Capital Bikeshare Analyseprojekt

Analyseteam: Tanja Koppenhöfer, Tanja Kirschner, Britta Öhlenschleger, Deborah Ullrich, Jasmin Stribik

0) Aufbau und Art der Dokumentation

Die Dokumentation des Analyseprojekts des Unternehmens Capital Bikeshare ist in mehrere Jupyter Notebooks (zukünftig als Notebook bezeichnet) aufgeteilt.

Dieses Kapitel stellt für den Leser einen Leitfaden dar, welche Informationen in welchem Notebook gefunden werden können.

Die Dokumentation des Projektes ist in den einzelnen Notebooks parallel zu den Analysetätigkeiten durchgeführt worden.

In den folgenden Notebooks wird zusätzlich erläutert, welche Rohdaten von welchen Quellen heruntergeladen werden können. Einige Rohdaten hat das Analyseteam selbst erstellt. Diese sind im "zip-Ordner Daten" Abgabeverzeichnis hinterlegt. Es handelt sich um folgende Dateien:

- sights_washington.kml (DATA_PATH = '../data/')
- Station_Data.csv (RAWDATA_PATH = '../data/raw')
- bike_idle_times.pkl (DATA_PATH = '../data/')
- bike_operation_times.pkl (DATA_PATH = '../data/')
- Feiertage.xlsx (DATA_PATH = '../data/')

In der folgenden Abbildung ist der Ablauf der Notebooks dargestellt.

Für den managementorientierte Leser kann das vorliegende Notebook ausreichende Informationen lieferen. Dieses fasst die Vorgehensweise, die verschiedenen Phasen und die Ergebnisse der Analyse zusammen.

Dem stark technisch versierten Leser empfehlen wir den in der Abbildung dargestellten Ablauf um die Details der Analyse näher kennenzulernen.

In dem vorliegenden Notebook wird an den jeweiligen Stellen auf die Notebooks verwiesen.

Für das Notebook ***60_DataUnderstanding*** werden alle in der Abbildung gelb markierten Dateien für die Ausführung benötigt. Ebenso verhält es sich bei Notebook ***70_DataPreparation*** für die Ausführung dieses Notebooks werden alle hellblau markierten Dateien benötigt.

Die gestrichelten Linien der Dateien zu den Notebooks zeigen, ob die Dateien als Input oder Output der Notebooks entstehen wobei auch beides möglich ist. Das Bild kann in einer besseren Darstellung auch im Ordner "Images" gefunden werden.

1) Erläuterung der Vorgehensweise

Für die Durchführung des Analyse-Projektes haben wir uns auf den CRISP-DM Prozess festgelegt, da dieser das Data Mining als Hauptziel hat und somit als geeignet für unsere Analyseaufgabe eingestuft worden ist.

Ein wichtiger Vorteil von CRISP-DM stellt der agile Projektmanagementansatz dar, welcher die Möglichkeit bietet, zwischen den unterschiedlichen Projekt-Phasen hin- und herzuwechseln, je nachdem, wie sich das Data Mining Projekt entwickelt.

Zur Sicherstellung einer effizienten und effektiven Zusammenarbeit einigten wir uns bereits zum Projektauftakt auf kurze aber in geringen Zeitabständen, regelmäßig wiederkehrende Rücksprachen.

Die einzelnen Phasen des CRISP-DM Prozesses und unsere umgesetzten Aufgaben in dieser Phase werden nachfolgend genauer beschrieben.

2) Business Understanding

Das Ziel des Business Understandings ist es, konkrete Ziele und Anforderungen für das Data Mining zu definieren. Hierzu haben wir zunächst die Ausgangssituation analysiert und Fragestellungen entwickelt, welche wir im weiteren Verlauf des Analyseprojekts beantworten möchten.

Um möglichst ergebnisorientiert zu arbeiten, haben wir uns den übergeordneten Kunden (sozusagen unseren "Datenlieferanten"), nämlich das Unternehmen Capital Bikeshare, näher angeschaut. Die folgenden Kerninformationen konnten wir aus der Unternehmenswebseite https://www.capitalbikeshare.com/ (https://www.capitalbikeshare.com/) herausfiltern:

Capital Bikeshare, das Fahrradleihsystem der Metro Washington D.C., ist Teil der Motivate Gruppe, welche der globale Anführer in Fahrradleihsystemen ist. Das Leistungsversprechen ist eine rund um die Uhr bestehende Fahrrad-Verfügbarkeit, also 24/7, 365 Tage im Jahr, an allen Ausleihstationen zu gewährleisten.

Um dieses Leistungsversprechen zu erfüllen, sollen verschiedene Vorhersagen getroffen werden:

- Vorhersage der Anzahl an Ausleihvorgängen insgesamt pro Stunde eines Tages
- Vorhersage der Anzahl an Ausleihvorgängen pro Stunde eines Tages je Station

Mittels unserer Prognosearbeit soll Capital Bikeshare sicherstellen können, dass zu jeder Stunde und an jeder Station die benötigte Anzahl Fahrräder zur Verfügung stehen. Gegebenenfalls lassen sich aus unseren Ergebnissen auch Handlungsempfehlungen zu weiteren, internen Verbesserungen (Standzeiten, Geographie, Mitgliedertypen) ableiten.

Als Ziel war gesetzt, ein Prognosemodell mit einer möglichst genauen Vorhersagegüte systematisch zu erarbeiten und die Vorgehensweise verständlich zu dokumentieren.

Fazit: Anhand der hier gewonnenen Erkenntnisse wurden die gegebenen Data Mining Ziele greifbar gemacht.

Ebenfalls wurde in dieser Phase damit begonnen, weitere mögliche, zusätzliche Datenquellen zu identifizieren.

3) Data Understanding

Ziel dieser Phase ist es, mithilfe von geeigneten Visualisierungen eine Übersicht über den Umfang und das Potenzial der Ausgangsdaten sowie der zusätzlichen, für das Prognosemodell als relevant eingestuften Datenquellen. Schließlich soll damit ein umfassendes erstes Verständnis für die Daten und das zugrundeliegende Analyseproblem geschaffen werden.

Ausgangspunkt dieser Phase waren die uns mit dem Analyseauftrag mitgelieferten Analysedaten, nämlich die Tourdaten zu den Ausleihvorgängen von CapitalBikeshare der Jahre 2015, 2016 und 2017. Diese Daten wurden im Rahmen des ***Notebook 45_ExploratoryDataAnalysis*** näher betrachtet und sowohl deskriptiv als auch explorativ erkundet. Hieraus haben sich weiterführende Fragen und erste Erkenntnisse zur Datenverteilung, des Umfangs der Daten als auch zu deren Qualität ergeben.
Fazit: Die Erkenntnis war, dass die Datenqualität ausreichend zur weiteren Prognose ist und nur geringe Mängel erkannt wurden.

Ebenso wurde klar, dass der Business Case nur unter Hinzuziehung weiterer Datenquellen vollständig begriffen werden konnte. An dieser Stelle der Phase des Data Understandings werden die benötigten Datenquellen identifiziert und hinsichtlich ihrer Qualität analysiert. Hierzu zählt auch die Analyse der Datentypen und der Datenqualität dieser Datenquellen.

Im Rahmen eines gemeinsamen Brainstormings haben wir folgende Use Cases identifizieren können:

Uhrzeiten:

- Welches ist die beliebteste Fahrradfahrzeit im Tagesverlauf? Gibt es Parallelen bei den unterschiedlichen Wochentagen?
- Gibt es Unterschiede bei den Fahrradfahrzeiten an Wochenenden/Feiertagen/Urlaubszeiträumen?

Jahreszeiten:

- Sind Unterschiede zwischen den Jahreszeiten erkennbar?
- Nimmt die Nutzungsdauer in den Wintermonaten ab?

Witterungsbedingungen:

- Haben unterschiedliche Witterungsbedingungen Auswirkungen auf die Ausleihvorgänge (z.B. Temperatur, Niederschlag, Wind, Luftfeuchtigkeit)?

Infrastruktur:

- Hat das Vorhandensein von Fahrradwegen, Hauptstraßen oder Landstraßen eine Auswirkung auf die Ausleihvorgänge?
- Beeinflusst die Anbindung an den öffentlichen Nahverkehr (z.B. Bushaltestellen, U-/S-Bahn-Stationen, Taxistationen, Mietwagenverleihe, Bahnhofs- oder Flughafennähe) die Ausleihvorgänge?

Beschaffenheit der Fahrräder:

- Gibt es in regelmäßigen Zeiträumen Reparatur/Instandhaltungsmaßnahmen?
- Kann man erkennen, nach wie vielen zurückgelegten Kilometeren ein Fahrrad aussortiert/entfernt wird?
- Wann muss ein Fahrrad zur Wartung und kann man den richtigen Zeitpunkt hierfür prognostizieren?
- Gab es im Laufe der zu analysierenden Jahre gegebenenfalls Innovationen oder Ausweitungen des Produkt- oder Serviceportfolios, wie beispielsweise die Vermietung von E-Bikes?

Gesellschaftliche Einflüsse:

- Schulen/Universitäten in der Nähe von Fahrradstationen oder eher in der Industriegegend?
- Gibt es gegebenenfalls einen Zusammenhang mit der Frequentierung einzelner Stationen und Kriminalitätsraten/Häufigkeit an Verkehrsunfällen in bestimmten Stadtvierteln/Straßenbereichen?

Nutzungsstrukturen/Kundenprofile:

- Wie ist das Verhältnis von "Casual"-Nutzern des Fahrradmietservices im Vergleich zu Mitgliedern ("Members")? Gibt es über die Jahre hinweg eine Veränderung dieses Verhältnisses, beispielsweise aufgrund verstärkter Werbemaßnahmen oder verschärfter Konkurrenz?
- Bei welchen Stationen können beliebte Orte/Touristikrouten/Sehenswürdigkeiten angenommen werden, weil dort die Fahrräder am längsten und häufigsten ausgeliehen werden, gemessen an Häufigkeit der ausgeliehenen Fahrrädern und der zurückgegebenen Fahrrädern an der gleichen Station?
- Wer sind die Kunden des Unternehmens? Alter, Zielgruppe, etc.? Lassen sich unterschiedliche Kunden-Klassifikationen (beispielsweise anhand einer Regressions- oder Assoziationsanalyse) erkennen und diese gegebenenfalls mit Orten paaren, an denen sich zu einer bestimmten Uhrzeit am meisten Leute dieser Gruppe aufhalten, um so die Vorhersagegenauigkeit unseres Modells zu verbessen?

Leitfragen aus den Daten:

- Wie lange dauert eine durchschnittliche Fahrt? Was ist die durchschnittliche Nutzungsdauer pro Tag / Monat?
- Wie viele Kilometer werden durchschnittlich zurückgelegt?
- Welche sind die beliebtesten Start- und End-Stationen, gemessen an der Häufigkeit der ausgeliehenen sowie der der zurückgegebenen Fahrräder?
- Welches sind die beliebtesten Routen, gemessen an höchster Ausleihdauer gepaart mit hoher Häufigkeit?
- Wie viele Kilometer Luftlinie werden pro Tour zurückgelegt? (Hierzu wurden verschiedenen Arten der Berechnung der Distanz zwischen zwei Punkten mit gegebenem Längen- und Breitengrad auf der Erdoberfläche erforscht und schließlich auch angewandt, vgl. Notebook ***51_DistanceStationsAnalysis***)
- An welchen Stationen stehen die meisten unbenutzten Fahrräder und wo könnte man Stationen daher auch verkleinern? An welchen Stationen sind die Fahrräder gut ausgelastet? Und wieso werden an manchen Stationen mehr Fahrräder genutzt als an anderen?
- Sind über die Jahre neue Fahrradstationen dazu gekommen?

Die beliebtesten 15 Sehenswürdigkeiten der Stadt Washington DC wurden durch Analyse der folgende Touristenempfehlungsseiten erstellt: https://washington.sehenswuerdigkeiten-online.de/sehenswuerdigkeiten/sights_washington.html (https://washington.sehenswuerdigkeiten-online.de/sehenswuerdigkeiten/sights_washington.html)
https://globusliebe.com/washington-dc-sehenswuerdigkeiten/ (https://globusliebe.com/washington-dc-sehenswuerdigkeiten/)
Um die Geodaten in unserem Projekt verwenden zu könenn wurde eine KML-Datei mittels Google Earth (Erreichbar unter: https://earth.google.com/web/ (https://earth.google.com/web/)) erstellt. Zusätzlich Quellen für die Feiertage von Washington DC wurde von folgenden Quellen verwendet:
https://dchr.dc.gov/page/holiday-schedules-2014-and-2015 (https://dchr.dc.gov/page/holiday-schedules-2014-and-2015)
https://dchr.dc.gov/page/holiday-schedules-2016-and-2017 (https://dchr.dc.gov/page/holiday-schedules-2016-and-2017)

Auch werden im Rahmen des Data Understandings die Daten mittels verschiedener Analysen hinsichtlich Mustern und Zusammenhängen untersucht, welche eine Relevanz für das zukünftige Modell haben könnten. Die notwendigen Analysen werden im Notebook ***60_DataUnderstanding*** durchgeführt. Details zu den Analysen sind diesem Notebook zu entnehmen.

Zusätzlich zu den Untersuchungen der Datenqualität haben wir einzelne Objekte, wie zum Beispiel die Stationen und deren Geo-Daten oder auch die Fahrräder näher betrachtet, um einen Eindruck von den Daten zu bekommen. Im Hinblick auf unser Ziel (eine Prognose der Anzahl der Ausleihvorgänge) konnten wir so einen ersten Eindruck über die Relevanz einzelner Einflussgrößen auf das Prognosemodell bekommen.

Des Weiteren sind Erkenntnisse aus der Data Understanding Phase für die Wahl der geeigneten Parameter relevant. Die wichtigsten Erkenntnisse aus der Data Understanding Phase sollen nachfolgend kurz zusammen gefasst werden - die Details finden sich dann im Data Understanding Notebook.

- Die Anzahl der Stationen nimmt pro Jahr zu. In weiteren Recherchen wurde in Erfahrung gebracht, dass CapitalBiksehare seinen Nutzern einräumt, Vorschläge für neue Fahrradstationen einzureichen. --> Wenn möglich, sollte in dem Modell ein positives Wachstum mit beachtet werden.
- Es gibt unterschiede im Tagesverlauf, d.h. es entstehen Auslastungsspitzen an denen mehr Fahrräder ausgeliehen werden --> Daher sollte die Uhrzeit für eine möglichst genaue Prognose beachtet werden.
- Des Weiteren gibt es Unterschiede im Wochenverlauf, es werden mehr Fahrräder an Wochentagen als am Wochendende ausgeliehen --> Deshalb sollte auch der Wochentag in der Prognose nicht vernachlässigt werden.
- Um über das ganze Jahr einen Überblick zu bekommen, wurden die Monate den Saisons Frühling, Sommer, Herbst und Winter zugeordnet, wobei sich dann auch erkennen lässt, dass im Winter deutlich weniger Fahrräder ausgeliehen werden als in den restlichen Jahreszeiten.
- Bei der Betrachtung der hinzugefügten Wetterdaten konnte festgestellt werden, dass die Betrachtung des Niederschlages allein nicht aussagekräftig ist.
- Ebenfalls kam aus dem Data Understanding die Erkenntnis wenn die Prognose noch genauer gemacht werden soll, dann sind auch die unterschiedlichen Mitgliedstypen (Mitglied und Gelegenheitsnutzer) relevant, da Mitglieder weitaus häufiger Fahrräder ausleihen als Gelegenheitsnutzer. Diese Betrachtung steht aber nicht im Fokus des Prognosemodells.

Ein wichtiger Aspekt des Data Understandig ist die Identifikation von Ausreißern und fehlenden Werten, da diese großen Einfluss auf die späteren Prognoseergbnisse nehmen können. Vorallem in den zur Verfügung gestellten Wetterdaten konnten einige Null-Werte identifiziert werden. Dieses Problem wird während der Phase „Data Preparation“ behandelt. Auch identifizierte Ausreißer in den Ausleihdaten müssen hier betrachtet werden.

4) Data Preparation

Wie im vorherigen Abschnitt beschrieben, haben wir die initialen Daten um weitere, zusätzliche Datenquellen ergänzt. Außerdem haben wir im Laufe unseres Analyseprozesses schrittweise Qualitätsmangel beseitigt und versucht, die Datenqualität konstant zu verbessern unter Beibehaltung einer möglichst hohen Vollständigkeit. Während manche Beobachtungen an einzelnen Merkmalsausprägungen lediglich transformiert werden konnten, mussten manche davon, zum Beispiel aufgrund fehlender Werte, entfernt werden.

Damit die Prognosemodelle möglichst genaue Vorhersagen treffen können, ist die Aufbereitung der Daten in dieser Phase besonders wichtig. Dabei sollten fehlende Werte und Ausreißer bereinigt werden, der Datentyp eventuell angepasst und die relevanten Merkmale ausgewählt (Feature Selection) werden. Auch das Zusammenführen und Aggregieren von Daten wird in dieser Phase realisiert (Data Mining and Predictive Analysis: Intelligence Gathering and Crime Analysis, S.50). Die Schritte der Datenaufbereitung sind in dem Notebook ***70_Data Preparation*** zu finden.

Wir haben uns aus Performance-Gründen dazu entschieden zunächst ein stationsunabhängiges Modell zu trainieren. Hierfür wird das Datenset basierend auf Datum und Stunde aggregiert, Member Typ und Station werden außer Acht gelassen. Für spätere Prognosen wird jedoch auch ein Datenset erstellt, bei dem die Ausleihvorgänge auf Zeit- und Stationsebene aggregiert werden. Für beide Datensets wird die Data Preparation Phase identisch durchgeführt.

Damit auch kategorische Variablen in den Prognosemodellen beachtet werden können, wird ein „One-Hot-Encoding“ durchgeführt. Dieses wandelt die Variablen in eine entsprechende Anzahl Spalten um und füllt diese mit 1 wenn das Feld den Wert enthalten hat und mit 0 wenn dies nicht der Fall war.

Ebenfalls haben werden in dieser Phase die Daten schon in Test und Trainingsdaten aufgesplittet. Dabei dienen die Jahre 2015 und 2016 als Trainingsdaten und das Jahr 2017 als Testdaten.

Während dieser Phase haben wir bereits begonnen, uns mit der Frage auseinanderzusetzen, welche Machine Learning Algorithmen uns dabei helfen könnten, unser eigenes Analysemodell zu entwickeln.

Im folgenden Abschnitt gehen wir auf die ausgewählten Algorithmen weiter ein und beschreiben unser Vorgehen zur Konzeptionierung des Analysemodells.

5) Modeling

Ziel dieser Phase ist es, einen geeigneten Machine Learning Algorithmus zur Prognose der Ausleihvorgänge auszuwählen.

Bei der Suche nach geeigneten Prognosemodellen haben wir uns im Wesentlichen auf "Supervised Learning" Methoden fokussiert, da Trainingsdaten zur Verfügung stehen, mit welchen ein Modell die Ergebnisse trainieren kann. Ein weiterer Anhaltspunkt bei der Recherche nach geeigneten Prognosemodellen war die Einordnung der Problemstellung zwischen Regression, Klassifikation und Anomalieerkennung. Da es sich bei der Zielgröße (Anzahl an auszuleiheneden Fahrrädern) um eine quantitative Zielgröße handelt, werden Regressionsmodelle benötigt.

Um ein möglichst geeignetes Modell für die Vorhersage zu finden wurden folgende Modellverfahren in der Gruppe geprüft:

- Multiples lineare Regression
- Random Forest Regression
- Support Vector Machine
- XGB (Decision Tree)

Um einen erweiterten Blick zu bekommen, wurde aus Interesse auch ein nicht ganz typisches Verfahren für diesen Anwendungsfall ausgewählt.

- Neuronales Netz

Um die verschiedenen Algorithmen miteinander zu vergleichen, wurde auch eine geeignete Kostenfunktion definiert. Kostenfunktionen messen den Unterschied zwischen dem vorhergesagten Wert und dem tatsächlichen Wert (vgl. The Machine Learning Workshop, S.162) und geben so Auskunft über die Performance des Modells.
Je nach Problemstellung kommen verschiedene Kostenfunktionen in Frage. Hier soll nun eine der am häufigsten verwendeten Funktionen, der Root Mean Squared Log Error (RMSLE) verwendet werden. Berechnet wird der RMSLE aus der logarithmierten Quadratwurzel des durchschnittlichen Prognosefehlers. Durch die Verwendung des Logarithmus ist RMSLE recht stabil gegenüber Ausreißern. Je größer der RMSLE ist, desto schlechter ist die Anpassung des Modells. Ziel ist es folglich, RMSLE zu minimieren, um so die Güte des Modells zu steigern.

Nachfolgend werden die einzelnen Prognosemodelle und ihre Ergebnisse kurz beschreiben. Weitere Details finden sich in den einzelnen ***80_xxx*** Notebooks. Da diese Phase sehr iterativ mit der nachfolgende Phase Evaluation zusammenhängt, werden erste Teilergebnisse der Evaluation schon in diesem Abschnitt erläutert.

Multiple Lineare Regression

Bei einer Linearen Regression soll eine Zielvariable (in unserem Fall die Anzahl an ausgeliehenen Fahrrädern \"count_out\") durch eine andere unabhängige Variable möglichst gut durch eine Geradengleichung beschrieben werden. Als Erweiterung werden bei der Multiplen Linearen Regression nicht nur eine unabhängige Variable zur Erstellung der Geradengleichung verwendet, sondern mehrere verschiedene Variabeln. Da wir in der Data Understanding Phase bereits erkennen konnten, dass mehrere Spalten (z.B. Wochentag oder Sasion) Auswirkungen auf die Ausleihvorgänge haben müssen, wird in unserem Anwendungsfall eine Mulitple Lineare Regression benötigt. Um eine Lineare oder Multivariante Regression in Python zu erstellen bieten sich die Möglichkeit über die Bibliothek sickit-learn oder über das statsmodel Package an. In dem Notebook ***80_Multiple Regression*** wurde sowohl eine Lineare Regression (mit der unabhängigen Variablen Temperatur) als auch eine Multiple Lineare Regression erstellt, wobei die Multiple Lineare Regression eine deutlich bessere Vorhersage lieferte. Bei der Anwendung dieses Prognoseverfahrens konnte nur eine Vorhersagegenauigkeit (Accuracy) von ca. 52% und ein RMSLE-Score von 1,867 erzielt werden.

Random Forest Regression

Bei der Random Forest Regression handelt es sich um eine Ensemble-learning Methode. Bei diesen Methoden werden die Ergebnisse mehrere Algorithmen miteinander kombiniert um ein besseres Ergebnis zu erzielen als mit einem einzelnen Modell (vgl. https://towardsdatascience.com/random-forest-and-its-implementation-71824ced454f (https://towardsdatascience.com/random-forest-and-its-implementation-71824ced454f)). Hierbei verwendet Random Forest "Bagging", das zufällige Ziehen (mit Zurücklegen) aus den Trainingsdaten. Die verschiedenen Bäume werden parallel erstellt und die einzelnen Ergebnisse dann aggregiert.
Bei Random Forest existieren weiterhin verschiedene Hyperparameter, mit denen die Güte des Modelles optimiert werden kann. Um die besten Hyperparameter zu ermitteln wurden verschiedene Verfahren verwendet. Zuerst wurde eine sog. Randomized Search durchgeführt. Hierbei können verschiedene Werte der Parameter festgelegt werden, welche dann zufällig miteinander kombiniert werden. Auch die Anzahl der verschiedenen Kombinationen kann festgelegt werden.
Um das Modell weiter zu optimieren, wurden die so ermittelten Werte anschließend verwendet um eine Grid Search durchzuführen. Auch hierbei werden die verschiedenen Werte der Parameter miteinander kombiniert. Im Gegensatz zur Randomized Search werden jedoch Modelle für alle Kombinationsmöglichkeiten trainiert, was die Performance beeinträchtig. Daher bietet es sich an, zuerst eine Randomized Search durchzuführen um die Werte der Parameter einzugrenzen und dann mittels Grid Search eine (noch) bessere Konfiguration der Hyperparameter ermitteln. Das mit optimierten Hyperparametern ausgeführte Modell erzielt einen r2-Wert von 0,796 sowie einen RMSLE-Score von 0,531.

Support Vector Machine

Support Vektor Maschinen klassifizieren unterschiedliche Datenpunkte in zwei Gruppen, indem sie zwischen den Datenpunkten eine Grenzlinie ziehen. Diese Methode lässt sich aber auch auf Regressionsprobleme anwenden. Das Verfahren *Support Vector Machine (SVM)* erzielte bei uns in der Basis-Konfiguration bereits nur eine sehr geringe Vorhersage-Genauigkeit. Zur Erhöhung der Genauigkeit haben wir bei diesem Verfahren versucht die Anzahl der Klassen für die Klassifikation zu verringern. Dazu haben wir die Zielvariable "Ausleihen" in verschiedene *Bins/Ranges* eingeteilt und jede Ausleihe einem entsprechenden *Bin* zugewiesen. Durch dieses Vorgehen und dem Tuning der *SVM*-Parameter war es uns jedoch nur möglich eine *Accuracy* von ca. 15,5% und einen *RMSLE* von 0.779 zu erreichen. Da andere Algorithmen mit auf den Basis-Daten mit einer ersten Basis-Konfiguration schon bessere Werte aufwiesen, haben wir uns entschlossen *SVM* nicht als unser Prognoseverfahren auszuwählen.

XGB (Decision Tree)

Beim "eXtreme Gradient Boosting" handelt es sich um ein Verfahren des maschinellen Lernens in Zusammenhang mit Entscheidungsbäumen.
Im Gegensatz zum Verfahren Random Forest besteht beim Aufbau des Entscheidungsbaums bei XGB eine Abhängigkeit in dem Sinne, dass mehrere Bäume parallel aufgebaut werden und nach einem schlecht performenden Entscheidungsbaum ein zumindest gering besser performender Baum gebaut wird. Hierfür macht XGB - wie der Name schon verrät - von Boosting Gebrauch.

Bei diesem Verfahren müssen die Daten zwar numerisch aufbereitet werden, sie müssen jedoch weder normalisiert noch skaliert werden.

In unserem Fall haben wir versucht, XGB sowohl für die Lösung eines Regressionsproblemes (XGBRegressor) als auch für die Lösung eines Klassifikationsproblemes (XGBClassifier) anzuwenden, da beide Varianten für die Zeitreihenprognose angewandt werden können. Hierbei haben wir auf normalisierte und skalierte, numerische Daten zugegriffen.
Während beim XGBRegressor der "reg:squarederror" das "objective" darstellt, tut dies beim XGBClassifier der "binary:logistic".

Da dieses Modell sehr ressourcenintensiv ist, haben wir das Verfahren für unser Prognosemodell nicht weiter in Betracht gezogen.

Neuronales Netz

Das Verfahren *künstliches neuronales Netz* wird in unserem Fall als überwachte Lernmethode verwendet, da die Eingabewerte und die Ausgabewerte beim Lernen berücksichtigt werden.

Im der einfachsten Konfiguration besteht ein künstliches neuronales Netz aus drei Schichten. Der Eingabeschicht, einer Zwischenschicht und einer Ausgabeschicht. In unserem Anwendungsfall haben wir uns entschlossen in der Eingabeschicht je 43 Neuronen (43 Input Neuronen da insgesamt 45 Attribute und maximal 2 Ausgabewerte) mit je einem Attribut zu berücksichtigen, in der Zwischenschicht jeweils 100 Neuronen einzusetzen und in der Ausgabeschicht zwei (Ausleihe- und Rückgabevoränge) beziehungsweise ein Neuron (Rückgabevorgänge) zu berücksichtigen.

Das Neuronale Netz wird in zwei Konstellationen getestet, die erste Konstellation berücksichtigt alle Eingabedaten sowie die "Ausleihvorgänge" und die "Rückgabevorgänge" als Ergebnisse. Die zweite Konstellation berücksichtigt nur die "Ausleihvorgänge" als Ergebnis (Anmerkung: Das Attribut Rückgabevorgänge ist in diesem Fall komplett aus dem Erfahrungsschatz des Neuronalen Netzes herausgenommmen). Im ersten Fall wird eine Accurancy von 54,89 % erreicht. Im zweiten Fall eine Accurrancy von 1,12 %. Bei der Vorhersage eines einzelnen Rückgabewertes können sicherlich durch das Tunen von Hyperparametern noch bessere Ergebnisse erzielt werden.

Da die Performance und Vergleichsparameter des neuronalen Netzes außer Konkurrenz stehen, werden diese nicht mit den anderen Verfahren verglichen.

6) Evaluation

In dieser Phase werden die Ergebnisse der Modellierungsphase analysiert und evaluiert. Dabei sollte auch geprüft werden, ob die gewählten Maschine Learning Algorithmen realistische Ergebnisse liefern oder zu einem Overfitting o.ä. führen.

Vergleich der verschiedenen Modelle

Zum Vergleich der verschiedenen Modellierungsverfahren wurden 3 Parameter verwendet:

1. **RMSLE**
   Der RMSLE **(Root Mean Squared Logarithmic Error)** berechnet den relativen Fehler zwischen Vorhersagewerten und tatsächlichen Werten. Durch die Verwendung des Logarithmus ist die Skalierung der Daten nicht relevant und auch Ausreißer haben einen deutlich geringeren Effekt auf den RMSLE-Wert. Eine andere Eigenschaft von RMSLE ist die höhere "Bestrafung" von zu gering prognostizierten Werten im Vergleich zu zu hoch geschätzten Werte (vgl. https://medium.com/analytics-vidhya/root-mean-square-log-error-rmse-vs-rmlse-935c6cc1802a). Damit eignet sich RMSLE sehr gut für die Bewertung der Modelle im Kontext des Geschäftsmodells von Capital Bikeshare. Eine zu geringe Prognose der Ausleihzahlen steht im Gegensatz zu dem Ziel, zu jedem Zeitpunkt genügend Fahrräder zur Verfügung zu stellen.
2. **Accuracy**
   Der Accurcy-Wert (multipliziert mit 100) gibt in Prozent an, wie gut die gesamte Vorhersagegenauigkeit des Modells ist. Sie steht für den Anteil der korrekt Vorhergesagten Werten im Verhältnis für die gesamten vorhergesagten Werte. Deshalb sollte für die Accurancy ein möglichst hoher Wert erreicht werden. (vgl. https://developers.google.com/machine-learning/crash-course/classification/accuracy). Zur Berechnung der Accuracy wird die cross_val_score Methode von sklearn verwendet. Hierbei wird mittels Kreuzvalidierung die durchschnittliche Genauigkeit des Modells berechnet (vgl. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html#sklearn.model_selectic
   Es wurde eine 10-fache Kreuzvalidierung verwendet.
3. **R2**
   Der R-Squared (R2) Wert sollte ein Wert möglichst nahe an 1,0 sein. Er misst, wie gut die unabhängigen Variablen X dazu geeignet sind, die abhängige Variable y vorherzusagen (vgl. https://medium.com/analytics-vidhya/r-squared-vs-adjusted-r-squared-a3ebc565677b). Hierbei wird der Anteil der durch die unabhängigen Variablen (X) erklärten Varianz im Vergleich zur Gesamtvarianz berechnet. Bildlich gesprochen beschreibt der R-Squared Wert bei einer Linearen Regression die Streuung der Datenpunkte, um die berechnete Regressionslinie. (vgl. https://statisticsbyjim.com/regression/interpret-r-squared-regression/). Ist dieser Wert sehr gering, eignen sich die Variablen nicht für die Prognose von y. Ist der Wert nahe an 1 besitzt das Modell eine gute Anpassungsgüte.

Nachfolgende Tabelle zeigt die ausgewählten Messwerte der einzelenen Modellierungsverfahren im Vergleich auf, wobei darauf hinzuweisen ist, dass die Ausführung des Notebooks zum Thema XGB sehr zeitintensiv ist (vgl. zugehöriges ***80_XGB (Decision Tree)***).

Prognosemodell	RMSLE	Accuracy	R2
Multiple Lineare Regression	1,867	0,528	0,64
Random Forest	**0,531**	**0,745**	**0,796**
XGB (Decision Tree)	1,32	0,07	0,004296
Support Vector Machine	1,051	0,1524	0,056

Prognosemodell	RMSLE	Accuracy	R2
Neuronales Netz	Wertermittlung nicht möglich	0,529	Wertermittlung nicht möglich

Im Vergleich der Tabelle ist zu erkennen, das bei unseren Tests das **Random Forest** Prognosemodell am Besten performt hat. Es konnte eine Genauigkeit von 74,5% erzielt werden und der RMSLE-Wert liegt bei 0,53. Mittels Random Forest lassen sich die Ausleihzahlen pro Stunde recht gut vorhersagen. Jedoch weißt auch der Random Forest Schwächen auf, die die Güte der Prognose reduzieren. Dazu zählt vorallem die Problematik, dass vorallem bei hohen Ausleihzahlen die Vorhersage recht stark von dem tatsächlichen Wert der Test-Daten abweicht. Außerdem scheint das Modell zum Overfitting zu tendieren. Genaueres kann dem zugehörigen Notebook entnommen werden.

Im Folgenden wird auf Basis der Prognosedaten ein Liniendiagramm erstellt, welches Prognose und tatsächlichen Wert gegenüberstellt. Dieses Diagramm zeigt, dass Random Forset vorallem mit der Vorhersage der Peaks Schwierigkeiten hat. Um dies zu verbessern könnten die Daten während der Data Preparation Phase noch weiter angepasst werden. Möglichkeiten hierfür sind im Kapitel "Weiterführende Schritte" beschrieben.

In [1]:

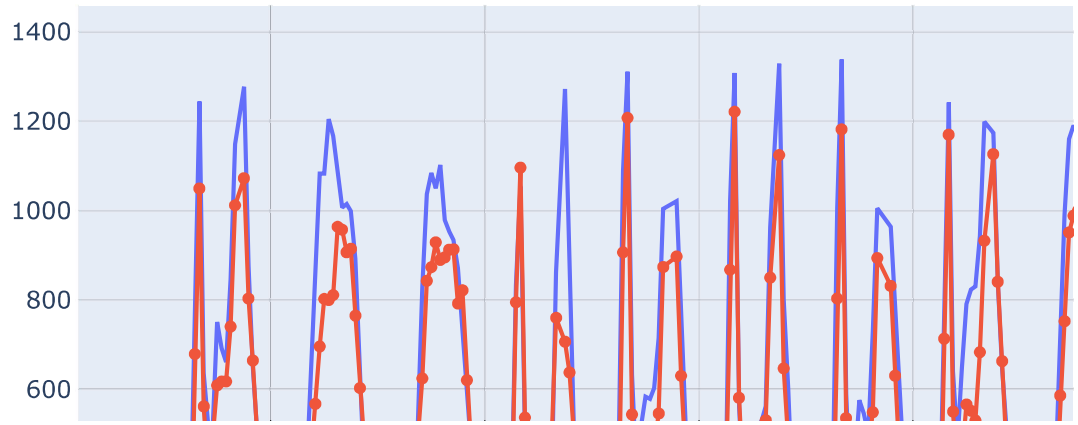
```
import plotly.graph_objects as go
import pandas as pd
DATA_PATH = '../data/'
```

In [2]:

```
X_test_range = pd.read_pickle(DATA_PATH+'Prognosedaten.pkl')

fig = go.Figure()
fig.add_trace(go.Scatter(x=X_test_range['DateTime'], y=X_test_range['count_out'],
                    mode='lines',
                    name='actual'))
fig.add_trace(go.Scatter(x=X_test_range['DateTime'], y=X_test_range['prediction'],
                   mode='lines+markers',
                         name='predicted'))

fig.show()
```



7) Deployment

Diese Phase ist Out-of-Scope unseres Projektes.

8) Weiterführende Schritte

Das durch unser Projekt entstandene Prognosemodell ist in der Lage, basierend auf Wetterdaten und Zeit die Anzahl der Ausleihvorgänge von Capital Bikeshare mit einer Genauigkeit von 74,5% zu prognostizieren. Es gibt noch weitere Möglichkeiten, die zur Verfügung stehende Daten anzupassen, mit dem Ziel diesen Wert weiter zu verbessern, welche wir in dem Umfang unserer Arbeit nicht betrachtet haben.

Außerdem besteht die Möglichkeit, während des Data Understanding und der Data Preparation weitere Optimierungsmöglichkeiten vorzunehmen:

- *Korrelation zwischen den Variablen*:
  Um die Güte des Prognosemodells zu optimieren, kann die Korrelation zwischen den verschiedenen Variablen untersucht werden. Wenn Variablen sich gegenseitig beeinflussen, kann dies einen negativen Einfluss auf die Prognose haben. Diese sollten dann entfernt und nicht im Modell berücksichtigt werden. Die Korrelationsanalyse wurde im Rahmen des Data Understanding kurz angerissen, könnte aber noch umfangreicher ausgeführt werden.
- *Skalierung der Daten*:
  Damit Daten mit einem sehr hohen Wertebereich in der Prognose nicht höher gewichtet werden, wäre eine Skalierung von Vorteil. Dabei werden die numerischen Werte auf ein einheitliches Skalenniveau angepasst. In dem vorliegenden Datensatz wäre dies hauptsächlich bei den Wetterdaten sinnvoll, da diese unterschiedliche Wertebereiche haben.
- *Auswirkungen der jährlichen Wachstumsrate auf die Ausleihzahlen:*
  Im Notebook ***60_DataUnderstanding*** haben wir herausgefunden, dass über die Jahre neue Stationen hinzu gekommen sind. Wir mutmaßen daher, dass es zusammenhängend eine Erhöhungen der Ausleihen über die Jahre gab. In der Prognose betrachten wir jedoch alle Jahre gleichgewichtet. Eine erhöhte Ausleihzahl unter den selben Ausgangsbedingungen könnte also das trainierte Modell verfälschen und die Prognosequalität verschlechtern. Dieses Einflussgröße wurde in unserem Modell nicht berücksichtigt.
- *Prognose der Ausleihvorgänge pro Station:*
  In der Modelling Phase wurde für alle Modelle zunächst mit dem Ziel, die Vorhersage der Ausleihvorgänge insgesamt vorherzugsagen, begonnen. Anschließend hatten wir das Ziel, auch die Werte auf Stationsebene zu prognostizieren. Dieses Vorhaben hat sich jedoch als für uns nicht durchführbar erwiesen, da die Datenmenge bei der uns zur Verfügung stehenden Hardware zu Performance-Schwierigkeiten geführt hat, sodass wir die Berechnung aus Zeitgründen nicht durchführen konnten.

Fazit: Durch die Implementierung der hier beschriebenen Optimierungsmöglichkeiten, könnte auch der Fall eintreten, dass ein anderer Algorithmus zu einem besseren Prognosemodell führt, als aktuell der Random Forest

In [ ]: