Cab Fare in Python

1. **INTRODUCTION**

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

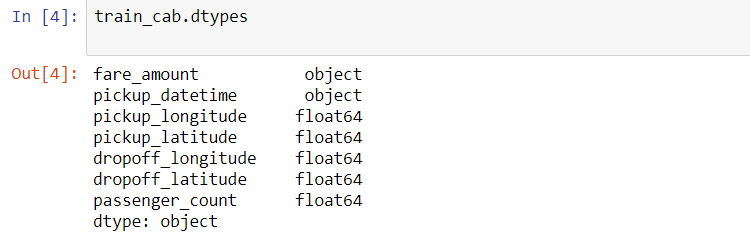
* 1. Data

Our task is to build regression model which will predict the fare of the cab depending on various others attributes.

The data-set contains the following variable :   
 1. fare\_amount – denotes the fare of the cab.  
 2. pickup\_datetime – denotes the time when the cab was   
 taken.  
 3. pickup\_longitude – denotes the location in terms of   
 longitude.  
 4. pickup\_latitude – denotes the location in terms of latitude.

5. dropoff\_longitude- float for longitude coordinate of where   
 the ride ended.  
6. dropoff\_latitude – float for latitude coordinate of where   
 the ride ended.  
7. passenger\_count – indicates the no of passenger in the   
 cab.

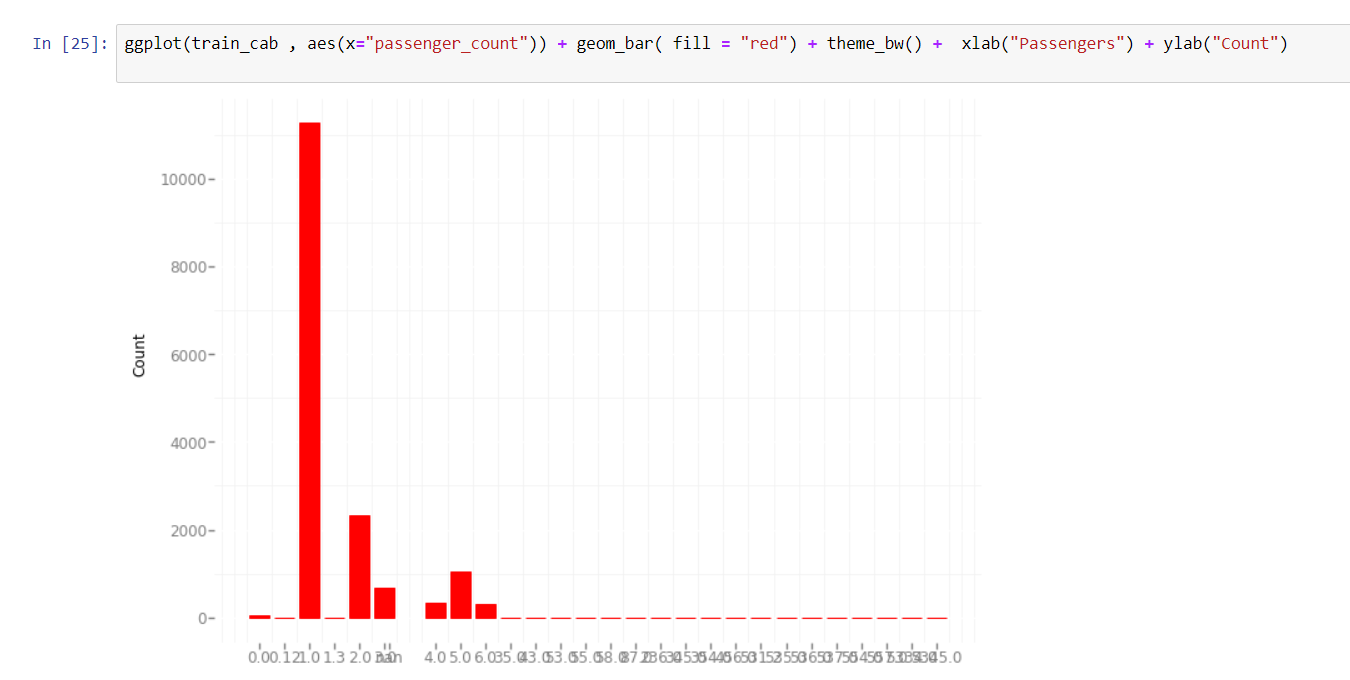
The details of the variables are in the dataset as follows :

  
2 METHODOLOGY

2.1Preprocessing   
 Prior to creating any machine learning models ,   
 it is imperative to have a feel of the data. Pre processing   
 involves data analysis with plots and graphs , recognizing   
 outliers , imputing missing values , feature selection and   
 feature selection etc.

2.1.1 Data Cleaning :

1.Invalid Passenger Count



As we can see , we have numerous invalid passenger\_count   
 that needs removal.

Before venturing into Missing Value Analysis or Outlier Analysis , it is imperative that we get rid of invalid rows i.e. rows   
having invalid attribute column.

First step is to get rid of invalid passenger\_count - Detecting invalid passenger\_count i.e less than 1 or greater than 6 as the underlying assumption is that more than 6 passengers can't enter the car.

There are total 58 rows with passenger\_count equivalent to 0 ; which needs removal.

Next removal of passenger\_count which exceeds 6 : total 20 in number

Total 78 rows are removed and number of rows goes from 16067 to 15989

1. Latitudes and Longitudes

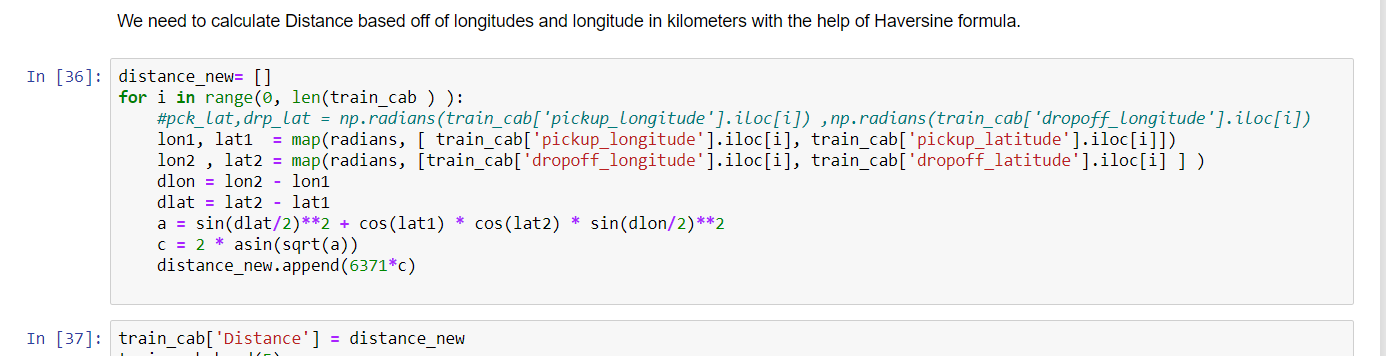
Valid values for latitudes and longitudes lies between -90 to 90 degrees and -180 t0 180 degrees respectively. Here’s a   
snippet of the code :



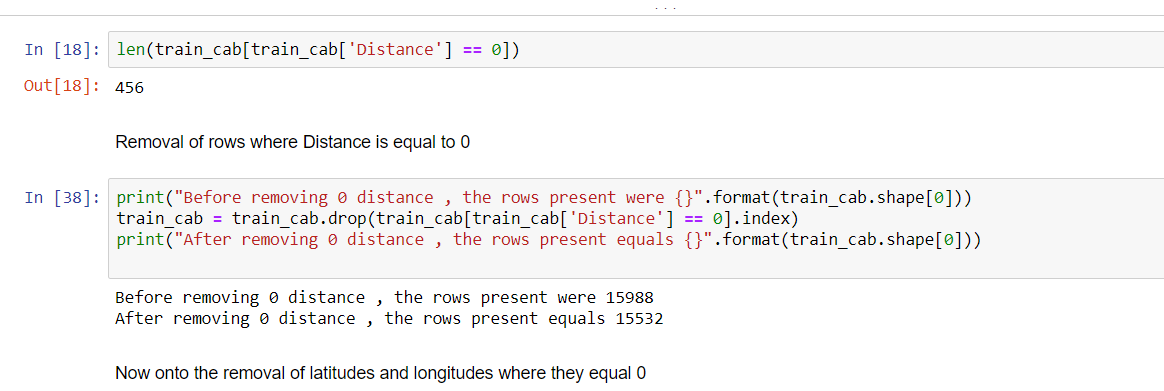
1 row exist with latitude exceeding 90 degrees which is subsequently removed upon its detection

Next is feature engineering Distance from latitude and longitude  
with the help of Haversine’s Formula.

The following line of code is used:



Upon inspecting further , its found that 456 rows exists with Distance valued 0 – which needs removal as 0 Distance route seems impossible. The following snippet depicts the removal of invalid Distance rows:

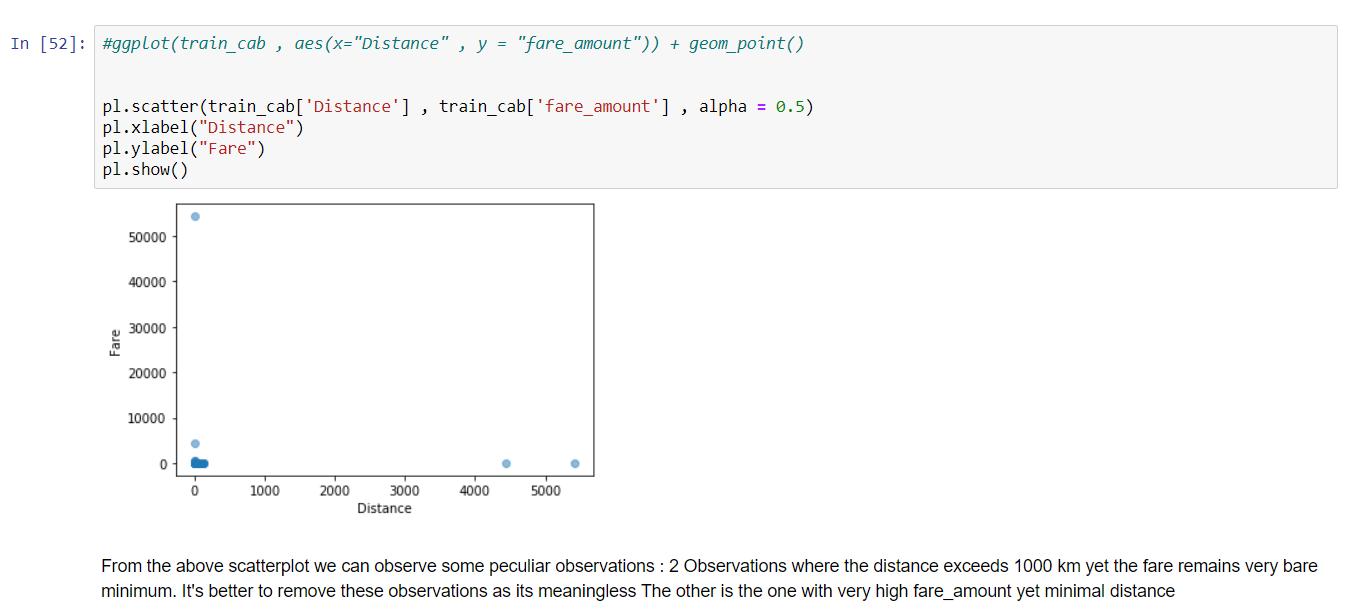


Total number of rows goes down from 15988 to 15532

Also, latitude and longitude valued 0 needs removal as its invalid.

The number of rows decreases further to 15511.

1. Invalid fare\_amount :



#From the above scatterplot we can observe some peculiar observations :

2 Observations where the distance exceeds 1000 km yet the fare remains very bare minimum.Upon further inspection , it is revealed that the passenger\_count is 1.

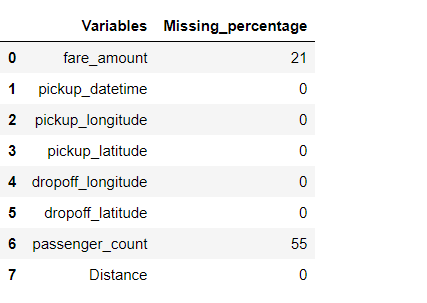
The other is the one with very high fare\_amount yet minimal distance

Thus ,it's better to remove these observations as its meaningless  
Also removal of rows with fare\_amount less than 1 gets removed as well.

The numbers we have left after data cleaning is 15502 i.e. 565 rows gets removed

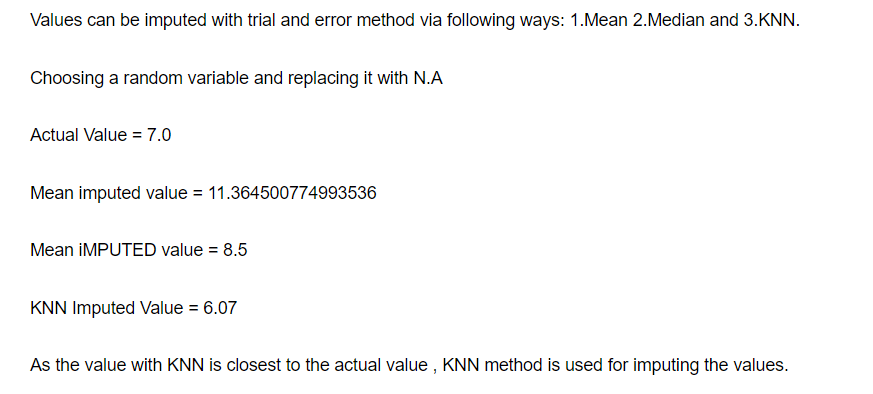
2.2 Missing value Analysis   
 Missing value involves imputing the missing values with   
 three methods : mean , median or KNN Imputation.

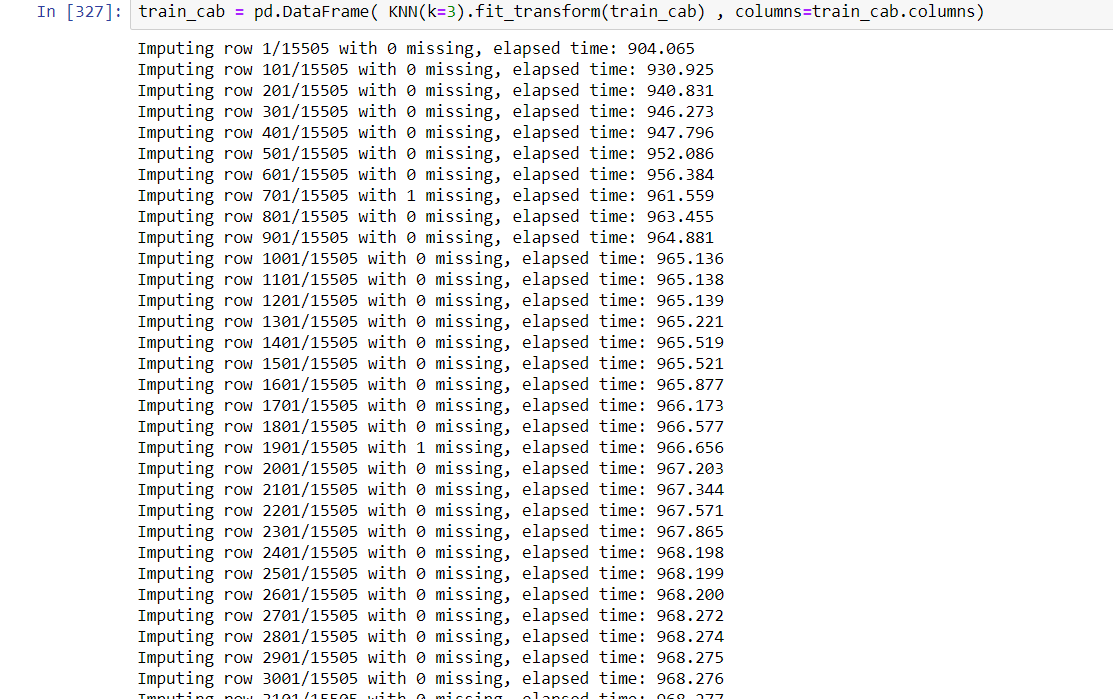
Before imputing missing values , it is important to discover   
 the number of missing values per column.



Also in Python , we will separate the pickup\_datetime and merge it later.

Following happens after Missing Value Analysis :





As the value imputed with KNN is the closest , we’ll select KNN.

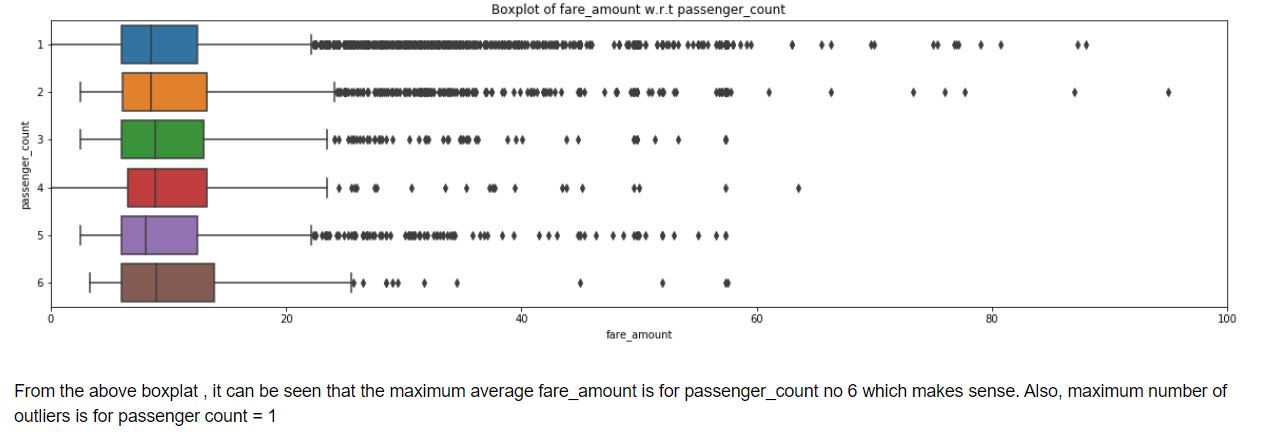
Feature Engineering takes place with pickup\_datetime and various derivatives are derived   
such as date , hour , year , month and day.

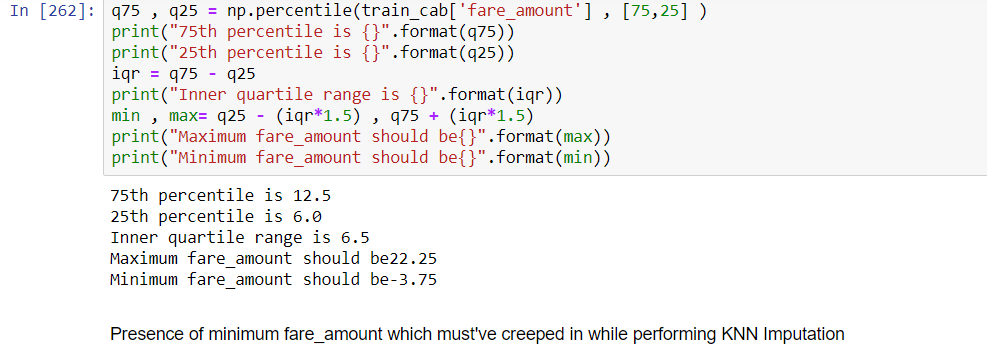
The derivatives are merged with the main dataframe which results in loss of data(around 500 rows)  
.

2.3 Outlier Analysis And Data Visualization

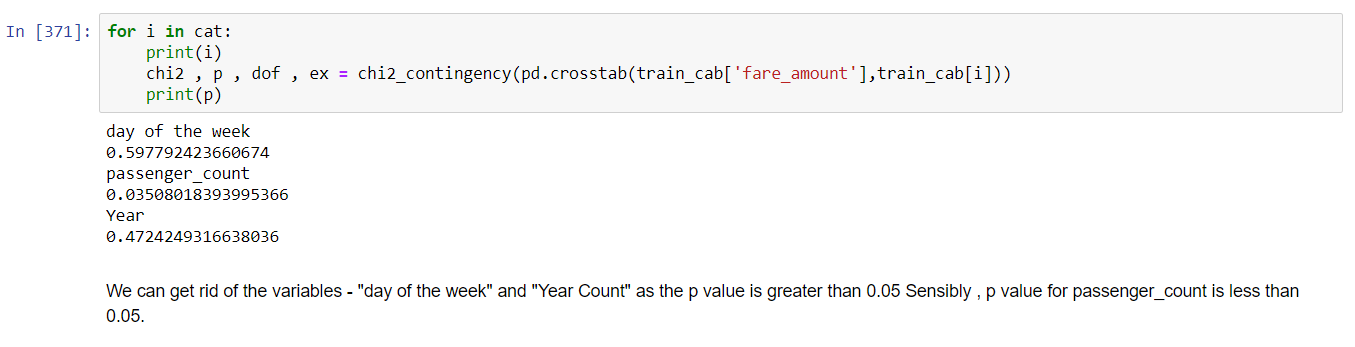
Outlier Analysis is performed to detect the outliers present in   
the data set in order to weed out the unnecessary variables. Graphical methods can be used, in this case Boxplot is used to visualize the presence of outliers.

It becomes imperative to get rid of outliers as their presence can generate ambiguity.



Upper quartile , lower quartile , maximum and minimum is calculated for fare\_amount.  


Outliers are detected and nullified and eventually imputed with KNN.

2.4 CORRELATION ANALYSIS: Correlation analysis is performed with the following result : 

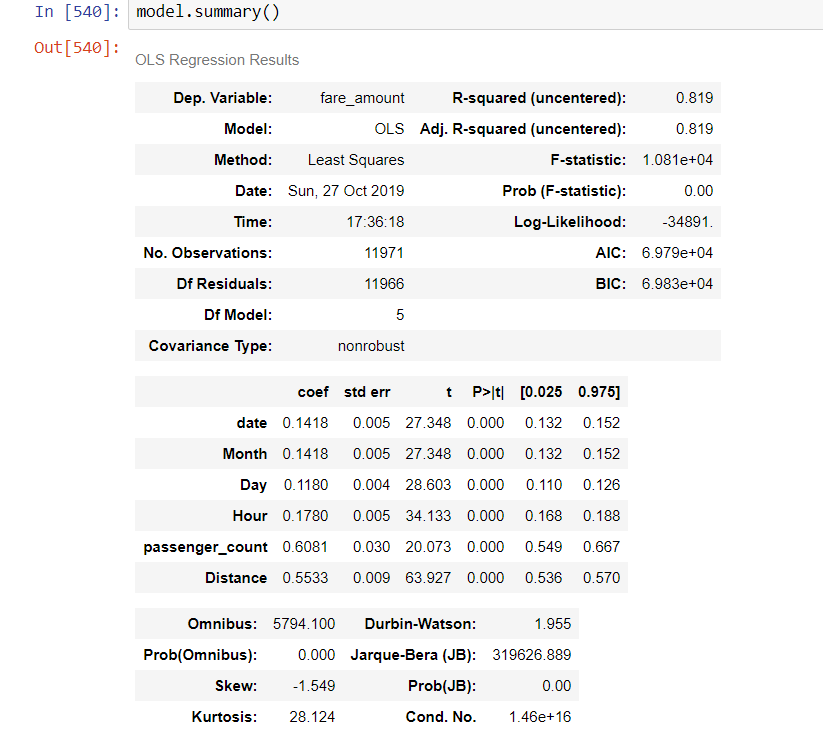
MACHINE LEARNING

Once the pre-processing steps are completed , it is the next step to build   
 model. Since it’s a regression model , we will select from one of the   
 three – Linear Regression , Decision Tree Model or Random Forest   
 Model.

First , we will sample the data into 2 portions : train and test with 80%  
 data going to train and 20 % to test.

Also , before the model is built , the arrangement of column in  
the train and test dataset is done.

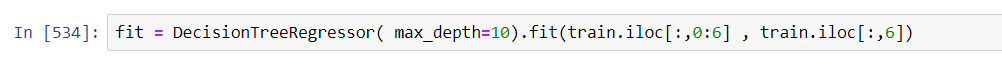
1.Linear Regression : In this case , we will use a Least Square Method.  
 Following is the snippet :



The result seems strange as all the p-values are 0.

We’ll move on to the Decision Tree Model.

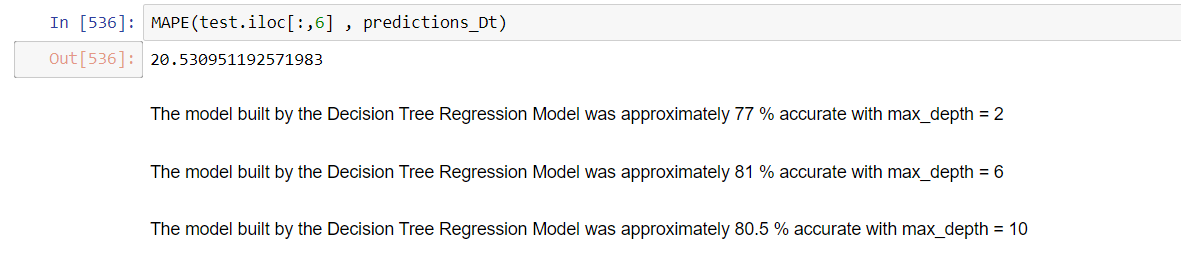
2. Decision Tree Model : Decision Tree Model is used as a regressor to   
 predict the value of fare\_amount. The same sample and testing   
 dataset is used to measure the accuracy of our model.

Here’s the snippet of the code used :   
 

max\_depth is the number of nodes on the tree.  
 The accuracy can measured with the help of MAPE i.e. Mean Absolute   
 Percentage Error.

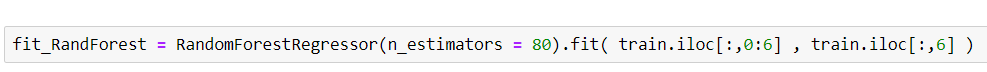
(Formula : abs(predicted value-actual value/actual value)\*100

Following results were obtained :



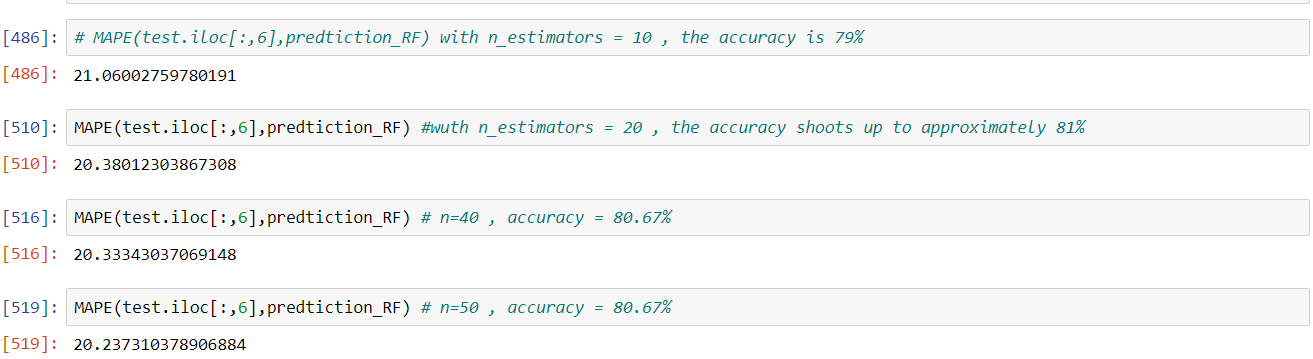
3. Random Forest: Random Forest is an extension of Decision Tree.  
 Here multiple decision trees are combined in order to deliver   
 a model.

Here’s a snippet :



Here the n\_estimators is an important parameter which represents the  
 number of trees.

Following result was obtained with the MAPE function :



Conclusion : Decision Tree delivered the highest accuracy of all the model.