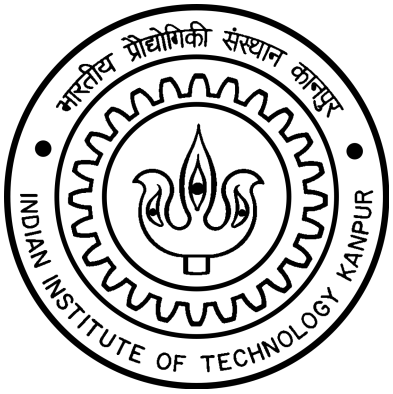

IME672A Project Report

Data Mining and Knowledge Discovery

Instructor:

Prof. Faiz Hamid

New York city Taxi Fare Prediction

Group - 4

Unnat Kumar Singh - 15807780

Pankaj Kumar - 15817475

Mayur Kumar Gupta - 15817398

Manish Kumar - 15817380

Utkarsh Sharma - 15807785

Manvendra Singh Rajput - 15807389

Contents

1 Introduction

This project aims to predict Taxi fare of New York City. We have been given real-world datasets, which consists of **55 M** tuples. Each tuple contains details about a particular taxi ride such as pickup and drops off locations, the pickup date and time, and the number of passengers traveling. The data which has been given is not ideal and has lots of errors and missing fields. Some of the tuples have negative fare which is not realistic. Also in some of the cases the number of passenger traveling in the taxi was greater than the taxi capacity. We also observed that some pickup and drop-off location are not valid points since they correspond to the points located in the water While some points are very far from the New York City which may be due to presence of outliers in the data. Our main challenge is to make use of this real-world data and model it to predict taxi fare in NY city based on given features.

2 Data Pre-Processing

2.1 Data cleaning

The first challenge in data cleaning was to read 55 Million rows, which takes longer processing time and requires a lot of memory(approx. 32 GB). After reading 5% of the whole dataset, we found that the dataset is randomly sorted by date, so we took first 4 million rows for processing since it was already randomly sampled. Generally, the performance of a machine learning model improves as the amount of training data increases. However, it may result in over fitting of a model sometimes. Therefore, we read only a small set of rows to explore the data keeping in mind the processing time and memory it takes.

Some of the challenges in the task of data cleaning were:

- Missing values
- Many tuples with negative fare
- Number of passenger in the taxi is greater than the taxi capacity
- Some drop-off and pickup locations are not in the NYC region and some corresponded to the location of the region covered in water

To solve these problems, We cleaned the data using Pandas library in python. The code for the same can be found in the following file data_cleaning.py:

To measure distance between two points, we defined a rectangular region based on the extreme coordinates of NY city. The motivation behind our approach was that the original data that was given consisted of many points which lies outside the city area. Therefore to keep the drop-off and pick-up location within the constraint area,we defined the rectangular area and did processing only on those points. Following are some functions for which we have written code :

- **remove datapoints from water**: This function removes tuples whose location lie in the region surrounded by water.
- **select boundary**: This function selects only those data points which lie inside the rectangular region defined earlier.
- **clean data**: This function removes all tuples whose fare is negative, and datapoints where the number of passengers traveling is outside the range of 0-6..

2.2 Feature Engineering

The attributes which are present in the original dataset are not directly relevant to us, so we implemented feature engineering to get a closer look into the data and found out the different features which affects the taxi fare. To search for more specific patterns in the data set effectively, we added few more features to our processed data. Some of these features are:

- **Distance**: It is the shortest possible path length between the starting and end point of each Taxi ride. We calculated it using formula taken from spherical geometry popularly known as Haversine's distance.

$$a = \sin^2\frac{\triangle\phi}{2} + \cos\phi 1.\cos\phi 2.\sin^2\frac{\triangle\lambda}{2} \tag{1}$$

$$c = 2.\arctan\frac{\sqrt{a}}{\sqrt{1-a}} \tag{2}$$

$$d = R.c \tag{3}$$

  where, ϕ is latitude, λ is longitude, R is earth's radius (mean radius = 6,371km);
  Note that angles need to be in radians to pass to trigonometric functions!

- **Splitting Date and Time**: Dataset contains date and timestamp of taxi ride in a single field. To analyze the effect of date and time of taxi ride on its fare,we need to separate the date in three categories i.e. day,month and year and timestamp seperately. Hence, we processed the data and separated the columns for **year, month, day, hour, and day of week**. This will help us to mine time series pattern in dataset more accurately.

- **Airport distance** : Further to explore if any specific zone or landmark in the city attracts more taxi ride, we added six more column to keep record of traffic rush associated with three specific Airport locations named Kennedy international airport(JFK), LaGuardia airport(LGA) and Liberty international airport(EWR) . Following are some parameters we created and used in further modelling :
  - pickup_distance_JFK: distance between any pickup point to Kennedy airport.
  - dropoff_distance_JFK: distance between any dropoff point to Kennedy airport.
  - pickup_distance_EWR: distance between any pickup point to Libert airport.
  - dropoff_distance_EWR: distance between any dropoff point to Libert airport.
  - pickup_distance_LGA: distance between any pickup point to LaGuardia airport.
  - dropoff_distance_LGA: distance between any dropoff point to LaGuardia airport.
- **borough distance**: We also divided the region into five major administrative division called borough named **Manhattan, Brooklyn, Queens, The Bronx and Staten Island.** Then we further categorized all the Taxi ride locations of the New York city into above five boroughs.
  - pickup_distance_manhattan:
  - pickup_distance_brooklyn:
  - pickup_distance_queens:
  - pickup_distance_bronx:
  - pickup_distance_statenisland:
  - dropoff_distance_manhattan:
  - dropoff_distance_brooklyn:
  - dropoff_distance_queens:
  - dropoff_distance_bronx:
  - dropoff_distance_statenisland:
- From the scatter plot we can easily figure out that Manhattan area has relatively high traffic rush. Hence, we further explored the coordinates of that area by creating two more feature named
  - is_pickup_lower_manhattan :
  - is_dropoff_lower_manhattan :

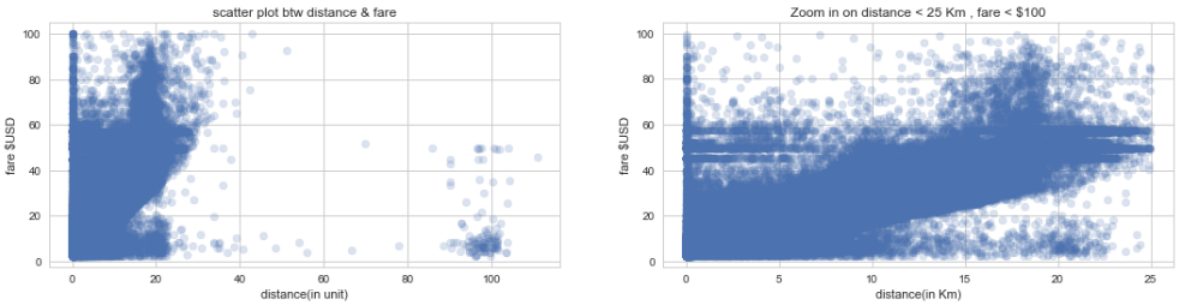

Figure 1: Scatter plot between fare and distance

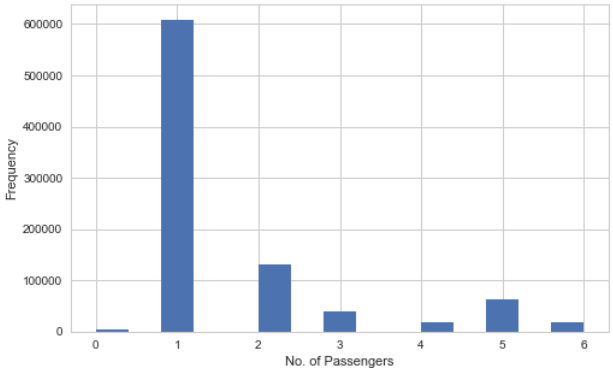

Figure 2: Histogram of passenger count

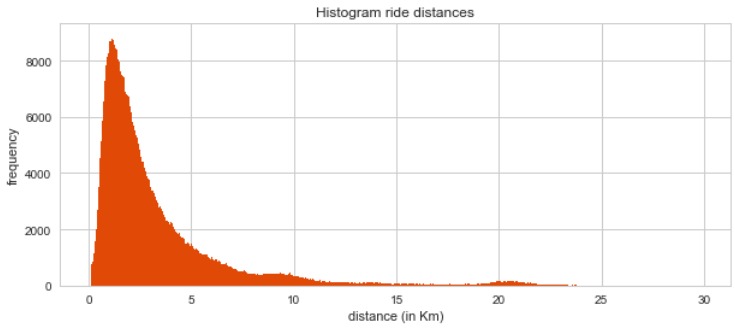

Figure 3: Histogram of distance

3 Literature Review

This problem was a competition on kaggle (online community of data scientists and machine learners), so this was a highly attempted problem statement on kaggle platform, and hence we

consulted some approaches of different authors to give a final appearance to our project. We solely used the technique to distinguish a pickup and dropoff point to be on land or water from Mr.Albert Breemen on Kaggle. We further read some blogs to explore the dataset broadly and to give an aesthetic touch to data visualization.

3.1 Linear Regression Model

Linear regression is a linear model which tends to interconnect relationship between input variables and final output by assuming a existence of linear relationship between them. Or in sort, output can be calculated from a linear combination of the input variables.

3.2 Random Forest

A Random Forest is an ensemble technique used to perform regression and classification tasks with the use of multiple decision trees and Bagging. Bagging in this context refers to training each decision tree on a different data sample where sampling is done with replacement.

3.3 LightGBM

Light GBM is a gradient boosting framework that uses tree based learning algorithm. It differs from other tree based learning algorithms due to growth pattern of tree in vertical manner or leaf wise. while trees in other algorithms grows level wise and horizontal.One of biggest advantage offered by this model is that it can handle the large size of data efficiently and takes lower memory to run.

3.4 XGBoost

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way.

4 Models and Results

In this section, we describe the models and methods that we have used to obtain our results. Following are the models in order:

- **Baseline** - In the baseline model,we have pre-processed the original data but we have not done the feature engineering part and implemented the four ML algorithms on this data. The four algorithms that we have used are **Linear regression model, Random**

forest, XGBoost and LightGBM. We calculated RMSE values for training data and test data for all four algorithms and also we have calculated the variance in each case which helped us to interpret over fitting in each respective mode.From the figure l we can see that there is a very high difference in the train and test RMSE for Random Forest, though overall the test-rmse of Random Forest is the best. Very high variance is a sign of overfitting. XGBOOST has done slighlty better than Light GBM in terms of training and test rmse - but variance is higher in XGBOOST In further work, we will consider LightGBM as the model and add features and tune this model.

- **With feature engineering** - Then we have implemented LightGBM algorithm on the processed data after including feature engineering part in data pre-processing step as explained in the data preprocessing code file. We found out RMSE for Light GBM with Feature Engineering = 3.6415988311191283. Train RMSE for Light GBM with Feature Engineering is 3.1609195102603014. Variance of Light GBM with Feature Engineering is -0.4806793208588269. With Feature Engineering, the RMSE has decreased from 3.78 to 3.64. The variance of the model has also come down from 0.5 to 0.48.

Method	RMSE(Baseline model)	RMSE(with feature engineering)
Linear regression	8.14	-
Random Forest	3.72	-
XGBoost	3.77	-
LightGBM	3.78	3.64

	model_name	test_rmse	train_rmse	variance
0	Linear Regression	8.140288	8.209541	0.069252
1	Random Forest	3.722761	1.412086	-2.310674
2	Light GBM	3.789988	3.301217	-0.488771
3	XGBOOST	3.772174	3.209402	-0.562772

Figure 4: Mode Comparision

5 Possible Modification

Based on our final result we realised that there is further scope to improve our analysis . Considering the above analysis, We would like to list out following possibilities for improving the current undertake:

- we worked on 5M rows out of given 55M rows provided to us. There are some nobel Framworks such as Dask which allow us to handle even massive datasets on a personal laptop. Moreover, learning how to set-up and use cloud computing, such as Amazon ECS, is a vital skill once the data exceeds the capability of your machine.
- Modelling location dependencies and avoid the direct point-to-point distance measure.
- Modelling the non-linear time (hour) dependency.

References

[1] r. https://www.kaggle.com/breemen/nyc-taxi-fare-data-exploration