2.2.3 Faktor Corona Maßnahmen

Es ist vorstellbar, dass Corona Lockdowns zu einer Verzerrung des allgemeinen Verkehrs geführt hat, was ebenso für Fahrräder gelten wird. Diese Effekte untersuchen Möllers et al. (2021) mittels Daten von Zählstation 10 deutschen Städten. Während Lockdowns zu einer Reduzierung von Fußgängern führte, ist der Effekte auf Fahrradfahrer uneindeutig. Bei dieser Betrachtung konzentrierten sie sich auf die erste Welle. Für ihr Modell verwenden sie Daten des RKI, dass tägliche Daten über neue Fälle zur Verfügung stellt pro Region. Zusätzlich kontrollieren sie auch die örtlichen Maßnahmen durch die Öffnungszeiten örtlicher Geschäfte und Schulen. Ihre Analyse zeigt, dass die Anzahl der Fahrradfahrer unter der Woche abgenommen hat, aber an Wochenenden zu genommen hat.

### 2.2.4 Faktor der städtischen Unterschiede

Die Studien von Wessel (2020), Möllers et al. (2021) und Goldmann and Wessel (2021) zeigten bereits, dass die Verwendungen von Daten aus verschiedenen Städten immer noch zu guten Vorhersagen führen, selbst wenn Unterschiede in der Höhe des Fahrradverkehrs in diesen Städten existieren. Dieser Punkt spielt für diese Hausarbeit eine große Rolle, denn die Daten von verschiedenen Städten zu vereinen, wäre ein hilfreicher Weg, um mehr Stichproben zu erhalten.

Damit die Aussagekraft des Modells jedoch in den Unterschieden der Städte nicht untergeht, empfiehlt es sich, Variablen in das Modell mit aufzunehmen, die diese Unterschiede erklären können. Goldmann and Wessel (2021) machen dies vor. Sie versuchen Unterschiede der Städte in der Wetterelastizität des Radverkehrs zu begründen. Dazu nutzen sie nicht nur Wetterdaten und Daten zu Feier- und Ferientagen sondern auch Daten zur demographischen Bevölkerungsstruktur der jeweiligen Stadt, Daten zum Autobesitz, die Unfallrate, die Verkehrsdichte, Radwegdichte und die Dichte des öffentlichen Nahverkehrs. Diese Auswahl ergibt einen guten Anhaltspunkt dafür, welche Variablen der Datensatz dieser Hausarbeit mit aufnehmen sollte. Weitere Angaben und Ergebnisse hierzu finden sich im Kapitel 3.3.

### 2.2.5 Faktor Radverkehrsunfälle

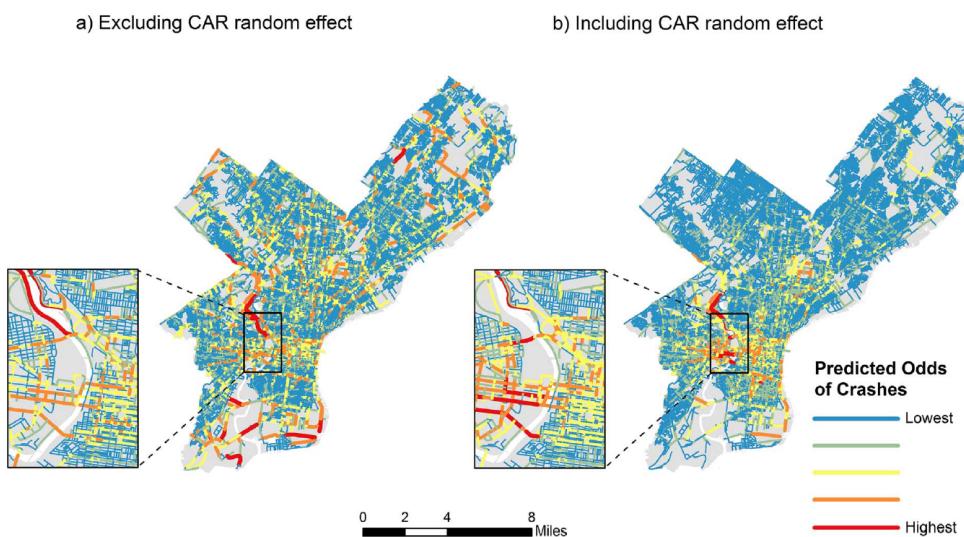
Da diese Arbeit beabsichtigt die Auslastung von Fahrradwegen und das allgemeine Verkehrsaufkommen von Fahrrädern hervorsagen zu können, haben sich die vorherigen drei Abschnitte auf Literatur konzentriert, die Ähnliches versuchte mit unterschiedlichen Datengrundlagen. Interessant wäre es, die Erkenntnisse aus diesen Daten mit dem Aufkommen von Fahrradunfällen zu verbinden, um z.B. kritische Stellen im Stadtbild als potentielle Verkehrsunfallorte ausfindig zu machen. Um in Erfahrung zu bringen, ob man durch statistische Methoden eine Antwort auf diese Nebenfrage finden kann, ist es wichtig, Literatur zu betrachten, die Unfallstatistiken als Datengrundlage verwendet.

Eine solche Studie stammte z.B. von Vandenbulcke et al. (2014), die der Frage nachgehen, wie Infrastruktur Unfälle hervorruft. Dazu verfolgen sie einen räumlichen bayesianischen Modellierungsentwurf mit Daten aus Brüssel. Ergebnisse dieser Studie sind zB, dass das Gefahrenpotential für Fahrradfahrer steigt, wenn sich Straßenbahnschienen auf der Fahrspur befinden, Brücken ohne separate Fahrradwege, komplizierte Kreuzungen, die Nähe zu Einkaufszentren oder Garagen und ein erhöhtes Bus Aufkommen.

Prati et al. (2017) verwenden ebenfalls einen bayesianischen Anssatz. Zum einem verwenden sie eine bayesianische Netzwerk Analyse und einen Chi-squared Automatic Interaction Detection (CHAID) Entscheidungsbaum, um die Schwere von Fahrradunfällen anhand von Charakteristiken wie Geschlecht und Alter der Fahrradfahrer, Art des Fahrzeuges des Unfallpartners, Art des Straßenabschnitts etc. vorherzusagen. Vorteil der CHAID Analyse ist es, dass die Verästelungen des Entscheidungsbaums Gabelungen mit mehr als zwei Pfaden zulässt. Als Datengrundlage verwenden Prati et al. (2017) italienische landesweite Unfallstatistiken von 2011 bis 2013. Das erlaubt den Vorteil einer großen Stichprobengröße mit 49621 Unfällen, bei denen mindestens ein Fahrradfahrer verletzt worden ist. Ein Validation Set Ansatz mit einem 70:30 Split vergleicht beide statistischen Modelle. Im Ergebnis zeigt sich, dass die wichtigsten Faktoren für die Schwere eines Unfalls der Straßentyp, der Unfalltyp, das Alter des Radfahrers, die Straßenbeschilderung, das Geschlecht

des Radfahrers, der Typ des gegnerischen Fahrzeugs und der Monat sind. Wobei das CHAID Modell im Test Set zu 98 % akkurat war.

Auch Kondo et al. (2018) verfolgen einen bayesianischen Ansatz, entwickeln aber ein räumliches Modell. Sie untersuchen wie Fahrradstreifen das Unfallrisiko senken können. So führen getrennte Fahrradstreifen zu einem 48 % geringeren Risiko an einer Viererkreuzung und zu einem 43 % geringeren Risiko auf Straßen mit hohem Verkehr. Zu diesen Erkenntnissen kommen sie auf Grundlage eines Datensatzes aus Philadelphia mit über 37000 beobachteten Unfällen zwischen 2011 und 2014 mit Charakteristiken der Straßenbeschaffenheit. Dazu verwendeten sie ein bayesianisches konditionelles autoregressives Logit Modell. Als unabhängige Variablen verwenden sie Straßen Charakteristiken, Charakteristiken von Kreuzungen und einen Traffic Indikator. Zu den Charakteristiken zählen z.B. die Anzahl der Keuzungszugänge, Stopnzeichen oder auch Fußgängerüberwege. Mithilfe dieser Daten ergibt sich ein Bild wie in Abbildung 2.3 zu finden: Wie viele der bisher genannten Studien



*Fig. 2.3: (a und b). Berechnete Wahrscheinlichkeiten für Radunfälle in Philadelphia*  
**Quelle:** Kondo et al. (2018)

konzentrierte sich Kondo et al. (2018) auf eine Stadt, was ein nicht unwesentliches Problem darstellen kann, denn unterscheidet sich der Straßenverkehr von Stadt zu Stadt, wie auch Goldmann and Wessel (2021) zeigen. Gera-

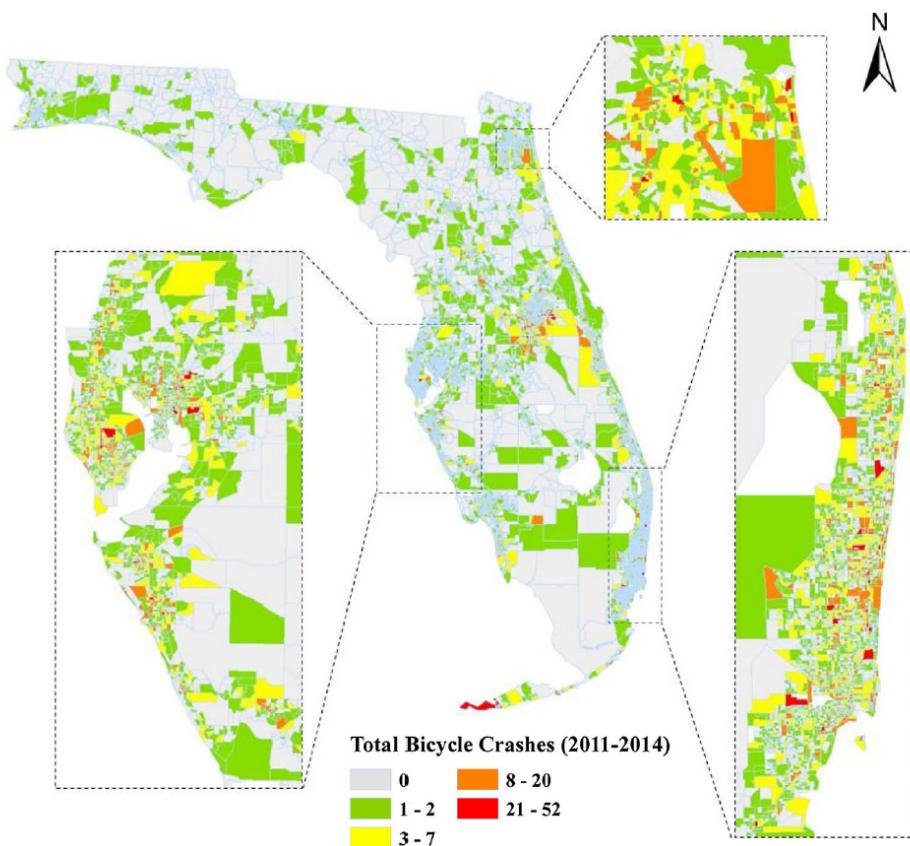
de die Ergebnisse aus Philadelphia, einer der fahrradfreundlichsten Städte der USA, lassen sich eventuell nicht eins zu eins auf andere Städte der USA übertragen, da Autofahrer in Philadelphia durch den höheren Verkehr von Fahrrädern an die Rücksichtnahme gewöhnt sein könnten, die Autofahrer aus Erfahrung walten lassen. Deswegen wäre es ein interessanter Schritt, diese Beobachtungen auf einer Makro Ebene zu tätigen, um zu sehen, welche Faktoren Städte übergreifend für Fahrrad Sicherheit sorgen.

Genau dies machen Saha et al. (2018), in dem Sie eine Analyse Floridas auf Zensus Block Größe vornehmen. Bei dieser Analyse stellen sie eine räumliche Konzentration und keine Gleichverteilung von Unfällen fest. Um dies zu erklären nutzen die Autoren ein bedingtes autoregressives Modell mit bedingten Variablen der Demographie, sozioökonomischen Daten, Straßen Infrastruktur und Radverkehr Charakteristiken. Davon wurden 21 Variablen als einflussreich identifiziert z.B. Bevölkerung, Alterskohorten, Autobesitz von Haushalten, Straßennetz Dichte oder Fahrradtour Intensität. Für das Modell werden Zensus Daten von 2011 bis 2014 verwendet, zu sehen auch in der Abbildung 2.4. Gerade die letzten zwei Studien zeigen, wie interessant die räumliche Verteilung von Fahrradunfällen ist und Kondo et al. (2018) zeigen auch das Verkehrsvolumen als Faktor der Unfallwahrscheinlichkeit.

Neben all dem ist interessant, dass Kondo et al. (2018) darauf verweisen, dass Unfallsicherheit ein starker Prädikator sei für den Fahrradverkehr laut Pucher et al. (2010), Thomas and DeRobertis (2013) und Winters et al. (2010). Dies ist eine gute Überleitung auf das nächste Thema. Die bisherige Betrachtung beschäftigte sich mit den Daten, die man verwenden kann für die abhängige Variable des Modells, dass am Ende dieses Projekts entstehen soll. Doch für eine zuverlässige Vorhersage braucht es Prädiktoren. Diese sind in der Literatur zu finden.

### 2.2.6 Sonstige Faktoren

Eine weitere interessante Datenquelle, die wir z.B. schon bei Alattar et al. (2021) kennen gelernt haben, ist Open Street Map. Dies ist ein 2004 gegründetes gemeinnütziges Projekt, dass das Ziel verfolgt Kartenmaterial zu



*Fig. 2.4: Räumliche Verteilung aller Fahrradunfälle (2011–2014) nach Zensus Blöcken in Florida.*

**Quelle:** Saha et al. (2018)

sammeln und online allen frei zur Verfügung zu stellen. Hier sind Daten über Straßen, Eisenbahnen, Flüsse, Wälder, Häuser und vielem weiteren zu finden. Auch Carl and Dror (2015) nutzt die frei verfügbaren Daten zu Höhenmetern des Geländes von Open Street Map, um die Planung von mehrtägigen Fahrradtouren zu erleichtern, die der Erholung dienen. Deswegen ist dieses Paper für die Frage dieser Hausarbeit weniger interessant, die Idee, diese Art von Daten zu verwenden, aber könnte hilfreich sein. Dass aber die Topographie eine Rolle auch für den Pendler Verkehr spielt, zeigt Rietveld and Daniel (2004).

### 2.3 Forschungslücken und Anknüpfungspunkte

Was die meisten der hier gezeigten Studien hier gemein haben, ist dass sie sich auf eine reine Zeitreihenanalyse beschränkt, jedoch die Daten nicht dazu nutzen die räumliche Verteilung der Daten zu untersuchen. Es gibt eine wenige Beispiele die das zwar tuen (Namen nennen), jedoch nie für den ganzen Verkehr, sondern nur für den Teil des Verkehrs, der im Rahmen von Leihradsystemen geschieht. Auch die Kontrolle der Routenwahl durch GPS Tracking, stellt immer nur einen verzerrten Teil des gesamten Verkehrs da, entweder durch zu geringe Stichprobengrößen oder durch die Selbstselektion der Studienteilnehmer.

Das lässt nur die Forschung übrig, die mit Daten von Zählstationen rechnet. Diese verfolgt aber meist das Ziel kausale Einflüsse auf den Radverkehr zu untersuchen wie bei (...). Andere Studien treffen zwar Vorhersagen, aber eben nicht in eine räumliche Dimension hinein, wie bei (Holmgren). Die Forschungsfrage dieser Masterarbeit erfordert aber eine räumliche Interpolation, um Vorhersagen für ein gesamtes Stadtgebiet zu treffen und nicht über einzelne Stationen, die vorher so beobachtet worden. Eine kausale Interpretation ist hierfür nicht zwangsläufig notwendig, wenn auch interessant. Außerdem würde eine kausale Interpretation die Verwendung von neuronalen Netzwerken ausschließen.

Nur wenige Studien untersuchen die Daten mehr als einer Stadt, mit ausnahme von (...). Aber die Verwendung von mehreren Städten wäre zwingend notwendig. Denn wie man im Vergleich sieht, bieten Studien mit Radzählstellen oft weniger Beobachtungspunkte pro Stadt, als z.B. Studien die GPS Tracking Daten verwenden oder Daten von Leihsystemen. Um diesen Nachteil auszugleichen, ist es ratsam, Daten mehrere Städte zu verwenden. Die Studie von ... zeigte, wie man die Unterschiede zwischen Städten hervor heben kann ist. So ist es ratsam demographische und soziale Daten über die Stadtstrukturen in einen Datensatz mit aufzunehmen. Bei ... zeigten diese Daten Unterschiede in der Wetterelastizität des Fahrradverkehrs auf. In dieser Studie würden die stadtbezogenen Daten Unterschiede im Radverkehr selbst kenntlich machen.

Zuletzt erfordert die Forschungsfrage Stationsbasierte Daten, möchte man Unterschiede im Radverkehrsaufkommen innerhalb der selben Stadt erkennen und auch vorhersagen können. Nur daraus ergeben sich räumliche Unterschiede im Stadtbild. Eine Studie, dient hier zB als Vorbild, in dem sie Daten von POIs in Google Streetview verwendet. ... Verwenden hierbei Bilderkennung. Soweit würde diese Masterarbeit nicht gehen. Allerdings könnte man Entfernung von Zählstationen und einigen POIs im Stadtbild mit in das Modell aufnehmen. Dies allgemein gibt einen guten Ausblick auf das kommende Kapitel im Allgemeinen. Anhand der bestehenden Literatur ließ sich festmachen, welche Datenstruktur notwendig ist, um die Forschungsfrage zu beantworten. Das kommende Kapitel gibt einen Einblick in den endgültigen Datensatz.

### 3. ZUSAMMENSETZUNG DES DATENSATZ

Das vorherige Kapitel zeigte den aktuellen Stand der Forschung zum Aufkommen des Fahrradverkehrs und endete mit einer Einschätzung, an welchen Studien sich diese Masterarbeit orientieren muss, um die gestellte Forschungsfrage zu beantworten. Darauf baute auch die Beschaffung von Daten für das Modell dieser Arbeit auf, angefangen über die Daten der Fahrradzähler, Daten zum Wetter, demographischen Daten und Daten der vorhandenen Infrastruktur erhoben durch Open Street Map und dazu gehörigen Points of Interest. Einen tieferen Einblick über die Datenbeschaffung und die Verteilung von Daten mit zugehörigen Abbildung beinhaltet dieses Kapitel.

Die Datenbeschaffung an sich beschränkt sich allein auf Deutschland aus Gründen der Einfachheit. Bestimmte Daten wie Daten zum Wetter und zur Demographie lassen sich so allein von einer Datenquelle beschaffen in diesem Fall dem Deutschen Wetter Dienst und dem statistischen Bundesamt DESTATIS. Der Nachteil davon ist, dass wiederum alle Aussagen des Modells allein für Deutschland gültig sind, da dieser Rahmen die Evaluierung für andere Regionen der Welt nicht zu lässt.

#### 3.1 *Fahrradzähler*

Notwendige Grundlage der Forschung sind Daten von Fahrradzählstationen. Dankbarerweise stellen diese Daten viele Kommunen öffentlich zur Verfügung oder teilen diese auf Nachfrage. In welchen Städten sich Fahrradzähler finden ließen, die in das Modell aufgenommen werden konnten, zeigt die Abbildung 3.1. Alle Fahrraddaten sind stündlich aufgelöst. D.h. die Anzahl der Fahrradfahrer, die eine Zählstation passierten, wurde innerhalb einer vollen Stunde aufsummiert.

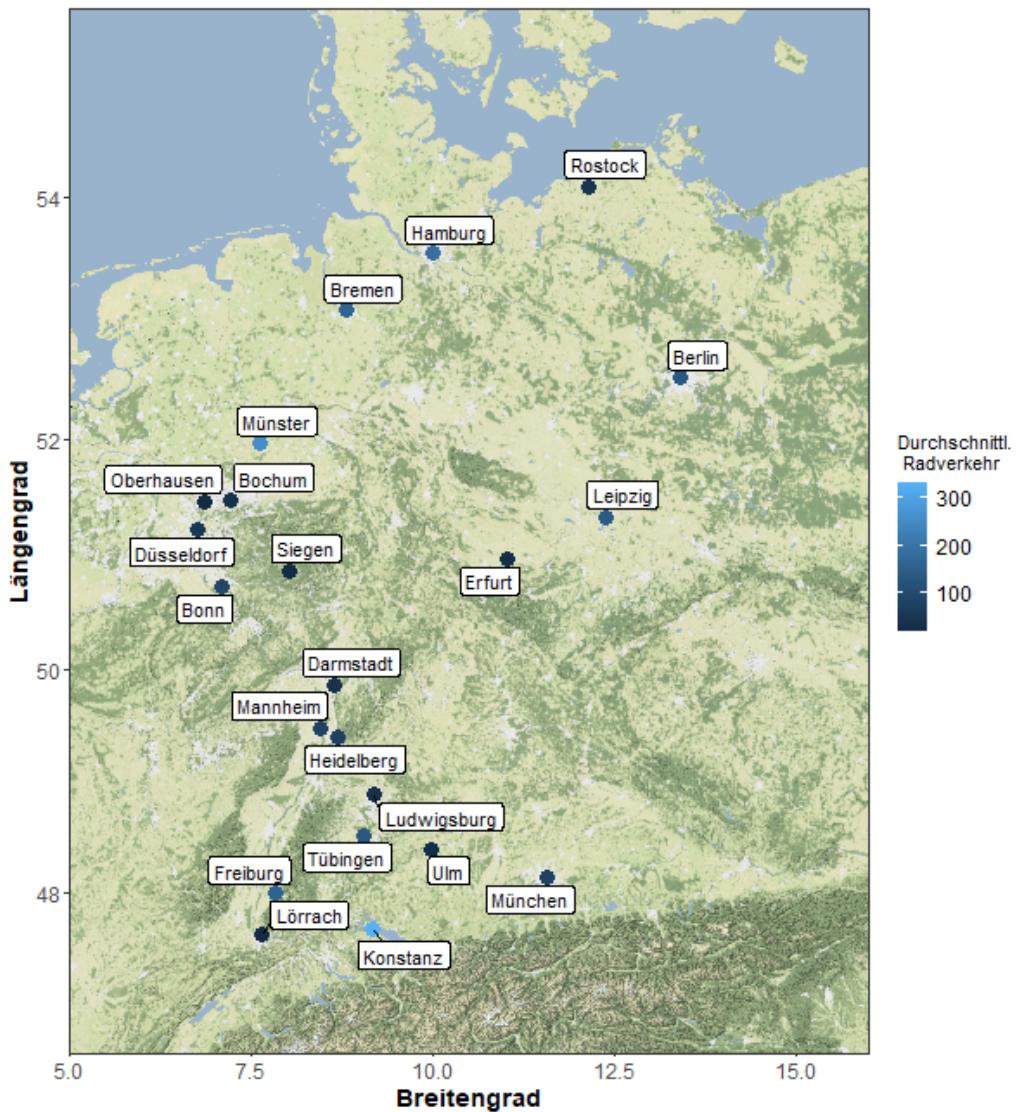


Fig. 3.1: Städte die im Datensatz vertreten sind.

Ein grundsätzliches Problem des Datensatzes, dass sich so leicht auch

nicht beheben lässt, dass zum einem Großstädte mehr Fahrradzähler errichten, zum anderem ihre Daten oft auch leichter zugänglich machen. Die Abbildung 3.2 zeigt deutlich, dass Großstädte mit mehr als 300 Tsd Einwohnern im Datensatz überrepräsentiert sind. Deshalb sollte man bei Vorhersagen für Städte mit weniger als 300 Tsd Einwohnern vorsichtig in der Interpretation sein.

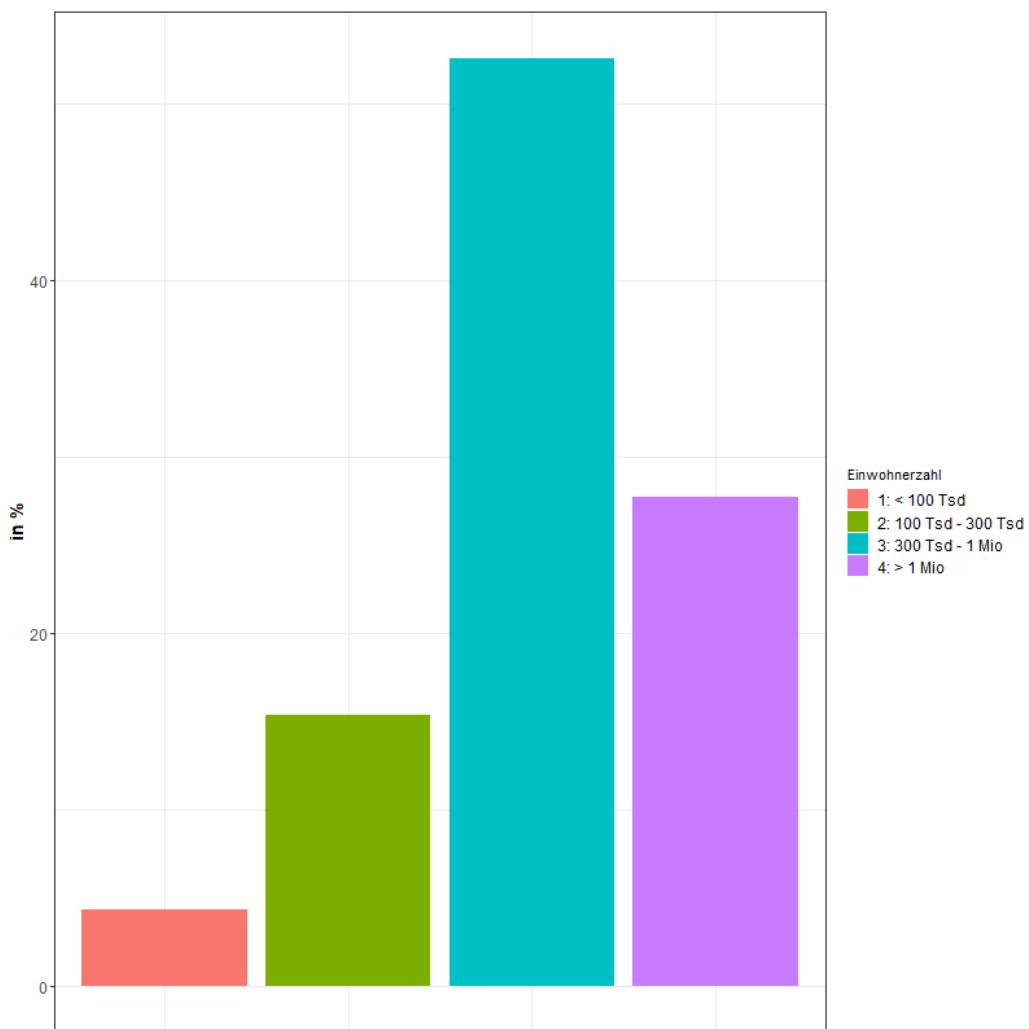


Fig. 3.2: Verteilung der Beobachtung nach Einwohnergröße der Städte

Die Datenquelle sind die Kommunen selbst. Eine Veröffentlichung aller Daten ist dabei nicht möglich, weil die Kommunen unterschiedliche Bedingungen für die Verwendung der Daten gestellt haben. Der Betrachtungszeit-

raum der Daten reicht von 2012 bis 2022 in einigen Fällen. Einige wenige Zählstellen in Hamburg und Siegen wurden erst 2022 aufgestellt. In allen anderen Städten reichen die Beobachtungen aber nur bis 2021. Die Abbildungen 3.3 zeigt wie sich die Daten über die Zeit verteilen. Dabei ist festzustellen, dass je weiter wir in die Vergangenheit gehen, desto weniger Daten finden wir, da nur wenige Zählstationen seit 2012 im Betrieb sind. (An dieser Stelle könnte man noch über die Fahrradzähler in Hamburg schreiben)

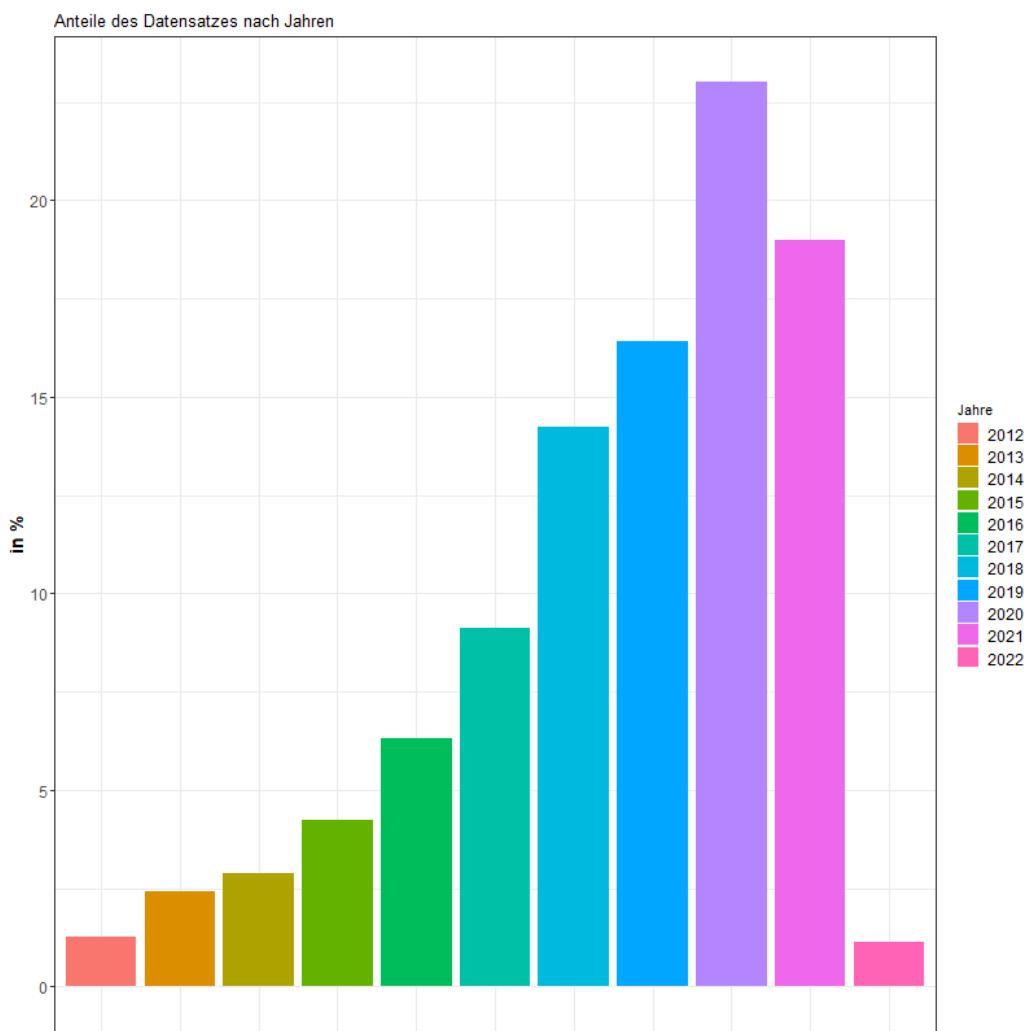


Fig. 3.3: Verteilung der Beobachtung nach Jahren

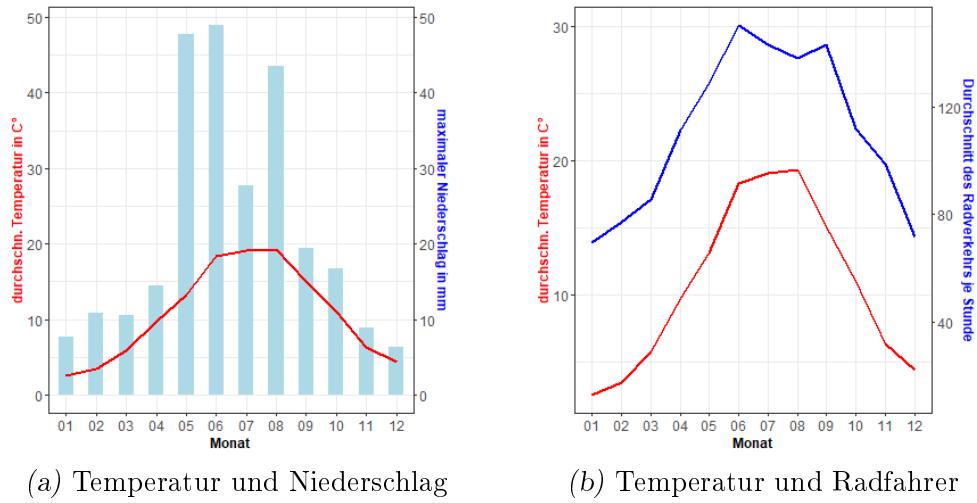


Fig. 3.4: Wetterdaten im Überblick

### 3.2 Wetterdaten

Daten zum Wetter stammen einheitlich vom Deutschen Wetterdienst. Dabei wurden die Fahrraddaten einer Stadt immer mit den Wetterdaten der nächstgelegenen Stadt verwendet. Nur Mannheim und Heidelberg teilen sich die Daten einer Wetterstation, da beide Städte sehr nah beieinander liegen. Insgesamt wurden Daten in den Datensatz mit aufgenommen zum Niederschlag in mm, zur Lufttemperatur in 2 m Höhe in Grad Celsius, zur Wolkenbedeckung in Achteln, zur relativen Feuchte in % und zur durchschnittlichen Windgeschwindigkeit. Genau wie die Fahrraddaten sind die Wetterdaten stündlich aufgelöst. Einen jährlichen Durchschnittsverlauf über alle Daten zeigt Abbildung 3.4a. Weiterhin zeigt die Abbildung 3.4b den Zusammenhang zwischen Radfahrer und Monatstemperaturen.

### 3.3 Demographische und soziale Statistiken

In diesem Abschnitt sammeln sich verschiedene Daten, die Unterschiede in den Städten in ihrer sozialen und demographischen Struktur in ihrer Bevölkerung hervorheben. Daten dafür stammen aus zwei Quellen. Die meisten Variablen nutzen als Datenquelle das statistische Bundesamt. Eine Varia-

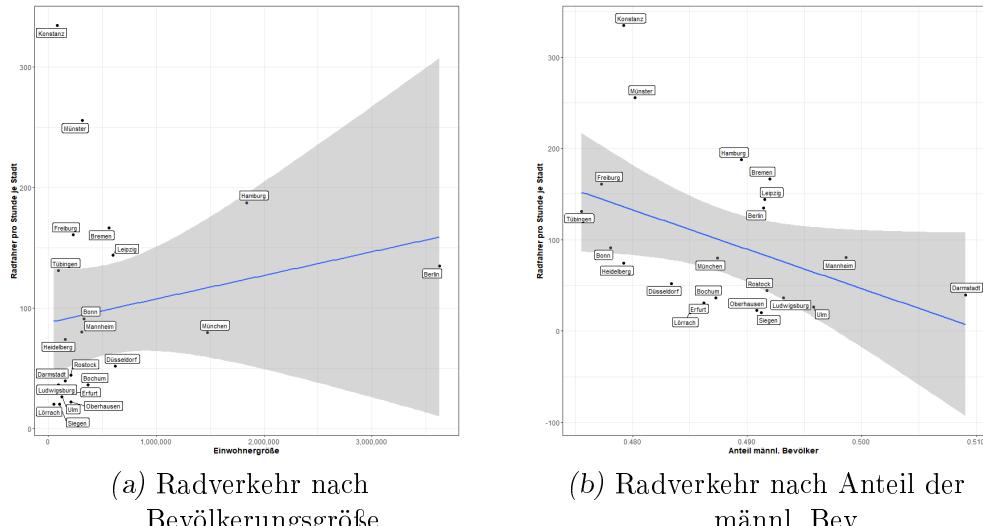


Fig. 3.5: Verhältnis von Bevölkerung und Fahrradaufkommen

ble bezieht sich auf den ADFC Fahrradklimaindex. Für beide Quellen gilt, dass Daten für das Jahr 2022 noch nicht vorhanden waren, da das Jahr zum Zeitpunkt der Recherche noch nicht abgeschlossen war. Für die wenigen Beobachtungen aus Hamburg z.B., die aus dem Jahr 2022 in den Datensatz mit aufgenommen worden sind, gilt, dass für diese Beobachtungen angenommen wurde, dass Variablen, die nicht erneuert werden konnten, konstant blieben. Das betrifft insgesamt jedoch nur wenige Beobachtungen, wie auch Abbildung 3.3 zeigt.

### 3.3.1 Statistisches Bundesamt

Das statistische Bundesamt, stellt verschiedene Tabellen zur Verfügung, die wertvolle Einblicke in die Unterschiede der Gemeinden geben, in denen sich die verschiedenen Fahrradzählstationen befinden.

Eine wichtige grundlegende Quelle ist das Gemeindeverzeichnis für alle politisch selbständigen Gemeinden (mit Gemeindeverband) in Deutschland, das jedes Jahr veröffentlicht wird. Dieses Verzeichnis beinhaltet nicht nur Landkreise und kreisfreie Städte, sondern hält auch Zahlen zu jeder Gemein-

de bereit. Dabei sind Daten zur Fläche in km<sup>2</sup>, zur Einwohneranzahl, sowohl männlich, weiblich wie auch insgesamt vorhanden. Diese fanden auch ihren Weg ins Modell. Wie sich Einwohnergröße und der Anteil an Männern in der Bevölkerung zum Aufkommen an Fahrradfahrern verhalten, kann man in der Abbildung 3.5 sehen. Außerdem verwendet das Modell die angegebenen Längen- und Breitengrade, aus denen der jeweilige Abstand der Fahrradzählstation zum Stadtzentrum berechnet wird, auch dargestellt in Abbildung 3.6. Das Modell greift dabei auf das Gemeindeverzeichnis von 2012 bis 2021 zurück.

Weitere Daten zur demographischen Verteilung der Bevölkerung in den jeweils betroffenen Landkreisen und kreisfreien Städten stammen aus der Tabelle 12411-0017 zur Bevölkerung nach Kreisen, Stichtag, Altersgruppen von Destatis (2022b). Mit diesen Daten ließ sich der Anteil der Bevölkerung berechnen, der jünger als 18, 25, 30 und älter als 40 und 60 ist. Wie sich das Radverkehrsaufkommen zum Anteil der Bevölkerung unter 30 verhält nach Städten zeigt auch die Abbildung 7.1 im Anhang.

Auch eine wertvolle Statistik bietet die Tabelle 46251-0020 über den Kraftfahrzeugbestand nach Kreisen, Stichtag, Kraftfahrzeugarten von Destatis (2022c). Mithilfe der Einwohneranzahl je Kreis ließ sich die Quote an Kraftfahrzeugen je Person berechnen. Außerdem zeigt die Tabelle 12521-0040 über die Anzahl der Ausländer nach Kreisen, Stichtag und Geschlecht von Destatis (2022a) ebenfalls wichtige Einblicke zu sehen in Abbildung 3.7.

### 3.3.2 ADFC Fahrradindex

Neben den stadspezifischen Daten, die von den Quellen des statistischen Bundesamtes stammen, nutzt der Datensatz die Daten zum ADFC Fahrradklimaindex. Der Allgemeine Deutscher Fahrrad-Club ist ein eingetragener Verein mit Sitz in Berlin und Bremen, der sich für die Interessen von Fahrradfahrern einsetzt und stand 2022 220.000 Mitglieder in Deutschland zählt. Im Zweijahresrhythmus veröffentlicht der ADFC den per Umfrage ermittelten Fahrradklimaindex. 2022 wurden z.B. 238 Tsd Menschen befragt aus verschie-

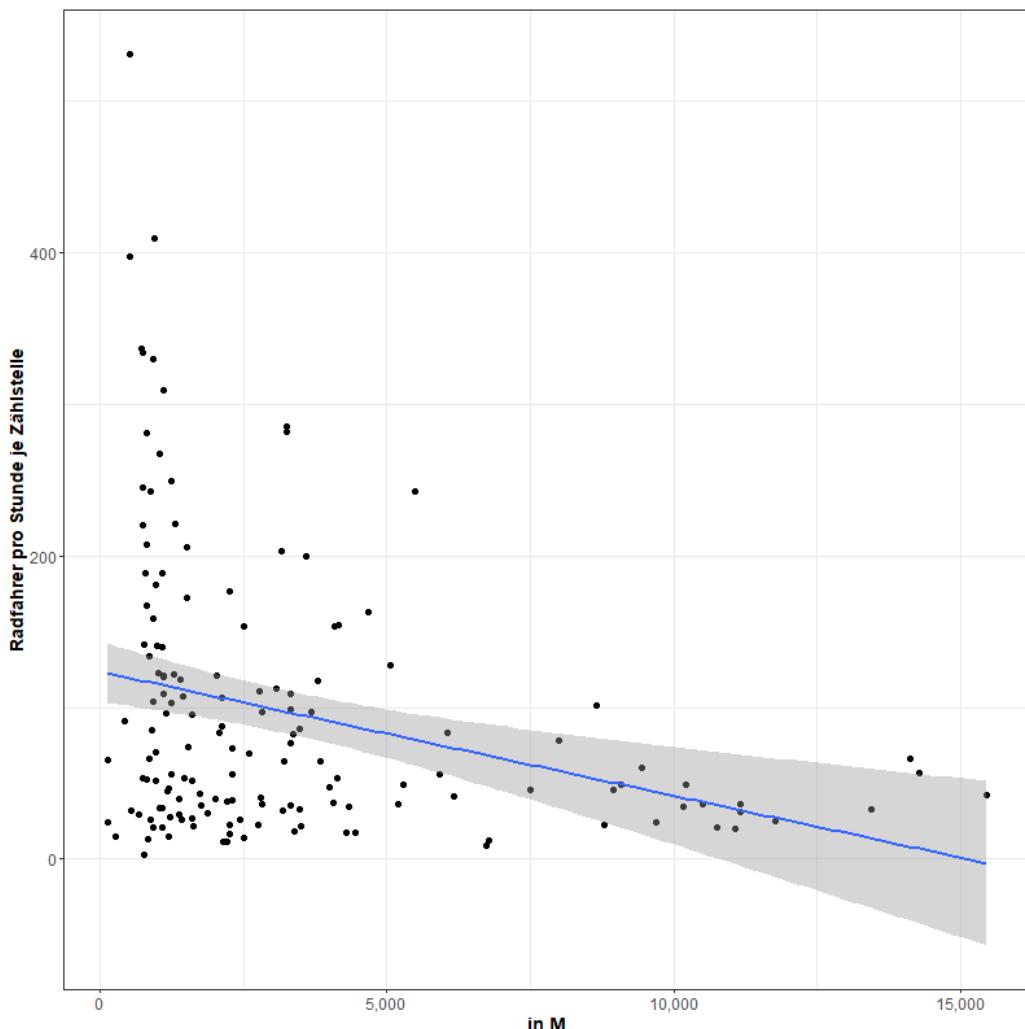


Fig. 3.6: Verteilung des Fahrradaufkommens in Entfernung zum Stadtzentrum

denen Städten. Neben Fragen zum persönlichen Fahrradfahren enthält der Fragebogen zum Fahrradklima Index Fragen betreffend des Fahrradklimas, dem Stellenwert des Radverkehrs in der jeweiligen Stadt, der Sicherheit, dem Komfort und der Infrastruktur sowie der Radverkehrsnetze. All diese Antworten fließen in eine Endnote, die sich zwischen 1 und 5 befindet, wobei 1 am Besten und 5 am schlechtesten ist. Wie sich das Fahrradklima ins Verhältnis zum aufgezeichneten Radverkehr ins Verhältnis nach Städten setzt, zeigt die Abbildung 7.2 im Anhang.

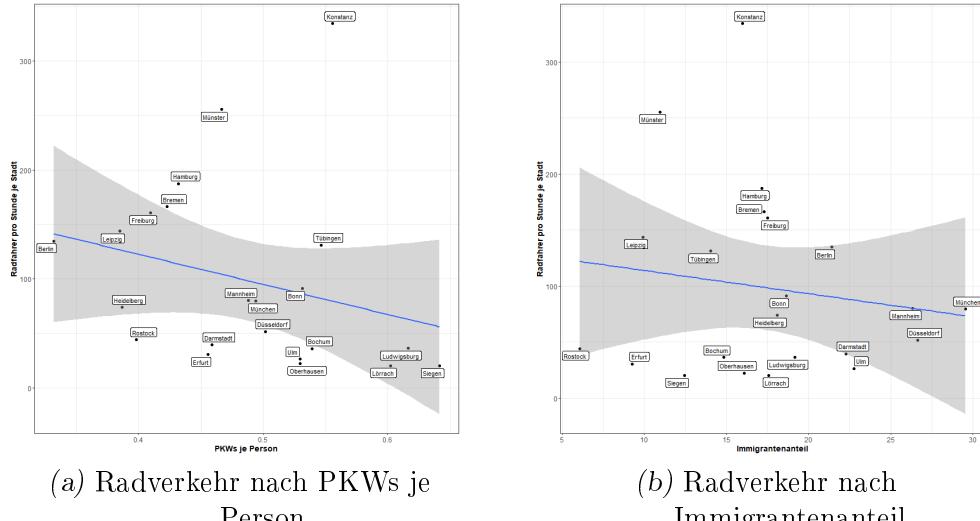


Fig. 3.7: Verhältnis von Bevölkerung und Fahrradaufkommen

### 3.4 Corona Daten

Wie Möllers et al. (2021) bereits zeigten, hatten die Corona Maßnahmen unterschiedliche und nicht ganz eindeutige Auswirkungen auf den urbanen Radverkehr. Dennoch ist es ratsam, Variablen in das Modell mitaufzunehmen, die Ausprägung von Corona Maßnahmen berücksichtigen. Wie bei Möllers et al. (2021) ließen sich dafür die Corona Inzidenzzahlen verwenden, oder aber eine Dummy Variable für Lockdowns.

Bei beiden Ansätzen stellen sich Probleme ein. War die Corona Inzidenz zu Beginn recht klein, waren die Auswirkungen für den Verkehr dennoch drastisch. Die Abbildung 3.8 gibt darüber einen Überblick. Deswegen beinhaltet der Datensatz zusätzlich eine Dummy Variable für die zwei bundesweiten Corona Lockdowns vom 22.3.2020 bis zum 4.5.2020 und vom 2.11.2020 bis zum 14.2.2020. In diese Zeitphasen fallen verschiedene Maßnahmen zur Kontaktbeschränkung, die jeweils bundesweit stattgefunden haben. Dabei kann es regional zu Abweichungen kommen, die sich im Nachhinein nicht ohne immensen Aufwand nachvollziehen lassen.

Diese Schwäche sollen die Daten zur Corona Inzidenz ausgleichen, die lokal mit Maßnahmen zur Kontaktbeschränkung korrelieren. Die Daten da-

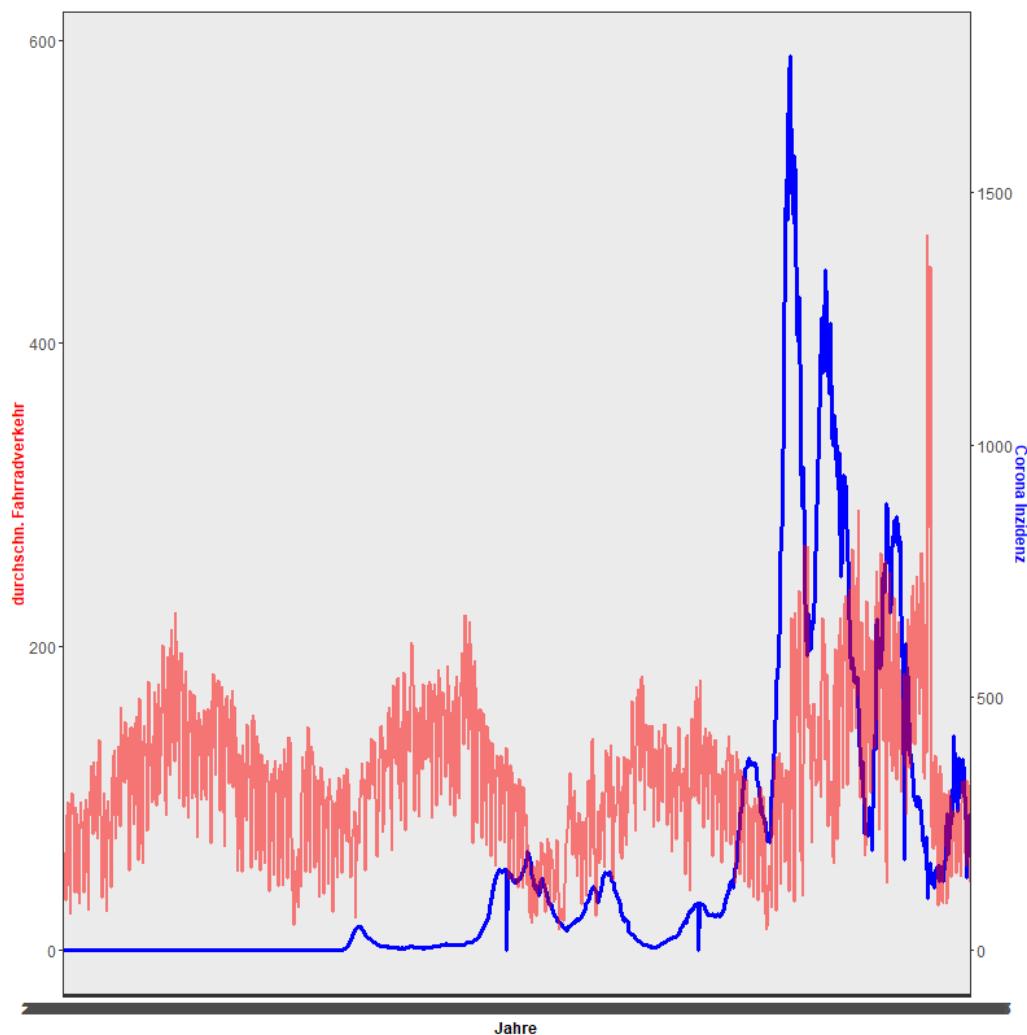


Fig. 3.8: Verlauf der Corona Inzidenz und des Radverkehrs

zu stammen vom Robert Koch Institut (Quelle: [https://www.rki.de/DE/Content/InfAZ/N/Neuartiges\\_Coronavirus/Daten/Inzidenz-Tabellen.html?nn=2386228](https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Daten/Inzidenz-Tabellen.html?nn=2386228)). Leider reichen die Daten des RKIs nicht für alle einzelnen Kommunen bis zum Beginn der Pandemie zurück. Im Zeitraum 10.3.2020 bis zum 6.5.2020 sind die im Datensatz enthaltenen Daten zur Corona Inzidenz bundesweit. Danach und bis zum 18.11.2020 verwendetet der Datensatz die Inzidenzen der jeweiligen Bundesländer. Erst danach sind die Inzidenzen der jeweiligen Landkreise und die der kreisfreien Städte verfügbar.

Bei der ersten Erstellung des Datensatz, waren die Corona Daten leider noch

nicht mit inbegriffen und erste Modelle wurden ohne diese Daten berechnet. Daraufhin wurde ein zweiter Datensatz erstellt, der nicht nur die Corona Daten zusätzlich berücksichtigte, sondern auch neuere Daten zu Straßen anlegte, dazu mehr im Detail im folgenden Abschnitt. Im weiteren Verlauf des Textes wird kenntlich gemacht, mit welcher Version des Datensatzes Berechnungen gemacht wird, ob mit der alten ohne Corona Daten oder der neuen mit Corona Daten.

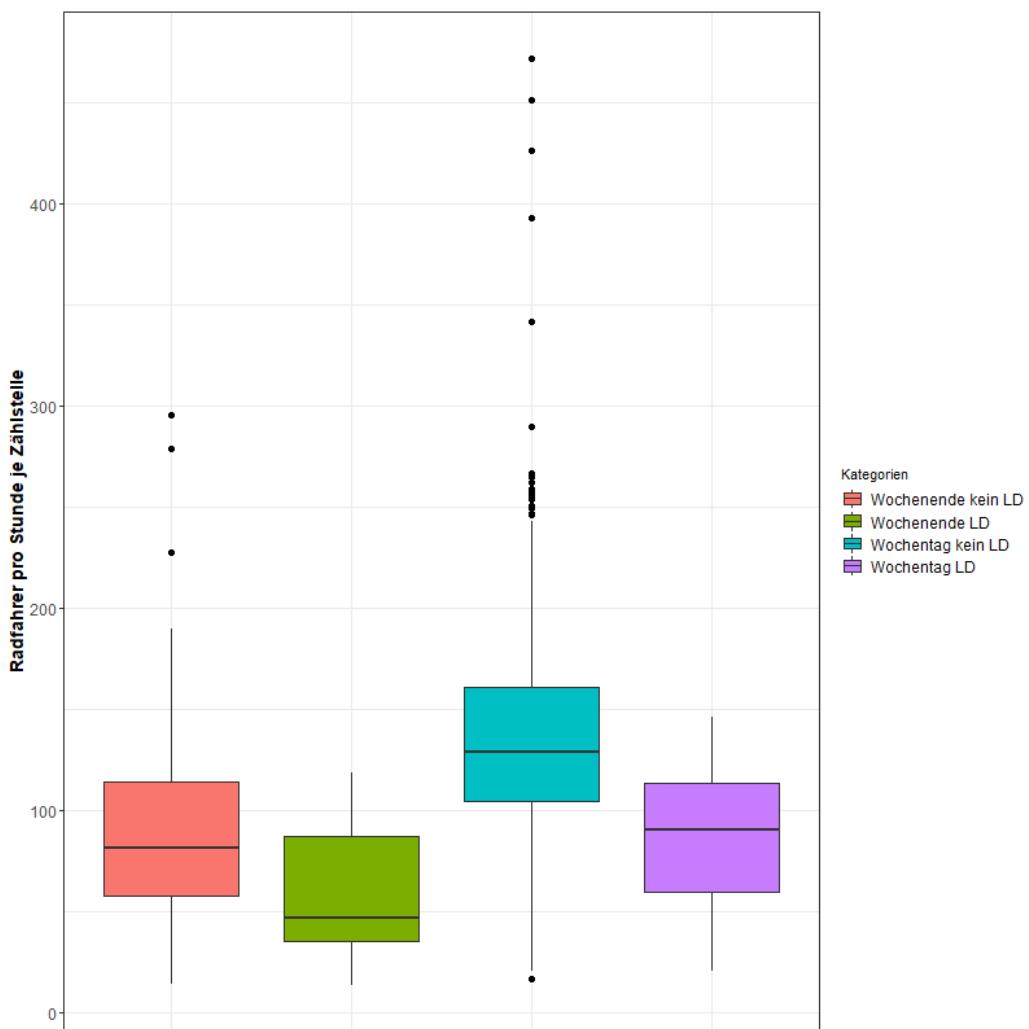


Fig. 3.9: Radverkehr im Lockdown

Anders als bei Möllers et al. (2021) gibt es in dem neuen Datensatz kein

zwiespältigen Effekt, wie die Abbildung 3.9 zeigt. Sowohl am Wochenende, als auch unter der Woche ist das Aufkommen des städtischen Radverkehrs gesunken.

### 3.5 Open Street Map Daten

Im vorherigen Abschnitt wurden die Datenquellen zu stadspezifischen Daten erläutert. Variablen zur demographie oder dem Autobesitz nach Städten sind wertvoll, um die Unterschiede im Radverkehr zwischen den Städten einzuschätzen zu können. Dies hilft jedoch nicht bei der räumlichen Unterscheidung. Verwendet man keine räumlichen Merkmale, so erhält man eine einheitliche Vorhersage je Stadt. Die räumliche Verteilung des Radverkehrs innerhalb jeder Stadt, ließe sich so nicht abbilden. Die einzige räumliche Variable, die bisher erhoben wurde, ist die Entfernung zum Stadtzentrum, beruhend auf den Daten des Gemeindeverzeichnis von 2012 bis 2021. Doch selbst mit dieser Variablen, gingen Abweichungen durch lokale Anlaufpunkte wie bei Bahnhöfen verloren.

Der daraus resultierende Schluss ist, Koordinaten zu solchen Anlaufpunkten oder auch Points of Interest (PoIs) dazu zu nutzen, Entferungen zu den Radzählstationen zu berechnen, und die Anzahl nach Radius und Entfernung zum nächstgelegenen PoI mit in das Modell aufzunehmen. Die beste Datenquelle hierfür bietet Open Street Map (OMS). Wie schon im Kapitel 2.2.4 beschrieben, handelt es sich beim Open Street Map um ein internationales Projekt, das in gemeinsamer Freiwilligenarbeit geographische Daten sammelt und öffentlich zur Verfügung stellt.

*Listing 3.1: OSM Daten Abfrage*

```

1 q <- getbb(toString(rawData$Town[1])) %>%
2 opq() %>%
3 add_osm_feature("amenity", "cinema")
4
5 cinema <- osmdata_sf(q)

```

Der Zugriff auf die OSM Daten funktioniert mit dem R-Paket „osmdata“ von Padgham et al. (2017). Dieses Paket bietet die Möglichkeit Daten verschiedener OSM Kategorien abzufragen über den Code 3.1. Dabei bezeichnet rawData\$Town[1] den ersten Eintrag in der Variable Town für den Grunddatensatz rawData. Da jede Stadt ihr eigenes Skript hat, ist dies jeweils die relevante Stadt. Für diese Stadt wird eine Anfrage gestellt, die bestimmte Features erfüllen muss. Im Code Beispiel werden Kinos gesucht. Mithilfe des R-Pakets „sf“ von Pebesma (2018) werden die Daten der Anfrage in ein simple Feature Format gespeichert in cinema. Dieses Format beinhaltet Koordinatenpunkten, mit jeweiliger ID und teils Namen und Adressen der verschiedenen Einrichtungen. Darüber hinaus aber auch Koordinaten der Linien, falls es sich z.B. um Straßennetze handelt, oder Koordinaten der jeweiligen Polygone der Gebäudegrundrisse.

### 3.5.1 Ausgewählte PoIs

Dieser öffentliche Zugang zu kategorischen Daten beinhaltet nicht nur Straßenverläufe, sondern auch Lageparameter zur öffentlichen Einrichtungen, Geschäften, Verkehrsknotenpunkten von Bus und Bahn, Ampeln und Straßenübergängen und vielem mehr. Es gibt also eine große Auswahl an Kategorien, die sich in das Modell aufnehmen ließen. Die Auswahl begrenzte sich dabei auf Kinos, Schulen, Universitätsgebäuden, Supermärkten, Kleidungsgeschäften und Fahrradwerkstätten. Die Auswahl dessen fand rein intuitiv statt. Schüler und Studenten sind häufiger Fahrradfahrer als der Rest der Bevölkerung. Kleidungsgeschäfte sind oft räumlich stark konzentriert im Stadtzentrum und bilden hierfür einen guten Indikator. Hat eine Stadt ein hohes Angebot an Fahrradwerkstätten, ist dies ein guter Hinweis darauf, dass hier ein hoher Radverkehr zu finden ist. Die Auswahl von Kinos und Supermärkten ist rein zufällig.

Noch empfehlenswert wäre es den Datensatz um noch mehr Kategorien zu erweitern und genau zu evaluieren, welche Kategorien die Performance des Modells erhöhen. Jedoch hätte dies den Arbeitsumfang dieser Masterar-

beit nochmals drastisch erhöht. In Abwägung des zeitlichen Aufwands und dem Nutzen für das Modell, beschränkt sich deswegen der Datensatz auf die erwähnten PoI's.

In jeder Kategorie wurde jeweils die Entfernung der jeweiligen Radzählstation zum jeweiligen PoI berechnet. Darüber hinaus wurden in jeder Kategorie die Anzahl an Einrichtungen in zwei verschiedenen Radien berechnet. Für Kinos und Fahrradwerkstätten wurde ein ein Kilometer und ein drei Kilometer Radius gewählt. Für Schulen, Universitätsgebäude und Kleidungsgeschäfte wurde ein 500 Meter und ein zwei Kilometer Radius gewählt. Für Supermärkte wurde ein 500 Meter und ein ein Kilometer Radius genutzt. Die Auswahl dieser Berechnungsgrenzen sind wiederum rein zufällig. Eine genaue Evaluation dieser Auswahl fand nicht statt. Zu Universitätgebäuden, Supermärkten und Kleidungsgeschäften finden sich die Abbildungen 7.3, 7.4 und 7.5 im Anhang. Für Städte, die über keine Universität verfügen, wurde angenommen, dass die nächste Universität mindestens 50 km weit entfernt sei.

*Listing 3.2:* Speichere die OSM Koordinaten

```

1 cinmat=matrix (1:3 * length (cinema$osm_polygons$osm_id) ,
2                 nrow = length (cinema$osm_polygons$osm_id) ,
3                 ncol = 3)
4
5 for (i in 1:length (cinema$osm_points$name)){
6     cinmat [i ,1]=cinema$osm_polygons$osm_id [ i ]
7     cinmat [i ,2]=as . data . frame (
8             cinema$osm_polygons $
9             geometry [[ i ]][1]) [1 ,1]
10    cinmat [i ,3]=as . data . frame (
11            cinema$osm_polygons $
12            geometry [[ i ]][1]) [1 ,2]
13 }
```

Der Code 3.2 zeigt die Umwandlung der Daten. Von Interesse sind nur die ersten Koordinatenpunkte der jeweiligen Polygone der Kinogebäude. Da-

zu wird die Matrix `cinmat` erstellt, die 3 Spalten hat und deren Reihenanzahl der Anzahl an ID's sind `length(cinma$osm_polygons$osm_id)` entspricht, damit in diesem Fall alle Kinos berücksichtigt werden. Mittels einem `for`-Loop werden alle Daten eingespeichert, so der Längengrad und Breitengrad in `cinma$osm_polygons$geometry[[i]][1]][1]`, der und die jeweilige ID in `cinma$osm_polygons$osm_id[i]`. Diese Matrix wird danach in ein Dataframe umgewandelt, um damit einfacher arbeiten zu können.

Nun muss nur noch für jede einzelne Zählstation die jeweilige Distanz zu den verschiedenen Kinos berechnet werden. Dazu bietet das R-Paket „geosphere“ von Hijmans (2021) die Funktion `distm`, mit deren Hilfe man mit zwei Koordination die Entfernung in Metern berechnen lassen kann. Pro Zählstation wird der `For`-Loop im Code 3.3 ausgeführt. Im code sind zwei Indexnummern zu finden, dabei steht `i` für die jeweilige Zählstation und `j` für das jeweilige Kino. Im Dataframe `d` sind die Koordinaten der jeweiligen Zählstationen gespeichert. Zusammen mit den Koordinaten der Kino im `cinmat` Datenframe kann die Funktion `distm` die Entfernung beider Koordinaten zueinander berechnen. Diese Entfernung wird dann im Vector `distc` gespeichert.

*Listing 3.3:* Berechnung der Entfernung

```

1 for (j in 1:length(cinmat$id)) {
2   cindist=distm(c(d$Lon[i],d$Lat[i]) ,
3             c(cinmat$lon[j],cinmat$lat[j]) ,
4             fun=distGeo)
5   distc[j]=cindist
6 }
```

Nun ist ein Vector vorhanden der alle Entfernungen gespeichert hat. Dieser kann nun dazu genutzt werden die benötigten Variablen zu berechnen. Zusammen mit dem Namen der aktuellen Station, der im Subset Datensatz `d[1,1]` zu finden ist wird die jeweilige Variable gespeichert. Dabei lässt sich mit der Funktion `min(distc)` die Entfernung zum nächstgelegenen Kino finden. Die Anzahl an Kinos in einem 1 km Radius findet sich mit der Funktion `sum(distc < 1000)`.

*Listing 3.4:* Berechnung der EntfernungsvARIABLEN

```

1 distmat_closest [ i ,1]=d [ 1 ,1]
2 distmat_closest [ i ,2]=min( distc )
3
4 distmat_1kmradius [ i ,1]=d [ 1 ,1]
5 distmat_1kmradius [ i ,2]=sum( distc < 1000)

```

Die so berechnet Variablen lassen sich einfach verbinden mit dem Befehl `merge`.

*Listing 3.5:* Füge neue Variablen dem Datensatz hinzu

```

1 rawData = merge(x = rawData ,y = distmat_closest ,
2 by = c("Station") ,
3 all = FALSE)
4
5 rawData = merge(x = rawData ,y = distmat_1kmradius ,
6 by = c("Station") ,
7 all = FALSE)

```

### 3.5.2 Ausgestaltung des öffentlichen Verkehrs

Neben PoI's, die als Anlaufstellen des Radverkehrs dienen, ist der öffentliche Nahverkehr ein weiterer Faktor, der den Radverkehr beeinflussen sollte mit verschiedenen Effekten. Zum einen ist ein substitueller Effekt, dass je besser der öffentliche Nahverkehr ausgebaut wäre, desto mehr Fahrradfahrer weichen auf den Nahverkehr aus. Eine weitere interessante Variable hierfür wären auch Preistarife des Nahverkehrs. Darüber hinaus gibt es auch komplementäre Effekte vor allem zwischen Bahnhöfen und Fahrrädern.

Im Datensatz vertreten sind Busstationen, Straßenbahnstationen, U-Bahn Stationen und Bahnhöfe. Darüber hinaus sind Daten zu Ampeln und Straßenübergänge ohne Ampeln im Datensatz vorhanden. Zu allen Daten ist wieder die nächstgelegende Entfernung angegeben, sowie die jeweilige Anzahl in zwei verschiedenen Radien. Für Ampeln, Straßenübergängen ohne Ampeln,

Busstationen, Straßenbahnstationen und U-Bahn Stationen betragen die Räden jeweils 250 Meter und 1 km. Für Bahnhöfe betragen die Räden 1 und 3 km. Bei den Bahnhöfen wurden nur diejenigen ausgewählt, deren Betreiber die DB Netz AG ist. Im Anhang finden sich zu Busstationen, Ampeln, Straßenbahnstationen und Bahnhöfen die Abbildungen 7.6, 7.7, 7.8 und 7.9. Für Städte, die nicht über Straßenbahnen oder U-Bahnen verfügen wurde angenommen, dass die nächsten Stationen jeweils mindestens 50 km weit entfernt seien.

Im Code hier besteht kein großer Unterschied, außer dass nun einzelne OSM Punkte relevant sind und keine Polygone z.B. für die Ampeln.

### 3.5.3 Straßentypen

Neben Informationen zu Gebäuden, Einrichtungen und einzelnen Punkten wie Ampeln, bietet Open Street Map auch Daten zu Straßennetzen. Der Datensatz, der zur Berechnung der Vorhersagen genutzt werden soll, sollte Informationen zum jeweiligen Straßennetz berücksichtigen. Primär gehört dazu die Information zum Straßentyp, wobei es zahlreiche Kategorien gibt. Eine begrenzte Auswahl an Straßentypen wird im ersten Datensatz berücksichtigt. Dazu zählen Radwege, Pfade, Wohngebietstraßen, Straßen in einem verkehrsberuhigten Bereich oder auch Spielstraßen, sekundäre und primäre Hauptstraßen. Die Aufteilung von Straßentypen nach Stationen im Datensatz sieht man auch in der Abbildung 3.10 links, wobei festzustellen ist, dass einige Stationen doppelt zugeteilt worden sind, durch sich kreuzende Straßen und Brücken. Auch Brücken wurden im Datensatz berücksichtigt und getestet, wie weit die nächstgelegene Brücke zur Zählstation entfernt ist. Eine Darstellung des Zusammenhangs zwischen dem Radverkehr und Brücken findet sich im Anhang in der Abbildung 7.10.

Ob eine Zählstation zu einem Straßentyp gehört oder nicht, wurde getestet, indem berechnet wurde, inwieweit eine Zählstation von einer Straße des betroffenen Straßentyp entfernt ist. Ist diese Entfernung geringer als 5 Meter, wurde angenommen, dass die Zählstation sich auf oder an einer solchen

Straße befände.

*Listing 3.6:* Teste den Straßentyp

```

1 for (i in 1:nlevels(as.factor(rawData$Station))){  

2     dist_mat$cycleways[i] =  

3         min(st_distance(DT2$geometry,  

4                           DT3cycleways))  

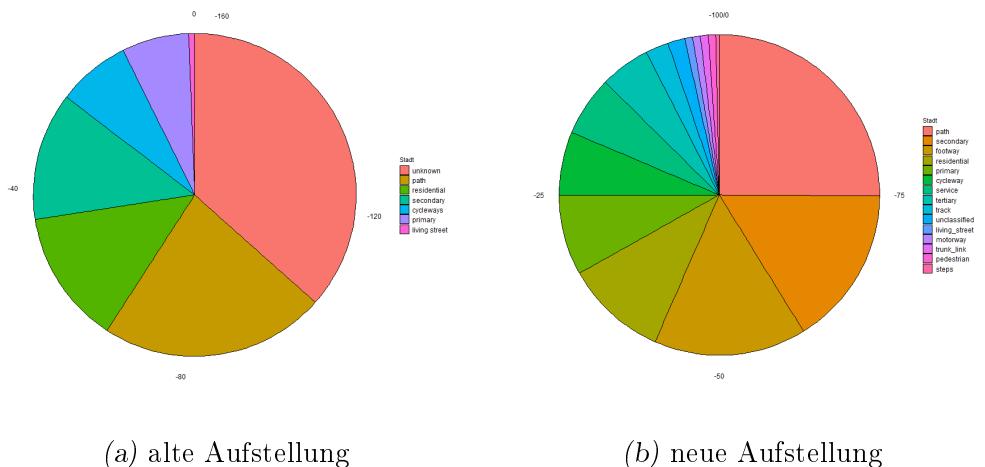
5  

6     if (dist_mat$cycleways[i]<5){  

7         bool_mat$cycleways[i]=1  

8     }  

9 }
```



*Fig. 3.10:* Verteilung von Straßentypen im Datensatz nach der alten Ermittlung

Dazu muss die jeweils kürzeste Entfernung zwischen einem Punkt und einer Linie berechnet werden. Dies funktioniert mithilfe der `st_distance` Funktion. Diese Funktion benutzt die Liste aller Koordinaten aller Straßenabschnitte des jeweiligen Straßentyps `DT3cycleways` und der Koordinate der Zählstation `DT2$geometry`, um daraus die Entfernung eines jeden Straßenabschnitts zum Punkt der Zählstation zu berechnen. Davon ist allein das Minimum interessant. Ist das Minimum geringer als 5 Meter, ist klar, dass

die Zählstation nahe des jeweiligen Straßentyps liegt.

Im zweiten Datensatz, der erstmals auch Corona Daten inkludierte, wurde dieses Problem anders gelöst, dargestellt in der Abbildung 3.11. In einem For Loop wurden alle Zählstationen betrachtet. Dabei wurden alle Straßen, die ein kleines Viereck (in der Abbildung 3.11 rot) um die Zählstation (blauer Punkt) herum berührten betrachtet. Die Zählstation wurde jeweils der nächstliegenden Straße (blaue Linie) zu geordnet. Das in der Abbildung gewählte Beispiel stammte von einer Zählstation in Hamburg. Auf diese Weise können aber nicht nur Daten zum Straßentyp festgestellt werden, sondern auch zur Straßenlänge, zum Straßenbelag, zur geltenden Höchstgeschwindigkeit und die Anzahl der Straßenspuren.

Genauer beschrieben wird auch wie in 3.7 zu sehen in jedem Loop Durchgang der Datensatz gefiltert nach Zählstationen und die Beobachtungen einer Station landen im Dataframe d. Für jede Station muss das rote Rechteck wie in der Abbildung 3.11 berechnet werden. Dies geschieht wie in 3.7 mittels der Variable radius und den Koordinaten aus d, also den Koordinaten der jeweiligen Zählstation. Die Koordinaten des Vierecks sind so in myLocation gespeichert.

*Listing 3.7:* Berechnung der Straßenvariablen

```

1 d=BikeData[BikeData$Station %in%
2   toString(levels(as.factor(
3     BikeData$Station ))[i]) ,]
4
5 radius = 0.0012
6 myLocation <- c(d$Lon[1]-radius ,d$Lat[1]-radius / 1.8 ,
7   d$Lon[1]+radius ,d$Lat[1]+radius / 1.8)

```

Mit diesen Koordinaten kann nun eine Abfrage von Open Street Map Daten zu der Klasse "highway" gestellt werden. Wie eine solche Abfrage gestellt wird, ist auch im Code 3.1 zu sehen. Darauf hin muss die Zählstation der nächstgelegnsten Straße zugeordnet werden. Dies passiert im Code 3.8.

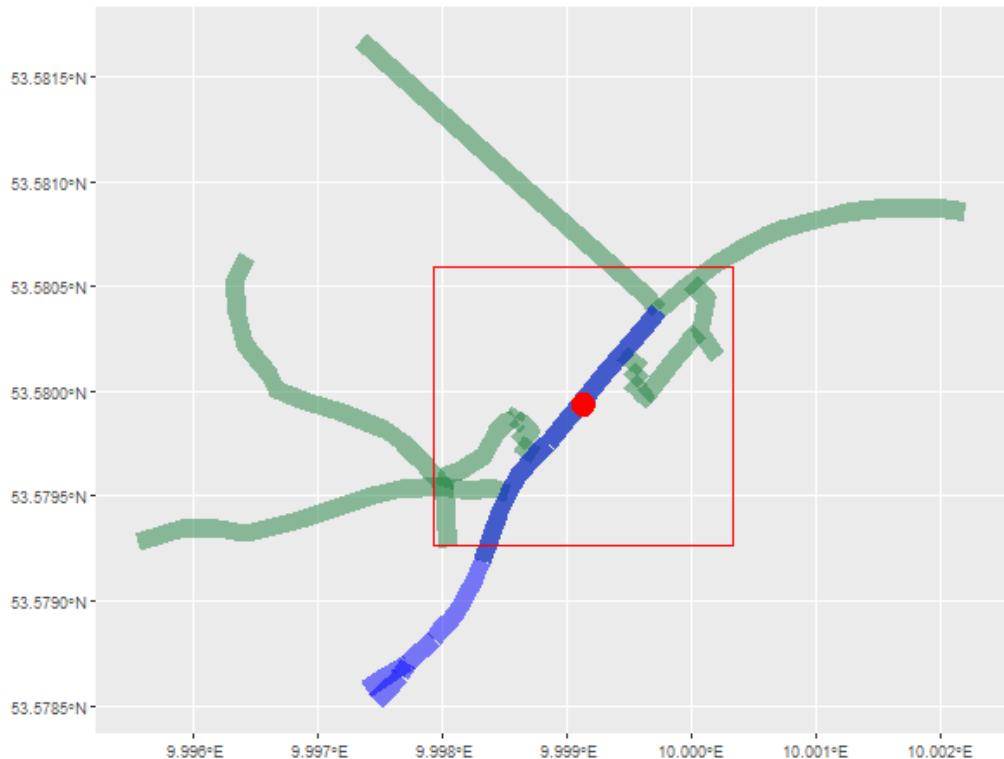


Fig. 3.11: Beispiel der Straßentypanalyse

Dieser besteht aus zwei kürzeren For Schleifen. Die erste wandelt alle Koordinaten der Straßen in ein passendes GPS Format um und ermittelt die jeweils kürzeste Distanz der Straße zu den Koordinaten der Zählstation gespeichert in `count_point$geometry`. Für den Fall, dass mehrere Straßen auf der selben Position liegen, ermittelt die Funktion `which(dist==min(dist))` welche dieser Straßen dies sind und wählt von diesen dann diejenige aus, bei der ein Straßennamen angegeben ist.

*Listing 3.8: Zuordnung zur nächsten Straße*

```

1 j=1
2 for(j in 1:nrow(streets$osm_lines)){
3   street_Points = st_transform(
4     streets$osm_lines$geometry[j], 4269)
5   dist[j] = min(st_distance(
6     count_point$geometry, street_Points))
7 }
8
9 for(j in which(dist==min(dist))){
10   if(length(streets$osm_lines$name[j])>0){
11     if(!is.na(streets$osm_lines$name[j])){
12       nearest=j
13     }
14   }
15 }
```

Die Variable `nearest` gibt nun an, welche Straße die nächstgelegene zur Zählstation ist und mit diesen Informationen lassen sich die weiteren notwendigen Variablen berechnen, wie auch in Code 3.9 beschrieben wird. Wie sich die Straßentypen unter der Anzahl der Zählstationen verteilen, zeigt die Abbildung 3.10 links. Weiter zeigt die Abbildung 3.12 die Aufteilung des Radverkehrsaufkommens nach der jeweiligen Straßenbeschaffenheit nach Straßenbelag und Straßentyp, wobei die Beschreibungen dem englischen Original der Open Street Map Beschreibung entspricht, denen teils spezifische Definitionen zu Grunde liegen.

*Listing 3.9:* Berechnung der Straßenvariablen

```

1 street_type = streets$osm_lines$highway[nearest]
2 surface = streets$osm_lines$surface[nearest]
3 lanes = streets$osm_lines$lanes[nearest]
4 maxspeed = streets$osm_lines$maxspeed[nearest]
5 streetlengths = st_length(street_Points[nearest])
```

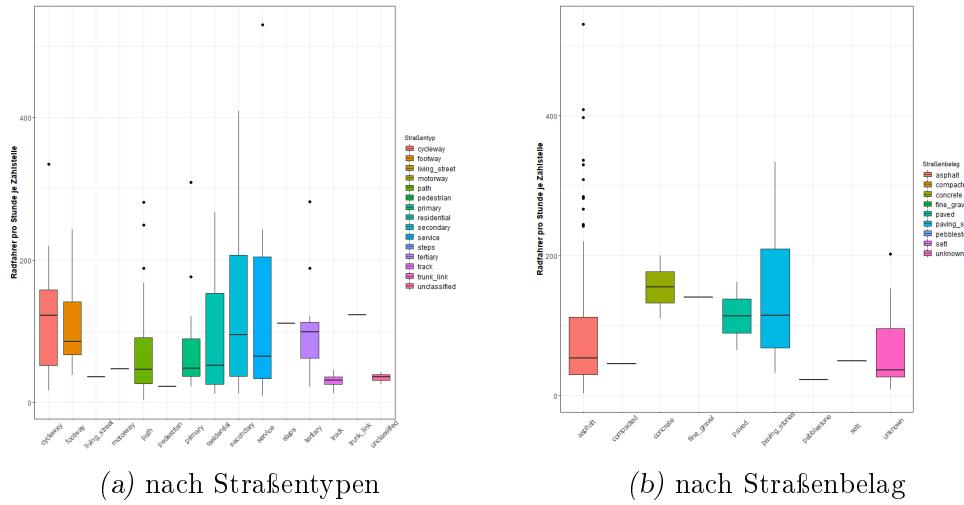


Fig. 3.12: Radverkehr nach Straßenbeschaffenheit

### 3.5.4 Sonstige

Weitere wichtige Variablen sind Feier und Ferientage. Dazu wurde eine Liste dieser Tage händisch für alle Bundesländer angelegt und diese automatisch in den Datensatz übertragen. Informationen zu den Ferien- und Feiertagen seit 2012 nach den jeweiligen Bundesländern stammen von der Seite [www.kalenderpedia.de](http://www.kalenderpedia.de).

## 4. VERWENDETE METHODEN

Das vorangegangene Kapitel gab einen genauen Überblick über die bestehende Literatur, die sich mit der Vorhersage des Aufkommens von Fahrrädern in urbanen Zentren beschäftigt. Immer häufiger wurden dabei Machine Learning Methoden zur Schätzung eines Modells verwendet. Um die Übersichtlichkeit der Arbeit zu gewährleisten, wurden Erläuterungen, über die verschiedenen verwendeten Methoden, bis zu diesem Zeitpunkt aufgespart, um einen gegliederten Überblick zu ermöglichen.

Im Folgendem wird die Funktionsweise von Regressionssystemen, Support Vector Systemen, Entscheidungsbaum Varianten und neuronalen Netzwerken erklärt. Die Auswahl dieser Schwerpunkte beruht auf der Verwendung in der bisher dargestellten Literatur. Hat man ein geeignetes Modell ausgewählt, lässt sich dessen Effizienz mit dem richtigen Validierungsverfahren noch weiter steigern. Deswegen beinhaltet dieses Kapitell ebenfalls eine kurze Erläuterung der Cross Validation.

### 4.1 *Problem zur Autokorrelation*

Vorab muss das Problem der Autokorrelation erwähnt werden. Autokorrelation bezeichnet wie der Name fast sagt, die Korrelation innerhalb einer Variable und nicht wie sonst die Korrelation zweier verschiedener Variablen. Diese Autokorrelation kann räumlich und zeitlich auftreten. Bei der zeitlichen Autokorrelation ähneln sich Werte, die zeitlich nah bei einander liegen. Bei der räumlichen Autokorrelation ähneln sich Werte, die räumlich nah bei einander liegen.

Dies kann zu Problemen führen. So beschreiben Liu et al. (2022), dass räumliche Autokorrelation die Annahme von unabhängig und identisch verteilt

te Zufallsvariablen verletzt. Dies kann zu Overfitting oder einem Bias der Vorhersagen führen. Im Fall von OLS Regressionen ist der Standardfehler nicht mehr konsistent, was Aussagen über die Signifikanz des Modells wertlos macht. So beschreibt es auch Stock and Watson Stock and Watson (2015a). Da das Ziel dieser Masterarbeit jedoch nur die Vorhersage ist, ist die Signifikanz nicht entscheidend. Um gute Vorhersagen zu treffen, ist es jedoch notwendig, das Risiko von Overfitting oder einer Verzerrung zu minimieren. Dazu gibt es drei Lösungsansätze. Diese wären die Verwendung von Lagged Variablen, Feature Engineering und Resampling.

#### 4.1.1 Lagged Variablen

Der Begriff Lagged Variable bezeichnet die Aufnahme einer versetzten Beobachtung, die als zusätzliche Variable mit in das Modell aufgenommen wird. Im Bereich von Zeitreihendaten wird also eine zeitlich vorhergehende Beobachtung der eigentlich abhängigen Variable in das Modell aufgenommen. Im Fall der Regression nennt man ein solches Vorgehen Autoregression. Dies beschreiben z.B. Stock and Watson (2015b). Die mathematische Formulierung einer allgemeinen Autoregression lautet:

$$y_t = \beta_0 + \beta_1 * y_{t-1} + \beta_2 * y_{t-2} + \dots + \beta_p * y_{t-p} + u_t \quad (4.1)$$

Diese Formulierung kann natürlich nicht nur in OLS Regressionen angewendet werden, sondern auch bei allen weiteren Machine Learning Algorithmen, über die hier noch beschrieben werden. Eine weiter Entwicklung dieses Ansatz ist die ARIMA-Modellierung, was die Abkürzung für Auto Regressive-Moving Average ist.

Problem an diesem Ansatz ist, dass er für die Vorhersage an unbekannten out of sample Orten nicht anwendbar ist, weil an unbekannten Orten auch vorhergehende Beobachtungen unbekannt sind, die man in das Modell zur Vorhersage mit aufnehmen müsste. Deswegen ist die Aufnahme von Lagged Variablen nicht geeignet für diese Masterarbeit.

### 4.1.2 Erklärende Variablen

Eine weitere Möglichkeit Autokorrelation zu vermeiden ist es, Variablen in das Modell aufzunehmen, die die Autokorrelation erklären können. Nimmt man in einem zeitlichen Modell z.B. die Stunden als Variable auf, die erklären können, dass in der Mittagszeit viele Fahrradfahrer auf den Straßen sind und Nachts nur sehr wenige, dann erklärt das auch, warum zu zwei Zeitpunkten, die nah beieinander z.B. um die Mittagszeit herum, ähnlich hohe Werte aufzeigen. Dies funktioniert ähnlich auch für räumliche Autokorrelation. Nimmt man als Variable die Entfernung zum Stadtzentrum auf, dann erklärt dies nicht nur, dass nahe des Stadtzentrums viele Radfahrer unterwegs sind und dafür weniger Radfahrer außerhalb des Stadtzentrums, es erklärt auch, wieso sich zwei nahe beieinander liegende Punkte ähneln, weil beide weit weg vom Stadtzentrum liegen.

Diesen Ansatz verfolgt auch die Studie von Liu et al. (2022), automatisiert jedoch diesen Prozess. Sie schlagen für Random Forest z.B. vor, Probleme mit Autokorrelation zu vermeiden, in dem gewichtete räumliche Variablen durch das LASSO Regressionsverfahren selektiert werden. Diese räumliche Variablen übernehmen die vorher besprochene Rolle. Die LASSO Regression ist neben der RIDGE Regression eines von zwei Regularisierungsverfahren. Dabei wird eine Regression aufgesetzt, in deren Optimierungsproblem nicht nur die quadrierte Summe der Fehlerterme steht, sondern dazu addiert die quadrierte (RIDGE) oder absolute (LASSO) Summe der Estimaten selbst. Ziel einer solchen Optimierung ist es, die Estimaten zu schrumpfen, wobei bei LASSO sogar Estimaten auf null setzen kann und somit Feature Selection betreiben kann.

Ein solches Modell aufzusetzen, übersteigt jedoch den Arbeitsumfang dieser Arbeit. Was die zeitliche Autokorrelation angeht, können die Variablen zur Stunde und zum jeweiligen Monat Auschluss geben. Was die räumliche Autokorrelation angeht, stehen so viele Variablen zur Auswahl, dass es hier durchaus Sinn machen könnte, eine genauere Feature Selektion zu betreiben. Ein solches Verfahren jedoch erfordert leistungsfähigere Hardware, würde aber gleichzeitig die Performance des Modells nicht drastisch erhöhen.

### 4.1.3 Resampling

Resampling bezeichnet eine Neuanordnung von Beobachtung eines Datensatzes, sodass dabei ein neuer Datensatz entsteht. Beispiele hierfür ist das Bootstrapping, bei dem eine Mehrfachziehung zufällig ausgewählter Beobachtungen erlaubt ist. Daneben ist Resampling auch ohne Mehrfachziehung von Beobachtungen nötig. Diese führt dazu, dass man weniger Beobachtungsdaten zieht, als im Datensatz vorhanden sind. Dabei handelt es sich um Undersampling.

Undersampling kann auch dabei helfen, Autokorrelation zu vermeiden, denn verringert man die Dichte an Beobachtungen, spielt die zeitliche oder räumliche Nähe einzelner Beobachtungen zu einander einer geringere Rolle. Der große Nachteil dieser Methode, ist dass durch ein Verzicht dieser Daten auch Informationen verloren gehen und das Modell durch die kleinere Sample Size eine schlechtere Performance vorweist. Im Fall dieser Arbeit steht ein Datensatz zur Verfügung, der mehr als groß genug ist, um Undersampling zuzulassen. Zudem kommt hinzu, dass der technische Flaschenhals im Arbeitsspeicher es notwendig machen wird, die Sample Size des Datensatzes zu verkleinern.

## 4.2 OLS Regression

Einer der einfachsten und geläufigsten Methoden ist die Regressionsanalyse. So nutzen Holmgren et al. (2017), Alattar et al. (2021) und Gao and Chen (2022) z.B. ein lineares Regressionssystem und Wessel (2020) ein log-lineares sowie ein negativ binomiales Regressionssystem.

Das Prinzip einer einfachen OLS (Ordinary Least Square) Regression besteht darin, den Zusammenhang zwischen zwei Variablen zu finden, der die Summe der quadrierten Fehlerterme, also die Abweichung tatsächlicher Beobachtungen zur Regressionsgerade, minimiert. Als Maß zur Validierung eines solchen Modells ließe sich z.B. das Bestimmtheitsmaß  $R^2$  nutzen, also der Anteil der Streuung, der durch die Regression erklärt werden kann, aber auch z.B. die Wurzel der summierten quadratischen Fehler (RMSE). Ein solches Fehler-

maß oder Bestimmtheitsmaß, dass die Performance des Modells bewertet ist nützlich, einen Vergleich zu ziehen, verwendet man mehrere Modelle, wie es zB Holmgren et al. (2017), Broucke et al. (2019) oder Gao and Chen (2022) machen.

### 4.3 Support Vector Regression

In der bisher betrachteten Literatur haben Holmgren et al. (2017), Broucke et al. (2019), Xu et al. (2013) und Gao and Chen (2022) Support Vector Machines bzw Support Vector Regressionen verwendet. Pisner and Schnyer (2020) schildern Support Vector Machines als eine supervised Classification Methode, die eine Hyperebene nutzt, um Daten nach unterschiedlichen Ausprägungsmustern zu trennen und so den jeweiligen Klassen zu zuordnen. Diese Vorgehensweise ist auch dargestellt in Abbildung 4.1. Nun könnte diese Trennlinie, auch Hyperplane genannt, so gewählt werden, dass sie das Margin, also den Abstand zwischen der Trennlinie selbst und der nächstgelegenen Beobachtung, maximiert. Dieses Vorgehen wird auch als Maximum Margin Classification bezeichnet. Jedoch falls innerhalb des maximalen Margin-Bereichs eine neue zufällige Beobachtung hinzu kommt, dann führt diese zu einer starken Veränderung der Trennlinie selbst. Ausreißer Daten können so zu einer starken Verzerrung führen und eine Generalisierbarkeit wäre nicht gegeben.

Deswegen nutzt man oft soft Margins, die die Missklassifikation von Ausreißern zu lassen. Dabei stellt  $\xi$  die Variable für die Toleranz von Missklassifikationen dar, auch oft Slack Variable genannt. Ist  $\xi = 0$  so erhalten wir ein hard Margin Classifier.

Mathematisch formulieren James et al. (2013a) eine Hyperplane in einem  $p$ -dimensionalen Raum wie folgt:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p = 0 \quad (4.2)$$

Dabei liegt der Normalenvektor  $\beta = (\beta_1, \beta_2, \dots, \beta_p)$  in der orthogonalen Richtung zur Hyperplane. Das Optimierungsproblem hinter der soft Margin ( $M$ )

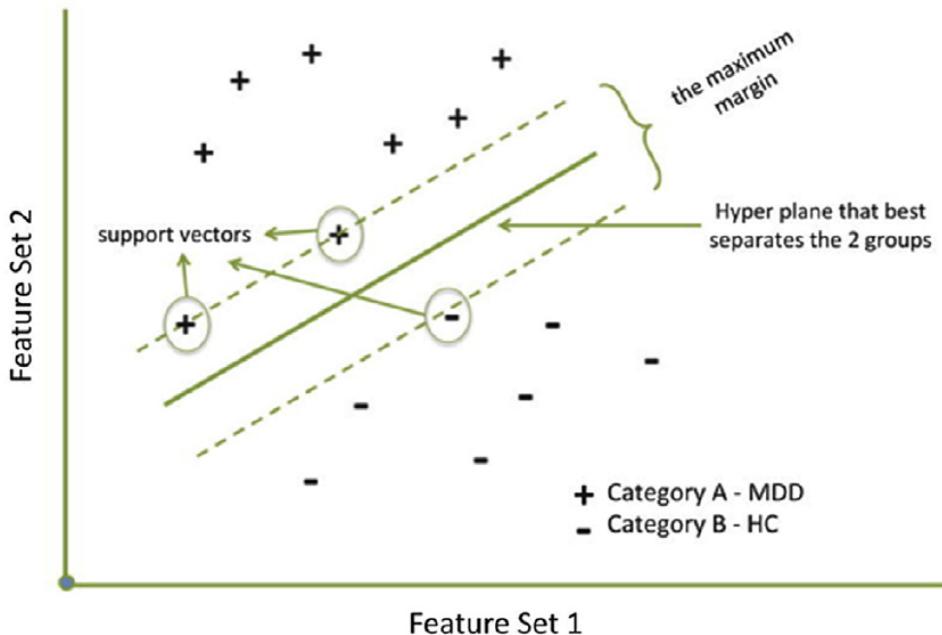


Fig. 4.1: Hyperebenen (Hyperplanes) in Support Vector Machines

Quelle: Pisner and Schnyer (2020)

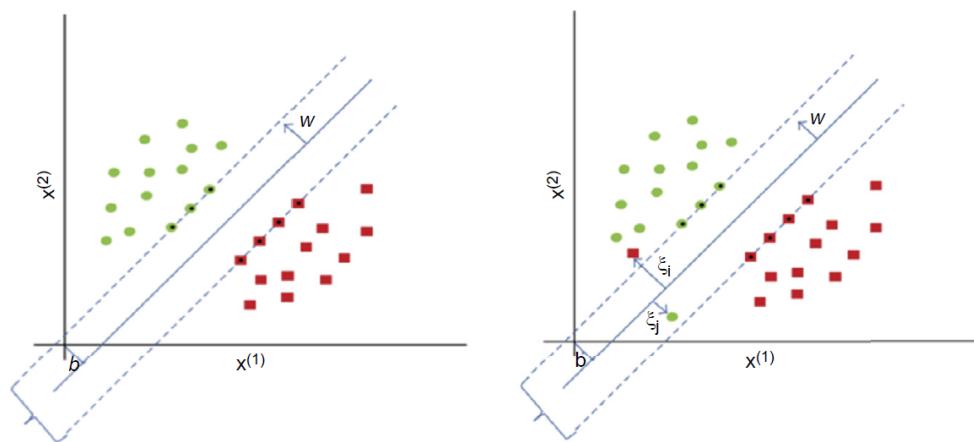


Fig. 4.2: Soft Margins in Support Vector Machines

Quelle: Pisner and Schnyer (2020)

Klassifikation sieht wie folgt aus:

$$\begin{aligned}
 & \max_{\beta_0, \beta_1, \dots, \beta_p, M, \xi_1, \dots, \xi_n} M \\
 & \text{und} \sum_{j=0}^p \beta_j^2 = 1, \\
 & y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \geq M(1 - \xi_i), \\
 & \xi_i \geq 0, \sum_{i=0}^n \xi_i \leq D,
 \end{aligned} \tag{4.3}$$

Hier taucht nun die Slack Variable auf, die einen Toleranzbereich für Missklassifikation ermöglicht. Setzt man  $\xi = 0$ , erhält man einen hard Margin Classifier. D ist ein nichtlinearer Tuning Parameter. Natürlich kann man in der Funktion des Hyperplanes (4.2) Polynomiale aufnehmen und so einen nicht linearen Classifier erhalten.

Grundlegendes Problem ist nun, dass in der Vorhersage des innerstädtischen Fahrradaufkommens kein Klassifikationsproblem besteht, so wie es eine Support Vector Machine lösen würde. Deswegen bietet es sich an, eine Support Vector Regression zu nutzen, so wie es z.B. Holmgren et al. (2017) macht. Während herkömmliche OLS Regression die summierten quadratischen Fehlerterme minimiert, minimiert Support Vector Regression, so beschreiben es Awad and Khanna (2015), eine Loss Function, die Fehlvorhersagen bestraft. Dieser Vorgang ist auch eine Verallgemeinerung des bisher beschriebenen Klassifikations Algorithmus.

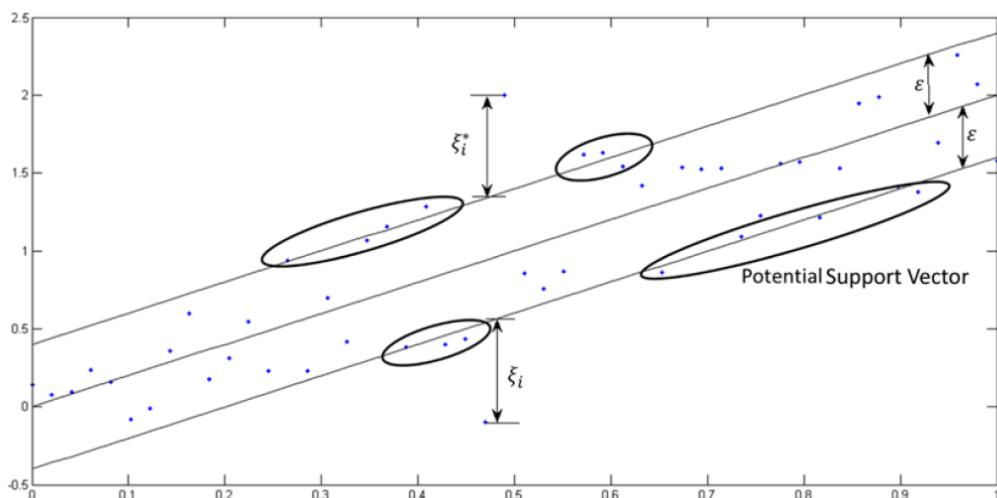


Fig. 4.3: Eindimensionale lineare SVR

**Quelle:** Awad and Khanna (2015)

Wo im Soft Margin Classifier  $\xi$  genutzt worden ist, um auch Missklassifikationen zuzulassen, wird nun ein unempfindlicher Bereich von  $\varepsilon$  um die Hyperplane herum gelegt, sodass ein Großteil der Beobachtungen im eigentlichen Margin Bereich zu finden sind. Aus der Hyperplane, die eigentlich

Beobachtungen trennen soll wird so die regressive  $\varepsilon$ -Tube, eine wertbeständige Funktion. Dargestellt ist dies auch in der Abbildung 4.3.

Die Slack Variable  $\xi$  wird so zur Toleranz Variable für Ausreißer von der  $\varepsilon$ -Tube und so zum Bestandteil der Loss Function, die man versucht zu minimieren. Zusammen mit C der Gewichtung der Minimierung ergibt das:

$$\begin{aligned} \min_{\beta_0, \beta_1, \dots, \beta_p, M, \xi_1, \dots, \xi_n} & \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N \xi_i + \xi_i^* \\ \text{und } & y_i - \beta^T x_i \leq \varepsilon + \xi_i^*, \quad i = 1 \dots N \\ & \beta^T x_i - y_i \leq \varepsilon + \xi_i, \quad i = 1 \dots N \\ & \xi_i, \xi_i^* \geq 0, \quad i = 1 \dots N \end{aligned} \tag{4.4}$$

Der Vorteil dieser Vorgehensweise ist, dass man durch den Optimierungsprozess ein Modell erhält, das sehr robust auf Ausreißer reagiert trotz hoher Präzession durch die Nutzung von  $\xi$ .

#### 4.4 Random Forests Regression

Entscheidungsbaum Methoden finden sich bei Holmgren et al. (2017), Broucke et al. (2019), Mitchell (2018) und Gao and Chen (2022). Dabei sind diese simpel, nicht immer aber besonders präzise so James et al. (2013b). Es sei denn man nutzt komplexere Weiterentwicklungen wie Random Forests. Diese nutzen auch Holmgren et al. (2017), Broucke et al. (2019) und Mitchell (2018).

Ein Vorteil für diese Hausarbeit ist weiter, dass Entscheidungsbaum Methoden sowohl zur Klassifikation, als auch zur Regression genutzt werden können so James et al. (2013b). Die Begrifflichkeit röhrt von der Struktur in der Entscheidungsbäume Daten aufbereiten, denn ihre Darstellungsweise erinnert an eine umgedrehte Baumkrone, in der jede Astgabelungen ein Statement darstellt, welches mit wahr oder falsch beantwortet werden kann. Die Beobachtungen aus dem Datensatz folgen entlang ihrer Ausprägung verschiedenen Astgabelungen und werden so einer Vorhersage zugeordnet. Dies kann z.B. aussehen, wie in Abbildung 4.4.

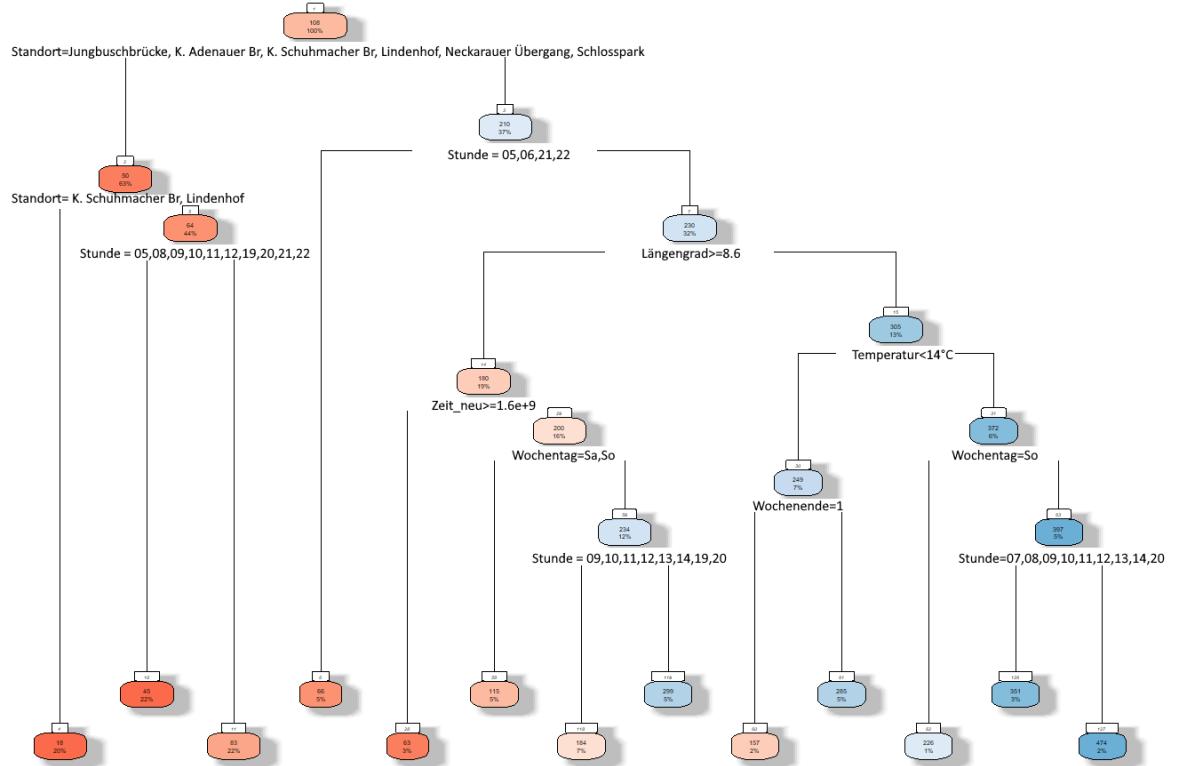


Fig. 4.4: Ein Entscheidungsbaum basierend auf Daten von Fahrradzählstationen in Mannheim und Daten des DWD 2016 bis 2022.

**Quelle:** Eigene Darstellung

Durch diese Entscheidungsgrenzen unterteilt der Entscheidungsbaum den Prädiktorenraum in distinkte sich nicht überlappende Regionen  $R_1, R_2, \dots, R_J$ . Das Optimierungsproblem besteht nun darin,  $R_1, R_2, \dots, R_J$  so zu wählen, dass die Summe der quadrierten Residuen (RSS) minimiert wird:

$$\sum_{j=1}^J \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 \quad (4.5)$$

Auf diese Weise lässt sich ein Entscheidungsbaum erstellen. Bei den zuvor angesprochenen Random Forests wird diese Vorgehensweise mit einem Bootstraps Verfahren kombiniert. Bootstrapping bedeutet, dass durch zufällige Ziehungen von Beobachtungen aus einem bestehenden Datensatz, wobei mehr-

fach Ziehungen erlaubt sind, ein neuer Bootstraps Datensatz erstellt wird, der dem ursprünglichen Datensatz ähnelt, jedoch davon abweicht. Diese Praxis führt man mehrere male durch und wendet auf jeden Bootstraps Datensatz einen Entscheidungsbaum an, so das man eine große Anzahl von Entscheidungsbäumen erhält. Möchte man nun auf neuen Beobachtungen eine Vorhersage treffen, lässt man diese durch die Bootstraps Entscheidungsbäume laufen, wobei man sich auf die Vorhersage festlegt, die am meisten von den Bootstraps Entscheidungsbäumen bestimmt wurde. Dieses Verfahren führt zu einer deutlichen Steigerung der Vorhersage Genauigkeit.

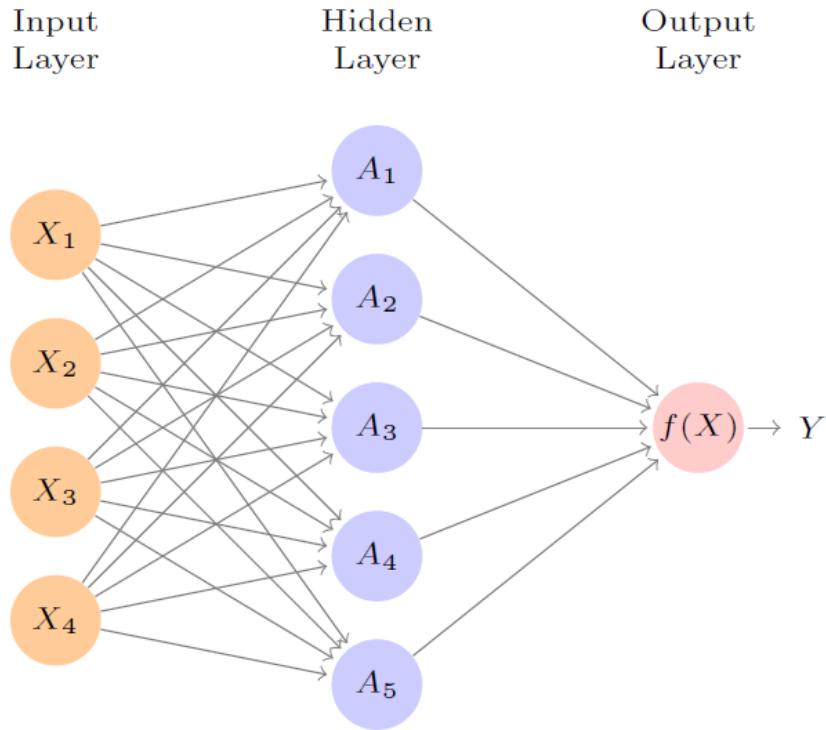
## 4.5 Neuronale Netze

Deep Learning Algorithmen und die dazu gehörigen neuronalen Netzwerke sind mit die fortgeschrittenste Form im Bereich des Machine Learnings. Sie tauchen bei Broucke et al. (2019) und Mitchell (2018) auf.

### 4.5.1 Aufbau eines neuronalen Netzes

Zur Verdeutlichung der Funktionsweise von neuronalen Netzwerken beschränkt sich diese Arbeit aufgrund von Übersichtlichkeit auf Single Layer Perceptrons, diese sind die einfachste Form eines neuronalen Netzes und bestehen nur aus einer Schicht von Neuronen. Eine graphische Darstellung eines solchen Netzwerkes ist in Abbildung 4.5 zu sehen.

Mathematisch ausgedrückt, gleicht dieses Netzwerk einer Vektor Multiplikation. Im Input Vektor (Input Layer), sind die Variablen einer jeden einzelnen Beobachtungen zu finden. Diese werden mit Gewichten im Hidden Layer multipliziert. Wobei jeder einzelner Punkt in diesem Layer, auch Neuron genannt, über ein separates Gewicht verfügt. Diese sind so eingestellt, dass im Output eine möglichst zielgenaue Vorhersage zustande kommt. Als Formel



*Fig. 4.5:* Neural network with a single hidden layer

**Quelle:** James et al. (2021)

sieht diese Vorgehensweise wie folgt aus:

$$\begin{aligned}
 f(x) &= \beta_0 + \sum_{k=1}^K \beta_k h_k(X) \\
 &= \beta_0 + \sum_{k=1}^K \beta_k g\left(w_{k0} + \underbrace{\sum_{j=1}^p w_{kj} X_j}_{\text{Aktivierungsfunktion}}\right)
 \end{aligned} \tag{4.6}$$

Ob und in welchem Umfang die Information, die ein Neuron passiert, weitergereicht wird, entscheidet die Aktivierungsfunktion  $g(z)$ . In ihr werden die Informationen des Inputs und die der Gewichte übertragen. Das kann z.B. durch eine Sigmoid Funktion (4.7) oder durch eine ReLU (Rectified Linear

Unit) Funktion (4.8) geschehen.

$$g(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}} \quad (4.7)$$

$$g(z) = \begin{cases} 0 & \text{wenn } z < 0 \\ z & \text{sonst.} \end{cases} \quad (4.8)$$

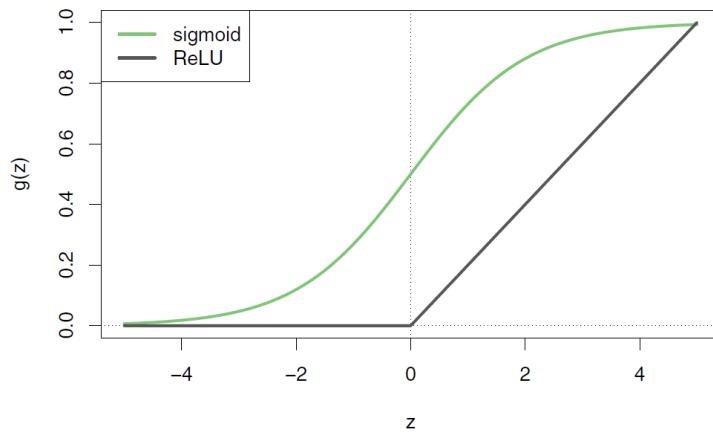


Fig. 4.6: Aktivierungsfunktionen

**Quelle:** James et al. (2021)

Die Sigmoid Funktion wird z.B. auch für die logistische Regression genutzt. Der Vorteil der ReLU Funktion besteht darin, dass Neuronen ausgeschaltet werden, sollte  $z$  zu klein sein, wodurch Rechenkraft gespart wird. Zu dem reagiert das Neuronale Netz stärker auf höhere Werte von  $z$  und lernt damit schneller.

#### 4.5.2 Berechnung eines neuronalen Netzes

Wichtig damit die Aktivierung eines Neurons zur richtigen Vorhersage führt, ist das richtig trainierte Gewicht  $w_{kj}$ . Sowohl die Parameter  $\beta_0, \dots, \beta_K$  als auch  $w_{10}, \dots, w_{Kp}$  müssen trainiert werden. Das Optimierungsproblem dazu besteht in:

$$\min_{\{w_k\}_1^K, \beta} \frac{1}{2} \sum_{i=1}^n (y_i - f(x))^2 \quad (4.9)$$

Um diese Optimierung vorzunehmen verwendet man den Vektor  $\theta$ , der alle zu optimierenden Variablen enthält. Der Fehlerterm ergibt dann:

$$R(\theta) = \frac{1}{2} \sum_{i=1}^n (y_i - f_\theta(x))^2 \text{ mit } \theta = (\{w_k\}_1^K, \beta) \quad (4.10)$$

Zu Beginn des Optimierungsprozesses wird  $\theta^{t=0}$  angenommen, dass heißt der Vektor beginnt ungewichtet. Darauf hin folgen Wiederholungen eines Prozesses, bei dem ein Vektor  $\delta$  gefunden werden muss, sodass  $\theta^{t+1} = \theta^t + \delta$  zu  $R(\theta^{t+1}) < R(\theta^t)$ . In jedem Durchlauf dieses Prozesses wird  $t$  erhöht. Eine eindimensionale Veranschaulichung dieses Prozesses ist in Abbildung 4.7 zu finden.

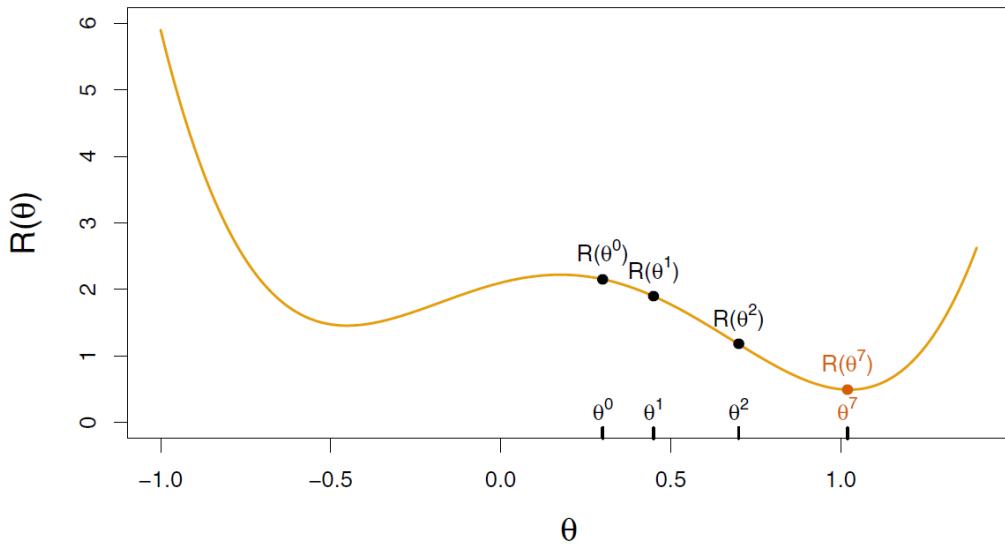


Fig. 4.7: Gradient Descent

**Quelle:** James et al. (2021)

Wie kann aber der Vektor  $\delta$  gefunden werden, sodass der Fehlerterm gesenkt wird? Dazu muss man den Vektor des Gradienten berechnen:

$$\Delta R(\theta^m) = \frac{\partial R(\theta)}{\partial \theta} |_{\theta=\theta^t} \quad (4.11)$$

Der partielle Vektor der Ableitung der aktuellen Schätzung von  $\theta^t$  zeigt aufwärts. Um also  $\delta$  zu finden gehen wir in die entgegen gesetzte Richtung gewichtet mit der Lernrate  $\rho$ . So ergibt sich:

$$\theta^{t+m} \leftarrow \theta^m - \rho \Delta R(\theta^m) \quad (4.12)$$

Die Ableitungen des Fehlerterms nach den Gewichten, die wir brauchen, um den Gradientenvektor in 4.11 zu bestimmen, kann man durch die Ketten Ableitungsregel vereinfachen:

$$\begin{aligned} \frac{\partial R_i(\theta)}{\partial \beta_k} &= \frac{\partial R_i(\theta)}{\partial f_\theta(x_i)} \frac{\partial f_\theta(x_i)}{\partial \beta_k} \\ &= -(y_i - f_\theta)g(z_{ik}) \\ \frac{\partial R_i(\theta)}{\partial w_{kj}} &= \frac{\partial R_i(\theta)}{\partial f_\theta(x_i)} \frac{\partial f_\theta(x_i)}{\partial g(z_{ik})} \frac{\partial g(z_{ik})}{\partial z_{ik}} \frac{\partial z_{ik}}{\partial w_{kj}} \\ &= -(y_i - f_\theta)\beta_k * g'(z_{ik})x_{ij} \\ \text{mit } z_{ik} &= w_{ko} + \sum_{j=1}^p w_{kj}x_{ij} \end{aligned} \quad (4.13)$$

## 4.6 Validation

Statistische Modelle wie ein neuronales Netz machen erstaunlich präzise Vorhersagen. Doch wie kann man eine solche Präzession messen? Dafür verantwortlich ist die Validierung. Mittels Maße wie dem Bestimmtheitsmaß  $R^2$  oder dem RMSE kann man die Performance der Vorhersagen im Vergleich mit den tatsächlichen Daten messen.

Doch wie validiert man die Vorhersagekraft eines Modells auf Daten, mit denen das Modell selbst nicht trainiert worden ist, um zu testen ob das Modell die eigenen Daten nicht overfittet? Hierzu ließe sich der Validation Set Approach nutzen. Teilt man die Beobachtungen in zwei unterschiedlichen Sets zufälligerweise auf, erhält man ein Set, mit dem man ein Modell trainieren kann und eines mit dem man es testen kann. Die Quote nach der man diese Beobachtungen aufteilt nennt man Split. Ein häufig verwendeteter Split ist 80 % für das Trainings Set und 20 % für das Test Set.

### 4.6.1 Cross Validation

Eine Weiterführung dieses Konzepts ist die Cross Validation. Hier wird der Datensatz nicht in 2 sondern in  $k$  viele und gleichgroße Sets unterteilt. Es werden dann  $k$  viele Validierungen vorgenommen. Bei jedem Durchlauf wird eines der Sets zur Validierung und die restlichen Sets zum Training verwendet. Das bietet den Vorteil, dass der gesamte Datensatz zur Validierung und zum Training verwendet wird, während beim herkömmlichen Ansatz immer auf ein Teil der Daten verzichtet werden musste. Außerdem werden so mehr Stichproben erhoben. Man vergleicht nicht ein Modell sonder  $k$ -viele Modelle. Es kann immer vorkommen, dass viele Modelle solide performen, während eines der Modelle zeigt, dass es zu drastischen Fehlern kann. Das heißt die Wahrscheinlichkeit Overfitting aufzudecken ist größer.

### 4.6.2 Conditional Validation Set Building

In dem Fall der Daten der Radzählstation gäbe es zusätzliches Problem, dass auch die Validierung mit dem Validation Set Approach Overfitting wahrscheinlich nicht richtig erkennen ließe. Denn durch eine rein zufällige Aufteilung der Daten in verschiedene Sets, ließe sich die wahre externe Validität des Modells nicht prüfen, denn auch dann träge das Modell immer vorhersagen für fixe räumliche Punkte, die bereits auch im Trainings Datensatz vorkamen.

Um dieses Problem zu umgehen, muss man die Aufteilung der Daten in die verschiedenen Sets auf die Ausprägung der Stationen konditionieren. Das heißt jede Zählstation darf in jedem Set nur einmal vorkommen, damit im Test Set immer Orte vorhanden sind, die dem Modell unbekannt sind. Diese Aufgabe übernimmt der folgende Code:

*Listing 4.1: Aufteilung der Zählstationen*

```

1 while (length(stations_not_chosen)>0)
2 {
3     x1 <- sample(1:length(stations_not_chosen), 1)
4     newStation = stations_not_chosen[x1]
5 }
```

```

6     stations_splits[stations_splits$Station ==
7             newStation ,]$Split = actual_split
8
9     observations_per_splits$Observations[actual_split] =
10    observations_per_splits$Observations[actual_split] +
11    Obs_perStation[Obs_perStation$Station ==
12    newStation ,]$Observations
13
14
15    stations_not_chosen = stations_not_chosen[-x1]
16
17
18    if(median(observations_per_splits$Observations) <=
19        observations_per_splits$Observations[actual_split]){
20
21        actual_split = actual_split + 1
22        if(actual_split>validation_splits)
23            {actual_split=1}
24
25
26    }
27

```

In dem Code 4.1 werden innerhalb einer `while()` Schleife alle Stationen einem der k-vielen Sets hinzugefügt, bis die Schleife die Anzahl der zu verordnenden Zählstationen in `stations_not_chosen` herunter gezählt hat. Die Anzahl von k ist 5 und ist in der Variable `validation_splits` gespeichert. Damit alle Validierungssets ungefähr gleich viele Beobachtungen haben, zählt der Code 4.1 mit, wie viele Beobachtungen mit jeder Station einem Set hinzugefügt werden. Diese Anzahl wurde zuvor in der Tabelle `Obs_perStation` gespeichert. Ob im folgendem Durchlauf der `while()` Schleife das nächste Validierungsset ausgewählt wird, oder nochmal dasselbe hängt an der Anzahl der Beobachtungen. Jedes Validierungsset wird solange mit Zählstationen

befühlt, bis der Median an Beobachtungen aller Sets gleich ist oder überschritten wurde. Dies wird in der `if()` Schleife abgefragt.

Auf diese Weise erhält man fast gleichgroße unterschiedliche Validierungsset, in denen jede Zählstation nur einmal vergeben ist. Teilweise kommt es auch vor, dass ganze Städte nur in je einem Validierungsset vertreten sind, weil es Städte gibt, die nur eine Zählstation mit sich bringen.

#### 4.6.3 Weighted Subset Building

Im Verlauf dieser Masterarbeit sind einige Probleme durch die Größe des Datensatzes entstanden. Da der Datensatz über vier Millionen Beobachtungen beinhaltet und viele Variablen inkludiert ist dessen Speicherplatz auf bis zu 2,7 GB angestiegen. Ein Problem von R ist es, dass alle Daten mit denen gerechnet wird, im Arbeitsspeicher passen müssen. Die Hardware mit deren Hilfe die Modelle der Arbeit entstanden ist, verfügt jedoch nur über 16 GB. Je komplexer die Modelle wurden, desto öfters versagten Skripte mit der Fehler Meldung "run out of memory". Es gäbe an dieser Stelle verschiedene Lösungsansätze. Einer wäre Cloud Computing oder High Performance Computing. Diese Lösungsansätze sind jedoch mit technischen Problemen verbunden.

Der Amazon Web Service bietet z.B. Cloud Computing Dienste an, die auf das Machine Learning Training spezialisiert sind. Um diesen Dienst jedoch in Anspruch zu nehmen, ist eine Kreditkarte notwendig. Dies stellte sich als Problem heraus. Darüber hinaus bietet die Westfälische Wilhelms-Universität Münster High Performance Computing Ressourcen an, die über eine Linux Kommandozeile steuerbar sind. Jedoch müsste dafür erst der Umgang mit Linux vertraut sein.

Beachtet man dies, ist die einfachste Lösung die Größe des Datensatzes zu reduzieren. Dies mag kontraintuitiv erscheinen, ist jedoch sogar mit Vorteilen verbunden. Denn erstellt man ein gutes Subset, dass den Datensatzes mit einem Teil von Beobachtungen darstellt, kann dies dabei helfen Overfitting zu verhindern oder besser zu entdecken.

Der Ansatz, der hier dabei verfolgt wurde, war es, möglichst gleich viele Be-

obachtungen von allen Zählstationen zufällig auszuwählen, dabei aber die Größe jedes Validierungsset auf 200000 Beobachtungen zu begrenzen, sodass der Datensatz auf eine Million Beobachtungen schrumpft. Durch diese Vorgehensweise ist der Anteil an Beobachtung im Datensatz nach räumlicher Verteilung gleichverteilter als zuvor, was dabei helfen sollte Overfitting besser aufzudecken und auch zu verhindern.

## 5. ERGEBNISSE

Das vorherige Kapitel erklärte die funktionsweise verschiedener statistischen Methoden, die sich zur Vorhersage des Radverkehrs anwenden ließen. Viele haben ihre Vor- und Nachteile, alle wurden in der bisherigen Literatur, die mit diesem Thema verwandt ist, verwendet. Folglich wurden alle vier Methoden angewendet und mit der beschriebenen Methode der Cross Validation evaluiert und lassen sich in ihrer Vorhersagekraft vergleichen. (Hier vielleicht Validation Set Building Code einfügen und erklären)

### 5.1 OLS Regression

Die grundlegenste Methode, die an dem erarbeiteten Datensatz getestet wurde, ist die OLS Regression. Diese Methode benötigt nur wenig Rechenkraft und stößt somit nicht an technische Flaschenhälse wie andere Methoden, leistet dafür jedoch nur ungenau Vorhersagen, auch zu entnehmen der Abbildung 5.1. Zum Vergleich wurden sechs unterschiedliche Log-Lin Regressionsmodelle erstellt und jeweils validiert, wobei jedes weitere Modell neue Variablen hinzufügt. Das erste Modell enthält nur zeitliche Variablen wie Jahr, Monat, Stunde, Wochenende, Nacht und Feiertage. Das zweite Modell fügt dem die fünf Wetter Variablen Regen, Temperatur, Bedeckung, relative Feuchte und Windgeschwindigkeit hinzu. Im dritten Modell demographische Daten hinzugefügt, im vierten Modell räumliche Variablen, im fünften quadrierte Variablen und im letzten kubische Variablen. Jedes Modell wurde fünf mal erstellt, für die fünf unterschiedlichen Trainings und Test Sets.

Die roten Linien in der Abbildung 5.1 zeigen den Verlauf des Bestimmtheitsmaßes  $R^2$  für die Trainingsdatensätze und zeigen, dass je mehr Variablen in das Modell aufgenommen werden, desto weiter steigt der erklärte Anteil des

Modells, wobei ein abnehmender Grenzeffekt der Variablenanzahl zu sehen ist. Die blauen Kurven hingegen zeigen den Verlauf des Bestimmtheitsmaßes in den Testdatensätzen. Damit zeigen sie die out of samples Validität der jeweiligen Modelle. Hier ist festzustellen, dass kein anhaltender steigender Trend nicht zu erkennen ist und sogar im Gegenteil, dass die Modelle mit mehr Variablen zu schlechteren Ergebnissen führen. Dies zeigt ganz klar, dass hier Overfitting stattfindet.

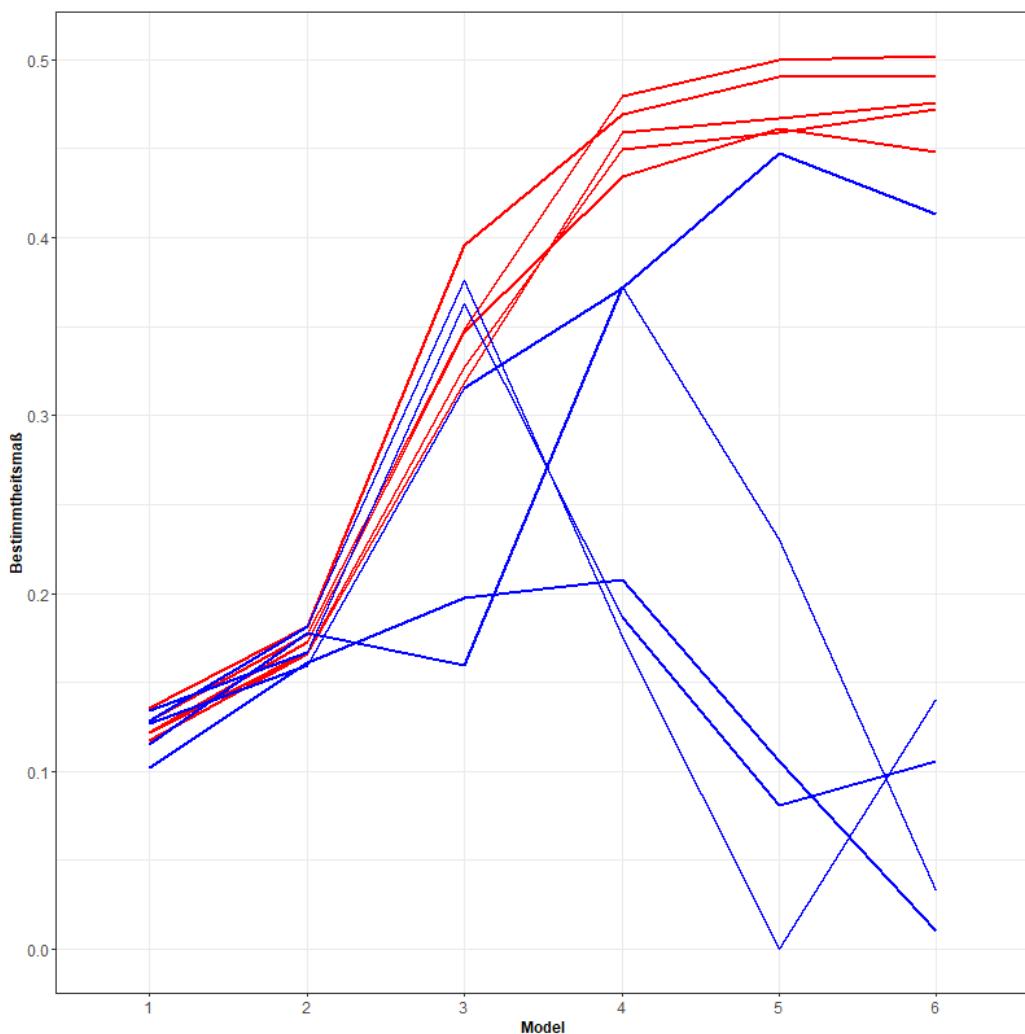


Fig. 5.1: Feature Selection Ergebnis der OLS Regression in 6 Modellen

Dass hier aber Overfitting auftritt ist kein Wunder, denn von dem Modell

Sets	Train $R^2$	Train RMSE	Test $R^2$	Test RMSE
1	0,448	127,441	0,105	184,489
2	0,502	124,486	0,413	113,692
3	0,49	124,969	0,01	153419513741494000
4	0,472	124,189	0,036	2239,056
5	0,476	119,869	0,141	196,462
Schnitt	0,478	124,191	0,14	30683902748299400

Tab. 5.1: Performance des OLS Modells

wird erwartet vorhersagen für, Vorhersagen für Stationen zu treffen, die im Trainingsdatensatz selbst nicht vorkommen und deswegen auch abweichende Charakteristiken aufweisen. Dies macht offensichtlich, dass es ein leistungsfähigeres Modell braucht und bei allen folgenden Machine Learning Algorithmen muss berücksichtigt werden, dass die Gefahr zum Overfitting relativ groß ist, und das vermutlich nie die Performance im Test Set erreicht wird, die sonst im Trainings Set erreicht wird. Die Abbildung 5.1 zeigt die Performance des endgültigen OLS Modells durch den Bestimmtheitswert und den root mean squared error in jedem einzelnen Split. Split 3 zeigt die tatsächliche Tragweite des Overfitting, denn wo hier im Trainings Set ein RMSE 125 steht, nimmt der RMSE im Test Set astronomische Größen an. Dieses Beispiel zeigt auch den Vorteil der Cross Validation, denn dadurch, dass bei der Cross Validation 5 unterschiedliche Test Sets genutzt werden, steigt die Wahrscheinlichkeit, solche Modellfehler offenzulegen.

Es wird also deutlich, dass sich ein Log Lineares OLS Modell nicht dazu eignet, Vorhersagen für einen ganzen Stadtbereich zu machen. Weiterhin lohnt es sich nicht, hier weitere Versuche zu unternehmen. Und damit bleiben noch drei weitere Algorithmen übrig, wobei der nächst folgende die Support Vector Regression ist.

## 5.2 Support Vector Regression

Unter allen Methoden ist die Support Vector Regression jene, die noch die größten Ähnlichkeiten hat zur herkömmlichen nicht linearen OLS Regression. Vorteile waren aber, dass die Support Vector Regression weniger stark

auf Ausreißer reagiert als die OLS Regression. Neben diesen Vorteil hat die Support Vector Regression den Nachteil, dass die Berechnung des Modells deutlich mehr Zeit in Anspruch nimmt und deswegen leistungsfähige Hardware erfordert.

Das R-Paket „e1071“, dass die passende Funktion bietet, stammte von Meyer et al. (2021). R hat den Nachteil, dass es eine single threaded Sprache ist. Es spielt also keine Rolle, wie Leistungsfähig die verwendete CPU ist, solange sich deren Leistung auf verschiedene CPU Kerne aufteilt. R nimmt zur Berechnung immer nur einen CPU Kern. Diese Limitation ließe sich umgehen mit dem Paket „foreach“ von Microsoft and Weston (2022) und „doParallel“ auch von Corporation and Weston (2022), jedoch verhinderte das der zur Verfügung stehende RAM. R legt alle Daten mit denen ein Skript rechnet in den Arbeitsspeicher ab. Würden verschiedene Modelle gleichzeitig von verschiedenen CPU Kernen berechnet werden, dann müssten auch verschiedene Tests und Trainings Datensätze erstellt werden müssen und dies würde in Summe den zur Verfügung stehenden Arbeitsspeicher von nur 16 GB überlasten.

Eine Möglichkeit dieses Problem zu umgehen, wäre es z.B. auf Amazon Web Service zurückzugreifen, die Server zum trainieren von Machine Learning Algorithmen anbieten. Einfacher ist es jedoch, nur einen kleinen zufällig ausgewählten Anteil des Datensatzes zu verwenden. Dabei besteht natürlich die Gefahr, dass die Vorhersagekraft des Modells leidet, wenn man zu wenig Datenpunkte des Datensatzes verwendet. Der Datensatz beinhaltet insgesamt 4.472.091 Beobachtungen. Davon wurden bei der OLS Regression jeweils 80 % zum trainieren des Modells genutzt. Reduziert man diese Trainingsdaten weiter und wählt zufällige Beobachtungen aus, dann verhält sich die Performance des Modells so, wie in der Abbildung 5.2 zu sehen.

Dabei wurde das Modell einmal mit 0.001 %, 0.01 %, 0.1 %, 0.25 %, 0,5 % und 1 % getestet. Als erstes fällt auf, dass die Support Vector Regression bereits bessere Ergebnisse aufweist, als die OLS Regression, so steigt der Bestimmtheitswert für das Trainingsset auf bis zu 75 % im Schnitt an, wo es zuvor noch um die 47 % waren. Aber auch hier findet wieder Overfitting statt, denn der Bestimmtheitswert in den Test Sets steigt nur auf im Schnitt 25 % an. Erhöht man den Anteil an Beobachtungen den man zum Training

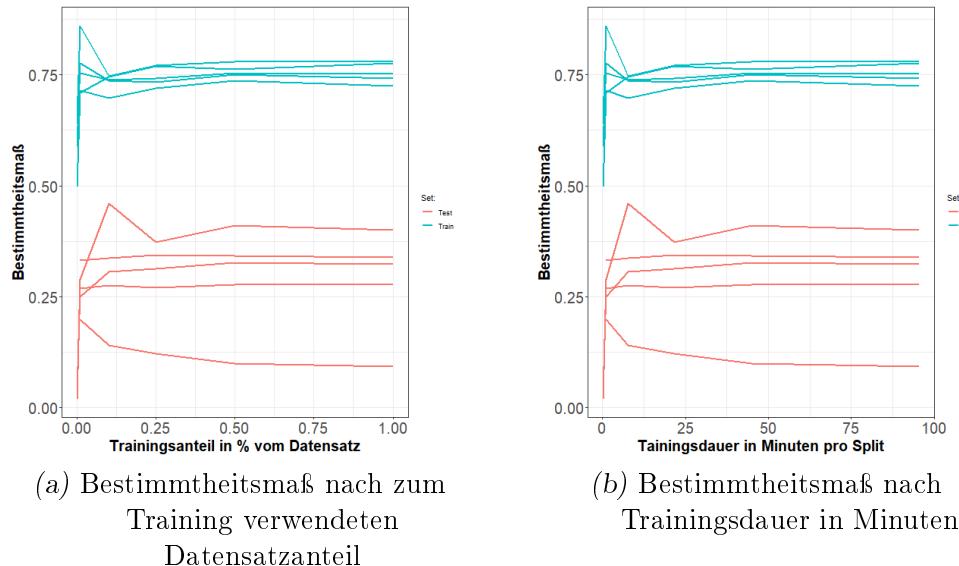


Fig. 5.2: Support Vector Regression Performance nach Anteil des Datensatzes

des Modells verwendet auf 0,25 %, dann findet kaum noch eine nennenswerte Veränderung in der Performance statt. Nur die Dauer die zur Berechnung benötigt wird, steigt signifikant an. Nutzt man 0,25 % der Daten zum Training, was ungefähr 9000 Beobachtungen entspricht, dann benötigt der Computer mit einer Intel Core i7-8750H CPU und 2,2 GHz Leistung im Schnitt 21 Minuten je Modell. Bei 1 % verwendeter Daten, was ungefähr 37.800 Beobachtungen entspricht, benötigt die Berechnung bereits im Schnitt 90 Minuten pro Modell. Da jeweils 5 Modell trainiert werden auf 5 unterschiedlichen Cross Validation Sets und das sechs mal zu unterschiedlichen Anteilen am Datensatz, hat die Erstellung der Ergebnisse in Abbildung 5.2 14,2 Stunden benötigt.

Darüber hinaus kann man auch hier die Performance des Modells anhand der Auswahl der Variablen im Modell vergleichen. Auch hier werden im ersten Modell ausschließlich zeitliche Variablen verwendet, im zweiten kommen die fünf Wettervariablen hinzu, im dritten Variablen, die Demographie, Stadt- ausdehnung, Autobesitz und Fahradklima beschreiben, im vierten räumliche Variablen von Open Street Map, im fünften Modell Variablen zu den Straßentypen und im sechsten nicht lineare Variablen. Kubische Variablen

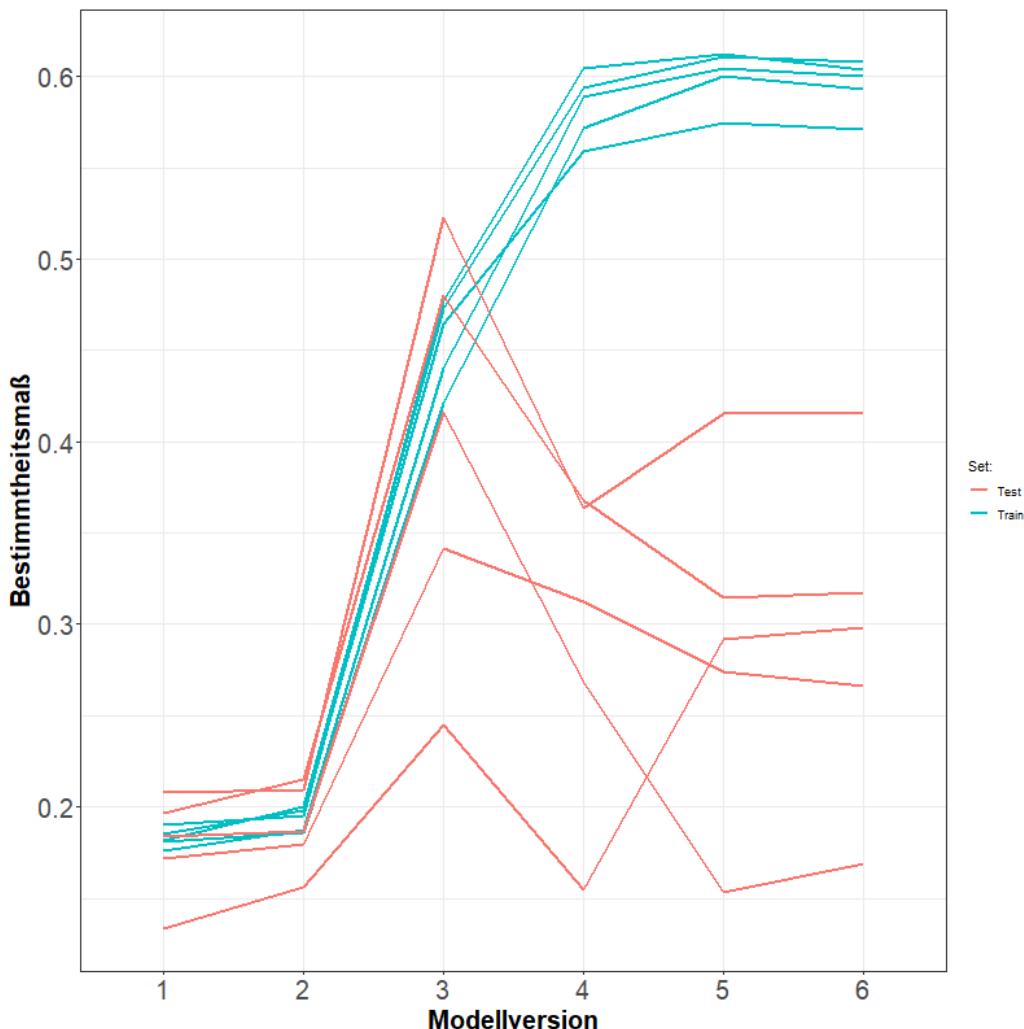


Fig. 5.3: Support Vector Regression Ergebnis der RF Regression in 6 Modellen

sind dieses Mal in keinem Modell enthalten, weil dies zu einer Überlastung des Arbeitsspeichers führte. Alle Modelle wurden mit einem Trainingsanteil von 0,05 % berechnet, also ungefähr 17.900 Beobachtungen. Den Vergleich zwischen den Modellen zeigt die Abbildung 5.3. Diese zeigt deutlich, dass das geringste Maß an Overfitting im dritten Modell stattfindet, erst danach nimmt das Overfitting zu. Das dritte Modell allein ist jedoch von wenig Nutzen, denn ohne die räumlichen Variablen, gibt es keine Möglichkeit räumliche Unterschiede in der Vorhersage hervor zu eben, was der eigentliche Nutzen des Modells sein soll. So ist auch die Support Vector Regression nur wenig

Sets	Train $R^2$	Train RMSE	Test $R^2$	Test RMSE
1	0,741	86,075	0,129	169,814
2	0,768	85,717	0,281	129,19
3	0,765	83,096	0,319	132,127
4	0,742	84,941	0,372	138,756
5	0,745	80,34	0,307	169,438
Mean	0,752	84,034	0,282	147,865

Tab. 5.2: Performance des SVR Modells

dazu geeignet zur räumlichen Vorhersage des urbanen Fahrradverkehrs.

Dies bestätigt auch ein weiterer Blick in die Tabelle 5.2, die die Performance des endgültigen Support Vector Modells zeigt, mit allen Variablen, bis auf den kubischen. Im Vergleich zum OLS Modell sehen wir keine übermäßigen Ausreißer mehr im RMSE der Test Datensätze, aber die Performance verbessert sich nicht ausreichend genug, um damit vernünftige Aussagen treffen zu können.

### 5.3 Random Forest Regression

Die bisherigen Versuche ein schlüssiges Modell zu berechnen förderten leider nur unzufriedenstellende Ergebnisse hervor. Deswegen wird es Zeit leistungsfähigere Methoden zu verwenden. Das R Paket „randomForest“ von Liaw and Wiener (2002). Ähnlich wie Support Vector Regressionen sind Random Forest deutlich rechenintensiver als eine OLS Regression. Deswegen wird auch hier nur ein Teil der Daten zum Training verwendet. Wie sich die Performance des Random Forest Models nach Anteil der zur Berechnung verwendeten Daten ändert, zeigt die Abbildung 5.4, deren Berechnung insgesamt 9,35 Stunden benötigte. Dabei wird sichtbar, dass qualitativ wie auch quantitativ die Random Forests performanter sind, als die vorherhige Support Vector Regression. In Abbildung 5.2 war noch zu sehen, dass die Berechnung mit 1 % der Daten ca 90 Minuten pro Modell benötigte. Hier sind es für 2 % nur noch 80 Minuten.

Auch hier findet sich wieder Overfitting. Dennoch ist die Vorhersagesicherheit bedeutend besser als zuvor. Im Trainingsset erreicht das Random

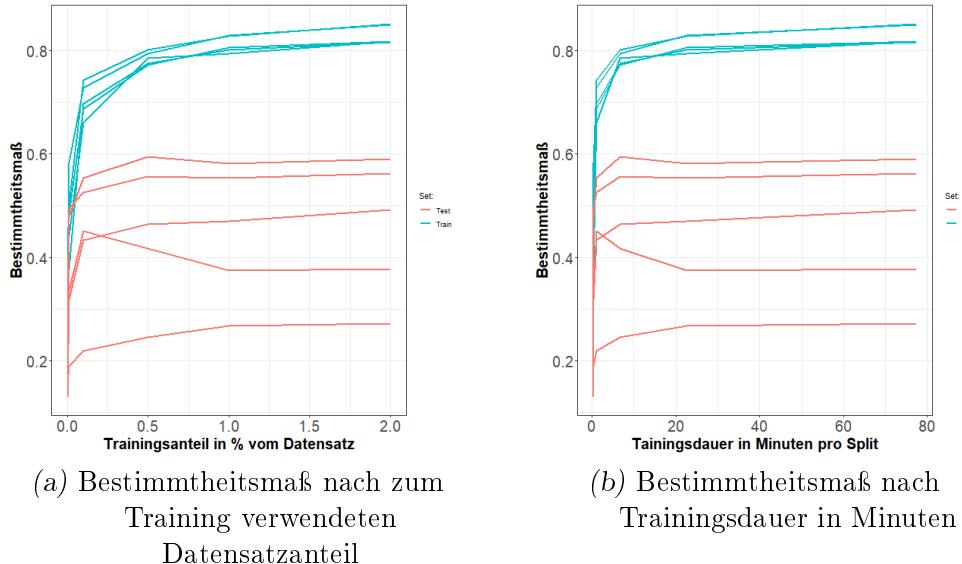
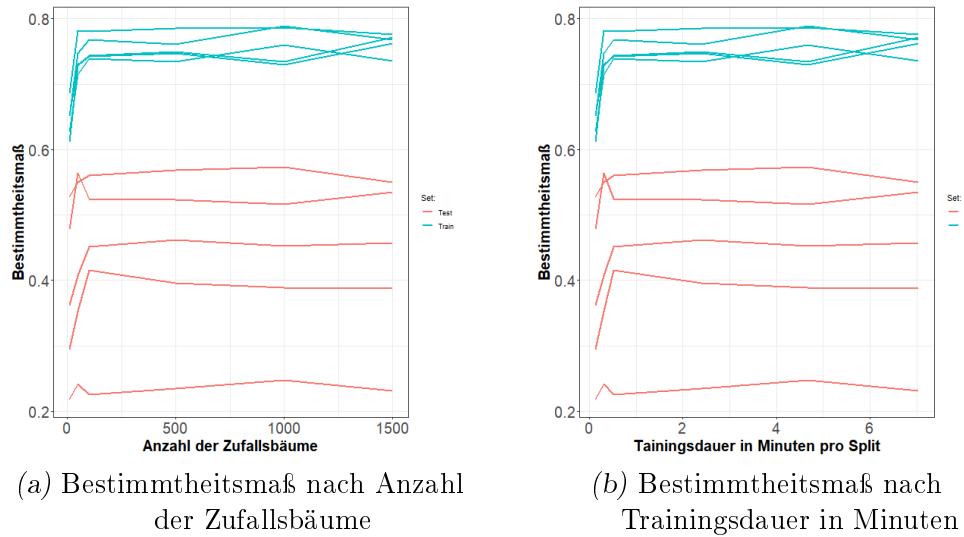


Fig. 5.4: Random Forest Performance nach Anteil des Datensatzes

Forest Modell einen Bestimmtheitswert von 80 % während die Support Vector Regression 75 % maximal erreichte. Im Test Set sieht der Unterschied noch besser aus. Hier werden nun bis zu 60 % erreicht. Zur Berechnung der Abbildung 5.4 wurden im Modell 500 Zufallsbäume verwendet. Diese Anzahl lässt sich variieren. Wie sich die Performance des Modells mit der Anzahl an verwendeten Zufallsbäumen verändert, zeigt die Abbildung 5.5, deren Berechnung 1,27 Stunden gebraucht hat. Dabei fällt schnell auf, dass das Maximum an Performance mit bereits relativ wenigen Zufallsbäumen bereits erreicht ist.

Darüber hinaus bleibt nur noch der Vergleich von Modellen mit verschiedenen Variablen übrig. Dazu fand die selbe Modellauswahl wie im Abschnitt zur OLS Regression statt. Das heißt das Modell 1 greift auf Variablen zum Jahr, Monat, der Stunde, zum Wochenende, Nacht, und den Ferien- wie auch Feiertagen. Modell 2 fügt Wettervariablen hinzu. Modell 3 nutzt zusätzlich demographische Variablen, Bevölkerungsanzahl, Fläche, Autobesitzrate, Immigrantenanteil und Fahrradklimaindex. Im Modell 4 kommen die Open Street Variablen hinzu. In den darauf folgenden Modellen kommen noch die nicht linearen Effekte hinzu. Zu sehen ist der Vergleich in der Performance



(a) Bestimmtheitsmaß nach Anzahl der Zufallsbäume

(b) Bestimmtheitsmaß nach Trainingsdauer in Minuten

Fig. 5.5: Random Forest Performance nach Anzahl der Zufallsbäume

Sets	Train $R^2$	Train RMSE	Test $R^2$	Test RMSE
1	0,77	83,783	0,368	139,04
2	0,809	79,122	0,473	116,077
3	0,813	78,826	0,265	134,78
4	0,778	84,372	0,542	140,007
5	0,784	77,452	0,575	154,054
Mean	0,791	80,712	0,445	136,791

Tab. 5.3: Performance des RF Modells

der unterschiedlichen Modelle in der Abbildung 5.6. Einen ganz signifikanten Anstieg in der Performance ist zwischen Modell 2 und 3 zu sehen. Danach finden kaum große Änderungen in der Performance sowohl im Trainings Datensatz, als auch im Test Datensatz.

Kombiniert man die so gewonnen Erkenntnisse zur Performance von Random Forest dann kommt man zu den Ergebnissen der Cross Validation in der Tabelle 5.3. Dieses Modell nutzte 0,5 % der Trainingsdaten, 250 Zufallsbäume, verzichtet auf nicht lineare Variablen und benötigte zur Berechnung 24 Minuten. Im Test Datensatz erreicht das Random Forest Modell ein Bestimmtheitsmaß von 44,5 %. Das ist mehr als dreimal so viel wie im OLS Modell und mehr als 1,5 mal so viel wie mit der Support Vector Regression.

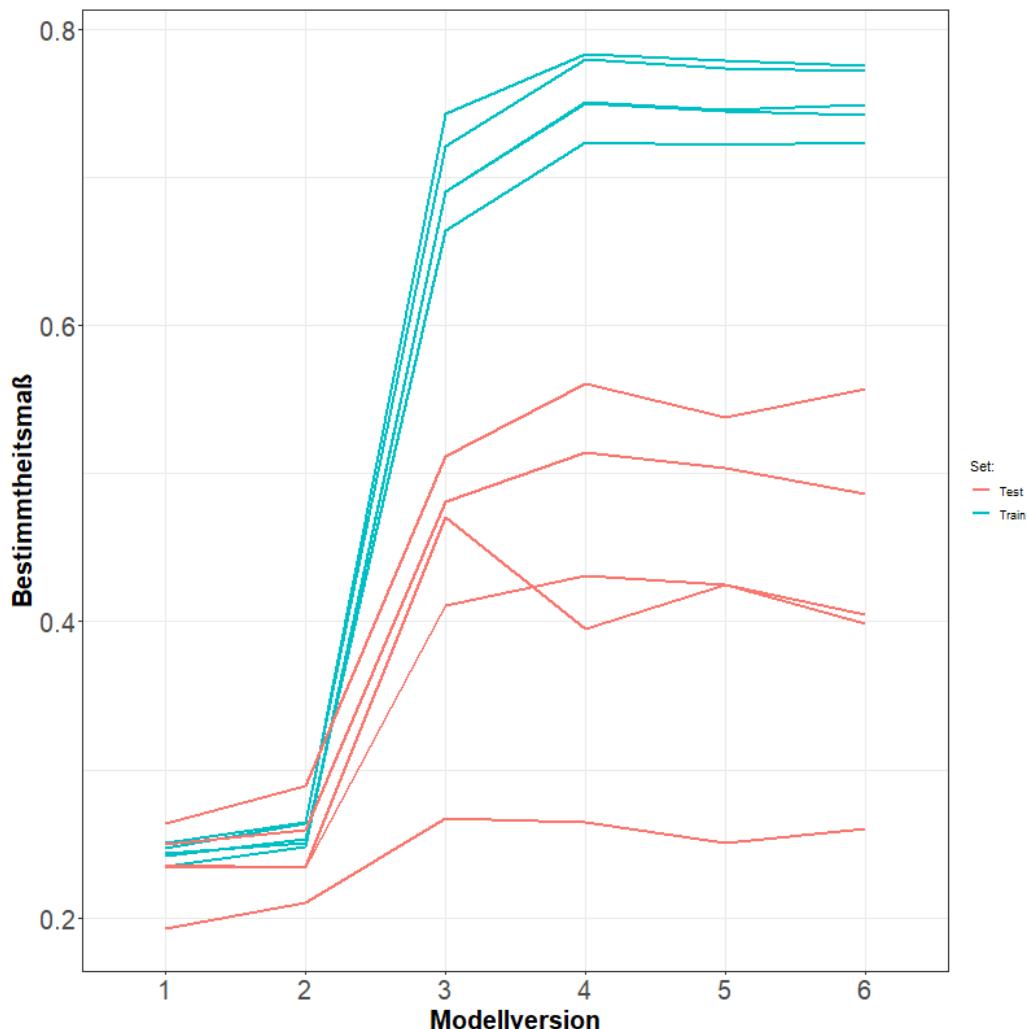


Fig. 5.6: Feature Selection Ergebnis der RF Regression in 6 Modellen

Auch wenn diese Werte bereits deutlich besser sind, als die der vorherigen Modelle, kann dies nicht darüber hinweg täuschen, dass sowohl die Höhe des Bestimmtheitsmaßes als auch des RMSE noch zu wünschen übrig lässt, doch in Anbetracht der schwierigen Aufgabe, die dieses Modell zu erfüllen hat, sind die Ergebnisse dennoch interessant, denn das Modell hat nur wenige Beobachtungsorte, auf deren Grundlage es Vorhersagen für ein ganzes Stadtgebiet treffen soll. Die letzte Methode, die nun noch übrig bleibt und die vielleicht in der Lage ist, noch bessere Vorhersagen zu treffen, ist das neuronale Netz.

Wendet man auf dieses Modell den zweiten Datensatz an, der Corona Daten beinhaltet und detailliertere Daten über die Straßenbeschaffenheit dann ändert sinkt das Bestimmtheitsmaß auf 41,16 % im Test Set und der RMSE steigt leicht auf 137,43, während im Trainingsset die Werte besser aussehen mit einem Bestimmtheitsmaß von 81,52 % und einem RMSE von 74,67. Das heißt hier ist das Overfitting sogar noch ein wenig größer.

#### 5.4 Neuronales Netz

Im Vergleich zu den vorherigen Methoden sind neuronale Netze die populärste Methode, vor allem bekannt für die verblüffende Präzision ihrer Vorhersagen. Doch verblüffen die neuronalen Netze auch bei der Vorhersage des urbanen Radverkehrs? Das hängt vor allem von den vielen Variablen ab, die man bei einem neuronalen Netz einstellen kann. Zum einen kann man festlegen, wie viele Ebenen ein neuronales Netz haben soll und wie viele Knotenpunkte jede Ebene haben darf, die Informationen an die nächste Ebene weiter geben.

Leider ist jeder Durchlauf eines neuronalen Netzes sehr zeitintensiv. Diese Limitation erschwert den Vergleich in der Performance vieler verschiedener Modelle. So dauerte die Durchlaufzeit des Modells aus der Tabelle 5.4 ganze 25,7 Stunden mit einer Anzahl an Beobachtungen von 20000. Würde man die Anzahl an Beobachtungen erhöhen, könnte das neuronale Netz unter Umständen bessere Vorhersagen treffen, dies würde jedoch auch die Durchlaufzeit erhöhen.

Grundsätzliche ist das Problem dieses Modells aber Overfitting. Von allen betrachteten Modellen ist hier das Overfitting am stärksten, betrachtet man die Unterschiede zwischen dem  $R^2$  Wert im Trainings-Set der im Schnitt bei 93 % liegt und im Test Set, wo der selbe Wert im Schnitt nur 9% beträgt. Die Differenz könnte nicht größer sein.

Das verwendete neuronale Netz nutzte sechs Neuronenreihen mit je 48, 26, 16, 8, 4 und 2 Neuronen. Hier müsste evaluiert werden, ob dieser Aufbau sich verbessern ließe. Die Anzahl von Trainingsschritten war auf 100000 begrenzt. Ein häufig auftretendes Problem war, dass das neuronale Netz keine

Sets	Train $R^2$	Train RMSE	Test $R^2$	Test RMSE
1	0,9223	162,471	0,069	220,811
2	0,937	376,894	0,005	620,447
3	0,9484	250,083	0,042	243,739
4	0,946	475,38	0,276	662,502
5	0,925	201,328	0,05	280,962
Mean	0,936	293,231	0,089	405,692

Tab. 5.4: Performance des neuronalen Netzes

Konvergenz zum Schwellenwert für die partielle Ableitung der Fehlerfunktion herstellen konnte, ganz gleich wie viele Trainingsschritte gemacht worden. Deswegen ist auch der Schwellenwert auf 0.025 hoch gesetzt. Dieser Wert ist zu hoch, war aber notwendig um überhaupt Ergebnisse hervor zu bringen. Je Modell wurden drei wiederholte Trainingsansätze genutzt. Diese Anzahl könnte man noch erhöhen um mehr Generalisierung zu erzielen.

Im Vergleich aller Modelle brachte das Random Forest Modell die besten Vorhersagen hervor. Im weiteren Verlauf wird also immer von dem Random Forest Modell die Rede sein.

## 5.5 Model Projektion

In den vorangegangenen Sktionen wurden die Ergebnisse vier verschiedener Machine Learning Modelle vorgestellt. Dies waren die Ergebnisse der OLS Regression, der Support Vector Regression, von Random Forest und die Ergebnisse eines neuronalen Netzes. Dabei haben die Ergebnisse des Random Forests die Ergebnisse mit der höchsten Plausibilität geliefert getestet durch die Cross Validation.

Dieses Modell lässt sich nun dazu nutzen, Vorhersagen auf ein ganzes Straßennetz zu projizieren. Die Daten des Straßennetzes stammen wiederum von Open Street Map. Der erste Schritt ist es, einen Kartenbereich auszuwählen mithilfe zweier Koordinaten wie in Code 5.1. Dabei ist anzumerken, je größer dieser Kartenausschnitt sein soll, desto höher ist natürlich auch die Datenlast. Bei ganzen Stadtgebieten ist es zu empfehlen, nur einzelne Straßentypen auszuwählen, oder den Prozess in mehrere Ausschnitte aufzuteilen. Die Koor-

dinaten werden in myLocation gespeichert. Der hier dargestellte Code würde einen Code Ausschnitt des Ringbereichs in Münster auswählen.

*Listing 5.1:* Wahl des Kartenausschnitts

```

1 myLocation <- c(7.597514856738869, 51.94573812395569,
2                               7.652382675482133, 51.9756143280805)
3
4 q <- myLocation %>%
5 opq() %>%
6 add_osm_feature("highway")
7
8 streets <- osmdata_sf(q)

```

Danach wird eine Anfrage an Open Street Map gestellt, die Daten des Typs "highway" im angegebenen Kartenbereich zu downloaden. Dieser Prozess ähnelt dem Prozess, wie wir ihn im Abschnitt 3.5 kennen gelernt haben. Darauf folgend müssen die ausgelesenen Daten in einem Format gespeichert werden, die in einen Datensatz für das Modell und zur Kartenprojektion verwendet werden könnten. Zunächst bestehen die Open Street Map Daten aus verschiedenen Straßenlinien, deren Koordinaten in `streets$osm_lines$geometry` gespeichert sind. Jede Straßelinie besteht aus mehreren und zusammenhängenden Vektoren, und diese Vektoren bestehen je aus zwei Koordinaten. Zunächst müssen die einzelnen Vektoren der Straßenlinien voneinander getrennt werden, damit jeder Vektor mit einem seiner beiden Koordinaten im Datensatz wie eine Zählstation behandelt werden kann. Dabei wird die erste Koordinate in `streetPositions$Lon` und `streetPositions$Lat` gespeichert und die zweite Koordinate in `streetPositions$Lon2` und `streetPositions$Lat2`. Aufgrundlage der ersten Koordinaten können dann Vorhersagen getroffen werden. Der Code 5.2 zeigt diesen Prozess.

Dabei werden zwei `for`-Loops verwendet.

*Listing 5.2:* Wahl des Kartenausschnitts

```

1 for(i in 1:length(streets$osm_lines$geometry)){
2   l = length(streets$osm_lines$geometry[[i]])

```

```

3
4      for(j in 1:(length(streets$osm_lines$geometry[[i]])/2 - 1)){
5
6          streetPositions$Lon[k]=streets$osm_lines$geometry[[i]][j]
7          streetPositions$Lat[k]=streets$osm_lines$geometry[[i]][1/2+j]
8          streetPositions$Lon2[k]=streets$osm_lines$geometry[[i]][j+1]
9          streetPositions$Lat2[k]=streets$osm_lines$geometry[[i]][1/2+j+1]
10
11      }
12
13  }
14
15 }
```

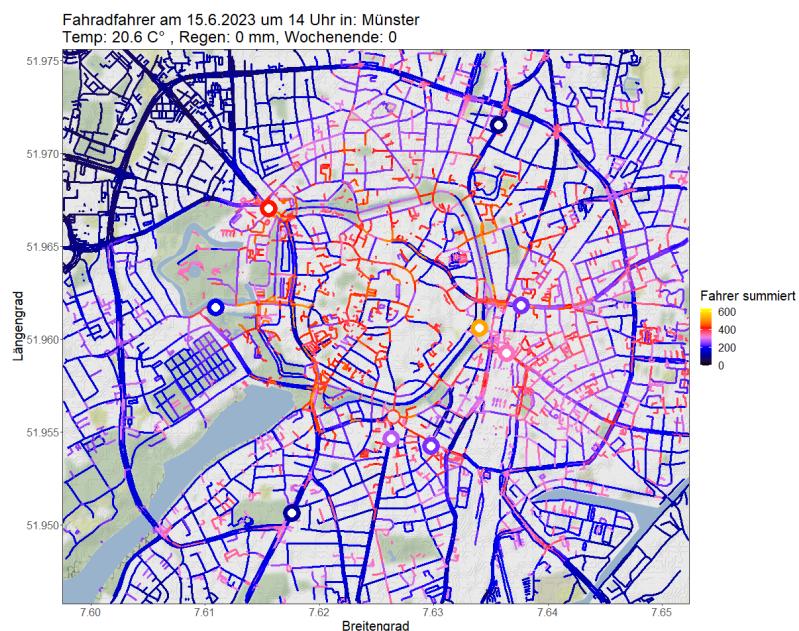
Der erste `for`-Loop geht alle Straßenlinien durch. Der zweite `for`-Loop geht alle Vektoren innerhalb der jeweiligen Straßenlinie durch. Innerhalb der zweiten Schleife werden dann die Einzelwerte der jeweiligen Koordinate in dem Dataframe `streetPositions` gespeichert, dieser wird darauffolgend als Grundlage für die Vorhersagen genutzt. Die Variablen die zu Vorhersagen benötigt werden, werden ähnlich wie in Kapitel 3 gebildet. Darüber hinaus kann man einzelne Faktoren auch überschreiben, um hypothetische Fälle zu betrachten.

Hat man nun einen solchen Datensatz, der die Vorhersagen des jeweiligen Modells enthält, ist die beste und anschaulichste Präsentation dieser Ergebnisse eine graphische Übersicht des Straßennetzes mit farblicher Markierung der Fahrradauslastung, wie sie für die in Abschnitt 5.3 zwei verschiedenen Random Forest Modell in Abbildung 5.7 zu sehen sind. Zusätzlich werden die im Datensatz enthaltenen Zählstellen als Ringe auf der Karte dargestellt, deren farblich die durchschnittliche Radauslastung an der jeweiligen Zählstation markiert.

Wie in der Abbildung 5.7 unterscheiden sich beide Modelle sehr stark. Während in der Abbildung zum ersten Modell zu sehen ist, dass das Auf-



(a) nach Modell 1



(b) nach Modell 2

Fig. 5.7: Räumliche Modell Projektion: Modellvergleich

kommen des Radverkehrs in der räumlichen Vorhersage flächenmäßig relativ gleichverteilt ist aber konzentriert auf das Stadtzentrum, ist die Vorhersage des zweiten Modells weniger gleichmäßig. In der zweiten Abbildung ist ein stärkeres Rauschen zu sehen. Dabei ergibt das Rauschen der Vorhersage an manchen Stellen Sinn, an anderen jedoch weniger Sinn. Was nützlich an Modell 2 ist, dass bestimmte einzelne Straßenzüge stärker betont werden, vor allem auch spezifische Kreuzungen im Bereich der Ringstraße Münsters. Es fallen z.B. auf die Kreuzung Niedersachsenring/Bohlweg, Niedersachsenring/Gartenstraße, Kaiser-Wilhelm-Ring/Warendorfer Straße oder auch die Kreuzung Kolde-Ring/Weseler Straße. Dies sind jedoch auch wichtige Verkehrsadern der Stadt Münster, dass eine Radverkehrsdichte im Bereich der Ring Straßen herrschen muss, scheint trivial. Doch kann das Modell auf vorhersagen, dass dies nicht im ganzen Ring Bereich so sein muss. Für den Nordwesten des Ringbereichs sagen beide Modelle eine niedrige Radverkehrsdichte hervor. Ohne einer weiteren Zählstation lässt sich dies nicht abschließend prüfen, wir kennen jedoch auch die gemessene Validität der Modelle. Darüber hinaus macht das zweite Modell für einige Nebenstraßen sehr hohe Vorhersagen. Diese erscheinen nicht plausibel. Eine Möglichkeit wäre es, diese Nebenstraßen und Parkgassen in der Anzeige auszuschließen, so wie es z.B. auch mit Gehwegen und Fußgängerzonen geschehen ist. Die graphische Ausgabe der Modellvorhersagen lässt sich also noch verbessern.

Mit der Absicht die Validität des Modells noch weiter zu erhöhen ist ein drittes Modell entstanden, dass alte und neue Straßentypen Variablen mit in das Modell aufnimmt, zusätzlich noch mehr nicht lineare Effekte und eine größere Stichprobenmenge von 80000 Beobachtung nutzt. Dieses dritte Modell hat in der Berechnung 28 Stunden benötigt. Die Ergebnisse zeigt die Tabelle 5.5 und die Abbildung 5.8. Graphisch ist dieses Modell fast identisch zum zweiten Modell. Da Modell 2 und 3 brauchbarere Vorhersagen machen, weil beide einzelnen Straßenzüge hervorheben und weil Modell 3 eine höhere Treffsicherheit als Modell 2 hat und mit mehr Beobachtungen trainiert worden ist, wird für weitere Vorhersagen Modell 3 verwendet.

Die Abbildungen 5.9 und 5.10 zeigen beispielhaft die Verkehrsprognosen für Hamburg, Leipzig, Mannheim und Oberhausen. Es wird offensichtlich,

Sets	Train $R^2$	Train RMSE	Test $R^2$	Test RMSE
1	0,842	72,581	0,357	97,473
2	0,84	69,092	0,508	142,219
3	0,866	63,878	0,282	153,957
4	0,844	67,856	0,56	139,983
5	0,832	69,49	0,538	142,357
Mean	0,845	68,579	0,443	135,198

Tab. 5.5: Performance des dritten RF Modells

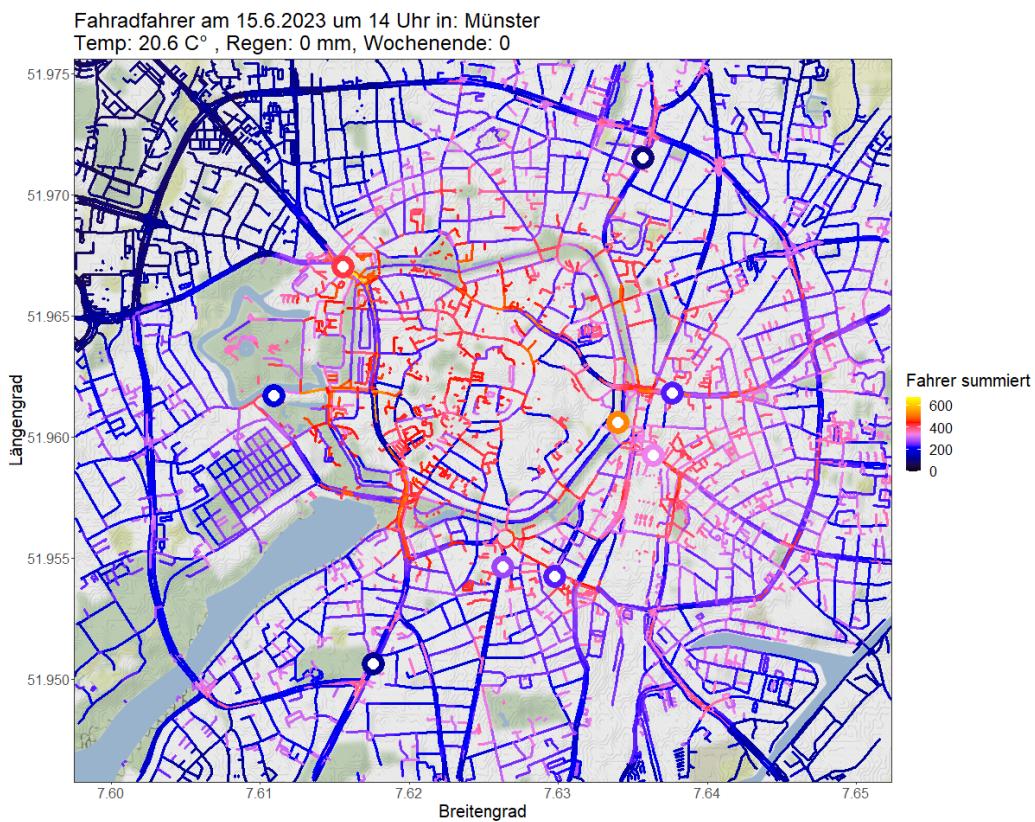
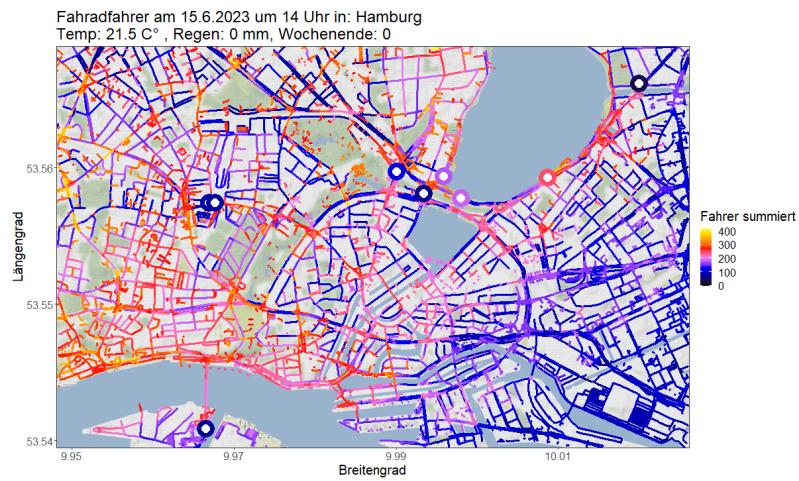
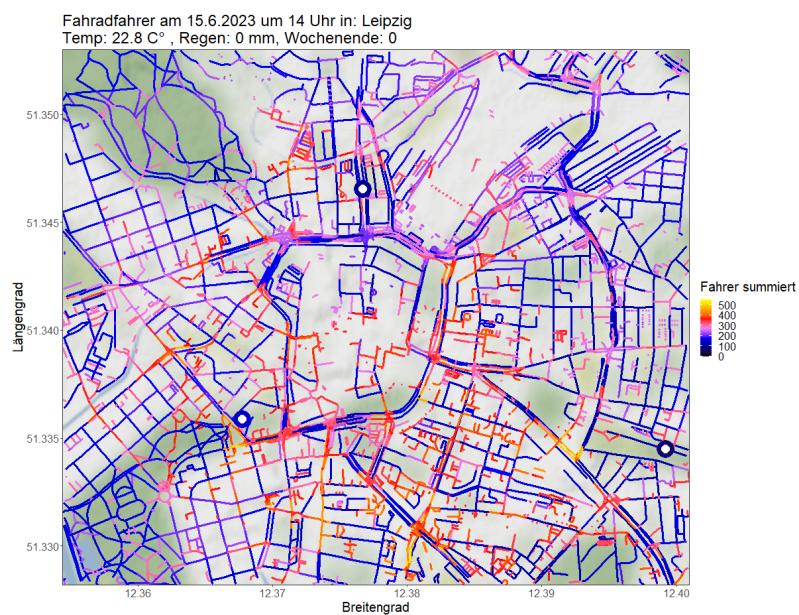


Fig. 5.8: Feature Selection Ergebnis der RF Regression in 6 Modellen

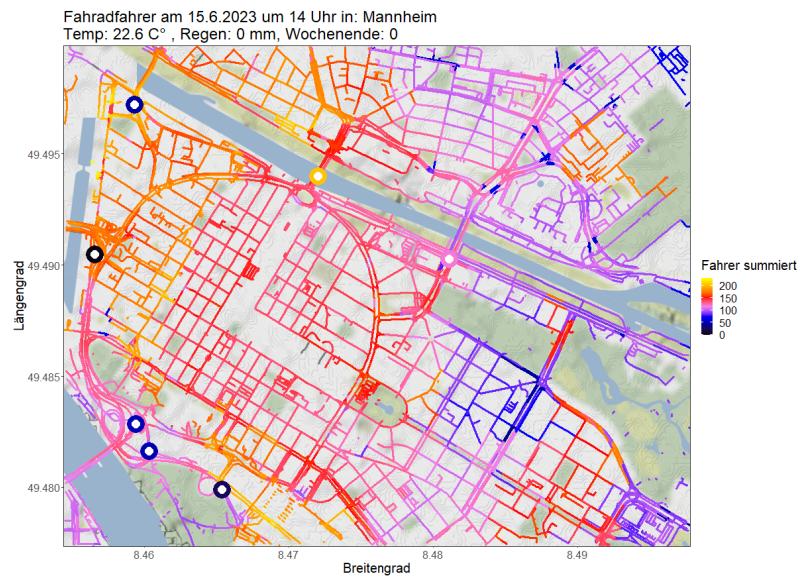


(a) Hamburg

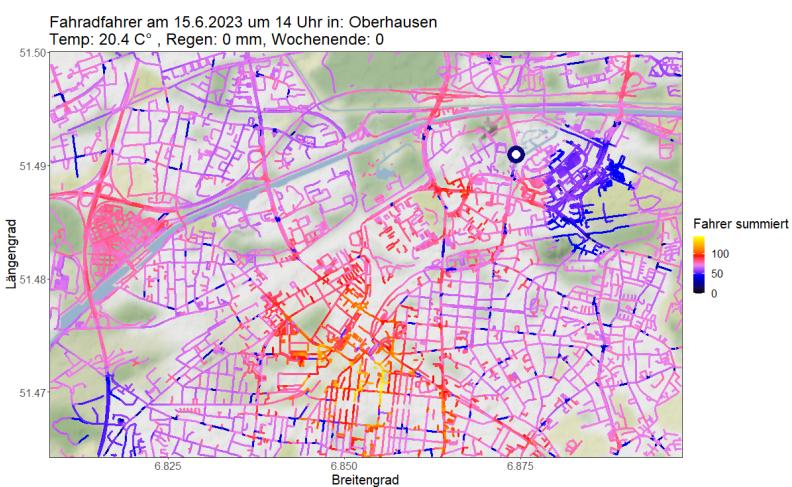


(b) Leipzig

Fig. 5.9: Modellprojektionen in Hamburg und Leipzig



(a) Mannheim



(b) Oberhausen

Fig. 5.10: Modellprojektionen in Mannheim und Oberhausen

dass das Modell sich dem Verkehrsniveau der Städte anpasst. An manchen Stellen scheint das Modell das Verkehrsaufkommen zu überschätzen, vergleicht man es zu den markierten Stationen. Die markierten Stationen zeigen jedoch nur den Durchschnitt über alle Messnungen hinweg an. Da die Modellprojektionen ein Datum in der Zukunft gewählt haben im Sommer, an dem Spitzenwerte auftreten ist hier ein Vergleich schwierig.

Um das Modell einem besonderen Stresstest zu unterziehen, muss Vorhersagen für eine ganze Stadt machen, die dem Modell unbekannt ist, Messwerte zu dieser Stadt aber vorliegen. Weil im Verlauf der Masterarbeit die Zählwerte aus Dresden erst spät zugänglich waren, konnten diese Daten selbst nicht mit in das Modell einfließen, sind nun aber nützlich für eine zusätzliche Validierung. Diese Daten aus Dresden stammen von 8 verschiedenen Zählstationen von 2018 bis 2021 und umfassen insgesamt 261576 Beobachtungen. Das Modell schätzt hier in Dresden über alle Beobachtungen den Durchschnitt der Radfahrer auf 40.95. Der tatsächliche Durchschnitt entspricht jedoch 83.66 und ist somit doppelt so hoch. Der vorhergesagte Medianwert liegt bei 27.06 und der tatsächliche Medianwert liegt bei 43. Das Modell scheint also den Verkehr in Dresden zu unterschätzen. Der RMSE für diese Daten liegt bei 115.76, was im Bereich der vorherigen Evaluationen lag, jedoch liegt das Bestimmtheitsmaß  $R^2$  nur noch bei 0.06.

Dies sind Hinweise darauf, dass Modellprojektionen in unbekannten Städten unzuverlässig sind. Diese Vermutung bestätigt sich beim graphischen Modelloutput in Abbildung 5.11. Dieser Modelloutput zeigt im Zentrum Dresdens wenig Verkehr in Randbezirken mehr und wirkt deswegen unglaublich. In Verbindung mit den Validierungswerten des RMSE und des Bestimmtheitswertes sowie des unterschiedlichen Medians und Durchschnitts in vorhergesagten und tatsächlichen Verkehrsdaten, zeigt das Dresdner Beispiel die Grenzen des Modells. Wo die Modellprojektionen für dem Modell bekannte Stationen glaubwürdig validiert werden konnten, sind die Modellvorhersagen für die unbekannte Stadt Dresden unglaublich.

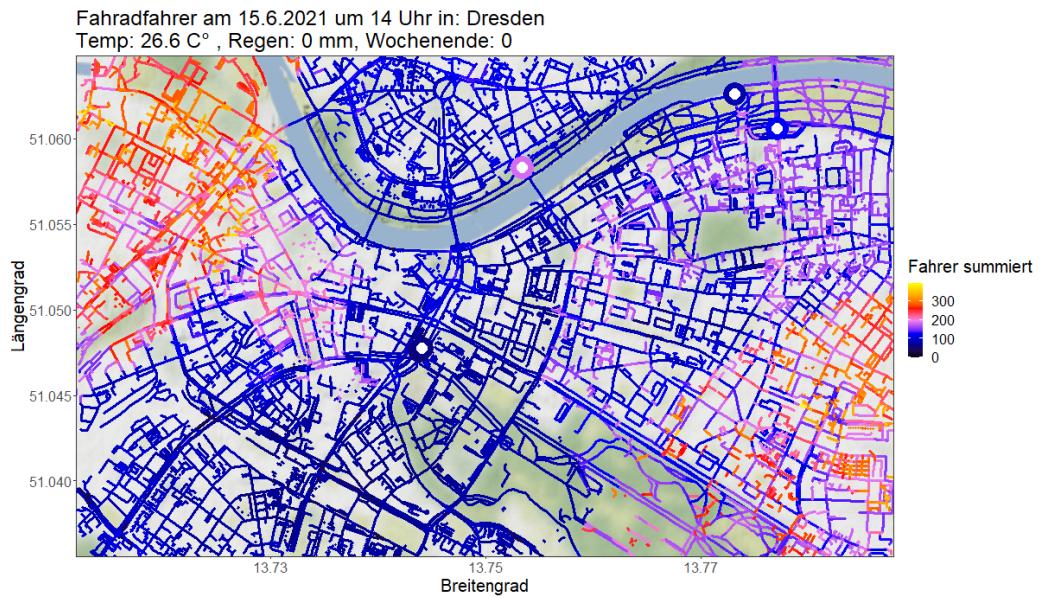


Fig. 5.11: Modellprojektionen in Dresden

## 5.6 Räumliche Korrelation zu Verkehrsunfällen

Auch wenn wie im vorherigen Abschnitt dargestellt die Vorhersagen für unbekannte Städte unbrauchbar sind, ist die Verwendung für die 22 deutschen Städte, die im Modellsatz vorkommen denkbar. Eine weitere interessante Frage ist, ob die Modellvorhersagen räumlich mit Radverkehrsunfällen korrelieren.

## 6. DISKUSSION

Zählstationen werden oft nicht zufällig gesetzt sondern mit Absicht an Stellen, wo man einen hohen Verkehr erwartet. Dies verletzt aber ebenfalls die Annahme von unabhängig und gleichverteilten Beobachtungen. Dies ist der Grund für das massive Overfitting, welches den Modellen allen zu Grunde liegt. Jedoch ließ sich durch Cross Validation dieses Overfitting aufdecken. Der Wert der für die externe Validität bemessen wurde steht. Hinter ihm verstecken sich keine weitere Überraschungen von Verzerrungen, die einem in der Anwendung des Modells noch begegnen können. Das heißt mithilfe der Cross Validation lässt sich die Performance des Modells getreu beurteilen. Die übrig bleibende Frage ist, ob diese Performance der Anwendung zur Verkehrsplanung und zur Unfallvermeidung ausreicht.

### 6.1 Fazit

## 7. ANHANG

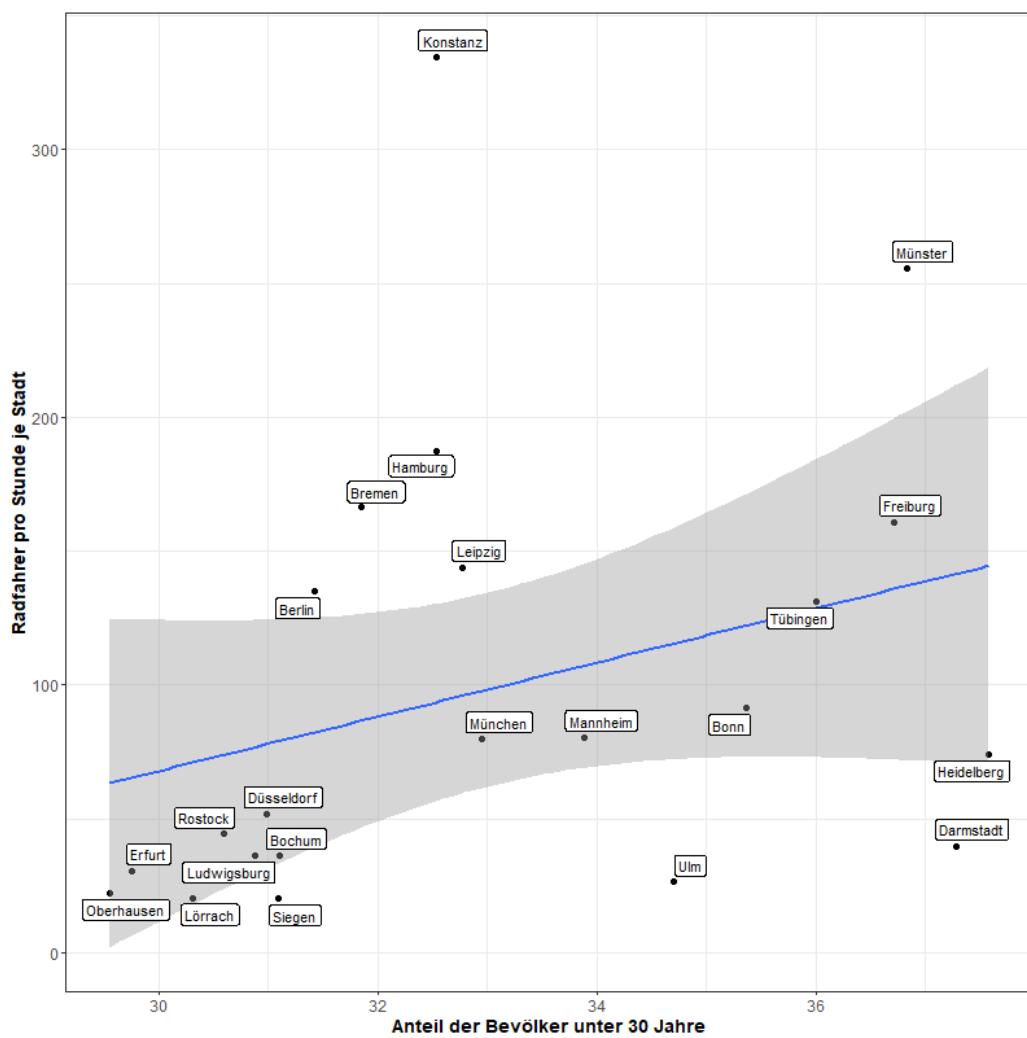


Fig. 7.1: Verteilung des Fahrradaufkommens nach Alter

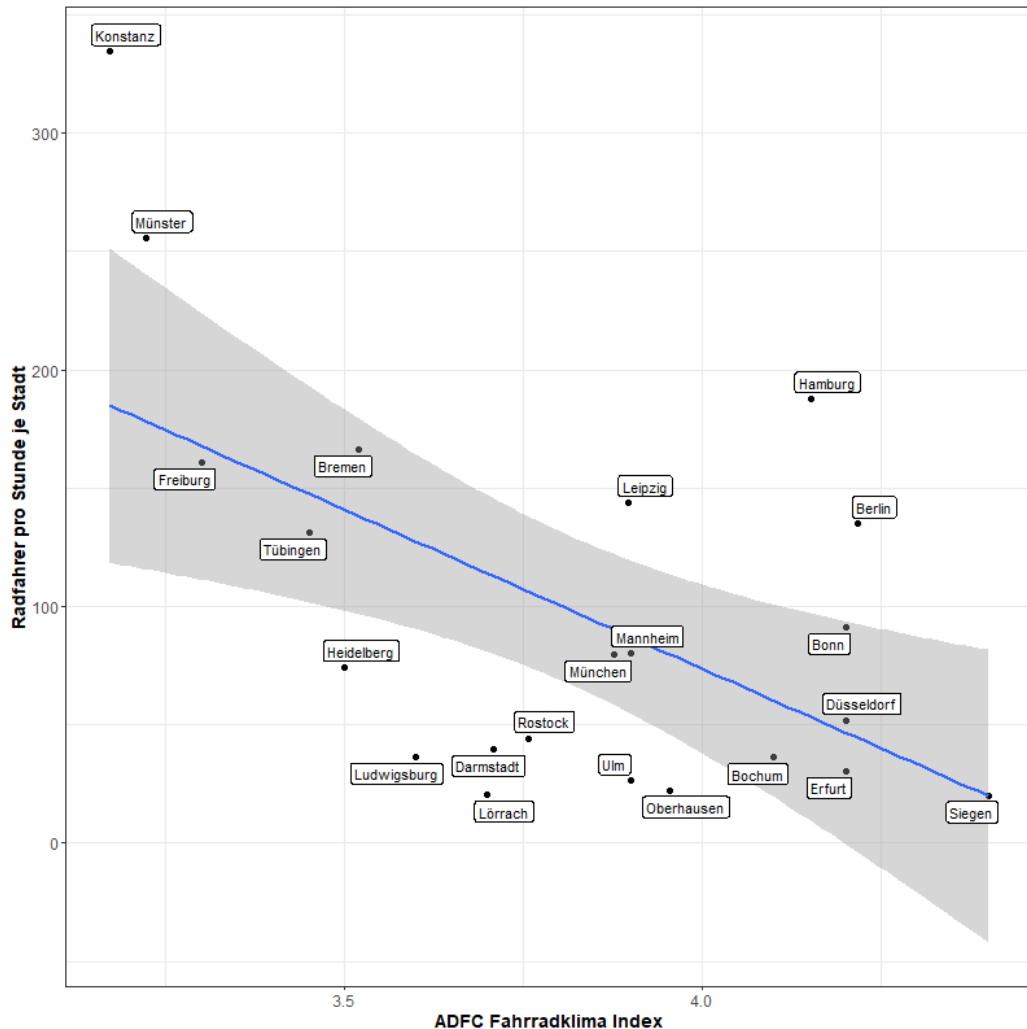


Fig. 7.2: Zusammenhang von Fahrradklima und Radverkehr

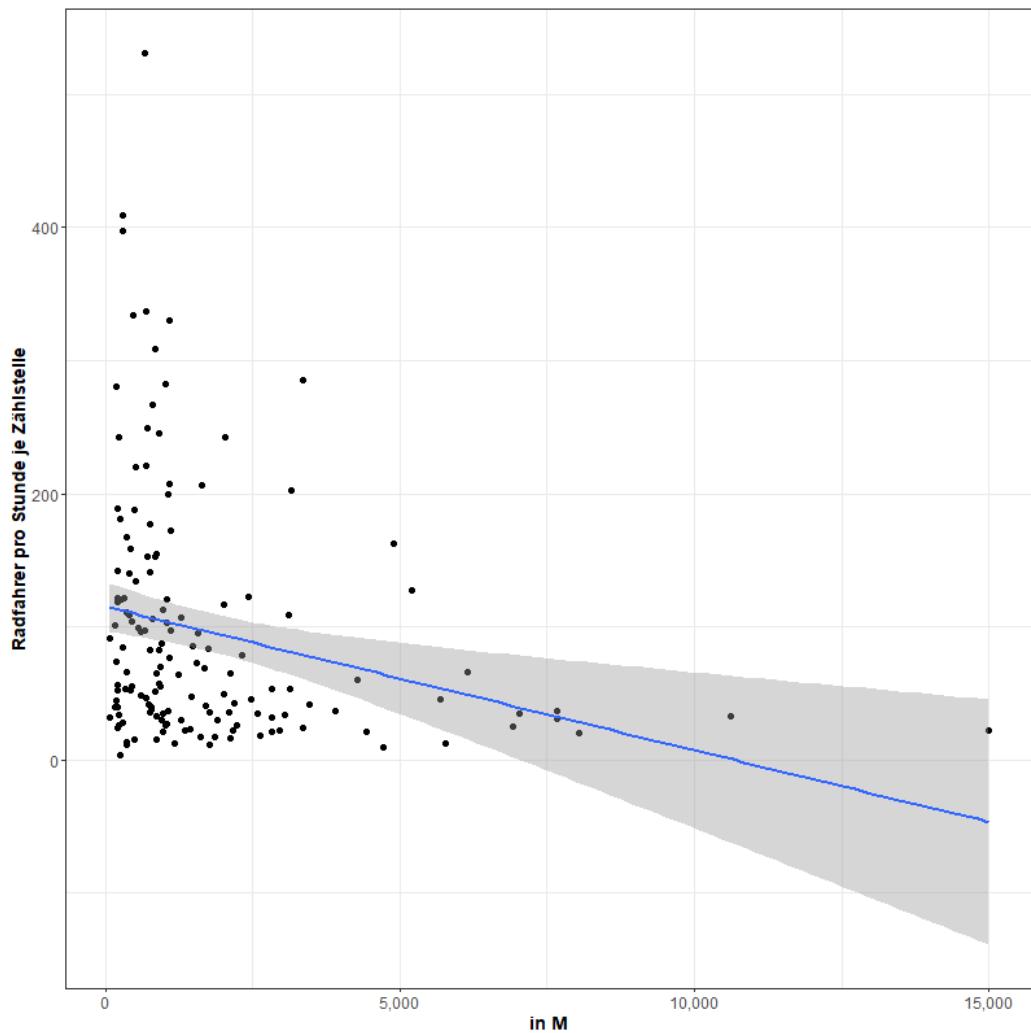


Fig. 7.3: Zusammenhang von Anzahl der Uni-Gebäude in einem 500 M Radius und Radverkehr

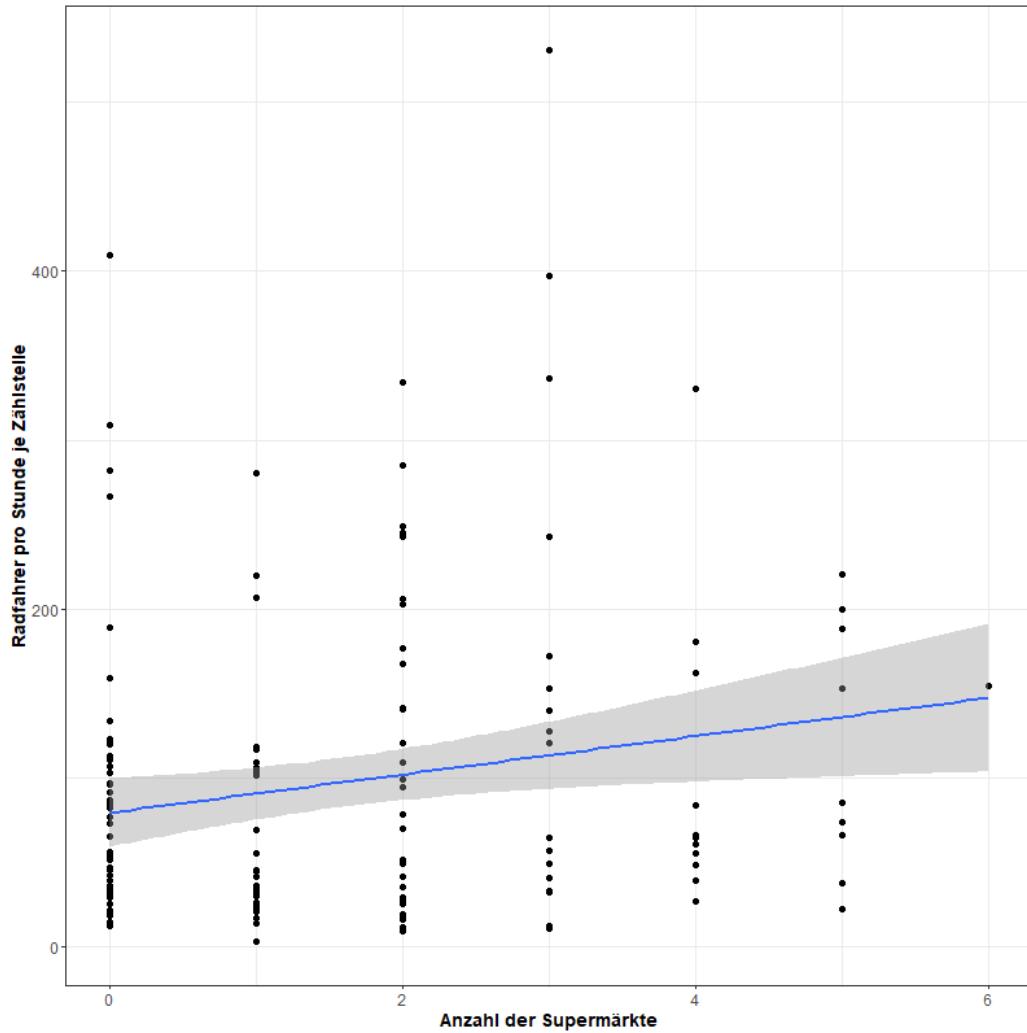


Fig. 7.4: Zusammenhang von Anzahl der Supermärkte in einem 1 km Radius und Radverkehr

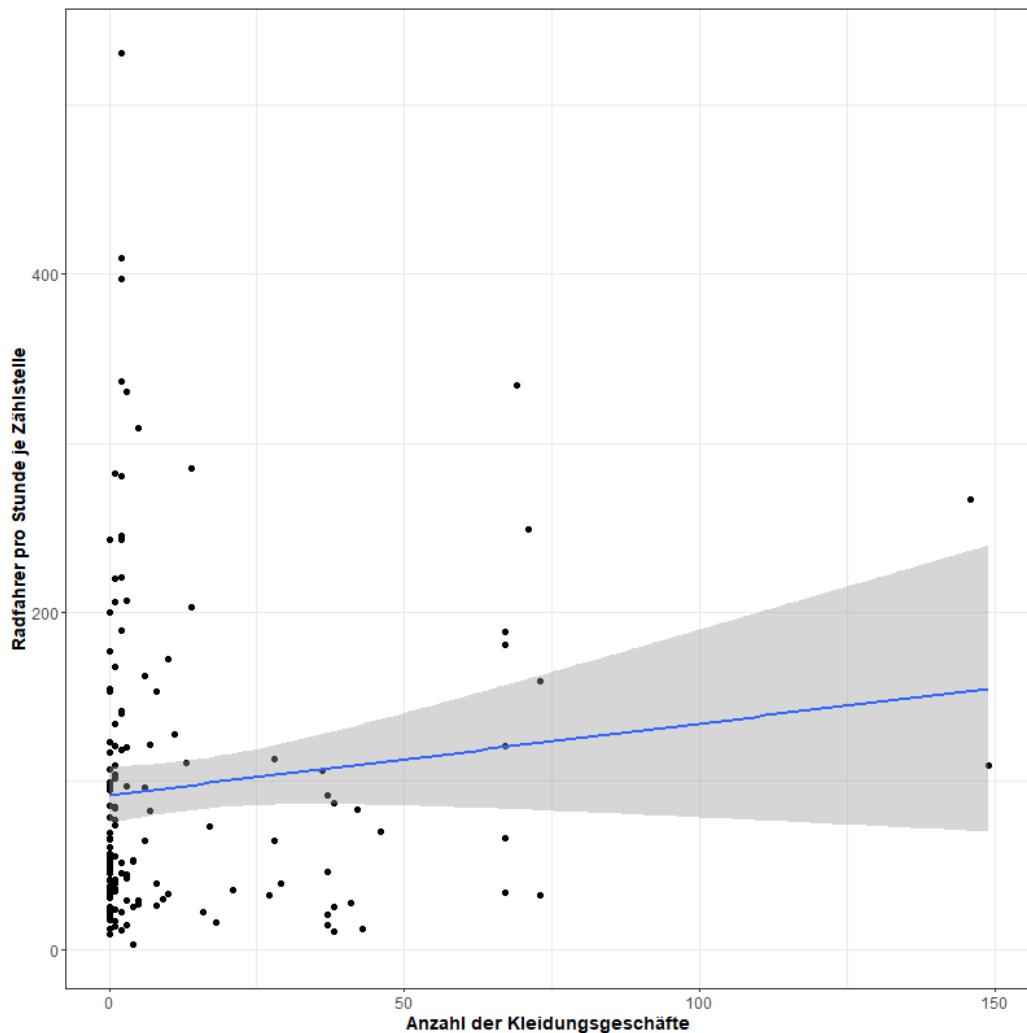


Fig. 7.5: Zusammenhang von Anzahl der Kleidungsgeschäften in einem 2 km Radius und Radverkehr

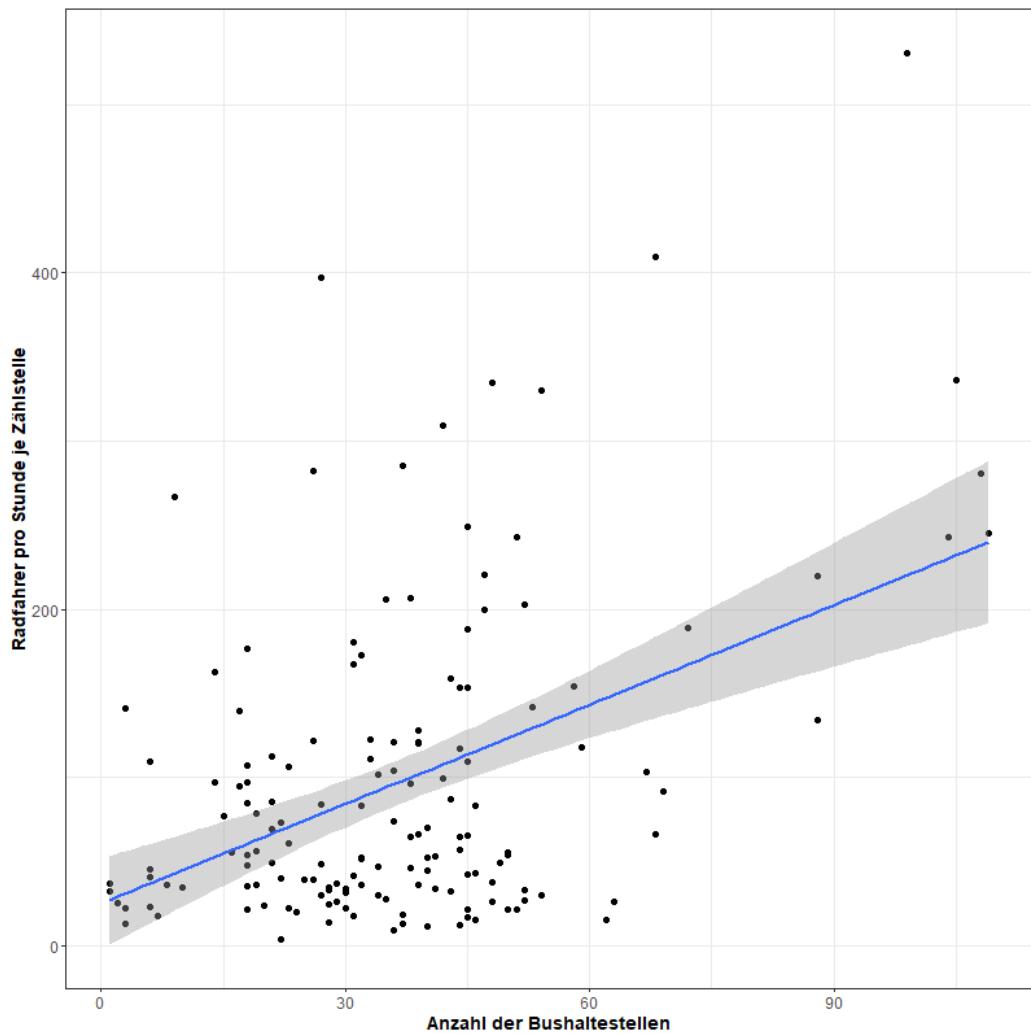


Fig. 7.6: Zusammenhang von Anzahl der Busstationen in einem 1 km Radius und Radverkehr

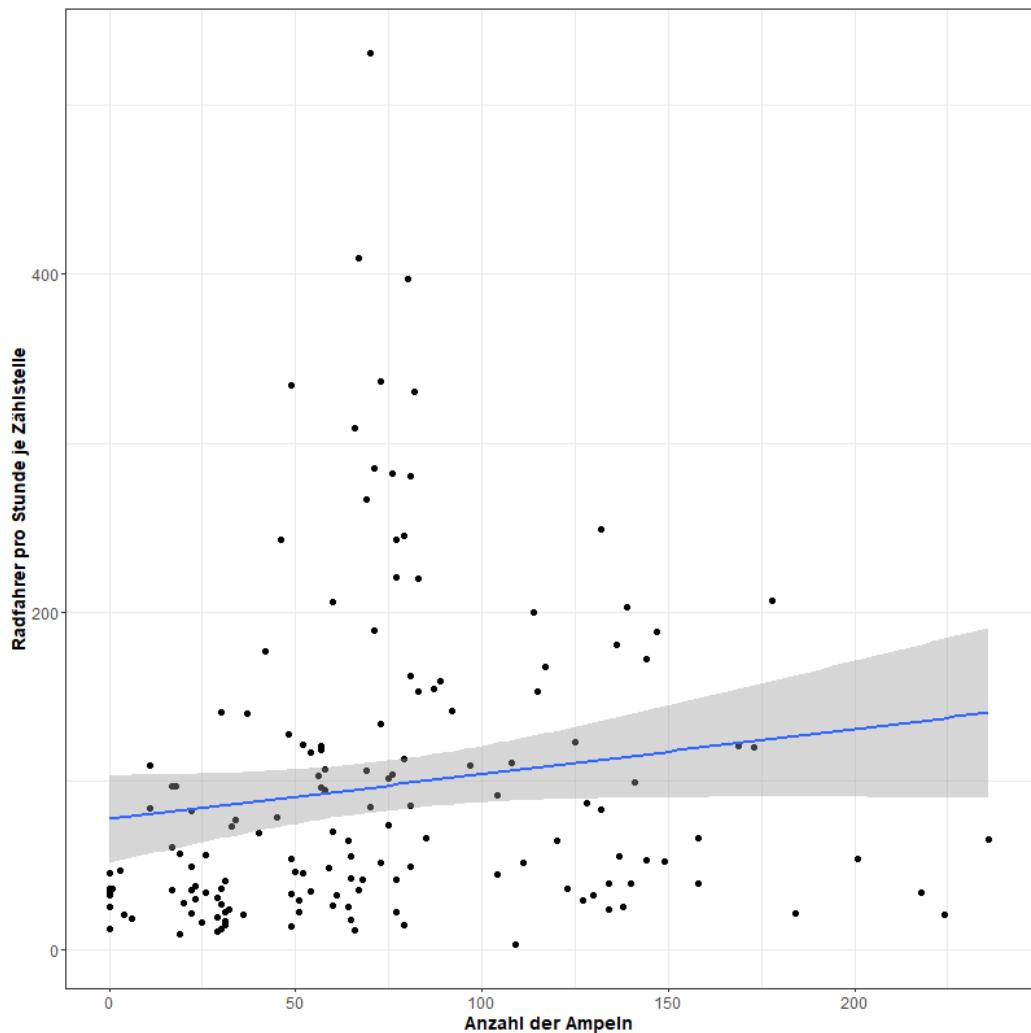


Fig. 7.7: Zusammenhang von Anzahl der Ampeln in einem 1 km Radius und Radverkehr

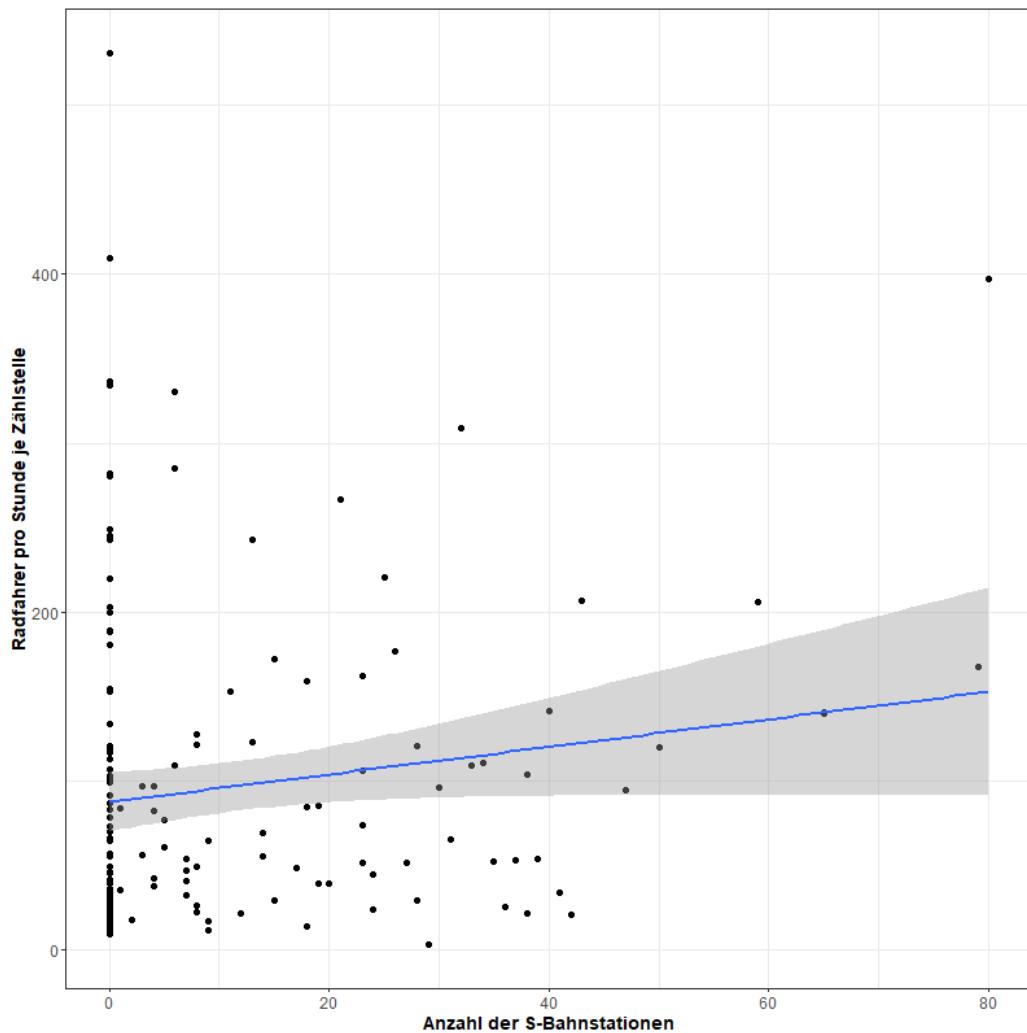


Fig. 7.8: Zusammenhang von Anzahl der Straßenbahnstationen in einem 1 km Radius und Radverkehr

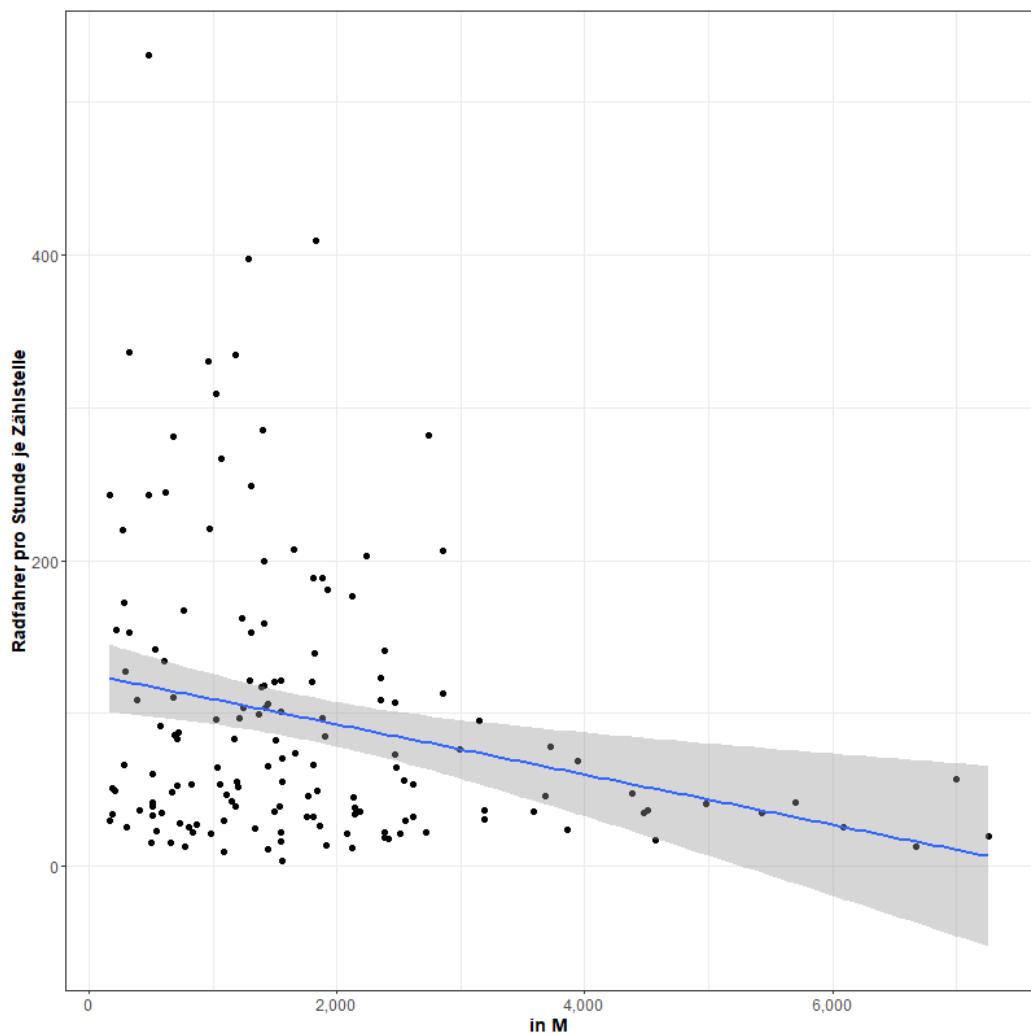


Fig. 7.9: Zusammenhang des nächsten Bahnhofes und dem Radverkehr

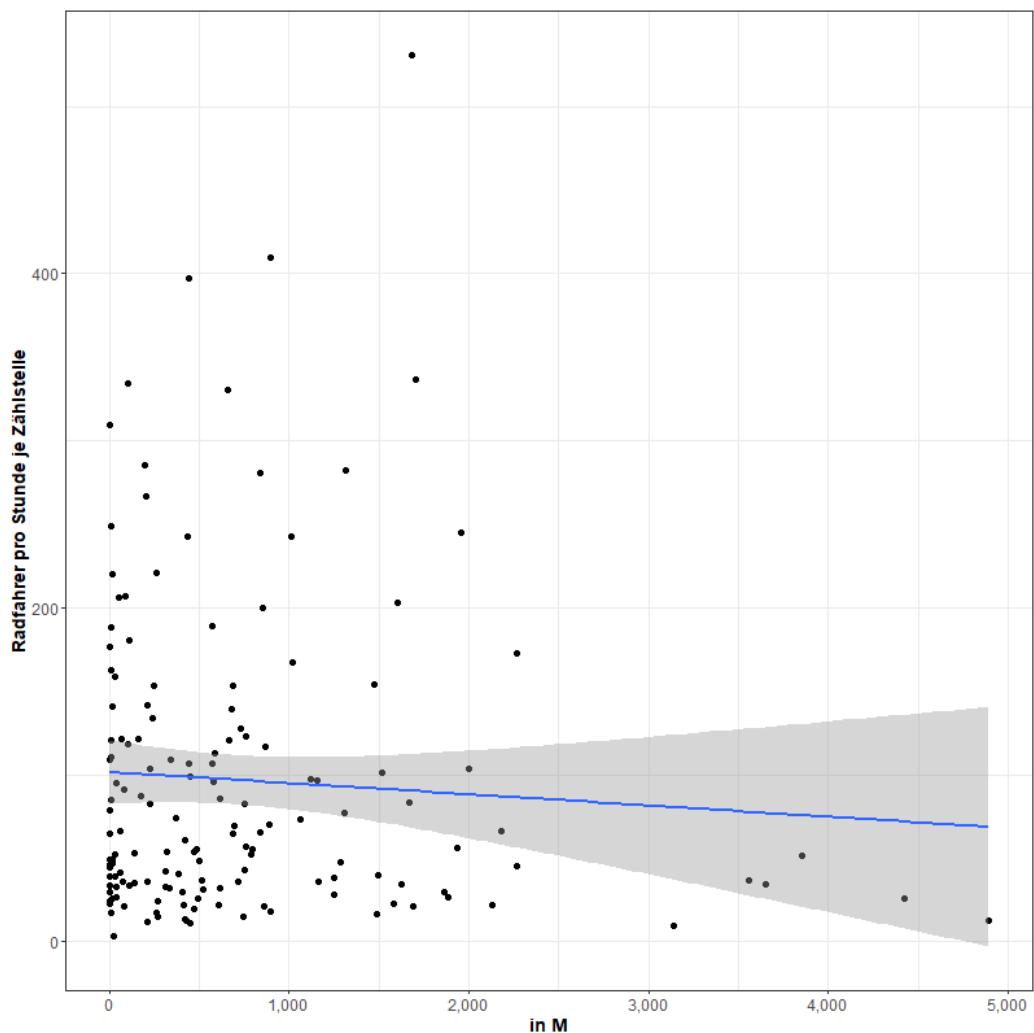


Fig. 7.10: Entfernung zur nächsten Brücke und Radverkehr

## LITERATURVERZEICHNIS

- Alattar, M. A., Cottrill, C. D., and Beecroft, M. (2021). Modelling cyclists' route choice using strava and osmnx: A case study of the city of glasgow. *Transportation Research Interdisciplinary Perspectives*.
- Awad, M. and Khanna, R. (2015). *Support Vector Regression*, pages 67–80. Apress, Berkeley, CA.
- BMDV (2020). Verwaltungsvereinbarung sonderprogramm „stadt und land“. *über die Gewährung von Finanzhilfen des Bundes an die Länder nach Artikel 104b des Grundgesetzes und aufgrund des Haushaltsgesetzes 2020 für Investitionen in den Radverkehr durch das Sonderprogramm „Stadt und Land“*.
- Broach, J., Dill, J., and Gliebe, J. P. (2012). Where do cyclists ride? a route choice model developed with revealed preference gps data. *Transportation Research Part A-policy and Practice*, 46:1730–1740.
- Broucke, S. V., Piña, L. M. V., Do, T. H., and Deligiannis, N. (2019). Brubike: A dataset of bicycle traffic and weather conditions for predicting cycling flow. In *2019 IEEE International Smart Cities Conference (ISC2)*, pages 432–437.
- Carl, K. and Dror, M. (2015). Construction of a topographical road graph for bicycle tour routes. *Sports Technology*.
- Colace, F., De Santo, M., Lombardi, M., Pascale, F., Santaniello, D., and Tucker, A. (2020). A multilevel graph approach for predicting bicycle usage in london area. In Yang, X.-S., Sherratt, S., Dey, N., and Joshi, A., edi-

- tors, *Fourth International Congress on Information and Communication Technology*, pages 353–362, Singapore. Springer Singapore.
- Corporation, M. and Weston, S. (2022). *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*. R package version 1.0.17.
- Destatis (2022a). Ausländer: Kreise, stichtag, geschlecht: Tabelle 12521-0040. *Statistisches Bundesamt*.
- Destatis (2022b). Bevölkerung: Kreise, stichtag, altersgruppen: Tabelle 12411-0017. *Statistisches Bundesamt*.
- Destatis (2022c). Kraftfahrzeugbestand: Kreise, stichtag, kraftfahrzeugarten: Tabelle 46251-0020. *Statistisches Bundesamt*.
- Eisenberger, D. (2015). Pressemitteilung: Zahlen – daten – fakten zum deutschen fahrradmarkt 2015sehr gutes jahr für die deutsche fahrradindustrie. *Zweirad-Industrie-Verband e.V.*
- Gao, C. and Chen, Y. (2022). Using machine learning methods to predict demand for bike sharing. In Stienmetz, J. L., Ferrer-Rosell, B., and Massimo, D., editors, *Information and Communication Technologies in Tourism 2022*, pages 282–296, Cham. Springer International Publishing.
- Goldmann, K. and Wessel, J. (2021). Some people feel the rain, others just get wet: An analysis of regional differences in the effects of weather on cycling. *Research in Transportation Business and Management*, 40:100541. Active Travel and Mobility Management.
- Harvey, F. and Krizek, K. (2007). Commuter bicyclist behavior and facility disruption. *Minnesota Department of Transportation, Research Services Section*.
- Hausman, J., Hall, B. H., and Griliches, Z. (1984). Econometric models for count data with an application to the patents-r and d relationship. *Econometrica*, 52(4):909–938.

- Heinen, E., van Wee, B., and Maat, K. (2010). Commuting by bicycle: An overview of the literature. *Transport Reviews*, 30(1):59–96.
- Hijmans, R. J. (2021). *geosphere: Spherical Trigonometry*. R package version 1.5-14.
- Holmgren, J., Aspegren, S., and Dahlströma, J. (2017). Prediction of bicycle counter data using regression. *Procedia Computer Science*, 113:502–507. The 8th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2017) / The 7th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2017) / Affiliated Workshops.
- Hong J, McArthur DP, S. J. K. C. (2022). Did air pollution continue to affect bike share usage in seoul during the covid-19 pandemic? *J Transp Health*.
- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013a). *An Introduction to Statistical Learning*, chapter Support Vector Machines, pages 337–372. Springer New York, New York, NY.
- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013b). *An Introduction to Statistical Learning*, chapter Tree-Based Methods, pages 303–335. Springer New York, New York, NY.
- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021). *An Introduction to Statistical Learning*, chapter Deep Learning, pages 403–458. Springer New York, New York, NY.
- Kaltenbrunner, A., Meza, R., Grivolla, J., Codina, J., and Banchs, R. (2010). Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. *Pervasive and Mobile Computing*, 6(4):455–466. Human Behavior in Ubiquitous Environments: Modeling of Human Mobility Patterns.
- Kondo, M. C., Morrison, C., Guerra, E., Kaufman, E. J., and Wiebe, D. J. (2018). Where do bike lanes work best? a bayesian spatial model of bicycle lanes and bicycle crashes. *Safety Science*, 103:225–233.

- Kursa, M. B. and Rudnicki, W. R. (2010). Feature selection with the boruta package. *Journal of Statistical Software*, 36(11):1–13.
- Lee, K. and Sener, I. N. (2020). Emerging data for pedestrian and bicycle monitoring: Sources and applications. *Transportation Research Interdisciplinary Perspectives*, 4:100095.
- Li, X., Xu, Y., Chen, Q., Wang, L., Zhang, X., and Shi, W. (2022). Short-term forecast of bicycle usage in bike sharing systems: A spatial-temporal memory network. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):10923–10934.
- Li, Y., Zheng, Y., Zhang, H., and Chen, L. (2015). Traffic prediction in a bike-sharing system. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL ’15, New York, NY, USA. Association for Computing Machinery.
- Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.
- Liu, X., Kounadi, O., and Zurita-Milla, R. (2022). Incorporating spatial autocorrelation in machine learning models using spatial lag and eigenvector spatial filtering features. *ISPRS International Journal of Geo-Information*, 11(4).
- Meng, M., Zhang, J., Wong, Y. D., and Au, P. H. (2016). Effect of weather conditions and weather forecast on cycling travel behavior in singapore. *International Journal of Sustainable Transportation*, 10(9):773–780.
- Menghini, G., Carrasco, N., Schüssler, N., and Axhausen, K. (2010). Route choice of cyclists in zurich. *Transportation Research Part A: Policy and Practice*, 44(9):754–765.
- Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. (2021). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071)*, TU Wien. R package version 1.7-9.

- Microsoft and Weston, S. (2022). *foreach: Provides Foreach Looping Construct*. R package version 1.5.2.
- Mitchell, P. P. (2018). Predicting bike-sharing traffic flow using machine learning.
- Musakwa, W. and Selala, K. M. (2016). Mapping cycling patterns and trends using strava metro data in the city of johannesburg, south africa. *Data in Brief*, 9:898–905.
- Möllers, A., Specht, S., and Wessel, J. (2021). The impact of the covid-19 pandemic and government intervention on active mobility. Technical Report 34. Publication status: Published.
- Nankervis, M. (1999). The effect of weather and climate on bicycle commuting. *Transportation Research Part A: Policy and Practice*, 33(6):417–431.
- Padgham, M., Rudis, B., Lovelace, R., and Salmon, M. (2017). osmdata. *The Journal of Open Source Software*, 2(14).
- Pebesma, E. (2018). Simple Features for R: Standardized Support for Spatial Vector Data. *The R Journal*, 10(1):439–446.
- Pisner, D. A. and Schnyer, D. M. (2020). Chapter 6 - support vector machine. In Mechelli, A. and Vieira, S., editors, *Machine Learning*, pages 101–121. Academic Press.
- Prati, G., Pietrantoni, L., and Fraboni, F. (2017). Using data mining techniques to predict the severity of bicycle crashes. *Accident Analysis and Prevention*, 101:44–54.
- Pritchard, R. (2018). Revealed preference methods for studying bicycle route choice—a systematic review. *International Journal of Environmental Research and Public Health*, 15:470.
- Pucher, J., Dill, J., and Handy, S. (2010). Infrastructure, programs, and policies to increase bicycling: An international review. *Preventive Medicine*, 50:S106–S125.

- Reddy, S., Shilton, K., Denisov, G., Cenizal, C., Estrin, D., and Srivastava, M. (2010). Biketastic: Sensing and mapping for better biking. volume 3, pages 1817–1820.
- Rietveld, P. and Daniel, V. (2004). Determinants of bicycle use: do municipal policies matter? *Transportation Research Part A: Policy and Practice*, 38(7):531–550.
- Romanillos, G., Austwick, M. Z., Ettema, D., and Kruijf, J. D. (2016). Big Data and Cycling. *Transport Reviews*, 36(1):114–133.
- Saha, D., Alluri, P., Gan, A., and Wu, W. (2018). Spatial analysis of macro-level bicycle crashes using the class of conditional autoregressive models. *Accident Analysis and Prevention*, 118:166–177.
- Stock, J. H. and Watson, M. W. (2015a). *Introduction to Econometrics*, chapter Estimation of Dynamic Causal Effects, pages 635–683. PEARSON, 3 edition.
- Stock, J. H. and Watson, M. W. (2015b). *Introduction to Econometrics*, chapter Introduction to Time Series Regression and Forecasting, pages 568–634. PEARSON, 3 edition.
- Thomas, B. and DeRobertis, M. (2013). The safety of urban cycle tracks: A review of the literature. *Accident Analysis and Prevention*, 52:219–227.
- Vandenbulcke, G., Thomas, I., and Int Panis, L. (2014). Predicting cycling accident risk in brussels: A spatial case-control approach. *Accident Analysis and Prevention*, 62:341–357.
- Wessel, J. (2020). Using weather forecasts to forecast whether bikes are used. *Transportation Research Part A: Policy and Practice*, 138:537–559. Publication status: Published.
- Winters, M., Davidson, G., Kao, D., and Teschke, K. (2010). Motivators and deterrents of bicycling: Comparing influences on decisions to ride. *Transportation*, 38:153–168.

- Xu, H., Ying, J., Wu, H., and Lin, F. (2013). Public bicycle traffic flow prediction based on a hybrid model. *Applied Mathematics and Information Sciences*, 7:667–674.
- Zhao, P., Li, S., Li, P., Liu, J., and Long, K. (2018). How does air pollution influence cycling behaviour? evidence from beijing. *Transportation Research Part D: Transport and Environment*, 63:826–838.
- ZIV (2022). Pressemitteilung: „freiheitsmobilität“ bleibt stark nachgefragt. *Der Zweirad-Industrie-Verband e.V.*