HSL Bike Planner

Jesse Hantula, Joel Pietiläinen, Lene Bierstedt

1. Introduction

Helsinki's city bike system, operated by HSL, is the most popular one in the world (Sweco, 2019). Nevertheless, city bike users often face a lack of available bikes at the stations, which has become the greatest point of dissatisfaction with the system according to a 2022 survey (HSL, 2022). HSL and third-party providers (Cityfillarit, 2023) offer the ability for users to check the real-time availability of city bikes across stations in Helsinki and Espoo. However, there is no way of knowing in advance what the availability of a station will look like. It thus becomes hard to plan journeys in advance, and users might refrain from using city bikes on time-sensitive journeys like the morning commute to work. This increases dissatisfaction with the city bike system, and discourages the use of city bikes in favor of other means of public transport or even private car use, less climate- and health-friendly choices.

We decided to tackle this problem through our project 'HSL Bike Planner'. Based on city bike usage data from HSL and weather data from the Finnish Meteorological Institute, we predict the busy-ness levels - a proxy for bike availability - of each city bike station in Helsinki and Espoo every day for the next 5 days. Our website displays these predictions through an interactive map. Hence, users are able to get advance information on how busy stations are, which enables them to plan their journeys ahead of time to ensure that bikes will be available when they travel. This improves people's daily journeys, and reduces dissatisfaction with the city bike system, helping Helsinki retain the best such system in the world.

This report first elaborates on our motivation for solving this problem and the desired objectives for, and added value created by, our solution (section 2) Then, we comment on data privacy and ethical issues (section 3) and delve into our data collection, preprocessing and exploratory data analysis (section 4). Next, we explain our learning task and approach (section 5) and finally the communication of results and visualizations (section 6) before concluding (section 7). Note that all steps from sections 4, 5 and 6 can be viewed in our GitHub repository with code and comments.

2. Motivation and Added Value

As city bike users ourselves, we have often been frustrated with the fact that it was not possible to know in the evening whether we could bike to university the next morning. Hence, we decided to investigate this issue and found that this is a problem for other city bike users too, as described in the introduction. Our project targets all city bike users in Helsinki and Espoo and enables them greater insight into city bike availability and thus better journey planning and reduced frustration. Moreover, using bikes instead of personal cars or even other public transport like buses is a better choice for both the environment and personal health. Hence, there are additional indirect benefits to our application beyond tackling the problem of bike availability we set out to solve.

After finding data on city bike usage from HSL (described further in section 4), we saw that it was hardly possible to predict the exact number of bikes that would be available at a station on a given day. This is because the data included information on arrivals and departures, but not on how many bikes were available at the start and end of the day. Since bike stations are not filled up with bikes every night, and it is possible to park more bikes at a station than it has capacity for, we could not use the capacity as a

baseline for bike availability at the start of the day either. Instead of predicting the exact availability, we thus decided to predict “busy-ness”, a function of how many departures and returns there are per day at a station (see section 5). This serves as a proxy for availability - for example, if a station is very busy, you are unlikely to find available bikes, but if it is not busy, you have a good chance. We also decided to include the weather as a variable in our prediction of busyness, as we assumed based on personal experience that bike stations would be busier on warm, sunny days compared to cold, rainy ones. In order to provide users with the ability to plan their journeys plenty of time ahead, we decided to not only offer predictions for the next day, but for the next five days. We chose five days, since weather forecasts beyond that might not be very accurate, and it encompasses the working week, enabling people to for example plan all their commutes at once. Based on the motivation and objectives outlined here, we began to refine our approach and implement our solution, as described in the next sections.

3. Data Privacy and Ethical Considerations

Since our data does not include any personal data, and our predictions are not about people and cannot be used against people in discriminatory ways, there are no data privacy or ethical considerations for our work. All our data is publicly available under the Creative Commons license, whose terms we follow - we are not using the data for commercial purposes, and are crediting the sources. This also applies to the fmiopendata Python library for fetching weather forecast data.

4. Data collection, preprocessing, exploratory data analysis

Our data collection consisted of downloading the HSL Origin-Destination data for city bikes, available for 2016-2021 as csv files (HSL, 2023). Since this data did not include coordinates of the city bike stations in Helsinki and Espoo, which we needed for creating a map on our website, we also downloaded a dataset on the city bike stations via Helsinki Region Infoshare (HRI, 2021). Moreover, we downloaded weather data for 2016-2021 from the Finnish Meteorological Institute as csv files (Ilmatieteenlaitos, 2023). As we were able to easily find these datasets which included all necessary information for our purposes, our data collection was quite simple.

Then, we began with preprocessing the collected data. The first step was to work with the HSL Origin-Destination data. The data included one csv file per year, with data on each bike trip taken during the year, including departure and return times, stations, trip distance and duration. After initial processing, we transformed the data into a dataframe with the departure and return counts per station per day, and saved it as a csv file. This meant that once we had processed the data, we did not need to run the code again, as we realized that working with large datasets like this was quite time-consuming. We processed the bike station data later on, during the preparation for map creation. This involved dropping all columns except the Finnish station name and the coordinates, and merging it with the dataframe of finalized predictions by station.

Secondly, we combined the processed HSL data and the weather data. The latter included a csv file for each year with information on the date (columns for year, month, day and timezone), as well as precipitation amount, air temperature, maximum and minimum temperature. We loaded these files into one dataframe. After merging the two dataframes by date, the last step was to add columns that could improve model performance. We initially only added a Weekend column where the row had value 0 if a

day was a weekday and 1 for a weekend. Later, we decided to investigate the effect of adding the average departure and return counts per station per month for up to the last three years. This required us to deal with missing values, as some stations were only added during later years and thus did not have any past years to calculate averages from. We decided to fill these missing values with the average of the average departure and return counts for all other stations for the relevant month. Due to the size of datasets, it was challenging to deal with these missing values and to check if our calculation of averages was successful. After this data processing, we also saved the resulting dataframe to csv.

Finally, we also plotted our data throughout this process, some final plots being available in our GitHub repository. The plots supported our assumption that there are more departures with higher temperature, as well as with less precipitation. We also found that there are more departures in summer months, and depending on the station, a day being on a weekend or a weekday also influences the departure counts. We also identified throughout working with the data that some stations like those at Rautatientori are the busiest. Overall, we roughly followed the ideas we had for these steps of our data analysis defined in the canvas, but as we got familiar with the data, the necessary steps became much clearer, and we were able to do some experimenting, such as adding the averages to the data, which ended up improving our model, as discussed next.

5. Learning task and approach

As outlined in sections 1 and 2 of this report, our learning task is to predict the departure and return counts - from which we calculate busyness - for each station, every day for the next 5 days, based on the weather forecast for those days. To achieve this, we first learn a model from the historical HSL and weather data, into which we can then feed current data to get the new predictions. Since we are trying to predict the dependent variables departure and return counts based on various independent variables, our learning task is a regression problem. We first tried to create a linear regression model, which ended up predicting some negative departure and return counts. After researching regression models, we realized that poisson regression is better suited to our needs of analyzing and predicting count data.

Our input (independent) variables are the date (day, month, year), weather data (precipitation, temperature), whether it is a weekend/weekday and 3 year average departure and return counts. We found that the averages improved our model, and the other variables' influence seemed to match the analysis of our plots from section 4. To better understand the variables' impact, we also created a function to test the prediction outcomes with fictitious input values. After learning some initial models, we realized that reducing the timeframe of input data from six to three years (but not any more) reduced model error. To remain within a reasonable project scope, we decided to leave out some variables mentioned in the canvas, like time of day or proximity to points of interest, but it would be interesting to try including them in the future. Another takeaway for the future is that an extensive initial data analysis of potential variables could have saved us some time with later experimenting for model optimization.

Finally, a note on the dependent variable: Since we wanted to give users information on busyness, rather than exact departure or return count, we created a formula for busyness turning the predicted departure and return counts and turned them into an ordinal variable: Not Busy, Moderately Busy, Quite Busy, Very Busy, or Extremely Busy. The formula goes as follows:

$$\frac{\frac{Departure\ count}{Last\ 3\ year\ average\ departures}+\frac{Return\ count}{Last\ 3\ year\ average\ returns}}{2} + \frac{(Departure\ count - Return\ count)}{\frac{(Last\ 3\ year\ average\ departures + Last\ year\ 3\ average\ returns)}{2}}$$

If the formula returns a value of 1, the station is averagely busy, below 1 is less busy and above 1 busier. We assigned ranges to each of the qualitative indicators, and converted the predictions accordingly. One downside of our way of calculating busyness and its relation to availability was already touched upon in section 2 - we are not including the real-time fill of stations. So we are assuming that if a station is not busy, bikes will be available. Yet, it could be that a station is empty, and there are no returns. Then, no bikes would be available. However, including this additional information would have drastically increased predictions in complexity, going beyond the scope of our project. Also, if some bikes were returned to the empty station during the day, and all of them used, the busyness would be average. Also, our predictions are based on historical use data, not last-day data. Hence, this downside seems acceptable, but future work could investigate if there is an approach combining the two.

Next, for our learning approach and implementation we used the scikit-learn PoissonRegressor, with a normal train-test split. We also selected mean squared error (MSE) as our model performance measure. Since we are predicting busyness (departures and returns) for each station, each station has its own MSE value. Hence, we cannot give one value to describe the entirety of our model. We instead looked at a subset of stations to check if MSE was going up or down depending on the choices of input variables. After creating the models, we saved them to a pickle file. The models are then used to do the predictions every day for the next 5 days. For that, we use the fmiopendata library to retrieve the current weather forecast. Figuring out how to set up these functionalities was also interesting. The predictions are then made by inputting to the models for each station the weather data, plus the up-to-date information about the next 5 days for the other variables. Since there is no bike trip data for 2022 available yet, inputs for the 2023 predictions include the 2019-2021 departure and return averages. The predictions made are saved and fed into the functions that create our visualization, discussed next. One thing to note is that since the city bike season lasts from April to October, predictions are only meant to be made during those months, since there is no use for them otherwise and no average values exist for the other months as inputs. However, we exceptionally added data for November 2023 based on October averages, to enable course instructors to view up-to-date predictions on our website even in November.

6. Communication of Results and Visualizations

Since we want users to be able to easily access our predictions, we decided to create a website that users can load from their desktop or mobile phone. We chose to create an interactive map on the website: It contains a map of Helsinki and Espoo with pins marking each city bike station. Users can zoom in and scroll to find their desired station. Upon clicking a station pin, a popup with the predictions is displayed. This includes a list with dates and the corresponding busyness values, which are appropriately color-coded for easy identification. The scale of the busyness values is explained at the top of the page. One feature that could be added in the future is the ability to search a particular bike station, and/or, since people may not know the name of a station, the ability to search the map for general locations. Nevertheless, our current design is very intuitive and easy to use, enabling users to quickly gain the desired information, and realizing the added value for the end-user, and hopefully as a consequence also positive climate- and health outcomes, described in section 2.

In order to create this map, we needed the coordinate of bike stations from the bike station data mentioned in section 3. After merging these with the final predictions, we transformed the dataframe for further use. This involved creating a dictionary, which included the Station name, the busyness score and matching color-codings for the next 5 days. This is then used to create the map using the Folium library, which enables the creation of custom HTML map files using Python. We then researched ways to publish this website, and found the easiest method to be using GitHub pages. We created a GitHub workflow to run our main.py script every night at 3:30 am to update the predictions, and commit these changes to publish our updated website through GitHub pages. Using these functionalities of GitHub was also new to us, but we overall found that we selected good tools for creating our website, which were simple to use for us without prior web development experience.

7. Conclusion

Overall, we found this project to be a great experience. Since we were able to implement our initial idea without major changes, it was a fairly straightforward process. Nevertheless, each of the steps described in this report involved a lot of learning and trying out new libraries and ways of working with data. We also worked well as a group: We worked during exercise classes and also often met up outside of class. We were thus able to discuss, code and debug together, as well as create the pitch and report. Hence, no one had a big burden of work on their own, but everybody was able to do small tasks on their own through a shared GitHub repository. In terms of our results, we managed to create a working website with predictions updating daily, which is simple to use. While there are of course various improvements that could be made both for the predictions themselves, for example adding additional variables, and our website, like implementing locations search, we did achieve our objectives for this project, and city bike users could start benefiting from our application at the start of the next city bike season.

8. Reference List

Cityfillarit (2023) *Helsinki city bikes*. Available at: https://www.cityfillarit.fi (Accessed 27 October 2023)

HRI (2021) *Helsinki Region Transport' s (HSL) City BIke Stations*. Available at: https://hri.fi/data/fi/dataset/hsl-n-kaupunkipyoraasemat (Accessed 27 October 2023)

HSL (2022) *City bike survey results: city bike users get their money's worth.* Available at: https://www.hsl.fi/en/hsl/news/news/2022/10/city-bike-survey-results-city-bike-users-get-their-moneys-worthksulleen (Accessed 27 October 2023)

HSL (2023) *Open Data*. Available at: https://www.hsl.fi/en/hsl/open-data (Accessed 27 October 2023)

Ilmatieteenlaitos (2023) *Download observations*. Available at: https://en.ilmatieteenlaitos.fi/download-observations (Accessed 27 October 2023)

Sweco (2019) S*weco designed the world's most popular city bike system for Helsinki – Tampere is next*. Available at: https://www.sweco.fi/en/insight/press-releases /sweco-designed-the-wor lds-most-popular-city-bike-system- for-helsinki-tampere-is-next/ (Accessed 27 October 2023)

9. Appendix

Website: https://jessehantula.github.io/data_science_project.github.io/

GitHub Repository: https://github.com/JesseHantula/data_science_project.github.io/tree/main