CAB FARE PREDICTION

This project is to predict cab fare after building a Machine Learning model on the train data. Here I received around 16k in train data & 9.9k in test data,which consist of 7 variables.& is a Regression problem

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

EXxplotary data analysis

R Code

```
cabdf$fare_amount=as.numeric(as.character(cabdf$fare_amount))

cabdf$pickup_datetime=as.POSIXct(strptime(cabdf$pickup_datetime,"%Y-%m-%d %H:%M:%S"))


#Extracting years,month & weedays number from time

cabdf$year=as.numeric(format(cabdf$pickup_datetime,"%Y"))

cabdf$month=as.numeric(format(cabdf$pickup_datetime,"%m"))

cabdf$wday=strftime(cabdf$pickup_datetime,"%u"))

cabdf$wday= as.numeric(as.character(cabdf$wday))
```

In the above code fare fare_amount is converted to numeric & pickup_datetime to POSIXct

Python Code

#Extracting years,month & weedays number from time

```
cabdf$year=as.numeric(format(cabdf$pickup_datetime,"%Y"))

cabdf$month=as.numeric(format(cabdf$pickup_datetime,"%m"))


#1-Monday to 7-Sunday

cabdf$wday=strftime(cabdf$pickup_datetime,"%u")


#converting character to numeric

cabdf$wday= as.numeric(as.character(cabdf$wday))
```

Missing Value: These are the values which are missing from dataset.

Cause of missing value:

-Human ERROR

-Refuse to answer while surveying

-Optional

After getting the missing value we can ignore or impute missing value.

3 types of method to impute missing value

1) Central Tendency

-Mean

-Median

- Mode

2)KNN Imputation

3)Prediction Method.

We have to voluntarily replace a data with NA to check which method is working accurartely.

OUTLIERS

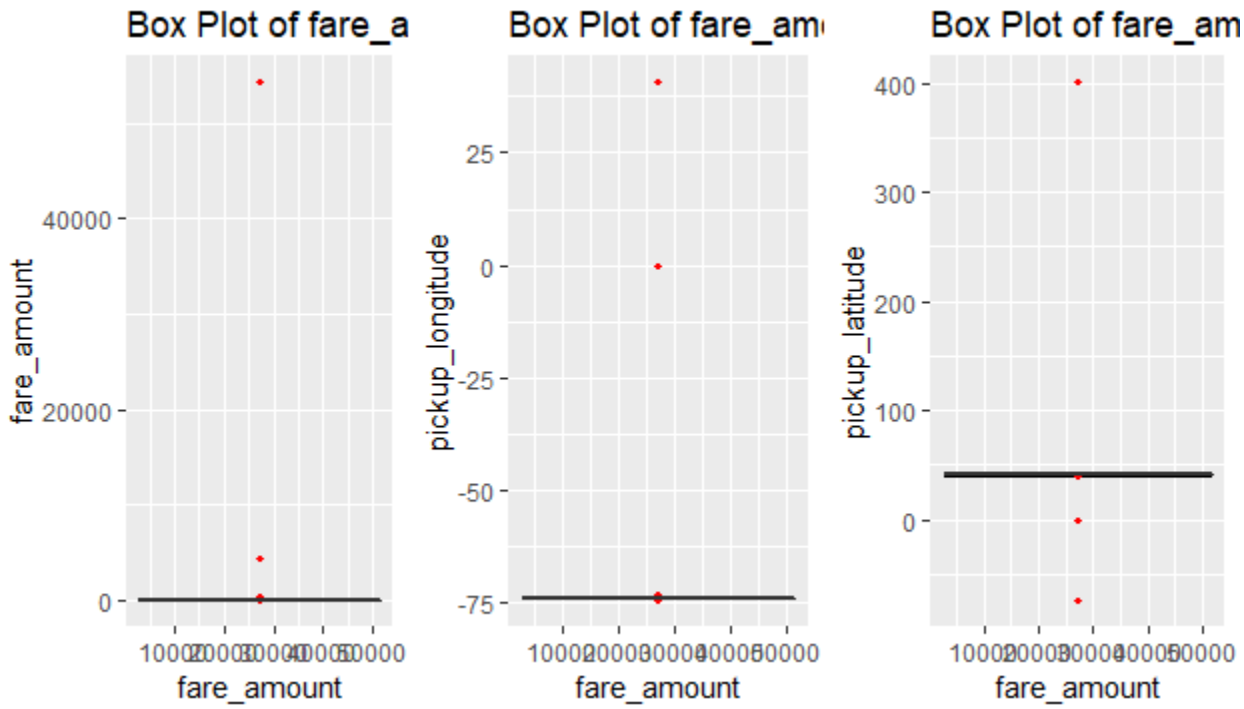
Outliers are the data which fall away from dataset.

Causes of outliers

-Poor Data quality/contamination

-Lower quality measurement,malfunctioning equipatient etc.

Box Plot



In the above boxplot pickup_longitude & pickup_latitude have outliers.

R code

```
for(i in cnames)
{
 print(i)
 outliers=cabdf[,i][cabdf[,i] %in% boxplot.stats(cabdf[,i])$out]
 cabdf=cabdf[which(!cabdf[,i] %in% outliers),]
}
```

Python Code

```
for i in cnames:
   q75,q25=np.percentile(cabdf.loc[:,i],[75,25])
   iqr=q75-q25
   min=q25-(iqr*1.5)
   max=q75+(iqr*1.5)
```

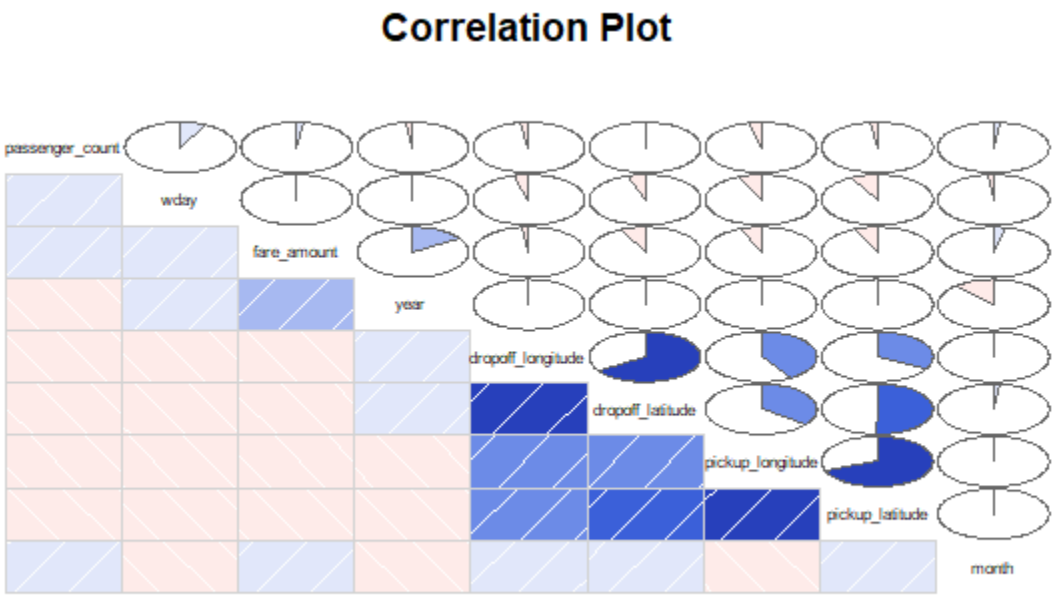
```
cabdf=cabdf.drop(cabdf[cabdf.loc[:,i]<min].index)

cabdf=cabdf.drop(cabdf[cabdf.loc[:,i]>max].index)
```

After deleting the rows we will still get some outliers,so we will considered that as Non Extreme outliers.

This will not affect the model as much.

#FEATURE SELECTION

Also known as variable selection means selecting a subset of relevant feature for use in the model selection.



In the above correlation plot we can clearly see almost correlation between any two variable is nearly 0 & none of any variable is highly negatively or positively correlated. So, we don't required to drop any variable as each variable will contribute to decide the target variable.

SAMPLING

I have divided the training data into train data (80%) to build the model & test data(20%) to test the data before deploying to actual test data.

R code

```
cabdf2=cabdf

train_index=sample(1:nrow(cabdf2),0.8*nrow(cabdf2))

train = cabdf2[train_index,]

test = cabdf2[-train_index,]
```

Python

```
train, test = train_test_split(cabdf, test_size=0.2)
```

ERROR METRICS

Their mainly three error metrics we use widely in regression model.

-**MAE(Mean Absolute Error)**

$$\mathrm{MAE} = 1/n \sum_{i=1}^{n} |fi - yi|$$

Where f is actual value & y is observed value ,n is no. of observation.

-**MAPE(Mean Absolute Percentage Error)**

$$\mathrm{MAPE} = 1/n \sum_{i=1}^{n} \left|\frac{At - Ft}{At}\right|$$

Where A is actual value & F is predicted value

-**RMSE(Root Mean Squared Error)**

$$RMSD = \frac{\sqrt{\sum_{i=1}^{n} (yi' - yi)\text{^}2}}{n}$$

WHY RMSE?

-Squaring the error which will give more accurate result.

-RMSE is used in time based measure(Time series data).As our car fare data is time series means the target variable depends on the pickup_datetime as the average fare will increase per year.

-RMSE is better for showing bigger deviation

LINEAR REGRESSION

It is a statistical Model,while in machine learning algorithm we save the pattern from historical data,where as in statistic model,it will save numbers in form of coefficients

Linear regression is measured by two parameters

- R squared(It is the proportion of variance in the dependent variable which can be explained by the independent variable.

  Adjusted R squared(Adjustment of the RSquared that penalizes the addition of extraneous predictors to the model.

DECISION TREE

-A predictive model based on a branching series of boolean test
-can be used for classification & regression

2 popular decision tree algorithm'

C5.0,CART

WHY RANDOM FOREST

Random Forest is ensemble which consist many number of decision trees, if the data is big we will use Random forest, single decision tree will not able to handle it.

Random forest use bagging method where once an error occurred in a decision tree,that error is fed to next decision tree to improve accuracy .So, this improve the correct prediction & decrease the error rate.

Rcode

RMSE LR	RMSE DT	RMSE RF
3.67	3.21	2.33

Python

RMSE LR	RMSE DT	RMSE RF
3.66	3.51	2.06

As RMSE of Random Forest is giving less error rate we will lock Random Forest to train the data.