CAB FARE PREDICTION Project

Problem Statement:

The Goal of this project is to predict the cab fare based on the given Information.

In this Problem, historical train data is given to us which has 16067 observations and 7 variables.

If we see the variable in data set, there are:

'pickup_datetime', 'pickup_longitude', 'pickup_latitude', 'dropoff_longitude', 'dropoff_latitude', 'passenger_count', 'year','month', 'day', 'hour', 'weekday', 'distance', 'fare_amount']

Where fare_amount is our target variable

Data Pre processing:

I have pre-process the train data to prepare it for modelling and will use different techniques

To preprocess the data, first I have created new variables like weekday, date, month, year, hour, from pickup_datetime and calculated distance between two geolocation using haversine formula.

After this, I have new variables:

['pickup_datetime', 'pickup_longitude', 'pickup_latitude','dropoff_longitude', 'dropoff_latitude', 'passenger_count', 'year','month', 'day', 'hour', 'weekday', 'distance', 'fare_amount']

Now the statistics of data:

	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	distance	fare_amount
count	15906	15906	15906	15906	15906	15906	15906
mean	-72.4754	39.89899	-72.4656	39.89953	1.649467	15.06822	15.06429
std	10.53715	6.185843	10.56508	6.185468	1.265771	311.6932	432.2966
min	-74.4382	-74.0069	-74.4293	-74.0064	0.12	0	0.01
25%	-73.9921	40.73494	-73.9912	40.73471	1	1.215848	6
50%	-73.9817	40.75263	-73.9802	40.75356	1	2.126809	8.5
75%	-73.9668	40.76738	-73.9636	40.76801	2	3.855717	12.5
max	40.76613	41.36614	40.80244	41.36614	6	8667.542	54343

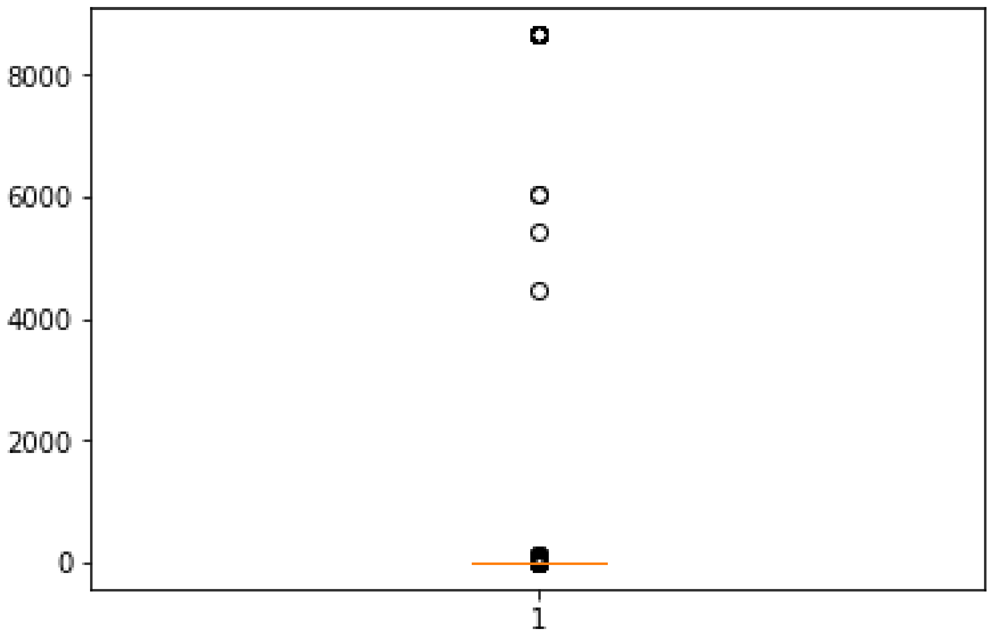
There are some anomalies we need to remove

- Longitude valid range should be +/-180 degree
- Latitude range should be +/- 90, but in pickup latitude max is 401.08 which is invalid
- Passenger count minimum is 0 which is not possible and maximum is 5345 which is also unreal
- so let's assume maximum passenger count be 200 (assume cab can be is a mini bus too)
- fare amount cannot be negative

So we need to remove these anomalies

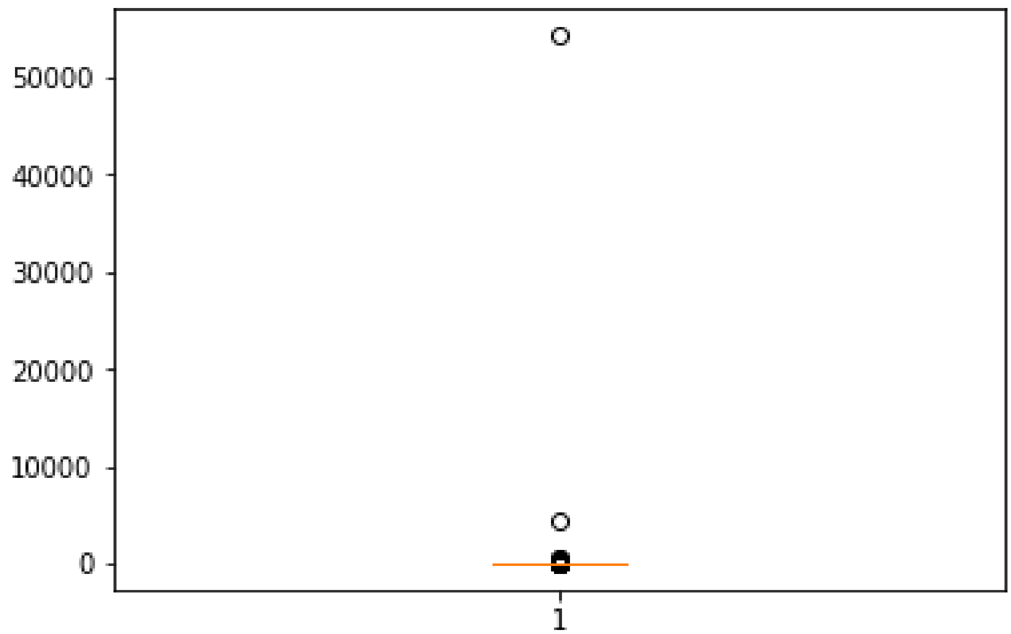
After removing above anomalies we have data size of (15906, 7)

Now check for outliers in "distance"



If we assume maximum distance available to book a cab is 200 then there are 23 outliers which is very small so we can remove them.

Now see outliers in fare_amount



Lets assume maximum fare amount be 200 then there are 4 outliers which is very small so we can remove them too

Now there are some conditions which are unreal :

- if distance = 0, fare_amount canot be greater than 0
- Remove observations where passenger_count = 0
- Remove observations where distance = 0 and fare amount =0
- After that there is only 1 missing data So we can remove that too

So Now the final observations are 15424

	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	distance	fare_amount
count	15424	15424	15424	15424	15424	15424	15424
mean	-73.9109	40.6886	-73.9099	40.6891	1.651776	3.442776	11.31444
std	2.679344	2.633265	2.679454	2.632917	1.267724	4.597316	9.472102
min	-74.4382	-74.0069	-74.227	-74.0064	0	0.000111	0.01
25%	-73.9924	40.73657	-73.9914	40.7363	1	1.277861	6
50%	-73.9821	40.75334	-73.9806	40.75424	1	2.191322	8.5
75%	-73.9682	40.7678	-73.9655	40.76831	2	3.93786	12.5

max	40.76613	41.36614	40.80244	41.36614	6	129.9505	165

Now data looks much clearer

Feature Selection

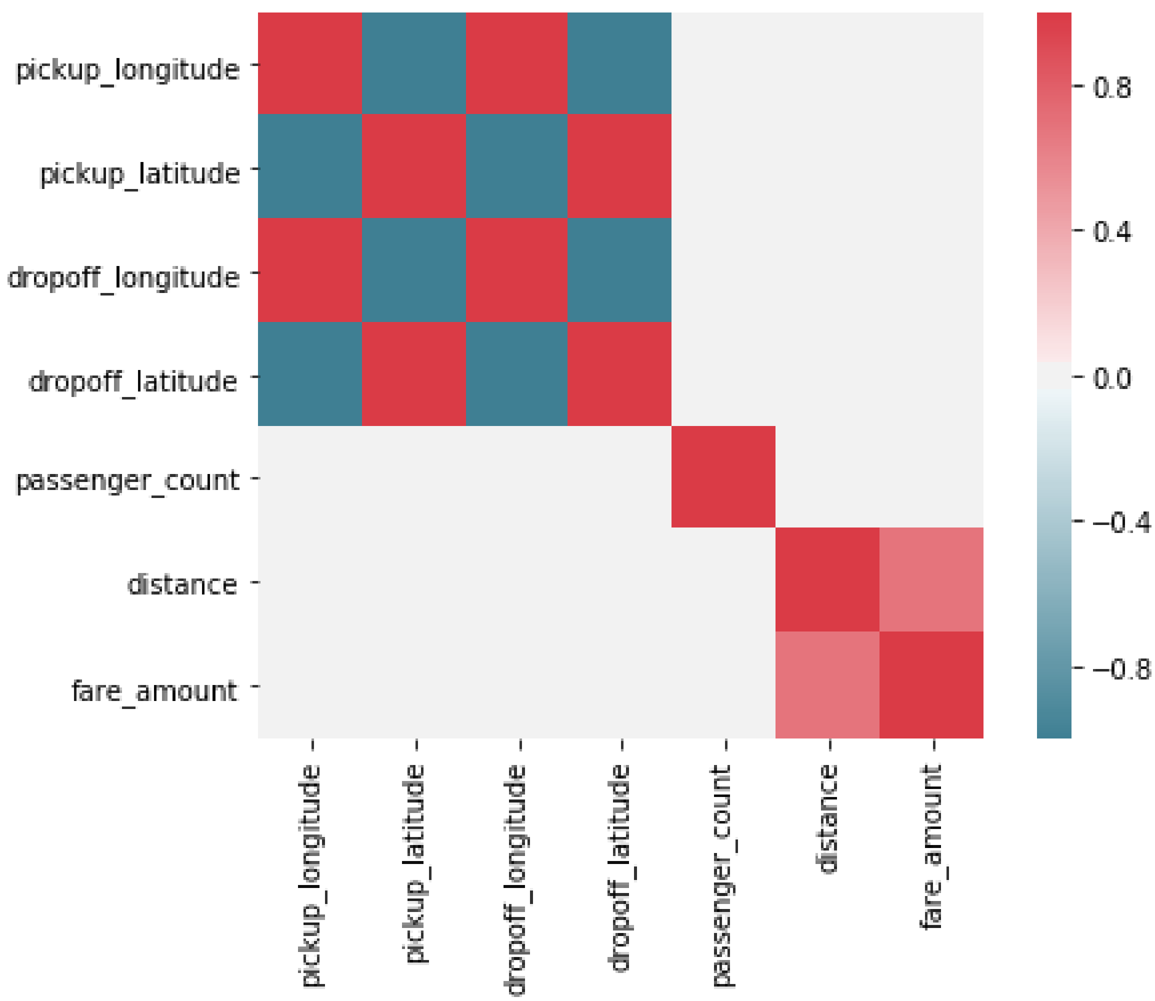
Hypothesis:

Now the next step to solve the problem should be hypothesis

We have following hypothesis

- More the distance, more the fare
- Fare amount may be different for weekdays and weekends
- During Peak hours, fares may be high

If we see the heat map of Numerical variables, we can see that fare_amount is highly corelated with distance.



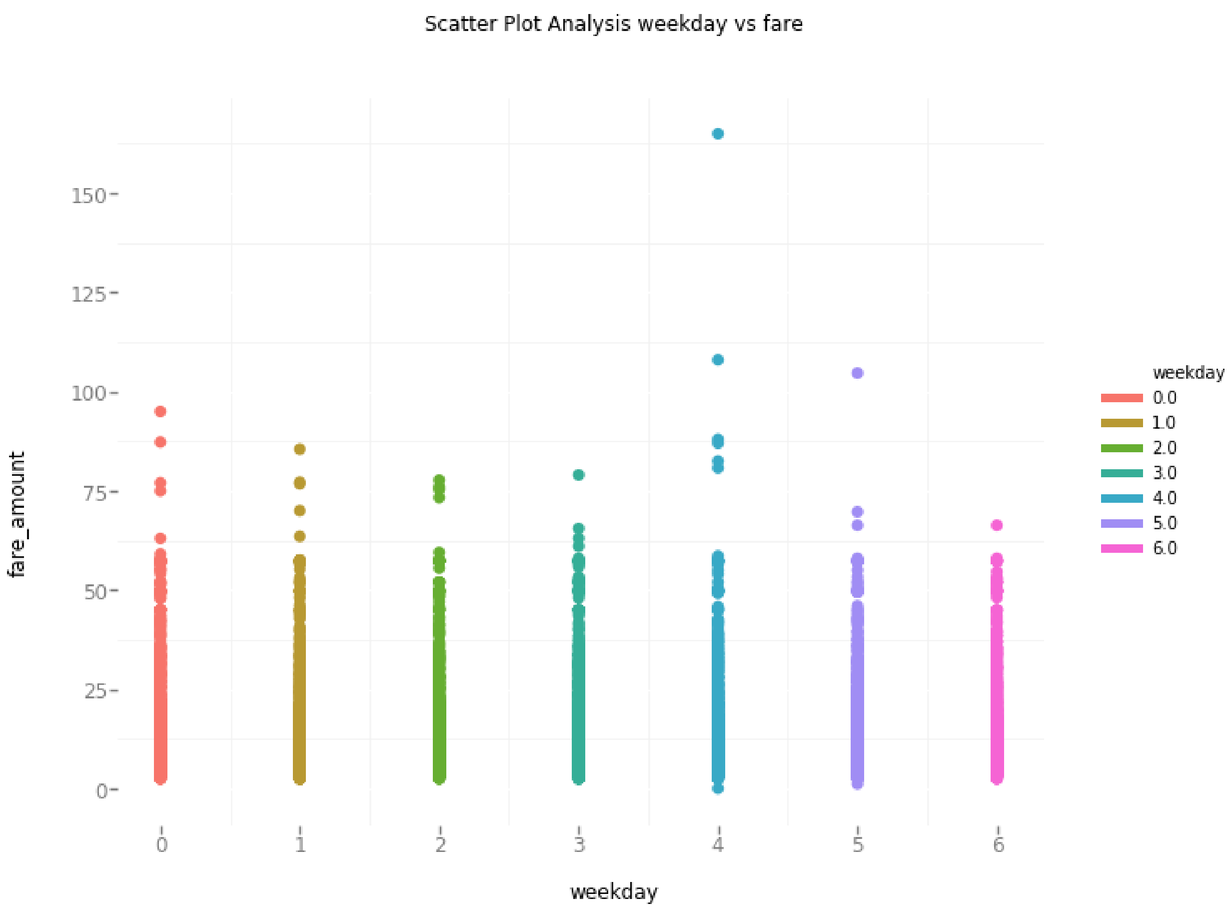
Also We have calculated distance from pickup and dropoff longitude/latitude, so we can remove these variables

We see that passenger and distance are highly independent

Now let’s see if weekday or time affects fare_amount or not,

For this see the below scatter plot in which 0-Monday, 6-Sunday

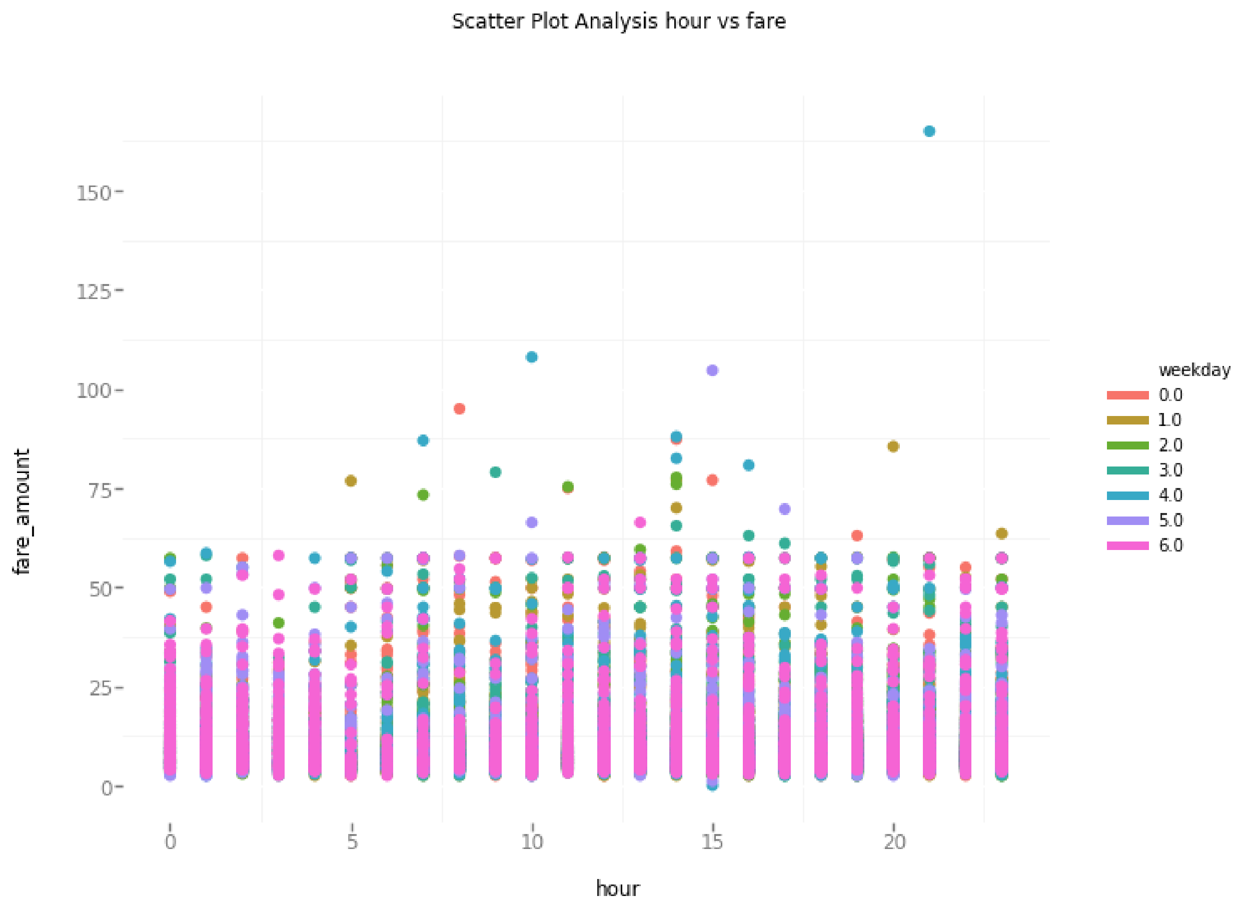
We see that the fare amount is almost same but on Monday, Tuesday and Friday it goes very high



Now Lets see the impact of hours on fare_amount

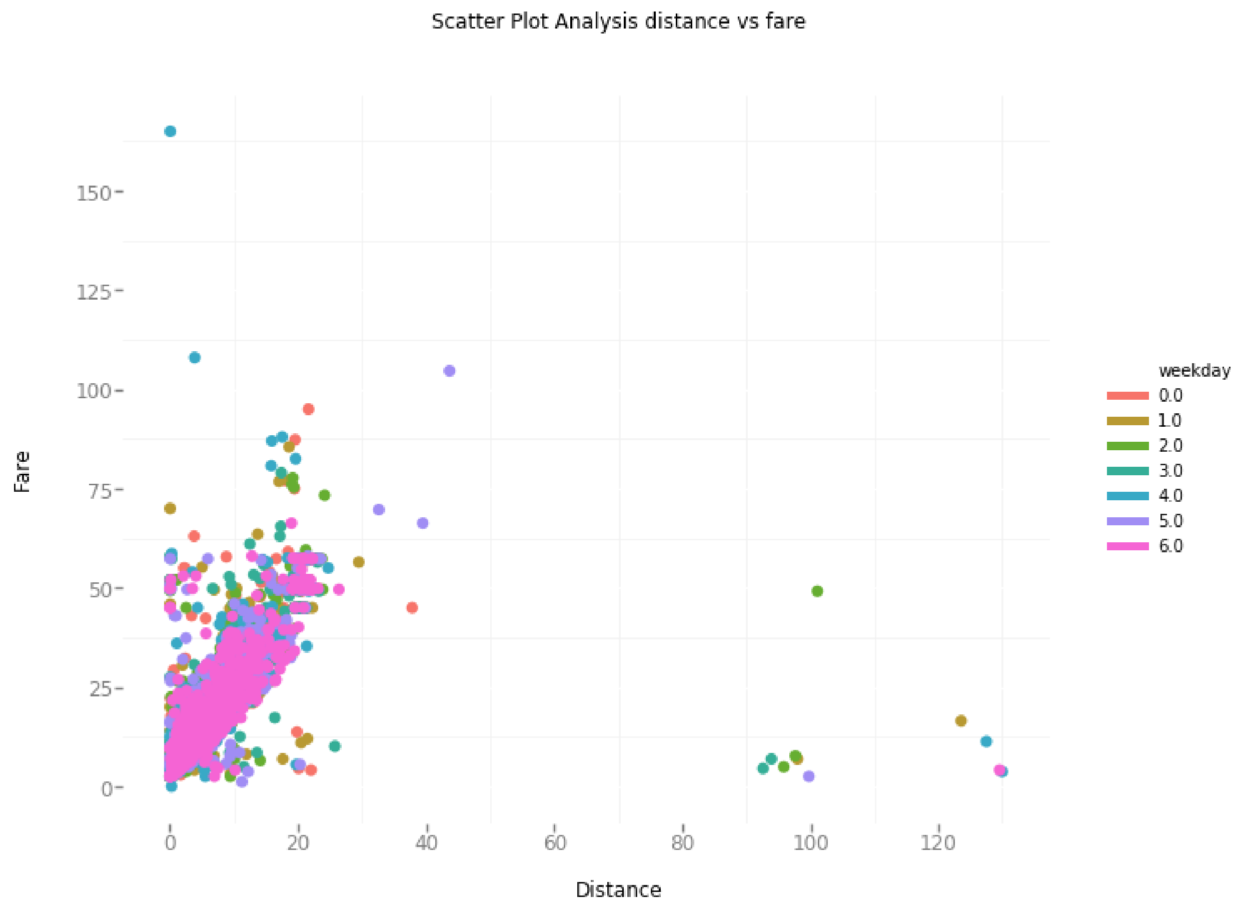
we can see that fare amount is less in the nights when weekdays but on weekends the fare amount is high in the night but very less from 5:00AM to 9:00 AM

If I see on weekdays, the fare_amount are high from 9:00 AM to 11:00AM and 2:00PM to 8:00 PM



Lets analyse distance vs fare_amount

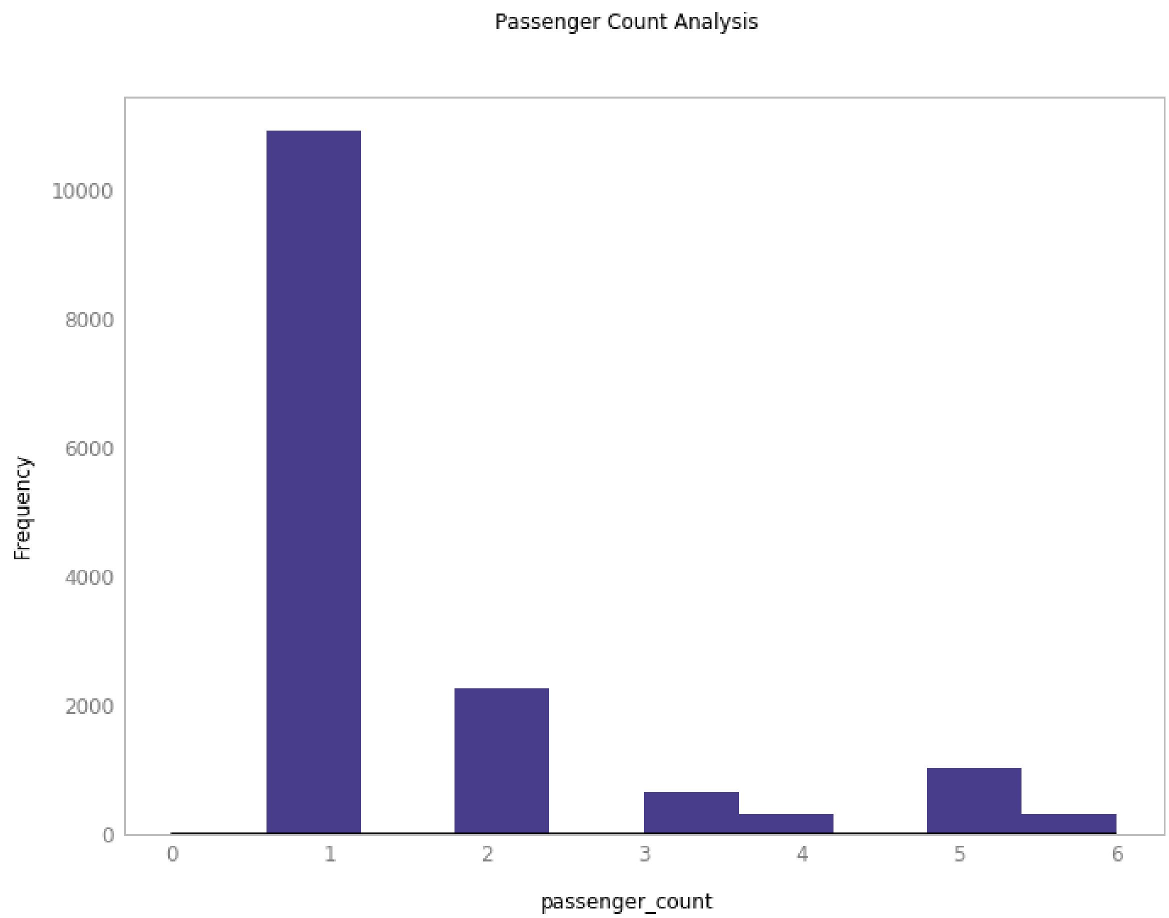
We can see that maximum rides were on Sunday

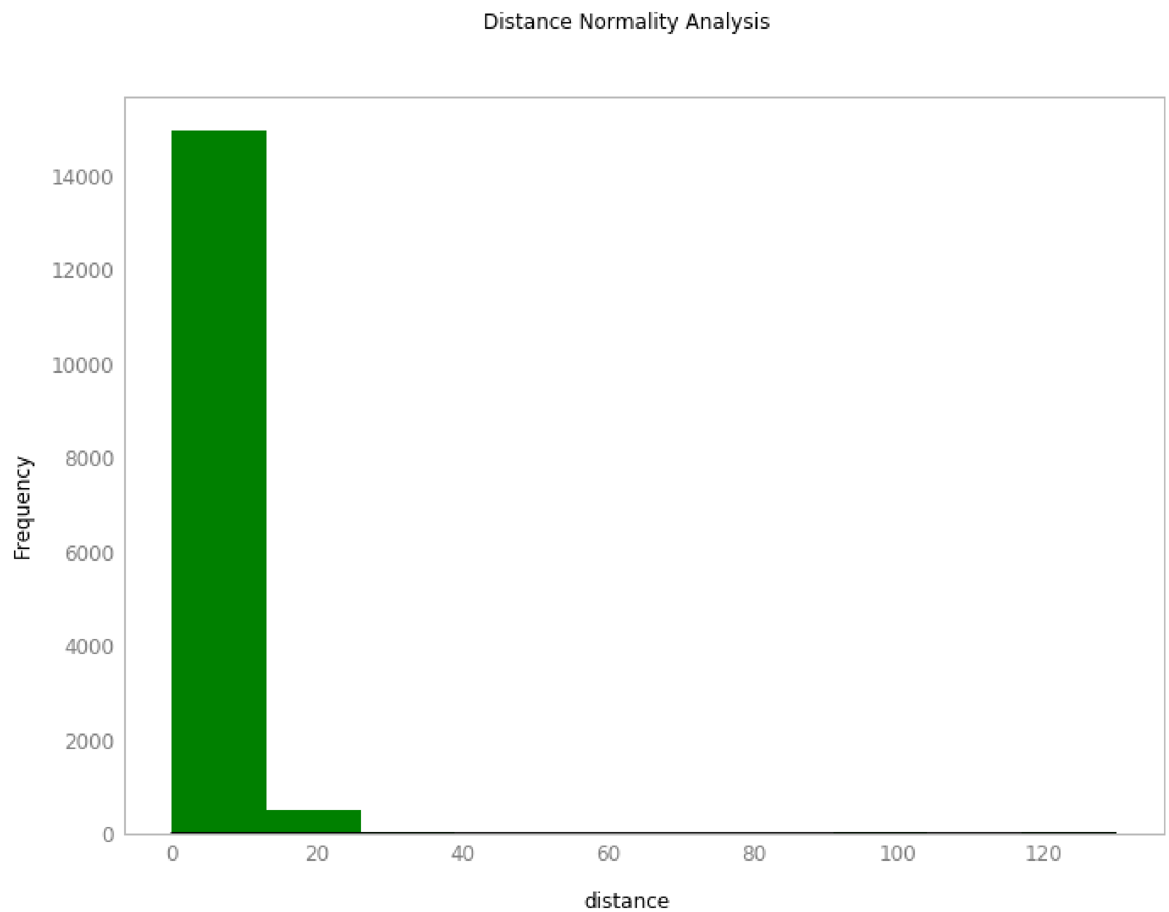


Feature Scaling:

We know feature scaling need to be done on Numerical variables and we have two independent numerical data here, passenger_count and distance

If we see the histogram of these Numerical data we can see that they are not normally distributed, So we should use normalization method to do feature scaling.


Passenger Count Analysis
Frequency
10000
8000
6000
4000
2000
0
0
1
2
3
4
5
6
passenger_count



Modelling

Now we have train data with 15424 observations and we are ready to model our data

For that I have sampled train data into train,test

As My problem is not related to classification, but a prediction problem, So We can use following Algorithms

1) Linear Regression
2) Decision Tree Algorithm
3) Random Forest Algorithm and
4) KNN Algorithm

    As this is Regression (Prediction) problem, We van use RMSE/MAPE error Metrics.
    Here I am using RMSE to see the model performance.
    Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. This means the RMSE should be more useful when large errors are particularly undesirable
    And here we need a fare prediction to expand our business and we have to predict correctly

Now lets see RMSE for different Models

1) Linear Regression Model
   R-squared = .797 (79.7 % of dependent variable can be explained by independent variables)
   RMSE = 6.058
2) Decision Tree Algorithm
   RMSE = 4.488
3) Random Forest Algorithm (with 150 tree/estimators)
   RMSE = 3.986
4) KNN Algorithm
   RMSE = 9.275

   Now we can see that Random Forest Algorithm is best match for this problem