qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm

|  |
| --- |
| CAB FARE PREDICTION PROJECT  EDWISOR DATA SCIENCE  7/22/2020  Submitted by  MUKESH KUMAR SABLANI |

**Table of contents**

**1.Introduction**

1.1 Problem Statement

1.2 Type of Problem

1.3 Data

1.4 Software and Hardware requirement

1.5 Exploratory Data Analysis

**2 Data Preprocessing**

2.1 Missing value analysis

2.2 Outlier analysis

2.3 Feature selection

2.4 Feature scaling

**3 .Model Development**

3.1 Linear Regression

3.2 Decision Tree

3.3 Random Forest

3.5 Prediction on test data

**4 Model Selection**

**5. Model fitting and Conclusion**

**6.References**

**1.Introduction**

Data Science project on Cab Fare Prediction, Machine learning algorithms are used to develop a regression model Problem Statement : The project is about a cab company who has done its pilot project and now they are looking to predict the fare for their future transactional cases. As, nowadays there are number of cab companies like Uber, Ola, Meru Cabs etc. And these cab companies deliver services to lakhs of customers daily. Now it becomes really important to manage their data properly to come up with new business ideas to get best results. In this case, earn most revenues. So, it becomes really important estimate the fare prices accurately.

* 1. **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 forfare prediction. You need to design a system that predicts the fare amount for a cab ride in the city.

* 1. Type of problem-

It is important to understand the idea of project behind the data set. The given data set is asking us to predict fare amount. And it really becomes important for us to predict the fare amount accurately.The problem clearly see above statement is a Regression problem as we need to predict the continues or catagoric variables of cab fare amount. So this can be solved any of the Regression techniques like:

1. Linear Regression
2. Decision Tree
3. Random Forest
   1. Data-

To get the the best result, to get the most effective model it is really important to our data very well. Here the given cab\_train data is a csv file consists 7 variable and 16067observation

*Number of attributes:*

* + - **fare\_amount** :fare of the given cab ride.
    - **pickup\_datetime** : timestamp value explaining the time of ride start.
    - **pickup\_longitude** : a float value explaining longitude location of the ride start.
    - **pickup\_latitude** : a float value explaining latitude location of the ride start.
    - **dropoff\_longitude :** a float value explaining longitude location of the ride end**.**
    - **dropoff\_latitude** : a float value explaining latitude location of the ride end **passenger\_count :** an integer indicating the number of passengers

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| fare\_amount | pickup\_datetime | pickup\_longitude | pickup\_latitude | dropoff\_longitude | dropoff\_latitude | passenger\_count |
| 4.5 | 2009-06-15  17:26:21 UTC | -73.844311 | 40.721319 | -73.84161 | 40.712278 | 1 |
| 16.9 | 2010-01-05  16:52:16 UTC | -74.016048 | 40.711303 | -73.979268 | 40.782004 | 1 |
| 5.7 | 2011-08-18  00:35:00 UTC | -73.982738 | 40.76127 | -73.991242 | 40.750562 | 2 |
| 7.7 | 2012-04-21  04:30:42 UTC | -73.98713 | 40.733143 | -73.991567 | 40.758092 | 1 |
| 5.3 | 2010-03-09  07:51:00 UTC | -73.968095 | 40.768008 | -73.956655 | 40.783762 | 1 |
| 12.1 | 2011-01-06  09:50:45 UTC | -74.000964 | 40.73163 | -73.972892 | 40.758233 | 1 |

**Table1.2 Cab\_train data**

Passenger following table shows how are the categorized

|  |
| --- |
| Independent Variables |
| pickup\_datetime |
| pickup\_longitude |
| pickup\_latitude |
| dropoff\_longitude |
| dropoff\_latitude |
| passenger\_count |

|  |
| --- |
| Dependent  Variables |
| Fare\_amount |

**Table 1.3 : Independent Variables Table 1.3 : Dependent/target variables**

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

pickup 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 snap of the test data provided

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| pickup\_dateti  me | pickup\_longitu  de | pickup\_latitud  e | dropoff\_longit  ude | dropoff\_latitud  e | passenger\_cou  nt |
| 2015-01-27  13:08:24 UTC | -73.97332 | 40.7638054 | -73.9814301 | 40.7438355 | 1 |
| 2015-01-27  13:08:24 UTC | -73.9868622 | 40.7193832 | -73.9988861 | 40.7392006 | 1 |
| 2011-10-08  11:53:44 UTC | -73.982524 | 40.75126 | -73.979654 | 40.746139 | 1 |

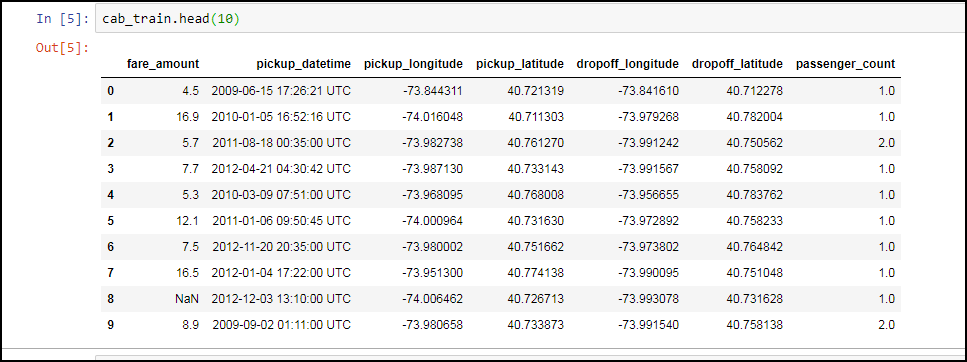
**Table1.4 test data**

* 1. **Software and Hardware requirement-**

1. R 3.6.1 for 64 bit
2. Anaconda 3 for 64 bit
3. R studio
4. 64 bit OS
5. Python 3
6. Jupyter Notebook
7. 4GB of RAM

**1.5 Exploratory Data analysis**

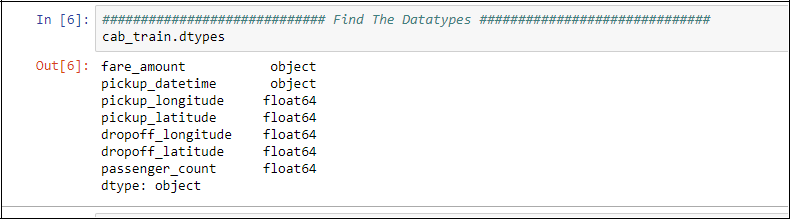
It is a approach where we have analyze data sets to summarize and performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.It is a good practice to understand the data first and try to gather as many insights from it. EDA is all about making sense of data in hand, before getting them dirty with it.



.

**Figure 1.5.1 Training Data**

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.



**Figure 1.5.2 Training Datatype**

**1.5.1 Assumption:**

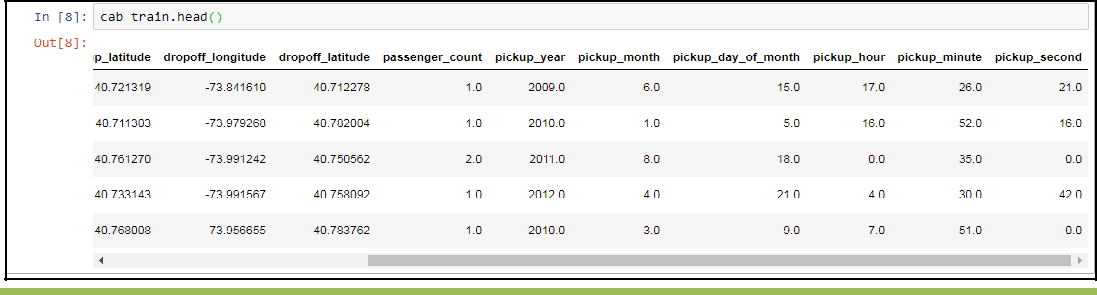
As we are talking about how independent variables will be effect on the target variable. So there will be the multiple assumptions.

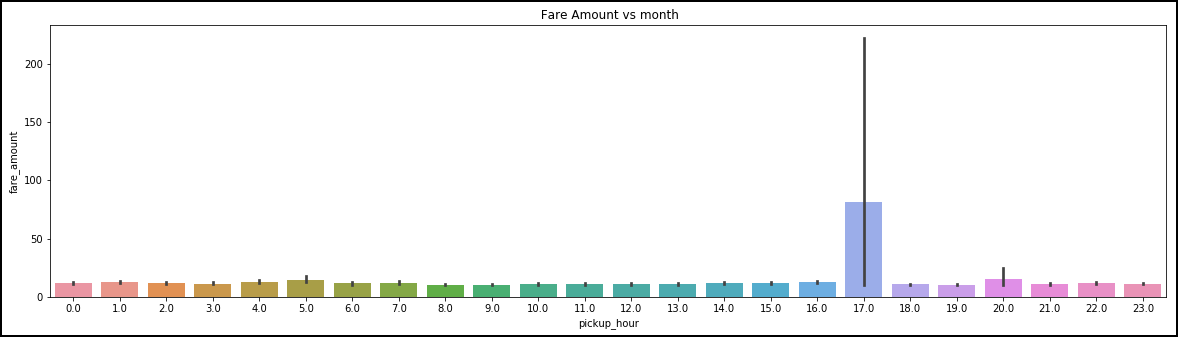
1. Fare amount is highly depend on **trip distance** which we can calculate from pickup and dropout latitude and longitude
2. Fare amount is depending on how **much time it will take to travel from one place to** **another place**. Because, in the **traffic** it may be take more time. So, indirectly it willeffect on fare amount.
3. 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.
4. Suppose any location from the yourCity **available multiple cabs** so, it may be possible fare rate will be less.

**1.5.2 Data Understanding and Cleaning**

* ***Pickup\_datetime***

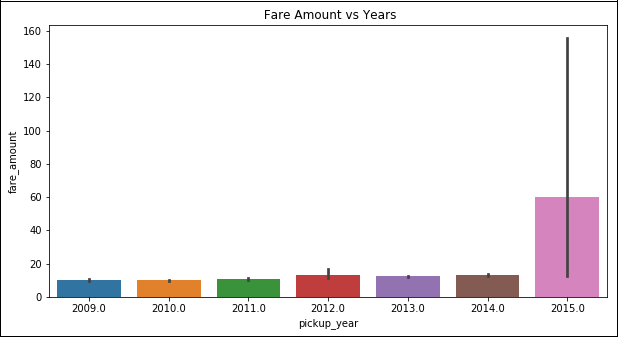
As we see in the train data set we have 6 independent and 1 target variables let’s discusses one by one. **Pickup\_datetime** telling us when the journey was stared like **2009-06-15 17:26:21UTC** so, what we can do is we differentiate the above attribute in year, month, day of month, hour,minute, second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



** Figure 1.5.3 Training Datatypes**

**Figure 1.5.4 Pickup\_hour veds fare amount**

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



**1.5. Pickup\_year and fare amount**

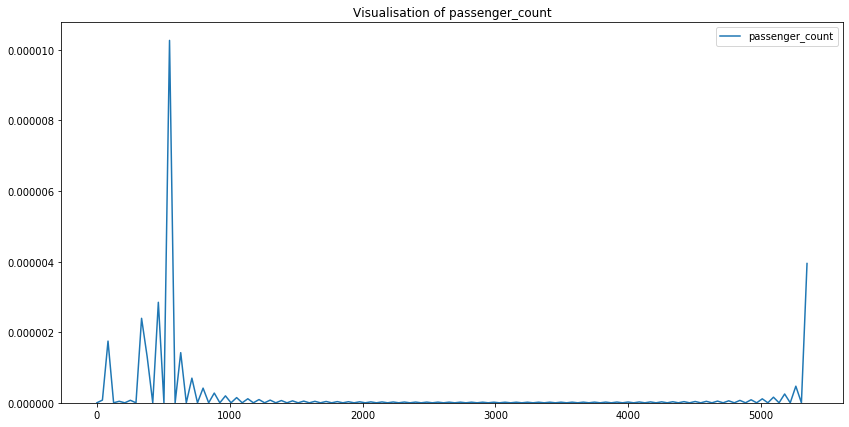
**Fare\_amount**

****

**Fig.1.6 Visualization 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

****

**Fig.1.7 Visualization of passenger\_count**

**2.Data Preprocessing**

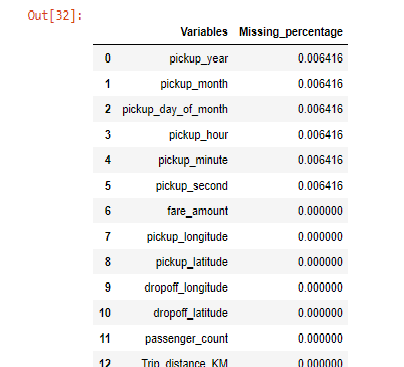
This step involves cleaning the data, dropping unwanted attributes, conversion of data-types to machine-understandable format and when our unstructured data come up into structure format then finally we will split the training data into train and validation sets. Here we already removed few unstructured data from our training data set in above chapter but still few impurities are there we will figure it out using below methods

**2.1 Missing Value Analysis-**

Missing values occur when no data is recorded for an observation; it was intended to make an observation, but because of some reason did not. Missing value analysis plays a vital role in data preparing. Due to improper handling, the result obtained by the researcher will differ from ones where the missing values are present.

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 then it is suggested to impute them with methods such as mean, mode, median, knn imputation **methods**.

As we can see in the below **Fig. 2.1** highest percentage of missing value is **0.006416** and it is negligible but, then also we impute this missing value with the help of central statistic method i.e. median as shown below.



**Methodology**

• Convert the negative values to zero in the variable ‘fare\_amount’ as it cannot be negative in real case scenario

• Create a data frame with total no of missing values for each variable

• Calculate percentage of missing values against each variable

• Sort in descending order

• In this project, missing values are decided to be imputed irrespective of their percentage, to avoid information loss

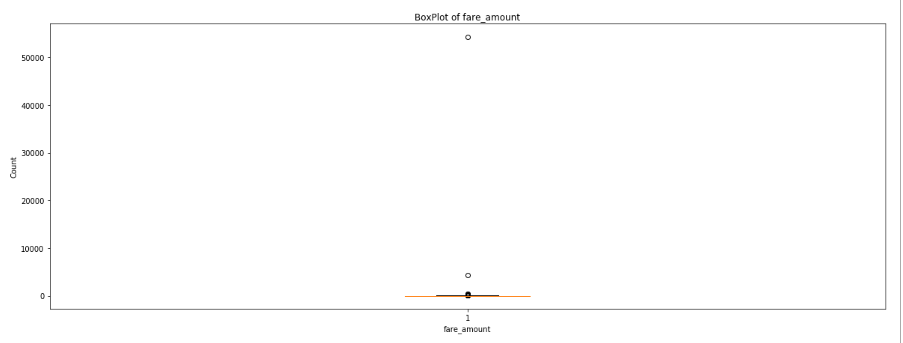
**2.2 Outlier Analysis-**

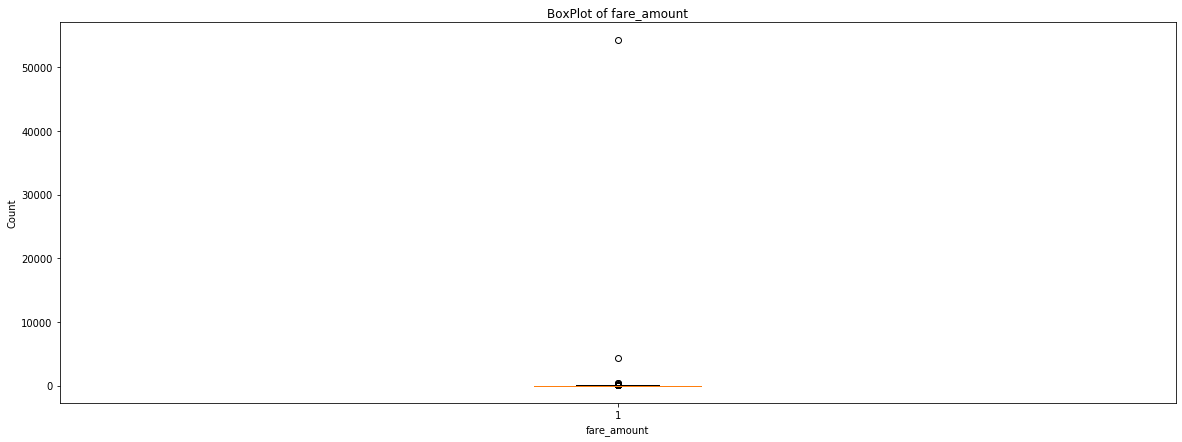
An outlier is an observation that is abnormal compared to other observations in that dataset. One of the most important tasks from large data sets is to find an outlier because outliers can significantly alter the results even though they are present in small proportions.

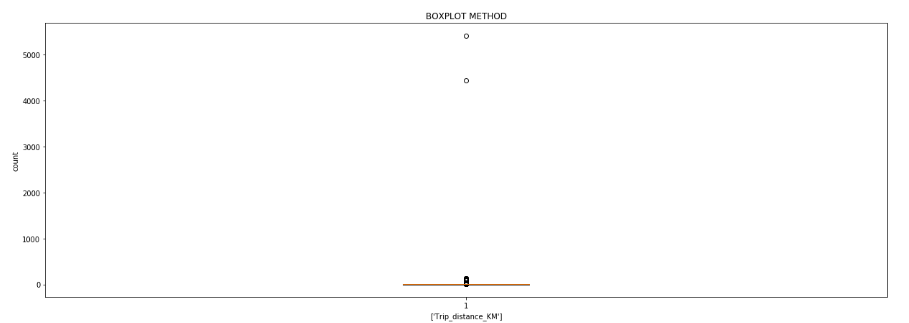
To find an outlier, inter quartile range (IQR) is found first. IQR represents the middle 50% of the data. The position of first quartile can be found using formula (N+1)/4 & third quartile can be found by 3\*(N+1)/4 where N is the total no. of observations. The difference of the values in the first & third quartiles is the IQR. If any observation falls below 1.5 times IQR from the first quartile value, or if it falls above 1.5 times IQR from the third quartile value, then the value can be qualified as an outlier.

Outliers can be found using box plot method which can be plotted in both R & Python. After finding the outliers, they can be removed from the dataset or they can be imputed by KNN method.

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 **Fig 2.2**

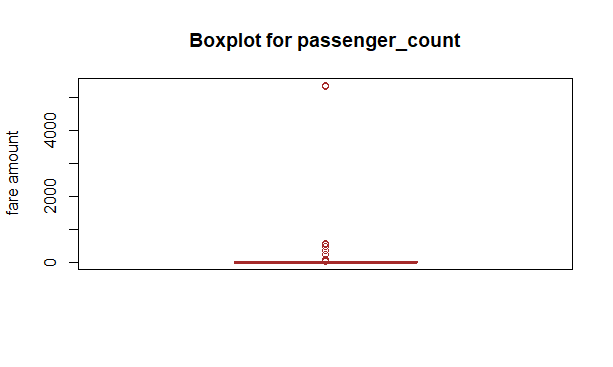
****



****

**Fig.2.2 Finding Outliers Using Boxplot**

**One of the box-plot is shown below in R:**



It could be concluded from the box plot that most of the observations were very close to each other and only few outliers were present.

Here what we did is we remove some outliers manually as discussed in **EDA**. 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.

**2.3 Feature Selection**

Feature Selection is the process of selecting those features which contribute most to the prediction variable. Having irrelevant features in data can decrease the accuracy of the models and make the model learn based on irrelevant features. Correlation analysis is used in feature selection for numerical variables 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

## The function of haversine function is described, which helped me to engineer our new variable, Distance.

**Used in Python :**

def haversine\_np(lon1, lat1, lon2, lat2):

# Convert latitude and longitude to radians

lon1, lat1, lon2, lat2 = map(np.radians, [lon1, lat1, lon2, lat2])

# Find the differences

dlon = lon2 - lon1

dlat = lat2 - lat1

# Apply the formula

a = np.sin(dlat/2)\*\*2 + np.cos(lat1) \* np.cos(lat2) \* np.sin(dlon/2)\*\*2

# Calculate the angle (in radians)

c = 2 \* np.arcsin(np.sqrt(a))

# Convert to kilometers

km = 6367 \* c

**return km**

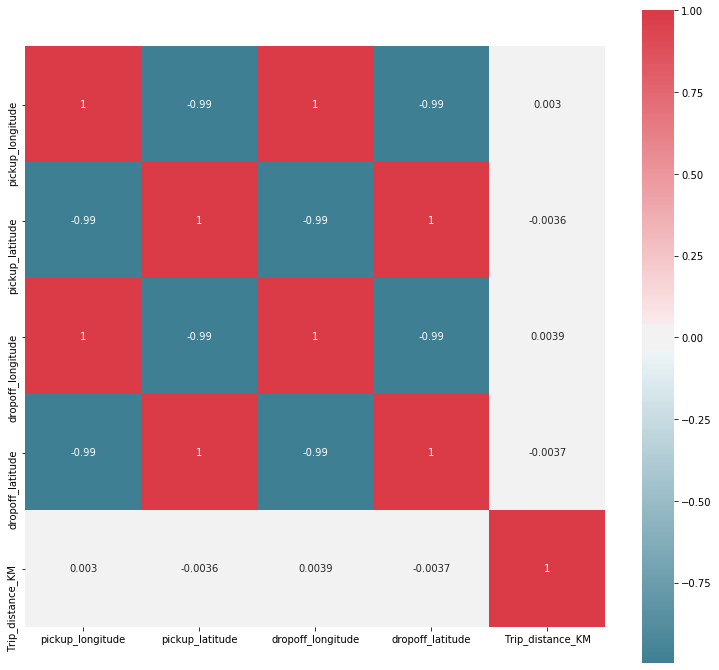
In below **Fig.2.3 pickup\_longitude, pickup\_latitude, dropoff\_longitude, dropoff** **latitude** those all attributes are highly correlated with each other. It means those variables carrysame information to explain the target variable actually in **EDA we can drop those all** **variables after calculation of Trip distance but below with the help of visualization method we analyses it batter**

In correlation analysis, correlation coefficient is calculated between two variables which ranges from -1 to +1. Correlation coefficient approaching -1 or +1 means that both the variables are strongly correlated (negatively & positively correlated respectively). Whilevalue close to 0 implies little or no correlation

Observations & Results

**In python, heat-map of correlation plot was made and is as shown below:**

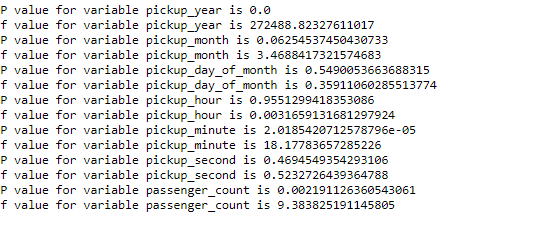




**Fig.2.3 Corrogram to Check Dependency of Continuous Variable**

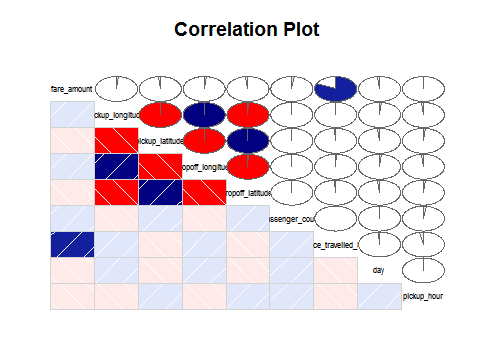
we can easily say pickup\_latitude,pickup\_longitude','dropoff\_longitude','dropoff\_latitude' are highly correalated with each other

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 in below



**Anova Test to Check Dependency of Categorical Variable**

In R, correlation plot was made as shown below:

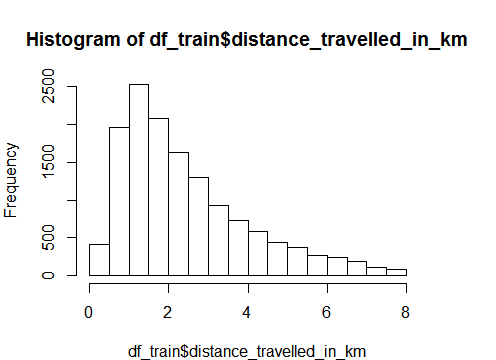


If correlation value is greater than 0.8 or less than -0.8, then it can be safely assumed that the two variables in consideration are highly correlated and one of them may be dropped. But, from correlation matrix in R, all the values were between the range -0.8 & 0.8, which indicates that they are independent of each other. Similar conclusion can be drawn from the heat map & correlation plot, in which low shade of red color which indicates dependency of the variables, was found. Hence, none of the numeric features were eliminated from the train data.

**2.4 Feature Scaling-**

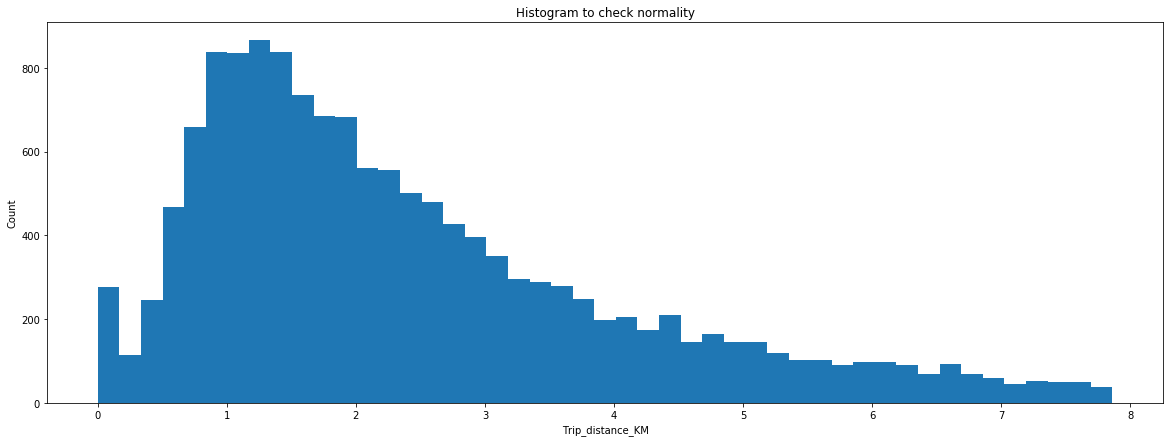
Feature scaling means adjusting data that has different scales into the same range As we know before passing the data to machine learning algorithm our data should be structure in format. To arrange our data into the desire structure feature scaling technique comes into the picture. In this two methods are playing the important role i.e. normalization and standardization. If our data is not normally distributed we go for normalization else for standardization. Normalization is the process of rescaling the features to the range of 0 to 1. Standardization is the process of rescaling data to have a mean of 0 and a standard deviation of 1. This is usually applied to the dataset which is normally distributed

**Histograms were plotted to check the normality of the data and the plots of some of the variables are as follows**

****

As we can see in **Fig.2.4** our data has skewed in nature. So, here we applied Normalization

**Normalization on continuous variable**

****

**Fig.2.4 Histogram to check normality of continuous variable**

From the above histograms, Here we can see when we can draw histogramme plot of ‘**Trip\_distance\_KM**’ variable is right skewed or positively skewed it can be concluded that the data is not normally distributed. So, to scale the data, normalization was applied As we can see in our final train data set there are many variables which carry multiple classes. The variables like hour, minute, second, month, year, day and passenger. If we consider the same for model development, it will assume them as a numeric value so what we can do is we just differentiate every single class into separate variable with the help of **Dummy variable** method.

**Range of the data before normalization = ( -74.03215, 40.81788 )**

**Range of the data after normalization = ( 0, 1)**

**3.Model Development**

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.

**3.1 Linear Regression**

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It’s used to predict values within a continuous range.

Advantages- linear Regression is to designed to purpose of a modelling one or more than one variables and it is greater understanding of variables that can impact its success coming weeks, months and years into the future.

Disadvantages- The disadvantages of linear regression are that it is only efficient for linear data. If the data shows the non-linearity the linear regression can not be useful for such kind of data. It is also not useful for noisy data, the noisy data occurs overfitting models.

**Train-Test split data** -

lm.fit(X\_train,y\_train)

#lets print the intercept

print("LM Intercept : ", lm.intercept\_)

predictions\_LR =lm.predict(X\_test)

RMSE\_LR = np.sqrt(metrics.mean\_squared\_error(y\_test,predictions\_LR))

r2\_LR = metrics.r2\_score(y\_test,predictions\_LR)

MAE\_LR = metrics.mean\_absolute\_error(y\_test,predictions\_LR)

def MAPE(y\_true,y\_pred):

mape = np.mean(np.abs((y\_true-y\_pred)/y\_true))

return mape

MAPE\_LR = MAPE(y\_test,predictions\_LR)

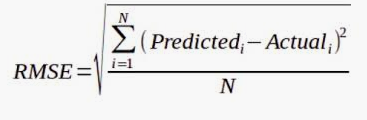
LR\_Results = {'RMSE\_LR':RMSE\_LR,'r2\_LR':r2\_LR,'MAE\_LR':MAE\_LR,'MAPE\_LR':MAPE\_LR}

print(LR\_Results)

**Error and Accuracy-**

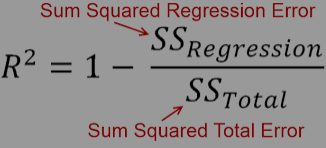
**1.RMSE-(Root Mean Square error)-**

RMSE is the standard deviation of the errors which occur when a prediction is made on a dataset. This is the same as MSE (Mean Squared Error) but the root of the value is considered while determining the accuracy of the model.



**2.Rsquared-**

It is also known as the **coefficient of determination**. This metric gives an indication of how good a model fits a given dataset. It indicates how close the [regression line](https://www.studytonight.com/post/classification-problem-introduction-to-logistic-regression) (i.e the predicted values plotted) is to the actual data values. The **R squared value lies between 0 and 1** where 0 indicates that this model doesn't fit the given data and 1 indicates that the model fits perfectly to the dataset provided.

`

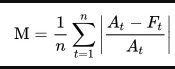
**3.MAE(Mean Absolute error)-**

We know that an error basically is the absolute difference between the actual or true values and the values that are predicted. Absolute difference means that if the result has a negative sign, it is ignored.

Hence, **MAE = True values – Predicted values**

**4.MAPE(Mean Absolute of percentage error)-**

The mean absolute percentage error (**MAPE**) is a statistical measure of how accurate a forecast system is. It measures this accuracy as a percentage, and can be calculated as the average absolute percent error for each time period minus actual values divided by actual values.



Where *At* is the actual value and *Ft* is the forecast value and n is total number of observations.

**Results-**

**In Python**

RMSE=80.468

Rsquared= -259.48

MAE=41.355

MAPE-5.43

**In R-**

RMSE=2.24

Rsquared=0.737

MAE=1.5938

MAPE=18.2

**3.2 Decision Tree**

Decision Tree is a supervised learning predictive model that uses a set of binary rules to calculate the target value/dependent variable.

Decision trees are divided into three main parts this are :

* + - **Root Node :** performs the first split
    - **Terminal Nodes :** that predict the outcome, these are also called leaf nodes
    - **Branches :** arrows connecting nodes, showing the flow from root to other leaves.

Advantages: Reduction in over-fitting and random forest classifier is more accurate than decision trees in most cases.

Disadvantages: Slow real time prediction, difficult to implement, and complex algorithm.

**# Building Decision Tree Model on training Data-**

fit\_DT = DecisionTreeRegressor(max\_depth = 4).fit(X\_train, y\_train)

# Apply model on splitted test data

predictions\_DT = fit\_DT.predict(X\_test)

MAE\_DT = metrics.mean\_absolute\_error(y\_test,predictions\_DT)

RMSE\_DT = np.sqrt(metrics.mean\_squared\_error(y\_test,predictions\_DT))

r2\_DT = metrics.r2\_score(y\_test,predictions\_DT)

MAE\_DT = metrics.mean\_absolute\_error(y\_test,predictions\_DT)

MAPE\_DT = MAPE(y\_test,predictions\_DT)

DT\_Results = {'RMSE\_DT':RMSE\_DT,'r2\_DT':r2\_DT,'MAE\_DT':MAE\_DT,'MAPE\_DT':MAPE\_DT}

print(DT\_Results)

**In Python**

RMSE=162.195

Rsquared= -1057.30

MAE=8.1735

MAPE-0.6434

**In R-**

RMSE=2.49

Rsquared=0.6758

MAE=1.841

MAPE=21.5

**3.3 Random Forest-**

Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.

Advantages: Reduction in over-fitting and random forest classifier is more accurate than decision trees in most cases.

Disadvantages: Slow real time prediction, difficult to implement, and complex algorithm.

**# Building Random Forest Model on training data**

RFModel = RandomForestRegressor(n\_estimators = 200).fit(X\_train, y\_train)

# Apply model on splitting test data

predictions\_RF = RFModel.predict(X\_test)

RMSE\_RF = np.sqrt(metrics.mean\_squared\_error(y\_test,predictions\_RF))

r2\_RF = metrics.r2\_score(y\_test,predictions\_RF)

MAE\_RF = metrics.mean\_absolute\_error(y\_test,predictions\_RF)

MAPE\_RF = MAPE(y\_test,predictions\_RF)

RF\_Results = {'RMSE\_RF':RMSE\_RF,'r2\_RF':r2\_RF,'MAE\_RF':MAE\_RF,'MAPE\_RF':MAPE\_RF}

print(RF\_Results**)**

**In Python**

RMSE=11.514

Rsquared= -4.333

MAE=2.3489

MAPE-0..27001

**In R-**

RMSE=2.210

Rsquared=0.7475

MAE=0.158

MAPE=18.20

**4.Model Selection**

So, now we have developed few models for predicting the target variable, now the next step is to identify which one to choose for deployment. To decide these according to industry standards, we follow several criteria. Few among this are, calculating the accuracy. MAE,RMSE,R2 and MAPE is used in our project .

## Accuracy= number of correct predictions / Total predictions made

**Accuracy=1- MAPE**

Model selection was done on the basis of given evaluation metrics which are summarized in the tabular form as given below:

|  |  |  |  |
| --- | --- | --- | --- |
| **MODEL-R** | **MAPE** | **ACCURACY** | **RMSE** |
| **Linear-regression** | **18.22** | **81.78** | **0.224** |
| **Decision tree** | **21.5** | **78.5** | **0.249** |
| **Random forest** | **18.20** | **81.8** | **0.221** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Model- Python** | **MAPE** | **ACCURACY** | **RMSE** |
| **Linear-regression** | **5.439** | **94.561** | **80.468** |
| **Decision tree** | **0.6434** | **98.56** | **162.19** |
| **Random forest** | **0.2700** | **99.73** | **11.514** |

From the table, it can be seen that Random forest scores better than the other 2 algorithms in all aspects. Both MAPE & RMSE are low for Random forest than Linear regression & Decision tree models. Also, the accuracy is also better than both of them. So, it can be easily concluded that Random forest is the better option among the three models. Further, MAPE value is 0.27 which is a good number as per industry standards

**5.Model Fitting and Conclusion**

After the selection of the best possible model, it was fit to the large test dataset for which the ‘fare\_amount’ was to be predicted. Data pre- processing was also done on the test data for maximum accuracy. No missing observation was found in missing value analysis. Feature scaling was also done because the original train dataset was trained on the scaled data, thus the predicted results would be accurate only if the model fitting is done on the scaled test data. After fitting the model, the results were in the range from 0 to 1 as the whole data was normalized.

**Prediction on TestData. csv**

#predicted values into dataframe

predictions\_RF = pd.DataFrame(predictions\_RF)

**#saving X\_test into directory**

**X\_test.to\_csv("xtest\_final.csv", index = False , header= True)**

df = pd.read\_csv('xtest\_final.csv')

#joining two dataframes

final\_result = pd.concat([df, predictions\_RF], axis=1)

#renaming column name

final\_result.rename(columns={0: 'fare\_amount'},inplace = True)

**#saving result in csv format**

**final\_result.to\_csv('cab\_final\_result\_.csv',header= True , index= False)**

**6.References**

1. Data Cleaning, Model Development and Data Visualization we used. <https://edwisor.com/career-data-scientist>
2. For Visualization using seaborn. <https://www.geeksforgeeks.org/plotting-graph-using-seaborn-python/>
3. To know best time to visit New York City. <https://santorinidave.com/best-time-to-visit-nyc>
4. Haversine algorithm for distance calculation. <https://www.movable-type.co.uk/scripts/latlong.html>

