MIE1512H: Predicting Station Demand in Bike Share Systems

TANYA TANG

1 INTRODUCTION

Docking station-based bike share systems rely on rebalancing trips taken throughout the day to replenish empty stations. These rebalancing trips rely on alerts that are triggered when a station's inventory drops below a predetermined level which is calculated from estimated station demand. Thus, being able to accurately predict station demand is crucial to increasing the efficiency of each rebalancing trip and also satisfying customer demand. The paper "*Towards Station-Level Demand Prediction for Effective Rebalancing in Bike-Sharing Systems*" by Pierre Hulot and Daniel Aloise [1] defines and tests a model architecture for this problem of demand prediction; in this report, we will adopt a section of their model architecture to apply on public datasets from a bike share system in San Francisco.

There are three main data analysis objectives of this project, the first two of which are also objectives of the selected paper:

- Synthesize a set of features that contribute to station demand by connecting data from different datasets.
- Construct several prediction models trained on historical data to predict station demand.
- Test prediction models and compare the predicted bike trips per station across all prediction models.

The sections of this report correspond to each step in the CRISP-DM methodology. The **Data Understanding** section will provide an comprehensive overview of the public datasets used in this report, including why the particular dataset is being used, data sources, missing/corrupted values, redundant data, etc. Then, the techniques used to transform and link the different datasets will be explained in the **Data Preparation** section. Next, the **Modelling** section describes the model architecture built in this report and provides implementation details. Finally, the **Evaluation** section assesses the overall project by summarizing and analysing the process for correctness. The report closes with concluding remarks as well as several proposals for continuations of this work.

2 DATA UNDERSTANDING

2.1 Ford GoBike Trip Data

Bike trip data comes from the Ford GoBike system in San Francisco. The data is given in csv format and split into separate files per month. There are 8 relevant columns in this dataset: start time, end time, start station name, start station latitude, start station longitude, end station name, end station latitude, and end station longitude. Data in this format exists from January 2018 to January 2020. Each month contains roughly 100,000 to 200,000 bike trips leading to around 4,700,000 total trips. There exist missing values for start or end station names in some rows, as well as some start and end timestamps that do not contain a date, only time. After removing all of these corrupted entries, we are left with a full trip history containing around 3,600,000 total trips.

The features we want to extract from this dataset are the aggregated departing trip and aggregated arriving trip counts for each distinct station at each distinct time slot. Each time slot is characterized by the hour, day, month, and year. A trip is counted as a departing trip for its starting station if the start time falls within the particular time slot, and a trip is counted as an arriving trip for its ending station if the end time falls within the particular time slot. This aggregation process will be further clarified in the **Data Preparation** section.

2.2 NOAA Hourly Weather Report

There is an obvious correlation between the probability that someone may rent a bike and the current weather conditions. If there is high wind or low visibility, for example, a person may choose to not rent a bike due to inclement weather. This connection is also recognized in the related paper, where both bike trip data and weather data are used to train the prediction models.

As aforementioned, the bike trip data is aggregated into hourly time slots with the total number of departures and arrivals per station per time slot. Thus, hourly weather data can be easily appended to the aggregated bike data. The National Oceanic and Atmospheric Administration (NOAA) records hourly weather data at land stations all over the world; the relevant features from this dataset are temperature, precipitation, wind speed, and visibility. So, NOAA data from the San Francisco International Airport weather station was collected from January 2018 to January 2020, fitting with the available bike data. This data set is quite small, with either one or two weather observations per hour, resulting in a total of around 20,000 entries.

There are several report types recorded by the NOAA in these datasets; for this project, we want to only take the FM-15 (Source 7) reports which contain all the relevant information. Thus, after filtering, the dataset contains around 18,000 entries. A small subset of hours are missing entries, so when joining the bike and weather data, any hour where weather data is missing is discounted.

2.3 San Francisco Police Department Incident Data

A connection that is mentioned in the related paper but not actually implemented is the correlation between perceived crime rates and someone's willingness to ride a bike through that neighbourhood. The last dataset describes police incidents in San Francisco, again from January 2018 to January 2020. This data comes directly from the San Francisco Police Department, which collects and publishes a summary of all incident reports. There are a total of around 310,000 entries in this dataset, and for each entry, we have the incident category, latitude, and longitude of the incident occurrence.

There are a total of 51 different incident categories, but only a subset of these categories refer to visible crimes that may affect a person's perception of a neighbourhood. For example, some irrelevant categories include courtesy reports, impounded vehicles, misplaced vehicles, gambling, traffic violations, and non-criminal incidents. After filtering out the irrelevant categories, we are left with a dataset containing around 210,000 entries.

This dataset cannot be directly joined with the aggregated bike data as opposed to the weather data. Therefore, we first need to assign the closest bike station to each incident based on minimizing the Euclidean distance between the incident and the bike stations. Then, once the stations are assigned, we need to apply the same aggregation process of counting how many relevant incidents occur in each time slot for each station. Again, this process will be further clarified in the **Data Preparation** section.

3 DATA PREPARATION

An example of the aggregation process introduced in the **Data Understanding** section is shown in Figure 1.

Joining the aggregated bike data and the weather data is very straightforward, each weather entry is appended to its corresponding bike data row in terms of hour, day, month, and year. After this step, we are left with a table containing all the relevant bike and weather data in its desired format.

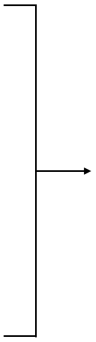
Hour	Day	Month	Year	Start Station	End Station
1	2	8	2018	Station 1	Station 3
2	5	8	2018	Station 2	Station 1
1	2	8	2018	Station 2	Station 3
1	2	8	2018	Station 1	Station 2

Hour	Day	Month	Year	Station Name	Depart Count	Arrival Count
1	2	8	2018	Station 1	2	0
2	5	8	2018	Station 1	0	1
1	2	8	2018	Station 2	1	1
2	5	8	2018	Station 2	1	0
1	2	8	2018	Station 3	0	2

Fig. 1. Aggregating historical bike trip data.

Time Slot Features	Weather Features	Station Name	Depart Count	Arrival Count
Time Slot 1	…	Station 1	2	0
Time Slot 2	…	Station 1	0	1
Time Slot 1	…	Station 2	1	1
Time Slot 2	…	Station 2	1	0
Time Slot 2	…	Station 3	0	2

Time Slot Features	Incident Category	Closest Station
Time Slot 1	1	Station 1
Time Slot 1	2	Station 1
Time Slot 2	6	Station 1
Time Slot 2	4	Station 2



Time Slot Features	Weather Features	Station Name	Depart Count	Arrival Count	Number of Incidents
Time Slot 1	…	Station 1	2	0	2
Time Slot 2	…	Station 1	0	1	1
Time Slot 1	…	Station 2	1	1	0
Time Slot 2	…	Station 2	1	0	1
Time Slot 2	…	Station 3	0	2	0

Fig. 2. Joining combined bike and weather data with crime data.

Next, we need to join the combined bike and weather data with the crime data. This step is shown in detail in Figure 2.

Lastly, there are two auxiliary time-related features to add to each row, a binary indicator equal to 1 if the row's time slot is during a weekday and equal to 0 otherwise, and a binary indicator equal to 1 if the row's time slow is during a holiday and equal to 0 otherwise.

In conclusion, the final dataset has the following schema:

- time slot year
- time slot month
- time slot day
- time slot hour
- station name
- wind speed rate
- visibility
- temperature
- precipitation
- number of crime incidents

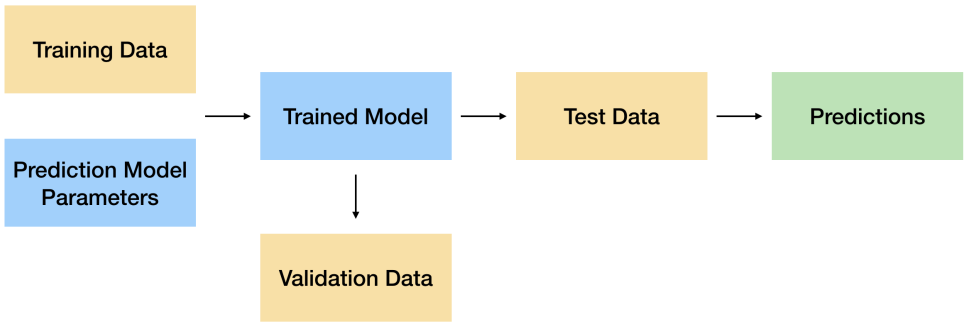


Fig. 3. Model architecture.

- weekday
- holiday
- **number of departing trips**
- **number of arriving trips**

The first 12 items are input features of the prediction models and the last two items (bolded) are output values.

4 MODELLING

The model architecture is described in Figure 3. The input data is used to train a prediction model to predict departing trips and a second prediction model to predict arriving trips for each station. Three prediction model methods were tested: linear regression, gradient-boosted tree, and random forest. There are a total of 341 stations in use throughout the entire time range from January 2018 to January 2020, resulting in a total of 682 prediction models for each model type and a grand total of 2046 prediction models created throughout the entire modelling process.

The dataset was split into three sets for training, validation, and testing. The training set contained data from January 2018 to July 2019, the validation set contained data from August 2019 too October 2019, and the testing set contained data from November 2019 to January 2020.

4.1 Hyperparameter Selection

Cross validation was used to select the prediction model hyperparameters.

- Linear Regression: regularization parameter $\in \{0.3, 0.5, 0.7\}$ and elastic net parameter $\in \{0.5, 0.8\}$
- Gradient Boosted Tree: maximum depth $\in \{3, 5\}$
- Random Forest: number of trees $\in \{50, 100\}$

The best model was chosen and applied to the validation set to obtain the R^2 value. Then, the model was used to predict either departure or arrival counts in the test set.

5 EVALUATION

After modelling, we now have data from November 2019 to January 2020 where we can compare the actual departure/arrival counts and the predicted departure/arrival counts, as well as the R^2 values of each prediction model. First, to get an overall idea of how the models compare to each other, Figure 4 graphs the average R^2 value of each model type.

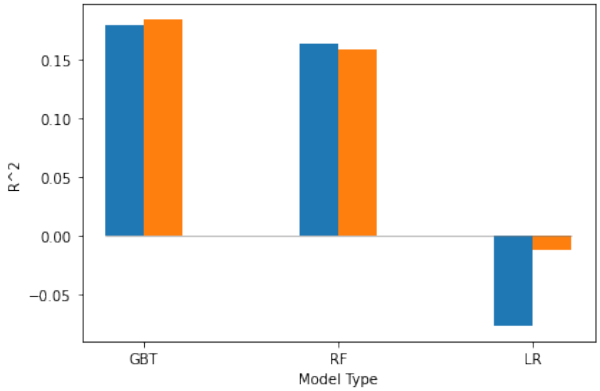
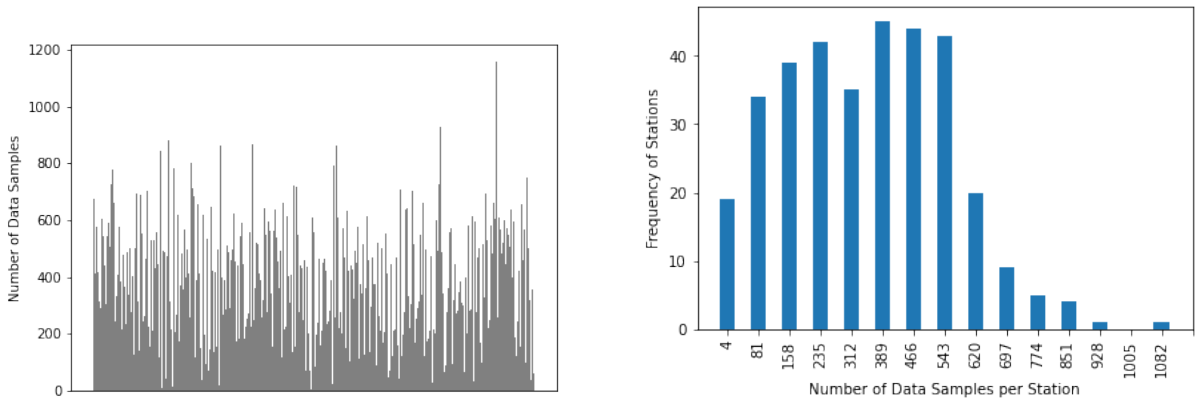


Fig. 4. Comparison of R^2 values across the three different models, the blue bars represent departure prediction models and the orange bars represent arrival prediction models.



(a) Number of data entries for each station.

(b) Histogram summarizing Figure 5a.

Fig. 5. Distribution of data entries across stations.

We can see that none of the models are performing particularly well. This may be due to an unbalanced distribution of data entries across stations. Figure 5 shows the distribution of data entries across all stations. It is clear that the majority of stations have a fairly small number of data entries. A lack of training data means that the prediction models for less popular stations may be fairly inaccurate.

Next, Figure 6 plots the discrepancy between predicted and actual departure/arrival counts for each model type. The prediction models are clearly underestimating both departure and arrival counts. Overall, GBT is the best model, however, the method of applying the same prediction model to all stations regardless of station popularity is clearly not the best technique.

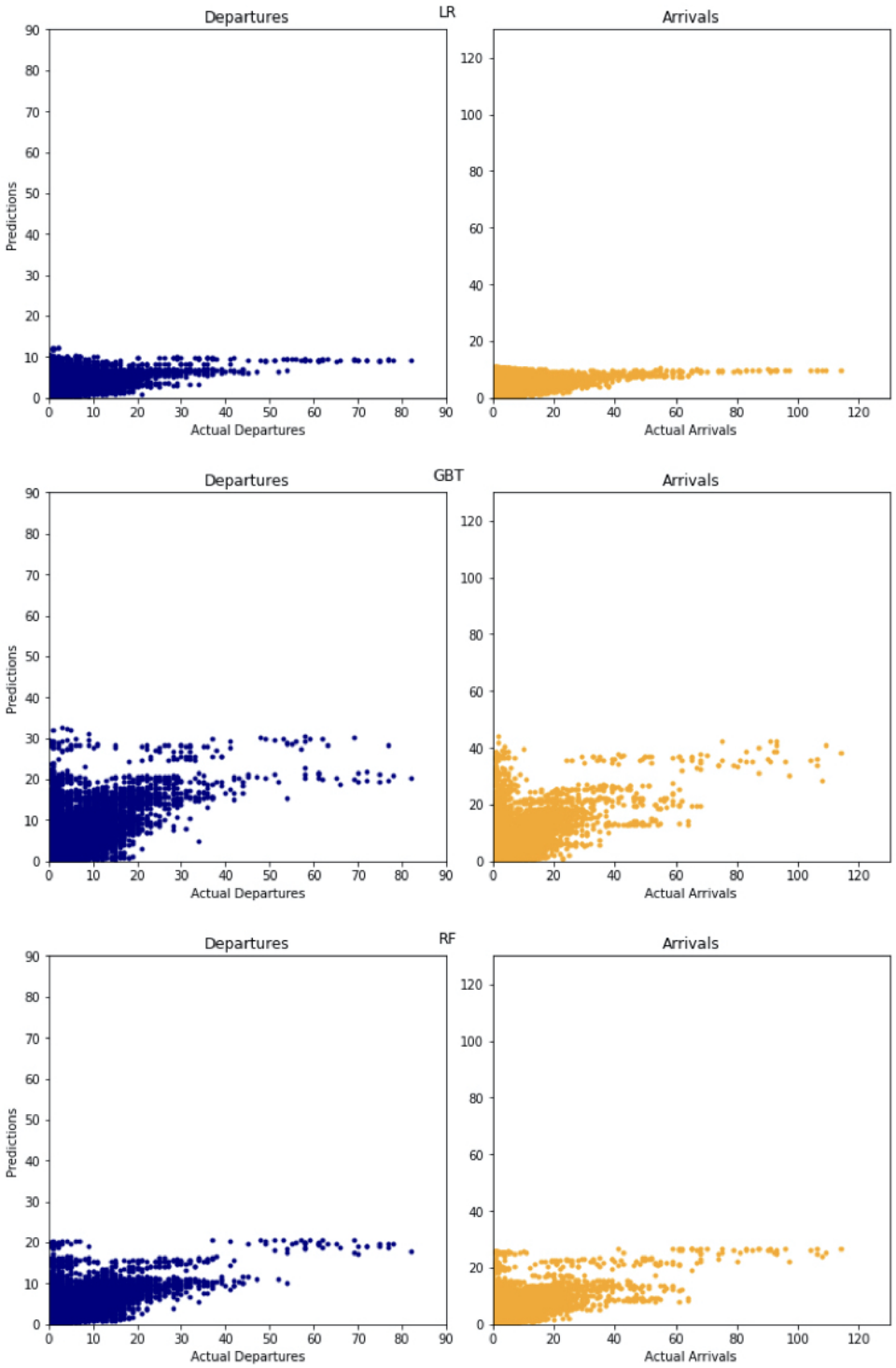


Fig. 6. Discrepancy between predicted and actual departure/arrival counts.

6 CONCLUSIONS AND FUTURE WORK

In conclusion, the three prediction methods tested in this report performed quite poorly given the prepared data. Gradient-boosted forest had the best relative performance and linear regression had the worst relative performance. All three models tended to underestimate both departure and arrival counts. There are three avenues of future work that could extend this project:

(1) Calculate some threshold for station popularity and only apply prediction models to stations above the threshold. Then, we can see how much a lack of training data contributes to model inaccuracy.
(2) Use the same data preparation techniques except without including crime data and compare how well the prediction models without crime data perform to the models with crime data. The original paper only used time and weather features, so we can test the impact of including crime data.
(3) The original paper had an extra step in their model architecture, reducing the prepared dataset into a smaller dimension then training the prediction models on the reduced dataset. This was done to prevent overfitting. Due to the highly theoretical nature of the reduction and eventual inversion back to the original dimension, we neglected to perform this operation within the scope of this project. However, this may improve model accuracy, so any future work should consider adding reduction back into the model architecture.

REFERENCES

[1] Hulot, P., Aloise, D. & Jena, S. (2018). *Towards Station-Level Demand Prediction for Effective Rebalancing in Bike-Sharing Systems*. KDD'18: Proceedings of the 24th ACM SIGKDD International Conference in Knowledge Discovery & Data Mining. pg. 378-386.