Lyft Travel Time Estimation

Candidate: X.L

Outline

- Summary
- Problem Statement and Overall Approach
- Data Exploration and Cleaning
- Feature Engineering
- Model Selection and Tuning
- Potential Improvement Ideas

Summary

In this data challenge, I have trained and built a XGB regression model that predicts travel time for rideshare applications given only limited information about the trip: starting and ending location coordinates and departure timestamp. The model achieved 25% lower RMSE than the benchmark Linear Regression model with a RMSE of around 300 secs on a hold-out validate dataset.

While the accuracy of the model can definitely be improved given more features and more time for the project, I believe the results are already showing good promise for real application purposes and some of the methodologies employed here are transferable when solving real problems at Lyft.

Problem Statement and Overall Approach

Problem Statement:

Build a predictive model that can estimate travel time between two specified locations at a given departure time

Overall Approach:

- Since the problem asks for predicting travel time, which is a continuous variable, the kind of model that applies are **regression** models.
- Available features are limited. Need to do some **Feature Engineering** to create more predictive features (e.g. trip length, weather, location, traffic, time etc).
- **Try out a suite of regression models** (using linear regression as starting point and benchmark) and **pick the best** performing model.
- **Fine-Tune the hyperparameters** of the best performing model from the previous step.

Data Exploration and Cleaning

Key Observations:

1. Average trip duration is 14 mins (837 secs)
2. Start and End coordinates are concentrated in a small region (starting location center coordinates: 40.75057, -73.97391) with standard deviations more than two orders of magnitude smaller than average. Google search shows that it is in New York City.
3. There are no null values.
4. The range of dates of the training and test data: 2015–01-01 to 2015-12-19
5. Trip Duration has an unbalanced distribution with some extreme values. Applied outlier removals for trip duration < 60 secs and > 7200 secs (Note that these thresholds are determined from Mean +/- 3 * Std of the log transform of trip duration).

Feature Engineering

Time Features:

- Created *hour_of_day, day_of_week, day_of_month* and *month_of_year* features from the departure timestamp
- Created *holiday* feature (binary) based on Public Holiday information available for year 2015.

Distance Features:

- Created *distance* feature by computing haversine distances for all training and test data using the haversine formula.
- Removed rows from training and test data where distance is 0.

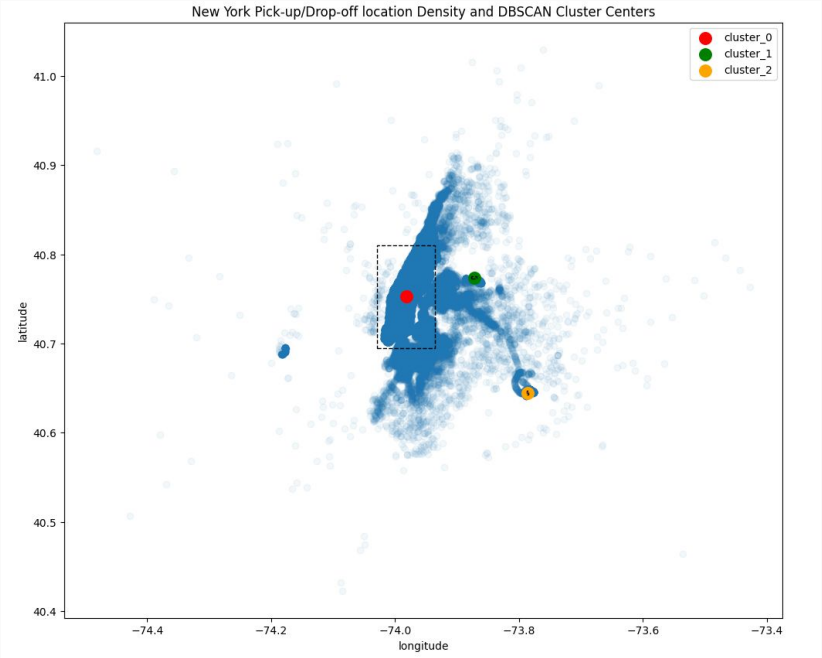
Weather Features:

- Downloaded weather data from NOAA for NYC for the entire year of 2015
- Created *hot_day* and *cold_day* features from Max/Min temperatures of day (*hot_day*: TMAX > 90; *cold_day*: TMIN < 32)
- Added 5 weather features to Train/Test data: *PRCP* (precipitation), *SNOW* (Snow fall), *SNWD*(Snow Depth), *hot_day* and *cold_day*.

Feature Engineering (continued)

Location Features:

- Used DBSCAN clustering on 1% of all start / end coordinates of data and identified 3 prominent clusters (City Center, LaGuardia Airport and JFK Airport).
- Created one-hot encoded Location_Cluster features for both training and testing data sets using a bounding box approach.



Model Selection

Model	RMSE (Root Mean Squared Error)
Linear Regression	406
Ridge regression	443
Lasso regression	414
XGB regression	302
Random Forest	371

- XGB regression performed best out with a 25% lower RMSE than benchmark Linear Regression.
- Picked XGB regression as the best model for further optimization.

Hyperparameter Tuning

Parameter Grid:

```
hp= {
    'max_depth': [7, 9, 11, 13],
    'learning_rate': [0.01, 0.05, 0.1],
    'n_estimators': [500, 1000]
}
```

Best results was achieved with *max_depth: 9, learning_rate: 0.05, n_estimators: 1000*; Model achieved RMSE of 303 secs on the hold-out validation data.

Picked the XGB regression model with these set of parameters as the final model.

Potential improvement Opportunities

Feature Engineering:

- **Neighborhood Features**: One way to potentially improve the models is to fetch address based on given coordinates of start and end location of trips. Neighborhood and Zip Code information can be derived from these addresses and used as features for our models.
- **Traffic Features**: One could look at potentially grab traffic information for each trip. E.g Using Google maps API to grab route information and get traffic information for roads in the route. A feature like number of high traffic roads in the route can be constructed.

Modeling:

Due to the limitation of time, only a suite of 5 different regression models have been tried. Other good options include **Support Vector Regressors**, **Neural Networks etc.**

Model Training:

Due to the limitation of computation resources (my laptop), I trained the model only with 1% of available training data, if more resources are available, training on 100% of available training data might help improve the performance of the final model.