**NYC Taxi Trip Time Prediction**

**Dharmeshbhai Patel**

**(PRO-Program)**

**Abstract:**

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 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.

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 data was sampled and cleaned for the purposes of this project. Based on individual trip attributes, we have to predict the duration of each trip in the test set.

* **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
* **dropoff\_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
* **dropoff\_longitude** - the longitude where the meter was disengaged
* **dropoff\_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

1. **Introduction:**

In today’s world it has become a race to gain more and more number of customers.

To gain more number of customers companies/vendors usually try to provide their customers with more comfort to attract them.

So here we will be predicting the time of trip duration our customers will take and which algorithm is best suited for that time prediction.

* The dataset contains 1458644 rows and 11 features(columns)
* Two categorical features ‘store\_and\_fwd\_flag’ and ‘vendor\_id
* Outliers present in all numerical features
* Data cleaning steps required for datetime features
* No null values present

1. **Steps Involved:**

* Exploratory Data Analysis

After loading the dataset we converted the pickup and drop-off date to days and timing into four category of morning, afternoon, evening and late night.

From pick up and drop-off latitude we find the trip distance which will help for predicting the duration

* Fitting different models

For modelling we tried various classification algorithms like:

1. Linear Regression
2. Decision Tree
3. Random Forest Classifier
4. XGBoost classifier
5. Gradient boosting Algorithm

* Tuning the hyperparameters for better accuracy

Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in case of tree based models

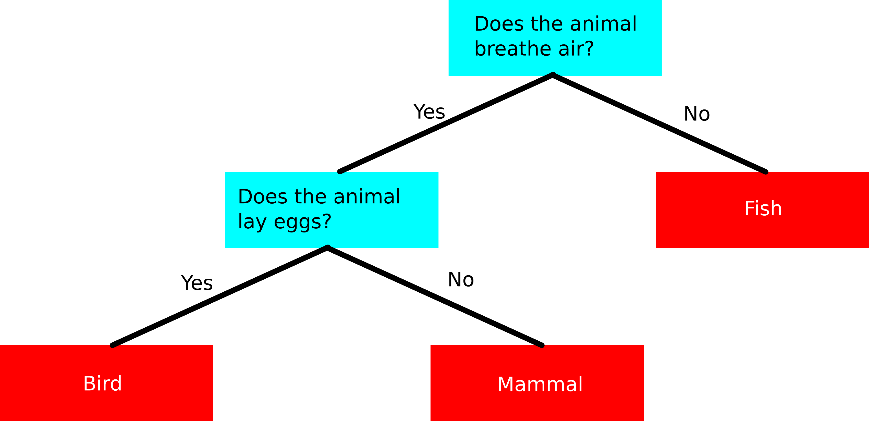
Random Forest Classifier and XGBoost classifier

1. **Algorithms:**
2. **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. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

1. **Decision Tree**

A decision tree is a flowchart-like diagram that shows the various outcomes from a series of decisions. It can be used as a decision-making tool, for research analysis, or for planning strategy. A primary advantage for using a decision tree is that it is easy to follow and understand.



1. **XGBoost**

To understand XGBoost we have to know gradient boosting beforehand.

● Gradient Boosting-

Gradient boosted trees consider the special case where the simple model is a decision tree

****

In this case, there are going to be 2 kinds of parameters P: the weights at each leaf, w, and the number of leaves T in each tree (so that in the above example, T=3 and w=[2, 0.1, -1]).

When building a decision tree, a challenge is to decide how to split a current leaf. For instance, in the above image, how could I add another layer to the (age > 15) leaf? A ‘greedy’ way to do this is to consider every possible split on the remaining features (so, gender and occupation), and calculate the new loss for each split; you could then pick the tree which most reduces your loss.

XGBoost is one of the fastest implementations of gradient boosting. trees. It does this by tackling one of the major inefficiencies of gradient boosted trees: considering the potential loss for all possible splits to create a new branch (especially if you consider the case where there are thousands of features, and therefore thousands of possible splits). XGBoost tackles this inefficiency by looking at the distribution of features across all data points in a leaf and using this information to reduce the search space of possible feature splits

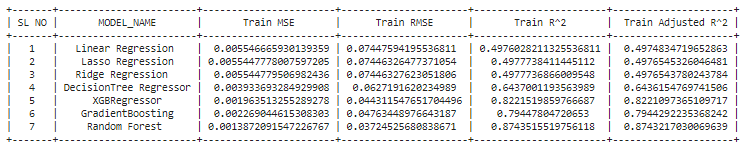
1. **Random Forest classifier**

Random Forest is a bagging type of Decision Tree Algorithm that creates a number of decision trees from a randomly selected subset of the training set, collects the labels from these subsets and then averages the final prediction depending on the greatest number of times a label has been predicted out of all.

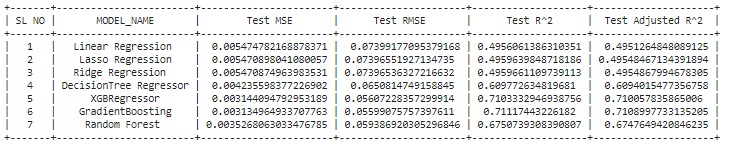


1. **Model Performance:**

* Training Set Comparison



* Test Set Comparison



1. **Conclusion:**

* In this project, we tried to predict the trip duration of a taxi in NYC.
* We are mostly concerned with the information of pick up latitude and longitude and drop off latitude and longitude, to get the distance of the trip.
* Gradient Boosting will be the best model to predict the trip duration for a particular taxi.