

USER MANUAL

INSTALLING THE PACKAGE

1. Download the package or clone it from GitHub.
2. Open the Anaconda prompt and navigate into the downloaded folder (using cd).
3. Install the package by entering the command `pip install -e .`.

START THE PACKAGE

- After installing the package enter `nextbike` into the console.

MANUAL USE OF THE PACKAGE

After the package is started, a menu appears in the console. In this menu you can choose between the following options, by entering the corresponding number <1-3>.

1 – Use default data to create all Dataframes -> Recreate all Dataframes necessary for later visualizations or predictions. It 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 to predict or train on new data. This method may take a few minutes, depending on the file size.

3 – Use saved Dataframes -> Load the Dataframes, which you created and saved before.

After the creation 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 you upload a 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.

VISUALIZATION MENU

1 – Trip duration -> Shows multiple visualizations covering the mean trip length in minutes and the standard deviation of trips lengths in two separate windows. To go on, you must close both windows.

2 – Number of Trips -> Shows multiple visualizations regarding the number of trips in different time intervals and compared with a normal distribution. To go on, you must close both 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 on, you must close those windows.

4 – Bikes per Station -> Shows a graphic and a map describing the number of bikes per station on a single point in time. The point in time is freely selectable. To go on, you must close those windows.

5 – Weather data -> Shows the temperatures and precipitation in Marburg 2019. To go on, you must 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.

PREDICTION MENU

1 – Trip duration -> The user has the choice between train or predict trip durations based on the previous loaded data. When the train option is chosen an evaluation is shown at the end.

2 – Direction of trips -> The user has the choice between train or predict trip directions based on the previous loaded data. When the train option is chosen an evaluation is shown at the end.

3 – Number of trips -> The user has the choice between train or predict the number of trips based on the previous loaded data. When the train option is chosen an evaluation is shown at the end.

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

A SAMPLE USE CASE

Train the model for predicting the trip duration 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 PACKET

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

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 a model on old data and transforms and predicts on new data:

```
```nextbike --train inputData.csv --transform marburg_test.csv --predict marburg_test.csv```
```