Project:

CAB FARE PREDICTION

Edwisor

Submitted By: Gautam.N.Pai

Dec 9, 2019

1. Introduction

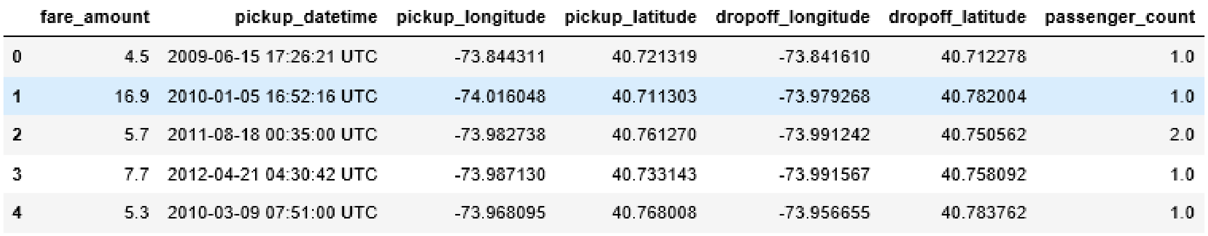
1.1 Problem Statement:

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

The objective of the project is to predict the fare amount based on some parameters like geo coordinates (latitude, longitude), passenger count and pickup datetime. The goal is to build up a regression model that can successfully predict the fare of rentals on the above factors.

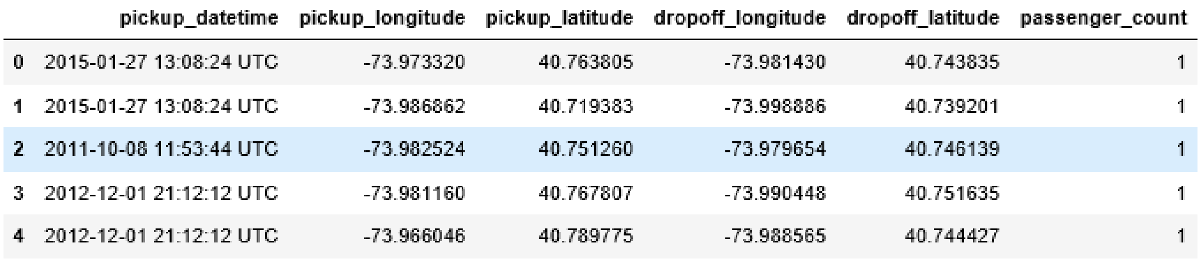
1.2 Data

Preview of Train Data

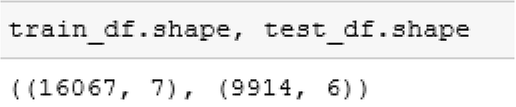


	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
0	4.5	2009-06-15 17:26:21 UTC	-73.844311	40.721319	-73.841610	40.712278	1.0
1	16.9	2010-01-05 16:52:16 UTC	-74.016048	40.711303	-73.979268	40.782004	1.0
2	5.7	2011-08-18 00:35:00 UTC	-73.982738	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

Preview of Test Data



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



```
train_df.shape, test_df.shape
```

```
((16067, 7), (9914, 6))
```

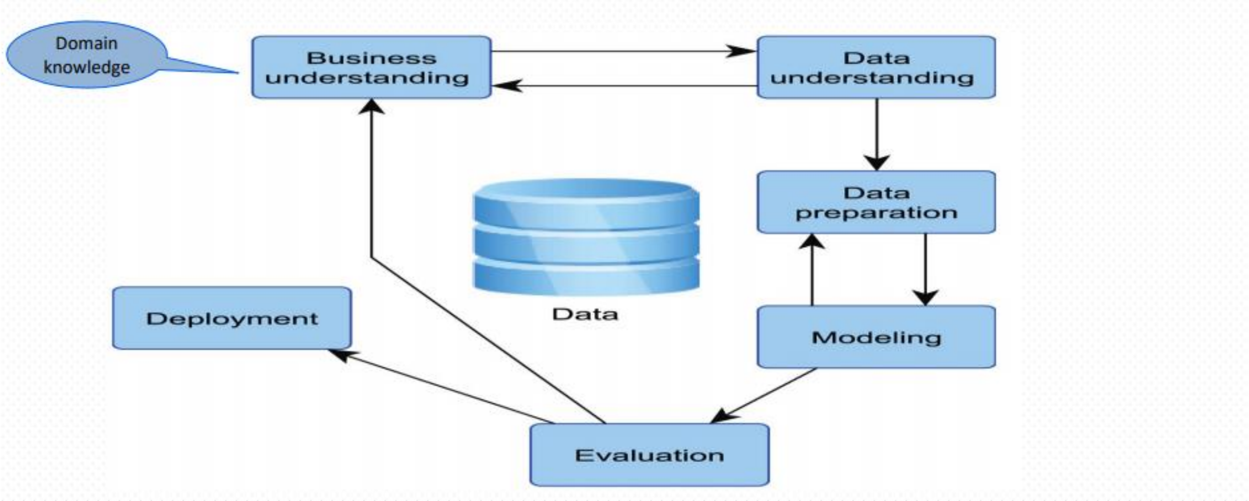
Here our response variable is fare_amount which we need to predict on the test data. The independent variables are pickup_datetime, pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude and passenger_count

2 METHODOLOGY:

2.1 CRISP-DM Process:

CRISP-DM stands for cross industry process for data mining. It includes 6 major phases

1. Business Understanding
2. Data Understanding
3. Data preparation
4. Modeling
5. Evaluation
6. Deployment



2.2 Data Preprocessing

Data preprocessing involves converting raw data into an understandable format. Real data is often incomplete, inconsistent and lacking in certain behavior/trends. Data preprocessing prepares raw data for further processing. The model accuracy and success of the metrics depends entirely on the Data preprocessing done. This accounts for 80% of any Data Science project. The following are the data pre

processing steps used for this project

1. Data Exploration
2. Missing Value Analysis
3. Outlier Analysis
4. Feature Engineering/Extraction
5. Feature Selection
6. Feature Scaling

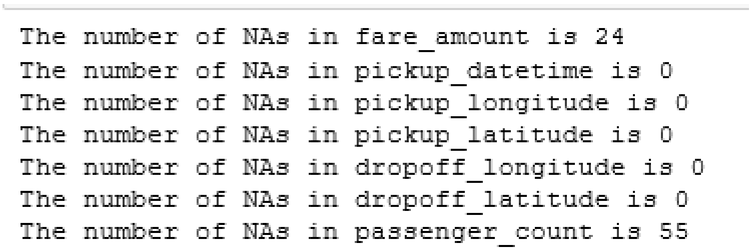
2.2.1 Data Exploration

In this stage, we get familiar with the data by identifying the class types of the variables and the data distribution of each and every variable like max, min, 1st quarter, 3rd quarter etc. The features could be continuous, discrete or categorical. Based on the data we do appropriate conversions. Also the impact of each variable to business case is determined and finding out data anamolies. In our project we determined the following anamolies

1) Invalid pickup date
2) Incorrect latitude/longitude entries (Latitudes range from -90 to +90 and longitudes range from -180 to +180)
3) Similar pickup and dropoff coordinates (This could be due to roundtrip or cancelled bookings)
4) Negative passenger_count, an unrealistic passenger_count or passenger_count with decimal values

2.2.2 Missing Value Analysis

In this stage we identify the missing values in each and every feature and gather the count of all the variables. Missing values can arise due to many reasons like business unable to capture the data due to privacy or error in capturing the data. Missing values are present as NA/NAN. The rule of thumb is if the missing values are more than 30%, it is advisable to drop that variable. If it is below to some percentage based on business scope, we can either drop those samples or else impute those values with either mean, median, knnimputation, random sampling imputation



```
The number of NAs in fare_amount is 24
The number of NAs in pickup_datetime is 0
The number of NAs in pickup_longitude is 0
The number of NAs in pickup_latitude is 0
The number of NAs in dropoff_longitude is 0
The number of NAs in dropoff_latitude is 0
The number of NAs in passenger_count is 55
```

In our project there were missing values in fare_amount and passenger_count variables. There were no missing values in the scoring dataset.

There are 3 types of missing values

1. MCAR (Missing Completely at Random)
2. MAR (Missing at Random)
3. MNAR (Missing Not at Random)

MCAR

The nature of missing values is not completely related to any of the variables whether missing or observed.

MAR

This means that the nature of the missing values is related to the observed data but not the missing data

MNAR

They exist when the missing data is neither MCAR or MAR. The missing values on the variables are related to that of both the observed and unobserved variables.

Which missing value imputation technique to be considered?

The rule of thumb says to identify which of the above category falls into and to select a missing value imputation technique which preserves the variance after the imputation and also preserves the distribution of the variable.

In Python: We tried with mean, median, knnImputation and constant value of a tail end of the distribution and we found that Imputation of a tail end with a value of mean + sd works well and preserves the variance

In R: We have several packages to deal with missing values and they are

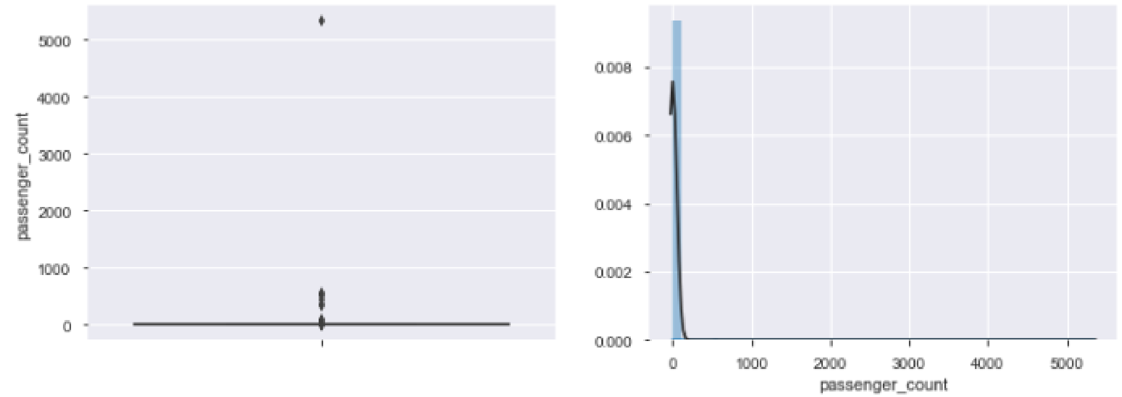
A. MICE (Multivariate Imputation via Chained Equations)
B. AMELIA
C. Hmisc
D. Mi
E. missForest

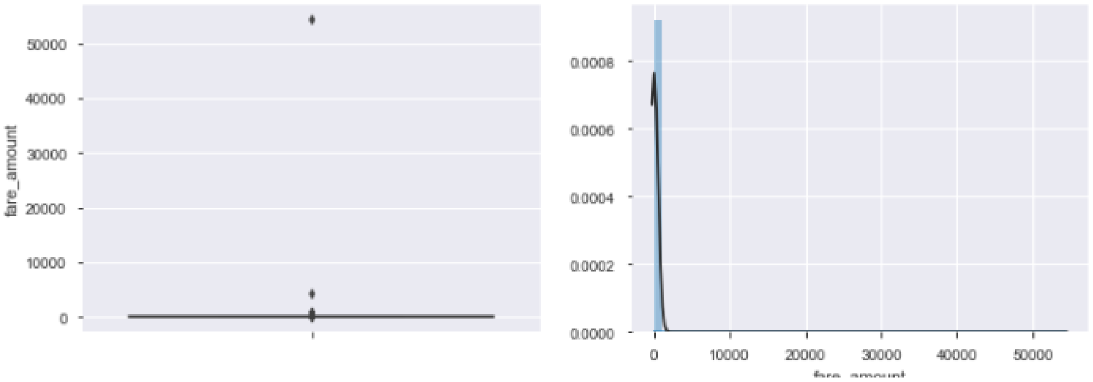
We tried almost all of the packages and we got better results with MICE since it imputes based on the values of other variables (MAR) in that sample (chained equations) and uses regression techniques in imputing missing values.

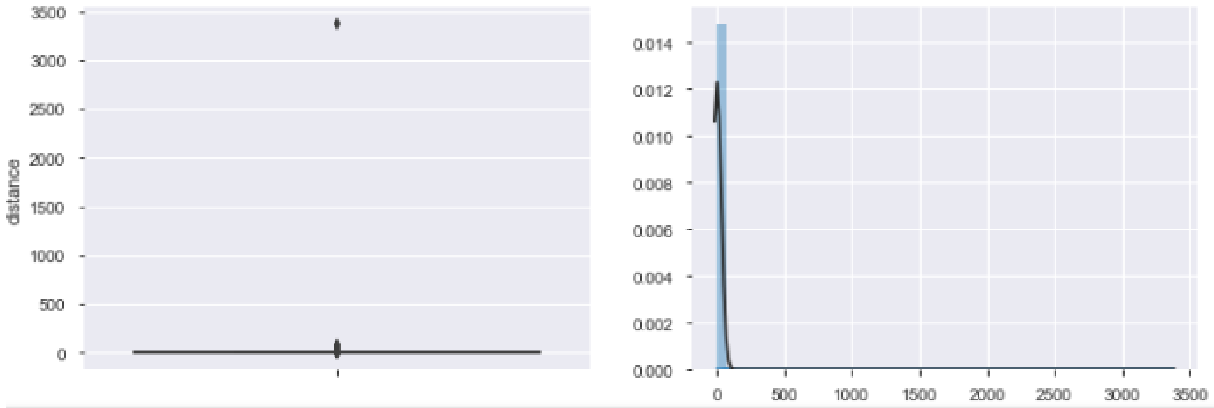
2.2.2 Outlier Analysis

An outlier is any inconsistent/abnormal value in a observation that deviates from the rest of the observations. These inconsistent value can be due to manual error, experimental error or invalid entry. Linear regression problems are very proned to outliers and presence of it can harm either the classifier or regressor. It can cause sufficient issues in the distribution of the variable as seen in this project. It is advisable to either remove the outliers or replace them with upper or lower cap values as per business scope.

We use boxplot visualization for doing outlier analysis and found that passenger_count, fare_amount and distance were having huge outlier and their distributions were majorly affected.



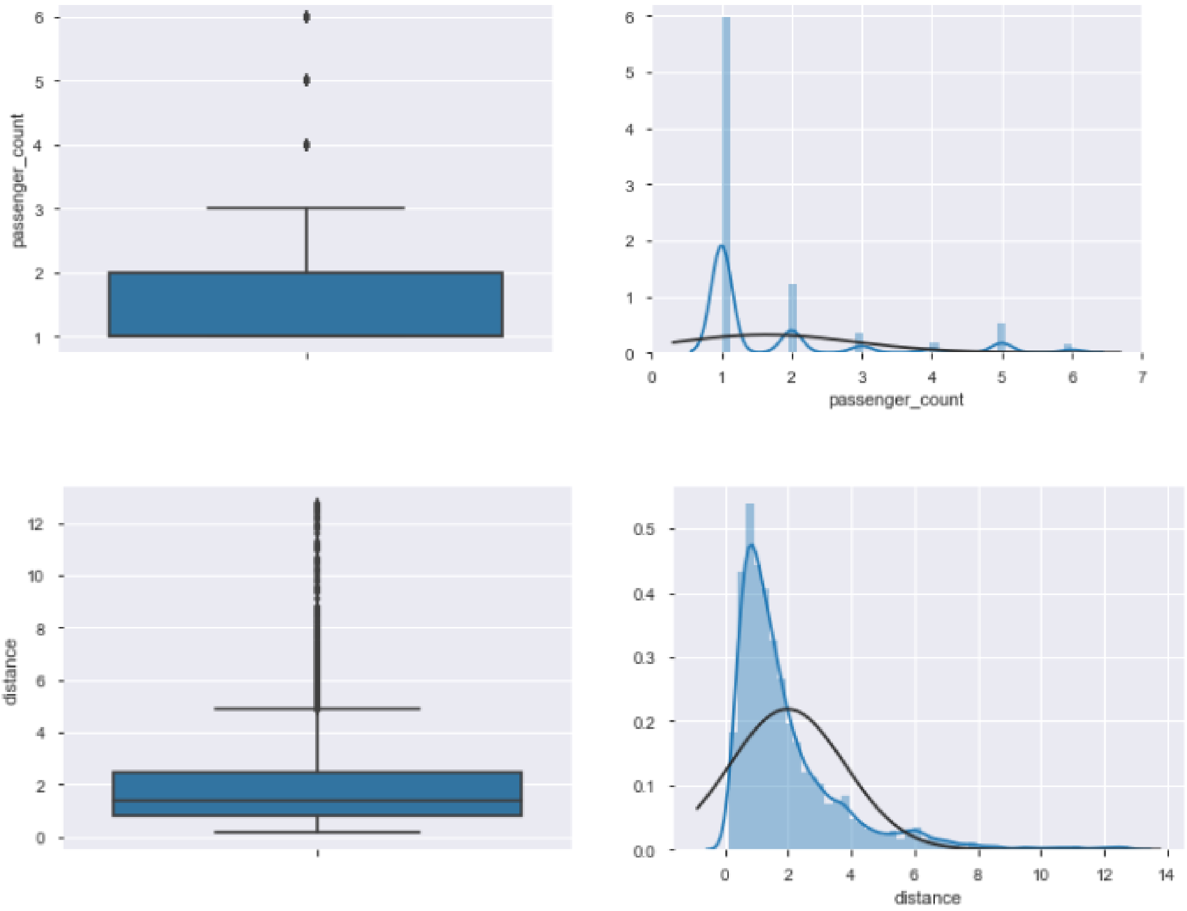


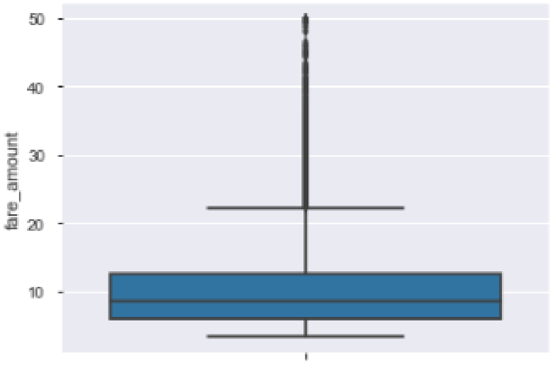


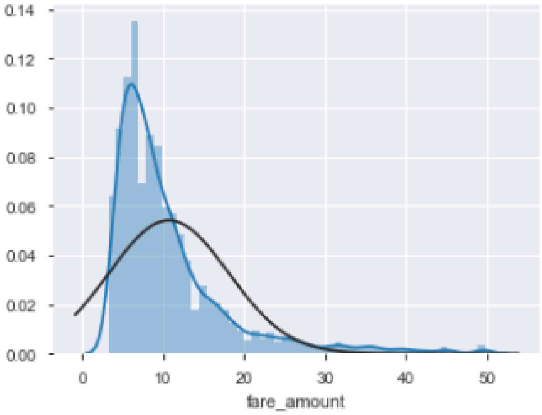
In Python: We tried to remove outliers for a very small set of observations. We selected <q1,>q99 quartiles on distance and fare_amount variables and dropped <q2 and >q98 values for passenger_count. The selection of the quartiles values depends at the discretion of the business. We selected these quartile upon constant experimentation to verify if valid data falls into this range or not.

In R: We had set NA values for the observations that fall into these quartile and then tried imputing with MICE package.

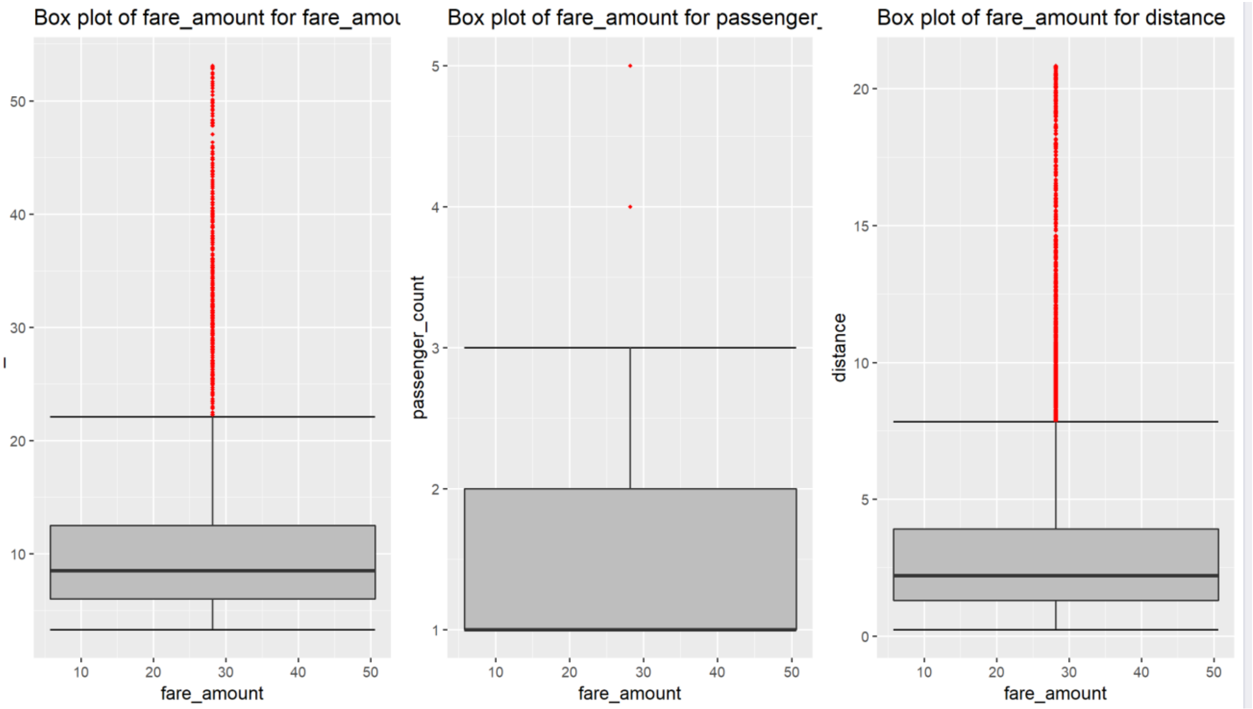
Boxplot after removal of outliers in python







Box plot after fixing outliers in R



In the above figures, we see that there are outliers but significant outliers are removed or fixed.

2.2.4 Feature Engineering/Extraction

Feature Engineering is the science of creating new variables from the existing set of variables. In our project we engineered pickup_latitude, pickup_longitude, dropoff_latitude and dropoff_longitude into distance. By doing this we were able to make the dataset very simple and extract valuable information from the distance feature where interpretation made easier. The motive behind feature engineering was that the correlation of fare_amount was not significant with the latitude/longitude variables.

We also extracted date relevant fields from the pickup_datetime variable into year, month, day, weekday, hour, minute, seconds

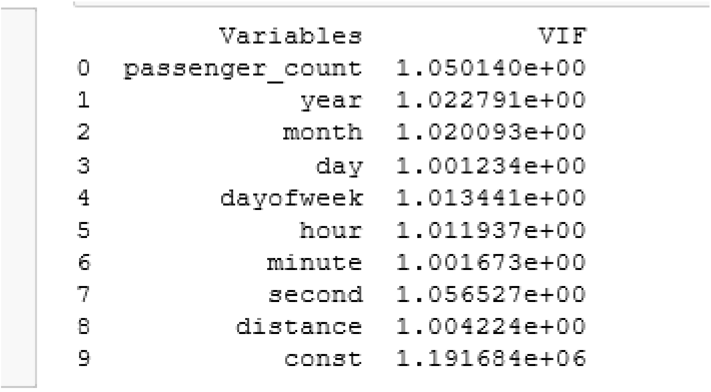
In Python: We used vincenty method from geopy.disatnce package.

In R: We used geosphere package for distance engineering

Once after the feature engineering is done, we need to make sure the outlier analysis is exercised on the new variable

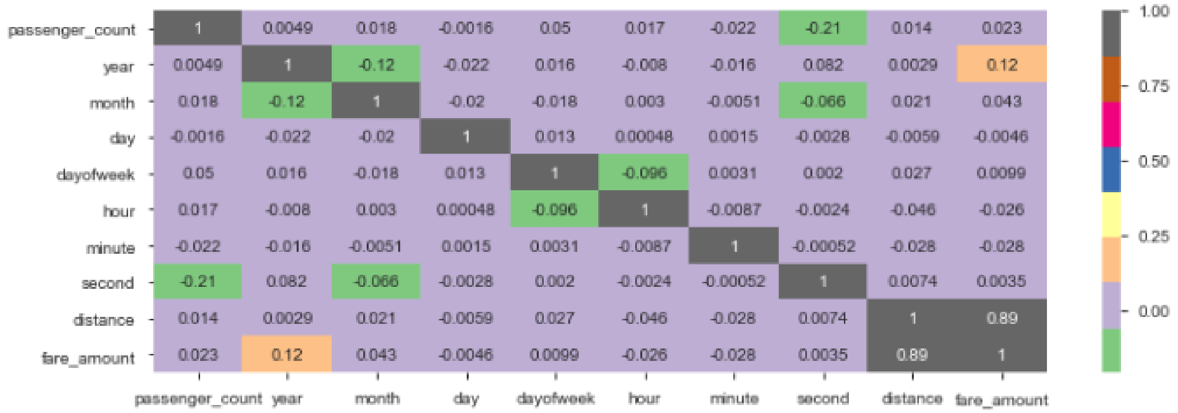
2.2.5 Feature Selection

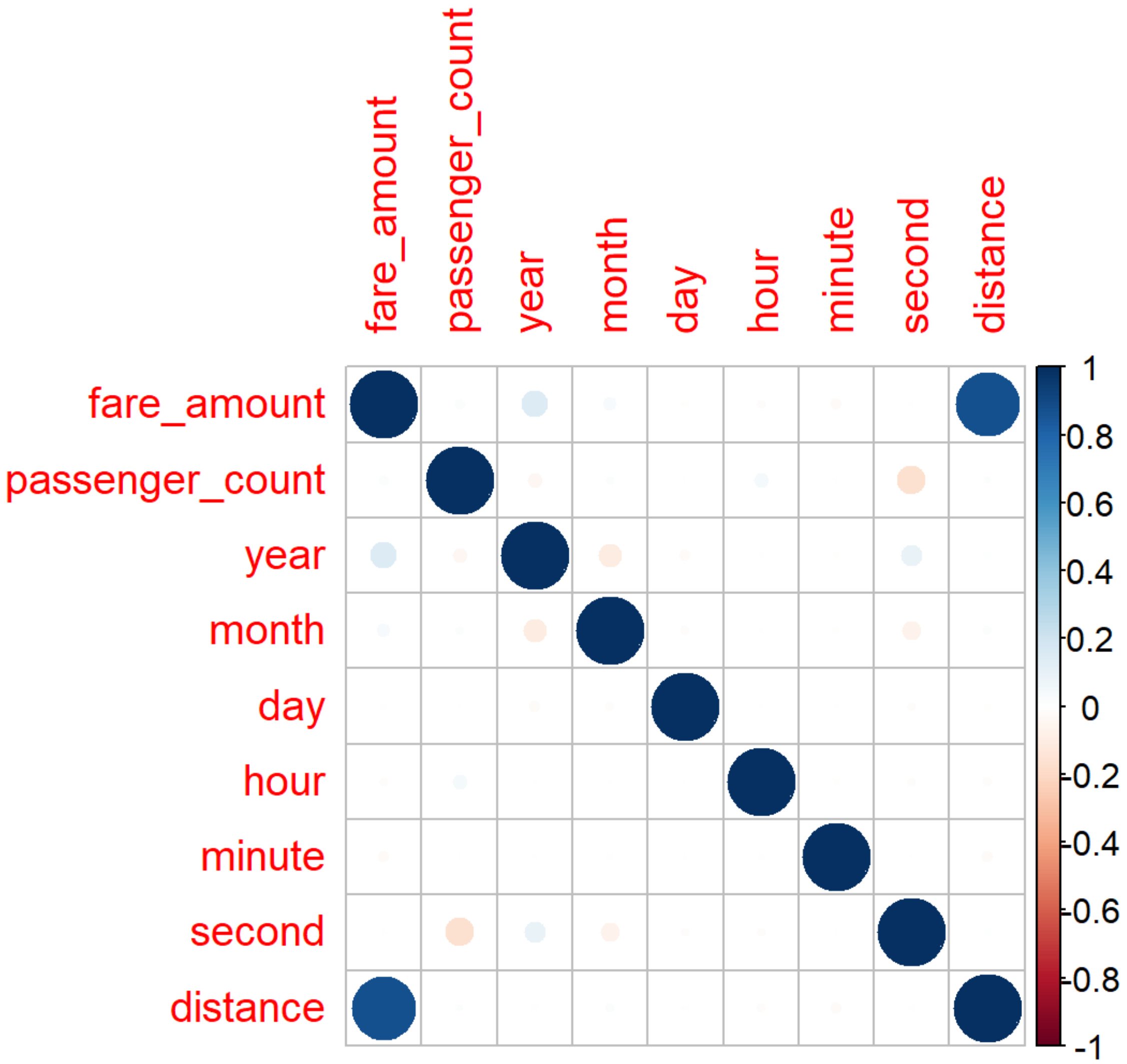
Before going with modeling, we need to assess the significance of variables that are contributing to the response variables. In simple terms, if the value of independent variable increases, do we find any change in the increase in the response variable and viceversa. There could be a possibility that many variables in our analysis are not significant. Hence feature selection should be exercised to identify the relevant variables as well as reduce the complexity of the model. Also linear regression assumption requires that there should be no multicollinearity between the variables. We used vif to verify for multicollinearity and all the values are within 2.



```
         Variables           VIF
0  passenger_count  1.050140e+00
1             year  1.022791e+00
2            month  1.020093e+00
3              day  1.001234e+00
4        dayofweek  1.013441e+00
5             hour  1.011937e+00
6           minute  1.001673e+00
7           second  1.056527e+00
8         distance  1.004224e+00
9            const  1.191684e+06
```

For numeric feature variables, we use correlation plots to find out significant continuous variables and for categorical variables to numeric variables we use ANOVA test and for categorical to categorical variable significance we use Chi Square test.


passenger_count
year
month
day
dayofweek
hour
minute
second
distance
fare_amount
1 0.0049 0.018 -0.0016 0.05 0.017 -0.022 -0.21 0.014 0.023
0.0049 1 -0.12 -0.022 0.016 -0.008 -0.016 0.082 0.0029 0.12
0.018 -0.12 1 -0.02 -0.018 0.003 -0.0051 -0.066 0.021 0.043
-0.0016 -0.022 -0.02 1 0.013 0.00048 0.0015 -0.0028 -0.0059 -0.0046
0.05 0.016 -0.018 0.013 1 -0.096 0.0031 0.002 0.027 0.0099
0.017 -0.008 0.003 0.00048 -0.096 1 -0.0087 -0.0024 -0.046 -0.026
-0.022 -0.016 -0.0051 0.0015 0.0031 -0.0087 1 -0.00052 -0.028 -0.028
-0.21 0.082 -0.066 -0.0028 0.002 -0.0024 -0.00052 1 0.0074 0.0035
0.014 0.0029 0.021 -0.0059 0.027 -0.046 -0.028 0.0074 1 0.89
0.023 0.12 0.043 -0.0046 0.0099 -0.026 -0.028 0.0035 0.89 1
passenger_count
year
month
day
dayofweek
hour
minute
second
distance
fare_amount
1.00
0.75
0.50
0.25
0.00


fare_amount
passenger_count
year
month
day
hour
minute
second
distance
fare_amount
passenger_count
year
month
day
hour
minute
second
distance
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

From the above plots, it is clear that the response variable fare_amount is highly correlated to distance and less correlated to year. So we select

fare_amount ~ distance + year

2.2.5 Feature Scaling

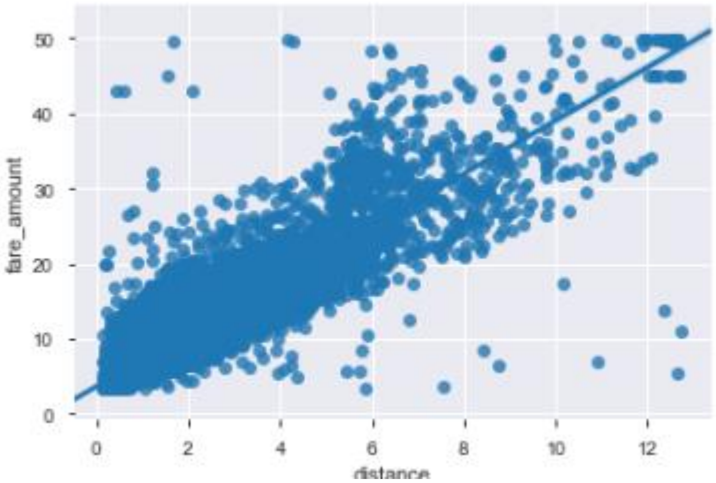
Sometimes the weight of some variables overwhelm the other variables in the dataset. In our project we have higher values for year variable and that could overwhelm the effect of distance variable and hence we require both the variables to be brought into a same scale. Feature scaling is used to normalize the range of features to a desired range. Widely used feature scaling methods are Standard Scaling and MinMax Scaling. Standard Scaling shifts the mean of the variable to zero with a standard deviation to 1. Whereas MinMaxScaler normalizes the data to a range from 0 to 1. Feature scaling is required in almost all the models except the ensemble trees. It is a must when we go for Deep Learning models.

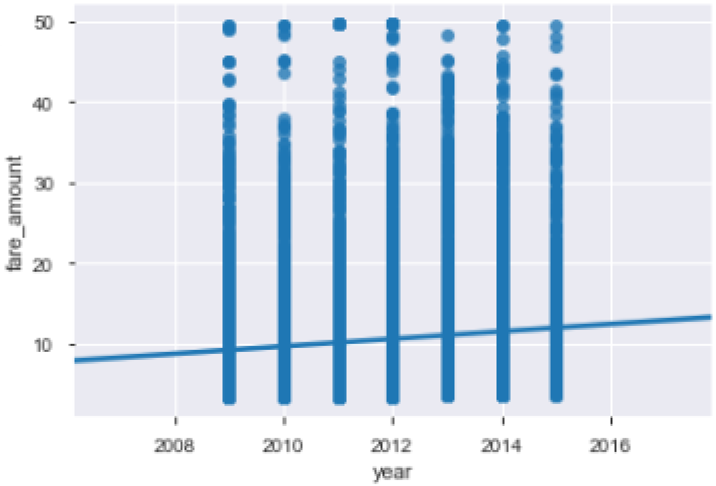
In R: We used fastScale method within dataPreparation package

In Python: We used StandardScaler method within sklearn.preprocessing package

3 BI Variate Analysis:

We plotted the response variable fare_amount with the distance and the year as below





4 Splitting the data into train and test:

To calculate the metrics of any model, we need to split the data into train and test datasets. We use the train data for training the model and the test data to calculate the metrics. We used sci-kit library's train_test_split to split the data into train and test sets and caret's createDataPartition in R. We had taken 70% of the data in train and the remaining in test.

5 Model Development:

The Cab Fare Prediction problem is a kind of regression problem under supervised learning. We are going to build the regression model on the training data and predict it on test data. The following are the attempted models being developed for this problem

1. Linear Regression
2. Decision Tree
3. Random Forests
4. Xgboost
5. ANN (Artificial Neural Network)

Since this is a regression problem, the metrics used for the evaluation of the model are

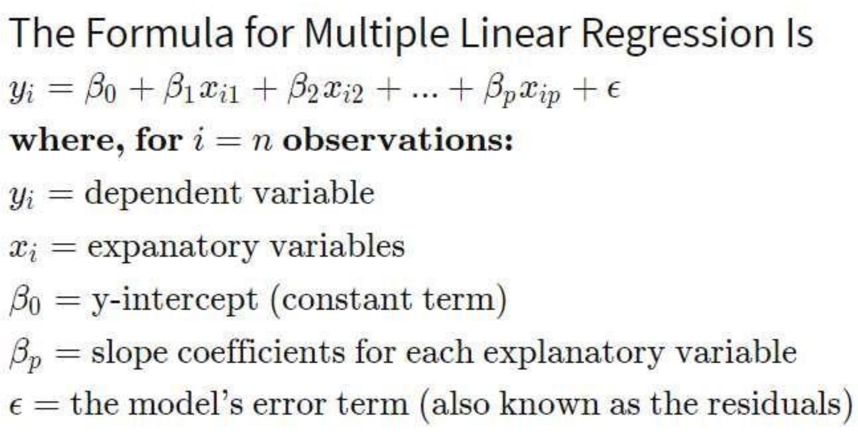
1. R2 score
2. Explained Variance Score
3. Mean Absolute Error
4. Mean Squared Error
5. Root Mean Squared Error
6. MAPE (Mean absolute percentage error)

5.1 Linear Regression:

Linear Regression is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of the linear regression is to fix the equation on the predictor variables to get the output. This is very useful when the problem statement is of forecasting type. The assumptions of this model are

1. Linear relation between features and target
2. Variables are normally distributed
3. No multicollinearity among the independent variables
4. Homoscedasticity (residual plot should be centred around 0)

The main pitfall of this technique is that it is sensitive to outliers. Presence of outliers may tend to change the best fit line.



The Formula for Multiple Linear Regression Is

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon$$

where, for $i = n$ observations:

y_i = dependent variable

x_i = expanatory variables

β_0 = y-intercept (constant term)

β_p = slope coefficients for each explanatory variable

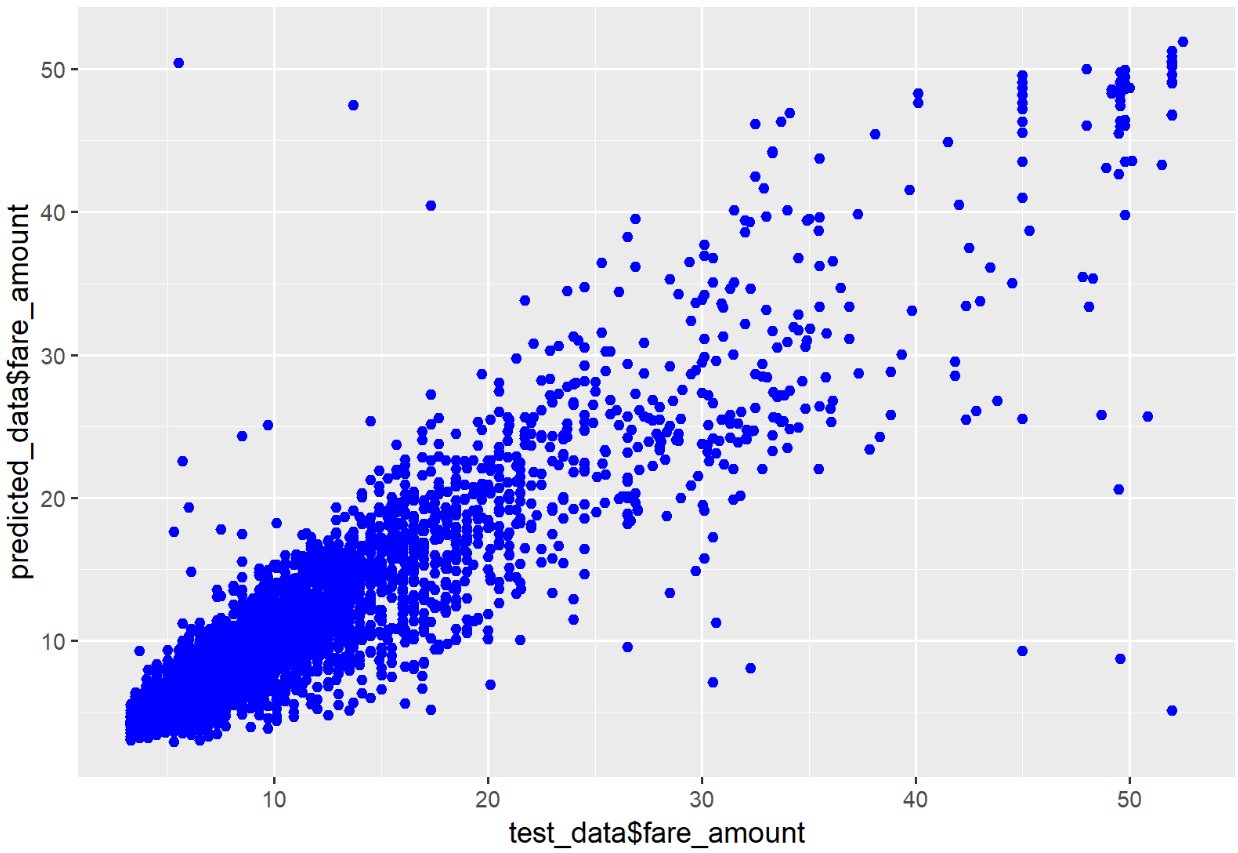
ϵ = the model's error term (also known as the residuals)

In R: Model Performance

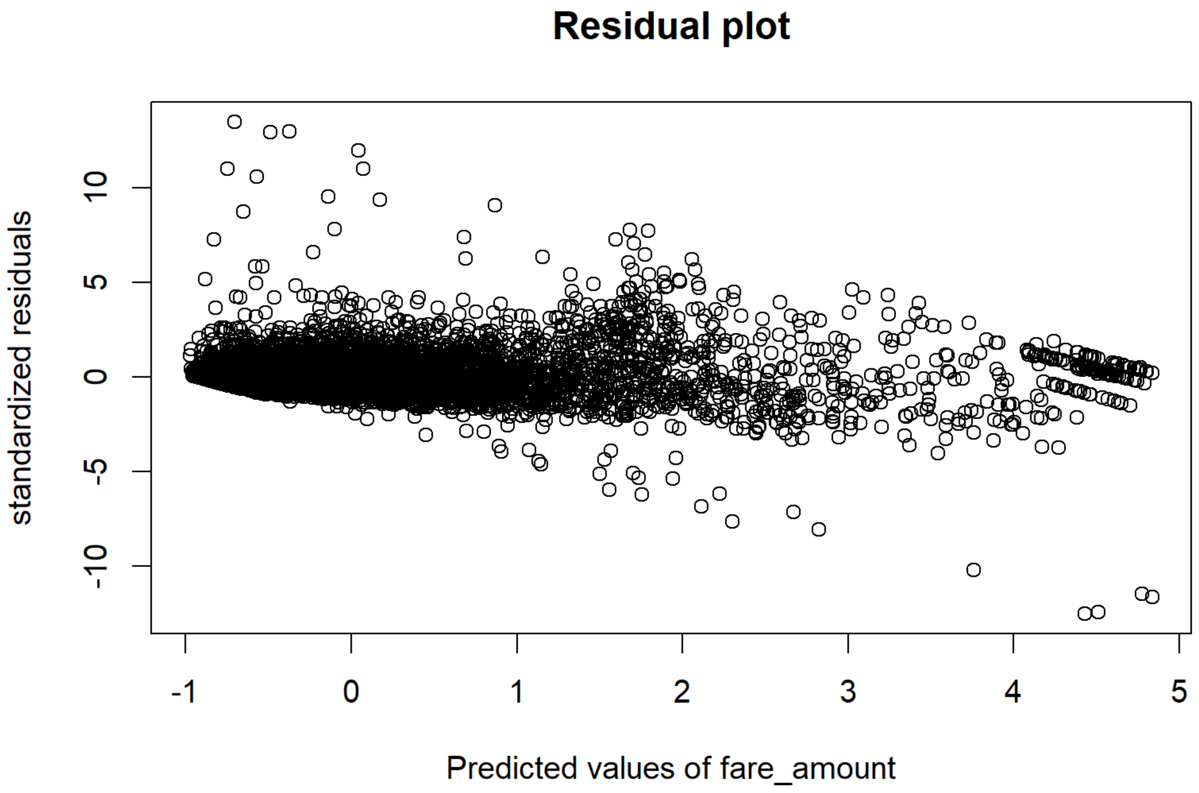
Error Metrics	R squared	MAE	MSE	RMSE	MAPE
Train	.8314	2.04	12.07	3.47	.197
Test	.8252	2.02	11.24	3.35	.195

In Python: Model Performance

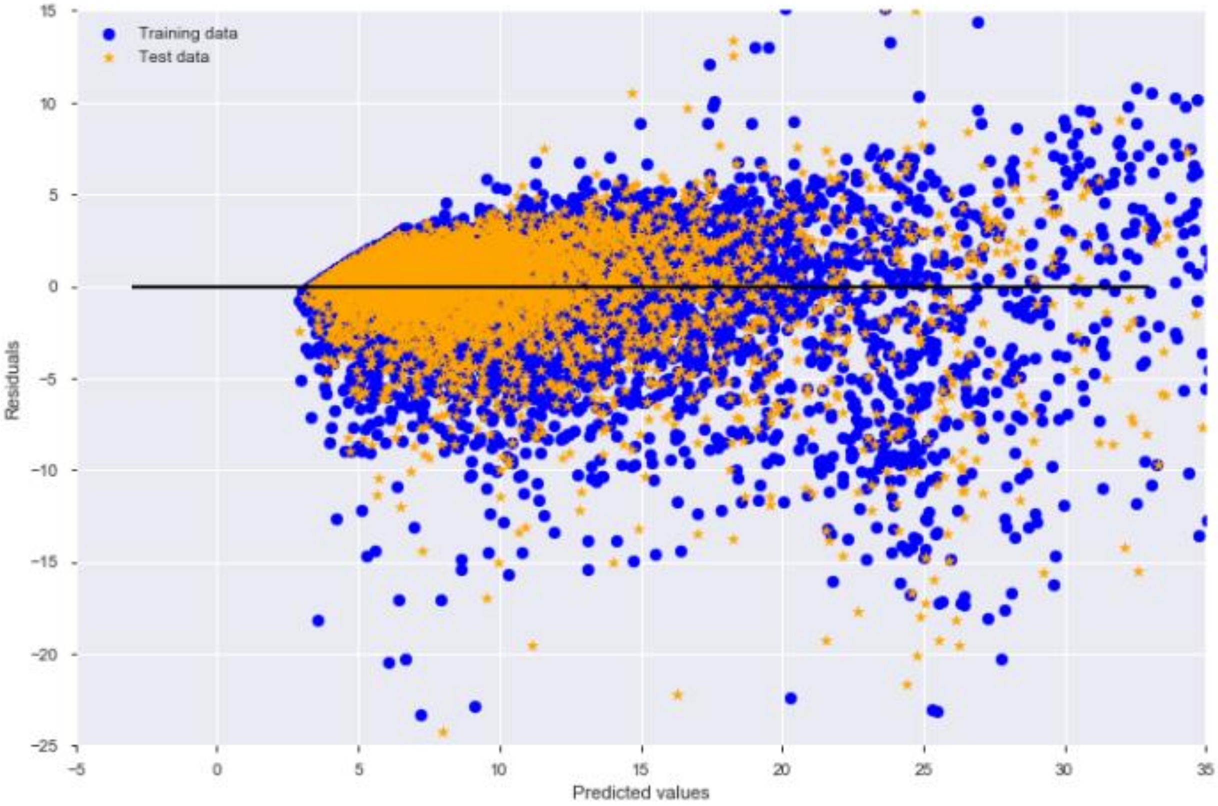
Error Metrics	R squared	MAE	MSE	RMSE	MAPE
Train	.80	1.94	10.0489	3.17	944.5
Test	.78	2.05	11.6964	3.42	969.15



Plot of predicted values with test values.



Residual plot in R

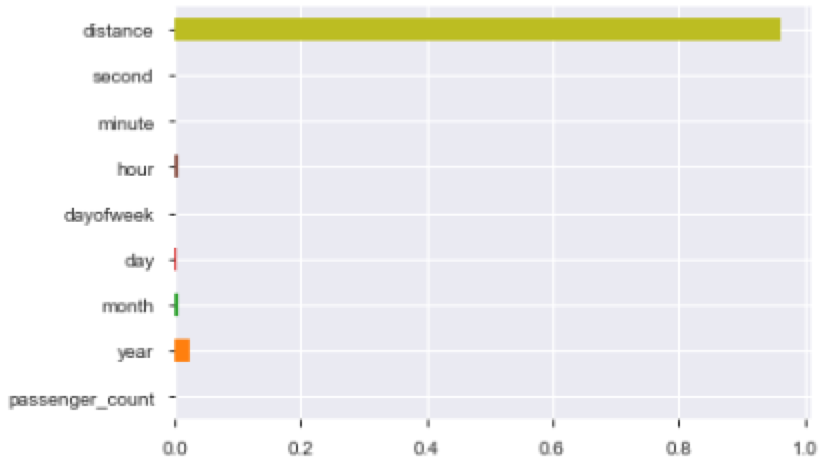


Residual Plot in Python

5.2 Decision Trees:

Decision Tree is a supervised machine learning algorithm, which is used to predict the data for classification and regression. It accepts both continuous and categorical variables. A decision tree is a decision support tool that uses a tree like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. Each branch connects nodes with "and" and multiple branches are connected by "or". Extremely easy to understand by the business users. It provides its output in the form of rule, which can easily understand by a non -technical person also.

One of the best advantage of modelling in decision trees is that it does not require more preprocessing of data and is robust to outliers. Also it provides the feature importance of the predictor variables as below.



In R: Model Performance

Error Metrics	R squared	MAE	MSE	RMSE	MAPE
Train	.80	2.33	14.03	3.74	.229
Test	.79	2.36	14.42	3.79	.22

In Python: Model Performance

Error Metrics	R squared	MAE	MSE	RMSE	MAPE
Train	.83	1.8	8.2369	2.87	945
Test	.76	2.07	12.46	3.53	969

5.2 Random Forests:

This is a sort of ensembling technique where multiple parallel trees are used to predict the outcome. In regression it uses the mean of all the estimators to predict the response variable and in classification it uses the maximum vote produced in each estimator to predict the response.

We used 100 estimators in Random Forests

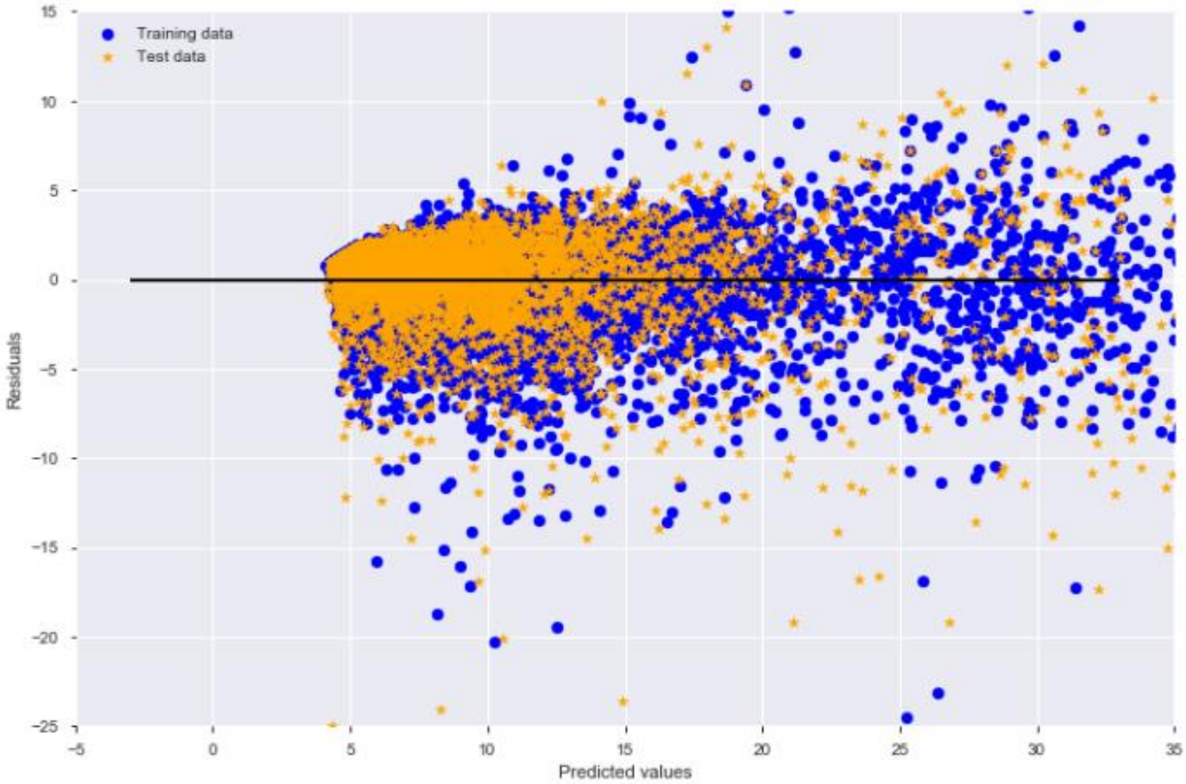
In R: Model Performance

Error Metrics	R squared	MAE	MSE	RMSE	MAPE
Train	.96	.92	2.54	1.59	.09
Test	.83	1.97	11.06	3.32	.196

In Python: Model Performance

Error Metrics	R squared	MAE	MSE	RMSE	MAPE
Train	.88	1.52	5.8	2.41	945
Test	.799	1.95	10.82	3.29	969

Residual Analysis

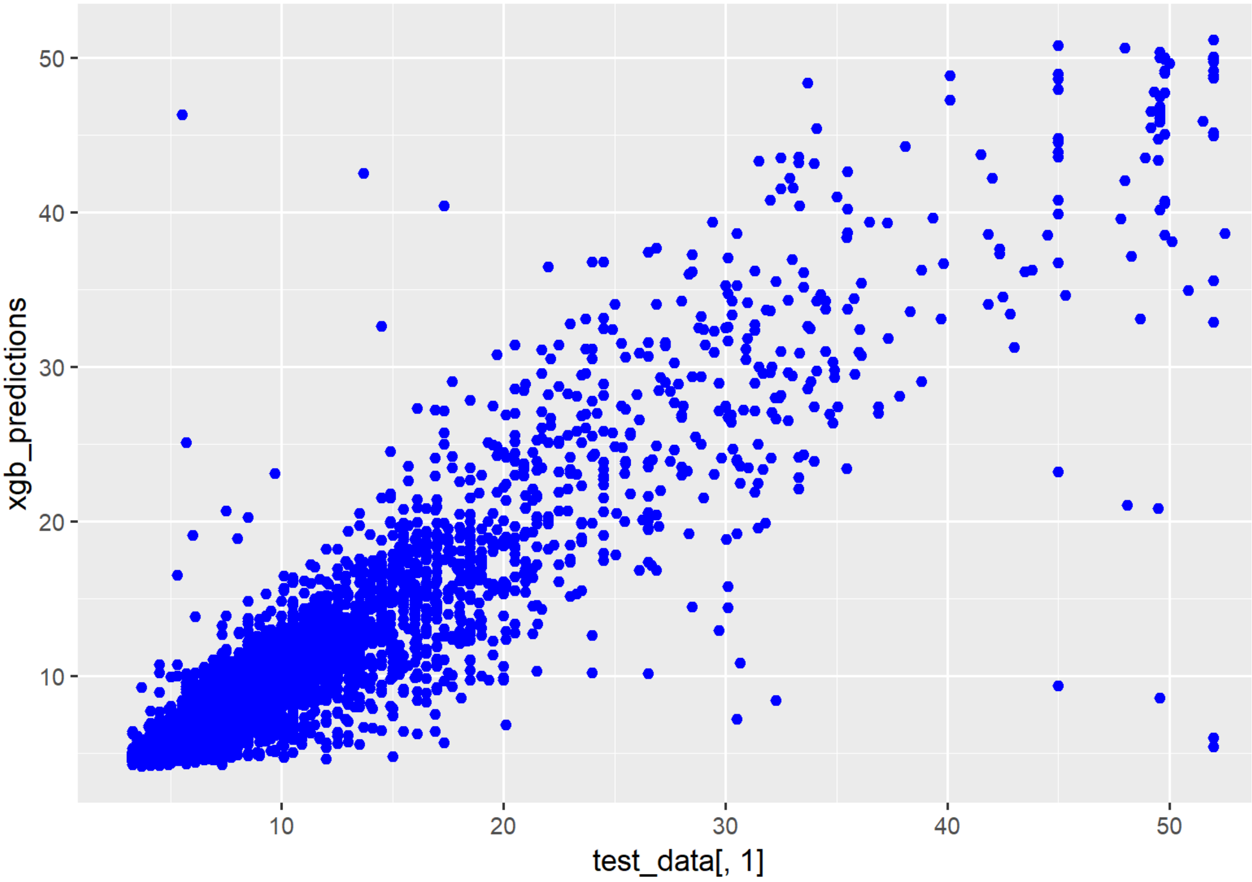


5.3 Xgboost:

Xgboost is a type of ensemble technique which is based on the concept of boosting wherein several weak learners from one classifiers are fed into another classifier with more weights given to the misclassified sample. The regularization term controls the complexity of the model which helps to control overfitting and thus gives better performance over other ensemble techniques.

We used Xgboost only in R

Error Metrics	R squared	MAE	MSE	RMSE	MAPE
Train	.90	1.65	7.16	2.67	.16
Test	.83	1.9	11.45	3.38	.181

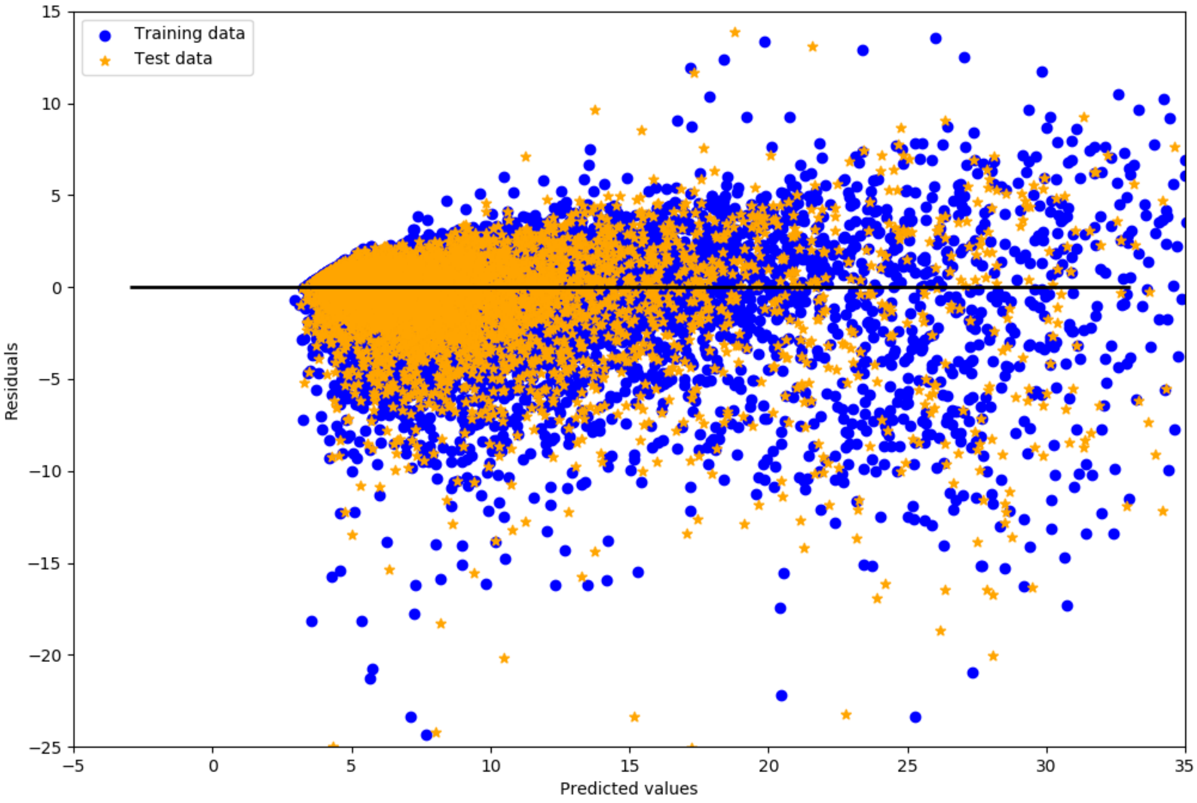


5.4 ANN:

ANN is a class of algorithm which works on neuron activations and it uses backpropogation in optimizing the gradient descent of a cost function. It provides random weights(guesses) in predicting the outcome and then optimizes the cost function to readjust the weights on the neuron.

We used 2 layered neural network for working on this problem statement. The first layer contained 9 units with relu activation function and the output unit corresponded to the fare_amount with an activation function 'linear' (regression). We used rmsprop for update rule in optimizing the loss function and the cost function used in our network was 'huber_loss'

Error Metrics	R squared	MAE	MSE	RMSE	MAPE
Train	.81	1.82	.0036	.06	951
Test	.81	1.97	.0036	.06	976



6 Conclusion:

From the above metrics, if we go on rmse calculation ANN model is winning from other models. When it comes to selection based on R2 score, the ensemble models like Random Forests and Xgboost are doing well.

The cab prices are dependent on the distance being travelled and it looks like there were varying prices every year on the distance travelled.

Prices are unaffected by the passenger_count as seen from the correlation plots.

Please find the predicted fare_amount in the test folder for the scoring dataset

test_DL.csv -> Prediction from ANN

test_forecasted.csv -> Predictions from python linear model (without outliers)

test_predictions_R_lm.csv -> Predictions from R linear model (fixing outliers)

test_predictions_R_RF.csv -> Predictions from R Random Forests model