Programming Data Science – Project Report

SS 2020

Authors: Tobias Olbrück, Lorenz Kriehn, Fabian Rehn & Jonas Fröhlich

Supervisor: Philipp Kienscherf

Department of Information Systems for Sustainable Society

Faculty of Management, Economics and Social Sciences

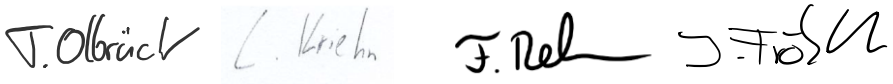
University of Cologne

May 31, 2020

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne die Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden. Ich versichere, dass die eingereichte elektronische Fassung der eingereichten Druckfassung vollständig entspricht.

Die Strafbarkeit einer falschen eidesstattlichen Versicherung ist mir bekannt, namentlich die Strafandrohung gemäß § 156 StGB bis zu drei Jahren Freiheitsstrafe oder Geldstrafe bei vorsätzlicher Begehung der Tat bzw. gemäß § 161 Abs. 1 StGB bis zu einem Jahr Freiheitsstrafe oder Geldstrafe bei fahrlässiger Begehung



Tobias Olbrück, Lorenz Kriehn, Fabian Rehn & Jonas Fröhlich

Köln, den 08.06.2020

Table of Content

1. Executive Summary

Looking at the problem of global warming it can be considered that the traditional mobility sector is a big contributor if it comes to the emission of carbon dioxide and other greenhouse gases. Therefore, newer and "greener" alternatives for transportation have to be developed and used. One of these alternatives is bike sharing, categorized as a sharing economy model called mobility-as-a-service (MaaS), especially for urban areas.

For the service providers of those bike sharing models it is important to analyze and participate the usage patterns of their service, to overcome shortcomings and to provide a good user experience to their customers. Data Science can help with this challenge.

In this project our team will handle bike sharing data for the city of Marburg in the year 2019 received from nextbike, Germanys biggest bike sharing provider. In different steps, we will achieve a common data understanding, clean and prepare the data, explore and describe it, present some advanced visualizations and finally build and evaluate prediction models for the trip duration, the number of trips and whether a trip goes to the University of Marburg or not.

We are able to predict the trip duration with a mean absolute error (MAE) of 184 seconds and a R^2 of 0.82 for the test set, using the scikit learn Random Forest Regressor, who outperformed various built polynomial models. Furthermore, we can predict if a trip heads towards university with an overall accuracy of 74 percent for the test set by a scikit learn KNN algorithm. On top, we can predict the number of trips per day with a MAE of 289 trips and a R^2 of 77 percent for the test set, again using a Random Forest Regressor.

Sadly, this positive values regarding the error metrics do not continue if it comes to the holdout set, which includes the data for the month of July, which was missing for train and test purposes. Just the prediction for the number of trips achieves quite good results on the holdout set, but the prediction for the trip duration and if a trip goes towards university suffer. Those predictions are quite as bad or even weaker than just taking the overall average as prediction result.

2. Data Cleaning and Data Preparation

For the raw data, four different sources have been considered. Nextbike provided all data regarding their bike sharing service in Marburg, the German weather service DWD provided

weather data and for geoinformations, two different approaches were used. The data was provided by OpenDataSoft and the Nominatim API.

2.1 Weather Data

From the perspective of business understanding it is known and logical, that the weather is an important factor with great influence on the usage patterns of bike sharing services. Therefore, we aggregated weather data from the German weather service DWD for Marburg, focusing on temperature and precipitation data, as we know from previous projects, that these are the most influencing ones. First, we retrieved hourly data, which unfortunately had high amounts of missing data over several weeks, which we could not interpolate. Nevertheless, the available data with a ten-minute interval was nearly complete. We joined the data on an minutely interval for the whole year of 2019 and used a forward fill to fill gaps, as the bike sharing data is also presented on a minutely datetime-level.

date	temperature	precipitation
2019-01-01 00:00:00	7.6	0.0
2019-01-01 00:01:00	7.6	0.0
2019-01-01 00:02:00	7.6	0.0
2019-01-01 00:03:00	7.6	0.0
2019-01-01 00:04:00	7.6	0.0

Table 1: Weather data

2.2 Bike Sharing Data

Reading in the raw bike sharing data provides a dataframe with around 1.5 million columns and exactly 13 attributes for each column.

	p_spot	p_place_type	datetime	b_number	trip	p_uid	p_bikes	p_lat	b_bike_type	p_name	p_number	p_lng	p_bike
0	True	0	2019-01-20 02:06:00	11281	first	4774295	5	50.808852	15	Biegenstraße/Cineplex	5155.0	8.773134	False
1	True	0	2019-01-20 14:16:00	11281	last	4774295	4	50.808852	15	Biegenstraße/Cineplex	5155.0	8.773134	False
2	True	0	2019-01-20 00:00:00	11169	first	4774543	5	50.795224	15	Südbahnhof	5173.0	8.763266	False
3	True	0	2019-01-20 01:55:00	11169	start	4774543	5	50.795224	15	Südbahnhof	5173.0	8.763266	False
4	True	0	2019-01-20 02:06:00	11169	end	4774368	4	50.804522	15	Frankfurter Straße/Psychologie	5159.0	8.770358	False

Table 2: Raw trip data

Each of the 13 attributes, except of datetime and trip, has either the **prefix p or b**. The prefix p indicates that the content of the column contains information about the place, where the bike is present at this moment in time. The prefix b indicates that the content of the column contains information about the unique bike itself.

Interpreting the exact **meaning of attributes**, the following information is maintained:

attribute name	meaning
p_spot	boolean indicator which is always the exact opposite of p_place_type
p_place_type	binary (0/1) indicator if the place is an official fixed station or not
datetime	timestamp of the recorded event
b_number	unique bike number/id
p_uid	unique place id, regardless of if it is an official fixed station or not
p_bikes	number of bikes present at this place
p_lat	latitude coordinate of the place
b_bike_type	type of the bike. According to nextbike: normal, electric and cargo bikes.
p_name	name of the place
p_number	unique place number if it is an official fixed station
p_lng	longitude coordinate of the place
p_bike	boolean indicator which is always equal to p_place_type

Considering the three different attributes p_spot, p_place_type and p_bike it was most clear, that p_place_type indicates at a binary level, whether the place is an official fixed station of nextbike (1) or not (0). It was found that the other two attributes are both exactly correlated with a correlation of 1 or -1 to p_place_type, discarding an outlier of only 6 records out of more than 1.5 million records, which all happened at the same date and place. Therefore, for the city of Marburg, we cannot see a different meaning for those two attributes. It could be possible, that this kind of data model has its sense for another city.

Until now, we did not describe the **meaning of the attribute trip**. This attribute is a categorical one with the four possible values ‘first’, ‘last’, ‘start’ and ‘end’. There are two general possibilities of event types, which are captured. First, it can indicate whether the record is the

start or the end of a trip, using the categorical values ‘start’ or ‘end’. Second, it seems that the values ‘first’ and ‘last’ are used as a status check before midnight (first) and after midnight (last) to get an updated information about the unique bike. For the analysis of mobility patterns, the records indicating the starts and ends of trips are more informative, as they represent the trips of users and show, generally spoken, at what times users go from what start place to what end place.

To do the further analysis and regarding the prediction goal it was useful to transform the bike sharing data into other structural representations than the described original event recording above. A **trip oriented data frame** is useful, which contains the attributes bike number, start time, end time, duration, start longitude coordinate, start latitude coordinate, end longitude coordinate, end latitude coordinate, weekend, bike type, start place number and end place number. The three additional attributes bike type, start place number and end place number have been chosen based on business understanding, as it can be expected that they are indicators for different types of trips and therefore could correlate with the duration or end position of trips, which would make them useful as possible input features for the prediction targets.

	bNumber	sTime	eTime	duration	sLong	sLat	eLong	eLat	weekend	bType	sPlaceNumber	ePlaceNumber	durationInSec
0	11169	2019-01-20 01:55:00	2019-01-20 02:06:00	00:11:00	8.763266	50.795224	8.770358	50.804522	True	15	5173	5159	660
2	11169	2019-01-20 11:58:00	2019-01-20 12:06:00	00:08:00	8.770358	50.804522	8.759248	50.804725	True	15	5159	5178	480
3	11169	2019-01-20 15:12:00	2019-01-20 15:27:00	00:15:00	8.759248	50.804725	8.774681	50.822927	True	15	5178	5150	900
4	11169	2019-01-20 15:39:00	2019-01-20 15:42:00	00:03:00	8.774681	50.822927	8.774681	50.822927	True	15	5150	5150	180
5	11270	2019-01-20 00:25:00	2019-01-20 00:37:00	00:12:00	8.775948	50.813203	8.775948	50.813203	True	15	5156	5156	720

Table 3: Trip data

For this data frame, data was cleaned in three different steps. First, during the creation of the trip format data frame, all trips are knowingly ignored, which have a missing start or end record in the raw data. These entries are flawed. Second, as we are interested in predicting the trip duration, we compare the trip duration of every trip to the calculated mean trip duration on a daily but also on a monthly level and drop those trips, which exceed one standard deviation on a daily level, or a half standard deviation on a monthly level. On a first view, this cleaning looks quite hard, as we lose around eight percent of all trips, but looking at mean and median trip duration on a daily level confirms our decision. While the change of the median trip duration is extremely small, the change of the mean trip duration is very impressive and now around three times smaller than before. This implicates that a small number of very long trips with durations of several hours, which could be even partially generated by failures in bike returns, corrupted

our data and added very high variance on it. The third method of data cleaning is explained in section 2.3.

As a second format of data representation it was decided to create a data frame with a minutely index for the whole year 2019. This data frame contains the different official fixed stations of nextbike in Marburg by their station number and stores the information about the number of bikes per fixed station available for every minute of the year.

	5140	5141	5142	5143	5144	5145	5146	5147	5150	5151	...	5168	5169	5171	5172	5173	5174	5175	5176	5177	5178
datetime																					
2019-01-01 00:00:00	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2019-01-01 00:01:00	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2019-01-01 00:02:00	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2019-01-01 00:03:00	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2019-01-01 00:04:00	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

Table 4: Bikes per Station Datetime Index

Third, a data frame containing all different fixed stations with station number, station name and station coordinates was created, to make this information easily accessible.

pNumber	pName	pLat	pLong
0	no fixed station	NaN	NaN
5140	Anatomie	50.816058	8.772242
5141	Ketzerbach/Zwischenhausen	50.813950	8.766160
5142	Friedrichplatz	50.803270	8.764060
5143	Interkulturelle Gaerten/ Am Richtsberg	50.794700	8.772230

Table 5: Stations

As fourth presentation of data, a data frame representing the number of trips per day was created. It contains the date, number of trips, temperature, and precipitation. Outlier dropping resulted in the deletion of eleven days, on which the number of trips per day exceeded the monthly standard deviation.

	date	tripsPerDay	temperatureMAX	temperatureAVG	temperatureMIN	precipitationAVG	day	month	dayOfWeek
0	2019-01-20	843	1.5	-7.373611	-11.6	0.000000	20	1	6
1	2019-01-21	1182	-0.7	-8.700694	-12.9	0.000000	21	1	0
2	2019-01-22	1038	-3.2	-5.825694	-12.3	0.000000	22	1	1
3	2019-01-23	1221	-1.7	-3.445139	-5.9	0.000000	23	1	2
4	2019-01-24	1458	-2.5	-4.198611	-5.2	0.001111	24	1	3

Table 6: Number of Trips per Day

2.3 Postal Codes Data

As postal codes for the start and end location of trips are needed for visualization purposes, two different methods for retrieving them were tested. As first option, a GeoJSON file which contains the border coordinates for all relevant postal code areas of Marburg was downloaded from OpenDataSoft (https://public.opendatasoft.com/explore/dataset/postleitzahlen-deutschland/table/?q=marburg). Using the shapely library, a point represented by one coordinate can be assigned to the postal code area. The method is accurate, but not fast enough for the massive number of trips we must consider.

As second option, a grid is created, which has nodes at evenly spaced intervals. The use of 12.000 nodes results in a distance of 167 meters between two nodes. For each of these nodes the postal codes were requested using the geopy library and the Nominatim API. Caused by the limitations of Nominatims free version (one request per second) this is a very time-consuming step. Fortunately, it only needs to be executed once. Through a saving of the results in a .csv file it is not necessary to repeat this. The created grid is then used to train a nearest neighbor algorithm. Subsequently, the algorithm is used to predict the postal codes for each start and end point. Once the data is present, it is much quicker than the first method.

To select which method to use, both methods were evaluated in terms of duration and accuracy:

Method	Execution Time in Seconds	Accuracy
Geojson	424	95%
Grid	22	95%

Since none of the methods is one hundred percent accurate, the grid method was chosen because of its significantly shorter runtime.

After assigning the postal codes to every start and end locations of all trips, once again the trip data was screened for flawed records. If a trip did not start and end inside the geographical borders of the city of Marburg, it was dropped. That only led to a small number of around 700 lost trips.

3. Exploration and Description

After the initial data cleaning and data preparation, we visualized our data to gain first insights. Based on previous project experience, we already knew that seasonality has an influence on the riding behavior of bike-sharing users. To make different figures comparable, we have decided to always use the same figure structure. Most of our figures are therefore divided into 3 parts (month, day & hour).

To test our initial hypothesis that seasonality has an influence on driving behavior, we first visualized the number of trips per month, day, and hour.

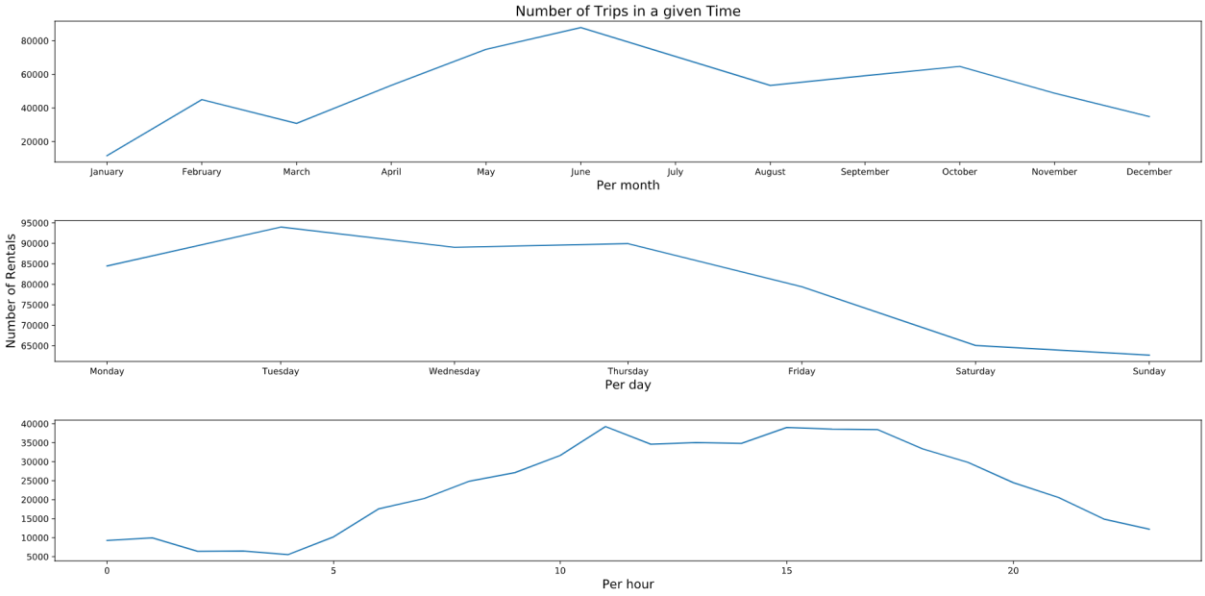


Figure 1: Number of Trips

As can be seen in figure 1, seasonality plays a major role for the number of trips. It can be clearly seen that the number of trips increases in summer month and decreases again in autumn. However, it must be mentioned that our underlying data set does not contain any data for the month of July. It can be assumed that these missing data will be provided at a later stage for evaluating our deployed prediction models.

Contrary to our expectations, the number of trips per day and hour is distributed rather unusually. The number of trips on weekends is decreasing very strongly. As a team we expected the opposite. For the weekend, the number of rides usually increases since many people use the rental bikes for recreational rides. Such a distribution is known for major German cities (e.g. Munich).

One explanation for this rather unusual distribution of trips could be that Marburg is known as a “Studentenstadt”. One quarter of Marburg’s 80,000 inhabitants are students (https://www.e-

fellows.net/Studium/Schule-Abi-und-dann-studieren/Uni-Staedte-von-A-bis-Z/Studieren-in-Marburg). It can therefore be assumed that the bike-sharing service is predominantly used by students.

This would also explain why the number of journeys reaches its maximum at approximately 11 o'clock relatively late. People who use a rental bike to go to work would most likely use the service much earlier. So, most people start working between 8 and 9 a.m. This in turn would support our statement that the rental bikes in Marburg are mainly used by students.

Further proof that it is mainly students who use the rental bikes was found after researching the price structure. From May 2019 Nextbike has adjusted its pricing structure. The first 30 minutes of a bike rental are free for everyone. Previously these were only free for students, but this change did not lead to significantly more and shorter rides.

In order to draw further conclusions from the data, the average trip length is examined in figure 2.

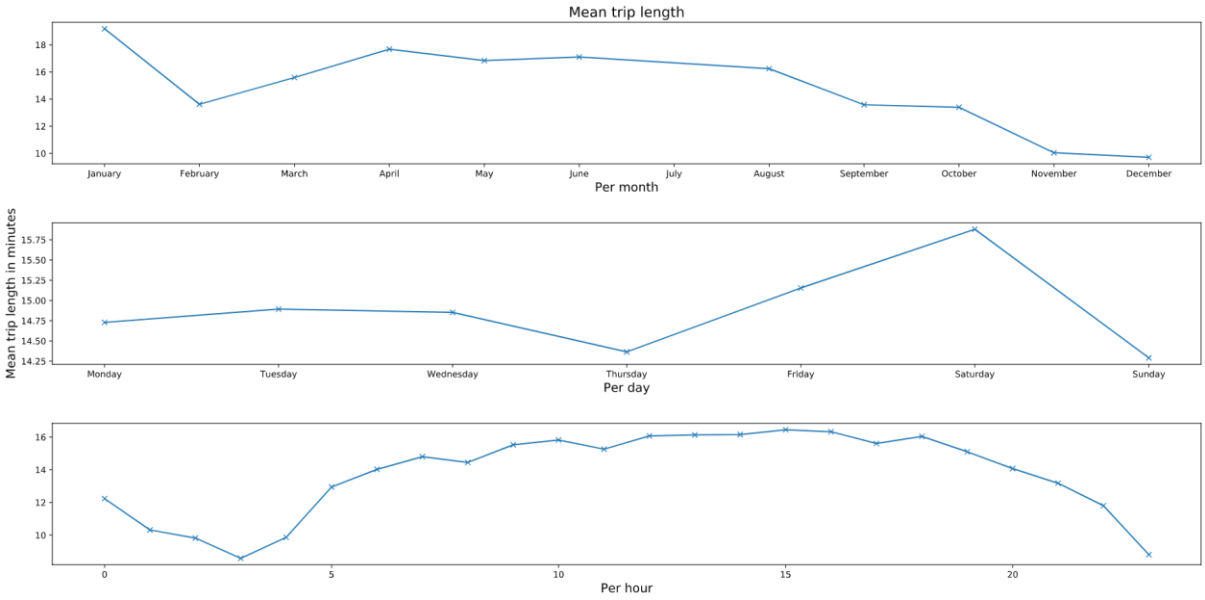


Figure 2: Mean Trip Length

It is particularly noticeable that journeys in January are comparatively long. Moreover, the monthly standard deviation, which can be seen in figure 3, is relatively high. In our dataset there is no data available before 20 January. Therefore, longer journeys are more influential. The longer trip duration cannot be explained by particularly good weather in January, as the weather data proves which will be evaluated later in this chapter.

Apart from the longer journeys in January, there is again a trend that the length of journeys increases towards summer and decreases again from autumn onwards. This can be explained by the typically more bicycle-friendly weather during summertime.

Although there are significantly fewer trips at the weekend, at least on Saturdays they are longer than during the week, but the mean trip length decreases again on Sundays.

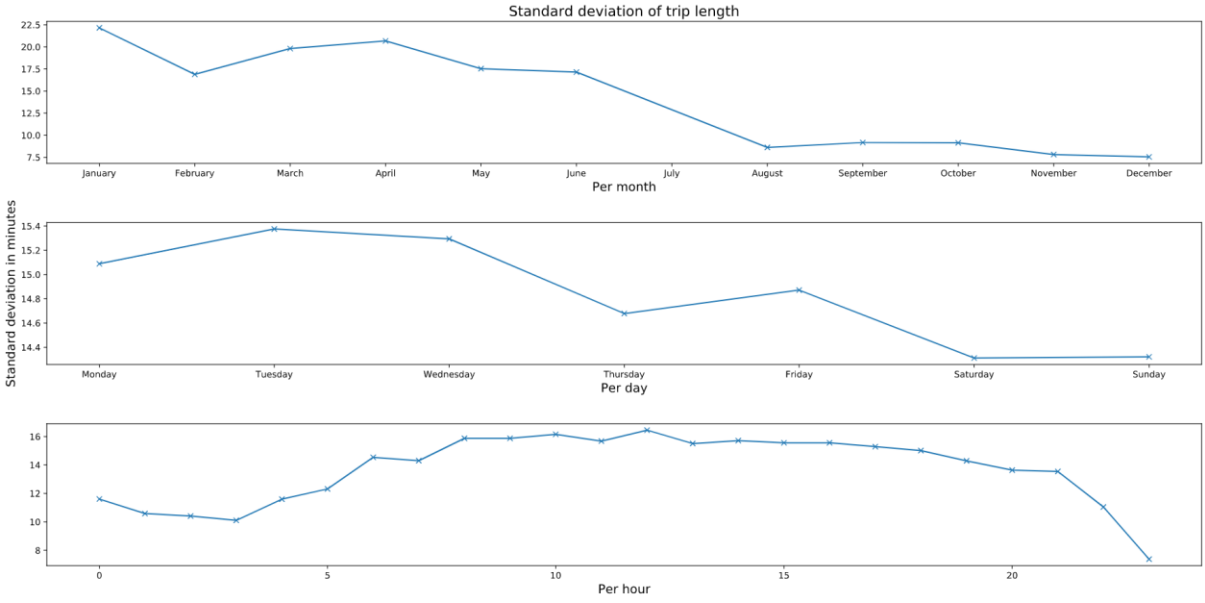


Figure 3: Standard Deviation of Trip Length

Further and detailed information, such as the different quartiles of the trip length, can be taken from the box plots in the appendix.

In order to be able to determine the trip length more accurately later, we have also evaluated weather data from Marburg from the German Weather Service.

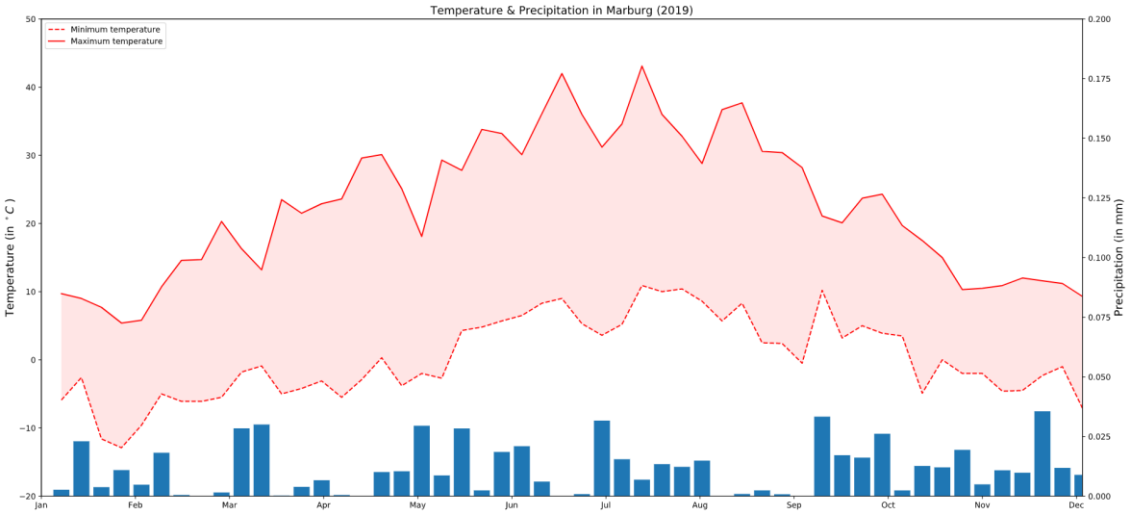


Figure 4: Weather data

The weather data in the form of temperature and precipitation are shown in figure 4. The x-axis shows the temporal course of the year in the form of months, while the y-axis shows temperature and precipitation. The temperature scale is shown on the left side and the precipitation scale on the right side of the figure X.

For a better understanding the temperature is given in the minimum and maximum temperature per day. The red area between the two lines represents the temperature reached during each day. While the temperature in winter fluctuates between -10 and 15 degrees, temperatures of up to 40 degrees are reached in summer.

In contrast to the temperatures, no trend can be determined for the precipitation values. However, it must be noted that the year 2019 was rather dry and there was comparatively little precipitation.

4. Advanced Visualization

4.1 Number of started trips per PLZ

The user interface of the developed program allows the user to view maps showing the number of trips per postcode area. Here he has the choice between starting points and end points of trips. He also has the possibility to choose a month of interest. In this section only the map for June is shown, as the proportional distribution among the postcode areas is the nearly identical in every month. The following map shows the distribution of the start trips according to their postal code. It is clear to see that most of the trips start in the center (35037, 35039) and only few in the peripheral areas (35094, 35043). This can be explained by significantly less fixed stations in the peripheral areas.

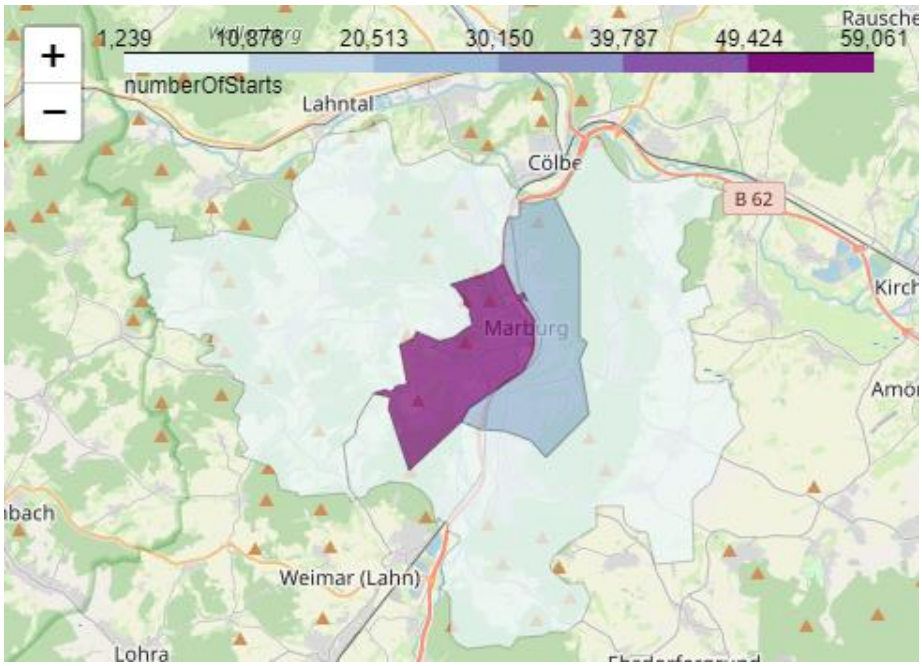


Figure 5: Number of trips started per postal code

The next graphics shows a map for the endpoint of trips according to its postal codes. As with the starting points, the relative distribution among the postcode areas is almost identical in each month. It can also be seen that almost as many trips start and end in the same area.

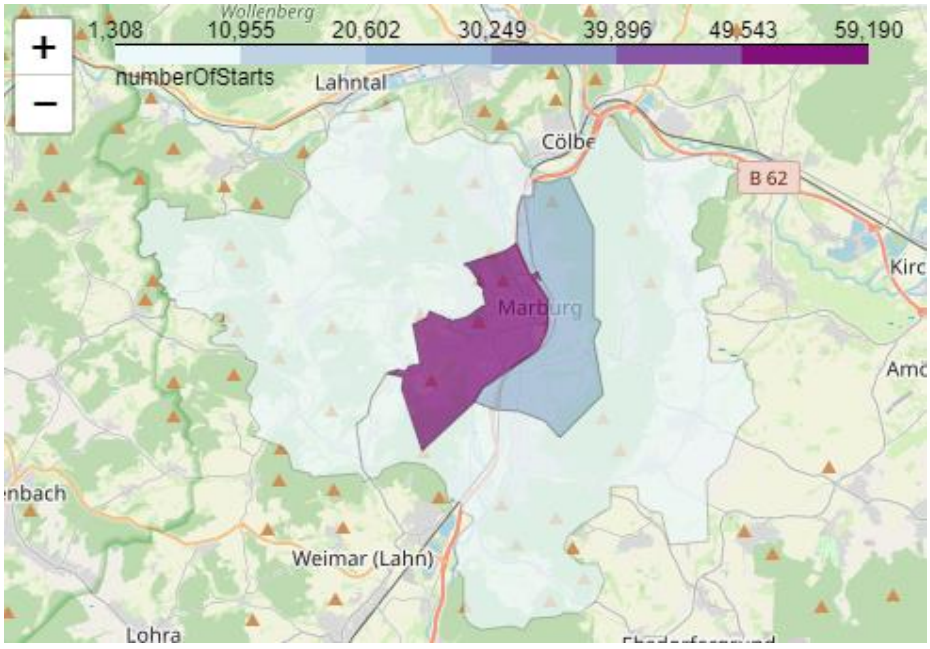


Figure 6: Number of trips ended per postal code

4.2 Number of Bikes at fixed Stations

Due to the in chapter 2.2 explained data frame, which contains the number of bikes available for all the fixed stations of nextbike on a minutely level for 2019, it is possible to visualize the number of bikes available on a folium map for every timestamp in 2019. Therefore, a color code and the radius of the as circle represented stations explains the number of bikes.

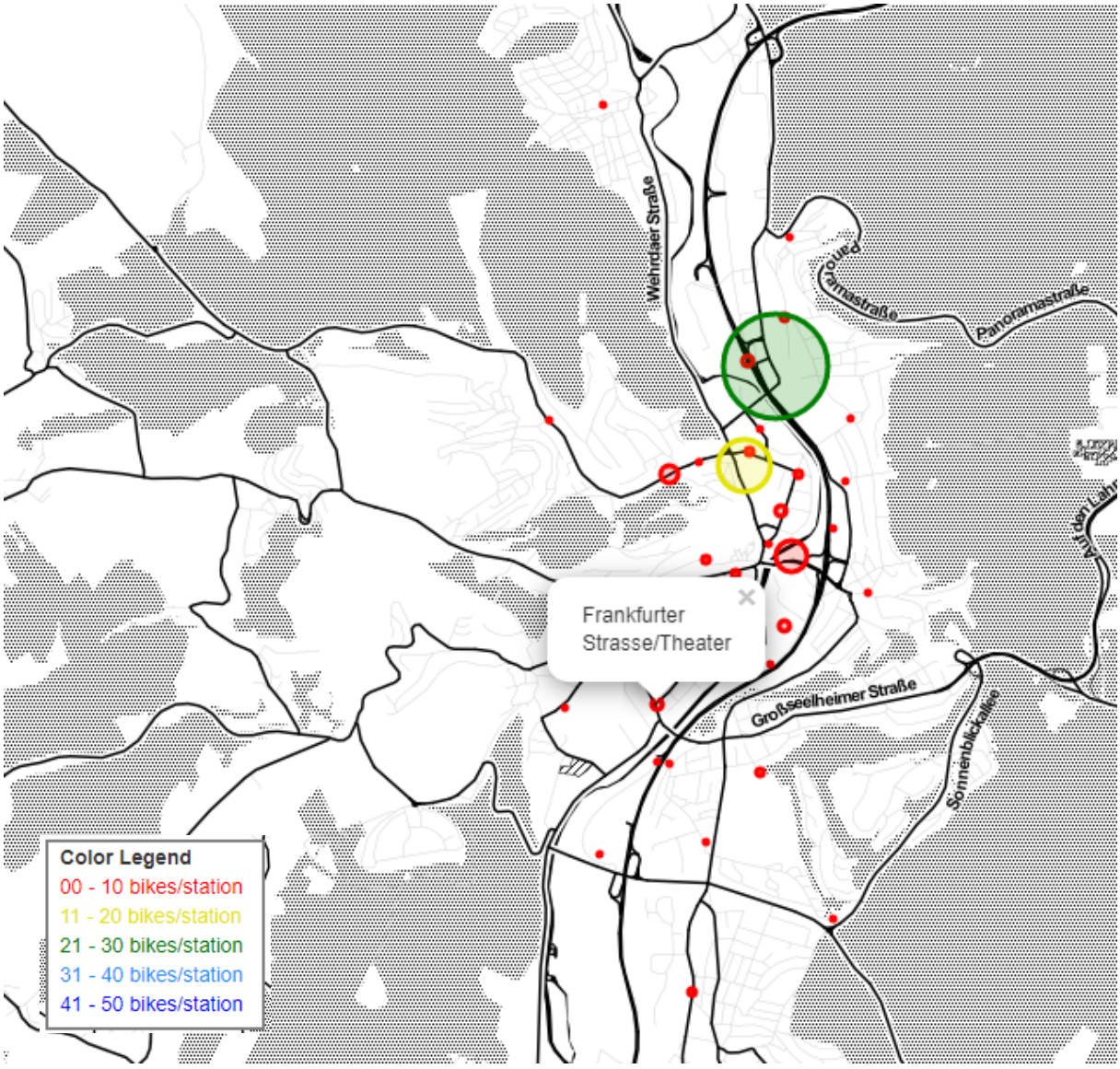


Figure 7: Number of bikes available per fixed station

The bigger the circle, the more bikes are available. Additionally, the color legend provides a categorized color scheme. The name of the station gets visible by clicking on the circles.

The following bar plot supports the visualization above if a simpler version without geographical representation is preferred.

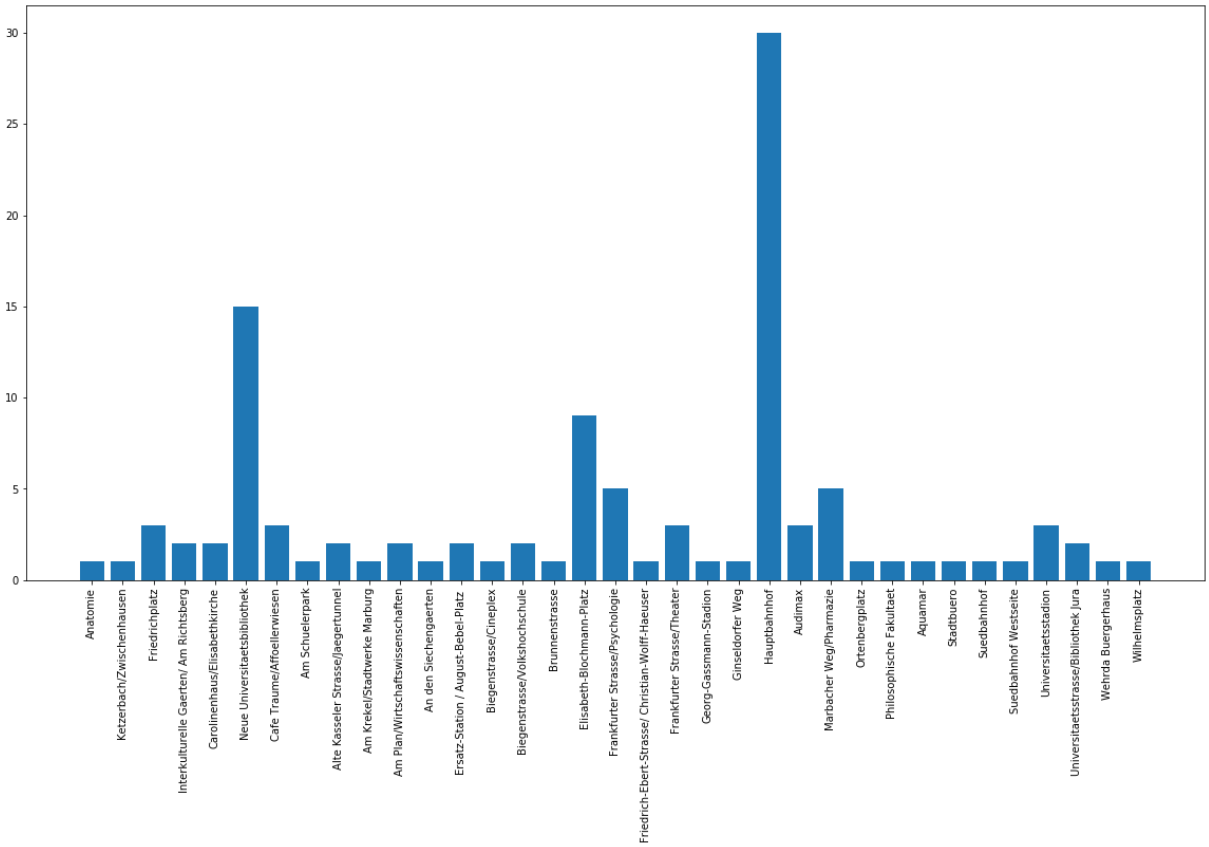


Figure 8: Number of bikes available per fixed station (BoxPlot)

4.3 Heatmap

This section handles the examination of data during a big event in Marburg. Trips in Marburg typically go from one station to another. Less than 3% of trips involve a start- or endpoint outside a station. This stems from the fact, that nextbike in Marburg does not offer a ‘flexzone’ (a ‘flexzone’ is a zone, where bikes can be returned at every corner). And every bike trip that ends outside a station costs 20€.

The event we investigate is the ‘Fridays for Future’ protest at the 2019-11-29 in Marburg. Nearly 5000 people attended the protest, which was coordinated on a global scale with many events in many different countries.

We plot the end location of trips five hours before the start and five hours after the start of the event. This ensures that the number of visualized trips is comparable and that all trips after the

event are captured, because the protest had no fixed duration. The black circles are the stations provided by nextbike.

The event took place at three stations distributed around Marburg. The event locations are marked with black info markers in the figure 9 and 10. The norther marker is the main station of Marburg, the marker in the middle of the map is the marketplace, and the marker at the south is at the ‘school-center’.

The main station of Marburg is a hotspot for nextbike. The station often has the highest number of available bikes (see an example in fig. 8) but before the start of the event the station had the highest number of incoming trips in Marburg and a high number of outgoing trips. This is different to the days leading up to the event. Normally the main station area is a place with lower incoming and higher outgoing bike traffic. We assume, the protestors used the bikes to travel to the event and the usual outgoing traffic continued.

The area around the marketplace is in the middle of multiple stations and featured an increased number of incoming trips in comparison to earlier days and to the other parts of the city.

The ‘school-center’ is a place with multiple schools in close vicinity. The pupils do not have free access to the bikes (unlike students of the University of Marburg) and have other means of travel for their normal route to school like public transportation or their own bikes. This could explain the low activity around the ‘school-center’ marker.

Another interesting observation is, that the areas around the University library and the ‘Philosophische Fakultät’ are areas with high incoming traffic at the first half of the day, but at the day of the event this peak shifted to the hours after the event. This could mean, that the University students (together with the pupils the two main groups attending the event) continued with their normal daily activity after the event with a few hours of lag.

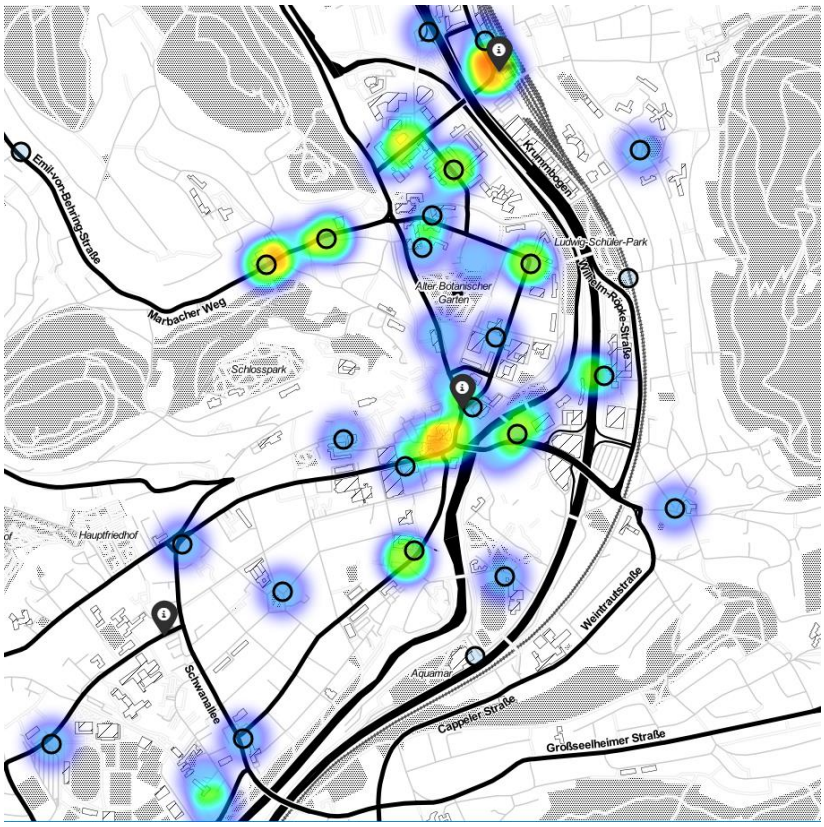


Figure 9: End location of trips five hours before the event

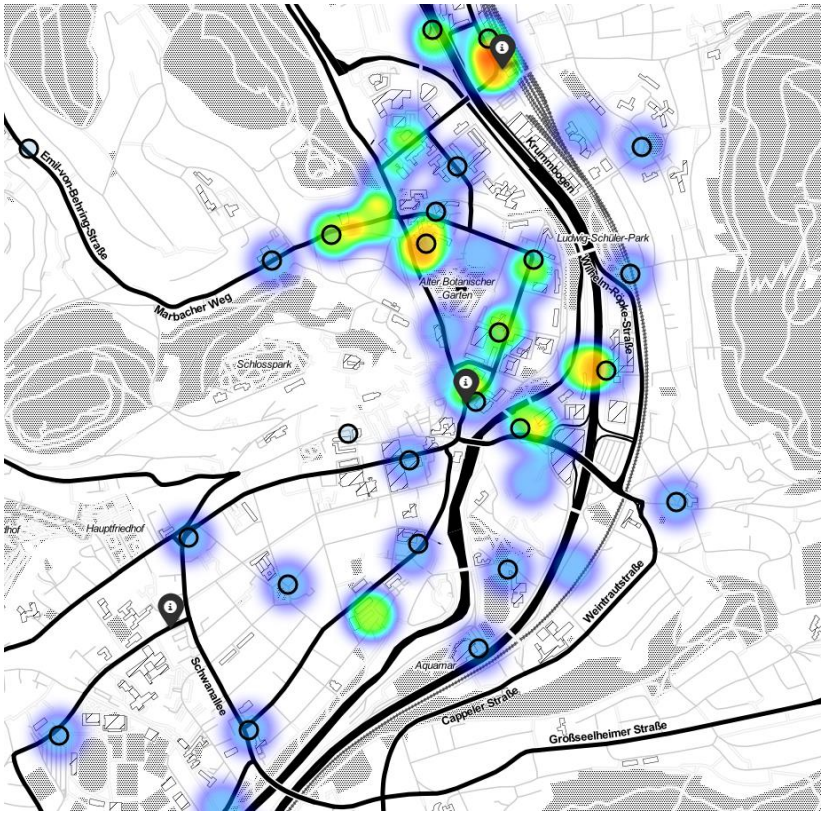


Figure 10: End location of trips five hours after the event

4.4 Normal Distribution

In a further step the distribution of the trip length is analyzed and compared with the normal distribution. To get more accurate results, the comparison is done monthly. As mentioned earlier, our data set does not include trips in the month of July. Therefore, this month is not included in the analysis.

Figure 11 illustrates the distribution of the trip length per month. The x-axis shows the length of the trip and the y-axis shows the probability that a trip has the length x. The calculated normal distribution to the data is shown in the red curve. It is calculated with the mean and the standard deviation of the data.

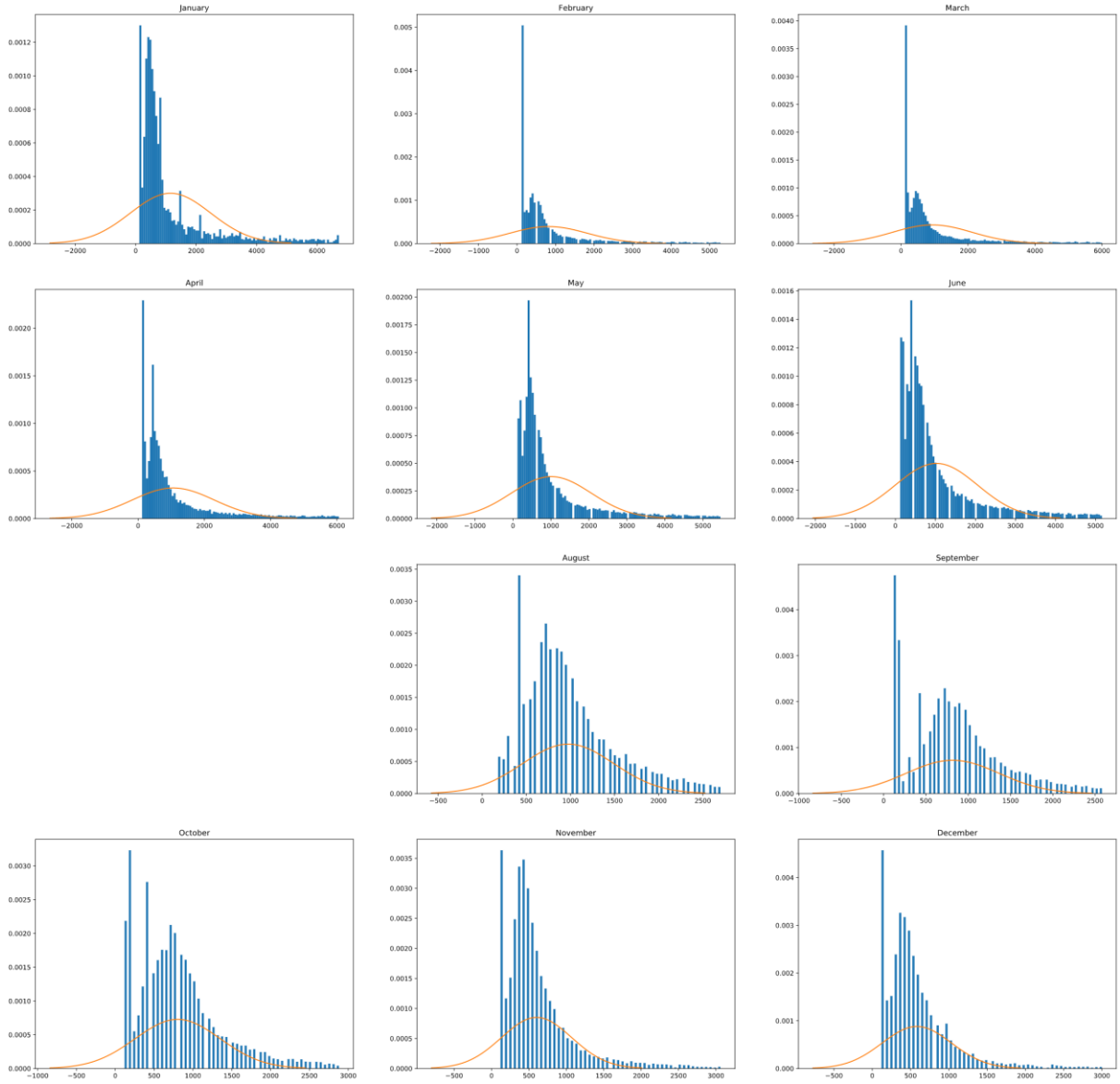


Figure 11: Distribution of trip length

As you can see in figure 11, no trip length is normally distributed. This has several reasons. First, the calculated normal distribution goes into the negative range, which is obviously not possible for trip lengths. On the other hand, it is clearly visible that most trips are very short, which always represent a peak.

However, clear differences can be seen between the months. Trips in the first half of the year (January - June) tend to be longer than in the second half. This leads to a higher standard deviation, which flattens out the normal distribution. The first 6 months can therefore be better represented by an exponential distribution.

The months August and September, on the other hand, can be represented much better by a normal distribution, if one ignores the negative trip lengths. This is mainly because there are much fewer long trips and that these are not as significant. Furthermore, the average trip length increases. Nevertheless, there are some peaks that stand out from the normal distribution.

If you go towards winter the distribution becomes exponential again. This is due to the fact that now most of the trips become much shorter again.

5. Predictive Analysis

In the prediction part, we predict three different target variables through regression. The target variables are the duration of a journey, the number of trips in a day, and whether a trip is heading to the university.

5.1 Hyperparameter Optimization

In this section a hyper parameter optimization used in two of the later following predictions is presented. This optimization refers to a random forest regression. The basic idea is to specify different values for each parameter. Then combine this parameter values and test them. Depending on the number of parameters and specified possible values, the number of possible combinations increases dramatically. To solve this problem a specific number of combinations is chosen randomly. Only these combinations are tested. For this a RandomizedSearchCV by scikit learn is used. It randomly selects 150 combinations and tests them for the random forest regressor. After testing all combinations, it delivers the best set of hyperparameters.

The number of trips hyperparameter optimization is available in the program. The optimization for the trip duration is to slow to perform at every start up, thus we performed the optimization once and implemented the best hyperparameters.

The following table shows the values of parameters for random combination:

Parameter	Values
n_estimators	np.linspace(start = 100, stop = 1500, num = 10)
max_features	‘auto’,’sqrt’
max_depth	np.linspace(10, 110, num = 11), None
min_samples_split	1,2,4
bootstrap	True,False

5.2 Trip Duration

With the first prediction we try to predict how long a started trip will be. Thus, we only use features that are available at the beginning of a trip. We consider all features that are available from the bike data and the current weather. In the feature selection process, we use a correlation matrix to get a first look about the prediction power of the features and the cross correlation between each feature. The features with the highest correlation are the starting place number (with one hot encoding), the month of the trip, the temperature at the start time, and the autocorrelation. Other features with low correlation but prediction power that we use for the regression are the minute of the trip, the hour of the trip, and whether a trip is during the weekend or not. The cross correlation between the selected features is acceptable with a maximum correlation of 0,33 between the autocorrelation feature and the monthly feature.

Trip duration also has autocorrelation as seen in the figure 12. All points for the lags between 1 and 100 lie above the 95% confident interval. The trip data is not a time series. Thus, we aggregate the average trip duration for each hour and use the hourly average trip duration of the hour before a trip starts. For the hours that had no trips or missing data, we use the mean of all hours as a replacement.

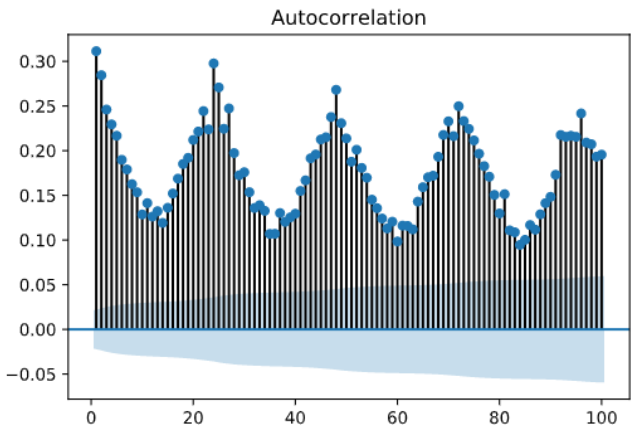


Figure 12: Autocorrelation

We use the random forest regressor implemented in the package scikit learn. The random forest regressor gives good speed and error performance in comparison to other tested regressions, like polynomial regressions customized for different hours of the day or different weekdays. The error metrics to validate the train and test performance of the model are the mean average error (MAE) and R^2. How the hyperparameter are chosen is detailed in the predictive analysis section. The model captures most of the variance of the train set with a performance of over 0.95 and a test set performance of 0.81 with better performance during the night hours than during the day. The MAE for the train set is 81 seconds and 180 seconds for the test set. The figure 13 shows that the model has difficulty to predict the long trip durations, which is understandable because the long trips are longer than a trip through Marburg takes and they often include brakes which are hard to predict.

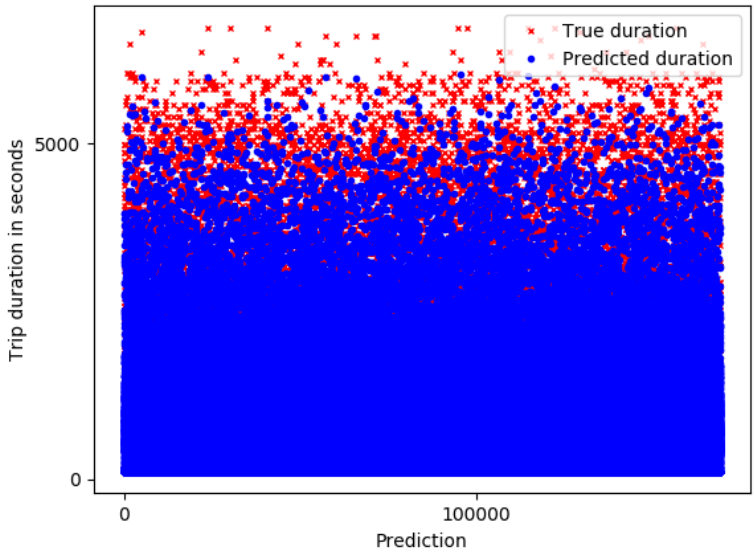


Figure 13: Model Evaluation of the Random Forest Regressor predicting Trip Duration

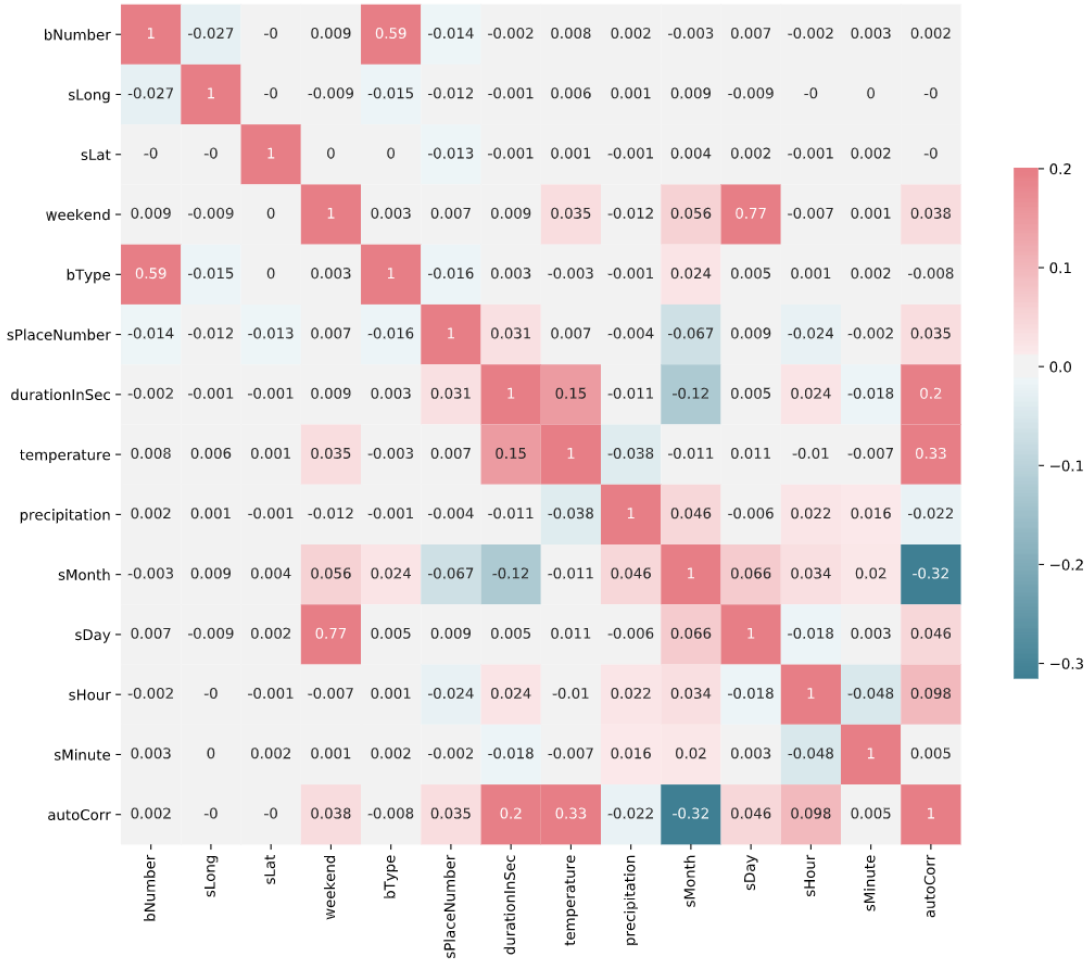


Figure 14: Correlation matrix for input features of trip duration

5.3 Number of Trips

In addition to the mandatory predictions, we decide to also predict the number of trips for the next day. For this an aggregated data frame (see chapter 2.2) is used. In addition to the existing features two additional features were created. First the feature “tripsLastDay” which is the number of trips of the previous day and second the feature “tripsOneWeekAgo” which is the number of trips seven days ago. For the selection of the features a correlations matrix was created. This matrix shows that the features ”temperatureAVG”, ”temperatureMin”, ”tripsLastDay” and “tripsOneWeekAgo” have a relative high correlation. Based on this values this features one the average temerature was selected for the prediction. Only one temerature value was selected to avoid redundancy. Furthermore, the features “precipitationAVG” and “dayOfWeek” were selected, although their correlation is low, because tests showed that these features improve the accuracy.

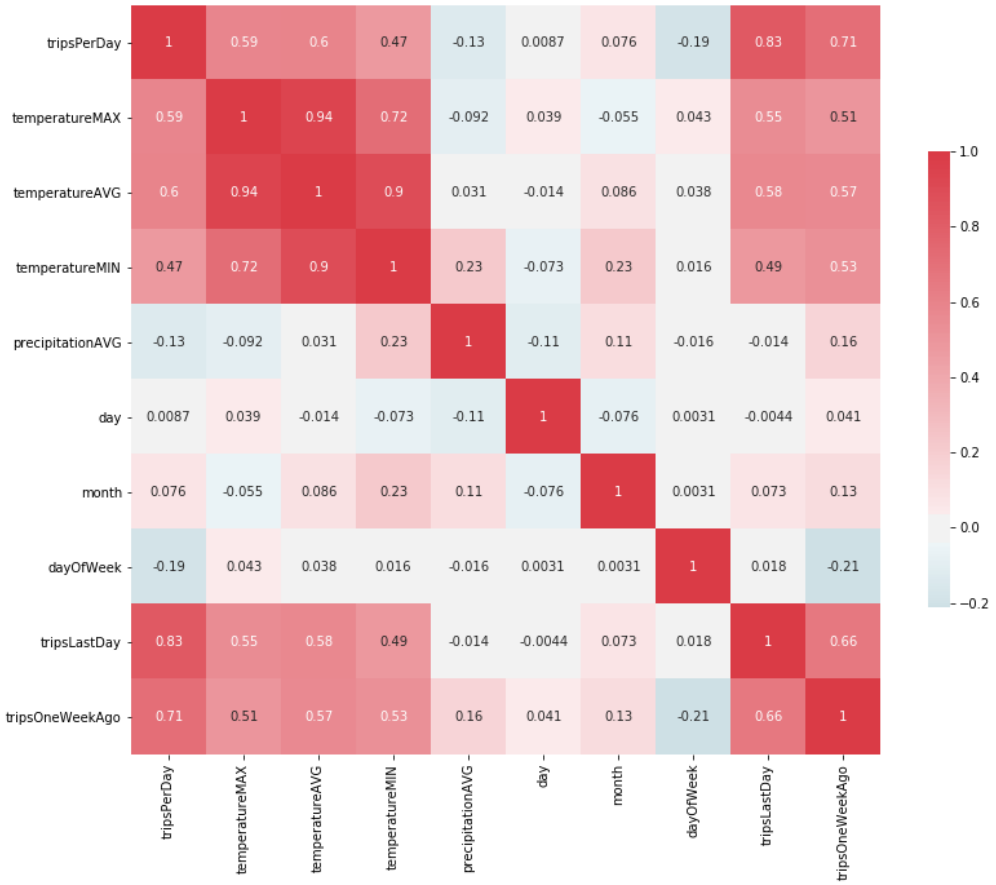


Figure 15: Correlation matrix for input features for number of trips prediction

As with the prediction of the journey duration, we also use the random forest regressor implemented in the package scikit learn, for the prediction of the number of trips per day. The random forest regressor gives good speed and error performance in comparison to other tested regressions. The hyper parameter optimization mentioned in 1.4 is used, too.

The evaluation of the model delivers the following results:

Train/Test	MAE	R^2
Train	110	0,96
Test	298	0,74

The following graph shows the real and the predicted number of trips per day. The prediction fits the real data quite well, but values above 3000 trips per day are more likely underestimated.

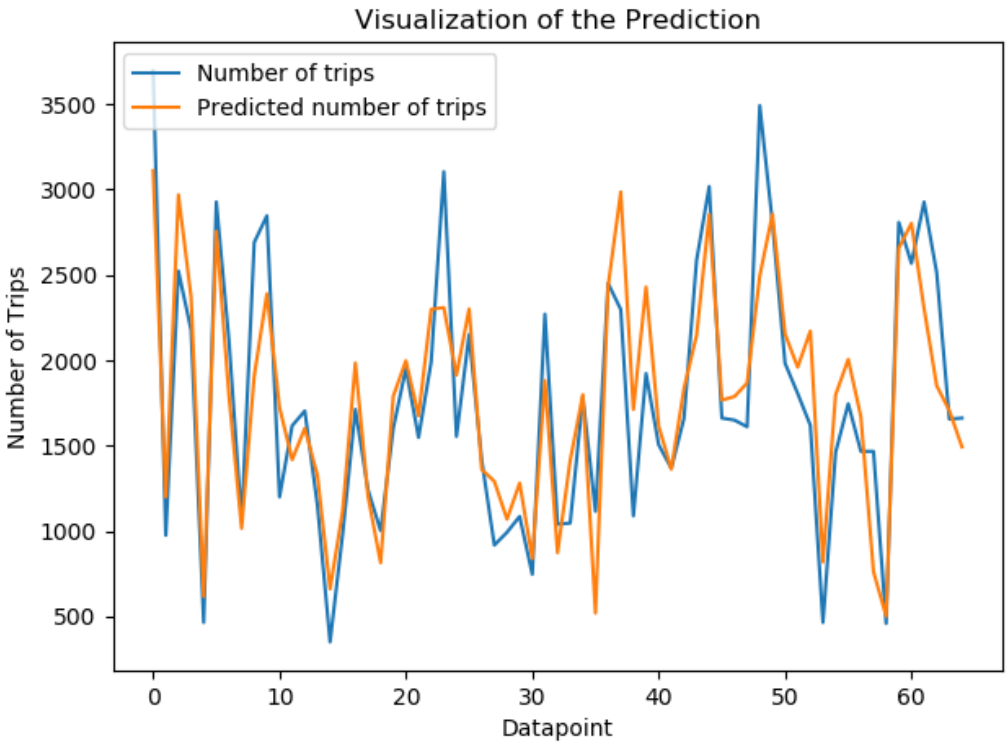


Figure 16: Evaluation of test set performance for number of trips

5.4 Trip Direction

Since the rental bikes in Marburg are mainly used by students, it is interesting to find out whether a trip is going to the university or not.

Generally, it must be noted that Marburg is not a campus university. The individual university buildings are distributed in Marburg, like the University of Cologne. To predict whether a trip goes to university or not, we first added the feature "tripToUniversity" to the existing dataset. This feature describes whether a trip goes to the university or not, encoded with 0 and 1. Therefore we looked at the different stations and marked the trips, which have as destination a station close to university buildings. We defined the following stations as close to university:

- Anatomie
- Neue Universitätsbibliothek
- Am Plan / Wirtschaftswissenschaften
- Biegenstraße / Volkshochschule
- Frankfurter Straße / Psychologie
- Audimax
- Marbacher Weg / Pharmazie
- Philosophische Fakultät
- Universitätsstadion
- Universitätsstraße / Bibliothek Jura
- Carolinenhaus / Elisabethkirche
- Elisabeth-Blochmann-Platz
- Am Schülerpark
- Aquamar

In the next step we developed different classification models (Linear Regression, Logistic Regression, KNN, SVM with various kernels) based on the training set and evaluated the performance based on the test set with a classification report.

Since the Accuracy was still too bad at first, we developed further features as "isTerm", which describes whether it is semester or semester break, and "isUniOpen", which describes whether the university is open or not. This feature refers to the time of day and the day of the week.

Based on the correlations, we have selected the most important features. These include "dayOfWeek", "isTerm", "isUniOpen", "month", "hour", "temperature", "precipitation" and

"sPlaceNumber", which is represented by dummy values. In figure 17, the correlations are shown.

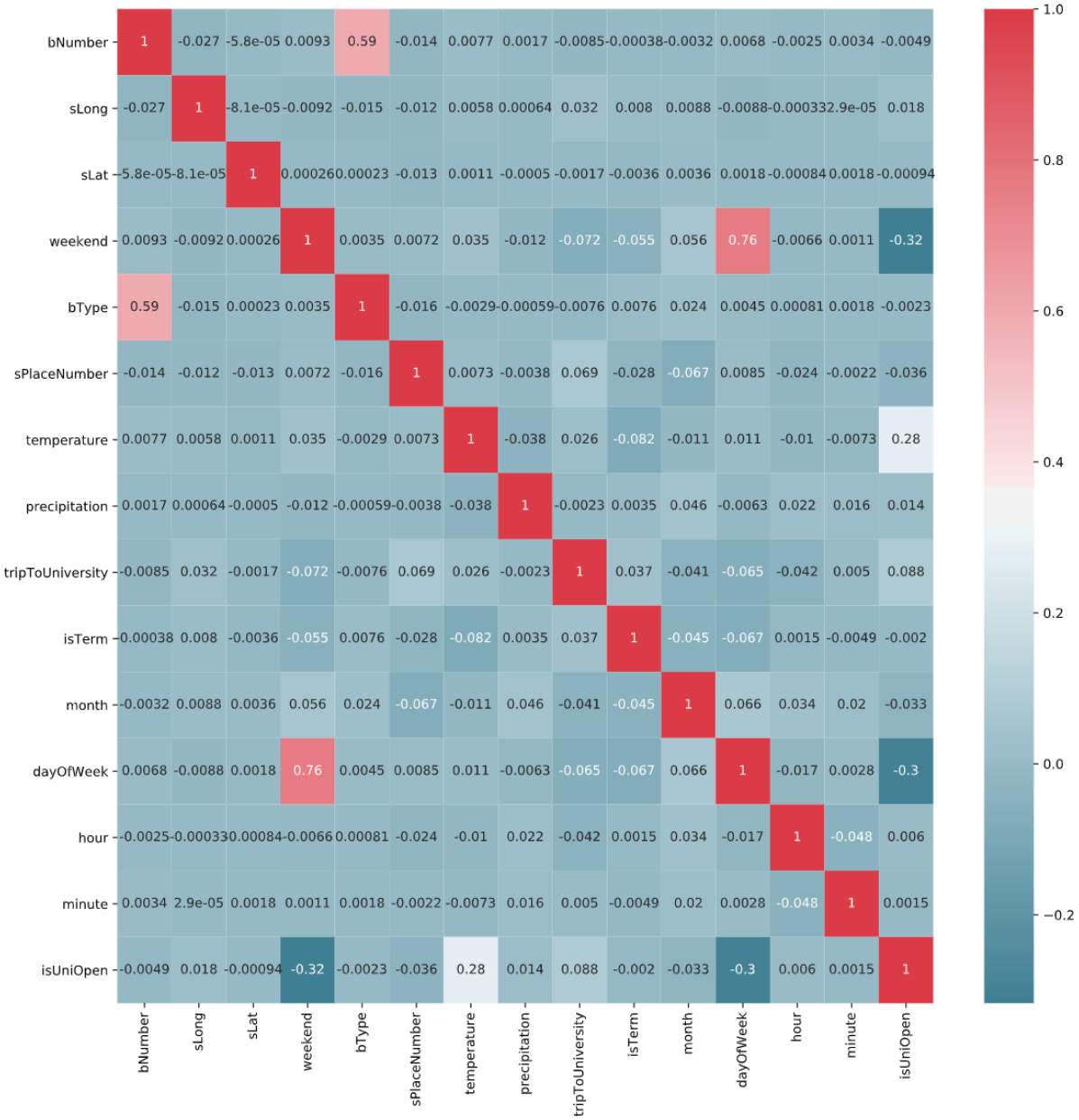


Figure 17: Correlation matrix for input features for trip direction

Using Principal Component Analysis (PCA), we have reduced the variance of features to 99% to increase the performance of our prediction. After adding the previously mentioned features, KNN proved to be the best model. The results are shown in table below.

	Precision	Recall
Trip does not go to university	0,77	0,78
Trip goes to university	0,7	0,68

The overall accuracy of our model is 0,74. We have also thought about whether it makes sense to optimize our model to a specific value. In medicine, for example, it can make sense to do so since you want to avoid giving sick people a wrong diagnosis. Since we do not attach such importance to this, we have decided to optimize the overall performance.

6. Performance Evaluation on Holdout-Set

As the evaluation of the predictive performance of our algorithms on the holdout set, which represents the month of July, returns quite unsatisfying results for at least two out of three algorithms, those outcomes and possible explanations are covered in an own chapter.

6.1 Trip Duration

For the trip duration, it must be recognized, that the mean absolute error on the holdout set (586 seconds) is more than three times higher than on the different test sets before. Moreover, the Random Forest Regressor prediction model seems to deliver worse results than just taking the average trip duration, as shown in the table below.

Method	MAE
Random Forest Regressor	586
Prediction by average (first 6 months)	500
Prediction by average (previous month)	506

Looking at the data for July, possible explanations could be found in the fact, that the standard deviation of the trip lengths decreases from June to August by 50 percent, as shown in figure 18.

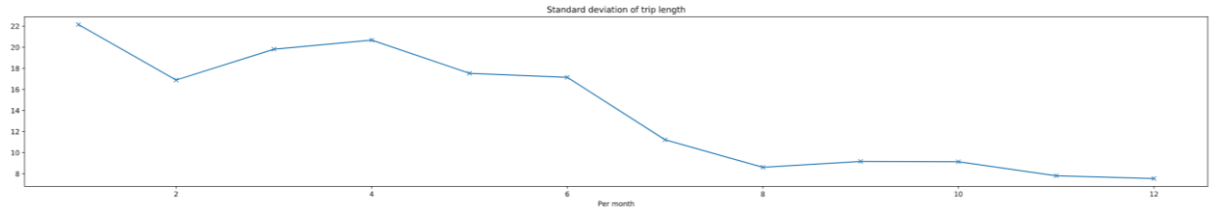


Figure 18: Standard Deviation of Trip Duration per Month

Furthermore, the median trip duration increases strongly from June to July, to decline again when going to August (figure 19).

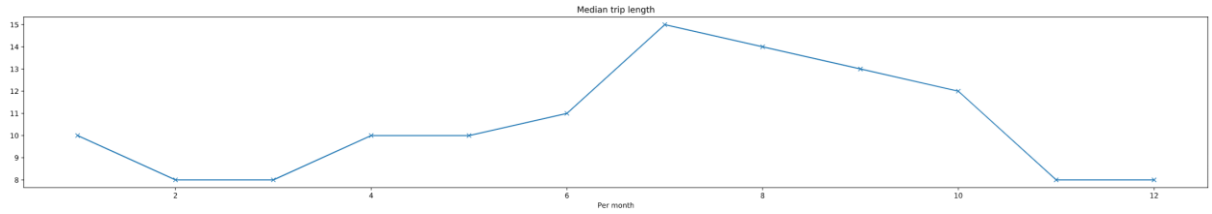


Figure 19: Median Trip Duration per Month

Analyzing the used Random Forest Regressor, it is hard to tell, if this meta estimator could be the cause of the bad holdout set performance, as it works like a black box where it is not possible to analyze the final model. It should be also taken into account, that the scikit learn train test split is distributed over all available month and therefore not the best choice, when testing the performance for a scenario, where the data for longer timespans is completely missing. However, other evaluated prediction algorithms have been evaluated and achieved similar error on the test set as the Random Forest Regressor model achieves on the holdout set. So maybe the difference between the test and holdout MAE could have been smaller using another algorithm, but that does not imply, that the holdout set performance would be better.

6.2 Number of Trips per Day

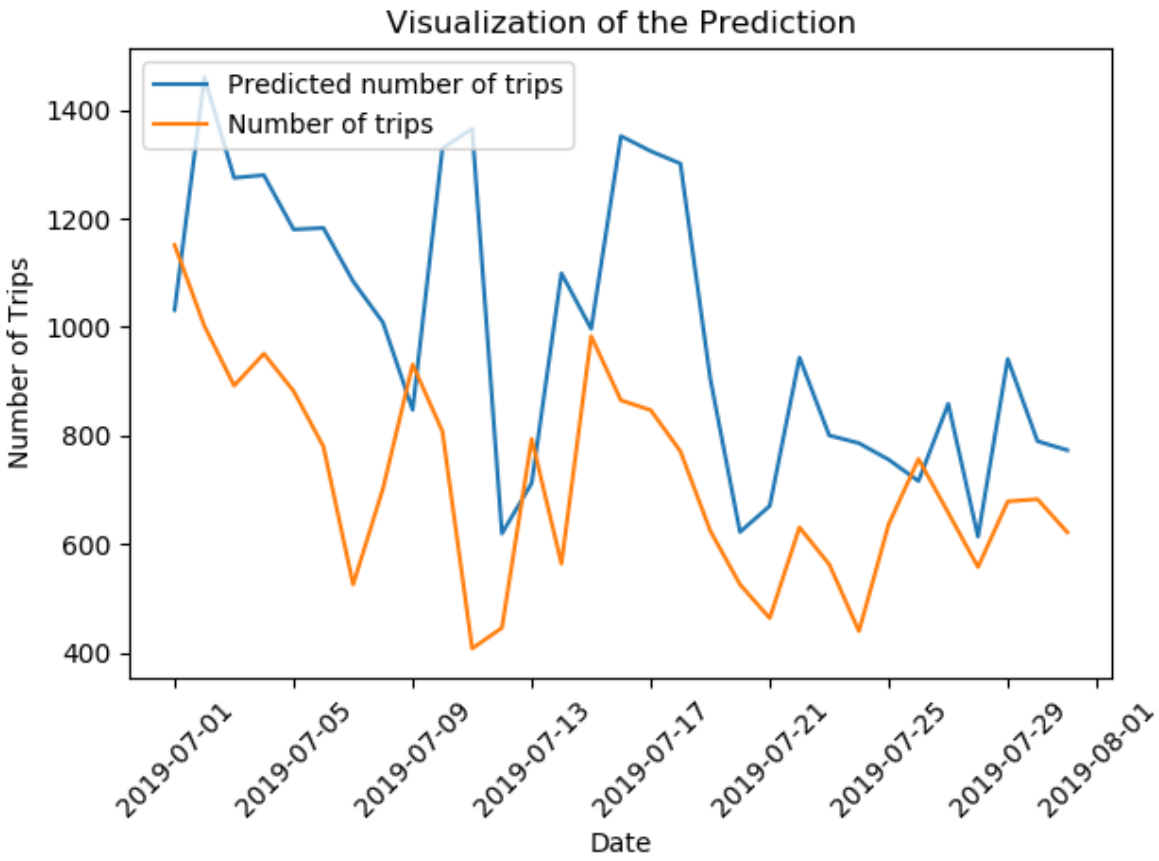


Figure 20: Holdout Set Prediction for Number of Trips per Day

The results of the evaluation of the algorithm on the holdout set differ from those of the test set. Although the MAE values differ only marginally, the real values are almost always exceeded on the holdout set.

To put the MAE values in relation to each other, two methods were tested, which use average values for prediction. The first method makes predictions based on the average value of the first 6 months, the second on the average value of the previous month. As you can see, the Random Forest Regressor delivers significantly better results.

Method	MAE
Random Forest Regressor	303
Prediction by average (first 6 months)	958
Prediction by average (previous month)	2207

The problems of the random forest regressor on the holdout set are probably related to the month tested. As you can see in figure 21, the number of trips per month in July (holdout set) is much lower than in other months.

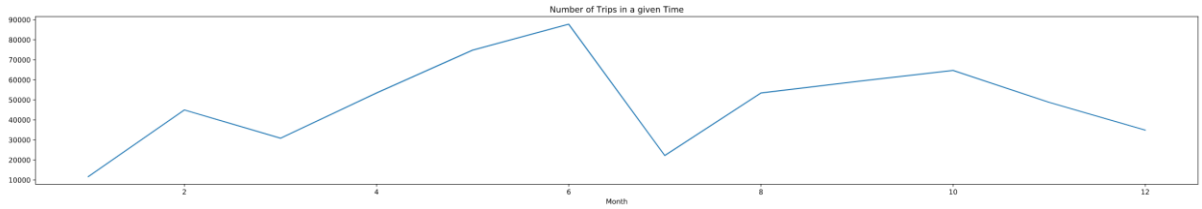


Figure 21: Number of Trips per Month

6.3 Trip Direction

Comparing the performance of the binary classification, if a trip goes to the university or not, from the test set to the holdout set, a big drop in accuracy can be seen. For the test set, the accuracy was about 74 percent. Now, for the holdout set, the accuracy is about 51 percent in July. This is, metaphorically speaking, just as accurate as tossing a coin for prediction. Finding concrete reasons for that bad accuracy is quite hard. Nevertheless, some assumptions can be made.

First, the task itself is complicated, as Marburg does not have a central university campus. The buildings belonging to the university are widely spread all over the city. In figure 22, the fixed stations are illustrated. Red markers symbolize fixed stations near to university buildings, blue markers symbolize the other fixed stations. In total, 14 of 35 fixed stations are related to the university, which is quite a high number. Furthermore, the semester break takes place in July, which influences the data quite hard, as Marburg is a traditional student city with 27.000 students and 77.000 citizens in total.

Second, it is possible that using the KNN Classifier is not the most suitable choice when predicting data for whole missing timespans. As its classification method is highly dependent on similar/near datapoints in regard of distance metrics, it could be very hard to predict datapoints in July, if all the data for July is missing. This is a difference to using the scikit learn train test split, which is distributed equal over all available month.

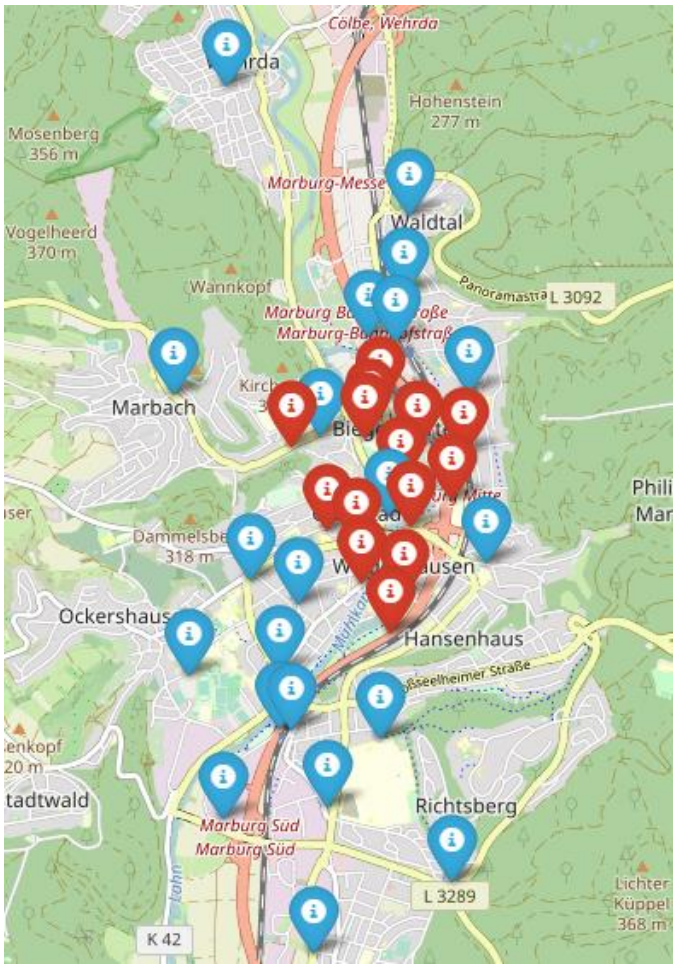


Figure 22: Fixed Stations in Marburg

Limitations

In summary, we have achieved acceptable results regarding the test set, what we were able to measure by our error values. But the results for predicting a whole month, which was missing in the data before, were significantly worse. Neither the prediction of the trip duration for July, nor the classification, if a trip goes towards the university or not, are use- or meaningful and sadly do not give any value. Their predictions are worse or equal to just taking the average over all values, as described in chapter 6. The business value is not given, except for the descriptive analytics and visualizations, which give insights to the data. A possible outlook would be the aggregation of more detailed data. Maybe an additional userID to the trip records could help to gather information about repetitive rides on a user level and improve the prediction performance.

Appendix