Data analysis of bike rental data from NextBike in Dortmund

Authors:
Emre Yildiz (7364253)
Erkin Altuntas (7323082)
Mostapha Ahaduch (7364245)
Marco Kurka (7369099)
Supervisor: Univ.-Prof. Dr. Wolfgang Ketter
Co-Supervisors: Philipp Kienscherf

June 10, 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.

Köln, den 10.06.2020

Executive Summary

Bike sharing offers have become more and more popular in recent years. Especially in big cities the offer is well accepted. This has led to an increased interest in deeper insights into how the rented bikes are actually used. This work has two objectives: First, the identification of factors influencing the trip duration. Subsequently, it is to be examined how precisely the trip duration can be predicted. The second goal is to analyze the direction of trips. In this case, trips are analyzed whether they are done towards a university station, away from a university station or without contact to a station near the university (of Dortmund). To reach these goals the booking data of the bike sharing provider NextBike for the city of Dortmund is analyzed by aggregating and processing the data. The analysis shows that trip duration is difficult to predict, as it varies greatly and is influenced by daily bookings. There are trends in the different months which show longer travel times in spring. Furthermore, the hourly trip duration is shorter at night than during the day.

However, by adding more data, such as weather data, and creating additional features as a result of feature engineering, the trip duration can be approximated. With the help of Grid Search, all common models are optimized regarding the setting of hyperparameters.

Random Forest had the best results with an R2 of 0.20 and a RMSE of 74.35 regarding trip duration. Regarding the prediction of a trip direction the best fitting model is a Random Forest as well. The resulting metrics of the Random Forest Classifier for the trip direction are a weighted average f1-score of 0.91, a macro average f1-score of 0.71 and an accuracy of 0.93 on the training data. On the test data the results are decreasing by about 10 percent in the macro average f1-score. The accuracy remains the same.

Contents

List of Figures

List of Tables

1 Problem description

1.1 Business Goal

The aim of this analysis is to evaluate the data of the bike sharing company NextBike for the city of Dortmund in such a way that a prediction can be made as accurately as possible for following aspects. First, how long will a trip with certain conditions take and second whether a trip will be a trip to the university of Dortmund. Furthermore, an understanding of the data should be created in order to make it clear which factors influence a journey. For this purpose, the given data as well as further data and features that has been added and self-generated will be evaluated.

1.2 Data Mining Goal

To determine the duration of trips for a given point in time it is crucial to identify which circumstances and conditions influence the figure size of the corresponding trip duration. Therefore, the technical goal of this work is to prepare the given data in such a way that evaluable objects are created from it. To achieve this, it is necessary to merge the appropriate postings, reduce them to necessary variables, filter out unusable lines and check the data quality. Furthermore, additional data should be used to further describe the trips. Finally, the dataset should be prepared in such a way that someone who is not familiar with it can use it for analyses.

2 Exploration and Description

The underlying dataset consists of two csv-files. The main dataset includes data of of the availability of bikes of the company NextBike. Furthermore, a dataset with weather data was added to further refine the analysis and prediction. The first step to get started with the analysis is to explore and clean the given raw data. Therefore, the meaning of the individual columns have to be determined. These are described in table 1.
Since the raw data only consists of bike availabilities or returns, but does not show complete trips, the data have to be mapped to each other to get the information of a whole trip. There are four different values in the column ***trip***: "first", "last", "start" and "end". At least two values are required to define whether the dataset belongs to the starting point or the end of the trip. This means that one trip is represented in two successively rows in the dataframe. One of the rows contains the values at the starting point like datetime or start position and the

other row contains the values at the ending point of the trip. Next, irrelevant columns are deleted. All bookings which have the values "first" or "last" for ***trip*** are dropped, since most of the trips in this dataframe have an unlikely long trip duration and a starting or end time at 00:01 and 23:59. Further, the columns ***p_spot, p_place_type, trip, p_uid, p_bikes, b_bike_type, p_bike*** are dropped because their relevance is not opposed to describe a trip. With the remaining columns a new dataset is created, which is the basis for the following analysis. Besides, hourly weather data is added to this dataframe.[1] The extracted and appended weather data contains the following attributes at the starting time of a trip: Temperature, precipitation in mm and precipitation as a boolean.

Column	Meaning
p_spot	True, if it is an official station and hour
p_place_type	The type of a bike-station
Datetime	Date and time of the start or end of a trip
b_number	Bike ID
trip	Values = ["first, last, start, end] defines if a trip starts or ends
p_uid	ID of the bike station / position
p_bikes	Number of available bikes at the postion
p_lat	Latitude coordinate of the position
p_ng	Longitude coordinate of the position
b_bike$_type$	Type of the used bike
p_name	Street or station name of the current position
p_number	ID of the postion / bike station
p_bike	unknown

Table 1: Columns of the raw data and the corresponding meaning

2.1 Feature Engineering

In addition to the existing attributes, there are added new features to the final data frame that contains a bike trip in each row. These additional features are useful for further visualization and prediction tasks.
The most important feature which has to be added first is the trip duration. This represents the label in the supervised machine learning algorithms which are used

[1]Source of the weather data is "Deutscher Wetterdienst", which can be found here: https://opendata.dwd.de/climate$_environment/CDC/observations_germany/climate/hourly/$

in the first of the two predictive tasks.
Moreover, the datetime at the start of a trip is splitted up in several feature columns, such as month, day and hour to use this important information in the predictive tasks. The columns weekday and weekend describe whether the trip starts at the weekend or within the week.
To better localize a trip the distances of the starting position to the University and to the Central Station of Dortmund are inserted. In addition to this, a KMeans++ algorithm is run to cluster the start position of each recorded trip to different areas within the city. The four integer values that are returned by KMeans++ divide the city in four areas where the trips starts. This feature is beneficial for the regression and classification of a trip. Last, features to determine whether a trip starts or ends at one of the five university station are added in order to determine the last added feature *tripLabel*. The trip label takes three values and represents whether a trips is going towards the university, away from the university or nowhere the university. This label is based on the binary variables ***awayFromUniversity*** and ***towardsUniversity*** which take the values 0 or 1 and determine whether the attribute is true or false.

3 Data Visualization

In order to obtain a better understanding of the data and to get a deeper insight, statistical evaluations are described and the results are explained in the following sections. First, the focus is on the trip duration of the respective trips, as the emphasis of this report is on them.

3.1 Trip Duration

count	207476.000000	count	179877.000000
mean	33.376010	mean	10.510465
std	83.413993	std	9.854895
min	2.000000	min	2.000000
25%	3.000000	25%	3.000000
50%	9.000000	50%	7.000000
75%	21.000000	75%	14.000000
max	1399.000000	max	48.000000

Figure 1: Comparison of the description of the trip duration of the dataset with and without outliers

As shown in figure 1, outliers make a clear difference when looking at the

trip duration. The difference can be assumed for day trips, which often have a duration of several hours. However, since these trips should be taken into account, outliers are not excluded. In the following figures all valid trips are considered. The resulting rentals are visualized to get an overview of the underlying data. The dataset is aggregated regarding the amount and/or duration of rentals over each month, weekday, day-type and hours. The key charts representing the trip duration will be explained here. A first impression is given in figure 2 indicating

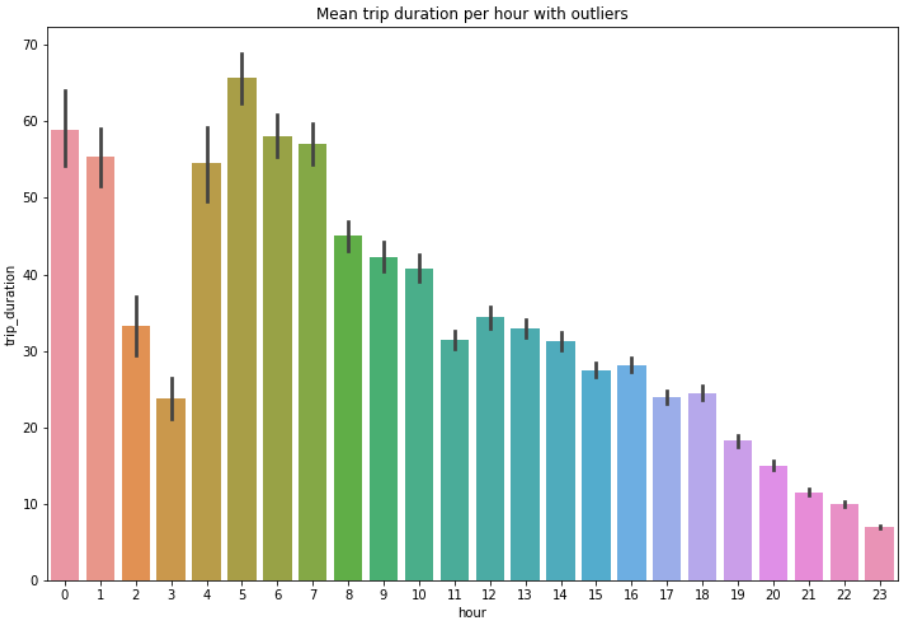

Figure 2: Mean trip duration per hour

a decline of the trip duration towards midnight. A possible explanation for this is the darkness which makes longer journeys unpleasant. A noticed phenomenon is that there are no trips beyond 12 o'clock at night, which increases this effect. An explanation for this has not been found. In general, an average trip lasts about 40 minutes (in the man mean) and the median is 8 minutes.

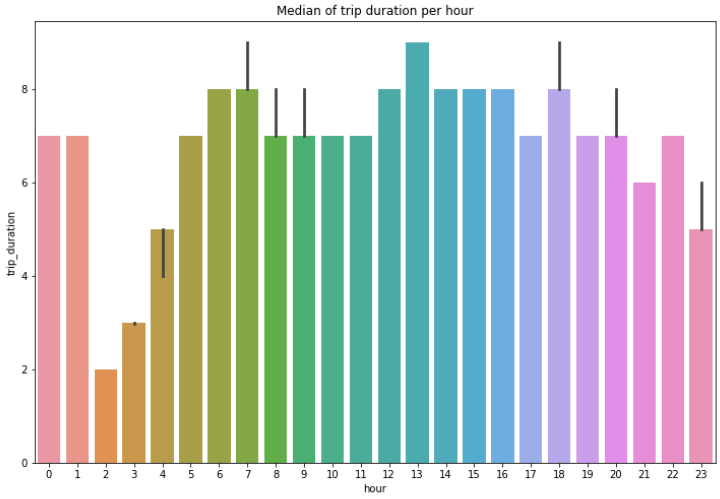

Figure 3: Median of trip duration per hour

In order to improve this distortion of very long journeys, the median is shown in Figure 3. Here one can see that the trips become less clearly shorter at the end of the day. An increase in travel times at 0 and 1 o'clock can also be seen here. In this case it can be assumed that these are caused by day trips that were booked in advance.

If one looks at the duration of the bicycle rentals on the weekdays, one notices a rather glaring duration. As shown in Figure 30 in the appendix, the average duration is slightly shorter on Mondays and longer on Saturdays. The longer rides on Saturdays could be explained by leisure riding.

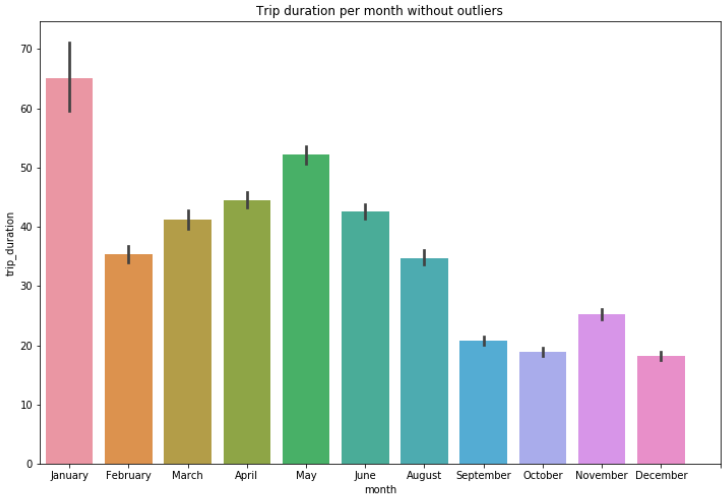

Figure 4: Mean trip duration per month

The mean trip duration of bicycle rentals increases until May, with January being

an exception. After that the average duration decreases again until winter. The big difference in January can be explained by a few trips in the dataset for that month.
When analyzing the trip duration further, one can see that the trips on weekends are on average only a few minutes longer than on weekdays, as shown in Figure 31 in the appendix.

3.2 Trip Distance

In the following, the distance travelled during a rental is analysed to gain a better understanding of the data. The trip distance is a value that measures the distance as the crow flies between the start station and the end station. This is important to note, as it is not the actual distance travelled that is measured.

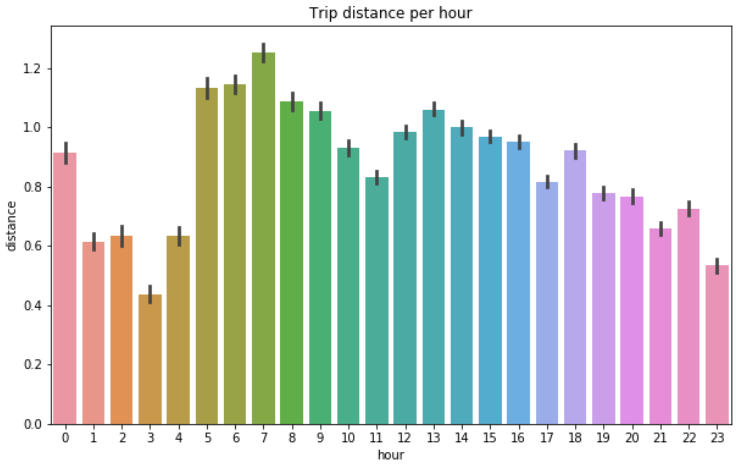

Figure 5: Mean trip distance per hour

Similar to the Trip Duration, there is an increase in the average distance until 7 o'clock. Since the distance travelled correlates with the trip duration, this also makes sense. From 7 o'clock on, the trip duration decreases until noon and then increases again at rush hour. From 18 o'clock on the average distance traveled decreases again. On average, customers drive about one kilometre when renting a bike.
Looking at the average distance on the individual days of the week, there is no significant difference on working days. On Fridays a slight increase and on Sundays a significant decrease of the average distance driven can be described. This also explains the slight difference in the average distance from weekdays compared to weekends. This can be seen in Figures 32 and 33 in the appendix

3.3 Amount of booked trips

In order to analyze the data set in more detail, it is necessary to take a closer look at the number of bookings. These give an insight into the actual demand for NextBike bicycles.

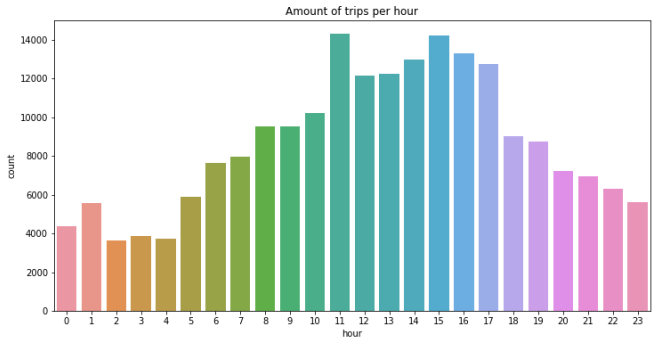

Figure 6: Amount of Bookings per hour

Looking at the number of bookings per hour, one can see that the demand is not higher during rush hour, which was to be expected. Instead, the demand is highest between 11 and 17 o'clock. This could be due to trips during lunch breaks or leisure trips by tourists. From 18 o'clock onwards, the number of bookings decreases.

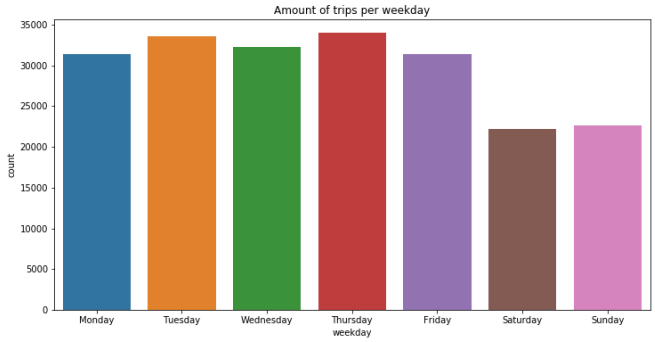

Figure 7: Amount of Bookings per weekday

On weekdays it can be seen that from Monday to Friday there are on average similar numbers of bookings. On weekends, however, significantly fewer bikes are rented. On average about 30% less than on weekdays. This suggests a more job-oriented use of NextBike, for exaple people who use the bike to get to and from work or use the bike during breaks.

3.4 Stations and Areas

The stations and areas of Dortmund are visualized within a map by using the folium-library. Therefore, the geographical data of the postal code areas of Dortmund are mapped to a folium-map in order to visualize the borders between the districts of Dortmund[2].

3.4.1 Demand per station

The map in figures 8 to 10 show the stations with the overall amount of rentals within the given dataset. The stations are shown as red circles which get bigger with their demand. As figure 14 indicates, the demand of the stations increases when moving towards the center of Dortmund. Further information can be accessed by clicking on the red circles. The name of the station aswell as the overall amount of rentals will be displayed, as it can be seen in figure 10.

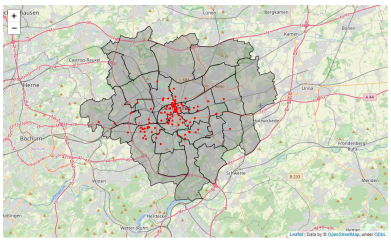

Figure 8: Stations of Dortmund

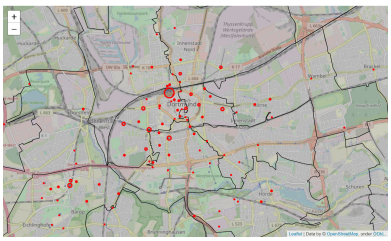

Figure 9: Stations of Dortmund

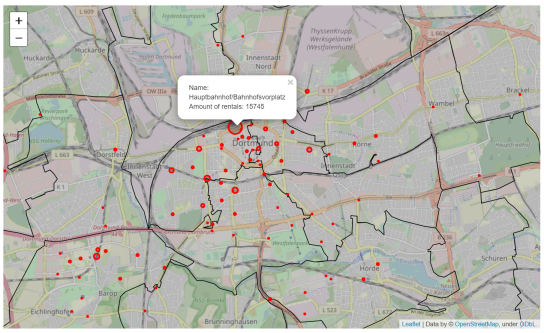

Figure 10: Details of stations of Dortmund

The following graphics 11 to 13 show the 20,000 trips [3] that have been used. Depending on the zoom level different clusters are formed. At the highest level of abstraction there are three different clusters, whereby it is clearly visible that the middle cluster dominates with almost 39,335 visited stations. By going further levels deeper, the clusters become finer, whereby it can be seen that the most used station is located in the center of Dortmund. However, it can also be observed

[2]Source of the geographical data: https://www.suche-postleitzahl.org/plz-karte-erstellens
[3]Map shows 40.000, because each trip has an end- and start-location

that certain stations are hardly ever visited, which is indicated by the yellow or green clusters.
The number of trips can be varied, because the graphic needs a lot of computing power, the trips are limited to 20.000 at first.

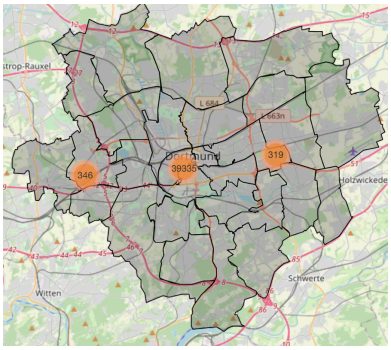

Figure 11: Top 20.000 used stations - Level 0

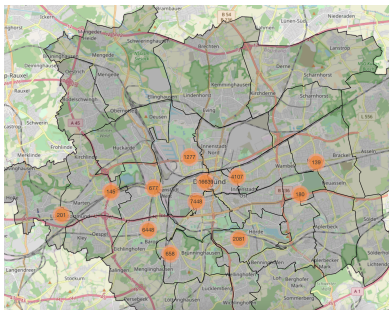

Figure 12: Top 20.000 used stations - Level 1

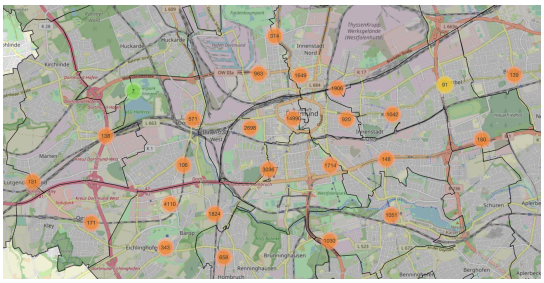

Figure 13: Top 20.000 used stations - Level 2

3.4.2 Amount of bikes per station

Similar to figures 8 to 10, figure 14 shows the stations as red circles. Instead of the amount of rentals at each station, the given map indicates the number of available bikes at each station for one point in time. To get this information, the number of available bikes is calculated for each station and for each hour of every day in the given dataset. For this example, the number of bikes on 31.12.2019 at 11 o‘ clock is displayed. Again, the circles get bigger with the described information and the information is shown by clicking on the stations. In contrast to the first map, it can be seen that the number in the periphery can be higher or similar to the center. This can be due to low demand in the periphery and higher demand (but also higher capacity) in the center.

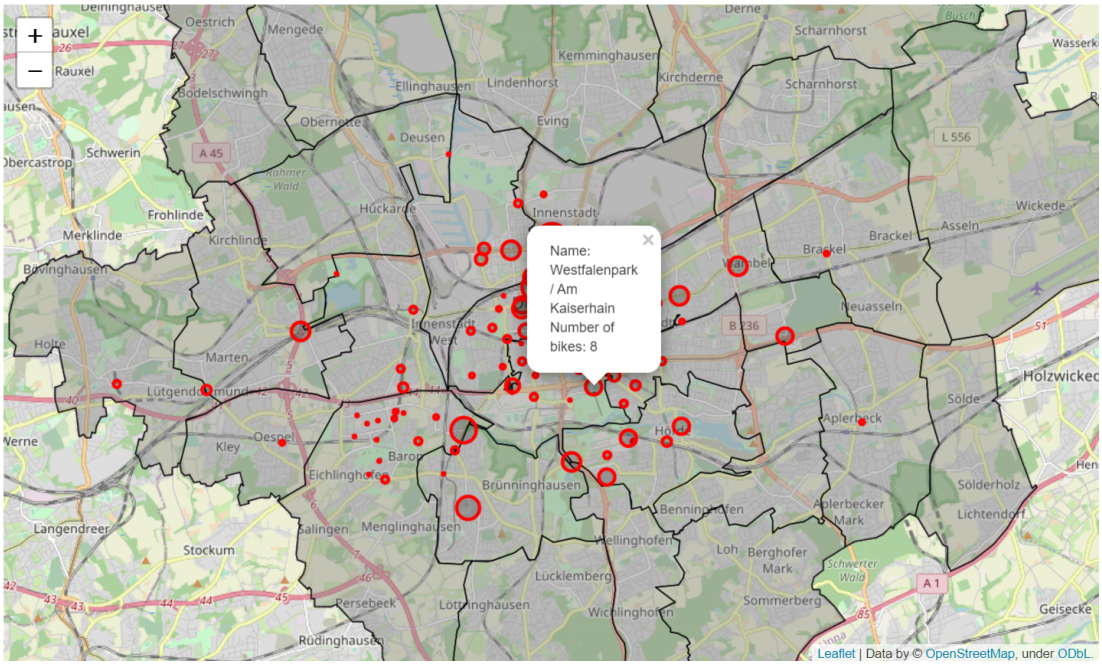

Figure 14: Available bikes per station for one point in time

3.4.3 Demand per postal code area

The next map indicates the amount of rentals per postal code area for the month with the highest demand. June is identified as the month with the highest amount of booked bikes, which can be explained by the aspect that people tend to go by bike in summer rather than in winter. As it can be seen in figure 15, the postal code areas are coloured regarding their demand. The dark greener the areas, the higher is he demand. The brown areas have no demand at all, because there are no stations located within these areas. Similar to the map before, the demand in the center of Dortmund is higher than in the periphery. In the center the demand goes up to more than 7500 areal bookings. However, there is one area in the South-West of Dortmund (44227) which shows a relatively high demand although it is not very central. The reason for this is that the stations of the University of Dortmund are located within this area. This indicates that students are an important customer group of NextBike which is also suspected by figures 16 to 19. Further information is shown when the mouse is hovered over the areas in the map as it is shown in the information box in figure 15. In addition to this, an interactive heatmap of the monthly course is created to see whether there are monthly differences in the demand of the postal code areas (figure 37). Different months can be selected by using the slider at the bottom left. However, no significant differences can be seen.

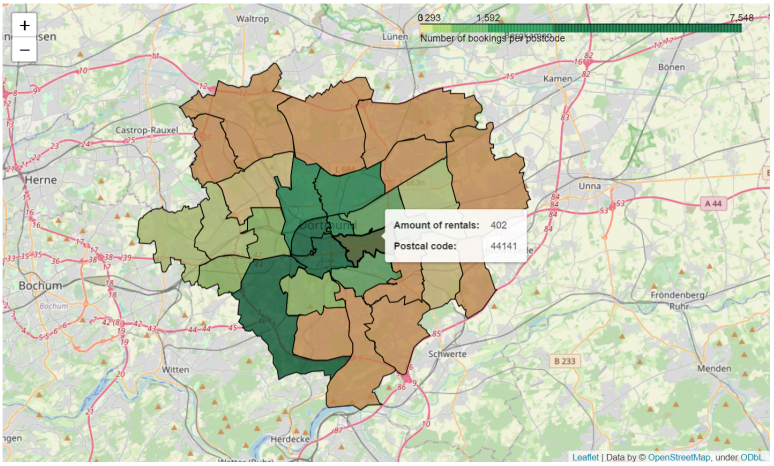

Figure 15: Areal demand in June

In order to map the stations to the postal code areas following procedure has to be done: The coordinates of the stations have to be converted to points (geometry objects of the library shapely) and compared to the polygons (also geometry objects) of the geodata/postal code areas in order to determine which station belongs to which postal code area. In addition to this, the (pandas-)dataframe of the given dataset has to be converted to a geodataframe to make the comparison and mapping of stations to postal code areas possible.

3.4.4 Daily demand course

The heatmaps in figures 16 to 19 visualize the intensity of rentals in different areas of Dortmund to different times of a day[4]. The intensity is shown by the heat, which is based on the rentals at each station, and different time periods of a day can be selected by using the slider at the bottom left. In detail, four time periods can be selected. These are from 5 to 9, 10 to 15, 16 to 20 and 21 to 4 oclock. As the figures show, the center of Dortmund shows a high demand over the whole day. However, from 10 to 15 there is also a high demand in the South-West of Dortmund which is probably due to the students which use the bikes to visit university since this is the rush hour of students. This demand decreases slightly from 16 to 20 until it reaches it minimum between 21 and 4 oclock.

[4]This heatmap is calculated in a different way than the heatmap of figure 37. While figures 18-21 visualize the heats based on the amount of rentals at each station, the figure 37 determines the intensity based on the demand in each postal code area. Since you need single points to work with heatmaps, the center of each area is calculated as the base of the heats.

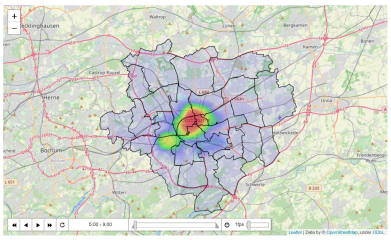

Figure 16: Daily heatmap 5 to 9

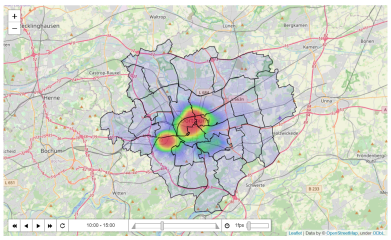

Figure 17: Daily heatmap 10 to 15

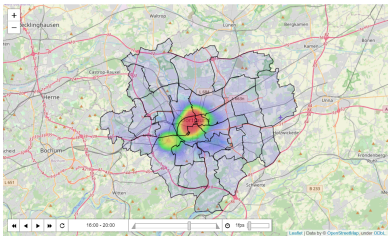

Figure 18: Daily heatmap 16 to 20

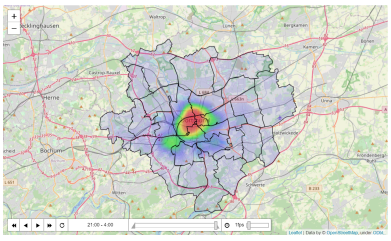

Figure 19: Daily heatmap 21 to 4

3.4.5 Demand at a given date and place

The starting point is the 20th of January in 2019. On this day, the derby between the two soccer rivals Dortmund and Schalke took place. It is decided to organize such an event, since data records for that date is available and furthermore the derby between the mentioned soccer teams is a classic and always fully booked.
The graphic can be abstracted into three parts. The first part is the marker. The marker in the picture below represents the stadium or place 'Signal Iduna Park' and is meant to mark the location of the stadium and to analyse possible connections later on. The second part corresponds to the heat map. The heat map is supposed to show the load, measured at the coordinates of the end positions in Dortmund on that day, to get a feeling of how the load distribution was on that day. The third component are the trips. The slightly bluish routes describe the trips that are made on that day.
As one can see, the highest load is in the center of Dortmund, although there is also a considerable load near the stadium. However, the trips do not show a route to a nearby (official) station, which suggests that only very few people use a bicycle to get to the stadium. The case that people have returned the bikes to an unofficial station, such as directly in front of the station, was not taken into account.
In summary, only a small fraction of visitors to the game have used NextBike bikes. This can also be justified, because most of them either use public transport, like bus and train or use a car.
The method which is responsible for creating the graphics can be changed at will, for example you can decide dynamically which date and which special event you

want to investigate, whereby you have to define the street for the "event". In a next step the road is transformed and the coordinates are obtained.

Figure 20: Demand at the Signal Iduna Park during the derby

4 Predictive Analytics

4.1 Trip duration

The aim of the third task is to predict the duration of one trip by knowing only the information about the starting point of it. Before starting with the prediction, the features for the prediction models have to be selected. A correlation matrix like it is shown in the appendix is always a good approach to learn how the features correlate with each other and which features are suitable for a well performing and powerful prediction model. By analysing the correlation matrix and by trying out several different features within the models, the following features are selected for further prediction models.

month, weekday, day_of_year, hour, minute, latitude_start, longitude_start, area_start, temperature, precipitation, distanceToUniversity, distanceToCentalStation

Some of them are given by the start information of the trip and the other ones are calculated or added as described in the feature engineering section.
The first prediction attempts are simple linear models. After fitting some ***Linear, Ridge and Lasso Regressions*** without setting any hyperparameters, it becomes clear that the complexity of these models is not sufficient for this prediction task. Figure 21 below compares three metrics of these predictive models. In the next step and with the hope of better results a Grid Search for hyperparameters is used for the Lasso and the Ridge Regression. In addition, the figure contains the regressions with defined hyperparameters and their metrics where

some of them become slightly better through the Grid Search. Furthermore the execution times for each regression are listed.

	Algorithm	RMSE	R2	MAE	Execution time (sec)	Description
0	Linear	80.519787	0.043494	39.654291	0.083323	
1	Lasso	80.605373	0.041460	39.360108	0.096022	
2	Ridge	80.519789	0.043494	39.654157	0.042007	
3	Ridge	80.519820	0.043493	39.652979	0.045681	Hyperparameters set after GridSearch
4	Lasso	80.519801	0.043494	39.653400	6.040680	Hyperparameters set after GridSearch

Figure 21: Results of Linear Models

Due to the low complexity of linear models the next approaches are made with ***Polynomial Regressions*** with a degree of three and four. The models get a little better than before. The performance on the training set is close to the performance on the test set. This means that the models aren't overfitted. In polynomial regressions the root mean squared error (RMSE) and the mean absolute error decrease and the R2 score increases. The execution time increases a lot in comparison to linear models. The following figure sums ups the metrics and the execution times for several polynomial regressions.

	Polynomial Regression with	Degree	R2	RMSE	MAE	Execution time (min)	Description
0	LinearRegression	3	0.087927	79.388925	38.314254	0.122343	LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
1	Linear Regression	4	0.101754	78.784868	38.547114	1.507270	LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
2	Ridge	3	0.085756	79.483344	38.294657	0.053588	Ridge(alpha=1.0, copy_X=True, fit_intercept=True, max_iter=None,\n normalize=False, random_state=None, solver='auto', tol=0.001)
3	Ridge	4	0.110136	78.416398	37.876588	2.770713	Ridge(alpha=1.0, copy_X=True, fit_intercept=True, max_iter=None,\n normalize=False, random_state=None, solver='auto', tol=0.001)
4	Ridge	4	0.109820	78.430360	37.859365	2.762630	Ridge(alpha=50, copy_X=False, fit_intercept=True, max_iter=40, normalize=False,\n random_state=None, solver='cholesky', tol=0.001)
5	Lasso	4	0.088976	79.343289	38.124820	6.450144	Lasso(alpha=1.0, copy_X=True, fit_intercept=True, max_iter=1000,\n normalize=False, positive=False, precompute=False, random_state=None,\n selection='cyclic', tol=0.0001, warm_start=False)
6	Lasso	4	0.071973	80.080274	38.485032	1.089877	Lasso(alpha=50, copy_X=False, fit_intercept=True, max_iter=40, normalize=False,\n positive=False, precompute=False, random_state=None, selection='cyclic',\n tol=0.0001, warm_start=False)

Figure 22: Results of Polynomial Regressions

The next predictive model which is taken into account is the ***Random Forest Regressor***. A random forest uses a number of decision trees on various subsamples of the dataset. One decision tree has a low bias and a high variance. The random forest averages the outputs of all used decision trees to improve the predictive accuracy and to avoid overfitting. So the random forest consists of many separate decision trees.
The first attempt to fit the random forest to the trip data leads to an overfitted predictive model since the performance on the test data is much worse than on the training data.

	Performance on	
	Training data	**Test data**
R2	0.866	0.140
RMSE	30.575	77.046
MAE	12.520	33.103

Table 2: Model performance on training and test data

The complexity of the model has to be decreased to avoid an overfitted Random Forest. Therefore some hyperparameters regarding to the properties of the used decision trees and the splitting of nodes in one decision tree are adjusted. This leads to a model with a lower variance. The R2 score increases, while the RMSE decreases. Here, a Grid Search is used again to identify the optimal hyperparameters and to get an even better and well performing model. The following figure shows the process how the metrics are improved with the different optimization steps. The last two rows belong to the predictions where the logarithm of the trip duration are used to fit the random forest. The idea is to reduce the effect of the trips with an unlikely long trip duration. The values have to be exponentiated after the prediction to get correct values in minutes for the durations. This method has the effect that on the one hand the MAE decreases even more, but on the other hand the RMSE increases again due to the usage of the logarithm of the durations and the reduction of the influence of long trip durations.

	Model	R^2	RMSE	MAE	Execution time in minutes	Description
0	RF	0.140957	77.046445	33.103462	1.494382	first try rfr, overfitted
1	RF	0.176323	75.443805	33.456218	0.415689	adjust hyperparameters to avoid overfitting
2	RF	0.20002	74.350667	32.003231	13.078619	hyperparameters with RandomizedSearch adjusted
3	RF	-	81.119785	26.244243	0.412784	using log of trip duration/ exp after prediction
4	RF	-	79.952002	25.575316	13.162461	using log of trip duration/ exp after prediction

Figure 23: Results of Random Forest Regressor

The last regressor which is used in this project is the ***Support Vector Regressor***. During this approach it becomes quickly clear that a SVR is not purposeful because the SVR is hard to optimize. Though the first SVR has an even stronger MAE than the Random Forest Regressor before, but the R2 is negative, which indicates a bad prediction model. The second attempt leads to a positive R2-score, but the RMSE and the MAE get much worse. Since a Grid Search on the hyperparameters for a SVR needs a lot of computational power, here is just a Randomized Search used to get the hyperparameters. Setting the returned hyperparameters from a Randomized Search also hasn't a positive effect on the metrics (last row in the figure below). The metrics become even worse.

	Name	R2	RMSE	MAE	Execution time (min)	Description
0	SVR	-0.080079	87.550234	29.554568	1.579823	SVR(C=1.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1, gamma='scale',\n kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)
1	SVR	0.019081	83.434611	40.239168	0.689221	SVR(C=100, cache_size=200, coef0=0.0, degree=3, epsilon=30, gamma='auto',\n kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)
2	SVR	-0.420239	100.394578	71.816326	0.027797	SVR(C=40, cache_size=200, coef0=0.0, degree=1, epsilon=30, gamma='auto',\n kernel='poly', max_iter=500, shrinking=True, tol=0.001, verbose=True)

Figure 24: Results of Support Vector Regressor

Because of the combination of all named metrics especially the RMSE with the MAE the ***Random Forest Regressor*** is chosen as the final prediction model for this task.
All in all, the Random Forest is not a really good predictive model, too. The model is not that powerful as it is possible in other predictive tasks. This also applies to the three named metrics. If they are put in relation with the standard deviation and the mean of the trip durations the values of the metrics aren't that impressive. This has many different reasons.
The trip duration of one bike trip is hard to predict with just knowing the information at the starting point of the trip. The length of a trip depends on several factors like for example the physical characteristics of the driver. How fast can the driver drive? How often does he have to stop driving for example due to traffic or traffic lights? Does the driver know the city or is he new in city or even a tourist? Sometimes the driver may not have a specific destination. Another problem in predicting the trip duration is that it is possible to rent a bike for over hours or a whole day. This counts as a full trip over the whole day with extreme long trip durations.
These are some reasons why it is really challenging to predict the duration of a bike trip.

4.2 University trips

Another aim of the prediction task is to classify whether a trip will be made towards the university of Dortmund by using only information about the start of a trip. The goal is to classify the attribute tripLabel. Again, before starting with the prediction, the features for the prediction models have to be selected. This procedure is done in the same style as the prediction of a trip duration. Finally, following attributes are chosen for this prediction task:

month, weekend, hour, area_start, distanceToUniversity

4.2.1 Logistic Regression

The first prediction attempts are binary logistic regressions. The binary logistic regression is used to predict the binary outcome of the variables ***awayFromUni-***

versity and ***towardsUniversity***. As the very first attempt, two separate logistic models are created to predict both variables individually. The multinomial logistic regression is build in order to predict the attribute ***tripLabel***. Similar to the first prediction task, a grid search is done to optimize the hyperparameters. This is done for both the binary and multionomial regressor to get a better comparison of both models. Following figure shows the results:

	Algorithm	Precision	Recall	F1score	Execution time (sec)	Description
0	Binary Logistic Regression	0.999299	0.999293	0.999294	0.752478	Predicts awayFromUniversity (is complementary to 2nd model)
1	Binary Logistic Regression	0.890065	0.918320	0.887933	0.250331	Predicts towardsUniversity (is complementary to 1st model)
2	Multinomial Logistic Regression	0.898683	0.913886	0.901719	8.989423	Predicts tripLabel
3	Binary Logistic Regression	0.996802	0.996674	0.996704	0.537564	Optimized hyperparameters of model in index 0
4	Binary Logistic Regression	0.891143	0.919107	0.889501	0.259306	Optimized hyperparameters of model in index 1

Figure 25: Results of Logistic Regressions

All models show a good overall performance as the average precision, recall and f1-score show in figure 25. However, when looking at the classification report of the binary classifier of the attribute ***towardsUniversity*** in figure 26, it becomes clear that the performance of predicting the value 1 is very low (f1-score = 0.12). The f1-score is an important measure since the classes are unevenly distributed. The value 1 occurs only 5031 times, while 0 occurs 57212 times. The complementary logistic regressor of the variable ***awayFromUniversity*** has a much better performance. This is because the probability that a trip which has started at the university will be done away from the university is very high. That classification report can be seen in the appendix in figure 38.

	precision	recall	f1-score	support
0	0.92	0.99	0.96	57170
1	0.53	0.07	0.12	5073
accuracy			0.92	62243
macro avg	0.73	0.53	0.54	62243
weighted avg	0.89	0.92	0.89	62243

Figure 26: Classification report of Logistic Regression for towardsUniversity

When looking at the classification report of the multinomial classifier in figure 27, similar problems as described before in the binary classifier of the attribute ***towardsUniversity***, but with a better overall performance.

	precision	recall	f1-score	support
awayFromUniversity	0.63	0.86	0.73	3072
noUniversityRide	0.96	0.98	0.97	54007
towardsUniversity	0.47	0.22	0.30	5164
accuracy			0.91	62243
macro avg	0.68	0.69	0.67	62243
weighted avg	0.90	0.91	0.90	62243

Figure 27: Classification report of Multionomial Logistic Regression

4.2.2 Support Vector Machine

The next predictive approach is the ***Support Vector Machine***. Therefore, a Support Vector Classifier model is build for both binary attributes in the same way as it is done in the Logistic Regression. Moreover, for each model a linear- and a rbf-kernel is approached. The results are shown in following figure 28.

	Algorithm	Precision	Recall	F1score	Execution time (sec)	Description
0	SVM with linear kernel	0.840158	0.916601	0.876716	105.369310	Predicts awayFromUniversity (is complementary to 2nd model)
1	SVM with linear kernel	0.843636	0.918497	0.879477	97.975079	Predicts towardsUniversity (is complementary to 1st model)
2	SVM with rbf kernel	0.840158	0.916601	0.876716	297.771291	Predicts awayFromUniversity (is complementary to 4th model)
3	SVM with rbf kernel	0.905629	0.921148	0.888784	592.105684	Predicts towardsUniversity (is complementary to 3rd model)

Figure 28: Results of SVM Classifiers

However, similar to the Logistic Regression models the average scores are that high, because the overpopulated classes manipulate the results. In addition to the problem of the logistic regressor, the attribute ***awayFromUniversity*** shows also a very low performance when classifying the value 1. The classification reports can be seen in the figures 39 to 42 in the appendix. As a next step, hyperparameters are tried to optimize with Grid Search, but due to very long execution time and poor initial performance this step is cancelled.

4.2.3 Random Forest Classifier

Last, a ***Random Forest Classifier*** is used to get a better comparison of different classification models. Again, models for the attributes ***awayFromUniversity*** and ***towardsUniversity*** aswell as for the attribute ***tripLabel*** are created. Following figure shows the first results of the different attempts.

	Algorithm	Precision	Recall	F1score	Execution time (sec)	Description
0	Random Forrest	0.915971	0.926144	0.918487	16.526235	Predicts tripLabel
1	Random Forrest	0.918142	0.929052	0.914644	131.005115	Optimized hyperparameters of model in index 0
2	Random Forrest	1.000000	1.000000	1.000000	5.359210	Predicts awayFromUniversity (complement)
3	Random Forrest	1.000000	1.000000	1.000000	18.891498	Optimized hyperparameters of model in index 2
4	Random Forrest	0.914250	0.925341	0.917834	13.856886	Predicts towardsUniversity (complement)
5	Random Forrest	0.915007	0.928859	0.912970	239.189682	Optimized hyperparameters of model in index 4

Figure 29: Results of Random Forest Classifiers

The first and second model predict the attribute ***tripLabel***. Additionally, the hyperparameters are optimized in the second one by doing a Randomized Search. The other models show the binary Random Forest Classifiers for the binary attributes (with and without optimized hyperparameters). As one can see, all models show a high performance, but when looking at the classification reports of the binary classifiers of the binary models one sees that the same problem as before occurs for the attribute ***towardsUniversity***, as figure 30 shows. Nevertheless, the values increase in comparison to the Logistic Regressor and the SVM. The classification report of the binary classifier for the attribute ***awayFromUniversity*** can be seen in the appendix in figure 43.

	precision	recall	f1-score	support
0	0.94	0.99	0.96	57201
1	0.67	0.24	0.35	5042
accuracy			0.93	62243
macro avg	0.80	0.61	0.66	62243
weighted avg	0.92	0.93	0.91	62243

Figure 30: Classification report of Random Forest Classifier for towardsUniversity

Finally, the Random Forest Classifier with optimized hyperparameters which classifies the attribute ***tripLabel*** (index 1 in figure 29) is chosen, because of following reasons. The binary classification models are not chosen for following reasons: First, the logistic regression models are not able to predict the value 1 of the attribute ***towardsUniversity***. This problem decreases when using a binaray Random Forest Classifier. However, it is more convenient to use one model. In addition to this, the execution time of the binary Random Forest Classifier increases when predicting the attribute ***towardsUniversity*** with optimized hyperparameters.

The Random Forest is chosen over the multinomial logistic regression, because

it shows a high overall performance with value of higher than 0.9 for the average precision, recall and f1-score. Moreover, when comparing its classification report to the report of the multionomial logistic regressor (see section 3.2.1), it can be seen that the f1-scores, accuracy and averaged precision and recall are higher in the case of the Random Forest. The classification report of the final model is shown in figure 31. The model is validated by an average cross validation score of 0.92 for ten groups of samples.

	precision	recall	f1-score	support
awayFromUniversity	0.70	0.86	0.77	3142
noUniversityRide	0.95	1.00	0.97	53964
towardsUniversity	0.69	0.26	0.37	5137
accuracy			0.93	62243
macro avg	0.78	0.70	0.71	62243
weighted avg	0.92	0.93	0.91	62243

Figure 31: Classification report of Random Forest Classifier

4.3 Evaluation

The last task of this project is to to evaluate the performance of the prediction models on test data which is not used in the process of developing the models. The test data has the same structure as the initial raw data and describes bookings of NextBike in Dortmund for the month July.
The received test data is used to evaluate the chosen models in the predictive tasks. Therefore the models are trained with the whole training set and evaluated with the new test data that.

4.3.1 Trip duration

The table below shows that the metrics for the Random Forest Regressor get worse if the model predicts the durations for the test data.

	training data	**test data**
R2 score	0.20	-0.14
RMSE	74.35	89.58
MAE	32.00	53.11

One reason is that the new test data contains only trips for July. The trained Random Forest Regressor have not seen any data for July before. Due to the usage of *month* and *day_ of_ year* as features in the prediction model, the Random Forest Regressor does not know how to handle the data for July. As described in the Prediction part the Random Forest uses several Decision Trees to predict the target value. The issues with Random Forests are that the results on the test set will get worse if the Regressor doesn't received all the values in training the model that are used by the test set. In this case the data for July is unknown for the trained model. If the training and test data would contain trips that are distributed over the whole year the results would be better than now.
Another reason is that the Random Forest Regressor is already a bit overfitted. If the new data is used for the prediction, it is not surprising that the model gets worse.
In retrospect using a Polynomial Regression would be the better choice to predict the trip durations.

4.3.2 Trip direction

Following figure 32 shows the performance of the model on the test data. When comparing this classification report to the report of the performance on the training data (figure 31) one can see that the performance decreased. In detail, the precision, recall and f1-score for the labels ***towardsUniversity*** and ***awayFromUniversity*** decreased by a little more than 10 percent. As a result, the macro averaged f1-score also decreased. However, the prediction of the label ***noUniversityRide***, accuracy and weighted average f1-score remained stable. However, the accuracy-score is not as meaningful as the f1-score when there is an overpopulation in a class. The reasons for the drop in performance can be given by the characteristics of a Random Forest Classifier. Since the model is trained, inter alia, by the month of a trip, it has not seen the values of the month of July as input before. Moreover, these type of classifiers are vulnerable when there is missing a whole range of values for a predictor-variable in the training data which is used in the test data afterwards. A reason for the stable performance when predicting the label ***noUniversityRide*** can be given by the overpopulation of this attribute in both training and test datasets.

	precision	recall	f1-score	support
awayFromUniversity	0.58	0.78	0.67	1391
noUniversityRide	0.95	1.00	0.97	25378
towardsUniversity	0.54	0.15	0.23	2469
accuracy			0.92	29238
macro avg	0.69	0.64	0.62	29238
weighted avg	0.90	0.92	0.90	29238

Figure 32: Classification report of Random Forest Classifier on test data

5 Conclusion

By analyzing the bike rental data from Dortmund we gathered many insights. We managed to visualize We were able to visually present concise features of the trips and describe patterns. Further we implemented weather data and engineered multiple features to describe the data better. Based on this, we were able to develop a prediction model for trip duration and trip direction, which perform relatively well. The trip duration can be predicted with a R of 0.20 and a RSME of 74.35, which is a reasonable result for a prediction with this kind of data.
Predicting the trip duration is is hindered by several limitations. The error rate is relatively high because the length of a trip is affected by multiple random factors and therefor simply hard to predict. A customer can sometimes drive slower and sometimes faster, more sporty or relaxed, or be in a greater hurry than usual on a particular day because of an event. This limitation of the randomness of the trip duration is very hindering for this work and leads to a rather imprecise prediction. The given trip data, which does not show any trips beyond 12 o'clock at night, further complicates the prediction. This is not a natural progression and is most likely due to incorrect data aggregation.
With regard to the trip direction, these limitations do not apply. Therefore, the test metrics are better here, with a accuracy of more than 0.9 in both training and test data. Moreover, the macro averaged f1-score is 0.71 on the training data and 0.62 on the testdata. It is therefore possible to predict well the direction in which a trip will go in connection with the university.
Furthermore, the prediction model predicts the overall demand of bikes based on the number of rental per time frame. It does not reflect a prediction in terms of load balancing which would also consider the amount of current rentals and returns within the viewed time frame.

In summary, it can be said that trip duration is difficult to predict. Weather data and feature engineering improve the result only slightly. The direction of a trip, however, can be reliably predicted.
More research is definitely needed to better aggregate the data and correct the missing trips, and to analyze other factors such as booking demand or specific customer groups.

A Appendix

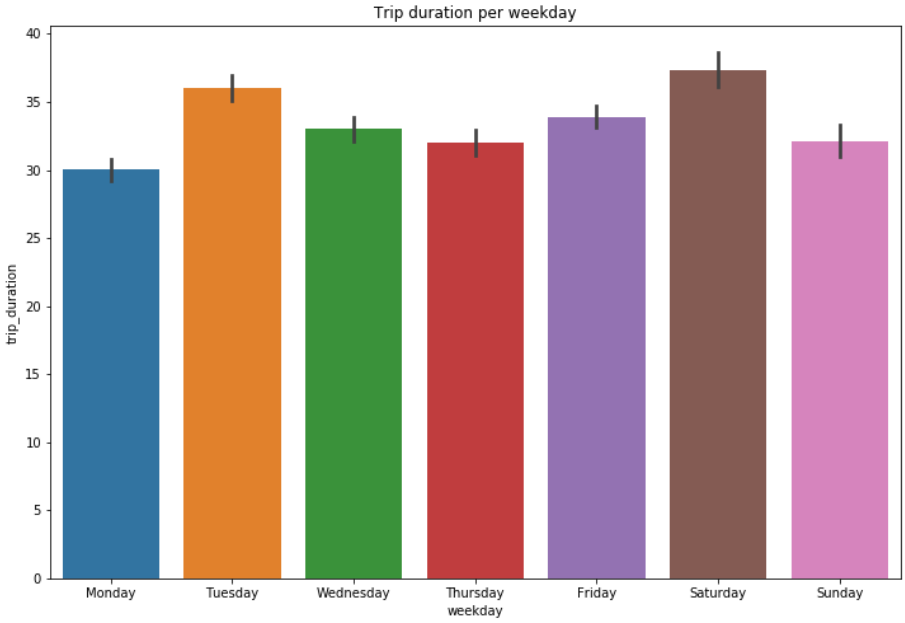

Figure 33: Mean trip duration on weekdays

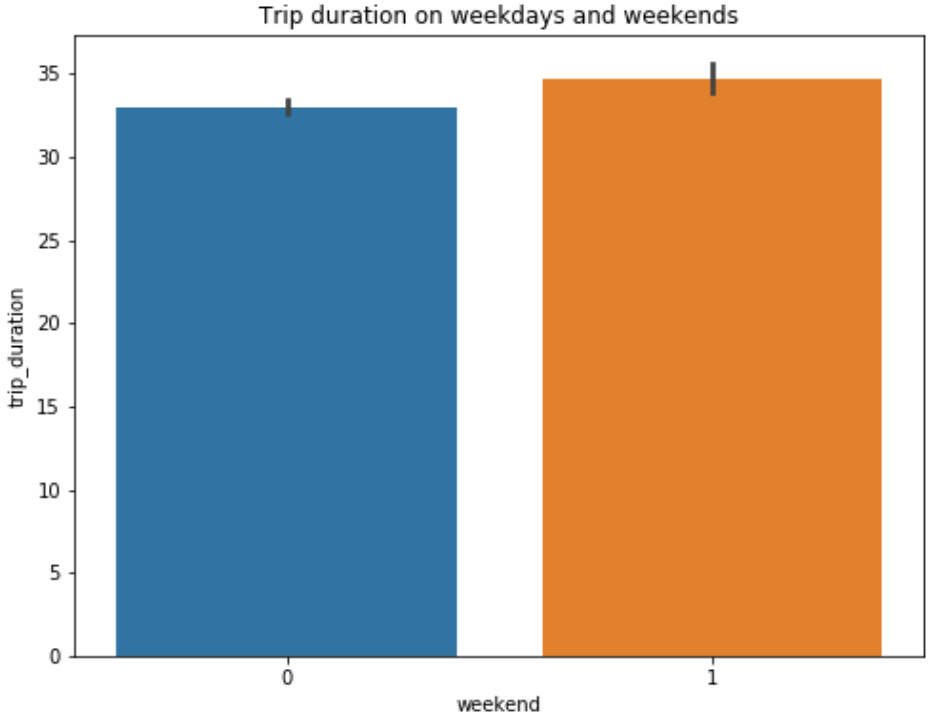

Figure 34: Mean trip duration on weekdays and weekends

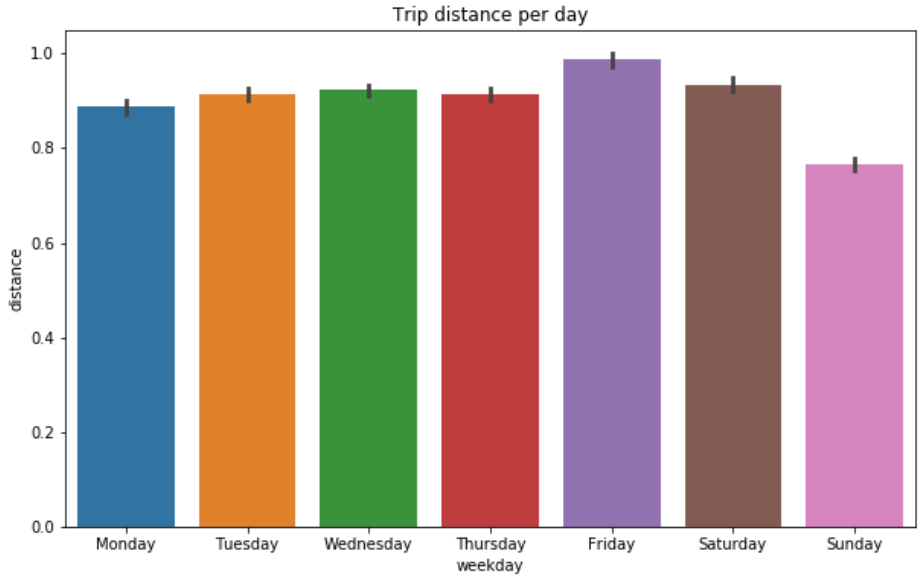

Figure 35: Mean trip distance per weekday

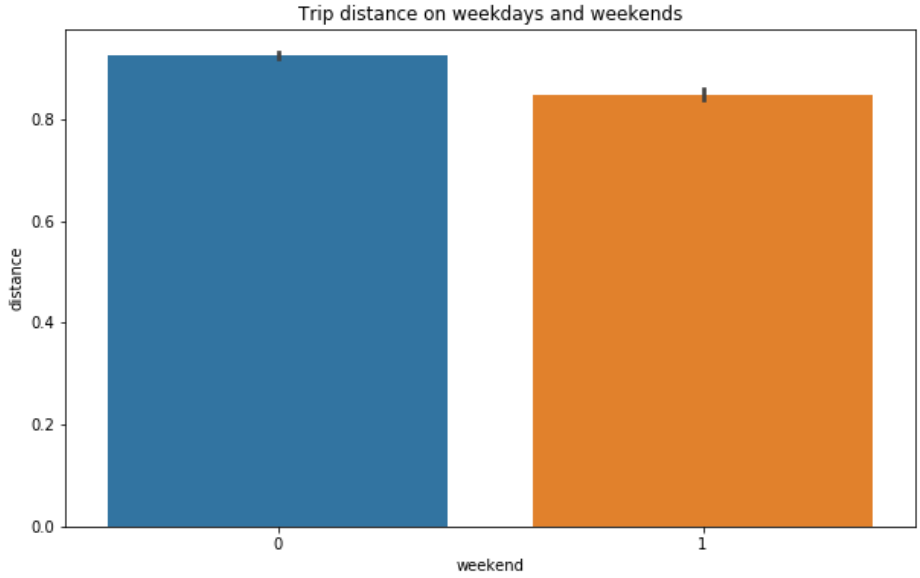

Figure 36: Mean trip distance on weekdays and weekends

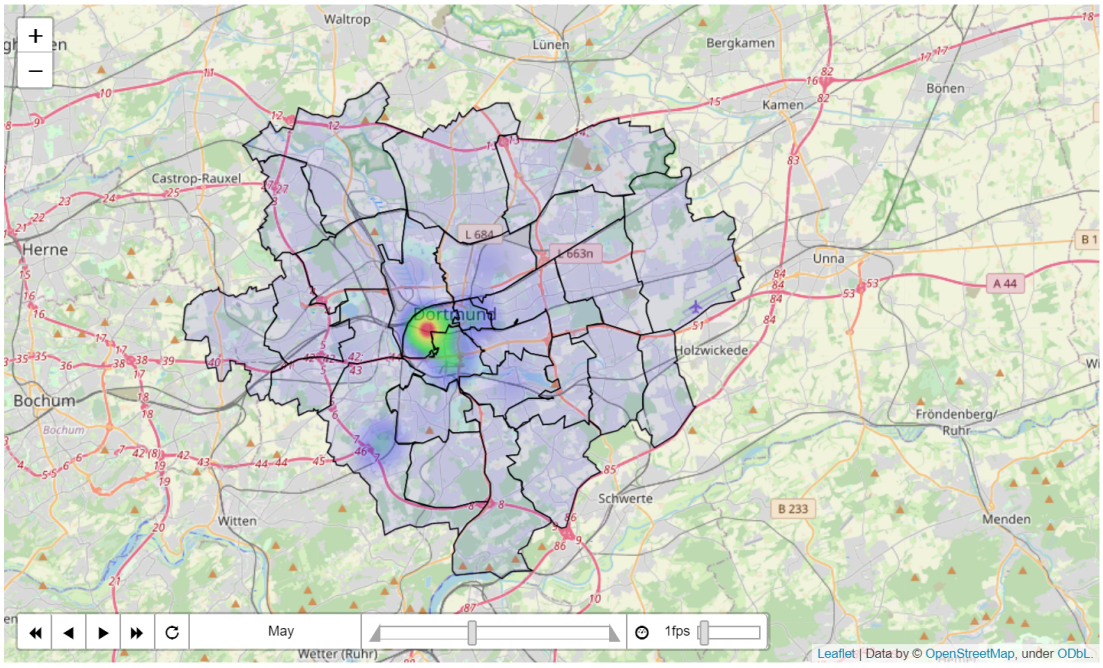

Figure 37: Monthly heatmap

	precision	recall	f1-score	support
0	1.00	1.00	1.00	57052
1	0.96	1.00	0.98	5191
accuracy			1.00	62243
macro avg	0.98	1.00	0.99	62243
weighted avg	1.00	1.00	1.00	62243

Figure 38: Classification report of logistic regression for awayFromUniversity

	precision	recall	f1-score	support
0	0.92	1.00	0.96	57052
1	0.00	0.00	0.00	5191
accuracy			0.92	62243
macro avg	0.46	0.50	0.48	62243
weighted avg	0.84	0.92	0.88	62243

Figure 39: Classification report of linear SVM for awayFromUniversity

	precision	recall	f1-score	support
0	0.92	1.00	0.96	57170
1	0.00	0.00	0.00	5073
accuracy			0.92	62243
macro avg	0.46	0.50	0.48	62243
weighted avg	0.84	0.92	0.88	62243

Figure 40: Classification report of linear SVM for towardsUniversity

	precision	recall	f1-score	support
0	0.92	1.00	0.96	57052
1	0.00	0.00	0.00	5191
accuracy			0.92	62243
macro avg	0.46	0.50	0.48	62243
weighted avg	0.84	0.92	0.88	62243

Figure 41: Classification report of rbf SVM for awayFromUniversity

	precision	recall	f1-score	support
0	0.92	1.00	0.96	57170
1	0.72	0.05	0.10	5073
accuracy			0.92	62243
macro avg	0.82	0.53	0.53	62243
weighted avg	0.91	0.92	0.89	62243

Figure 42: Classification report of rbf SVM for towardsUniversity

	precision	recall	f1-score	support
0	1.00	1.00	1.00	56886
1	1.00	1.00	1.00	5357
accuracy			1.00	62243
macro avg	1.00	1.00	1.00	62243
weighted avg	1.00	1.00	1.00	62243

Figure 43: Classification report of Random Forest for awayFromUniversity

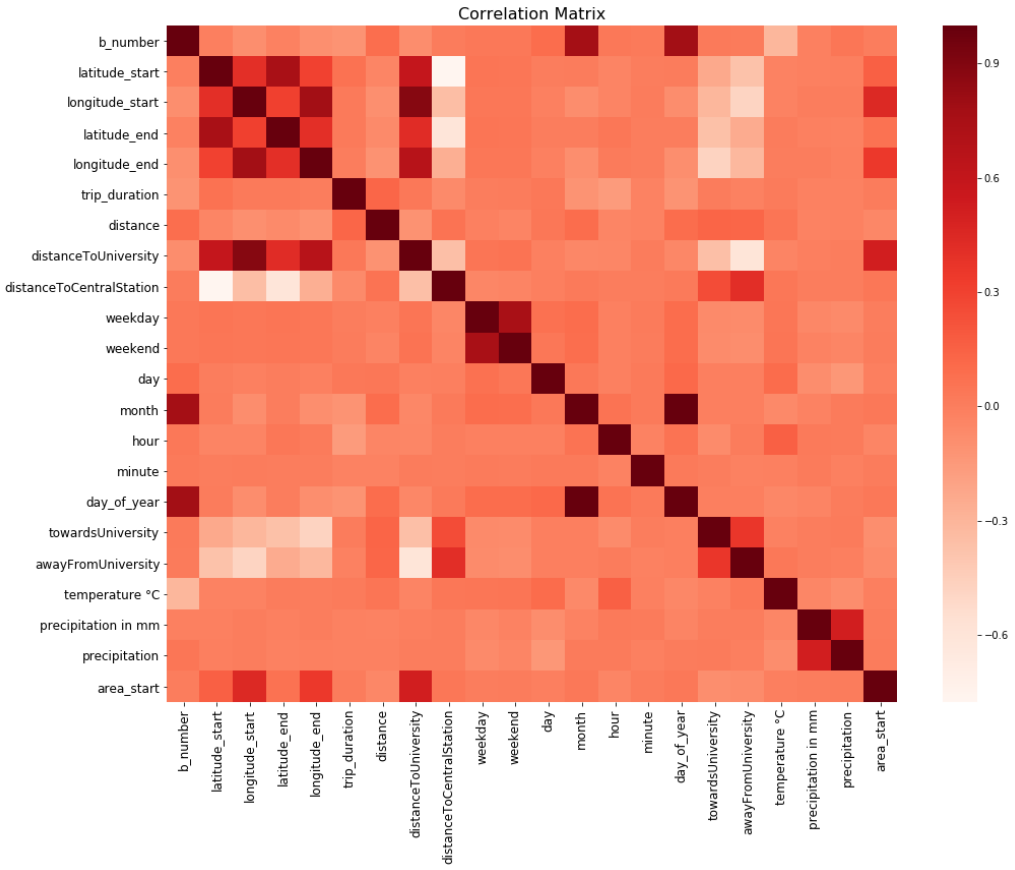

Figure 44: Correlation Matrix

References