

Analytics and Applications

WS 21/22

Instructor Prof. Dr. Wolfgang Ketter
Teaching Assistant Nastaran Naseri

Team Assignment Report

Dataset Chicago 2019



Team 06 Data Miners CABIB

Carsten Stukenborg
Andrej Kotsovolos
Bastian Schneider
Isabel Wittmann
Björn Reibke



Department of Information Systems for Sustainable Society
Faculty of Management, Economics and Social Sciences
University of Cologne

Executive Summary

The increasing competition between different operators of bike sharing systems is becoming an important factor that has to be taken into account when trying to offer a viable alternative to present modes of public transport, especially under the light of the soaring debate regarding climate change. In order to make the day-to-day operations more efficient and value-adding, the goal of this project was to apply cutting-edge data analytics on transactional ride data. Thus, we performed analytical investigations and applied several Machine Learning methods in the context of data derived from the Chicago divvy bike fleet in 2019. This data comprised records of all individual rides that took place, the associated weather during the rental and the geographical information where the ride started and ended. For this project, the data was analyzed, clustered and various algorithmic models were applied to predict the hourly demand of the bike fleet. We start by elaborating the main findings by stating that a seasonality and geographical demand pattern exists. Both the usage and utilization are highest during the summer and lowest during the winter months. Furthermore, the weather is correlated with the demand, as warmer temperatures lead to higher and rainfall leads to a lower usage. Regarding the geographical pattern, a three-ring structure was observed, which contains the core of the city with the most dense network, the surroundings of that core that are still centrally located in Chicago, and lastly, the outskirts, where only a minor amount of rides is started. Next, the revenue from customers is volatile and peaks during summer, while the revenue from the subscriptions is quite stable. The patterns observed for the usage of the bike fleet and duration of the trips apply likewise to the revenue. Therefore, the mean revenue per trip is higher in summer than in winter due to the on average longer duration of trips, attributed to the weather. The mean revenue per trip per day is the highest on the weekend. Moreover, long rides during the summer seem to account for the absolute majority of the system's revenue, stressing the importance of such leisure rides for the platform. It could hence be advisable to approach those customer groups specifically, for example through targeted advertising. When taking into account the fact that many of the heavily utilized stations are located close to recreational sites, the importance of said group is emphasized even more. Next, the number of rides per individual bike was compared with typical benchmark values. This results in the insight that during the summer, a favorable target value is met, however during winter, spring and autumn, this figure is comparably low. We suggest that a reduction of the number of bikes during those seasons could be advisable, or at least a follow-up optimization project of bike allocation. Furthermore, a daily pattern can be observed, which can be attributed to commuters. People ride to work in the morning hours and return in the afternoon. The number of rides for the afternoon trips are higher than for the morning trips, so it is suspected that additional leisure trips begin after work. Likewise, the coverage of the individual stations was investigated. It was discovered that the vast majority of the stations is equipped with a sufficient capacity. Only very few stations come near to a situation where they can be considered overloaded. However, especially in the more peripheral regions of Chicago, the capacity seems to be constantly too high, as their maximum capacity is nearly never reached. Consequently, it could be wise to reduce the number of stations or the capacity of those stations. Moreover, the KPIs referring to rides per unique bikes and to the ratio of demand and station capacity indicate that commuters are the main drivers of high figures for those KPIs. Similar to the leisure riders during summer, as described above, targeted advertising could be performed to attract more of those valuable customer types. Besides the aforementioned main findings, the development of a relatively accurate model for demand prediction was accomplished which can be applied on an hourly basis in the operational business. This can enhance the ability to respond to time- and weather-related changes and thus, reduce the cost of operation. However, we strongly recommend focussing more thoroughly on the aspects of prediction to ensure even more precise predictions, possibly on a bike station level.

Contents

1 Problem Description	1
1.1 Business and Data Mining Goal	1
2 Data Description and Preparation	1
3 Descriptive Analytics	1
3.1 Temporal Demand Patterns and Seasonality	1
3.2 Geographical Demand Patterns and Top X Stations	2
3.3 KPI 1: RPUB (Rides Per Unique Bike)	4
3.4 KPI 2: Demand-Capacity (Implementation of the Concept of Coverage)	6
3.5 KPI 3: Average Rental Durations	8
3.6 KPI 4: Revenue	8
4 Data Analytics	9
4.1 Cluster Analysis	9
4.2 Predictive Analysis	16
5 Conclusion	19
A Appendix	20
A.1 Data Description	20
A.2 Temporal Demand Patterns and Seasonality	20
A.3 KPI 1: RPUB (Rides Per Unique Bike)	24
A.4 KPI 2: Demand-Capacity	26
A.5 KPI 3: Average Rental Durations	27
A.6 KPI 4: Revenue	28
A.7 Cluster Analysis	28
A.8 Predictive Analysis	44
B References	49
C Supplementary Document	50

List of Figures

1	Fleet usage	2
2	Daily bike utilization of fleet for April	2
3	Top 15 stations in terms of starting rides	3
4	Histogram for the distribution of station interactions over all stations	3
5	Logarithmic number of starting rides per station	4
6	Average RPUB (KPI 1) per day	5
7	Average hourly RPUB (KPI 1)	5
8	Average RPUB (KPI 1) with unique daily bikes - Comparison weekdays and weekends	6
9	Maximum utilization category count	6
10	Average Demand-Capacity (KPI 2) value per hour	7
11	The average maximum Demand-Capacity (KPI 2) value per day of the 20 most utilized stations with standard deviation	7
12	Average hourly rental duration aggregated over the whole year	8
13	Revenue in 2019 for user types including annual subscription	9
14	Cluster result with 3 cluster for input feature duration	9
15	Cluster result K-Means	10
16	Cluster result K-Means with and without precip	10
17	Classification of the average hourly revenue over both customer types using Gaussian Mixture Models	11
18	Share of each cluster on revenue per month	12
19	Classification of stations with 3 centers based on their average KPI2 value	12
20	Clustered stations according to their Demand-Capacity (KPI2) value	13
21	Development of the average daily Demand-Capacity (KPI2) value per cluster over 2019	14
22	Number of total rides starting in 2019 per station and average ride duration per station	14
23	Stations clustered based on the number of starting rides and average ride duration	16
24	Predictive analytics results	16
25	Best L1 model - Comparison of actual vs predicted values for demand on the test set and distribution of absolute difference between actual and predicted demand values	17
26	Hyperparameter evaluation of number of estimators	18
27	Random Forest - Comparison of actual vs predicted values for demand on the test set and distribution of absolute difference between actual and predicted demand values	18
28	Fleet usage over the year 2019	20
29	Bike utilization over the year 2019	21
30	Daily bike usage of fleet with precip for each month	22
31	Daily bike utilization of fleet with precip for each month	23
32	Variance of hourly RPUB split by Warmday and Rainday	24
33	Relationship between maximum temperature and RPUB	25
34	Average hourly KPI 2 value grouped by month	26
35	Average hourly KPI 2	26
36	Average Demand-Capacity per day per month	27
37	Results KPI 3 average rental durations	27
38	Monthly revenue in 2019 per user type	28
39	Mean revenue per trip per day in 2019	28
40	Sum of squared distances (loss) of samples to their closest cluster center for each number of cluster for duration	28

41	Silhouette score for each number of cluster for duration	29
42	Cluster results of Gaussian Mixture Model	29
43	Sum of squared distacnes (loss) of samples to their closet cluster center for each number of cluster (duration and temperature)	30
44	Number of clusters vs. silhouette score (duration and temperature)	30
45	Cluster results K-Means - duration and temperature	31
46	Cluster results Gaussian Mixture Model - duration and temperature	32
47	Silhouette score for clustering based on revenue	33
48	Optimal number of clusters using K-Means instead of GMM based on revenue	34
49	Clustering of the average hourly revenue over both customer types using K-Means	35
50	Optimal number of clusters based on Demand-Capacity	36
51	Optimal number of clusters	37
52	Silhouette score for clustering based on demand-capacity	38
53	Clustering of the average hourly revenue over both customer types and the average hourly temperature	39
54	Classification of stations based on their average yearly temperature and their average yearly KPI2 value	39
55	Stations clustered based on KPI2 and average yearly temperature	40
56	Classification of the number of total rides startig in 2019 per stations	41
57	Average number of starting rides per hour over 2019 split by cluster	41
58	Stations clustered according to the number of starting rides	42
59	Optimal numbers of clusters for numer of starting rides per station and average duration .	43
60	Developed features	44
61	Schematic representation Random Forest (taken from Jagannath, 2020)	46
62	Cost complexity of a tree	47
63	XGBoost result - Comparison of actual vs predicted values for demand on the test set and distribution of absolute difference between actual and predected demand values	48

List of Tables

1	Data cleaning and preparation steps	1
2	Cluster rings	15
3	Data description of most relevant data inputs	20
4	Final hyperparameter values for Random Forest	45
5	Different model results for Random Forest	45
6	Final hyperparameters results for XGBoost	48
7	Project responsibilities	50

1 Problem Description

Note: A more detailed version of this report can be found in our git repository:
<https://github.com/bschne11/DATAMINERS-CABIB>

1.1 Business and Data Mining Goal

As bike sharing platforms are a part of the transformation of the mobility system towards mobility as a service and on-demand, they are relatively new and have a potential to improve the ecological impact of transportation, especially in urban areas. In order to leverage the whole potential of a bike fleet operator, data analytics can be used. The business goal of this project is to monitor and optimize operations, boost profitability and increase the service level of fleet operators. First, the project focuses on system monitoring, which provides information about the usage and utilization of the fleet. Keeping track of the fleet's performance should be the basis for all operational and business decisions. Secondly, predicting demand precisely can lead towards a high service level by using the information to allocate the right number of bikes at the right time. The data mining goal of our project is to gain a deep understanding of the daily, seasonal and geographical bike fleet usage patterns, to identify clusters and to predict the hourly total system demand using the available data for the year 2019.

2 Data Description and Preparation

The provided dataset of bike sharing rentals in Chicago in 2019 is cleaned and prepared for descriptive analytics, clustering and predictive analytics. Data cleaning includes handling missing, erroneous and non-contextual data (Table 1). The available data is described in (Appendix Table 3).

Step	Description
Null and NaN values	These values mark missing data or erroneous data not representing a valid number. This type of data has been removed.
Duplicate values	Duplicate values are based on duplicate rows, lead to redundant data and have been removed.
Negative or non-contextual data	Negative values, for example negative duration times, have been removed. Further data anomalies are discovered which potentially lead to context bias. These include, in particular, the shifted weather data. These were shifted by 5 hours so that minimum and maximum temperature data correspond to the typical course of the day.
Merge and interpolation	The weather data and divvy bike set data were merged. The merge of these two separate data sets and the eliminated data from previous steps resulted in data losses which were compensated by linear interpolation.

Table 1: Data cleaning and preparation steps

3 Descriptive Analytics

3.1 Temporal Demand Patterns and Seasonality

The Usage and Utilization metrics are introduced on a monthly, daily and hourly basis and further contrasted with the temperature and precip (rain/snow) rate. The usage of bike fleet describes how many rentals are started on the respective aggregation level. The utilization rate is calculated by the ratio (x/n), where x describes the number of unique bikes, based on their bike ids, that have been used at least once and n is assigned the total number of unique bikes. The summer months record the highest

usage and utilization numbers and the winter months the lowest ones. This is referred to as annual seasonality, attributed to the seasons and can be observed in Figure 1, Appendix 28 and Appendix 29. The positive correlation with temperature values supports this observation. Furthermore, a daily pattern can be seen and can be attributed to commuters. People ride to work in the morning hours and return in the afternoon. Afternoon trip numbers are higher than morning trips, so it is suspected that additional leisure trips begin after work. Finally, the negative correlation between rainfall and usage or utilization should be mentioned, visualized in Figure 2, Appendix 30 and Appendix 31. High rainfall rates indicate low utilization rates or usage numbers and vice versa.

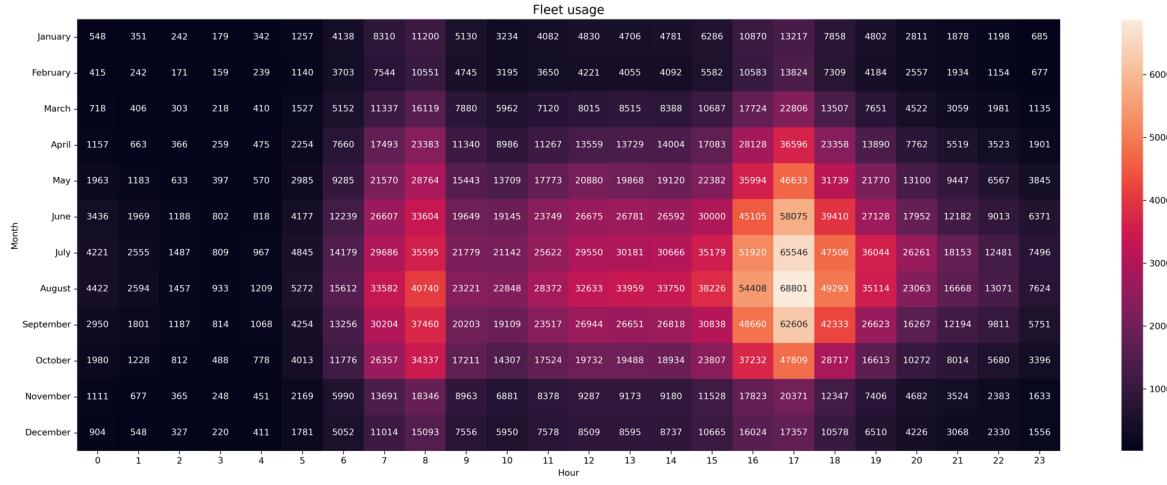


Figure 1: Fleet usage

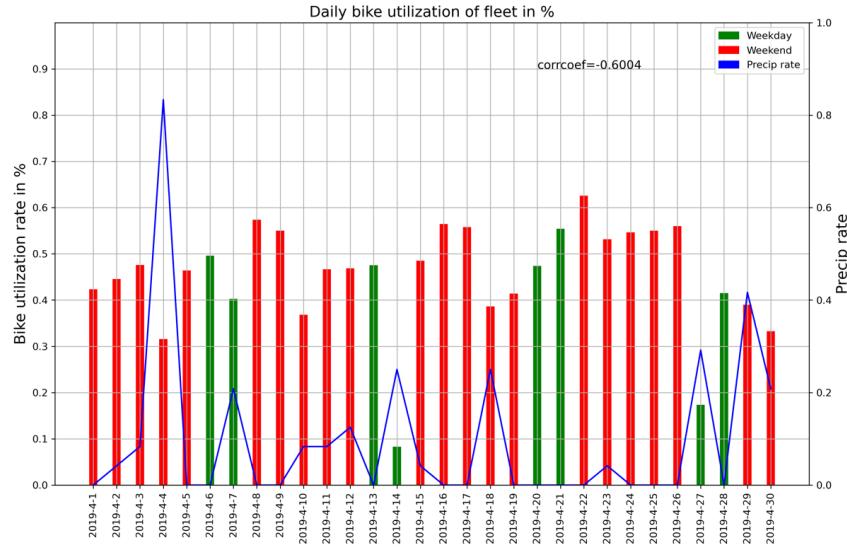


Figure 2: Daily bike utilization of fleet for April

3.2 Geographical Demand Patterns and Top X Stations

The following section deals with the investigation of the utilization of the individual stations. Several metrics were used to investigate the stations under different perspectives. The first metric we took was to measure the total sum of rides that start and end at each station and to plot the most heavily utilized stations under this definition.

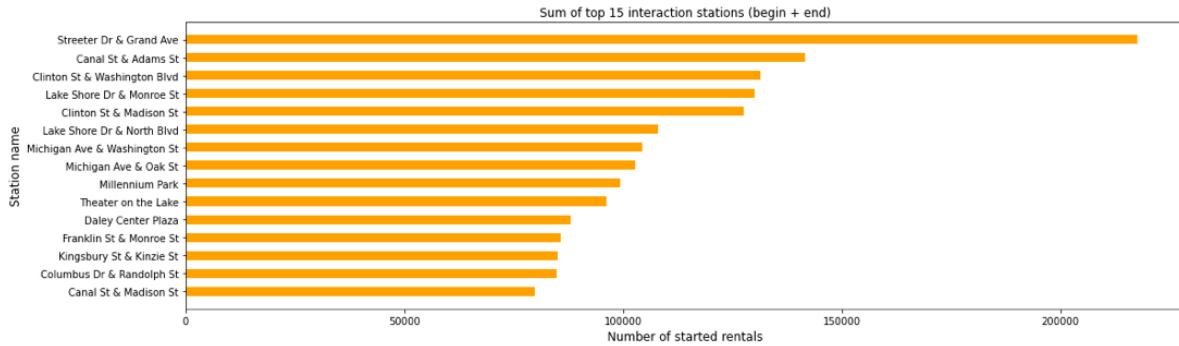


Figure 3: Top 15 stations in terms of starting rides

Figure 3 illustrates the respective top 15 stations. As can be seen in the plot, the first five stations have a significant lead compared to the places 6 to 15. As a dashboard user, these stations can be seen as most important and should be monitored carefully to avoid longer maintenance. Building further upon those 15 stations, a highly skewed distribution of station interactions can be seen in Figure 4.

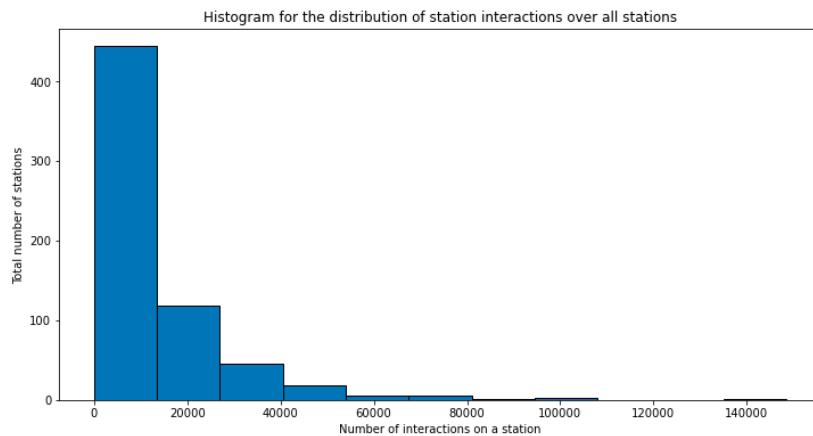


Figure 4: Histogram for the distribution of station interactions over all stations

Most stations only have a limited number of interactions. However, the top 50 stations already account for one third of the overall interactions. Also, the 0.75-quantile is very meaningful, stating that 75 percent of all stations have 16,850 or less interactions. Nevertheless, one should not diminish the importance of these stations, as 31 percent of all interactions are represented by them, stating that also the smaller stations are vital for daily business.

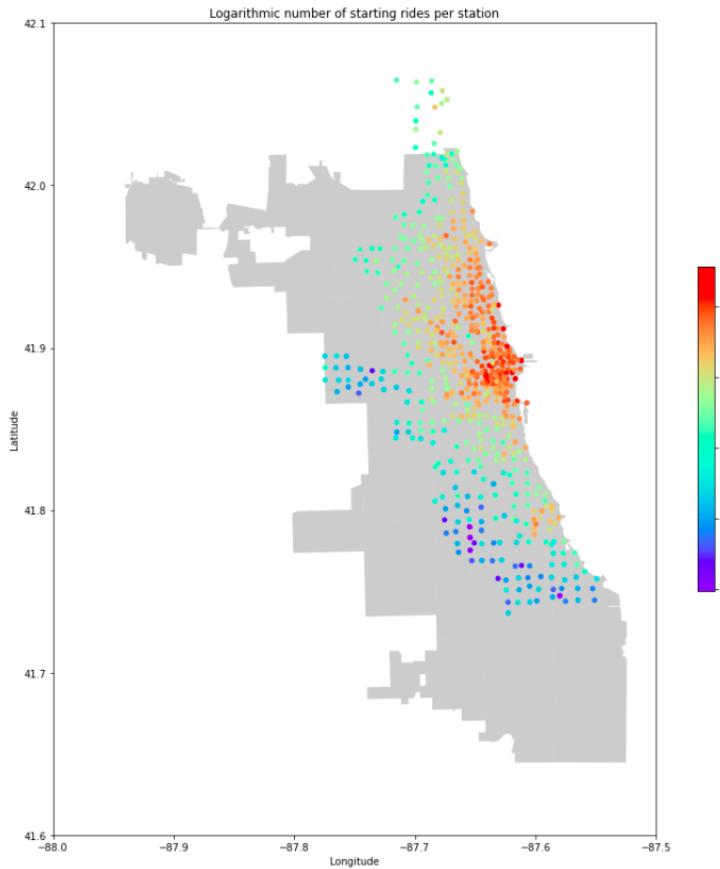


Figure 5: Logarithmic number of starting rides per station

Another perspective was taken in Figure 5. Here, the logarithmic number of starting rides per station is plotted. It can be easily seen that most of the rides start in the center of Chicago and along the coast, whilst moving to the outskirts of the city decreases the density of rides. The data can thus be clustered into three areas: The core of the city, with the most dense network, then the surroundings of that core that are still centrally located in Chicago and lastly, the outskirts, where only minor amounts of rides are started. We will refer to this structure as "three-ring structure" for the rest of this assignment.

3.3 KPI 1: RPUB (Rides Per Unique Bike)

The first KPI scrutinizes the number of trips per unique bicycle. The concept of "unique bicycle" is defined as all the bikes counted uniquely per day. The KPI is derived from an article by Sven Boor (2019) and an article by Yanocha, Mason, Patlán, Benicchio et al. (2018), although we adapted the KPI by also calculating hourly values, not only daily ones, in order to fulfill the task. The first insight that can be derived is that a clear seasonality can be exhibited with the highest values during summer and the lowest during winter (Figure 6).

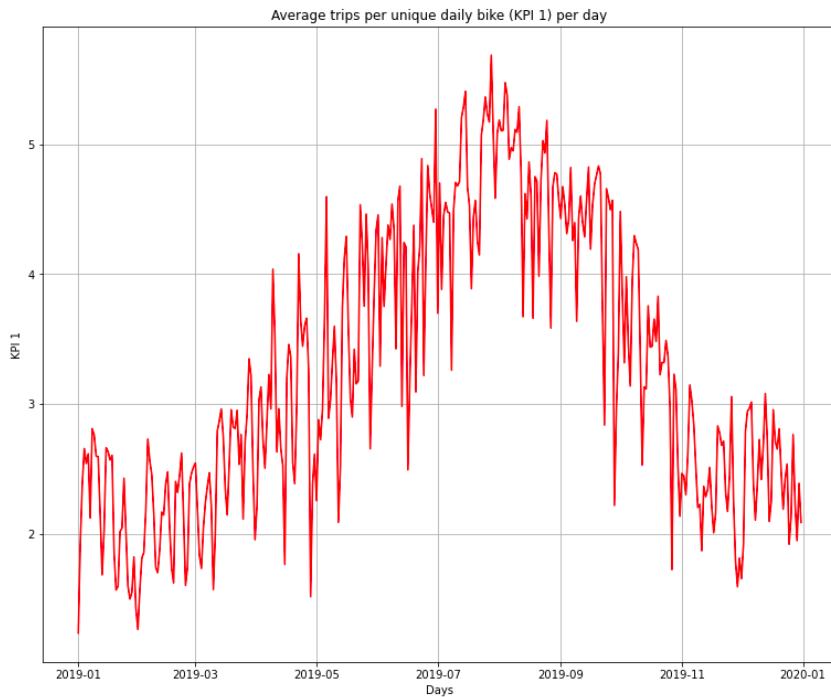


Figure 6: Average RPUB (KPI 1) per day

RPUB based on unique daily bikes always lies within the target interval defined by Boor (2019) of 4-8 trips per bike. During summer, it averages around a value of 5, which is appropriate, however during winter, spring and autumn, it falls below the lower boundary of 4 defined by Boor (2019). This could indicate that the number of bikes during winter is too high.

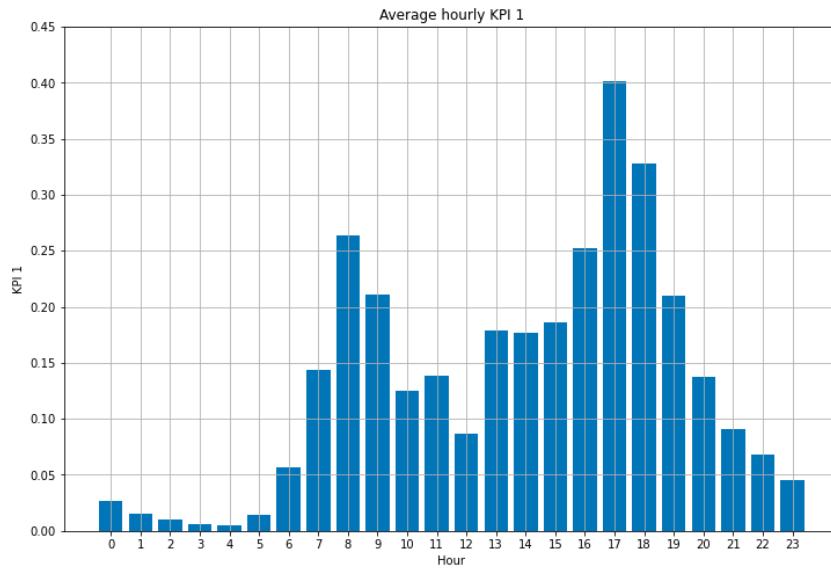


Figure 7: Average hourly RPUB (KPI 1)

When looking at the hourly RPUB values averaged over the entire year (Figure 7), a very explicit commuter pattern can be observed, as the values peak at around 8 and 17 o'clock. Interestingly, at 5 o'clock, the KPI stands at around 0.4, which is more than twice as high than the average monthly value in July, the most "active" month. A bike at that time performs around 0.4 rides per hour. Thus, commuting activity has a tremendous impact on the utilization of the divvy system, much more than

monthly seasonality. Comparing weekend and weekdays hourly (Figure 8), it becomes clear that the weekend RPUB values are on average not higher or lower than the weekday ones, except for the situation during noon and at around 9/17 o' clock. Thus, it can again be reasoned that commuting is a significant driver of the divvy bike system and the peak around noon is attributed to leisure activities. As the influence of the weather data was rather obvious, it can be consulted in the Appendix (32 and 33).

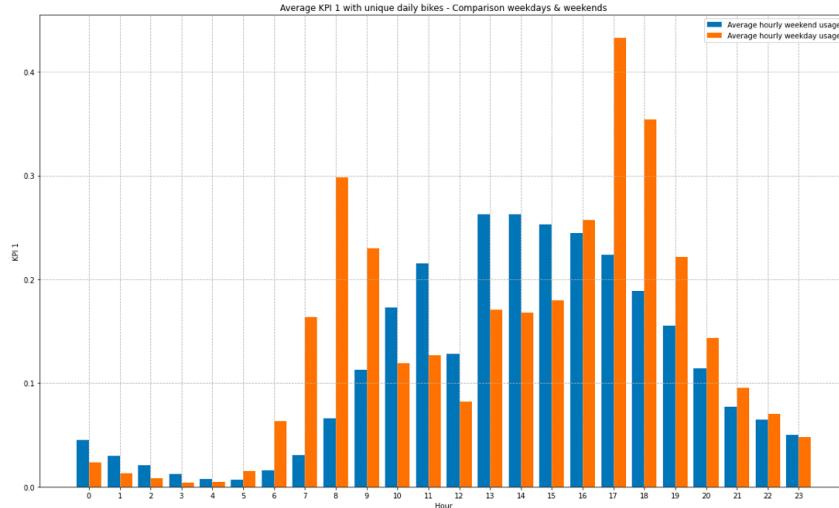


Figure 8: Average RPUB (KPI 1) with unique daily bikes - Comparison weekdays and weekends

3.4 KPI 2: Demand-Capacity (Implementation of the Concept of Coverage)

This KPI is related to "coverage" and investigates, whether the distribution and the number of stations is sufficient. Demand-Capacity divides the maximum of the number of arriving and departing rides per hour per station by the capacity in terms of how many bikes that station can hold. For further explanation, please refer to the Jupyter notebook. For Figure 9, it was counted how many rides were started at a station exhibiting high Demand-Capacity values (category 2) or mediocre/low (category 1 and 0) values.

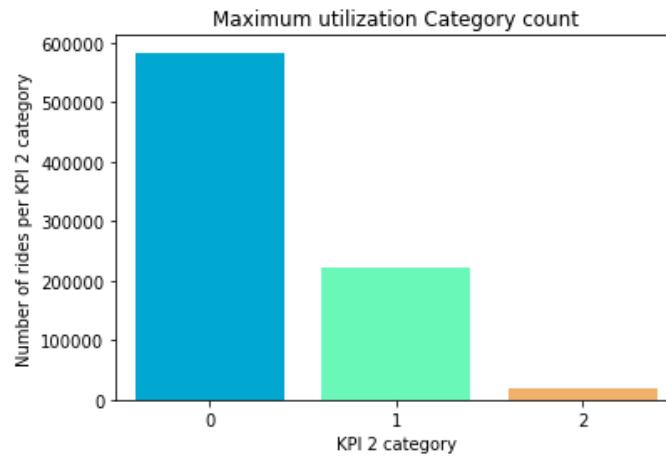


Figure 9: Maximum utilization category count

As can be clearly seen, the number of rides starting at a category 2 station is very low. A similar notion arises when investigating the average Demand-Capacity value per month (Appendix 34).

Moreover, when investigating the average hourly Demand-Capacity value (Appendix 35), whilst a commuting pattern can be observed similar to that of RPUB, even during the most industrious time of

17 o'clock, the KPI does not exceed an average value of 0.4.

Splitting the same hourly illustration into summer and winter values (Figure 10a), the value peaks at around 0.4. This notion does not change when distinguishing the hourly values in terms of weekday vs weekend (Figure 10b). Interestingly, the values peak at 5 pm at just below 0.35, whilst the maximum was around 0.4 during summer. Thus, the season has a higher influence on the Demand-Capacity than time alone.

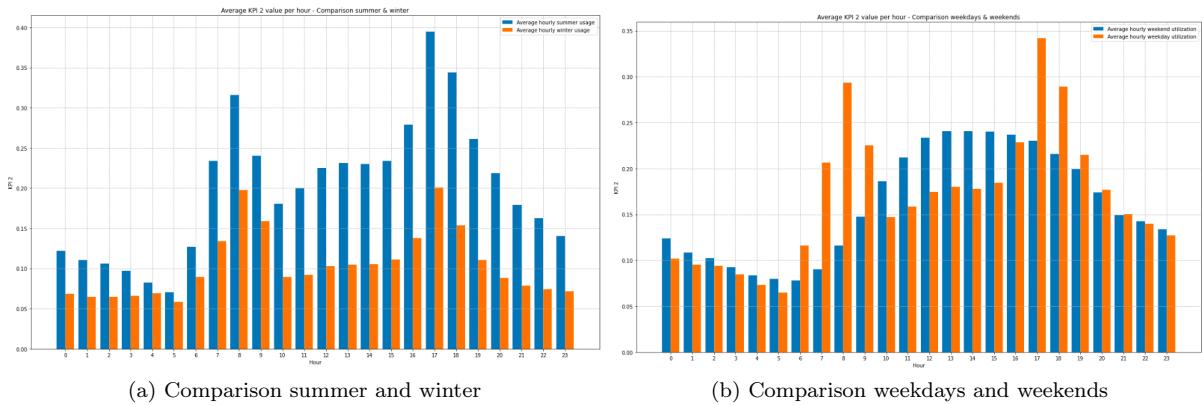


Figure 10: Average Demand-Capacity (KPI 2) value per hour

In order to investigate stations that exhibit problematic behavior in terms of the Demand-Capacity, we limited the data to the 20 most "problematic" stations, that is, the stations that exceeded a Demand-Capacity value of 1 most often. Investigating those 20 stations, it can be seen (Figure 11) that during non-summer months, the Demand-Capacity values are not problematic, as they rarely exceed a value of 0.4. However, during summer, the upper boundary of the standard-deviation intervals exceeds values of up to 1.4, which can be considered problematic.

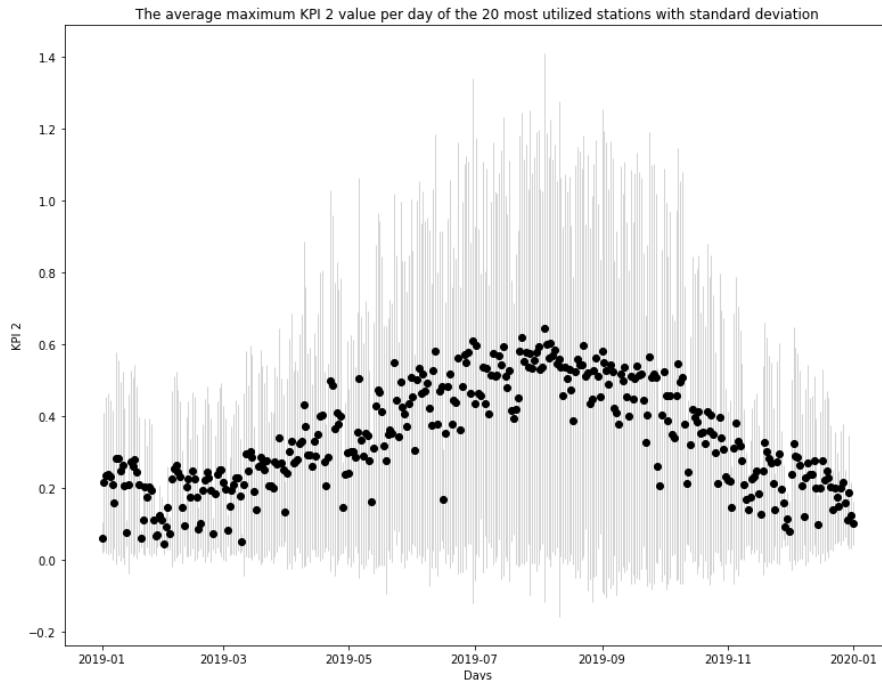


Figure 11: The average maximum Demand-Capacity (KPI 2) value per day of the 20 most utilized stations with standard deviation

3.5 KPI 3: Average Rental Durations

As a next KPI, the average hourly rental durations over different aggregated time slots were observed in order to get a detailed insight on different patterns. The main key aspects we drew from that analysis are described in the following. First, there is a common pattern that the longest duration rides occur between 10am and 5pm, so during the midday, the average duration is the highest. Between 12am and 4am, the average hourly duration is at its second-highest peak. Between 5 am and 9 am, the average duration is lowest during the day, meaning that those bike rides are more often used for short trips. This pattern is visualized in Figure 12, whereby the pattern did not only occur on an aggregation level over the whole year but also in summer, winter, weekends and weekdays. Next, different aggregation levels were analyzed, and the final results can be seen in Appendix 37. Again, a seasonality pattern can be observed, in which in summer, longer durations occur than in winter. Additionally, durations on weekends in summer are on average 5 minutes longer than on weekdays and in winter there are overall shorter durations during the entire day regardless of weekend or weekday. Furthermore, a weather pattern can be observed which tells us that during precipitation, the average durations are usually shorter while during warmer temperature, they are longer.

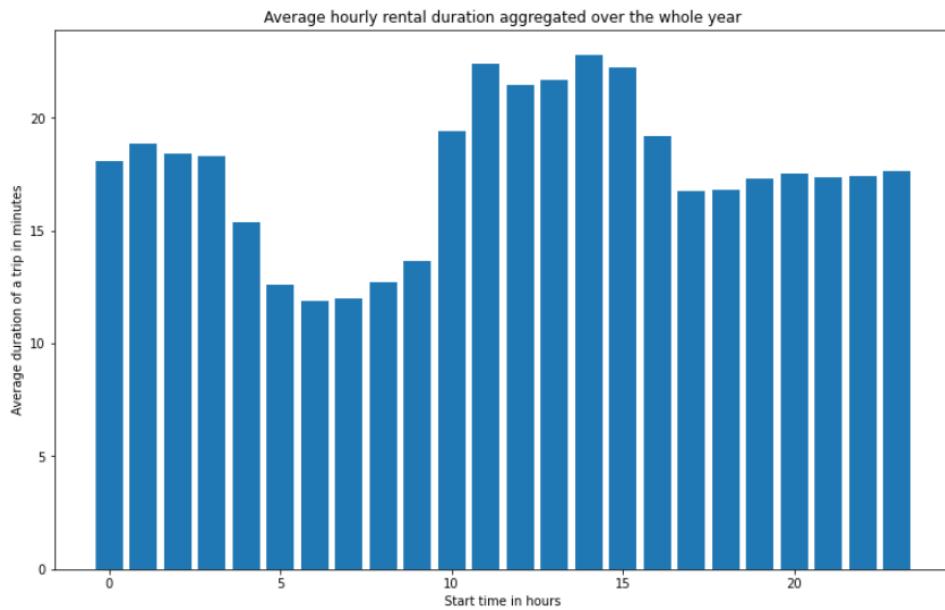


Figure 12: Average hourly rental duration aggregated over the whole year

3.6 KPI 4: Revenue

The revenue is the last KPI. It totals 4.656.803 \$ excluding subscriptions and including subscriptions 8.652.803 \$ (Figure 13). The profit from subscribers can only be estimated, because they pay an annual subscription and we do not have the number of subscribers nor the user IDs to calculate the numbers (Figure 13). The customers' share is directly depending on the number of trips from customers. Also, it is dependent on the duration of the trips for both types of users as trips longer than 30 (customer) /45 minutes (subscriber) are paid by minute (derived from *divvy website*). Thus, the patterns observed for the usage of the bike fleet and duration of trips apply to this KPI as well. The revenue from customers is volatile and peaks in summer while the revenue from the subscriptions is quite stable (Appendix 38). Also, the mean revenue per trip is higher in summer than in winter due to the longer duration of trips on average, leading back to the weather. The mean revenue per trip per day is highest on the weekend (Appendix 39).

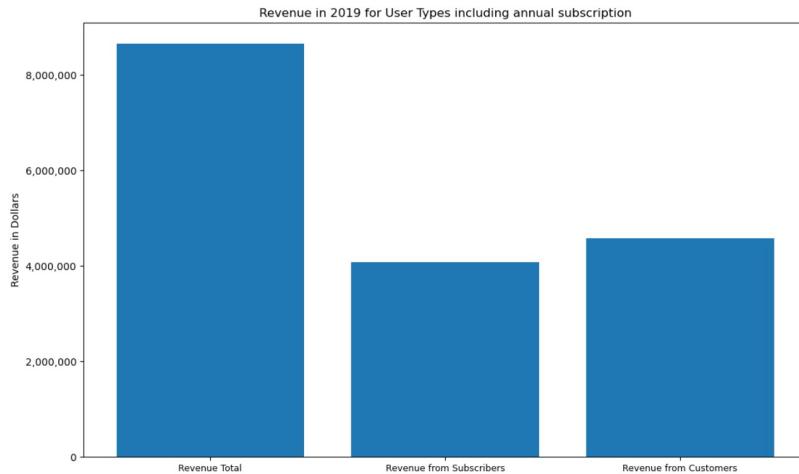


Figure 13: Revenue in 2019 for user types including annual subscription

4 Data Analytics

4.1 Cluster Analysis

Firstly, we will create clusters based on trip durations, weather (temperatures and precipitation) and revenue characteristics as described and calculated in the descriptive section. Thus, we develop station-related clusters and transaction-related clusters. Secondly, we will try to develop clusters of stations based on activity measures (KPI2, number of starting rides per station). During clustering, we would like to refer to the mean duration of each available hour to one ‘trip’. The application of K-Means as a hard clustering method and the Gaussian Mixture Model (GMM) as soft clustering method results in three reasonable clusters describing duration as a single input feature. The number of clusters is derived from the elbow point and silhouette scores, visualized in Appendix 40 and Appendix 41. The visualization of the clusters reveals short trips (<14 min.), medium trips (<24 min.) and long trips (>24 min.) (Figure 14). The Gaussian Mixture model result suggests, instead of a cluster for long trips, a cluster for extreme trips, which includes both very short and long trips (Appendix 42).

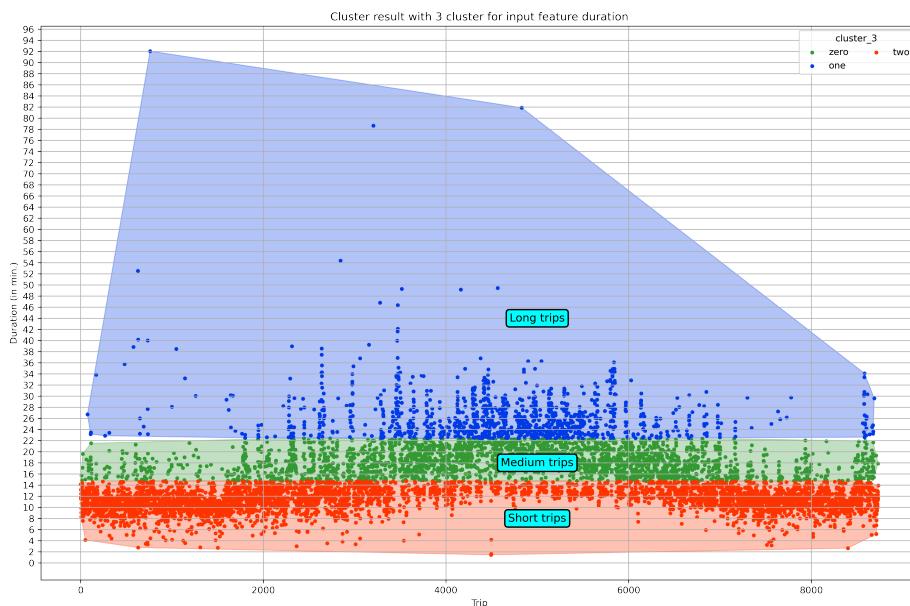


Figure 14: Cluster result with 3 cluster for input feature duration

The clustering with duration and temperature leads to six differentiations, visualized in Figure 15. Following the elbow method (Appendix 43), one would choose the cluster result with 3 clusters (Appendix 45), but this cluster assignment seems too superficial and does not represent the different combined intervals of duration and temperature values in an adequate manner. The GMM adds a new perspective, as the data is now largely represented by elliptical cluster structures. The clusters leave a large scope for interpretation and are highly subjective. Despite the better coverage of elliptical structures, a more reasonable classification as with K-Means cannot be found (Appendix 44 and Appendix 46).

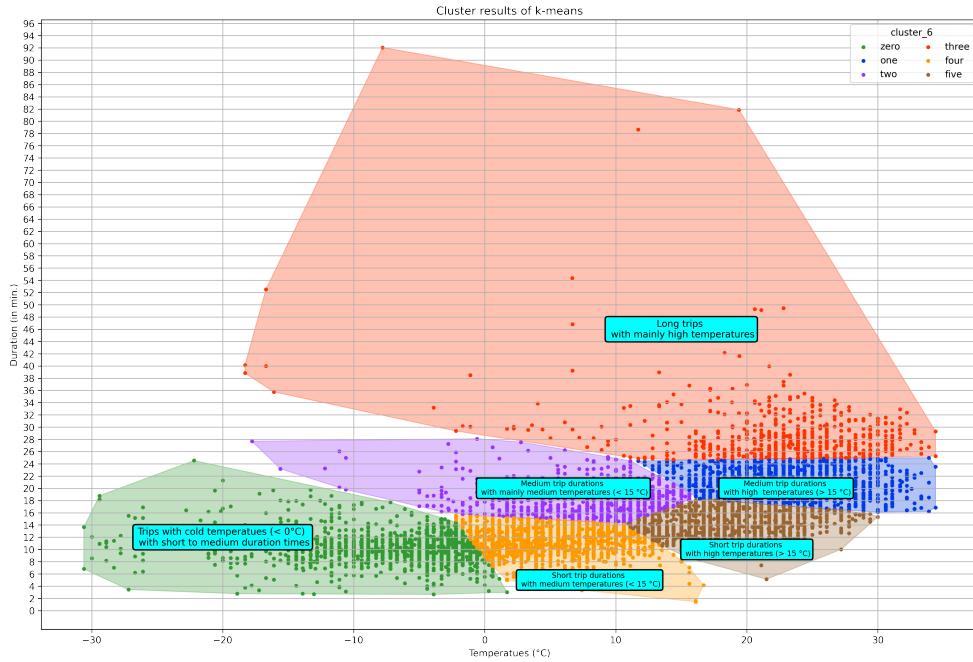


Figure 15: Cluster result K-Means

Taking precip into account, the complementary cluster to the K-Means result with 6 cluster is visualized in Figure 16. The K-Means algorithm found the same 6 clusters, but additionally differentiate rain trips by adding two rain clusters.

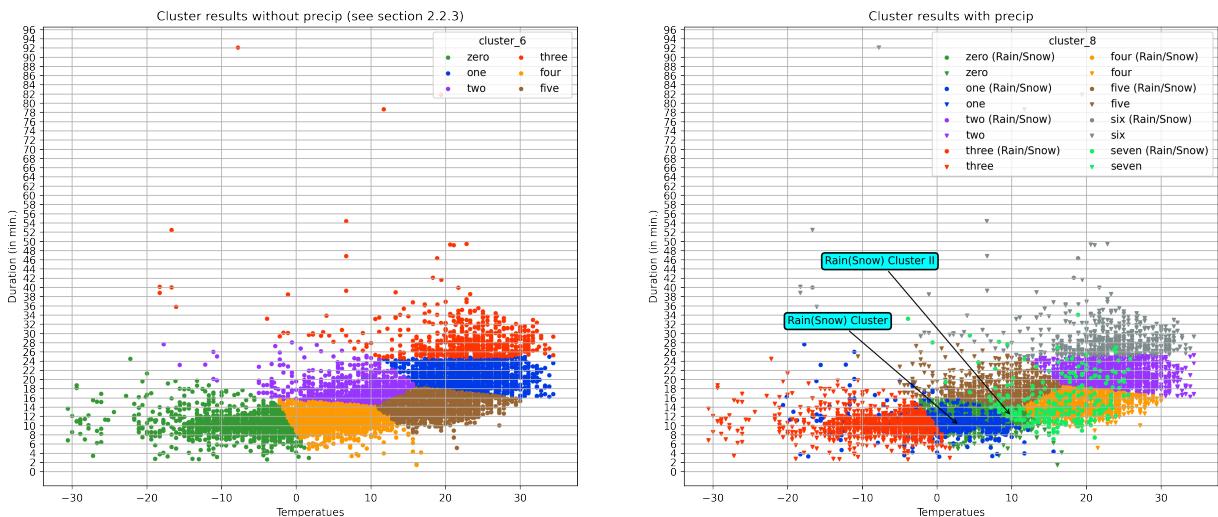


Figure 16: Cluster result K-Means with and without precip

Moving to the analysis of the revenue, we decided to cluster the revenue hourly (averaged) to yield a

reasonable amount of data with K-Means and GMM. The silhouette score for the latter one (Appendix 47) indicates that three clusters are optimal. The results of the K-Means are found in Appendix 48 and 49). We found the GMM clusters to be more reasonable, thus we continued to work with those (Figure 17).

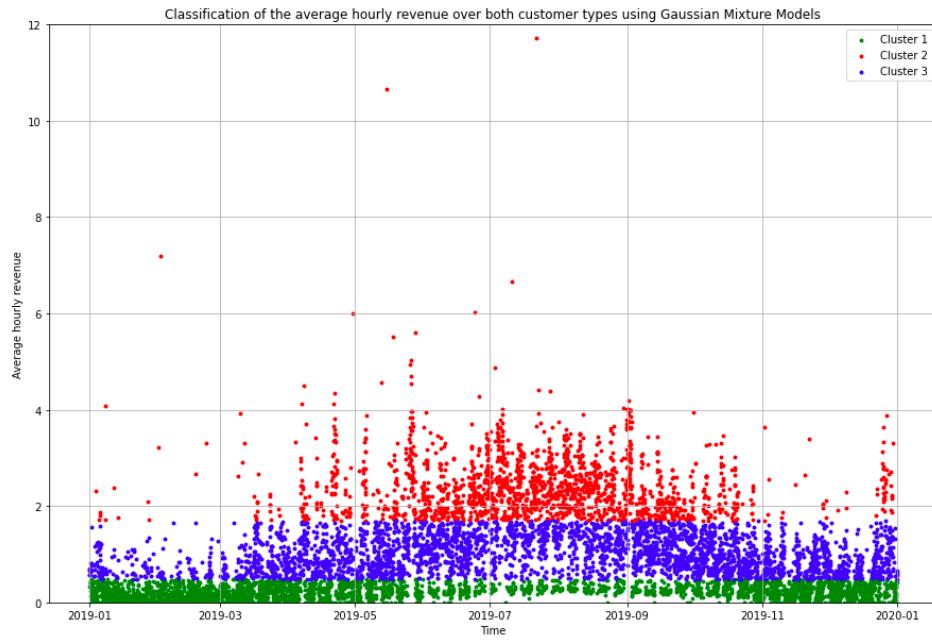


Figure 17: Classification of the average hourly revenue over both customer types using Gaussian Mixture Models

Looking at the monthly share of revenue per cluster (Figure 18), it is evident that during winter, autumn, and spring, usually the blue cluster accounts for most of the revenue, although in January and February, the share of the green cluster is nearly equally high. During summer however, especially in July and August, the red cluster accounts for the absolute majority of the revenue, indicating, as also seen in the above sections, the importance of long rides during the warm summer months for the ride sharing system. Adding the average hourly temperature as an input feature does not yield relevant insights, which is described in (Appendix 51 - 53).

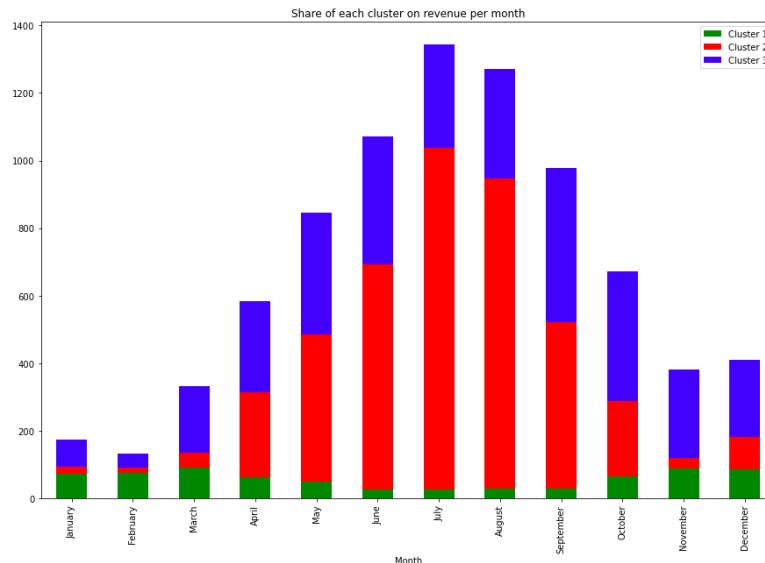


Figure 18: Share of each cluster on revenue per month

Now, the individual stations were clustered, beginning with the Demand-Capacity (see the descriptive part for the explanation) as the sole input feature. Applying K-Means and again using the elbow method (Appendix 50) yields a relatively high number of optimal clusters, however we discovered a clear three-ring structure in the descriptive analysis and thus chose three clusters.

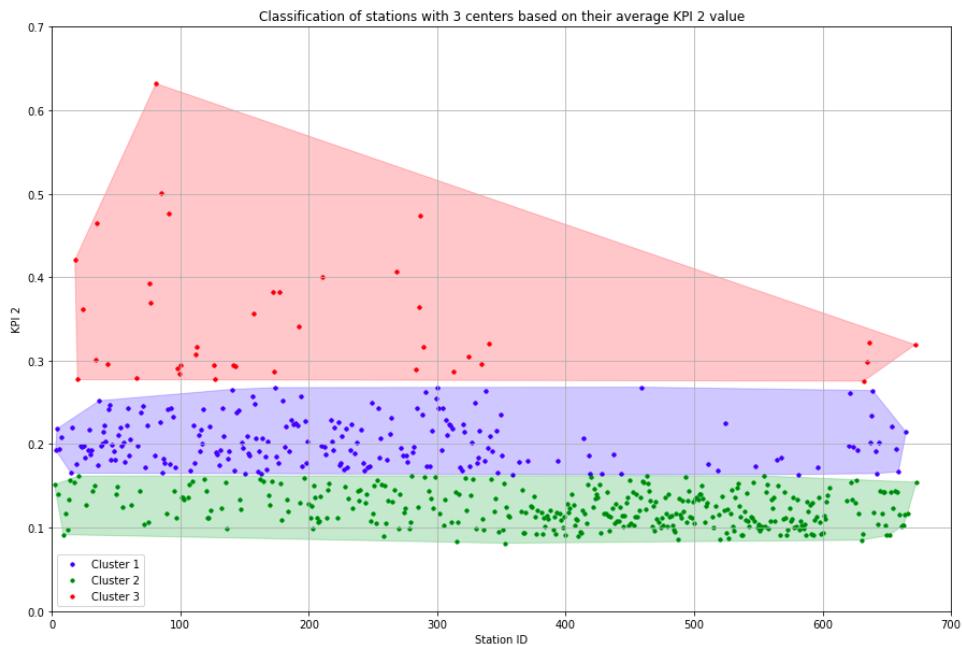


Figure 19: Classification of stations with 3 centers based on their average KPI2 value

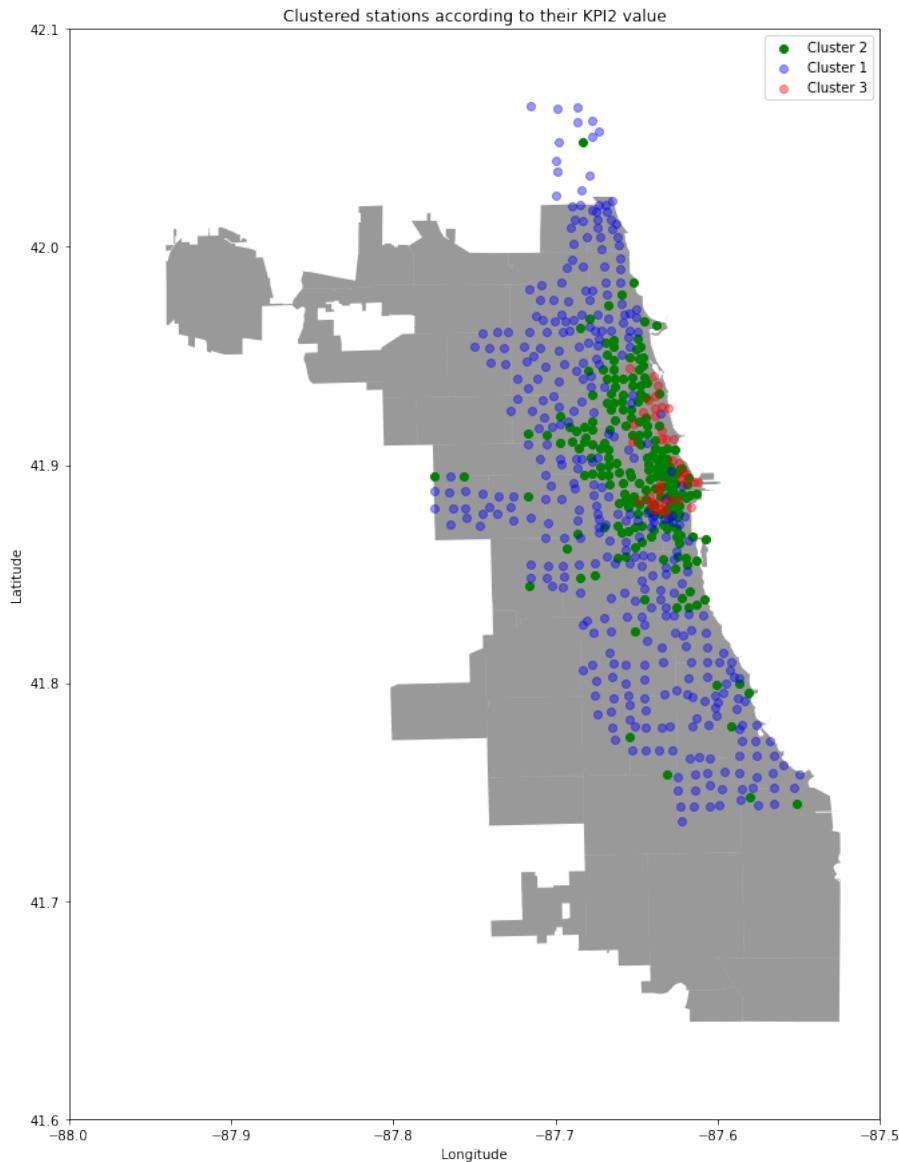


Figure 20: Clustered stations according to their Demand-Capacity (KPI2) value

Figure 19 illustrates the clusters and serves only as a supporting graph to understand the mapped values in Figure 20. Here, we can clearly observe the above-mentioned three-ring-structure. Comparing the sketch map in the notebook with a Google Maps representation of Chicago, it gets clear that the majority of the red stations are located near recreational areas close to lake Michigan. A second smaller cluster of red stations seems to be located in downtown/roughly north to the University of Illinois/Chicago. We deduce the following hypothesis: The red stations comprise mainly leisure stations, that are used for recreational purposes. This is backed by Figure 21, where the “red” Demand-Capacity development is largely season dependent - this makes sense, as during summer, there are obviously more rides related to recreational purposes. The blue and green cluster cannot be significantly distinguished. Several other input features were applied in Appendix 54 - 58 without new insights.

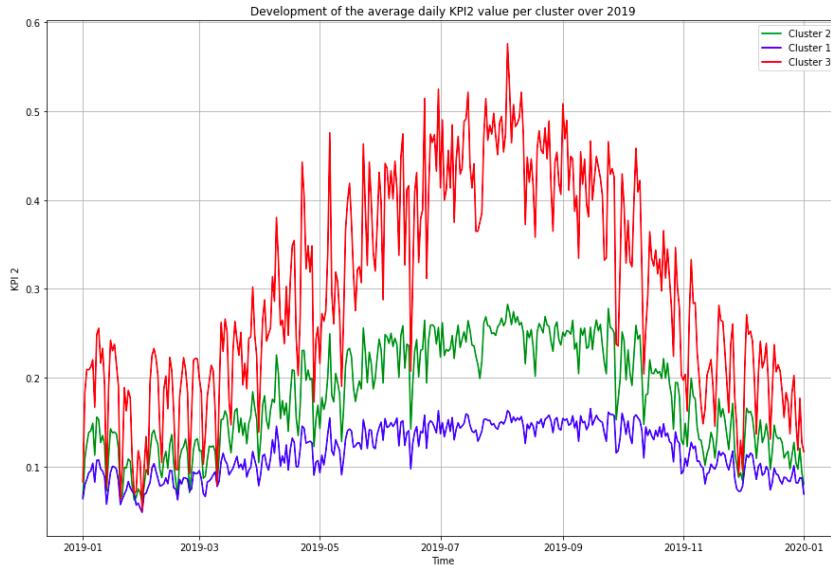


Figure 21: Development of the average daily Demand-Capacity (KPI2) value per cluster over 2019

A last insightful clustering using K-Means was created based on the total number of starting rides per station and the average duration of the rides starting there. Four clusters were found to be suitable (Appendix 59) and plotting them (Figure 22) shows, that the red cluster stands out as it exhibits the least proximity to the group of the other three clusters. The yellow and green cluster are relatively similar in terms of their total number of starting rides but seem to show significant differences in terms of the average ride duration starting at the respective stations. When mapping the stations accordingly (Figure 23), Table 2 summarizes the ring structure.

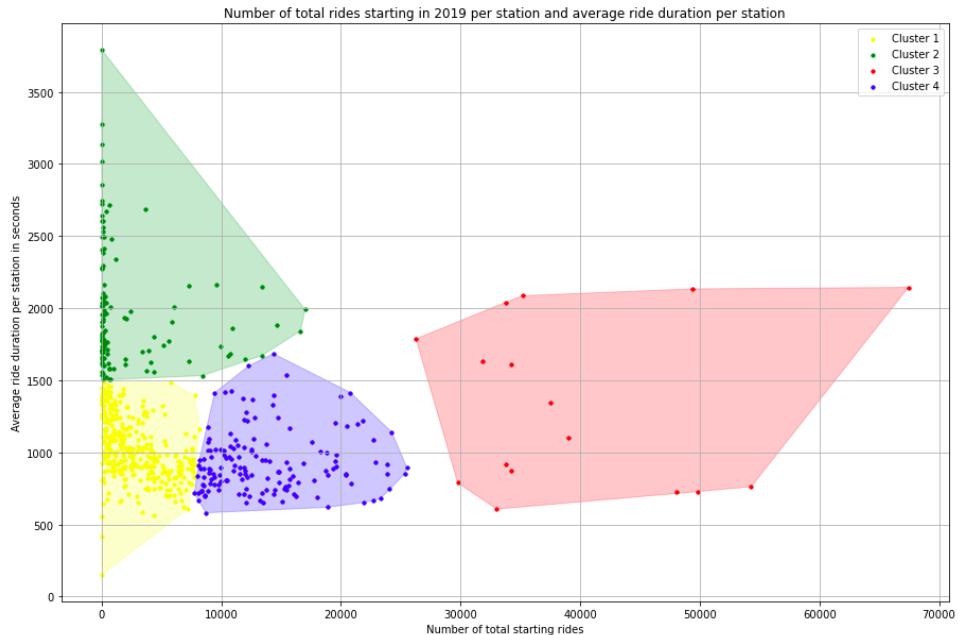


Figure 22: Number of total rides starting in 2019 per station and average ride duration per station

Ring	Description
Red Cluster	<ul style="list-style-type: none"> The red stations are not as predominant as in earlier plots. They are again mainly located around recreational sites.
Yellow and Green	<ul style="list-style-type: none"> The ring comprised by the green and yellow stations exhibits a new structure. The green stations seem to form a circle around the yellow stations and thus, around the city. This makes sense, as the green stations exhibit roughly the same number of rides as the yellow stations, however, they showed a higher average riding duration. Thus, the rides from the green outskirt ring seem to be heading towards the center of the city, as their average ride duration is the highest compared to all other clusters. The other clusters do not exhibit this behavior. They might be considered "housing" areas.
Blue Cluster	<ul style="list-style-type: none"> The second ring was also exhibited earlier. It corresponds to rides roughly as long as the yellow ones. The rides starting there will thus be heading to stations within the blue, red or close yellow stations.

Table 2: Cluster rings

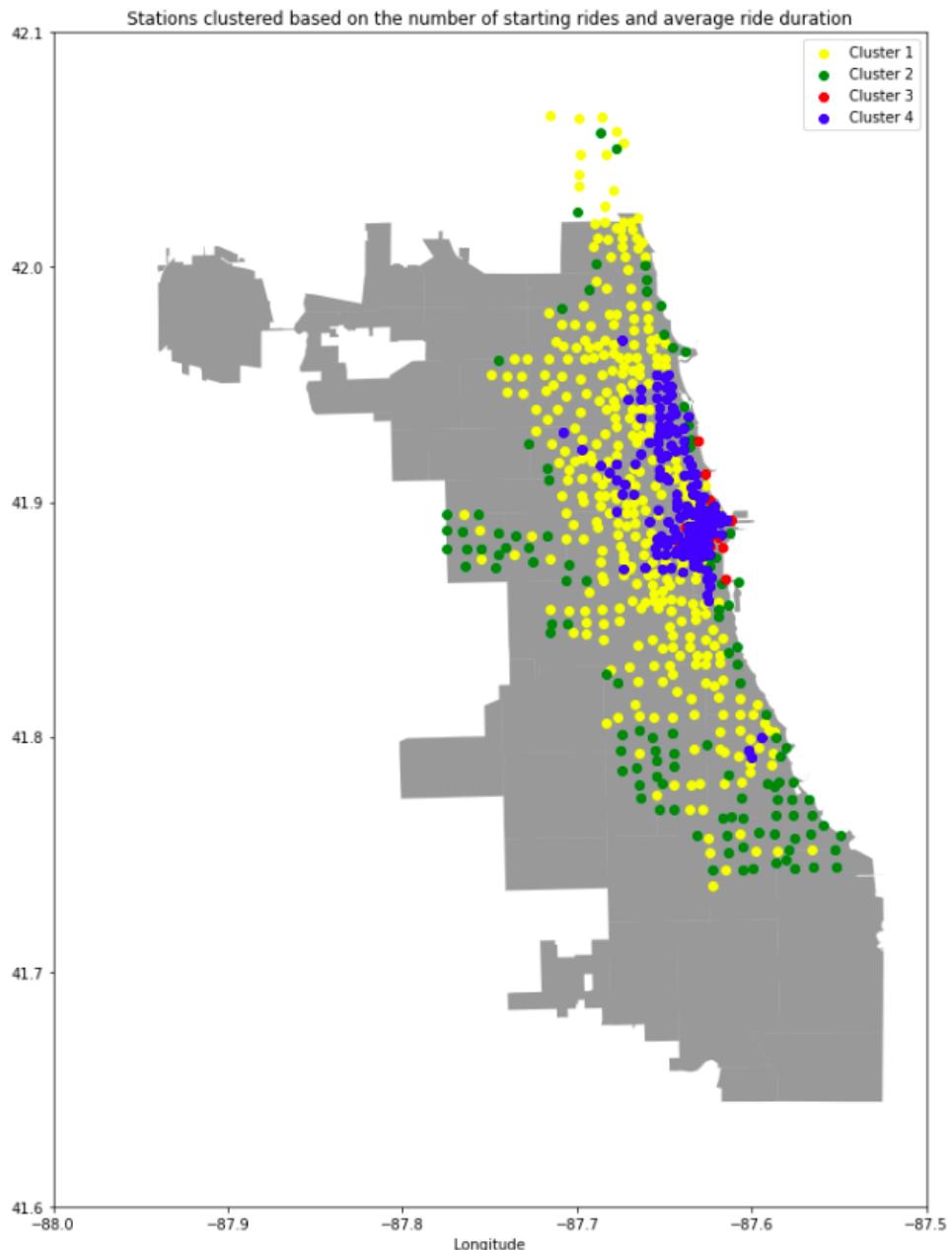


Figure 23: Stations clustered based on the number of starting rides and average ride duration

4.2 Predictive Analysis

As the future demand is a key factor that will guide operational decision-making of Divvy Bikes, total system-level demand in the next hour was forecasted. In order to do that, we developed prediction models using three different algorithms and features (Appendix 60) available in and derived from the dataset. The results can be seen in Figure 24 below. As the Random Forest algorithm produced the best model for us, we focus more deeply on it in this report.

Model	R-squared Train	R-squared Test	MAE	MSE	RMSE	MAPE
0 Lasso (L1) Regression	0.907344	0.870992	110.891394	33496.118119	183.019447	1.118139
1 Random Forest	0.988062	0.916826	80.205573	21595.745800	146.954911	0.502030
2 XGBoost	0.973565	0.917122	81.938185	21518.907884	146.693244	0.652096

Figure 24: Predictive analytics results

But first, we start with linear regression. We decided to use linear regression models due to their simplicity in application and understanding. While doing so, we immediately faced the problem of missing linearity in our data. Nevertheless, we used Multi-Linear Regression as a first look into regression. Our best model excludes the features ‘season’ and ‘isWeekday’ and reached an R^2 of roughly 0.6. Because of the missing linearity, we decided to also try Polynomial Regression with the same features, which resulted in strongly improved results. In addition, we also applied both L1 and L2 regularization. In the end, we yield our best Polynomial Regression model with L1 regularization, with an R^2 of about 0.8 to 0.9 for the prediction and an MAE around 100.

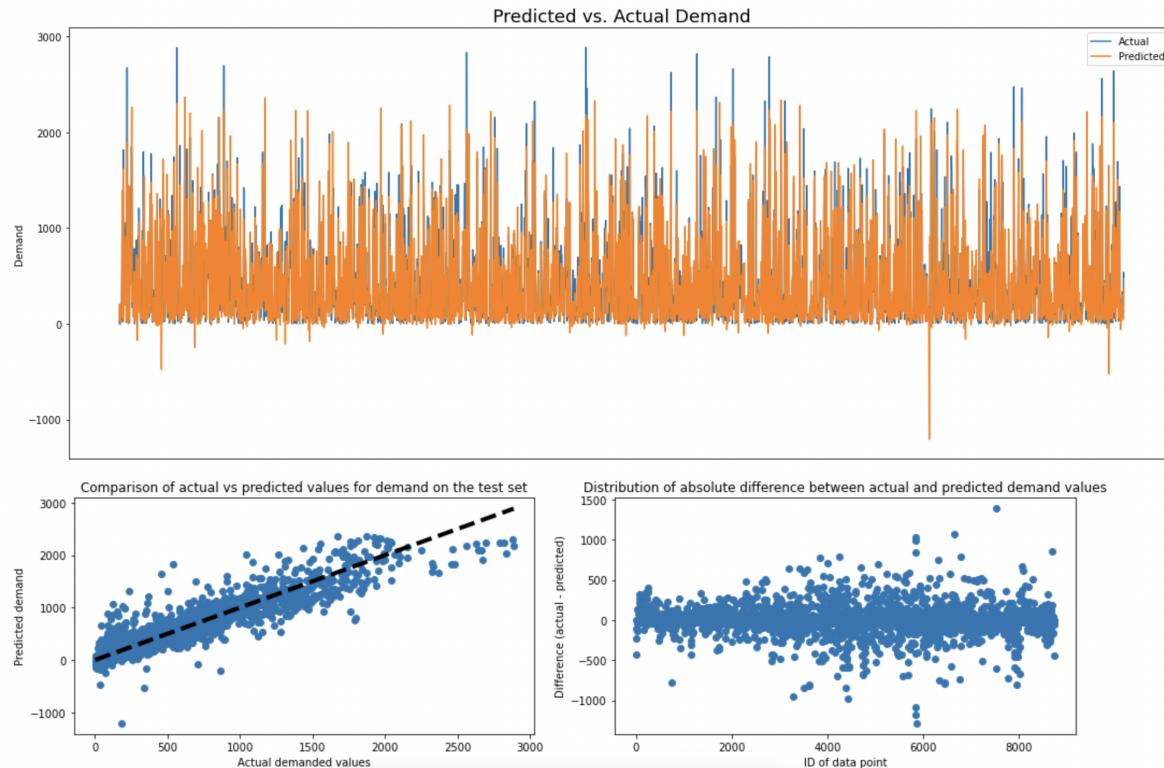


Figure 25: Best L1 model - Comparison of actual vs predicted values for demand on the test set and distribution of absolute difference between actual and predicted demand values

For our second and best model, we used the Random Forest algorithm. It is an ensemble method which uses multiple decision trees to make predictions based on a mean computation (in regression) of all trees (see Appendix 61 for a visualization). The exact algorithm description and reasons why we chose it as one of our model can be found in the Appendix. To find the best model possible, first a feature selection process was conducted in which different features were selected, and their impact was assessed. Based on their impact on the R^2 and MAE and their increase in complexity, the features were then either removed or included in the model. The MAE error metric was used because it has practical relevance, stating that on each hourly prediction x bikes were predicted too much or too less. The R^2 error metric describes the proportion of variance in the dependent variable that is accounted for by the model and thus, is also a good measurement of overfitting. The table 5 in appendix illustrates the process. At the end, we decided to select model 5, as model 4 has three more features and is more complex, and the metrics did not differ significantly. Afterwards, the hyperparameters were tuned and selected. For that, we first looked at the number of estimators (trees in the forest) with cross validation, as can be seen in Figure 26. At around 250 estimators, the score does not really change at all, so we picked the number of trees in the forest as 250. To facilitate the hyperparameter tuning and not only focus on one single

parameter but on different combinations, we implemented a grid search. Because a whole grid search is computationally heavy, first a random grid search was performed, which limits the possible search space for each optimal parameter. The final hyperparameter results are listed in the Appendix Table 4 . With these values for the hyperparameters, we executed the model on the test set and retrieved an R^2 of 91,68 percent.

This means that 91,68 of the proportion of the variance in the demand can be accounted for by the model. The mean absolute error (MAE) is 80,21 which means that in each prediction of the bike demand in one hour, there are 80 too many or too less bikes predicted on average. Benchmarking this, the absolute deviation in the model still seems relatively high, and the Divvy bike provider would still need to maintain 80 additional bikes which are not needed. However, there is a total of more than 6.000 bikes in the Divvy bike fleet, so the prediction error accounts only for about 1 of all bikes which is handleable for the provider. The results of the model are visualized in Figure 27. The dashed line would be the optimal prediction line, e.g. if the actual demand is 1.000 the predicted demand would ideally be 1.000 as well. It can be seen that the majority of prediction points reside near the optimal line. In the second figure, the deviations can be inspected more precise. Most of the prediction points lie between -100 and 100 which is as also expressed through the mean absolute error (MAE) of 80. However, some outliers to exist with the extreme of approximately -1.500 demanded bikes on one day.

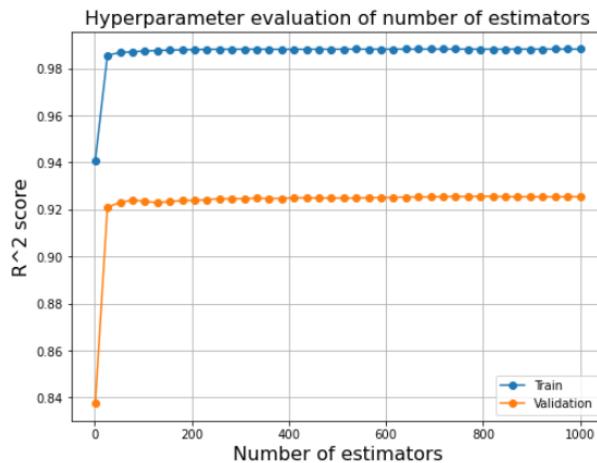


Figure 26: Hyperparameter evaluation of number of estimators

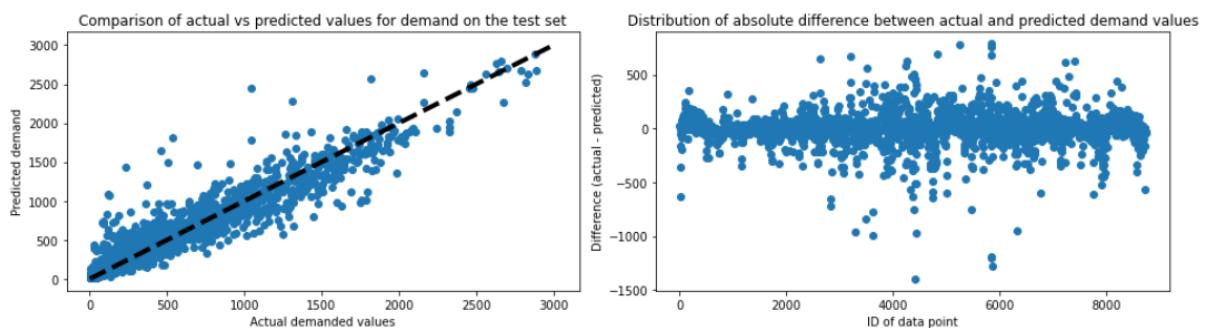


Figure 27: Random Forest - Comparison of actual vs predicted values for demand on the test set and distribution of absolute difference between actual and predicted demand values

As third regression algorithm we decided to use XGBoost, an ensemble method. The "state-of-the-art" machine learning algorithm (Shyam, Ayachit, Patil and Singh, 2020) can deal well with structured data and is fast, especially compared to other implementations of gradient boosting. The results of the

first model are good with an R^2 of 97 for the train and 92 for validation. Also, the mean absolute error (MAE) and the Root Mean Squared Error (RMSE) indicate that the model is relatively accurate. Using grid search to tune the hyperparameters improved the model, but only slightly, resulting in an R^2 of 91,7 for the test set (Appendix 63).

5 Conclusion

In conclusion, the project delivers valuable insights into the bike fleet's performance and hourly demand prediction.

Averaging over all yearly bikes, the fleet does not meet the target number of rides of 4-6 defined by Boor (2019), so there might be too many bikes or improperly distributed bikes, especially during winter. The vast majority of stations are properly equipped in terms of capacity. "Outskirt" capacity might even be too high, thus it could be considered to relocate some stations towards the center.

A three ring structure could be exhibited, with the red, most centered ring being that with the highest demand and activity. The red stations are predominantly in recreational sites. Combining this insight with the fact that the duration of rides in summer is significantly longer, we can conclude that leisure rides during summer and commuters seem to be the most important customer groups. Those groups could be advertised specifically.

In terms of revenue, there are different possibilities to change the pricing in order to increase the profitability. First, looking at the "three-ring-structure", different pricing for the rings could be adapted. Furthermore, the bike rentals in winter should be made more attractive and thus, the pricing in winter could be adjusted. Another idea is to introduce a different pricing structure for the extra minutes. During the day, the longest duration is between 10am and 5pm and the second longest is during 12pm and 4 am, hence the trips could be more expensive per minute in these time intervals.

However, there are some limitations to our project. The demand is predicted for the whole system, but predicting the demand on single stations is more accurate and enables better maintenance and refilling of bikes. Moreover, time series analysis over multiple years could be used to improve prediction. Also, we were limited by the given data and collecting more detailed data offers new options for further analysis.

A Appendix

A.1 Data Description

Data inputs	Description
start_time	Time where trips are started : datetime
end_time	Time where trips are finished : datetime
start_station_id	ID for start station : integer
start_station_name	Name for start station : string
end_station_name	Name for end station : string
bike_id	ID for unique bike : integer
user_type	Subscriber and customer are distinguished : string
duration_time	Duration of trip (in sec., min., hour) : integer
weekday_start	Indicating weekday (true/false) : boolean
full_hour_start	Start time rounded to full hour : datetime
max_temp	Describing temperature values in Celsius : float
precip	Indicating rain/snow (true/false) : boolean
total_docks	Theoretical maximum capacity of the station : integer
docks_in_service	Actual capacity of the station : integer
status	Availability of the stations : string
start_latitude	Latitude of the starting station : float
start_longitude	Longitude of the start station : float

Table 3: Data description of most relevant data inputs

A.2 Temporal Demand Patterns and Seasonality

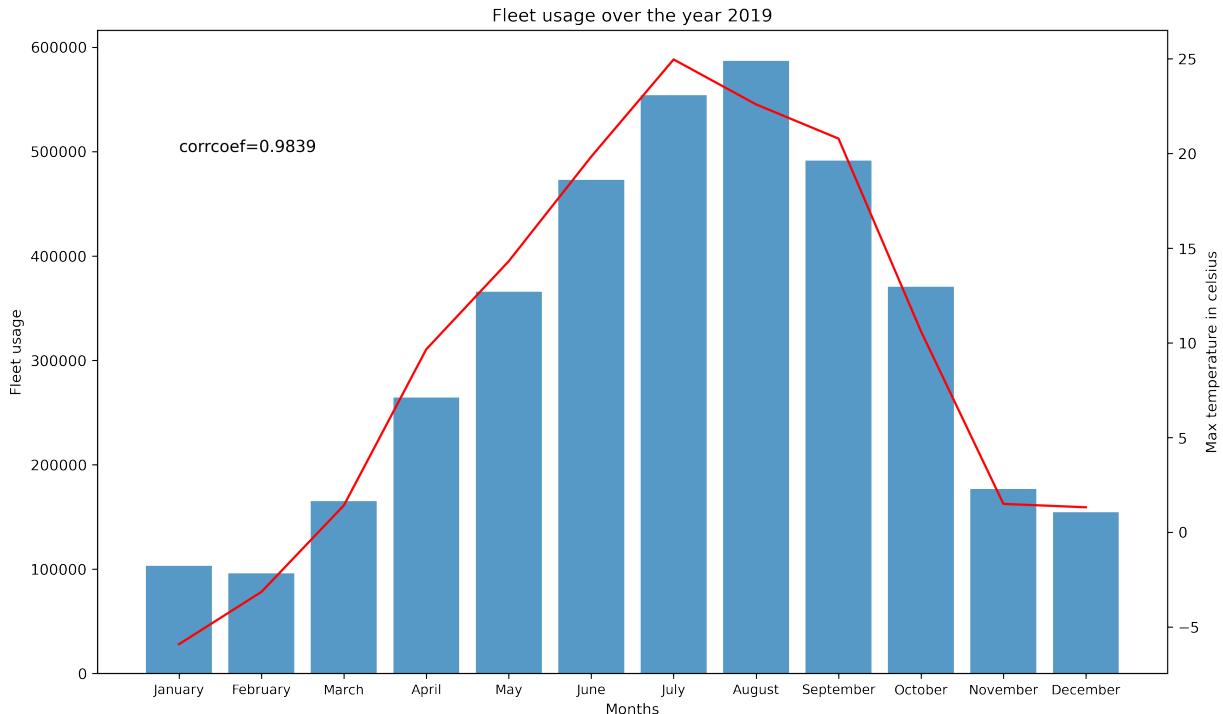


Figure 28: Fleet usage over the year 2019

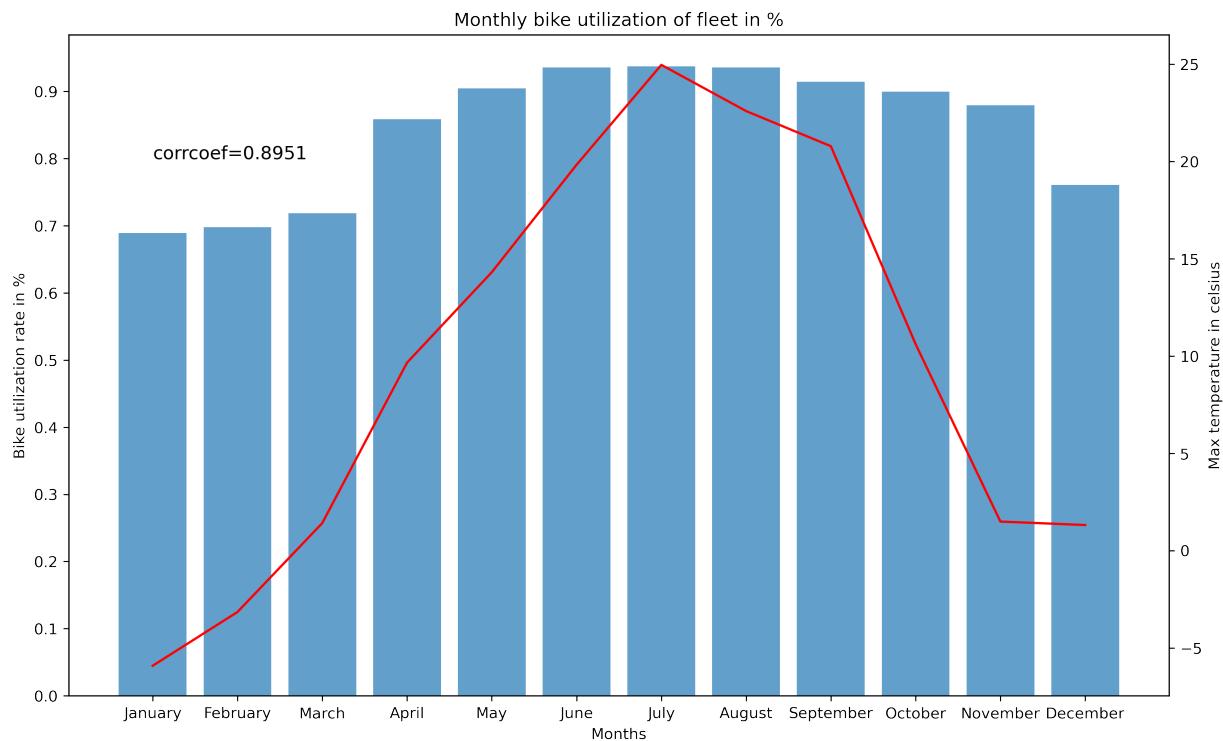


Figure 29: Bike utilization over the year 2019

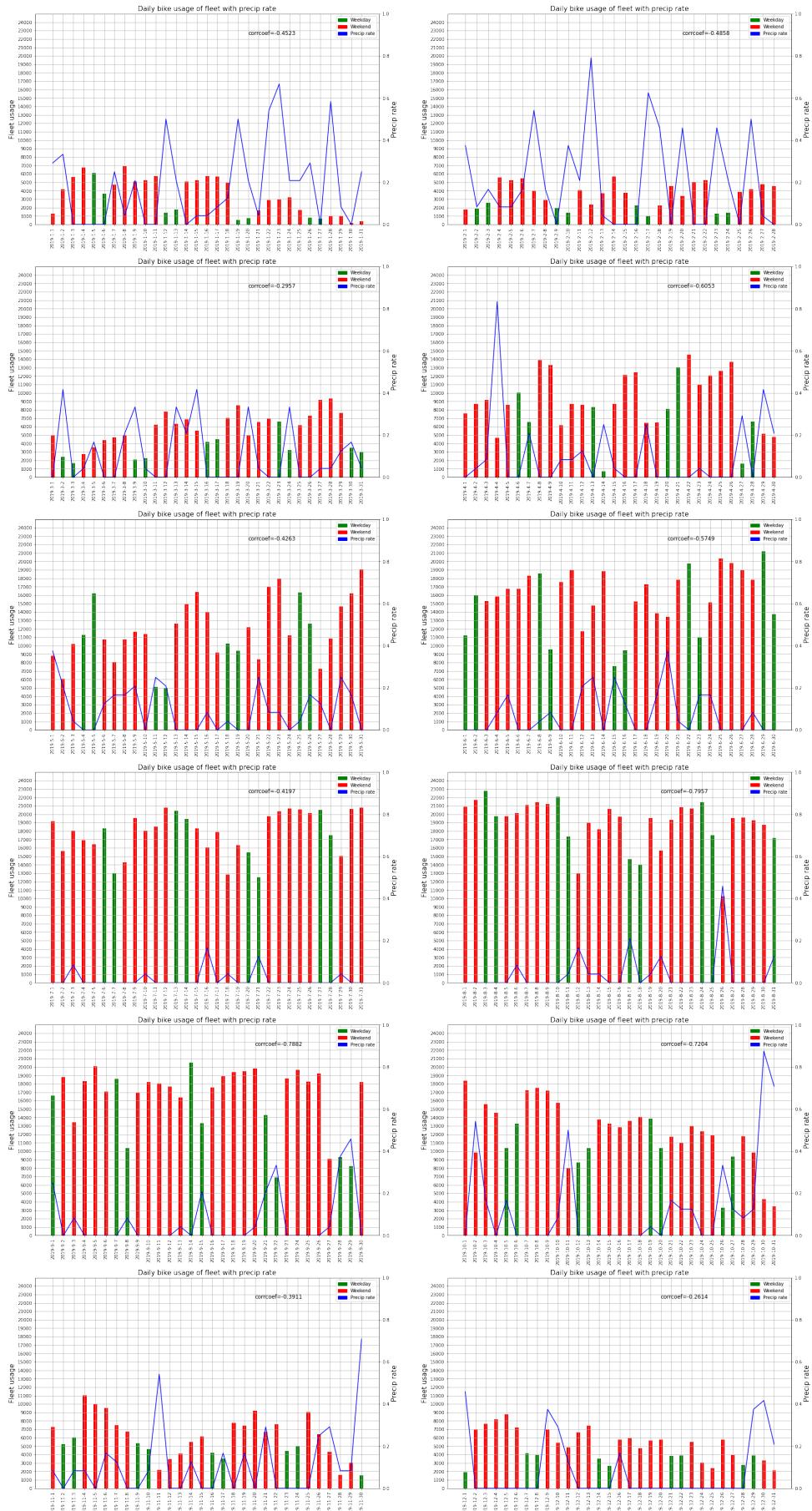


Figure 30: Daily bike usage of fleet with precip for each month



Figure 31: Daily bike utilization of fleet with precip for each month

A.3 KPI 1: RPUB (Rides Per Unique Bike)

Weather-based investigation of RPUB

To be able to also take into account some of the weather data, two metrics were developed. Firstly, if more than half of all rides on one day started during rain, then the day was counted as "rainday". Secondly, if the average temperature was above ten degrees, the day was counted as "warmday". In the first boxplot, the data (RPUB with unique yearly bikes) is split in terms of weather the day was rainy and weather it was warm. It becomes clear that the variance of the RPUB on a rainday is significantly smaller than on a non-rainy day. This might be due to the fact that regardless of the temperature or other aspects that might influence usage, people will refrain from using the bike, leading to less fluctuation in terms of the RPUB. As to expect, the RPUB is much higher on warm days.

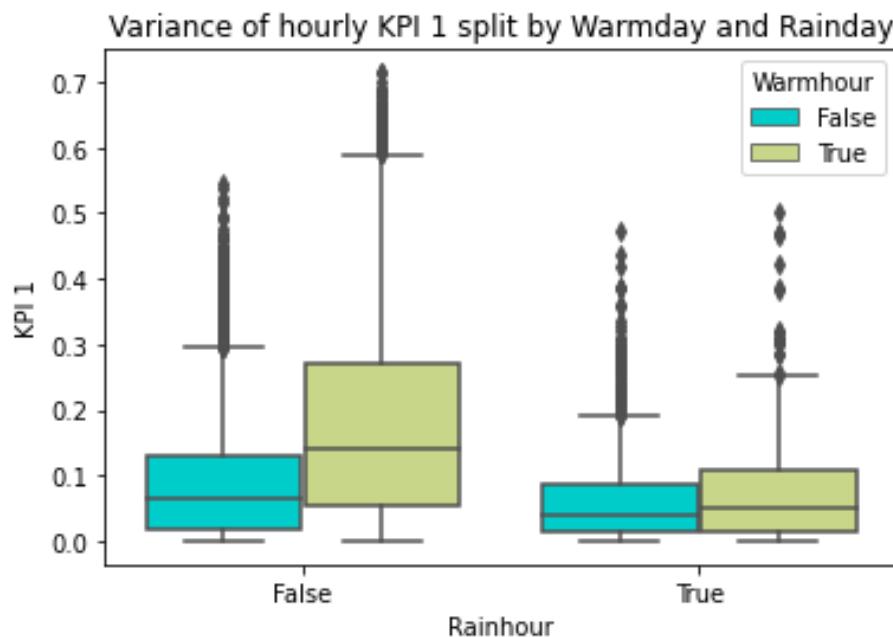


Figure 32: Variance of hourly RPUB split by Warmday and Rainday

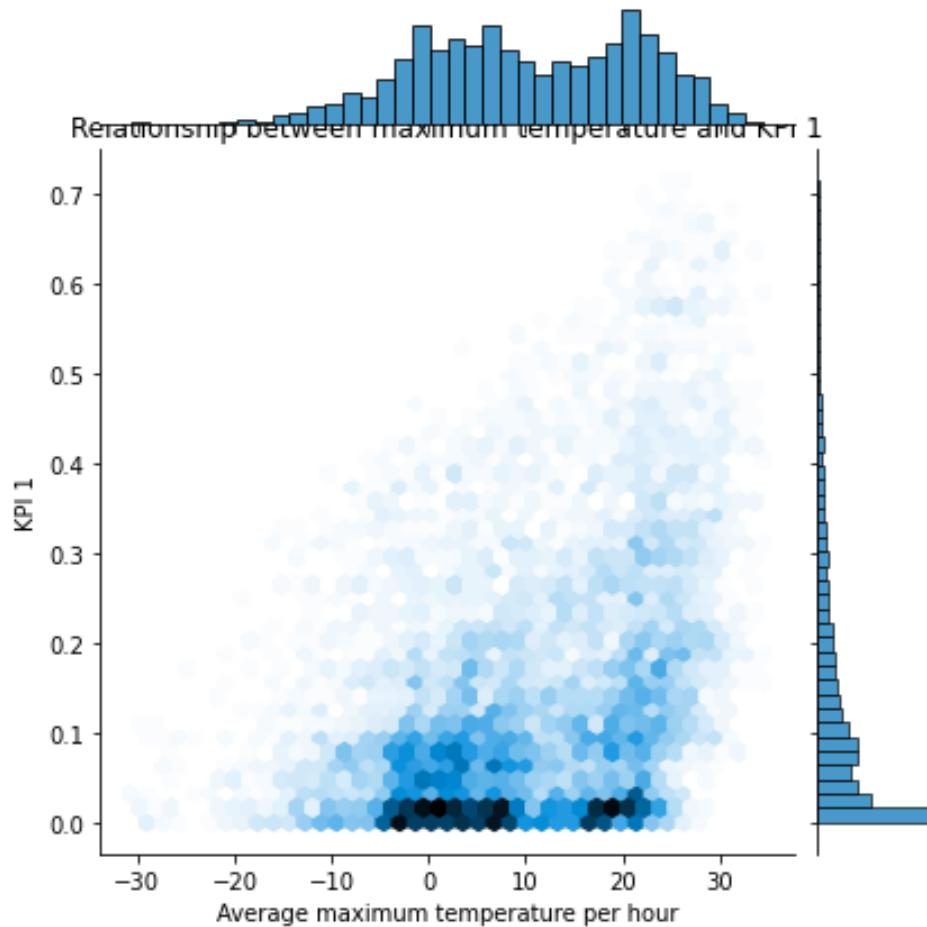


Figure 33: Relationship between maximum temperature and RPUB

Investigating a jointplot illustrating the relationship between maximum temperature and RPUB with unique yearly bikes, a linear positive correlation between the average maximum daily temperature and the RPUB (thus, bike usage) can be exhibited. However, the histogram of RPUB values is comparably equally distributed, being slightly skewed towards low values.

A.4 KPI 2: Demand-Capacity

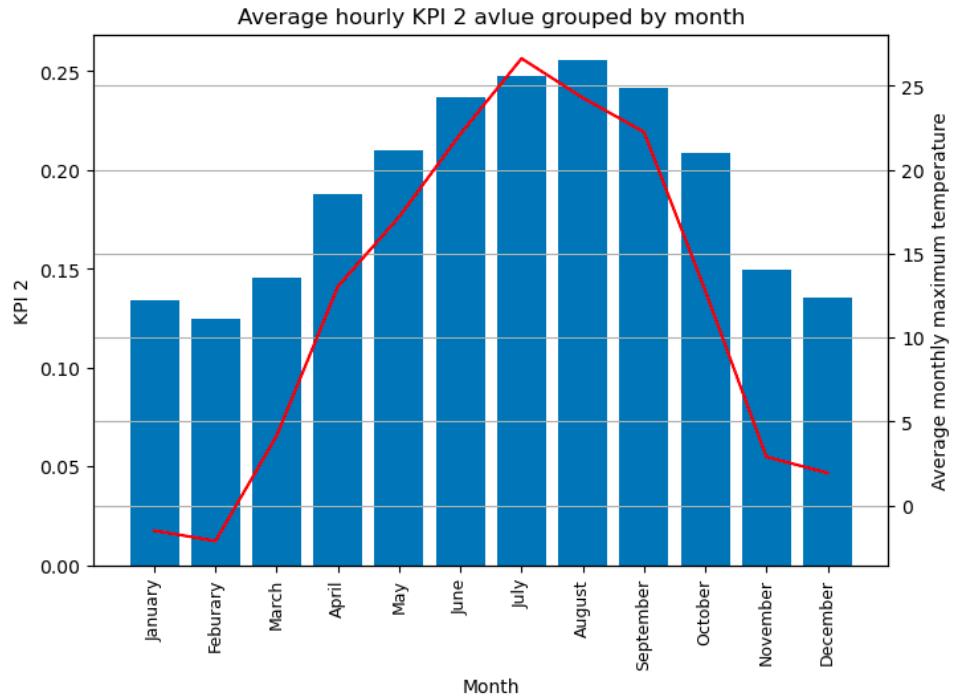


Figure 34: Average hourly KPI 2 value grouped by month

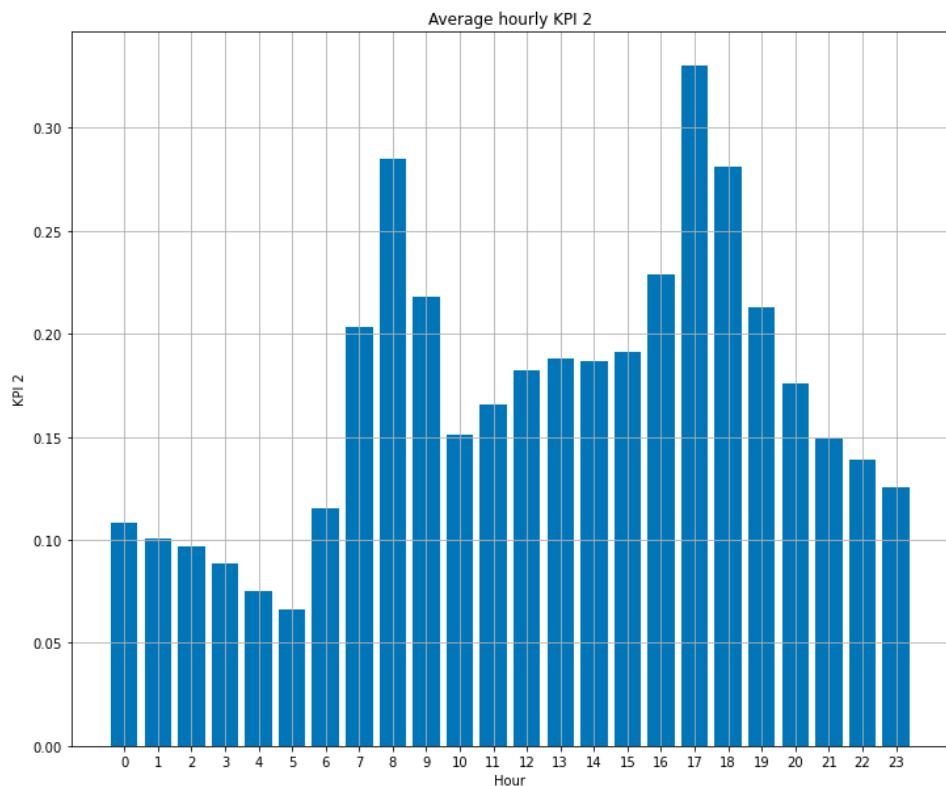


Figure 35: Average hourly KPI 2

More thorough investigation of Demand-Capacity

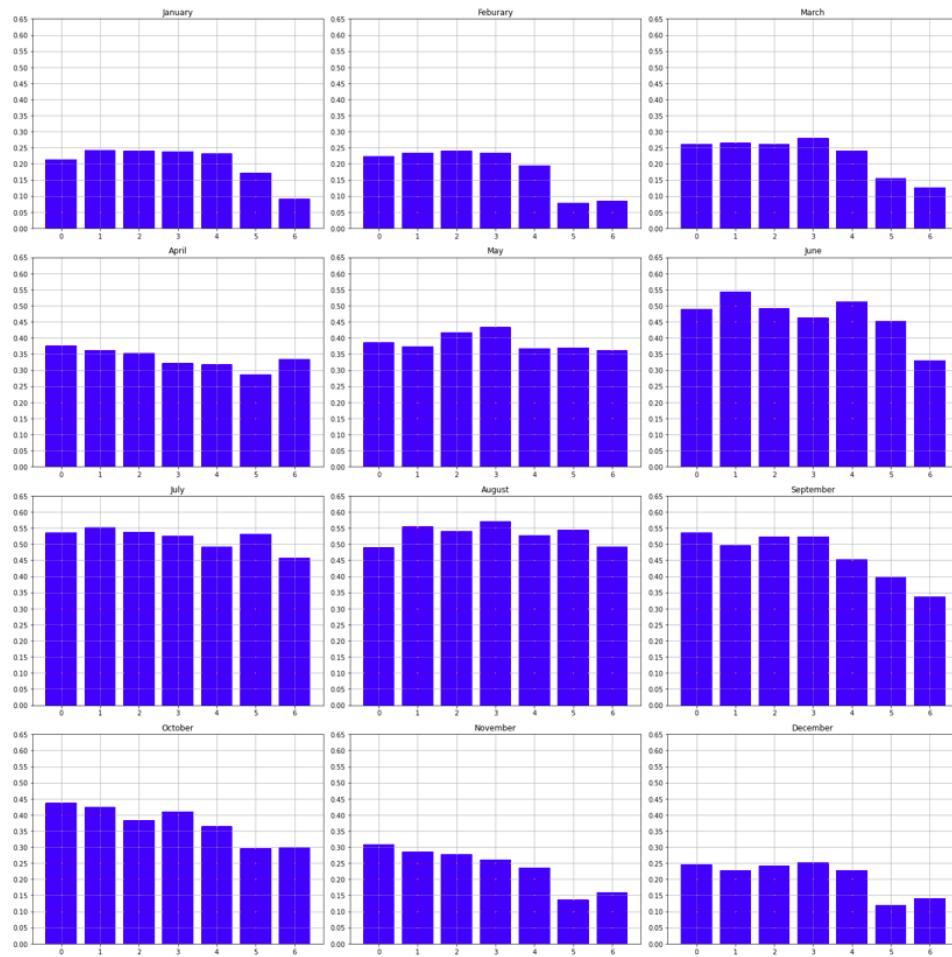


Figure 36: Average Demand-Capacity per day per month

The illustration above depicts the average demand-capacity value per day per month. This stresses the notion that was already gained in the main part of this assignment in a sense, that the figures do not, on average, come close to a problematic value of 1.

A.5 KPI 3: Average Rental Durations

	Time Scale	KPI: Average duration
0	Whole Year	17.574653
1	Summer Weekends	24.249130
2	Summer Weekdays	19.667574
3	Winter Weekends	11.477111
4	Winter Weekdays	11.733988
5	Precip	13.723743
6	Temp above 15°C	19.674434

Figure 37: Results KPI 3 average rental durations

A.6 KPI 4: Revenue

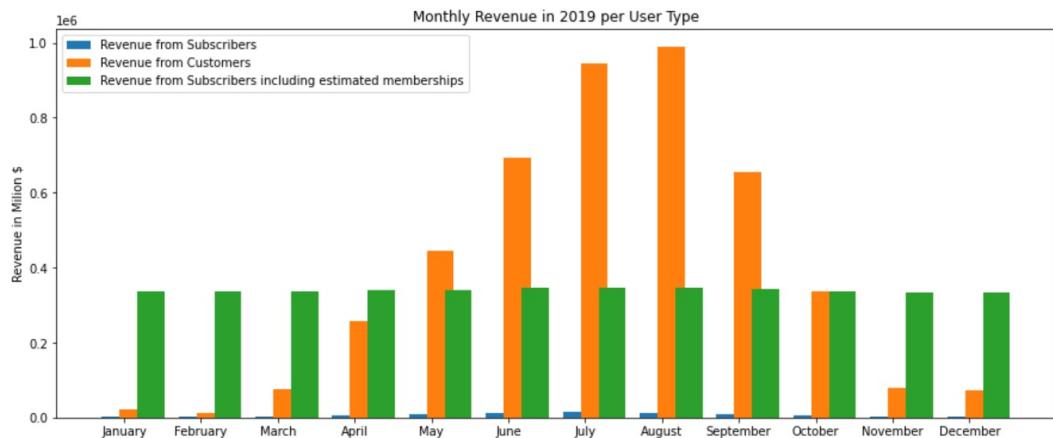


Figure 38: Monthly revenue in 2019 per user type

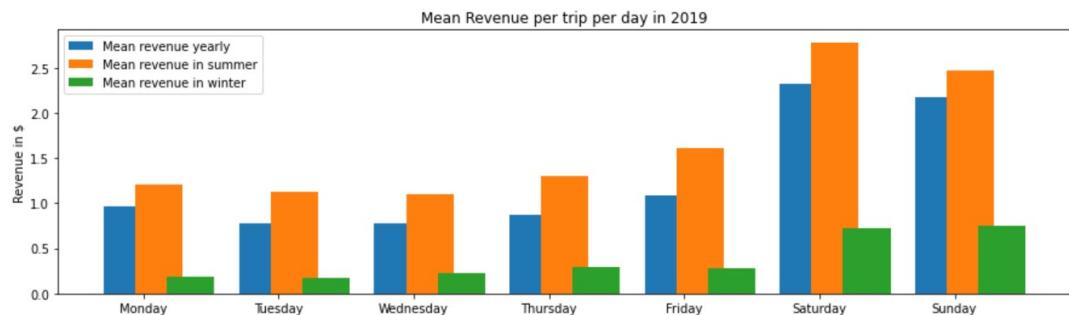


Figure 39: Mean revenue per trip per day in 2019

A.7 Cluster Analysis

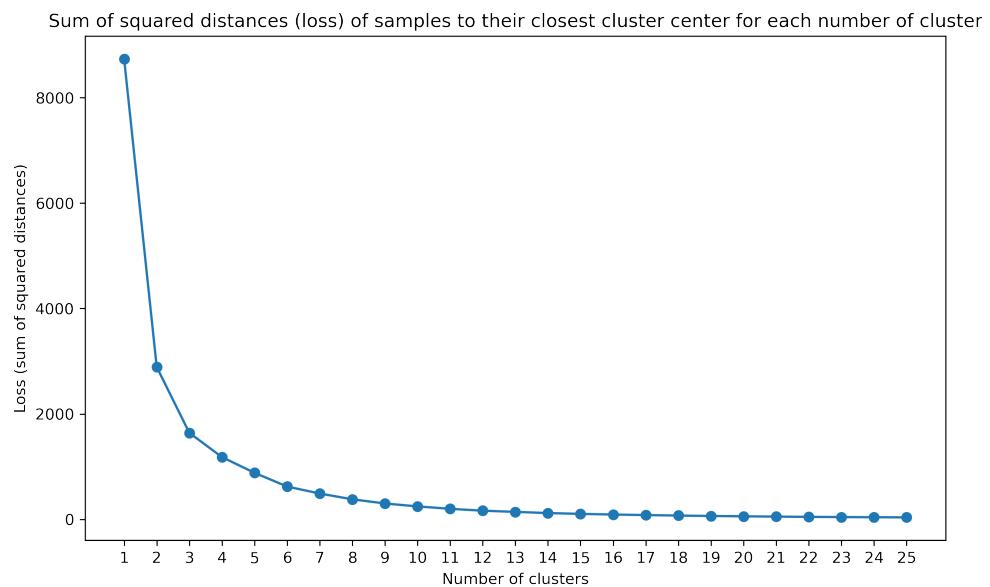


Figure 40: Sum of squared distances (loss) of samples to their closest cluster center for each number of cluster for duration

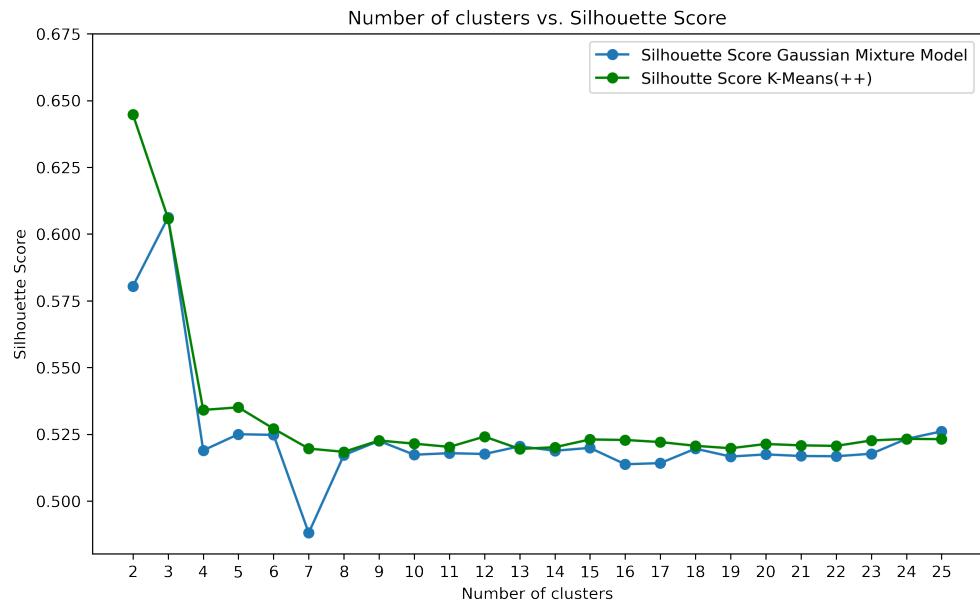


Figure 41: Silhouette score for each number of cluster for duration

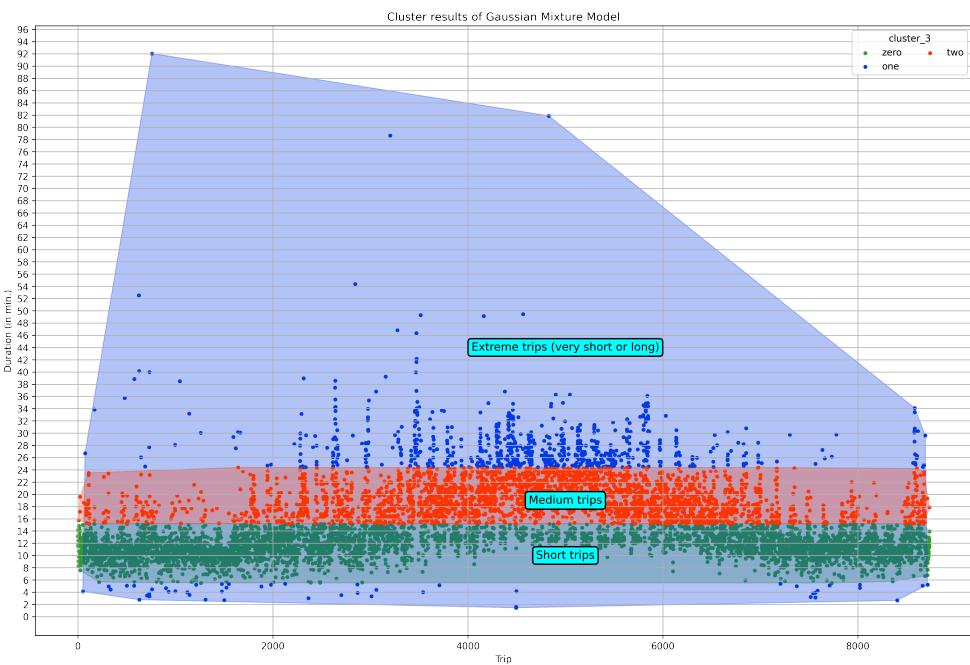


Figure 42: Cluster results of Gaussian Mixture Model

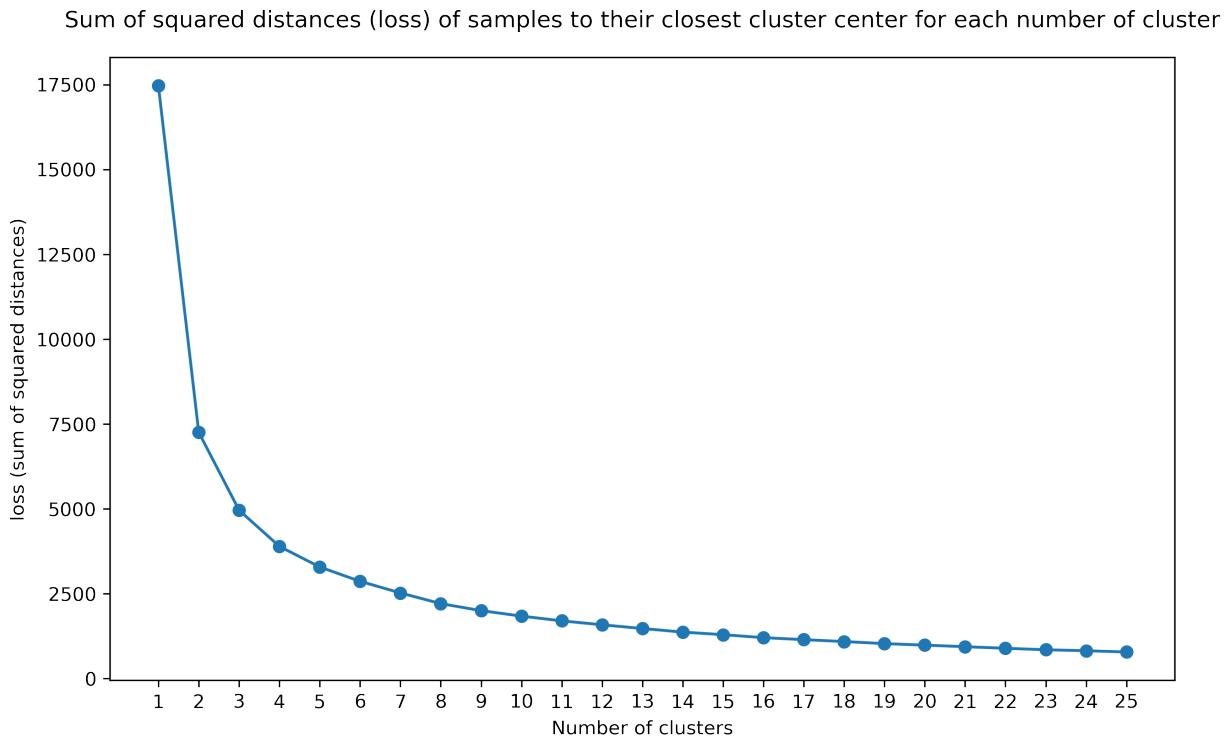


Figure 43: Sum of squared distacnes (loss) of samples to their closet cluster center for each number of cluster (duration and temperature)

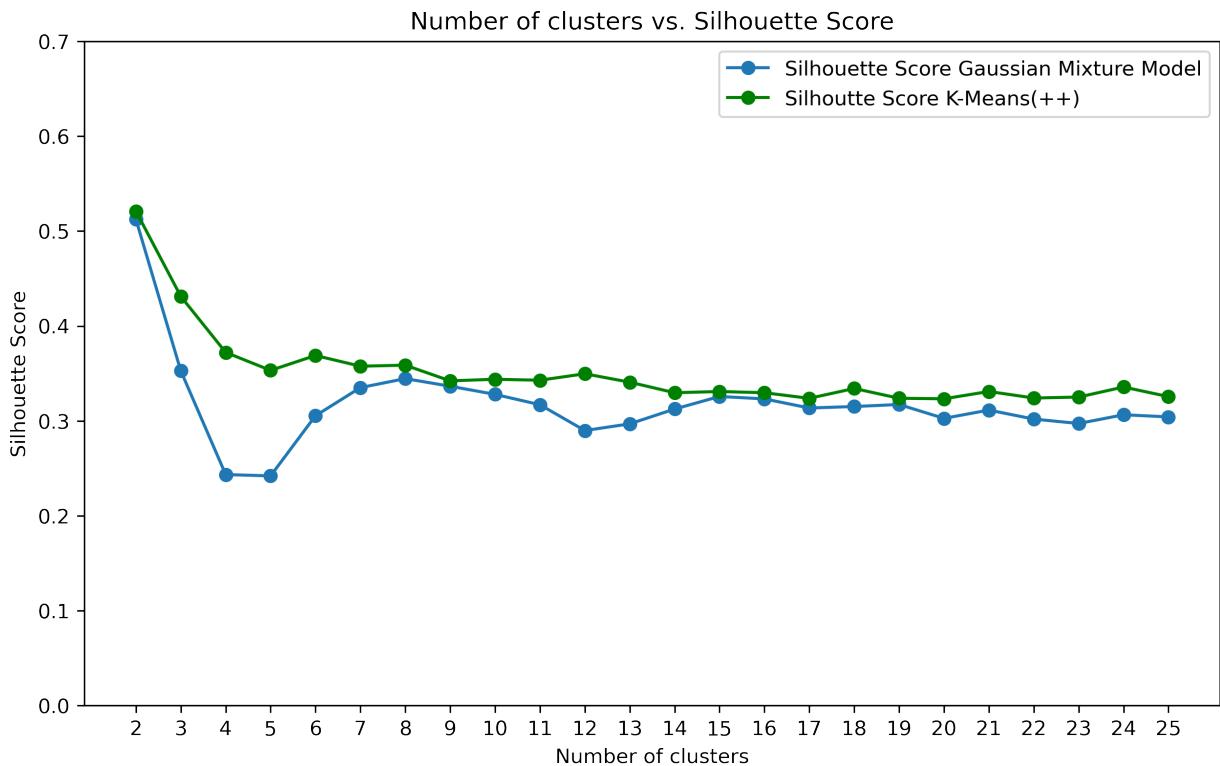


Figure 44: Number of clusters vs. silhouette score (duration and temperature)

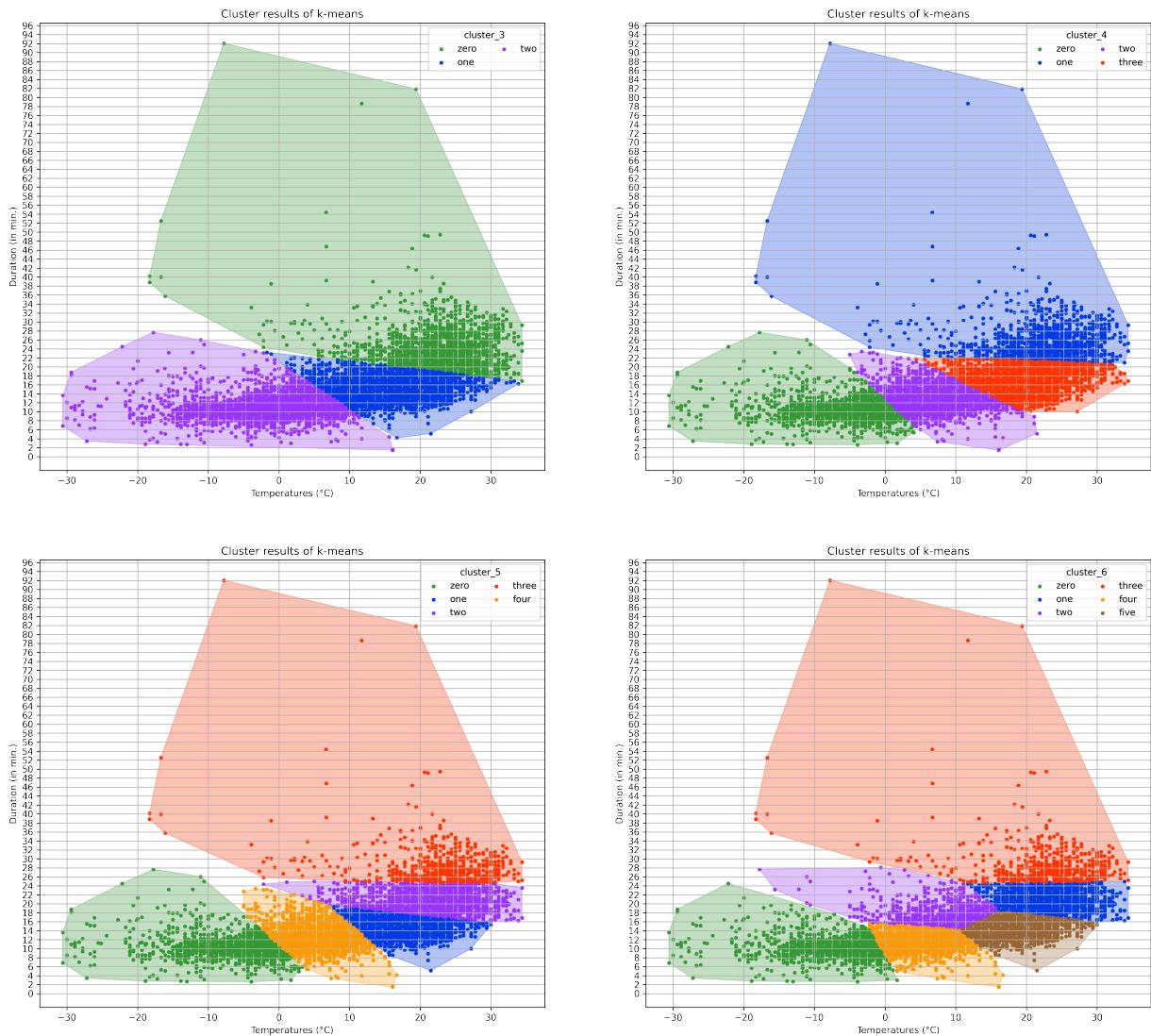


Figure 45: Cluster results K-Means - duration and temperature

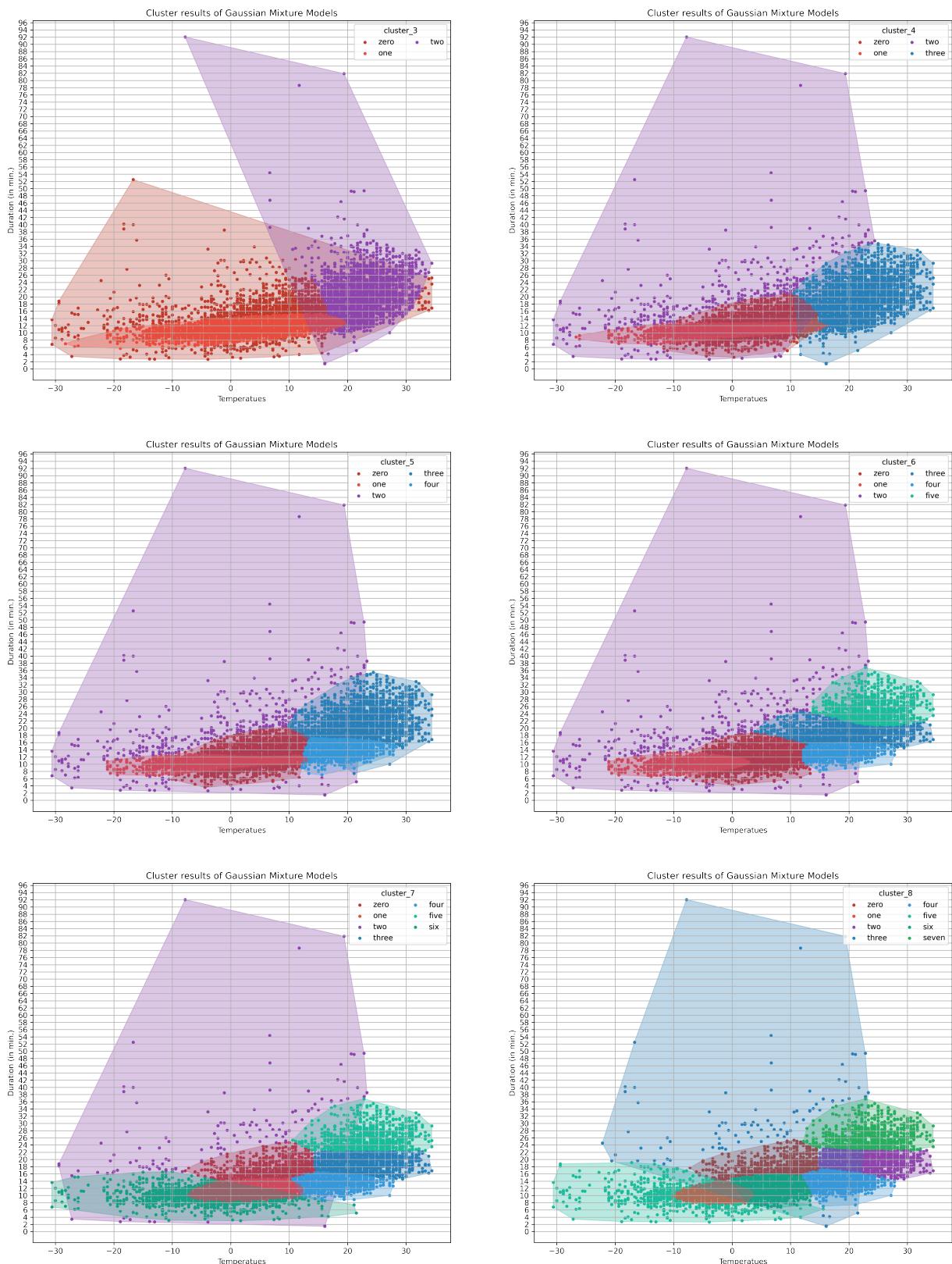


Figure 46: Cluster results Gaussian Mixture Model - duration and temperature

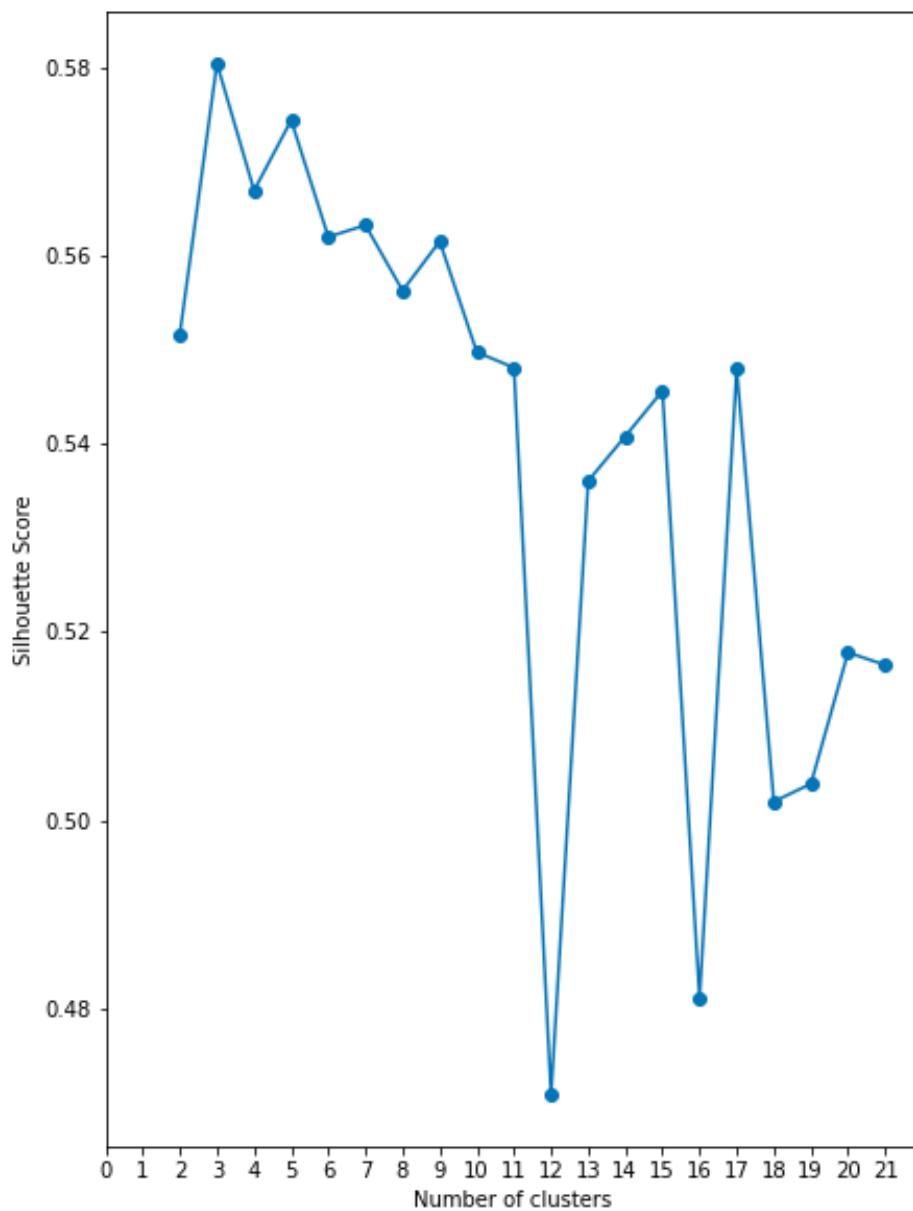


Figure 47: Silhouette score for clustering based on revenue

Revenue clustering using K-Means

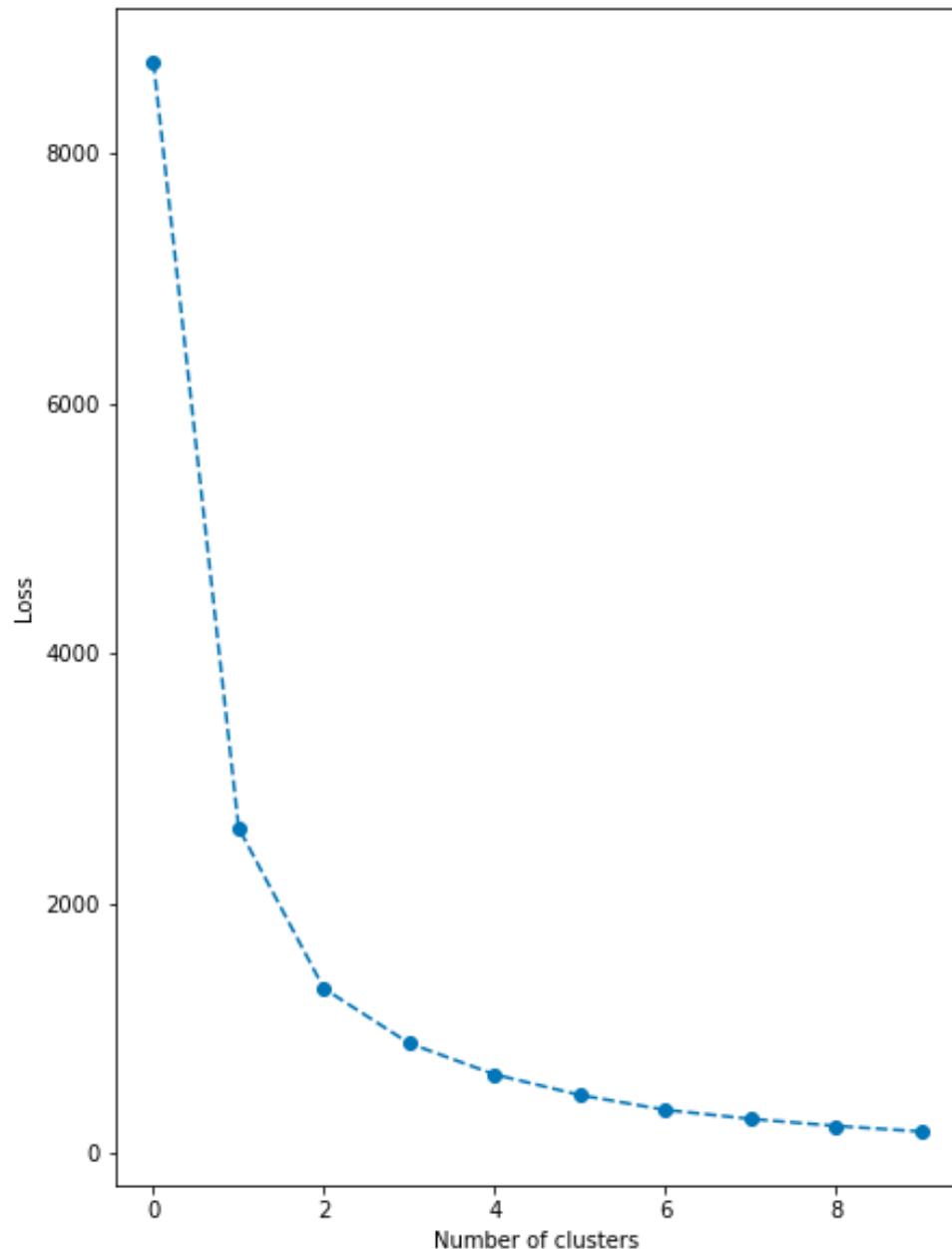


Figure 48: Optimal number of clusters using K-Means instead of GMM based on revenue

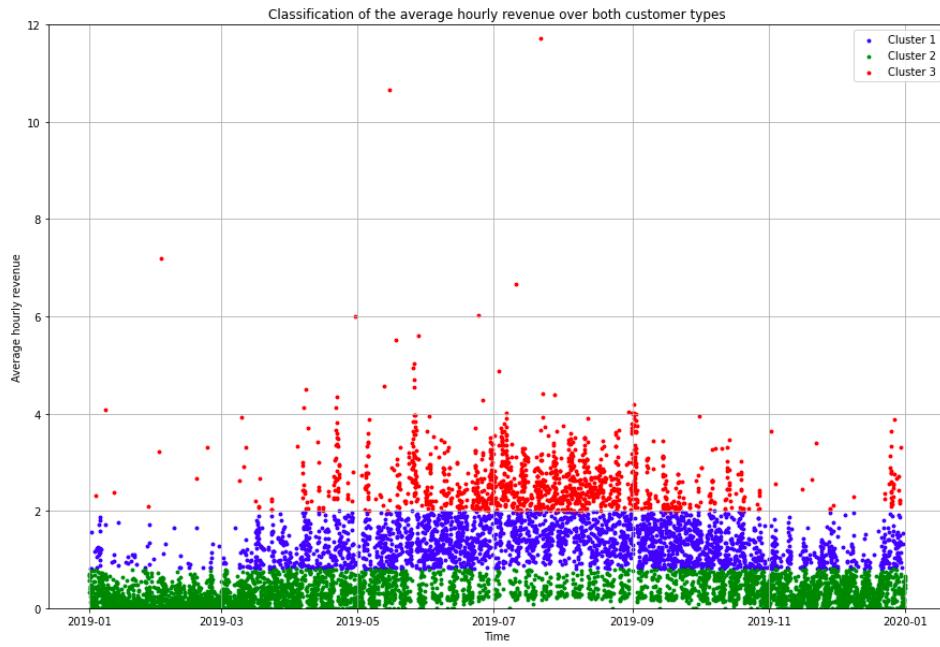


Figure 49: Clustering of the average hourly revenue over both customer types using K-Means

Comparing the here depicted results of K-Means and GMM in the main part, the green layer here is larger. As we consider the GMM result to be more realistic (there are often very short rides below 30 minutes that create only low revenue) we continued to work with the clustering created by GMM in the main section.

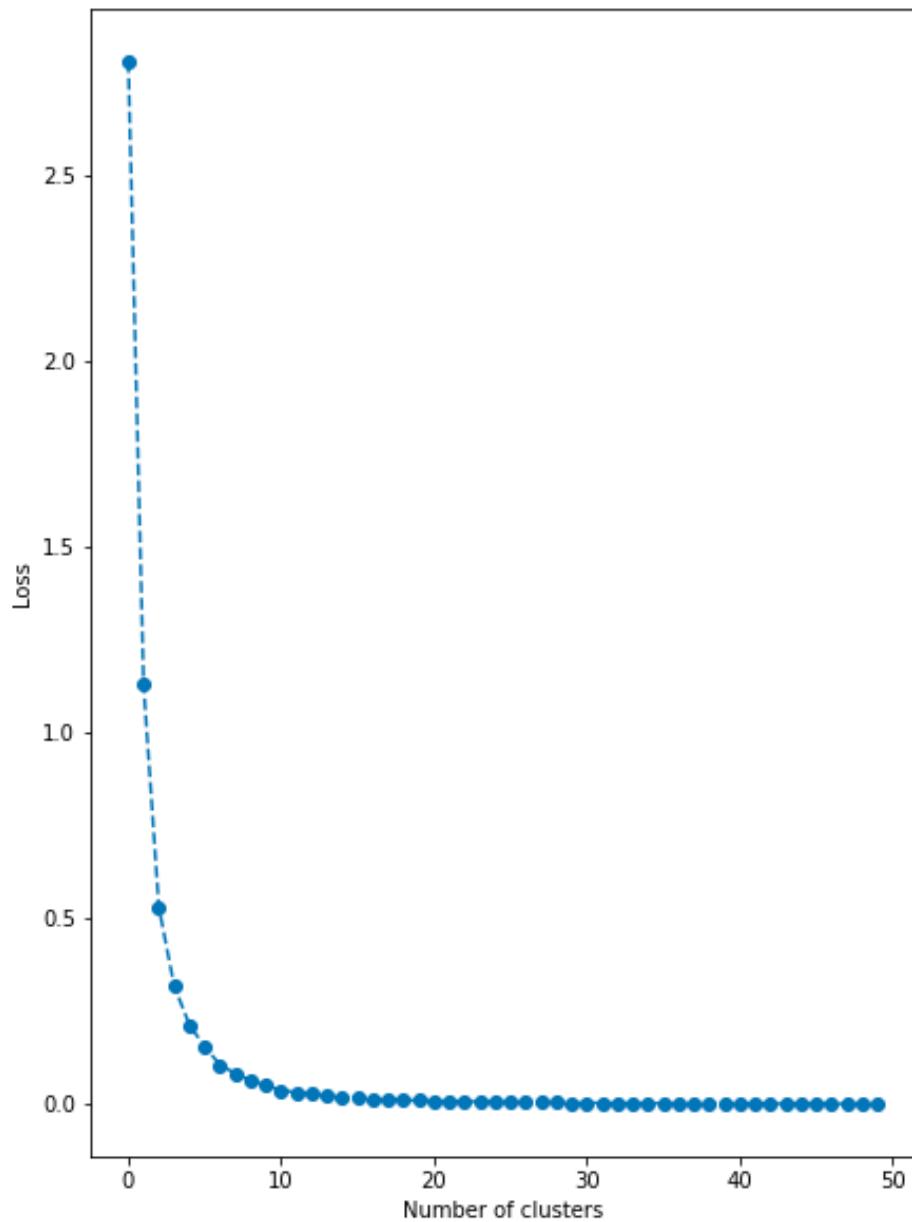


Figure 50: Optimal number of clusters based on Demand-Capacity

Clustering based on revenue and average hourly station temperature

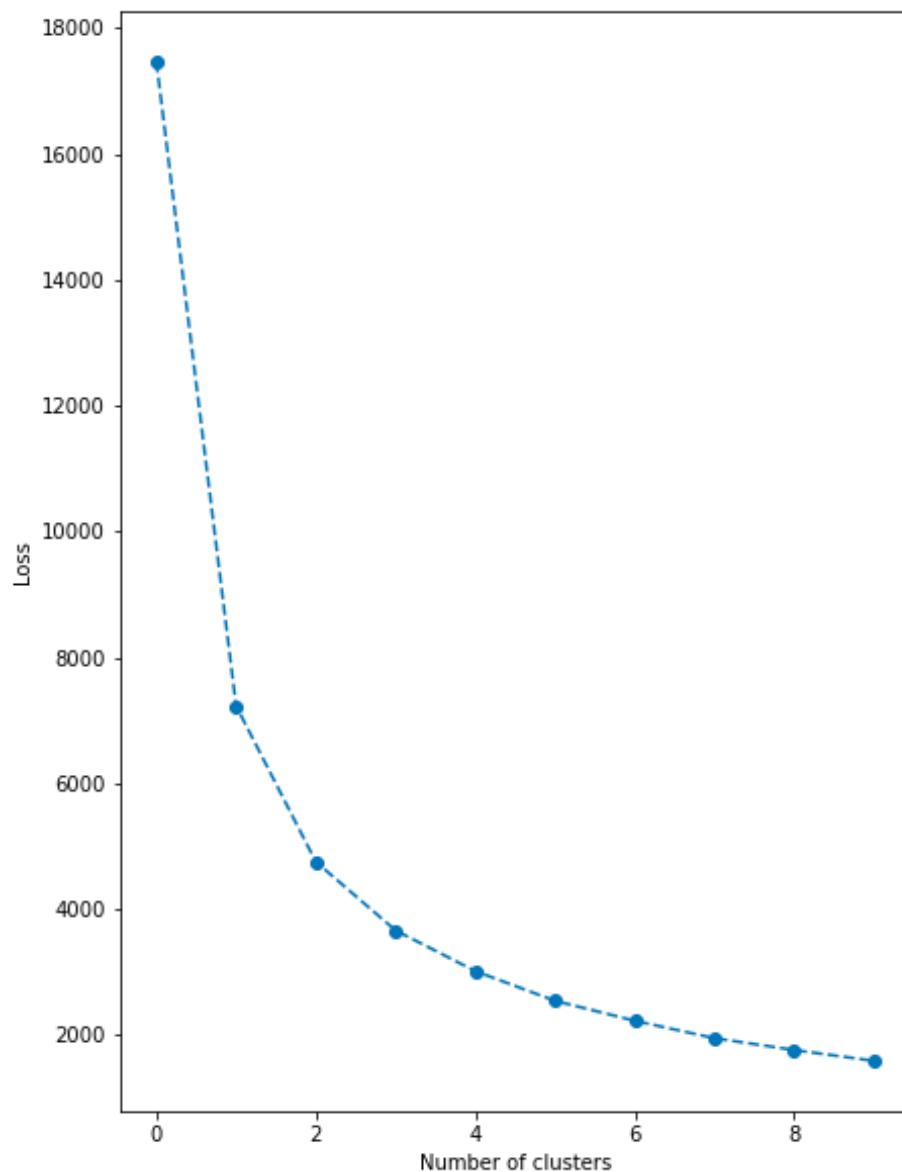


Figure 51: Optimal number of clusters

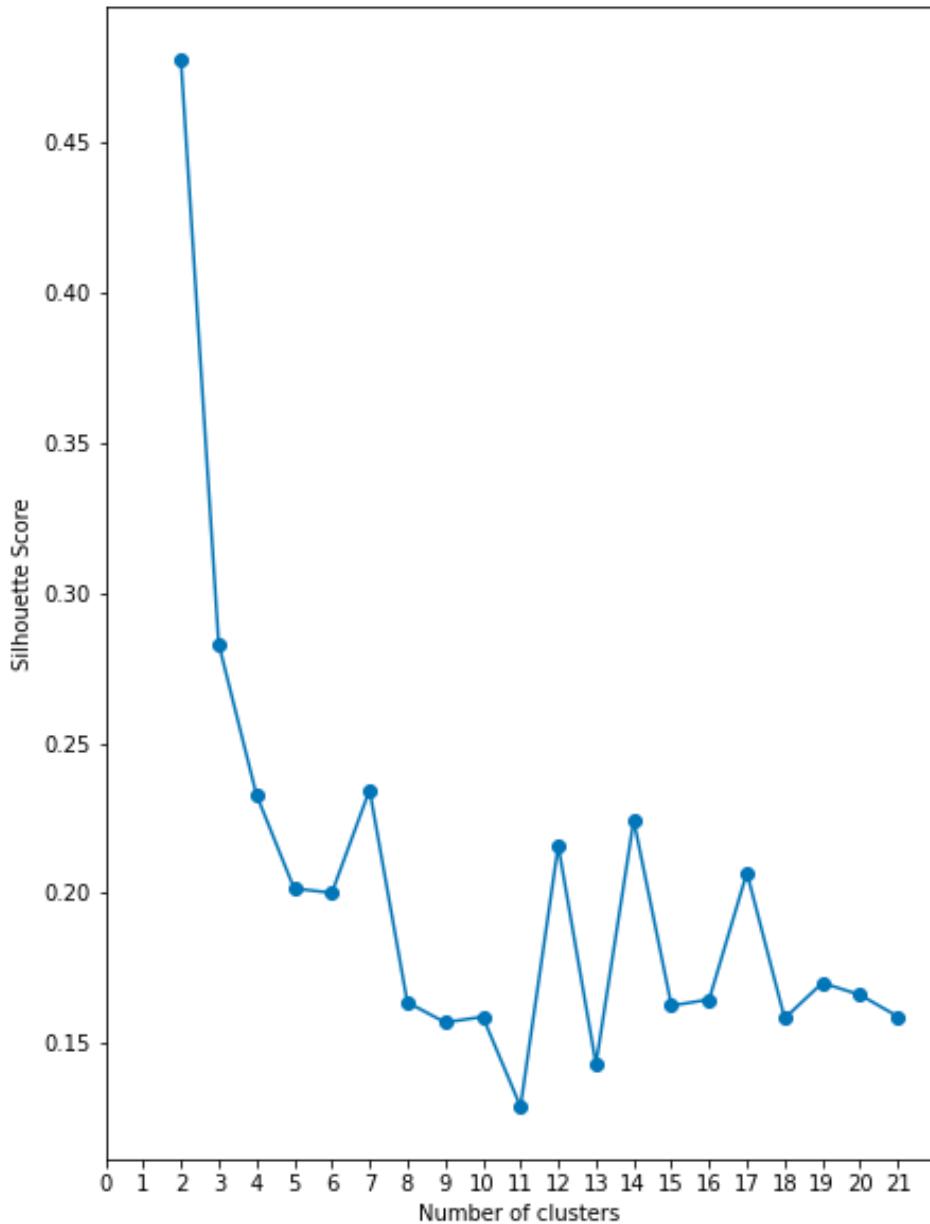


Figure 52: Silhouette score for clustering based on demand-capacity

When we add the average hourly temperature as an input feature, K-Means again yields that 3 clusters are optimal (Appendix 51). GMM indicated an optimum of 2 clusters (Appendix 52), which we decided was too low and thus, GMM are inappropriate.

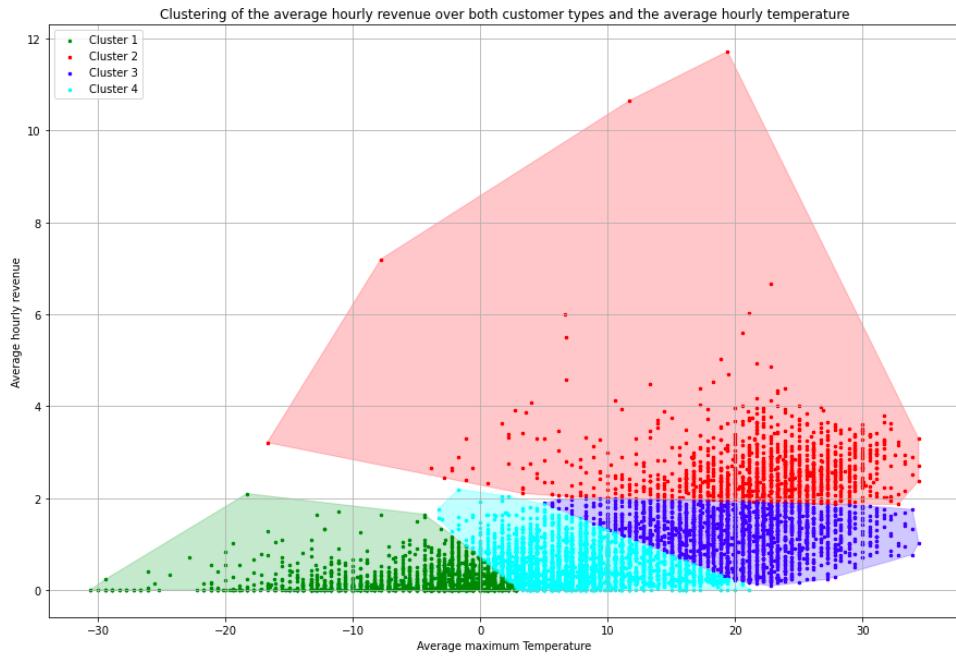


Figure 53: Clustering of the average hourly revenue over both customer types and the average hourly temperature

Investigating the results of the K-Means clustering (Appendix 53), the green, cyan and blue clusters exhibit roughly the same revenue and are mainly distinguished by the average maximum temperature per respective hour. The red cluster exhibits the highest average hourly revenue, together with the highest variance in terms of average hourly temperature, although, as it is to expect, the data points are nevertheless skewed towards temperatures above 0 degrees.

Clustering based on revenue and average yearly Demand-Capacity value

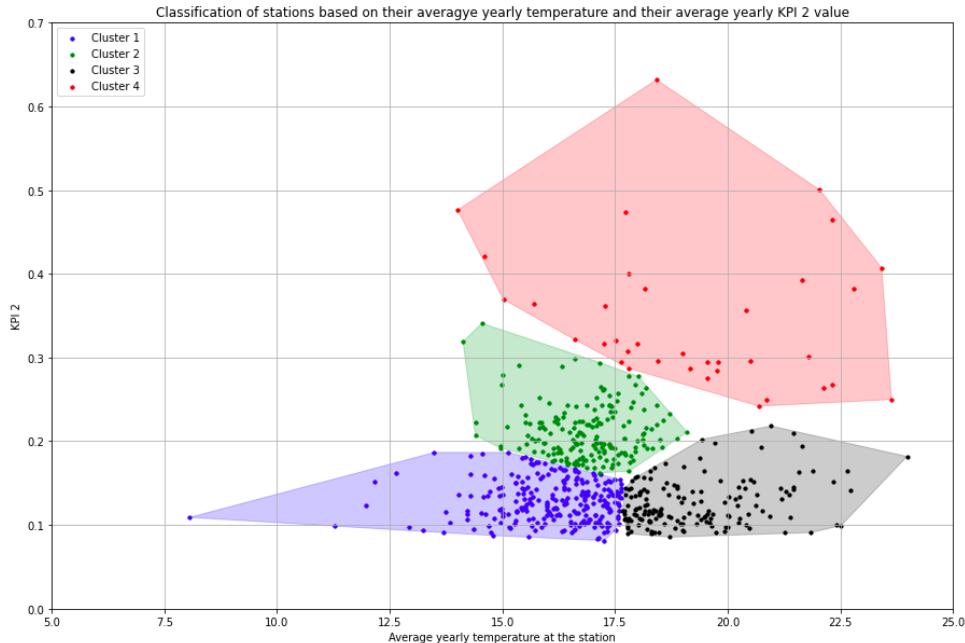


Figure 54: Classification of stations based on their average yearly temperature and their average yearly KPI2 value

Our goal is to try to further differentiate the (former/above described) green and blue clusters to see if any new patterns can be derived. Therefore, we add another feature, the average yearly temperature per station that was present when a ride started. In terms of Demand-Capacity, again three layers arise. However, the lowest layer (black and blue), while exhibiting roughly the same distribution in terms of Demand-Capacity, shows a significant difference in terms of the average temperature of starting rides there. The red cluster is pretty predominant in terms of both temperature and Demand-Capacity.

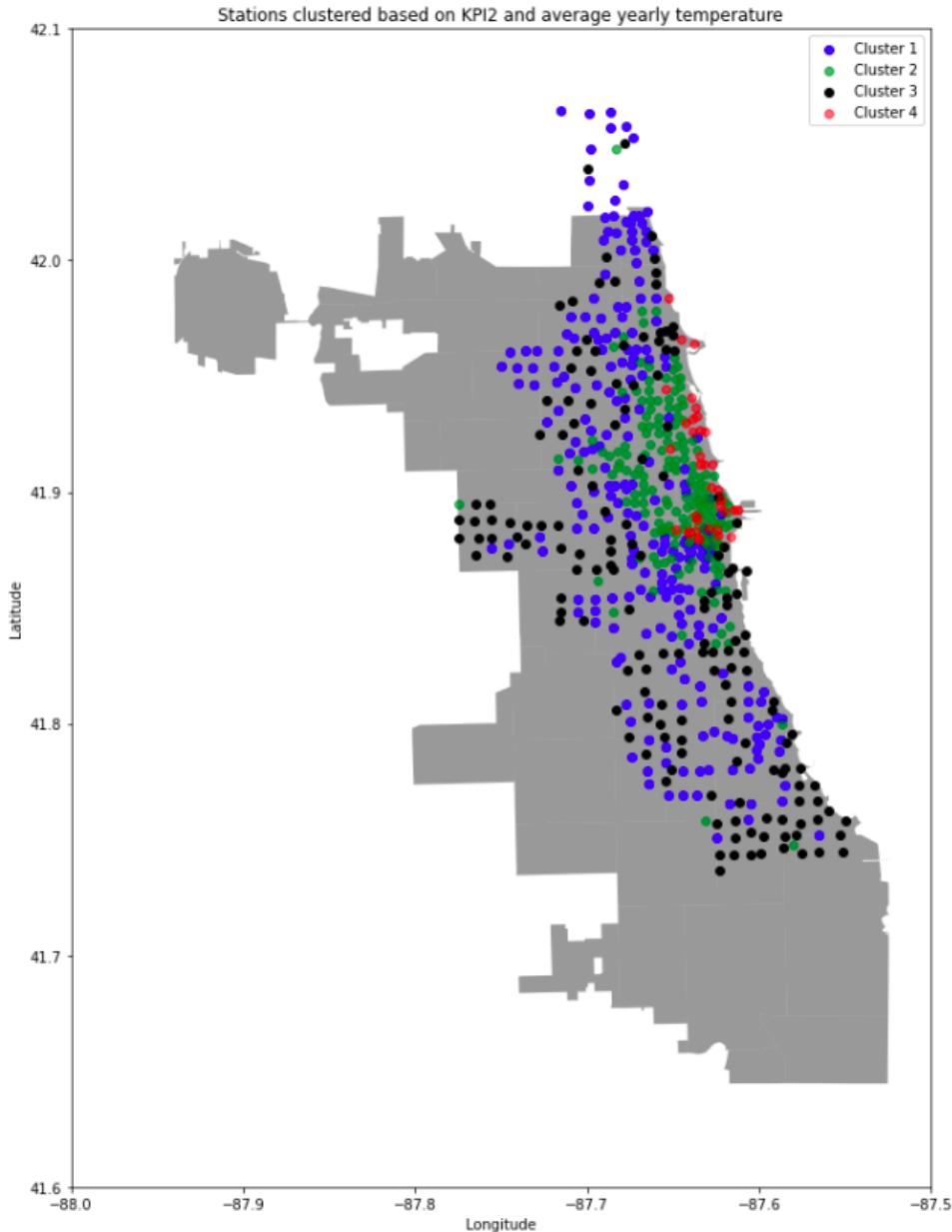


Figure 55: Stations clustered based on KPI2 and average yearly temperature

Interestingly, although we now implemented four clusters, the general structure derived from only taking Demand-Capacity into account remains steady: We still see highly used stations around the lake and two large rings around them. The red stations are still situated mainly around parks etc. and at one small cluster downtown, just as above. However, the "outskirt" stations were differentiated by adding another cluster. While occasionally, there are some areas where black stations are predominant and others where blue stations are predominant, no real differentiation can be made. Thus, average yearly ride starting temperature does not seem to be a very good way to further distinguish the stations.

Clustering based on the total number of starting rides per station

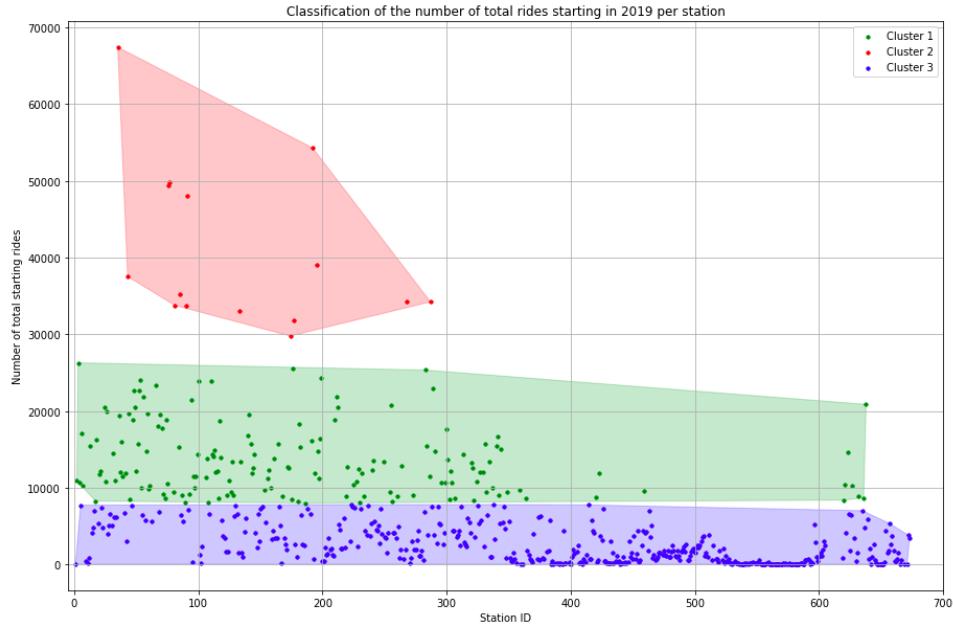


Figure 56: Classification of the number of total rides startig in 2019 per stations

We will introduce an alternative to the Demand-Capacity for two reasons. Firstly, Demand-Capacity is relatively complex conceptually and thus not that easy to understand. Secondly, we want to gain a different perspective and see, what clusters arise then. Thus, we will simply use the total number of starting rides per station over the year 2019 as input feature here. The red cluster stands slightly out as the difference of the distribution of the number of starting rides for that cluster is higher than the difference between the blue and the green cluster.

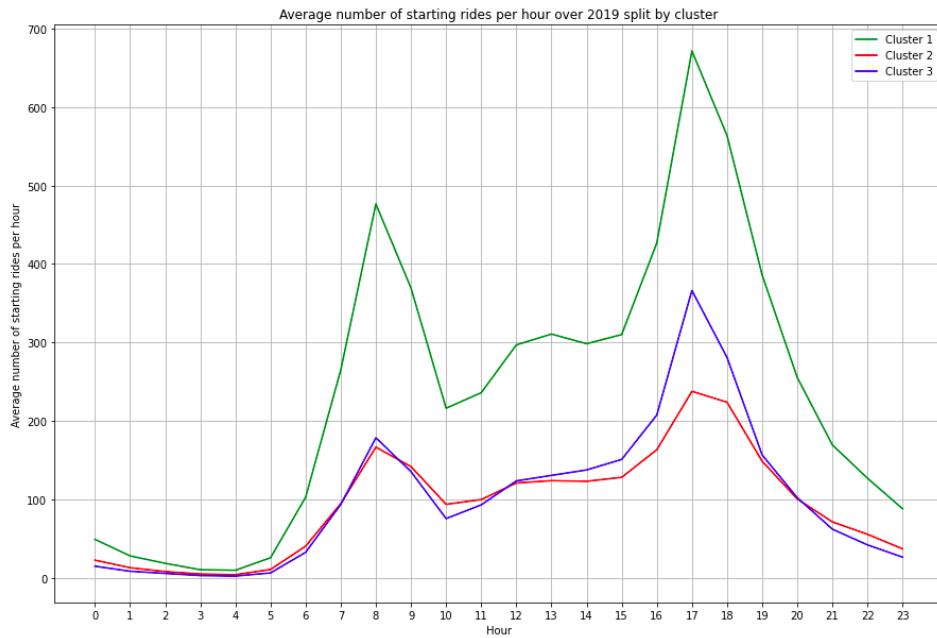


Figure 57: Average number of starting rides per hour over 2019 split by cluster

In the graph above we plot the average number of starting rides per hour over the year 2019. Clearly

and as already observed in the descriptive part of the assignment, the commuter pattern can be seen - all three graphs peak at around 8 and 17 o' clock, respectively. Interestingly, comparing the blue and the red cluster, it becomes clear that both peaks for the red cluster are roughly the same value, while the 17 o' clock peak for the blue cluster is significantly higher than the 8 o'clock peak. However, it is difficult to deduce something related to the type of trips/customers based solely on that observation. Interestingly, although the red clustered stations above account for the highest number of starting rides, the "mediocre" cluster in green stands out significantly in the figure below.

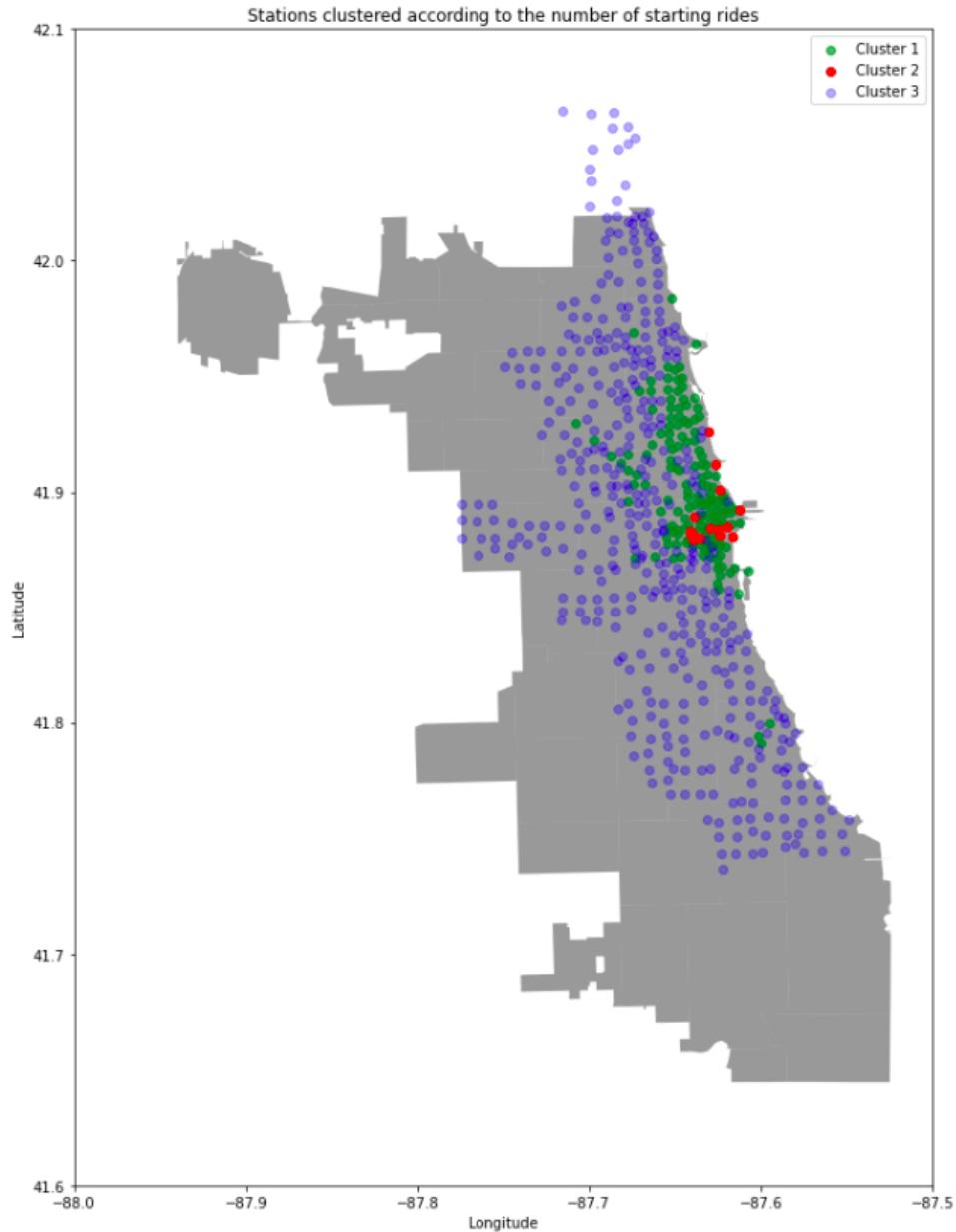


Figure 58: Stations clustered according to the number of starting rides

Interestingly, when creating the same geospatial plot as above (then based on Demand-Capacity values), the number of red stations seems to be smaller, however the sites were those are located is similar to the clustering based on Demand-Capacity. The blue stations seem to be predominant compared to the green ones and the diffusion of those two station types does not be as high as based on Demand-Capacity. This might be due to the fact that Demand-Capacity dissects the different activities on each station more thoroughly and thus yields a more specific, distinguishable image. Nevertheless, the three-ring-structure

as above can still be observed. Hence, both Demand-Capacity and the number of starting rides per station yield, overall, the same distribution of stations and thus, seem to be reasonable measures.

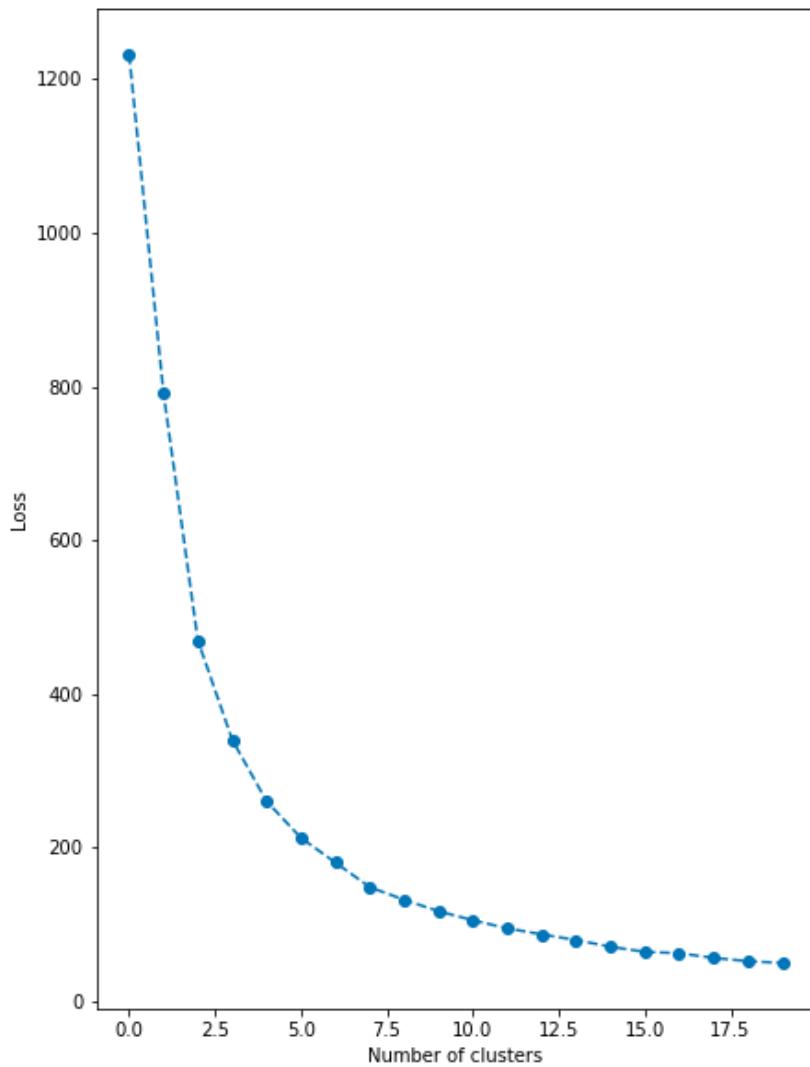


Figure 59: Optimal numbers of clusters for numer of starting rides per station and average duration

A.8 Predictive Analysis

Feature	Description	Correlation	Source/Domain knowledge
Temperature	Average temperature in Celsius for the corresponding time in Chicago	Colder temperature --> Less trips, Warmer temperatures --> More trips	Seasonality pattern, descriptive analytics 2.1
Precip	Indicates if it was raining or not for the corresponding time in Chicago True - Rainfall during time False - Otherwise	If it is raining, less people rent a bike	Precip impact, descriptive analytics 2.1.2.2
Season	Seasons of the year 1 - Spring (May to March) 2 - Summer (June to August) 3 - Autumn (September to November) 4 - Winter (December to February)	Summer season has a higher demand than the winter season	Seasonality pattern, descriptive analytics 2.1
Month	Month of corresponding time	Certain months have higher demands than others (e.g. comparing August and February)	Impact of different months, descriptive analytics 2.1.2.1
IsWeekday	Indicates if the corresponding time is on a weekday or not True - Time is on weekday False - Otherwise	Usually demand is higher on weekdays (people getting to work etc.)	Weekday/Weekend impact, descriptive analytics 2.1.2.2
Weekday	Indicates which weekday is given for the corresponding time 1 - Monday 2 - Tuesday 3 - Wednesday 4 - Thursday 5 - Friday 6 - Saturday 7 - Sunday	The demand can vary from weekday to weekday	Weekday/Weekend impact, descriptive analytics 2.1.2.3
Hour	The hour of the corresponding time	The demand varies during the whole day, e.g. during the night the demand is way lower than during the midday	Hour impact, descriptive analytics 2.1.2.4 & 2.1.2.5
isDay	Indicates if the corresponding time is during the day or not True - From 6 am to 8 pm False - Otherwise	We decided the period from 6 am - 8 pm to be day due to the people's activity and time of sunrise and sunset. During the day the demand of bikes is higher than during the night	Hour impact, descriptive analytics 2.1.2.4 & 2.1.2.5
IsEveningRushHour	Indicates if the corresponding time is during the evening rush hour True - From 4 pm to 6 pm False - Otherwise	During the evening rush hour the demand is very often at its peak, so more bikes are needed.	Literature: "Excess demand prediction for bike sharing systems" (Liu, 2021), "Analysis and Prediction of Bike Sharing Traffic Flow" (Hamad, 2020), "Predicting Bike Usage for New York City's Bike Sharing System" (Singhvi, 2015) + evening rush hour impact, descriptive analytics 2.1.2.5
IsMorningRushHour	Indicates if the corresponding time is during the morning rush hour True - From 7 am to 8 am False - Otherwise	During the morning rush hour the demand is very often at its second highest peak.	Literature: "Excess demand prediction for bike sharing systems" (Liu, 2021), "Analysis and Prediction of Bike Sharing Traffic Flow" (Hamad, 2020), "Predicting Bike Usage for New York City's Bike Sharing System" (Singhvi, 2015) + morning rush hour impact, descriptive analytics 2.1.2.5

Figure 60: Developed features

Hyperparameter	Description	Final Value	Interpretation
Bootstrap	Boolean if bootstrap samples are used	True	Bootstrap is used, i.e. that the trees are not fitted on all available data but on data subsets which can reduce the variance.
Alpha	Parameter for regulating the tree size	0.6	Relatively high value, meaning that there is a larger penalty on the tree size.
Max Depth	Maximum depth of three, meaning how many levels of nodes it can have	90	A tree size of 90 seems relatively high, and overfitting could occur, but the alpha parameter regulates this.
Max Features	Maximum number of features to split at each node	5	For each split 5 random features of our dataset, which comprises 6 in total, are looked at. In CART all features are taken into account, as CART use a greedy algorithm. However, this results in structural similar trees and as we know ensemble methods work best if the sub-models are uncorrelated. Taking 5 features reduces this problem.
Min Samples Leaf	Minimum number of samples required in a leaf node	1	One sample in the leaf means that the tree could be recursively partitioned until each sample is in one leaf. This would overfit the data, however the max depth and min samples split reduce that risk.
Min Samples Split	Minimum number of samples needed to split a node	3	In order to split a node, a minimum number of 3 samples is needed. This can reduce the generalization error, as the tree can not split until each leaf only has one sample.
Estimators	Number of trees on which the prediction is calculated based on mean computation	250	250 decision trees are used, which lies in the typical range of 100 to 1.000 trees and should be sufficient for generalizable prediction.

Table 4: Final hyperparameter values for Random Forest

Model	1	2	3	4	5
R ² (in %)	74.35	95.85	98.81	98.78	98.80
MAE	158.84	119.54	76.30	74.47	74.52

Table 5: Different model results for Random Forest

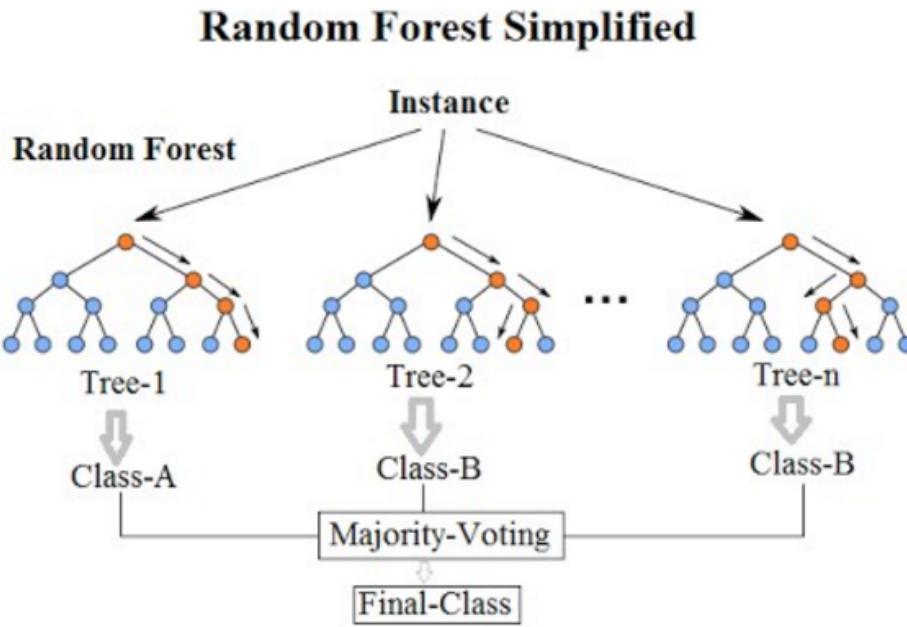


Figure 61: Schematic representation Random Forest (taken from Jagannath, 2020)

Addendum Random Forst: Algorithm and reasons for choosing it

As we know that a Random Forest is a composition of several decision trees, we need to understand the decision tree algorithm. In a decision tree for regression, the tree is built top-down from a root node and each node is recursively partitioned into two parts based on the provided features in order to achieve a minimum variance reduction (in classification information gain). On each node level, each feature is looked at and split various times to see how large the variance would be in that new node. Then the idea is to pick the feature and split which maximizes the variance reduction, i.e. which split reduces the variance from parent to child the most. However, if we let the algorithm run to an end, each training target might result in one leaf. This would introduce massive overfitting. To tackle this issue, pruning the tree is necessary. So the pruning process trades off a higher squared error in the validation dataset against the number of decision nodes in the pruned tree to arrive at a tree that captures the patterns but not the noise. This can be done by using cost complexity, which takes both mentioned aspects into account Figure 62. The alpha value can be varied, meaning at $\alpha = 0$ there is no penalty on tree size, while if α is getting larger the tree size decreases. The idea here is to start with a full-grown tree and then start to increase α gradually until the cost complexity of the full tree exceeds the one of a subtree. This is then repeated until a set of different sized trees and error rates is built. Then one can choose the minimum error tree for a model. The prediction in a leaf is made by taking the average of all instances in that node. On top of this algorithm, a Random Forest has two other hyperparameters (besides these for the decision trees). These are the number of decision trees to calculate the mean computation on and the number of features taking into consideration to split a node. The number of estimators typically lies within a range of 100 to 1.000. Generally speaking, the more trees, the better the results should get. However, the improvement only improve slightly as the number of trees grows, so there is a trade-off between the number of trees and computation time. Via cross-validation, one can select the optimal number. Decision trees are greedy, and they decide which variable to split on by looking which minimizes the error the most. So multiple decision trees can have a lot of structural similarities and as we

know combining predictions work best with uncorrelated sub-models. Here, the second hyperparameter of number of features comes into play because only a subset of features is taking into consideration while splitting a node. This can create different structured trees and improve the generalization ability. We decided to make use of the Random Forest because ensemble methods can lead to a smaller variance and can avoid overfitting. In the case of the Random Forest, the algorithm reduces the risk of overfitting in a single decision tree by using several estimator trees. As another reason, the algorithm works well with both categorical and continuous values, which we both had in our data set, and there is no need for feature scaling, like normalization or standardization, as it uses a rule based approach. Additionally, it can have advantages over bagging algorithms because at each split on a Random Forest not all features are taken into account but only a subset of those available (as aforementioned). This can lead to variations in the structure of trees, which again can reduce the risk of overfitting.

$$CC(T) = Err(T) + \alpha L(T)$$

$CC(T)$ = cost complexity of a tree

$Err(T)$ = proportion of misclassified records

α = penalty factor attached to tree size (set by user)

Figure 62: Cost complexity of a tree

Hyperparameter	Description	Final Value	Interpretation
colsample-bytree	Percentage of features used per tree	0.5	High values can lead to overfitting.
learning rate	The learning rate corresponds to how quickly the error is corrected from each tree to the next and is a simple multiplier	0.08	Chosen to prevent overfitting, usually a value between 0.05 and 0.3 is chosen.
OLR1			
n-estimators	Number of trees on which the prediction is calculated based on mean computation	500	500 decision trees are used, which lies in the typical range of 100 to 1.000 trees and should be sufficient for generalizable prediction. The relatively small learning rate requires a certain amount of trees.
max_depth	Determines how deeply each tree is allowed to grow during any boosting round	6	A tree size of 6 is higher than the default of 3, meaning it can grow more than default.
subsample	Percentage of samples used by tree	1	Low values can cause underfitting. The hyperparameter tuning increased the subsample value from 0.8 to 1.
gamma	Regularization parameter controlling the splitting of nodes (Minimum loss reduction required to make a further partition on a leaf node of the tree).	0.6	The larger gamma is, the more conservative the algorithm will be, that is why 0.6 was chosen.

Table 6: Final hyperparameters results for XGBoost

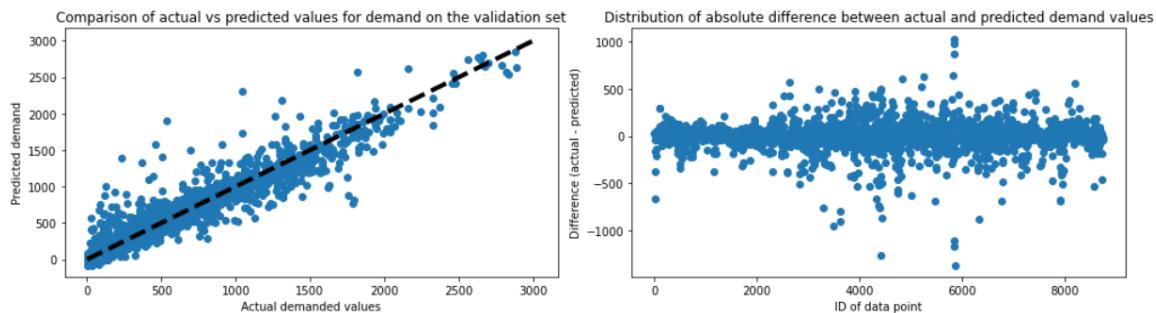


Figure 63: XGBoost result - Comparison of actual vs predicted values for demand on the test set and distribution of absolute difference between actual and predicted demand values

B References

Boor, S., 2019. Impacts of 4th generation bike-sharing, Delft University of Technology. URL: <http://resolver.tudelft.nl/uuid:0ac0d41a-5d86-430a-b6c4-af6b44371f8c>

Yanocha, D., Mason, J., Patlán, M., Benicchio, T., Alfred, T., Laksmana, U., 2018. The Bikeshare Planning Guide. URL: https://3gozaa3xxbpb499ejp301xc8-wpengine.netdna-ssl.com/wp-content/uploads/2013/12/BSPG_digital.pdf

Bayen A. M., Siauw T. (2015). An Introduction to MATLAB® Programming and Numerical Methods for Engineers. Chapter 14: Interpolation. Academic Press. ISBN 978-0-12-420228-3

Jagannath, V., 2020, Random Forest Template for TIBCO Spotfire URL: <https://community.tibco.com/wiki/random-forest-template-tibco-spotfire>

Shyam R., Ayachit S. S., Patil V. and Singh A., "Competitive Analysis of the Top Gradient Boosting Machine Learning Algorithms", 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2020, pp. 191-196, doi: 10.1109/ICACCCN51052.2020.9362840. URL: <https://ieeexplore.ieee.org/abstract/document/9362840>

Divvy, Lyft Inc., URL: <https://ride.divvybikes.com/>

C Supplementary Document

Chapter	Lead responsible, responsible
Data and Preparation	<u>Bastian Schneider</u> , <u>Carsten Stukenborg</u> , Andrej Kotsovolos, Björn Reibke, Isabel Wittmann
Descriptive Analytics	<u>Bastian Schneider</u> , <u>Björn Reibke</u> , <u>Carsten Stukenborg</u> , <u>Isabel Wittmann</u> , Andrej Kotsovolos
Data Analytics (Cluster Analysis)	Björn Reibke, Carsten Stukenborg
Data Analytics (Predictive Analysis)	Andrej Kotsovolos, Bastian Schneider, Isabel Wittmann
Team Report	Andrej Kotsovolos, <u>Bastian Schneider</u> , <u>Björn Reibke</u> , <u>Carsten Stukenborg</u> , <u>Isabel Wittmann</u>

Table 7: Project responsibilities