NYC Taxi Trip Time Prediction

Piyush Lanjewar
Pruthvi Raj
Yogesh Reddy

Abstract:

NYC Taxi trip time prediction to find the expected time taken for a particular trip. We design a method of predicting taxi trip time by finding historical similar trips. Trips are clustered based on origin, destination, and start time. Then similar trips are mapped to road networks to find frequent sub-trajectories that are used to model the travel time of the various parts of the routes. The data 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).

Keywords: EDA, Feature Engineering, Linear Regression, Decision tree, Random Forest, Gradient Boost, XGBoost, R2 Score, Visualizations.

1. Problem Statement

Your task is to build a model that predicts the total ride duration of taxi trips in New York City. Your primary dataset is one released by the NYC Taxi and Limousine Commission, which includes pickup time, geo-coordinates, number of passengers, and several other variables.

2. Introduction

In New York City every day too many taxis get hired by consumers to reach their destination. Several factors affect taxi time to reach a destination like traffic condition, weather condition, speed of the vehicle and festival seasons, etc. We have to take care of the components affecting time to reach the destination. There are too much of factors that correlate to the taxi time. In this project, we are taking only those factors which affect a lot the taxi time. The Data was explored and concluded two vendors give services in NYC. The data initially have 11 columns and 1458644 observations. We are taking the data from Google Cloud Platform and the data based on the 2016 NYC Yellow Cab trip record. After many operations, some business insights were calculated. From the previous data, we observed and based on that we have to build a model, which is a **supervised-learning regression problem**. I will approach the problem in several steps, namely exploratory data analysis, preprocessing phase, feature engineering, and various modeling techniques.

3. Features

The dataset has approximately 1458644 rows and 11 columns.

- Vendor_id : The Vendors who give the taxi service to NYC since 2016
- Pickup_datetime: The pickup date and time in ddmmyyyy and hhmmss format.
- Dropoff_datetime : The dropoff date and time in ddmmyyyy and hhmmss format.
- Passenger_count : How many passengers taking the taxi service at that particular ride.
- Pickup_longitude and latitude : The pickup location of passengers
- Drop_off longitude and latitude : The drop off location of the passengers
- Store_and_fwd_flag: The flag indicates whether the trip record was held in-vehicle memory before sending to the vendor
- Trip_duration: The total time taken to reach the destination in seconds

5. Observations

- **Distribution of Trip_duration:**

The total time is taken to reach from pickup location to the drop-off location. The column is taken as a dependent variable as it needs to be predicted. To take care of this column as too many outliers are present in the data. We have to remove the outliers with the help of the box plot outlier removal technique. We are removing the values which are more than one day which come under outlier and are not taken as a short ride in NYC. After plotting concluded that the data is right-skewed and then log transformation was applied to remove skewness. Finally, our dependent variable is normally distributed.

- **Latitude and Longitudes:**

  The data have pickup and dropoff locations with longitudes and latitudes. From the location, we calculated the distance between specific points. The geopy and vincenty are applied to calculate the difference between the pickup and dropoff location and the distance calculated in km.

- **Date and time:**
  The time is calculated by converting the pickup datetime and dropoff datetime column to calculate the respective time duration for specific rides.

6. Steps involved:

- **<u>Exploratory Data Analysis</u>**
  Our data has null values but too many outliers in it. We first concluded the types of vendors giving services, the distance between the respective location. To see the total number of passengers for specific trips

visually. The trip duration of more than one day was calculated and treated as outliers. Box plot used to see outliers for columns in the dataset. Visualization of pickup at different time zones. Graphical visualization of pickup and drop off locations.

- **Outliers Treatment:**

IQR is part of Descriptive statistics and is also called midspead, the middle is 50%. IQR is the first Quartile minus the Third Quartile (Q3-Q1). For finding out the Outlier using IQR we have to define a multiplier which is 1.5 ideally that will decide how far below Q1 and above Q3 will be considered as an Outlier. Any value below **Q1-1.5*IQR** or above **Q3+1.5*IQR** is an Outlier.

- **Feature Engineering:**

  We have data of date and time in object format we have change the format from which can get more insights out of it. We imported datetime library and concluded the day and month in which the ride happened. We have to understand at what time zone the trip taken like morning, evening or night. From this data we can define the traffic conditions and observed that usually at evening the traffic condition is busy while in early morning there is light traffic. We have also concluded visually that on which day usually there is a high number of trips which results to traffic. After this conclusions, we have to estimate the distance between pickup and dropoff point. The distance is calculated and based on the distance and time we calculated speed where speed = distance/time. Feature engineering is the most important step above all. We have to determine the factors which actually affect the dependent variable.

- **Training and testing dataset:**

  In this step we are dividing the dataset into two parts i.e dependent variable genreally denoted as X and dependent variable generally denoted as y. After assigning X and y we have to split the data in train and test. The data of 100% id splitted into 70% train and 30% split and the splitting percentage is based on nature of the data.

- **Model Building:**

  There were different models are used to see which model gives maximum accuarcy such as Linear regression, Lasso, Rigde, Elastic Net, Decision trees, Random forest Model, Gradient Boosting techniques, XGBoost. Finally we came to result that Random forest performing well when we are passing data into it. It give 99.20% of r2 score which is also known as goodness of fit.

- **Cross-validation:**

  Cross validation or CV is mostly used to find optimal pair of hyperparameters. We can call it as a hyperparameter tuning technique. In this step, we have to maximize the total score obtained by the desired

model, by hyperparameter tuning we can tune the parameters such that it gives optimal result. We have two methods to do hyperparameter tuning:

- Random Search CV:
  Random Search from the name we can get the idea that our data will have randomly picked combinations and out of that mathematical and statistical tools are used, we get our desired results.
- Grid Search CV:
  Grid Search CV has mostly used cross-validation techniques in the industry. In our case, we have used several hyperparameters to tune parameters. We used Grid Search CV and increased our accuracy to 0.20%.

Linear Regression:

It is machine learning algorithm based on supervised learning. These models are target prediction based on independent variables. It is moslty used for finding out the relationship between variables and forecasting.

Decision Tree:

It is the most powerful and popular model for classification and prediction. It is a flow chart like tree structure, where each internal node denotes a test on an atribute each branch represents an outcome of the test and each leaf node.

RandomForest:

It is a supervised machine learning algorithm that is used widely in classification and regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

AdaBoost:

It is also called adaptive boosting technique in machine learning used as an ensemble method The most common algorithm used with adaboost is decision trees with one level that means with decision trees with only 1 split.

Gradient Boosting:

It is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessors error. In contrast to adaboost, the weights of the training instances are not tweaked, instead each predictor is trained using the residual error of predecessor as labels.

XgBoosting:

It stands for Extreme Gradient Boosting. It is an implementation of Gradient Boosted decision trees. In this algorithm decision trees are created sequentially, where weights play an important role in XgBoost. Weights are assigned to all the independent variables. These are fed to decision tree and the output is the fed to second decision tree. This process repeats for total decision trees assigned. It can work on regression, classification, ranking and user defined prediction problems.

7. Conclusion:

- There are two vendors giving taxi service to NYC.
- There are 1450599 numbers taxis who not store and forward for the trip and there are 8045 taxis that store the number of trips and forward

- The passenger count varies from 1 to 9 where 1 passenger take the taxi service most.
- Trip duration are given in seconds and there are 4 observations in the data which show trip duration more than 1 day.
- There is a high chance to get busy traffic at evening and light traffic at early morning based on the demand of taxis at different timezones.
- Vendor 2 gives more taxi services than vendor 1.
- Our dependent variable i.e trip duration was right-skewed after being converted to normal.
- There is no high correlation between the features observed from the heatmap.
- Regression models like Linear, Lasso, Ridge, Lasso, Elastic Net are not suitable for time prediction whereas Decision trees Regressor, Random Forest, Gradient Boost Algorithm, and XGBoost.
- From the r2 score we select our model and the importance of features it is taking in the model building. Mostly The models are taking Distance, Speed, Pickup_timezone, and vendor_id to be the most important features.
- From the score we got w applied hyperparameter tunning and the score got optimized by 0.20% with Grid Search CV.

References-

1. Medium
2. GeeksforGeeks
3. Analytics Vidhya
4. StackOverflow