Cab Rent Prediction

Prity Vijay

1 August 2019

**Chapter 1**

**Introduction**

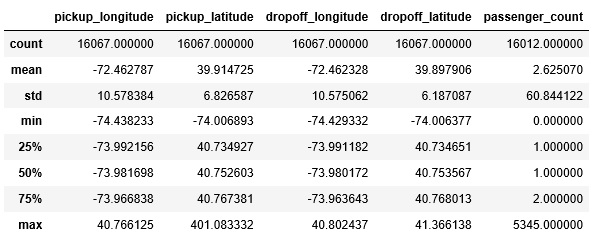
* 1. **Problem Statement**

An immense growth of cab service has been observed in a past few year .cab ride is becoming part of common people life. Many of us take cabs for travelling purpose but something strange we see every time during the ride of cab and that is in its fare. It’s quite known that cab price depends upon the distance we travel. But does every time price of a cab is depended upon distance? No, apart from distance there are many other factors which is directly proportional to the fare of a cab, some of them are availability of a cab, timing of booking cab, day of travelling, etc. Cabs are always expensive when there is less availability of cabs and during peak time i.e, a time when most of the people are in travelling mode. At late night, early morning also cab fare price varies. Here from the historical data by apply analytics I will develop a model for cab fare prediction. Basically, I will design a system that predicts the fare amount for a cab ride in the city.

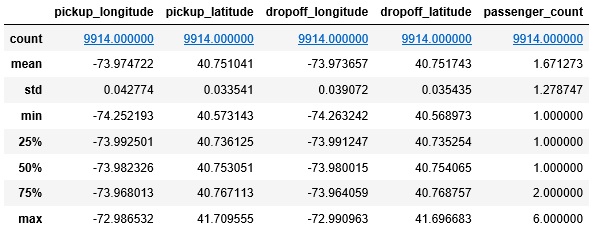
* 1. **Data**

The problem discussed above is a regression problem. Objective is to predict cab fare, based on co-ordinates and time of travel. Train data and test data are two sets of data, through which we will try to predict cab fare. Train data is consists of fare amount, pick up and drop off co-ordinates along with passenger count where as test data, as name suggest will be use for testing of predicted fare, all the column are just like train data, only the difference is fare amount is missing in test data which we have to predict. Given below is a glimpse of sample data set that we are using to predict cab fare prediction.

**Table 1.1**: Cab Prediction Data Summary (Train)



**Table 1.2**: Cab Prediction Data Summary (Test)



As we can see above table 1.1 1nd 1.2 is showing summary of train and test set.

Let us have a name of all the columns individually.’ 'fare\_amount’ is a dependant variable for which we have to develop a model for prediction of cab fare.

**Input:**

print(data\_train.columns)

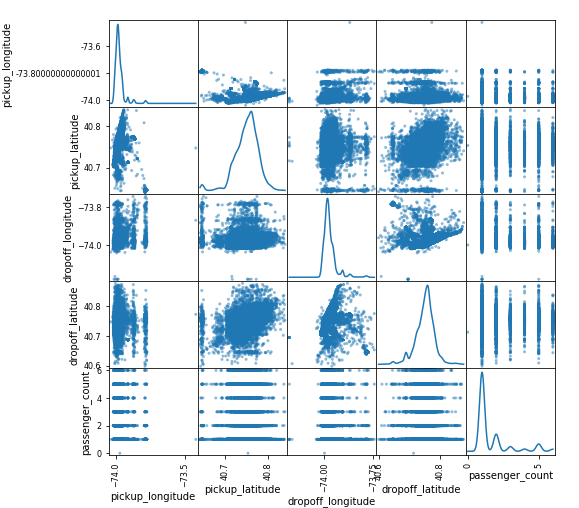
Index(['fare\_amount', 'pickup\_datetime', 'pickup\_longitude', 'pickup\_latitude', 'dropoff\_longitude', 'dropoff\_latitude', 'passenger\_count'], dtype='object')

print(data\_test.columns)

Index(['pickup\_datetime', 'pickup\_longitude', 'pickup\_latitude', 'dropoff\_longitude', 'dropoff\_latitude', 'passenger\_count'], dtype='object')

* 1. **Column Description:**
* pickup\_datetime - timestamp value indicating when the cab ride started.
* pickup\_longitude - float for longitude coordinate of where the cab rides started.
* pickup\_latitude - float for latitude coordinate of where the cab rides started.
* dropoff\_longitude - float for longitude coordinate of where the cab rides ended.
* dropoff\_latitude - float for latitude coordinate of where the cab rides ended.
* passenger\_count - an integer indicating the number of passengers in the cab ride.

Let us visualize all continuous variable of a sample dataset in Fig. 1.1 In a next chapter we will further analyze the data in more detail.



**Figure 1.1 Scatter matrix of continuous variable of a Cab Prediction Dataset (Train)**

**Python Code:**

pd.scatter\_matrix(data.loc[:,col],figsize=(8,8),diagonal='kde')

**Chapter 2**

**Data Analysis**

Data Analysis is a process of inspecting, cleaning, transforming and modelling data with the goal of discovering useful information, suggesting conclusions and supporting decision-making. For any data science project, data is like a fuel of an engine therefore knowing the data is first and foremost task which should be done very efficiently and proficiently. Therefore we will start with knowing our data and then we more further to other stages which is as follow:

1. Emphasize for understand the problem and data
2. Data exploration / data cleaning
3. Feature engineering / feature selection
4. Model evaluation and selection
5. Model optimization
6. Interpretation of results and predictions

First thing to perform whenever we deal with any kind of data is know the shape, data type and dimension of a data.

**Input:**

data\_train=pd.read\_csv("train\_cab.CSV",sep=',')

print("Training Data Shape :", data\_train.shape)

data\_test=pd.read\_csv("test.CSV",sep=',')

print("Test Data Shape :", data\_test.shape)

**Output :**

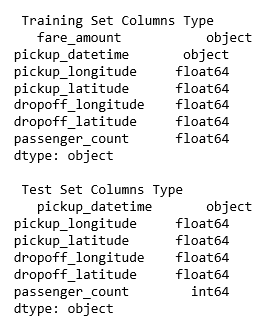
**Training Data Shape:** (16067, 7)

**Test Data Shape:** (9914, 6)

**Input:**

print("\n Training Set Columns Type\n ",data\_train.dtypes)

print("\n Test Set Columns Type\n ",data\_test.dtypes)



**Figure 2.1 : Column Data Type ( Train & Test)**

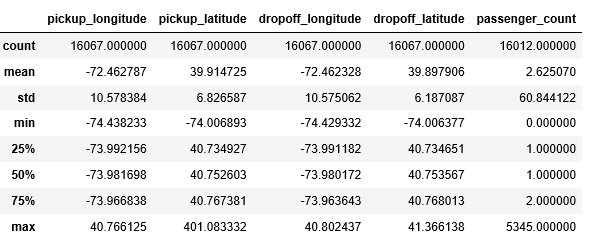
Next, let us check descriptive statistics of train and test set, which is very important for knowing the distribution of the data.

**Input:**

# Descriptive summary

data\_train.describe()

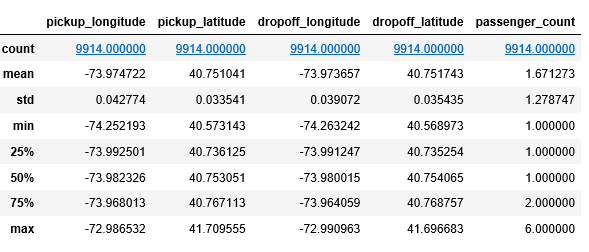
**Output:**



**Input:**

# Descriptive summary

data\_test.describe()

**Output:** 

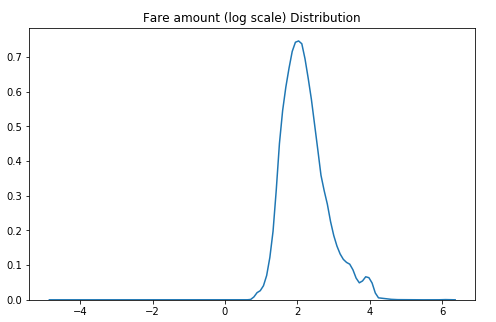
**Distibution of Dependant Variable:**

As fare\_anount is not normally distributed we are taking log scale for all the values, and disbution of fare\_amount is below:

**Code:**

plt.figure(figsize=(8,5))

sns.kdeplot(np.log(data\_train['fare\_amount'].values)).set\_title("Fare amount (log scale) Distribution")

****

**Figure 2.2 : Log Distribution of Fare Amount**

From descriptive analysis it is clear that data set is consists of lots of NaN, Zero, columns like passenger\_count is highly deviated from the mean, co-ordinates columns

| ( pickup\_longitude | pickup\_latitude | dropoff\_longitude | dropoff\_latitude ) |  |
| --- | --- | --- | --- | --- |

is consists of the value having no meaning, therefore these values should be identified and should be treated. In the next section we will see how step should be taken to clean the data.

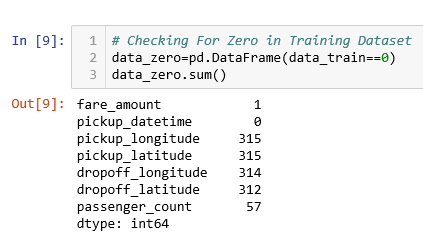
**Chapter 3**

**Data Pre-Processing**

**3.1 Data Pre Processing:**

Data is growing day-by-day. Extracting knowledge from it is a real threat otherwise it is nothing but garbage. Data, when loaded into the database from various source, emerge as a messy dataset. These datasets are of no use because extracting valuable information from it is very tough. Therefore data pre-processing is the first and mandatory step for any data scientist before mining [2]. According to Lour, data scientists spend 50% - 80% of their valuable time and effort in data collection and preparing disorderly digital data, before it can be explored for useful nuggets.

In descriptive analysis we have seen columns contain zero values, which does not play any role during fare prediction therefore, we will identify such columns and replace it with null values. Those null values will be further treated during missing value analysis.



**Table 3.1 : Number of Missing Value (Train)**

**3.1.1 Missing Value Analysis:**

In the first step of data pre-processing we look into the dataset to collect missing values in different variables. Missing value should be recognized and replace (mean, median or knn imputation) or drop it because in the presence of missing values, a machine learning algorithm doesn’t perform well. Cab Prediction dataset contain many missing values therefore lets identify it and treat it accordingly.

In the table 3.1, we can see missing values in almost all columns except fare\_amount and pickup\_datetime. We will replace those with mean or median depending which will suitable for a particular column.

**3.1.2 Outlier Detection**

Outliers are extreme values in an observation of dataset that falls beyond the range from other records that are more or less similar to each other [1]. Outliers can occur in a dataset due to some experiment error variation in measurement etc. Outliers in a dataset should be taken care of before application of machine learning algorithm to the dataset in order to get an appropriate result because machine learning algorithm is sensitive towards the distribution of the dataset. A massive amount of outliers in a dataset can cause poor and inaccurate models with longer training times by misguiding machine learning algorithms. In a dataset, outliers can be part of records during the collection, processing or analyzing records. Machine learning algorithm like linear and logistic regression is very much affected in its training process with the presence of outliers. Human error, instruments error, experimental errors, data processing errors, etc. are some of the causes of outliers. Cab fare prediction dataset is consists of 7 columns. Here we will try to detect outliers.

**fare\_amount:** It is a dependant variable and therefore very important. But we can see the value range from 0 to 54343. As fare cannot be zero, we drop the values whose value less than 0 or more that 99 percentile of a data.

**passenger\_count:** Passenger counts are showing very high deviation where maximum count of passenger is 5345. But in test data passenger counts vary from 1 to 6. We observed 21 such deviated variables, we detect those value and replace it with mean.

pickup\_longitude, pickup\_latitude, dropoff\_longitude, dropoff\_latitude

Some of the co-ordinates values seem to be out of range i.e. those values are not possible for pick up and drop off location. Therefore we compare co-ordinates of test data with that of train data where Minimum Longitude of test data is -74.263242, Maximum Longitude of test data is -72.986532, Minimum Latitude of test data is 40.568973 and Maximum Latitude of test data 41.709555. We are using these values as a boundary and below or above of these are discarded from the dataset. We got 28 such values which are out of range

**3.1.3 Feature Engineering:**

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. If feature engineering is done correctly, it increases the predictive power of machine learning algorithms by creating features from raw data that help facilitate the machine learning process. Feature Engineering is an art of deriving new columns from given column in a dataset.

Pick up date is provided in a dataset which is very important. But as we know we cannot use date directly for model building. We extract year, month, day, weekday, hour from it with the help of function extract\_date. This function is applied to train as well as test data.

def extract\_date(data) :

data['pickup\_datetime'] = pd.to\_datetime(data['pickup\_datetime'], format= "%Y-%m-%d %H:%M:%S.%f")

data["year"] = pd.DatetimeIndex(data["pickup\_datetime"]).year

data["month"] = pd.DatetimeIndex(data["pickup\_datetime"]).month

data["weekday"] = data["pickup\_datetime"].dt.weekday\_name

data["day"] = data["pickup\_datetime"].dt.day

data["hour"]= data["pickup\_datetime"].dt.hour

return data

Distance is another variable which is derived from the pickup and drop off longitude and latitude.

# Lets Calcute Distance From Cordinate

def distance(data):

pickup\_latitude = np.radians(data["pickup\_latitude"])

pickup\_longitude = np.radians(data["pickup\_longitude"])

dropoff\_latitude = np.radians(data["dropoff\_latitude"])

dropoff\_longitude = np.radians(data["dropoff\_longitude"])

dlon=dropoff\_longitude-pickup\_longitude

dlat=dropoff\_latitude-pickup\_latitude

a = (np.sin(dlat/2))\*\*2 + p.cos(pickup\_latitude)\*np.cos(dropoff\_latitude)\*(np.sin(dlon/2))\*\*2

c = 2 \* np.arctan2( np.sqrt(a), np.sqrt(1-a) )

R = 6373.0

d = R \* c

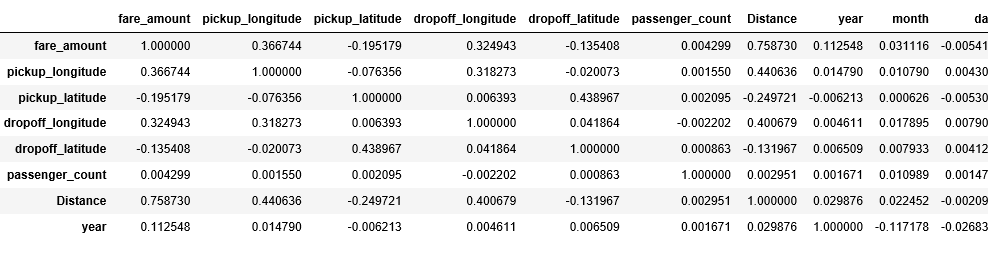
data["Distance"] = d

return data

3.1.4 Feature selection

It is not realistic to perform a complex data analytics on a massive dataset. Data reduction is a technique to achieve a smaller dataset. Prior to any machine learning modelling we should measure the significance of each independent variable of our dataset. It can happen that many variables are not performing important role during model building. These types of variable should be recognized and dropped before any predictive analysis. Let us look in our dataset for insignificant independent variable.

As we can see that apart from distance no other variable is correlated to fare\_amoun t(dependant variable), we will visualize the data from different aspects and try to find out the relation. We will drop pickup\_date before modelling.



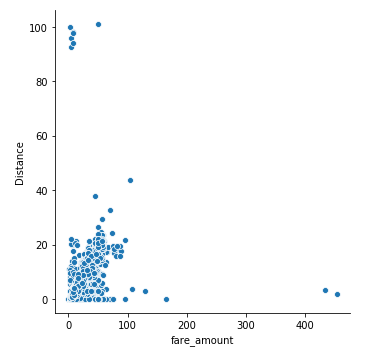
**Figure 3.2 : Table After Data Wrangling**

After entire step for data cleaning we are left with the following columns, in the next section we will make some visualization and try to discover some relationship among the variables.

**Chapter 4**

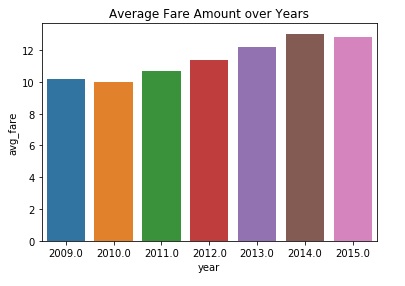
**Visualization**

Let us first check how fare is related with distance

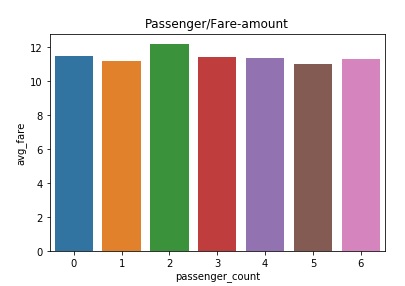
****

**Figure 4.1 : Distance/ Fare scatter plot**

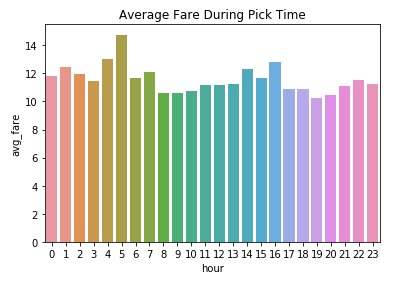
Next we will check how fare changes over the period of time.

****

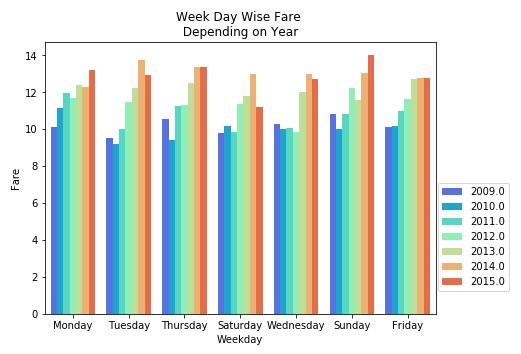
**Figure 4.2 : Distribution of Fare Amount Changes with year**

****

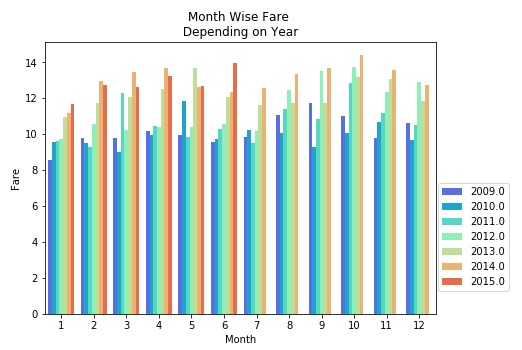
**Figure 4.3 : Distribution of Fare Amount Changes with Number f Passenger**



**Figure 4.4: Distribution of Fare Amount Changes with Pick Time**

****

**Figure 4.6: Distribution of Fare Amount Changes/ Weekdays over the period of Year**

****

**Figure 4.7: Distribution of Fare Amount Changes/ Month over the period of Year**

**Conclusion:**

Fare and distance is showing positive relationship.

Fare of a cab is increases over the years.

No such relationship of fare with passenger count is identified. Although fare is maximum when passenger count is 2.

During pick time fare is high i.e, early morning, office timing etc.

Sunday and Monday cabs are demanded more over period of time

No such pattern of fare can be identified with the month, but we can see slightly high graph during feb and nov.

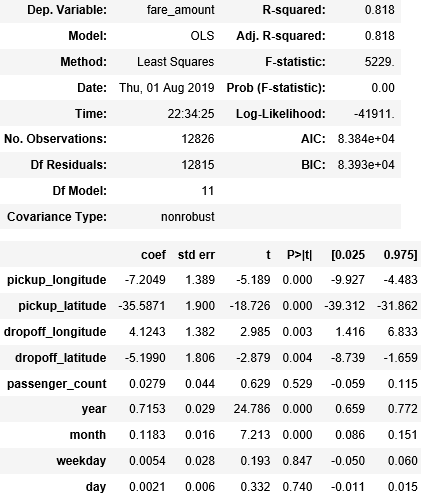
**Chapter 5**

**Data Modeling**

**5.1.1 Model Selection**

From the previous step of analysis it is clear that fare\_amount which is our dependant variable is a continuous variable and therefore this is a regression problem. To solve regression problem we have to select machine learning regression algorithm. To predict cab fare here we are using multiple linear regression, decision tree and random forest and try to discover best and accurate result. We should always try with simpler algorithm and check its result. Let us first try decision tree on our dataset and check the result.

**5.1.2 Multiple Linear Regression**

****

*Adjusted R-squared* value, we can see 81% of the data can be explain by using our multiple linear regression model. Let us evaluate the model by using MAPE. This is good but error rate is very high, some of the predictor we can see that there is some insignificant variable let us remove them and check the result.

**Model Evaluation**

After building a model we have to evaluate it to check the performance of a model. There are numerous error matrixes for the evaluation of regression model like MAE, MAPE and MSE. We are using MAPE here because it is easy to understand as it gives a percentage of error during prediction.

**#Calculate MAPE**

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

mape = np.mean(np.abs((y\_true - y\_pred) / y\_true))\*100

return mape

**# Result of Linear Regression**

# Accuracy=69.05

# Error 30.94

As we see error rate is very high. We will drop the model by rejecting null hypothesis. Next we will try with other model of MLR by dropping insignificant independent variable.

**Code:**

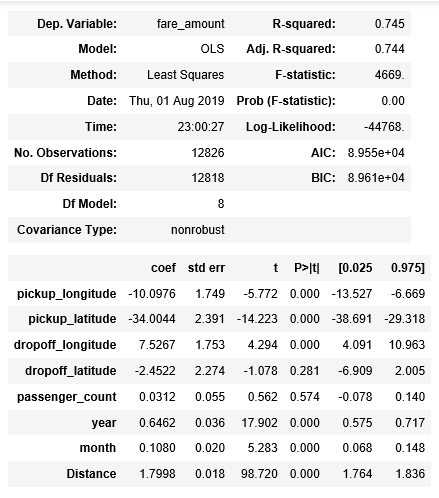
# After Removing In-Significant Variables

data\_train\_dup=data\_train.copy()

data\_test\_dup=data\_test.copy()

data\_train\_dup=data\_train\_dup.drop(['weekday','day','hour'],axis=1)

data\_test\_dup=data\_test\_dup.drop(['weekday','day','hour'],axis=1)



Here also result is not satisfactory. Although adj. R-squared can explain 74% of the data yet error rate is too high. Same error matric (MAPE) is performance evaluation.

# Result of Linear Regression by removing insignificant variable

# Accuracy=68.90

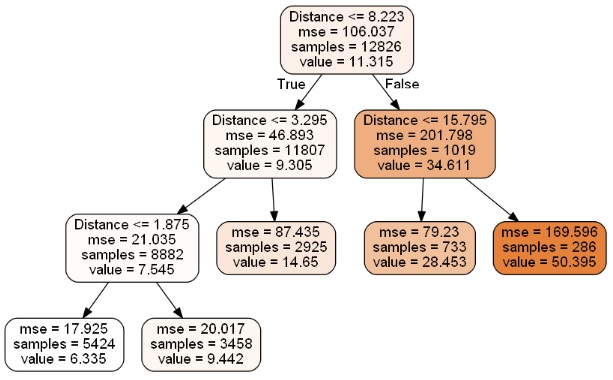
# Error=31.09

We will try with some advance model like Decision tree and random forest.

**5.1 Decision Tree**

By using the default values of decision tree doesn’t provide satisfactory result and therefore we tried pruning trees. Our decision tree was built based on following criteria.

DecisionTreeRegressor(criterion='mse', max\_depth=5, max\_features=None, max\_leaf\_nodes=5, min\_impurity\_decrease=0.0, min\_impurity\_split=None, min\_samples\_leaf=10, min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0, presort=False, random\_state=None, splitter='best')



**Fig 5.1: Decision Tree Regression Plot**

**5.1.1 Model Evaluation**

After building a model we have to evaluate it to check the performance of a model. There are numerous error matrixes for the evaluation of regression model like MAE, MAPE and MSE. We are using MAPE here because it is easy to understand as it gives a percentage of error during prediction.

**MAPE:**

Mean absolute percentage error (MAPE) is used here for model evaluation.

Error: 27.48

Accuracy: 72.51

**5.2 Random Forest**

As decision tree doesn’t provide satisfactory result and we look farward for random tree

rf = RandomForestRegressor(n\_estimators = 100, random\_state = 883,n\_jobs=-1)

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

rf\_pred= rf.predict(X\_test)

**5.1.1 Model Evaluation**

After building a model we have to evaluate it to check the performance of a model. There are numerous error matrixes for the evaluation of regression model like MAE, MAPE and MSE. We are using MAPE here because it is easy to understand as it gives a percentage of error during prediction.

**MAPE:**

Mean absolute percentage error (MAPE) is used here for model evaluation.

# Error 19.54

# Accuracy 80.45

**Conclusion:**

As observed random forest performance is far better than decision tree and multiple linear regressions. Therefore we will drop 1st three models and accept the last model constructed using random forest.

**References:**

1. Jingke Xi, Outlier detection algorithm in data mining, 2008 Second International Symposium on Intelligent Information Technology Application, December 2008.
2. Prity Vijay, Bright Keswani,ICCS 2018, Enhanced Approach to Attain Competent Big Data Pre-Processing