Cab Fare Prediction

(A model development to predict the cab fare)

by

Surjeet Singh

Index

Chapter 1
Introduction

1.1 Problem Statement

The given problem provides a data of pilot project for different parameters for a cab company. The company willing to launch the project on a larger scale and for that, the right prediction of the cab fare is required. We have to develop a machine learning model to solve this problem so that accurate prediction of the cab fare can be done.

1.2 Data Summary

Given data for training has 16067 rows and 7 columns. The seven columns are as follow fare_amount, pickup_datetime, pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude, passenger_count.

	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

Using this data, we have to develop a model to predict the fare amount correctly. We can create more data using the given data in order to feed the model the appropriate data. For example, using the pickup coordinates and drop-off coordinate distance between the pick point and drop off point can be determine which is better data to feed the model.

Chapter 2
Pre-processing

Machine learning models accept the structured data with proper column and rows. But in real life the data is unstructured. In the pre-processing techniques we analyze the data thoroughly and the clean the data so that the data can feed into a machine learning models.

2.1 Analyzing the raw data

Just take look at the raw data so that we can thoroughly analyze the data. We can use the describe function for it.

	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
count	16067.000000	16067.000000	16067.000000	16067.000000	16012.000000
mean	-72.462787	39.914725	-72.462328	39.897906	2.625070
std	10.578384	6.826587	10.575062	6.187087	60.844122
min	-74.438233	-74.006893	-74.429332	-74.006377	0.000000
25%	-73.992156	40.734927	-73.991182	40.734651	1.000000
50%	-73.981698	40.752603	-73.980172	40.753567	1.000000
75%	-73.966838	40.767381	-73.963643	40.768013	2.000000
max	40.766125	401.083332	40.802437	41.366138	5345.000000

As we can see from the above that it reveals a lot about the data. For example the maximum number of passenger in the cab is which is not possible for the cab because maximum sitting capacity of the cab is 6. We have to clean each for such entries but before that just take check of missing values or NA's for each column and remove them.

```
df_train.isna().sum()
```

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

So there are 24 NA's in fare_amount column and 55 NA's in the passenger_count column. We have to remove them because the model does not accept the missing values of NA's. 69 entries having NA's have been removed. We have other options to handle it such as fill the missing values with mean or median or KNN. But for now we are removing them.

Machine learning models accept the numerical values of categorical values. Checking the data type for the given data so that we can convert the data into its appropriate type.

```
df_train.dtypes
```

```
fare_amount          object
pickup_datetime      object
pickup_longitude    float64
pickup_latitude     float64
dropoff_longitude   float64
dropoff_latitude    float64
passenger_count     float64
dtype: object
```

The fare_amount column must have numerical values but it is given in the format of a string object. Because it consist strings so it may contain special characters or any non numerical characters. Pick_datetime column is also need to modify and we will extract the year, month, date and hour in the separate column so that we can use them for our analysis.

2.2 Outlier Analysis and Data Cleaning

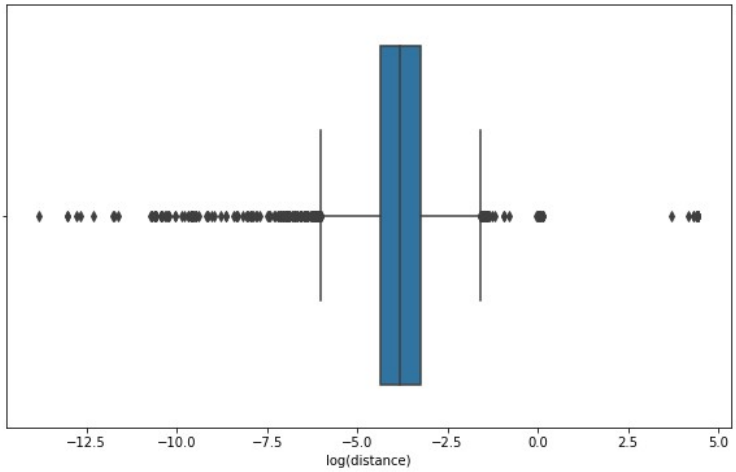
As we have discussed earlier that the real life data is not well structured. It consists of a lot of values which are not meaning full. If the column contains few relatively high or low values, those values are considered as outliers. Outlier analysis is completely objective so it varies from problem to problem. In our case, we don't need outliers so we have to remove them.
We have applied the following filters on the data:

- Zero fare or negative fare is not possible
- Passenger count must lies between 1 to 6
- Passenger count must be integer values
- Values of latitude are restricted between -90 to 90
- Values of longitudes are restricted between -180 to 180

After applying these filters 381 entries have been removed from the data. Similar analysis is also done for the test data.

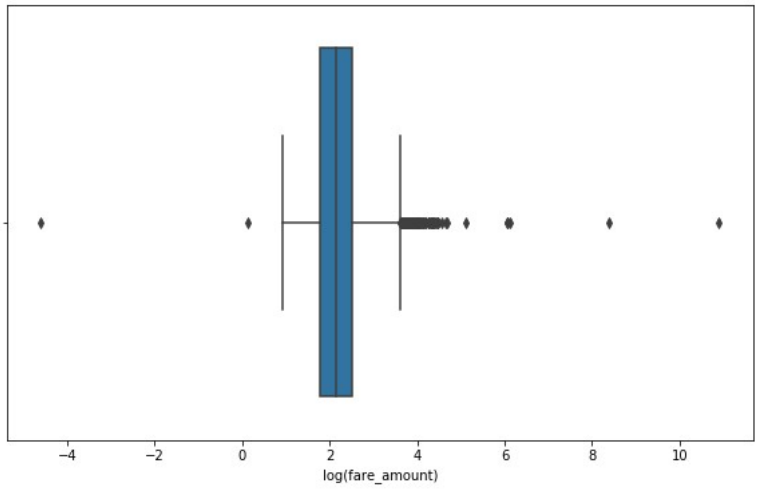
Now, we can add a new column to our data. We have added a new column with the name distance. The distance is calculated by using the Euclidean distance formula. After adding the new column, the entries having 0 distance has been removed because 0 distance is not a meaningful information for our model.

distance
1.515900e+04
1.578993e-01
3.167522e+00
1.000000e-06
1.303066e-02
2.217934e-02
3.932840e-02
8.448094e+01



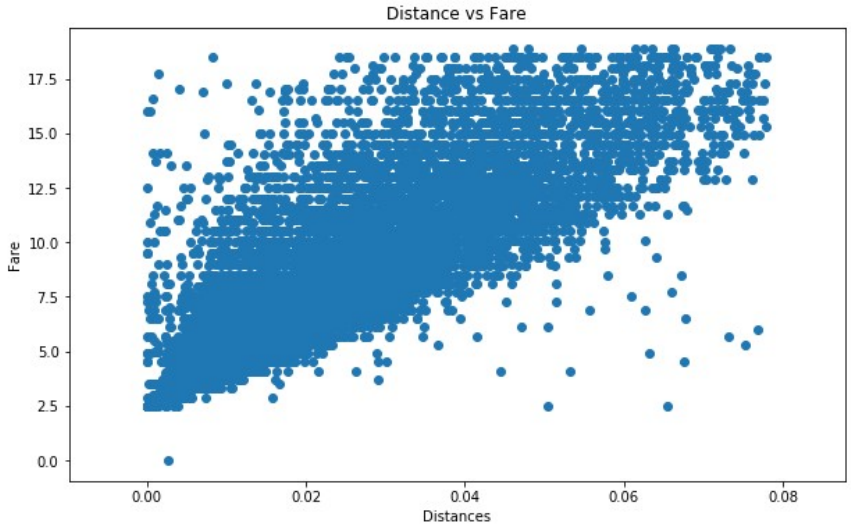
It is observed that the distance column itself have outliers. The distance contains very high values but the most of the distance column has smaller values. We have removed the entries lying outside the upper fence and the lower fence. 227 instances are removed in this cleaning step.

In the similar way, the fare amount column also contains few very large values in comparison to its most of the values. These values may affect the model significantly so we need to remove them too. 1326 entries have removed in the data cleaning step.

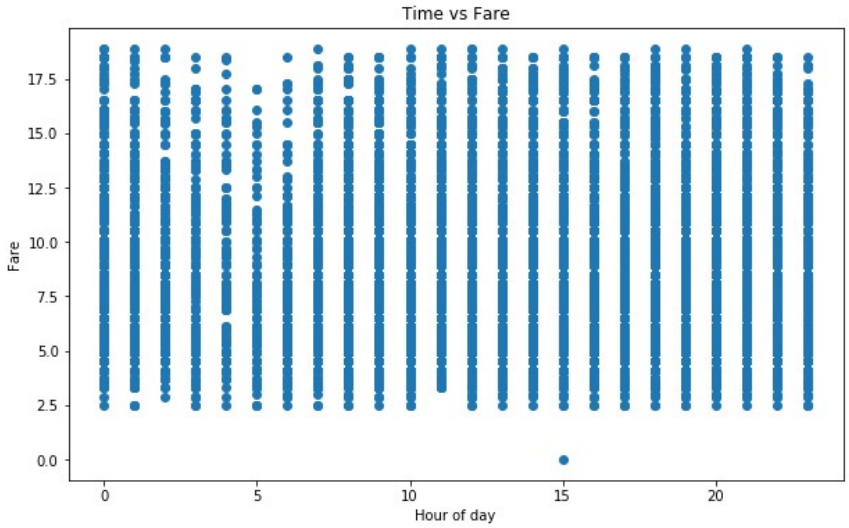


2.3 Plots for each Column

Analyzing each column by plotting its graphs is very important because it given the idea about the variation of the target function. It also roughly gives the importance of the particular feature in our prediction. We can avoid the columns to feed the model having the insignificant variation with the target function.



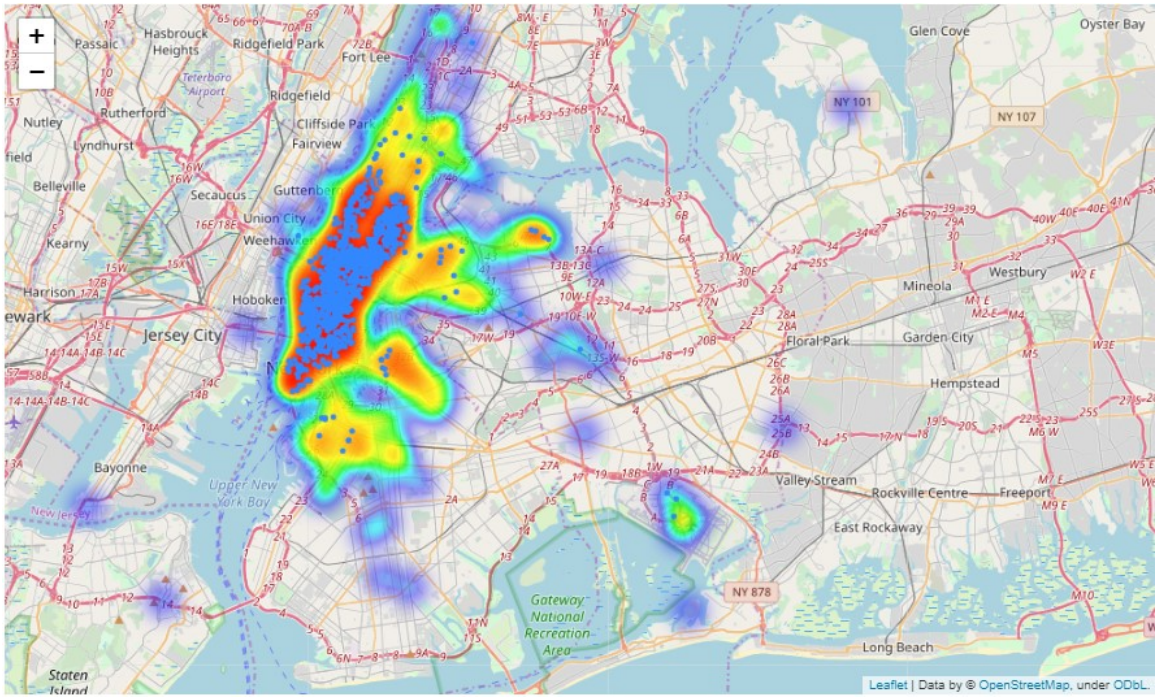
It can be concluded from the above plot that the fare amount is increasing with the distance which is quite obvious. We have to include that feature to the machine learning model because it plays an important role in the fare prediction.



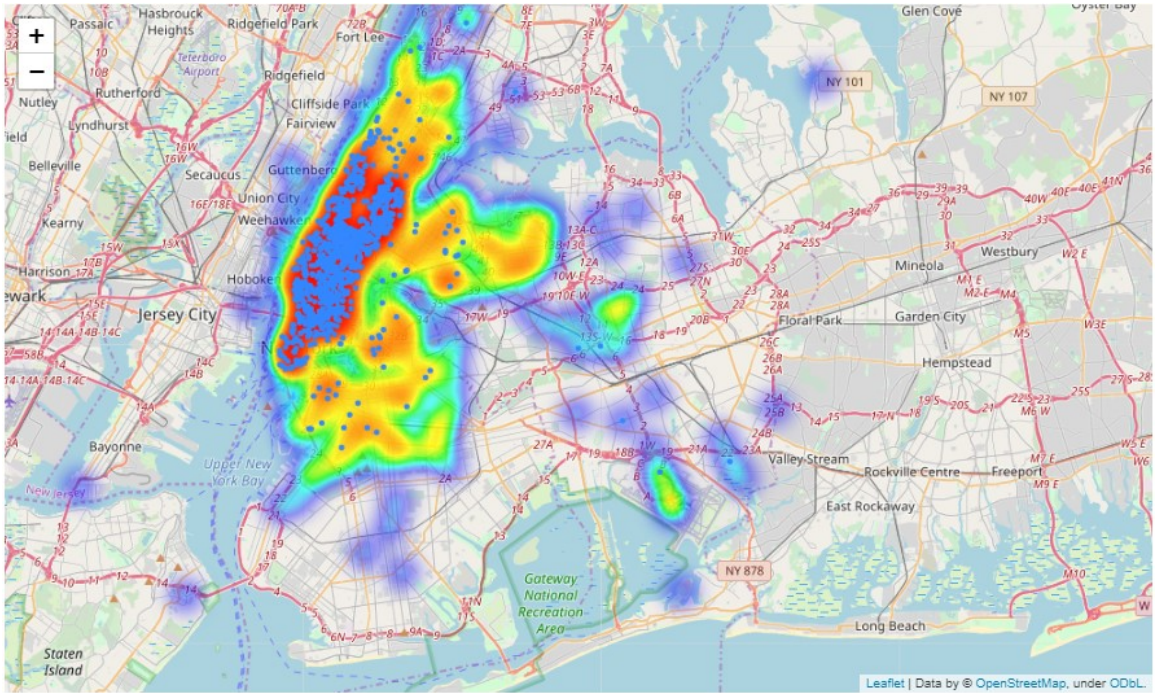
The fare amount is spread in similar manner for every hour of the day. Therefore, the importance of this feature is less and may have less impact on our prediction.

The pickup coordinates and drop off coordinates column can be visualized on the map to understand in better way. We can observe that which region most frequent to pick and drop. The python have library named “folium” to plot the coordinates on the map in the form of heat map.

We can avoid the pickup coordinates and drop off coordinates to pass into the machine learning model because these columns are used to create the new column with the name distance. Distance is much significant in the prediction.

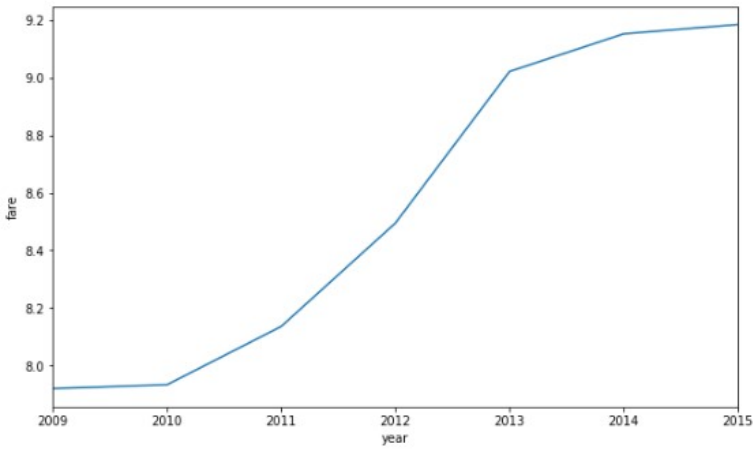


Pickup coordinates

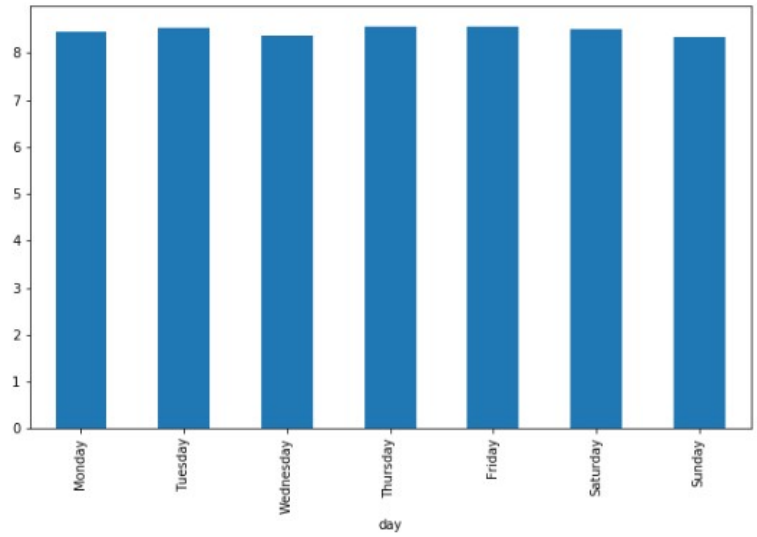


Drop off coordinates

Drop off coordinates have more spread than the pickup coordinates but both of the coordinates are most concentrated in the same region. The given coordinates are corresponding to Manhattan.

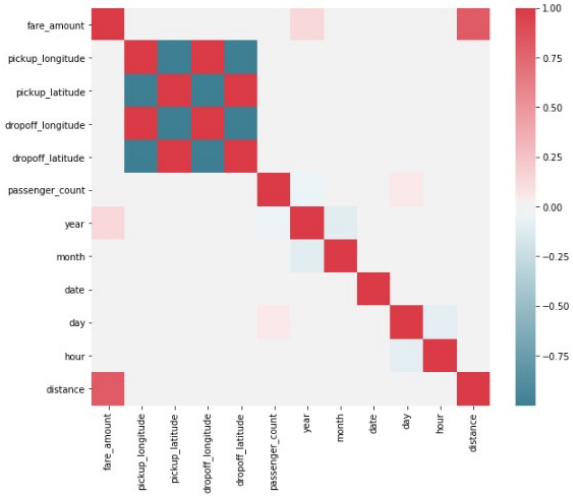


The average fare amount is increasing as the year increase. The increase in the fare amount with year is may be due to the inflation in the market.



The average fare amount with respect to the day of the week is almost same. Similar behavior is also observed in the months.

2.4 Correlation Plot



The above plot is called correlation plot which tell us the correlation for every possible pair of the features. It can be concluded from the correlation plot that the most important feature are year and the distance. As the year increase, the fare amount is slightly increases due to inflation because it shows a small positive correlation between the year and the fare amount. The positive correlation between the distance and the fare amount is obvious and discussed earlier.

Chapter 3
Machine Learning Models

In this chapter, we are going to discuss the different which can be trained on the given data and then predict fare using the given test data. We will begin with the simplest Linear Regression model and then move towards the advanced models like Random Forest, GridSearchCV etc.

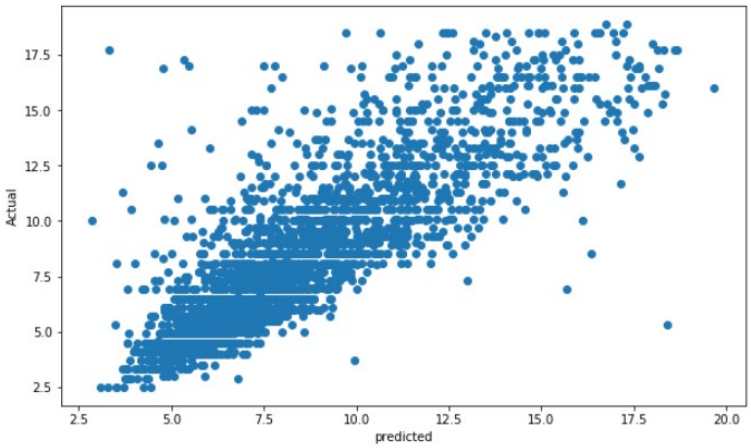
3.1 Features Choice & Splitting the Test Train Data

After cleaning the data and before applying the model to it, it is necessary to chose the feature properly and then split the data as train data and test data. Train data is used to train the model and then test data is used to test the accuracy of our trained model. Usually, 80% of the data assigned to the training data and 20% of the data is assigned to the test data. While splitting the data X stands for feature and Y stands for the target. We chose "distance", "passenger_count", "year" ,"hour", and "day" as the feature and “fare_amount” as the target because it is our objective our problem statement.

```
X = df_train[["distance","passenger_count","year","hour","day"]]
Y = df_train["fare_amount"]
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
```

3.2 Linear Regression Model

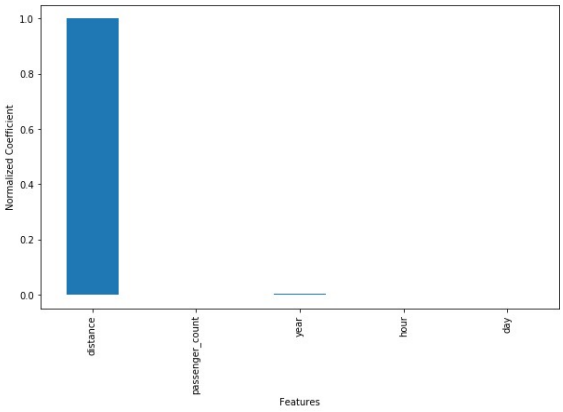
Linear Regression is the simplest model to apply and it fit a straight line to our data by adjusting the intercept and the slop of the line. After fitting the linear regression model, we predict the fare amount using the test data and then calculating the error to check the accuracy of our model. We chose RMSE to check the accuracy of the model. The RMSE of our model is 1.97. To check whether the model is under fitted or over fitted, we have to determine the RMSE for both the test data and the train data. The value of RMSE for test data and the train must be close to each other. The RMSE for the train data is also 1.97 which means our model is fitted well.



The above plot is between the predicted fare amount and the actual fare amount. The linear behavior of the graph shows the good predictions but still there are some predictions which are off the pattern and we have to rectify them in our further models.

Linear regression model can also tell us the coefficient for each feature. The coefficient of the feature shows the impact of that feature on the prediction and we can retrain our model by eliminating the feature having very low value of the coefficient.

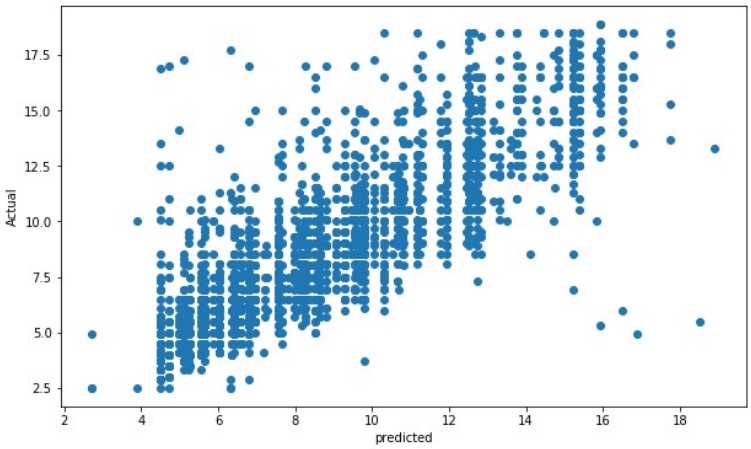
	Coefficient	Normalized Coefficient
distance	195.036463	1.000000
passenger_count	0.045475	0.000233
year	0.322265	0.001652
hour	0.011788	0.000060
day	-0.018547	-0.000095



This dataframe shows the coefficient and the normalized coefficient for each feature used to train the model. It is very clear from the data that the distance is the most important feature because it has maximum impact and all the other features have very low impact on the prediction.

3.3 Decision Tree Model

Decision Tree is very famous model in the machine learning and can be used for both classification as well as regression. We have fitted the Decision Tree model on the same data set used in the linear regression model. The RMSE error of the Decision Tree model is 1.95. The Decision Tree has slightly lower error than the linear regression model but there is no significant improvement in the model. The RMSE for the training data is 1.81 which is lower than the RMSE for the test data. This shows characteristics of over fitting the model because the error for the training data is lower than the test data.



The above graph shows the plot between the fare predicted from the Decision Tree model and the actual value of the fare. The spread of the predicted and the actual fare is expected to be lower for a better model. Further we will apply more advanced model. One of the advanced models is Random Forest. The parameters of this model can be fine to achieve better accuracy by using the Cross Validation.

3.4 Random Forest Model

Random Forest Model is based on Decision Tree model. It train the number of Decision Tree models and then combine their result to get a better accuracy of the model. We can control the number of the Decision Tree to be trained in the Random Forest so that the best number of Decision Trees to be trained. GridSearchCV help us to fine tune the parameters of the any model by cross validating for every possible combination of the parameter given to it. The RMSE for Random Forest is comes out to be 1.87. It shows a significant improvement in the prediction. But the error for the training data is 1.71 which is lower than the test data. It again shows the over

fitting of the model. This over fitting may occur because of the over fitting of the decision tree itself.

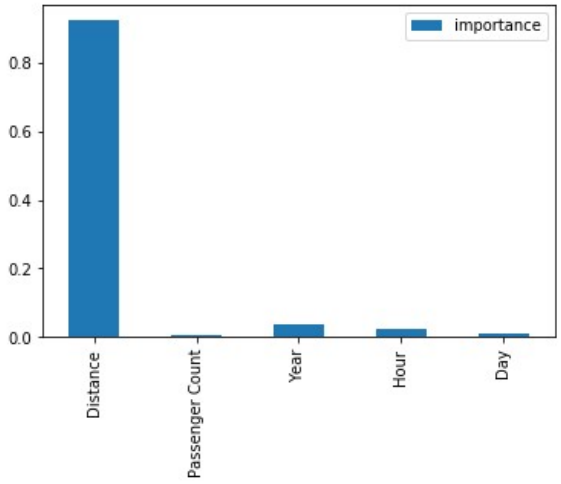
Using the GridSearchCV we can determine the best parameters to be work with.

```
grid_search.best_params_
```

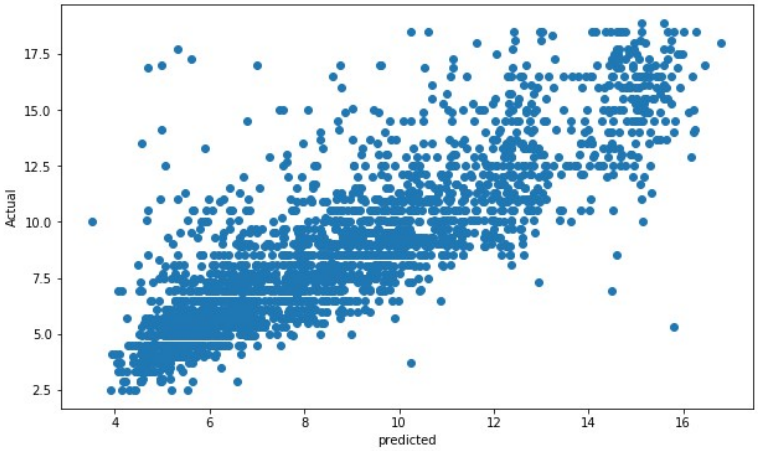
```
{'max_depth': 8, 'max_features': 3, 'n_estimators': 28}
```

The Random Forest is able to determine the feature importance of each parameter given to it. It equivalent to the coefficient of Linear Regression model which also tell the impact of the feature on the predictions.

	importance
Distance	0.921693
Passenger Count	0.004857
Year	0.039390
Hour	0.024547
Day	0.009513



It is again clear that the most important feature for fare prediction is the distance only and the year also impact a little bit in the predictions because as year increase the inflation affect the prices.



It can be observed from the graph that there is a little bit less spread in the graph which is a sign of good prediction. Sometimes the plots look similar but we have already calculated the RMSE for each model. RMSE is minimum for Random forest model.

Chapter 4
Conclusion

We have used three models to train which are Linear Regression, Decision Tree and Random Forest. The errors of the corresponding models are 1.97, 1.95 and 1.85. The error is lowest for the Random Forest but there is no much improvement even after applying the advanced model. It can happen because the prediction is limited to the one feature only which is distance. All the other features have almost no effect on the prediction except year. But impact of the year in comparison to the distance is negligible. As from the error analysis, it can be concluded that the Linear Regression fit well to the model because the distance feature vary linearly with the fare amount. Other two models have shown a little bit over fitting. We can go with any of the model because all of the models have similar error. We can prefer the linear regression model because of its simplicity.