**Cab Fare Prediction Report**

We have two CSV files given train\_cab.csv and test.csv

Submitted the python file and R code in separate files

R code can be run in Rstudio and python code HTML format has also been submitted

Using the Jupyter notebook to import the data in python

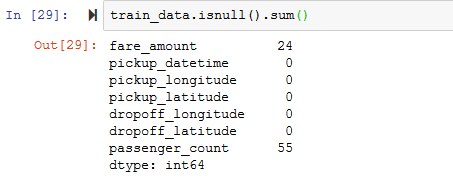
Importing the required libraries

Setting the current working directory using chdir command.

Reading the CSV files using read\_csv command into a pandas dataframe.

Considering these values as train\_data and test\_data

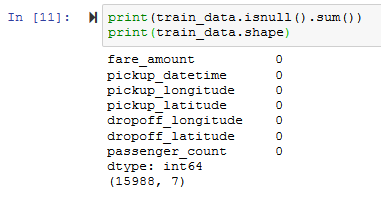
Checking number of null values in all columns



Removing nulls as there are only 55 null out of 16067 records which is aproximately 0.3 % of the whole data

train\_data.dropna(how='any',inplace=True)

We can observe in the below screenshot, the number of records has been dereased

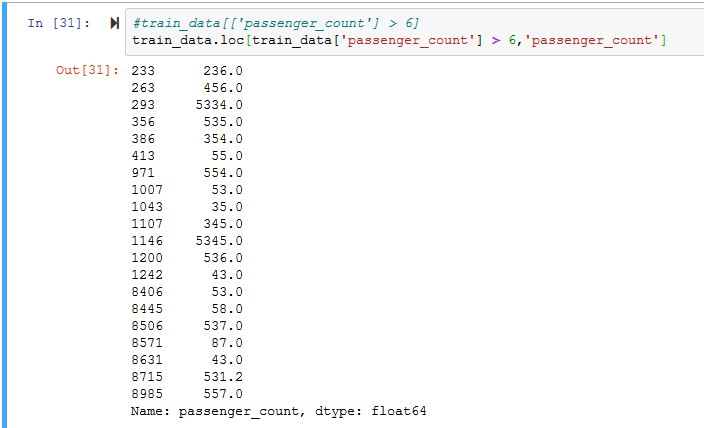


Removing ' - ' symbol from fare\_amount column as its returning error when converting to numeric as shown in the below screenshot



train\_data['fare\_amount']= train\_data['fare\_amount'].apply(pd.to\_numeric)

Number of passengers in a cab can be few like 6 or 7 as expected, but we have outliers as observed in the below screenshot



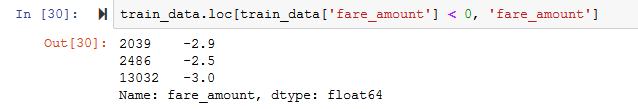
Few columns have negative fare amount

Calculating outliers boxplot

Quartile range feature scaling

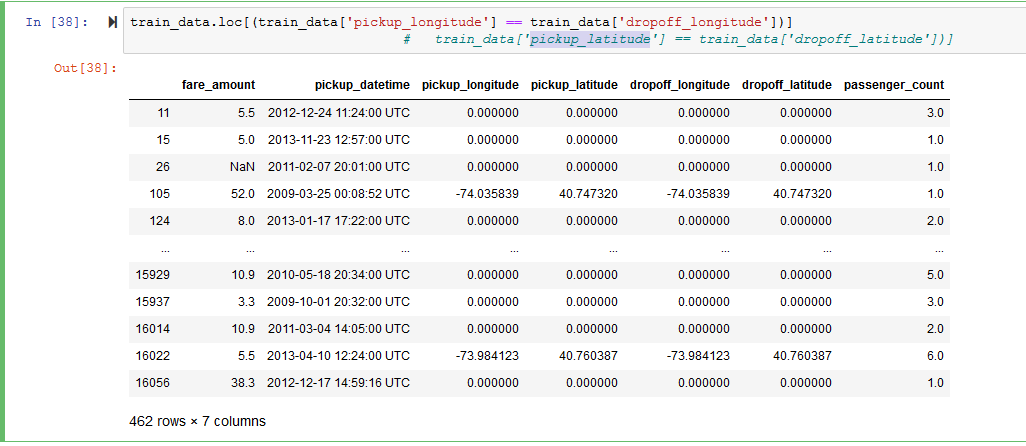
Corelation

Zscore



# Eliminating rows which has same pickup and drop latitude

We can see in the below screenshot the for most of the rows, pickup latitude and longitude, dropoff latitude and longitude are zero



We have identified and removed the missing values and outliers

Now, let's prepare data to develop the model

Consider target variable in train data as y\_train and remaining columns as x\_train since we need to split data to pass source and target variables to the model

y\_train = train\_data['fare\_amount']

x\_train = train\_data.drop(['fare\_amount'], axis=1 )

Consider test data as x\_test. Since we dont have target variable in test data

x\_test = test\_data.drop(['pickup\_datetime'], axis = 1)

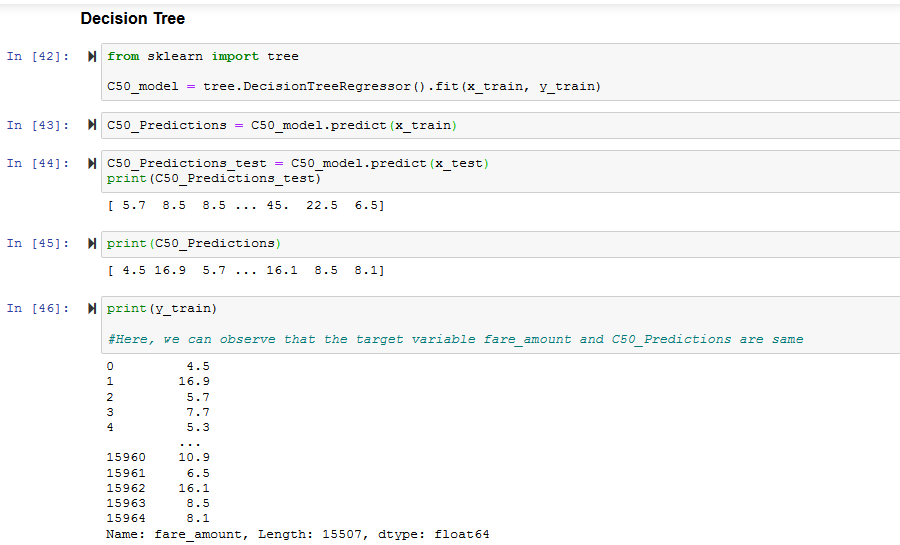
We have tried multiple models like Linear Regression, Decision Tree and Random forest

As we can observe in the below screenshot for the Decision Tree model, it is very accurate

Also, we have derived RMSE (Root Mean Squared Error) value, which is very low and close to 0.

In the below screenshot, we can see that the variables y\_train which is fare amount and predicted data C50\_predictions are same

Hence, we can consider Decision Tree as best fit for the given data as the predicted values for the given data are almost similar with high accuracy rate as observed below



Final predicted values for the given data can be found in the variable named as C50\_Predictions\_test in Python