**NYC Taxi trip Duration Prediction**

Almabetter, Bangalore

Neha Pasi

**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 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.**

**Introduction to EDA**:

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.

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.

If we don’t perform exploratory data analysis, we won’t be able to find inconsistent or incomplete data that may pose trends incorrectly to our model.

This step also serves as the basis for answering our business questions.

**Introduction to Supervised Regression Machine learning**:

In supervised learning, you train your model on a labelled dataset that means we have both raw input data as well as its results. We split our data into a training dataset and test dataset where the training dataset is used to train our network whereas the test dataset acts as new data for predicting results or to see the accuracy of our model.

Hence, in supervised learning, our model learns from seen results the same as a teacher teaches his students because the teacher already knows the results. Accuracy is what we achieve in supervised learning as model perfection is usually high.

The model performs fast because the training time taken is less as we already have desired results in our dataset. This model predicts accurate results on unseen data or new data without even knowing a prior target. In some of the supervised learning models, we revert back the output result to learn more in order to achieve the highest possible accuracy.

Data Summary:

Based upon the initial assessment we found that the data was pretty much clean except for Outliers in some columns. We draw out the following key insights about the data:-

1. The dataset has a shape of (1458644, 11) which means that it contains approximately 1458644 rows and 11 columns.
2. Our dataset has 4 columns with float64 dtype, 3 columns with int64 dtype and 4 columns with object dtype.
3. In our dataset, we observed that we don’t have any null values in any columns:

We have the following column provided to us in the dataset:

#### 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

#### Steps Involved:

#### Exploratory Data Analysis

After loading the dataset we performed this method by comparing our target

variable that is Trip duration with other independent variables. This process

helped us figuring out various aspects and relationships among the target and

the independent variables. It gave us a better idea of which feature behaves

in which manner compared to the target variable.

* Null Values Treatment

We don’t have any null values in our dataset.

* 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.

* Independent and Dependent Variable Selection

In this step we select appropriate dependent and and independent variable for making prediction.

* Split the Dataset into Train and Test Sets

We need to split a data set into train and test sets to evaluate how well our machine learning model performs. The train set is used to fit the model. The second set is called the test data set, this set is solely used for predictions.

* Fitting different Models

For modeling we tried various regression algorithms like:

1. Linear Regression
2. Decision Tree Algorithm
3. Random Forest Algorithm
4. XG Boost Algorithm

**Algorithms**

**Linear Regression**:

Linear Regression is of two types: Simple and Multiple. Simple LinearRegression is where only one independent variable is present and the model has to find the linear relationship of it with the dependent variable

Whereas, In Multiple Linear Regression there are more than one independent variables for the model to find the relationship.

Equation of Simple Linear Regression, where bois the intercept, b1 is coefficient or slope, x is the independent variable and y is the dependent variable.

linear regression 1

Equation of Multiple Linear Regression, where bo is the intercept, b1,b2,b3,b4…,bn are coefficients or slopes of the independent variables x1,x2,x3,x4…,xn and y is the dependent variable.

linear regression 2

A Linear Regression model’s main aim is to find the best fit linear line and the optimal values of intercept and coefficients such that the error is minimized.  
Error is the difference between the actual value and Predicted value and the goal is to reduce this difference.

Let’s understand this with the help of a diagram.

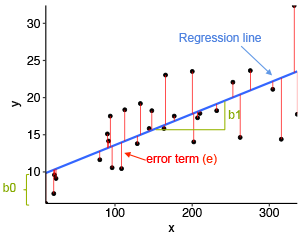


Image Source: Statistical tools for high-throughput data analysis

In the above diagram,

* x is our dependent variable which is plotted on the x-axis and y is the dependent variable which is plotted on the y-axis.
* Black dots are the data points i.e the actual values.
* bo is the intercept which is 10 and b1 is the slope of the x variable.
* The blue line is the best fit line predicted by the model i.e the predicted values lie on the blue line.

The vertical distance between the data point and the regression line is known as error or residual**.** Each data point has one residual and the sum of all the differences is known asthe Sum of Residuals/Errors.

**Mathematical Approach:**

Residual/Error = Actual values – Predicted Values

Sum of Residuals/Errors = Sum(Actual- Predicted Values)

Square of Sum of Residuals/Errors = (Sum(Actual- Predicted Values))2

i.e

linear regression

**Lasso Regression**

This is a regularization technique used in feature selection using a Shrinkage method also referred to as the **penalized regression method.** Lasso is short for **L**east **A**bsolute **s**hrinkage and **s**election **o**perator, which is used both for regularization and model selection. If a model uses the **L1 regularization** technique, then it is called lasso regression.

### Lasso Regression for Regularization

In this shrinkage technique, the coefficients determined in the linear model from equation 1.1. above are shrunk towards the central point as the mean by introducing a penalization factor called the alpha α (or sometimes lamda) values.

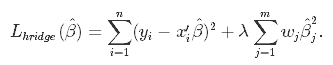
https://res.cloudinary.com/dyd911kmh/image/upload/v1648205671/image17_ryqgum.png

Alpha (α) is the penalty term that denotes the amount of shrinkage (or constraint) that will be implemented in the equation. With alpha set to zero, you will find that this is the equivalent of the linear regression model from equation 1.2, and a larger value penalizes the optimization function. Therefore, lasso regression shrinks the coefficients and helps to reduce the model complexity and multi-collinearity.

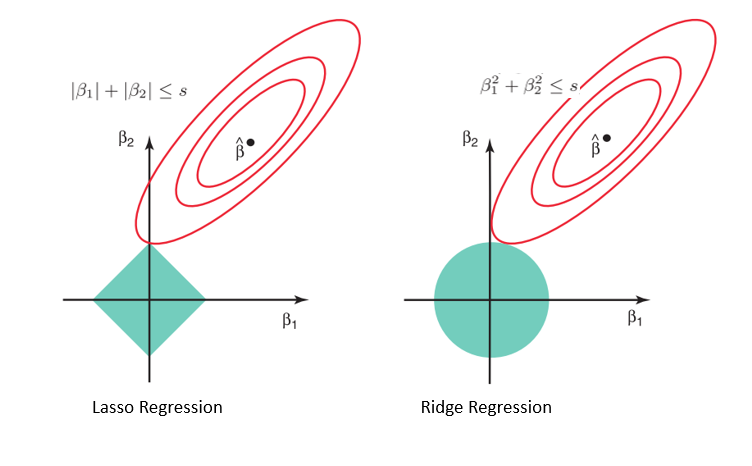
Alpha (α) can be any real-valued number between zero and infinity; the larger the value, the more aggressive the penalization is.

## **Ridge Regression**

Similar to the lasso regression, ridge regression puts a similar constraint on the coefficients by introducing a penalty factor. However, while lasso regression takes the magnitude of the coefficients, ridge regression takes the square.



Ridge regression is also referred to as **L2 Regularization**.



Random Forest Classifier Algorithm

A random forest is a supervised machine learning method built from decision tree techniques. This algorithm is used to anticipate behaviour and results in a variety of sectors, including banking and e-commerce.

A random forest is a machine learning approach for solving regression and classification issues. It makes use of ensemble learning, which is a technique that combines multiple classifiers to solve complicated problems.

A random forest method is made up of a large number of decision trees. The random forest algorithm’s ‘forest’ is trained via bagging or bootstrap aggregation. Bagging is a meta-algorithm ensemble that increases the accuracy of machine learning algorithms.

The outcome is determined by the (random forest) algorithm based on the predictions of the decision trees. It forecasts by averaging or averaging the output of several trees. The precision of the outcome improves as the number of trees grows.

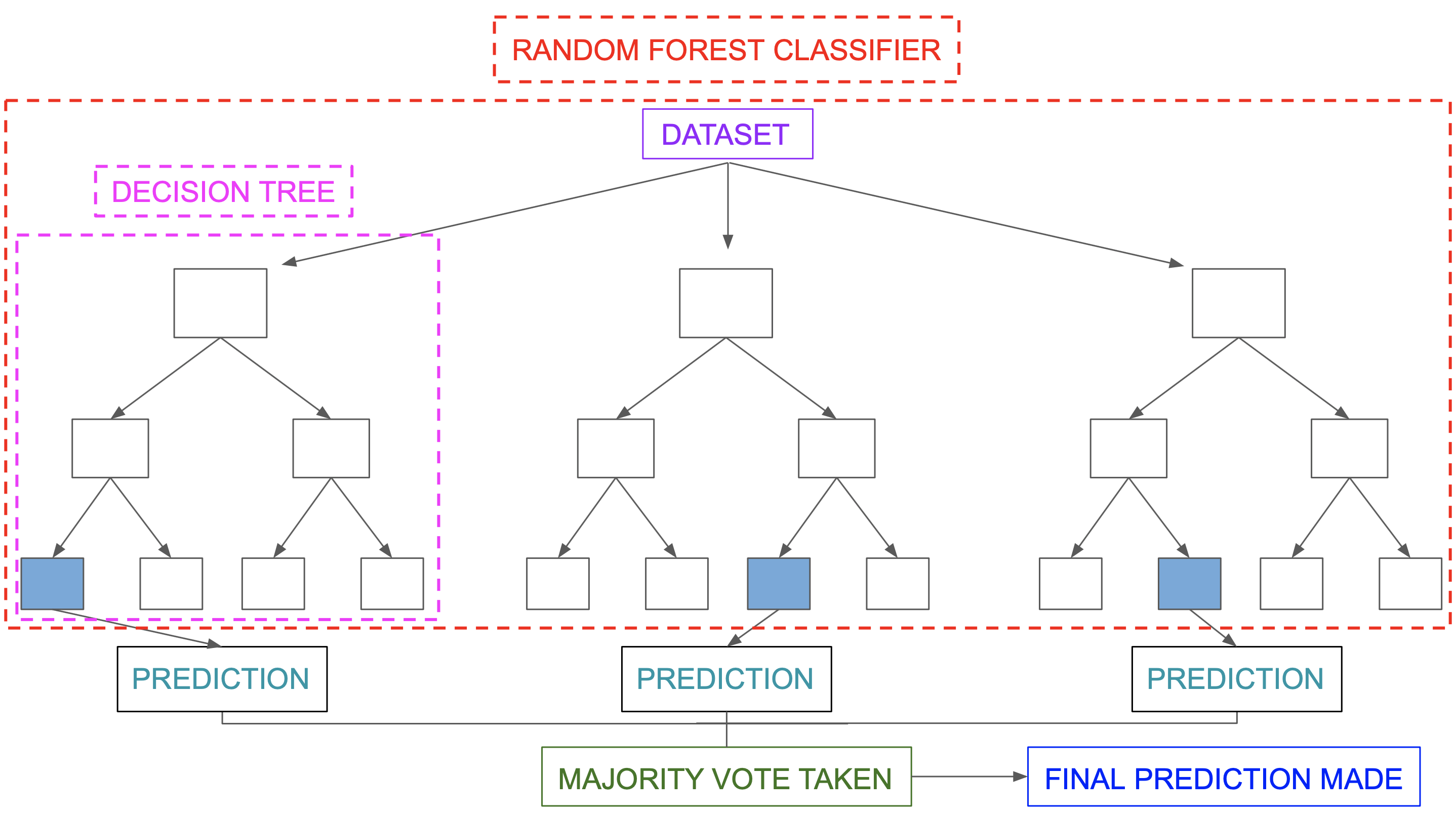


Image Source: https://miro.medium.com/max/5752/1\*5dq\_1hnqkboZTcKFfwbO9A.png

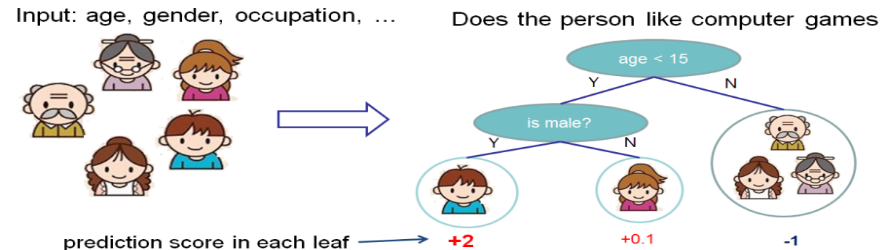
A random forest system is built on a variety of decision trees. Every decision tree is made up of nodes that represent decisions, leaf nodes, and a root node. The leaf node of each tree represents the decision tree’s final result. The final product is chosen using a majority-voting procedure. In this situation, the output picked by the majority of the decision trees becomes the random forest system’s ultimate output. Let us now implement the random forest algorithm.

XGBoost Algorithm

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.

**Conclusion**:

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

Starting with loading the data so far we 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 90 to 99%.

And there is no such improvement in accuracy score even after applying Lasso and Ridge Regression in linear regression model

So the accuracy of our best model is XGBoost is 99%