

***“Trip Distance Prediction & Result Comparison using Machine Learning”***

A PROJECT REPORT  
SUBMITTED IN THE PARTIAL FULFILMENT OF THE REQUIREMENTS  
FOR THE AWARD OF DEGREE  
OF  
MASTER OF TECHNOLOGY  
IN  
SOFTWARE ENGINEERING

Submitted By

**Rishabh Malpani**  
**(2K19/SWE/11)**

Under the supervision of

**Dr. Rajni Jindal**  
Head Of Department  
Department of Computer Science & Engineering  
Delhi Technological University, Delhi



**DEPARTMENT OF SOFTWARE ENGINEERING**

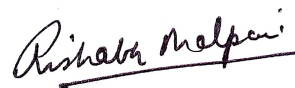
DELHI TECHNOLOGICAL UNIVERSITY  
(Formerly Delhi College of Engineering)  
Bawana Road, Delhi-110042

JUNE, 2021

## **DECLARATION**

I, **Rishabh Malpani, 2K19/SWE/11** student of **M.Tech (SWE)**, hereby declare that the project entitled “**Trip Distance Prediction & Result Comparison using Machine Learning**” is submitted by me to the Department of Computer Science & Engineering, **Delhi Technological University**, Shahbad Daulatpur, Delhi. I have done my project in partial fulfilment of the requirement for the award of the degree of Master of Technology in Software Engineering and it has not been previously formed the basis for any fulfilment of the requirement in any degree or other similar title or recognition.

This report is an authentic record of my work carried out during my degree under the guidance of **Dr. Rajni Jindal**.

A handwritten signature in black ink, reading 'Rishabh Malpani', with a horizontal line drawn underneath it.

Place: Delhi

Date: 24<sup>th</sup> June, 2021

**Rishabh Malpani**

**(2K19/SWE/11)**

## **CERTIFICATE**

I hereby certify that the project entitled “**Trip Distance Prediction & Result Comparison using Machine Learning**” which is submitted by **Rishabh Malpani (2K19/SWE/11)** to the Department of Computer Science & Engineering, Delhi Technological University, Shahbad Daulatpur, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology in Software Engineering, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any degree or diploma to this university or elsewhere.



Place:

**Dr. Rajni Jindal**

Date:

**SUPERVISOR**

**Head Of Department**

**Dept. of Computer Science & Engineering**

## **ACKNOWLEDGEMENT**

I am very thankful to **Dr. Rajni Jindal** (Head Of Department, Department of Computer Science & Engineering) and all the faculty members of the Department of Computer Science at DTU. They all provided me with immense support and guidance for the project.

I would also like to thank the university for providing the laboratory, infrastructure, test facilities and environment so that I can work without obstacles.

I would also like to thank our lab assistants, seniors, and peer groups for providing me with all the knowledge on various topics.

**Rishabh Malpani**

**(2K19/SWE/11)**

## **ABSTRACT**

We are probably living in the clearest era of human history. An era in which computing has moved from large-scale mainframes to PCs and the cloud. But that's not what happened, it's something we can think of over the years to come. **Trip Distance Prediction** is important in the development of mobility-on-demand and travel information systems. Accurate estimates of travel distance support the decision-making process for riders and drivers using such systems. In this project, the static trip distance of a taxi trip trajectory is predicted by applying some regression model to a highly conditioned set of trips. We are using the NYC Taxi data set which is available on Kaggle in which, so many rich features are present like Locations, duration Distance etc. Also, we are going to use classification on the datasets and predict the Trip type. It is important to compare the results for all the different Algorithms so that we can analyse the best algorithm.

## CONTENTS

CANDIDATE'S DECLARATION.....	II
CERTIFICATE.....	III
ACKNOWLEDGEMENT.....	IV
ABSTRACT.....	V
<b>CHAPTER 1: INTRODUCTION.....</b>	<b>10</b>
1.1 Machine Learning.....	10
1.2 Type of Machine Learning.....	10
1.2.1 Based on the nature of the learning "signal" or "feedback" .....	10
1.2.2 Based on the "output" required for machine learning systems.....	12
1.3 Terminologies of Machine Learning.....	12
1.4 Objective of the Project.....	13.
<b>CHAPTER 2: THEORITICAL CONCEPT.....</b>	<b>14</b>
2.1 Trip Distance Prediction.....	14
2.2 Why Trip Distance Prediction.....	14
2.3 Datasets.....	14
2.3.1 Multicollinearity (Using VIF).....	14
2.4 Algorithm.....	14
2.5 Metrics for performance (Regression).....	19
2.6 Metrics for performance (Classification).....	20
<b>CHAPTER 3: PROPOSED MODEL FOR DISTANCE PREDICTION.....</b>	<b>23</b>
3.1 Datasets.....	23
3.2 Data Manipulation.....	23
3.3 EDA(Exploratory Data Analysis).....	24
3.4 Data Pre-processing.....	27
<b>CHAPTER 4: RESULTS.....</b>	<b>29</b>
4.1 Result for Regression.....	29
4.2 Result for Classification.....	31
<b>CHAPTER 5: CONCLUSION AND FUTURE SCOPE.....</b>	<b>33</b>
REFERENCES.....	34

## **List of Figures**

Fig.1: AI and Machine Learning.....	10
Fig.2: Supervised & Unsupervised Learning.....	11
Fig.3: Classification & Regression.....	12
Fig.4: Training & Prediction.....	13
Fig.5: Linear Regression.....	15
Fig.6: Logistic Regression.....	16
Fig.7: Part of Decision Tree.....	18
Fig.8: AdaBoost for Decision Tree.....	18
Fig.9: Confusion Matrix.....	20
Fig.10: AUC and ROC.....	21
Fig.11: Confusion Matrix and different Metrics.....	22
Fig.12: Box Plot for All numerical features.....	24
Fig.13: Univariate Analysis.....	25
Fig.14: Bivariate Analysis.....	26
Fig.15: Histogram.....	26
Fig.16: HeatMap.....	27
Fig.17: Correlation Heat Graph.....	27
Fig.18: Result Graph (Actual V/S Predicted).....	29
Fig.19: ROC curve Graph.....	31

## **List of Tables**

Table.3: Original Dataset .....	23
Table.2: Dataset Dictionary.....	23
Table.3: Manipulated Dataset.....	24
Table.4: Correlation Matrix.....	27
Table.5: VIF Table (Multicollinearity).....	27
Table.6: One-Hot Encoding.....	28
Table.7: Comparison Between all the Regression Algorithm.....	30
Table.8: Comparison Between all the Classification Algorithm.....	32



### **List of symbols and abbreviations**

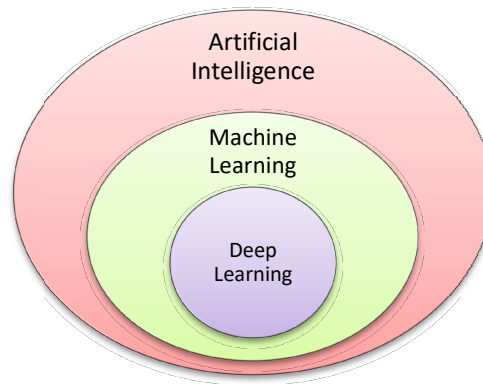
<b>Abbreviations</b>	<b>Full Form</b>
ML	Machine Learning
AI	Artificial Intelligence
ROC Curve	Receiver Operating Characteristic Curve
AUC	Area Under ROC Curve
R <sup>2</sup>	R-Square Score
MAE	Mean Absolute Error
MSE	Mean Square Error
RMSE	Root Mean Square Error
VIF	Variance Inflation Factor

# CHAPTER 1

## INTRODUCTION

### 1.1 Machine Learning

It is the most interesting technology. It is a area of learning which allows system to learn with no need of complex programming. As the name implies, this gives us the system the ability to get more learning. ML is probably used in more places today than we might imagine.



*Fig.4: AI and Machine Learning*

### 1.2 Types of Machine Learning

#### 1.2.1 Based on the nature of the learning

**Supervised Learning:** The example of the input and required output given by the "teacher" is presented to the computer. The objective is to understand the important and basic rules for comparing input data and output data. The training process goes on as the model reaches to the desired level of performance score in the training data. Here are some real-time examples:

**Classification of Image:** Train with images / labels. It then gives a new image in the future, hoping that the system will recognize the new object.

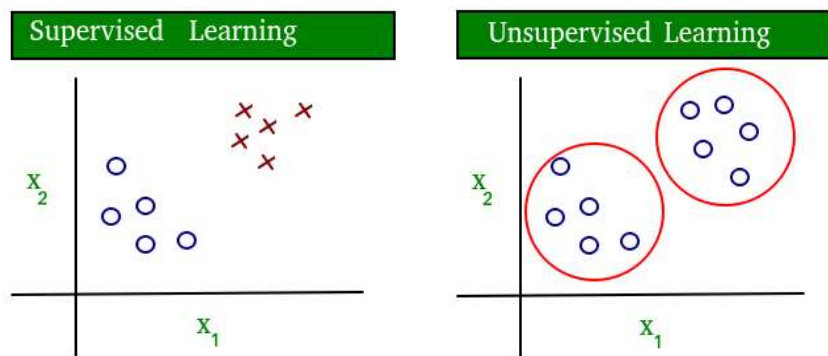
**Market Forecast / Regression:** Train your system with previous market data and get new data from your system to predict new prices in the future.

**Unsupervised Learning:** Algorithms of learning is unlabeled, leaving behind its own algorithm for finding structures in the input. This type of learning is used to cluster the population of different groups or area. This method of learning can be a objective in its own right.

**Clustering:** It is use to instructs the system to divide same type of data into groups. In research and science, we use this method mostly.

**High-dimensional visualization:** We use our system to visualize data having high-dimensionality.

**Generative model:** More data can be generated when the model gets the probability distribution of the input data. This is very helpful in making the classifier more robust.



*Fig.2: Supervised & Unsupervised Learning*

Of course, supervised learning data is labeled, but unsupervised learning data is unlabeled.

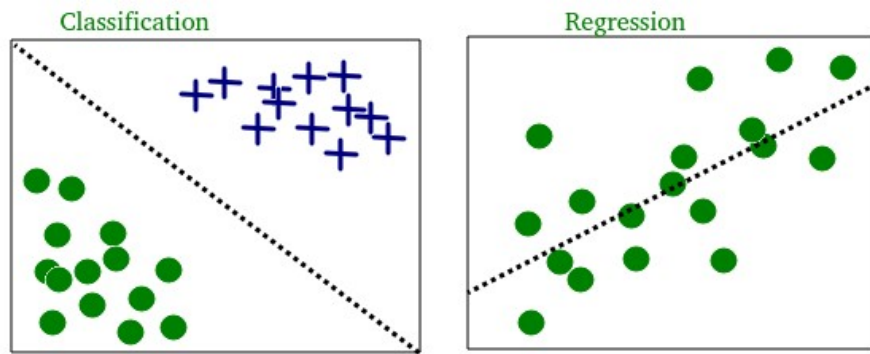
**Semi-supervised learning:** When a problem contain a very big data and less of labled data, so this type of learning is known as semi-supervised learning problem. Those issue lie between both the supervised learning and unsupervised learning. For example, Suppose we have a photo gallery (dogs, cats, people, etc.) and some photos are labeled and most images are unlabeled.

**Reinforcement learning:** Computer programs interact with dynamic environments that need to perform specific goals (such as driving a car or playing games with opponents).

### 1.2.2 Based on the "output" required for machine learning systems

**Classification:** In this Method, we have the input and we divide or categorise that input into two or more classes, and then a model will be created by the learner which assigns hidden input to the classes which is known as multi-label classification. This is generally handled in a supervised way. Filtering the spam is a best example of it, where input will be the email sent by user and the classes will be spam or not spam.

**Regression:** This method also comes under supervised learning problem. Here we see that output is continuous unlike in Classification where output is discrete. For example, use historical data to predict stock prices.



*Fig.3: Classification & Regression*

### 1.3 Some Machine Learning Terminologies

**Model:** A model or we say **hypothesis** is a **specific representation**. When we apply Machine Learning Algorithms, then the model learns from it and gives the required Outputs.

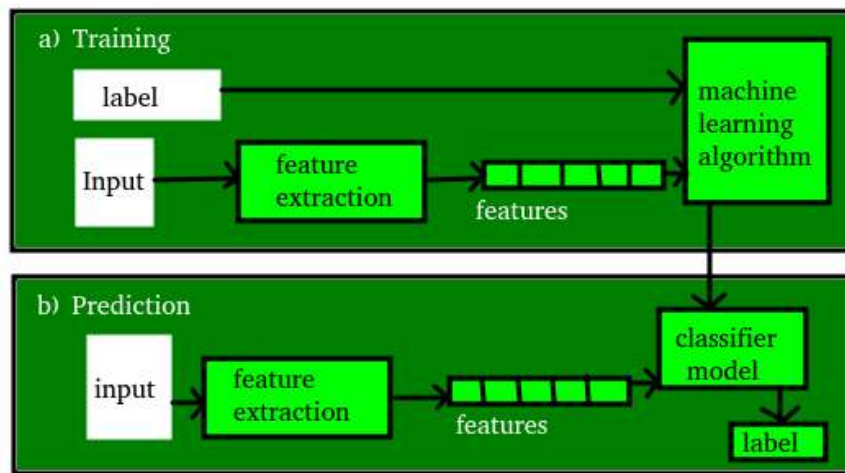
**Features:** Features are the individual measurable properties of the data. A series of numerical features can be easily explained by **feature vectors**. The feature vector is provided as input to the model. For example, you can predict fruits from characteristics such as color, smell, and taste.

**Target:** The values which we predicted by the model is known as target variable or label. For example, Suppose we need to find the name of fruits which

is given in Features Section, so Fruit will be the label which gives the fruit value.

**Training:** This is a process of mapping where we have already the input data, features and target (label), and required outputs. So in this we use all the data and create a model which maps the new data to its trained classes.

**Forecast:** When the model is ready, you can enter a set of inputs that provide predictive output (labels).



*Fig.4: Training & Prediction*

### 1.4 Objectives of the Project

In this project, the following objectives need to be achieved:

1. Generating a custom dataset for training and validating the ML model.
2. Training and validating the performance of ML models.
3. Comparing the performance of different ML models available.

## **CHAPTER 2**

### **THEORITICAL CONCEPT**

This section presents the basic theoretical concepts needed to understand the main processes and tasks of the experiments studied in this project. This section details the concepts of machine learning, transfer learning, and the available pre-trained models. It also gives rise to the idea of using different types of layers that are used in different pre-trained models. The concepts presented in this section will help you understand the proposed Trip Distance Prediction.

#### **2.1 Trip Distance Prediction**

This is the process to predict the distance of a vehicle going to covered. We use Supervise Learning here because we need to first see the past records related to the vehicle so that we can calculate and estimate the distance which will be going to covered by the vehicle under the circumstances.

#### **2.2 Why Trip Distance Prediction**

This Project help us for so many fields:

- Keep proper maintenance of the vehicle.
- Easy to estimate the fuel quantity.
- Very Helpful for Taxi /Cab company.
- Automobiles sector can maintain the service records.

#### **2.3 Datasets**

Datasets are more important for any supervise learning because it help us to understand the features related to vehicle. Datasets can be available publically related to the field. After getting the proper datasets we can manipulate it according to the requirements.

##### **2.3.1 Multicollinearity (Using VIF)**

We use Variance inflation factor (VIF), when we need to calculate the value of multicollinearity . This is use generally for Regression model. So the VIF is define as the ratio of the variance of the whole model to the variance of the model containing only its one independent variable Mathematically.

## 2.4 Algorithms

In this Project, Classification & Regression algorithms are used so we will discuss the algorithm having high performance in its field.

- **Linear Regression:** As we know that the objective of ML is to calculate the relationship between the input and the output data, so we use input variable(x) to get the output variable(y) with the help of Linear Regression. In linear regression, the equation of relationship is expressed as  $y = a + bx$ . Hence, the purpose of this regression is to calculate the coefficients value a and b. Where

a : intercept

b : slope of a line.

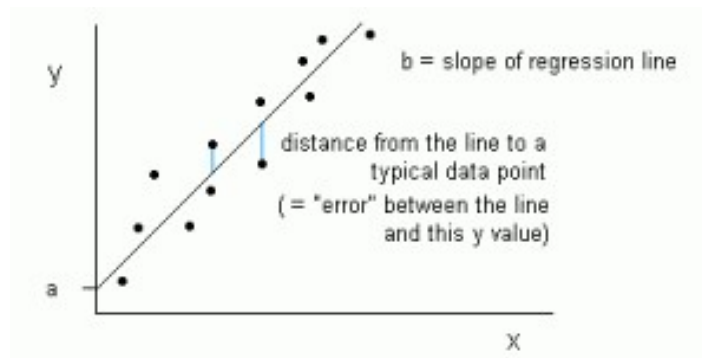


Fig.5: Linear Regression

Figure 5, here we can see the graph representation for x and y against the data. Here the objective is to draw the line nearest to most of points. So we can understand that if the distance is reduced means the error percentage is also reduced.

- **Logistic Regression:** As we know in Linear Regression, we predict the numerical results like marks of student in subjects, but in logistic regression we predict the discrete values like whether the student is pass or fail.

We use binary classification as a term in this method . That is, for datasets with  $y = 0$  or  $1$ ,  $1$  indicates the default class. For example, there are only two possibilities for predicting whether an event will occur. Does it occur (indicated by  $1$ ) or does not occur ( $0$ )? Therefore, if you want to predict whether a patient is ill, use the value  $1$  in the dataset to label the sick patient. Here in this type of classification technique, we use probability for each of the class unlike any of regression techniques where we generate output directly. This is why, the range of output will be  $0$  to  $1$ . For example, if you are trying to predict if a patient is ill, you already know that the sick patient will be displayed as  $1$ , now if the method has a patient value is  $0.91$ , the patient may be ill.

Logarithmically, by converting input value( $x$ ) using logistic function which is  $[h(x) = 1 / (1 + e^{-x})]$ , we generate the output value( $y$ ). Then we convert this probability into a binary method under classification.

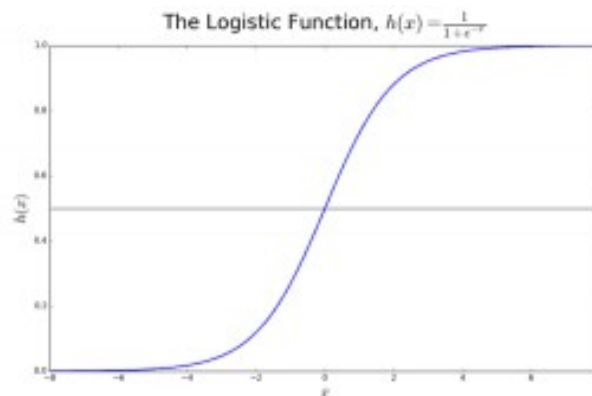


Fig.6: Logistic Regression

By using this method, main objective is to train the data and find the coefficients and minimize the errors. We will have actual and predicted value so according to the result we can find the results and minimize the error.

- **Naïve Bayes:** We use a theorem in this method, which is known as Bayes Theorem as we need to calculate the probability here which will tell us that given event is going to happen if another one has already happened. So we find the probability that hypothesis ( $h$ ) is true, we use the theorem as follows, given prior knowledge ( $d$ ).



$$P(h|d) = (P(d|h) P(h)) / P(d)$$

We use the term "naive" here as the variables are independent of each other.

- **KNN:** KNN stands for K-Nearest Neighbour. As we know that first we divide whole data into two set (ie. Test datasets and train datasets). But in KNN we use whole data set as a train data set. Means we do not use test data set separately.

When we need to find result using new dataset, This method of classification examines the data and get the k sections related to the data where k is define by the user like how many section user want. And then it gets the k number of results from all section so it take the average of the results for regression method and also take the most frequent class for classification method. We calculate similarities between instances using Euclidean distance and Hamming distance.

- **Decision Tree:** In this machine learning method, we use the classification and regression based on tree where non-leaf nodes in classification tree are the root node and the non-leaf nodes in regression tree are the internal node. Non-leaf node is also known as non-terminal nodes as terminal node is known as leaf node where leaf node represents the output value(y) and non-leaf node is represented as single input value(x). Here we do prediction by traversing the tree to reach to the leaf node using non-leaf node as there are certain conditions to select the non-leaf node. And hence we reach to the result node.

In Figure 7 below, we see that if a person is going to buy a sports car or minivan. This prediction is taking place using Decision Tree method where, it classifies based on the age and the marriage status. This figure shows that if a person's age is 30 years and that person is not married, then what will be the result. So this Decision Tree is showing that the person will buy a sports car as shown below:

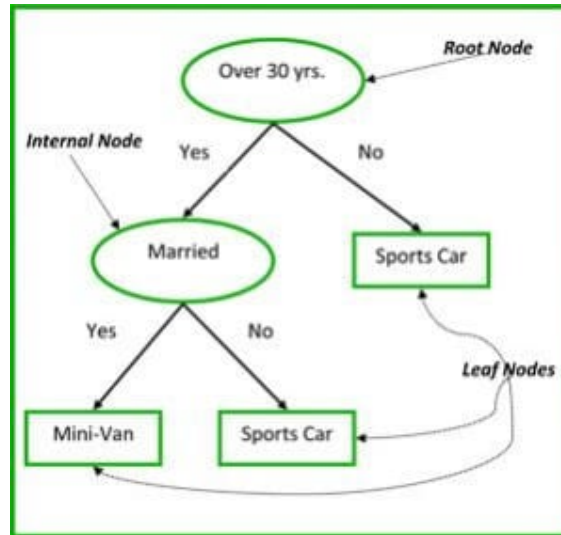


Fig.7: Part of Decision Tree

- AdaBoost** : In simple word we use term AdaBoost for Adaptive Boosting. This is a ensemble techniques which comes under Classification method. The difference between Bagging and Boosting is that as each model built independently that is why Bagging is parallel ensemble and Boosting is sequential ensemble. Because each model build based on the previously build model.

Bagging is like a voting process where classifier votes for the final result as the bagging is parallel ensemble. But in Boosting its kind of a Weighted Voting where based on majority, classifiers give vote to get the final result.

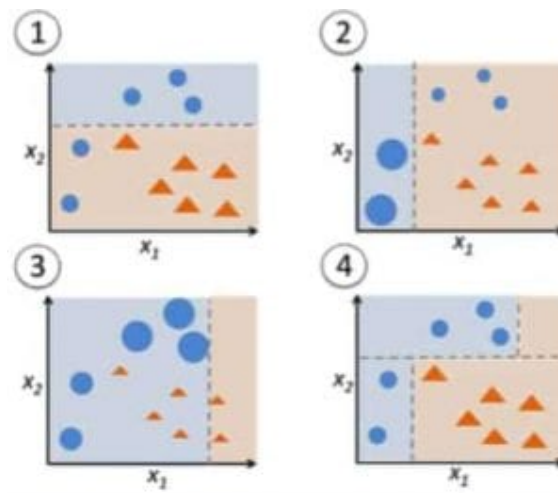


Fig.8: AdaBoost for Decision Tree

## 2.5 Metrics for performance (Regression)

As regression comes under the supervise learning , we are going to map the actual value and the predicted value and the relationship shows that how good we built our model.

The metrics for performance used to evaluate the regression model are as follows:

### 1. MAE

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

MAE measures the average magnitude of a series of prediction errors, regardless of the direction of the prediction. Measures the precision of continuous variables. The equations are listed in the library reference. In simple words, we get the average score of all the error scores per records.

### 2. MSE

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

Mean squared error (MSE) helps us to visualize the results by drawing a curve which shows that the regression line is close to the data set or not. if it is close means the error is less. So we calculate the distance from the regression line to the data points and that distance is nothing but the errors. So we calculate the square of errors. We need a square to get rid of the negative sign. It also gives more weight to the big difference. This is called the root mean square error because it averages a series of errors. The lower the MSE, the better the prediction.

### 3. RMSE

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$$

RMSE is nothing but a square root of the MSE hence RMSE is stands for Root Mean Square Error. It is also use to find the average value of the errors. It is generally used when the error is large enough that is why we calculate square root.

#### 4. R-Squared

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

We use R-square to find the performance of the model. Which means that it shows that how the input variable is helping the model to predict the final output. In other words, It explains how input variables explain fluctuations in output / predictor variables. The coefficient of determination (R-square) is calculated by following steps:

1. Find residual sum of squares (SSres) of the regression model.
2. Find the error sum of squares (SStot) of the mean model.
3. Divide SSres by SStot.
4. Subtracting it from 1.

A coefficient of determination of 0.85 indicates that the model is performing 85% good with the help of input variable(x). Means higher the coefficient of determination will give better model. However, this metric has limitations and is resolved by an **adjusted coefficient of determination**.

#### 2.6 Metrics for performance (Classification)

The following metrics are used for analyzing the performance of the models that are calculated using confusion matrix (as shown in Fig. 9) generated on the dataset:

		Actual Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Fig.9: Confusion Matrix

##### 1. Accuracy

Accuracy is the ratio of correct outcomes to the total outcomes of the experiment. Accuracy is calculated using the following formula:

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN}$$

Accuracy represents the correctness of predictions made by the model.

## 2. AUC (Area Under ROC Curve)

AUC represents the worthiness of model predictions (Fig. 10). It is the degree of how superior a model is capable to discern between positive and negative occurrences.

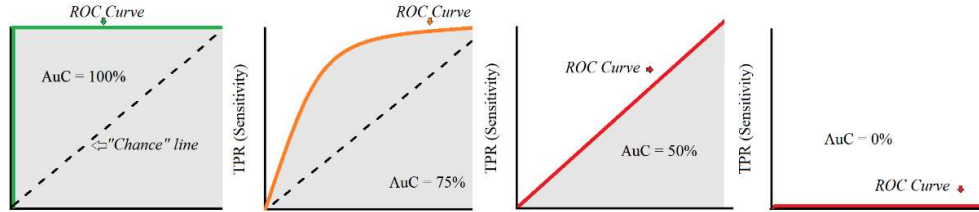


Fig.105: AUC and ROC

1.0 value in AUC means model prediction is 100% accurate and 0.5 means model prediction is worthless for unknown instances prediction.

## 3. Precision

Precision is the ratio of true positive classified records to the total positive outcomes for a class. Precision is calculated using the following formula:

$$Precision = \frac{TP}{TP+FP}$$

Precision signifies how many positive outcomes are actually correct for a class.

## 4. Recall

Recall is also known as sensitivity which is defined as the ratio of true positive classified records to the total number of real positive instances. The recall is calculated using the following formula:

$$Recall = \frac{TP}{TP+F}$$

Recall signifies the positive predictions that are classified incorrectly.

The following figure (Fig. 11) represents the confusion matrix along with the metrics calculations:

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN)	<b>Recall or Sensitivity</b> $\left( \frac{TP}{TP + FN} \right)$
	Negative	False Positive (FP)	True Negative (TN)	
		<b>Precision</b> $\left( \frac{TP}{TP + FP} \right)$		<b>Accuracy</b> $\left( \frac{TP + TN}{TP + FP + TN + FN} \right)$

*Fig.11: Confusion Matrix and different Metrics*

## **CHAPTER 3**

### **PROPOSED MODEL FOR DISTANCE PREDICTION**

In this Chapter, all the working is given for preparing the model. Here we can understand about the data which can be useful for make a query and also to select the suitable features required to build the model for getting good outcomes.

#### **3.1 Datasets:**

In this Project, we are going to look at the NYC Taxi Data which is available publically on kaggle.

The Data table is given below:

lpep_pickup_datetime	lpep_dropoff_datetime	Pickup_longitude	Pickup_latitude	Dropoff_longitude	Dropoff_latitude	Passenger_count	Trip_distance	Trip_type
01-09-15 0:02	01-09-15 0:02	-73.979485	40.684956	-73.979431	40.685020	1	0.00	2.0
01-09-15 0:04	01-09-15 0:04	-74.010796	40.912216	-74.010780	40.912212	1	0.00	2.0
01-09-15 0:01	01-09-15 0:04	-73.921410	40.766708	-73.914413	40.764687	1	0.59	1.0
01-09-15 0:02	01-09-15 0:06	-73.921387	40.766678	-73.931427	40.771584	1	0.74	1.0
01-09-15 0:00	01-09-15 0:04	-73.955482	40.714046	-73.944412	40.714729	1	0.61	1.0

Table.1: Original Datasets

Features/Columns	Description
lpep_pickup_datetime	Start Timing of trip from origin
lpep_dropoff_datetime	End Timing of trip at destination
Pickup_longitude	Start position longitude
Pickup_latitude	Start position latitude
Dropoff_longitude	End position longitude
Dropoff_latitude	End position latitude
Passenger_count	Number of passenger in car
Trip_distance	Distance covered by vehicle (in miles)
Trip_type	Type of trip (1 or 2)

Table.2: Dataset Dictionary

### 3.2 Data Manipulation:

Now we can manipulate the Dataset as per the requirements. In this project, we need to change/update some features as given below:

- Calculate the Time duration of trip by using Start and End timing of trip.  
 $\text{Time\_duration(in minutes)} = (\text{end trip time} - \text{start trip time})$
- Calculate arc distance with the use of longitude and latitude using **haversine formula**.
- Calculate the Direction from origin to destination.
- Convert Date into Days.
- Divide the Hour into 4 part: 12am-6am, 6am-12pm, 12pm-6pm, 6pm-12am.
- Divide the Trip into 2 type: Short Trips(Distance $\leq$ 1mile), Long Trips(Rest)
- Remove outliers from the data.

	Passenger_count	Trip_distance	arc_distance	Duration	day_of_week	Direction	Time_interval	Trip
0	1	0.00	0.003893	0.0	Tuesday	West	MidNight-Morning	Short
1	1	0.00	0.001057	0.0	Tuesday	South	MidNight-Morning	Short
2	1	0.59	0.485087	3.0	Tuesday	South	MidNight-Morning	Short
3	1	0.74	0.700255	4.0	Tuesday	North	MidNight-Morning	Short
4	1	0.61	0.765232	4.0	Tuesday	South	MidNight-Morning	Short

Table.3: Manipulated Dataset

So After Pre-processing, the table is changed to given table.

### 3.3 EDA(Exploratory Data Analysis):

Here we will analyse the dataset in form of graph or values. We can use here so many Graphs as shown below:

- **Box Plots:**

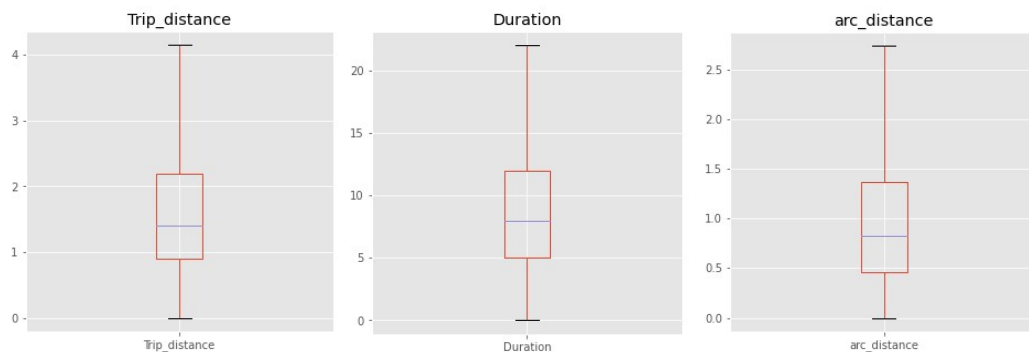


Fig.12: Box Plot for All numerical features



- **Bar Graph: (Univariate Analysis)**

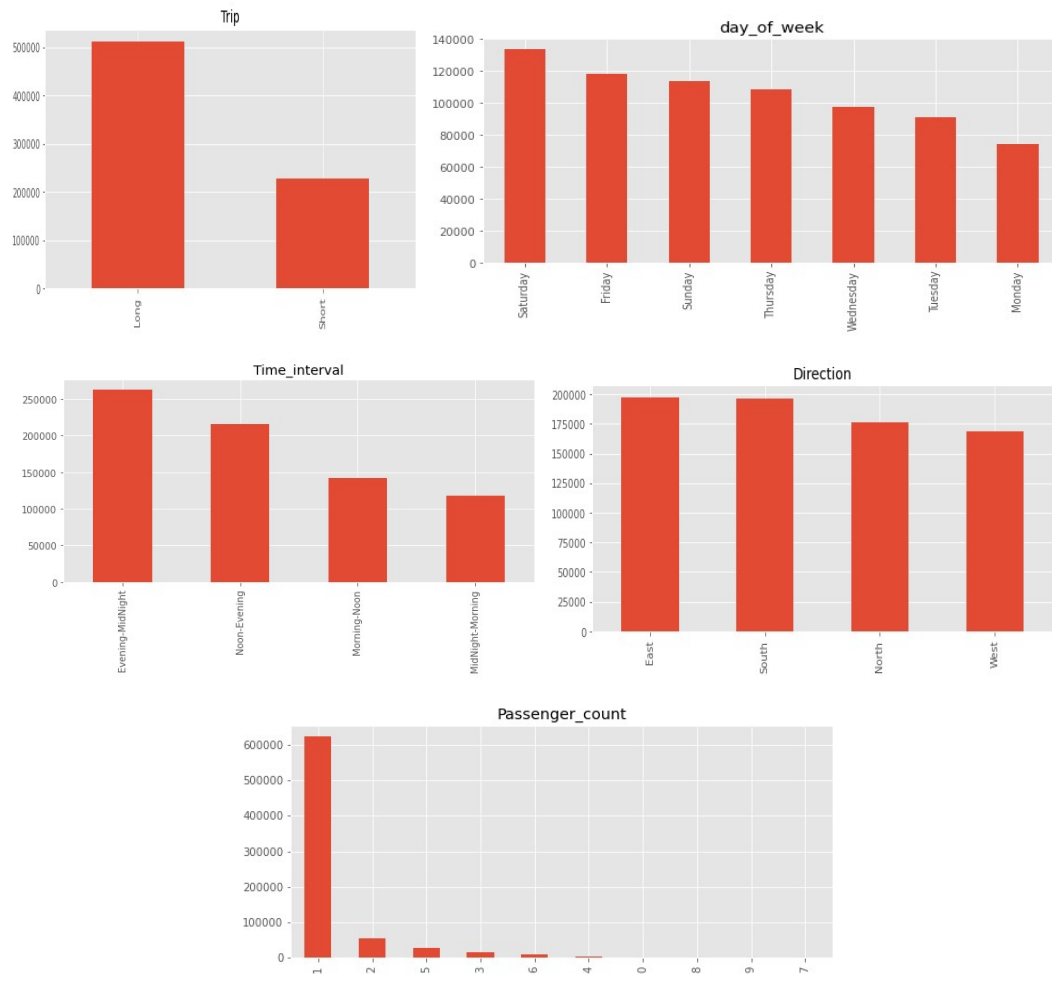
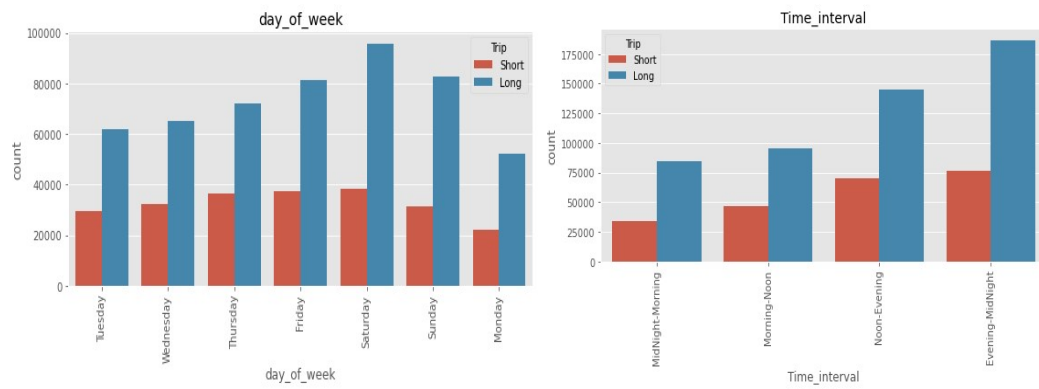


Fig.13: Univariate Analysis

- **Bar Graph: (Bivariate Analysis)**



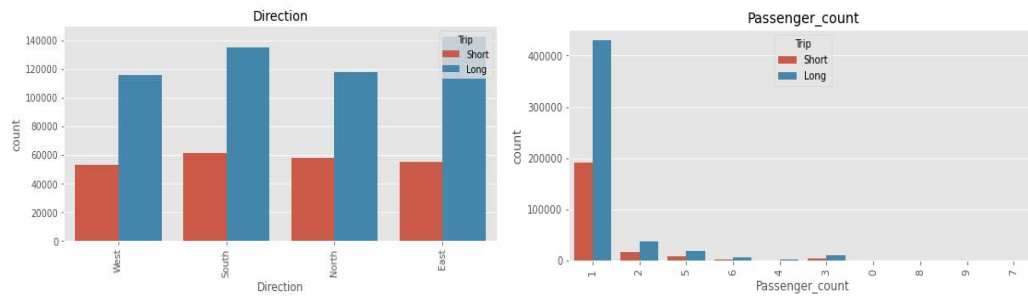


Fig.14: Bivariate Analysis

- Histogram:**

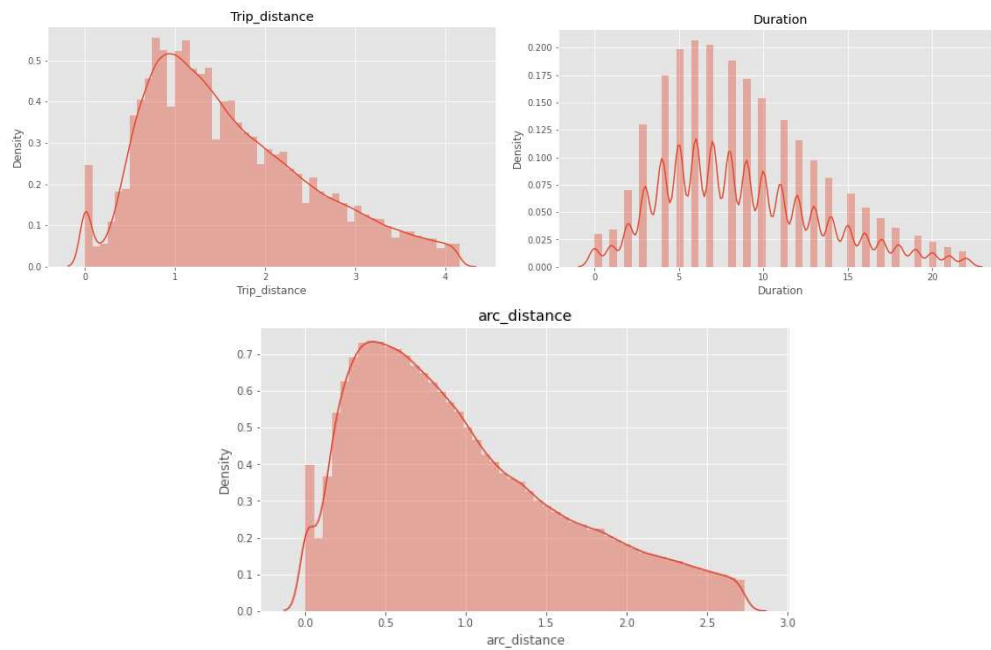
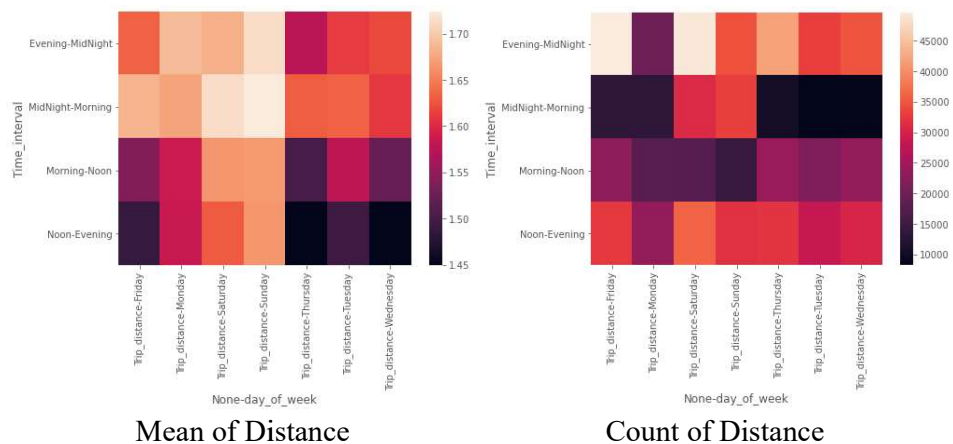


Fig.15:Histogram

- Heat Map Graph:**



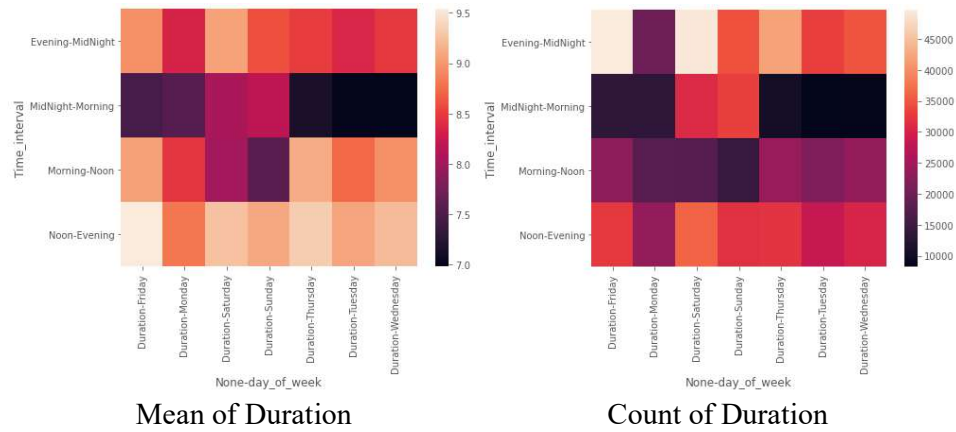


Fig.16:HeatMap

- Correlation Matix:**

	Trip_distance	arc_distance	Duration
Trip_distance	1.000000	0.797942	0.091232
arc_distance	0.797942	1.000000	0.069468
Duration	0.091232	0.069468	1.000000

Table.4:Correlation Matrix

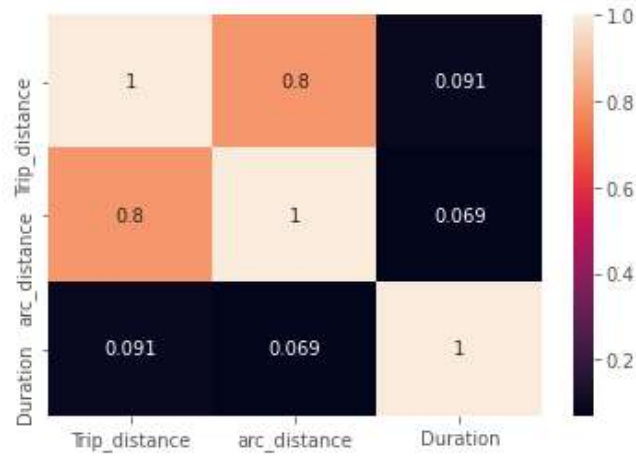


Fig.17:Correlation Heat Graph

### 3.4 Data Pre-processing:

- Feature Selection/Rejection:(using Multicollinearity)**

	variables	VIF
0	arc_distance	4.378515
1	Duration	4.378515

Table.5:VIF Table (Multicollinearity)

- **One-Hot Encoding:**

Passenger_count	arc_distance	Duration	Short	Weekend	MidNight-Morning	Morning-Noon	Noon-Evening	North	South	West
1	2.341173	7	0	1	1	0	0	0	1	0
1	0.962292	10	0	0	0	0	1	0	1	0
1	0.334970	8	0	1	1	0	0	0	0	0
1	1.036021	18	0	0	0	0	1	0	0	1
1	2.306022	11	0	1	0	0	1	1	0	0

Table.6: One-Hot Encoding

### 3.5 Train Model:

- **Train Test Division:**

Train Dataset: 70%

Test Dataset: 30%

- **Train Model For Regression:**

Here, we use regression to predict Distance so we need to make Distance as a Dependent Variable and rest variable will be Independent Variable.

Next step is going to use Algorithm on Train Datasets and validate with Test Datasets. and generate the Results.

- **Train Model For Classification:**

Here, we use Classification to predict Type of Trip so we need to make Trip as a Dependent Variable and rest variable will be Independent Variable.

Next step is going to use Algorithm on Train Datasets and validate with Test Datasets. and generate the Results.

## CHAPTER 4

### RESULTS

In this Chapter, we will see the results of the algorithms having Top Performing Results. Then we will compare and analyse the results for both Regression & Classification.

#### 4.1 Results For Regression:

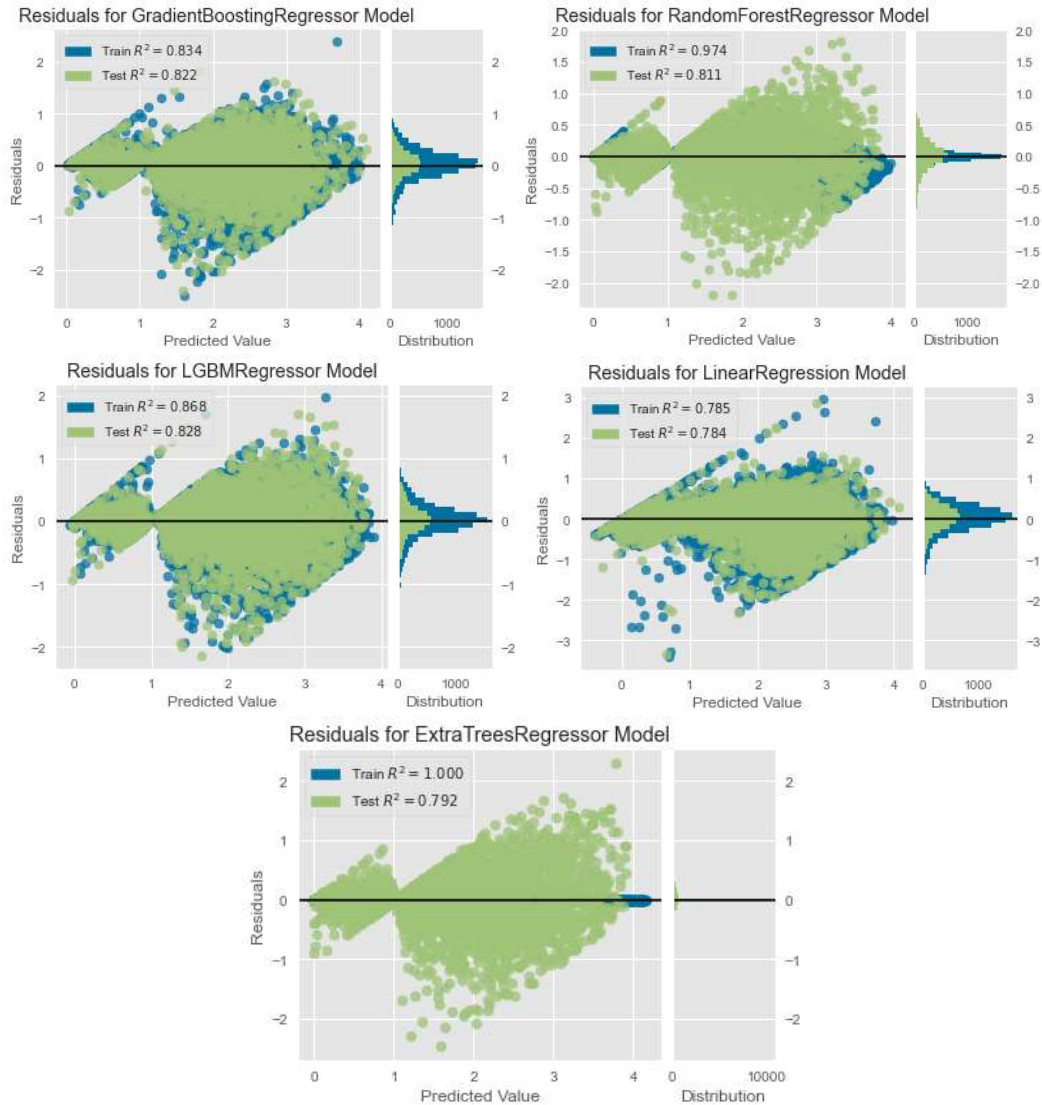


Fig.18: Result Graph (Actual V/S Predicted)

- **Comparison of Results for all Regression Methods:**

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
lightgbm	Light Gradient Boosting Machine	0.2562	0.1404	0.3745	0.8311	0.1245	0.1902
gbr	Gradient Boosting Regressor	0.2671	0.1472	0.3834	0.8229	0.1298	0.2155
rf	Random Forest Regressor	0.2693	0.1544	0.3929	0.8139	0.1296	0.1934
et	Extra Trees Regressor	0.2855	0.1760	0.4195	0.7878	0.1387	0.2074
br	Bayesian Ridge	0.3022	0.1786	0.4224	0.7851	0.1534	0.2759
ridge	Ridge Regression	0.3024	0.1789	0.4228	0.7847	0.1535	0.2764
lr	Linear Regression	0.3025	0.1791	0.4231	0.7844	0.1535	0.2766
lar	Least Angle Regression	0.3025	0.1791	0.4230	0.7844	0.1535	0.2765
huber	Huber Regressor	0.2938	0.1835	0.4282	0.7793	0.1498	0.2510
knn	K Neighbors Regressor	0.3243	0.2221	0.4712	0.7323	0.1661	0.2617
ada	AdaBoost Regressor	0.3930	0.2531	0.5029	0.6948	0.1766	0.3392
omp	Orthogonal Matching Pursuit	0.3565	0.2546	0.5044	0.6935	0.1795	0.3029
dt	Decision Tree Regressor	0.3567	0.2788	0.5276	0.6639	0.1723	0.2490
par	Passive Aggressive Regressor	0.3872	0.2984	0.5358	0.6387	0.1823	0.3108
en	Elastic Net	0.4223	0.3228	0.5681	0.6113	0.2130	0.4739
lasso	Lasso Regression	0.4540	0.3538	0.5947	0.5742	0.2281	0.5564
llar	Lasso Least Angle Regression	0.7418	0.8326	0.9122	-0.0015	0.3585	1.0779

Table.7:Comparison Between all the Regression Algorithm

## 4.2 Results For Classification:

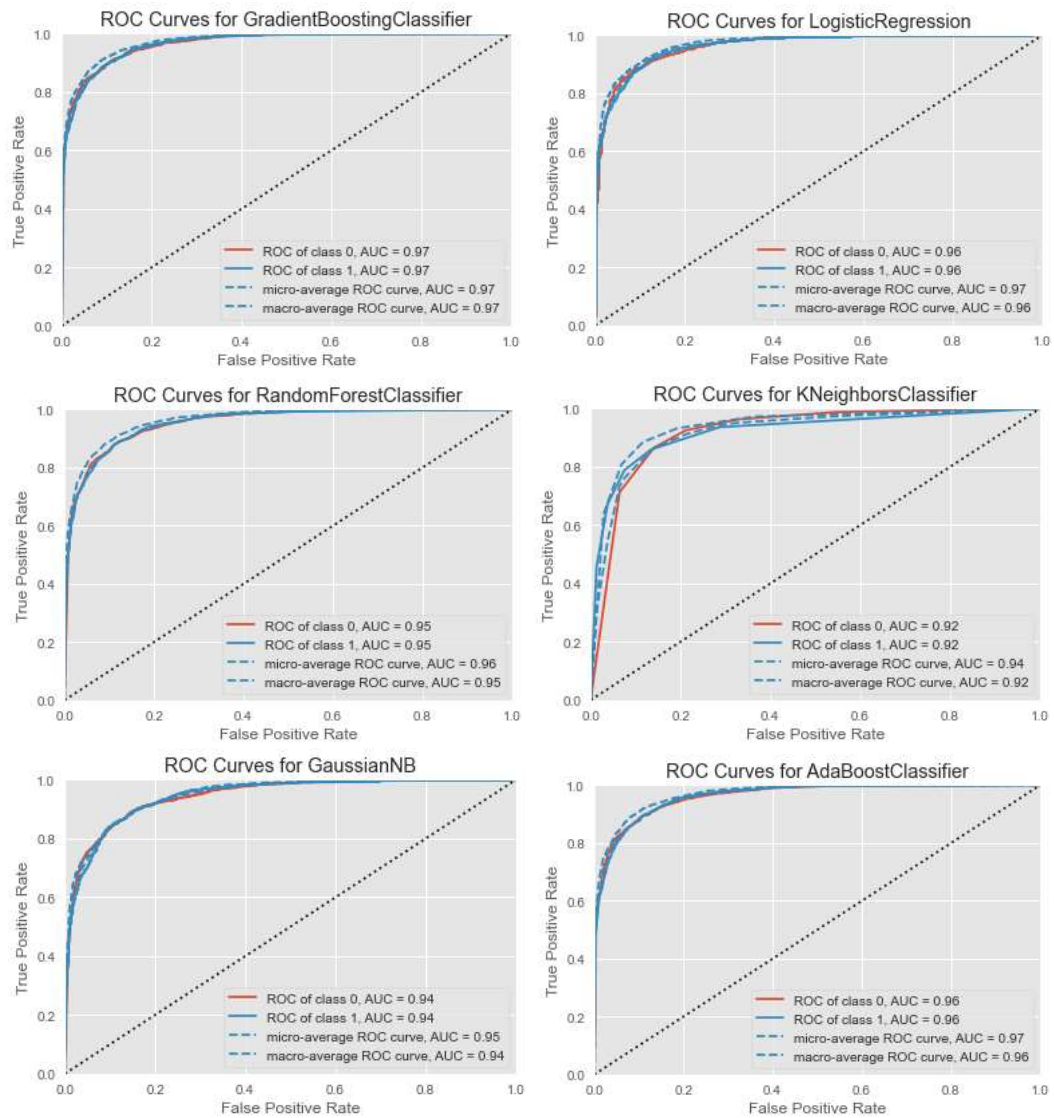


Fig.19: ROC curve Graph



- **Comparison of Results for all Classification Methods:**

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
<b>gbc</b>	Gradient Boosting Classifier	0.9054	0.9627	0.8250	0.8571	0.8405	0.7734	0.7739
<b>lr</b>	Logistic Regression	0.9041	0.9601	0.8372	0.8445	0.8406	0.7720	0.7722
<b>ada</b>	Ada Boost Classifier	0.9030	0.9607	0.8170	0.8560	0.8358	0.7670	0.7677
<b>ridge</b>	Ridge Classifier	0.9026	0.0000	0.8311	0.8445	0.8375	0.7680	0.7683
<b>lightgbm</b>	Light Gradient Boosting Machine	0.9024	0.9611	0.8301	0.8448	0.8372	0.7675	0.7678
<b>lda</b>	Linear Discriminant Analysis	0.9017	0.9578	0.8522	0.8280	0.8397	0.7689	0.7693
<b>svm</b>	SVM - Linear Kernel	0.8932	0.0000	0.8337	0.8269	0.8255	0.7490	0.7536
<b>rf</b>	Random Forest Classifier	0.8877	0.9498	0.8007	0.8232	0.8116	0.7317	0.7320
<b>et</b>	Extra Trees Classifier	0.8768	0.9317	0.7841	0.8038	0.7937	0.7059	0.7061
<b>knn</b>	K Neighbors Classifier	0.8739	0.9140	0.7671	0.8061	0.7859	0.6967	0.6973
<b>nb</b>	Naive Bayes	0.8654	0.9435	0.8797	0.7307	0.7981	0.6985	0.7054
<b>dt</b>	Decision Tree Classifier	0.8626	0.8369	0.7719	0.7736	0.7725	0.6741	0.6743
<b>qda</b>	Quadratic Discriminant Analysis	0.8119	0.8871	0.7035	0.6929	0.6908	0.5568	0.5630

Table.8:Comparison Between all the Classification Algorithm



## **CHAPTER 5**

### **CONCLUSION AND FUTURE SCOPE**

**The following conclusion has been made:**

- Existing pre-trained Machine Learning models have been studied and compared in this study.
- It has been found that Gradient Boosting Method performed best in both Regression & Classification among other algorithm.
- The Accuracy is not more than 90% which means we need more rich features in the model.
- In Regression, R-Square Value is good but Mean Percentage Error is also 20% because the data is Right Skewed.
- Along with this, the performance of different Machine Learning models has been studied.

**The future scope of this project is as follows:**

- In the Model, We can add more features like Weather Condition, Traffic Report, fuel used, etc.
- Adding more rich features like above can results in a good performance using Regression as well as Classification.

## **REFERENCES**

- **Machine Learning Concepts:**
  1. <https://www.geeksforgeeks.org/ml-introduction-data-machine-learning/>
- **Datasets:**
  1. [https://s3.amazonaws.com/nyc-tlc/trip+data/green\\_tripdata\\_2015-09.csv](https://s3.amazonaws.com/nyc-tlc/trip+data/green_tripdata_2015-09.csv)
  2. <https://www.kaggle.com/amit07/nyc-greendata-taxi>
- **Calculations Reference:**
  1. <https://www.sisense.com/blog/latitude-longitude-distance-calculation-explained/>
- **Implementation for Algorithms:**
  1. <https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/>
  2. <https://www.dataquest.io/blog/top-10-machine-learning-algorithms-for-beginners/>
- **AutoML Concepts:**
  1. <https://pycaret.org/regression/>
  2. <https://pycaret.org/classification/>
- **Performance Measure in Classification and Regression:**
  1. <https://iq.opengenus.org/performance-metrics-in-classification-regression/>
- **Others:**
  1. <https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/mean-squared-error/>
  2. <https://www.geeksforgeeks.org/ways-to-apply-an-if-condition-in-pandas-dataframe/>
  3. [https://www.tutorialspoint.com/machine\\_learning\\_with\\_python/index.htm](https://www.tutorialspoint.com/machine_learning_with_python/index.htm)