Project Report On

CAB FARE PREDICTION

Submitted by
POONAM LAL

INDEX

CHAPTER 1

INRODUCTION

Now a day's cab rental services are expanding with the multiplier rate. The ease of using the services and flexibility gives their customer a great experience with competitive prices.

1.1 Problem Statement

You are a cab rental start-up company. You have successfully run the pilot project & 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.

1.2 Data Understanding

Understanding of data is the very first and important step in the process of finding solution of any business problem. Here in our case our company has provided a data set with following features, we need to go through each and every variable of it to understand and for better functioning.
Size of Dataset Provided: - 16067 rows, 7 Columns (including dependent variable) Missing

Values: Yes
Outliers Presented: Yes

Attribute Information:
1. pickup_datetime - timestamp value indicating when the cab ride started.
2. pickup_longitude - float for longitude coordinate of where the cab ride started.
3. pickup_latitude - float for latitude coordinate of where the cab ride started.
4. dropoff_longitude - float for longitude coordinate of where the cab ride ended.
5. dropoff_latitude - float for latitude coordinate of where the cab ride ended.
6. passenger_count - an integer indicating the number of passengers in the cab ride.

CHAPTER 2

Methodology

1. Pre-Processing

When we required to build a predictive model, we require to look and manipulate the data before we start modelling which includes multiple preprocessing steps such as exploring the data, cleaning the data as well as visualizing the data through graph and plots, all these steps is combined under one shed which is **Exploratory Data Analysis**, which includes following steps:

- Data exploration and Cleaning
- Missing values treament
- Outlier Analysis
- Feature Selection
- Features Scaling
    - Skewness and Log transformation
- Visualization

2. Modelling

Once all the Pre-Processing steps has been done on our data set, we will now further move to our next step which is modelling. Modelling plays an important role to find out the good inferences from the data. Choice of models depends upon the problem statement and data set. As per our problem statement and dataset, we will try some models on our preprocessed data and post comparing the output results we will select the best suitable model for our problem. As per our data set following models need to be tested:

- Linear regression
- Decision Tree
- Random forest
- Gradient Boosting

We have also used hyper parameter tunings to check the parameters on which our model runs best. Following are two techniques of hyper parameter tuning we have used:

- Random Search CV
- Grid Search CV

3. Model Selection

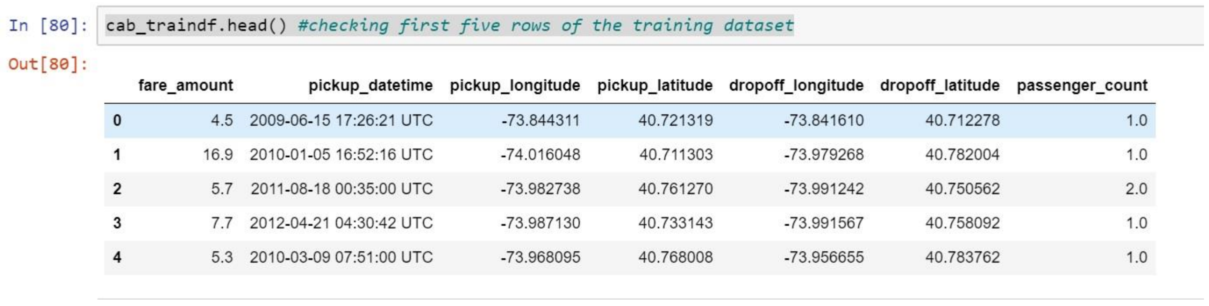
The final step of our methodology will be the selection of the model based on the different output and results shown by different models. We have multiple parameters which we will study further in our report to test whether the model is suitable for our problem statement or not.

CHAPTER 3

Pre-Processing

3.1.1 Exploratory Data Analysis

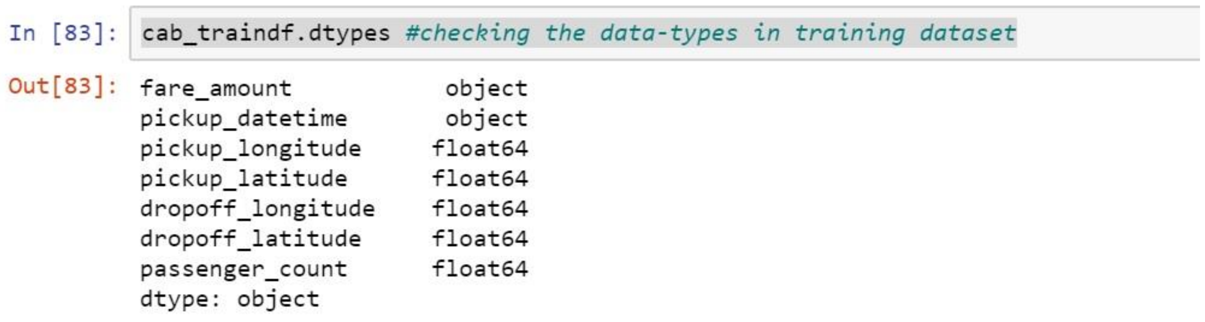
The very first step which comes with any data science project is data exploration and cleaning. In the given project, we have training data set of cab fare .The data which we have is unstructured in nature so, here we need to spend more time for data understanding, data cleaning, and data visualization to figure out new features that are better predictors of cab fare.
The data we have looks like this:



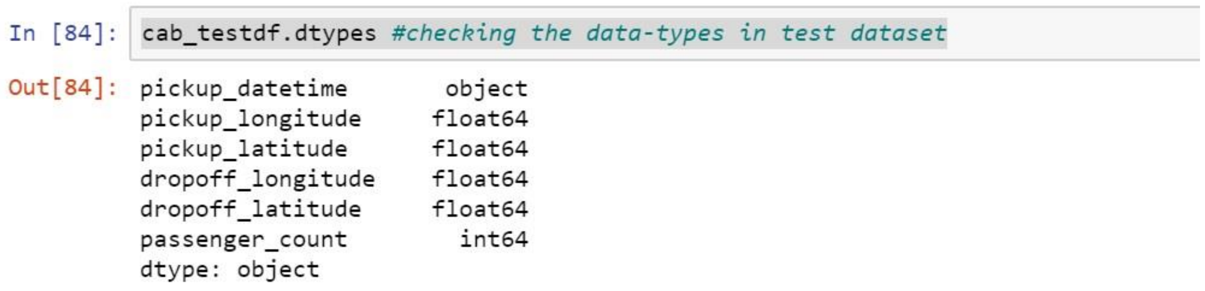
```
In [80]: cab_traindf.head() #checking first five rows of the training dataset
Out[80]:
```

	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

Datatype of train and test data:



```
In [83]: cab_traindf.dtypes #checking the data-types in training dataset

Out[83]: fare_amount           object
         pickup_datetime       object
         pickup_longitude     float64
         pickup_latitude      float64
         dropoff_longitude    float64
         dropoff_latitude     float64
         passenger_count      float64
         dtype: object
```



```
In [84]: cab_testdf.dtypes #checking the data-types in test dataset

Out[84]: pickup_datetime       object
         pickup_longitude     float64
         pickup_latitude      float64
         dropoff_longitude    float64
         dropoff_latitude     float64
         passenger_count        int64
         dtype: object
```

we can see the data-types of the given attributes. Here we need to change our data-types because pickup-date time which is indicating when the trip started it should be in timestamp. Fare_amount in dollar is nothing but our target variable it should be float. As we can see the target variable is not in test data set.

Assumption:

As we are talking about how independent variables will be effect on the target variable. So there will be the multiple assumptions.

1. Fare amount is highly depend on **trip distance** which we can calculate from pickup and dropout latitude and longitude

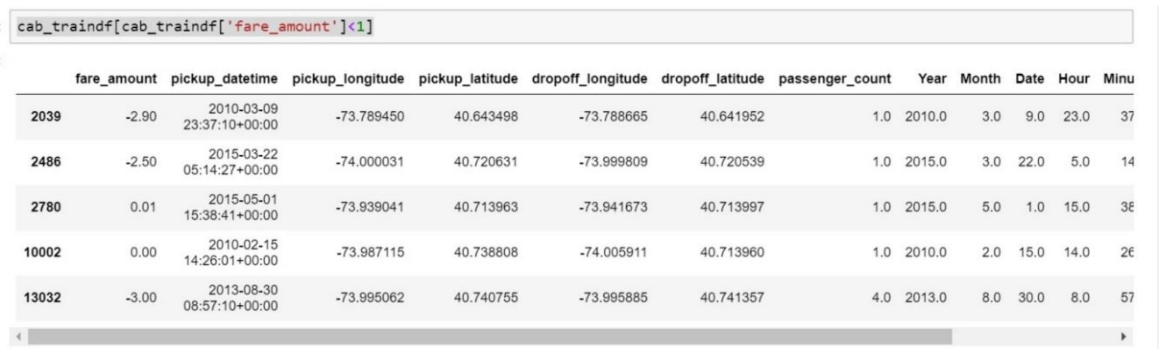
2. Fare amount is depending on how **much time it will take to travel from one place to another place**. Because, in the **traffic** it may be take more time. So, indirectly it will effect on fare amount. Pickup time is also impact on fare charges like suppose journey may be start in **night time so night charges** will be impact on fare amount.
3. Suppose any location from the New York City **available multiple cabs** so, it may be possible Fare rate will be less.

3.1.2 Data Cleaning

➢ ***Pickup_datetime***

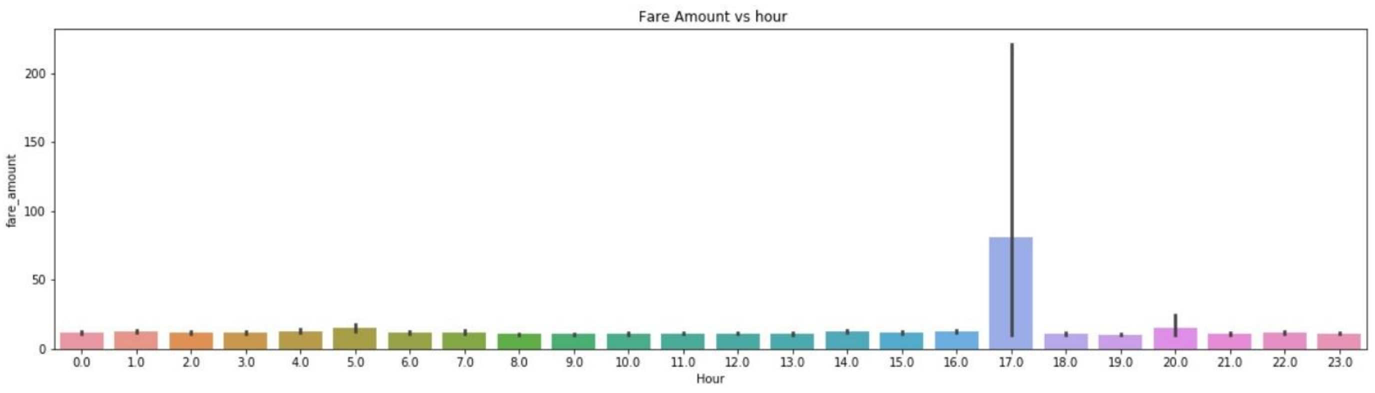
As we see in the train data set we have 6 independent and 1 target variables let's discusses one by one. **Pickup_datetime** telling us when the journey was stared like **2009-06-15 17:26:21 UTC** so, what we can do is we differentiate the above attribute in year, month, day of month, hour, minute, second as shown below.

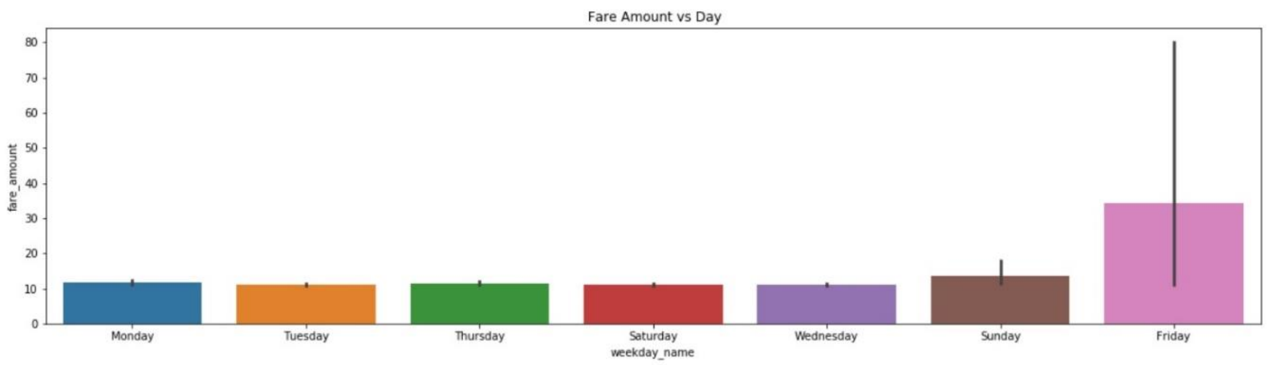
As we can see hour 17 is impacting more on the target variable. In this particular time people use more cab service.



```
cab_traindf[cab_traindf['fare_amount']<1]
```

	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	Year	Month	Date	Hour	Minu
2039	-2.90	2010-03-09 23:37:10+00:00	-73.789450	40.643498	-73.788665	40.641952	1.0	2010.0	3.0	9.0	23.0	37
2486	-2.50	2015-03-22 05:14:27+00:00	-74.000031	40.720631	-73.999809	40.720539	1.0	2015.0	3.0	22.0	5.0	14
2780	0.01	2015-05-01 15:38:41+00:00	-73.939041	40.713963	-73.941673	40.713997	1.0	2015.0	5.0	1.0	15.0	38
10002	0.00	2010-02-15 14:26:01+00:00	-73.987115	40.738808	-74.005911	40.713960	1.0	2010.0	2.0	15.0	14.0	26
13032	-3.00	2013-08-30 08:57:10+00:00	-73.995062	40.740755	-73.995885	40.741357	4.0	2013.0	8.0	30.0	8.0	57





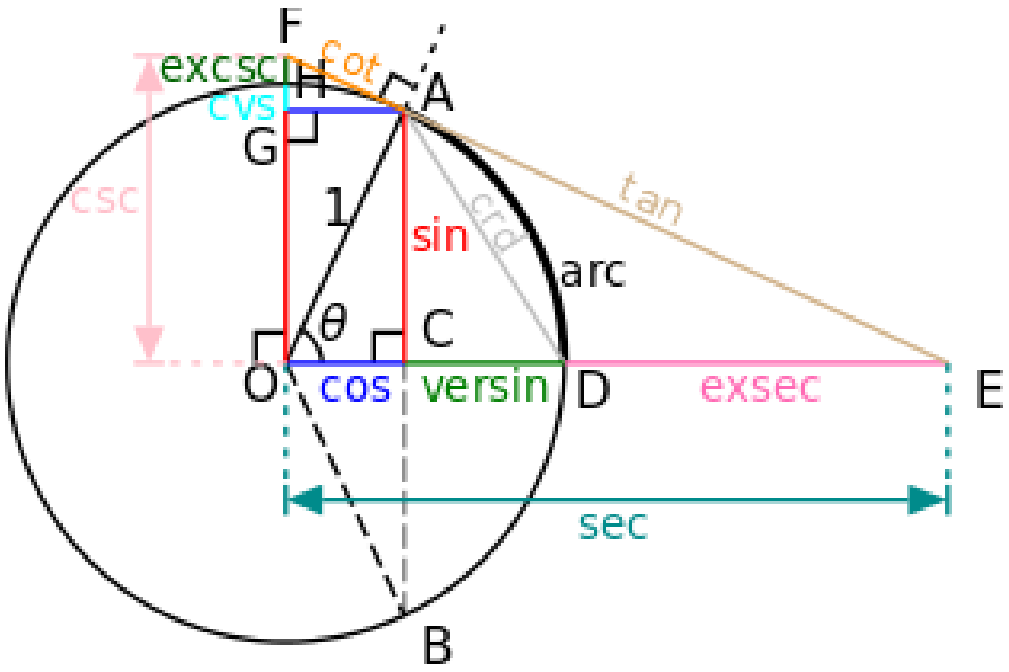
Above figure shows the relation between the days and fare amount.

- ***Pickup and Dropout Location :***

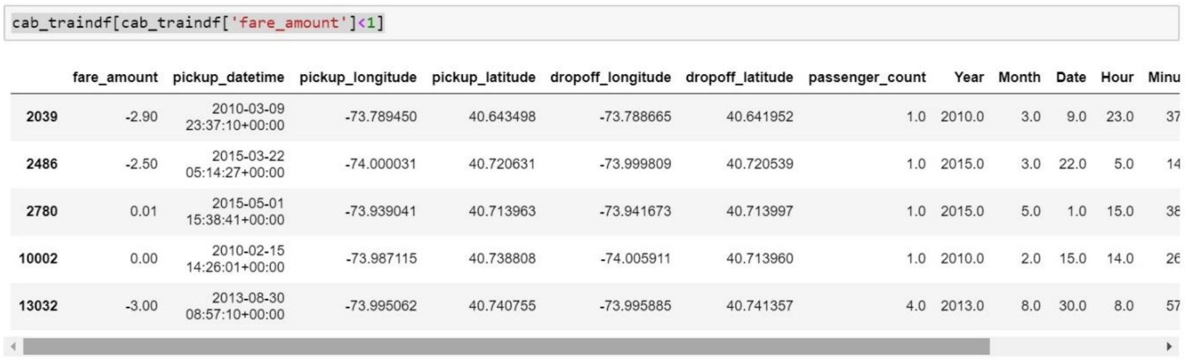
  In our data set pickup latitude, pickup longitude, drop-off latitude and drop-off longitude telling us the pickup and drop-off location. With the help this attributes here we can find the distance of trip using haversine formula. As we know the earth share is spherical or elliptical in nature so the 'haversine' formula help us to calculate the great-circle distance between two points that is, the shortest distance over the earth's surface distance. The formula to calculate the distance is shown below. *:*

  $a = \sin^2(\Delta\varphi/2) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2(\Delta\lambda/2)$
  $c = 2 \cdot \operatorname{atan2}(\sqrt{a}, \sqrt{(1-a)})$
  $d = R \cdot c$
  Where, **φ** *is latitude in radian,* **λ** *is longitude,* R *is earth's radius (6,371km)*

  The latitude and longitude point has to be entered within the range latitude: [−90, 90] longitude: [−180, 180]. But in few cases we can see latitude is 401.1 and longitude -73.9 also in some cases values of latlong is 0 it lies inside ocean and value in the range of latitude 39.61607524 longitude 73.9644596 are also lies within water. Practically speaking, it is not possible because cab service is not available inside the water and ocean those values are



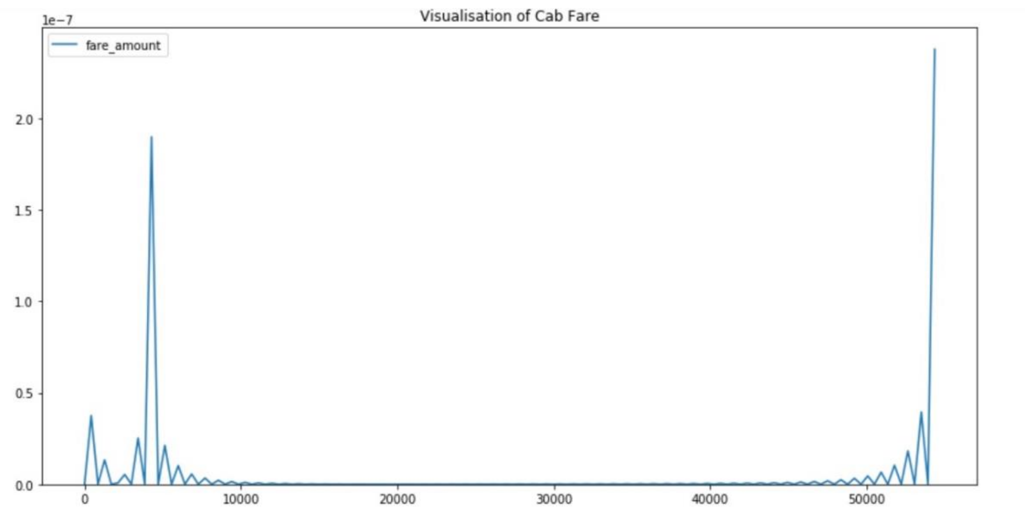
➢ ***Fare Amount*** **:**
When we plot the values of fare amount we came to know few values are negative.The min value is -3 and max 54343, but, as we know the cost of the journey cannot be negative so it is nothing but the impure values we must filter it out from our existing data set.



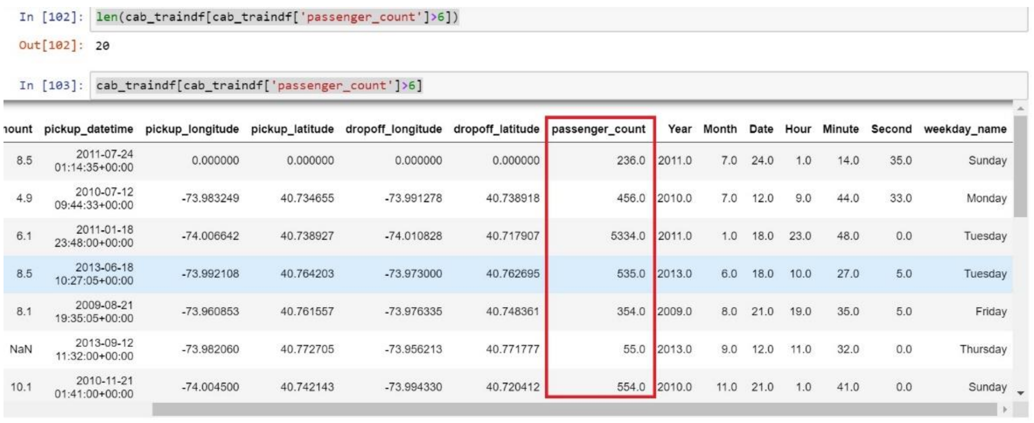
```
cab_traindf[cab_traindf['fare_amount']<1]
```

	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	Year	Month	Date	Hour	Minu
2039	-2.90	2010-03-09 23:37:10+00:00	-73.789450	40.643498	-73.788665	40.641952	1.0	2010.0	3.0	9.0	23.0	37
2486	-2.50	2015-03-22 05:14:27+00:00	-74.000031	40.720631	-73.999809	40.720539	1.0	2015.0	3.0	22.0	5.0	14
2780	0.01	2015-05-01 15:38:41+00:00	-73.939041	40.713963	-73.941673	40.713997	1.0	2015.0	5.0	1.0	15.0	38
10002	0.00	2010-02-15 14:26:01+00:00	-73.987115	40.738808	-74.005911	40.713960	1.0	2010.0	2.0	15.0	14.0	26
13032	-3.00	2013-08-30 08:57:10+00:00	-73.995062	40.740755	-73.995885	40.741357	4.0	2013.0	8.0	30.0	8.0	57

Visualisation of fare amount:

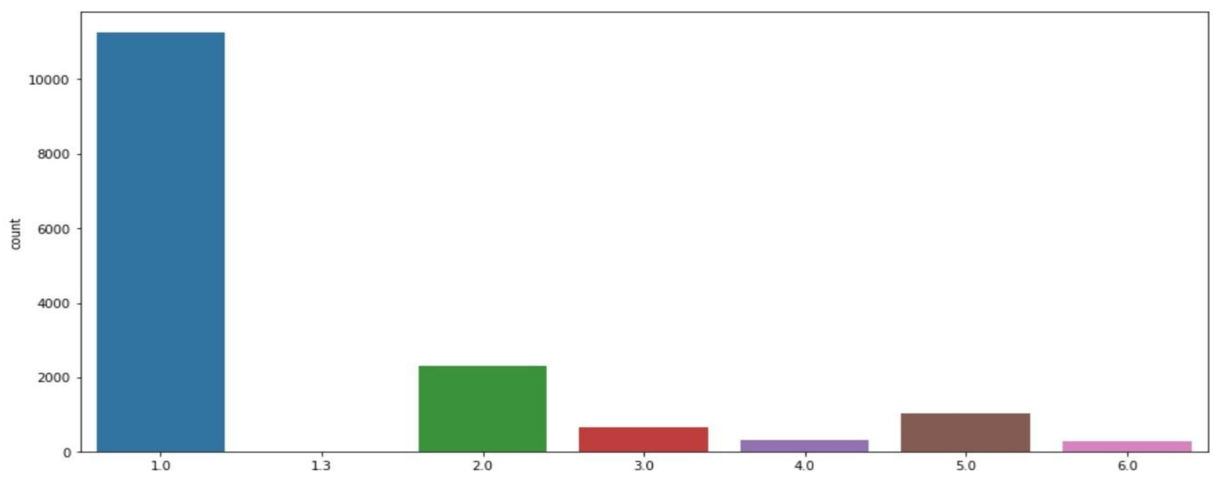


➢ ***passenger_count***
In train data set we can see the min value of passenger in single trip is 0 and maximum is 5345. Practically it is not possible because cab has desire setting capacity. Suppose we have 7 sitter cabs so max capacity is 6 passengers and 1 driver so in our data set beyond 6 we will consider as an outlier.



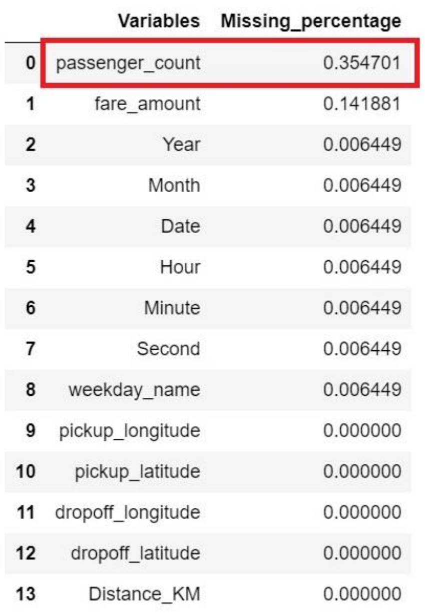
```
In [102]: len(cab_traindf[cab_traindf['passenger_count']>6])

Out[102]: 20

In [103]: cab_traindf[cab_traindf['passenger_count']>6]
```

ount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	Year	Month	Date	Hour	Minute	Second	weekday_name
8.5	2011-07-24 01:14:35+00:00	0.000000	0.000000	0.000000	0.000000	236.0	2011.0	7.0	24.0	1.0	14.0	35.0	Sunday
4.9	2010-07-12 09:44:33+00:00	-73.983249	40.734655	-73.991278	40.738918	456.0	2010.0	7.0	12.0	9.0	44.0	33.0	Monday
6.1	2011-01-18 23:48:00+00:00	-74.006642	40.738927	-74.010828	40.717907	5334.0	2011.0	1.0	18.0	23.0	48.0	0.0	Tuesday
8.5	2013-06-18 10:27:05+00:00	-73.992108	40.764203	-73.973000	40.762695	535.0	2013.0	6.0	18.0	10.0	27.0	5.0	Tuesday
8.1	2009-08-21 19:35:05+00:00	-73.960853	40.761557	-73.976335	40.748361	354.0	2009.0	8.0	21.0	19.0	35.0	5.0	Friday
NaN	2013-09-12 11:32:00+00:00	-73.982060	40.772705	-73.956213	40.771777	55.0	2013.0	9.0	12.0	11.0	32.0	0.0	Thursday
10.1	2010-11-21 01:41:00+00:00	-74.004500	40.742143	-73.994330	40.720412	554.0	2010.0	11.0	21.0	1.0	41.0	0.0	Sunday



3.1.3Missing Value Analysis

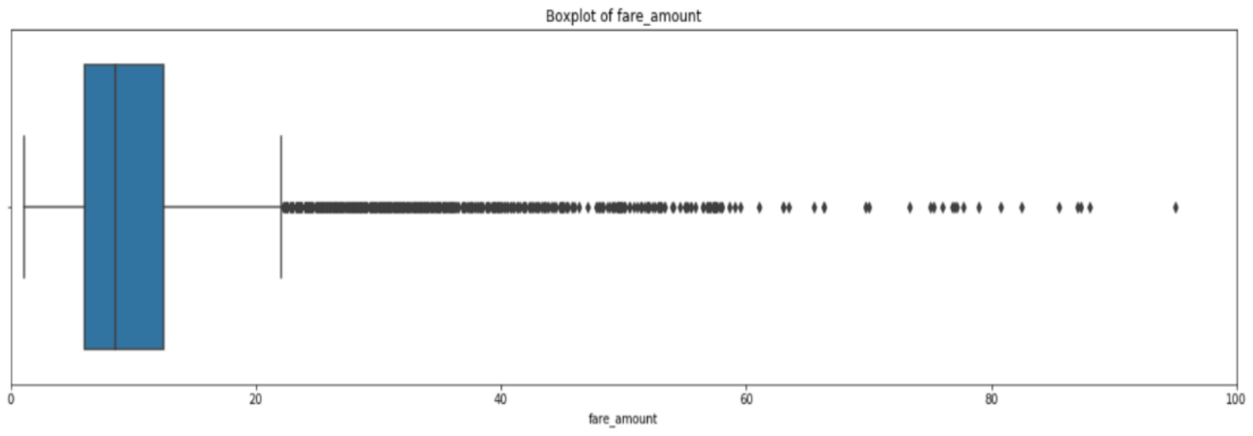
Missing value analysis plays a vital role in data preparing.
There are many reasons to occur missing values. In statistics while calculating missing values, if it is more than 30% we just drop the particular attribute because it does not carry much information to predict our target variables. As we can see in the below highest percentage of missing value is **0.354701** so we impute this missing value with the help of central statistic method i.e. median as shown below.

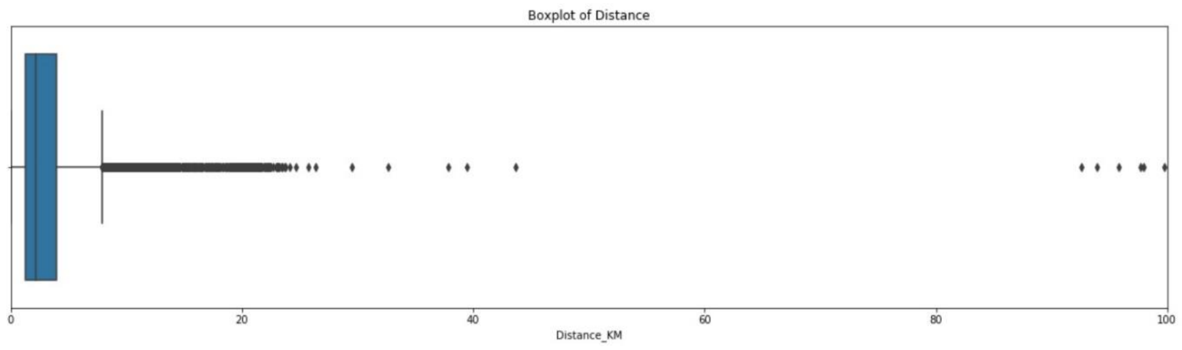


	Variables	Missing_percentage
0	passenger_count	0.354701
1	fare_amount	0.141881
2	Year	0.006449
3	Month	0.006449
4	Date	0.006449
5	Hour	0.006449
6	Minute	0.006449
7	Second	0.006449
8	weekday_name	0.006449
9	pickup_longitude	0.000000
10	pickup_latitude	0.000000
11	dropoff_longitude	0.000000
12	dropoff_latitude	0.000000
13	Distance_KM	0.000000

3.1.4 Outlier Analysis

Outlier is the observation which is inconsistent related with all data set. The values of Outliers are the accurate but it is far away from the set of actual values and it heavily impact on the mean so that we consider it as an outlier. Here we have used box plot method to detect outliers as shown below:





Here what we did is we remove some outliers manually as discussed in **chapter 1**. But in some cases we find the extreme values and if we consider the same it will impact on mean. So we must have to remove it from our train data set. There are two method use to remove an outlier i.e. KNN and box plot here we used box plot method.

3.2 Creating some new variables from the given variables.

Here in our data set our variable name pickup_datetime contains date and time for pickup. So we tried to extract some important variables from pickup_datetime:

• Year

• Month

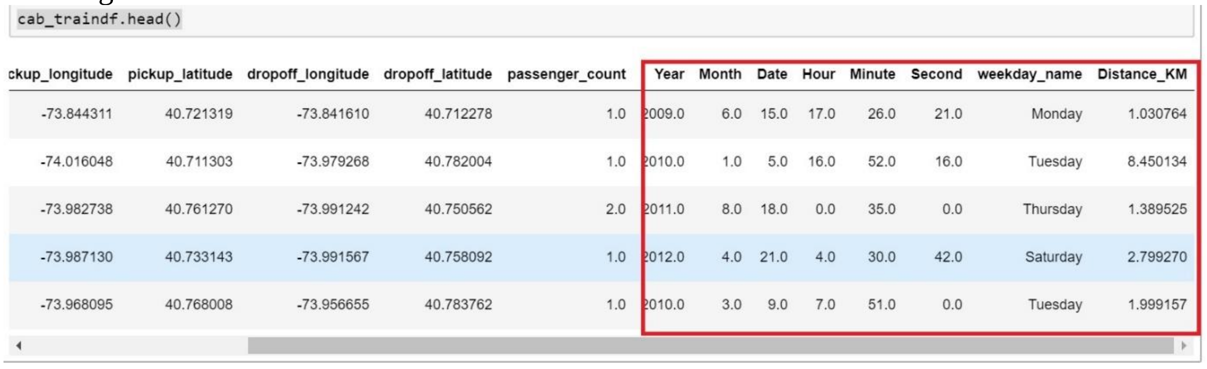
• Date

• Day of Week

• Hour

• Minute

Also, we tried to find out the distance using the haversine formula which says:
The **haversine formula** determines the great-circle distance between two points on a sphere given their longitudes and latitudes. Important in navigation, it is a special case of a more general formula in spherical trigonometry, the law of haversines, that relates the sides and angles of spherical triangles.

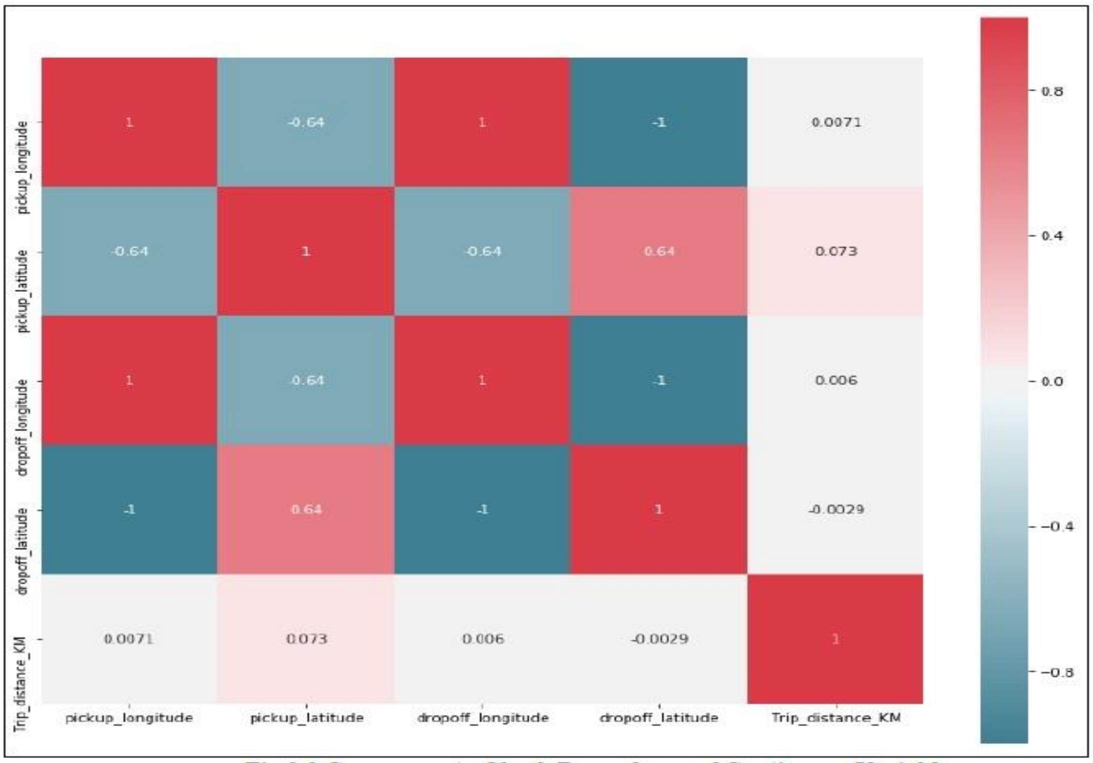


```
cab_traindf.head()
```

ckup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	Year	Month	Date	Hour	Minute	Second	weekday_name	Distance_KM
-73.844311	40.721319	-73.841610	40.712278	1.0	2009.0	6.0	15.0	17.0	26.0	21.0	Monday	1.030764
-74.016048	40.711303	-73.979268	40.782004	1.0	2010.0	1.0	5.0	16.0	52.0	16.0	Tuesday	8.450134
-73.982738	40.761270	-73.991242	40.750562	2.0	2011.0	8.0	18.0	0.0	35.0	0.0	Thursday	1.389525
-73.987130	40.733143	-73.991567	40.758092	1.0	2012.0	4.0	21.0	4.0	30.0	42.0	Saturday	2.799270
-73.968095	40.768008	-73.956655	40.783762	1.0	2010.0	3.0	9.0	7.0	51.0	0.0	Tuesday	1.999157

3.3 Feature Selection

As we know, while developing the model if we consider the independent variables which carries the same information to explain the target variables it will create the problem of multi-collinearity. So to avoid our model from the multi-collinarity problem we need to applied Feature Selection or dimensional reduction on the top of our data set. It helps us to sort out the variables which are highly correlated with each other. In our case we applied **correlation analysis** for numeric variables and **ANOVA** for the categorical variables In below figure **pickup_longitude, pickup_latitude, dropoff_longitude, dropoff latitude** those all attributes are highly correlated with each other. It means those variables carry same information to explain the target variable actually in **Chapter 1 we can drop those all variables after calculation of Trip distance but below with the help of visualization method we analyses it batter.**



When we applied **ANOVA** test on categorical variables the p value of all the variables is **0** it means there no any dependency, as shown below

```
P value for variable Year is 0.0
f value for variable Year is 830571802.5619931
P value for variable Month is 0.0
f value for variable Month is 3440.745340710477
P value for variable Date is 0.0
f value for variable Date is 2674.202779706927
P value for variable Hour is 5.618781442514664e-263
f value for variable Hour is 1224.907602036647
P value for variable Minute is 0.0
f value for variable Minute is 14740.115953957082
P value for variable Second is 2.9598793244452687e-224
f value for variable Second is 1039.913730777895
P value for variable passenger_count is 0.0
f value for variable passenger_count is 16926.93974348641
```

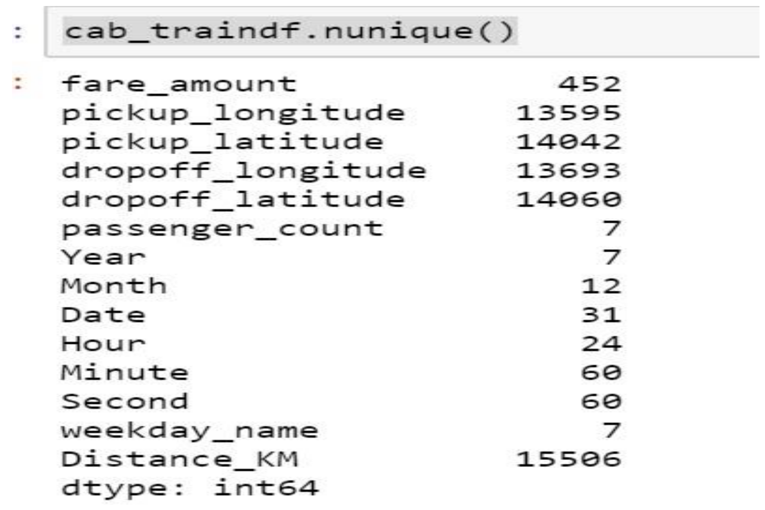
3.4 Some more data exploration

3.4.1 Dependent and Independent Variables:

- ***Independent variables:*** passenger_count, year, Month, Date, Day of Week, Hour Distance_KM
- ***Our Dependent variable*** : fare_amount

3.4.2 Uniqueness in Variable :

We need to look at the unique number in the variables which help us to decide whether the variable is categorical or numeric. So, by using python script ‘nunique’ we tried to find out the unique values in each variable. We have also added the table below:



```
: cab_traindf.nunique()

: fare_amount              452
  pickup_longitude       13595
  pickup_latitude        14042
  dropoff_longitude      13693
  dropoff_latitude       14060
  passenger_count            7
  Year                       7
  Month                     12
  Date                      31
  Hour                      24
  Minute                    60
  Second                    60
  weekday_name               7
  Distance_KM            15506
  dtype: int64
```

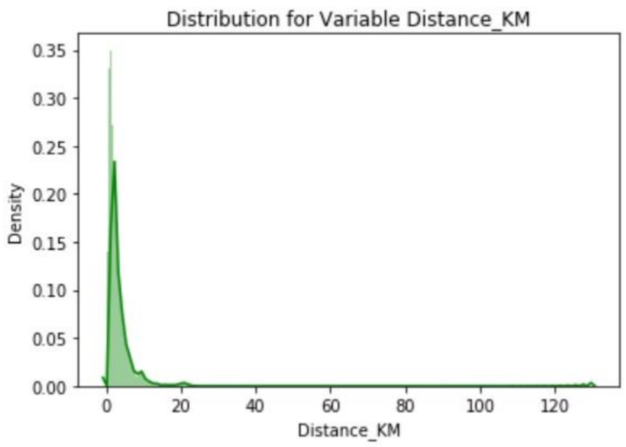
3.4.3 Dividing the variables

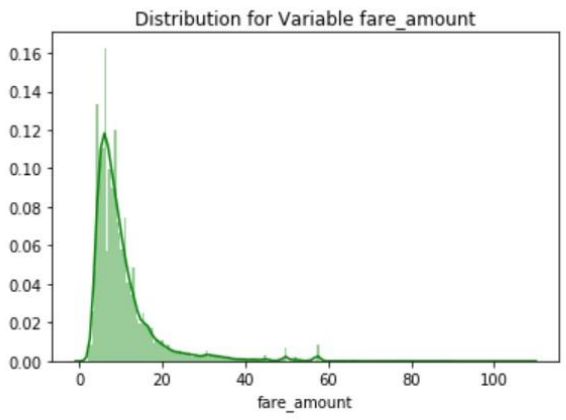
Dividing the variable into two categories basis their data   types:

- ***Continuous variables*** **-** 'fare_amount', 'distance'.
- ***Categorical Variables*** **-** 'year', 'Month', 'Date', 'Day of Week', 'Hour', 'passenger_count'

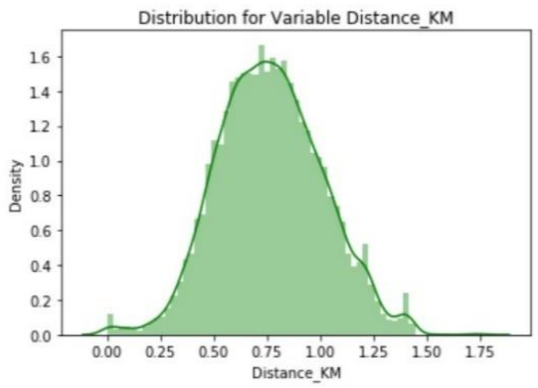
3.5 Feature Scaling

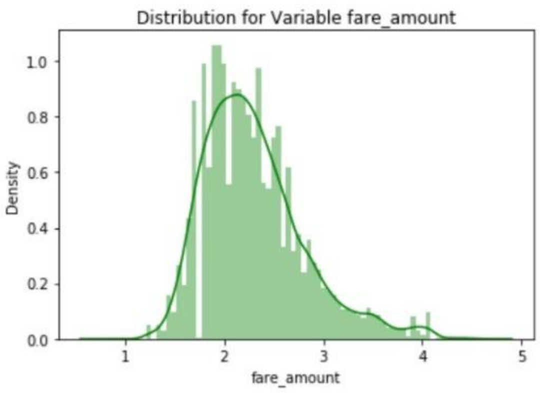
Skewness is asymmetry in a statistical distribution, in which the curve appears distorted or skewed either to the left or to the right. Skewness can be quantified to define the extent to which a distribution differs from a normal distribution. Here we tried to show the skewness of our variables and we find that our target variable absenteeism in hours having is one sided skewed so by using **log transform** technique we tried to reduce the skewness of the same.
Below mentioned graphs shows the probability distribution plot to check distribution before log transformation:





Below mentioned graphs shows the probability distribution plot to check distribution after log transformation:





CHAPTER 3

Modelling

After data cleaning and exploratory data analysis phase, we finally arrived at the model building phase. In this chapter we will applied multiple machine learning algorithm to predict the test case. In cab fare prediction project our target variable i.e. fare amount is numeric (predicting and forecasting type of problem) so that here we are using regression models on structure data to predict test case.

The next step is to differentiate the train data into 2 parts i.e. train and test. The splitting of train data into 2 parts is very important factor to verify the model performance and to understand the problem of over-fitting and under-fitting. Over-fitting is the term where training error is low and testing error is high and under-fitting is the term where both training and testing error is high. Those are the common problem of complex model.

In this analysis, since we are predicting fare amount which is the numeric variable. So, we come to know that, our problem statement is predicting (forecasting) type. So, what we can do is we will apply supervise machine learning algorithms to predict our target variable. As we know our target variable is continuous in nature so, here we will build regression matrix model. **Root Mean Square Error** (RMSE) to measures how much **error** there is between two data sets. In other words, it compares a predicted value and an observed or known value. The RMSE is directly interpretable in terms of measurement units, and so is a better measure of goodness of fit. So, in our case any model we build should have lower value of an **RMSE** and higher value of variance i.e**. R square.**

4.1 Multiple Liner Regression

Multiple Linear Regression is one of the statistical methods of prediction. It is used to find a linear relationship between the target and one or more predictors. It means the target variables should be continuous in nature. The main idea is to identify a line that best fits the data. To build any model we have some assumptions to put on data and model. This algorithm is not very flexible, and has a very high bias. Below we calculated RMSE and R^2 values using liner regression.

	Training	Test
R^2	0.779	0.720
RMSE	0.237	0.268
MAPE	7.459	7.926
Adj R^2	0.779	0.720

4.2 Decision Tree

A tree has many analogies in real life, and turns out that it has influenced a wide area of machine learning, covering both classification and regression. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions.

Below is the screenshot of the query we executed and the result shown, we will compare the results of each model in a combined table later on.

	Training	Test
R^2	0.703	0.668
RMSE	0.275	0.292
MAPE	9.067	9.432
Adj R^2	0.703	0.667

4.3 Random Forest

Random forest is the collection of multiple decision trees. In Random Forest, output is the average prediction by each of these trees. For this method to work, the baseline models must have a lower bias. The idea behind Random Forest is to build n number of trees to have more accuracy in dataset. Random Forest uses bagging method for predictions. It can handle large no of independent variables without variable deletion and it will give the estimates that what variables are important. The RMSE and R^2mvaalue of our project are shown below.

	Training	Test
R^2	0.969	0.746
RMSE	0.087	0.255
MAPE	2.716	7.608
Adj R^2	0.969	0.745

4.4 Gradient Boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

	Training	Test
R^2	0.823	0.766
RMSE	0.212	0.245
MAPE	6.748	7.239
Adj R^2	0.823	0.765

4.5 Hyper Parameters Tunings for optimizing the results

Model hyperparameters are set by the data scientist ahead of training and control implementation aspects of the model. The weights learned during training of a linear regression model are parameters while the number of trees in a random forest is a model hyperparameter because this is set by the data scientist.

Hyperparameters can be thought of as model settings. These settings need to be tuned for each problem because the best model hyperparameters for one particular dataset will not be the best across all datasets. The process of hyperparameter tuning (also called hyperparameter optimization) means finding the combination of hyperparameter values for a machine learning model that performs the best - as measured on a validation dataset - for a problem.

- Random Search CV
- Grid Search CV

Here we have used two hyper parameters tuning techniques

- **Random Search CV**: This algorithm set up a grid of hyperparameter values and select random combinations to train the model and score. The number of search iterations is set based on time/resources.
- **Grid Search CV**: This algorithm set up a grid of hyperparameter values and for each c combination, train a model and score on the validation data. In this approach, every single combination of hyperparameters values is tried which can be very inefficient.

CHAPTER 5

Conclusion

In above chapters we applied multiple pre-processing to frame our data into the structural format and different machine learning algorithm to check the performance of model. In this chapter we finalize one of them.

5.1 <u>Model Evaluation</u>

The main concept of looking at what is called residuals or difference between our predictions f(x[I,]) and actual outcomes y[i].
In general, most data scientists use two methods to evaluate the performance of the model:

- **RMSE** (Root Mean Square Error): is a frequently used measure of the difference between values predicted by a model and the values actually observed from the environment that is being modelled.
- **R Squared(R^2):** 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. In other words, we can say it explains as to how much of the variance of the target variable is explained.
- We have shown both train and test data results, the main reason behind showing both the results is to check whether our data is overfitted or not.
- Here we drop MAPE and Adj R square method because MAPE is showing way high values as compare to RMSE and R^2 method.
  On the other hand Adj R^2 value gives/low result as compare to R^2.

Below table shows the model results before applying hyper tuning:

	RMSE(Train)	RMSE(Test)	R^2(Train)	R^2 (Test)
Multiple LR	0.237	0.268	0.779	0.720
Decision Tree	0.275	0.292	0.703	0.668
Random Forest	0.087	0.225	0.969	0.746
Gradient Boosting	0.212	0.245	0.823	0.766

Below table shows results post using hyper parameter tuning techniques:

	RMSE(Test)	R^2 (Test)
Random Search CV (Random Forest **)**	0.250	0.76
Random Search CV (Gradient Boosting)	0.270	0.72
Grid Search CV (Random Forest)	0.249	0.76
Grid Search CV (Gradient Boosting)	0.254	0.75

5.2 Model Selection

On the basis RMSE and R Squared results a good model should have least RMSE and max R Squared value. So, from above tables we can see:

- From the observation of all RMSE Value and R-Squared Value we have concluded that,
- Both the models- Gradient Boosting Default and Random Forest perform comparatively well while comparing their RMSE and R-Squared value.
- After this, I chose Random Forest CV and Grid Search CV to apply cross validation technique and see changes brought about by that.
- After applying tunings Random forest model shows best results compared to gradient boosting.
- So finally, we can say that Random forest model is the best method to make prediction for this project with highest explained variance of the target variables and lowest error chances with parameter tuning technique Grid Search CV

Finally, I used this method to predict the target variable for the test data file shared in the problem statement. Results that I found are attached with my submissions.

############### End of Report##############

References

- For Data Cleaning and Model Development – https://edwisor.com/career-data-scientist
- For other code related queries – https://stackoverflow.com