NYC Taxi Trip Time Prediction

DSN Naveen Kumar,
Data science trainee,
AlmaBetter, Bangalore

Abstract:

In New York City many of people commute to different regions of city via taxi. A lot of streets and roads in New York city are quite busy due to traffic jams, construction or road blockage etc. Therefore, it is very important to predict the trip duration of taxi so that the user will know how much time it will take to commute from one place to other. Also, due to the increasing popularity of app-based taxi such as ola or uber and there competitive pricing levels. Decisions has to be taken by the user for opting which one to choose based on trip pricing and duration. This prediction also helps drivers to choose route having lesser trip time. They provided the dataset which is released by NYC Taxi and Limousine Commission. This dataset contains pickup time, drop-off time, geocoordinates, number of passengers, trip duration and several other variables

My primary motive is analyse the dataset, perform feature engineering to comes up with suitable independent features and building a good model that will help in predicting the trip duration of NYC taxi. Here, for prediction the taxi trip duration I applied linear regression, lasso and ridge regression and then I applied XGBoost and LightGBM. To find out which will give better accuracy and with lesser amount of prediction time. At last, a comparison of the two mentioned algorithms facilitates me to decide that XGBoost is more efficient than Multi-Layer Perceptron for taxi trip duration-based predictions

Keywords:machine learning,surge pricing,dynamic pricing,classified labels

1.Problem Statement

The dataset is based on the 2016 NYC Yellow Cab trip record data made available in Big Query on Google Cloud Platform. The data was originally published by the NYC Taxi and Limousine commission (TLC).

The main objective is to build a predictive model, which could help them in predicting the trip duration of taxi. This would in turn help them in matching the right cabs with the right customers quickly and efficiently.

- id - a unique identifier for each trip
- vendor id - a code indicating the provider associated with the trip record
- pickup datetime - date and time when the meter was engaged
- drop off datetime - date and time when the meter was disengaged
- passenger count - the number of passengers in the vehicle (driver entered value)
- pickup longitude - the longitude where the meter was engaged
- pickup latitude - the latitude where the meter was engaged
- drop off longitude - the longitude where the meter was disengaged

- drop off_latitude - the latitude where the meter was disengaged
- store and fwd flag - This flag indicates whether the trip record was held in vehicle memory before sending to the vendor because the vehicle did not have a connection to the server - Y=store and forward; N=not a store and forward trip
- trip duration - duration of the trip in seconds

2. Introduction

More than 7 billion people exist on earth. With necessities of food, water and shelter there also a key requirement of commutating from one place to other. Rapid advancement in technology in the last two decades leads to adaption of a more efficient way of transport via internet and app-based transport system. NewYork city is one of such advanced city with extensive use of transport via subways, buses and taxi services. NewYork has more than 10,000 plus taxi and nearly 50% of population doesn't have a personal vehicle. Due to this fact most people used taxi has their primary mode of transport and it accounts for more than 100 million taxi trips per year.

The dataset is based on the 2016 NYC Yellow Cab trip record data made available in Big Query on Google Cloud Platform. The data was originally published by the NYC Taxi and Limousine commission (TLC). This dataset contains around 1458644 records and 11 features.

Out of numerous machine learning algorithm we have selected Xgboost and LightGBM repressors for our used case. More accurately prediction will lead to make better taxi trip duration prediction not only in NewYork but also applicable to other city as well in future and make user taking better decision for choosing right taxi to commute.

3. Trip duration & Trip Duration Variation

Trip duration normally calculated based on the distance between pickup and drop off point and average speed of the vehicle covering this distance. However, in reality there are many factors which affects the trip duration. Following are some of the factors:

- Peak hours: there are certain hours where route get busy due to moment of peoples commutating from office to home or vice versa.
- bad weather conditions (rain, snow, etc)
- big events or festivals
- traffic conditions

6. Steps involved:

- **Data Loading and general checkups :**
  I loaded the data from the given csv files using a function from pandas library. Then checked the general information about data. I observed that the data contains 1458644 records and 11 features. My data set contain three different data types i.e. floats, strings and datetime objects.
- **Null values Treatment**
  I inspected the dataset and found that my dataset has no null value present in it. So, luckily i skip this step.

- **Exploratory Data Analysis**

I begin EDA by first checking the distribution of our dependent variable i.e. trip duration. I observed that the data is highly positively skewed. I plotted the box plot and observed that there are many outliers present in the variable. To cross check this trip duration I calculated the difference in pickup and drop off timing and matched with trip duration. I observed no difference so there is no miscalculation or falsified entries. To eliminate the outliers I segregated the data variable into different segments and observed that majority of trip duration is within an hour some observation are within two days but a very few observation are having more than two days. I eliminated such values from out dataset.

I removed id variable as it doesn't give much interpretation. Then calculated the distance based on haversine formula from pickup and drop off latitude and longitude. Then I plotted the box plot for the variable and observed there are many outliers so I segregated this variable and see that most of the trip are within 10km, some trips are within 50km while a very few trip crosses 50km. so I eliminated trip with 0 and above 50km distance. I checked for categorical variable store and fwd flag and passenger_count. I observed the store and fwd flag contains majority of one category So I drop this feature. Passenger count variable has entries from 0 to 9. Since there is no trips with 0 passenger either this a miss entry or the driver forgot to enter passenger count of that trip. In a taxi maximum six person are allowed to sit including minor So I eliminate 0 and 7-9 records from our dataset.

I also created some more feature i.e. pickup month, pickup weekday and pickup hour. To get a good insight of trip duration and drop pickup date and drop-off time column. Then I checked for correlation between variables and observed that geographic coordinates are very less correlated and VIF is also high between this variables so I drop off this variable from our data set.

- **Encoding of categorical columns**
  Since some of categorical variable are in string format. So I cannot passed this variable to our model directly I use one hot encoding to convert it into numerical variable having binary integers 0 and 1.

- **Standardization of features**
  This is one of the important step for getting good accuracy as you can see there are some colomns having different ranges of values then other column. Therefore. It is important to do scaling the data so that our data set will have uniformity and to get good accuracy. So, here I use MinMaxscaler function.

- **Fitting different models** For modelling I tried various classification algorithms like:

1. **Linear Regression**
2. **XGBoost classifier**
3. **LightGBM**

- **<u>Tuning the hyperparameters</u> <u>for better accuracy</u>**

  Tuning the hyper parameters of respective algorithms is necessary for getting better accuracy.

7.1. Algorithms:

1. Linear Regression:

Linear Regression is a regression of dependent variable on independent variable. It is a linear model that assumes a linear relationship between dependent (y) and independent variables (x). The dependent variable (y) is calculated by linear combination of independent variable (x).

$$Y=B_0+B_1x_1+B_2x_2$$

The cost function for linear regression is given by: Minimum sum of square error

MSSE$= \sum_1^n (Y_i act - Y_i pred)^2$

2. XGBoost:

Sometime in building a model I cannot just rely on the result of a single model. Ensemble offer a systematic solution for this by combining the prediction of multiple model. The resultant model is superior then individual model called base learner and is obtained from aggregation of base learner prediction. Bagging and boosting are two types of ensemble method. XGBoost comes under boosting is known as extra gradient boosting. GBM first calculates the model using X and Y then after the prediction is obtain. It will again calculate the model based on residual of previous model, here loss function will give more weightage to error of previous model and this process continuous until MSE gets minimizes. XGBoost is just an extension of GBM with following advantages.

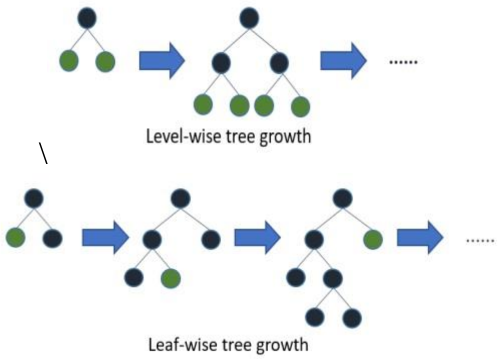
- Regularization
- Parallel Processing
- High Flexibility
- Handles Missing values
- Tree pruning
- Built in cross validation
- Continuous on existing model

3. LightGBM:

Sometime in building a model

LightGBM is a fast, distributed high performance gradient boosting framework. It is widely used for ranking, classification, regression and many other machine learning task.

LightGBM is based on decision tree algorithm. But it splits the tree leaf wise rather than level wise like other boosting algorithm.So when growing on the same leaf in Light GBM, the leaf-wise algorithm can reduce more loss than the level-wise algorithm and hence results in much better accuracy which can rarely be achieved by any of the existing boosting algorithms.



\

7.2. Model performance:

The model performance can be evaluated by various regression metrics such as:

1. Mean Squared Error (MSE) Mean squared error is the most widely used evaluation metric for regression task. It is the average of squared difference between actual and predicted value of dependent variable
   MSE$= \frac{1}{n} \sum_{1}^{n}(Y_i act - Y_i pred)^2$

2. R2 score coefficient of determination or R^2 is a metric that compares model with a constant base line model and that tell how much model is better. The constant base line model is choosen by taking the mean of the data and drawing a line at mean. The range of R^2 is always less than or equal to one no matter whether the MSE value is high or low.

$$R^2 = 1 - \frac{MSE(model)}{MSE(Baseline)}$$

7.3. Hyper parameter tuning:

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

I used Grid Search CV, Randomized Search CV and Bayesian Optimization for hyperparameter tuning. This also results in cross validation in this case I divided the dataset into different folds. The best performance improvement among the three was by Bayesian Optimization.

1. **Grid Search CV-**Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model's performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is it traverses a specific region of the parameter space and cannot understand which movement or region of the space is important to optimize the model.

2. **Randomized Search CV-** In Random Search, the hyperparameters are chosen at random within a range of values that it can assume. The advantage of this method is greater

chance of finding regions of the cost minimization space with more suitable hyperparameters, since the choice for each iteration is random. The disadvantage of this method is that the combination of hyperparameters is beyond the scientist’s control.

8. Conclusion:

That's it! We reached the end of our exercise.

Starting with loading the data so far I have done EDA , null values treatment, encoding of categorical columns, feature selection and then model building.

In all of these models our accuracy revolves in the range of 70 to 74%. And there is no such improvement in accuracy score even after hyperparameter tuning.

So the accuracy of our best model is 73% which can be said to be good for this large dataset. This performance could be due to various reasons like: no proper pattern of data, too much data, not enough relevant features.

References- 1. MachineLearningMastery
2. GeeksforGeeks
3. Analytics Vidhya