NYC Taxi Trip Time Prediction

Alok Kumar

Data science trainee,

Almabetter, Bangalore

Abstract:

New York City taxi rides form the core of the traffic in the city of New York. The many rides taken every day by New Yorkers in the busy city can give us a great idea of traffic times, road blockages, and so on. Predicting the duration of a taxi trip is very important since a user would always like to know precisely how much time it would require of him to travel from one place to another. Prediction of trips duration can help users to plan their trips properly, thus keeping potential margins for traffic congestions.

It can also help drivers to determine the correct route which in-turn will take lesser time as accordingly.

1. Problem Statement

A typical taxi company faces a common problem of efficiently assigning the cabs to passengers so that the service is smooth and hassle free. One of main issue is determining the duration of the current trip so it can predict when the cab will be free for the next trip.

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

Successful prediction of the taxi trip duration would eventually be much useful in the future to make better taxi trip duration predictions applicable to multiple cities.

2. Dataset description

1) **id** - a unique identifier for each trip
2) **vendor_id** - a code indicating the provider associated with the trip record
3) **pickup_datetime** - date and time when the meter was engaged
4) **dropoff_datetime** - date and time when the meter was disengaged
5) **passenger_count** - the number of passengers in the vehicle (driver entered value)
6) **pickup_longitude** - the longitude where the meter was engaged
7) **pickup_latitude** - the latitude where the meter was engaged

8) **dropoff_longitude** - the longitude where the meter was disengaged
9) **dropoff_latitude** - the latitude where the meter was disengaged
10) **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
11) **trip_duration** - duration of the trip in seconds

3. Introduction

New York City is one of the highly advanced cities of the world with extensive use of taxi services. Along with a vast population, the requirement of commonly available transportation serves the common purpose as it provides a very large transportation system. New York facilitates one of the largest subway systems in the world and comprises various green and yellow cabs which approximately count of around 13,000 taxis. Most of the population of New York depends upon public transport, and it has been estimated that 54 percent of the people do not own a car or a personal vehicle. As a matter of fact, it accounts for almost 200 million taxi trips per year.

The data set contains the data regarding several taxi trips and its duration in New York City. I will now try and apply different techniques of Data Analysis to get insights about the data and determine how different variables are dependent on the target variable Trip Duration.

4. Steps involved

1. **Exploratory Data Analysis**
   Exploratory Data Analysis is investigating data and drawing out insights from it to study its main characteristics. EDA can be done using statistical and visualization techniques. We performed this method by comparing our target variable that is trip_duration with other independent variables. Exploring and analyzing the data is important to see how features are contributing to the target variable, identifying anomalies and outliers to treat them lest they affect our model, to study the nature of the features, and be able to perform data cleaning so that our model building process is as efficient as possible.

2. **Encoding of categorical columns**
   We used One Hot Encoding to produce binary integers of 0 and 1 to encode our categorical features because categorical features that are in string format cannot be understood by the machine and needs to be converted to numerical format.

3. **Outliers Treatment**

   An outlier is an object that deviates significantly from the rest of the objects. They can be caused by measurement or execution error. Our dataset contains a large number of outliers values which might tend to disturb our accuracy hence we removed extreme values at the beginning of our project in order to get a better result.

4. **Feature Engineering**

   Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in supervised learning. In order to make machine learning work well on new tasks, it might be necessary to design and train better features.

5. **Standardization of features**

   Standardization is scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation. Our main motive through this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it.

6. **Fitting different models**

   For modelling we tried various regression algorithms like:

   1. **Linear Regression**
   2. **DecisionTree Regressor**
   3. **XGBRegressor**
   4. **GradientBoosting**

7. **Hyperparameters tuning**

   Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in case of tree based models like DecisionTree Regressor, XGBRegressor and Gradient Boosting.

8. **Feature Importance**

   We have applied feature importance on models to determine the features that were most important while model building and the features that didn't put much weight on the performance of our model. Inspecting the importance score provides insight into that specific model and which features are the most important and least important to the model when making a prediction.

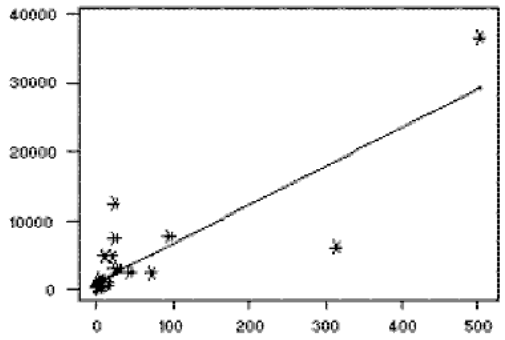
5. Algorithms

1. Linear Regression:

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory

variable, and the other is considered to be a dependent variable.

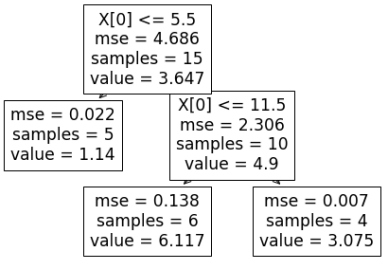
A linear regression line has an equation of the form Y = a + bX, where X is the explanatory variable and Y is the dependent variable. The slope of the line is b, and a is the intercept (the value of y when x = 0).



The most common method for fitting a regression line is the method of least-squares. This method calculates the best-fitting line for the observed data by minimizing the sum of the squares of the vertical deviations from each data point to the line

2. Decision Tree for Regression

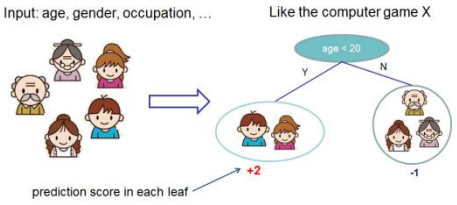
A regression tree is basically a decision tree that is used for the task of regression which can be used to predict continuous valued outputs instead of discrete outputs.



The basic idea behind the algorithm is to find the point in the independent variable to split the data-set into 2 parts, so that the mean squared error is the minimized at that point. The algorithm does this in a repetitive fashion and forms a tree-like structure.

3. XGBoost for Regression

XGBoost stands for "Extreme Gradient Boosting" and it is an implementation of gradient boosting trees algorithm. In boosting, the trees are built sequentially such that each subsequent tree aims to reduce the errors of the previous tree. Each tree learns from its predecessors and updates the residual errors.



The base learners in boosting are weak learners in which the bias is high, and the predictive power is just a tad better than random guessing. Each of these weak learners contributes some vital information for prediction, enabling the boosting technique to produce a strong learner by effectively combining these weak learners. The final strong learner brings down both the bias and the variance.

4. GradientBoosting

The main idea behind this algorithm is to build models sequentially and these subsequent models try to reduce the errors of the previous model. When the target column is continuous, we use Gradient Boosting Regressor. The objective here is to minimize the loss function by adding weak learners using gradient descent. Since it is based on loss function hence for regression problems, we have a loss functions like Mean squared error (MSE). The principle behind boosting algorithms is first we built a model on the training dataset and then a second model is built to rectify the errors present in the first model.

6. Model performance

The essential step in any machine learning model is to evaluate the accuracy of the model. The Mean Squared Error, Root Mean Squared Error, and R-Squared or Coefficient of determination metrics are used to evaluate the performance of the model in regression analysis.

1. **Mean Squared Error**
   The Mean Squared Error represents the average of the squared difference between the original and predicted values in the data set. It measures the variance of the residuals.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

2. **Root Mean Squared Error**
   Root Mean Squared Error is the square root of Mean Squared error. It measures the standard deviation of residuals.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

3. **R-squared**
   The coefficient of determination or R-squared represents the proportion of the variance in the dependent variable which is explained by the linear regression model. It is a scale-free score i.e. irrespective of the values being small or large, the value of R square will be less than one.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

4. **Adjusted R squared**
   Adjusted R squared is a modified version of R square, and it is adjusted for the number of independent variables in the model, and it will always be less than or equal to R^2.In the formula below n is the number of observations in the data and k is

the number of the independent variables in the data.

$$R^2_{adj} = 1 - \left[\frac{(1 - R^2)(n-1)}{n-k-1} \right]$$

7. Conclusion

- The summary of all models that we've trained and tested so far. We conclude that **GradientBoosting** has the highest score of 0.737 followed by **XBGRegressor** 0.731. Whereas **DecisionTree Regressor** and **Linear Regression** had a score of 0.62 and 0.492 respectively, hence making them unfit to be used for our data