Project

July 4, 2024

1 Project Summary

1.1 1. Description of the dataset and what's challenging about it

The dataset includes bus line trips across their . It includes identification information such as trip_id, line_id, station_id, etc

In addition it has data we are trying to predict, such as time of arrival, and number of passengers that go on in the stop

There are a few difficult issues when predicting on the data:

- High skew on 0 values, which can cause the model to overfit and just predict ~0 all the time:
- Some lines are more represented than others, some lines will appear in 80 trips, while others in 1, which will give a bias on the lines that appear more

In general, there we a lot of different features and combiniations that could be correlated, and knowning which one is worth the time was hard to determine at first. Plus some data like x,y position is useless when raw, and needs to be processed heavily.

1.2 2. Data cleaning and preprocessing.

We cleaned the data from outlier trip durations

in the exploration of the model predicting passengers up, we decided to add the following coloumns : total_distance (from its first station to the last) line_count (how many other bus lines pass through this current station) max_line_count (saving the max line_count it passes through in it's route) : we believed that it can affect how many people might board the bus density (how many other stations are around the current one according to gaussian kde algorithm): we used the kde algo that

distance from to next station time period (morning being 7-11, Noon 11-15, Afternoon 15-19, Evening 19-23, Night 23-7) hour of the day (0-23)
start_hour (what hour the bus started) drive_fraction (how much of the trip was completed) max_density (similar to max line_count - the maximum density of a station the line passes through)

we droped the following coloumns : mekadem_nipuach_continue, mekadem_nipuach_luz, lat and long we belived that the mekadem nipuch was not usefull enough to calculate the data, and the lat and long were used in other col's calculations (distance from to next station, total_distance, density) but were not useful on their own.

adjusting values: In passengers continue, if there were negative values, we deduced it's a human error and chose to put instead the abs value.

we filtered for outliers in the trip_duration that was greater than 2 standart deviation units from the median, and as well in passengers continue .

The considerations that guided your design of learning systems

The various methods you tried and ended up not using them, and the results you obtained with them. we tried using linear regression, however grading boosting worked a lot better which indicated that the data is not linearily seperable we created the features into rush_hour or things that have to do with the door_close col , yet we didnt see enough impact on our model. we tried engineering using sevral features, but ended up using only

- The learning system you ended up using For both tasks, we ended up using XGBoost regression, trained on handpicked features engineered by us.
- The test error you expect your system's predictions would have. Explain why you expect such error.

we used ChatGpt, it helped us in creating plots, converting dateTime, geographical data etc. we think it allowed us under the short time period we were given to create meaningful data.

1.3 3. The considerations that guided our design of learning systems.

We started with quick and dirty baseline to see our starting point (linear regression with one feature) and an mse 3.67 for task 1 and mse = 20000 for task 2. Then we iterated and added features as we went. We split the data to train/dev/test and evaluated via kfold cross validation.

For feature engenieering we used features based on intuition and verified them using correlation to the y column. We saw using a correlation heatmap between the columns how the different features might affect the passengers_up field, and notably the passengers_continue was the most correlated one. we also saw that the density was correlated with a significance to the pass_up field. moreover, we used dummmies to use the categorized data, we did that for `cluster`, `direction`, and `time_period`.

1.4 4. The various methods we tried and ended up not using them, and the results we obtained with them.

We tried using linear regression, however gradient boosting worked a lot better which indicated that the data is not linearily seperable.

We tried creating features like rush_hour or things that have to do with the door_close col, but we didnt see enough impact on our model.

1.5 5. The test error we expect our system's predictions would have is:

2.5 on task 1 and **80** on task 2.

Based on our splitting of the data, we kept a test set rather seperate, and made sure it looks similar to how a real test set would be created.

We decided to sample entire trips instead of random rows based on how the test data looked, which affects how the model will train and hopefully make it more aligned with the test set.