CAB FARE PREDICTION

BY:

UMANG

Contents

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

1.2 Data:

Number of attributes:

- 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.

Let's take a quick view of a sample of the given train data:

	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
0	4.5	2009-06-15 17:26:21 UTC	-73.844311	40.721319	-73.841610	40.712278	1.0
1	16.9	2010-01-05 16:52:16 UTC	-74.016048	40.711303	-73.979268	40.782004	1.0
2	5.7	2011-08-18 00:35:00 UTC	-73.982738	40.761270	-73.991242	40.750562	2.0
3	7.7	2012-04-21 04:30:42 UTC	-73.987130	40.733143	-73.991567	40.758092	1.0
4	5.3	2010-03-09 07:51:00 UTC	-73.968095	40.768008	-73.956655	40.783762	1.0

2 Methodology:

In the first step, all the required libraries were installed and imported i.e. numpy, pandas, matplotlib, seasborn etc. to run the code.

2.1 Data Pre-processing:

After importing the required libraries, data pre-processing is the first step to do with data. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. It involves transforming data into a basic form that makes it easy to work with. Data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in one data set, and best out of them is chosen.

After analysing the data, I see that there is a date and timestamp variable (pickup_datetime). To make things in a better way, I have extracted year, month, date, hour and minute from date and timestamp variable.

Further, in this stage, we'll deal with outliers, missing values and some unwanted data that we don't want in our model.

2.2 Missing value analysis:

A missing value table is created to know how much amount of information is missed in our raw data. After analysing the data, I found that there are two attributes i.e. fare_amount and pasenger_count which have missing data 0.149% and 0.342% respectively.

	Variables	Missing_percentage
0	passenger_count	0.342317
1	fare_amount	0.149374
2	pickup_datetime	0.000000
3	pickup_longitude	0.000000
4	pickup_latitude	0.000000
5	dropoff_longitude	0.000000
6	dropoff_latitude	0.000000

The missing values can be filled by using various mathematical techniques i.e. Mean, Median, Mode, KNN imputation.

In python, missing values in fare_amount were filled by using Median and missing values in passenger_count were filled by using Mode method.

In R, I have used KNN imputation method to fill the missing values in fare_amount and Mode method for passenger_count.

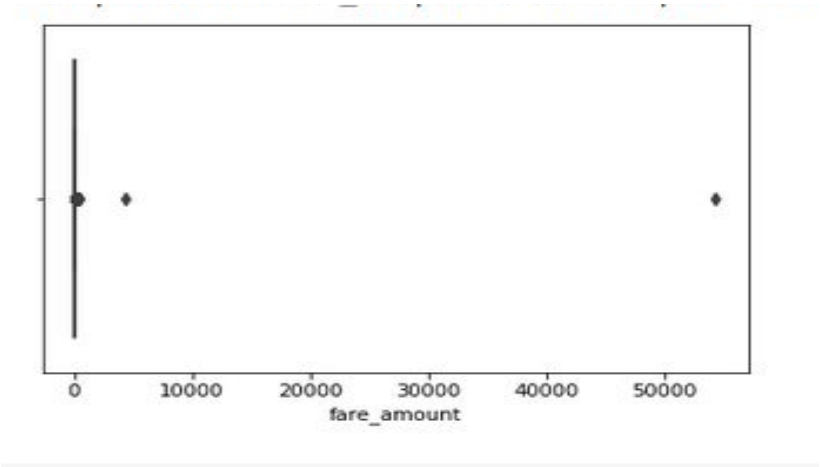
In passenger_count, I have considered the practical scenario and taken the values from 1 to 6 because a cab can't take more than 6 passengers comfortably at a time. There are some observations in data which shows the passenger_count in cabs are in double & triple digits and some observations in passenger_count has "0" value, hence they all are removed.

2.3 Outlier analysis:

After imputing the missing values, the next step performed is outlier analysis. An outlier is a data point that differs significantly from other observations. We can deal with outliers by using two methods:

- ➢ Remove the entire rows which have outliers
- ➢ Replace outliers with NA and then apply missing value analysis.

We can see the outliers by plotting the boxplot graph. I have pasted an image of boxplot graph of fare_amount.



In the boxplot graph, we are unable to see the upper fence and lower fence because the scale has very large value i.e > 50,000. I have removed the outliers by using upper fence and lower fence. There is a formula to calculate the interquartile range.

Maximum **=** *Q75 + 1.5*IQR*

*Minimum = Q25 -1.5*IQR*

first quartile (Q25 /25th Percentile): the middle number between the smallest number (not the “minimum”) and the median of the dataset.

third quartile (Q75 /75th Percentile): the middle value between the median and the highest value (not the “maximum”) of the dataset.

interquartile range (IQR): 25th to the 75th percentile.

The values which fall beyond “maximum” and “minimum” are treated as outliers. We have almost 10% outliers in fare_amount, if we remove them, we might lose some important information so I decided to change these values in NA and applied missing value analysis. These values were imputed by using Median and KNN imputation method in Python and R respectively.

Now, there are 4 columns related to longitude and latitude. These columns are pickup_longitude, pickup_latitude, dropoff_longitude and dropoff_latitude. We already know that the range of “longitude” is (-180, +180) and for latitude, it is (-90, +90). There is one row which contains observation beyond the limit, hence removed.

2.4 Distance calculation

After setting the range of longitude and latitude, I have calculated the distance(in km) between two points of longitude and latitude by using "*Haversine formula".* The *Haversine* formula calculates the shortest distance between two points on a sphere using their latitudes and longitudes measured along the surface. It is important for use in navigation. A function was created to calculate the distance by using Haversine.

dlon = lon2 - lon1

dlat = lat2 - lat1

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

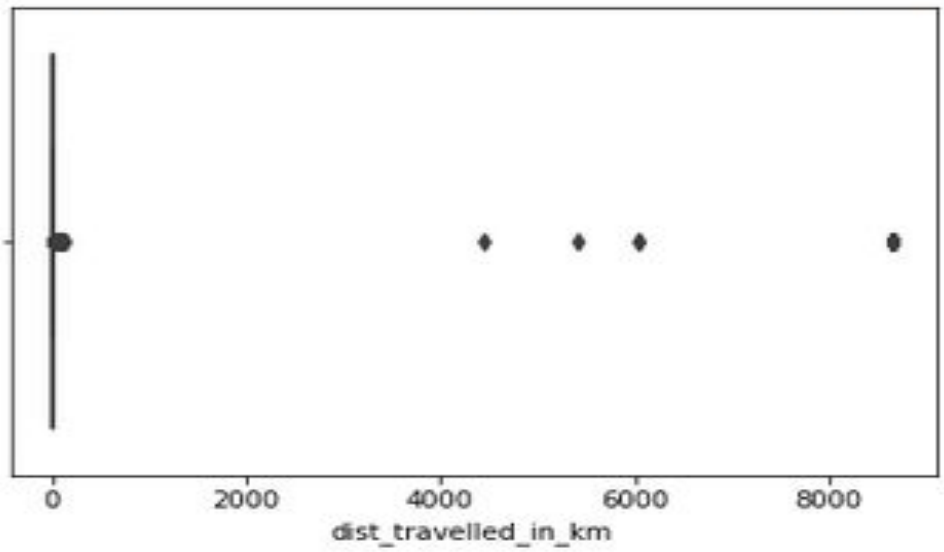
c = 2 * asin(sqrt(a))

Where, dlon = difference of pickup_longitude and dropoff_longitude

dlat = difference of pickup_latitude and dropoff_latitude

After calculating the distance, a boxplot graph of travelled distance is plotted and found that there are some observations which were treating as outliers. In R, NA was filled in place of outliers and missing value analysis was applied. In Python, outliers were removed.

Boxplot of distance_travelled:



2.5 Data Visualisation

Now, bar graphs were plotted between categorical variable and target variable to visualize the relationship.

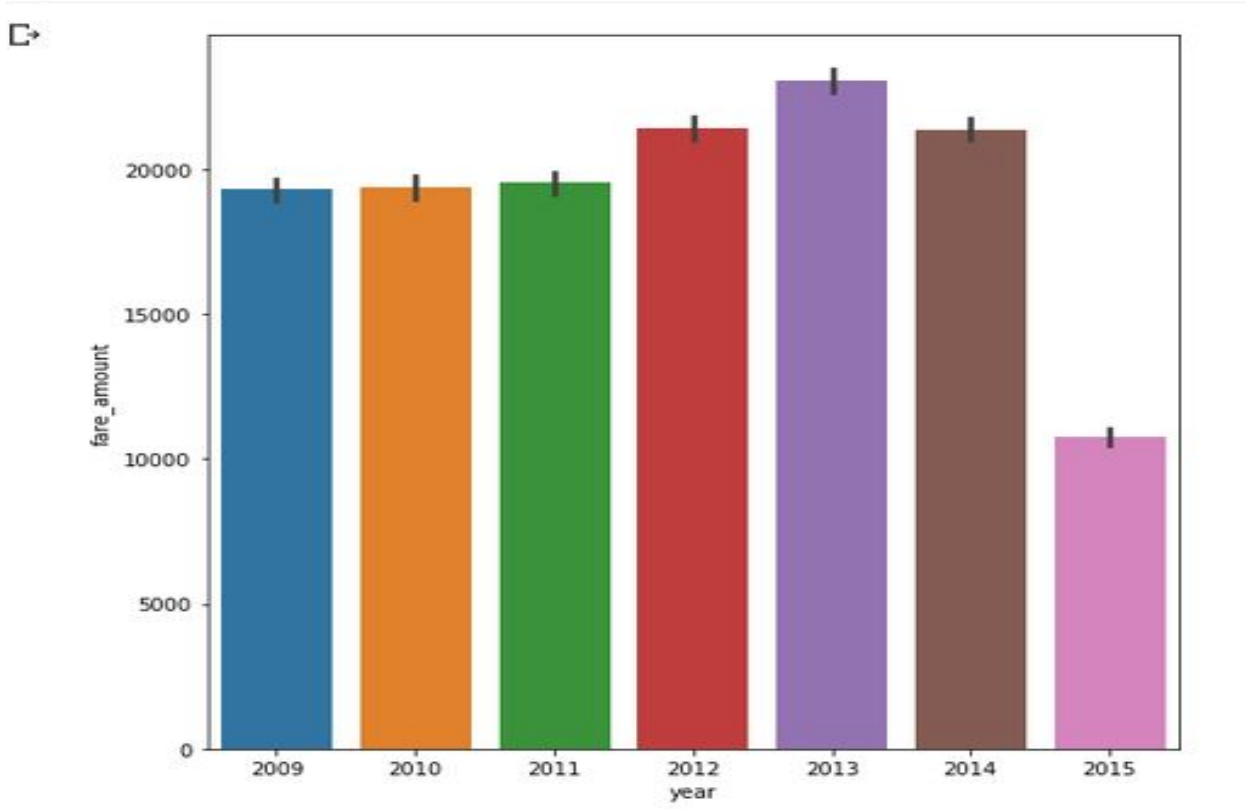


Fig1: Bar graph between year and total sum of fare_amount

We can clearly see in the graph that fare_amount is less in year 2015 as compared to others. As we can see there are six categories of different years so we don't need to categorize it further.

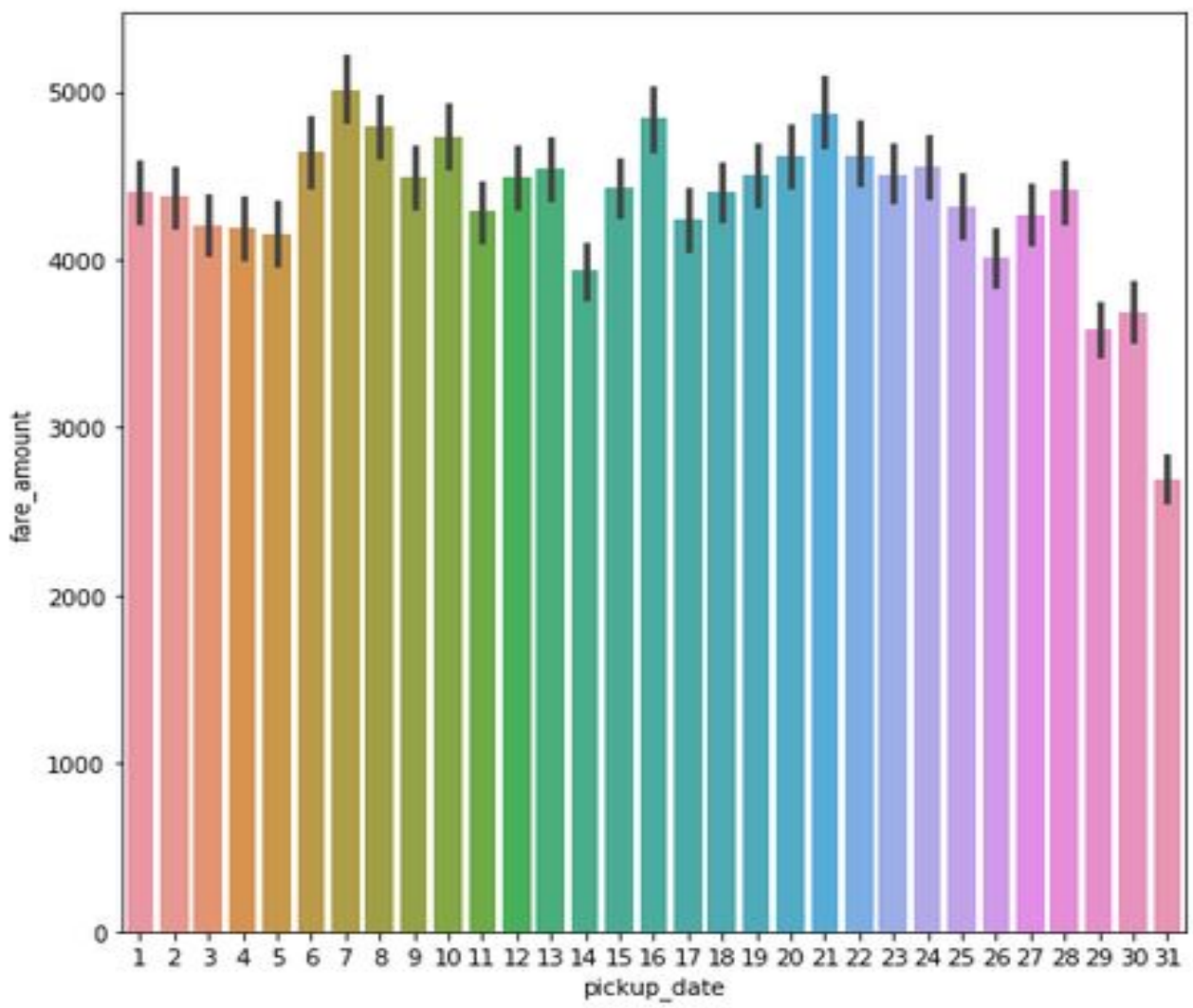


Fig2: graph between pickup_date and total sum of fare_amount

We can clearly see in the graph that fare_amount is less for the last 3 days of the month as compared to other days so I divided the days section in 2 categories.

- From date 29th to 31st, fare_amount is less so it was categorised as “1” .
- From date 1st to 28th, the fare_amount is high so it was categorised as “2” .

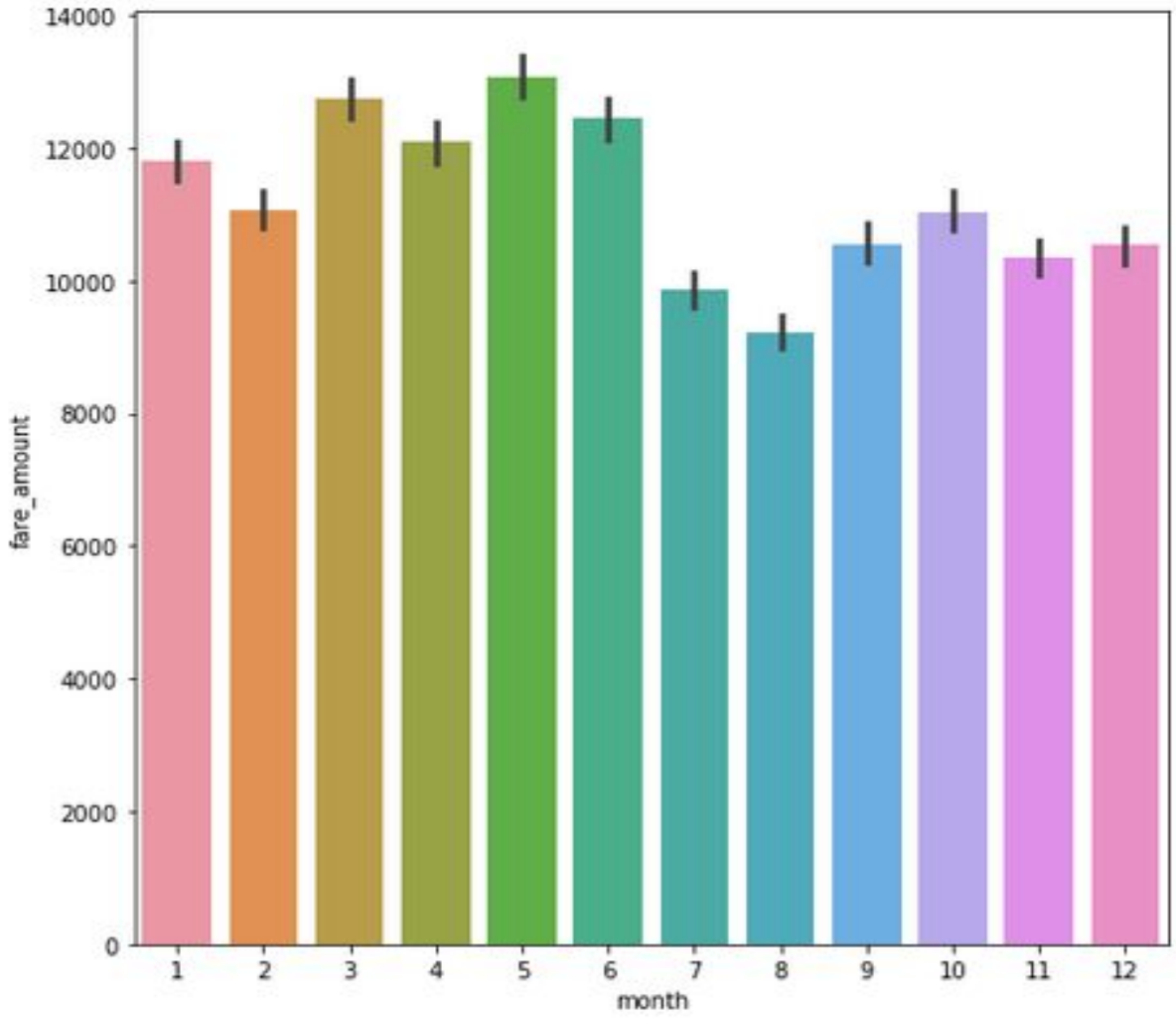


Fig3: graph between total sum of fare_amount and month

- This graph indicates that fare_amount is varying according to the different months in a year but there is not very much deviation in fare_amount in different months and it is already categorised into 12 different numbers.

- “1” for january, “2” for february and so on.

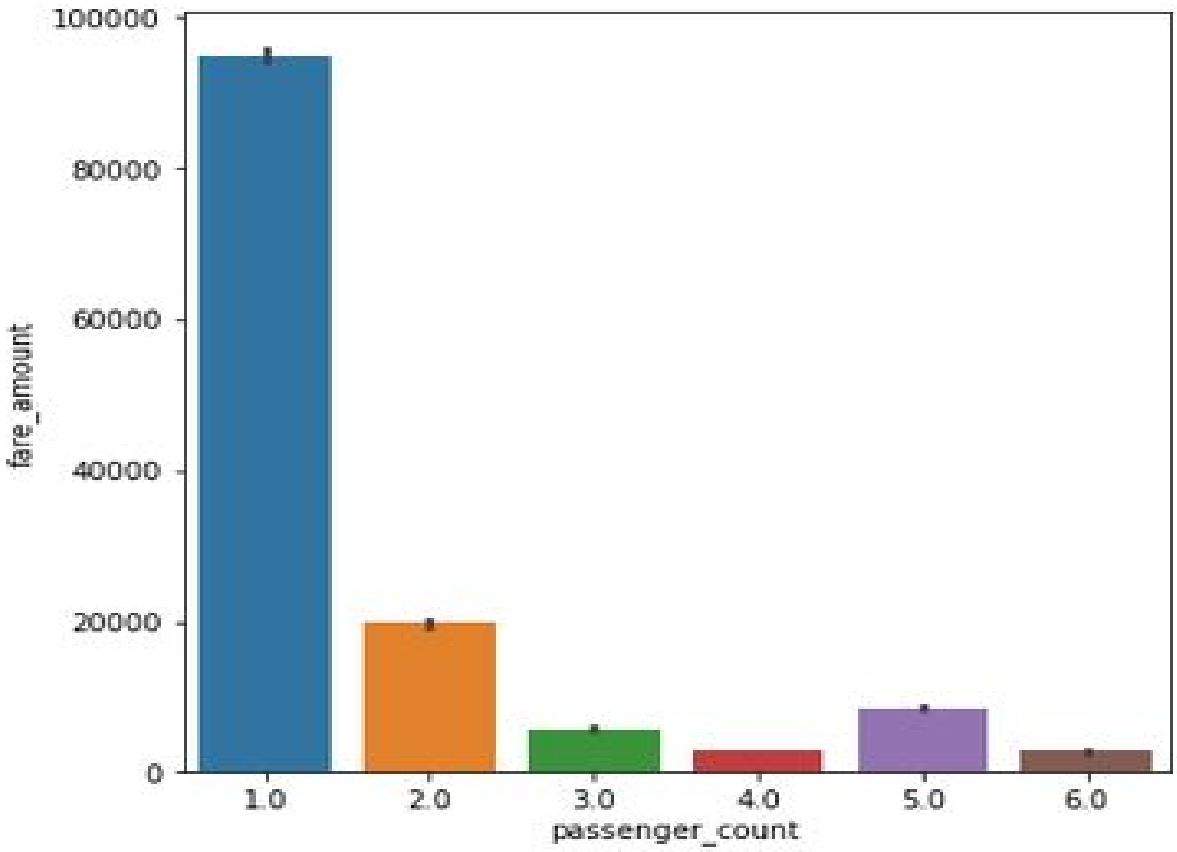


Fig4: graph between total sum fare_amount and passenger_count

We can clearly see in the bar graph that passenger_count is already divided into the categories so no need to change it further.

3. Feature Selection:

Feature selection is the process where we select those features which contribute most to our prediction variable. Having irrelevant features in your data can decrease the accuracy of the models and make your model learn based on irrelevant features. We can remove the irrelevant features by using some statistical techniques:

3.1 ANOVA test:

In both Python and R, I have done ANOVA test, it compares categorical variables with numerical variable.

Null hypothesis: No relationship exists on the categorical variables. The two variables are independent of each other.
Alternate hypothesis: The two variables are not independent.

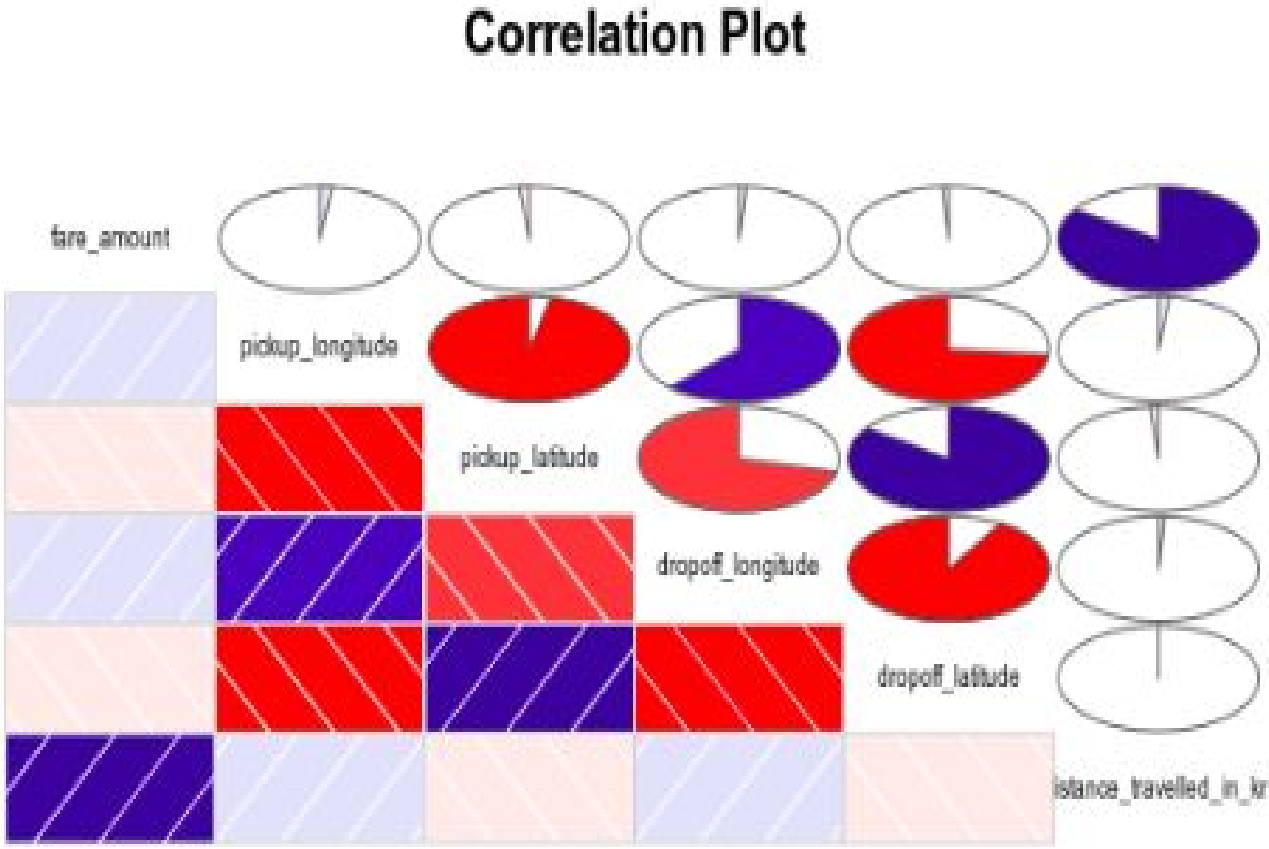
- If p-value < 0.05 then we reject the null hypothesis, saying that these two variables are independent. If p-value > 0.05 then we can't reject the null hypothesis, saying that these two variables are not independent.
- If p-value > 0.05, remove the variable from our data set and if p-value < 0.05, keep the variable.

```
[1] "passenger_count"
                Df Sum Sq Mean Sq F value Pr(>F)
df_train[, i]    5    240   47.93   2.515 0.0278 *
Residuals    15442 294321   19.06
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
[1] "Year"
                Df Sum Sq Mean Sq F value Pr(>F)
df_train[, i]    6   6299  1049.9   56.24 <2e-16 ***
Residuals    15441 288261    18.7
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
[1] "month"
                Df Sum Sq Mean Sq F value   Pr(>F)
df_train[, i]   11    858   78.03   4.101 4.74e-06 ***
Residuals    15436 293702   19.03
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
[1] "pickup_sessions"
                Df Sum Sq Mean Sq F value   Pr(>F)
df_train[, i]    2   1159   579.4    30.5 6.01e-14 ***
Residuals    15445 293402    19.0
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
[1] "pickup_days"
                Df Sum Sq Mean Sq F value Pr(>F)
df_train[, i]    1      4   3.765   0.197  0.657
Residuals    15446 294557  19.070
```

Table of ANOVA test

- In the above table, we can see that the p-value of pickup_days is greater than 0.05, hence dropped this variable.

3.2 Correlation analysis: It compares two numerical variables in a contingency table to see if they are related or not.



Correlation plot of numeric variable

- After observing the above table, we can see that some variables were highly correlated, hence removed dropoff_longitude and drop_off latitude.

3.3 Variance Inflation Factor(VIF) **:** We have performed VIF test using function VIF which is used to check whether variables have multicollinearity .

```
const                1.190010e+06
pickup_longitude     2.257470e+04
pickup_latitude      1.758946e+04
dropoff_longitude    2.255091e+04
dropoff_latitude     1.759864e+04
passenger_count      1.000590e+00
year                 1.014183e+00
pickup_days          1.001064e+00
month                1.014201e+00
pickup_session       1.002957e+00
dist_travelled_in_km 1.022340e+00
dtype: float64
```

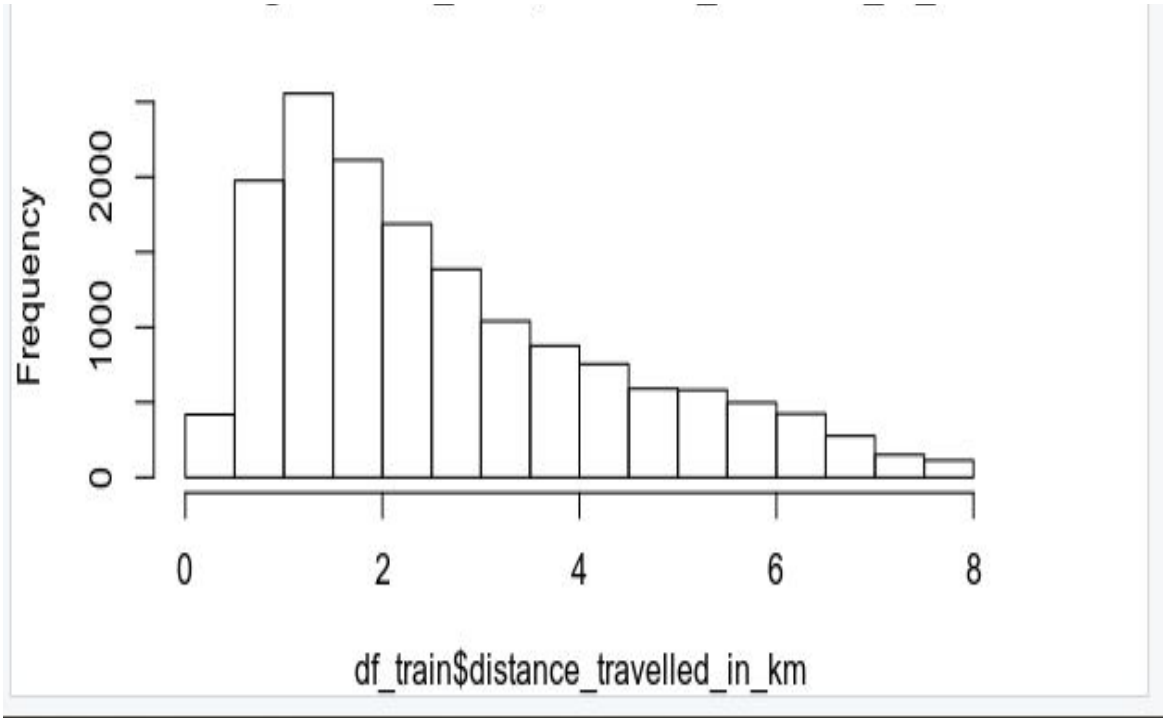
VIF table

4. Feature Scaling:

It is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization. Sometimes, it also helps in speeding up the calculations in an algorithm. I have done normalisation in both python and R.

4.1 Normalisation:

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values.



Histogram of distance travelled

- As we can clearly see in the above histogram that our data of "dist_travelled_in_km" is left skewed, I have normalised the numeric variable in the same range of (0,1).

5. Model development:

After preprocessing of data we must proceed with model development. Firstly, we split the clean data into test & train and then develop and apply different models. I have applied three different models.

5.1 Linear Regression Model: 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.

```
# Building LinearRegression Model on training Data
lm = LinearRegression()
lm.fit(X_train,y_train)
```

5.2 Decision Tree model: Decision Tree is like a rule, it can be used in both Regression and Classification problem statements. Each branch connects node with "and" multiple branches are connected by "or".

```
# Building DTModel on training data
fit_DT  = DecisionTreeRegressor(max_depth = 4).fit(X_train, y_train)
fit_DT
```

```
# Apply model on splitted test data
predictions_DT = fit_DT.predict(X_test)

predictions_DT
```

5.3 Random Forest model: Random Forest is an ensemble that consists of many decision trees. It can be used in both types of problem statements i.e. Regression and Classification.

```
# Building RFModel 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)
```

6. Result:

In Python:

	RMSE	r2	MAE	MAPE
Linear Regression	2.429565	0.621539	1.655037	0.203424
Decision Tree	2.374108	0.638619	1.657890	0.198723
Random Forest	2.364459	0.641550	1.649496	0.197446

In R:

Linear Regression:

```
    RMSE Rsquared      MAE     MAPE(X_test[,1],RF_Predictions)
2.244280 0.737322 1.593874   0.1822015
```

Decision Tree:

```
     RMSE  Rsquared      MAE     MAPE(X_test[,1],DT_prediction)
2.4909875 0.6758457 1.8411142   0.2150518
```

Random Forest:

```
    RMSE  Rsquared      MAE     MAPE(X_test[,1],RF_Predictions)
2.2100726 0.7475652 1.5838288   0.1822015
```

7.Conclusion:

As it can be clearly seen from the result ,Random Forest Regression model is performing good for Cab fare dataset (for all the metrics i.e. RMSE, MAE, MAPE, r^2). So, I have selected Random Forest Regression for the prediction of the model.

Further, MAPE value is 0.18 which is a good number as per industry standards. Mean absolute percentage error is commonly used as a loss function for regression problems and in model evaluation, because of its very intuitive interpretation in terms of relative error.