Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Thu-Bao Cao

Vorhersage eines Taxifahrpreises in New York City mit maschinellen Lernverfahren

Fakultät Technik und Informatik
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

Thu-Bao Cao

Vorhersage eines Taxifahrpreises in New York City mit maschinellen Lernverfahren

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Bachelor of Science Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Kai von Luck
Zweitgutachter: Prof. Dr. Tim Tiedemann

Eingereicht am: 11.Februar 2020

Thu-Bao Cao

Thema der Arbeit

Vorhersage eines Taxifahrpreises in New York City mit maschinellen Lernverfahren

Stichworte

Machine Learning, KDD, Logistic Regression, Random Forest, Yellow Taxi Dataset, Imbalance Dataset

Kurzzusammenfassung

In dieser Studie werden die Probleme und Methoden zur Steigerung der Datenqualität eines Realdatensatzes New York City Yellow Cab aufgezeigt. Es wird diskutiert, welche externen Daten und neuen Features durch das Feature Engineering für die Vorhersage des Taxifahrpreises in New York City relevant sind. Schließlich werden verschiedene Machine-Learning-Algorithmen und Versionen des Trainingdatensatzes getestet, um deren Auswirkungen auf die Vorhersageleistung gegeneinander zu evaluieren.

Thu-Bao Cao

Title of Thesis

Keywords

Machine Learning, KDD, Logistic Regression, Random Forest, Yellow Taxi Dataset, Imbalance Dataset

Abstract

This study shows the problems and methods for increasing the data quality of a real data set New York City Yellow Cab. It is discussed which external data and new features through feature engineering are relevant for predicting the taxi fare in New York City. Finally, different machine learning algorithms and different versions of the training data set are tested in order to evaluate their effects on the prediction performance against each other.

Inhaltsverzeichnis

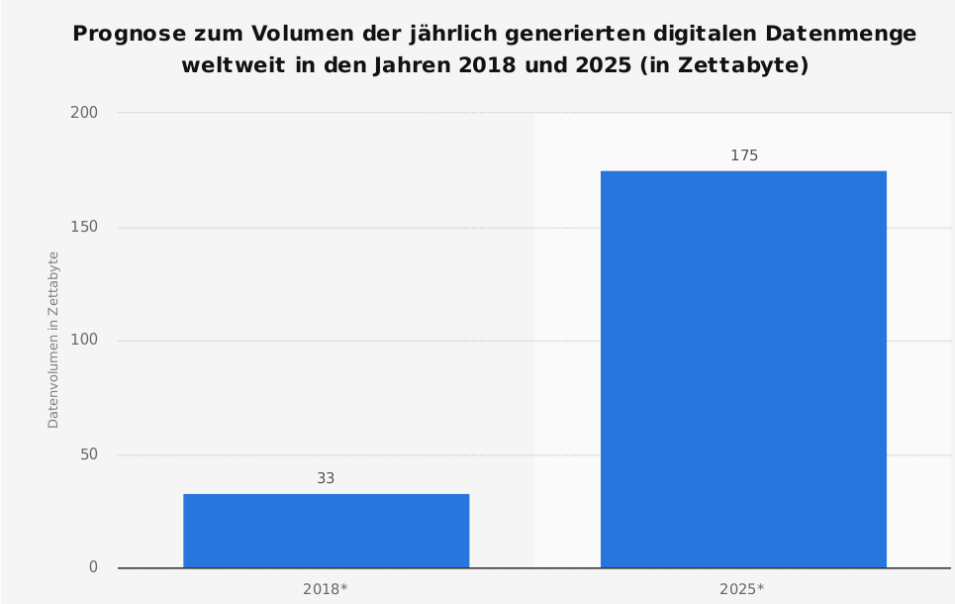
Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

> "Die Information in der Welt verdoppelt sich etwa alle 20 Monate" — Thomas A. Runkler [3]

Im Jahre 2025 wird weltweit mit einer Datenmenge von 175 Zettabyte gerechnet. Dieses Volumen entspricht einem Anstieg von 142 Zettabyte gegenüber dem Jahr 2018. Abbildung 1.1 verdeutlicht die Relationen.



Quelle:[4]

Abbildung 1.1: Prognose zum Volumen der jährlich generierten digitalen Datenmenge weltweit in den Jahren 2018 und 2025

Mit dem Anstieg der Datenmengen und der geringen Kosten der leistungsstarken Hardware wird es für Forschungseinrichtungen und Unternehmen immer attraktiver diese großen Datenmengen zu analysieren. So gab es in den letzten Jahren erheblich Fortschritte im

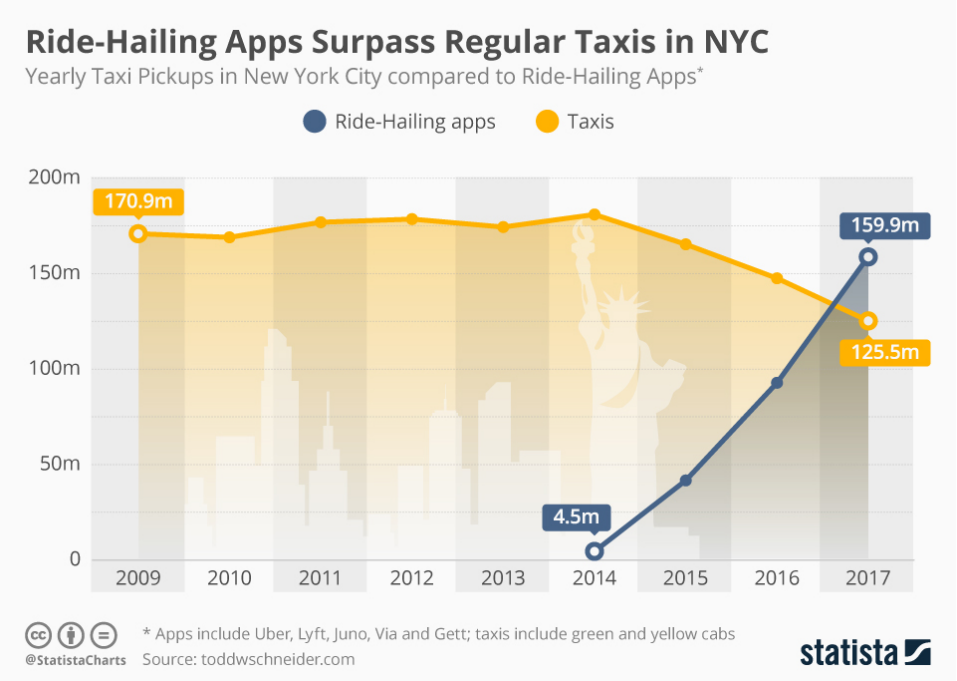
Bereich des Machine Learnings. Das Forschungsfeld bildet eine Schnittmenge aus Statistik, künstlicher Intelligenz und Informatik und ist ebenfalls als prädiktive Analytik oder statistisches Lernen bekannt [5]. Heutzutage wird Machine Learning in vielen kommerziellen Anwendungsgebieten, Forschungsprojekten und Unternehmen eingesetzt. Beispiele sind Medizin, E-Commerce, IT-Security und Predictive Logistic. Hierbei wird ein künstliches System auf Trainingsdaten trainiert, das daraus Wissen extrahiert, um eine gegebene Fragestellung zu beantworten. Dieses Wissen soll den Anforderungen der Gültigkeit, Neuartigkeit, Nützlichkeit und Verständlichkeit genügen, um die Relevanz der angeführten Fragestellung zu repräsentieren.

Dennoch liegt das Hauptproblem eines Machine-Learning-Projekts nicht in der Auswahl des geeigneten Algorithmus oder in der Auswahl der geeigneten Metriken, die zur Bewertung eines Projekts herangezogen werden, sondern eher in den Daten selbst. Denn meistens sind reale Daten ungenau, nicht vollständig oder sie enthalten sogar falsche Angaben. Heutzutage liegt 80 % eines Machine-Learning-Projekts in der Datenvorverarbeitung und in der Auswahl der passenden Features zur gegebenen Problemstellung [6]. Der Prozess der Datenvorverarbeitung und der Auswahl der passenden Features soll anhand des realen Datensatzes der New York City Yellow Cab durchgeführt werden. Im nächsten Abschnitt werden die Probleme der heutigen Taxifahrer in New York City analysiert und wie Machine Learning das Problem beheben könnte. Anschließend werden die Ziele der Arbeit vorgestellt, die mit dem Yellow-Cab-Datensatz unternommen werden [7].

1.1 Problemstellung und Motivation

Über Jahrzehnte war das Taxi weltweit das dominierende Transportmittel in den Großstädten, mit dem gegen einen zu zahlenden Mietpreis Personen befördert werden. Deshalb wurden im Jahre 1930 in den meisten Städten Vorschriften verabschiedet, um die Dominanz der Taxiunternehmen zu regulieren. Mit dieser Regelung sollten die Preise stabilisiert sowie die Sicherheit und Qualität des Taxis gewährleistet werden. Die Taxiunternehmen profitierten bis vor Kurzem von einem mangelnden Wettbewerb auf dem Markt für Miettransportmittel. Im Zuge der Digitalisierung sehen sich Taxiunternehmen mit neuen Wettbewerbern konfrontiert, die Ride-Sharing-Anwendungen nutzen. Ride-Sharing-Unternehmen wie Uber nutzen eine Smartphone App, um Kunden mit Fahrern zu vermitteln, die sie zum Ziel bringen. Der Hauptvorteil von Uber ist, dass sie keine

Fahrzeuge oder Fahrer bereitstellen, sondern nur die Vermittlung zwischen Fahrer und Kunden als Dienstleistung bereitstellen. Da Uber keiner Regulierung unterliegt und Fahrten zu niedrigeren Preisen vorschlagen, haben diese Angebote die New Yorker Taxis im Jahr 2017 bei den Abholungen übertroffen.



Quelle:[8]

Abbildung 1.2: Ride-Hailing Apps vs. Regular Taxis in New York City

Aus Abbildung 1.2 wird ersichtlich, dass Uber als Transportmittel exponentiell an Bedeutung gewonnen hat, wohingegen das Taxi stetig an Bedeutung einbüßt. Die Folge ist, dass die Taxifahrer länger arbeiten müssen, um mit dieser Beschäftigung den Lebensunterhalt sichern zu können, und trotzdem immer noch weniger verdienen als in den Zeiten vor Uber.

Damit die Taxifahrer von den unregulierten Uber-Fahrer nicht in Existenznot geraten, könnte man mithilfe von Machine Learning vorhersagen, zu welcher Zeit und an welchen Orten potenziell mit hohen Einnahmen gerechnet werden kann. Diese Informationen könnten die Suchzeit des Taxifahrers verringern und höhere Einnahmen generieren.

1.2 Fragestellung

Ziel der Arbeit ist es, unter Anwendung des KDD-Prozesses anhand von Realdaten zu überprüfen, ob die Daten für Vorhersagen anwendbar sind. Es sollen ausgewählte Ansätze getestet und miteinander verglichen werden. Der Schwerpunkt dieser Arbeit liegt in der Datenaufbereitung und im Feature Engineering. Diese Arbeit soll als Grundlage für das weitere Vorgehen mit Machine Learning und dem KDD-Prozess zum Yellow-Taxi-Dataset dienen.

1.3 Aufbau der Arbeit

In Kapitel 2 wird das relevante theoretische Wissen ausgearbeitet. Danach folgt die Analyse der vergleichbaren Arbeiten, die zur Vorhersage des Taxifahrpreises bereits veröffentlicht wurden.

In Kapitel 3 werden zunächst die Datensätze vorgestellt, die für die Analyse herangezogen werden. Anschließend geht es um die Frage, welche weiteren Anwendungsszenarien mit den Daten möglich sind. Das Kapitel wird mit der zu beantwortenden Fragen abgeschlossen, im welchen Zustand sich die Daten befinden und welche weiteren Features für die Vorhersage der Taxifahrpreise relevant sein können.

In Kapitel 4 wird der KDD-Prozess praktisch durchgeführt. Hier werden alle Phasen des KDD-Prozesses durchlaufen und mit einer Evaluierung der Ergebnisse abgeschlossen.

In Kapitel 5 werden alle Erkentnisse der Arbeit zusammengefasst und ein Ausblick für mögliche Verbesserungen der Ergebnisse gegeben.

2 Problemanalyse

In diesem Kapitel werden die theoretischen Grundlagen vorgestellt, die für die Vorhersage des Taxifahrpreises in New York City relevant sind. Zu diesem Zweck werden die einzelnen Konzepte und Methoden beschrieben. Anschließend folgt eine Analyse der Studien zum Thema Vorhersage der Taxifahrpreise in New York City.

2.1 Knowledge Discovery in Databases

Mehr als die Hälfte der Weltbevölkerung lebt heutzutage in Städten [9]. Für einen einwandfreien Ablauf müssen die Dienste effektiv, effizient und nachhaltig gestaltet werden. Die Städte haben angefangen, umfangreiche Daten zu publizieren, um zum Beispiel die Stadtdynamik zu analysieren wie die Taxifahrten in New York City [7]. In New York City werden täglich etwa 500.000 Taxifahrten registriert.

Für die Analyse und Verarbeitung dieser umfangreichen Datenmengen muss ein Mechanismus bereitgestellt werden, der einer strukturierten Vorgehensweise folgt. Zu diesem Zweck könnte das KDD-Verfahren nach Fayyad u.a. [10] eingesetzt werden:

> "KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" — Fayyad u.a. [11]

Das KDD ist ein Gesamtprozess, der iterativ und interaktiv verläuft und Wissen aus dem Daten extrahiert. Das Wissen wird mit einem Muster gleichgesetzt, das die Eigenschaften allgemeingültig, nicht trivial, neu, nützlich und verständlich aufweist. Bevor die Phasen des KDD-Prozesses durchlaufen werden, wird in der Initialphase das Domänenwissen und das Ziel der Anwendung spezifiziert. Nachdem das Domänenwissen und das Ziel klar ist, wird der KDD-Prozess in fünf Schritten durchlaufen (siehe Abbildung 2.1).

In der ersten Phase werden die Daten gesammelt, die für die Anwendung als geeignet erscheinen. Hierbei können zusätzlich zu den Basisdatenbestand auch externe Daten für die

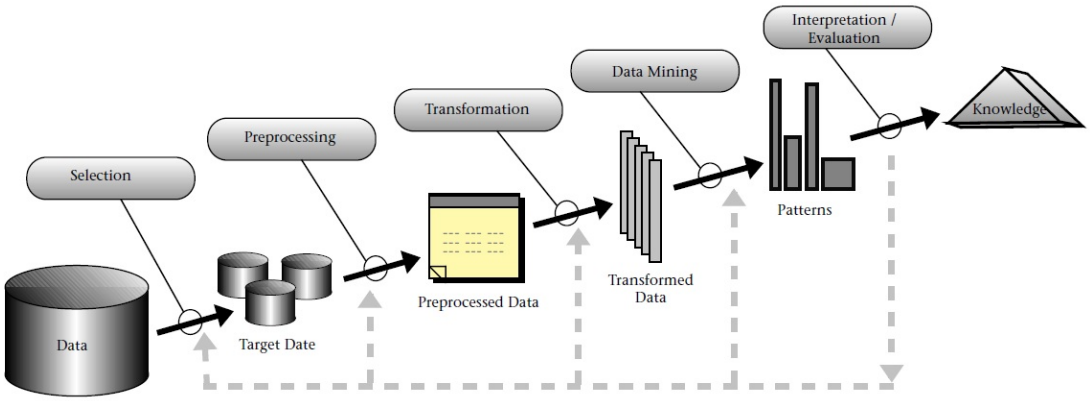


Abbildung 2.1: Der KDD-Prozess nach Fayyad [1]

Analyse herangezogen werden. In der zweiten Phase wird die Qualität der Daten geprüft und gegebenfalls mit passenden Methoden gesäubert. In der dritten Phase werden die vorverarbeiteten Daten in ein Format transformiert, das vom Machine-Learning-Algorithmus verarbeitet werden kann. In der vierten Phase werden die Machine-Learning-Algorithmen ausgewählt und auf die transformierten Daten trainiert. Diesbezüglich ist anzumerken, dass in der vorliegenden Studie anstatt des Data Mining das Machine Learning eingesetzt wird. Ein KDD-Prozess legt den Schwerpunkt auf die Daten, daher kann für die Studie das Machine Learning eingesetzt werden. In der fünften Phase werden die Ergebnisse des Machine-Learning-Algorithmus evaluiert und interpretiert.

In den folgenden Abschnitten werden die Phasen genauer erläutert und es wird auf Probleme während der einzelnen Phasen eingegangen. Die Analyse orientiert sich an Cleve und Lämmel [12].

2.1.1 Datenselektion

In der Phase der Datenselektion werden nach der Spezifizierung des Domänenwissens und des Ziels die relevanten Daten ausgewählt und gegebenfalls bei gegebenen Datenquellen exportiert. Hierfür ist eine Sichtung der Daten erforderlich, die anschließend mit dem Domänenwissen bewertet wird. Falls auch externe Daten anhand des Domänenwissens relevant sein können, können sie für die weitere Analyse herangezogen werden. Die Anreicherung der Datenbasis mit externen Daten ist ein notwendiger Prozess für die Datenselektion, weil die externen Daten im Anwendungsfall mögliche Beziehungen zum

Ziel aufzeigen. Durch die Hinzufügung von anderen Datenquellen können Redundanzen auftreten, die vermieden werden sollten, weil dadurch die Daten höhere Dimensionen aufweisen und eine höhere Komplexität für den Machine-Learning-Algorithmus vorliegen würde. Ziel ist es, einen vereinheitlichten Basisdatenbestand zu erreichen, um die Daten möglichst kompakt zu erhalten und eine hohe Informationsdichte zu ermöglichen. Die Phase der Datenselektion kann die weiteren Phasen negativ beeinflussen, wenn nicht relevante Daten ausgewählt oder umgekehrt die relevanten Daten nicht ausgewählt werden. Deshalb ist die Spezifizierung des Domänenwissens und des Ziels eine wichtige Komponente für die Datenselektion. Weitere Probleme können aufseiten der Technik und des Rechts auftreten. Beispiele hierfür könnten technische Restriktionen wie Kapazitäts- und Datentypenbeschränkungen des Zielsystems sein. Eine mögliche Lösung wäre die Beschränkung auf ein Teildatensatz. Die rechtlichen Probleme des Datenschutzes könnten durch die Anonymisierung der Daten gelöst werden.

2.1.2 Datenvorverarbeitung

In der Phase der Datenvorverarbeitung wird die Qualität des Datenbestands geprüft und durch den Einsatz geeigneter Verfahren verbessert. Laut Cleve u.a. enthalten bis zu 5 % der Feature eines realen Datenbestands falsche Angaben [13]. Die falschen Angaben entstehen häufig aufgrund von technischen und menschlichen Fehlern. Zum Beispiel können Anomalien aufseiten der Messeinheiten auftreten oder Personen falsche Angaben machen. Die Phase der Datenvorverarbeitung ist einer der aufwendigsten Schritte. Für ein zuverlässiges und unverfälschtes Ergebnis ist ein solcher Aufwand notwendig. Zusätzlich zu der Bereinigung von falschen Angaben muss auch eine einheitliche Datenstruktur gewährleistet sein. Falls die Daten zum Beispiel syntaktisch unterschiedlich, aber semantisch gleich sind, müssen sie vereinheitlicht werden. Dieses Problem tritt häufig bei der Zusammenführung des Basisdatenbestands und der externen Daten auf, beispielsweise wenn Feiertags- und Verkaufsdaten eines Betriebs zusammengeführt werden. Hier könnte zum Beispiel das Feature Datetime eine unterschiedliche Syntax haben, aber semantisch identisch sein. Andersherum kann es auch möglich sein, dass sie syntaktisch identisch sind, aber eine andere Zeitzone meinen. Außerdem sollten Redundanzen bei den Daten vermieden werden, weshalb sie verworfen werden können. Für die Identifikation von falschen Angaben ist aus diesem Grund Domänenwissen unerlässlich. Das Ziel der Datenvorverarbeitung besteht darin, eine einheitliche Struktur zu erlangen und die Datenqualität zu verbessern. Meistens weisen die Erkenntnisse, die aus der Datenvorverarbeitung resultie-

ren, auf eine zu verbessernde Datenqualität im System hin. Im weiteren Verlauf werden die Typen der fehlerhaften Daten und die anzuwendende Methodik für eine systematische Behandlung besprochen.

Fehlende Daten

Mit fehlenden Daten sind fehlende Einträge in den Attributen gemeint. Oft ist es einfacher, fehlende Daten zu erkennen, weil kein Domänenwissen vorhanden sein muss. Die fehlenden Daten erkennt man durch den Vergleich der Anzahl der Einträge unter den Attributen. Vielmehr sollte überprüft werden, ob die fehlenden Daten an sich Informationen bereitstellen. Die meisten Machine-Learning-Algorithmen können keine fehlenden Daten verarbeiten, die deshalb behandelt werden müssen. Im Folgenden werden die Methoden zur Bereinigung von fehlenden Daten vorgestellt:

- Eine Art der Behandlung fehlender Daten ist die Ignorierung des gesamten Attributs. Hierbei gehen vermutlich wichtige Informationen verloren, weshalb es nur angewendet wird, wenn keine Einträge im Attribut vorhanden sind.
- Bei nur wenigen fehlenden Daten, ist eine manuelle Korrektur eine Option. Oftmals sind aber mehrere Attribute von diesen Problem betroffen, was zu einem hohen Arbeitsaufwand führt und deshalb nicht anzuraten ist.
- Im Gegensatz zur Löschung des Attributs können auch die fehlenden Daten durch globale Konstanten ersetzt werden. Beispiele hierfür wären „unbekannt“oder „minus unendlich“. Dieses Verfahren ist geeignet, wenn viele Einträge der Attribute betroffen sind oder gewisse Informationen enthalten.
- Eine andere Möglichkeit ist, die fehlenden Daten mit den Durchschnittswert für die statistische Auswertung zu ersetzen. Diese Möglichkeit besteht, wenn es sich um metrische Werte handelt.
- Bei einem kategorischen Wert wird der häufigste Wert herangezogen.
- Attribute können bestimmte Relationen untereinander aufweisen. Fehlende Daten können unter Umständen berechnet werden, wie die fehlende Entfernung zwischen Start- und Zielpunkt.

Falsche Daten

Falsche Daten können aufgrund von mehreren Fehlerquellen entstehen. Sie können aufgrund von Schreibfehlern entstehen oder auf der strukturellen Ebene während der Datenintegration eintreten. Beispiele für falsche Daten wären zum Beispiel ungültige Werte in einem Zahlenbereich, die außerhalb des Definitionsraums liegen oder unplausibel sind. Nach der Identifikation der falschen Daten werden verschiedene Methoden zur Säuberung eingesetzt, die auch für die fehlenden Daten geeignet sind. Auch hier stellt sich die Frage, ob die falschen Daten Informationen beinhalten, die zur Wahl der Behandlungsmethode der falschen Daten führen.

Ausreißer

Mit Ausreißern sind Daten gemeint, die sich häufig von den anderen Datenpunkten erheblich unterscheiden oder die außerhalb des Definitionsraums liegen. Ein Beispiel wäre, wenn ein Datensatz aus 30- bis 50-jährigen Personen besteht, aber auch ein Wert eines 90-Jährigen enthalten ist. In dem Fall könnte es sich um einen Ausreißer handeln, aber auch ein falscher Wert wäre möglich. Die Entscheidung, ob der Wert als Ausreißer behandelt wird oder ob ein falscher Wert vorliegt, ist häufig von dem Anwendungsfall abhängig. Für die Glättung von Ausreißern sind folgende Methoden hilfreich:

- Eine Möglichkeit ist, die Ausreißer zu gruppieren und durch den Mittelwert zu ersetzen. Dieses Vorgehen kann den Einfluss der Ausreißer auf den Machine-Learning-Algorithmus abschwächen.
- Eine andere Möglichkeit ist, die Ausreißer durch eine mathematische Funktion wie die Regression zu glätten. Hier wird der Ausreißer durch die berechnete Funktion ersetzt.
- Im Zweifelsfall lassen sich die Ausreißer durch die Methode der fehlenden Daten behandeln.

Hierbei ist anzumerken, dass die vorgestellten Methoden nur einen Teilbereich der Behandlung von fehlerhaften Daten abdecken und für weitere Informationen auf die zu Beginn erwähnte Literatur hingewiesen wird.

2.1.3 Datentransformation

In der Phase der Datentransformation werden die vorverarbeiteten Daten in ein Format transformiert, das vom eingesetzten Machine-Learning-Algorithmus abhängig ist. Die meisten Machine-Learning-Algorithmen können nur numerische Features oder Labels verarbeiten, weshalb die Methoden der Transformation in numerische Daten hauptsächlich beschrieben werden. Eine weitere Möglichkeit ist die Skalierung der numerischen Werte, weil unterschiedliche Wertebereiche einen Einfluss auf die Gewichtung der einzelnen Features ausüben. Im Folgenden werden die häufig eingesetzten Verfahren der Datentransformation beschrieben. Hierbei ist zu beachten, dass zwischen syntaktischen und semantischen Transformationen unterschieden wird.

Syntaktische Transformation

Die syntaktische Transformation dient dazu, das Format der Daten zu verändern, ohne dabei ihre Aussage zu verfälschen und das Verarbeiten mit dem Machine-Learning-Algorithmus zu ermöglichen.

Kodierung

Machine-Learning-Algorithmen wie die logistische Regression können nicht mit kategorischen Datentypen arbeiten. Deshalb ist die Konvertierung eines kategorischen in einen numerischen Datentyp unerlässlich. Es gibt bei den kategorischen Datentypen zwei Formen. Es handelt sich um ordinale Daten, wenn es kategorische Datentypen mit einer Rangfolge oder einer Ordnungsrelation wie z.B. klein, mittel oder groß sind. Für die Transformation von ordinalen Daten wird häufig das Label Encoding benutzt. Hierbei ist zu beachten, dass die Rangfolge der ordinalen Daten nach der Transformation nicht verfälscht wird. Ein Beispiel hierfür ist:

$$\begin{gathered}\text{klein} \rightarrow 1 \\ \text{mittel} \rightarrow 2 \\ \text{groß} \rightarrow 3\end{gathered}$$

Aber auch eine andere Darstellung ist möglich:

$$\begin{gathered}\text{klein} \rightarrow 1 \\ \text{mittel} \rightarrow 2 \\ \text{groß} \rightarrow 4\end{gathered}$$

Beide Kodierungsformen behalten die Ordnung bei: klein < mittel < groß. Der Unterschied der beiden Kodierungen liegt in den Abständen zwischen mittel und groß. Hierbei könnten durch den unterschiedlichen Abstand der Kodierungen abstandsbasierte Algorithmen wie der k-Nearest-Neighbour jeweils unterschiedliche Ergebnisse erzielen.

Besteht bei den kategorischen Variablen keine Ordnungsrelation, lässt sie sich mithilfe der One-Hot-Kodierung in ein binäres Format umwandeln. Ein Beispiel könnte wie folgt aussehen:

Color
Red
Red
Yellow
Green

$\rightarrow$

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

Tabelle 2.1: One-Hot-Kodierung für den kategorischen Feature Color

In Tabelle 2.1 wird der One-Hot-Mechanismus dargestellt. Dabei wird vom Feature Color für jede Kategorie jeweils ein Feature Red, Yellow und Green in binärer Form dargestellt.

Normalisierung

Umfangreiche Wertebereiche haben bei Machine-Learning-Algorithmen eine höhere Gewichtung. Zur Vermeidung könnten die metrischen Werte auf ein numerisches Intervall wie [0, 1] transformiert werden. Hierbei wird die Normalisierung angewendet, für die der minimale Wert *min(xi)* und maximale Wert *max(xi)* eines metrischen Features benötigt wird. Hierbei lässt sich der normalisierte Werte wie folgt berechnen:

$$x_{new} = \frac{x - min(xi)}{max(xi) - min(xi)} \tag{2.1}$$

Lässt sich mit der Normierung auf das Intervall [0,1] kein zufriedenstellendes Ergebnis erzielen, kann der Wertebereich vergrößert werden, indem x-beliebige Werte gesetzt werden [0,X].

Semantische Transformation

Bei der semantischen Transformation werden die Aussagen der Daten verändert. Ein Beispiel ist die Diskretisierung von metrischen Werten auf einer Ordinalskala, um die

Dimension der Features zu reduzieren und den Machine-Learning-Algorithmus zu verallgemeinern oder den Prozess zu beschleunigen.

Binning

Binning ist eine Methode, um kontinuierliche in kategorische Werte umzuwandeln, indem Werte in eine vordefinierte Anzahl von Bins gruppiert werden. Der kontinuierliche Wert wird durch einen kategorischen Wert ersetzt, der den Bin beschreibt. Durch die Gruppierung der metrischen Werte in Bins wird der Einfluss der Ausreißer auf den Algorithmus geringer und die Auswirkungen auf das Machine-Learning-Modell werden abgeschwächt.

2.1.4 Data Mining/Machine Learning

In der Phase des Machine Learnings wird der ausgewählte Algorithmus auf den transformierten Datensatz trainiert. Üblicherweise findet das Data Mining der transformierten Daten statt, um nach Mustern, Trends oder Zusammenhängen zu suchen.

> "Machine learning is the training of a model from data that generalizes a decision against a performance measure." [14]

Der Machine-Learning-Algorithmus muss lernen, einen Datensatz zu verallgemeinern, anstatt ihn einfach auswendig zu lernen. Machine Learning dient der automatisierten Entscheidungsfindung anhand von Algorithmen und als Methodik für komplexe Probleme. Machine Learning kann grob in drei Kategorien eingeteilt werden:

- überwachtes Lernen (supervised learning)
- unüberwachtes Lernen (unsupervised learning)
- bestärkendes Lernen (reinforcement learning).

Beim überwachten Lernen sind die Labels für die gegebenen Features bekannt. Der Machine-Learning-Algorithmus wird auf die gegebenen Features mit den bekannten Labels trainiert und in der Testphase mithilfe der gegebenen Labels evaluiert.

Beim unüberwachten Lernen sind die Labels für die gegebenen Features unbekannt, indem der Machine-Learning-Algorithmus für die Mustersuche angewendet wird.

Beim bestärkenden Lernen ist das Ziel, dass der Agent selbstständig eine Strategie entwickelt, um seine Belohnung zu maximieren. Hierbei erhält er zu bestimmten Zeiten eine Belohnung oder Bestrafung für seine Aktionen.

Für die vorliegende Studie ist lediglich das überwachte Lernen relevant (vgl. Abschnitt 2.2).

2.1.5 Interpretation/Evaluation

In der Phase der Interpretation und Evaluation des KDD-Prozesses werden die identifizierten Muster evaluiert und interpretiert. Die Muster sollen nach folgenden Kriterien bewertet werden:

- Die Gültigkeit ist ein Maß dafür, wie warscheinlich es ist, dass die gefundenen Muster auch für neue Daten gültig sind.
- Die Neuartigkeit ist ein Maß dafür, ob die gefundenen Muster die bisherigen Kenntnisse ergänzen oder ihnen widersprechen.
- Die Nützlichkeit beschreibt die Anwendbarkeit des gefundenen Musters.
- Die Verständlichkeit beschreibt, wie gut die gefundenen Muster vom Anwender verstanden werden können.

In dieser Arbeit werden die Ergebnisse der Machine-Learning-Algorithmen (Abschnitt 2.3) mit den Aufstellungen des Basisdatenbestands evaluiert. Für die Evaluation der Ergebnisse wurden die Metriken Genauigkeit (precision), die Sensitivität (recall) und das F-Maß genutzt.

Die Formel für die Genauigkeit wird in Gleichung 2.2 definiert:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{2.2}$$

Hierbei steht True Positive für die korrekt vorhergesagte positive Klasse und False Positive für die inkorrekte Vorhersage der positiven Klasse.

Die Formel für die Sensitivität wird in Gleichung 2.3 definiert:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2.3}$$

False Negative steht für die inkorrekt vorhergesagte negative Klasse.

Die Genauigkeit beschreibt die Korrekheit der Vorhersage der positiven Labels. Die Sensitivität beschreibt die Erfassung der positiven Labels von der positiven Klasse. Das F-Maß entspricht der Kombination dieser beiden Metriken und wird in Gleichung 2.4 definiert:

$$F\text{-}Maß = 2X\frac{precisionXRecall}{precision + Recall} \tag{2.4}$$

Sind nach der Evaluierung die Ergebnisse nicht erfolgreich, wird der KDD-Prozess wie beschrieben wiederholt.

2.2 Überwachtes Lernen

Beim überwachten Lernen sind die Labels für den Fall Y und die dazugehörigen bekannten Features X gegeben. Der Machine-Learning-Algorithmus erlernt eine Zuordnungsfunktion mithilfe der gegebenen Features zum Label, wie in Gleichung 2.5 dargestellt wird:

$$Y = f(X) \tag{2.5}$$

Der Machine-Learning-Algorithmus soll die Funktion f(X) so gut wie möglich anpassen, damit sie für neue unbekannte Features Vorhersagen treffen kann. Es wird als überwachtes Lernen bezeichnet, weil der Machine-Learning-Algorithmus, der anhand der Trainingsdaten lernt, wie ein Lehrer betrachtet wird, der den Lernprozess überwacht. Nachdem die Lernphase des Algorithmus beendet ist, wird das nun trainierte Modell auf die Testdaten ohne die angegebenen Labels getestet und die Ergebnisse der Vorhersagen mit den vorgegebenen Labels verglichen, um die Leistung zu evaluieren.

2.2.1 Klassifikation

Bei der Klassifikation wird zu den gegebenen Beobachtungen die Klassenbezeichnung vorhergesagt. Der Klassifikator wird für die gelabelten Datenpunkte trainiert, um das erlernte Wissen für ungelabelte Datenpunkte anzuwenden. Die Zuordnungsfunkion in Gleichung 2.5 sagt die Klassenbezeichnung für die ungelabelten Datenpunkte voraus. Die Klassifikation kann mit binärer Klassifikation erfolgen, indem für die Unterscheidung zwei Klassen genau definiert oder auch als Klassifikator für mehrere Klassen unterteilt werden. Meistens werden nicht die Klassen direkt vorhergesagt, sondern es wird eine Warscheinlichkeit der Zugehörigkeit zu den Klassen angegeben, wobei die Klasse mit der höchsten Warscheinlichkeit als Vorhersage ausgewählt wird. Ein Beispiel hierfür ist der Iris-Datensatz [15], mit dem anhand der Features wie die Länge und Breite eines Kelchblatts und die Länge und Breite des Blütenblatts versucht wird, die Klassen Iris Setosa, Iris Virginica und Iris Versicolor vorherzusagen. Es konnten bei einer Auswertung die Warscheinlichkeiten 0.2 für Iris Setosa, 0.1 für Iris Virginica und 0.7 für Iris Versicolor vorhergesagt werden. Hier wird die Klasse Iris Versicolor ausgewählt, weil sie die höchste Warscheinlichkeit hat. Neben den in Abschnitt 2.1.5 vorgestellten Metriken ist die Klassengenauigkeit die am häufigsten eingesetzte Metrik zur Abschätzung der Qualität des Vorhersagemodells. Es ist aber zu beachten, dass diese Metrik eher weniger für nicht balancierte Datensätze geeignet ist, weil kaum Erkenntnisse daraus abzuleiten sind, ob das Modell einen Lernprozess durchlaufen hat.

In dieser Arbeit wird die Klassifikation als Methode verwendet.

2.2.2 Regression

Neben der Klassifikation beim überwachten Lernen bildet die Regression eine weitere Kategorie. Sie ähnelt der Klassifikation mit der Ausnahme, dass anstatt der Klassenbezeichnung eine kontinuierliche Größe ausgegeben wird. Auch die Methoden zur Evaluierung des Modells sind anders als bei Klassifikationen. Anstatt der Klassengenauigkeit werden zum Beispiel die mittleren quadratischen Fehler angewendet, der sogenannte Root mean squared error (RMSE).

2.3 Eingesetzte Methoden und Verfahren

In diesem Abschnitt werden die Algorithmen und ihre Besonderheiten vorgestellt, die für diese Studie relevant sind.

2.3.1 Logistische Regression

Die logistische Regression ist ein Klassifikationsverfahren, das anders als die lineare Regression anstatt eines kontinuierlichen Werts eine Klasse vorhersagt. Dieser Zusammenhang lässt sich zum Beispiel an der Frage festmachen, ob es sich bei einer eingehenden E-Mail um eine Spam- oder Nicht-Spam-Mail handelt. Die logistische Regression transformiert die Ausgabe in eine Warscheinlichkeit mithilfe der logistischen Sigmoid-Funktion, die in Abbildung 2.2 dargestellt wird.

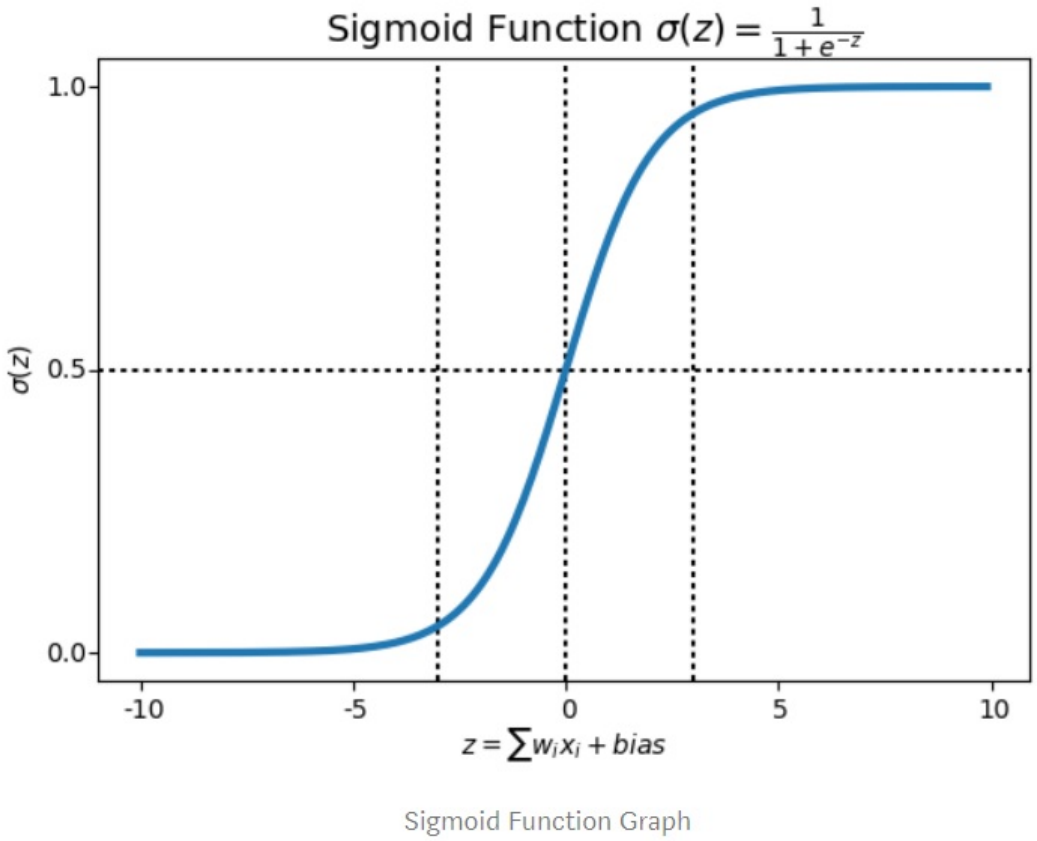


Abbildung 2.2: Die Sigmoid-Funkion [2]

Dabei steht z für die gewichtete Summe der Eingabemerkmale X_i, wobei die Gewichte als W_i angegeben sind. Nachdem die Funktion z berechnet wurde, wird sie der Sigmoid-Funktion übergeben, die eine Warscheinlichkeit zwischen 0 und 1 angibt. Je nach gewähltem Threshold, das in der Abbildung 2.2 bei 0.5 liegt, wird eine Klasse vorhergesagt. Wenn z.B. für Warscheinlichkeiten unter 0.5 die Klasse Katze und für Warscheinlichkeiten über 0.5 die Klasse Hund definiert ist, wird bei einer Warscheinlichkeit von 0.7 die Klasse Hund vorhergesagt.

2.3.2 Random Forest

Der Random Forest ist ein Klassifikationsverfahren, das aus vielen Entscheidungsbäumen besteht, wobei die Bäume sich voneinander unterscheiden. Der Random Forest besteht aus einem Ensemble von Entscheidungsbäumen. Ensembles sind Methoden, die mehrere schwache Machine-Learning-Algorithmen trainieren, um ein mächtigeres Modell zu erschaffen [16]. Ein wesentlicher Nachteil bei Entscheidungsbäumen ist das Auswendiglernen des Trainigsdatensatzes. Dieses Problem wird beim Random Forest dadurch umgangen, dass mehrere Entscheidungsbäume erstellt werden, die unterschiedlich overfitten und am Ende mit den Mittelwert des Overfittings reduziert werden. Zudem ist die Generalisierungsfähigkeit des Random Forest häufig besser, weil durch die Zufälligkeit die Varianz verringert wird. Im Folgenden wird der Algorithmus in vier Schritten erklärt:

1. Es werden zunächst zufällig Bootstrap samples der Größe n aus dem Trainigsdatensatz entnommen. Hierbei wird n-mal zufällig ausgewählt, sodass der dadurch neu enstandene Datensatz so groß ist wie der ursprüngliche. Aufgrund der zufälligen Auswahl der Daten können manche Daten mehrmals und andere eher seltener vorkommen.

2. Nun wird mithilfe der durch Bootstrap Samples entstandene Datensatz ein Entscheidungsbaum erstellt. Hierbei werden bei jedem Knoten folgende Schritte ausgeführt:

   2.1 Wähle d-zufällige Features. d beschreibt eine Teilmenge der gegebenen Features.

   2.2 Teile den Knoten anhand des Features. Die beste Teilung wird anhand des eingesetzten Kriteriums gemessen.

3. Schritt 1 und Schritt 2 werden k-mal wiederholt.

4. Die Ergebnisse der einzelnen Entscheidungsbäume werden zusammgengefasst, indem der Mehrheitsentscheidung entscheidet, welche Klasse genommen wird.

Anzumerken ist, dass die Anzahl der Entscheidungsbäume frei wählbar ist. Für einen tieferen Einblick wird die Literatur von Zhang [16] empfohlen.

2.4 Vergleichbare Arbeiten

Für die Umsetzung eines Vorhersagemodells der Taxifahrpreise in New York City wurden in dieser Arbeit die relevanten Methoden analysiert.

Der Großteil der Arbeiten über die Vorhersage des Taxifahrpreises in New York City sieht die Aufgabe als ein Regressionsproblem an. Zum Beispiel wurde in der Arbeit von Antoniades u.a. [17] mihilfe der Regressionsmodelle wie Lineare Regression, Lasso und Random Forest der Taxifahrpreis als kontinuierliche Größe vorhergesagt. In der Arbeit von Cordor [18] wurde zusätzlich zu den davor genannten Modellen das Gradient Boosting eingesetzt. In der Arbeit von Ojha u.a. [19] wurde noch zusätzlich die einfache KNN-Regression benutzt, um sie mit der Baseline-lineare-Regression zu vergleichen. Zusätzlich wurden die Vorhersagen anhand der bekannten Fahrstrecke als Feature übergeben. Obwohl die genannten Arbeiten einen anderen Use Case und die Fragestellung als Regressionsproblem aufgefasst haben, konnten die Erkenntnisse zur Datenexploration und das Feature Engineerings zur Arbeit beitragen.

In der Arbeit von Upadhyay u.a. [20] wurde mithilfe der Modelle Deep Neural Network und Stacking Classifier die Taxipreisklasse vorhergesagt. Es wurden die Klassen low/normal/medium/high/very high definiert, die auf der Ordinalskala von 1 bis 5 angegeben wurden. Der Basisdatenbestand enhält Angaben zur Startzeit und zum Startort anhand der Längen- und Breitengrade. Hierbei haben Upadhyay u.a. wichtige Beiträge zum Feature Engineering geleistet. Zusätzlich zur Startzeit und zum Startort wurden die Distanzen zum Airport, Stadtzentrum und Touristenplatz sowie Angaben zum Urlaubstag und die Diskretisierung der Zeit in morgens, nachmittags und nachts und die Angaben zum Wochenende als neue Features hinzugefügt.

In der Arbeit von Jolly [21] wurde mithilfe der Modelle Random Forest, Fully Connected Neural Network und das Long Short-Term Memory für die Vorhersage des Taxipreisklasse benutzt. Hierbei wurden die gleichen Eingabemerkmale herangezogen, jedoch mit dem Zusatz, dass mithilfe der k-Means-Clustering-Methode die Angaben zu dem Taxi-Zones verfeinert wurden.

Einige dieser Methodiken können zu einer erfolgreichen Umsetzung des Vorhersagemodells beitragen.

3 Datenanalyse

Nachdem das theoretische Wissen aufgearbeitet und wissenschaftliche Arbeiten analysiert wurden, werden die Datensätze vorgestellt, die für die Vorhersage der Taxifahrpreise in New York City herangezogen werden. Anschließend werden andere Vorhersagen beschrieben, die mit der Datenbasis möglich sind. Für einen ersten Eindruck über die Datenbasis wird die Datenqualität, die explorative Datenanalyse und das Feature Engineering analysiert.

3.1 Vorstellung der Datensätze

Im Folgenden werden die Datensätze beschrieben, die für die Vorhersage des Taxifahrpreises in New York City verwendet wurden. Zusätzlich werden weitere Datensätze vorgestellt, die für die Analyse in Erwägung gezogen wurden, aber aufgrund fehlender Daten in die weitere Analyse nicht einbezogen werden konnten.

3.1.1 NYC Yellow Taxi Dataset

Die New Yorker Taxi- und Limousinenkommission (TLC) [22] wurde 1971 gegründet und ist für die Lizenzierung und Regulierung der New Yorker Medaillontaxis (Yellow Cabs), Borotaxis (Green Cabs), Liverytaxis (For-Hire Vehicles), Pendlerwagen (Commuter Vans) und Paratransit-Fahrzeuge (Paratransit Vehicles) verantwortlich. Die TLC sammelt Fahrinformationen zu jeder Taxi- und Mietwagenfahrt, die von den lizenzierten Fahrern und Fahrzeugen durgeführt wird. Die Daten zu den Taxifahrten werden von den Technologiedienstleistern (TSP) bereitgestellt, die elektronische Messungen in jedem Fahrzeug vornehmen.

So haben sich im Laufe der Zeit über 2 Milliarden Reisedaten für die Yellow Cabs in den Jahren von 2009 bis 2018 angesammelt. Die Daten sind auf der Trips-Record-Seite

des TLC [7] aufrufbar und werden als CSV-Datei pro Monat für jedes Jahr hinterlegt. Das Datenvolumen für einen Monat umfasst zwischen 1,7 und 2,8 Gigabyte. Ich habe mich für einen Zeitraum von Januar 2009 bis Juni 2016 der Daten entschieden, weil die Koordinaten nach Juni 2016 nicht mehr als Längen- und Breitengrad vorliegen, sondern mithilfe Location IDs, die einer größeren Fläche entsprechen, ersetzt worden sind.

In Abbildung 3.1 ist ein Ausschnitt aus der Datentabelle Yellow Cab 2009 für den Monat Januar dargestellt, wo jede Zeile für eine durchgeführte Fahrt steht, die von einem TLC-lizenzierten Fahrzeug durchgeführt wurde.

	vendor_name	Trip_Pickup_DateTime	Trip_Dropoff_DateTime	Passenger_Count	Trip_Distance	Start_Lon	Start_Lat	Rate_Code	store_and_forward	End_Lon	End_Lat	Payment_Type	Fare_Amt	surcharge	mta_tax	Tip_Amt	Tolls_Amt	Total_Amt
0	VTS	2009-01-04 02:52:00	2009-01-04 03:02:00	1	2.63	-73.991959	40.721565	NaN	NaN	-73.993805	40.695923	CASH	8.898438	0.5	NaN	0.000000	0.0	9.398438
1	VTS	2009-01-04 03:31:00	2009-01-04 03:38:00	3	4.55	-73.982101	40.736290	NaN	NaN	-73.955849	40.768028	Credit	12.101562	0.5	NaN	2.000000	0.0	14.601562
2	VTS	2009-01-03 15:43:00	2009-01-03 15:57:00	5	10.35	-74.002586	40.739746	NaN	NaN	-73.869980	40.770226	Credit	23.703125	0.0	NaN	4.738281	0.0	28.437500
3	DDS	2009-01-01 20:52:58	2009-01-01 21:14:00	1	5.00	-73.974266	40.790955	NaN	NaN	-73.996559	40.731850	CREDIT	14.898438	0.5	NaN	3.050781	0.0	18.453125
4	DDS	2009-01-24 16:18:23	2009-01-24 16:24:56	1	0.40	-74.001579	40.719383	NaN	NaN	-74.008377	40.720348	CASH	3.699219	0.0	NaN	0.000000	0.0	3.699219
5	DDS	2009-01-16 22:35:59	2009-01-16 22:43:35	2	1.20	-73.989807	40.735004	NaN	NaN	-73.985023	40.724495	CASH	6.101562	0.5	NaN	0.000000	0.0	6.601562
6	DDS	2009-01-21 08:55:57	2009-01-21 09:05:42	1	0.40	-73.984047	40.743546	NaN	NaN	-73.980263	40.748924	CREDIT	5.699219	0.0	NaN	1.000000	0.0	6.699219
7	VTS	2009-01-04 04:31:00	2009-01-04 04:36:00	1	1.72	-73.992638	40.748363	NaN	NaN	-73.995583	40.728306	CASH	6.101562	0.5	NaN	0.000000	0.0	6.601562
8	CMT	2009-01-05 16:29:02	2009-01-05 16:40:21	1	1.60	-73.969688	40.749245	NaN	NaN	-73.990410	40.751083	Credit	8.703125	0.0	NaN	1.299805	0.0	10.000000
9	CMT	2009-01-05 18:53:13	2009-01-05 18:57:45	1	0.70	-73.955170	40.783043	NaN	NaN	-73.958595	40.774822	Cash	5.898438	0.0	NaN	0.000000	0.0	5.898438
10	CMT	2009-01-05 08:15:38	2009-01-05 08:16:44	1	0.30	-73.986824	40.750893	NaN	NaN	-73.984116	40.751438	Cash	2.900391	0.0	NaN	0.000000	0.0	2.900391
11	CMT	2009-01-05 06:21:43	2009-01-05 06:28:41	1	2.30	-74.006104	40.748432	NaN	NaN	-73.978439	40.762482	Cash	7.699219	0.0	NaN	0.000000	0.0	7.699219
12	DDS	2009-01-20 13:44:02	2009-01-20 13:52:43	2	2.10	-73.983337	40.744781	NaN	NaN	-73.981163	40.720837	CASH	7.300781	0.0	NaN	0.000000	0.0	7.300781
13	CMT	2009-01-05 16:19:53	2009-01-05 16:26:48	2	1.20	-73.973511	40.760281	NaN	NaN	-73.962936	40.775120	Cash	6.699219	0.0	NaN	0.000000	0.0	6.699219
14	CMT	2009-01-05 17:22:16	2009-01-05 17:27:25	1	0.80	-73.984428	40.757481	NaN	NaN	-73.982521	40.767166	Cash	5.898438	0.0	NaN	0.000000	0.0	5.898438
15	CMT	2009-01-05 16:02:52	2009-01-05 16:18:43	1	4.50	-73.991066	40.727654	NaN	NaN	-73.945770	40.777649	Cash	13.898438	0.0	NaN	0.000000	0.0	13.898438
16	CMT	2009-01-05 12:15:06	2009-01-05 12:27:58	1	1.70	-74.001678	40.747299	NaN	NaN	-73.978958	40.750393	Cash	8.500000	0.0	NaN	0.000000	0.0	8.500000
17	CMT	2009-01-05 07:49:57	2009-01-05 07:54:11	1	1.00	-73.982460	40.731476	NaN	NaN	-73.973007	40.743385	Credit	4.500000	0.0	NaN	1.000000	0.0	5.500000
18	DDS	2009-01-23 23:57:34	2009-01-24 00:12:40	2	5.00	-73.992554	40.724476	NaN	NaN	-73.953064	40.777554	CREDIT	13.296875	0.5	NaN	3.449219	0.0	17.250000
19	CMT	2009-01-05 10:23:13	2009-01-05 10:33:56	1	1.30	-73.990089	40.759026	NaN	NaN	-73.983971	40.746937	Cash	7.300781	0.0	NaN	0.000000	0.0	7.300781

Abbildung 3.1: Ausschnitt aus der Datentabelle Yellow Cab 2009 für den Monat Januar

Aus Abbildung 3.1 ist zu entnehmen, dass jede Zeile eine Fahrt detailliert beschreibt. Hierzu zählen Daten zu Datum und Uhrzeit der Abholung, der Zustellung, Anzahl der Fahrgäste, Distanz der Fahrtrecke, Abholort nach Längen- und Breitengrad, Zielort nach Längen- und Breitengrad usw. Es sind mit dem Umfang der Daten auch andere Vorhersagen möglich (vgl. Abschnitt 3.2).

Die Beschreibung der Attribute aus der Datentabelle Yellow Cab und die Angaben der Datentypen befinden sich im Anhang dieser Studie.

3.1.2 Weather Dataset

Das Wetter könnte einen Einfluss auf den Taxifahrpreis in New York City haben. Durch extreme Wetterereignisse wie starker Regenfall, Schneefall oder auch Stürme könnte die Verkehrlage erheblich beeinflusst worden sein, weil die Taxifahrer aufgrund der extremen

Wetterbedingungen langsamer fahren oder sogar anhalten müssen. So entstehen vermutlich zusätzliche Kosten für die Fahrt. Deshalb wurden Wetterdaten für die Erweiterung der Datenbasis einbezogen.

Das National Climatic Data Center (NCDC) der USA wurde 1951 gegründet. Es gilt als das größte Archiv für Wetterdaten. Es sind momentan Daten der letzten 150 Jahre gespeichert und täglich kommen ca. 224 Gigabyte neu hinzu [23]. Die Wetterdaten sind auf der Seite des NCDCs [24] verfügbar und werden von der National Oceanic and Atmospheric Administration (NOAA) betrieben. Es wurde die Wetterstation im Central Park für die täglichen Messungen ausgewählt, weil die Wetterstation in New York City liegt. Die CSV-Datei umfasst ein Datenvolumen von 499 Kilobyte.

In Abbildung 3.2 ist ein Ausschnitt aus der Datentabelle Weather dargestellt, in der jede Zeile die Wetterinformationen für einen gegebenen Tag liefert.

STATION	NAME	LATITUDE	LONGITUDE	ELEVATION	DATE	AWND	FMTM	PGTM	PRCP	SNOW	SNWD	TAVG	TMAX	TMIN	TSUN	WDF2	WDF5	WSF2	WSF5	WT01	WT02	WT03	WT04	WT05	WT06	WT07	WT08	WT09
USW00094728	NY CITY CENTRAL PARK, NY US	40.77898	-73.96925	42.7	2009-01-01	11.179688	615.0	47.0	0.000000	0.000000	0.0	NaN	26.0	15.0	NaN	300.0	310.0	23.0	36.9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
USW00094728	NY CITY CENTRAL PARK, NY US	40.77898	-73.96925	42.7	2009-01-02	6.261719	959.0	852.0	0.000000	0.000000	0.0	NaN	34.0	23.0	NaN	230.0	220.0	17.0	30.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
USW00094728	NY CITY CENTRAL PARK, NY US	40.77898	-73.96925	42.7	2009-01-03	10.070312	1209.0	1106.0	0.000000	0.000000	0.0	NaN	38.0	29.0	NaN	290.0	280.0	17.0	25.9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
USW00094728	NY CITY CENTRAL PARK, NY US	40.77898	-73.96925	42.7	2009-01-04	7.609375	13.0	2353.0	0.000000	0.000000	0.0	NaN	42.0	25.0	NaN	270.0	270.0	19.9	25.1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
USW00094728	NY CITY CENTRAL PARK, NY US	40.77898	-73.96925	42.7	2009-01-05	6.929688	1023.0	1129.0	0.000000	0.000000	0.0	NaN	43.0	38.0	NaN	290.0	300.0	14.1	21.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
USW00094728	NY CITY CENTRAL PARK, NY US	40.77898	-73.96925	42.7	2009-01-06	6.710938	2318.0	2329.0	0.080017	0.000000	0.0	NaN	38.0	31.0	NaN	70.0	60.0	15.0	17.9	1.0	NaN	NaN	NaN	1.0	NaN	NaN	NaN	NaN
USW00094728	NY CITY CENTRAL PARK, NY US	40.77898	-73.96925	42.7	2009-01-07	10.507812	609.0	604.0	1.190430	0.000000	0.0	NaN	38.0	31.0	NaN	60.0	60.0	25.1	32.0	1.0	NaN	NaN	NaN	1.0	NaN	NaN	NaN	NaN
USW00094728	NY CITY CENTRAL PARK, NY US	40.77898	-73.96925	42.7	2009-01-08	11.406250	1813.0	1605.0	0.000000	0.000000	0.0	NaN	38.0	29.0	NaN	270.0	280.0	21.0	31.1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
USW00094728	NY CITY CENTRAL PARK, NY	40.77898	-73.96925	42.7	2009-01-09	9.617188	1542.0	1431.0	0.000000	0.000000	0.0	NaN	32.0	26.0	NaN	290.0	270.0	17.9	29.1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Abbildung 3.2: Ausschnitt aus der Datentabelle Weather

Aus Abbildung 3.2 ist zu entnehmen, dass es zusätzlich zu den Wettertypen auch metrische Werte für Temperaturen, Windstärke, Niederschlag, Schneefall und Schneetiefe gibt. Diese Attribute nennen weitere Details über das Wetter.

Auch hier befinden sich die Beschreibung der Attribute aus der Datentabelle Weather und die Angaben zu den Datentypen im Anhang.

3.1.3 US Federal Holiday Dataset

Bundesfeiertage könnten einen Einfluss auf den Taxifahrpreis in New York City haben. An manchen Feiertagen könnte das Verkehrsaufkommen geringer sein. Dadurch entstehen vermutlich keine Zusatzkosten infolge von Wartezeiten im Stau oder von längeren Wartezeiten. Deshalb wurden Daten zu den amerikanischen Bundesfeiertagen für die Erweiterung der Datenbasis hinzugefügt.

Die Daten über amerikanische Bundesfeiertagen werden auf der Kaggle-Seite [25] als CSV-Datei für den Zeitraum von 1966-2020 bereitgestellt. Die CSV-Datei umfasst ein Datenvolumen von 15 Kilobyte.

In Abbildung 3.3 ist ein Ausschnitt aus der Datentabelle Holidays dargestellt, wo jede Zeile einen Feiertag anzeigt.

	Date	Holiday
110	2009-01-01	New Year's Day
111	2009-01-19	Martin Luther King, Jr. Day
112	2009-02-16	Washington's Birthday
113	2009-05-25	Memorial Day
114	2009-07-03	Independence Day
115	2009-09-07	Labor Day
116	2009-10-12	Columbus Day
117	2009-11-11	Veterans Day
118	2009-11-26	Thanksgiving Day
119	2009-12-25	Christmas Day
100	2010-01-01	New Year's Day
101	2010-01-18	Martin Luther King, Jr. Day
102	2010-02-15	Washington's Birthday
103	2010-05-31	Memorial Day
104	2010-07-05	Independence Day
105	2010-09-06	Labor Day
106	2010-10-11	Columbus Day
107	2010-11-11	Veterans Day
108	2010-11-25	Thanksgiving Day
109	2010-12-25	Christmas Day
0	2010-12-31	New Year's Day

Abbildung 3.3: Ausschnitt aus der Datentabelle Holidays

Aus Abbildung 3.3 ist zu entnehmen, dass die Spalte Date als Datetime angegeben wird und die dazugehörigen Feiertage als String.

3.1.4 Sonstige Datensätze

Zusätzlich zu den Wetter- und Feiertagsdaten könnten Daten über das Verkehrsvolumen, Verkehrsunfälle und Großveranstaltungen zur Anreicherung der Datenbasis vorgesehen werden, weil die Daten einen direkten Einfluss auf den Verkehr ausüben und dadurch den Taxifahrpreis verändern können. Auf der Seite Open Data für New York City [26] sind die Daten abgelegt und als CSV-Dateien abgespeichert. In Tabelle 3.1 werden die sonstigen Datensätze mit dem Grund für die Nichtübernahme in die Datenbasis aufgezeigt.

Datensatz	Grund für nicht Übernahme
NYC Traffic Volume Counts	Datensatz startet im Jahr 2011 und enhält keine Daten aus den Jahren 2013 und 2014
NYPD Motor Vehicle Collisions	Datensatz startet im Jahr 2012
NYC Permitted Event Information	Datensatz enthält den Standort der Veranstaltung als City Block

Tabelle 3.1: Gründe für die Nichtübernahme der Daten

Aus Tabelle 3.1 ist zu entnehmen, dass die sonstigen Datensätze häufig wegen der fehlenden Daten aus bestimmten Jahren für die Erweiterung der Datenbasis nicht eingesetzt werden können.

3.2 Mögliche Vorhersagen mit dem NYC-Yellow-Taxi-Dataset

Neben der Vorhersage des Taxifahrpreises in New York City lassen sich anhand der Datenbasis (siehe Abschnitt 3.1.1) auch weitere Ziele realisieren.

3.2.1 Taxi Demand Prediction

Die Vorhersage der Nachfrage nach Taxis an bestimmten Orten und zu bestimmten Uhrzeiten könnte die Effizienz der Taxifahrer in New York City erhöhen und die Wartezeit für die Fahrgästen verringern. Dieses Ziel verfolgen Smith u.a. [27] mit ihrer Arbeit. Hier wird mithilfe von Features wie Datum/Uhrzeit, Wohngebiet, frühere Anzahl der Tweets und Nachfragen die zukünftige Zahl der Nachfragen nach Taxis vorhergesagt. Es wurde ein

Random-Forest-Regressions-Modell für mehrere Untermengen der Features angewendet, um die Aussagekraft der einzelnen Features zu testen.

3.2.2 Taxi Dispatch Planning

Eine Erweiterung der Vorhersage der Nachfrage nach Taxis in New York City ist die Entwicklung eines intelligenten Taxiversandsystems. Hierbei können Taxifahrer dem Fahrgast so zugeordnet werden, dass Leerfahrten des Taxifahrers verrringert werden und er am Ende der Fahrt dem nächsten Fahrgast zugeordnet wird. Dieses System besteht aus zwei Modellen, die Vorhersagen zu den Anfragen und über das Ziel ermitteln. Die Entwicklung eines intelligenten Taxiversandsystems beschreiben Xu u.a. [28] in ihrer Arbeit. Es wurden Features wie Datum, Wochentag, Tageszeit, Wetterinformationen und Wohngebiet genutzt. Für die Vorhersage der Anfragen nach Taxis wurde das LSTM-Netz und für die Vorhersage des Ziels wurde ein Feed-Forward neuronales Netz genutzt. Dabei wird bei der Vorhersage die Warscheinlichkeitsverteilung für ein Ziel in einem Wohngebiet und nicht der genaue Zielort ermittelt.

3.2.3 Taxi Travel Time Prediction

Die Vorhersage der Reisezeit zwischen Start- und Zielort einer Taxifahrt könnte ein Indikator für die jeweilige Beschreibung der Verkehrslage sein. Die Arbeit von Wang u.a. [29] enthält nicht die üblichen routenbasierten Reiseschätzungen, sondern mithilfe von historischen Daten, die ähnliche Strecken bzw. Orte aufweisen, wird der Mittelwert für die Fahrzeiten berechnet. Die Geschwindigkeit wird als Feature genutzt, weil sie ein Indikator für die jeweilige Verkehrslage ist.

3.2.4 Airport Pickup Decision

Laut dem NYC Taxi Limousine Commission Factbook [30] befindet sich 14 % der Abholorte der gelben Taxis am La Guardia Airport und am JFK Airport. Um einen Taxifahrer gegen Ende einer Fahrt bei der Entscheidung zu unterstützen, ob er im jeweiligen Ort nach Fahrgästen suchen oder zum Airport fahren soll, könnte man auf ein binäres Vorhersagemodell zurückgreifen, das Yazici u.a. [31] vorstellt. Dabei werden Features wie Wetterbedingungen, Datum und Uhrzeit sowie der letzte Zustellungsort genannt. Es wird ein binäres Logistic-Regression-Modell entwickelt, das als Output “airport pickup„ oder

“cruising for customers„ enthält. Es wurde festgestellt, dass Taxifahrer bei schlechtem Wetter die Strategie verfolgen, mehrere kurze Fahrten durchzuführen, die am Ende zu einem höheren Fahrpreis führen, anstatt zum Airport zu fahren und mit den Wartezeit die Flatrate anzunehmen.

3.2.5 Taxi Hourly Rate Prediction

Eine andere Möglichkeit, die Einnahmen einer Fahrt zu bewerten, besteht darin, den Stundenlohn des Taxifahrers für eine bestimmte Fahrt vorherzusagen. Hier sind die Features ähnlich wie die zur Vorhersage des Taxifahrpreises geeignet.

3.3 Qualität der Daten

Die Qualität einer Vorhersage ist abhängig von der Qualität der Eingabedaten. Dieses Prinzip „Garbage in - Garbage out“ist von Lidwell u.a. [32] analysiert worden. Hierbei liegen die Daten in ihrer Ursprungsform häufig nicht in der gewünschten Qualität vor. Im Folgenden werden die Kriterien zur Prüfung der Datenqualität vorgestellt:

- **Verständlichkeit**: Sind die Daten verständlich oder können wir sie interpretieren?
- **Nützlichkeit**: Sind die Daten für das angestrebte Ziel wertvoll?
- **Gültigkeit**: Handelt es sich um valide Daten?
- **Glaubwürdigkeit**: Passen die Daten zu unseren Erfahrungen?
- **Exakheit**: Sind die Daten präzise? Sind sie widerspruchsfrei und vollständig?
- **Aktualität**: Sind die Daten aktuell?
- **Volatilität**: Die vorliegenden Daten sollten über eine gewisse Periode repräsentativ sein.

Quelle:[33]

In dieser Arbeit werden die Kriterien Verständichkeit, Nützlichkeit, Gültigkeit, Glaubwürdigkeit und Exaktheit zur Bewertung der Datenqualität genutzt. Beispiele für mangelnde Datenqualität wären hier falsche oder fehlende Werte und Ausreißer.

Bevor es um die Bewertung der Datenqualität geht, werden die Beobachtungen zu der ersten Sichtung dargestellt. Es zeigt sich, dass die Spaltennamen nicht immer einheitlich sind. Dieses Problem wird in Abbildung 3.4 deutlich.

December 2009

```
dfs_2009[11]
```

	vendor_name	Trip_Pickup_DateTime	Trip_Dropoff_DateTime	Passenger_Count	Trip_Distance	Start_Lon	Start_Lat	Rate_Code	store_and_forward	End_Lon	End_Lat	Payment_Type	Fare_Amt	surcharge
0	VTS	2009-12-17 07:35:00	2009-12-17 07:40:00	1	0.11	-73.987930	40.737885	NaN	NaN	-73.990334	40.748451	Credit	4.898438	0.0
1	VTS	2009-12-21 14:19:00	2009-12-21 14:24:00	1	1.07	-73.956009	40.779556	NaN	NaN	-73.967300	40.787834	CASH	4.898438	0.0
2	VTS	2009-12-18 03:09:00	2009-12-18 03:34:00	1	8.98	-73.955742	40.689503	NaN	NaN	-73.937729	40.737465	CASH	23.703125	0.5
3	VTS	2009-12-14 21:24:00	2009-12-14 21:33:00	2	1.66	-73.983986	40.754646	NaN	NaN	-73.986198	40.737610	Credit	6.898438	0.5
4	VTS	2009-12-18 08:17:00	2009-12-18 08:29:00	1	1.55	-73.959129	40.769264	NaN	NaN	-73.976265	40.760616	CASH	7.699219	0.0

...

January 2010

```
dfs_2010[0]
```

	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	trip_distance	pickup_longitude	pickup_latitude	rate_code	store_and_fwd_flag	dropoff_longitude	dropoff_latitude	payment_type	fare_amount
0	VTS	2010-01-26 07:41:00	2010-01-26 07:45:00	1	0.75	-73.956779	40.767750	1	NaN	-73.965958	40.765232	CAS	4.500000
1	DDS	2010-01-30 23:31:00	2010-01-30 23:46:12	1	5.90	-73.996117	40.763931	1	NaN	-73.981514	40.741192	CAS	15.296875
2	DDS	2010-01-18 20:22:20	2010-01-18 20:38:12	1	4.00	-73.979675	40.783791	1	NaN	-73.917854	40.878559	CAS	11.703125
3	VTS	2010-01-09 01:18:00	2010-01-09 01:35:00	2	4.70	-73.977921	40.763996	1	NaN	-73.923904	40.759724	CAS	13.296875
4	CMT	2010-01-18 19:10:14	2010-01-18 19:17:07	1	0.60	-73.990921	40.734680	1	0	-73.995514	40.739086	Cre	5.300781

Abbildung 3.4: Ausschnitte aus der Datentabelle Yellow Cabs für Dezember 2009 und Januar 2010

Aus Abbildung 3.4 ist zu entnehmen, dass sich die Spaltennamen nicht nur geändert haben, sondern die Groß- und Kleinschreibung willkürlichen Wechseln unterliegt.

Eine andere Auffälligkeit ist der veränderte Datentyp in Bezug auf die Werte im Zeitablauf. Dies wird in Abbildung 3.5 deutlich.

December 2014

```
dfs_2014[11]
```

	vendor_id	pickup_datetime	dropoff_datetime	passenger_Count	trip_distance	pickup_longitude	pickup_latitude	rate_code	store_and_fwd_flag	dropoff_longitude	dropoff_latitude	payment_type	fare_amount
0	VTS	2014-12-12 18:16:00	2014-12-12 18:35:00	3	4.03	-74.014053	40.711708	1	NaN	-73.995628	40.759460	CSH	16.0
1	VTS	2014-12-12 18:18:00	2014-12-12 18:36:00	1	4.10	-73.945877	40.780525	1	NaN	-73.972557	40.740459	CRD	15.0
2	VTS	2014-12-12 18:31:00	2014-12-12 18:35:00	1	0.96	-73.961449	40.796261	1	NaN	-73.955627	40.787762	CSH	5.5
3	VTS	2014-12-08 01:53:00	2014-12-08 01:55:00	5	0.76	-73.955276	40.768677	1	NaN	-73.948975	40.777363	CRD	4.5
4	VTS	2014-12-12 17:58:00	2014-12-12 18:34:00	1	11.19	-73.862694	40.768959	1	NaN	-73.745811	40.766701	CSH	35.0

...

January 2015 ¶

```
dfs_2015[0]
```

	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	pickup_longitude	pickup_latitude	RateCodeID	store_and_fwd_flag	dropoff_longitude	dropoff_latitude	payment_type	f
0	2	2015-01-15 19:05:39	2015-01-15 19:23:42	1	1.59	-73.993896	40.750111	1	N	-73.974785	40.750618	1	
1	1	2015-01-10 20:33:38	2015-01-10 20:53:28	1	3.30	-74.001648	40.724243	1	N	-73.994415	40.759109	1	
2	1	2015-01-10 20:33:38	2015-01-10 20:43:41	1	1.80	-73.963341	40.802788	1	N	-73.951820	40.824413	2	
3	1	2015-01-10 20:33:39	2015-01-10 20:35:31	1	0.50	-74.009087	40.713818	1	N	-74.004326	40.719986	2	
4	1	2015-01-10 20:33:39	2015-01-10 20:52:58	1	3.00	-73.971176	40.762428	1	N	-74.004181	40.742653	2	

Abbildung 3.5: Ausschnitte aus der Datentabelle Yellow Cabs für Dezember 2014 und Januar 2015

Aus Abbildung 3.5 ist zu entnehmen, dass die Werte häufig von String zu integer konvertieren. Ein Beispiel hierfür wäre wie in Abbildung 3.5 die Spalte vendor_id.

Die Zahl der Einträge ist im Monat Februar und März 2010 nicht immer identisch mit der Zahl der Spalten, wie in Abbildung 3.6 dargestellt wird.

```
ParserError: Error tokenizing data. C error: Expected 18 fields in line 265240, saw 19
```

ParserError, because there are Rows with more Entries than Number of Columns ¶

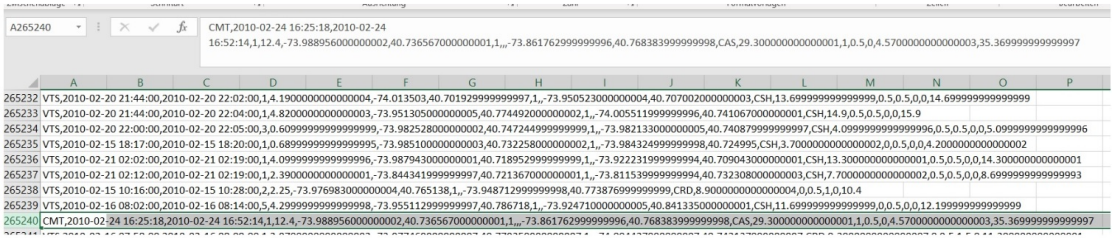


Abbildung 3.6: Fehlermeldung für eine nicht stimmige Zahl an Einträgen mit Spalten

Für die Wetterdaten und Bundesfeiertagsdaten wurden keine Auffälligkeiten festgestellt, die als eine CSV-Datei vorliegen.

Bevor die Yellow-Cab-Daten für die weitere Analysen weitergereicht werden können, müssen sie zunächst in eine einheitliche Form überführt werden. Dazu zählen die homogene Namensgebung der Spalten, die Konvertierung der String-Werte vor dem Jahr 2015 in

Integer-Werte und die Löschung der Kommata in den CSV-Dateien im Februar und März 2010 mithilfe eines Skripts.

3.3.1 Fehlende Daten

Nachdem die Yellow-Cab-Daten in eine einheitliche Form überführt worden sind, werden die Daten auf fehlende Einträge geprüft. Es hat sich jedoch gezeigt, dass nur die drei Features RateCodeID, Dropoff_Lon und Dropoff_Lat von diesem Problem betroffen sind (vgl. Abbildung 3.7).

	Passenger_Count	Trip_Distance	Pickup_Lon	Pickup_Lat	RateCodeID	Dropoff_Lon	Dropoff_Lat	Fare_Amt	Pickup_Year
count	2.000000e+06	2.000000e+06	2.000000e+06	2.000000e+06	1.749721e+06	1.999989e+06	1.999989e+06	2.000000e+06	2.000000e+06
mean	1.681454e+00	inf	-7.257114e+01	3.996286e+01	1.038570e+00	-7.259656e+01	3.997035e+01	inf	2.012500e+03
std	1.333940e+00	NaN	1.297025e+01	1.086378e+01	5.887309e-01	1.306104e+01	1.079283e+01	NaN	2.291288e+00
min	0.000000e+00	0.000000e+00	-3.358101e+03	-3.114292e+03	1.000000e+00	-3.386093e+03	-3.486646e+03	-1.000000e+02	2.009000e+03
25%	1.000000e+00	1.009766e+00	-7.399204e+01	4.073508e+01	1.000000e+00	-7.399136e+01	4.073408e+01	6.101562e+00	2.010750e+03
50%	1.000000e+00	1.730469e+00	-7.398177e+01	4.075269e+01	1.000000e+00	-7.398006e+01	4.075323e+01	8.500000e+00	2.012500e+03
75%	2.000000e+00	3.169922e+00	-7.396697e+01	4.076724e+01	1.000000e+00	-7.396339e+01	4.076828e+01	1.300000e+01	2.014250e+03
max	2.080000e+02	inf	3.380714e+03	3.351468e+03	2.100000e+02	2.443999e+03	3.351468e+03	inf	2.016000e+03

Abbildung 3.7: Tabelle der zusammengefassten Statistik für Yellow Cab

Aus Abbildung 3.7 lässt sich entnehmen, dass bei den Dropoff_Lon und Dropoff_Lat jeweils 11 Einträge fehlen, die ca. 0,0011 % des gesamten Datensatzes ausmachen. Bei den RateCodeIDs fehlen 250.279 Einträge, die ca. 12,5 % des Datensatzes entsprechen. Eine Besonderheit der fehlenden Werte für Drop-off ist, dass für die Fahrt trotzdem ein Fare Amount erhoben wurde, wie aus Abbildung 3.8 zu entnehmen ist.

	Pickup_DateTime	Dropoff_DateTime	Passenger_Count	Trip_Distance	Pickup_Lon	Pickup_Lat	RateCodeID	Dropoff_Lon	Dropoff_Lat	Fare_Amt
817158	2012-12-11 12:36:38	2012-12-11 12:51:28	0	0.0	-73.979042	40.755409	0	NaN	NaN	14.351562
841102	2012-12-11 13:29:54	2012-12-11 13:41:08	0	0.0	-73.972008	40.757191	0	NaN	NaN	11.851562
841592	2012-12-11 13:17:00	2012-12-11 13:44:37	0	0.0	-73.996437	40.737808	0	NaN	NaN	20.500000
976192	2012-12-11 12:12:16	2012-12-11 12:21:15	0	0.0	-74.008751	40.731773	0	NaN	NaN	10.601562
995619	2012-12-11 13:06:07	2012-12-11 13:27:52	0	0.0	-73.967072	40.763653	0	NaN	NaN	30.500000
1056690	2013-07-03 16:10:39	2013-07-03 16:10:39	0	0.0	-74.001541	40.740929	0	NaN	NaN	6.648438
1108561	2013-06-21 22:06:57	2013-06-21 22:06:57	0	0.0	-73.966858	40.757732	0	NaN	NaN	15.000000
1132571	2013-06-21 14:53:08	2013-06-21 14:53:08	0	0.0	-73.991463	40.749828	0	NaN	NaN	22.906250
1161334	2013-01-09 19:49:27	2013-01-09 20:01:14	0	0.0	-73.967171	40.756767	0	NaN	NaN	13.000000
1174853	2013-03-23 20:46:25	2013-03-23 21:01:42	0	0.0	-73.985863	40.736195	0	NaN	NaN	12.000000
1226106	2013-07-04 03:24:40	2013-07-04 03:24:40	0	0.0	-73.985596	40.742348	0	NaN	NaN	11.000000

Abbildung 3.8: Einträge der fehlenden Werte für die Drop-off-Location

Dieser Hinweis könnte darauf hindeuten, dass eine Stornierungsgebühr erhoben wurde. Die fehlenden Einträge für den Drop-off können gelöscht werden, weil sie nur einen geringen Anteil des gesamten Datensatzes ausmachen.

Bei den fehlenden Werten der RateCodeIDs fällt auf, dass sie nur für 2009 vorliegen. Vermutlich wurde zu dieser Zeit die Systematik des RateCodeIDs noch nicht eingeführt. Für die RateCodeID könnte man die fehlende Werte durch eine globale Konstante wie 0 ersetzen, weil zum Beispiel für die RateCodeID wie in Abbildung 3.9 ungültige Werte wie 0 u.a. enhalten sind.

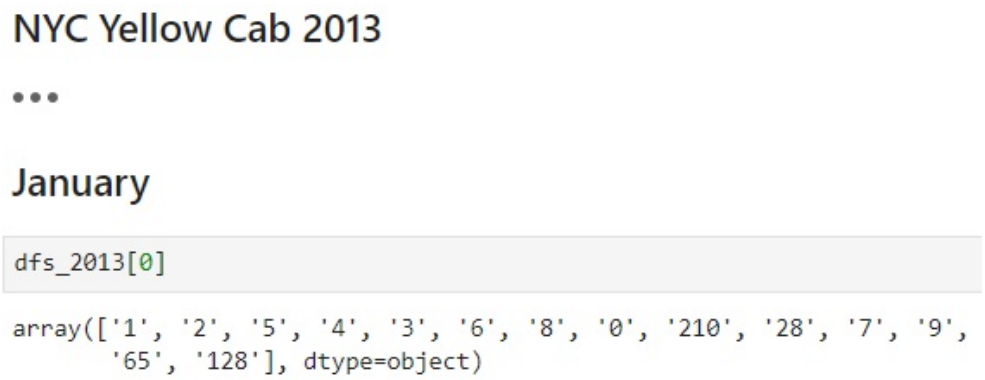


Abbildung 3.9: Ausschnitt der Werte für die RateCodeID für Januar 2013

Eine Löschung der Einträge für die fehlenden Werte des RateCodeID ist nicht vorgesehen, weil sie 12,5 % des gesamten Datensatzes entsprechen.

Abschließend wird in Tabelle 3.2 das Vorgehen für die fehlenden Werte im Datensatz Yellow Cab gezeigt.

Feature	Prozent an fehlende Daten	Lösung
RateCodeID	12,5 %	globale Konstante
Dropoff Lon & Lat	0,0011 %	Einträge löschen

Tabelle 3.2: Zusammenfassung des Vorgehens für fehlende Daten

Bei den Wetterdaten enthält nur das Feature Average Temperature fehlende Einträge. In Abbildung 3.10 werden die fehlenden Einträge anhand der Count-Zeile ersichtlich.

	TAVG	TMAX	TMIN	WT01	WT11	WT16	WT18
count	0.0	2738.000000	2738.000000	2738.000000	2738.000000	2738.000000	2738.00000
mean	NaN	inf	inf	0.293280	0.000365	0.211103	0.03908
std	NaN	18.390625	16.734375	0.455349	0.019111	0.408166	0.19382
min	NaN	15.000000	-1.000000	0.000000	0.000000	0.000000	0.00000
25%	NaN	48.000000	36.000000	0.000000	0.000000	0.000000	0.00000
50%	NaN	64.000000	49.000000	0.000000	0.000000	0.000000	0.00000
75%	NaN	79.000000	63.000000	1.000000	0.000000	0.000000	0.00000
max	NaN	104.000000	84.000000	1.000000	1.000000	1.000000	1.00000

Abbildung 3.10: Tabelle der zusammengefassten Statistik für Wetterdaten

Aus Abbildung 3.10 ist zu entnehmen, dass für das Feature Average Temperature alle Einträge im Datensatz fehlen. Das Feature Average Temperature wurde aus dem Datensatz gelöscht, weil aus der Mittelwertberechnung der beiden Features TMAX und TMIN der Average Temperature berechnet werden kann.

Die Bundesfeiertagsdaten sind als korrekt und vollständig anzusehen, weil sie von der Kaggle-Seite aufbereitet wurden.

3.3.2 Falsche Daten

Im Anschluss an die Bereinigung der fehlenden Daten wurden die Datensätze auf falsche Werte geprüft. Zu diesem Zweck werden zunächst die Distribution der einzelnen Features und die Koordinaten auf der Map dargestellt.

Bei der Distribution der Geokoordinaten in Abbildung 3.11 fällt auf, dass es Koordinaten gibt, die außerhalb von New York City liegen.

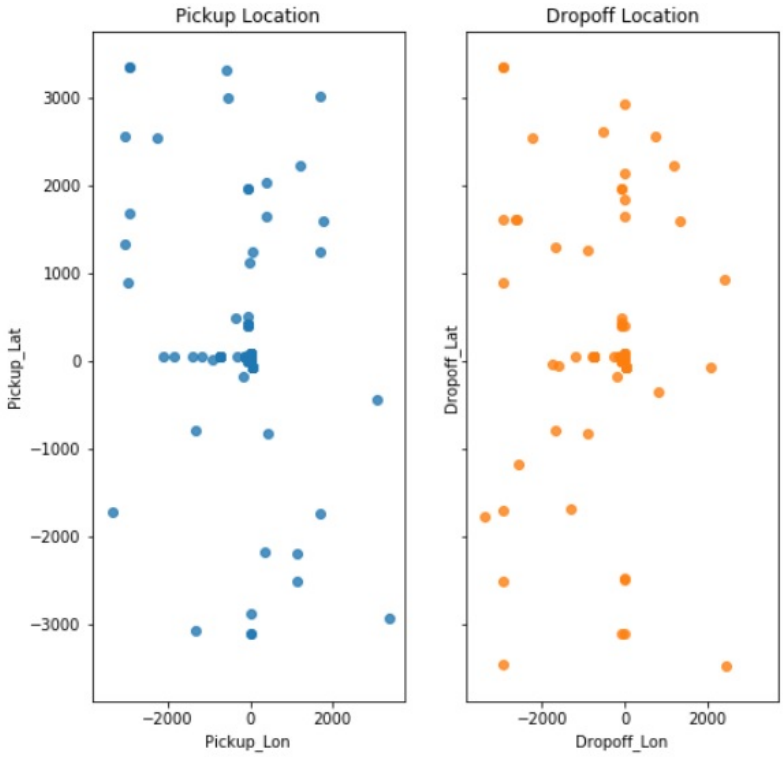


Abbildung 3.11: Distribution der Geokoordinaten

Vermutlich gab es Anomalien aufseiten des GPS-Sensors bei der Übermittlung der Daten vom Service-Provider. Die Bounding Box für New York City ist -74.25909, 40.491721, -73.700181, 40.916178. Der Anteil der Koordinaten, die außerhalb von New York City liegen, ist jedoch gering. Diese Daten wurden daher aus dem Datensatz entfernt.

Abbildung 3.12 zeigt die Verteilung der Zahl der Fahrgäste.

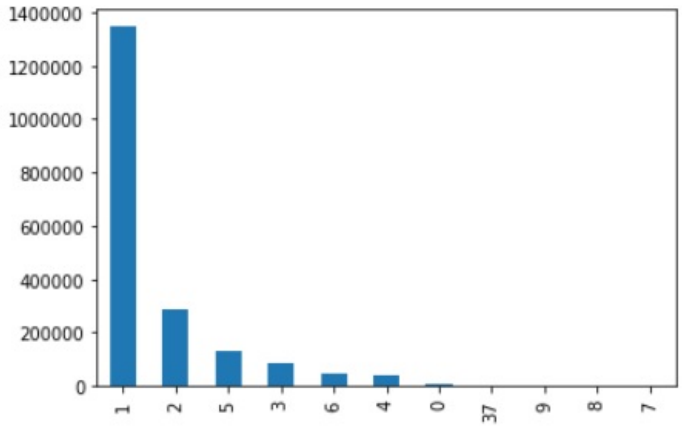


Abbildung 3.12: Distribution der Passenger Count

Die Auswertung hat ergeben, dass 0,3 % der Fahrten ohne Fahrgäste stattfanden und 0,0002 % der Fahrten aus mehr als 6 Fahrgäste bestanden. Möglicherweise hatten die Taxifahrer die Zahl der Fahrgäste in diesen Fällen falsch eingegeben. Der Anteil der offensichtliche unkorrekten Angaben zu den Fahrgästen wurde deshalb aufgrund des geringen Anteils aus dem Datensatz gelöscht.

Abbildung 3.13 zeigt die Verteilung der RateCodeID.

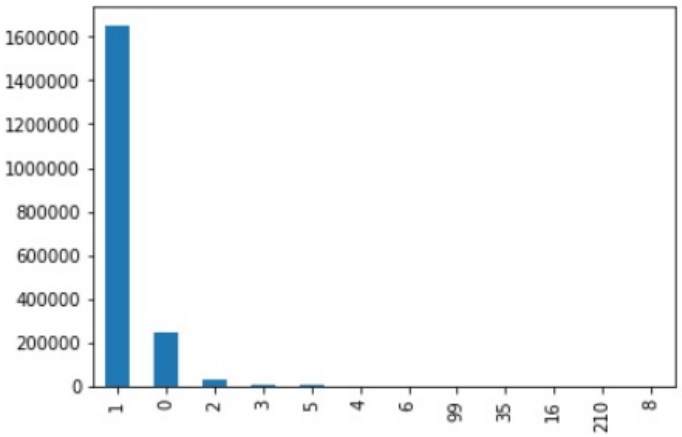


Abbildung 3.13: Distribution der RateCodeID

Es fällt auf, dass 12,5 % der RateCodeID den Wert 0 aufweisen und 0,0006 % der RateCodeID Werte über 6 haben. Die gültigen RateCodes liegen laut TLC zwischen den Werten 1 und 6. Für Werte über 6 wurden die Einträge gelöscht, weil sie nur einen geringen Anteil ausmachen. Der Wert 0 wurde beibehalten, weil die Einträge wichtige Informationen für die weitere Analyse bereithalten.

Abbildung 3.14 zeigt die Verteilung der Distanz für die Fahrstrecke in Intervallen.

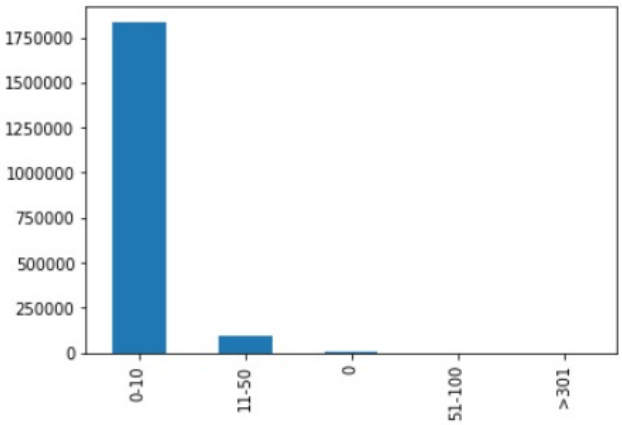


Abbildung 3.14: Distribution der Trip Distance

Es fällt auf, dass 0,2 % der Strecken eine Länge von 0 Meilen aufweist und 0,001 % der Strecken über 50 Meilen lang sind. Es könnten aufseiten der Taxifahrer Anomalien im Taximeter bei der Übertragung der Daten vorliegen. Einträge mit einer Streckenlänge von 0 Meilen wurden aus dem Datensatz gelöscht, weil sie einen geringen Anteil erfassen, aber die Einträgen mit der Distanzlänge von über 50 Meilen wurden beibehalten, weil sie uns vermutlich für die weitere Analyse weitere Hinweise über das Fahrverhalten des Taxifahrers geben.

Abbildung 3.15 zeigt die Verteilung des Fare Amounts.

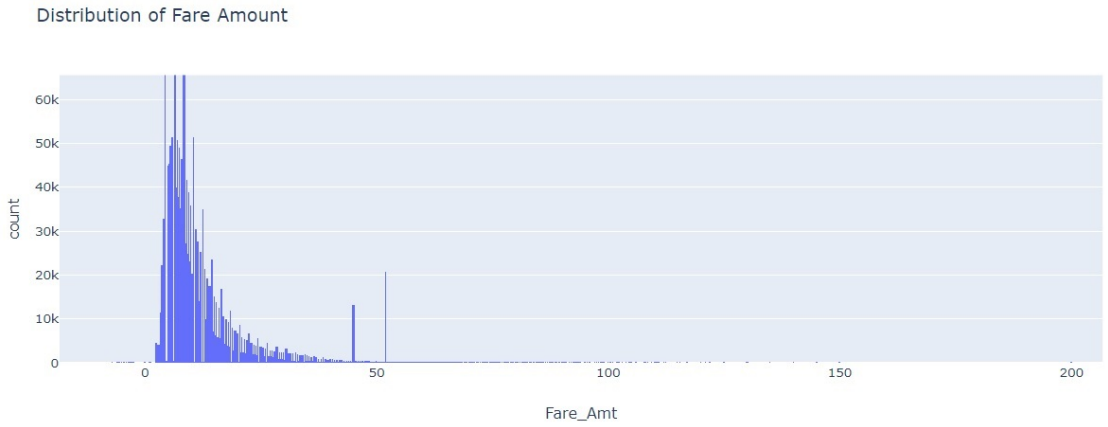


Abbildung 3.15: Distribution des Fare Amounts

Es fällt auf, dass 0,0096 % der Fahrten negative Dollar, 0,005 % genau 0 Dollar und 0,01 % über 100 Dollar erwirtschaftet haben. Die Einträge negative Dollar und 0 Dollar wurden aus dem Datensatz gelöscht, weil diese Angaben vermutlich auf inkorrekte Zahlvorgänge hinweisen. Es wurden auch Einträge gelöscht, die kleiner gleich 2,5 Dollar sind, weil der Grundpreis bereits bei 2,5 Dollar liegt.

Nachdem die Distribution aufgezeigt und auch bereinigt wurde, wurden die Geokoordinaten auf der Map geplottet. In Abbildung 3.16 stehen die blauen Punkte für die Abholorte und die roten Punkte für die Zustellungsorte.

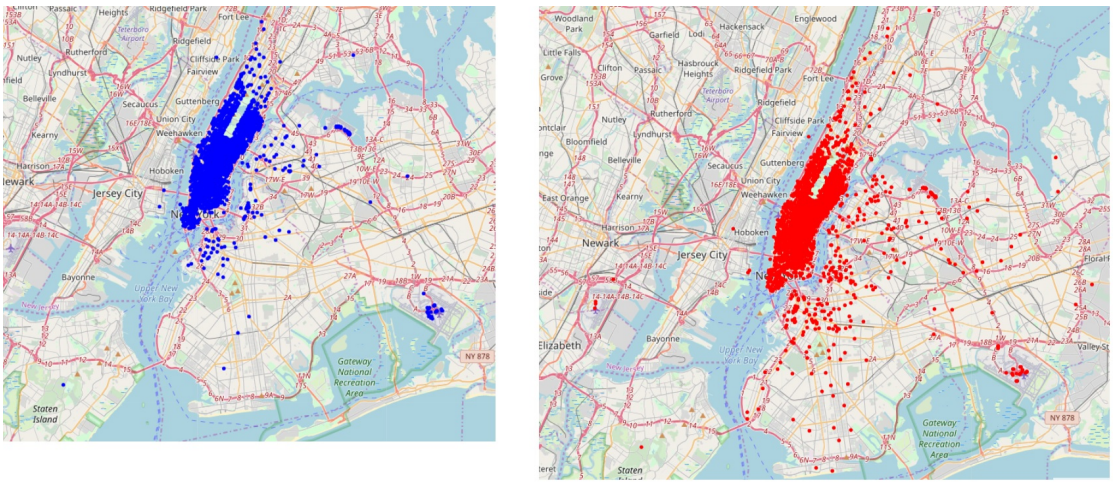


Abbildung 3.16: Abhol- und Zustellungsorte auf der Karte

Die Punkte konzentrieren sich auf Abholorte in Manhattan, am La Guardia Airport und am JFK Airport. Bei den Zustellungsorten ist dasselbe Muster zu erkennen, aber mit dem Zusatz, dass die Bezirke Brooklyn und Queens vermehrt zu den Zielorten gehören. Ein Problem ist, dass beide Abbildungen Punkte zeigen, die im Atlantischen Ozean liegen. Diese Fehler sind möglicherweise auf Anomalien im GPS-Sensor zurückzuführen.

Die TLC stellt auf ihrer Seite [7] zusätzlich zu den Reisedaten auch ein Shapefile zur Verfügung, das die zu den Koordinaten dazugehörigen Bezirke und Taxizonen beschreibt. Diese Daten habe ich nicht nur für die Anreicherung der Geokoordinaten genutzt, sondern auch zum Plotten der Orte außerhalb von New York City. In Abbildung 3.17 ist dies dargestellt.

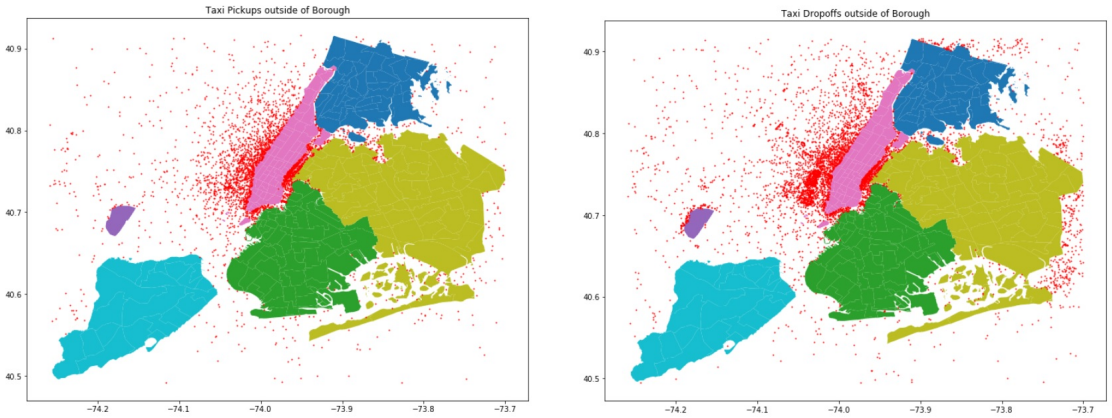


Abbildung 3.17: Plots der Koordinaten außerhalb von New York City

Alle Einträge, d.h. 0,18 % der Abholorte und 0,28 % der Zustellungsort, die außerhalb von New York City oder im Atlantischen Ozean liegen, wurden aus dem Datensatz entfernt.

Weiterhin wurde geprüft, ob es Einträge gibt, wo der Abholort und Zustellungsort identisch sind. Abbildung 3.18 zeigt einen Überblick.

	Pickup_DateTime	Dropoff_DateTime	Passenger_Count	Trip_Distance	Pickup_Lon	Pickup_Lat	RateCodeID	Dropoff_Lon	Dropoff_Lat	Fare_Amt
61	2009-07-31 19:11:23	2009-07-31 19:11:42	1	0.000000	-73.981522	40.690323	0	-73.981522	40.690323	2.500000
99	2009-11-12 11:07:00	2009-11-12 11:08:00	1	0.000000	-73.978523	40.761028	0	-73.978523	40.761028	2.500000
133	2009-11-24 19:16:00	2009-11-24 19:21:00	1	1.059570	-73.966858	40.763287	0	-73.966858	40.763287	4.898438
192	2009-06-28 18:45:00	2009-06-28 18:51:00	1	1.339844	-73.907974	40.883327	0	-73.907974	40.883327	6.101562
326	2009-06-09 23:48:36	2009-06-09 23:53:13	1	1.000000	-73.986847	40.766396	0	-73.986847	40.766396	4.898438
493	2009-07-31 13:40:00	2009-07-31 14:04:00	5	6.511719	-73.993629	40.633442	0	-73.993629	40.633442	19.296875
622	2009-07-20 08:32:25	2009-07-20 08:32:27	1	0.000000	-73.994553	40.761234	0	-73.994553	40.761234	2.500000
691	2009-09-25 19:13:51	2009-09-25 19:34:31	1	4.601562	-74.015060	40.639397	0	-74.015060	40.639397	14.101562
741	2009-08-23 07:56:58	2009-08-23 07:57:02	1	0.000000	-73.981575	40.743629	0	-73.981575	40.743629	2.500000
764	2009-12-04 22:27:39	2009-12-04 22:53:45	1	15.703125	-73.987389	40.724239	0	-73.987389	40.724239	35.312500
995	2009-12-25 10:36:00	2009-12-25 10:42:00	5	1.799805	-73.956223	40.778839	0	-73.956223	40.778839	6.500000
1031	2009-02-17 09:14:42	2009-02-17 09:17:22	1	0.700195	-73.931519	40.799900	0	-73.931519	40.799900	4.101562

Abbildung 3.18: Einträge mit gleichem Abholort und Zustellungsort

Diese Einträge entsprechen nur 0,89 % des Datenvolumens und wurden daher gelöscht.

Auch bei den Zeitfeatures wurden Anomalien identifiziert. Es wurde geprüft, ob die Abholzeit größer oder gleich der Zustellzeit ist. Dies wird in Abbildung 3.19 dargestellt.

	Pickup_DateTime	Dropoff_DateTime	Passenger_Count	Trip_Distance	Pickup_Lon	Pickup_Lat	RateCodeID	Dropoff_Lon	Dropoff_Lat	Fare_Amt
1052	2009-01-26 21:48:00	2009-01-26 21:47:00	5	1.269531	-73.955246	40.820042	0	-73.951843	40.804348	6.101562
1323	2009-01-06 14:53:00	2009-01-06 14:52:00	1	1.169922	-73.995743	40.754200	0	-73.998360	40.741287	5.699219
3035	2009-01-24 11:59:00	2009-01-24 11:53:00	1	0.429932	-73.976051	40.765560	0	-73.972878	40.761585	3.300781
3075	2009-01-16 10:41:00	2009-01-16 10:36:00	5	0.700195	-73.984444	40.732208	0	-73.982185	40.740177	3.699219
3248	2009-02-02 12:01:00	2009-02-02 11:57:00	5	0.549805	-73.984512	40.746662	0	-73.982224	40.740921	4.500000
4888	2009-01-15 12:33:00	2009-01-15 12:32:00	1	0.919922	-73.977829	40.745937	0	-73.983124	40.753448	5.300781
4904	2009-01-16 02:15:00	2009-01-16 02:03:00	1	0.819824	-73.989151	40.762955	0	-74.000031	40.761402	4.500000
6941	2009-05-04 11:08:00	2009-05-04 10:59:00	1	1.290039	-73.972870	40.761257	0	-73.977623	40.774029	6.500000

	Pickup_DateTime	Dropoff_DateTime	Passenger_Count	Trip_Distance	Pickup_Lon	Pickup_Lat	RateCodeID	Dropoff_Lon	Dropoff_Lat	Fare_Amt
184	2009-06-03 16:58:00	2009-06-03 16:58:00	1	0.209961	-73.972672	40.795925	0	-73.974419	40.793137	2.900391
391	2009-10-20 20:55:00	2009-10-20 20:55:00	1	0.029999	-73.989540	40.750584	0	-73.989540	40.750584	45.000000
523	2009-01-10 03:38:00	2009-01-10 03:38:00	3	1.599609	-73.997719	40.720936	0	-73.998131	40.738873	6.500000
833	2009-09-26 14:20:00	2009-09-26 14:20:00	1	0.000000	-73.983978	40.762123	0	-73.984016	40.762123	2.500000
1075	2009-08-11 11:07:00	2009-08-11 11:07:00	5	0.140015	-73.987640	40.760017	0	-73.986198	40.761784	2.500000
1752	2009-04-29 21:39:00	2009-04-29 21:39:00	5	0.280029	-73.957878	40.779198	0	-73.955490	40.782669	2.900391
2099	2009-02-09 18:04:00	2009-02-09 18:04:00	1	1.730469	-74.008568	40.704376	0	-74.015915	40.715118	7.699219

Abbildung 3.19: Einträge mit höherer Abhol- als Zustellzeit

Diese Anomalien treten mit 0,02 % und 0,11 % nur sehr selten auf und werden daher aus dem Datensatz entfernt.

Die Analysen und Quellcodes sind der Arbeit in mehreren Jupyter-Notebook-Dateien beigelegt.

3.3.3 Exploration der Daten

Nachdem die fehlenden und fehlerhaften Daten bereinigt worden sind, folgt eine explorative Datenanalyse. Die explorative Datenanalyse dient dazu, die Zusammenhänge zwischen den Daten zu erkennen und mithilfe von Diagrammen darzustellen. Dabei wird gezielt auf die Einflussfaktoren der einzelnen Eingabemerkmale auf das Ausgabemerkmal untersucht. Hierbei können falsche Daten oder Ausreißer genauer eingegrenzt werden. Für den Ablauf der explorativen Datenanalyse wurden folgende Fragen in Tabelle 3.3 verfasst.

Merkmalstypen	Fragen
Zeitliche Merkmale	1. Wie beeinflusst die Zeit den Fahrpreis? 2.Wie beeinflussen die Bundesfeiertage den Fahrpreis?
Räumliche Merkmale	1. Wird der Fahrpreis höher, wenn die zurückgelegte Strecke größer ist? 2.Wie beeinflusst der Abholort und Zustellungsort den Fahrpreis?
Wetterbedingte Merkmale	1.Wie beeinflusst das Wetter den Fahrpreis?
Andere Merkmale	1.Wie beeinflusst die Anzahl der Fahrgäste den Fahrpreis?

Tabelle 3.3: Fragen zur explorativen Datenanalyse

Zeitliche Merkmale

Aus dem Merkmal Pickup_Datetime werden weitere Merkmale wie das Jahr, der Monat, die Wochentage und die Uhrzeit auf Stundenbasis generiert, um den Einfluss der einzelnen Merkmale auf das Ausgabemerkmal Fahrpreis zu untersuchen.

Jahr

In Abbildung 3.20 werden die Anzahl der Fahrten und der Mittelwert des Fahrpreises für den Zeitraum von 2009-2016 dargestellt.

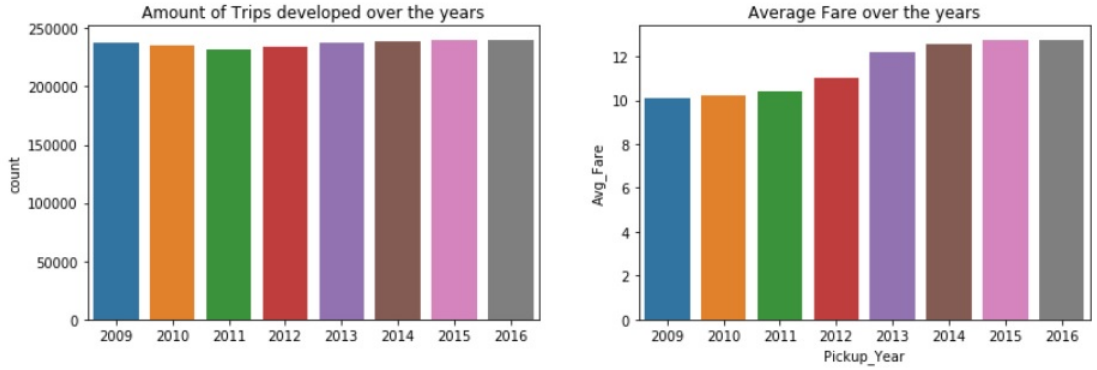


Abbildung 3.20: Anzahl der Fahrten und Mittelwert des Fare Amounts verteilt von 2009-2016

Die Anzahl der Fahrten ist im genannten Zeitraum weitgehend konstant geblieben, wohingegen der Fahrpreis über die Jahre kontinuierlich anstieg. Es könnte sein, dass sich die Preisstruktur in den Jahren verändert hat und so für die gleiche Fahrt ein höherer Fahrpreis entstanden ist.

Monat

In Abbildung 3.21 werden die Anzahl der Fahrten und der Mittelwert des Fahrpreises für ein Jahr dargestellt.

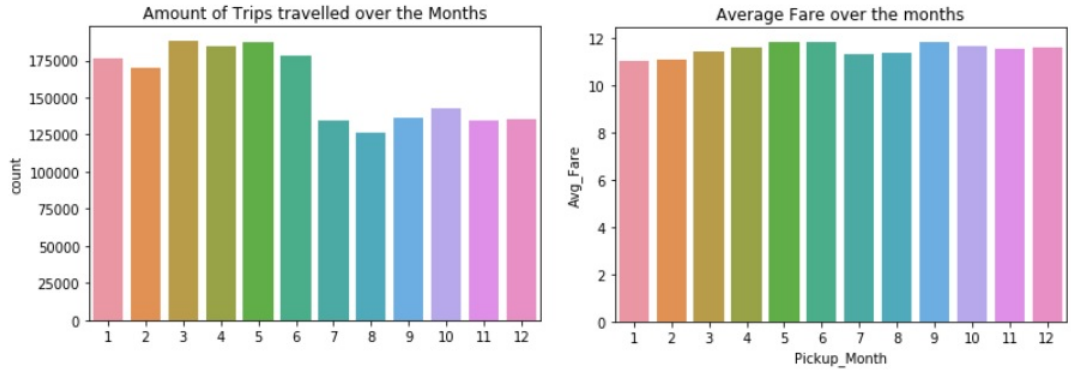


Abbildung 3.21: Anzahl der Fahrten und Mittelwert des Fare Amounts verteilt über 12 Monate

Die Anzahl der Fahrten ist von Juli bis Dezember signifikant geringer als von Januar bis Juni, wohingegen der Fahrpreis über die Monate hinweg nur geringe Veränderungen aufweist.

Wochentag

Abbildung 3.22 zeigt die Anzahl der Fahrten und den Mittelwert des Fahrpreises für eine Woche.

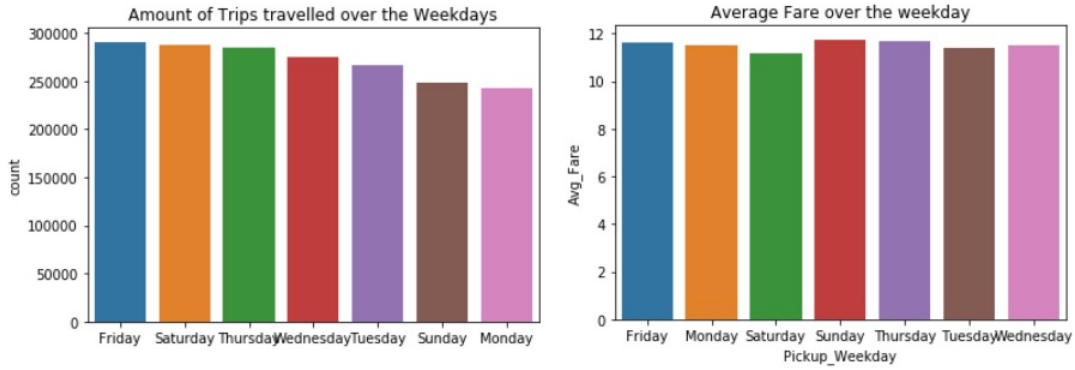


Abbildung 3.22: Anzahl der Fahrten und Mittelwert des Fare Amounts für eine Woche

Es ist zu erkennen, dass die Tage vor dem Wochenende wie Donnerstag und Freitag sowie der Samstag das höchste Fahrtaufkommen aufweisen. An diesen Tagen gehen die Menschen ihren Freizeitaktivitäten nach, erledigen Einkäufe, gehen ins Kino und treffen sich mit Familie und Freunden. Am Sonntag und Montag ist das niedrigste Fahrtaufkommen zu verzeichnen. Die höchsten Fare Amounts sind Donnerstag, Freitag und Sonntag, obwohl am Sonntag eher ein niedriges Fahraufkommen besteht. Möglicherweise bewirkt das niedrige Verkehrsaufkommen mehr Taxifahrten.

Tageszeit

Abbildung 3.23 zeigt die Anzahl der Fahrten und den Mittelwert des Fahrpreises für eine Tag.

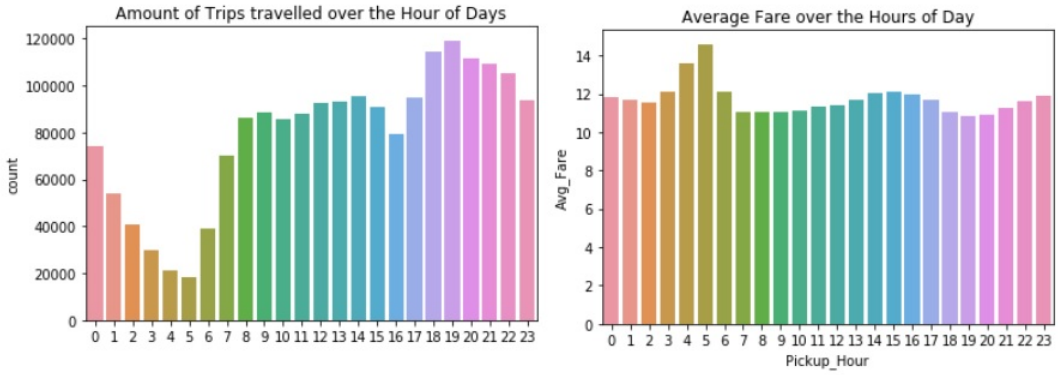


Abbildung 3.23: Anzahl der Fahrten und Mittelwert des Fare Amounts für einen Tag

Das Fahrtaufkommen geht in der Nacht stetig zurück und wächst während des Tages wieder an. Zwischen 18 und 19 Uhr wird das Maximum erreicht. Am Ende des Arbeitstages nehmen viele Menschen offenbar ein Taxi auf den Weg in die Wohnung oder sie treffen sich noch mit Freunden. Es ist ein lokales Minimum gegen Nachmittag zu verzeichnen. Den höchsten Fahrpreis erhält der Taxifahrer zwischen 4 und 5 Uhr morgens. Es könnte sein, dass die Menschen um diese Uhrzeit, um dem Verkehr zu umgehen, zum Airport fahren und der Taxifahrer dadurch ein höheres Einkommen erzielt.

Bundesfeiertage

Abbildung 3.24 zeigt die Anzahl der Fahrten und den Mittelwert des Fahrpreises über alle Tage im Jahr 2015.

Abbildung 3.24: Anzahl der Fahrten und Mittelwert des Fare Amounts verteilt über die Tage im Jahr 2015

In der Abbildung sind die Bundesfeiertage entsprechend gekennzeichnet. Die Feiertage haben einen direkten Einfluss auf das Fahrtaufkommen der Taxifahrten. An den Feiertagen ist die Anzahl der Fahrten häufig geringer und an den Tagen darauf ist ein signifikanter Anstieg zu verzeichnen. Die Taxifahrer verdienen tendenziell weniger als an normalen Tagen. Der Trend des niedrigeren Verdienstes an Feiertagen ist aber nicht immer einheitlich, weil die arbeitsfreie Tage eher vertraglich festgelegt und es von den Taxifahrern abhängig ist, ob sie arbeiten wollen oder nicht.

Räumliche Merkmale

In Abbildung 3.25 wird der Zusammenhang zwischen Fahrstrecke und Fahrpreis mithilfe eines Scatterplots dargestellt.

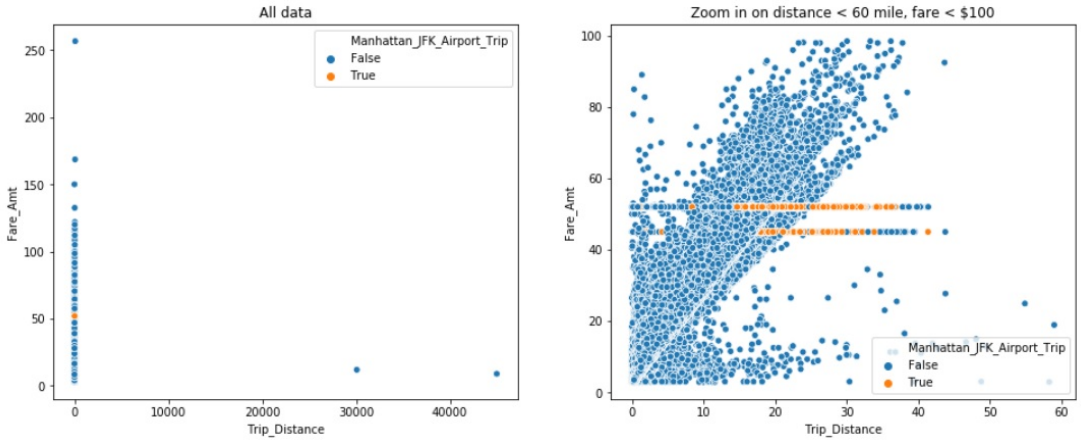


Abbildung 3.25: Trip Distance vs Fare Amount

Die Abbildung zeigt, dass die Fahrstrecke größtenteils mit den Fahrpreis linear korreliert. Drei Fälle sind typisch: kurze Fahrstrecken mit hohem Fahrpreis, hohe Fahrstrecken mit geringem Fahrpreis und die zwei horizontalen Linien bei 45 und 52 Dollar. Die zwei horizontalen Linien entsprechen hauptsächlich der Fahrt zwischen Manhattan und JFK Airport, weil die Strecke einer Flatrate von 45 Dollar vor Juli 2012 und nach Juli 2012 52 Dollar entsprechen [34]. Vermutlich gab es bei der Datenübertragung bei den Merkmalen Fahrstrecke und Fahrpreis Anomalien, die die tatsächlichen Werte verrauscht haben. Es wurde daher erprüft, ob die angenommene Fahrstrecke der Realität entspricht. Zu diesem Zweck wurde die Haversine-Distanz zwischen Abholort und Zustellungsort berechnet [35]. War die tatsächliche Reisestrecke kleiner als die Haversine-Distanz, wurde sie korrigiert und dementsprechend markiert. In diesem Fall wurden die Fahrtkosten ebenfalls korrigiert, falls der neu berechnete Fahrpreis mit der Haversine-Distanz größer war als der tatsächliche Fahrpreis. Es wurden auch Fahrpreise korrigiert, falls der tatsächliche Fahrpreis mit dem neu berechneten Fahrpreis mit der echten Fahrstrecke kleiner war.

Das Ergebnis der Korrektur zeigt Abbildung 3.26.

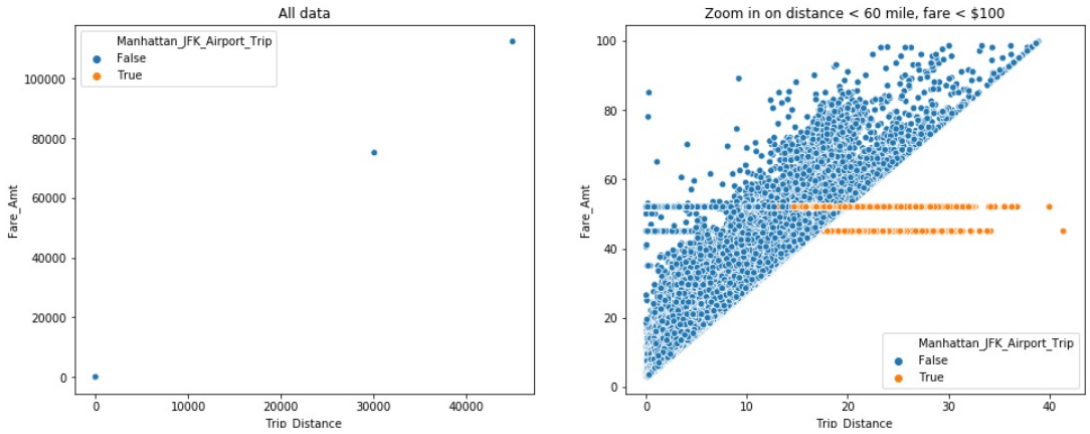


Abbildung 3.26: Trip Distance vs Fare Amount

Hier wurden die 2 Fälle kurze Fahrstrecken mit hohem Fahrtkosten und lange Fahrstrecken mit niedrigen Fahrtkosten größtenteils bereinigt.

Abbildung 3.27 zeigt die Anzahl der Fahrten und den Mittelwert des Fahrpreises der Abholungen für einzelne Bezirke.

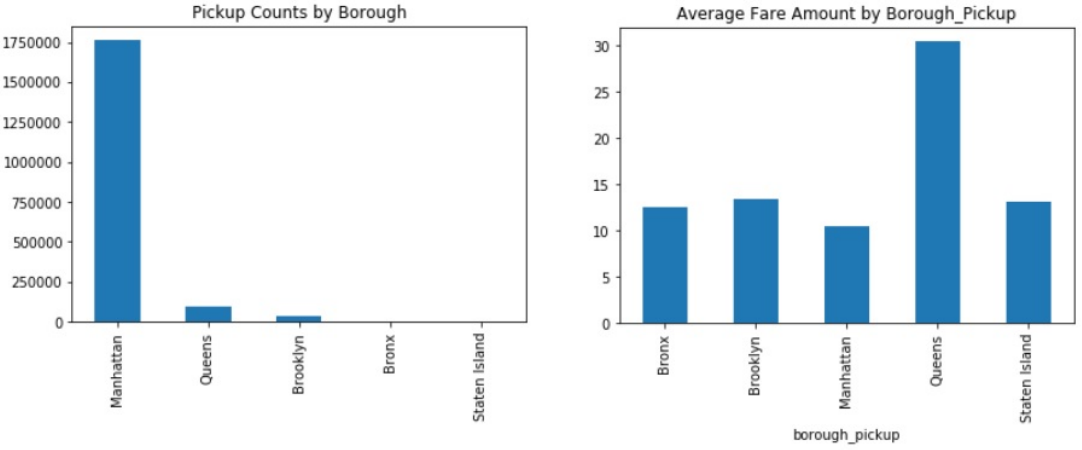


Abbildung 3.27: Anzahl der Fahrten und Mittelwert des Fahrpreises der Abholungen für einzelne Bezirke

Die meisten Abholungen finden in Manhattan statt. Den höchsten Fahrpreis erreicht man in Queens. Vermutlich ist die Flatrate zwischen Manhattan und JFK Airport der Grund für den hohen Fahrpreis.

Abbildung 3.28 zeigt die Anzahl der Fahrten und den Mittelwert des Fahrpreises der Zustellungen für einzelne Bezirke.

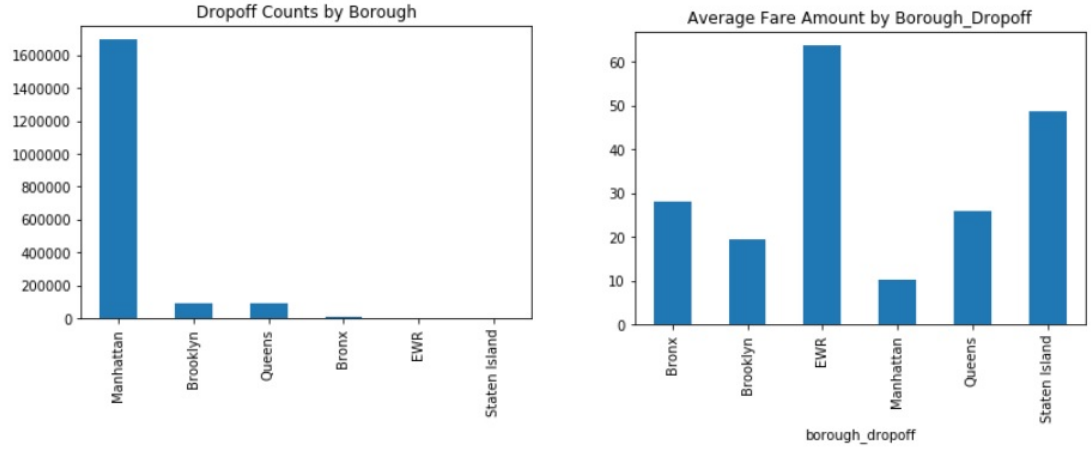


Abbildung 3.28: Anzahl der Fahrten und Mittelwert des Fahrpreises der Zustellungen für einzelne Bezirke

Die meisten Zustellungen werden in Manhattan abgeschlossen. Die höchsten Fahrtkosten werden mit Fahrten zum Newark Airport erreicht. Die Fahrtstrecke von Manhattan und Newark Airport ist lang und dadurch ist der Fahrpreis höher als auf anderen Strecken.

Abbildung 3.29 zeigt die Verteilung der Fahrpreise für einzelne Bezirke.

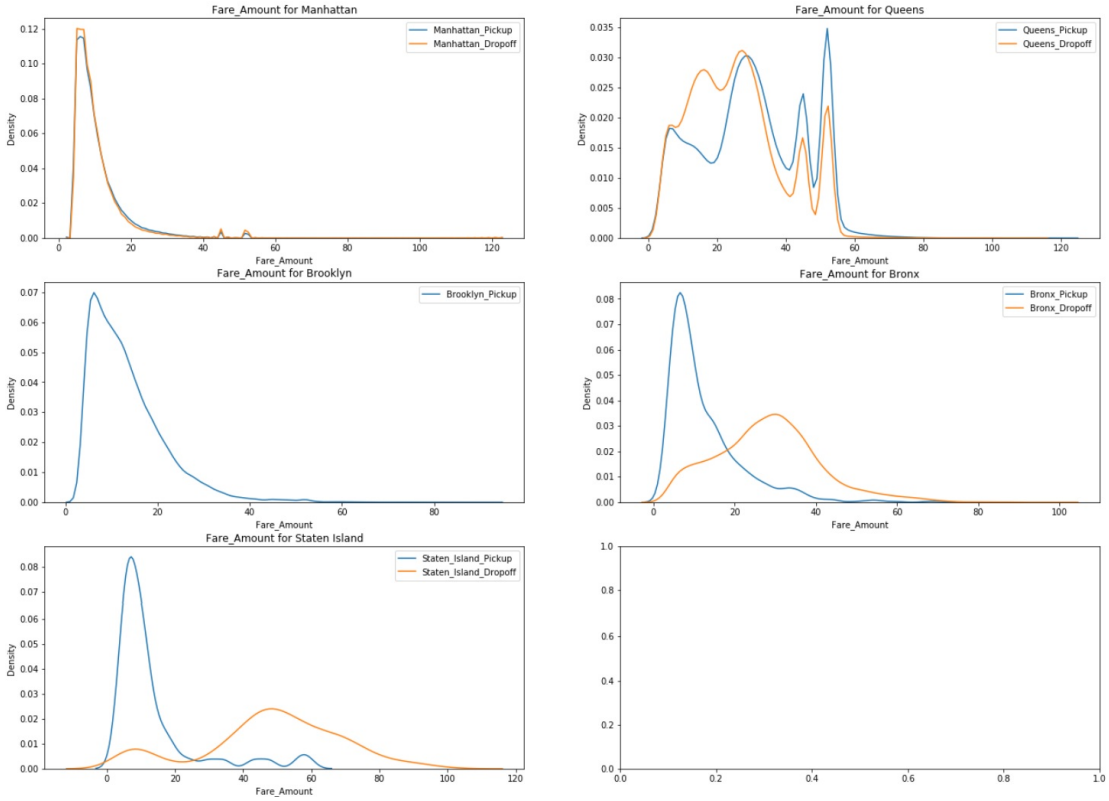


Abbildung 3.29: Verteilung der Fahrpreise für einzelne Bezirke

Es ist anzunehmen, dass die Verteilung der Fahrpreise für einzelne Bezirke unterschiedlich ist, außer in Manhattan. Bei einer Abholung in Queens sind die Fahrten tendenziell teurer als in den anderen Bezirken. Außerdem zeigt sich, dass sich in Staten Island die Fahrtkosten zwischen Abholung und Zustellung erheblich unterscheiden.

Abbildung 3.30 zeigt die Verteilung der Fahrpreise zwischen Lower Midtown Manhattan und dem übrigen Manhattan.

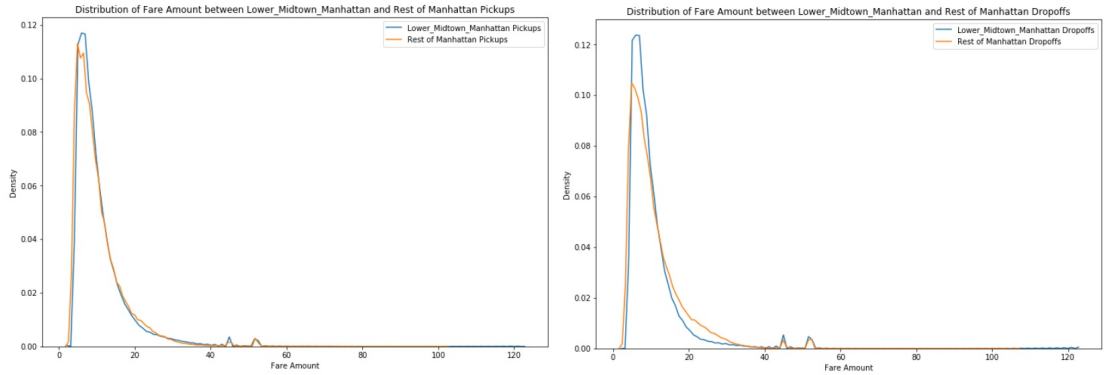


Abbildung 3.30: Verteilung der Fahrpreise zwischen Lower Midtown Manhattan und dem übrigen Manhattan

Innerhalb von Manhattan gibt es nur geringe Unterschiede in Bezug auf die Fahrpreise.

Wetterbedingte Merkmale

Da in diesem Anwendungsbeispiel die genaue Temperaturangabe nicht wichtig ist und anhand der Wetterdaten eine Tendenz analysiert werden soll, wurde die Durchschnittstemperatur am jeweiligen Tag mithilfe der Minimum- und Maximumtemperatur berechnet und mithilfe des Binnings in Temperaturkategorien eingeordnet. Tabelle 3.4 zeigt die Transformation der Average Temperature in Temperaturkategorien mit den jeweiligen Intervallen.

Intervall	Kategorie
(−26 °C) - (−13 °C)	cool
(−12.9 °C) - (0 °C)	little cool
(0.1 °C) - (20 °C)	pleasant
(20.1 °C) - (26 °C)	little warm
(−26.1 °C) - (32 °C)	warm
(32.1 °C) - (38 °C)	hot

Tabelle 3.4: Transformation der Average Temperature in Temperaturkategorien

Nach der Transformation wird in Abbildung 3.31 die Anzahl der Fahrten und der Mittelwert der Fahrpreise anhand der Temperaturkategorien dargestellt.

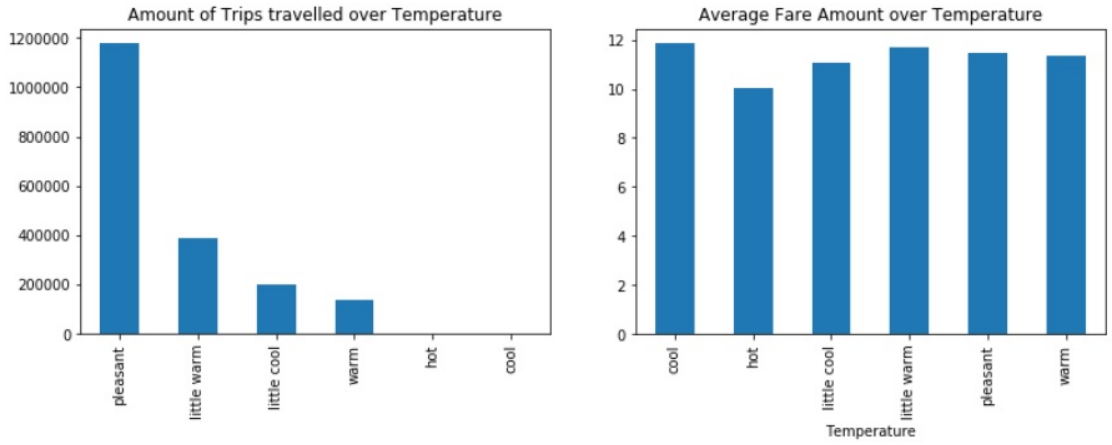


Abbildung 3.31: Anzahl der Fahrten und den Mittelwert der Fahrpreise anhand von Temperaturkategorien

Es gibt mehr Fahrten bei durchschnittlichen Temperturen als bei hohen oder niedrigen Temperaturen. Im Durchschnitt verdient der Taxifahrer an kalten Tagen mehr als an heißen oder angenehmen Tagen ohne wetterbezogene Widrigkeiten. Es ist anzunehmen, dass Menschen bei heißen oder angenehmen Tagen häufiger zu Fuß oder mit dem Fahrrad unterwegs sind.

Abbildung 3.32 zeigt die Anzahl der Fahrten und den Mittelwert der Fahrpreise anhand der Wetterlage.

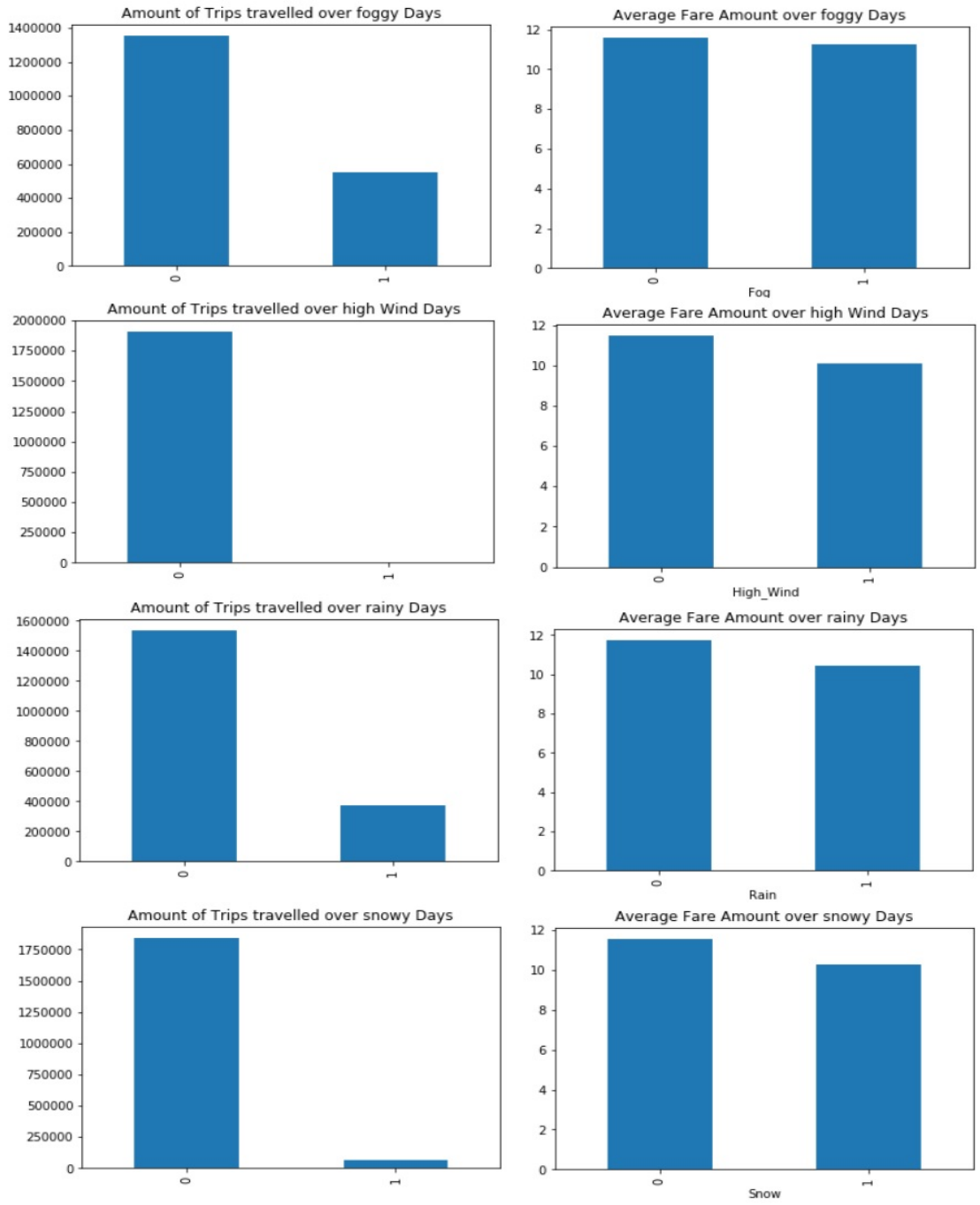


Abbildung 3.32: Anzahl der Fahrten und der Mittelwert der Fahrpreise anhand der Wetterlage

Die jeweiligen Wetterlagen wurden mit dem Datentyp boolean ermittelt. Bei allen Wetterlagen ist das Verkehrsaufkommen geringer als an normalen Tagen. Es wird vermutet,

dass die Wetterlagen extreme Wetterbedingungen beschreiben, die Einfluss auf das Verkehrsaufkommen haben und dadurch der Mittelwert des Fahrpreises geringer ist als unter normalen Wetterbedingungen.

Andere Merkmale

Abbildung 3.33 zeigt den Mittelwert der Fahrpreise in Bezug auf die Anzahl der Fahrgäste.

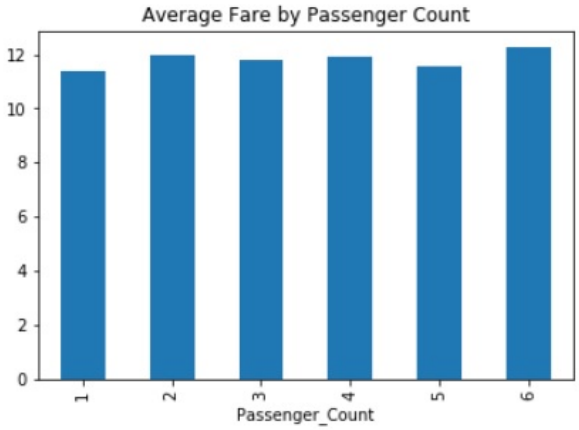


Abbildung 3.33: Mittelwert der Fahrtkosten in Bezug auf die Anzahl der Fahrgäste

Den höchsten Fahrpreis erreicht ein Taxifahrer mit 6 Fahrgästen.

Abbildung 3.34 zeigt die Verteilung der Fahrpreise in Bezug auf die Anzahl der Fahrgäste.

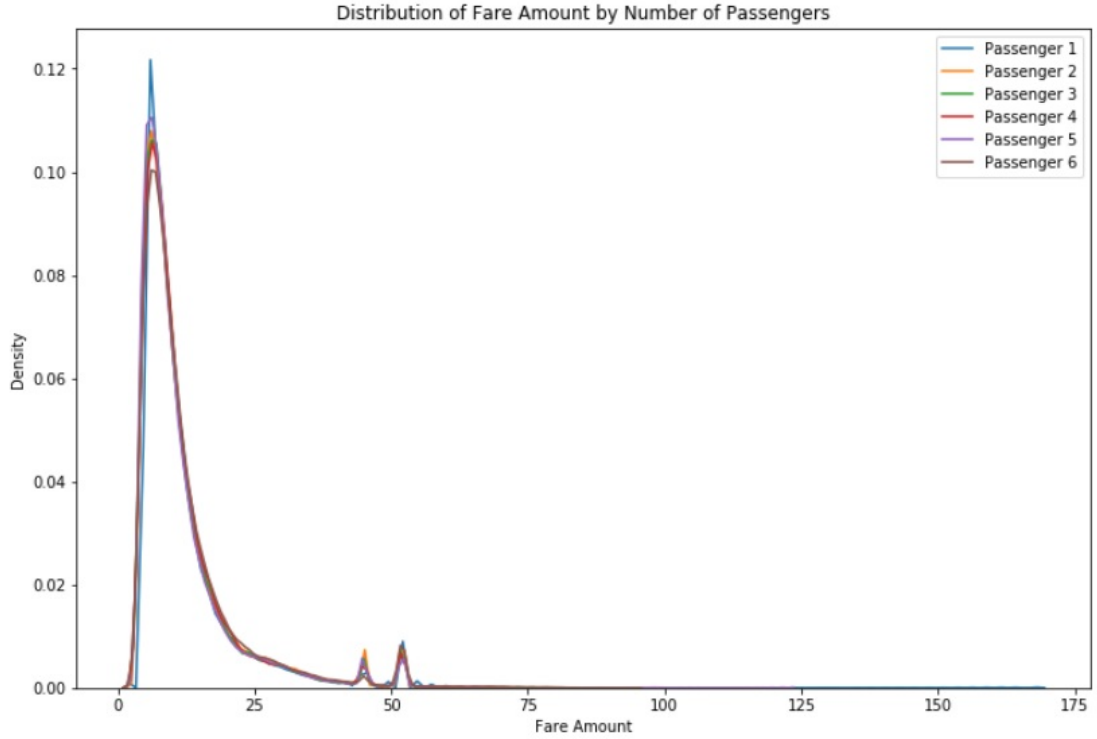


Abbildung 3.34: Verteilung der Fahrtkosten in Bezug auf die Anzahl der Fahrgäste

Bei der Verteilung der Fahrpreise in Bezug auf die Anzahl der Fahrgäste ist ein ähnlicher Verlauf zu erkennen.

3.4 Feature Engineering

> "Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data." — Dr. Jason Brownlee [36]

Nachdem die Zusammenhänge der Eingangsmerkmale mit dem Ausgabemerkmal Fare Amount untersucht worden sind, folgt das Feature Engineering. Das Feature Engineering dient dazu, neue relevante Eingangsmerkmale aus den existierenden Merkmalen zu generieren, um die Vorhersageleistung des Lernalgorithmus zu verbessern. Die neuen Merkmale zeigt Tabelle 3.5.

Feature Type	Feature Engineering	Data Type
Temporal	Pickup_Year	Numeric
	Pickup_Month	Numeric
	Pickup_Day	Numeric
	Season	Categorical
	Pickup_Weekday	Categorical
	Pickup_Hour	Numeric
	Time_Of_Day	Categorical
	Rush_Hour	Dichotomous
Spatio	borough_pickup	Categorical
	Pickup_Is_Area_Lower_Midtown_Manhattan	Dichotomous
	Distance_to_JFK_Airport	Numeric
	Distance_to_Newark_Airport	Numeric
	Distance_to_Rossville_Woodrow	Numeric
	Distance_to_rockaway_park	Numeric
Holiday	Holiday	Dichotomous
Weather	Temperature	Categorical
	Fog	Dichotomous
	High_Wind	Dichotomous
	Rain	Dichotomous
	Snow	Dichotomous

Tabelle 3.5: Feature Engineering und deren Datentypen

Zusätzlich zu zeitlichen Merkmalen wurden die Jahreszeit und die Frage, ob es sich um einen Wochentag oder um ein Wochenende handelt, die Tageszeit und die Frage, ob es sich um einen Rush Hour handelt, als Merkmal generiert. Sie sollten die zeitlichen Unterschiede feiner unterteilen, wobei das Merkmal Rush Hour ein Repräsentant für den Verkehr sein sollte. Bei den räumlichen Merkmalen steht zone_pickup für die Taxizonen, die eine feinere Unterteilung als die Boroughs gewährleisten. Es wurden Distanzen zu populären Orten mit hohen Fahrpreisen hinzugefügt, weil die Fahrstrecke eine lineare Korrelation zum Fahrpreis bildet.

4 Vorhersage des Taxifahrpreises

In diesem Kapitel wird die praktische Durchführung des KDD-Prozesses analysiert und dokumentiert. Abgeschlossen wird das Kapitel mit der Evaluierung der Ergebnisse und der Zusammenfassung.

4.1 Eingesetzte Software

In dieser Arbeit wurde größtenteils die Skriptsprache Python [37] in der Version 3.7 verwendet. Python bietet zahlreiche Bibliotheken von der Analyse bis hin zur Evaluierung der Datensätze. Für die KDD-Prozesse wurden mehrere Bibliotheken benutzt. Für die Datenselektion und Übersicht wurde die Bibliothek pandas [38] verwendet, um sie in Dataframes anzuzeigen. Im Anschluss an die Datenselektion wurden die Daten in der SQLite-Datenbank gespeichert. Für die Visualisierung der Daten wurden die Bibliotheken Matplotlib [39] und Seaborn [40] verwendet. Die Datenmanipulation wurde mithilfe von pandas realisiert. Für die Transformations-, Machine-Learning- und Evaluierungsphase wurde die Bibliothek Scikit-Learn [41] verwendet, die bereits vorimplementierte Machine-Learning-Algorithmen und Evaluierungsmethoden bereitstellt.

4.2 Datenselektion

In der Phase der Datenselektion wurden alle Features der Datensätze analysiert und mithilfe des zuvor erworbenen Domänenwissen bewertet, welches Feature den Taxifahrpreis möglicherweise beeinflussen könnte, und eine vollständige Taxifahrt beschreiben. Die Datensätze, die zur Vorhersage des Taxifahrpreises in New York City herangezogen wurden, werden in Abschnitt 3.1 detailliert beschrieben.

Die Bewertung, welche Features zur Vorhersage des Taxifahrpreises herangezogen wurden, ist aus dem Anhang zu entnehmen.

Nachdem die relevanten Features für jeden Datensatz bestimmt wurden, wurde der Yellow-Cab-Datensatz auf Syntax und Semantik geprüft, um eine einheitliche Datenform für die Datenvorverarbeitung zu gewährleisten (vgl. Abschnitt 3.3). Anschließend wurden die Datensätze in eine geeignete Datentabelle in der SQLite-Datenbank gespeichert. Für den Yellow-Cabs-Datensatz wurde jeweils für ein Jahr eine Datentabelle erstellt. Für den Wetter- und Feiertagsdatensatz wurde jeweils eine eigene Datentabelle erstellt.

4.3 Datenvorverarbeitung

In der Phase der Datenvorverarbeitung wird die Qualität der Daten geprüft und mit passenden Methoden verbessert. Die praktische Durchführung der Datenvorverarbeitung wurde in Abschnitt 3.3 detailliert beschrieben und dokumentiert. Im Folgenden werden die identifizierten fehlerhaften Daten für jeden Datensatz zusammengefasst:

Yellow-Cab-Datensatz

Die fehlenden Daten wurden wie folgt verarbeitet.

- Fehlende Daten lagen bei dem Features Drop-off Longitude, Latitude und der RateCodeID des Yellow-Cab-Datensatzes vor.
- Hier wurden die fehlende Werte der Drop-off Locations gelöscht, die nur einen geringen Anteil umfassen.
- Bei den RateCodeID wurden die fehlende Werte mit einer globalen Konstante ersetzt, weil die Einträge wichtige Informationen bereithalten.

Falsche Daten wurden wie folgt bearbeitet:

- Falsche Daten lagen ebenfalls nur in geringem Umfang vor. Sie wurden daher aus dem Datensatz entfernt.
- Nur angesichts des Features Trip_Distance und Fare_Amt wurden die Originalwerte durch approximierte Werte ersetzt, wenn die Originalwerte kleiner waren als die approximierten Werte.

Wetterdatensatz

Fehlende Daten wurden wie folgt bearbeitet:

- Bei den Wetterdaten war das Feature Average Temperature betroffen. Diese Information wurde aus dem Datensatz gelöscht und mit der Minimumtemperatur und Maximumtemperatur verrechnet, weil die Average-Temperatur keine Einträge enthält.

Falsche Daten wurden wie folgt bearbeitet:

- Ansonsten wurden die Wetterdaten als korrekt betrachtet.

Die Feiertagsdaten wurden aus der Kaggle-Plattform bereits aufbereitet. Sie wurden als korrekt betrachtet und deswegen nicht weiter geprüft.

4.4 Datentransformation

In der Phase der Datentransformation werden die vorverarbeiteten Daten in ein Format transformiert, das vom Machine-Learning-Algorithmus verarbeitet werden kann. Zusätzlich wurde zu den vorverarbeiteten Daten neue Features generiert (vgl. Abschnitt 3.4), die einen möglichen Einfluss auf den Taxifahrpreis haben. Die endgültigen Eingabefeatures enthalten Angaben zu Zeit und Ort, die neu generierten Features des Feature Engineering sowie Wetter- und Feiertagsdaten. Die Eingabefeatures und das Ausgabelabel zeigt Tabelle 4.1.

Features	Skalentyp	Transformation
Pickup_Year	Nominal	One-hot-Kodierung
Pickup_Month	Nominal	One-hot-Kodierung
Pickup_Day	Nominal	One-hot-Kodierung
Season	Nominal	One-hot-Kodierung
Pickup_Weekday	Nominal	One-hot-Kodierung
Pickup_Hour	Nominal	One-hot-Kodierung
Time_Of_Day	Nominal	One-hot-Kodierung
Rush_Hour	Nominal	One-hot-Kodierung
borough_pickup	Nominal	One-hot-Kodierung
Pickup_Is_Area_Lower_Midtown_Manhattan	Nominal	One-hot-Kodierung
Distance_to_JFK_Airport	Metric	Normalisierung
Distance_to_Newark_Airport	Metric	Normalisierung
Distance_to_Rossville_Woodrow	Metric	Normalisierung
Distance_to_rockaway_park	Metric	Normalisierung
Holiday	Nominal	One-hot-Kodierung
Temperature	Nominal	One-hot-Kodierung
Fog	Nominal	One-hot-Kodierung
High_Wind	Nominal	One-hot-Kodierung
Rain	Nominal	One-hot-Kodierung
Snow	Nominal	One-hot-Kodierung
Label	Skalentyp	Transformation
Fare_Rate	ordinal	Label-Encoding

Tabelle 4.1: Transformierung der Eingangsfeatures und des Labels

Auf der Seite der Eingabefeatures sind nominale und metrische Werte enthalten. Damit die Eingabefeatures für das Machine-Learning-Modell im passenden Format vorliegen, wurden die nominalen Werte mithilfe der One-Hot-Kodierung in binäre Werte umgewandelt. Die metrischen Werte wurden auf dem Intervall [0,1] normalisiert, damit die höheren Werte durch eine stärkere Gewichtung beim Machine-Learning-Algorithmus die Vorhersage nicht beeinflussen. Die Zielgröße Fare_Amt wurde vom metrischen Skalentyp auf einen ordinalen Skalentyp mithilfe der Clustering-Methode k-Means in eine neue Zielgröße Fare_Rate transformiert, weil während der Arbeit die Preisklasse für den Taxifahrer relevant ist und nicht die genaue Angabe des Fare_Amounts. Diese haben die Werte Low, Medium und High, die mithilfe der ordinalen Darstellung geordnet werden können.

Abbildung 4.1 zeigt die Einteilung der Fahrpreise anhand der jeweiligen Cluster.

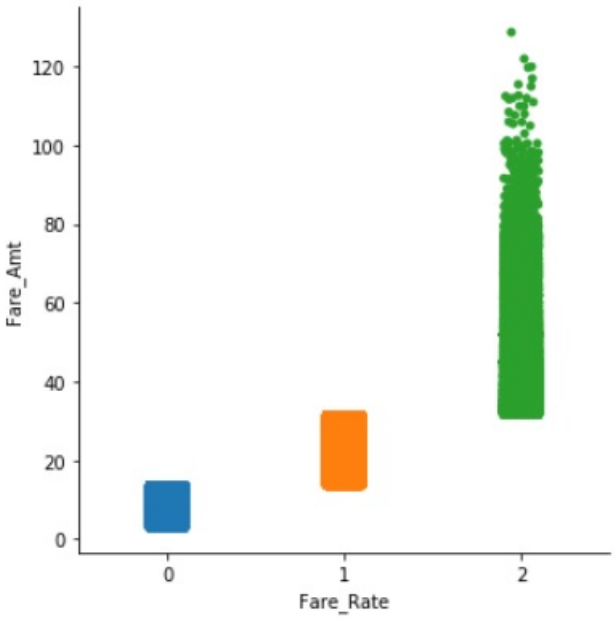


Abbildung 4.1: Einteilung des Fare_Amts in die Klassen Low, Medium und High

Es wurden 3 Cluster von 0 bis 2 erstellt. 0 steht für eine niedrigen Fahrpreis, 1 steht für ein mittleren Fahrpreis und 2 steht für ein hohen Fahrpreis. Die hohen Fahrpreise beginnen bei 32 Dollar und reichen bis 128 Dollar, obwohl wenige Einträge über 100 Dollar vorliegen. Die mittleren Fahrpreise beginnen bei 14 Dollar und reichen bis 32 Dollar. Die niedrigen Fahrpreise beginnen bei 3 Dollar und reichen bis 14 Dollar.

Nachdem die Transformationen abgeschlossen sind, wurden unausgeglichene Fahrpreisklassen festgestellt. Dieser Zusammenhang ist aus Abbildung 4.2 zu entnehmen.

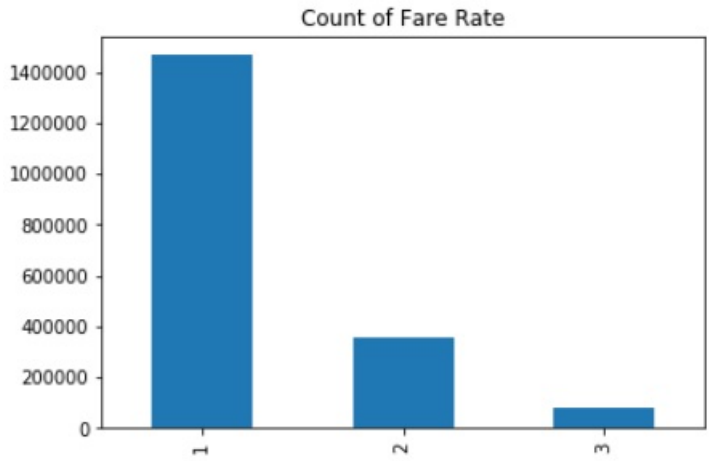


Abbildung 4.2: Unausgeglichene Aufteilung der Klassen

In Bezug auf die Aufteilung der Klassen besteht ein Ungleichgewicht, das möglicherweise einen Einfluss auf den Machine-Learning-Algorithmus hat, weil durch die vermehrten Beispiele die Klasse Low dazu tendiert, die Klasse 1 vorherzusagen. Entsprechende Methoden wie dem Problem begegnet werden kann, werden im folgendem Abschnitt vorgestellt.

4.5 Unausgeglichener Datensatz

Die ungleichmäßige Verteilung der Fahrpreisklassen könnte dazu führen, dass der Machine-Learning-Algorithmus Schwierigkeiten hat, die Muster der weniger repräsentativen Fahrpreisklassen zu erlernen [42]. Für die Überwindung des Ungleichgewichts gibt es mehrere Verfahren wie das Undersampling und Oversampling. Beim Undersampling wird die repräsentative Klasse auf die Größe der unterrepräsentierten Klasse reduziert. Beim Oversampling ist es andersherum, hier wird die unterrepräsentierte Klasse auf die Größe der repräsentativen Klasse erhöht. In der Arbeit von Batista u.a. [42] wurden Datensätze mit verschiedenen Over- und Undersampling-Methoden getestet und ausgewertet. Es wurde festgestellt, dass die Oversampling-Methoden bessere Ergebnisse liefern als die Undersampling-Methoden. In der Arbeit wird die Oversampling-Methode verwendet, weil beim Undersampling viele wichtige Datenpunkte verloren gehen und eine Erhöhung der Fahrpreisklasse Medium und High geeignet erscheint.

4.6 Machine Learning

In der Phase des Machine Learnings werden die ausgewählten Machine-Learning-Algorithmen (vgl. Abschnitt 2.3) für die transformierten Daten trainiert. Die Datenbasis in Tabelle 4.1 wird in Trainings- und Testdatensätzen in einem Verhältnis von 80 zu 20 % aufgeteilt. Damit die Machine-Learning-Algorithmen und der Einfluss des Oversamplings miteinander verglichen werden können, wurden bei den Trainings- und Testdatensätzen jeweils der gleiche random_state in dem Fall gleich eins gesetzt, damit die Teilung immer identisch bleibt. Die Trainingsdatensätze wurden in 6 Versionen aufbereitet.

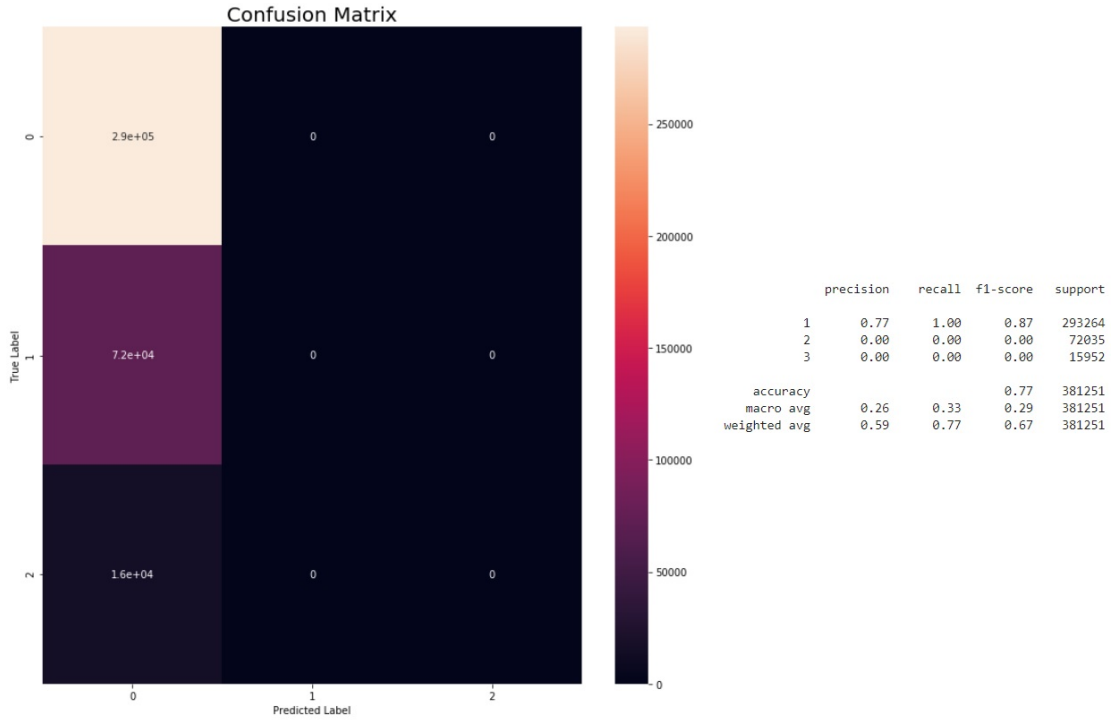
	Time & Location	Time & Location & Feature Engineering	Time & Location & Feature Engineering & external Data
imbalance	Version 1	Version 3	Version 5
oversampling	Version 2	Version 4	Version 6

Tabelle 4.2: Die 6 Versionen des Trainingsdatensatzes mit den jeweiligen Unterscheidungsmerkmalen

Aus Tabelle 4.2 wird ersichtlich, dass bei den Machine-Learning-Algorithmen jeweils 6 Modelle instanziiert werden müssen, die mit unterschiedlichen Ausprägungen in den Eingabefeatures und mit unausgeglichenen sowie ausgeglichenen Trainingsdatensätze trainiert werden. Deshalb werden die Trainingsdaten mit der gleichen Verteilung versehen, damit sie miteinander verglichen werden können. Für die hier vorliegenden unausgeglichenen Testdatensätze wurden die Metriken Precision, Recall und F1-Score verwendet. Die Accuracy wurde ausgeschlossen, weil bei einem unausgeglichenen Testdatensatz die Vorhersagequalität des Modells nicht bewerten werden kann.

4.6.1 Baseline

Bei der Baseline handelt es sich nicht um ein Machine-Learning-Algorithmus. Für die Klassifikation soll anhand der Baselines das Worst-Case-Szenario abgebildet werden, wenn der Klassifizierer nur die überrepräsentative Klasse vorhersagt. Die Baseline sagt immer die Klasse Low voraus. Abbildung 4.3 zeigt die Konfusionsmatrix und den Klassifizierungsbericht für den Testdatensatz.



	precision	recall	f1-score	support
1	0.77	1.00	0.87	293264
2	0.00	0.00	0.00	72035
3	0.00	0.00	0.00	15952
accuracy			0.77	381251
macro avg	0.26	0.33	0.29	381251
weighted avg	0.59	0.77	0.67	381251

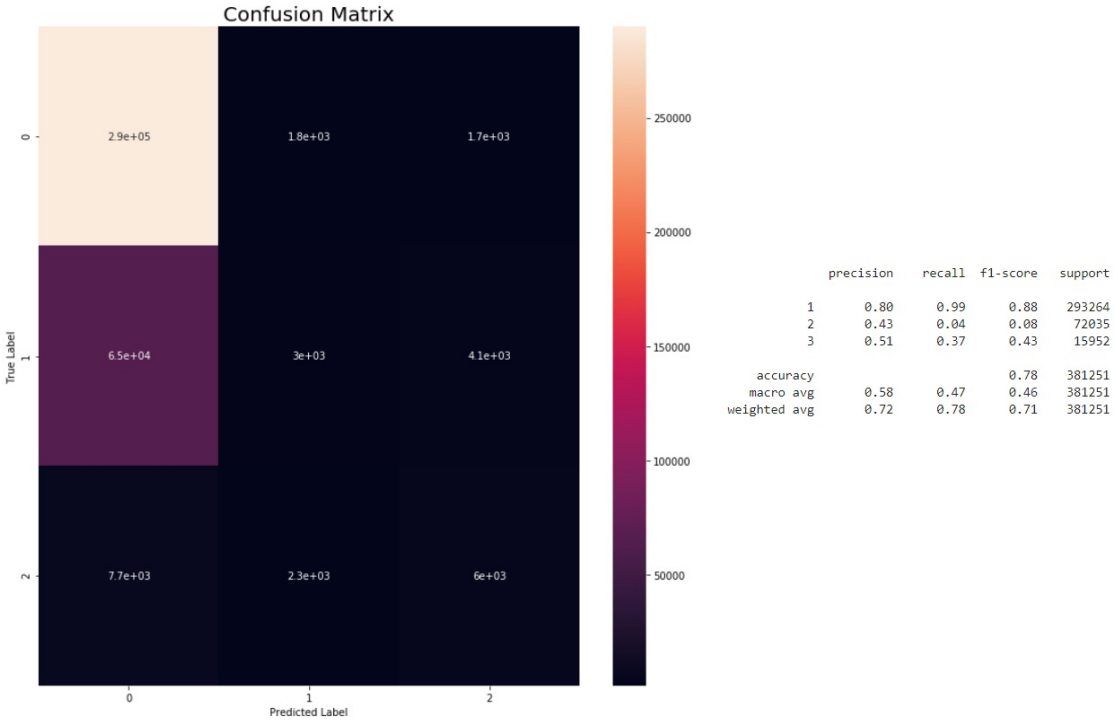
Abbildung 4.3: Konfusionsmatrix und Klassifizierungsbericht der Baseline für den Testdatendatensatz

Es wird ersichtlich, dass die Baseline-Methode ungeeignet ist. Sie hat nur eine Precision von 26 %, einen Recall von 33 % und einen F1-Score von 29 %. Diese Angaben gelten als Richtwert für die anderen Machine-Learning-Algorithmen.

4.6.2 Logistische Regression

Bei der logistischen Regression wurden in scikit-learn die Standareinstellung übernommen. Nur die Mindestanzahl der Iterationen wurde auf 800 Läufe hochgesetzt, weil sonst die Parameter nicht konvertieren würden.

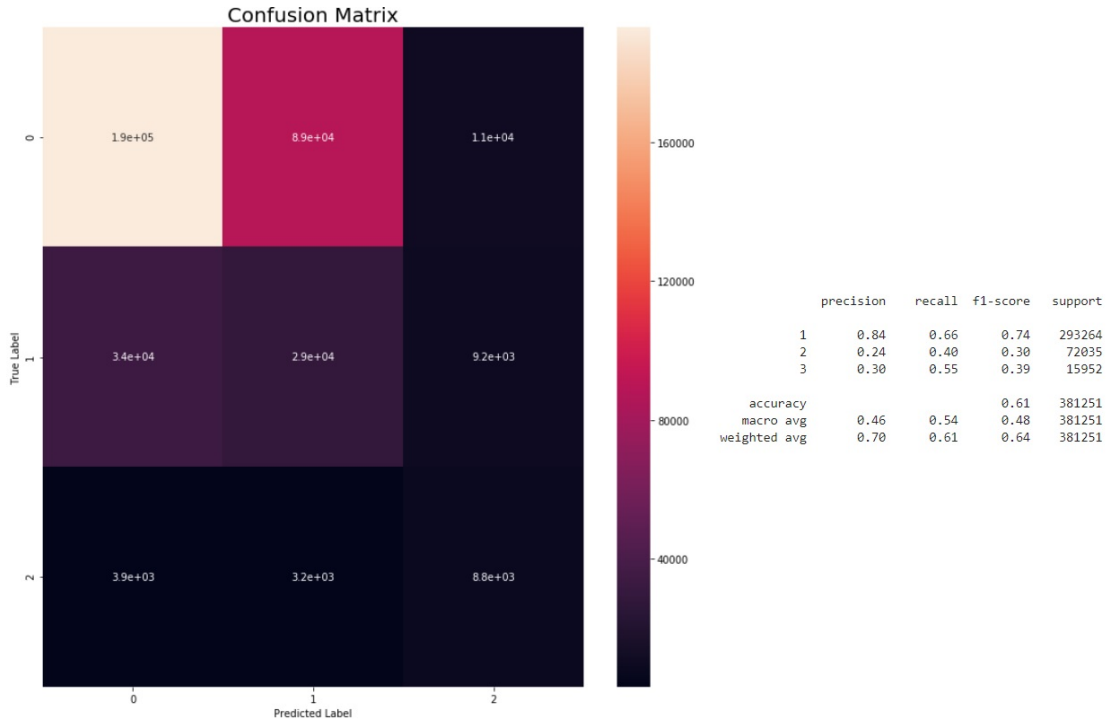
Eingabefeatures: Time und Location



	precision	recall	f1-score	support
1	0.80	0.99	0.88	293264
2	0.43	0.04	0.08	72035
3	0.51	0.37	0.43	15952
accuracy			0.78	381251
macro avg	0.58	0.47	0.46	381251
weighted avg	0.72	0.78	0.71	381251

Abbildung 4.4: Konfusionsmatrix und Klassifizierungsbericht der logistischen Regression für den Testdatensatz mit dem Trainingsdatensatz Version 1

Abbildung 4.4 zeigt die Konfusionsmatrix und den Klassifizierungsbericht der logistischen Regression für den Testdatensatz. Hierfür wurde das Modell mit dem unausgeglichenen Trainingsdatensatz trainiert. Das Modell erzielt eine Precision von 58 %, einen Recall von 47 % und einen F1-Score von 46 %.

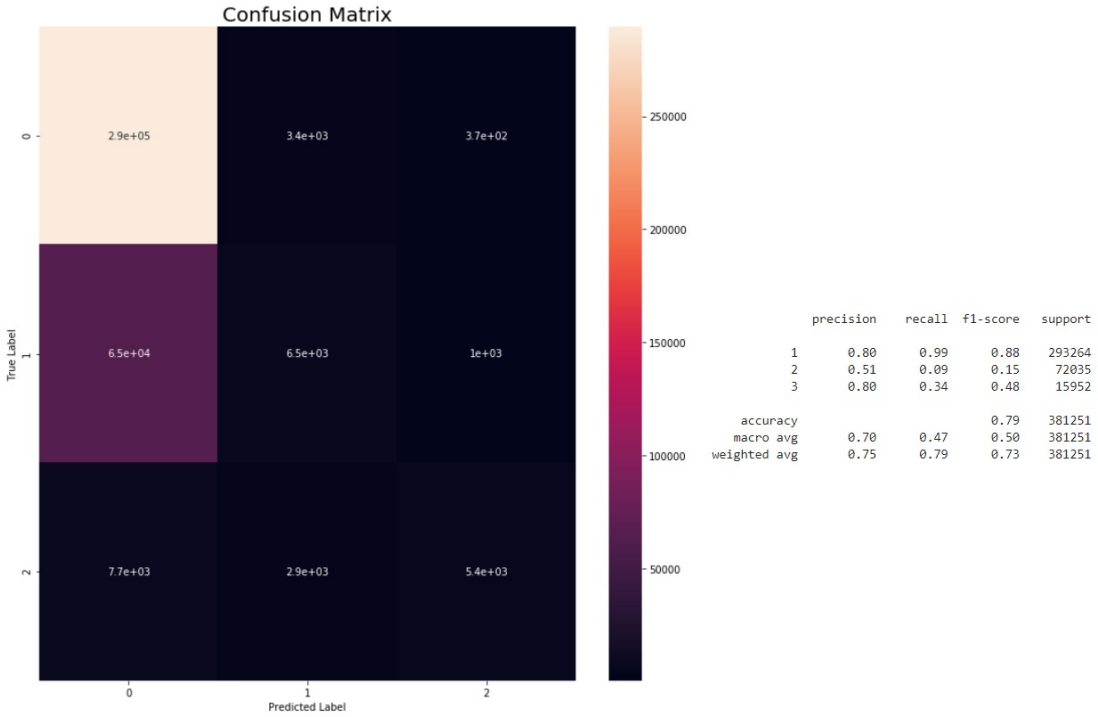


	precision	recall	f1-score	support
1	0.84	0.66	0.74	293264
2	0.24	0.40	0.30	72035
3	0.30	0.55	0.39	15952
accuracy			0.61	381251
macro avg	0.46	0.54	0.48	381251
weighted avg	0.70	0.61	0.64	381251

Abbildung 4.5: Konfusionsmatrix und Klassifizierungsbericht der logistischen Regression für den Testdatensatz mit dem Trainingsdatensatz Version 2

Abbildung 4.5 zeigt die Konfusionsmatrix und den Klassifizierungsbericht der logistischen Regression für den Testdatensatz. Hierfür wurde das Modell mit dem Oversampling-Trainingsdatensatz trainiert. Das Modell erzielt eine Precision von 46 %, einen Recall von 54 % und einen F1-Score von 48 %.

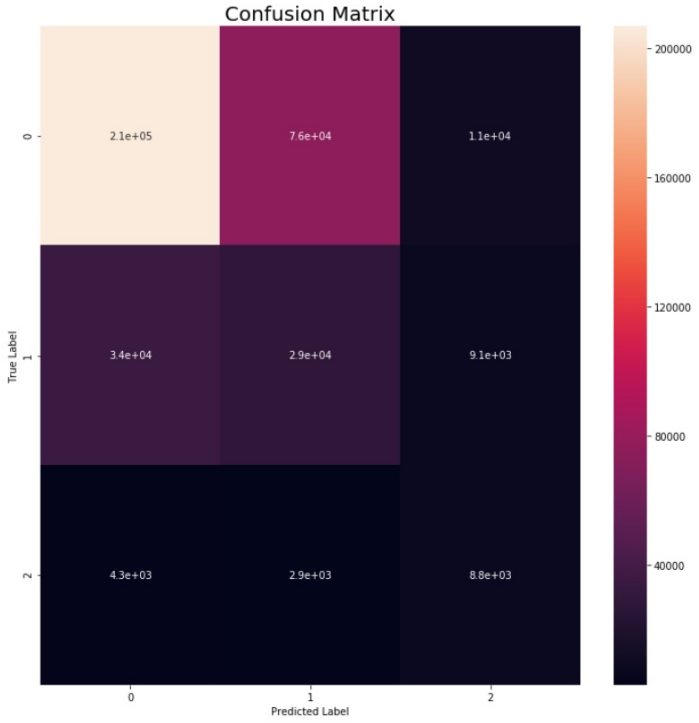
Eingabefeatures: Time und Location with Feature Engineering



	precision	recall	f1-score	support
1	0.80	0.99	0.88	293264
2	0.51	0.09	0.15	72035
3	0.80	0.34	0.48	15952
accuracy			0.79	381251
macro avg	0.70	0.47	0.50	381251
weighted avg	0.75	0.79	0.73	381251

Abbildung 4.6: Konfusionsmatrix und Klassifizierungsbericht der logistischen Regression für den Testdatensatz mit dem Trainingsdatensatz Version 3

Abbildung 4.6 zeigt die Konfusionsmatrix und den Klassifizierungsbericht der logistischen Regression für den Testdatensatz. Hierfür wurde das Modell mit dem unausgeglichenen Trainingsdatensatz trainiert. Das Modell erzielt eine Precision von 70 %, einen Recall von 47 % und einen F1-Score von 50 %.



	precision	recall	f1-score	support
1	0.84	0.70	0.77	293264
2	0.27	0.40	0.32	72035
3	0.30	0.55	0.39	15952
accuracy			0.64	381251
macro avg	0.47	0.55	0.49	381251
weighted avg	0.71	0.64	0.67	381251

Abbildung 4.7: Konfusionsmatrix und Klassifizierungsbericht der logistischen Regression für den Testdatensatz mit dem Trainingsdatensatz Version 4

Abbildung 4.7 zeigt die Konfusionsmatrix und den Klassifizierungsbericht der logistischen Regression für den Testdatensatz. Hierfür wurde das Modell mit dem Oversampling-Trainingsdatensatz trainiert. Das Modell erzielt eine Precision von 47 %, einen Recall von 55 % und einen F1-Score von 49 %.

Eingabefeatures: Time und Location with Feature Engineering und external Data

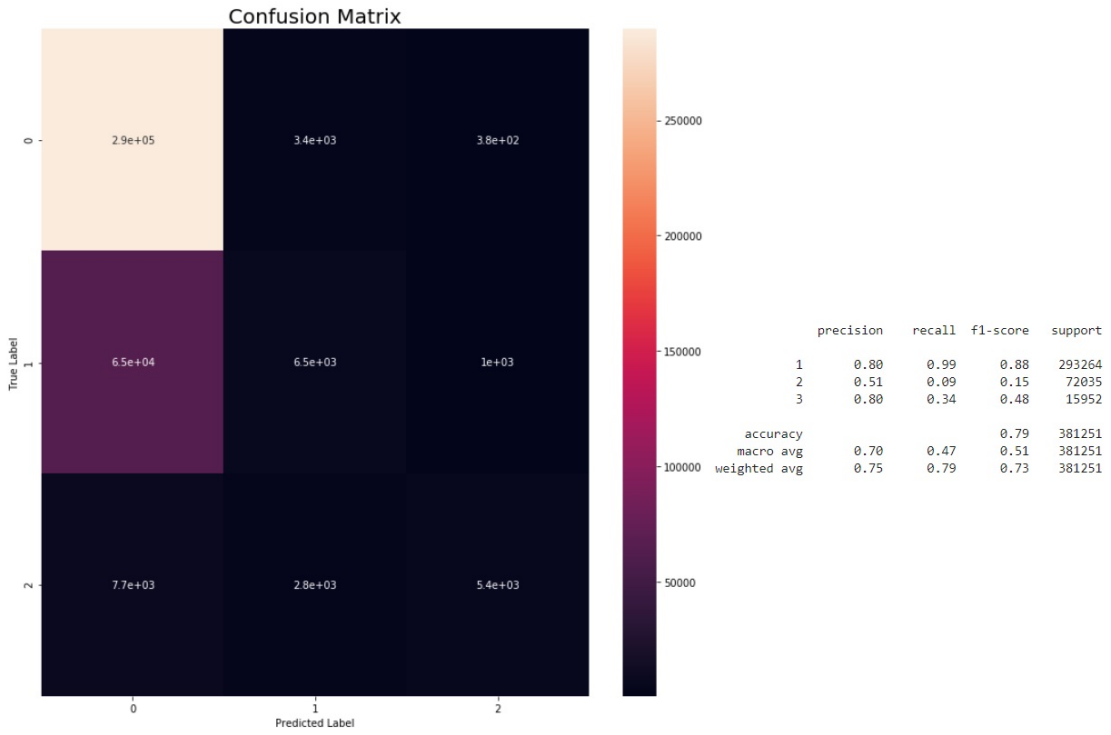
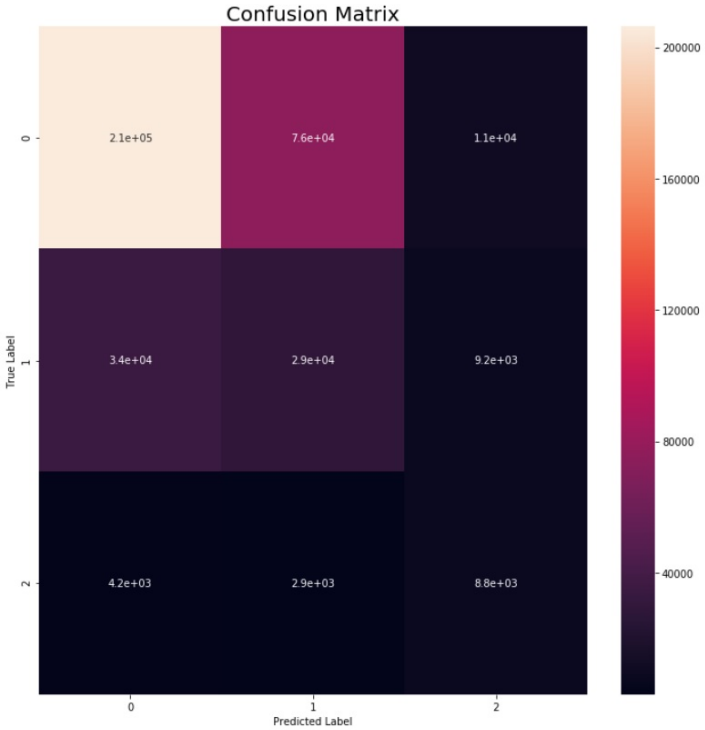


Abbildung 4.8: Konfusionsmatrix und Klassifizierungsbericht der logistischen Regression für den Testdatensatz mit dem Trainingsdatensatz Version 5

Abbildung 4.8 zeigt die Konfusionsmatrix und den Klassifizierungsbericht der logistischen Regression für den Testdatensatz. Hierfür wurde das Modell mit dem unausgeglichenen Trainingsdatensatz trainiert. Das Modell erzielt eine Precision von 70 %, einen Recall von 47 % und einen F1-Score von 51 %.



	precision	recall	f1-score	support
1	0.84	0.70	0.77	293264
2	0.27	0.40	0.32	72035
3	0.30	0.55	0.39	15952
accuracy			0.64	381251
macro avg	0.47	0.55	0.49	381251
weighted avg	0.71	0.64	0.67	381251

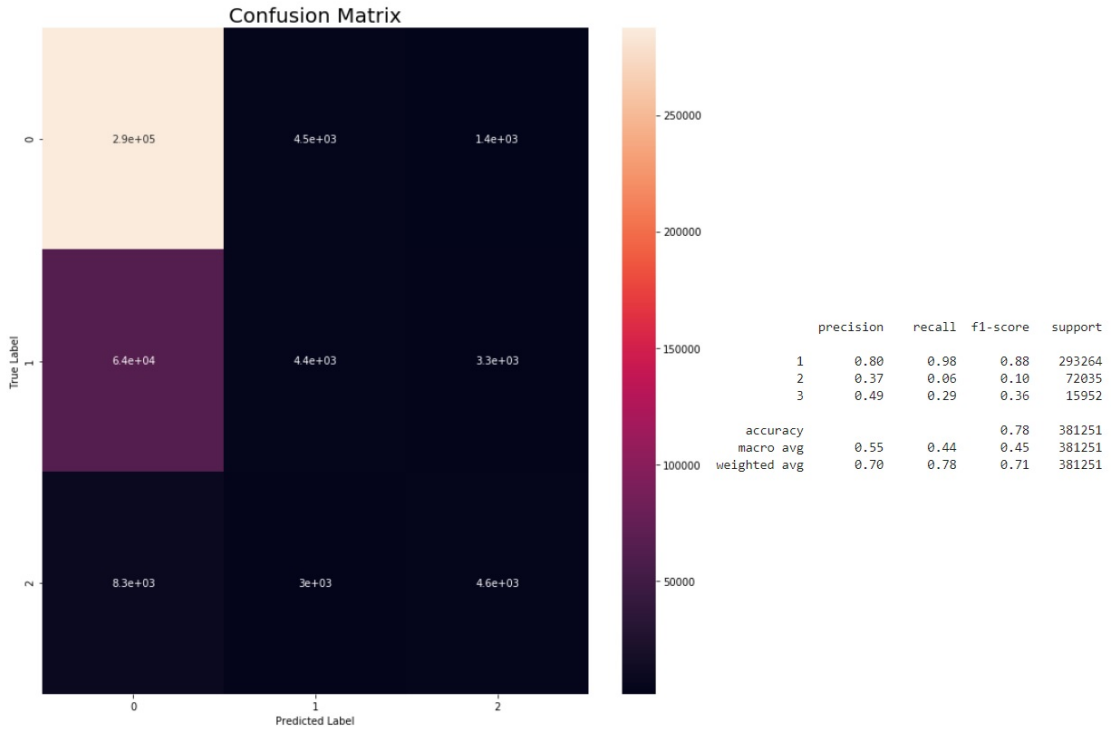
Abbildung 4.9: Konfusionsmatrix und Klassifizierungsbericht der logistischen Regression für den Testdatensatz mit dem Trainingsdatensatz Version 6

Abbildung 4.9 zeigt die Konfusionsmatrix und den Klassifizierungsbericht der logistischen Regression für den Testdatensatz. Hierfür wurde das Modell mit dem Oversampling-Trainingsdatensatz trainiert. Das Modell erzielt eine Precision von 47 %, einen Recall von 55 % und einen F1-Score von 49 %.

4.6.3 Random Forest

Beim Random Forest wurde in scikit-learn auch die Standareinstellung übernommen. Nur die Mindestanzahl an Bäumen wurde auf 40 gesetzt, um ein repräsentatives Ergebnis zu erhalten.

Eingabefeatures: Time und Location



	precision	recall	f1-score	support
1	0.80	0.98	0.88	293264
2	0.37	0.06	0.10	72035
3	0.49	0.29	0.36	15952
accuracy			0.78	381251
macro avg	0.55	0.44	0.45	381251
weighted avg	0.70	0.78	0.71	381251

Abbildung 4.10: Konfusionsmatrix und Klassifizierungsbericht des Random Forest für den Testdatensatz mit dem Trainingsdatensatz Version 1

Abbildung 4.10 zeigt die Konfusionsmatrix und den Klassifizierungsbericht des Random Forest für den Testdatensatz. Hierfür wurde das Modell mit dem unausgeglichenen Trainingsdatensatz trainiert. Das Modell erzielt eine Precision von 55 %, einen Recall von 44 % und einen F1-Score von 45 %.

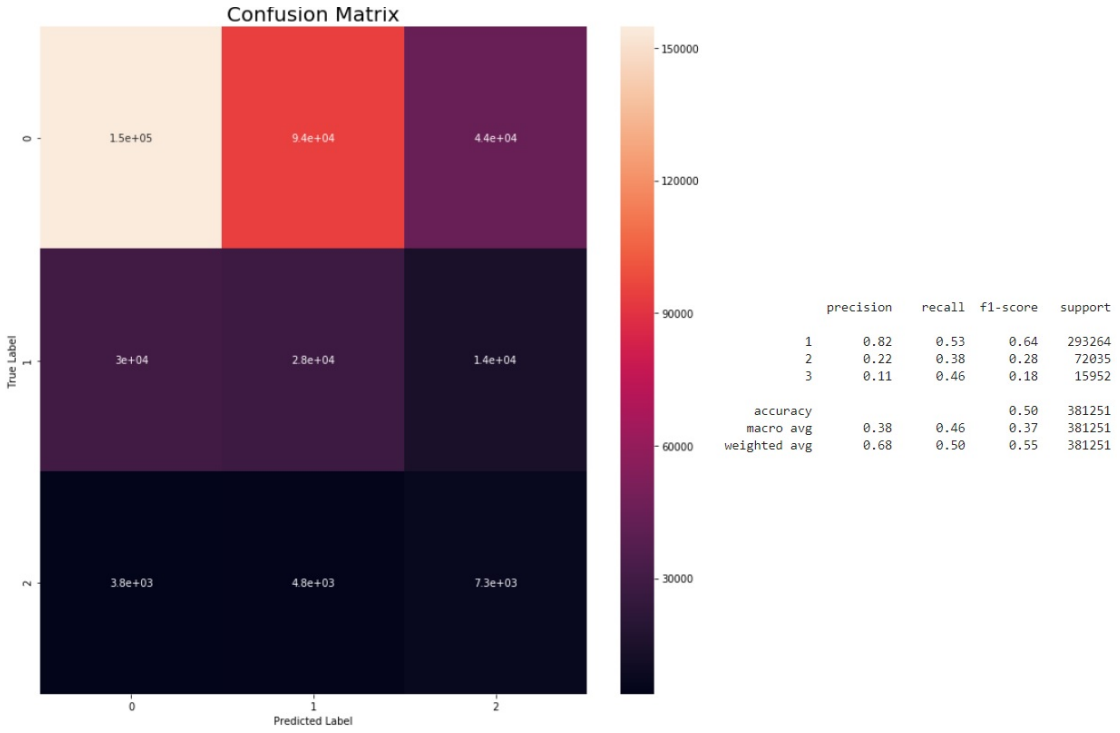


Abbildung 4.11: Konfusionsmatrix und Klassifizierungsbericht des Random Forest für den Testdatensatz mit dem Trainingsdatensatz Version 2

Abbildung 4.11 zeigt die Konfusionsmatrix und den Klassifizierungsbericht des Random Forest für den Testdatensatz. Hierfür wurde das Modell mit dem Oversampling-Trainingsdatensatz trainiert. Das Modell erzielt eine Precision von 38 %, einen Recall von 46 % und einen F1-Score von 37 %.

Eingabefeatures: Time und Location with Feature Engineering

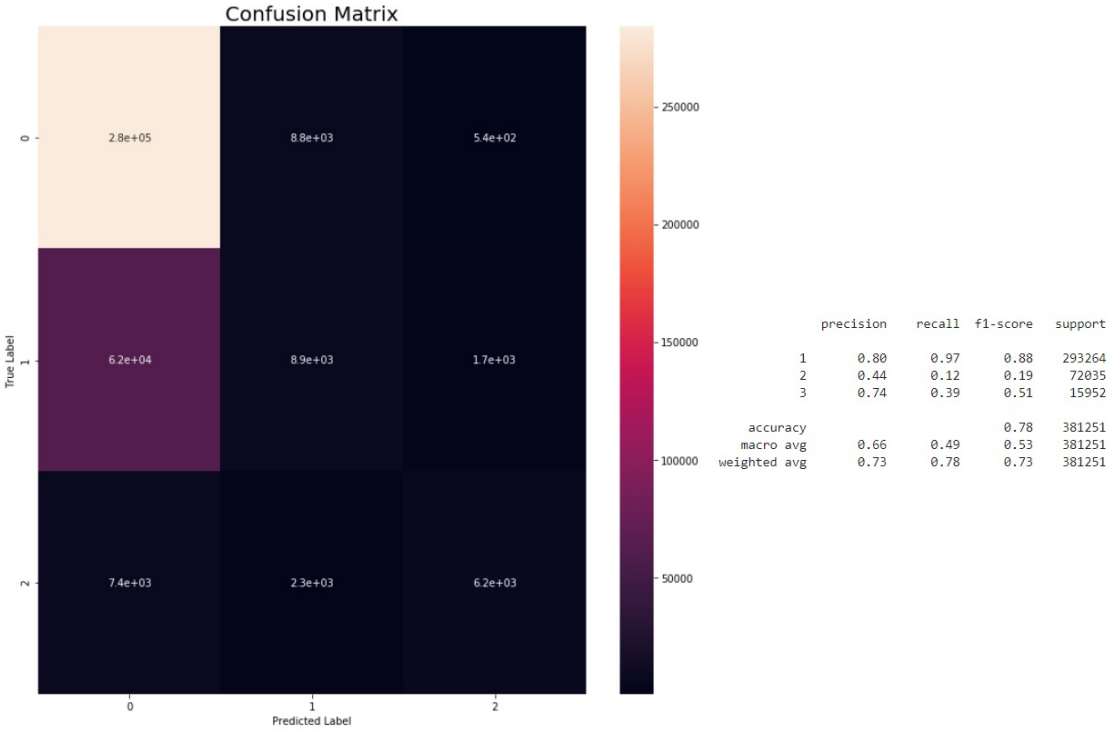
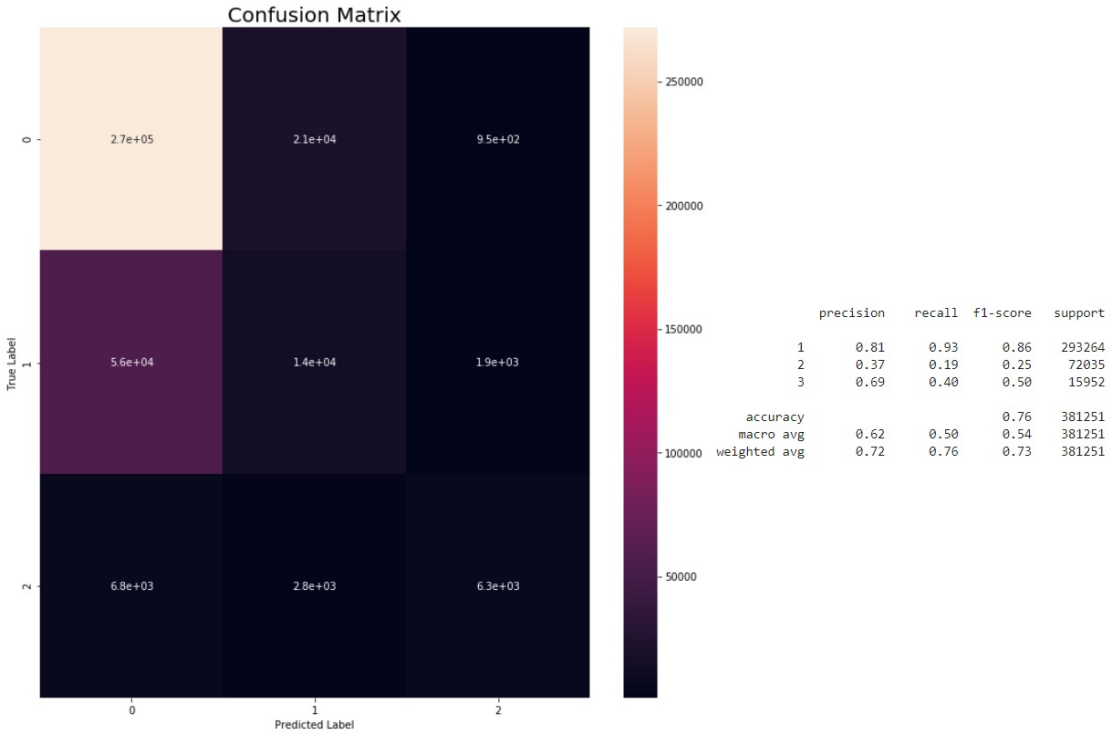


Abbildung 4.12: Konfusionsmatrix und Klassifizierungsbericht des Random Forest für den Testdatensatz mit dem Trainingsdatensatz Version 3

Abbildung 4.12 zeigt die Konfusionsmatrix und den Klassifizierungsbericht des Random Forest für den Testdatensatz. Hierfür wurde das Modell mit dem unausgeglichenen Trainingsdatensatz trainiert. Das Modell erzielt eine Precision von 66 %, einen Recall von 49 % und einen F1-Score von 53 %.

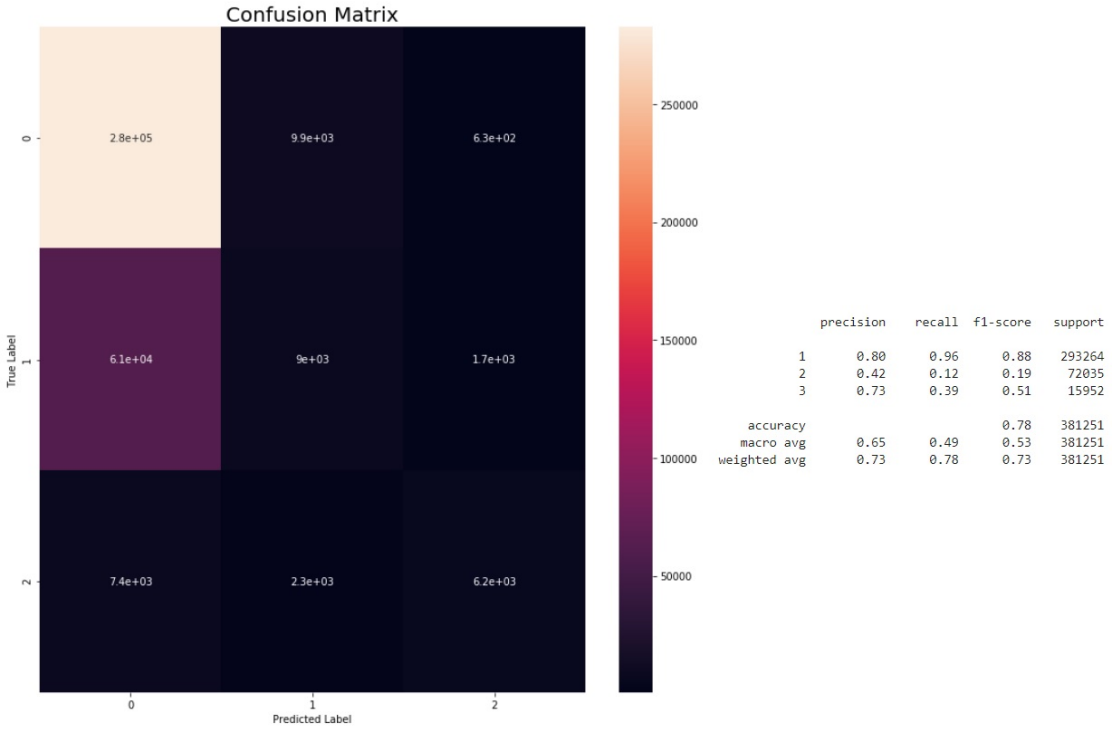


	precision	recall	f1-score	support
1	0.81	0.93	0.86	293264
2	0.37	0.19	0.25	72035
3	0.69	0.40	0.50	15952
accuracy			0.76	381251
macro avg	0.62	0.50	0.54	381251
weighted avg	0.72	0.76	0.73	381251

Abbildung 4.13: Konfusionsmatrix und Klassifizierungsbericht des Random Forest für den Testdatensatz mit dem Trainingsdatensatz Version 4

Abbildung 4.13 zeigt die Konfusionsmatrix und den Klassifizierungsbericht des Random Forest für den Testdatensatz. Hierfür wurde das Modell mitf dem Oversampling-Trainingsdatensatz trainiert. Das Modell erzielt eine Precision von 62 %, einen Recall von 50 % und einen F1-Score von 54 %.

Eingabefeatures: Time und Location with Feature Engineering und external Data



	precision	recall	f1-score	support
1	0.80	0.96	0.88	293264
2	0.42	0.12	0.19	72035
3	0.73	0.39	0.51	15952
accuracy			0.78	381251
macro avg	0.65	0.49	0.53	381251
weighted avg	0.73	0.78	0.73	381251

Abbildung 4.14: Konfusionsmatrix und Klassifizierungsbericht des Random Forest für den Testdatensatz mit dem Trainingsdatensatz Version 5

Abbildung 4.14 zeigt die Konfusionsmatrix und den Klassifizierungsbericht des Random Forest für den Testdatensatz. Hierfür wurde das Modell mit dem unausgeglichenen Trainingsdatensatz trainiert. Das Modell erzielt eine Precision von 65 %, einen Recall von 49 % und einen F1-Score von 53 %.

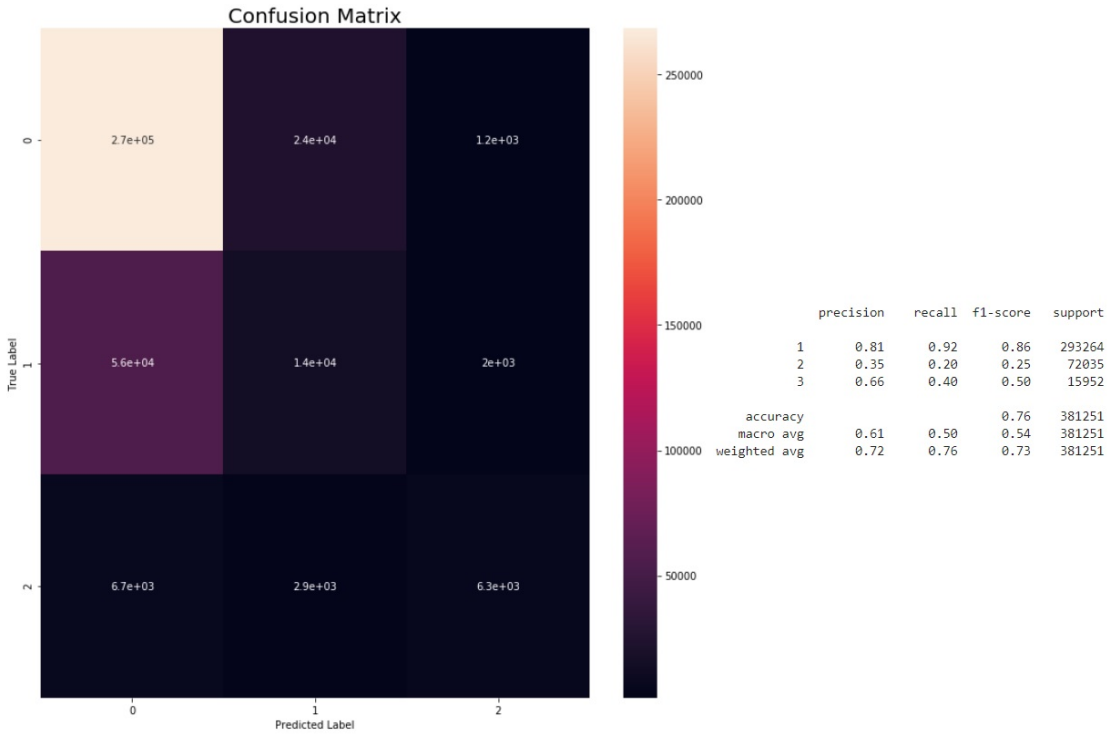


Abbildung 4.15: Konfusionsmatrix und Klassifizierungsbericht des Random Forest für den Testdatensatz mit dem Trainingsdatensatz Version 6

Abbildung 4.15 zeigt die Konfusionsmatrix und den Klassifizierungsbericht des Random Forest für den Testdatensatz. Hierfür wurde das Modell mit dem Oversampling-Trainingsdatensatz trainiert. Das Modell erzielt eine Precision von 61 %, einen Recall von 50 % und einen F1-Score von 54 %.

4.7 Evaluierung der Ergebnisse

Nachdem die Ergebnisse der verschiedenen Machine-Learning-Algorithmen, die auf unterschiedlichen Versionen der Trainingsdatensätze mit den Standardeinstellungen trainiert wurden, vorgestellt wurden, folgt nun in diesem Abschnitt die Bewertung der Ergebnisse. Die Ergebnisse des Experiments werden in Tabelle 4.3 zusammengefasst.

Algorithmus	Trainingsdaten	Features	Precision	Recall	F1-Score
Baseline	ohne Training	Zeit & Ort	26 %	33 %	29 %
Logistische Regression	unausgeglichen	Zeit & Ort	58 %	47 %	46 %
Logistische Regression	oversampled	Zeit & Ort	46 %	54 %	48 %
Logistische Regression	unausgeglichen	Zeit & Ort mit Feature Engineering	70 %	47 %	50 %
Logistische Regression	oversampled	Zeit & Ort mit Feature Engineering	47 %	55 %	49 %
Logistische Regression	unausgeglichen	Zeit & Ort mit Feature Engineering & externen Daten	70 %	47 %	51 %
Logistische Regression	oversampled	Zeit & Ort mit Feature Engineering & externen Daten	47 %	55 %	49 %
Random Forest	unausgeglichen	Zeit & Ort	55 %	44 %	45%
Random Forest	oversampled	Zeit & Ort	38 %	46 %	37 %
Random Forest	unausgeglichen	Zeit & Ort mit Feature Engineering	66 %	49 %	53 %
Random Forest	oversampled	Zeit & Ort mit Feature Engineering	62 %	50 %	54 %
Random Forest	unausgeglichen	Zeit & Ort mit Feature Engineering & externen Daten	65 %	49 %	53 %
Random Forest	oversampled	Zeit & Ort mit Feature Engineering & externen Daten	61 %	50 %	54 %

Tabelle 4.3: Vergleich der Machine-Learning-Algorithmen

Bei allen Modellen wurde eine Steigerung in den Metriken Precision, Recall und F1-Score in Bezug zur Baseline festgestellt. Bezüglich des Oversamplings weisen alle Modelle unterschiedliche Merkmale auf. In Bezug zur Precision schnitten die auf Oversampling trainierten Modelle schlechter ab als die mit den unausgeglichenen Trainingsdatensätzen trainierten Modelle. Beim Recall ist es genau umgekehrt. Hier schnitten die mit dem Oversampling-Trainingsdatensatz trainierten Modelle besser ab als die mit den unausgeglichenen Trainingsdatensätzen trainierten Modelle. Beim F1-Score ließ sich keine wirkliche Tendenz ermitteln. In Bezug auf das Feature Engineering und die Heranziehung der externen Daten sind signifikante Verbesserungen für alle Metriken vorhanden. In dieser Studie wird auf die Konzentration einer Metrikgröße verzichtet, weil eine allgemeine Sicht zur Bewertung des Modells erwünscht ist. Zum Beispiel würde man hier die Metrik Recall verwenden, wenn es um die korrekte Klassifizikation der gesamten Datenpunkte der Fare_Rate-Klassen ginge, und die Precision, wenn es um die Korrektheit der Vorhersage des Modells zu den Fare_Rate-Klassen ginge. Beim Precision schnitt die logistische Regression, die mit allen Features und den unausgeglichenen Trainingsdaten trainiert wurde, am besten ab. Beim Recall schnitt die logistische Regession, die mit allen Features und dem Oversampling-Trainingsdaten trainiert wurde, am besten ab. Beim F1-Score schnitt der Random Forest, der mit allen Features und den Oversampling-Traininsdaten trainiert wurde, am besten ab.

Es wird vermutet, dass durch die Heranziehung der Hyperparameteranpassung und die zielgerichtete Feature Selection die Ergebnisse verbessert werden können. Zusätzlich könnten aber auch mehr Daten zur Vorhersage des Taxifahrpreises verwendet und mehr relevante Features generiert werden. Die ermittelten Ergebnisse sollen in dieser Studie als Ansatzpunkt für das weitere Vorgehen dienen.

4.8 Zusammenfassung

In diesem Kapitel wird die praktische Durchführung entlang des KDD-Prozesses exemplarisch erläutert. Es wurden zwei Machine-Learning-Algorithmen angewendet, die deutliche Steigerungen in Bezug auf die Baseline für alle drei Metriken Precision, Recall und F1-Score verzeichnen. Es konnte bewiesen werden, dass mit dem Feature Engineering und mit der Heranziehung von externen Daten die Ergebnisse der Metriken verbessert werden können. In Bezug auf das Oversampling wurden verschiedene Auswirkungen auf

die Metriken erkannt. Nur beim Recall konnte eine steigende Tendenz aufgezeigt werden. Es wurden akzeptable Ergebnisse erzielt, die sicherlich noch mit weiteren Methoden verbessert werden könnten. Es hat sich in der Phase der Datenvorverarbeitung herausgestellt, dass die Realdaten mit einigen Problemen behaftet sind. Ein großes Problem sind falsche Daten oder Ausreißer, die man nicht ohne Domänenwissen beurteilen könnte. Diese Einschränkungen hat erhebliche Auswirkungen auf die Glaubhaftigkeit und Gültigkeit der Ergebnisse. Auch im Bereich des Feature Engineerings ist nicht immer klar, ob bestimmte Features eher zu einem Cluster zusammengefasst werden sollten oder nicht, weil bestimmte Features auch miteinander korrelieren können.

5 Fazit und Ausblick

Ziel dieser Studie ist es, die Probleme von Realdaten am Beispiel des New-York-City Yellow-Cab-Datensatzes aufzuzeigen, die Datenqualität zu steigern und neue Features zu generieren, die die Vorhersagequalität steigern. Außerdem sollte die Anwendbarkeit auf Vorhersagen des New-York-City Yellow-Cab-Datensatz mithilfe des KDD-Prozesses praktisch durchgeführt werden. Hier sollten die Einflüsse des Feature Engineerings, die Hinzufügung von externen Daten sowie unausgeglichene und Oversampling-Trainingsdatensätze mit den verschiedenen Algorithmen verglichen werden. In der Phase der Datenselektion und Datenvorverarbeitung zeigten sich Mängel in der Datenqualität, die sowohl syntaktische als auch semantische Ausprägungen haben. Syntaktische Ausprägungen wie wechselnde Datentypen im Laufe der Jahre oder Veränderungen der Spaltennamen wurden durch Vereinheitlichungen gelöst. Semantische Ausprägungen wie fehlende Werte wurden identifiziert und aufgrund ihrer geringen Anzahl aus dem Datensatz entfernt. Falsche Werte sind häufig komplexer, weil zur Identifikation der falschen Daten Domänenwissen vorausgesetzt wird. Es wurden falsche Daten aus dem Datensatz gelöscht, falls sie nur einen geringen Anteil am Datensatz ausmachten. Es wurden teilweise aber auch falsche Daten durch synthetische Daten ersetzt, weil sonst viel mehr Daten aus dem Datensatz gelöscht werden müssten. Ein Beispiel hierfür wären eine zu hohe Fahrstrecke oder zu hohe Fahrpreise. Hiermit haben wir in dieser Studie die Gültigkeit und Exakheit der Datenqualität verletzt. Die Verletzung der Kriterien Gültigkeit und Exakheit der Datenqualität war nicht vermeidbar, weil sonst entweder weniger Daten vorhanden wär oder falsche Daten in die Vorhersage einbezogen werden müssten. Die falschen Daten konnten durch Einbeziehung von anderen Features häufig identifiziert und gemeinsam behandelt werden sowie sich an den Beispielen Fahrtstrecke und Fahrpreis verdeutlichen lässt, weil sie einen linearen Zusammenhang aufweisen. In der Phase der Datentransformation wurden nach der Exploration der Daten neue Features generiert, die Einfluss auf den Fare_Amount haben. Dazu zählen zum Beispiel die Distanzen zu Orten, die höhere Fahrpreise erwarten lassen oder die Hinzufügung von externen Datenquellen wie das Wetter (vgl. Abschnitt 3.4). Es hätten auch Features wie die Distanzen zu Orten mit

geringen Fahrpreisen interessiert, damit das Modell eine stärkere Einteilung von großen und kleinen Fahrpreisen in den Daten erhält. Es wurden die nominalen Eingabefeatures mithilfe der One-Hot-Kodierung in eine binäre Form umgewandelt und metrische Werte normalisiert, damit die höheren Werte nicht automatisch eine stärkere Gewichtung erlangen. Nach der Transformation des Fare_Amounts von einer metrischen zu einer ordinalen Skala wurde ein Ungleichgewicht zwischen den Klassen festgstellt. Es wurde versucht dieses Problem mithilfe des Oversamplings zu beheben, der die unterepräsentative Klasse auf die Ebene der überepräsentativen Klasse mit Kopien der unterpräsentativen Klasse anreichert. In der Phase des Machine Learning wurden mehrere Versionen der Trainingsdatensätze generiert, um den Einfluss der unausgeglichene Datensätze und der Oversampling-Datensätze sowie des Feature Engineerings und externe Daten zu vergleichen. Durch die Einbeziehung von externen Daten und Feature Engineering wurde die Vorhersagequalität signifikant verbessert. Auf Seiten des Oversamplings wurde wechselnde Auswirkungen auf den Metriken erkannt (vgl. 4.7).

Die hier durchgeführten Experimente haben wertvolle Erkenntnisse geliefert und sollen als Grundlage für das weitere Arbeiten mit dem New-York-City-Yellow-Cab-Datensatz dienen. Diese Studie wurde auf einem lokalen Rechner ausgeführt und unterliegt deshalb einigen technischen Restriktion. Durch die Ausführung auf leistungsstärkeren Servern oder GPUs könnten mehr Daten für das Training hinzugefügt werden und auch mehr Durchläufe vermutlich genauere Ergebnisse erzielen. Anstatt den Dimensionsraum zu vergrößern, könnte eine Dimensionsreduktion sinnvoll sein. Hier könnte das Feature Selection eine Abhilfe schaffen, um den Merkmalsraum auf die wichtigsten Features zu reduzieren, und dem Modell dadurch weniger komplexe Daten zur Verfügung stehen. Zusätzlich könnte durch Hinzufügung von weiteren Features wie die Distanzen zu den Orten mit niedrigen Fahrpreisen die Einteilung in den Klassen weiter verfeinert werden. Durch die zielgerichtete Hyperparameteranpassung könnte das Modell mithilfe des Gridsearchs eine höhere Genauigkeit erreichen. Zusätzlich könnten komplexere Modelle wie das Long-Short-Term-Memory (LSTM) oder auch einfache neuronale Netze zu einer höheren Genauigkeit führen.

Literaturverzeichnis

[1] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):41, 1996.

[2] Ayush Pant. Introduction to logistic regression, 2019. URL https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148.

[3] T.A. Runkler. *Data Mining: Modelle und Algorithmen intelligenter Datenanalyse*, page V. Computational Intelligence. Springer Fachmedien Wiesbaden, 2015.

[4] F.Tenzer. Prognose zum volumen der jährlich generierten digitalen datenmenge weltweit in den jahren 2018 und 2025, 2019. URL https://de.statista.com/statistik/daten/studie/267974/umfrage/prognose-zum-weltweit-generierten-datenvolumen/.

[5] Andreas C Müller and Sarah Guido. *Einführung in Machine learning mit Python: Praxiswissen data science*, page 1. O'Reilly, 2017.

[6] Gill Press. Cleaning big data: Most time-consuming, least enjoyable data science task, survey says, 2020. URL https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-su#1d1005e96f63.

[7] NYC Taxi Limousine Commission. Tlc trip record data, 2019. URL https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

[8] Patrick Wagner. Ride-hailing apps surpass regular taxis in nyc, 2019. URL https://www.statista.com/chart/13480/ride-hailing-apps-surpass-regular-taxis-in-nyc/.

[9] Susanne Ehneß. Die meisten menschen leben in städten, 2019. URL https://www.egovernment-computing.de/die-meisten-menschen-leben-in-staedten-a-731941/.

[10] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37, 1996.

[11] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):40–41, 1996.

[12] Jürgen Cleve and Uwe Lämmel. *Data Mining*. De Gruyter Studium. De Gruyter Oldenbourg, Berlin, 2. auflage edition, 2016.

[13] Jürgen Cleve and Uwe Lämmel. *Data Mining*. De Gruyter Studium. 2. auflage edition.

[14] Sunila Gollapudi. *Practical machine learning*. Community experience distilled. Packt Publishing, Birmingham, UK, online-ausg. edition, 2016. URL http://proquest.tech.safaribooksonline.de/9781784399689.

[15] E. ANDERSON. The irises of the gaspe peninsula. *Bull. Am. Iris Soc.*, 59:2–5, 1935. URL https://ci.nii.ac.jp/naid/10000141584/en/.

[16] Cha Zhang and Yunqian Ma. *Ensemble machine learning: methods and applications*. Springer, 2012.

[17] Christophoros Antoniades, Delara Fadavi, and Antoine Foba Amon Jr. Fare and duration prediction: A study of new york city taxi rides. 2016.

[18] Cyril Cordor, Matthew Sims, and Eung Keun Kim. New york taxi fare prediction. 2018.

[19] Monalisha Ojha, Nisha Rani, and Ankit Tewari. Taxi fare prediction. 2019.

[20] Rishabh Upadhyay and Simon Lui. Taxi fare rate classification using deep networks.

[21] Paul Jolly, Boxiao Pan, and Varun Nambiar. Caesar's taxi prediction services predicting nyc taxi fares, trip distance, and activity.

[22] NYC Taxi Limousine Commission. Nyc taxi limousine commission, 2019. URL https://www1.nyc.gov/site/tlc/index.page.

[23] Wikipedia. National climatic data center, 2019. URL https://de.wikipedia.org/wiki/National_Climatic_Data_Center.

[24] NOAA. National centers for environmental information, 2019. URL https://www.ncdc.noaa.gov/.

[25] Snehaa Ganesan. Federal holidays usa 1966-2020, 2019. URL https://www.kaggle.com/gsnehaa21/federal-holidays-usa-19662020#usholidays.csv.

[26] NYC Gov. Open data for all new yorkers, 2019. URL https://opendata.cityofnewyork.us/.

[27] Austin W Smith, Andrew L Kun, and John Krumm. Predicting taxi pickups in cities: which data sources should we use? In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pages 380–387. ACM, 2017.

[28] Jun Xu, Rouhollah Rahmatizadeh, Ladislau Bölöni, and Damla Turgut. Taxi dispatch planning via demand and destination modeling. In *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, pages 377–384. IEEE, 2018.

[29] Hongjian Wang, Xianfeng Tang, Yu-Hsuan Kuo, Daniel Kifer, and Zhenhui Li. A simple baseline for travel time estimation using large-scale trip data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):19, 2019.

[30] NYC Taxi Limousine Commission. Factbook, 2019. URL https://www1.nyc.gov/site/tlc/about/fact-book.page.

[31] M Anil Yazici, Camille Kamga, and Abhishek Singhal. A big data driven model for taxi drivers' airport pick-up decisions in new york city. In *2013 IEEE International Conference on Big Data*, pages 37–44. IEEE, 2013.

[32] William Lidwell, Kritina Holden, and Jill Butler. *Universal principles of design, revised and updated: 125 ways to enhance usability, influence perception, increase appeal, make better design decisions, and teach through design*. Rockport Pub, 2010.

[33] Jürgen Cleve and Uwe Lämmel. *Data Mining*, page 202. De Gruyter Studium. De Gruyter Oldenbourg, Berlin, 2. auflage edition, 2016.

[34] Memo. How to transfer between jfk airport and manhattan?, 2019. URL https://www.new-york-city-travel-tips.com/how-transfer-between-jfk-airport-manhattan/.

[35] Wikipedia. How to transfer between jfk airport and manhattan?, 2019. URL https://en.wikipedia.org/wiki/Haversine_formula.

[36] Dr. Jason Brownlee. Discover feature engineering, how to engineer features and how to get good at it, 2019. URL https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good

[37] Python. python, 2019. URL https://www.python.org/.

[38] pandas. Python data analysis library, 2019. URL https://pandas.pydata.org/.

[39] John Hunter u.a. matplotlib, 2019. URL https://matplotlib.org/.

[40] Michael Waskom. seaborn: statistical data visualization, 2019. URL https://seaborn.pydata.org/.

[41] scikit learn. scikit-learn machine learning in python, 2019. URL https://scikit-learn.org/stable/.

[42] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.

A Anhang

A.1 Beschreibung der Datentabellen und Angaben zu den Datentypen

A.1.1 Yellow-Cab-Datensatz

Field Name	Description	Data Type
vendor_name	A designation for the technology vendor that provided the record. CMT=Creative Mobile Technologies VTS= VeriFone, Inc. DDS=Digital Dispatch Systems	String
Trip_Pickup_DateTime	The date and time when the meter was engaged.	Datetime
Trip_Dropoff_DateTime	The date and time when the meter was disengaged.	Datetime
Passenger_Count	The number of passengers in the vehicle. This is a driver-entered value.	Integer
Trip_Distance	The elapsed trip distance in miles reported by the taximeter.	Float
Start_Lon	Longitude where the meter was engaged.	Float
Start_Lat	Latitude where the meter was engaged.	Float
Rate_Code	The final rate code in effect at the end of the trip. 1= Standard rate 2=JFK 3=Newark 4=Nassau or Westchester 5=Negotiated fare 6=Group ride	String
store_and_forward	This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka "store and forward," because the vehicle did not have a connection to the server. Y= store and forward trip N= not a store and forward trip	String
End_Lon	Longitude where the meter was disengaged.	Float
End_Lat	Latitude where the meter was disengaged.	Float
Payment_Type		String
Fare_Amt	The time-and-distance fare calculated by the meter.	Float
surcharge	Miscellaneous extras and surcharges. Currently, this only includes the 0.50 Dollar and 1 Dollar rush hour and overnight charges.	Float
mta_tax	automatically triggered based on the metered rate in use.	Float
Tip_Amt	Tip amount – This field is automatically populated for credit card tips. Cash tips are not included.	Float
Tolls_Amt	Total amount of all tolls paid in trip.	Float
Total_Amt	The total amount charged to passengers. Does not include cash tips.	Float

Abbildung A.1: Attribute in der Datentabelle Yellow Cab 2009 für den Monat Januar und die dazugehörigen Datentypen

Aus Abbildung A.1 ist zu entnehmen, dass die Daten aus der Datentabelle überwiegend als kategorische und numerische Variablen gespeichert werden.

A.1.2 Weather-Datensatz

Field Name	Description	Data Type
STATION	station identification code	String
STATION_NAME	name of the station	String
LATITUDE		Float
LONGITUDE		Float
ELEVATION		Float
DATE	the year of the record (4 digits) followed by month (2 digits) and day (2 digits)	Datetime
AWND	Average daily wind speed (meters per second or miles per hour as per user preference)	Float
FMTM	Time of fastest mile or fastest 1-minute wind (hours and minutes, i.e., HHMM)	Float
PGTM	Peak gust time (hours and minutes, i.e., HHMM)	Float
PRCP	Precipitation (mm or inches as per user preference, inches to hundredths on Daily Form pdf file)	Float
SNOW	Snowfall (mm or inches as per user preference, inches to tenths on Daily Form pdf file)	Float
SNWD	Snow depth (mm or inches as per user preference, inches on Daily Form pdf file)	Float
TAVG	Average Temperature during the day	Float
TMAX	Maximum temperature (Fahrenheit or Celsius as per user preference, Fahrenheit to tenths on Daily Form pdf file	Float
TMIN	Minimum temperature (Fahrenheit or Celsius as per user preference, Fahrenheit to tenths on Daily Form pdf file	Float
TSUN	Daily total sunshine (minutes)	Float
WDF2	Direction of fastest 2-minute wind (degrees)	Float
WDF5	Direction of fastest 5-second wind (degrees)	Float
WSF2	Fastest 2-minute wind speed (miles per hour or meters per second as per user preference)	Float
WSF5	Fastest 5-second wind speed (miles per hour or meters per second as per user preference)	Float
WT01	Fog, ice fog, or freezing fog (may include heavy fog)	Bool
WT02	Heavy fog or heaving freezing fog (not always distinguished from fog)	Bool
WT03	Thunder	Bool
WT04	Ice pellets, sleet, snow pellets, or small hail	Bool
WT05	Hail (may include small hail)	Bool
WT06	Glaze or rime	Bool
WT07	Dust, volcanic ash, blowing dust, blowing sand, or blowing obstruction	Bool
WT08	Smoke or haze	Bool
WT09	Blowing or drifting snow	Bool
WT11	High or damaging winds	Bool
WT14	Drizzle	Bool
WT16	Rain (may include freezing rain, drizzle, and freezing drizzle)	Bool
WT17	Freezing rain	Bool
WT18	Snow, snow pellets, snow grains, or ice crystals	Bool
WT19	Unknown source of precipitation	Bool
WT22	Ice fog or freezing fog	Bool

Abbildung A.2: Attribute in der Datentabelle Weather und die dazugehörigen Datentypen

A.2 Bewertung der Features in den Datentabellen

A.2.1 Yellow-Cab-Datensatz

Field Name	Necessary
vendor_name	No
Trip_Pickup_DateTime	Yes
Trip_Dropoff_DateTime	Yes
Passenger_Count	Yes
Trip_Distance	Yes
Start_Lon	Yes
Start_Lat	Yes
Rate_Code	Yes
store_and_forward	No
End_Lon	Yes
End_Lat	Yes
Payment_Type	No
Fare_Amt	Yes
surcharge	No
mta_tax	No
Tip_Amt	No
Tolls_Amt	No
Total_Amt	No

Abbildung A.3: Auswahl der relevanten Features aus dem Datensatz Yellow Cab für eine Taxifahrt

A.2.2 Weather-Datensatz

Field Name	Necessary
STATION	No
STATION_NAME	No
LATITUDE	No
LONGITUDE	No
ELEVATION	No
DATE	Yes
AWND	No
FMTM	No
PGTM	No
PRCP	No
SNOW	No
SNWD	No
TAVG	Yes
TMAX	Yes
TMIN	Yes
TSUN	No
WDF2	No
WDF5	No
WSF2	No
WSF5	No
WT01	Yes
WT02	No
WT03	No
WT04	No
WT05	No
WT06	No
WT07	No
WT08	No
WT09	No
WT11	Yes
WT14	No
WT16	Yes
WT17	No
WT18	Yes
WT19	No
WT22	No

Abbildung A.4: Auswahl der relevanten Features aus dem Weather-Datensatz für eine Taxifahrt

Die Angaben beschränken sich auf die Features Temperatur und zu den Wettertypen Nebel, Windstärke, Regen und Schnee. Durch die Begrenzung des Featureraums vom Weather-Datensatz wird eine kompakte Beschreibung des Wetters ermöglicht und Duplikate werden entfernt. Beispiele hierfür wären wie der Feature Average Windspeed und der Wettertyp High Wind. Es wird vermutet, dass der Wettertyp High Wind einen größeren Einfluss auf den Verkehr hat als die Angabe des Average Windspeed, der aus diesem Grund gelöscht wurde.

A.2.3 US-Holiday-Datensatz

Die Features des Feiertage-Datensatzes wurden direkt übernommen, der genügend Informationen über die Feiertage in New York City bereithält.

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Ort	Datum	Unterschrift im Original