

University of Cologne  
Faculty of Management, Economics and Social Sciences

San Francisco 2019

# **Team 'We Want a Toast'**

## **Group 1 - Assignment WS 22/23**

For the study program Information System

Authors:           Laurenz Bihlmayer  
                      Leo Niggemann  
                      Manuel Heeren  
                      Tobias Wißmach

Repository Link:   <https://github.com/major-mayer/aawewantatoast>

Lecture:           Analytics and Application  
Instructor:       Prof. Dr. Wolfgang Ketter  
Submission date:   31.01.2023

# Executive summary

The main objective of this project was to analyze trip data from Lyft's bike rentals in San Francisco for the year 2019 in order to monitor and predict demand. To achieve this, we used three data sets that were cleaned and joined together; these included bike rental data for San Francisco, weather data for 2019 from weather.com, and geodata provided by Lyft's system data for the bike stations. Additionally, we established key performance indicators (KPIs) to monitor the operational performance of the bike fleet.

Through cluster analysis, we were able to gain a deeper understanding of the customer groups and station relevance. We found the following customer groups: 'Nightriders', which mainly ride in the early morning and late evening hours, the 'Workers' which take a bike to commute to work and lastly the 'Midday rides'.

Our findings also showed that the highest demand for bike rentals occurred during the weekday morning and evening rush hours, with the busiest bike stations located in downtown San Francisco. Lastly, we implemented a predictive analysis to forecast demand and optimize the performance of the fleet in the future. For the model, we chose the K-Nearest-Neighbor model, because it showed the best  $R^2$  value of 0.956773. We also found out, that the lag feature from one week before as well as the `is_workday` feature are the most important features for the prediction.

To optimize the performance of the fleet, we recommend that Lyft focus on increasing the availability of bikes at these busy stations during peak hours, as well as providing promotions and incentives for both short-trip and weekend-customers. Additionally, our analysis revealed the existence of two customer clusters: short-trip customers who primarily use bikes for commuting purposes, and weekend-customers who use bikes for leisure and recreation. To better serve these customers, Lyft should consider offering corporate subscriptions for short-trip customers and promotions for long-trip customers on weekends.

The results of our analysis are based on a limited dataset, which includes only one year of bike rental data and lacks information on user demographics and user distinction as distinction. Nevertheless, our findings and recommendations provide a solid foundation for Lyft to improve its bike-sharing service and better meet the needs of its customers.

# Contents

<b>List of figures</b>	<b>IV</b>
<b>1. Problem Description</b>	<b>1</b>
<b>2. Data Description and Preparation</b>	<b>1</b>
2.1. Ride Data . . . . .	1
2.2. Weather Data . . . . .	2
<b>3. Data Analysis</b>	<b>2</b>
3.1. Descriptive Analysis . . . . .	2
3.1.1. Temporal Demand Patterns . . . . .	2
3.1.2. Geographical Demand Patterns . . . . .	3
3.1.3. Key Performance Indicators (KPIs) . . . . .	5
3.2. Cluster Analysis . . . . .	5
3.3. Trip Clustering . . . . .	5
3.3.1. Weather Clustering . . . . .	6
3.3.2. Station Clustering . . . . .	6
3.4. Predictive Analysis . . . . .	9
3.4.1. Preparation . . . . .	9
3.4.2. Feature Engineering . . . . .	9
3.4.3. Grid Search Validation and Train/ Test Splitting . . . . .	9
3.4.4. Predictive Models . . . . .	10
<b>4. Conclusion</b>	<b>12</b>
<b>Literatur</b>	<b>13</b>
<b>Appendix A. K-nearest Neighbors hyperparameter optimization</b>	<b>13</b>
<b>Appendix B. Distribution of Work</b>	<b>13</b>

## List of Figures

1.	An excerpt from the trip dataset. . . . .	1
2.	An excerpt from the weather dataset. . . . .	2
3.	Ride counts plotted on monthly, weekday, and hourly basis (from left to right). . .	3
4.	Temperatures plotted on monthly, weekday, and hourly basis (from left to right). .	4
5.	Heat map that shows the relation between weekday (x-axis), hour (y-axis) and trip count (colour). . . . .	4
6.	Plot that shows the number of relocations per hour. . . . .	5
7.	Trip clustering based on start_month, start_weekday, start_hour, user type, trip duration and workday/weekend. . . . .	7
8.	Weather clustering based on month, weekday, hour, max. temperature and precipitation. . . . .	8
9.	Station clustering based on location and usage. . . . .	9
10.	KFold and TimeSeriesSplit Algorithm. Source: <a href="https://scikit-learn.org">https://scikit-learn.org</a> . . .	10
11.	Comparison between values predicted by the KNN model and real test values. . .	11
12.	Feature importance as returned by the XGBoost Random Forest Regressor. . . . .	12

# 1. Problem Description

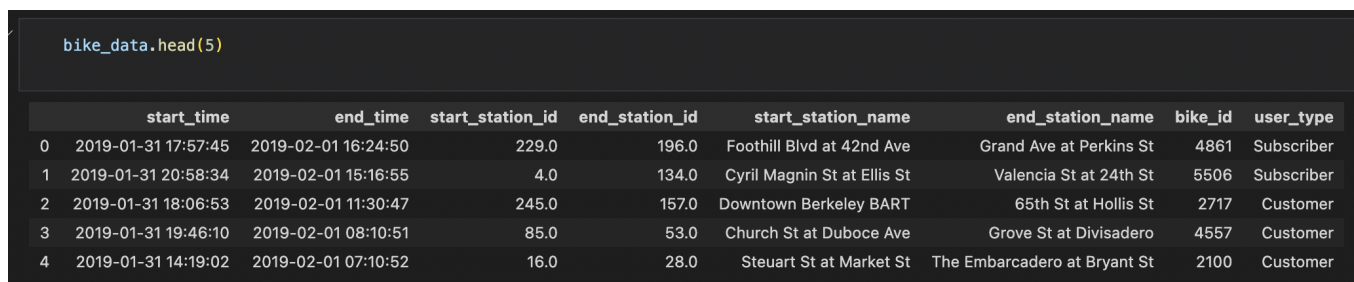
The fundamental goal of this project is to inform and improve business decisions for the bike-sharing platform Lyft. By conducting analysis on historical data, we aim at identifying patterns and trends that unfold demand-driving factors and thereby enhance allocation. To gain a more comprehensive understanding of these behavioural patterns, we generate costumer, time and spatial based clusters and provide hourly demand forecasting.

## 2. Data Description and Preparation

To conduct the data analysis, we utilized the following data sets: [sf\\_2019.csv](#), [weather\\_hourly\\_sf.csv](#), and [Sf\\_2019\\_full.csv](#). In the following section, we will provide an overview of the data and explain the steps we took to prepare it.

### 2.1. Ride Data

The data for the bike rides is located in the [sf\\_2019.csv](#) file and includes 2,506,982 trips that were taken in San Francisco in 2019. The file provides us with information about the start and end time, start and end station, as well as the user type, which can be either a subscriber or a customer.



	start_time	end_time	start_station_id	end_station_id	start_station_name	end_station_name	bike_id	user_type
0	2019-01-31 17:57:45	2019-02-01 16:24:50	229.0	196.0	Foothill Blvd at 42nd Ave	Grand Ave at Perkins St	4861	Subscriber
1	2019-01-31 20:58:34	2019-02-01 15:16:55	4.0	134.0	Cyril Magnin St at Ellis St	Valencia St at 24th St	5506	Subscriber
2	2019-01-31 18:06:53	2019-02-01 11:30:47	245.0	157.0	Downtown Berkeley BART	65th St at Hollis St	2717	Customer
3	2019-01-31 19:46:10	2019-02-01 08:10:51	85.0	53.0	Church St at Duboce Ave	Grove St at Divisadero	4557	Customer
4	2019-01-31 14:19:02	2019-02-01 07:10:52	16.0	28.0	Steuart St at Market St	The Embarcadero at Bryant St	2100	Customer

Figure 1: An excerpt from the trip dataset.

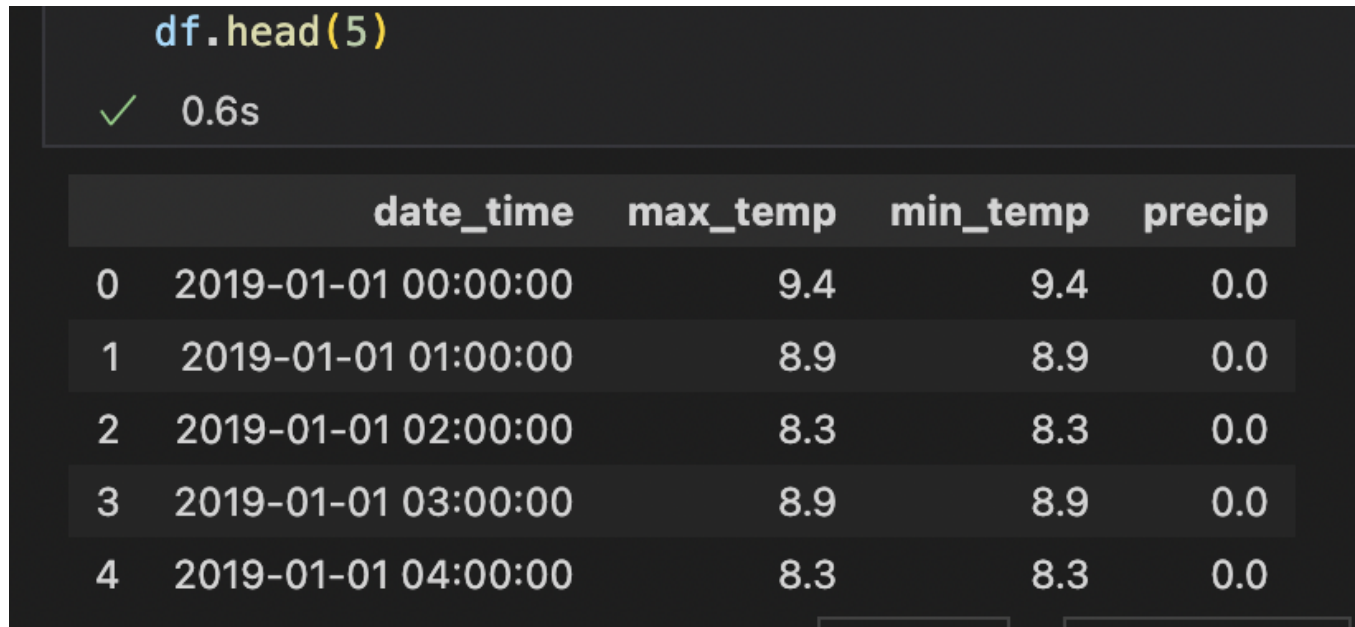
In the first step, we examined the data for missing or incomplete values, and any duplicate entries. Thankfully, we did not find any duplicate values.

To improve our further analysis, we then added additional features to the dataset such as numeric features for `start_hour`, `start_weekday`, `start_month` as well as the `trip_duration` measured in seconds.

We then combined the external data set [Sf\\_2019\\_full.csv](#) with the trips data set. Therefore, we created a look-up table including the `station_id` as well as the station's actual geo coordinates from external data set and joined the geo data on the original trip data set. The enhanced geo features allowed us to refine the dataset to only include data from stations located in San Francisco. Trips with empty `station_id` values are assumed to happen within the San Francisco area. That applies to around 3 % of all trips and has a noticeable impact on the overall demand. We also removed any trips with suspicious data constellations in the trip duration as negative trip durations or outliers. We eliminated the top 0.5% of trip durations, which were all trips that lasted longer than two hours.

## 2.2. Weather Data

The file [weather\\_hourly\\_sf.csv](#) gives us information about the weather in San Francisco in 2019. It includes the columns `date_time`, `max_temp`, `min_temp` and the precipitation `precip`. The values of `date_time` are UTC timestamps and needed to be shifted by -8 hours to match the ride data.



```
df.head(5)
```

✓ 0.6s

	date_time	max_temp	min_temp	precip
0	2019-01-01 00:00:00	9.4	9.4	0.0
1	2019-01-01 01:00:00	8.9	8.9	0.0
2	2019-01-01 02:00:00	8.3	8.3	0.0
3	2019-01-01 03:00:00	8.9	8.9	0.0
4	2019-01-01 04:00:00	8.3	8.3	0.0

Figure 2: An excerpt from the weather dataset.

For weather data cleaning, we checked for any missing values or duplicate entries, similar to the process used for the rides data set. We did not find any missing values, but 79 duplicate rows across all columns of the data frame and 371 duplicate entries in the `date_time` column.

As we did not want to drop any columns easily, we determined the ratio of hourly temperature changes across the entire dataset. Temperature changes on an hourly basis at odds of roughly 2:1. The duplicated rows in the data frame can therefore be a result of duplicated entries in the `date_time` column co-occurring with a natural steady temperature trend. Since temperature values tend to not change within the course of an hour in non-duplicates as well, we did not declare the duplicates as redundant data. Instead, we kept the entries but adjusted the `date_time` column by shifting these timestamps forward until no `date_time` duplicates remain. Dropping duplicates and interpolating new values was not necessary.

## 3. Data Analysis

### 3.1. Descriptive Analysis

#### 3.1.1. Temporal Demand Patterns

We aggregated the variable time on the different hierarchy levels month, weekday and hour. The monthly demand shows no clear trend, probably due to low temperature variation between the

seasons in San Francisco. On weekends there is less demand, indicating that bike usage is mainly for trips to and from work. This assumption is supported by the hourly demand patterns. Peaks are at the morning hours around 8am and at around 5pm. Between 1am and 5am, demand is at a minimum.

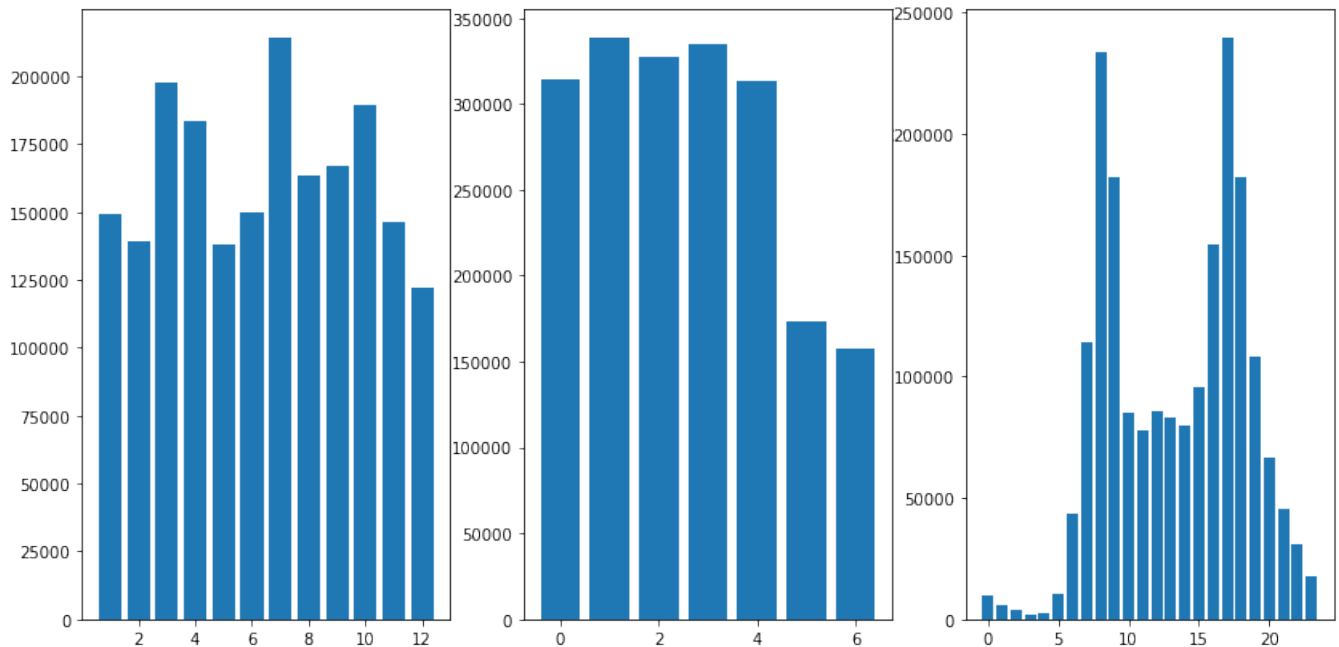


Figure 3: Ride counts plotted on monthly, weekday, and hourly basis (from left to right).

To investigate the weather dataset we plotted the average temperature over the hierarchy levels month, weekday and hour. The average temperatures get to a maximum of 19 °C in July and August and drop to a minimum of 10 °C in February. The demand pattern follows no clear temperature pattern, since there are maxima in demand in March and April. The mild climate in San Francisco seems to have no influence on the demand since the temperature range allows bike trips all year long.

The heat map displays weekly trips through the combination of weekday and hour dimensions. It highlights that weekdays are busier than weekends during the peak commuting hours of 8 am and 5 pm, possibly indicating the presence of people traveling to work.

### 3.1.2. Geographical Demand Patterns

The heatmap plots geographical locations of stations by latitude and longitude. Stations that are used frequently are yellow to red, and stations used sparsely are coloured purple. The main activity takes place at daytime in downtown San Francisco, more specifically in Valencia Street and Market Street. Other frequent stations are located around King Street and the docks. At nighttime stations are less frequently used and turn purple or disappear from the map. Interactive use of the heatmap is not possible via PDF, but with the notebooks in the repository that is linked.

Plotting the monthly interactive Heatmap shows that trips taken at the beginning of the year are more centralized and trips taken towards the end of the year spread across a greater area, especially in the western city.

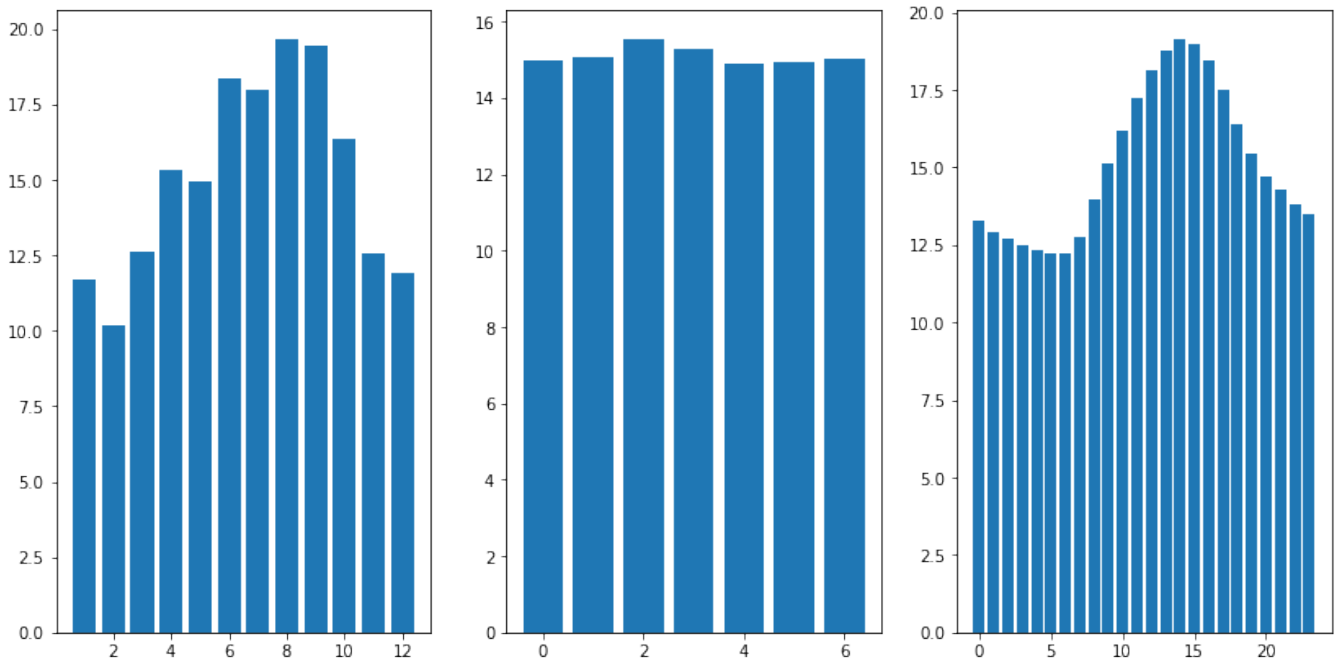


Figure 4: Temperatures plotted on monthly, weekday, and hourly basis (from left to right).

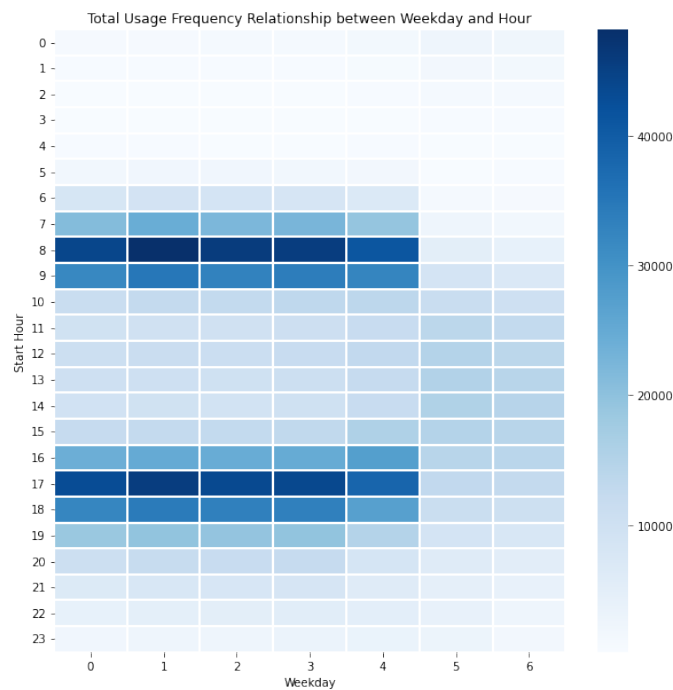


Figure 5: Heat map that shows the relation between weekday (x-axis), hour (y-axis) and trip count (colour).



### 3.1.3. Key Performance Indicators (KPIs)

To monitor the fleet operations from a management perspective, we introduced the following KPIs:

First, we investigate the ratio of trips taken by subscribers and non-subscribers to differentiate between these pricing models. The average share of trips taken by subscribers across the whole dataset is around 79%. Towards the end of the year, the ratio drops significantly.

To answer the question of how often it is necessary to relocate a bike in a given time interval, we introduced the 'Relocations' KPI. Therefore, we compared end stations to subsequent start stations to determine if relocations occurred. There is a peak of relocations at the end of July that could be potentially due to maintenance efforts.

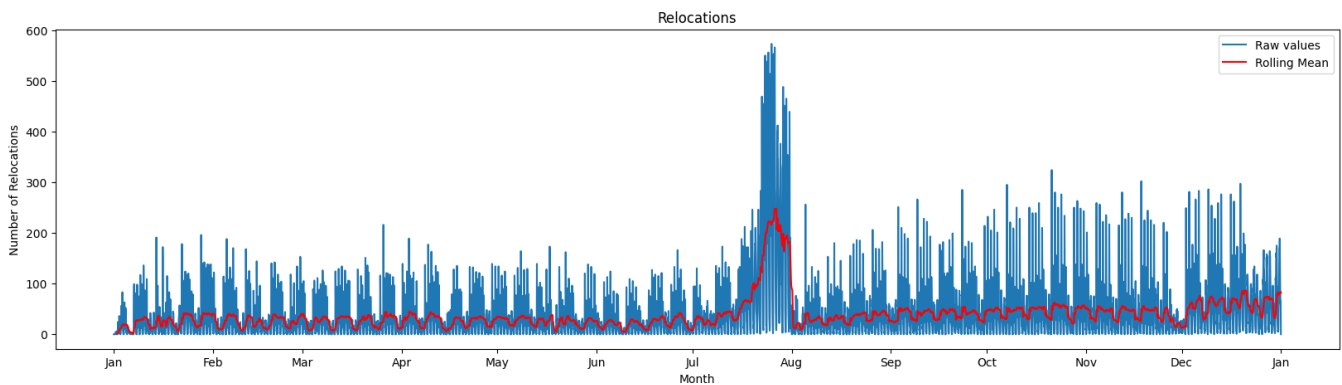


Figure 6: Plot that shows the number of relocations per hour.

Our third KPI shows the distinct bikes in use per hour. On average, there are around 200-300 bikes in use per hour. Furthermore, the weekday patterns are identifiable. The peak at the end of July could be related to the peak of the 'Relocation' KPI. To monitor the distribution we compared trips taken with distinct bikes. Values close to zero indicate a bad distribution of the bikes, whereas values near to one mean that the number of bikes used and trips taken is almost identical and thus indicate a good distribution of the bikes.

As our fourth KPI, we chose the revenue per hour. On average, Lyft has a revenue of about 900 USD per hour in San Francisco. The range of revenue per hour lies between a minimum of 3.49 USD and a maximum of 5760 USD.

We only show the plot for the number of relocations here as an example. You can find all the other plots in the script.

## 3.2. Cluster Analysis

### 3.3. Trip Clustering

Our trip clustering was based on the features `start_hour`, `start_weekday`, `start_month`, `is_weekend`, `user_type` and `trip_duration`. Method wise we used the k-means clustering approach since it guarantees convergence and it scales to large datasets. On the downside we had to choose `k` the number of cluster manually. Therefore, we did a grid search to identify the elbow. After comparing `k=3` and `k=4` clusters we preferred to set `k=3`, because additional cluster mainly just

split the `user_type` in two different daytime clusters. The 3 resulting clusters all were sufficient in size and easily distinguishable, leading to a high content of information per cluster.

0. 'Early Rides to work': Customers are characterized by a short trip duration compared to Cluster 1. They are taking trips in the typical work hour pattern described in the analysis section, but only in the morning hours. The cluster size of Cluster 0 and Cluster 2 are evenly distributed.
1. 'Weekend rides': Weekend rides are distinguishable by their weekday distribution, taking trips exclusively on weekends. Users on weekends are taking less short rides and longer trips, assuming that these are related to leisure activities. In terms of customer type the distribution shows no large deviation to the customer trip ratio of around 79% across the whole dataset.
2. 'Late rides from work': The most distinctive feature is the time of day as of cluster one, but including only evening hours. Their subscriber share resembles the one of cluster 1 related to typical working hours, with only weekdays being included in the cluster as well.

Analyzing the clusters created by K-Means algorithm, we presumed that the monthly feature is of less distinctive character, what was confirmed in the predictive analysis.

### 3.3.1. Weather Clustering

The following features were used for K-Means Weather Data Clustering: precipitation, maximal temperature, months, weekday and hour. And we identified for  $k = 3$  the clusters:

0. 'Nightriders': Cluster 0 has a greater share of rides with precipitation and a lower average temperature. Rides are distributed around early morning hours and late evening hours when temperatures are obviously lower.
1. 'Workers': This cluster reconstructs the work and weekday patterns of the trip clustering. Temperature wise, the cluster is evenly distributed.
2. 'Midday rides': Cluster 2 has the highest average temperature and rare precipitation. Trips are taken around midday more often compared to the other clusters. There is no weekday variation.

The clustering of the weather data shows once again that neither temperature nor precipitation have a significant influence on demand. The identified clusters rather describe temporal correlations.

### 3.3.2. Station Clustering

The station clusters are computed based on location and usage. Therefore, they show the stations that are heavily frequented, less frequented and an in between cluster. As described in the Geographical Data Analysis, there are stations that can be characterized by being used less frequent and being distributed across a large area. These are grouped together in the cluster called

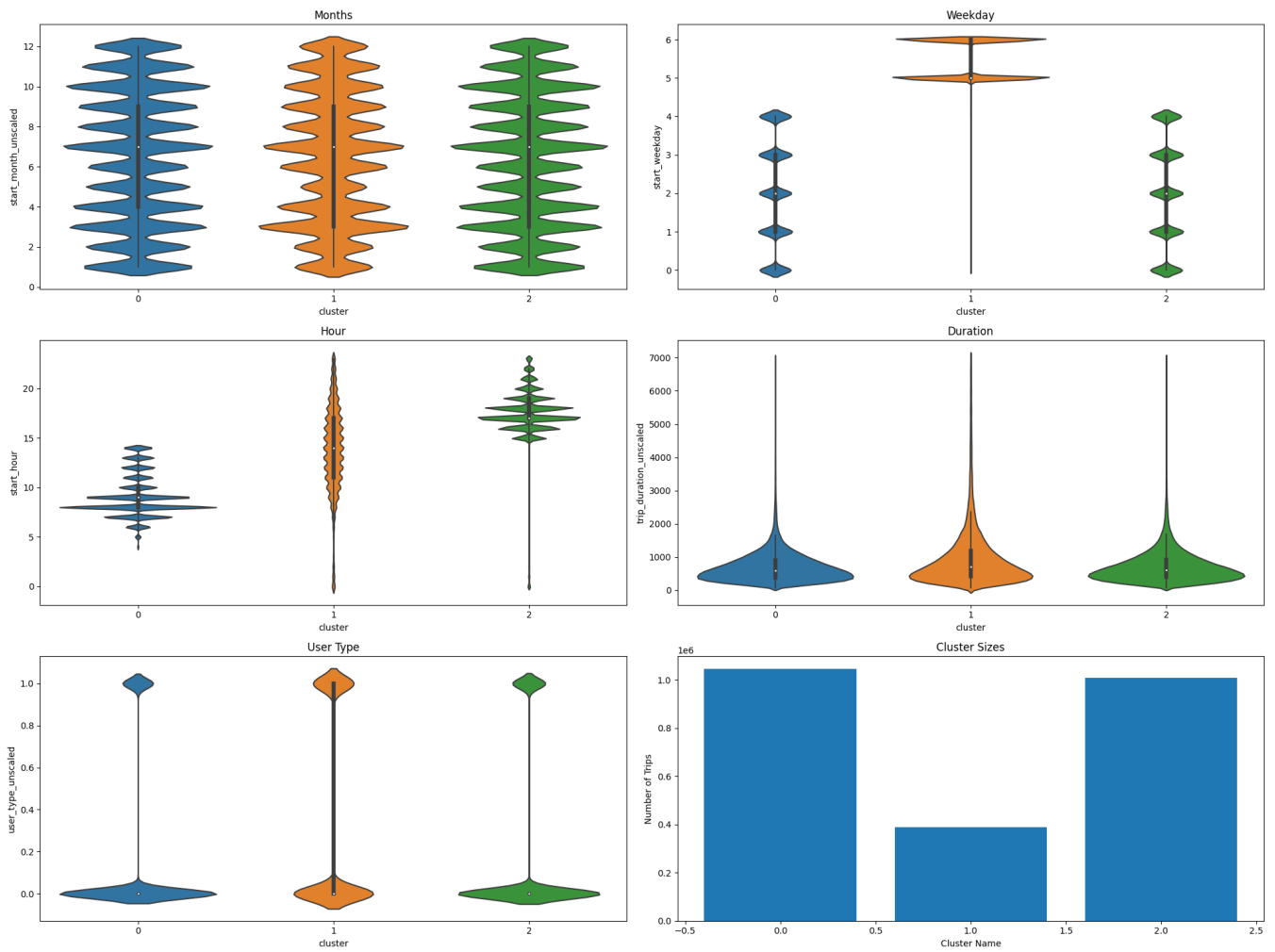


Figure 7: Trip clustering based on start\_month, start\_weekday, start\_hour, user type, trip duration and weekday/weekend.

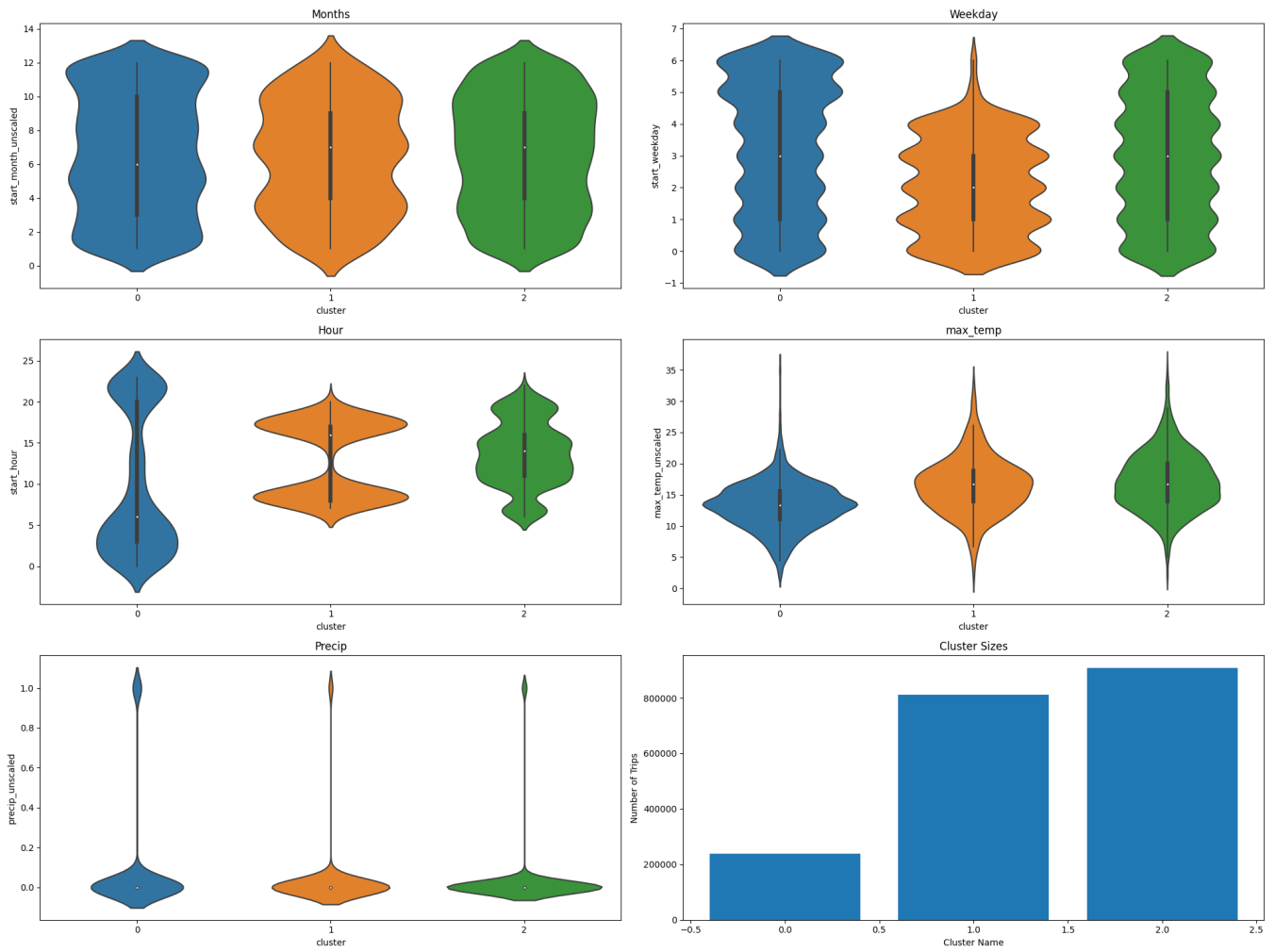


Figure 8: Weather clustering based on month, weekday, hour, max. temperature and precipitation.

'Low Performer'. The 'High Performer' - cluster are exactly the stations along the market street together with the eastern stations close to the docks. These are the stations that are frequented the most. Stations in the 'Medium Performer' - Cluster are located mainly in downtown as well, but spread around a greater area than the 'High Performer' - Cluster. These stations are also popular, but not as popular as the 'High Performers'

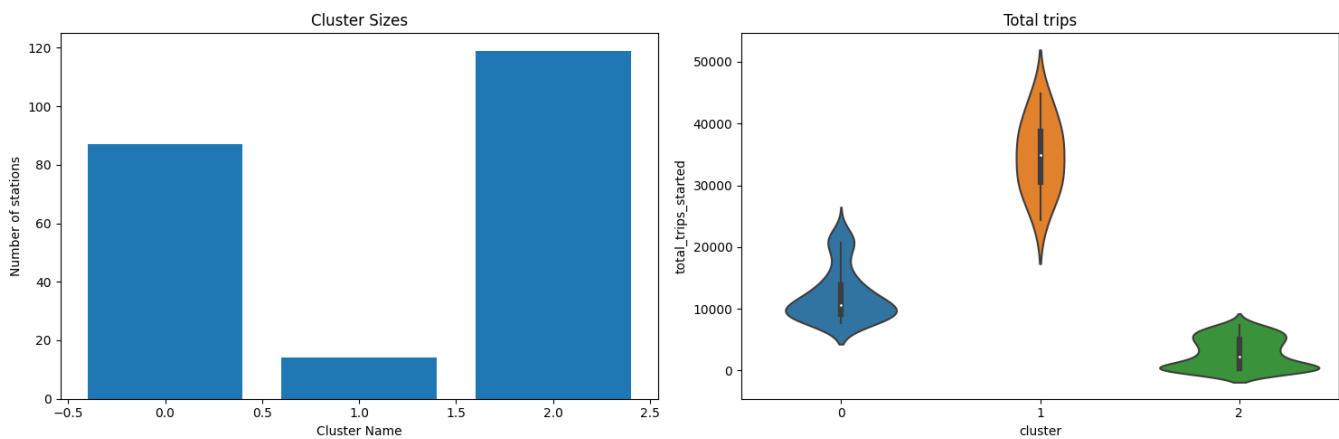


Figure 9: Station clustering based on location and usage.

### 3.4. Predictive Analysis

#### 3.4.1. Preparation

First, we created a small data set for predicting the hourly usage of the bike fleet. Bike trip data is aggregated, and the hourly demand is calculated. Location data is not part of the `prediction_df` since the hourly demand of the whole bike fleet needs to be determined. Information on station granularity is therefore not considered. The weather data is reduced to the columns `max_temp` and `precip`.

#### 3.4.2. Feature Engineering

We added additional features based on the analysis and clustering. Since a significant temporal correlation was discovered, these are for example seasonal features and lag features. We displayed the correlation of these features in the correlation matrix. We used a high correlation with the feature `'number_of_trips'` for assessing features.

#### 3.4.3. Grid Search Validation and Train/ Test Splitting

To find the best combination of parameters for our models, we are using `GridSearchCV`. It executes an exhaustive search over all parameters. `GridSearchCV` splits the data into multiple training and test sets with the same percentage of samples (cross validation). By default, the `'KFold'` algorithm is used which assumes independent samples and is not suitable for our time series dataset. This is because time series data always has the same sort of dependency to data from the past.



Figure 10: KFold and TimeSeriesSplit Algorithm. Source: <https://scikit-learn.org>

As shown in the graph 10, the KFold algorithm would not ensure that the model works only on training data, instead, we are using the TimeSeriesSplit algorithm which is a drop-in replacement. TimeSeriesSplit ensures that a fitted model is only evaluated on ‘future’ data.

For a final evaluation after the grid search, especially for visualization purposes we cannot simply take the train/test samples that are generated using TimeSeriesSplit, because after the best parameters are found, the model is refitted on the whole dataset. Therefore, we preserve a 30% share of our whole dataset for the final re-evaluation.

#### 3.4.4. Predictive Models

Three different regression models (Random Forest, K-Nearest Neighbors, and XGBoost Random Forest) were used to predict an outcome based on a dataset. The models were first tested with default parameters, and then fine-tuned hyperparameters using GridSearchCV. For performance evaluation, we chose the  $R^2$  score as an easy-to-understand measure. The score approximates how well the regression models approximate the real data points. Deviation in model performance was low with around 1%. The K-Nearest Neighbors model showed the best performance with an  $R^2$  value of 0.956773.

Model Name	Unoptimized	Optimized
RandomForestRegressor	0.954284	0.954296
KNeighborsRegressor	0.955088	<b>0.956773</b>
XGBRFRegressor	0.946470	0.955571

Table 1: Comparison of the  $R^2$  value of the different models.

The most important features for the prediction were found to be the lag values from one week before, as well as the `is_workday` feature. Temperature and precipitation were found to have little impact on the prediction. Since K-Nearest Neighbors does not allow feature importance analysis, the values were calculated based on XGBRFRegressor and are show in figure 12.

Further improvement could be achieved by using enhanced weather data for the features. The mild climate in San Francisco with low precipitation and temperature ranges allowing riding a bike all year long have a little influence on the demand. Possible enhancements could be achieved through wind strength and direction, since San Francisco’s location in a bay area.

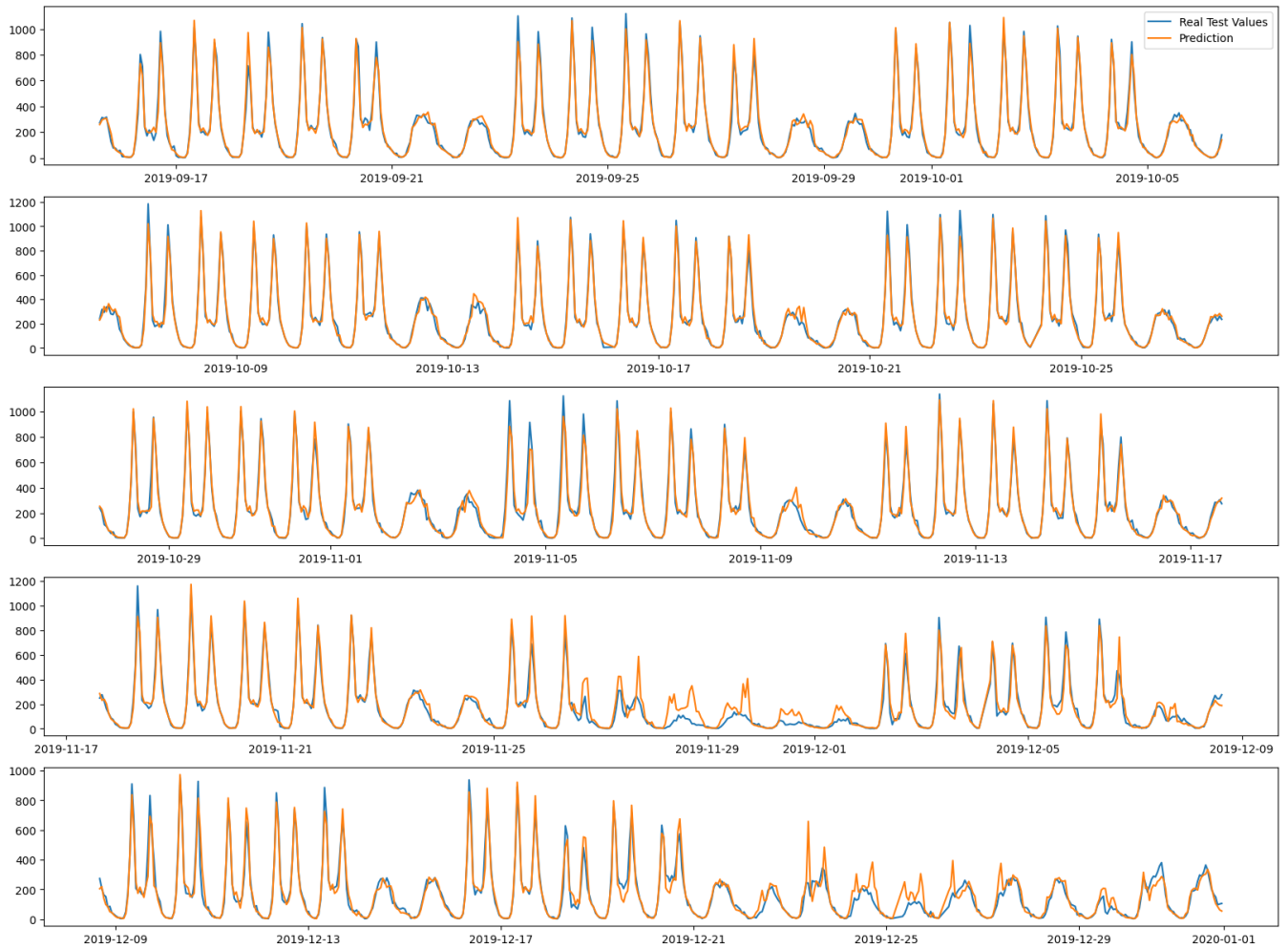


Figure 11: Comparison between values predicted by the KNN model and real test values.

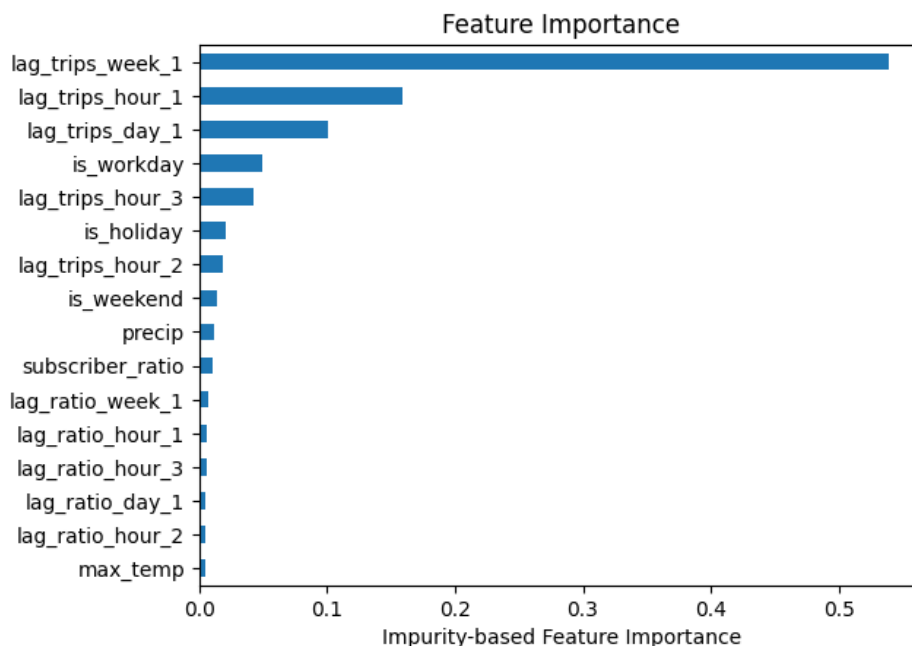


Figure 12: Feature importance as returned by the XGBoost Random Forest Regressor.

In case of the selected prediction models, improvements can be achieved by an advanced hyperparameter tuning. Our computational resources are limited, and especially tuning the XGBoost hyperparameters is cost-intensive. For reference, we show a detailed visualization of the effect of hyperparameter tuning in the appendix [A](#) (using the KNN model as an example).

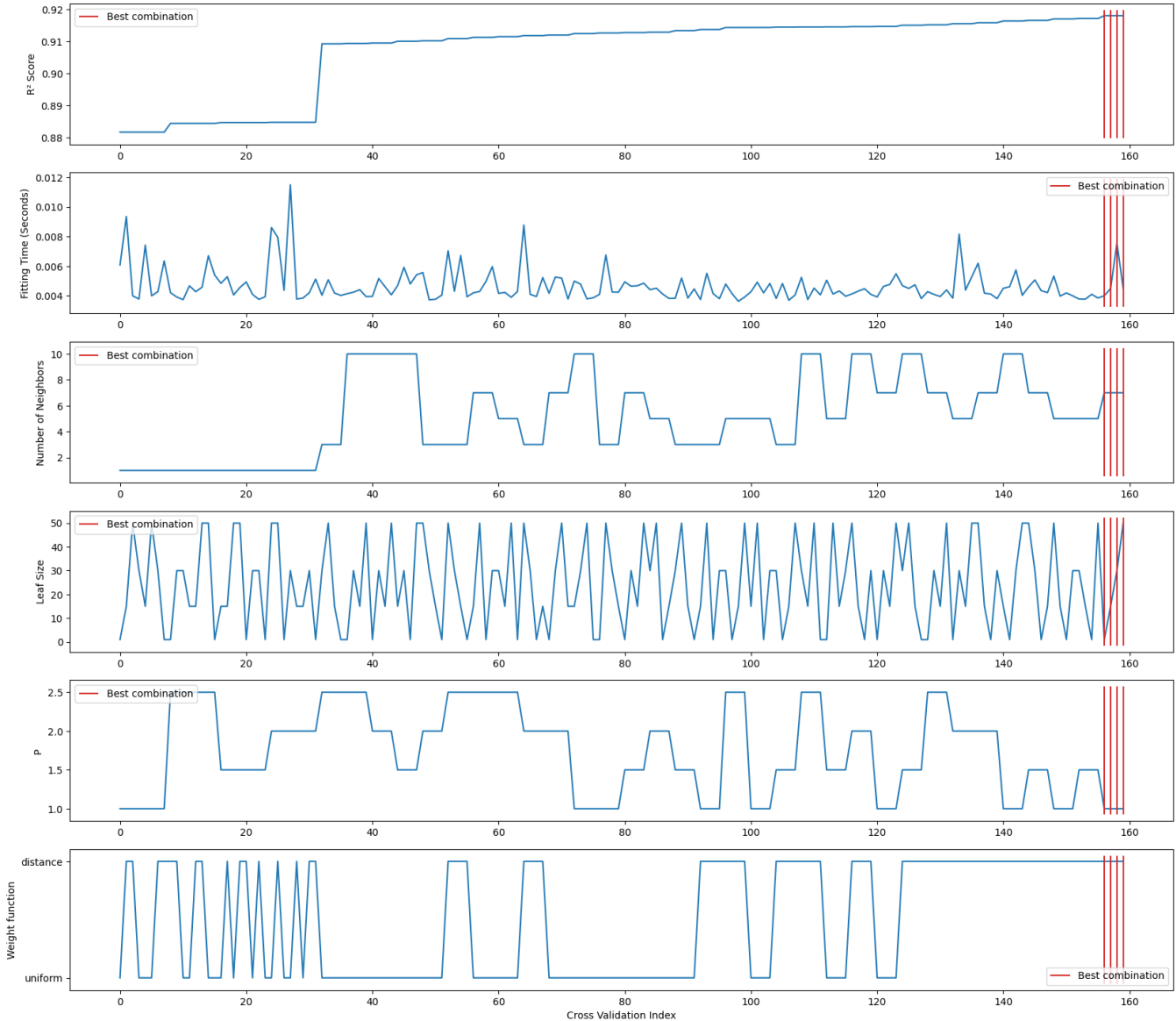
## 4. Conclusion

Our goal was to inform and improve business decisions for the bike-sharing platform Lyft. The analysis of the bike-sharing data has provided insights into the temporal and geographical demand patterns, as well as customer and time-based clusters. Based on the findings, it is recommended that Lyft focuses on increasing the availability of bikes at stations located in downtown San Francisco during the weekday morning and evening rush hours, as well as at the docks. Additionally, Lyft should provide promotions and incentives for short-trip customers during the weekdays, and for long-trip customers on weekends. Finally, as bike sharing is a popular option for the 'last mile connectivity' on the customer's way to work, Lyft should consider corporate subscriptions in the B2B market. The scope of the project was limited to data from only one year of the bike-sharing system, which could be expanded to multiple years to provide more comprehensive findings and analysis of usage and system growth. Additionally, the dataset's features were limited, such as the lack of information on user demographics, which would have allowed for more advanced user clustering and segmentation. In terms of revenue calculation, the results were limited by the fact that revenue calculation for 2019 data was based on the 2023 pricing model.



## A. K-nearest Neighbors hyperparameter optimization

This figure shows the importance of different hyperparameters of the K-nearest Neighbors model regarding the  $R^2$  score.



## B. Distribution of Work

We split up on the implementation and documentation between our group members, the following list shows who was mainly involved in which task:

- Weather Data Cleaning: Tobias
- Bike Data Cleaning: Leo, Manuel

- Geo Data Cleaning: Laurenz, Leo, Manuel, Tobias
- KPI: Laurenz, Tobias
- Descriptive Analytics: Leo, Manuel
- Bike and Weather Clustering: Leo
- Geo Clustering: Laurenz, Leo, Manuel, Tobias
- Predictive Analytics: Leo, Laurenz
- Report: Laurenz, Leo, Manuel, Tobias