User manual

Installation the package

- First download the package or clone it from GitHub.
- Then open the Anaconda prompt.
- Navigate into the folder (using cd).
- Install the package by entering the command “’pip install -e . “’.

Start the package

- After installing the package enter “’nextbike “’ into the console.

Manual use the package

After the package is started a menu appears in the console. In this menu it can be chosen between the following items, by entering the corresponding number into the console <1-3>.

1 – Use default data to create all Dataframes -> Recreate all Dataframes necessary for later visualizations or predictions. This method uses the default .csv as input. This method will take a few minutes.

2 – Use new data to create all Dataframes-> Recreate all Dataframes necessary for later visualizations or predictions. This method uses a custom .csv as input. Use this method for predict or train on new data. This method may take a few minutes, depending on the file size.

3 – Use saved Dataframes -> Loads the Dataframes, created in the last rebuild. This method is quite fast.

After the creating of the Dataframes is completed, the main menu appears. It can be chosen between the following items, by entering the corresponding number into the console <1-4>. Please note that visualization is only possible on the default data set. It does not matter whether this was newly created or loaded. If it is an custom data set, only options 2-4 will be displayed.

1 – Visualize -> Go into visualization menu.

2 – Predict -> Go into prediction menu.

3 – Go Back -> Go back to the previous menu.

4 – End -> Close the program.

First lets take a look on the visualization menu. It can be chosen between the following visualizations, by entering the corresponding number into the console <1-7>.

1 – Trip duration -> Shows multiple visualizations about the mean trip length in minutes and the standard deviation of trips lengths in two separate windows. To go one close these windows.

2 – Number of Trips -> Shows multiple visualizations about the number of trips in different time units and compared with a normal distribution. For this again 2 windows open. To go one close these windows.

3 – Start/End point of Trips -> Shows a map on which trips are assigned to postal code areas. There is the choice between start and end points and the month of interest. To go one close these windows.

4 – Bikes per Station -> Shows graphic and a map about the number of bikes per station on a single point in time. The point in time is freely selectable. To go one close these windows.

5 – Weather data -> A graphic about the temperatures and precipitation in Marburg 2019. To go one close this window.

6 – Heat map -> Opens a heatmap, which shows the connection between trips and events in Marburg.

7 – Go back -> Go back into main menu.

Now lets take a look on the prediction menu. It can be chosen between the following predictions, by entering the corresponding number into the console <1-4>.

1 – Trip duration -> Here the user has the choice between Training or Prediction of the Dataframes, loaded at the beginning. When the train option is chosen a evaluations is shown at the end.

2 – Direction of trips -> Here the user has the choice between Training or Prediction of the Dataframes, loaded at the beginning. When the train option is chosen a evaluations is shown at the end.

3 – Number of trips -> Here the user has the choice between Training or Prediction of the Dataframes, loaded at the beginning. When the train option is chosen a evaluations is shown at the end.

4 – Go back -> Go back into main menu.

A sample use case:

Train the model for trip duration prediction with the default data and predict the holdout set:

Use default data to create all Dataframes -> Predict -> Trip duration -> Train -> Go back -> Go Back -> Use new data to create all Dataframes -> Predict -> Trip duration -> Predict

Automated use of the package

Besides the possibility to use the user interface, the packet can be used automatically with 3 different commands. This makes it possible to e.g. combine the package with another interface.

nextbike -- transform <filename.csv> -> Transforms the given file into the trip format, necessary for later predictions. The result is saved in the output folder.

nextbike --train <filename.csv> -> Trains the model on the given file. If no transform is given, the algorithm will be trained on the saved data.

nextbike --predict <filename.csv> -> Make a prediction based on the data in the file. The result is saved in the output folder.

Example that trains the model on old data and transforms and predicts on new data: ```nextbike --train inputData.csv --transform marburg_test.csv --predict marburg_test.csv```