Car Fare Prediction Project Report

Dishank Bari

12th August 2019

Project Name: Car Fare Prediction

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

Dataset:

1) **train_cab dataset**
2) **test dataset**

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 cabride.

Missing Values: Yes

1. Aim of the project

The aim of this project is to create an almost production-ready model that predicts a car ride's fare based on the information like pickup_datetime, pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude, and passenger_count variables would be able to provide the driver or customer who planning to book the car ride.

2. Understanding the Problem statement

As we see, the data we get in both type like train_cab data, test data. Means here, we need to do analysis on the train_cab dataset and finalize the model from it, then apply selected model on the test dataset to predict the car fare.

Predicting fare and duration of a ride can help passengers decide when is the optimal time to start their commute, or help drivers decide which of two potential rides will be more profitable, for example. Furthermore, this visibility into fare will attract customers during times when ride sharing services are implementing surge pricing.

In order to predict the fare, the data which we have consist of

- pickup_datetime,
- pickup_longitude,
- pickup_latitude,
- dropoff_longitude,
- dropoff_latitude, and
- passenger_count

variables. As we know that, this is the regression dataset. For that reason we need to consider regression machine learning model. That we discussed later, when we reach at modelling and predictions.

3. Data Exploration: Understanding the data

Before moving forward, let's understand the dataset which we have:

Here, we see that two dataset present i.e. train_cab dataset, which we used for training our machine learning model and other is test dataset on which we have to implement our model and predict the car fare for observations.

After loading the dataset in jupyter notebook or rstudio, first we find out the number of observations in train and test dataset.

As it shows, we have 16067 observations & 7 variables in train dataset and 9914 observations & 6 variables in test dataset.

	Number of Observations	Number of Variables
train_cab dataset	16067	7
test dataset	9914	6

One variable extra in train_cab dataset means, there is one dependent variable i.e. fare_amount, which we need to predict for test dataset.

Variables in train_cab dataset:

- fare_amount – dependent variable
- pickup_datetime, - independent variable
- pickup_longitude, - independent variable
- pickup_latitude, - independent variable
- dropoff_longitude, - independent variable
- dropoff_latitude, - independent variable
- passenger_count - independent variable

Variables in test dataset

- pickup_datetime, - independent variable
- pickup_longitude, - independent variable
- pickup_latitude, - independent variable

- dropoff_longitude, - independent variable
- dropoff_latitude, - independent variable
- passenger_count - independent variable

4. Data Conversion

To perform various operations, we need to get these variables in proper form. For that reason, we need to check the dypes of all the variables.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16067 entries, 0 to 16066
Data columns (total 7 columns):
fare_amount          16043 non-null object
pickup_datetime      16067 non-null object
pickup_longitude     16067 non-null float64
pickup_latitude      16067 non-null float64
dropoff_longitude    16067 non-null float64
dropoff_latitude     16067 non-null float64
passenger_count      16012 non-null float64
dtypes: float64(5), object(2)
memory usage: 878.7+ KB
```

As we see that train_cab dataset has fare_amount and pickup_datetime in wrong dtypes. (datatypes) For that reason we need to convert them in proper dtypes

- fare_amount convert into float dtypes
- pickup_datetime convert into datetime dtypes

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9914 entries, 0 to 9913
Data columns (total 6 columns):
pickup_datetime      9914 non-null object
pickup_longitude     9914 non-null float64
pickup_latitude      9914 non-null float64
dropoff_longitude    9914 non-null float64
dropoff_latitude     9914 non-null float64
passenger_count      9914 non-null int64
dtypes: float64(4), int64(1), object(1)
memory usage: 464.8+ KB
```

Similarly, we need to check the dtypes for test dataset. Here we found that pickup_datetime and passenger_count has wrong dtypes, then we need to convert them in useable dtypes.

- pickup_datetime convert into datetime
- passenger_count convert into float.

While dealing with these dataset, I found some of the wrong entries (observations) in train_cab dataset. To get train dataset in proper structure I need to discard these observations.

5. Data Preprocessing:

5.1 Missing Value Treatment:

Now next step in our analysis is to do missing value treatment. Missing value treatment is an important part of our analysis, the data we get has many missing values due to many reason because these data gather from various resources.

Let's check the missing value in train_cab dataset.

```
passenger_count      55
fare_amount          24
dropoff_latitude      0
dropoff_longitude     0
pickup_latitude       0
pickup_longitude      0
pickup_datetime       0
dtype: int64
```

As we see, that there are 55 missing values in passenger_count variable and 24 missing values in fare_amount. Compare to our larger train_cab dataset these values are very small in count, however we can discard or drop these missing values. In these we drop around some observations.

6. Data Manipulation

6.1 Outliers:

Some of the observations are inconsistent with rest of the dataset are called outliers. This is because of poor data quality, low quality measurements, manual error and more.

In this data manipulation, we need to find out the inconsistent data from both train_cab and test dataset. These inconsistent data will affect our analysis during modelling. That why we need to deal them here.

We deal these outliers in variables, one by one.

1) fare_amount:

As we know that fare_amount should be a positive number, due to we are dealing with the car fare_amount.

```
count    15986.000000
mean        15.030453
std        431.213944
min         -3.000000
25%          6.000000
50%          8.500000
75%         12.500000
max      54343.000000
Name: fare_amount, dtype: float64
```

As you see, in above images, the minimum fare_amount is -3.00000 which is not possible as amount in rupees, dollar, or euro.

Need to find the count of negative values if there is any zero fare_amount value present in train_cab dataset. After counting analysis, we found that there are 4 observations have negative and zero values. So we need to drop these observations.

Also the maximum values of fare_amount showing is 54343.000, which is not possible in Car Fare Amount. It will lead wrong conclusion. So either I need to correct that value or need to drop that observations

After Checking highest fare_amount, fount that there 3 values which are highest and not possible in fare_amount i.e. fare-amount = 54343.0000, fare_amount = 4343.0000 and fare_amount = 0.01. I need to drop these values from the train_cab dataset.

After these operations,

- The highest fare_amount is 453.00
- The lowest fare_amount is 1.14

2) passenger_count:

```
count    15979.000000
mean         2.623795
std         60.905468
min          0.000000
25%          1.000000
50%          1.000000
75%          2.000000
max       5345.000000
Name: passenger_count, dtype: float64
```

As the above image shows, that the minimum passenger_count in train_cab dataset is 0 and maximum passenger_count in train_cab dataset is 5345.00. Lol which is not possible even in train.

After Checking, passenger_count ranges from 35 to 5345. As I am analyzing dataset of Cabs, then this is not possible. So these values are strictly outliers. I need to drop them.

For car fare prediction, I am consider the Max Number of passengers are 6. Which make sense if car is "SUV".

Also there are some observations with passenger_count = 0 & 0.12, So these values are stictly outliers. I need to drop them. Because this thing is not possible in case of passenger number (passenger_count).

After dropping these outlier observations, the

- Minimum passenger_count = 1
- Maximum passenger_count = 6

Now, we need to clear passenger_count in case of test dataset. Because passenger_count variable also present in test dataset.

```
count    15902.000000
mean         1.649686
std          1.265840
min          1.000000
25%          1.000000
50%          1.000000
75%          2.000000
max          6.000000
Name: passenger_count, dtype: float64
```

The image shows that, in test dataset, the minimum passenger_count is 1 and maximum passenger_count is 6, means we are on right track. Our test dataset is free from this outliers.

3) pickup_latitude and pickup_ longitude

First we explore the pickup_latitude and pickup_longitude in train_cab dataset.

Exploring the pickup longitute and latidude it is difficult to know the location for that reason I just Google these things. Found one useful website to identify the location https://www.latlong.net/Show-Latitude-Longitude.html

The above website gives the Actual Location based on longitude and latitude after doing so random check on website and using describe function, I conclude that the pilot project run in New York City of United States.

Goggling gives the New York City Longitude and Latitude i.e.

> New York City Latitude and longitude coordinates are: 40.730610, -73.935242.

After using describe function on train_cab dataset pickup_longitude and pickup_latitude are near to the actual coordinates of New York City. For that I am considering the longitude and latitude ranges for our train_cab dataset

- Latitude ranges from 39 to 43
- Longitude ranges from -72 to -76

The map shows the exact picture, why I am using latitude and longitude range for these analysis. The map is taken from following website:

https://www.mapsofworld.com/usa/states/new-york/lat-long.html

As the image shows, the latitude and longitude of pickup are ranges from

- Latitude ranges from 39 to 43
- Longitude ranges from -72 to -76

The observations, which are not include, in these ranges are dropped. Because there are many observations with 0.0000 longitude and latitude coordinates.

After, these operations, same will be applied for dropoff_longitude and dropoff_latitude.

4) dropoff_latitude ad dropoff_longitude

Similar Longitude and Latitude coordinates are used to drop the outliers from dropoff_longitude and dropoff_latitude.

These operations dropped some of the observations that are outliers.

7. Feature Selection

Feature selection is an important part, in this selecting a subset of relevant features like variables for use in model construction. Here we focus upon the input variables, which learn the algorithms and predict the results should be consider and ignoring the rest.

For Numeric Variables: We go with correlation analysis. Correlation tells you the association between two continuous variables.

Here, we do the Correlation Analysis:

Now we generate the correlation matrix and correlation plot

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
fare_amount	1.000000	0.298960	-0.131786	0.252538	-0.105001	0.004734
pickup_longitude	0.298960	1.000000	0.196346	0.390560	0.154156	-0.004675
pickup_latitude	-0.131786	0.196346	1.000000	0.158357	0.505780	-0.003151
dropoff_longitude	0.252538	0.390560	0.158357	1.000000	0.260921	-0.007609
dropoff_latitude	-0.105001	0.154156	0.505780	0.260921	1.000000	-0.003112
passenger_count	0.004734	-0.004675	-0.003151	-0.007609	-0.003112	1.000000

To understand the above correlation matrix, this ranges from -1 to 1

1: highly positively correlated variables

-1: highly negatively correlated variables

0: no correlation between variables

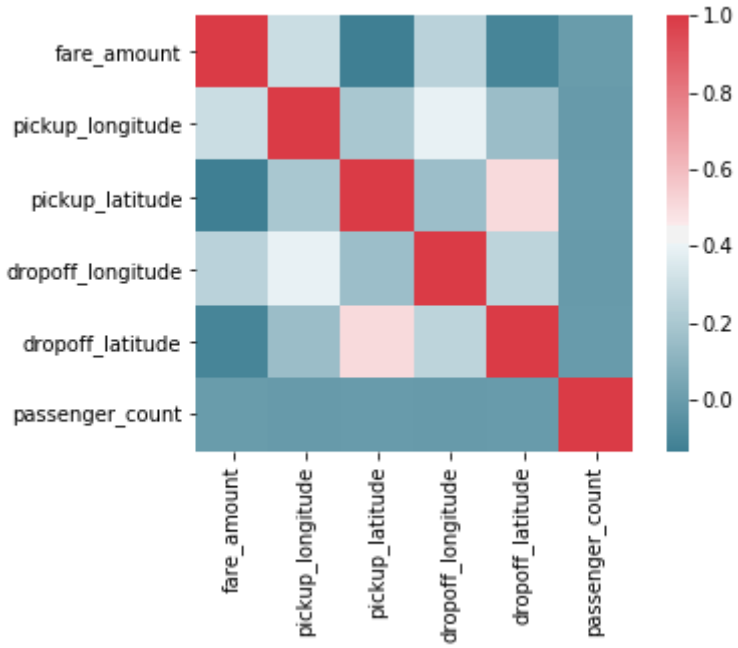
Acceptance and rejection of variables based on:

- Correlation between two variables ranges from 0.7 to 1 and -1 to -0.7 then these variables, will not help to predict the final results, so you can drop any of the variable.
- Other than this accept these variables for model building.

In our case, none of the two variables correlation ranges 0.7 to 1 and -1 to -0.7. So we accept all these variables for model building.

Now, let's do correlation analysis with graph. Here you see that color code which help you to analyze the correlation between two variables. Below graph is self-explainer.

- Red Color - variables are negatively correlated.
- Blue Color – variables are positively correlated.



The above correlation analysis shows that, each variable in dataset is independent and not correlated with each other. So, each variable or feature play an important role to predict the fare_amount.

8. Exploratory Data Analysis

For this analysis, we do some assumptions based on our dataset as follows:

1) Does the passenger_count (Number of Passengers) affect the fare_amount (fare)?
2) Does the pickup_datetime (Pickup Date & Time) affect the fare_amount (fare)?
3) Does the day of the week affect the fare_amount (fare)?
4) Does the distance travelled affect the fare_amount (fare)?

To clarify these assumptions we need to visualize the each variables against each other. We need to split the pickup_datetime in required form. As follows:

- Year
- Month
- Date
- Hour
- Day of the week

	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	Year	Month	Date	Day_of_week	Hour
0	4.5	2009-06-15 17:26:21	-73.844311	40.721319	-73.841610	40.712278	1.0	2009	6	15	0	17
1	16.9	2010-01-05 16:52:16	-74.016048	40.711303	-73.979268	40.782004	1.0	2010	1	5	1	16
2	5.7	2011-08-18 00:35:00	-73.982738	40.761270	-73.991242	40.750562	2.0	2011	8	18	3	0
3	7.7	2012-04-21 04:30:42	-73.987130	40.733143	-73.991567	40.758092	1.0	2012	4	21	5	4
4	5.3	2010-03-09 07:51:00	-73.968095	40.768008	-73.956655	40.783762	1.0	2010	3	9	1	7

After this splitting and extracting, our train_cab dataset look like the above image. The new variables in train_cab dataset are:

Year, Month, Date, Date_of_week and Hour

We need to implement this thing to our test dataset, and our test dataset look like below image.

	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	Year	Month	Date	Day_of_week	Hour
0	2015-01-27 13:08:24	-73.973320	40.763805	-73.981430	40.743835	1.0	2015	1	27	1	13
1	2015-01-27 13:08:24	-73.986862	40.719383	-73.998886	40.739201	1.0	2015	1	27	1	13
2	2011-10-08 11:53:44	-73.982524	40.751260	-73.979654	40.746139	1.0	2011	10	8	5	11
3	2012-12-01 21:12:12	-73.981160	40.767807	-73.990448	40.751635	1.0	2012	12	1	5	21
4	2012-12-01 21:12:12	-73.966046	40.789775	-73.988565	40.744427	1.0	2012	12	1	5	21

Now the most important, need to calculate the distance travelled by cab from the pickup latitude & longitude and dropoff_latutude & longitude. To know how to calculate the distance from give data, I need to find formula.

After Googling, I found formula, named as **Haversine Formula.**

Also creating a new filed 'distance' to fetch the distance between pickup and drop location

We can calculate the distance in a sphere when latitudes and longitudes are given by Haversine formula

$\text{haversine}(\theta) = \sin^2(\theta/2)$

Eventually, the formual boils down to the following where φ is latitude, λ is longitude, R is earth's radius (mean radius = 6,371km) to include latitude and longitude coordinates (A and B in this case).

$a = \sin^2((\varphi B - \varphi A)/2) + \cos \varphi A \,.\, \cos \varphi B \,.\, \sin^2((\lambda B - \lambda A)/2)$

$c = 2 * \text{atan2}(\sqrt{a}, \sqrt{(1-a)})$

$d = R \cdot c$

d = Haversine distance

Here, we find out the distance using pickup_longitude, pickup_latitide, dropoff_longitude and dropoff_latitude by Haversine formula. After calculating our dataset get one more variable name as H_Distance. This implies to both train_cab & test dataset.

Image of test dataset with H_Distance variable.

	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	Year	Month	Date	Day_of_week	Hour	H_Distance
0	2015-01-27 13:08:24	-73.973320	40.763805	-73.981430	40.743835	1.0	2015	1	27	1	13	2.323259
1	2015-01-27 13:08:24	-73.986862	40.719383	-73.998886	40.739201	1.0	2015	1	27	1	13	2.425353
2	2011-10-08 11:53:44	-73.982524	40.751260	-73.979654	40.746139	1.0	2011	10	8	5	11	0.618628
3	2012-12-01 21:12:12	-73.981160	40.767807	-73.990448	40.751635	1.0	2012	12	1	5	21	1.961033
4	2012-12-01 21:12:12	-73.966046	40.789775	-73.988565	40.744427	1.0	2012	12	1	5	21	5.387301
5	2012-12-01 21:12:12	-73.960983	40.765547	-73.979177	40.740053	1.0	2012	12	1	5	21	3.222549
6	2011-10-06 12:10:20	-73.949013	40.773204	-73.959622	40.770893	1.0	2011	10	6	3	12	0.929601
7	2011-10-06 12:10:20	-73.777282	40.646636	-73.985083	40.759368	1.0	2011	10	6	3	12	21.540102
8	2011-10-06 12:10:20	-74.014099	40.709638	-73.995106	40.741365	1.0	2011	10	6	3	12	3.873962
9	2014-02-18 15:22:20	-73.969582	40.765519	-73.980686	40.770725	1.0	2014	2	18	1	15	1.099794

Here I also varify the actual distance I get through this formula and distance provided by website: https://www.geodatasource.com/distance-calculator based on longitude and latitude.

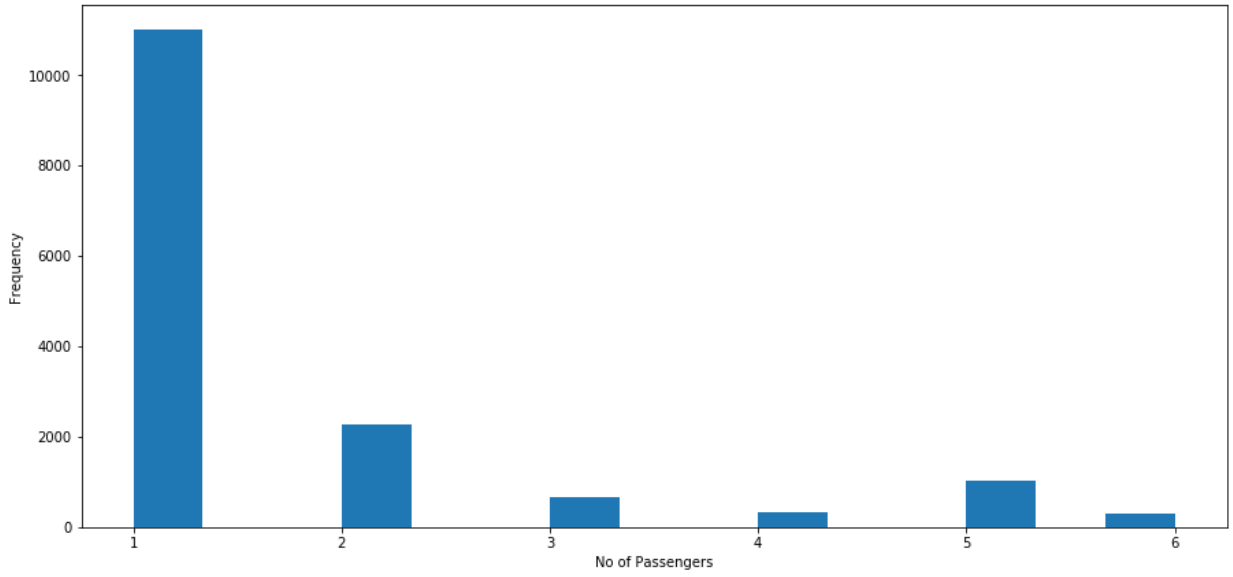
From this I conclude that I am on right track.

Now let's clarify assumptions one by one that I made earlier:

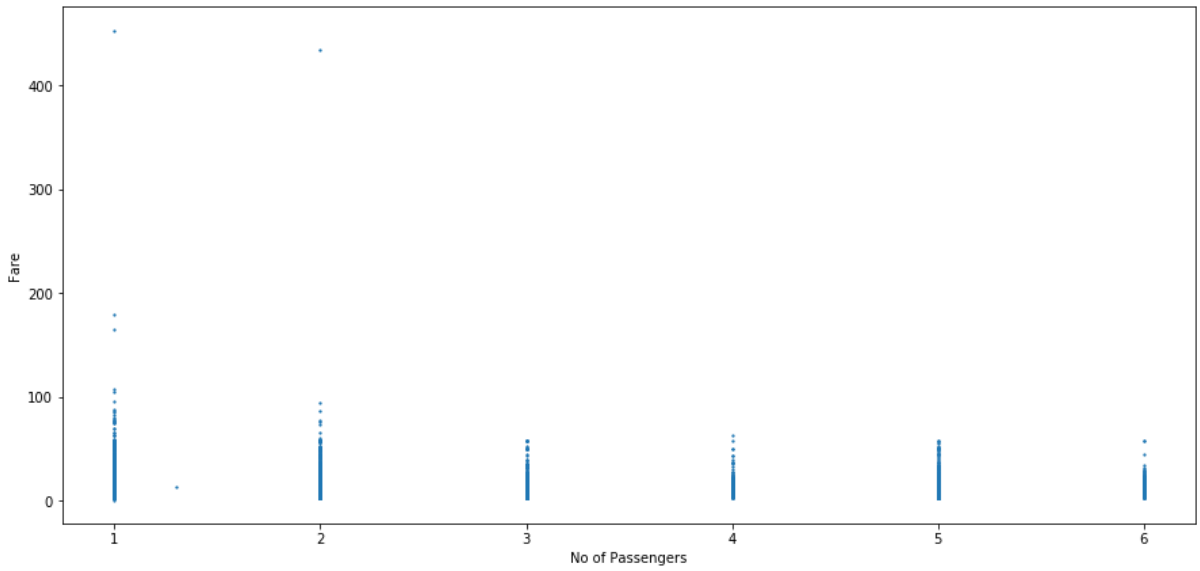
1) Does the passenger_count (Number of Passengers) affect the fare_amount (fare)?

Here, we plot the graph, Number of Passengers vs frequency and another graph, Number of Passenger vs fare

Number of Passengers vs Frequency



Number of Passenger vs Fare



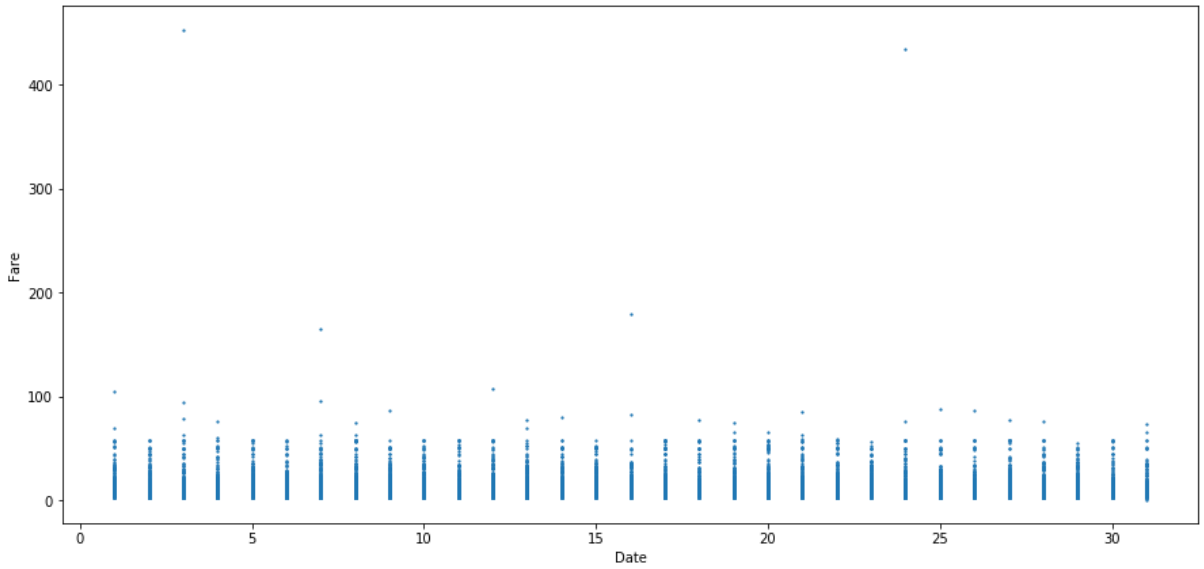
From above two graphs, we see that single passenger is most frequent travelers, the highest fare also seems to come from the cabs which carry just 1 passenger. Means we can say that, number of passenger won't make much difference in car fare. May be the fare amount will be same for 1 or more passengers.

2) Does the pickup_datetime (Pickup Date & Time) affect the fare_amount (fare)?

Many time we see, the timing plays an important role in amount calculations of any service. If you take service at odd hours of the day or during holiday, then amount or prices may vary. But this is the general thing, here we need get clarification from the dataset.

Let's plot the Date vs Fare graph

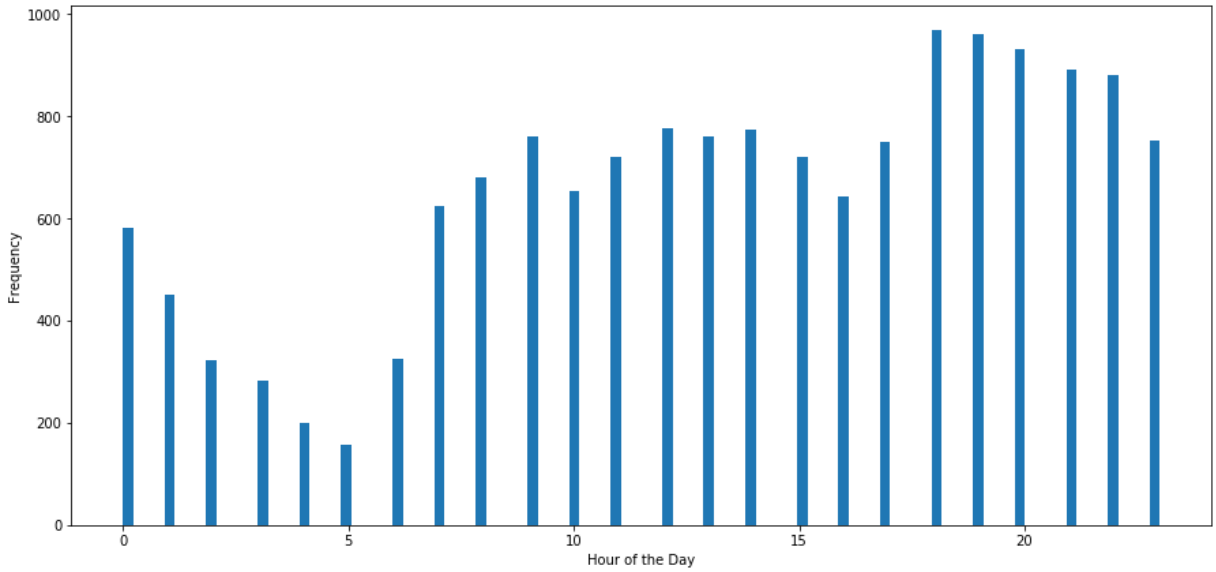
Date vs Fare



From the above graph, the fare_amount throughout the month is seem to be uniform, with maximum fare received on the 3rd of the month.

Another graph, Hour of the day vs frequency

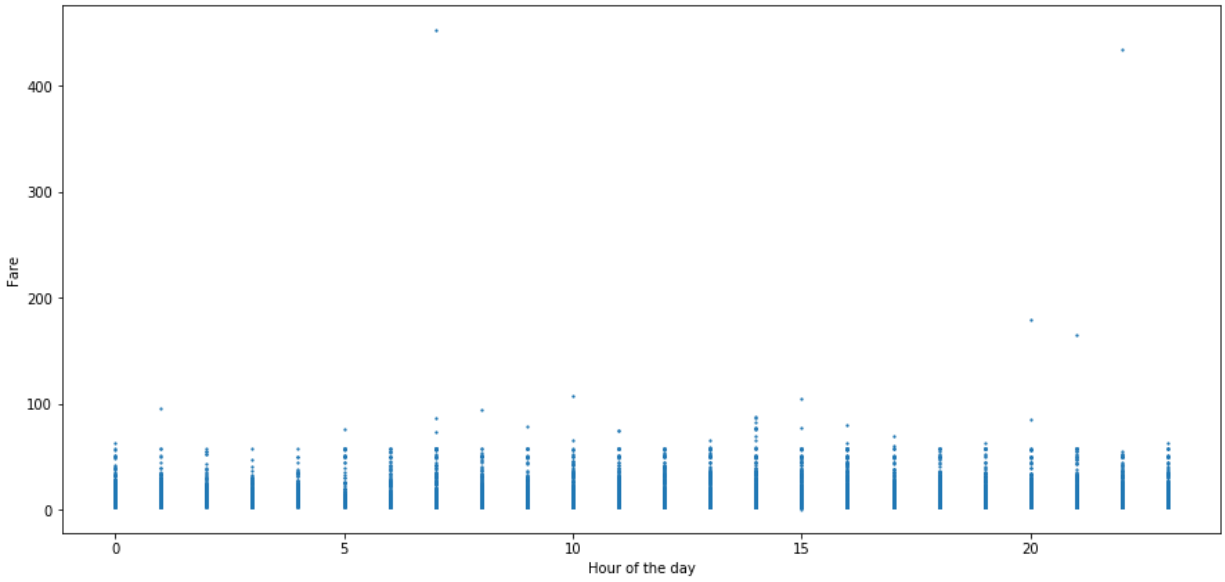
Hour of the Day vs Frequency



As the above graph shows, the time of the day plays an important role. The frequency of cab rides seem to be lowest at 5AM and the highest at 6PM. That is understandable. Many office work time over at 6PM, and people take the car to reach home. On the other hand many peoples are sleeping on 5AM that why the car used at 5AM is lowest.

Now we check the hour of the day will affect the fare_amount, plotting

Hour of the Day vs Fare

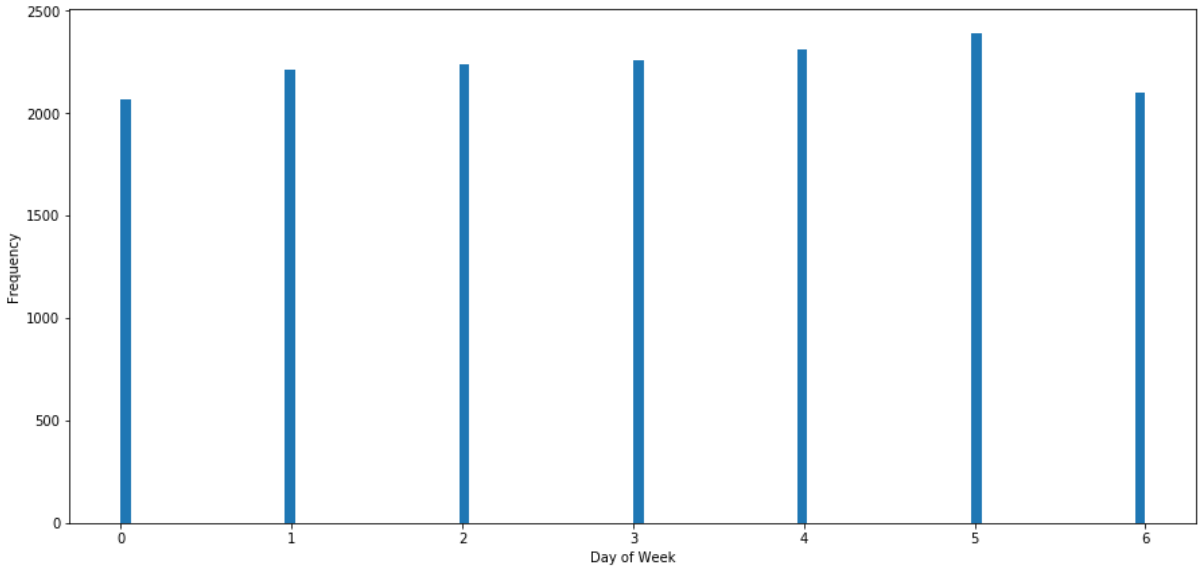


The fares, seem to be the high between 5AM to 10AM and 1PM to 4PM. Maybe people who leave early to avoid traffic and cover large distance. At the morning may be there is traffic due to rush hours, many office working time is 10AM to 6PM.

3) Does the day of the week affect the fare_amount (fare)?

First, we will understand the how many cars ride used at particular day of the week for that, we plot the graph Day of the week vs Frequency

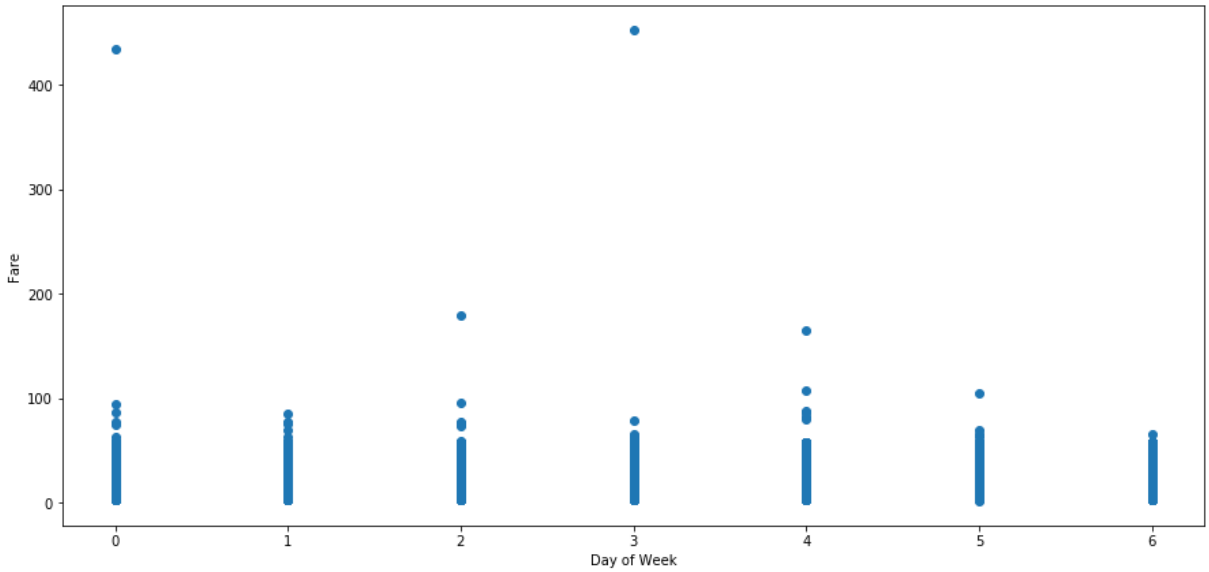
Day of the week vs Frequency



Note: The day of the week with Monday=0, Sunday=6.

The day of the week doesn't seem to have that much effect on number of cab rides.

Day of the week vs Fare

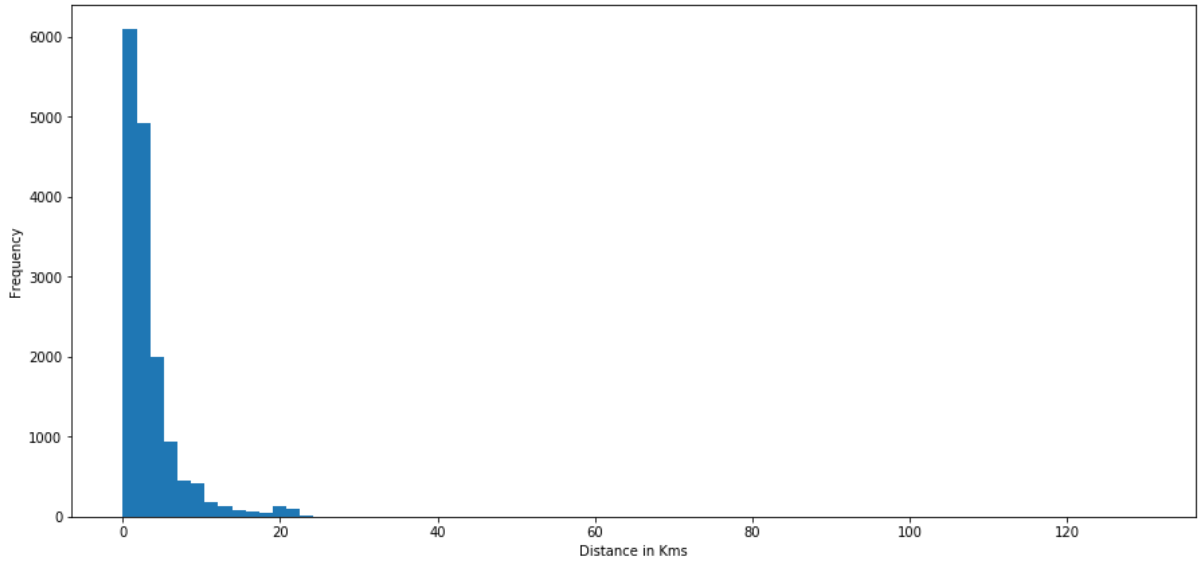


Above graph visualization not much affected by the day of week, the highest fares seem to be on Monday and Friday and the lowest on Thursday and Sunday. Maybe people travel far distances on Monday to reach offices and Friday to reach back home, hence there is high fares. Many people prefer to stay at home on Sunday or has low fare due to holiday of offices shut.

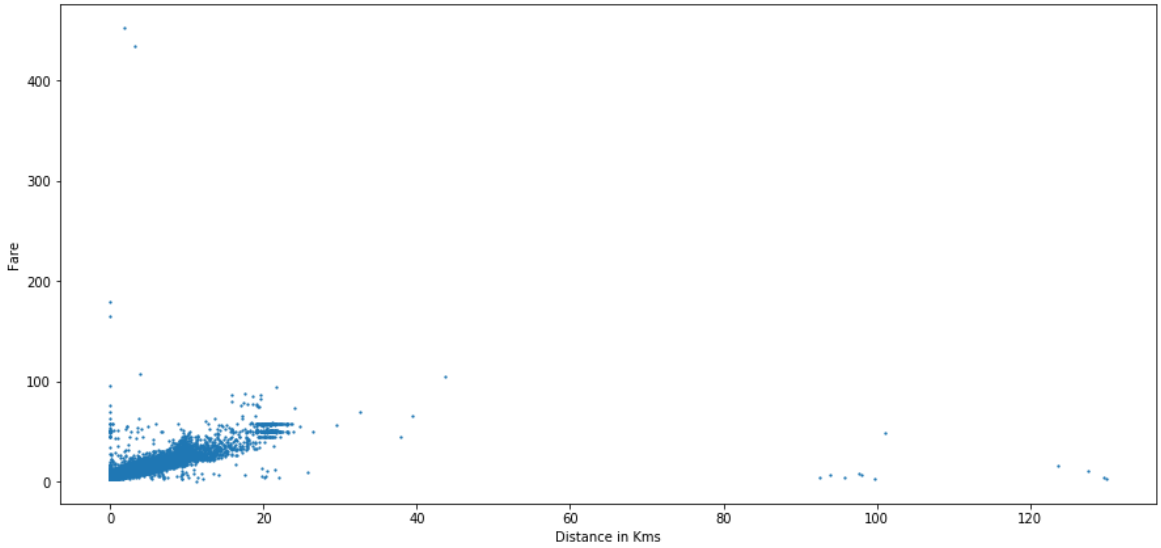
4) Does the distance travelled affect the fare_amount (fare)?

This is the obvious answer and we are confident about it that the distance travelled absolutely affect the fare_amount. But I will check with visualization

Distance vs Car Frequency



Distance vs Fare



There are values which are greater than 100Kms! As this data is from New York City, I am not sure why people would take a car to travel more than 100Kms. Since there is not much data present beyond 50Kms. So this is the outliers.

As we see, each variable has small and big effect on fare_amount variable which is our dependent variable. In that, Distance travelled is play an important role in calculating fare_amount.

For good modeling analysis, we need to remove the distance outliers. More than 50kms.

Also, we found that, some distances are showing 0 value. This may be because of the customer book the cab ride and cancel later on without travelling. In this scenario, the distance travelled is zero. So this is wrong data, because if customer won't travelled then he/she won't ready to pay the amount.

That why, I am considering H_Distance = 0 as outliers or wrong measurement and remove these observations from train_cab dataset and test dataset.

9. Modeling & Predictions:

Finally, Finally, Data Cleaning is done! Now let's build the model and predict the results

In machine learning there is two main types:

- **Supervised Machine Learning:** knowledge of output. Target Variable is fix
- **Unsupervised Machine Learning:** No knowledge of Output. Self-Guided Learning Algorithms.

Selecting model is main Part of Modelling, We have various model algorithms some of the basic algorithms are:

- **Linear Regression :** Best suitable for Regression Model
- **Logistic Regression:** Suitable for Classification Model
- **Decision Tree:** Best suitable for Regression & Classification model
- **Random Forest:** Mostly used for Classification model analysis but can be used for Regression model
- **KNN algorithms:** Can be used for Regression and Classification model
- **Naive Bayes:** used for Classification Model

Currently we are dealing with Regression Model, so we are considering following algorithms:

- Linear Regression
- Decision Tree
- Random Forest
- KNN Regression Algorithms

9.1 Model Evaluation:

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future.

For classification problems, we have only used classification accuracy as our evaluation metric. But here we used Error Metrics to evaluate the model.

After building a number of different regression models, there is a wealth of criteria by which they can be evaluated and compared. List of error metrics are:

Mean Absolute Error (MAE): is the mean of the absolute value of the errors: In [0,∞), the smaller the better.

Mean Squared Error (MSE): is the mean of the squared errors: In [0,∞), the smaller the better.

Mean Absolute Percent Error (MAPE): is the mean of the absolute percent value of the errors: In [0,1] the smaller the better.

Root Mean Squared Error (RMSE):is the square root of the mean of the squared errors: In [0,∞], the smaller the better.

For more clarification,

- MAE gives less weight to outliers means it is not sensitive to outliers.
- MAPE is similar to MAE, but normalized the true obeservations. When true observation is zero then this metric will be problematic
- MSE is a combination measurement of bias and variance of predictions. It is more popular.
- RSME is square Root of MSE, Root Square is taken to make the units of the error be the same as the units of the target. This measure gives more weight to large deviations such as outliers, since large differences squared become larger and small (smaller than 1) differences squared become smaller.

Selection: Out of these 4 error metrics, MSE and RMSE are mainly used for Time-Series dataset. As I know, current working data is not a time depended or time-series data.

For that Reason the Model Evaluation is based on **MAPE Error Metrics**.

But here, we check all error metrics of each model we consider.

Before going for the algorithm testing, we need to prepare train and test dataset from the train_cab, these train and test datasets are different from our main train_cab and test dataset.

This can be achieved by splitting train and test dataset from our train_cab dataset. In that we do,

- Preparing training dataset,
- Applying the same preparation on test dataset
- And finding the error metrics to evaluate the algorithms

To simulate a train and test set we are going to split randomly train_cab dataset into 80% train and 20% test.

Later on we check, we predict the fare_amount output of the test set and compare it with our actual fare_amount of test set.

9.2 Algorithm Modeling

First model we built is Linear Regression Model:

1) Linear Regression

This is the simplest form of regression model which describe relation among variables. The one simple case is where a dependent variable may be related to independent variable or explanatory variable.

Key assumptions for Linear Regression mode used:

- Linear Relationship
- Multivariate normality
- No or little multi-colinearity
- No auto correlation

After building the model, A few things, I learn from this output

- pickup_longitude, pickup_latitude, dropoff_latitude, year, month, hour, H_distance have small p-values, whereas dropoff_longitude, passenger_count, date have a larger p-values

- Here I reject the null-hypothesis for pickup_longitude, pickup_latitude, dropoff_latitude, year, month, hour, H_distance
    - There is association between these variables and fare_amount

- Fail to reject the null hypothesis for dropoff_longitude, passenger_count, date
    - There is no association

R-squared (0.848) means this model provides better fit for the given data

But selecting the model with the highest R-squared is not a reliable approach for choosing the best linear model.

Dep. Variable:	y	R-squared:	0.848
Model:	OLS	Adj. R-squared:	0.848
Method:	Least Squares	F-statistic:	6257.
Date:	Mon, 12 Aug 2019	Prob (F-statistic):	0.00
Time:	16:24:58	Log-Likelihood:	-39437.
No. Observations:	12323	AIC:	7.890e+04
Df Residuals:	12312	BIC:	7.898e+04
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
pickup_longitude	12.3931	1.419	8.736	0.000	9.612	15.174
pickup_latitude	9.6711	1.969	4.911	0.000	5.811	13.531
dropoff_longitude	0.1102	1.382	0.080	0.936	-2.599	2.819
dropoff_latitude	-10.8352	1.791	-6.049	0.000	-14.346	-7.324
passenger_count	0.0105	0.042	0.250	0.803	-0.072	0.093
Year	0.4851	0.028	17.537	0.000	0.431	0.539
Month	0.0721	0.016	4.626	0.000	0.042	0.103
Date	-0.0068	0.006	-1.107	0.268	-0.019	0.005
Day_of_week	-0.0272	0.027	-0.998	0.319	-0.081	0.026
Hour	0.0046	0.008	0.552	0.581	-0.012	0.021
H_Distance	2.2069	0.015	145.387	0.000	2.177	2.237

Omnibus:	34886.856	Durbin-Watson:	1.996
Prob(Omnibus):	0.000	Jarque-Bera (JB):	3555322451.300
Skew:	36.997	Prob(JB):	0.00
Kurtosis:	2633.360	Cond. No.	9.16e+04

Error Metrics of Linear Regression

	Python	R
MAE	2.309174314217753	2.153135
MSE	26.008942624250416	13.82451
MAPE	0.21449896785866907	0.2162309
RMSE	5.099896334657246	3.718132

2) Decision Tree Regression Algorithms

In decision tree regression algorithms the predictions is based on branching series of Boolean tests. There are number of different types of decision trees that can be used in Machine Learning Algorithms. Decision Tree is a rule. Each branch connects nodes with "and" and multiple braches are connected by "or".

Error Metrics of Decision Tree Regression Algorithms

	Python	R
MAE	2.9550429514573167	2.581969
MSE	30.4739633637039	18.36605
MAPE	0.3068516321964189	0.2577115
RMSE	5.520322759015445	4.285564

3) Random Forest Regression Algorithms

Random Forest is an ensemble that consists of many decision trees. The method combines Breiman's "bagging" idea and random selection of features. This algorithms can be used for regression problem.

	Python	R
MAE	2.1157140798442025	1.951885
MSE	35.247137087961804	15.31187
MAPE	0.21443544640231504	0.2085321
RMSE	5.936929937936088	3.913038

4) KNN Regression Algorithms

K stands for K-Nearest Neighbors. It is the simple algorithms that scores all variable cases and classifies new cases based on similarity measures. It is the lazy learning algorithms.

	Python	R
MAE	2.7093073677377477	2.520559
MSE	31.841585311262577	18.97416
MAPE	0.2727668612728445	0.2618314
RMSE	5.642834864787607	4.355934

9.3 Selecting best suitable model for final analysis

As we discussed earlier, we are considering the MAPE for model evaluation because, it calculate average absolute percent error for each time period minus actual values divided by actual values.

Reason I already explain, lets explain again:

Selection: Out of these 4 error metrics, MSE and RMSE are mainly used for Time-Series dataset. As I know, current working data is not a time depended or time-series data.

Comparing the MAPE error metrics of all the algorithms

MAPE	Python	R
Linear Regression	0.21449896785866907	0.2162309
Decision Tree Regression	2.9550429514573167	2.581969
Random Forest Regression	0.21443544640231504	0.2085321
KNN Regression	0.2727668612728445	0.2618314

As we see from above table, Random Forest Regression Algorithms has lowest MAPE values. That the reason we reject other algorithms and accept Random Forest Algorithms for future predictions. As we get more data, our Random Forest Regression Algorithms MAPE will reduces and it will be more suitable for future dataset.

Finally, we implement the developed model algorithms on our processed test dataset and prepare the final output results.

10. Conclusion and Future Work

Considering what is and what not considered for models built in this project, their predicting results are fairly accurate. To further improve the prediction accuracy, more variabilities need to be considered and modeled. Although car ride duration to get understanding of traffic. Also, modeling traffic and the effect of location in between pickup and drop-off points should be considered as well as difference in drivers' speed.

To get more accurate results, analyze the larger dataset to infer relationships and effects of location and traffic at different times. Also consider the speed limitation and traffic data etc.