Semester Project – Nextbike

Nextbike is an operator of shared bikes (both free-floating and station-based). They operate in many cities in Germany and around the world. You may know them as the operator of KVB bikes in Cologne. Your data set consists of trip data of Nextbike's bikes in one German city for most of 2019. Note that the data set has not been (extensively) preprocessed. Therefore, there might be missing data and there might even be false values/measurements hidden in the data.

Based your data set, following tasks should be carried out and included in your presentations and reports:

Tasks

1 Exploration and Description

a) The data set shows columns with prefixes p and b. What do you think do they represent? Also try to find good assumptions for the meanings of the columns

b) The trip column in your data set shows different values. Explain why there are not only two. Are examples with certain values for trip more informative for the analysis of mobility patterns than others?

c) Based on the given data, create a new DataFrame that stores (at least) the following trip information ("trip format"):

- **Bike Number**
- **Start Time** (Either as appropriate data type or as several columns from "Start Month" down to "Start Minute"),
- **Weekend** (binary),
- **Start Position** (Either as appropriate data type or as two columns for Longitude and Latitude),
- **Duration**,
- **End Time** (see above),
- **End Position** (see above).
- If you find more columns helpful for later analyses, that's fine – but explain why you added more! Save the new DataFrame to a file. Make sure that the whole routine of transformation is reproducible, i.e, given a new file in the same format as the original data set, you should be able to automatically create and save such a new DataFrame.

d) Calculate the aggregate statistics (i.e., mean and standard deviation) for the trip duration per month, per day of week, and per hour of day. Are there visible differences between weekdays and weekends?

These exploration tasks are the minimum requirement. Investigate further wherever it makes sense.

2 Visualization

a) For the summer month (i.e., June, August, or September) with most trips, visualize the number of started trips per PLZ region (you'll have to find geo data for that yourselves!) in a map.

b) For one moment in time, visualize the number of bikes at fixed stations meaningfully.

c) Create a heatmap based on an interesting aspect of the data, e.g., end locations of trips shortly before the start of a major public event.

d) Visualize the distribution of trip lengths per month. Compare the distributions to normal distributions with mean and standard deviation as calculated before (1.d))

These visualizations are the minimum requirement. Use more visualizations wherever it makes sense.

3 Prediction

a) Predict the journey duration. You may use any Start information in the data set. Also consider creating new features that may help prediction quality (Hint: In the coding week, we discussed creating polynomial features. How would you engineer features for periodic quantities with known period lengths, e.g, a day, a year, …?).
b) All cities in your datasets have universities. Based on Start information, predict whether a trip will be towards, or away from the university. Analyze your predictive performance on different subsets of your data (months, …). Do you see differences?

You my other predictive targets as you please.

Repository/Code

- You will be required to create a GitHub repository for the project. You may fork https://github.com/IS3UniCologne/PDS_Project as a starting point.
    - Use issues, pull requests, and other features of GitHub's workflow.
    - Create a Python package, i.e., not just scripts (see lecture). If run in a virtual (conda) environment with only jupyter installed, installation of your own package (see example notebook) should be sufficient (no other installations beyond your own package should be necessary). Create meaningful submodules in your package.
    - No functions or complex data types should be defined in your notebooks.
    - Comment your code and provide sufficient documentation for an outsider to use your work.
    - Create a command-line interface for your machine learning algorithm (Prediction). It should provide at least three functions:
        - `<cli_name> train` should train an ML model based on data (in the format that data was provided to you) in a defined location and store the model.
        - `<cli_name> predict new_data.csv` should load a previously trained model and predict journey duration and direction (see above) based on Start information given in new_data.csv (Provided in the same format as other data provided to you). Predictions should be saved to disk.
        - `<cli_name> transform new_data.csv` should transform a data set in the provided format to a data set in the "trip format" and save it to disk (see 1 c)).

Assessment

The course has 5 deliverables. The first is due May 4th and consists of a written timeline (max. 1 page) of your team's project. The written timeline is mandatory but does (if submitted) not affect the overall grade. The remaining 4 blocks have weights as follows:

- Code 30%
    - Clean, performant, well-documented, reproducible
- Repository 20%
    - Utilization of GitHub's features, structure as mandated
- Report 30%
    - Scientific style
    - Explanation of key steps in the project (you may use the tasks as guidance)
    - Meaningful visualizations
- Presentation 20%
    - Persuasiveness
    - Visual appeal

If there are any questions, contact philipp.kienscherf@wiso.uni-koeln.de.