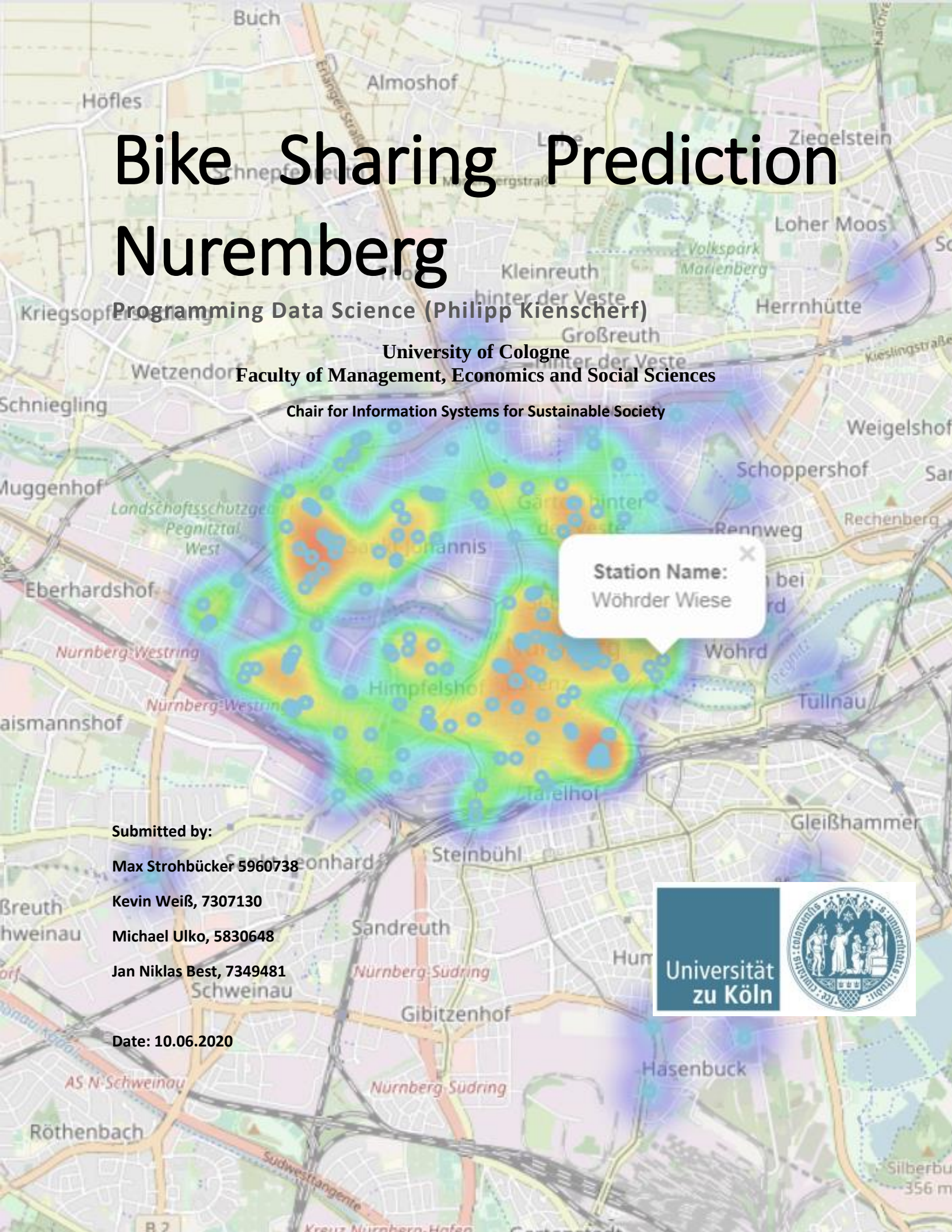
# Bike Sharing Prediction Nuremberg

## Programming Data Science (Philipp Kienscherf)


**University of Cologne**
**Faculty of Management, Economics and Social Sciences**

**Chair for Information Systems for Sustainable Society**



**Submitted by:**

**Max Strohbücker 5960738**

**Kevin Weiß, 7307130**

**Michael Ulko, 5830648**

**Jan Niklas Best, 7349481**


**Date: 10.06.2020**

# Table of content

List of tables ..... 3
List of figures ..... 4
Executive Summary ..... 5
Problem Description ..... 6
Data Description ..... 7
Data Preparation ..... 8
Create initial trip data ..... 8
Creation of Additional Features ..... 8
Remove Quantiles of not Meaningful Data ..... 8
Calculating Postal Codes ..... 9
Calculate Trip Direction ..... 9
Visualization ..... 10
Data Exploration ..... 10
Distribution of trip lengths ..... 10
Trips of August ..... 10
Bikes at fixed stations ..... 10
Trips endings on Christmas ..... 11
Predictive Analysis ..... 12
Feature Analysis and Engineering ..... 12
Data splitting ..... 13
Data Scaling and Principle Component Analysis ..... 14
Model creation and training ..... 14
Duration prediction and evaluation ..... 15
Predictive analysis on the direction of a trip ..... 16
Conclusion ..... 18
Advances and Limitations ..... 18
Further Research ..... 18
References ..... 20
Appendix - Graphics ..... 21

# List of tables

Table 1: Original dataset ..... 7
Table 2: Additional features ..... 8
Table 3: Machine Learning Models applicated on the validation data ..... 15
Table 4: Confusion Matrix classification (positive=towards the university; negative=away from university) ..... 17
Table 5: Performance metrics of different classifiers ..... 17

# List of figures

Figure 1: Distribution of trip duration on the 90% quantile ..... 21
Figure 2: Mean Duration of trips per day of year ..... 21
Figure 3: Mean and standard deviation of trip duration (left). Distribution of trips on weekends and workings days (right). ..... 22
Figure 4: Mean and standard deviation of trip duration for every day of week (left). Distribution of trips for every day of week (right). ..... 22
Figure 5: Trip duration distribution per month and normalized over a year ..... 23
Figure 6: Month with most trips (August) for each postal code area ..... 23
Figure 7: Fixed stations and bikes at stations (01.10.2019 08:00) ..... 24
Figure 8: Heatmap of trip ends at Christmas (24.12.20109) ..... 25
Figure 9: Visualized trips on the 24th December 2019 with start and end point ..... 25
Figure 10: Training and validation loss of NN compared without and with weather data ..... 26
Figure 11: Correlation Matrix ..... 26
Figure 12: PCA Analysis of components ..... 27
Figure 13: Training and validation loss of Neural Network trained in 100 epochs ..... 27
Figure 14: Residuals of Neural Network prediction on validation set ..... 28

# Executive Summary

The main task is to predict the expected duration and direction of a trip within Nuremberg based on past booking data from nextbike. The given dataset is not good enough to get an accurate result. In the beginning, the booking data is transformed into trips and invalid data and noises are removed. Additional information like postal codes and weather data are added to receive more meaningful data.

The analysis of the data follows the cross-industry standard process for data mining consisting of business understanding, data understanding, data preparation, modeling, evaluation, and deployment. (Shearer 2000, p. 14) To prepare the data for further analysis, unnecessary or irrelevant information is dropped and additional features such as the duration of a trip are created and added. Then the dataset is split into start and end information, cleaned from duplicated or wrong entries and merged so that each row of the new dataset corresponds to one trip (later also named booking). The corresponding start and end postal codes are added, entries not recorded in Nuremberg are dropped and the final trip data is saved as a CSV file.

The descriptive analysis revealed, that over 75% of the trips are shorter than 10 minutes with a mean trip duration of about 8 minutes. In general, trip durations increase towards the weekend, most of the trips take place in the afternoon, in the center of Nuremberg and about 30% of the trips were towards the university.

For the predictive analysis, all trip attributes are scaled or reformatted to fit the requirements of successful training and prediction. Different models are tested, and the neural network was identified as the best usable one. Nevertheless, based on the given data, machine learning algorithms cannot solve the task perfectly.

The best prediction results compared to the runtime were produced by a neural network using 50 epochs and 5 layers with an RMSE of 8.09, MAE of 6.47, which means an average error of about 6 minutes and 29 seconds when applied on the test set from July. With an error of over half of the duration mean we assume that the neural network could be biased or we are missing some explanatory features. To improve these results, more data has to be collected over a longer period or additional features should be added (e.g. user data). Including weather data, unfortunately, did not improve these results significantly. With an accuracy of about 75%, the neural network could correctly predict whether trips are made towards the university or not.

The deployed software has a modular structure of the provided python package which enables easy installation and usage of the modular components. These four easy-to-use commands are provided by the package:

- Transformation of new bike rental data into data containing trip information
- A descriptive analysis of new bike rental data, including the generation of several plots
- (Re-)Training models with the existing data
- Prediction of trip duration and trip direction based on an input file

Each prediction runs completely independent of the others, so that it is possible to only predict the duration of a trip, the direction of a trip, or both together. Pre-trained models exist to instantly start the predictions without training the model before.

## Problem Description

Climate change has already begun worldwide and is a global challenge that cannot be ignored anymore. The emission of greenhouse gases is one of the facets of climate change. Its consequences are serious and affect both mankind and the whole nature (Rodt et al. 2010, p. 9). About one-fifth of Germany's greenhouse gas emissions are currently produced by the transport sector (Hornberg et al. 2017, p. 3). The largest share of greenhouse gas emissions comes from road traffic (Hornberg et al. 2017, p. 4), which continues to be a growing source of CO2 emissions worldwide (Schinke et al. 2010, p. 43). A rethinking in the transport sector offers the opportunity to significantly reduce this development, to increase the safety of cyclists and pedestrians in road traffic (Rodt et al. 2010, p. 34) and to positively influence people's living conditions (Hornberg et al. 2017, p. 4). CO2 emissions can be reduced through the stepwise introduction and usage of climate-friendly forms of transport (Hornberg et al. 2017, p. 71). One such climate-friendly approach is bike rental, which has become an important part of municipal strategies for sustainable urban mobility concepts in many places. nextbike, a bike rental company founded in Leipzig in 2004, is one of the largest providers of modern bike-sharing systems in Europe (nextbike GmbH 2020). However, the management of a bike-sharing platform is complex and depends on many interrelated factors such as total rentals, the intensity of use in an area, trip duration and direction. Our goal is to apply and optimize machine learning models, to effectively predict the duration of individual trips and the direction of travel using available information to improve customer satisfaction and increase the usage based on improvements from nextbike regarding the results of this study. (Jupyter Explorers 2019, p. 2; A Team 2019, p. 1)

## Data Description

The data set of bike-sharing-rentals from the provider nextbike was was provided to us by our client, the University of Cologne. The dataset contains various information about bike bookings in Nuremberg in 2019. It includes data such as bike IDs or location-related data. An overview of the original dataset including the important attributes is provided in Table 1.

*Table 1: Original dataset*

| **Column** | **Data type** | **Description** |
|---|---|---|
| p_spot | bool | Indicator whether a bike is placed on a bike spot |
| p_place_type | int64 | Place type: '0' = bike spot, '12' = outside of bike spot |
| datetime | DateTime object | Start or end DateTime of the rental for one special bike |
| b_number | int64 | Bike number |
| trip | object | Bike tracker for 1) bike availability (*first & last*) and 2) booked trips (*start & end*) |
| p_uid | int64 | Unique identifier for place |
| p_bikes | int64 | Number of bikes placed on the selected position for the special DateTime |
| p_lat | float64 | Latitude of the bike position |
| b_bike_type | int64 | Bike type |
| p_name | object | Name of the bike position |
| p_number | float64 | Bike position ID |
| p_lng | float64 | Longitude of the bike position |
| p_bike | bool | Indicator whether a bike is placed outside of a bike spot |

The prefix *b* is an abbreviation of "bike" and *p* is an abbreviation of "place". "Place" is a more comprehensive term for bike stations and provides information on bike station type, its unique ID, name, geographic coordinates, and the number of bikes placed on the selected position for the rental date. It also provides information, whether a bike is stationed on or outside of the fixed station. The *trip* column contains four different values: "first", "last", "start", and "end". "first" and "last" belong together as much as "start" and "end". According to our analysis, these value pairs provide various information about the status of every bike. Based on the available data, it is evident that "first" and "last" provide information on the bike availability on each day and are recorded once daily, "start" and "end" represent the start and end time of a rental transaction.

# Data Preparation

For the data preparation steps, the process will be described in the following paragraphs. The original CSV file is imported by using pandas and the following steps are applied to this raw data. During the data preparation, valid trips are created and saved as a CSV file.

## Create initial trip data

The first step is to clean duplicated entries. Here the geographical data are excluded because of little differences in the raw data in longitude and latitude values. Usually, the values have six decimal places, which has an accuracy of 11.1 cm whereas some values have 12 decimal places, which would result in an accuracy of 30 nano millimeters.

After that elimination, only entries in the column *trip* with the values “start” and “end” are considered as described in chapter

Data Description. At the same time, some entries were identified and deleted where “start” and “end” occur multiples times in sequence in the data frame. The time difference between these values differs in most cases just a few seconds, which indicates trouble in the booking process. The nature of such errors remains unknown due to no further information. Afterwards, a merge of each *start* entry with its appropriate *end* entry for each bike is performed to receive a detailed and valid trip information dataset.

## Creation of Additional Features

Now a dataset with detailed information for booked trips for each bike rental transaction is available for closer analysis. These data are used for the creation of additional features, which have positive effects for further model training and prediction. The created features are presented in Table 2 for the start- and endpoint of a trip.

*Table 2: Additional features*

| **Additional Feature** | **Data Type** | **Description** |
|---|---|---|
| Weekend | bool | Weekend day or not |
| Duration | int64 | Trip duration in minutes |
| Month | int64 | Trip month (1=January; 12=December) |
| Day | int64 | Trip day (1-31) |
| Hour | int64 | Trip hour (0-23) |
| Minute | int64 | Trip minute (0-59) |
| Day_of_year | int64 | Numerated days in the year (1-365) |
| Day_of_week | int64 | Numerated days of the week (0=Monday; 6=Sunday) |
| Season | Int64 | 1=winter; 2=spring; 3=summer; 4=fall |

## Remove Quantiles of not Meaningful Data

Extremely short and extremely long trips seriously distort model training and prediction. Therefore, we decided to drop trips shorter than one minute and bigger than the 90% quantile. The remaining data and the distribution of duration are displayed in Figure 1.

## Calculating Postal Codes

Furthermore, information regarding postal codes is imported and merged into the trip data. By intersecting the locations with the postal code regions, it is possible to filter all data points inside of Nuremberg. The enriched dataset with postal code information is saved.

## Calculate Trip Direction

As next, the trip direction is calculated. In this way, it will be discovered whether a journey has been made towards the university or away from the university. The university is defined as the “Friedrich-Alexander-University Erlangen and Nuremberg”. The main building is in Erlangen, so the faculty of economics is defined as the university spot of Nuremberg due to the more central location than the faculty of educational sciences. For this purpose, the distance between the start points of the journeys and the university is calculated and analogously the distance of endpoint and university. Finally, both distances are compared and if the distance between the endpoint and the university is shorter than between the start point and the university, then it was a journey towards the university.

# Visualization

In the next step, we will visualize some important facts resulting from our analysis of the transformed data.

## Data Exploration

First, we look at the mean trip time for each calendar day presented in Figure 2. Here two interesting facts are considered. The first one is that the mean trip duration was longer in the first four months. The second fact consists in the impossibility to indicate the mean trip duration for May and July. There are various explanations for every month.

In the original dataset data for May is available. However, processing the data preparation steps showed the necessity to proceed without these data due to the geographical indicators. Because the geographic coordinates point to Marburg they are not suitable for further analysis for Nuremberg. One possible reason for this anomaly can be the dismantling of bike rental system for Nuremberg from nextbike in April 2019 and the planned introduction of a new bike rental system in May 2019 (Judith Horn 2019). There may have been a delay in the introduction of the new system, so the bike rental service was available again only from June 2019.

The data for July wasn't included in the dataset due to the reason, our client want to test the developed prediction model on these data.

Secondly, we try to determine how driving behavior differs between working days and weekends. To do this, all trips from Monday to Friday and from Saturday to Sunday are grouped and visualized. From the visualized results in Figure 3, it is evident that the average trip duration on working days is a bit lower than on weekends and about 23% of all trips take place on weekends. Figure 4 gives a closer look at the average trip duration and distribution of trips for each weekday.

## Distribution of trip lengths

Figure 5 shows the distribution of trips every month except May and July. It is remarkable that in February suddenly the durations drop to a minimum and up to the summer months the distributions are getting longer again. Also in comparison to the rest of the year in the spring months, the standard deviations vary more.

## Trips of August

Figure 6 presents the started trips per postal code for each postal code region. The more purple the postal code region is colored, the more trips were started. Additional information is shown in blue circles which indicates the fixed stations and the black marker points to the university. This figure visualizes that bikes are more often used in the city center of Nuremberg and towards the direction to Fuerth. In the border of Nuremberg are not so many fixed stations installed so far. Additionally, in the city center, there are free-floating bikes available, which can be returned everywhere and are not bound to specific stations. This increases the number of trips as well.

## Bikes at fixed stations

In the following, Figure 7 shows the available bikes in the city center at fixed stations. There are many stations in the city center with only few bikes available because free-floating bikes are allocated around,

too. Outside of the city center, where the customer has to return the bikes at a fixed station, the stations have more available bikes. The black marker displays the university.

## Trips endings on Christmas

The heatmap plotted in Figure 8 displays ended trips in that region. The more spots are colored red, the more trips ended there. One red spot is at the “Frauenkirche”. This is a church, where the Christmas market is located. The other red-colored spot is the biggest graveyard of Nuremberg. Additionally, no trips were made away or towards the university at Christmas.

For December 24$^{th}$ the visualized trips in Figure 9 indicate that more people used the bikes to visit the church, the graveyard or to ride between both places. Green circles mark startpoints and red circles mark endpoints of trips. At points, where only red circles occur, these red circles may overlay green circles. This means, the trip ended where it started, or this bike was used more often this day. This visualization also confirms that there were no trips around the university at Christmas.

# Predictive Analysis

Predictive analytics creates new possibilities for planning. If a bike rental provider would know the duration of booking at the moment of the booking, it would know how long the booked bike is not available to other customers. This trip duration prediction can be used for planning how many bikes are needed to fulfill customers' demand. Also, the analysis of typical trip destinations creates an overview of where trips are likely to end. Though it would be possible to analyze places with many bikes and places with less. This information could be used to distribute the bikes fitted to the customer's demand around these hotspots. In a city like Nuremberg with a big university, the university could be one of them. Thus, it is meaningful to evaluate if a trip goes towards or away from the university. With this classification, the number of bikes needed at different times to be able to fit the demand of students on the way to university and away from it can be calculated.

The predictive analysis is divided into two main tasks:

- **(a)** Regression on trip durations at the moment where a bike is booked. The duration prediction is divided into the following subtasks: (1) Feature analysis and creation, (2) data splitting, (3) data scaling and principal component analysis (PCA), (4) model creation, and training, (5) duration prediction and evaluation.
- **(b)** Classification if a trip will be towards or away from the University of Nuremberg in the moment of bike's booking. The steps are like those of the duration regression, especially feature analysis, data splitting, data scaling, and PCA. Therefore, they will not be discussed in detail. But all differences to the duration prediction analysis will be stated.

## Feature Analysis and Engineering

During feature analysis, the features which are used to predict the duration of a trip are selected. First the variables *Place_start* and *Start_Time* are dropped. *Place_start* contains the names of the different stations. The same information is included in the variable *p_uid_start*, but as a numerical representation of the names and can, therefore, be used for calculations during training and prediction. The *Start_Time* variable is split into its atomic information. At this point, a correlation analysis is used to take a first look at the dependencies of the features. Next, all information about the end of a trip will be dropped. It would not make any sense to predict the duration of a trip at the start of it and therefore using information that only exists if the trip has ended. After that, dummy variables are created for the Boolean variables *p_spot_start*, *p_bike_start,* and *Weekend*.

This step is followed by the feature creation, where the squared features of the atomic start time information and the squared latitude and longitude of the start points are added to the dataset. This choice has been made because of the assumption that there is a causal dependency between the time of a trip and its duration e.g. trips to the office or train station in the morning and trips away from the office and train station in the afternoon. The squared features receive a higher influence in the model as there will be more variance due to two squared scaling. Including these squared features results in a better model performance on the validation set.

Additionally, weather data for temperature per hour and rain in mm per hour are used (Stadt Nürnberg 2019). The weather data from 01.01.2019 to 31.12.2019 are included. Missing values are filled by values

of the previous measurement. As these weather data need to be fetched from an external source for the start of trips, the inclusion of weather data has been made optional. If one decides to use the weather data one should keep in mind that the quality of weather data is very important for the prediction. Against our assumptions, the performance of the models does not clearly increase. The impact of weather data on our models depends on the hyperparameters of the models. Therefore, look at Figure 10.

When all features are created, another correlation analysis is done. Features with high positive or high negative correlations could be a sign of redundant information in the data and therefore lead to bad results in model performance. But as some features could have a high explanation power on *Duration* when they are combined with others' we decide not to drop features depending on correlations. Also doing repeated hypothesis tests for the coefficients of multiple linear regressions could lead to an accumulation of type I errors. So the decision was made to use a PCA analysis for feature transformation as described later. After that, a forward selection method is used to select PCA components to include in our models. The neural network is trained on all combinations of components. Afterwards, the performance on the train and the validation set is measured and saved. In the end, the different metrics are compared and 17 components received the best result for our neural network regression. As there was not enough time to do the same for the classification of trip directions the decision was to also include 17 components in the direction prediction. Statistical theory tells us, that a model that includes irrelevant regressors results in a higher variance than the perfect model. But a model that excludes regressors which would have explanatory power to the dependent variable would create a bias in the model (Verbeek 2018, p. 66 ff.). Therefore we decided to minimize the bias and accept some more variance in our models.

As it can be seen from correlation analysis especially the time depending features are correlated with the duration of a trip. This follows our assumptions that the duration of a trip depends on the time at which the trip starts e.g. the day of the year, the season, the hour of start or weekends, which is represented in Figure 11.

## Data splitting

The final features data are split into train 50% and test 30% and validation set 20%. The train set is used to train the parameters of the regression and classification models e.g. coefficients of the linear regression. After training the validation set is used to look at the models' performance on predicting the validation data. With the results the hyperparameters are fitted e.g. number of hidden layers in the neural network. The reason for the train and validation split is that there will be no fitting of hyperparameters on the test set. It is guaranteed that the test set is only used to evaluate the results after the models are fully trained. To receive each time the same split we use a random state of 42 by splitting. To ensure that we do not overfit our model choice on exactly this data split, we tried out different random states at a testing stage in the project. This testing stages showed that the results on different splits are very similar and therefore no k-fold cross validation is needed. The only difference in splitting between regression of *Duration* and classification of *Direction* is that we use different dependent variables. For the regression split we use *Duration* and for the classification split we use *Direction* as the feature which should be predicted.

## Data Scaling and Principle Component Analysis

The features are standardized as $U = \frac{x - \mu}{\sigma}$ with x as the value which should be scaled, μ as the mean over the whole feature and σ as the feature's standard deviation. This overcomes problems of magnitude and is also necessary for the PCA because otherwise features with a larger range of values create more variance and therefore could bias the PCA. The standard scaler of scikit-learn is only fitted on the train data. Otherwise, one would look at the test set and use information from it which in real-world application are not given. In comparison to the standard scaler with previous noise-cleaning a robust scaler less noise cleaning (0.95 quantile instead 0.90) was used. The results of the regression were worse by using the robust scaler. Therefore, we decide to stick with standard scaler and noise-cleaning.

After scaling the PCA is used to create components of the features. PCA is not just used for dimensional reduction but also to nearly eliminate covariance between the features. For some models, a covariance between independent variables generates a worse prediction performance. The number of components is set to 17. These components capture ≈100% of the variance in the features for regression and classification. Therefore, as much variance as possible is captured with not including features that explain only minimum amounts of variance. This also results in a lower number of dimensions such that the models do not run in problems such as long runtimes. The variance captured variance by each component can be seen in Figure 12. As the captured variance of a component is not crucial for the influence of an independent variable on a dependent variable, also the components with a low amount of captured variance are included. The resulting components are then used in the next steps.

## Model creation and training

As the models are trained on the 90% quantile of the data, the models are much better in predicting short trip durations which are smaller than 40 minutes but worse in predicting longer trips. This is a decision that has been made because of the high number of short durations in the data and only fewer long trips.

To find the best model for predicting the duration of a trip, different models are trained on the train set and the results are compared by Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and $R^2$. The models which are created can be seen in Table 3. The dummy regressions were used to evaluate other regression models. The assumption is that "intelligent" machine learning algorithms have a better performance on the data than basic dummy regressions that always put out the mean or median of the train set on predictions. The model with the best results is the neural network. This has been evaluated by brute force testing over 600 hyperparameters on different models and comparing validation, and training set performance. Therefore, in the following, only its parameters are explained in detail.

The neural network has an input layer with an input shape of 92,507. All five layers, except the output layer, have a dimension of 36, use the rectified linear unit (ReLU) as activation function and their kernel is initialized by a standard normal distribution of weights. As an optimizer, the RMSprop optimizer is used (tensorflow.org 2020). For optimization, a learning rate of 0.001 is used. For the regression on *Duration,* an output layer of dimension 1 is needed as a prediction, if only one value is necessary. These model parameters have been evaluated by testing different settings and comparing the results of training and validation loss. The final decision has been made by choosing the smallest model with appropriate performance. Such that model complexity has been reduced to a minimum. This helps to prevent

overfitting on the training set. The final step is the choice of epochs to train on. To evaluate the number of epochs, the training and validation loss is plotted per epoch and presented in Figure 13. The validation loss decreases together with training loss approximately up to epoch 50. Therefore, the maximum number of epochs which should be used amounts 50 epochs. Thus, overfitting is prevented. At this point, the neural network with 50 epochs has the same performance on the data with and without weather data, which is visualized in Figure 10.

The neural network has a MAE ≈ 3.89426 and RMSE ≈ 6.7261 when fitted on the training set. This means that the trained model has an average bias when predicting a trip duration which is at about 4 minutes. The residuals plot confirms this assumption, too. This plot shows for a perfect model a line with $f(x) = y$. The trained neural network does not capture the duration longer than 30 minutes and tends to underestimate the actual value. It is visualized in Figure 14. As the RMSE is clearly larger than the MAE, it can be stated that different errors are not of the same magnitude as RMSE penalizes big errors stronger than MAE. The presence of higher errors has also been seen in the residuals plot. This bias should be kept in mind when looking at the prediction of new data.

## Duration prediction and evaluation

The different models create the following results on the test set:

*Table 3: Machine Learning Models applicated on the validation data*

| Model | MAE (≈) | RMSE (≈) | $R^2$ (≈) |
|---|---|---|---|
| Dummy regression (mean) | 5.1190 | 7.4736 | -6.5641 |
| Dummy regression (median) | 4.7430 | 7.6723 | -0.0538 |
| Linear regression (17 regressors) | 4.9311 | 7.2374 | 0.0622 |
| Support Vector Machine | 4.1958 | 7.3810 | 0.0246 |
| Neural Network (50 epochs) | 3.9716 | 6.5361 | 0.2351 |

Again, the results state that the neural network is the best model in predicting the duration of trips on the test set. Therefore, it is chosen to predict the duration of the new data for the hackathon.

It is evident that the MAE and RMSE are not getting worse when looking at the test set performance. That means that the model predicts with some bias learned during training but does not overfit to training data. Additionally, the $R^2$ value of the neural network is much better than in the other models. But only 23.51% of the variance in the real data is explained by the predictions of the neural network. This states that at about 75% of the variance in the actual trip durations cannot be explained by the neural network. Therefore, we recommend using the model very cautiously.

Reasons for the bad prediction results could be the wrong creation of trip data from raw data. To create trip data, one assumption is that for one bike the timeline of bookings (start and end data) can be used to match one trip start with the following trip end. As there were a lot of errors in the raw data regarding double or triple bookings for single trip starts, this assumption could be wrong. Incorrect trip data would lead to incorrect predictions, of course. To overcome this problem, it would be needed to dive deeper in the way of how these data are collected and where the errors come from. To find out about this we would recommend contacting nextbike or the service provider who makes the data available.

## Predictive analysis on the direction of a trip

The direction of a trip is a classification problem. Hence, supervised machine learning algorithms are used to solve it. As during the feature analysis for the regression problem also for classification, all trip information regarding the end of a trip are not used. Dummies for Boolean variables are created. Again, squared features are used to receive more explanation power in the independent variables for the dependent *Direction*. The only difference in feature preparation is that the dependent variable is the Boolean *Direction* which is True if a trip goes towards the university and False if not.

After a train-test-split, the matrix of independent variables is used to train a standard scaler and after that is scaled by it. Furthermore, a PCA is used to reduce the dimension of the matrix and to eliminate covariance between the features. This PCA is as the scaler fitted by training data. The PCA creates 17 components that explain ≈100% of the variance in data. The resulting matrix of training components is again split in a train and validation set. The training set holds 50% of all data and the validation set 20%. The train- and validation set is then used in the training process of the following classifiers:

(1) A dummy classifier that learns the most frequent classification in the data and always classifies with this label. This classifier is used to evaluate the other models' performance compared to this very primitive classification. (2) A K-Neighbors classifier which calculates the 10 nearest neighbors to each point and uses the label which is the most common in this class. As a weight function, the inverse of the distances is used. This results in the fact that closer neighbors have a higher influence than those which are further away. The decision on which algorithm to take to compute, the nearest neighbors is set to auto such that the most appropriate algorithm will be taken. A distance measure of the Euclidean distance is used. (3) A Decision Tree classifier for which only the maximal depth is set to 5. This choice is made to avoid overfitting. (4) A Random Forest classifier which also has a maximum depth of 5 and uses 10 trees in the forest. (5) The Neural Network classifier with default settings of scikit-learn (Scikit-Learn 2020) instead of the number of maximal iterations which is set to 1,000.

To estimate the best model again, a brute force approach is used to train the models on over 640 hyperparameters. The result was created in a twelve-hour training and validation of the models. The results are saved and compared in precision, accuracy, recall, and f1 score.

To calculate these metrics, the confusion matrix of the classifiers should be analysed. The confusion matrix of the neural network is shown in Table 4 in comparison to the confusion matrix of the dummy most frequent model. The true positive values show the number of trips towards the university which were predicted as those. The true negative values are the number of correctly predicted trips away from the university. Trips, which were actually towards the university (positive), but predicted as "away from the university" (negative) are called false negative values and trips actually away from the university (negative), but predicted as "towards the university" (positive) are called false positive values. While only comparing these values may be hard and not that meaningful, different metrics like accuracy, precision or recall are derived from the confusion matrix to be able to compare machine learning algorithms.

*Table 4: Confusion Matrix classification (positive=towards the university; negative=away from university)*

| | | Predicted | | | Predicted | |
|---|---|---|---|---|---|---|
| | Neural Network | Positive | Negative | Dummy most frequent | Positive | Negative |
| Actual | Positive | 9117 (TP) | 6922 (FN) | Positive | 0 (TP) | 16039 (FN) |
| | Negative | 4367 (FP) | 35099 (TN) | Negative | 0 (FP) | 39466 (TN) |

Coming back to the brute force approach, those metrics derived from the confusion matrixes of the tested classifier algorithms should be considered. The best results for each algorithm can be seen in Table 5.

*Table 5: Performance metrics of different classifiers*

| Classifier | Accuracy (≈) | Precision (≈) | Recall (≈) | F1 Score (≈) |
|---|---|---|---|---|
| Dummy (most frequent) | 0.7110 | 0.0* | 0.0 | 0.0 |
| K-Neighbors | 0.7760 | 0.6549 | 0.4758 | 0.5512 |
| Decision Tree | 0.7525 | 0.6363 | 0.3356 | 0.4395 |
| Random Forest | 0.7561 | 0.7308 | 0.2471 | 0.3693 |
| Neural Network | 0.7966 | 0.6761 | 0.5684 | 0.6176 |

Before the interpretation of the results, it is important to mention that 30% of the trips in the test set go towards and 70% away from the university. The trained models learn that trips away from the university are more than twice as much as trips towards the university. Additionally, one should also keep in mind, that the dummy classifier has to be handled with caution as it will divide by zero when calculating the precision. That's why the precision is interpreted as zero.

The accuracy of a classifier shows how many directions were classified correctly in general (so the percentage of true positive and true negative labelled trips of all trips). The neural network produces the best accuracy with 79.66%, which is about 8% more than the dummy classifier, which simply labels all trips as the most frequent one. The best precision was reached by a random forest classifier, meaning that 73.08% of the trips which were labelled as “towards the university” were also actual trips towards the university. None of the classifiers managed to get significantly good recall values. The recall metric describes, how many of the actual trips towards the university were also label as “towards the university”. Therefore, even the neural network as the best classifier did not perform very good with a recall score of 56.84%. All in all, the neural network performed best, also regarding a F1 score of 61.76%. Therefore, the neural network has been taken for further predictions as it is not that complex in comparison to other models which created a precision only 6% better than the neural network.

Regarding the direction prediction of different subsets of our data, it was remarkable, that there are much more trips towards the university during the semester in comparison to the summer months, where the semester holidays took place. But it should be mentioned here, that this is a not very precise analysis, because we are missing dates for May and July, which are the months in the middle of the semester and the beginning of the semester holidays.

# Conclusion

## Advances and Limitations

As it does not make any sense to train a model on all trip data which would result in high errors during prediction, the aim was to take restrictions and therefore to achieve better performance. The restriction of the model is to predict the most frequent trips (< 10 min) with higher accuracy. But do not predict trips that are longer than around 50 minutes very well. This limitation seems for the prediction of the most frequent trips of nextbike useful. For short trips that are under 50 minutes, the estimated duration is good. The weakness of the model is clearly longer trips. As it was needed to create trip data from raw data a lot of data cleaning had to take place. This has been seen in the number of data points which was over ≈2,100,000 and only ≈185.000 resulting trips. Therefore, cleaning is needed at the start of a prediction if raw booking data is handed. This cleaning process needs trips end information in the test set, which would not be available for application during rental transactions in real time. Therefore, by cleaning the test dataset it becomes corrupted due to the reason of usage of non-existing information. To overcome this problem two possibilities come up.

1. A new machine learning model that works as a cleaner should be trained on the training data. It learns which trip starts are redundant or noisy. Then this model is used to clean the trips start data without knowing the end information.
2. Receive better data from scratch. The received data is of bad quality. Therefore, one should look inside the rental tracking system and eliminate the noise at the point of booking. Multiple pushes of booking to the server would otherwise result in redundancy.
   Both possibilities are not applicable for the horizon of this project.

Both possibilities are not applicable for the horizon of this project. As a workaround, our test dataset got the same cleaning as the training dataset before the prediction was done. Our results are: RMSE: 8.09 MAE: 6.47. Without this cleaning, the results are more than four times worse.

## Further Research

For further research, a higher amount of data is useful to train the model on it. This would reduce the problems regarding missing months. Also, in the project in can be seen that predictions of completely missing months are worse than predictions of single data points in a month on which the model was trained. To receive these data analysis on general using patterns in multiple cities seems useful.

Additionally, a more extensive feature analysis should take place to investigate relationships between independent variables and the variables to predict. The perceived data does not explain the variable duration very well. So, searching for the missing variance which is not explained and finding predictors for it should be the next step. One entry point could be the analysis of spatial data which can be included e.g. population density. Another point to enter the feature analysis could be more complex weather data or analyzing specific user groups e.g. students, commuters, or tourists which could be a little bit difficult to find in data.

It is important to execute a real-world data science project with business experts to get more business insides which then is connected to the data. Also, daily news of the company and the development could

create inside information. The improvements arrived due to business knowledge on one hand can improve the models and on the other can be used by the business experts to apply in the real-world.

The last promising approach which we thought of is the use of ensemble methods. The idea is to train a classification model that evaluates different trip sizes. On the resulting subsets, regression models can be used to estimate the duration of a trip more precisely. This approach would work if the classification problem is simpler to solve than the regression. Possible splits could be trips under 10 minutes and from there on the steps should become larger each step as there are not that much trips anymore.

All in all, the project report tells the story of very messy data for which a lot of creativity was needed to get even some meaningful results. This is different from typical toying problems from teaching in data science. Nevertheless, the project mirrored a realistic problem which thought us that data need a lot of massaging to use machine learning algorithms in a meaningful way.

# References

A Team (2019): Bike Sharing prediction Marburg. University of Cologne - Information Systems for Sustainable Society.

Hornberg, Claudia Prof. Dr.; Niekisch, Manfred Prof. Dr.; Callies, Christian Prof. Dr.; Kemfert, Claudia Prof. Dr. (2017): Umsteuern erforderlich: Klimaschutz im Verkehrssektor. Available online at https://www.umweltrat.de/SharedDocs/Downloads/DE/02_Sondergutachten/2016_2020/2017_11_SG_Klimaschutz_im_Verkehrssektor.pdf?__blob=publicationFile&v=25, checked on 6/10/2020.

Judith Horn (2019): Mit 300 Rädern: VAG startet im Mai mit neuem Verleihsystem. Edited by Nordbayern.de. Available online at https://www.nordbayern.de/region/nuernberg/mit-300-radern-vag-startet-im-mai-mit-neuem-verleihsystem-1.8801979, updated on 4/14/2019, checked on 6/10/2020.

Jupyter Explorers (2019): Optimizing Bike Sharing Rentals. A Prediction Model for Bike Sharing Demand. University of Cologne - Information Systems for Sustainable Society.

nextbike GmbH (2020): nextbike Unternehmen. Available online at https://www.nextbike.de/de/unternehmen/, updated on 6/10/2020, checked on 6/10/2020.

Rodt, Stefan; Georgi, Birgit; Huckstein, Burkhard; Mönch, Lars; Herbener, Reinhard; Jahn, Helge et al. (2010): CO2-Emissionsminderung im Verkehr in Deutschland - Mögliche Maßnahmen und ihre Minderungspotenziale. Available online at https://www.umweltbundesamt.de/sites/default/files/medien/461/publikationen/3773.pdf, checked on 6/10/2020.

Schinke, Boris; Harmeling, Sven; Schwarz, Rixa; Kreft, Sönke; Treber, Manfred; Bals, Christoph (2010): Globaler Klimawandel: Ursachen, Folgen, Handlungsmöglichkeiten. Dritte überarbeitete Ausgabe 2010. Available online at https://germanwatch.org/sites/germanwatch.org/files/publication/1186.pdf, checked on 6/10/2020.

Scikit-Learn (2020): scikit-learn 0.23.1 documentation. sklearn.neural_network.MLPClassifier. With assistance of David Cournapeau, Matthieu Brucher. Available online at https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html, updated on 6/10/2020, checked on 6/10/2020.

Shearer, Colin (2000): The CRISP-DM model: the new blueprint for data mining. In *Journal of Data Warehousing* (4). Available online at https://mineracaodedados.files.wordpress.com/2012/04/the-crisp-dm-model-the-new-blueprint-for-data-mining-shearer-colin.pdf, checked on 6/10/2020.

Stadt Nürnberg (2019): Umweltdaten Nürnberg: Archiv. With assistance of Harald Bauer. Edited by Stadt Nürnberg. Umweltdaten Nürnberg. Available online at http://umweltdaten.nuernberg.de/wetterdaten/messstation-nuernberg-flugfeld/archiv.html, updated on 6/10/2020, checked on 6/10/2020.

tensorflow.org (2020): Keras Optimizer RMSprop. Available online at https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/RMSprop, updated on 6/3/2020, checked on 6/10/2020.

Verbeek, Marno (2018): A guide to modern econometrics. Fifth edition. Hoboken, NJ: Wiley Custom.

# Appendix - Graphics

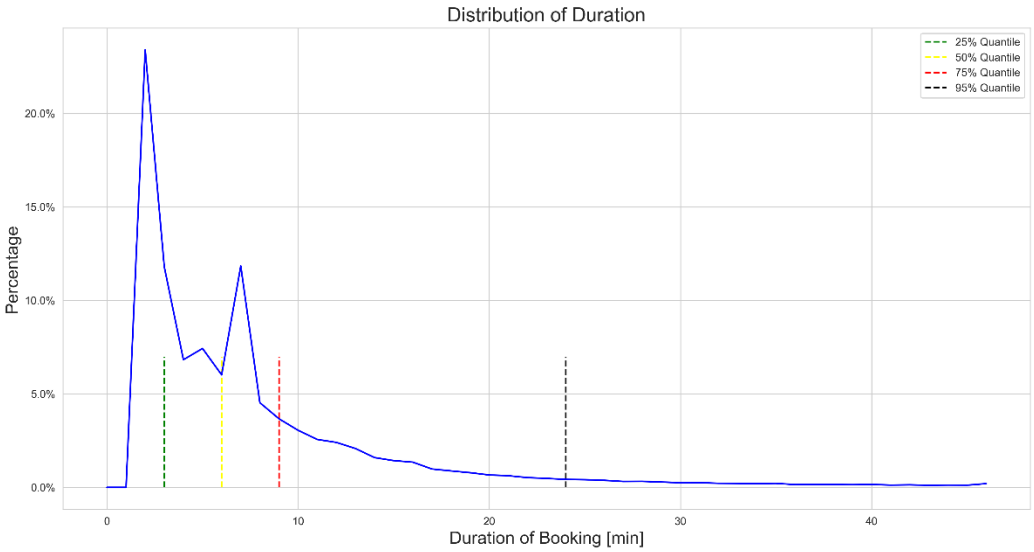



*Figure 1: Distribution of trip duration on the 90% quantile*

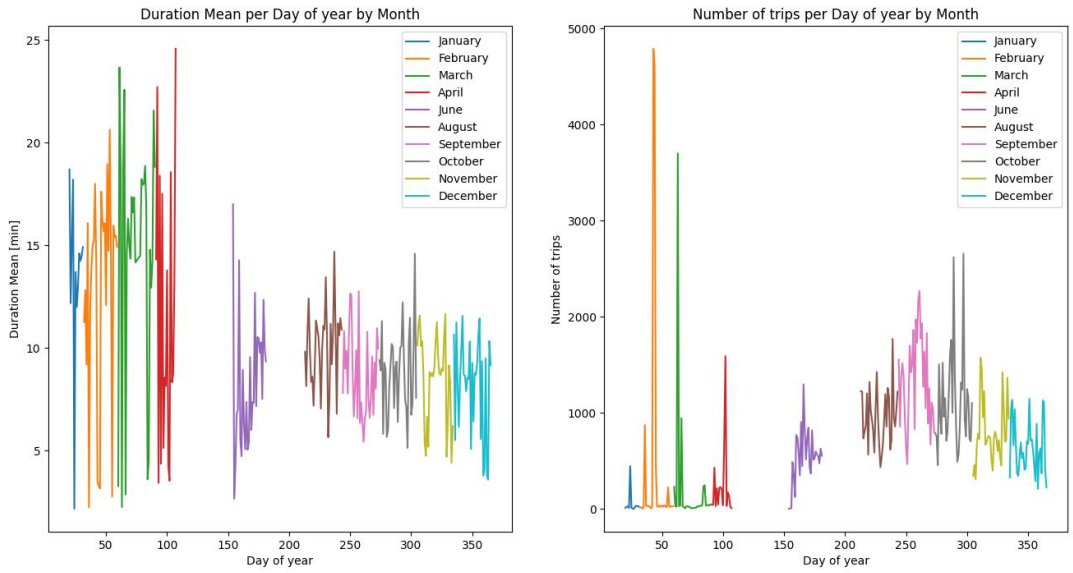



*Figure 2: Mean Duration of trips per day of year*

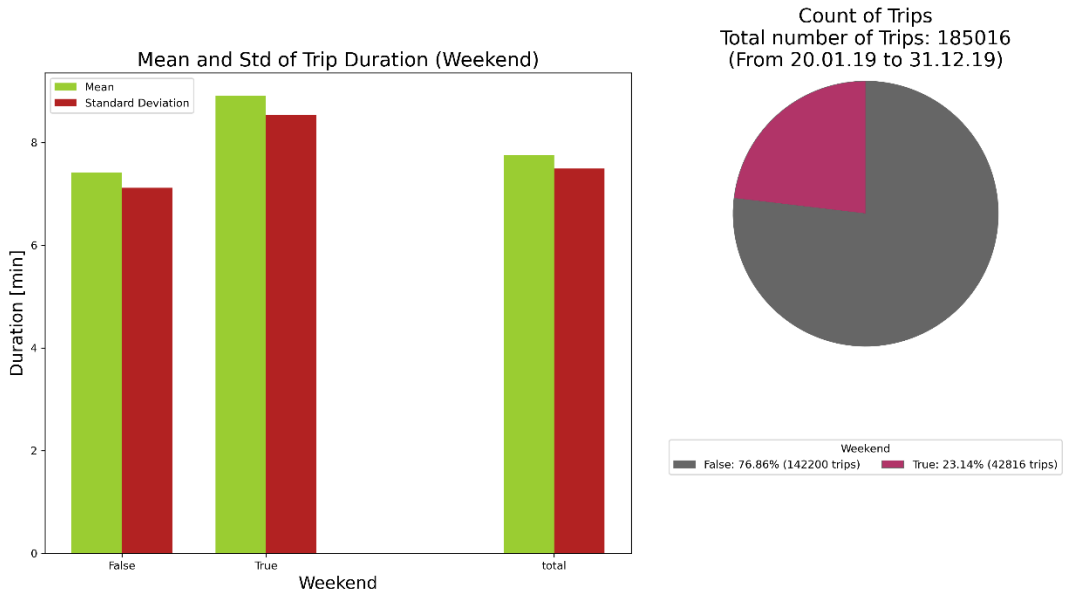



*Figure 3: Mean and standard deviation of trip duration (left). Distribution of trips on weekends and workings days (right).*

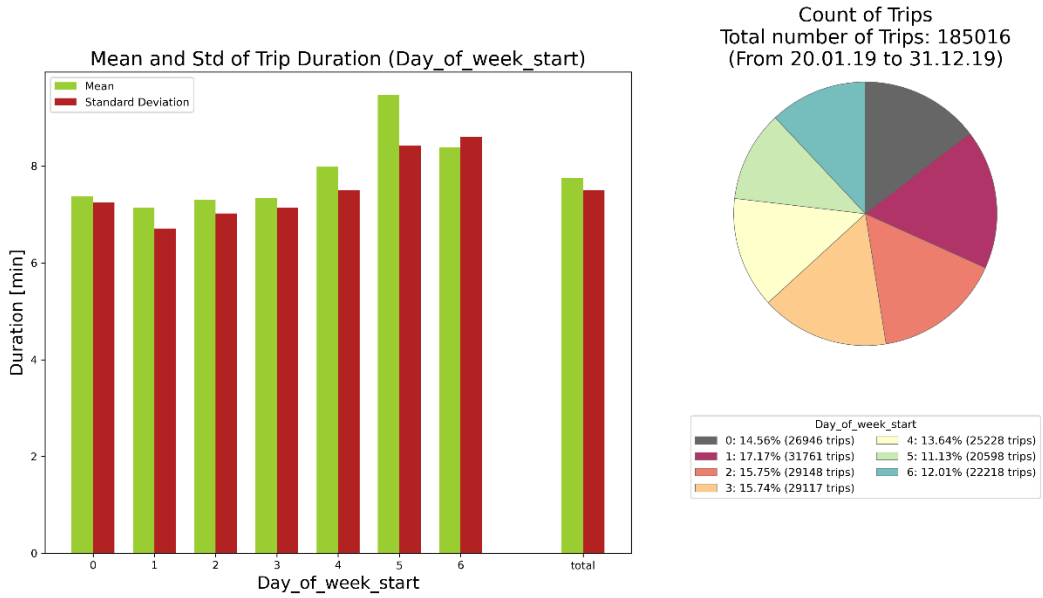



*Figure 4: Mean and standard deviation of trip duration for every day of week (left). Distribution of trips for every day of week (right).*

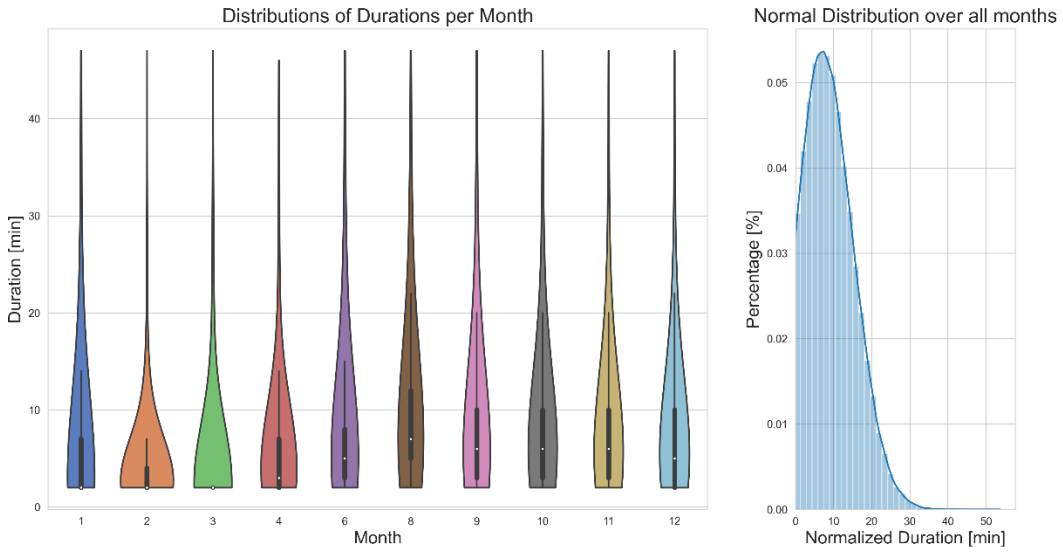



*Figure 5: Trip duration distribution per month and normalized over a year*

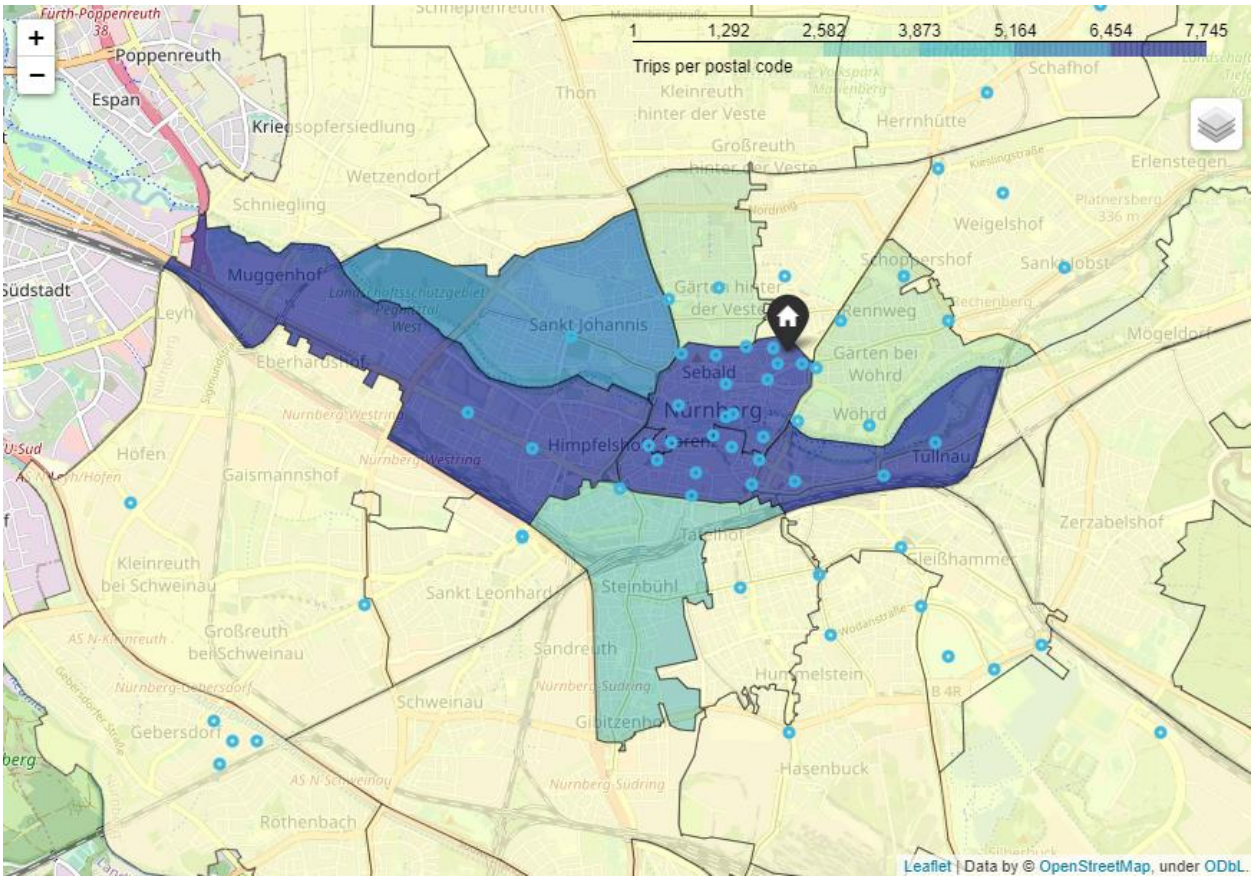



*Figure 6: Month with most trips (August) for each postal code area*

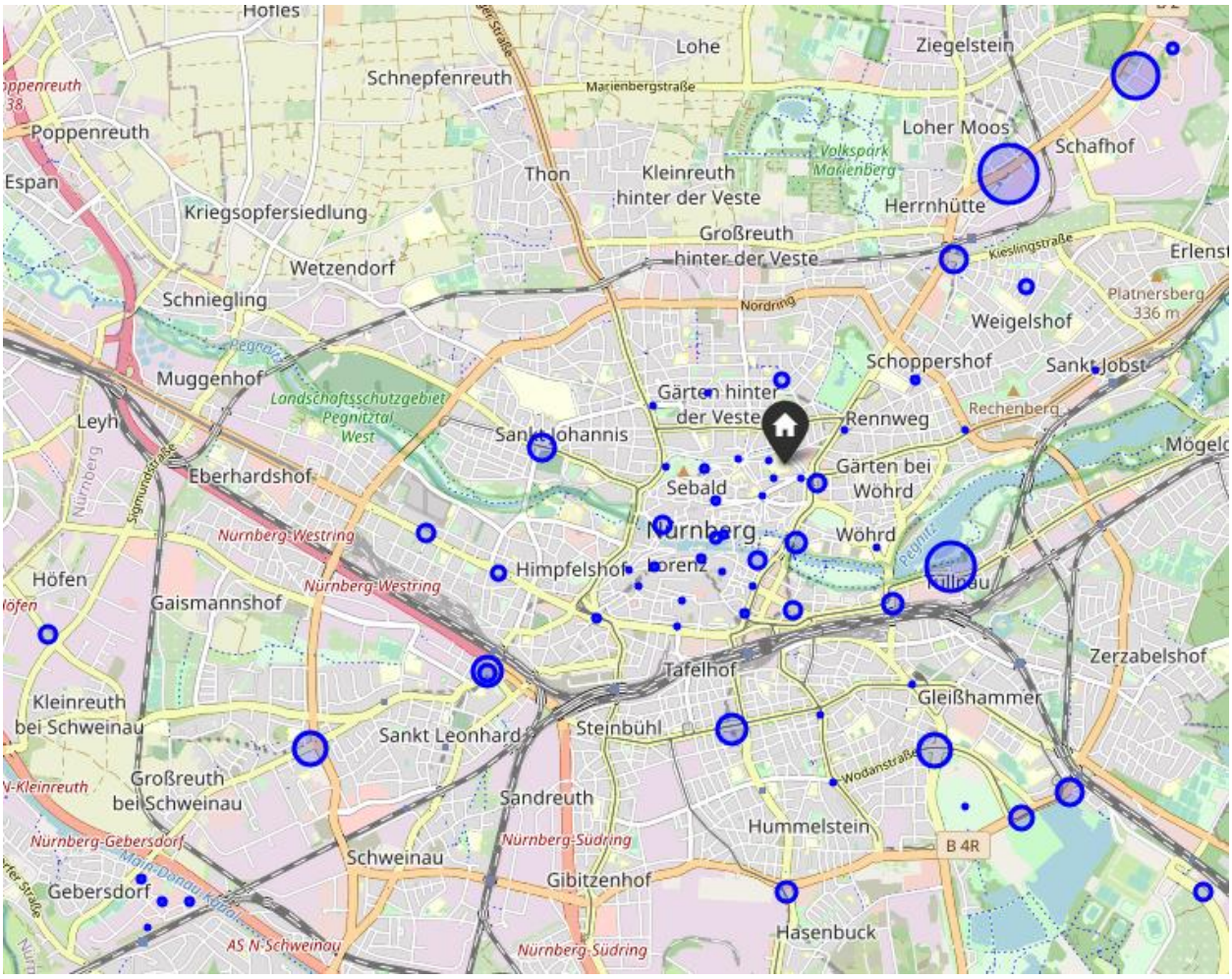



*Figure 7: Fixed stations and bikes at stations (01.10.2019 08:00)*

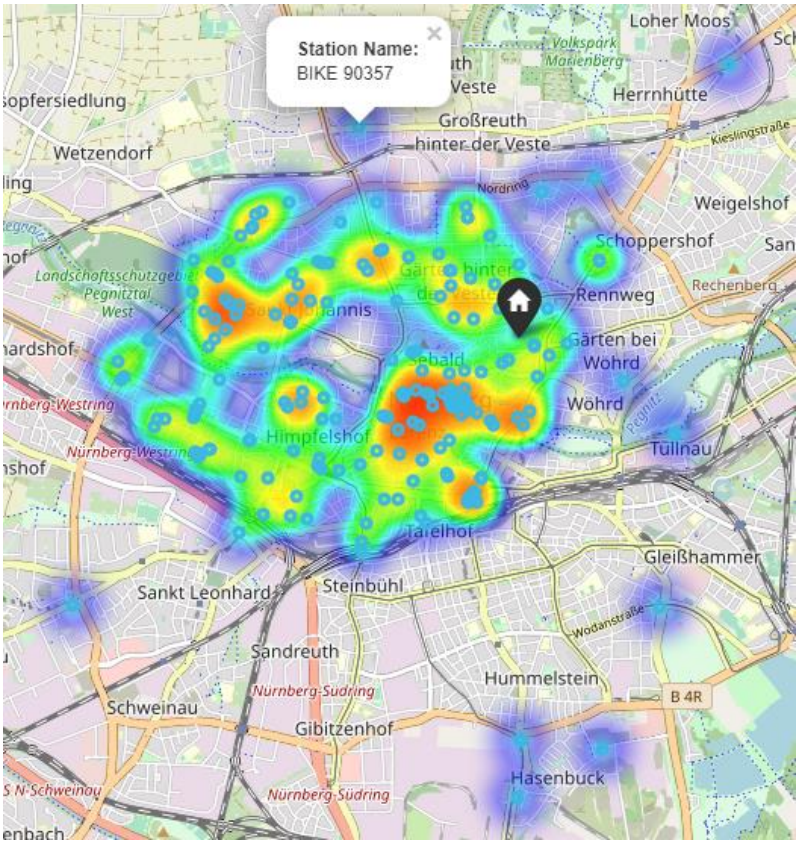



*Figure 8: Heatmap of trip ends at Christmas (24.12.20109)*

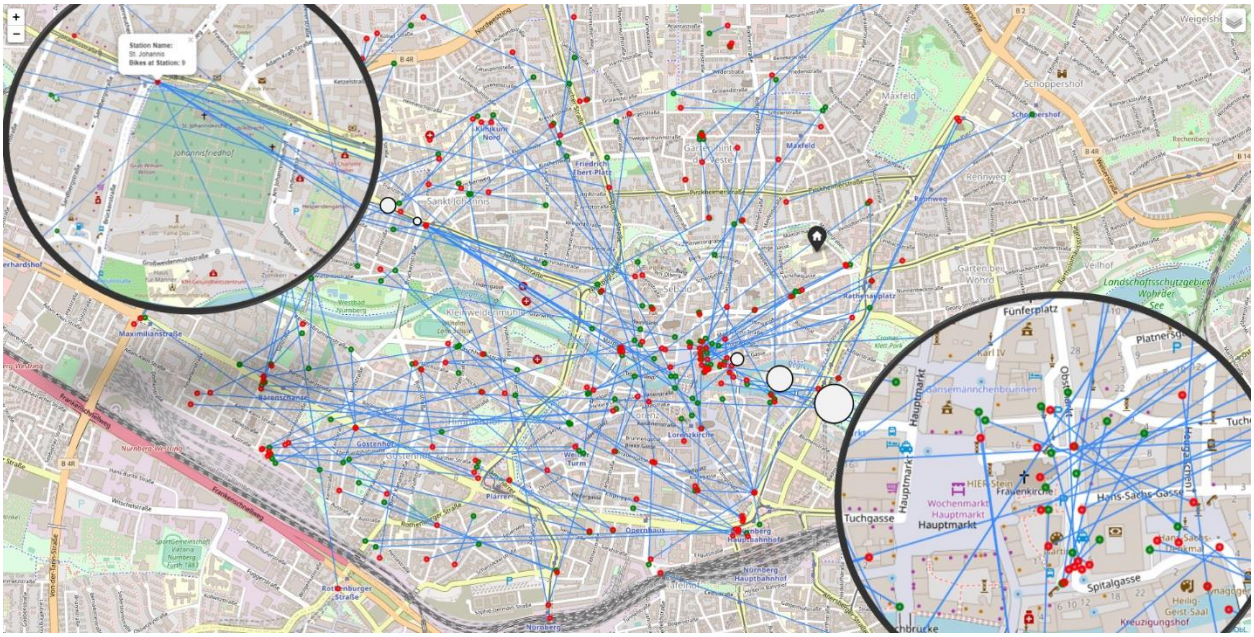


*Figure 9: Visualized trips on the 24th December 2019 with start and end point*

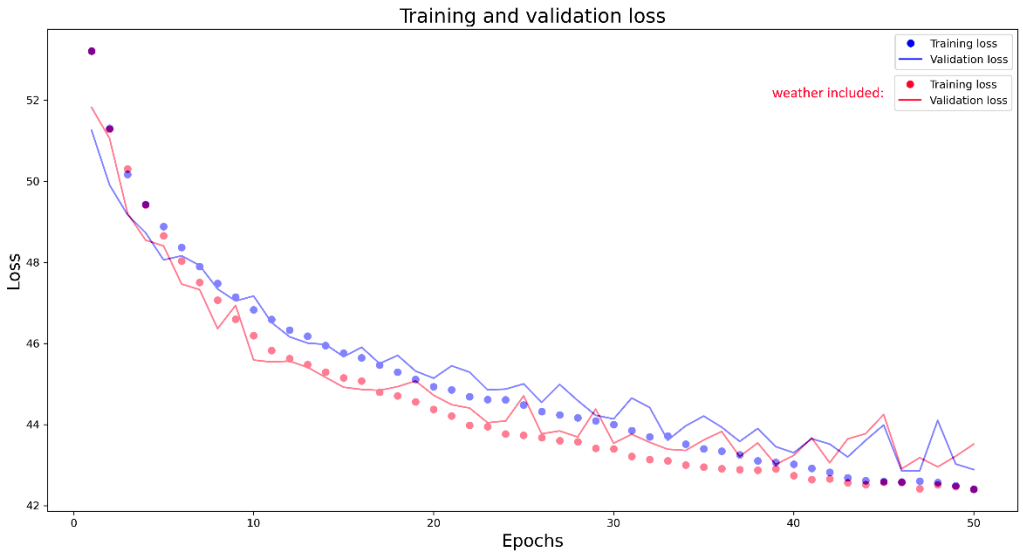



*Figure 10: Training and validation loss of NN compared without and with weather data*

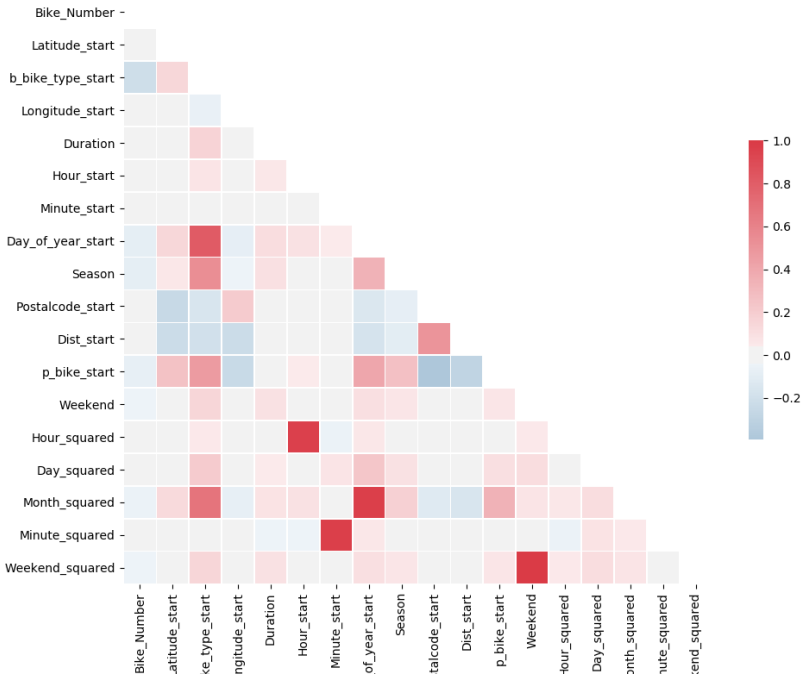



*Figure 11: Correlation Matrix*

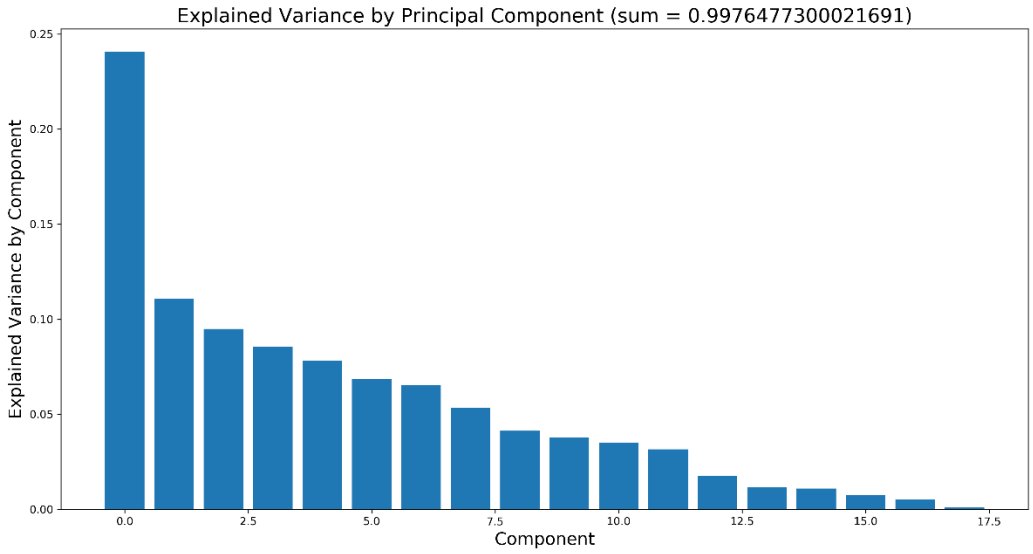



*Figure 12: PCA Analysis of components*

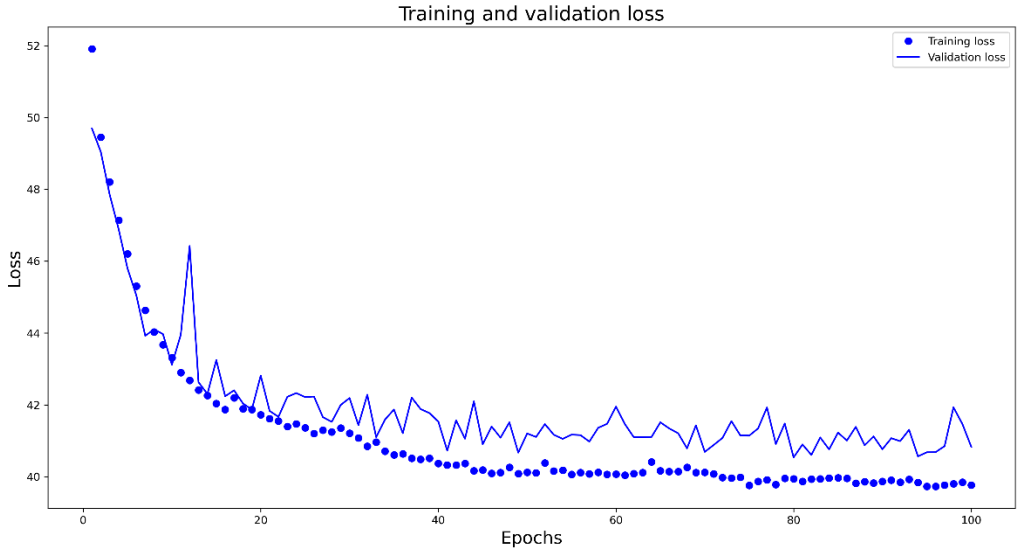



*Figure 13: Training and validation loss of Neural Network trained in 100 epochs*


*Figure 14: Residuals of Neural Network prediction on validation set*