

New York City Taxi Fare Prediction

November 30, 2019



by -

DHRUVINKUMAR RADADIYA (MT2019035)

DHAMELIYA VATSALKUMAR (MT2019033)

DEVANS SOMANI (MT2019032)

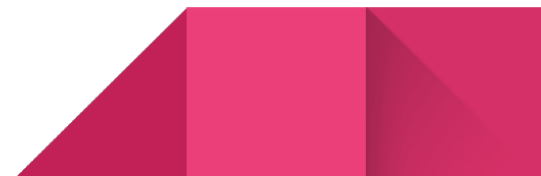
Index

| | |
|---|-----------|
| Introduction | 3 |
| Problem Statement | 4 |
| Description Of Dataset | 4 |
| ID | 5 |
| Features | 5 |
| Target | 5 |
| Data Visualization & Exploring | 5 |
| Data Preprocessing | 10 |
| Training Models | 10 |
| Support Vector Machine Classifier | 10 |
| Random Forest | 10 |
| XGBoost | 11 |
| Light GBM | 11 |
| Conclusion | 12 |
| References | 12 |

Introduction

New York City taxi rides paint a vibrant picture of life in the city. The millions of rides taken each month can provide insight into traffic patterns, road blockage, or large-scale events that attract many New Yorkers. With ridesharing apps gaining popularity, it is increasingly important for taxi companies to provide visibility to their estimated fare and ride duration, since the competing apps provide these metrics upfront. Predicting fare and duration of a ride can help passengers decide when is the optimal time to start their commute, or help drivers decide which of two potential rides will be more profitable, for example. Furthermore, this visibility into fare will attract customers during times when ridesharing services are implementing surge pricing.

In order to predict duration and fare, only data which would be available at the beginning of a ride was used. This includes pickup and dropoff coordinates, trip distance, start time, number of passengers.



Problem Statement

The goal is to create a model that predicts a taxi ride's fare based only on the information any rider would be able to provide to the driver at the time of booking.

Description Of Dataset

The train dataset provided from Kaggle is of ~55 million row each contains six features and one target column of fare amount to be predicted.


ID

- key - Unique string identifying each row in both the training and test sets. Comprised of pickup_datetime plus a unique integer, but this doesn't matter, it should just be used as a unique ID field.

Features

- pickup_datetime - timestamp value indicating when the taxi ride started.
- pickup_longitude - float for longitude coordinate of where the taxi ride started.
- pickup_latitude - float for latitude coordinate of where the taxi ride started.
- dropoff_longitude - float for longitude coordinate of where the taxi ride ended.
- dropoff_latitude - float for latitude coordinate of where the taxi ride ended.
- passenger_count - integer indicating the number of passengers in the taxi ride.

Target

- fare_amount - float dollar amount of the cost of the taxi ride. This value is only in the training set; this is what you are predicting in the test set and it is required in your submission CSV.
- 

Data Visualization & Exploring

A quick summary of the train dataset's statistics obtained from Figure 1, reveals some questionable numbers.

| | fare_amount | pickup_longitude | pickup_latitude | dropoff_longitude | dropoff_latitude | passenger_count |
|-------|---------------|------------------|-----------------|-------------------|------------------|-----------------|
| count | 3.900000e+06 | 3.900000e+06 | 3.900000e+06 | 3.899973e+06 | 3.899973e+06 | 3.900000e+06 |
| mean | 1.133998e+01 | -7.249905e+01 | 3.991828e+01 | -7.250126e+01 | 3.991341e+01 | 1.684525e+00 |
| std | 9.796856e+00 | 1.270449e+01 | 8.763235e+00 | 1.281956e+01 | 9.565379e+00 | 1.338279e+00 |
| min | -1.000000e+02 | -3.426609e+03 | -3.488080e+03 | -3.412653e+03 | -3.488080e+03 | 0.000000e+00 |
| 25% | 6.000000e+00 | -7.399207e+01 | 4.073490e+01 | -7.399140e+01 | 4.073403e+01 | 1.000000e+00 |
| 50% | 8.500000e+00 | -7.398182e+01 | 4.075262e+01 | -7.398016e+01 | 4.075314e+01 | 1.000000e+00 |
| 75% | 1.250000e+01 | -7.396712e+01 | 4.076710e+01 | -7.396370e+01 | 4.076811e+01 | 2.000000e+00 |
| max | 6.981600e+02 | 3.439426e+03 | 3.310364e+03 | 3.457622e+03 | 3.345917e+03 | 2.080000e+02 |

Figure 1

- Negative fare is present in the dataset which could be due to refunds or just erroneous. According to the TLC 3, they charge an initial \$2.50 for every trip, so there shouldn't be any fares under that amount.
- Next, there are some trips with 0 passengers.
- The latitudes and longitudes, present in the dataset are not only of New York City but all around the globe.

The original dataset contains features as pickup and dropoff locations, as longitude and latitude coordinates, time and date of pickup and dropoff, ride fare and passenger count. The data was processed to extract separate features for year, month, day, weekday and hour from the date and time of each ride, as well as trip duration as the difference between dropoff and pickup time.

The fare of a taxi ride is function of the mileage and the duration of the ride (sum of drop charge, distance charge and time charge). The drop charge is constant and the distance can easily be estimated but evaluating the duration is not a trivial task. It is the result of complex traffic processes that are nonlinear.

So, we have computed the distance between the pickup and drop location using "Haversine Distance Function".

The relations between extracted features like month, day of week etc and fare amount/frequency can be seen in the following figures-

- Frequency of ride VS Day of week (Figure 2)

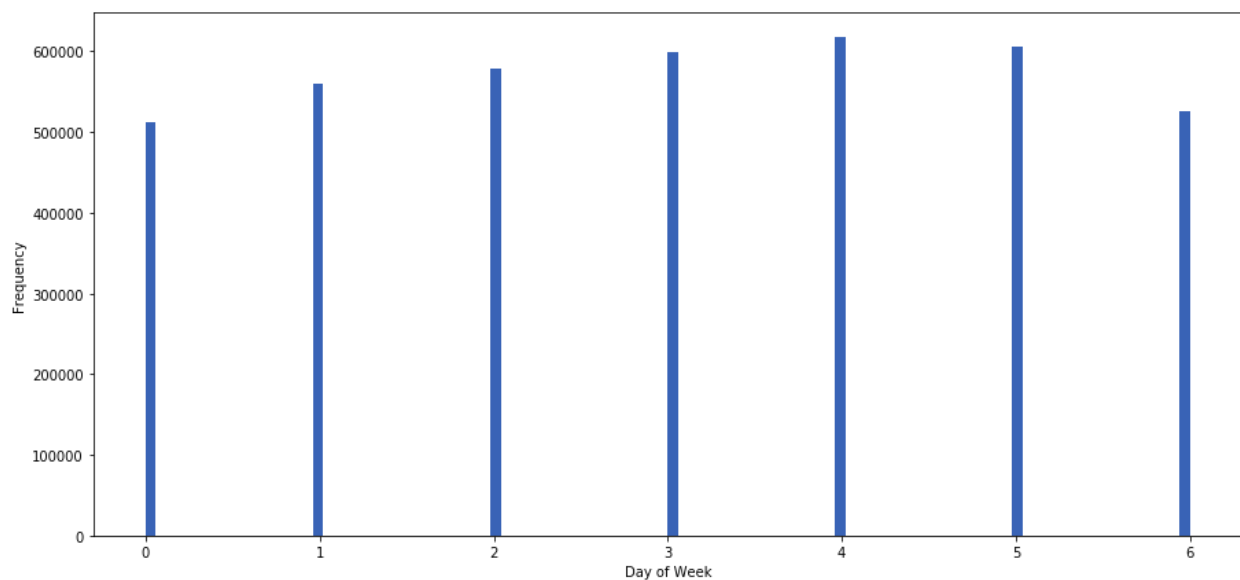


Figure 2

On weekdays the number of taxi rides is more than the number of rides on weekends.

- Frequency VS Hour (Figure 3)

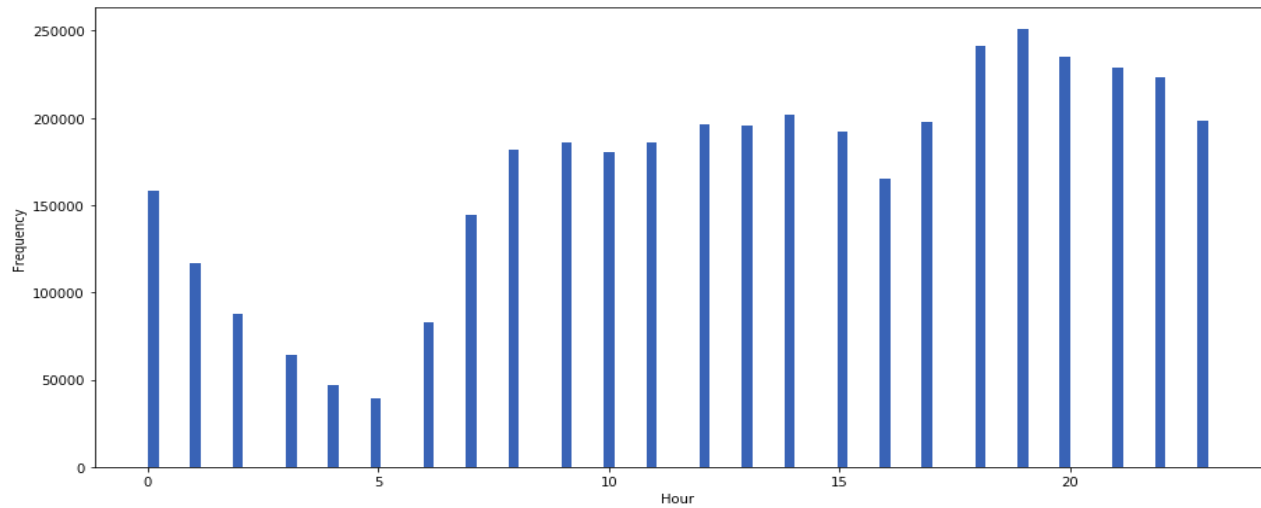


Figure 3

Number of taxi rides is more during working hours. So fare amount will be higher.

- Frequency VS Passenger count (Figure 4)

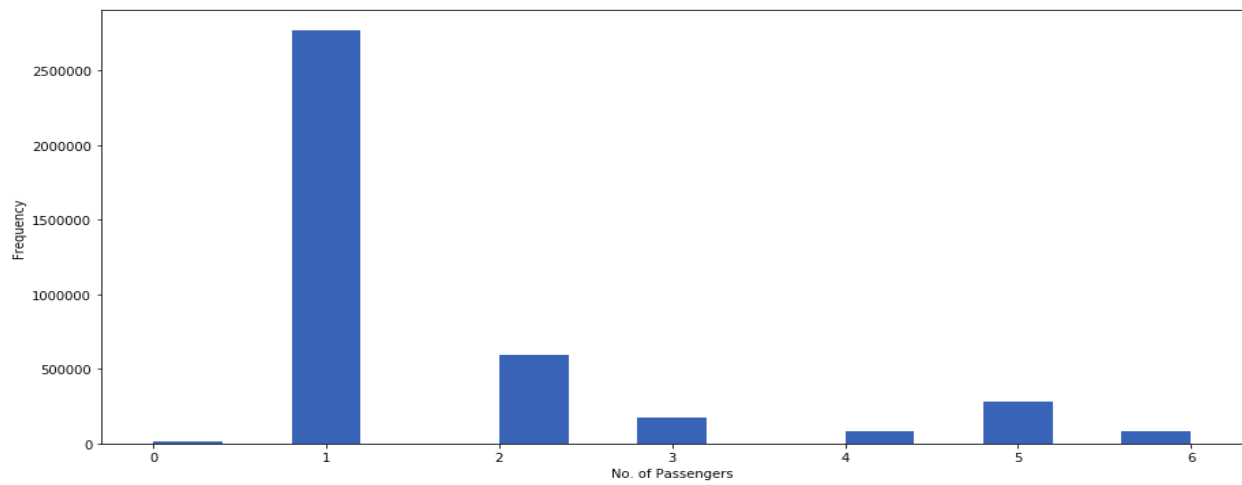


Figure 4

- Fare VS Distance (Figure 5)

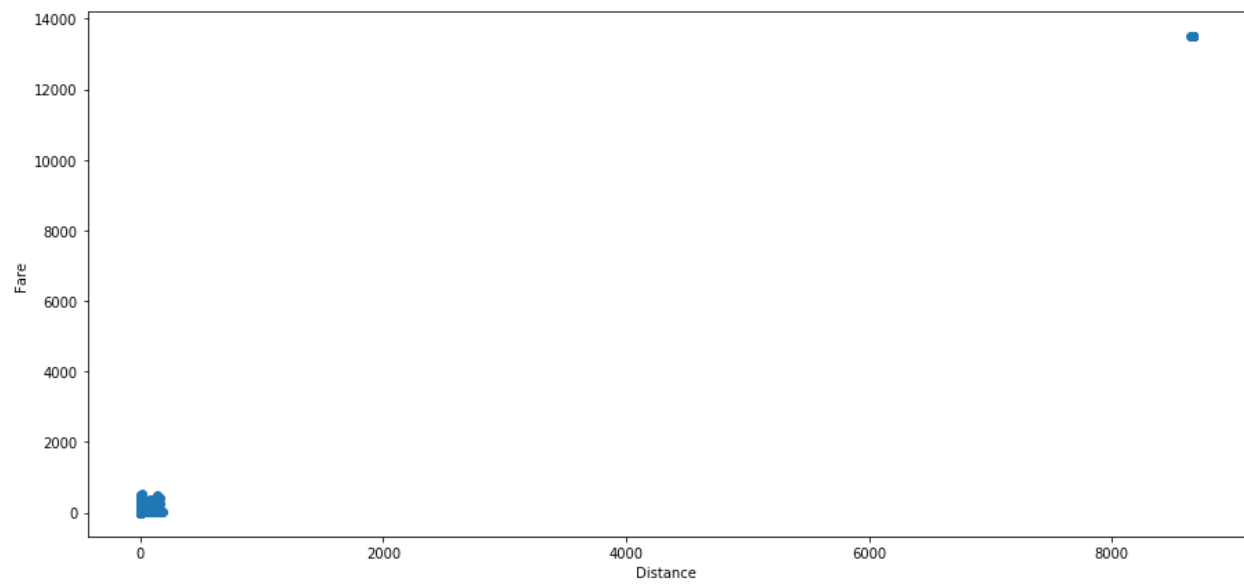


Figure 5

- Frequency VS Distance(Figure 6)

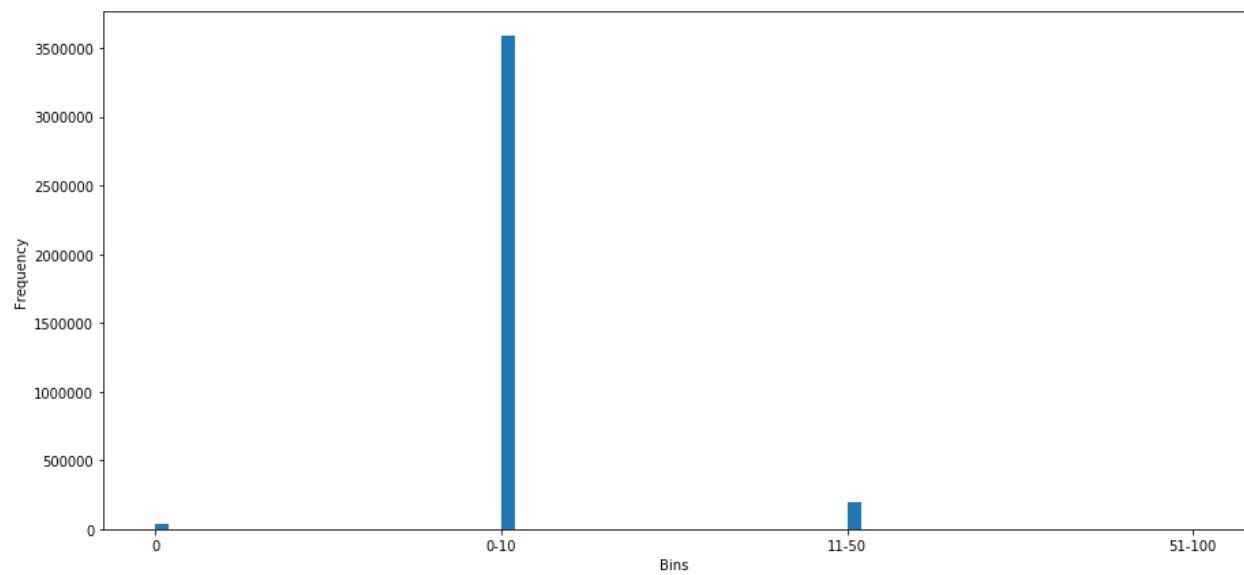


Figure 6

Data Preprocessing

- Read dataset
- Many features of the train dataset contain missing values. Since dataset is very large so we drop the rows with missing values.
- We extracted year, month, day, day of week and hour from the pickup_datetime feature.
- The four coordinates of pickup and dropoff outside the range of the latitude and longitude of New York City were dropped.
- Using four geolocation and Haversine Distance function we calculated the distance for each trip.
- We did One Hot Encoding Day of Week.
- We have controlled fare amount based on rush_hour and high_distance.

Training Models

We tried four different types of model for predicting fare amount.

I. Support Vector Machine Classifier

Support Vector Machines is considered to be a classification approach, but it can be employed in both types of classification and regression problems. It can easily handle multiple continuous and categorical variables. SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplane in an iterative manner, which is used to minimize an error. The core idea of SVM is to find a maximum marginal hyperplane(MMH) that best divides the dataset into classes. It is known for its kernel trick to handle nonlinear input spaces.

II. Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude



of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

III. XGBoost

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class right now.

IV. Light GBM

Light GBM is a gradient boosting framework that uses tree based learning algorithm. Light GBM grows tree vertically while other algorithm grows trees horizontally meaning that Light GBM grows tree leaf-wise while other algorithm grows level-wise. It will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise algorithm can reduce more loss than a level-wise algorithm.

Accuracy of model was calculated based on RMSE(Root Mean Squared Error).Models were tested multiple times and least error measured for each model is given in Table 1.

| No. | Model | RMSE |
|-----|------------------------|------|
| 1 | Support Vector Machine | 5.8 |
| 2 | Random Forest | 11.7 |
| 3 | XGBoost | 6.09 |
| 4 | LGBM | 4.73 |

Table 1

Conclusion

Considering what is and what is not accounted for in the models built in this study, their predicting results are fairly accurate. To further improve the prediction accuracy, more variable features need to be considered and modeled. Although the rides in hour and average speed in hour work as proxies for traffic, more modeling on the effect of location is needed. These quantities could be calculated for different areas to further model local effects of traffic. Also, modeling traffic and the effect of location in between pickup and dropoff points should be considered as well as difference in drivers' speed. These further steps could be taken both by analyzing larger sets of the data to infer relationships and effects of location and traffic at different times, as well as aggregation with other datasets, as data on traffic, speed limitations, etc.

References

- <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
- <https://lightgbm.readthedocs.io/en/latest/>
- <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>
- https://xgboost.readthedocs.io/en/latest/python/python_api.html