Project

CAB FARE PREDICTION

Submitted By:

Vikas Suresh Dudhe

INDEX

ABSTRACT

Predictive analytics uses repository knowledge to predict the longer-term events. Typically, past knowledge is employed to make a mathematical model that captures necessary trends. That predictive model is then used on current knowledge to predict the longer term or to counsel actions to require for best outcomes. Predictive analytics has received a great deal of attention in recent years because of advances in supporting technology, significantly within the areas of huge knowledge and machine learning. firms additionally use Predictive analytics to make a lot of correct forecasts, like prediction the fare quantity for a cab ride within the town. These forecasts change resource coming up with for example, programming of assorted cab rentals to be done a lot of effectively. For a cab rental start-up company, the fare quantity relies on a great deal of things. This analysis aims to know all patterns and to use analytics for fare prediction. The projected work is to style a system that predicts the fare quantity for a cab ride within the town. The aim is to make regression models, which can predict the continual fare quantity for every cab ride and facilitate prediction counting on multiple time-based, point and general factors.

INTRODUCTION

Machine learning (ML) is closely associated with technique statistics, that focuses on creating predictions victimization computers. process (DM) may even be a field of study at intervals cc and focuses on alpha data analysis through Unsupervised learning. In its application across business issues, machine learning is additionally spoken as prognostic analytics. Machine learning tasks unit of measurement classified into many broad classes.

In supervised learning, the rule builds a mathematical model from a collection of information that contains each the inputs and in addition the desired outputs. Classification algorithms and regression algorithms unit of measurement samples of supervised learning Regression algorithms are named for his or her continuous outputs, that suggests they go to own any value at intervals a range.

In Unsupervised learning, the rule builds a mathematical model from a collection of information that contains solely inputs and no desired output labels.

Unsupervised learning algorithms unit of measurement accustomed notice structure within the information, like grouping or clump of information points. Unsupervised learning will discover patterns within the information and will cluster the inputs into classes, as in feature learning. property reduction is that the strategy of reducing the quantity of "features", or inputs, in Associate in Nursing exceedingly} terribly set of information.

Machine learning and process usually use constant ways in which and overlap considerably, however whereas cc focuses on prediction, supported acknowledged properties learned from the employment data, process focuses on the invention of (previously) unknown properties within the information. this might be the analysis step of data discovery in databases (KDD). DM uses several cc ways in which, however with absolutely whole totally different goals; on the opposite hand, cc additionally employs process ways in which as "unsupervised learning" or as a preprocessing step to strengthen learner accuracy.

A. Supervised Learning Approach

Supervised learning algorithms embrace classification and regression. Classification algorithms area unit used once the outputs area unit restricted to a restricted set of values, and regression algorithms area unit used once the outputs could have any numerical worth at intervals a spread.

To build any model, the primary step is to acknowledge the matter statement and select the suitable class that matches in. Since the discourse statement “fare quantity for

a cab ride within the city" fits into foretelling, Regression is employed because it helps in predicting continuous fare quantity for the longer term. Regression may be a supervised Learning approach because the target variable, that is fare-amount and is understood beforehand. Regression is employed because it helps in predicting continuous fare quantity for the longer term.

B. Data

The aim is to create regression models that may predict the continual fare quantity for every of the cab-rides looking on multiple time-based, point and generic factors. This drawback statement falls underneath the class of prediction that deals with predicting continuous values for the longer term (the continuous value is that the fare quantity of the cab ride). Figure 1 shows a sample of the information set that may be want to predict the fare quantity of a cab

	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
0	4.5	2009-06-15 17:26:21 UTC	-73.844311	40.721319	-73.841610	40.712278	1.0
1	16.9	2010-01-05 16:52:16 UTC	-74.016048	40.711303	-73.979268	40.782004	1.0
2	5.7	2011-08-18 00:35:00 UTC	-73.982738	40.761270	-73.991242	40.750562	2.0
3	7.7	2012-04-21 04:30:42 UTC	-73.987130	40.733143	-73.991567	40.758092	1.0
4	5.3	2010-03-09 07:51:00 UTC	-73.968095	40.768008	-73.956655	40.783762	1.0

Figure 1: Train Data

C. Number of attributes:

- fare_amount - float value indicating the fare amount for the ride.
- pickup_datetime - timestamp value indicating when the cab ride started.
- pickup_longitude - float for longitude coordinate of where the cab ride started.
- pickup_latitude - float for latitude coordinate of where the cab ride started.
- dropoff_longitude - float for longitude coordinate of where the cab ride ended.
- dropoff_latitude - float for latitude coordinate of where the cab ride ended.
- passenger_count - an integer indicating the number of passengers in the cab ride.

Independent Variables
pickup_datetime
pickup_longitude
pickup_latitude
dropoff_longitude
dropoff_latitude
passenger_count

Dependent Variables
Fare_amount

From the given train data, it is understood that, we have to predict fare amount, and other variables will help me achieve that, here pickup_latitude/longitude, dropoff_latitude/longitude this data are signifying the location of pick up and drop off. It is explaining starting point and end point of the ride. So, these variables are crucial for us.

Passenger_count is another variable, that explains about how many people or passenger boarded the ride, between the pickup and drop off locations. And pick up date time gives information about the time the passenger is picked up and ride has started. But unlike pick up and drop off locations has start and end details both in given data. The time data has only start details and no time value or time related information of end of ride. So, during pre-processing of data we will drop this variable. As it seems the information of time is incomplete.

Also, there is a separate test data given, in the format of CSV file containing 9914 observations and 6 variables. All of them are the independent variables. An in these data at the end we have to predict the fare or the target variable. Following is a sample of the test data provided.

	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
0	2015-01-27 13:08:24 UTC	-73.973320	40.763805	-73.981430	40.743835	1
1	2015-01-27 13:08:24 UTC	-73.986862	40.719383	-73.998886	40.739201	1
2	2011-10-08 11:53:44 UTC	-73.982524	40.751260	-73.979654	40.746139	1
3	2012-12-01 21:12:12 UTC	-73.981160	40.767807	-73.990448	40.751635	1
4	2012-12-01 21:12:12 UTC	-73.966046	40.789775	-73.988565	40.744427	1

Figure 2: Test Data

PROBLEM STATEMENT

You are a cab rental start-up company. You have successfully run the pilot project and now want to launch your cab service across the country. You have collected the historical data from your pilot project and now have a requirement to apply analytics for fare prediction. You need to design a system that predicts the fare amount for a cab ride in the city.

index	Column	Non-Null Count	Dtype
0	fare_amount	16043 non-null	object
1	pickup_datetime	16067 non-null	object
2	pickup_longitude	16067 non-null	float64
3	pickup_latitude	16067 non-null	float64
4	dropoff_longitude	16067 non-null	float64
5	dropoff_latitude	16067 non-null	float64
6	passenger_count	16012 non-null	float64

METHODOLOGY

A. Exploratory Data Analysis

After characteristic the approach, consecutive step is pre-processing the info. viewing knowledge refers to exploring the info, purification the info in addition as visualizing the info through graphs and plots. this is often referred to as Exploratory Data Analysis (EDA).

1) Exploring passenger count features

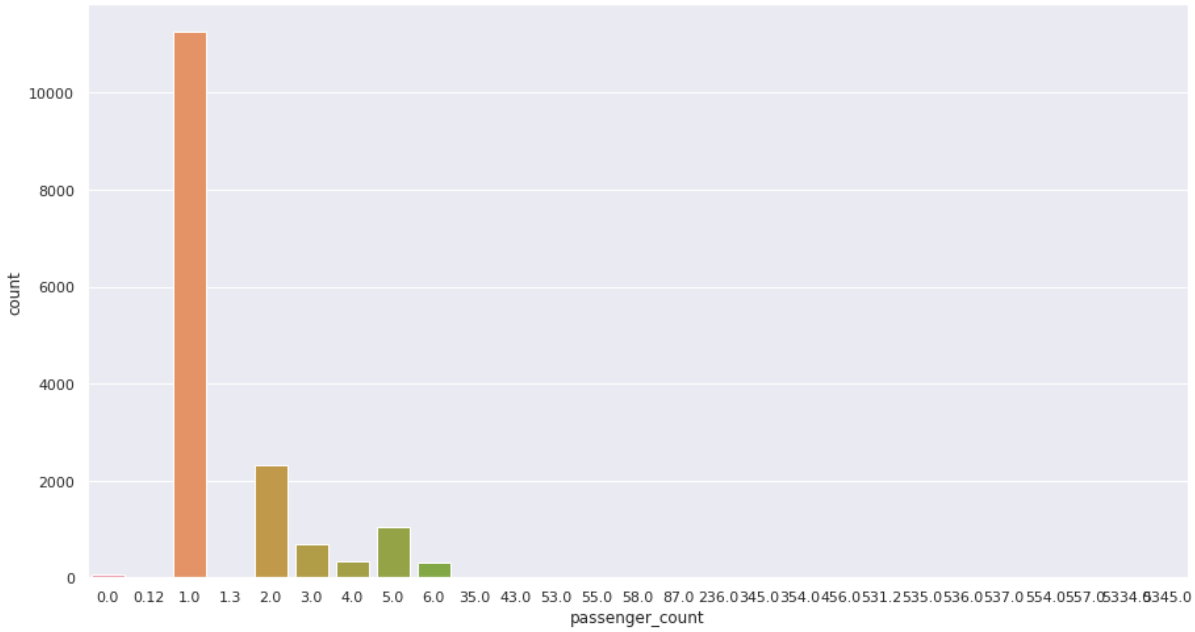

Figure 3: Passenger Count Plot

Above graph shows the training dataset contain maximum number of passenger is having 1 and max passengers we can accommodate in Cab service is 6. Hence we have to take care and further analysis and certain assumption and will remove in outlier analysis.

- passenger count having zero doesn't make any sense to data
- passenger count above 6 need to be treated as max capacity

To initiate this method, any of the likelihood distributions of the variables square measure thought-about. Most analysis like regression, need the info to be usually distributed. this will be envisioned at a look by viewing the likelihood distributions or likelihood density functions of the variable.

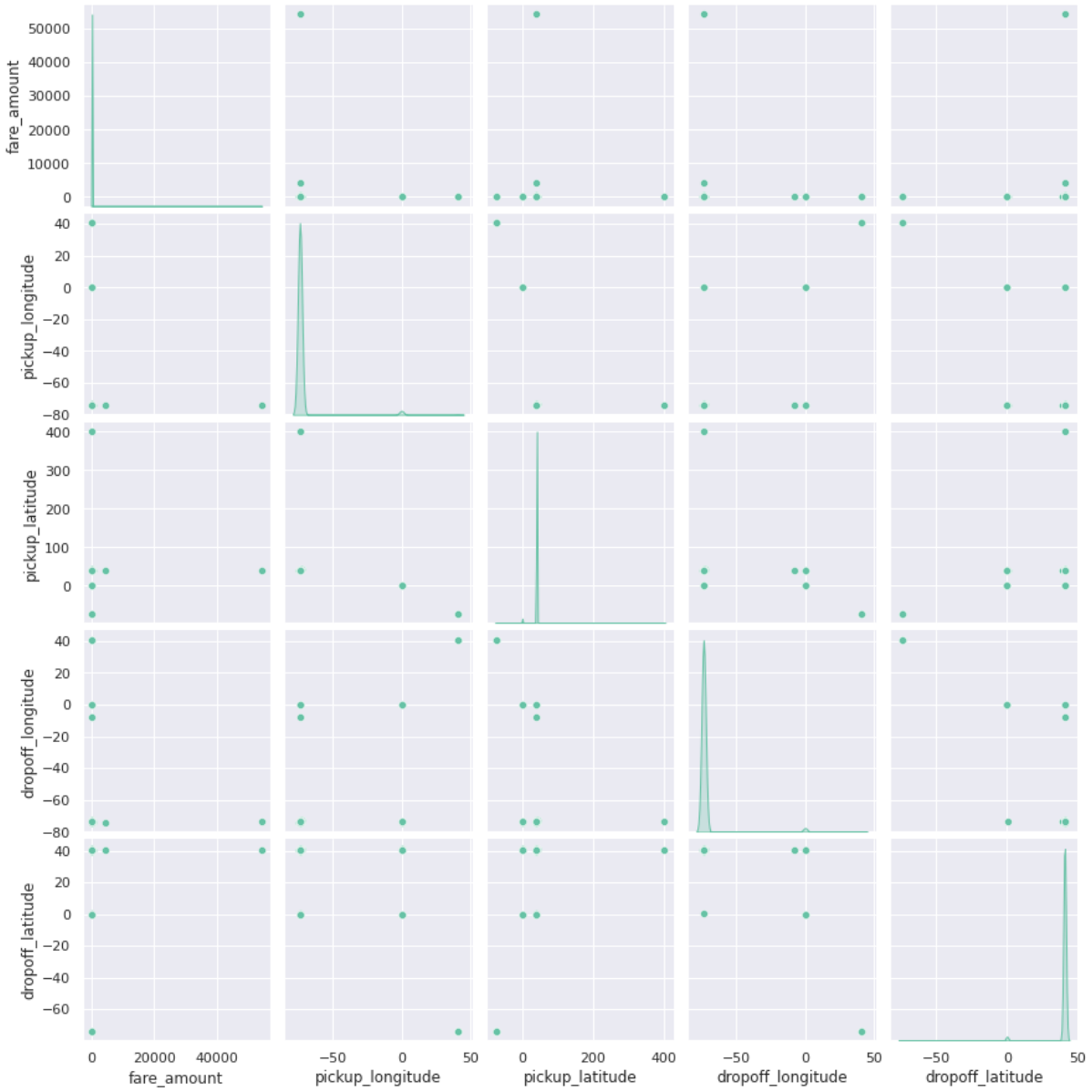

Figure 4: Pair plot on Train data

To plot multiple pairwise bivariate distributions in a dataset, we use the pairplot() function. This shows the relationship for (n, 2) combination of variable in a DataFrame as a matrix of plots and the diagonal plots are the univariate plots.

2. Exploring pickup and drop-off features

Location features need to take care because cab fare will be decided on this factor before analysing its values first, we will check the valid ranges for latitudes and longitudes. The valid range for latitudes is -90 to +90 and for Longitude are -180 to +180.

When we applied above validation we get the following results

```
pickup_Longitude        above +180  is 0
pickup_Longitude        below -180  is 0
pickup_Latitude         above +90   is 1
pickup_Latitude         below -90   is 0
dropoff_Longitude       above +180  is 0
dropoff_Longitude       below -180  is 0
dropoff_Latitude        above +90   is 0
dropoff_Latitude        below -90   is 0
```

From the above observation we can see that only 1 row having inappropriate range so will treat in outlier analysis.

One more interesting factor we need to address here is if location features are having zero values that doesn't make any sense of travel.

3. Exploring fare_amount features

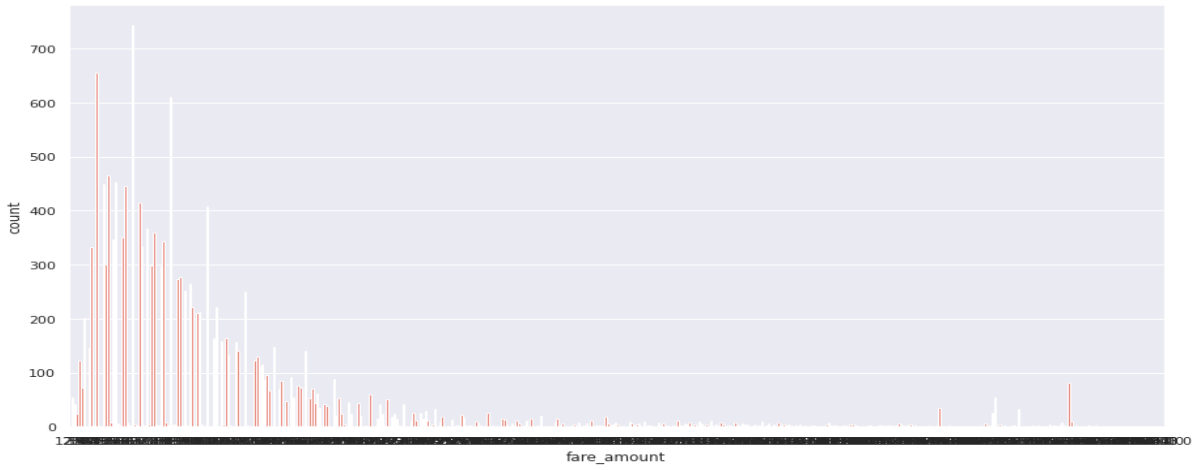

Figure 5: count plot for fare_amout

This graph shows the variation of fare amount ranges from 0 to 54343 Rs which is practically for some cases are invalid and out of range, need to take care in outlier analysis.

B. Missing Value Analysis

Missing value is availability of incomplete observations in the dataset. This is found because of reasons like, incomplete submission, wrong input, manual error etc. These Missing values affect the accuracy of model. So, it becomes important to check missing values in our given data.

Once proper data conversion is done next step is to analyse the missing values. As per data, the missing value percentages of the variables are as in Figure 6. Note: pickup_datetime column has 1 NA value which already removed in preprocessing.

	Variables	Missing_%
0	passenger_count	0.351213
1	fare_amount	0.140485
2	pickup_datetime	0.000000
3	pickup_longitude	0.000000
4	pickup_latitude	0.000000
5	dropoff_longitude	0.000000
6	dropoff_latitude	0.000000

Figure 6: Missing Percentage of features

As solely passenger_count and fare_amount have missing values and therefore the share is a smaller amount than thirty percent, that the missing values area unit imputed. On random assignment NA to at least one of those values for passenger_count and so filling the content using the 3 methods: mean, median and K-NN, it's found that the median provides the nearest estimate to the present actual worth. once filling in missing values data looks like as shown in Figure.7.

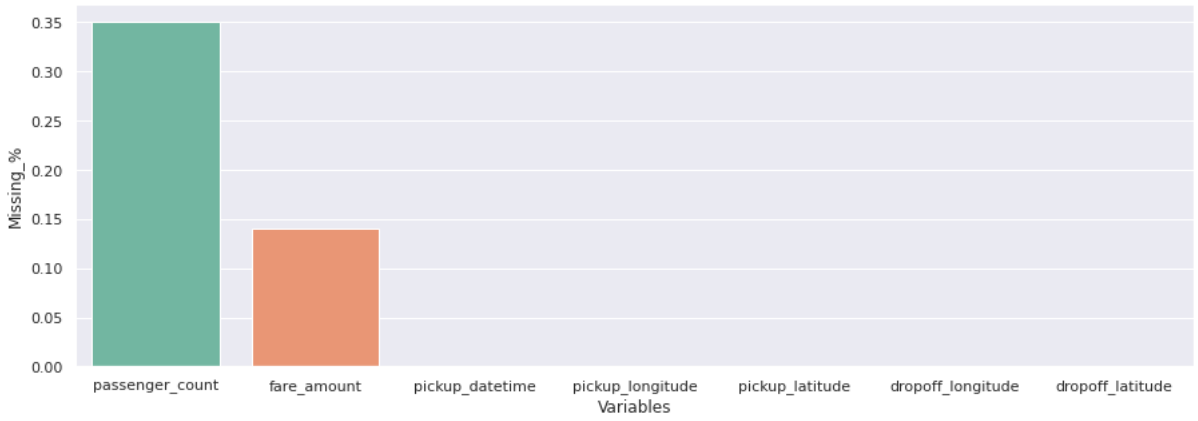

Figure 7: Visualization of missing values

1) Missing Value Analysis in Given Data:

In the given dataset it is found that there are lot of values which are missing. It is found in the following types:

- **Blank spaces:** Which are converted to NA and NaN in R and Python respectively for further operations
- **Zero Values:** This is also converted to NA and Nan in R and python respectively prior further operations
- **Repeating Values:** There are lots of repeating values in pickup_longitude, pickup_latitude, dropoff_longitude and dropoff_latitude. This will hamper our model, so such data is also removed to improve the performance.

Following the standards of percentage of missing values, we now have to decide to accept a variable or drop it for further operations. Industry standards ask to follow following standards:

1. Missing value percentage < 30% : Accept the variable
2. Missing value percentage > 30 % : Drop the variable

It is found from the above graph plot that the there is no variable exceeding the 30% range so we not need to exclude any of our variable.

2) Impute the missing value:

After the identification of the missing values the next step is to impute the missing values. And this imputation is normally done by following methods.

1. Central Tendencies: by the help of Mean, Median or Mode
2. Distance based or Data mining method like KNN imputation
3. Prediction Based: It is based on Predictive Machine Learning Algorithm

To use the best method, it is necessary for us to check, which method predicts values close to the original data. And this done by taking a subset of data, taking an example variable and noting down its original value and the replacing that value with NA and then applying available methods. And noting down every value from the above methods for the example variable we have taken, now we chose the method which gives most close value.

In this project, KNN imputation worked the best. So, I am using KNN method to impute missing Values.

C. Outlier Analysis

Outlier is an abnormal observation that stands or deviates away from other observations. These happens because of manual error, poor quality of data and it is correct but exceptional data. But, It can cause an error in predicting the target variables. So we have to check for outliers in our data set and also remove or replace the outliers wherever required.

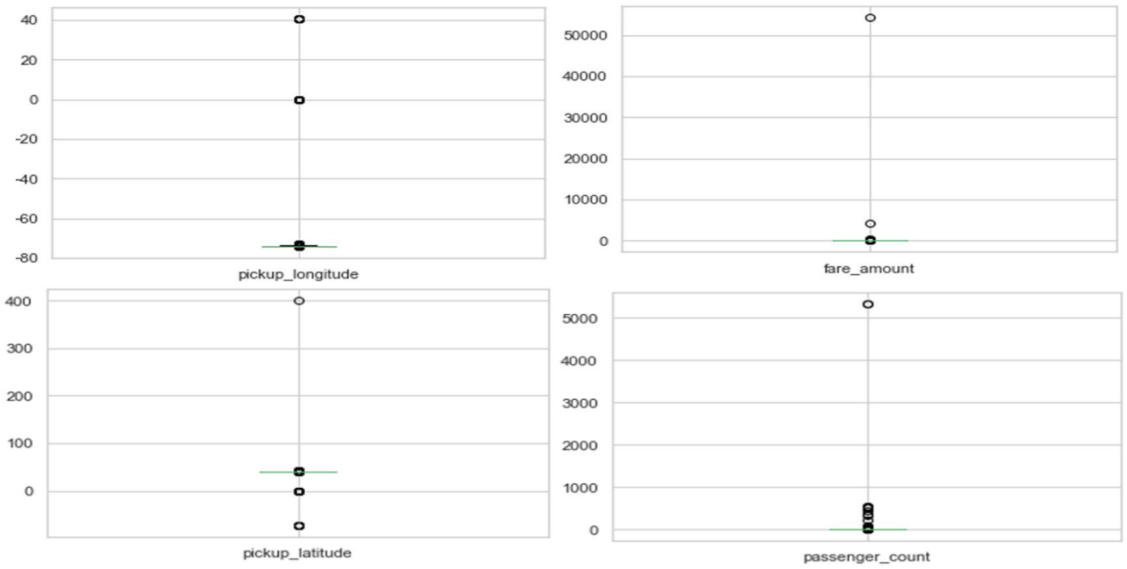

Figure 8: Noisy Data

As depicted in Fig.6, there are a lot of noisy data so it's important to clean the data for better model performance. We visualize the outliers using boxplots. Figure 8 shows the boxplots of four of the six predictor variables (as a sample) and the target variable. A lot of useful inferences can be made from these plots such as a lot of outliers and extreme values are seen in each of the data sets.

Let's dive in for details of outlier found in each data feature

1) Fare_Amount :

I have always seen fare of a cab ride as positive, I have never seen any cab driver, giving me money to take a ride in his cab. But in this dataset, there are many instances where fare amount is negative. and having zero Given below are such instances:

	fare_amount	pickup_datetime	...	dropoff_latitude	passenger_count
2039	-2.90	2010-03-09 23:37:10	...	40.641952	1.0
2486	-2.50	2015-03-22 05:14:27	...	40.720539	1.0
2780	0.01	2015-05-01 15:38:41	...	40.713997	1.0
10002	0.00	2010-02-15 14:26:01	...	40.713960	1.0
13032	-3.00	2013-08-30 08:57:10	...	40.741357	4.0

[5 rows x 7 columns]

Figure 9: Fare Outliers

2) Passenger_count:

I have always found a cab with 4 seats to maximum of 6 seats if we consider sedan and Hatchback. But in this dataset, I have found passenger count more than this, and in some cases a large number of values. This seems irregular data, or a manual error. Thus, these are outliers and needs to be removed. Few instances are following.

	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
233	8.5	2011-07-24 01:14:35	0.000000	0.000000	0.000000	0.000000	236.0
263	4.9	2010-07-12 09:44:33	-73.983249	40.734655	-73.991278	40.738918	456.0
293	6.1	2011-01-18 23:48:00	-74.006642	40.738927	-74.010828	40.717907	5334.0
356	8.5	2013-06-18 10:27:05	-73.992108	40.764203	-73.973000	40.762695	535.0
386	8.1	2009-08-21 19:35:05	-73.960853	40.761557	-73.976335	40.748361	354.0
413	NaN	2013-09-12 11:32:00	-73.982060	40.772705	-73.956213	40.771777	55.0

Figure 10: Passenger count Outlier

3) Location points:

When I checked the data it is found that most of the longitude points are within the 70 degree and most of the latitude points are within the 40 degree. This symbolizes all the data belongs to a specific location and a specific range. But I also found some data which consists location points too far from the average location point's range of 70 Degree Longitude and 40 Degree latitude. It seems these far point locations are irregular data. And I consider this as outlier. I have collected the maximum and minimum values of location point as a reference to identify the outliers.

Column Name	Min Range	Max Range
pickup_Longitude	-74.438233	40.766125
pickup_Latitude	-74.006893	401.083332
dropoff_Longitude	-74.42933199999999	40.802437
dropoff_Latitude	-74.006377	41.366138

Figure 11: Location Range Outlier

From above table we can see only 1 row having out of range data after removing the passenger count outliers. so as per valid range of longitude i.e. -180 to +180 and for latitude -90 to +90 we will keep this ranges and rest will be dropped.

One more outlier is location having zero values in pickup and dropoff location which is also an outlier hence we have to remove those rows also. following are the summary for having location zero.

```
pickup_longitude      equal to 0 is 311
pickup_latitude       equal to 0 is 311
dropoff_longitude     equal to 0 is 312
dropoff_latitude      equal to 0 is 310
```

All these outliers mentioned above happened because of manual error, or interchange of data, or may be correct data but exceptional. But all these outliers can hamper our data model. So there is a requirement to eliminate or replace such outliers. And impute with proper methods to get better accuracy of the model. In this project, I used KNN method to impute the outliers in passenger count, location Points and fare amount.

```
fare_amount          0
pickup_longitude     0
pickup_latitude      0
dropoff_longitude    0
dropoff_latitude     0
passenger_count      0
dtype: int64
```

Figure 12: Imputation Treatment for missing values

Upon removing the outliers and missing values, data is now refined as shown below,

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude
count	15658.000000	15658.000000	15658.000000	15658.000000	15658.000000
mean	11.364818	-73.911499	40.689704	-73.906299	40.687644
std	10.780425	2.659305	2.613556	2.711095	2.632652
min	1.140000	-74.438233	-74.006893	-74.429332	-74.006377
25%	6.000000	-73.992390	40.736547	-73.991369	40.736294
50%	8.500000	-73.982050	40.753312	-73.980552	40.754238
75%	12.500000	-73.968078	40.767805	-73.965362	40.768314
max	453.000000	40.766125	41.366138	40.802437	41.366138

Figure13: Cleaned Train data set

D. Feature Selection

Because all the variables are numeric the important features are extracted using the correlation matrix. As is seen from Fig.14, all the variables are important for predicting the fare_amount since none of the variables have a high correlation factor (considering the threshold as 0.9), so all the variables for model building are kept.

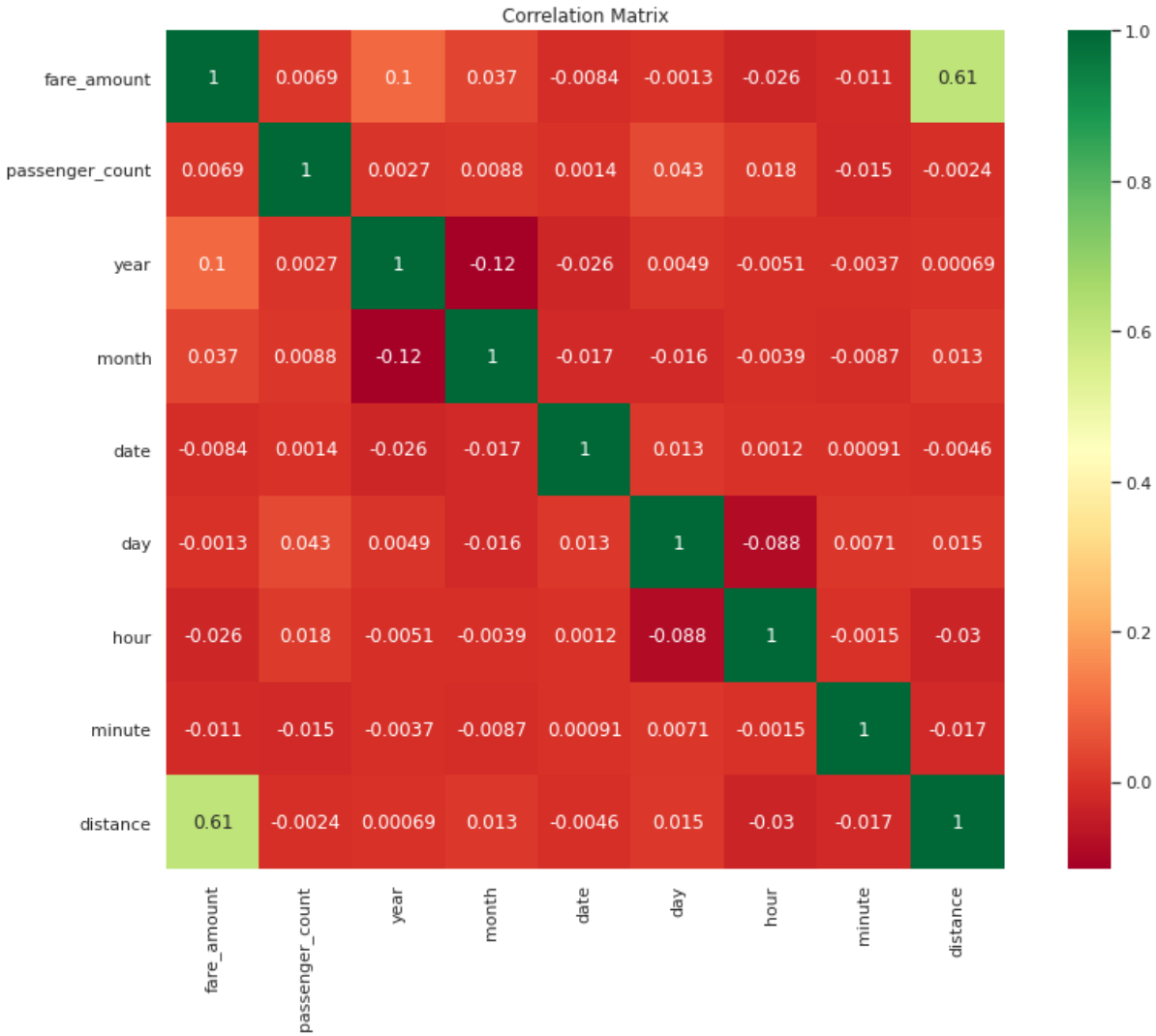

Figure 14 Correlation Matrix

As is seen, distance has the highest prediction power for fare_amount whereas passenger_count and day have the least prediction power.

E. Feature Engineering

Sometimes, all the variables in our data may not be accurate enough to predict the target variable, in such cases we need to analyze our data, understand our data and select the dataset variables that can be most useful for our model. In such cases we follow feature selection. Feature selection helps by reducing time for computation of model and also reduces the complexity of the model.

After understanding the data, preprocessing and selecting specific features, there is a process to engineer new variables if required to improve the accuracy of the model.

In this project the data contains only the pick-up and drop points in longitude and latitude. Also as the longitude, latitude points are there, the distance traveled per ride is easily calculated to derive a relationship between the fare amount and the distance Thus, we have to create a new variable prior further processing the data. And in this project the variable I have created is Distance variable (distance), which is a numeric value and explains the distance covered between the pick-up and drop of points.

After researching I found a formula called The haversine formula, that determines the distance between two points on a sphere based on their given longitudes and latitudes. This formula calculates the shortest distance between two points in a sphere.

We have to convert pickup_datetime variable from object to datetime and convert them into year, month, day as individual entities to take deep dive into data understanding.

After executing the haversine function in our project, I got new variable distance and some instances of data are mentioned below.

	pickup_datetime	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	year	month	date	day	hour	minute	distance
0	2009-06-15 17:26:21	4.5	-73.844311	40.721319	-73.841610	40.712278	1.0	2009	6	15	0	17	26	1.030764
1	2010-01-05 16:52:16	16.9	-74.016048	40.711303	-73.979268	40.782004	1.0	2010	1	5	1	16	52	8.450134
2	2011-08-18 00:35:00	5.7	-73.982738	40.761270	-73.991242	40.750562	2.0	2011	8	18	3	0	35	1.389525
3	2012-04-21 04:30:42	7.7	-73.987130	40.733143	-73.991567	40.758092	1.0	2012	4	21	5	4	30	2.799270
4	2010-03-09 07:51:00	5.3	-73.968095	40.768008	-73.956655	40.783762	1.0	2010	3	9	1	7	51	1.999157

Figure 15: Featured Columns

F. Feature Scaling

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

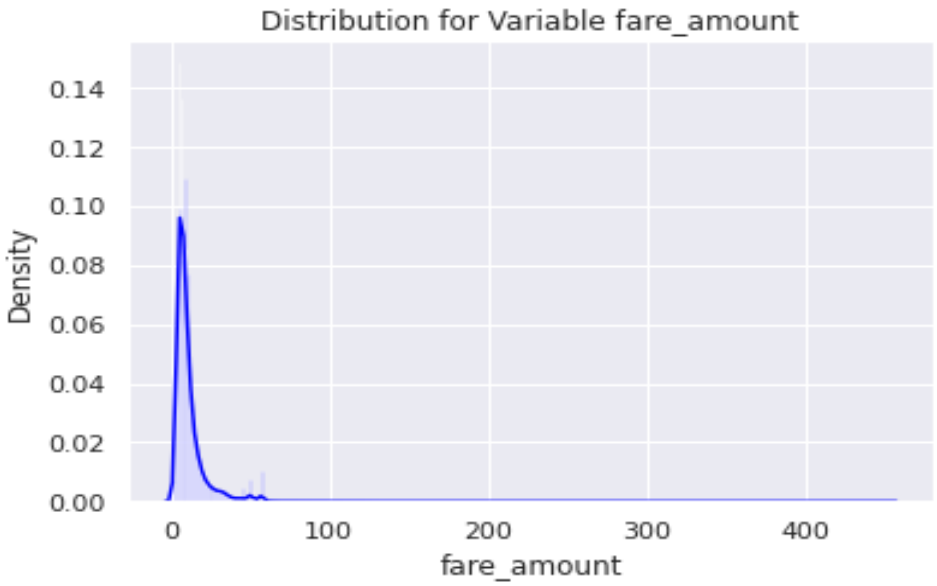

Figure 16: Distribution graph for fare amount before scaling

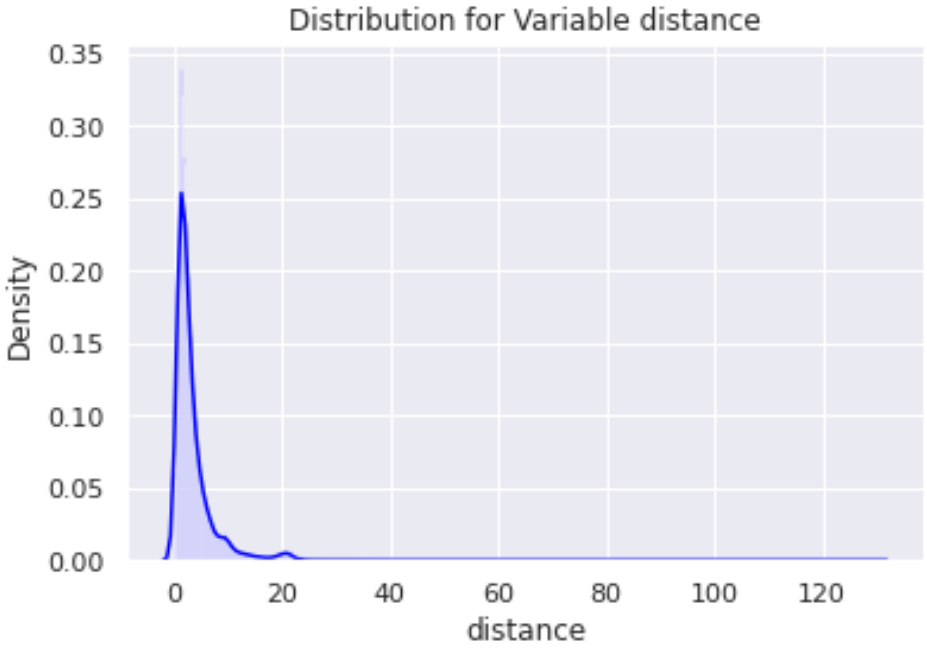

Figure 17: Distribution graph for distance before scaling

For scaling we use Numpy.log1p() is a mathematical function that helps the user to calculate the natural logarithmic value of x+1 where x belongs to all the input array elements. log1p is reverse of exp(x) – 1.

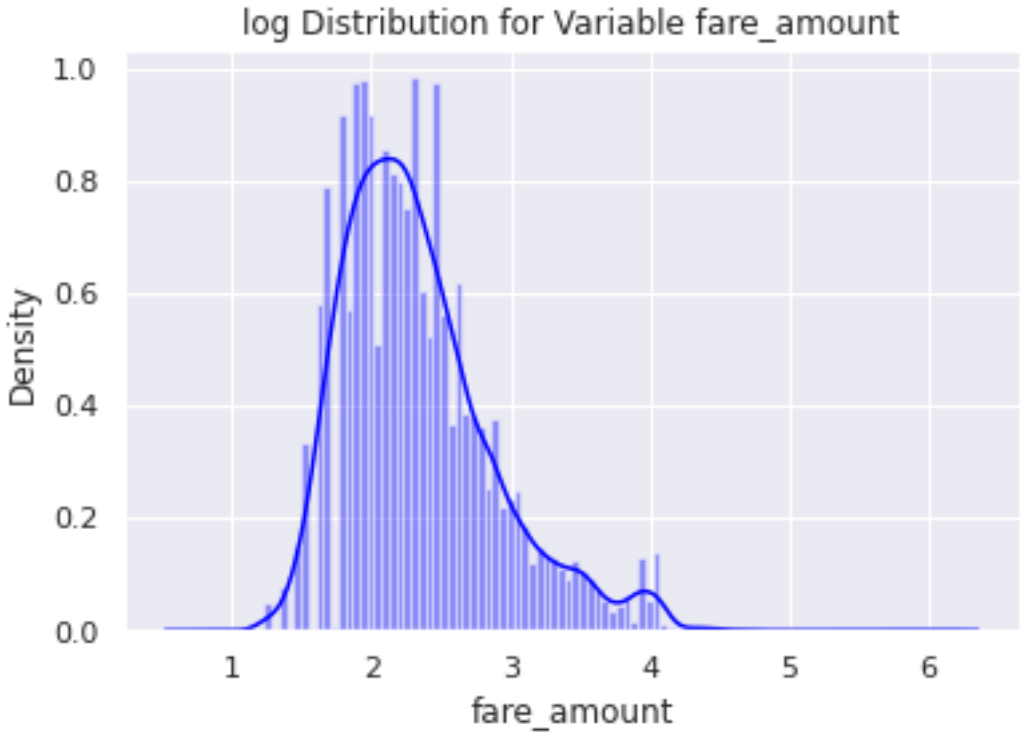

Figure 18: Distribution graph for fare amount after scaling

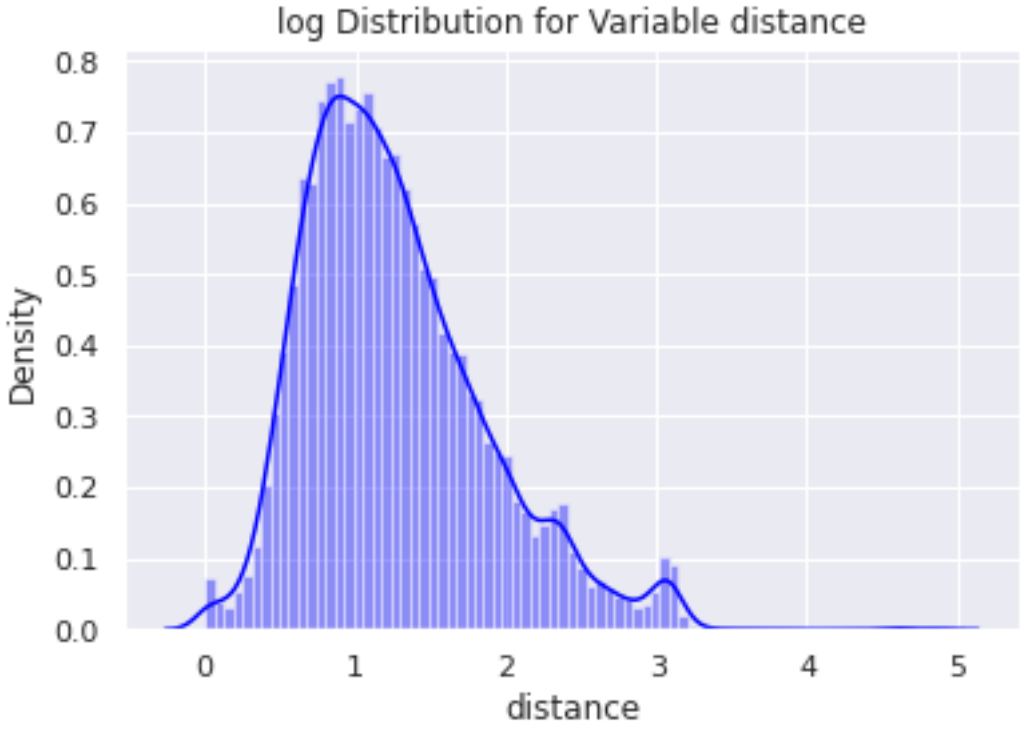

Figure 19: Distribution graph for distance after scaling

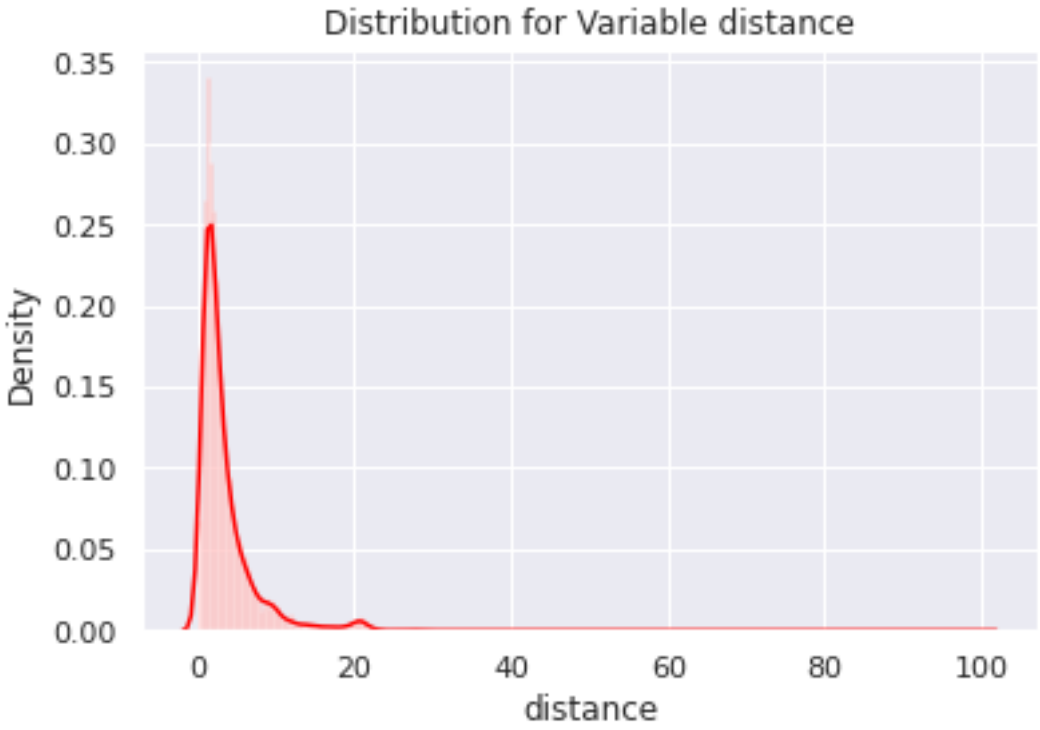

Figure 20: Distribution graph for distance before scaling on test data

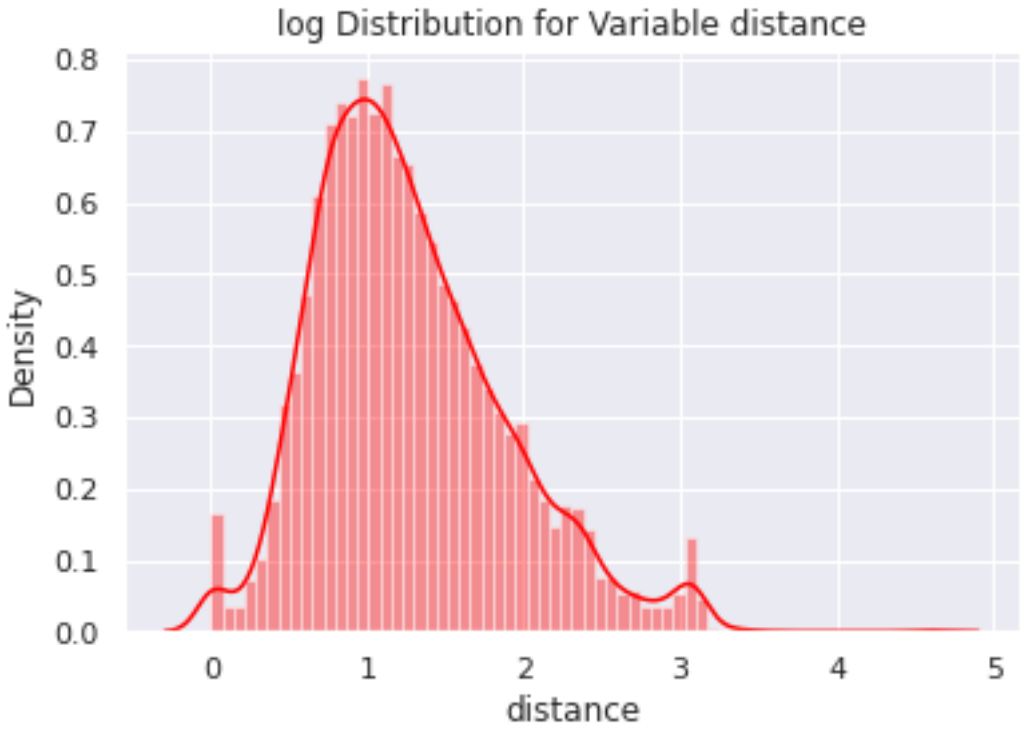

Figure 21: Distribution graph for distance after scaling on test data

We have done feature scaling for both test and train data so that we will get proper model trained.

DATA VISUALIZATION

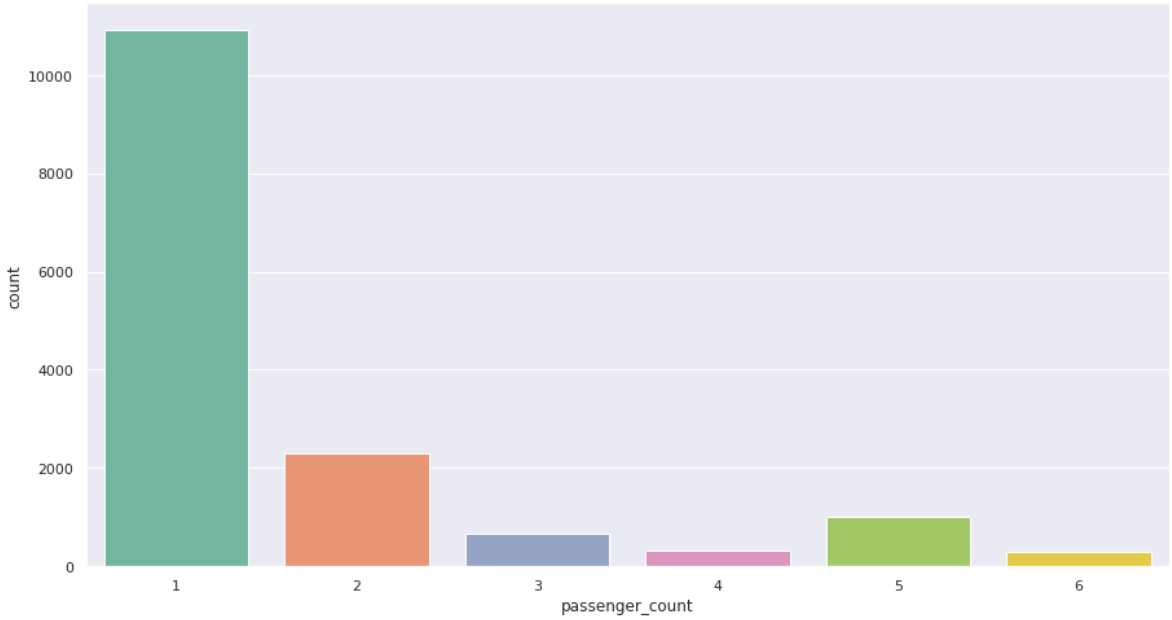

Figure 22: Count Plot of passenger

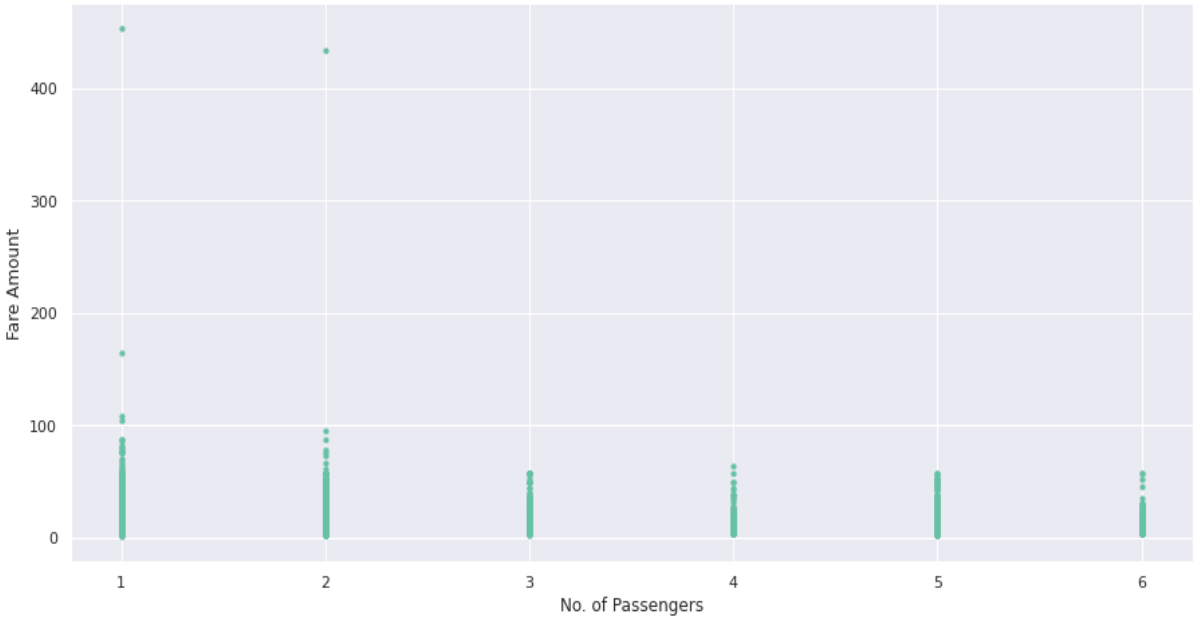

Figure 23: Relationship between Fare_amount and no of passenger

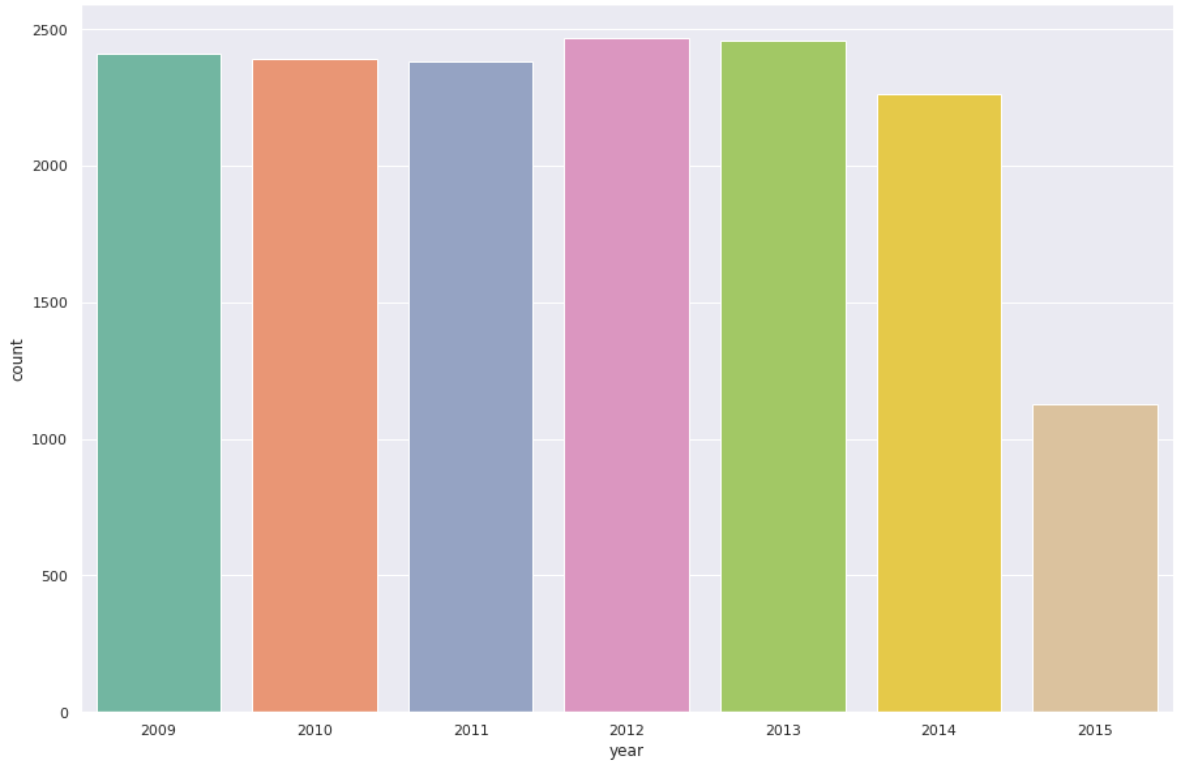

Figure 24: Count plot year wise

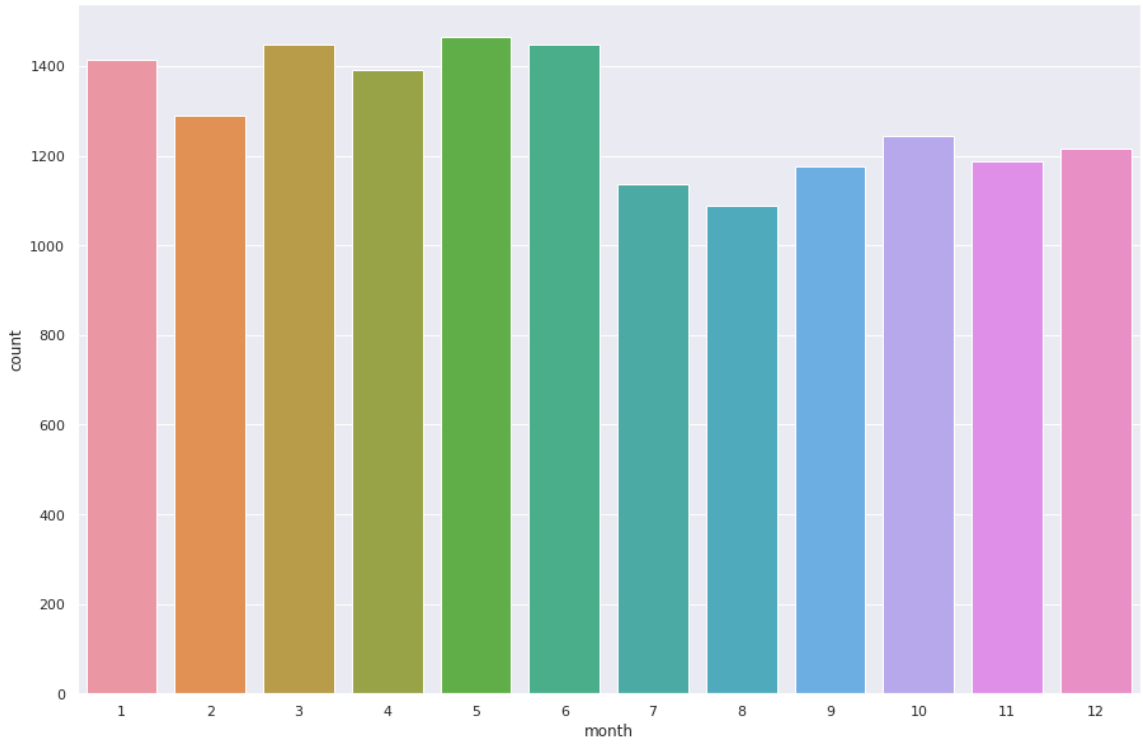

Figure 25: Count plot month wise

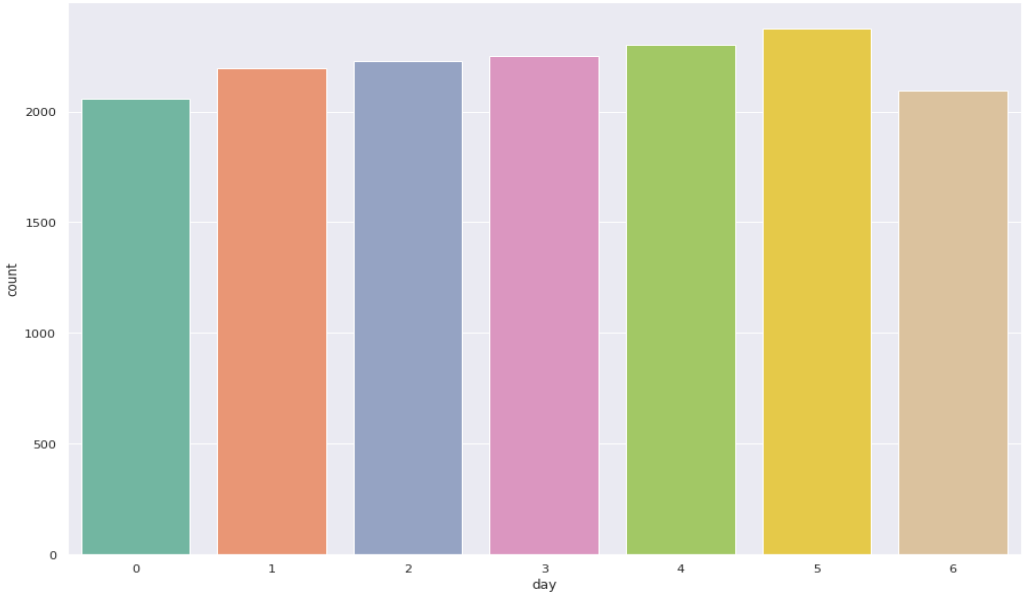

Figure 26: Count plot day wise

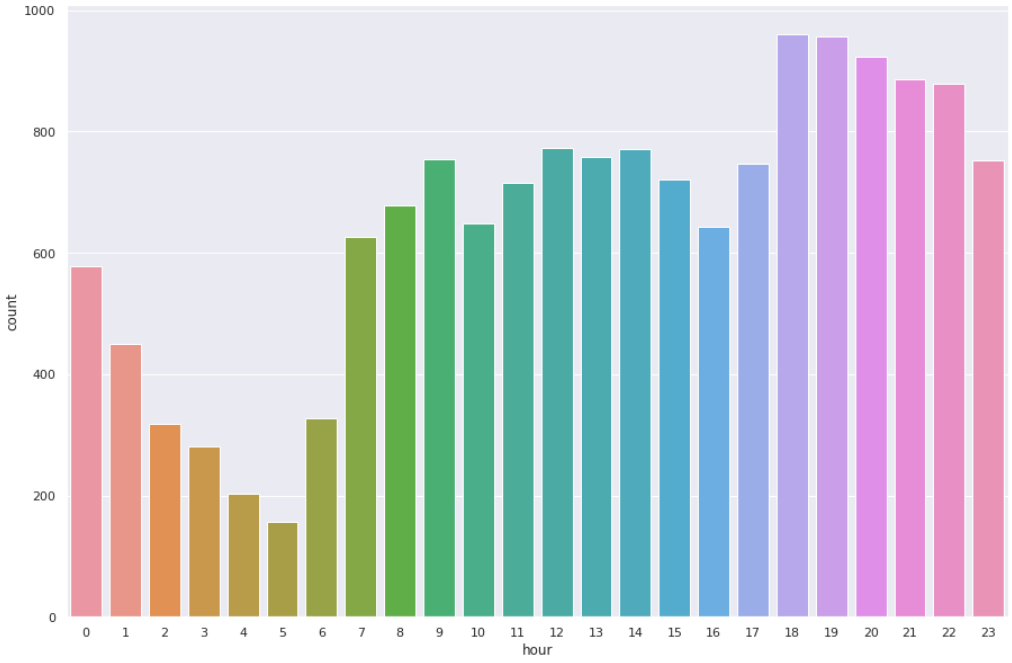

Figure 27: Count plot hour wise

Observations from above graphs

1) utilization of service from year 2014 seems downfall
2) 3rd quarter of service is affected than other quarters
3) weekday utilization of service is more as compared to weekend
4) cab service utilization is more in night time as compared to day time

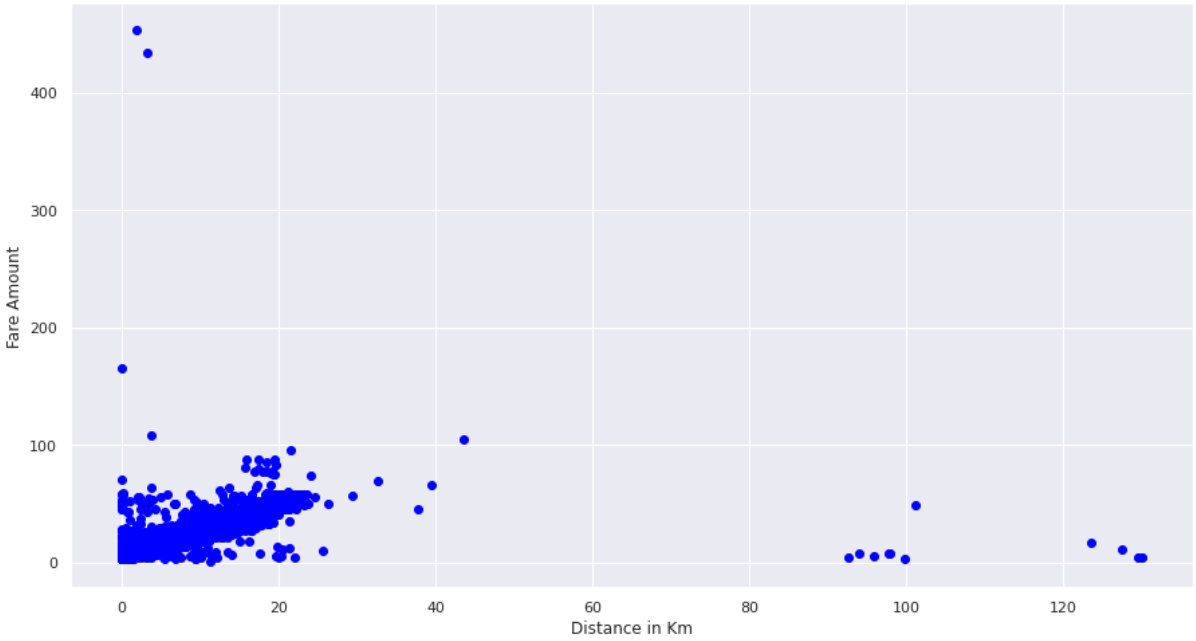

Figure 28: Relationship between fare amount and distance

Above graph shows clear effect of fare and distance based on distance fare amount increase

	fare_amount	passenger_count	year	month	date	day	hour	minute	distance
0	1.704748	1	2009	6	15	0	17	26	0.708412
1	2.884801	1	2010	1	5	1	16	52	2.246029
2	1.902108	2	2011	8	18	3	0	35	0.871095
3	2.163323	1	2012	4	21	5	4	30	1.334809
4	1.840550	1	2010	3	9	1	7	51	1.098331

Figure 29: Final scaled data for model

MODELING

A. Model Selection

In the early stages of research throughout pre-processing, it's understood that fare_amount depends on multiple behaviors. Therefore, it is important to make a model in such the simplest way that it takes all told the specified inputs and fits the model in such the simplest way that it offers the foremost correct result amongst all the opposite models. The variable will fall in any of the four categories: Nominal, Ordinal, Interval, and Ratio. 3 approaches are taken and compared:

After all the above processes the next step is developing the model based on our prepared data. In this project we got our target variable as "fare_amount". The model has to predict a numeric value. Thus, it is identified that this is a Regression problem statement. And to develop a regression model, the various models that can be used are Linear Regression, Random Forest, XGBoost and KNN imputation.

1) Linear Regression

The next method in the process is Linear regression. It is used to predict the value of variable *Y* based on one or more input predictor variables *X*. The goal of this method is to establish a linear relationship between the predictor variables and the response variable. Such that, we can use this formula to estimate the value of the response *Y*, when only the predictors (*X- Values*) are known.

Model Evaluation with default parameters

```
#metric calculation RMSE for test data
##calculating RMSE for test data
RMSE_test_LR = np.sqrt(mean_squared_error(y_test, prediction_test_LR))
print("Root Mean Squared Error For Test data = "+str(RMSE_test_LR))

##calculating RMSE for train data
RMSE_train_LR= np.sqrt(mean_squared_error(y_train, prediction_train_LR))

print("Root Mean Squared Error For Training data = "+str(RMSE_train_LR))
```

```
Root Mean Squared Error For Test data = 0.24855938527489574
Root Mean Squared Error For Training data = 0.2759378227590264
```

```
#calculate R^2 for train data
from sklearn.metrics import r2_score
r2_score(y_train, prediction_train_LR)
```

```
0.7426851654594733
```

```
#calculate R^2 for test data
r2_score(y_test, prediction_test_LR)
```

```
0.7948204715133892
```

Linear Regression Model with tuned parameters

```
# Setup the parameters and distributions to sample from: param_grid
param_grid = {'copy_X':[True, False],
          'fit_intercept':[True,False]}
# Instantiate a Decision reg classifier: Lregg
Lregg = LinearRegression()

# Instantiate the gridSearchCV object: Lregg_cv
Lregg_cv = GridSearchCV(Lregg, param_grid, cv=5,scoring='r2')

# Fit it to the data
Lregg_cv.fit(X, y)

# Print the tuned parameters and score
print("Tuned Decision reg Parameters: {}".format(Lregg_cv.best_params_))
print("Best score is {}".format(Lregg_cv.best_score_))
```

```
Tuned Decision reg Parameters: {'copy_X': True, 'fit_intercept': True}
Best score is 0.755864978762299
```

```
# Create the regressor: reg_all
reg_all = LinearRegression(copy_X= True, fit_intercept=True)

# Fit the regressor to the training data
reg_all.fit(x_train,y_train)

# Predict on the test data: y_pred
y_pred = reg_all.predict(x_test)

# Compute and print R^2 and RMSE
print("R^2: {}".format(reg_all.score(x_test, y_test)))
rmse = np.sqrt(mean_squared_error(y_test,y_pred))
print("Root Mean Squared Error: {}".format(rmse))
```

```
R^2: 0.7948204715133892
Root Mean Squared Error: 0.24855938527489574
```

Summary of Model

Metric	Train Data	Test Data
RMSE	0.2759378227590264	0.24855938527489574
r2_score	0.7426851654594733	0.7948204715133892

2) Random Forest

The next model to be followed in this project is Random Forest. It is a process where the machine follows an ensemble learning method for classification and regression that operates by developing a number of decision trees at training time and giving output as the class that is the mode of the classes of all the individual decision trees.

Important features from random forest algorithm

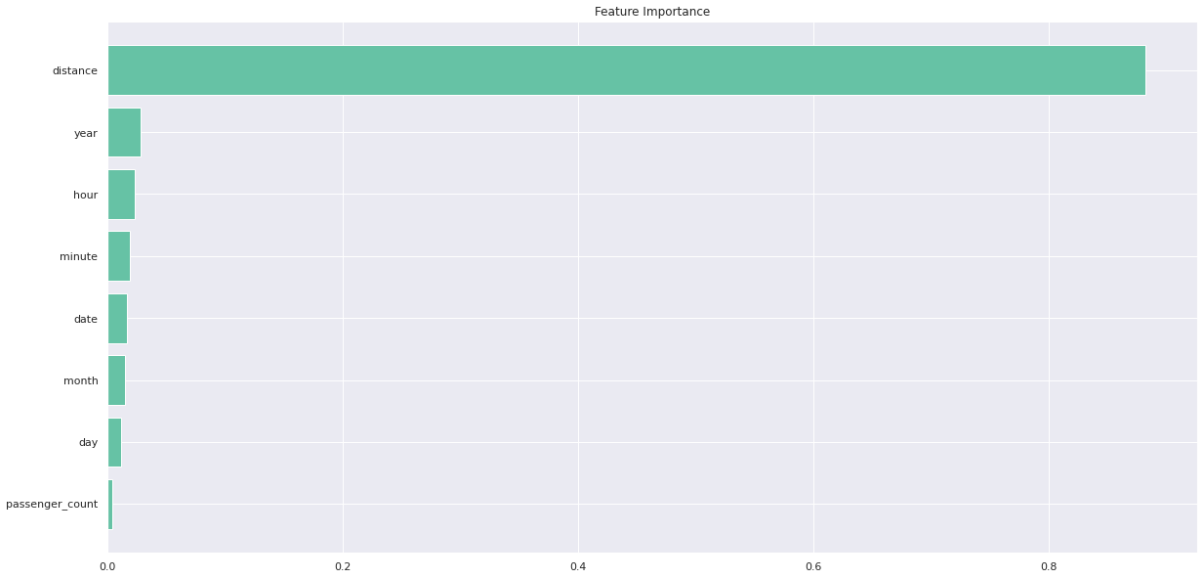

Figure 30: Important Features through Random Forest

Summary of model

```
<<<------------------- Training Data Score ---------------------->>>

r2 square   0.9298203593402904
Adjusted r square:0.9297720262544366
MAPE:3.89264377088736
MSE: 0.020766761829189864
RMSE: 0.14410677232243413

<<<------------------- Test Data Score ---------------------->>>

r2 square   0.8140047426681034
Adjusted r square:0.8136199580653998
MAPE:7.349878071773215
MSE: 0.05600517714349311
RMSE: 0.23665412978330447
```

3) XGBoost

Boosting is a sequential technique which works on the principle of an ensemble. It combines a set of weak learners and delivers improved prediction accuracy. At any instant t, the model outcomes are weighed based on the outcomes of previous instant t-1. The outcomes predicted correctly are given a lower weight and the ones miss-classified are weighted higher. Note that a weak learner is one which is slightly better than random guessing. For example, a decision tree whose predictions are slightly better than 50%. XGBoost is well known to provide better solutions than other machine learning algorithms. In fact, since its inception, it has become the "state-of-the-art" machine learning algorithm to deal with structured data.

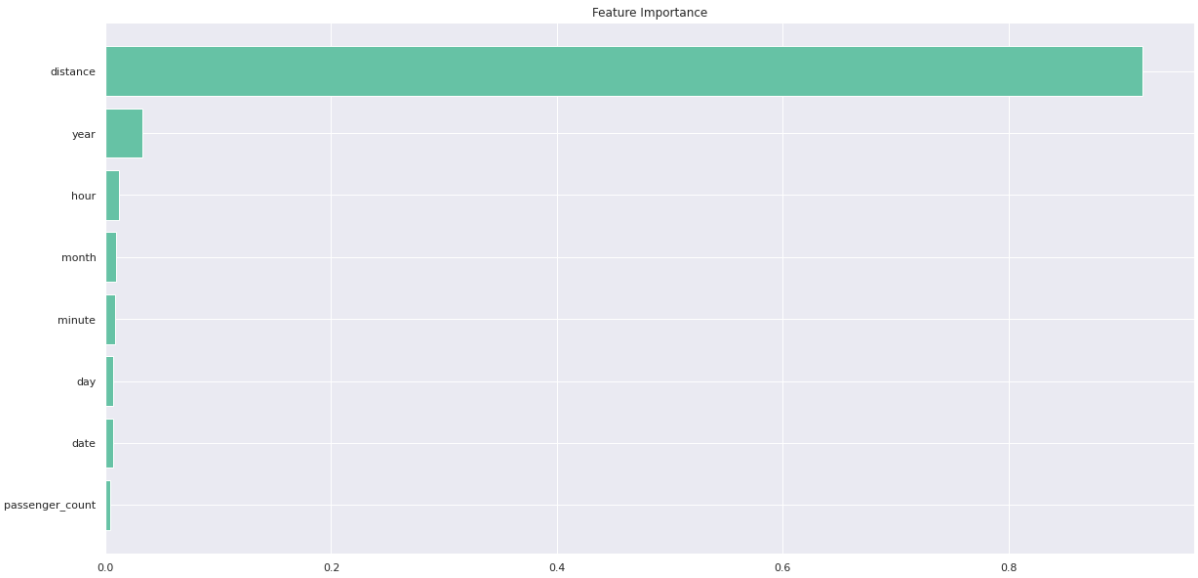

Figure 31: Important Features through XGBoost

Summary of model

```
<<<------------------- Training Data Score ---------------------->>>

r2 square   0.8554948600757166
Adjusted r square:0.8553953386294878
MAPE:6.284308617554664
MSE: 0.04276031902831015
RMSE: 0.20678568380888981

<<<------------------- Test Data Score ---------------------->>>

r2 square   0.8121022048988186
Adjusted r square:0.8117134843503806
MAPE:7.275888375932081
MSE: 0.056578051776532017
RMSE: 0.23786141296253163
```

MODEL EVALUATION

The quality of a regression model is how well its predictions match up against actual values, and Error metrics are used to judge the quality of a model, which enables us to compare regressions against other regressions with varied parameters.

Above mentioned, Linear Regression, Random Forest, and XGBoost Method are the various models that can be developed for the given data. At first place, The Data is divided into train and test. Then the models are developed on the train data. After that the model is fit into it to test data to predict the target variable. After predicting the target variable in test data, the actual and predicted values of target variable are compare to get the error and accuracy. And looking over the error and accuracy rates, the best model for the data is identified and it is kept for future usage.

A. Root Mean Squared Error (RMSE)

The Root Mean square Error (RMSE) and R-Squared area unit used for managing statistic prediction and continuous variables. The RMSE indicates absolutely the work of the model to the info, whereas R-Squared may be a relative live of work

RMSE should be compared with the variable quantity as RMSE is within the same units because the variable quantity.

Smaller The Result, Better The Performance Of The Model: To understand how well the independent variables "explain" the variance in the model, the R-Squared formula is used.

For The R-Squared, The Closer the Value To 1, The Better The Performance Of The Model: According to the underlying model, Table-1 describes its error metrics.

B. Mean Absolute Error (MAE/ MAPE):

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

MAE or Mean Absolute Error, it is one of the error measures that is used to calculate the predictive performance of the model. In this project we will apply this measure to our models.

The mean absolute percentage error (MAPE) is the mean or average of the absolute percentage errors of forecasts. Error is defined as actual or observed value minus the forecasted value.

C. R_squared:

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

The definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model. Or:

R-squared = Explained variation / Total variation
R-squared is always between 0 and 100%:
0% indicates that the model explains none of the variability of the response data around its mean.
100% indicates that the model explains all the variability of the response data around its mean.
In general, the higher the R-squared, the better the model fits your data. However, there are important conditions for this guideline that I'll talk about both in this post and my next post.

Model Evaluation Summary

1) Training Data

Algorithm	RMSE	MAPE	R2 score
Linear Regression	0.275937822759026	7.600500307764352	0.7426851654594733
Random Forest	0.144106772322434	3.89264377088736	0.9298203593402904
XGBoost	0.227030195765692	6.810463392740986	0.8258154602339608

2) Testing Data

Algorithm	RMSE	MAPE	R2 score
Linear Regression	0.248559385274895	7.50899013353772	0.7948204715133892
Random Forest	0.236654129783304	7.349878071773215	0.814004742668103
XGBoost	0.236271594805483	7.17992379856057	0.8146055535428931

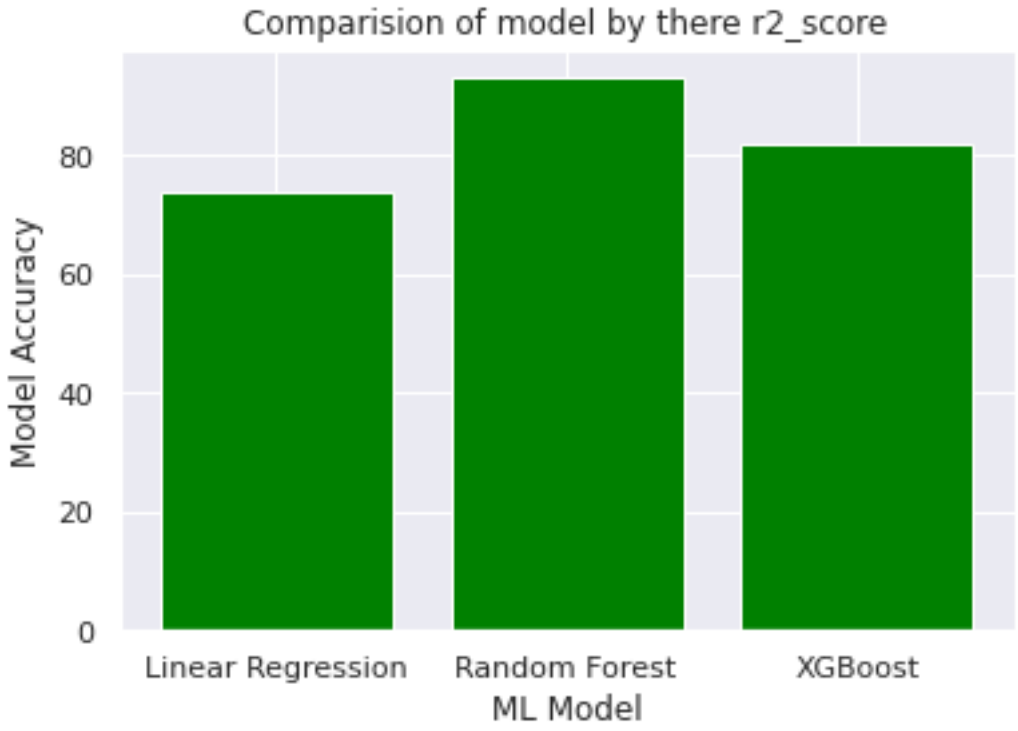

Figure 31: Model Comparison

Acceptance of Model

R-squared is a goodness-of-fit measure for linear regression models. This statistic indicates the percentage of the variance in the dependent variable that the independent variables explain collectively. R-squared measures the strength of the relationship between your model and the dependent variable on a convenient 0 – 100% scale.

From above summary we can see the maximum r2 score given by Random Forest algorithm with tuning parameters in both train and test data. Hence, we are going to finalize the Random Forest Algorithm for prediction of cab price.

Rejection of Model

Since MAPE is a measure of error, high numbers are bad and low numbers are good. we can see that linear regression and XGBoost having high numbers in term of MAPE and r2 score is also low for both the algorithm hence we are rejecting the linear regression and XGBoost for final prediction.

Conclusion

The quality of a regression model depends on the matchup of predictions against actual values. In regression problems, the dependent variable is continuous. In classification problems, the dependent variable is categorical. Linear Regression helps us to understand and predict the continuous values. Random Forest can be used to solve both regression and classification problems. The K-NN algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. XGBoost is the ensemble technique which take the collective decision of predictor value and give the output. Out of the three models left, Random Forest is the best model as it has the lowest RMSE score and highest R-Squared score, which explains the highest variability and tells us how well the model fits in this data.

DEPLOYMENT

1) First Step is to make environment ready for Executing all our codes that we can ensure using following command and we should have python 3.7 version and above

pip install -r /path/to/requirements.txt

2) Second Step is to load the Train and Test Data File as a command line argument to python file which we are going to use for deployment

The command will look like

D:\edwisor_details\project**> python .\cab_fare_prediction_deployment.py **
**"D:\edwisor_details\project\cab_fare_1\train_cab.csv" **
"D:\edwisor_details\project\cab_fare_1\test.csv"

3) After executing this above command, we will get model pickle file and test.csv file with predicted values in the same project folder

4) Another Approach is load the model file which is in pickle file give input as Test.csv and you will get an and predicted csv file as a output