Trip Duration Prediction
Bangalore Metropolitan Transport Corporation

Team : Pheonix

Saurabh IMT2016098
Rishikesh IMT2016131
Abhinil IMT2016015

Machine Learning Project Report

Introduction

About 23 million lines of data collected over a week by the government of Karnataka with the Bangalore Metropolitan Transport Corporation. Applied machine learning algorithms to predict the time taken by bus trips in the future. We have trained various ML models on the data and compared the results. Data Analysation has been the main focus our project approach.

The Dataset

Training Data :

- Id- Unique device id of the bus.
- TimeStamp - Time at which bus is currently at.
- Latitude - Current latitude value.
- Longitude - Current longitude value.

Test Data :

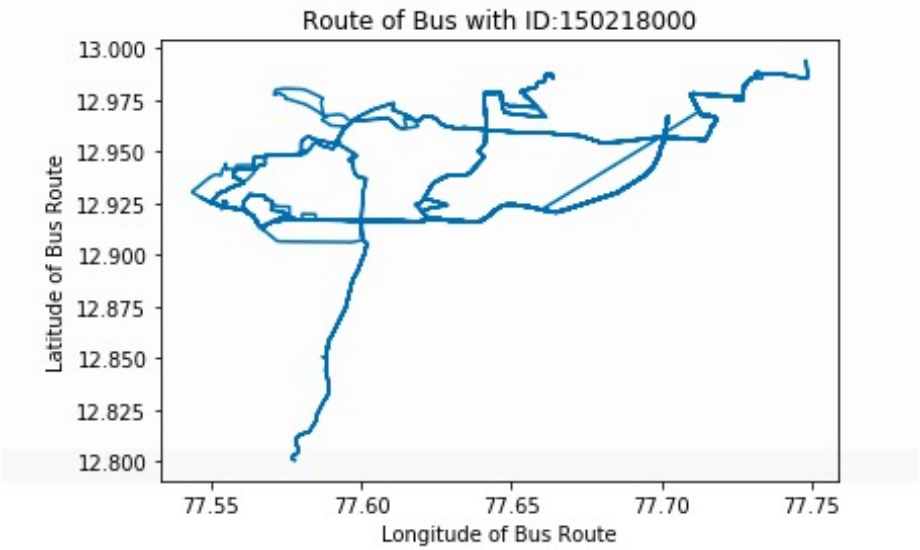
- Id- Unique device id of the bus.
- TimeStamp- Starting time of the journey. Explicitly at TimeStamp the bus is starting from LATLONG1.
- Duration- Duration of the journey to be predicted from LATLONG1 to LATLONG100 in the test set for each Id
- LATLONG1- Starting point of the journey with latitude and longitude values separated by ':'.
- LATLONG100- Ending point of the journey with latitude and longitude values separated by ':'.

Data Analysis

When we started analysing the data, there were 230 million data points. There were about 6000 unique bus Ids each following a different route.

Splitting the data

As a first step we had to sort the data according to bus ids and their times, but this was not possible due to hardware restrictions. So we started by splitting the data according to bus Ids. We created one file for each bus.



This is the route taken by a particular bus.

Removing Outliers

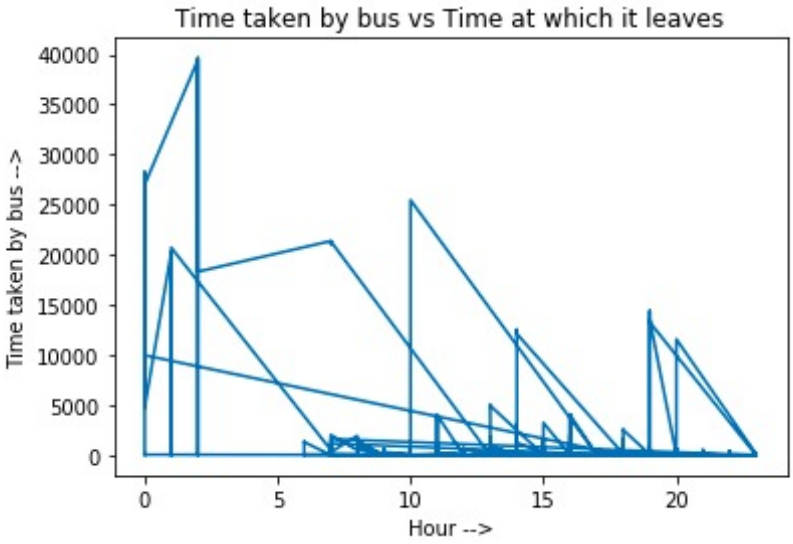
Then, we started detecting outliers in latitude and longitude columns and we as all the values should be within 12.5 to 13.2 for latitudes and 77.3 to 78 for longitudes. We removed all the values falling outside the range.

Sort the data

We separated the Date and time column into the week, hour, minutes and seconds.
Now we sorted each data file based on weekday - hour - minutes - seconds.

Further Cleaning and analyzing

On further analysing the data we found that there are a lot of cases where the bus doesn't move for a certain amount of time.



First, we thought to remove all the cases where the bus was at the same set of coordinates for a certain time. We did this to get the direct time taken to reach the next set of coordinates. Then we realised that the location precision was not that good and the coordinates were not accurate. We realised this by taking some data-points and putting them on the google map.
So, instead, we dropped rows when the speed of the bus became zero for more than 10 seconds.

Adding new features

On further thinking, we came to the conclusion that the Bangalore traffic will follow a certain pattern and that would affect the time taken by the buses. During the peak hours, the time taken by the bus will be more due to more traffic and it will subsequently decrease during the night, early morning and between office hours. So the time taken follows a kind of sinusoidal curve.
So first we decided to add two columns of sin(w1*hour) and cos(w2*hour) and since we don't know the exact sinusoidal shape we were taking w1 and w2 as hyperparameters. Since there were many possible combinations of w1 and w2 it was practically impossible to try every combination manually. So, we decided to put sin(2i/k) and cos(2i/k) where i goes from 1 to k and k will be the hyperparameters. We set k as 14.

Now out data is ready for training

Training The Model

We changed the data format by creating a row for each pair of coordinates. Taking the timestamp of first prediction we predicted the next one and so on.

Due to hardware limitations we trained the model by randomly selecting 300 bus_ids from a set of 6000.

Random Forest

This model was overfitting the data as we were using very less number of trees (about 10) due to hardware limitations.

XGBoost

We thought that since it is a boosting algorithm we will train it in mini-batches(every busId corresponds to one batch) but as we start training the model it was not able to learn properly may be beacuse as we train it on another bus the route will change completely and because of this error function is changing completely.
So we train it on randomly selected 300 busID's but since the 300 buses are not covering the entire Bengaluru city so it is overfitting on test data.

Linear Regression

Form this model we are getting the best result. We got RMSE of about 17k.

Evaluation

We used RMSE for our Evaluation because RMSE is an absolute measure of fit. As the square root of a variance, RMSE can be interpreted as the standard deviation of the unexplained variance, and has the useful property of being in the same units as the response variable. Lower values of RMSE indicate better fit.

Future Improvements

We can train the model on complete data if we have enough hardware and can use more number of trees in Random Forest and XGBoost. This will surely decrease the RMSE.

References

- https://github.com/ravi03071991/ML$_T A_I IITB_2$018$https://scikit-learn.org/stable/$
- https://xgboost.readthedocs.io/en/latest/
- https://pandas.pydata.org/

Thank You!!