New York Taxi Fare Prediction

Anirudh C(IMT2017006)
Arjun P(IMT2017007)
Deep Inder Mohan(IMT2017013)
Team Name: GG

Contents

1 Introduction

We attempt to predict the fare amount for a New York cab ride. The dataset used for this task contains the following attributes:

- Key: Unique string identifying each cab ride
- Pickup Datetime: Pickup timestamp
- Pickup Location: Latitude and longitude
- Dropoff Location: Latitude and longitude
- Passenger count: Number of passengers
- Fare Amount: Fare paid for cab ride

2 Data Preprocessing and Exploration

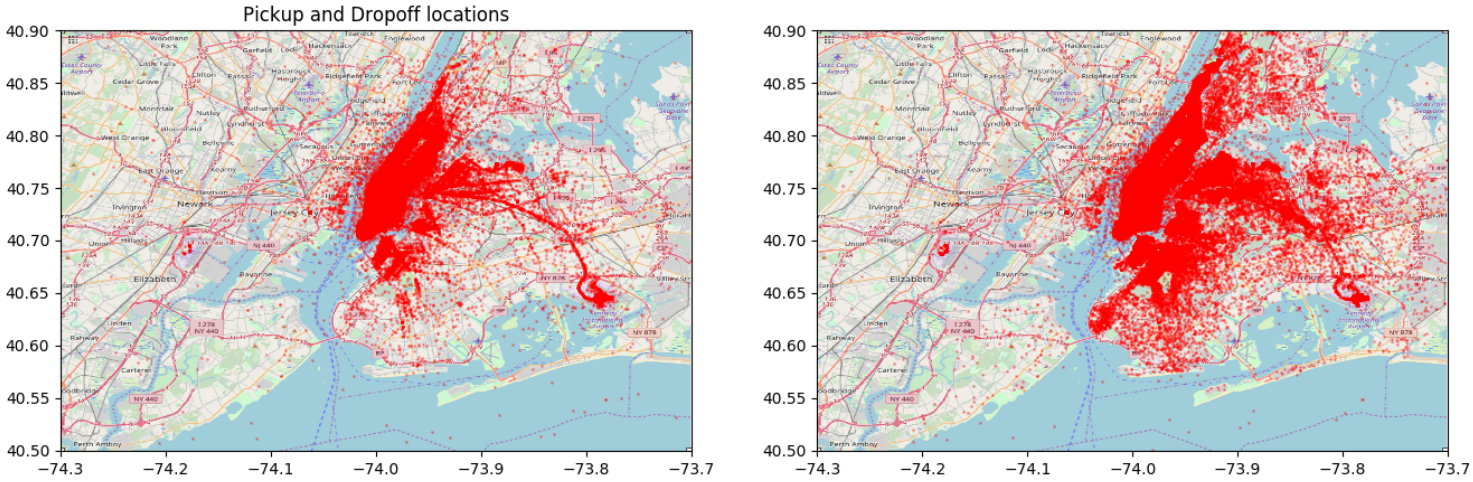
Real-world data is prone to semantic errors and even missing data points. Thus, we inspect the data to account for the same.

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
count	999994.0	999994.0	999994.0	999994.0	999994.0	999994.0
mean	11.332852307113846	-72.49888687543535	39.91797554189262	-72.49967779988924	39.9211666716207	1.6837691026146158
std	9.792374269157087	12.723840482600034	7.883370079789801	11.96171796186543	10.116248631785915	1.3383789292864057
min	-60.0	-3007.20545	-3458.6647020000005	-2913.518675	-3461.540872	0.0
25%	6.0	-73.992047	40.734951019287 11	-73.991395	40.734095	1.0
50%	8.5	-73.981832	40.752627	-73.980122	40.753177	1.0
75%	12.5	-73.96713199999998	40.767095	-73.963661	40.768135	2.0
max	450.0	2814.475637	2210.174975	2842.47403	3345.9173530000007	208.0

From the above, we can observe that certain values are logically invalid such as:

- Passenger counts of 0 or greater than 7 (physical limitation of capacity of cab)
- Fare amounts less than $2.5 (minimum fare for a cab ride in New York)

Let us look at the distribution of the pickup and dropoff location on a map of New York.



From the above we can notice the following anomalies.

- Certain locations are outside of New York City with invalid latitude and longitude values.
- Certain locations are inside water bodies.

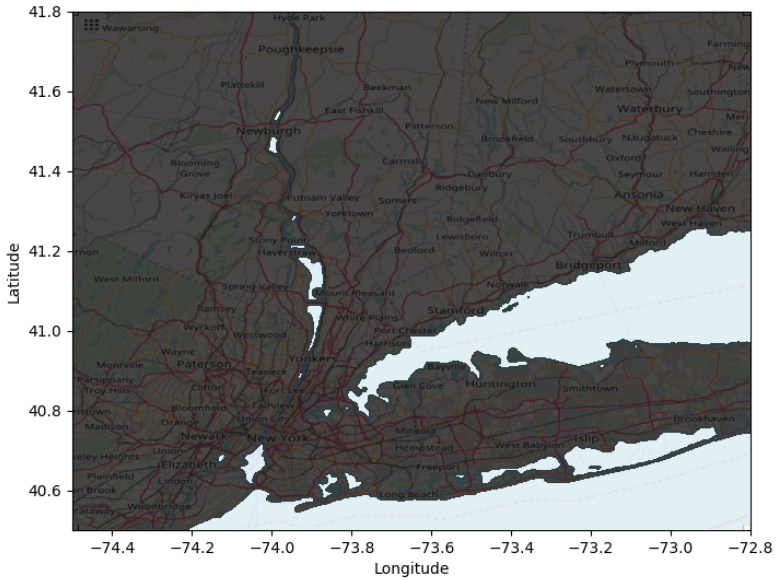
To detect points with pickup or dropoff locations outside of New York from the dataset:

We obtain a rectangular bounding box for New York City in the following format ($\text{longitude}_{\text{min}}$, $\text{longitude}_{\text{max}}$, $\text{latitude}_{\text{min}}$, $\text{latitude}_{\text{max}}$). We use the bounding box (-74.5, -72.8, 40.5, 41.8) further in the project.

Detecting pickup and dropoff locations in water is more involved.

- We create a boolean mask on an image of the map of New York (within our bounding box) that assigns a boolean '1' or '0' to each pixel on the image. The mask value for a particular

pixel is '1' if the locations described by that pixel are on land and '0' otherwise. This is depicted in the image below.



- We now map every latitude and longitude value in the data to a pixel in the image and then apply the boolean mask to decide if that location is in water or not.

We also remove the data points that have missing attributes (null values).

3 Feature Engineering

A primary concern in this task is the manner in which the pickup date and time are encoded.

Note that the time of day is cyclic with a period of 24 hours, the day of week is cyclic with a period of 7 days and the months are cyclic with a period of 12 months. We want to capture this aspect of the date and time with our encoding. We define the following encoding:

We map time of day (0 - 86400 minutes), day of week (1 - 7 days) and month (1 - 12 months) to points on a circle using an angle measure

$$\theta = 2\pi \frac{x}{P}$$

where, x is the value of the attribute and P is the period.

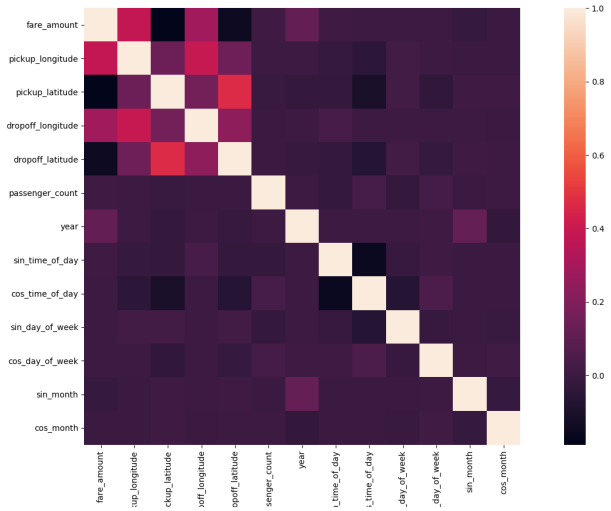
We then add 2 new features as sine and cosine of θ (mapping onto unit circle).

4 Experiments

4.1 Experiment 1

We remove all points in water and outside the bounds of New York.

We look at the correlation of the features.



We can see that some of the features highly correlated. Hence, to be robust against this and to allow in built feature selection we use a tree based learning algorithm with boosting (ensemble).

Due to the size of the dataset, we need an optimal algorithm to train the random forest. Hence, we use LightGBM as a boosting algorithm.

We normalise the data and use min-max scaling for the categorical features passenger count and year.

This model gave an RMSE of 9.9 on the test dataset. Upon further inspection we notice that the predicted values on the test data are all similar. This may be due to the fact that we scaled the data thereby reducing the variance. We establish this as the baseline which we aim to optimize.

4.2 Experiment 2

A factor that affects cab rides that hasn't been addressed is the distance of the trip. We estimate this using the *Haversine* formula that approximates the distance between 2 locations on a sphere. We compute this distance estimate and use it as a feature.

We now use the estimate of distance without normalising the data. Also, the sine and cosine features for time are highly correlated. Therefore, we attempt to train a model using only the radian measure (without cyclising).

This model gave an RMSE of 9.5 on the test dataset. This implies that our model could not generalise to the test data.

4.3 Experiment 3

Upon inspection of the test data, we notice that there are trips outside the bounds of New York. To be able to generalise better for these points we introduce a feature "invalid" that categorizes the data points as follows:

- $invalid = 0$, if pickup and dropoff locations are outside New York.
- $invalid = 1$, if pickup and dropoff locations are inside New York but in water.
- $invalid = 2$, if pickup and dropoff locations are inside New York and on land.

New York cab rides to airports are fixed. Hence, we add an estimate of distance to the airport from the pickup location as a feature to capture patterns in the fare amount with regard to this. Once again we use the haversine formula to estimate this.

We train LightGBM on this data, which gave an RMSE of 5.1 on the test data.

4.4 Experiment 4

We use a bootstrapping technique to sample from the data to learn a better distribution. Here we sample randomly from the dataset and thus increase the variance in the data the model uses in training. This allows us to generalise better on the test data.

Trips to 2 more airports are frequent from New York City. Thus, we add haversine distance estimates to these airports as well. This allows us to fine tune the model.

We also notice that the correlation between the passenger count and the fare is the least correlated (passenger counts in the range 1-4 don't affect the fare amount). Thus, we do not use passenger count as a feature.

This model gave an RMSE of 4.7 on the test data. This indicates that the model generalised better.

4.5 Experiment 5

To depict the effect of the boosting algorithm we train a regression tree using XGBoost which resulted in an RMSE of 10.9 on the test data. This validates the choice of LightGBM over XGBoost.

5 Conclusion

Cab fares in general are directly related to the time of travel and distance. The haversine distance estimate does not depict the actual distance travelled by the cab.

Due to the varying nature of cab rides there is a persistence of outliers. There are a variety of cab rides that can be classified as outliers. For instance, a short distance estimate could still amount to a large fare due to delays in time.

This results in the model being less generalisable and therefore an RMSE of 4.7 on the test data.

Due to the unavailability of computing resources, k-fold validation could not be performed on the entirety of the data. Also, k-fold validation on a small subset resulted in an RMSE of 5.6 on the test data, further justifying its ineptness in optimizing the predicted fare.

6 Acknowledgements

We drew inspiration from the following resources:

- Albert van Breemen: https://www.kaggle.com/breemen/nyc-taxi-fare-data-exploration
- Susan Li: https://towardsdatascience.com/how-taxis-arrive-at-fares-predicting-new-york-city-

We also discussed approaches with Rahul Murali Shankar (IMT2017033), Shubham (from MPL@IIITB) and Anuj Shah(MT2018019) the TA for this project.