Predicting city-bike departures based on time and weather observations

April 10, 2022

1 Introduction

Helsinki city-bikes are a popular means of transportation during the spring, summer and autumn seasons in the city. However, an usual problem seems to be that some bike-stations are either empty of them, or overflowing with them. This calls for a better optimized redistribution of bikes from other stations depending on time and weather, which means one needs to be able to predict peak times of departures from specific stations. This work does just that, by training two different models (polynomial regression and multilayer perceptron) on a combination of city-bike departure and weather data for a specific bike-station in downtown Helsinki. The structure of the report can be seen in the following table of contents section.

Contents

2 Problem Formulation

The Helsinki Regional Transport Authority's (HSL) city bikes are a convenient method of transportation during the spring-autumn seasons (April-October). One problem with them, however, is that when you need them the most the nearest station is empty of them and when you need to return one the nearest station has no empty spaces. This is a result of the popularity of the bikes which leads to natural hot-spots of departures and returns at specific bike stations depending on the time of day, date and, probably, weather.

This work attempts to predict the number of departures at a specific bike station (Kaisaniemenpuisto) based on time, day and weather, to give an indication of whether the station needs more bikes. The data is acquired from two sources: The city bike data is accessible at the HSL website [3] and the weather data is accessible at the Finnish Meteorological Institute (FMI) website [2]. Both sources publish their open data under the Creative Commons BY 4.0 International Licence.

The city-bike data consists of data points that hold information regarding individual trips with a bike. One data point holds the timestamp of departure, the timestamp of return, the name and id of the station of departure, the name and id of the station of return, the covered distance with the bike, and the duration of the whole trip. In this project we are mainly interested in modifying this data into a format in which we can see the number of departures and number of returns at specific dates and times of day for one station (Kaisaniemenpuisto). The weather data consists of observations at specific hours of a day, regarding various weather measurements. One data point holds the date, hour, precipitation amount, precipitation intensity and the air temperature. These data observations are made at the Kaisaniemi station, in downtown Helsinki.

The **labels** in the data are going to be in some form the number of departures at the chosen bike station. The **features** are going to be the day of week, hour of day and precipitation intensity. Two methods are presented in this work, in which different subsets of the aforementioned features are used. In the first method, a polynomial regression model, only the hour of day is used as a feature, while in the second model, a multilayer perceptron model, the hour of day, weekday classifier and precipitation intensity are used as features.

For this project, data sets spanning the years 2016-2021 were downloaded. Further data preprocessing is needed to match the weather and bike data observation times, as well as to obtain the number of departures and returns at the specific bike station.

3 Methods

3.1 The Data

The data is a combination of the Helsinki Regional Transport Authority's city bike interaction data for Kaisaniemenpuisto station, and weather observations from the Finnish Meteorogical Institute's Kaisaniemi station, from May to October, over a span of 5 years (2016-2021). These two data sets were combined, with the relevant features and labels obtained. The resulting data set has 21 228 data

points including the following columns: Year, month, day, hour, date, weekday, weekday classifier, departures, returns, normalized departures, normalized returns, air temperature, rain intensity and rain intensity class. The number of data points and specific labels/features are specified further for both methods in subsections 3.2 and 3.3.

Year and **month** are self explanatory (integers), **day** specifies the number of the day in the current month, **hour** accounts for a whole hour of the day (e.g. 23 = 23:00-00:00), **date** holds the number of the day and the month together, **weekday** specifies which day of the week it is (Mon-Sun = 0-6), the **weekday classifier** specifies if it is a workday or weekend (0 or 1), **departures** and **returns** are the total number of departures and returns from the bike station during the specified hour, **normalized departures and returns** are the number of departures and returns normalized with respect to their annual maxima, **air temperature** is measured in degrees Celsius, **rain intensity** is measured in mm/h, and the **rain intensity classifier** categorizes the rain intensity into light rain (0-2.5 mm/h), moderate rain (2.5-7.6 mm/h) and heavy rain (>7.6 mm/h). A more detailed look of the data can be found at this link [1].

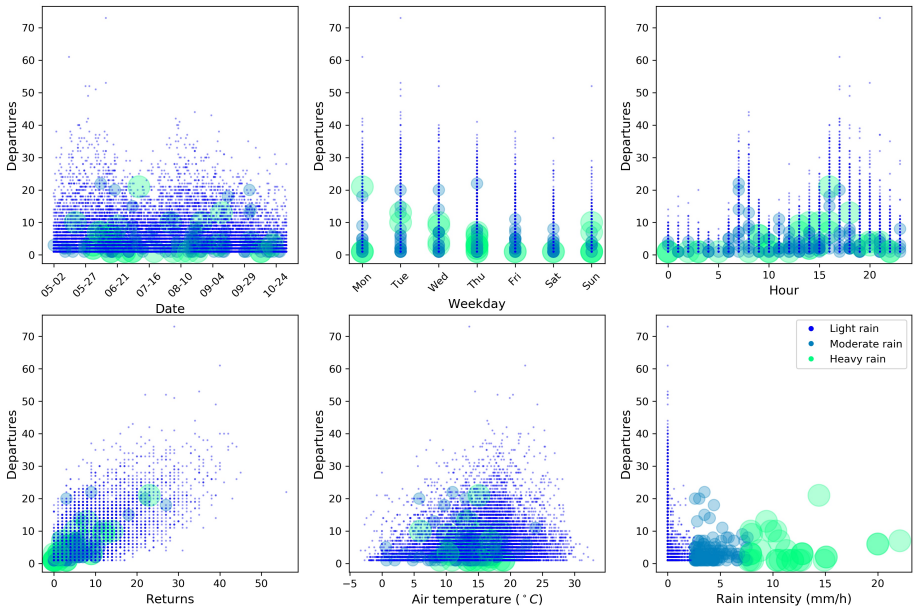

Figure 1: The number of departures plotted against relevant features

A summary of the data is visualized in Fig. 1, with some transparency to get a better picture of the density of data points. From these plots we can see that the day of the week, hour of the day, and rain intensity clearly have some effect on the departure trends. E.g. in the upper right plot we can see that the upper contour fluctuates strongly depending on the hour, with three peaks situated at 1am, 7-8am and 16-18pm. This makes our selection of features (weekday, hour and rain intensity) justified. However, we will consider different features depending on the specific machine learning method used.

The reason to why normalized departures and returns with respect to their annual maxima were introduced, was the large fluctuation in city bike usage throughout the years. We want to measure

the popularity of the city bikes overall, but to take advantage of the full data set we need to see to it that the data is on common ground for each year. If we consider Fig. 4 in the appendix section A.2 for example, we can see that in 2016, 2020 and 2021 the mean number of departures from the station was a lot smaller than for 2017-2019. This is simply because the city bikes were in their infancy in 2016, and the Covid-19 pandemic hit at the end of 2019, both phenomena resulting in low usage of the bikes.

3.2 Method 1 - Polynomial Regression

For the first method, polynomial regression was chosen to predict the worst case scenario of bicycle departures from the Kaisaniemi station. The "worst case", in this context would refer to the upper contour of the departures plotted against the hour of the day in the upper right plot of Fig. 1. Due to the regular, albeit non-linear form of these upper contours, polynomial regression should work fine to model them. This choice of method means the loss function to be minimized is the average (or mean) squared error. The label and feature for this specific model are the normalized departures and hour of day, respectively. The rest of the features, as proposed in the problem formulation, i.e. the air temperature and precipitation intensity will be used in the second method.

To be able to model the worst case scenario for the above plots, we need to filter out most of the data points below it. The ultimate worst case would be to select only the largest value for departures for each hour, but a single data point for each hour might result in severe distortion of the model due to outliers (e.g. a festival downtown which results in a peak of departures). For this project, the 20 largest departures were selected for each hour, resulting in a total of $20 \times 24 = 480$ data points for both workdays and weekends (960 data points). Increasing the number of data points for each hour would simply lower the fitted model vertically, while keeping the same form, but decreasing the number of data points might distort it. The data was separated into training, validation, and testing sets, as 60%, 30% and 10%, respectively. This splitting ensures a reliable amount for the training, while being able to validate our model and test our result at the end.

3.3 Method 2 - Multilayer Perceptron

In the second method, a multilayered perceptron regression model was chosen to predict the worst case scenario of bicycle departures, but this time with three features: the hour of day, the weekday classifier, and the precipitation intensity. This model choice is motivated by the problem still being one of regression, and the added complexity of the two new features. The optimized loss function was the squared error because that is the loss function used by `sklearn.neural_network.MLPRegressor`. A limited memory BFGS was used as the solver, due to the small amount of data, and the rectified linear unit function (ReLU) was used as the activation function due its effectiveness in a multilayered network. Several different combinations of layers and numbers of neurons were tested, but for the purposes of this project the investigation was quickly restricted so that 15 numbers of neurons were used in a varying number of hidden layers (this was a rabbit hole, indeed). Again, the 20 largest normalized departures were included in the final data, for each hour, both working days and weekends, as well as all precipitation intensities. This time resulting in a total of 1126 data points. The splitting of the data was the same as in the previous method.

4 Results

In addition to the splitting of the data, as explained in the previous section, K-fold model validation with $k = 5$ was also used on the data as a whole to ensure that the models presented in this chapter are overall good choices. But, this is beyond the scope of this project, which is why those figures are found in the appendix for those interested (App. A.4). The final chosen models were based on both the single splitting of the data, plots of which can be found in App. A.3, and the K-fold calculations. For the polynomial models, degrees of 10 and 7 were used for the workday and weekend models, respectively, while for the multilayered perceptron 7 hidden layers each with 15 neurons were used.

Figs. 2 and 3 show the final results for the polynomial, and multilayered perceptron models. On a quick glance, we can see that both models fit the data quite well. The polynomial models, that were trained separately on workday and weekend data, both capture the important highs and lows of the data. In the workdays we can clearly identify the morning and evening rush (6-9 and 16-20), and the weekend has an interesting rise in the middle of the night (party-people), and a stretched out hill from midday forward. The multilayered perceptron model, which is a single model for the combined workday and weekend data, also manages to capture the same highs and lows as the polynomial model with an additional advantage: Taking into account the rain. We can clearly see that the same rush hours are there, but that the overall number of departure sinks if there is rain. One oddity is apparent though, and it is the high peak in departures on the weekends (6-11) when it is raining. This can be explained by missing data points for rainy weekends at the corresponding time. The model seems to somehow infer this peak from the not so rainy working days, which is interesting.

The final training, validation and testing errors can be seen in Figs. 2 and 3. We can see that overall the errors are lower for the multilayered perceptron model, than for the polynomial models. Especially the final test errors are a lot closer to the validation error for the multilayered perceptron model, making it the model of choice. Nevertheless one needs to remember that these models are not perfectly comparable, due to the differences in the used data. Although, one could further argue that due to the added advantage of being able to predict with the help of weather, the multilayered perceptron is in the end the best method. Still, one has to take into account the one odd peak in the weekend predictions. Therefore, the final chosen model is the multilayered perceptron with 7 hidden layers, all with 15 neurons, with the caveat that further data is also required (hoping for rainy weekends).

5 Conclusion

In this work, we have investigated two different models (polynomial and multilayered perceptron) in predicting departures from a specific bicycle station (Kaisaniemi), based on both weather observations and time. The final conlusion for this particular work, was that a multilayered perceptron model was better equipped to predict the number of departures from the station, with an increased advantage in taking into account the weather. The results are satisfactory for the purposes of this project, with one interesting oddity in the multilayered perceptron (the weekend peak in the rain data of the model). This can be improved once further data is acquired. This work could be taken further by using the models in confluence with the live feed of the number of bicycles at the bike station. With this combination, one could recommend if bikes need to be redistributed to the station.

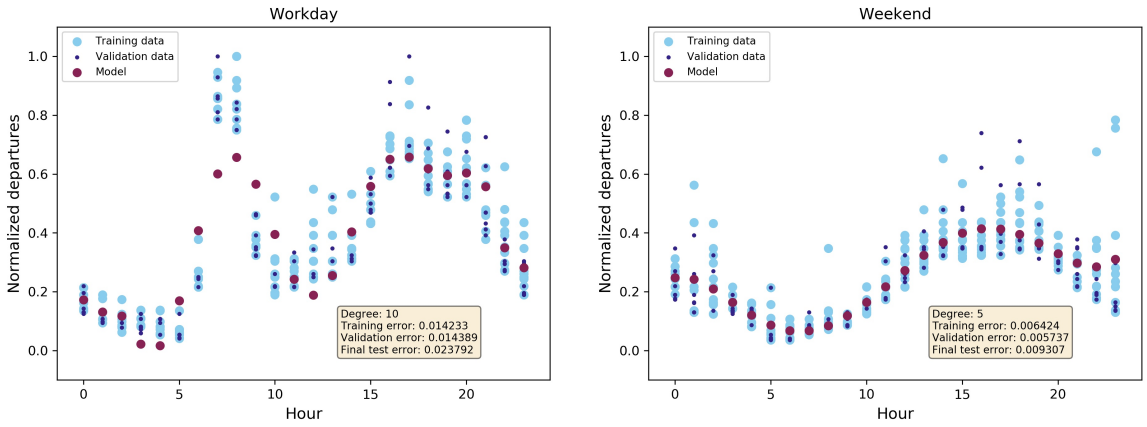

Figure 2: Resulting polynomial model, alongside with training and validation data, separated into workday and weekend plots.

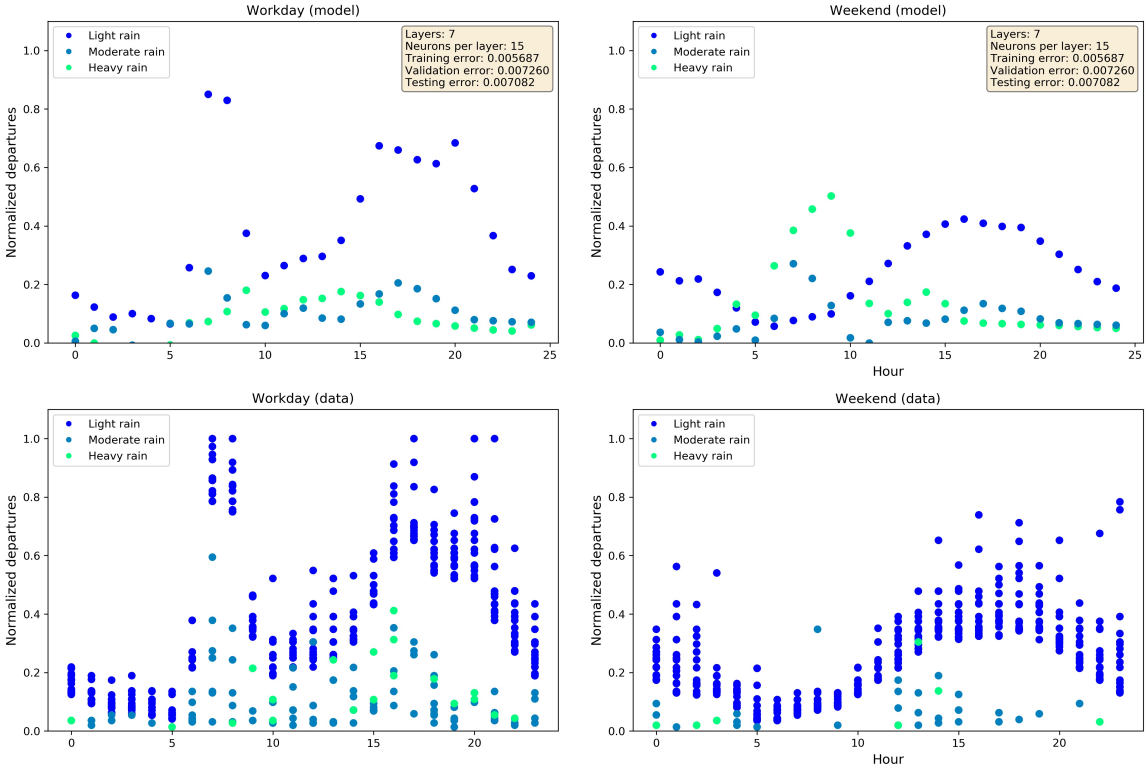

Figure 3: Resulting multilayer perceptron model (top), and the training and validation data (bottom), separated into workday and weekend plots. The training and validation data have not been further specified in the plot, to avoid messiness.

References

[1] CurlyNikolai. *City-bike ML*. "`https://github.com/CurlyNikolai/Citybike_ml`". 2022.

[2] Finnish Meteorological Institute. *Download Observations*. "`https://en.ilmatieteenlaitos.fi/download-observations`". 2021.

[3] Helsinki Regional Transport Authority. *Open data*. "`https://www.hsl.fi/en/hsl/open-data`". 2021.

A Appendix

A.1 The Code

The code can be found in its entirety at this link [1].

A.2 Annual Mean Departures

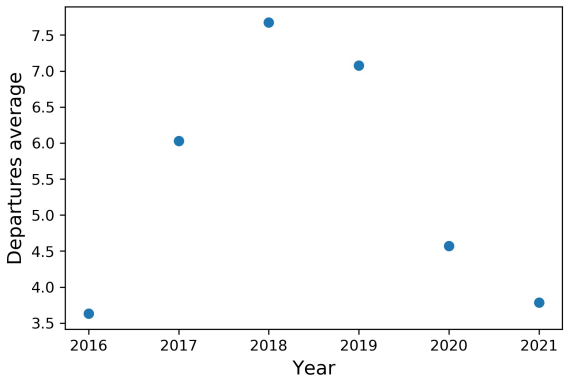

Figure 4: The average number of departures throughout the years. Note the drop in usage after 2019 when the Covid-19 pandemic hit.

A.3 Training and validation errors

Figure 5: Training and validation errors for polynomial models. We can see that for workdays and weekends polynomials of degrees 10 and 5, respectively, should be sufficient.

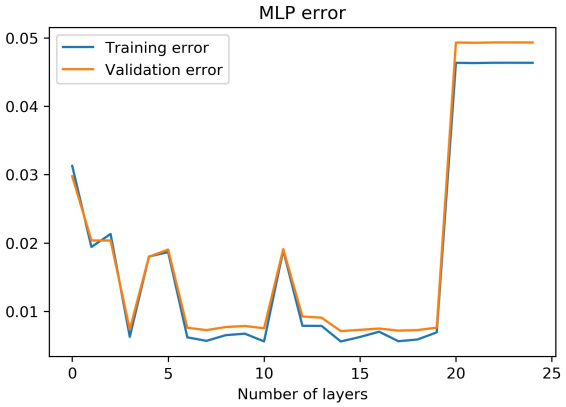

Figure 6: Training and validation errors for the multilayered perceptron regression models. We can see that for example 7 hidden layers should be sufficient.

A.4 K-fold reassurance

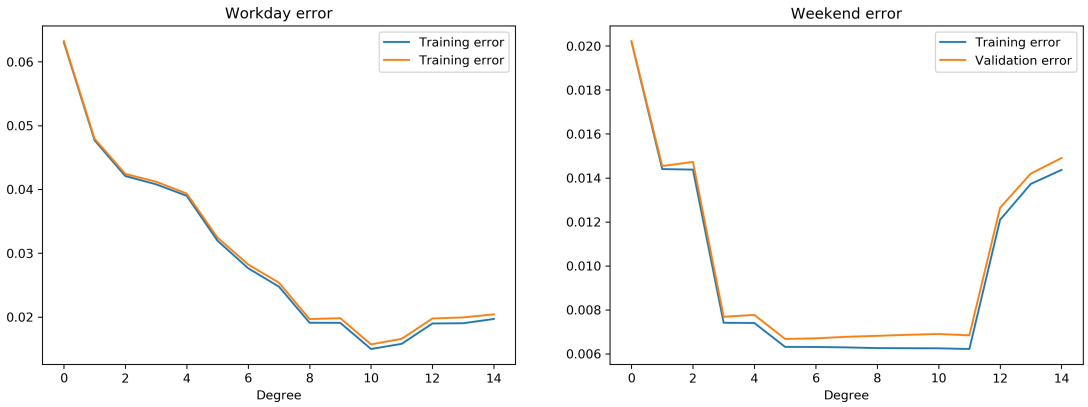

Figure 7: Average training and validation errors from KFolding ($k = 5$) for the polynomial models. We can see that the chosen degrees of 10 and 5 for the workdays and weekends, respectively, should be alright.

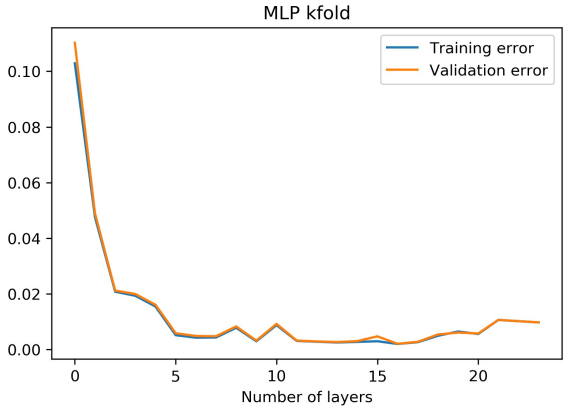

Figure 8: Average training and validation errors from KFolding ($k = 5$) for the multilayered perceptron regression models. We can see that the chosen number of layers 7 should be alright.