DSML Group Work

**Introduction**

During this project we have developed an analysis for the German Enterprise Nextbike GmbH, which works in bicycle rental all over the world.

This report entails our findings and the work required to reach them based on a rental dataset from Nextbike between the 20 January 2019 and the 20 January 2020 and corresponding weather information distributed by the German Weather Service (DWD) for the cities of cologne and essen.

By analyzing the data in timesteps of hours we were able to produce results, with a meaning for real time business reactions.

**Data Preparation:**

The task of preparing data for analysis is always a quite difficult task, because of the effect of the decisions on the outcome of the analysis. Therefore we decided to use the maximum data possible from the rental data with only minimal cleaning. The dataset provided for the task entails all trips made within the year marked with the respective starting and endpoints in coordinates, the corresponding duration of the trip in the timedelta format and the time the trip was taken including the specific day. While we would have liked to use all data without cleaning, when checking the starting and finishing locations of the recorded trips, we found some data which is impossible to achieve physically, for example a trip from the United States of America to cologne in less than three hours. While it is certainly interesting how such data can be achieved, for our analysis this type of data produces errors and thus we dropped all trips with coordinates outside of a square from 50 to 52 latitude and 6 to 8 longitude.

For the weather data we decided upon, we had the goal in mind to produce a common weather indicator specifically tied to the bike rental of Nextbike. In this effort we decided upon a large quantity of different types of data. Air Temperature, Air Pressure, Cloud Coverage, Precipitation Amount, Form of Precipitation, relative Humidity, Soil Temperature, Sunshine duration and Wind velocity to name them shortly at this point. It should be obvious why we decided on aspects such as air temperature or the amount of precipitation. On the other hand air pressure and soil temperature are not as easy to connect to the task. Keeping in mind our analysis there will follow an explanation to the less self-explanatory datasets at a later point in this report.

The weather data while rather simple required some preparation for usage, because of the formats used to save date and time. For further reference we converted the timestamp from an Integer format to a split date and time format similar to the one in the rental data. (ipynb Data Preparation Cell 5) The overall quality and accuracy of the weather aspects we computed using the given quality bytes. (A IMG 1)

Furthermore, we added a column to the dataset containing the weekday as a String to simplify the comprehension of analytics for specific weekdays.

**Real Time Monitoring & Descriptive Analytics Report:**

Definition of Key Performance Indicators (KPI)

In this part of the project the focus is on analyzing the rental data to create indicators of the performance of the renting system with an hourly significance in each city. These should give, when compared to the same indicators computed with data of other enterprises or cities, a value showing a business advantage or disadvantage.

The first choice of a KPI is the percentage of used bikes. It is meant to show the performance of the fleet as a whole. For example if there are less bikes than needed then the fleet would not be as good as it can be. On the other hand, if there are far too many bikes available then potential income is lost and the risk that bikes might get broken is increased, thus capital is risked.

With this reasoning the computed percentage of used bikes for each hour of the year is the first KPI. It has an error though which is quite difficult to fix. The dataset only provides a number of bikes used throughout the year. There is no proof for how many bikes were really available for each hour. Thus, an approximation of the total amount of used bikes for the year is used as the size of the fleet. Compared to the amount of bikes, which are only used a single time the output is:

In a visualization of this KPI the following graph is generated:

Second KPI Utilization of bikes in relation with weather data

The visualization of the relative use of bikes of Cologne and Essen shows some sudden drops in overall usage of bikes in the interval of a year (Utilization\_Bikesin%\_CITY.pdf). One cause for those drops might be related to the weather condition on those specific days

With the help of the correlation coefficient, we determine how much relation exists between the weather data and utilization. First step is to create a data frame with the right columns. Then the data frame is sliced to the right index size to fit with the utilization data frame. Otherwise the resulting values are incorrect. Because of missing weather data from one of Cologne's weather stations, we were forced to use only one corr. matrix to represent Cologne. Both data frames are merged into one and plotted to give a correlation matrix. The first matrix for cologne and essen give the coefficient of the whole year ( corrMatrix\_CITY\_Year.pdf)  while the second one was restricted by a time interval  (corrMAtrix\_CITY\_Interval.pdf)

The result shows an overall small absolute value close to zero for weather datas with irregular inputs, which means this specific data gives us only a value when these related weather conditions appear (e.g. precipitation amount), otherwise it gives us zero. This phenomenon is explained by the fact that in our correlation function the zeros adulterate the results. On a day without rain utilization of bikes still fluctuates. On the opposite, weather data with regular inputs has a higher coefficient, because regular inputs means less adulterated comparisons. This can be seen by looking at the second matrix with the restricted period of time (corrMAtrix\_CITY\_Interval.pdf). The overall higher absolute values prove our hypothesis. There are also values, which were positive in the first but negative in the second matrix. This is related to the season of the year. The weather is different between for example summer and winter. Certain weather factors have a different impact on other weather data depending on the season, which also explains certain drops of values from the first matrix to the second. There are also coefficients with an absolute value close to zero without a recognizable change when the first and second matrix is compared. Those weather conditions have a low relation to the utilization and do not need to be as close examined as the other conditions. The dataframe for the  second matrix of both cities is not as perfectly cleaned as hoped for, which is seen by the relative low coefficient value (under 0.5), but the results show us, there exists a relation between utilization of bikes and certain weather conditions.

The third KPI is the distribution of usage of bikes throughout the city. It will show where the most bikes are needed and whether a redistribution is needed at any point in a day. It can also show commonly used places, which shows good points for rental stations. A visualization can provide us with any amount of business intelligence, while at the same time summing up the performance.

The next KPI analyzes how well Nextbike did in terms of revenue. This KPI is important for the obvious reason that any company needs to know how much it earns. Furthermore, it can also serve as a source of ideas for future improvements.

The revenue calculation is based on the Nextbike pricing model found here:<https://www.nextbike.de/de/preise/>. At the “Basistarif” it costs 1€ for every 30 minutes that you rent a bike with a maximum of 9€ per day. There is also an option to pay a monthly fee, but since we don’t have access to the Nextbike customer data, the simplifying assumption was made, that all rentals were charged on the Basistarif.

To get the most insights from this KPI, it was observed over different timespans, including:

* Total revenue generated per hour of the day over the whole period of time
* Average revenue per hour of the day generated on an average single day
* Total Revenue generated per hour per bike
* Total hourly revenue on workdays vs. weekends
* Total revenue generated per day of the week
* Total Revenue generated per month

To keep this report short, we will only focus on the most interesting findings and attach diagrams for the remaining revenue KPI’s to the appendix.

As expected, the total revenue generated in Cologne is much higher than in Essen due to the sheer difference in size of the two cities. To make the values more comparable, we looked at the average revenue generated per bike. But since the number of bikes in Essen (1365) is almost as high as in Cologne (1394), even in this comparison Cologne performs better by a factor of over 16 (Cologne: 906,78€/bike, Essen: 55,23€/bike, Appendix: CE\_average\_revenue\_per\_hour\_per\_bike). A possible explanation for this could be that the fleet in Essen got replaced with new bikes.

The other interesting finding has to do with the distribution of revenue over the day. Both cities have two revenue peaks during the day. A smaller one in the morning at 7am where a lot of people have to get to work. And a bigger one in the afternoon at 3pm where people get home from work, go shopping or may use a bike for free time activities. What’s remarkable is that while in Essen these two peaks are almost at the same level, the peak in the afternoon in Cologne is much higher than in the morning (Appendix: CE\_total\_revenue\_per\_hour). In this difference lies an opportunity for improvement, because it means that Nextbike doesn’t tap into its full potential for rides in the morning in Cologne, while in Essen it doesn’t tap into its full potential in the afternoon. With marketing campaigns tailored specifically towards these different market segments, they could improve their overall revenue.

The final KPI shows the distribution of the starting and endpoints of the trips. It can show the amount of work that is needed from Nextbike to always have the perfect distribution, meaning enough bikes in each borough or district to satisfy all customers. This KPI will be computed as a list for each city, which decreases its comparability. What can be compared though is the standard deviation or the standard variance of the lists. Here the rule is the smaller the deviation the better the distribution performance. Furthermore a visual comparison of the differences ist possible as well. In the map XXX is shown the distribution of the renting of bikes at 9 o’clock on the 23423432432. The following map shows the distribution of the return of the same bikes. The greatest difference can be seen in the borough of “Poll” where there was a drop of bikes available. In comparison the map XXX shows the renting for the same date and time in Essen. A great difference that can be observed is that in Cologne the main demand is close to the center of the city, while in Essen there are two centers of demand. The first center in Essen is similar to the one in cologne in the heart of the city slightly to the north. In the very south there is the second center.

Another visualization shows the difference between the number of bikes that started and ended in a borough for one hour of a day as a mean value over the year. The image XXX entails the maps for Cologne and the image YYY for Essen. These maps highlight the boroughs that are most likely to be out of bikes without some redistribution by Nextbike and show where a great number of bikes are piled up. If these values are equalizing over the day then it could be that there is never a need to redistribute the bikes.

If the deviations for Cologne and Essen are compared now, they show a slightly better performance of the fleet in Cologne, as Cologne has a deviation of 3.197621055337373 and Essen has one of 8.918119531309129. This value can be heavily influenced if there are hours where the number of bikes is very low because then the percentage of bikes changing the borough can grow faster. The Histograms in X and Y show that in Essen the value is strongly influenced by the extreme values at -30 and 50.

Taking into account all of these KPI’s will create an image of the performance of the individual fleets of Cologne and Essen. The result can show advisable actions that can be performed by Nextbike to improve the performance. For example the KPI showing percentage of usage and revenue clearly state that the fleet in Essen is not performing good enough for the number of bikes given. Furthermore the KPI concerning the distribution shows that in Essen the everyday maintenance work is more than in Cologne. Overall, improving the performance of the fleet in Essen is advisable.

**Predictive Analytics Report**

The goal of this predictive analysis is to give an accurate prediction model for the hourly rental bike demand for the city of Cologne and Essen. The prediction shall be made on the basis of the input values described in the previous chapter.

Starting with the report, a new column named “numOfRentedBikes”, which basically aggregates all the rows for a specific hour at a day, where a bike is rented, was added.

Following the supporting hypothesis: People are using the bikes to get to work and to get home from work. That prediction was mapped that by the columns “hour”, which is simply the corresponding hour compared to the dependent variable and the column “weekday” or rather “is weekday”, which are mapping the work schedules of the Next Bike customers, who usually work between 8 to 5.

The second aspect of the hypothesis was the weather. More bicycles are generally rented when:  
it is not raining/the precipitation amount is low, the weather is warm, it is not windy and the weather is generally pleasant. This was mapped by the weather data, which we mentioned above.

Summarized, the independent variables are: air pressure, air temperature, cloud coverage, wind velocity, precipitation amount, relative humidity, weekday, isWeekend and hour - as an input for the regression during every hour of the observed time span from 2019-20-01 to 2020-20-01 on the cities of Cologne and Essen.

**On the subject of feature selection and feature extraction:**

Since we are only working with a handful of features (9 pieces) and have enough data samples to cover our regression models there is no need for dimensionality reduction. Regardless, it was still checked. “dayofweek” and “isweekend” are obviously highly correlated and some features like “windVelocity” had low variance.

For even more information about the data the technique of feature extraction was used. It resulted in a variable for: the hour of rent (“hour”, the day of the week (“dayoftheweek”) and whether the date was on the weekend or not (isWeekend). These features could all be extracted from the timestamp provided from the initial data set of the bike rental data.

Starting with the regression selection: with that in mind and with the datasets available through weather prediction, the rental network could predict its demand through the following regressions we chose. Moreover, since the input data was labeled, supervised learning was applied.

Because there are many regressions which were examined in the lectures, a benchmark, on which the models that come in question could be compared, was necessary. A relative error metric as a loss function was not acceptable, since it is not a metric for comparing different regression models. So the loss function was formulated as, the Mean Average Error (MAE) and the lowest Root Mean Squared Error (RMSE). With these it is simple to compare all regressions and choose the best one, by just picking the lowest error.

*What regressions were picked and why?*

Not counting in the unambiguously better error values from the loss functions, we are starting with the random forests. Here is what made it better than other regressions. In addition, the simplicity of the implementation of the regression algorithms was examined.

Except in the Artificial Neural Network (ANN) we used the Scikit-learn library, which is probably the most useful library for machine learning in Python, plus the methods we were taught in the workshops.

**1. Random forests**

Random Forests are training N trees in parallel and predicting, via majority vote, between tree outputs Random forests are an ensemble method that combines multiple decision trees for classification or regression, where the term “forest” refers to multiple trees, i.e., multiple learners (typically N = 100 to 1000) Predictions are made via a mean computation (regression)

**Strengths**: It generates an internal unbiased estimate of the generalization error as the forest building progresses, it has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing, it is one of the most accurate learning algorithms available. For many data sets, it produces a highly accurate classifier, it runs efficiently on large databases.

it can handle thousands of input variables without variable deletion.

**Drawbacks**: RFs are no longer easy to interpret and do not perform well on small data sets, random forests have been observed to overfit for some datasets with noisy classification/regression tasks, for data including categorical variables with different number of levels, random forests are biased in favor of those attributes with more levels. Therefore, the variable importance scores from random forest are not reliable for this type of data.

(source:<https://towardsdatascience.com/random-forest-and-its-implementation-71824ced454f>, and lecture 8)

**2. Artificial neural network (ANN):**

**what are anns?**

**pros?**

**cons?**

**benny**

**3. L1 -Lasso - Regression:**

The formulation of the L1 algorithm trades off loss on the training set with a penalty on absolute values of the parameters. This type of regularization (L1) can lead to zero coefficients of particular features so LASSO also helps in feature selection. The problem is that the Lasso algorithm can be biased, because the selection is arbitrary in nature - LASSO will select only one feature from a group of correlated features. In this task this might be not decisive, but in the future, if there were more features than 9, it would be because it ignores nonsignificant variables that may, nevertheless, be interesting or important.

.

However, in this task it was the 3rd most accurate regression.

**Why not the other regressions?**

Why not KNN regression?

Since KNN is a nonparametric regression algorithm and we have enough data samples to define a functional relationship between input features and the target value, we did perform better by not averaging the mean of the k- nearest neighbours. The MAE/RMSE did also show worse performance than the other 3 algorithms we chose for this task.

Why not Linear Regression/Polynomial Regression/ RBF/ridge/decision trees/ etc.?

At long last, the other regressions did not make it into the top 3 regressions because they were simply not performing well enough on the given data set and the selected features.

*To the subject Training of the models:*

First of all we divided the dataset into 50% for the training set, 20% for the holdout set and 30% for the test set. Later we switched into 70% for the test and validation set and 30% for the test data. (?xxx)

Teilung in 50, 20, 30.

Random forest und lasso regression erzielen bessere werte mit großerem trainings set von 70%. Außerdem vermeidet lasso regression sowieso over, under fitting. Deswegen keine holdout set nötig

Then we created a multidimensional x-vector as an input for our independent variables and ran the regression against our dependent variable. We mainly used scikit and the methods we were taught in the workshops.

*ANN: how did we train it? - benny*

**What do the coefficients in your regression model mean?**

As a general rule, if the coefficient value is zero or near zero, it means that the corresponding feature can be dropped. Air pressure with the value of 0.23680493 for Cologne and -0.00754672 for Essen seems to be insignificant in this context, as the multiple linear regression showed. (nachweis aus dem workbook) On the other hand, all other coefficients had a value above 0.5, so they correlate with the prediction from the regression that was done.

**The following results were achieved:**

The first best performing algorithm:

|  |  |  |
| --- | --- | --- |
| **ANN** | Essen | Cologne |
| RMSE |  | 32,0107 |
| MAE |  | 22,2160 |

The second best performing:

|  |  |  |
| --- | --- | --- |
| **Random forest** | Essen | Cologne |
| RMSE | 4.175551016130484 | 33.405665387009165 |
| MAE | 3.0715896439748285 | 23.626626563012735 |

The third best performing:

|  |  |  |
| --- | --- | --- |
| **L1 - Lasso - Regu.** | Essen | Cologne |
| RMSE | 4.395851626717146 | 37.21070119868697 |
| MAE | 3.2361923269025863 | 27.749865209238273 |

(note: it was found out that an alpha/lambda regularization value of 0.1 would be suitable for this regression)

The achieved error values through the loss functions  can be interpreted as following : e.g. there is a  MAE/RMSE of 23 for Cologne: by having the independent variables as an input for the x-vector, the number of rented bikes each hour can be forecast with an error of the given MAE or RMSE value.

**How well do the models perform? What are the scores?**

The data that was provided for Essen (circa 54000 samples) performed “better” than the data from Cologne (circa 1 million samples), if only the pure numbers from the loss function are considered. So on average there is a **(xxx)** error value for Essen and a 24 error value for Cologne. Nevertheless it has to be considered that there are significantly more bikes rented than in Cologne in an one hour period than in Cologne.

That is furthermore the difference between Cologne and Essen. Starting at the population of both cities, where Cologne has nearly double the amount of rented bikes, making Cologne much more favorable for bike rentals. By looking at the average value of rented bikes in each city It becomes clear how the data should be viewed:

Cologne:

(977089bikes÷ 365d ÷24h)

≈ 112 average rented bikes per hour

*Average error:*

(22,2160+23.626626563012735+27.749865209238273)/ 3

≈ 24 avg. error

Essen:

(54165bikes ÷ 365d ÷ 24h)

≈ 6 average rented bikes per hour

*Average error:*

(.....)/ 3

≈ 3avg. error

[source: predictive analysis.ipynb, paragraph: 4. Material for report]

The city of Cologne has a deviation of exactly 20% per hour on rented bicycles, while Essen has a **(xxx%)**  average deviation per hour, which is in conclusion the result of the research task.

Overall the regression was more precise on the Cologne data set, as the results above have shown. Important to note is that it was not the regression but, the deviation can be traced back to the given data sets, where Cologne has nearly 20 times the amount of samples in comparison to Essen.

The more data samples one regression got, the better it performs.

**Which model would we select for deployment and why?**

The regression which would be best suited for deployment is the random forest regression. It may have a slightly higher error but is much simpler to implement. From a business perspective, if Next Bike would have the sufficient technical knowledge and personnel, the ANN would be recommended, because the loss is clearly smaller than the random forest regression.

Either way the task was found to be solvable and through enough corresponding data to the real time data monitoring of the Next Bike API, the bike rentals, an almost exact prediction can be made for Cologne but not for Essen, or at least one with a **(xxx)** difference.

**How could the selected model be improved further? What are the improvement levers of the model?**

The predictive power of the model could be further improved if the model was trained better, for example by increasing the amount of input data, e.g. a longer investigation period of the rental data instead of just one year.

Another aspect is the precision of the weather data. The DWD offers a quality byte for their measured values. This is classification of how could and precise the provided value is. A stronger emphasis on only very accurate weather information could affect the main precision of the model.

*Ann Benny hyperparameter*

Furthermore, if Next Bike could provide additional data, which would help to enhance the prediction power of the regression the predictions could be improved. (?)

**Are there any additional findings?**

The observation of the geographic coordinates for the start and finish point of the bike rents showed that there are some bikes that registered coordinates from outside germany. This is clearly a mistake and an error in the collected data. This should be fixed in future analyses.

The dataset division: it was interesting that a 70/30 ratio performed better than the 50/20/30 ratio we were shown in the lecture.

The data did not need to be standardized, at least for the L1 and random forest.

**Distribution of tasks:**

1. Data Collection and Preparation:

* Github repo: Creation and Maintenance (Marvin)
* Correcting Data Types for Input Features (Tolga)
* Weather data collection (Felix, Tolga, Joonseop)
* Aggregation of Bike Rental Data to hourly values and filling of empty hour entries with 0 for rented Bikes (Marvin)
* Deletion of empty rows without collected rental data (16-18.03.19) (Marvin)
* Merging of weather data to bike rental data (Marvin)
* Export of feature data frames (Tolga)
* Code Review and Improvement (Tolga)
* Delete rows with start or destination outside of defined coordinate range (Felix)

2. Real-time monitoring and descriptive analytics (Felix,Julian,Joonseop):

* KPI Selection (Felix, Joonseop, Julian)
* Percentage of used bikes (Felix)
* Utilization of bikes in relation with weather data (Joonseop)
* Revenue KPI (Julian)
* KPI to distribution of start and endpoints of trips (Felix)

3. Predictive Analytics (Marvin,Tolga,Benny):

* Research on suitable Regression Models (Tolga)
* Split of data sets in training/holdout/test (Tolga)
* Feature extraction (Marvin)
* Lasso Regression: Standardization of Input features (Marvin)
* Lasso Regression: Creation (Tolga)
* Lasso Regression: Lambda fitting (Marvin)
* Visualizations: Deviation, Loss functions (Marvin)
* ANN: Creation and fitting (Benny)
* Code Review and Improvement (Tolga)

APPENDIX

Reference Image 1)



Reference Image 1)



Reference Image 1)



Reference Image 1)



Reference Image 1)



Reference Image 1)

