Cab Fare Prediction Model

Data Science Project (Edwisor)

Riddhesh Sajwan
February 26, 2020

Contents

Introduction

1. Problem Statement

We have to make a prediction model for a cab renting company. This prediction model should be able to predict the amount of fare on a particular ride given the source, destination and count of passengers. We are provided with a training dataset which has date of journey, source, destination and count of passengers. Also, we are given a test dataset with the same parameters to run our prediction model on for calculation of accuracy.

2. Data

Following is a sample of data provided to us to make our prediction model. We will preprocess the data provided to avoid any missing values or outliers.

Table 1.2.1 Training Data (Rows 1-13 and Columns 1-4)

fare_amount	pickup_datetime	pickup_longitude	pickup_latitude
4.5	2009-06-15 17:26:21 UTC	-73.844311	40.721319
16.9	2010-01-05 16:52:16 UTC	-74.016048	40.711303
5.7	2011-08-18 00:35:00 UTC	-73.982738	40.76127
7.7	2012-04-21 04:30:42 UTC	-73.98713	40.733143
5.3	2010-03-09 07:51:00 UTC	-73.968095	40.768008
12.1	2011-01-06 09:50:45 UTC	-74.000964	40.73163
7.5	2012-11-20 20:35:00 UTC	-73.980002	40.751662
16.5	2012-01-04 17:22:00 UTC	-73.9513	40.774138
	2012-12-03 13:10:00 UTC	-74.0065	40.72671
8.9	2009-09-02 01:11:00 UTC	-73.9807	40.73387
5.3	2012-04-08 07:30:50 UTC	-73.9963	40.73714
5.5	2012-12-24 11:24:00 UTC	0	0

Table 1.2.2 Training Data (Rows 1-13 and Columns 5-7)

dropoff_longitude	dropoff_latitude	passenger_count
-73.84161	40.712278	1
-73.979268	40.782004	1
-73.991242	40.750562	2
-73.991567	40.758092	1
-73.956655	40.783762	1
-73.972892	40.758233	1
-73.973802	40.764842	1
-73.990095	40.751048	1
-73.9931	40.73163	1
-73.9915	40.75814	2
-73.9807	40.73356	1
0	0	3

As we can observe from the data that below are the variables that we will be using as predictors for our model:

Table 1.2.3 Predictors

SR. NO.	**PREDICTOR**
1	pickup_longitude
2	pickup_latitude
3	dropoff_longitude
4	dropoff_latitude
5	passenger_count

Methodology

1. Pre-Processing

Data pre-processing is an important initial step in any data mining, data science or machine learning project. In simple terms, it is the process of converting raw data into a much readable, understandable format. The raw data usually received is always incomplete or has errors, hence data pre-processing is done to make sure that complete and correct data is sent to the deployment model to get a flawless and more accurate prediction result. The following steps take place in the process of pre-processing:

a. Missing Data Analysis
b. Outlier Analysis
c. Encoding Categorical Data
d. Standardization of Data
e. Dimensionality Reduction
f. Splitting of Data Set

a. Missing Data Analysis

Missing Data is a common problem with databases. This situation usually arises due to technical glitches while collecting data or people refusing/forgetting to fill data completely while the time of data collection. Missing data analysis is a technique used to find out the presence of missing values in a dataset and treat this situation by either removing that observation or imputing values into that observation using various methods like mean, median, mode, KNN imputation etc.

Observing the above sample data, we can see that the dataset provided to us has missing values. The best way to figure out what to do with this data is by calculating the percentage of missing data in all the variables.

We will be going by the following two rules: -

1. If the percentage of missing data is less than 5% then we can eliminate the observations as their absence won't affect the model much.
2. If the percentage of missing data is more than 30% then won't consider imputation as it might change the entire outcome of the model.

We will be using R and python to perform missing value analysis.

We have imported the csv file provided to us with the condition that all blank values, extra spaces and zeroes will be replaced by NA into a dataframe.

R code:

```
ds <- read.csv("C:/Users/riddh/OneDrive/Documents/Edwisor/Cab Fare Prediction/train_cab.csv",
sep = ",", header = T, na.strings = c(" ", "", "NA","0"))
```

Dataframe Output:

Fig 2.1.a.1 Dataframe of Training Data

	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
1	4.5	2009-06-15 17:26:21 UTC	-73.84431	40.72132	-73.84161	40.71228	1
2	16.9	2010-01-05 16:52:16 UTC	-74.01605	40.71130	-73.97927	40.78200	1
3	5.7	2011-08-18 00:35:00 UTC	-73.98274	40.76127	-73.99124	40.75056	2
4	7.7	2012-04-21 04:30:42 UTC	-73.98713	40.73314	-73.99157	40.75809	1
5	5.3	2010-03-09 07:51:00 UTC	-73.96810	40.76801	-73.95665	40.78376	1
6	12.1	2011-01-06 09:50:45 UTC	-74.00096	40.73163	-73.97289	40.75823	1
7	7.5	2012-11-20 20:35:00 UTC	-73.98000	40.75166	-73.97380	40.76484	1
8	16.5	2012-01-04 17:22:00 UTC	-73.95130	40.77414	-73.99009	40.75105	1
9	NA	2012-12-03 13:10:00 UTC	-74.00646	40.72671	-73.99308	40.73163	1
10	8.9	2009-09-02 01:11:00 UTC	-73.98066	40.73387	-73.99154	40.75814	2
11	5.3	2012-04-08 07:30:50 UTC	-73.99634	40.73714	-73.98072	40.73356	1
12	5.5	2012-12-24 11:24:00 UTC	NA	NA	NA	NA	3
13	4.1	2009-11-06 01:04:03 UTC	-73.99160	40.74471	-73.98308	40.74468	2
14	7	2013-07-02 19:54:00 UTC	-74.00536	40.72887	-74.00891	40.71091	1
15	7.7	2011-04-05 17:11:05 UTC	-74.00182	40.73755	-73.99806	40.72279	2

Now, we will calculate the missing value percentage of every column using R. We get the following dataframe:

Fig 2.1.a.2 Dataframe of Missing Value Percentage

	Columns	Missing_percentage
1	pickup_longitude	1.9605402
2	pickup_latitude	1.9605402
3	dropoff_longitude	1.9543163
4	dropoff_latitude	1.9418684
5	passenger_count	0.6970810
6	fare_amount	0.1555984
7	pickup_datetime	0.0000000

As we can observe from the above figure, the missing data percentage of all the variables is less than 5% (rather 2%), so it is quite safe to assume that they don't contribute as much in the outcome, hence it is only sensible to eliminate these observations.

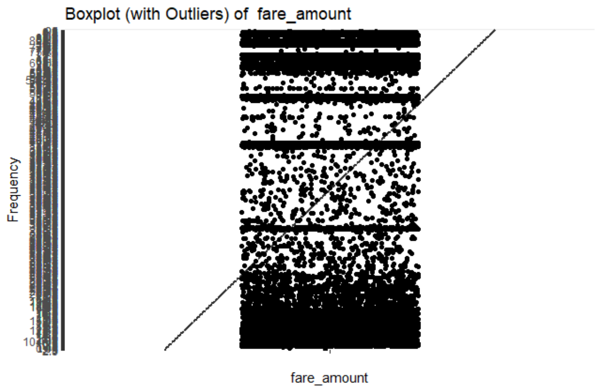
After performing the omission of these observations, we move onto the next step in pre-processing.

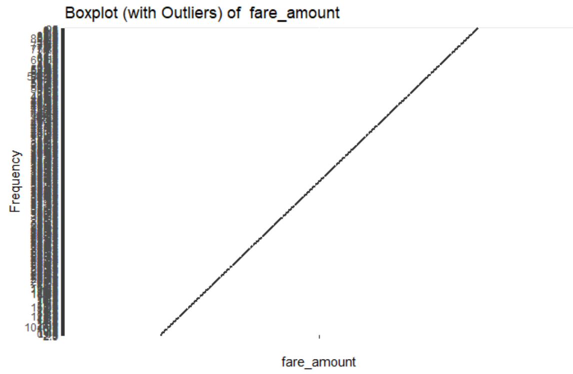
b. Outlier Analysis

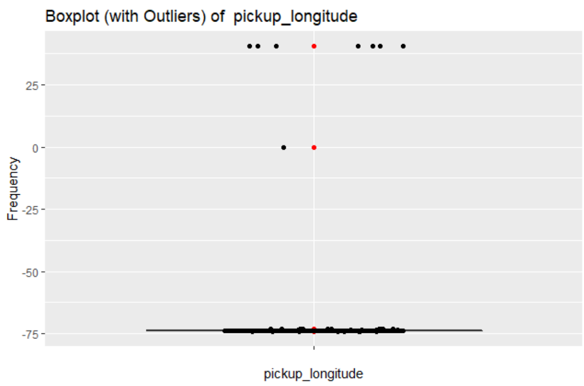
“Observation which deviates so much from other observations as to arouse suspicion it was generated by a different mechanism” — Hawkins(1980)

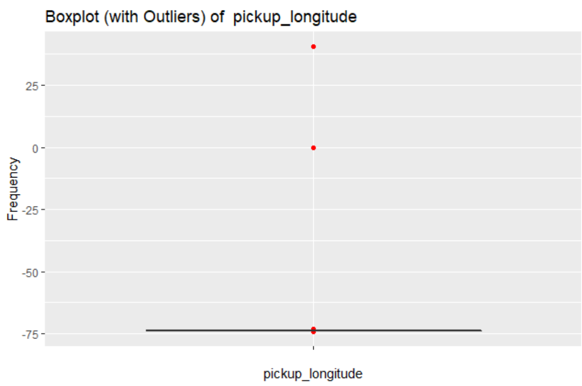
We have clearly observed from our visualizations that some predictors like *windspeed* had a few data points that weren’t as usual and hence they skewed the entire data outcome. We also proved the effects of outliers on the data using visualizations. The histograms and boxplots in the Fig 2.1.b.1 precisely depict how the outliers affected the outcome.

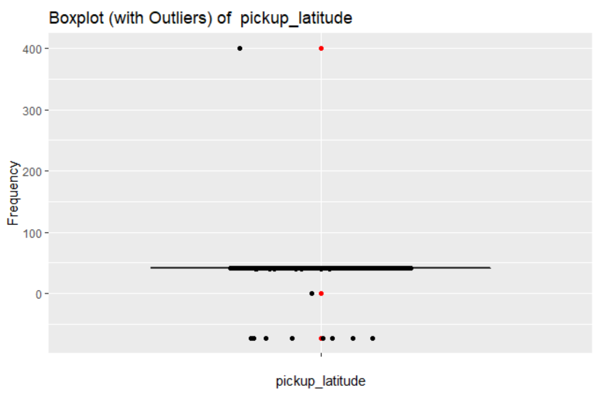
b.1 Boxplot with Outliers

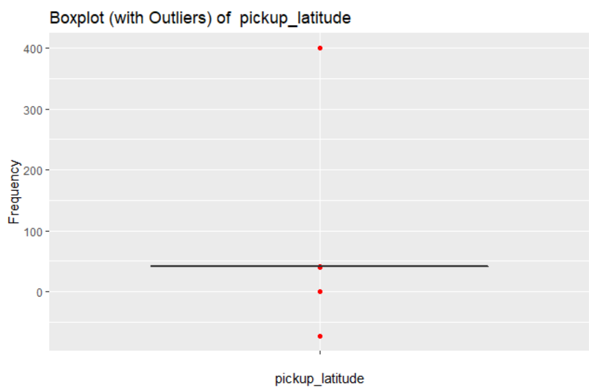












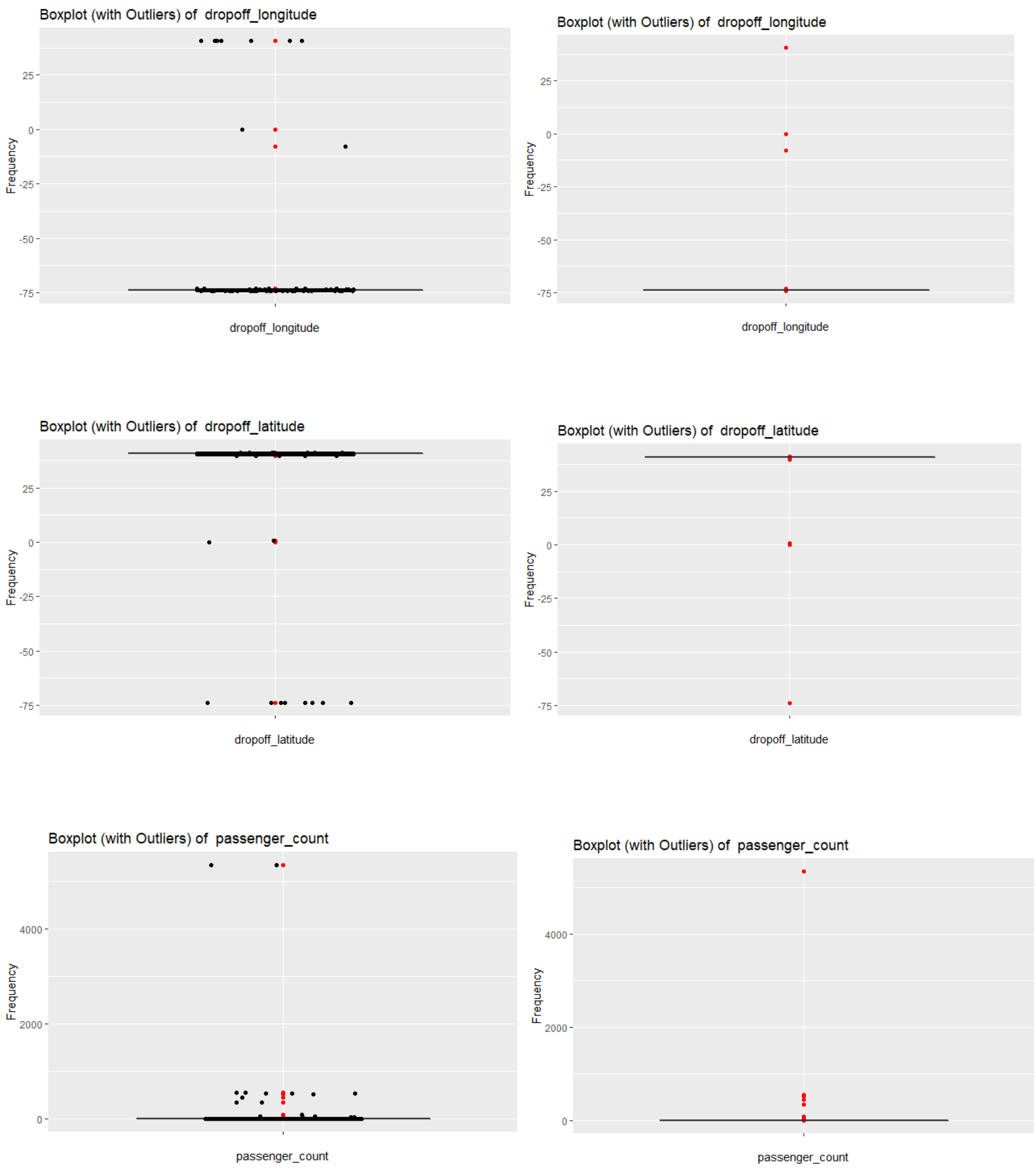


Fig 2.1.b.1 Boxplot (with outliers)

The red dots observed in the above diagrams are outliers and their presence can actually affect the graphs. In the above graphs, it is quite clear that the predictor *windspeed* has a lot of outliers and removing them can totally help the outcome.

Hence, below Fig 2.1.b.2.b is a representation of boxplots after the removal of outliers from *passenger_count* and refer Fig 2.1.b.2.a to see the effect of outliers on data, depicted by the use of boxplot and histogram.

b.2 <u>Effect of Outliers</u>

Fig 2.1.b.2.a with outliers

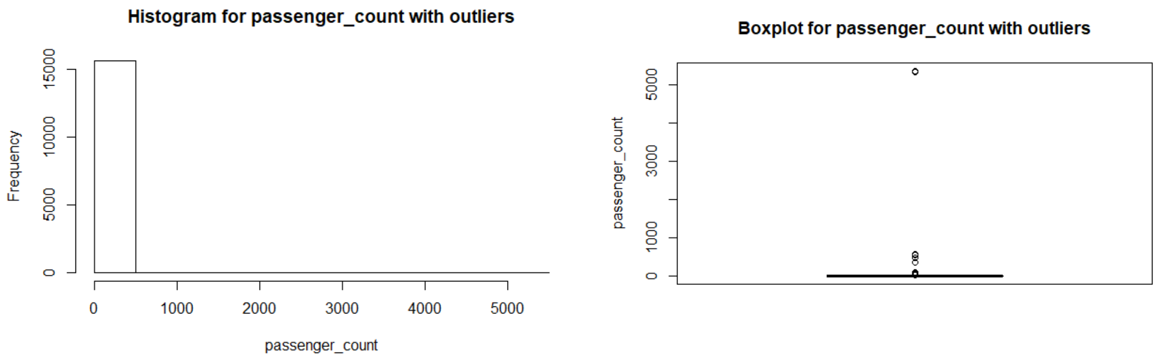
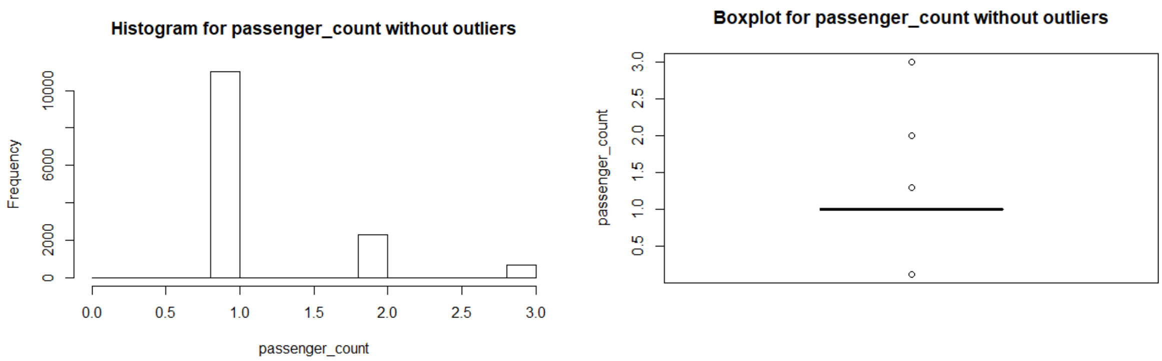


Fig 2.1.b.2.a without outliers



b.3 Boxplots After Removal of Outliers

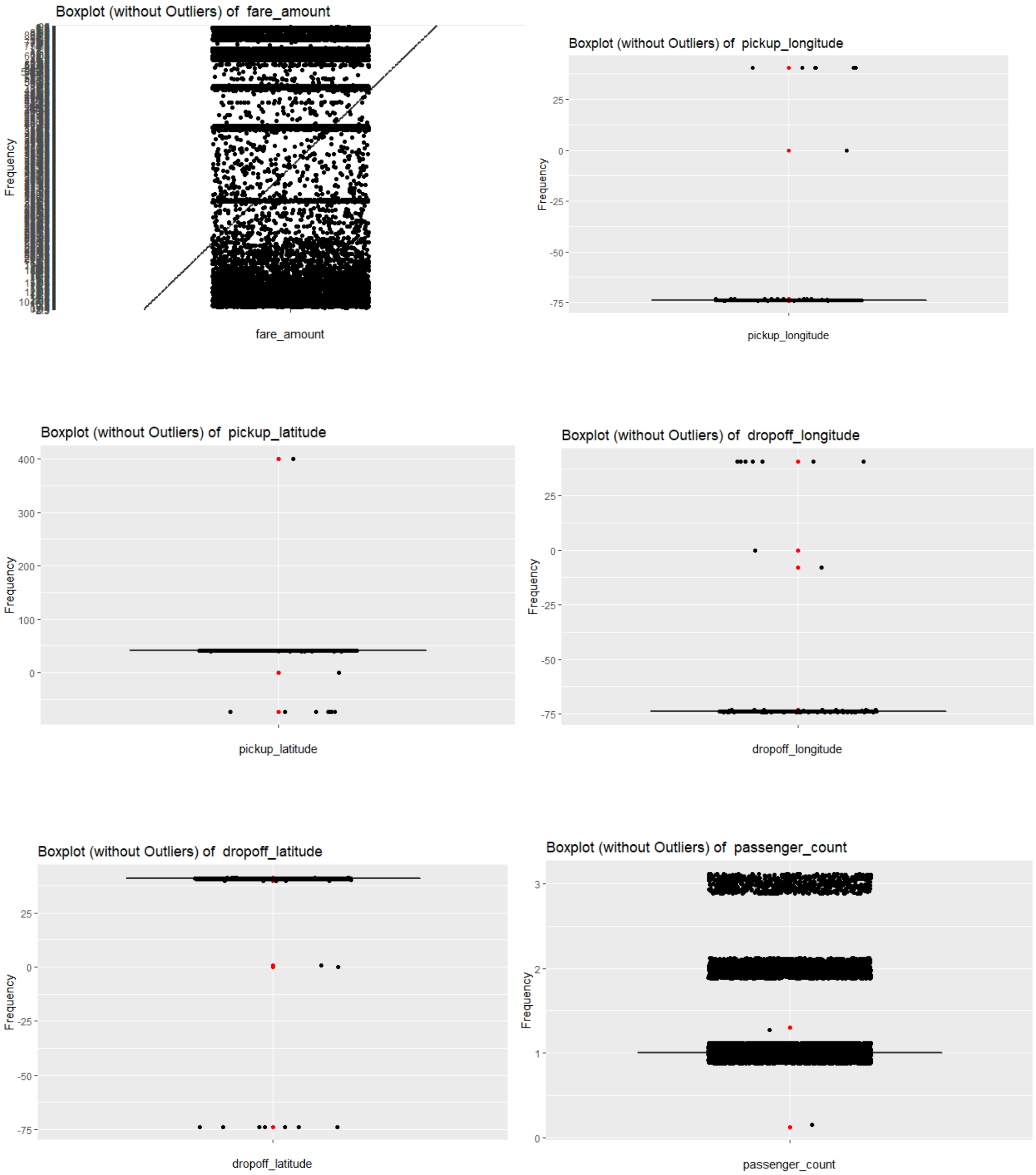


Fig 2.1.b.2.b Boxplots without outliers

c. Feature Selection

This is a technique used to make our dataset contain less but more contributing predictors. Given, we have a long list of predictors, it is quite possible that some of them are dependent on each other and hence do not contribute significantly in the final outcome. The process of discovering and eliminating such predictors is called feature selection. The features selected after this process become the part of our final dataset which is passed through our prediction model.

Table 2.1.c.1 Continuous Variables/Predictors

SR. NO.	**PREDICTOR**
1	pickup_longitude
2	pickup_latitude
3	dropoff_longitude
4	dropoff_latitude
5	passenger_count

As we can observe from the above data, the number of predictors are quite less and hence going for dimensionality reduction won't be as beneficial. Also, in this case, where the number of predictors is already less and all have equal importance (pick-up and drop locations and passenger count are equally important for cab fare prediction) we should not be considering feature selection as it may lead to loss of data.

d. Encoding Categorical Data

It is very important to encode categorical data into numbers, but as we can observe from the data, all our variables are already numeric. Hence, the question arises, are all of them continuous. So, we run the following line of code in R to convert any categorical data to continuous data.

```
for (i in 2:7){if(class(ds_new[,i]) == "factor"){ds_new[,i] <- as.numeric(ds_new[,i])}}
```

We found out that *fare_amount* was turned into continuous by this code.

Fig 2.1.d.1 Categorical to Continuous

	pickup_datetime	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
1	2009-06-15 17:26:21 UTC	300	-73.84431	40.72132	-73.84161	40.71228	1
2	2010-01-05 16:52:16 UTC	57	-74.01605	40.71130	-73.97927	40.78200	1
3	2011-08-18 00:35:00 UTC	372	-73.98274	40.76127	-73.99124	40.75056	2
4	2012-04-21 04:30:42 UTC	431	-73.98713	40.73314	-73.99157	40.75809	1
5	2010-03-09 07:51:00 UTC	369	-73.96810	40.76801	-73.95665	40.78376	1

e. Rest of the Pre-Processing

As it is quite evident from the data that we don't need the rest of the pre-processing techniques in our model. We rejected dimensionality reduction in the above section c and standardization doesn't seem necessary at all for our data. Apart from that, the data provided to us was already divided into training and testing set, so splitting of data is also not required.

Let's move onto the next section, where we perform data visualization on our data set.

2. Data Visualization

This is a very important aspect for any data science project as this step enables the engineer to observe the data carried by the predictors in a much more informative way. In data visualization we use various graphical representations to showcase the observations we have with us, which helps us to understand the future steps a bit more. We have the following kind of graphs in our report: (R code in Appendix)

a. Probability Density Function with Histogram and normal fit. (Fig 2.2.a.1)

b. Histogram and Mean of Predictors (Fig 2.2.b.1)

c. Boxplots for each Predictor (Before Fig 2.1.b.1 and After Fig 2.1.b.2.b removal of outliers)

a. Probability Density Function with Histogram and Normal Fit

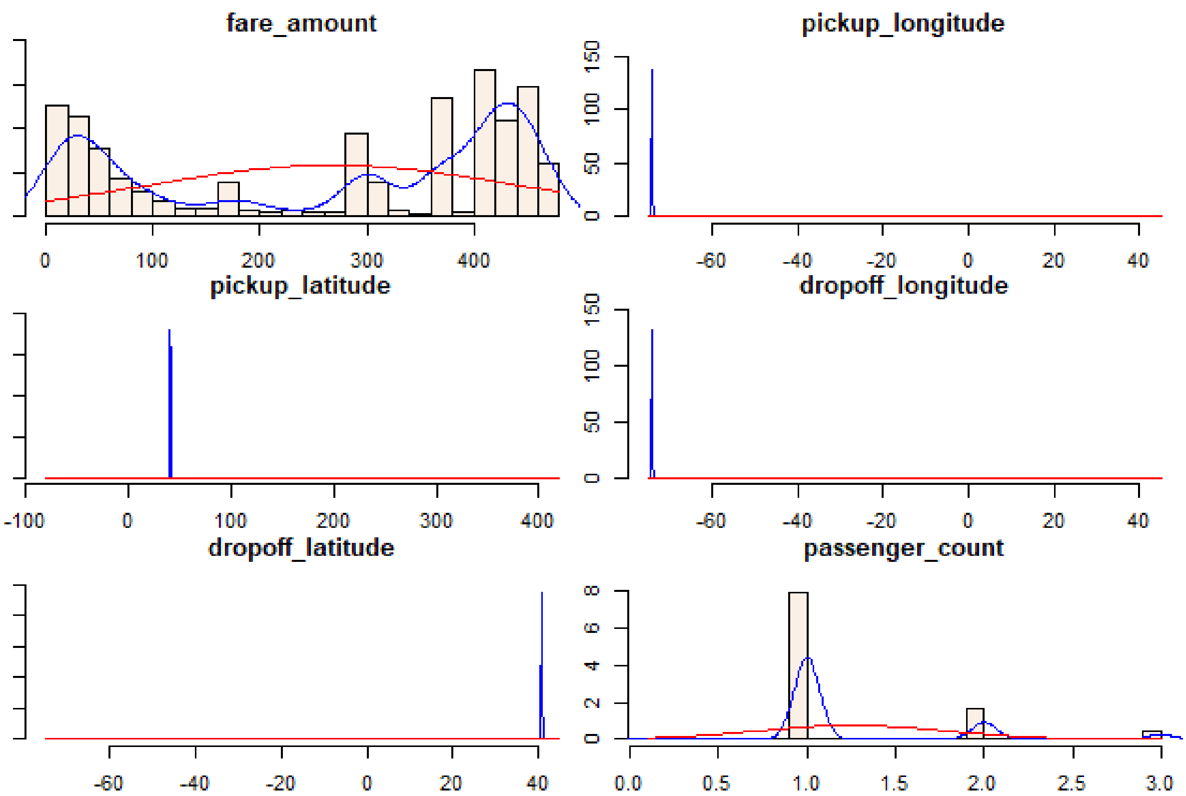
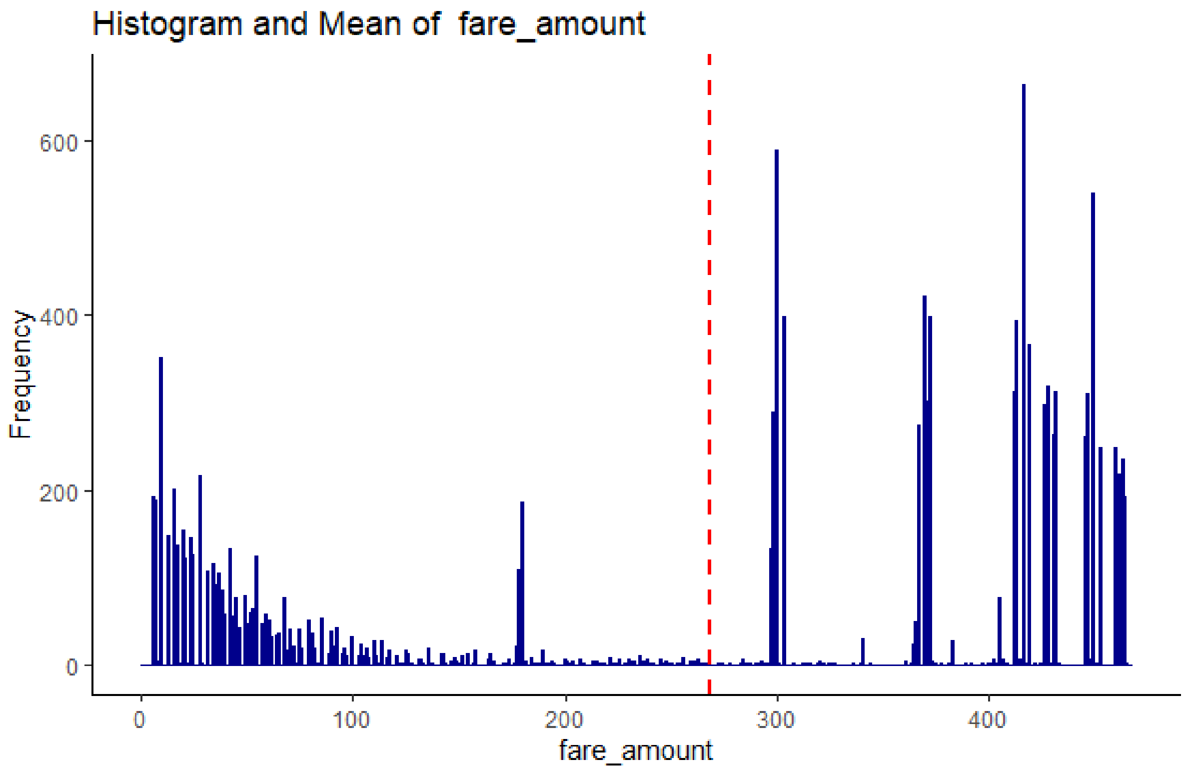
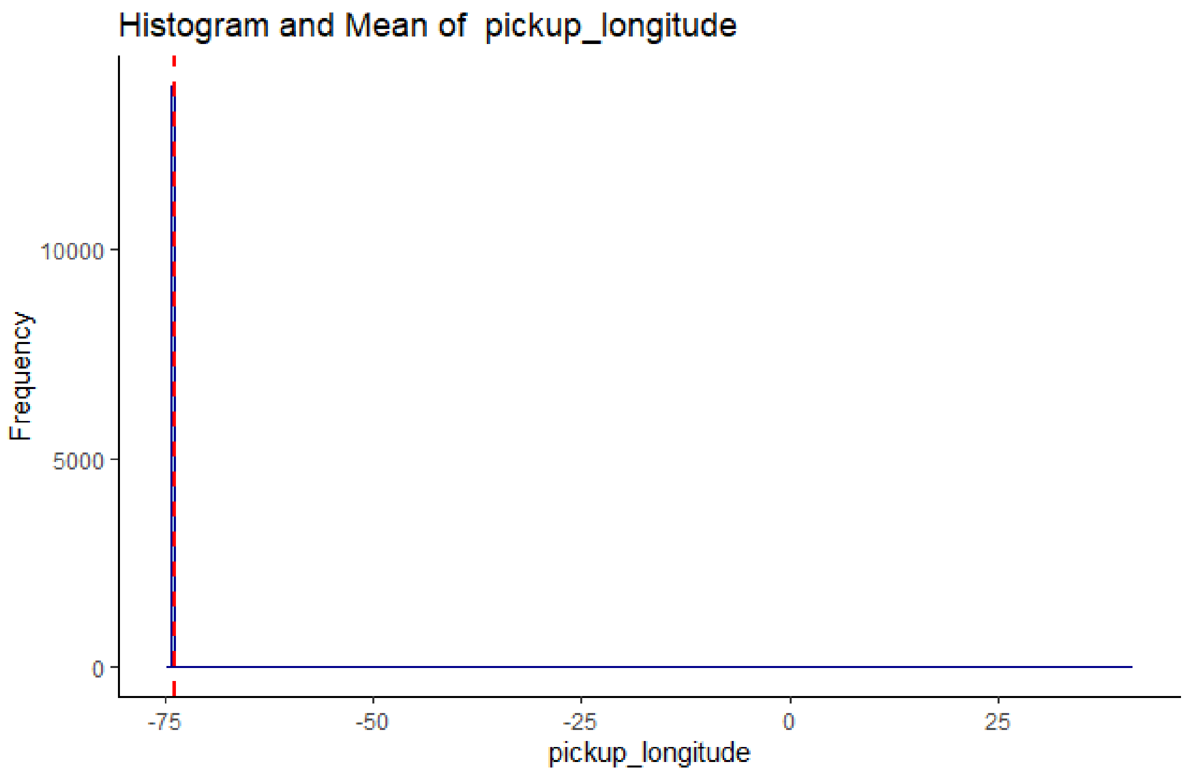
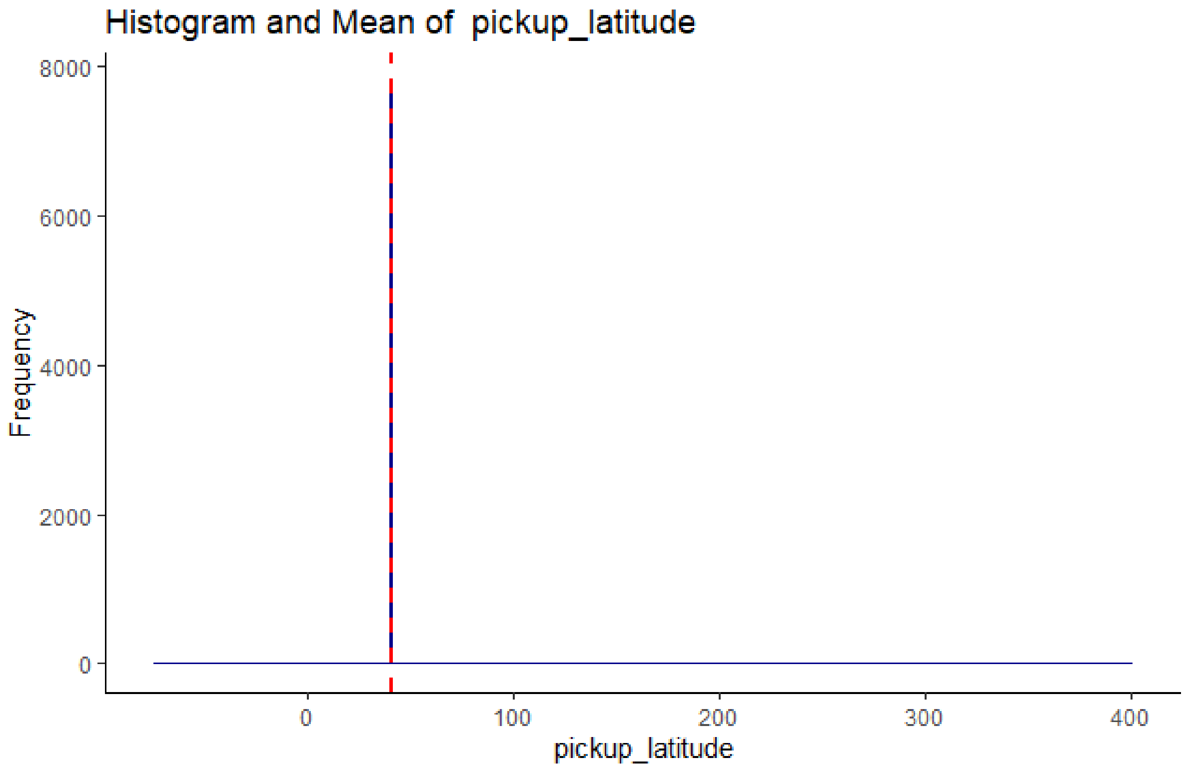


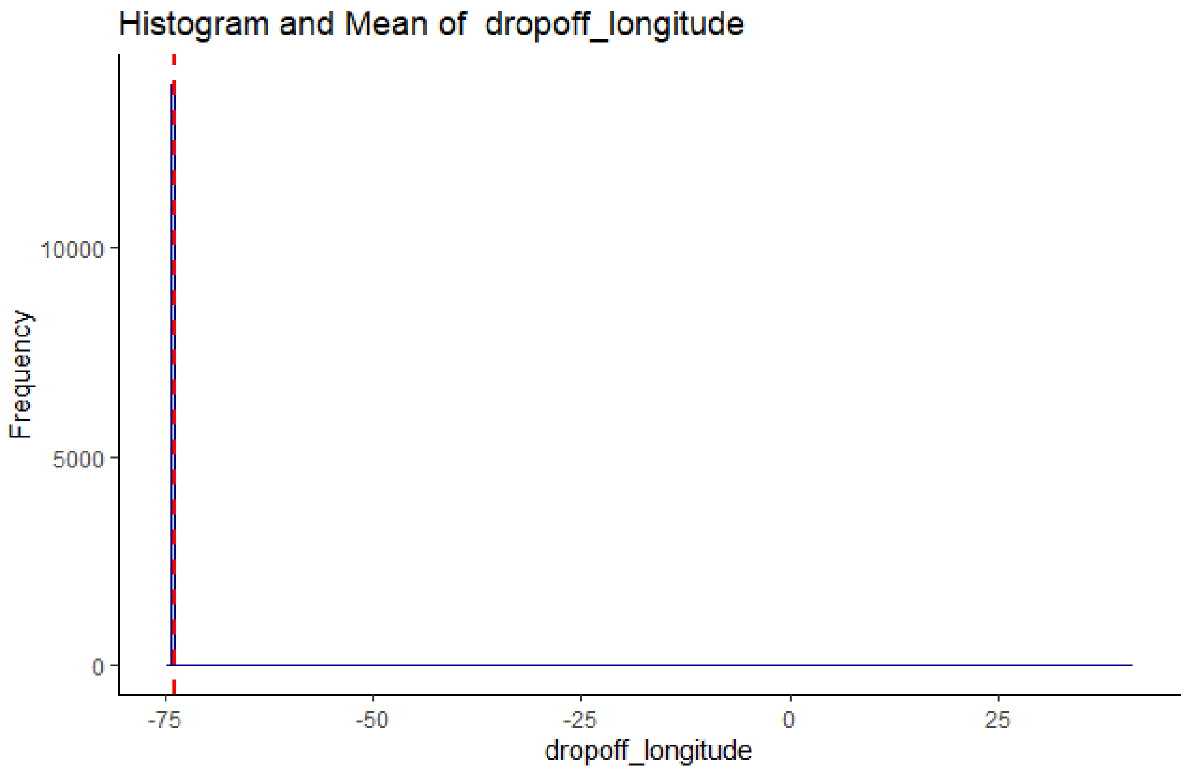
Fig 2.2.a.1 PDF with Normal

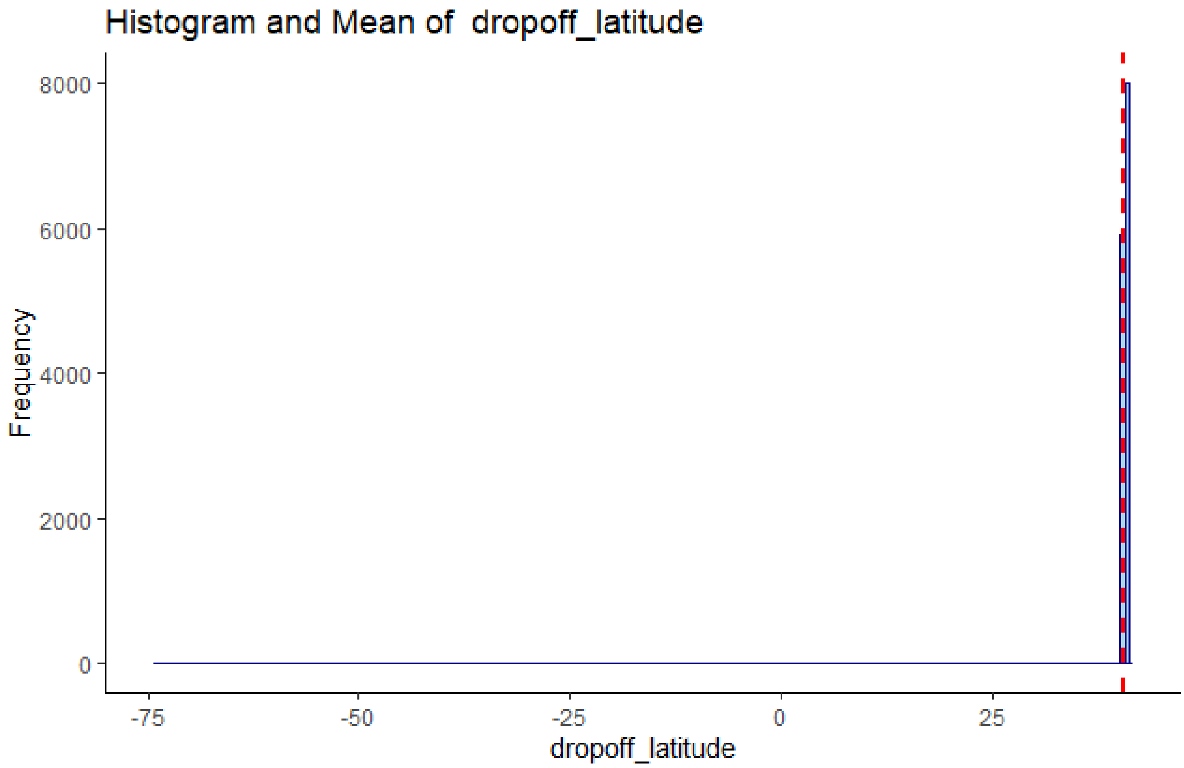
b. Histogram and Mean of Predictors






Histogram and Mean of pickup_latitude
8000
6000
4000
2000
0
Frequency
0
100
200
300
400
pickup_latitude


Histogram and Mean of dropoff_longitude
10000
5000
0
Frequency
-75
-50
-25
0
25
dropoff_longitude



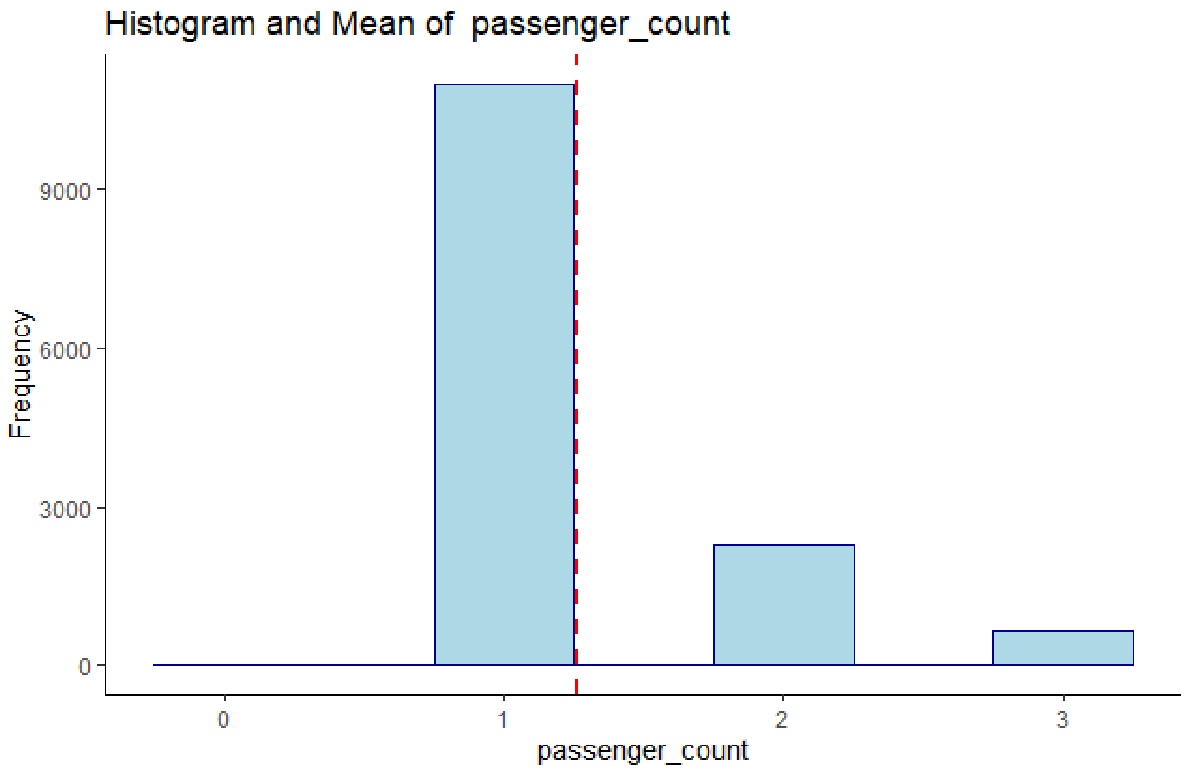


Fig 2.2.b.1 Histogram with Mean line

Modelling

In the above analysis done by us using various methods, we have been able to identify that the variable *fare_amount* is our target variable. Hence, we only need to predict that variable using different methods. Given, our target variable is continuous (converted into continuous from encoded categorical), we will be relying only on various models to make our predictions. Below are the models we have used in our project.

1. Training the Model

We will be starting with the most basic of prediction models and see how this goes and then jump onto complex models like random forest.

Multiple Linear Regression

Considering that we have multiple predictors, it is safe and efficient to use multiple regression on our variables in order to get a prediction model out of them.

Output of MLR

```
lm(formula = fare_amount ~ ., data = dsPredictors)

Residuals:
Min      1Q  Median      3Q     Max
-267.72 -208.63   71.86  157.28  282.71

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)       164.2615   111.9022   1.468    0.142
pickup_longitude   -3.0052     4.1818  -0.719    0.472
pickup_latitude    -0.2432     0.4765  -0.510    0.610
dropoff_longitude   1.3025     2.5994   0.501    0.616
dropoff_latitude   -0.2795     3.3195  -0.084    0.933
passenger_count    -0.2531     2.7222  -0.093    0.926

Residual standard error: 172.1 on 13946 degrees of freedom
Multiple R-squared:  0.0003745,      Adjusted R-squared:  1.614e-05
F-statistic: 1.045 on 5 and 13946 DF,  p-value: 0.3891
```

It is quite evident from the output that multiple linear regression has significantly failed. Multiple R-squared value suggests that we can explain about 0.03% of the data. Also, the p-value is greater than 0.05 which means that not even a single variable is dependent on the target. We should confirm this hypothesis from correlation matrix.

Fig 3.1.a.1 Correlation Matrix

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
fare_amount	1.0000000000	-0.018337422	0.008461619	-0.017056565	0.017912028	-0.0008454372
pickup_longitude	-0.0183374216	1.000000000	-0.643833625	0.978251134	-0.985744429	0.0037213911
pickup_latitude	0.0084616190	-0.643833625	1.000000000	-0.629971458	0.642246478	-0.0049977123
dropoff_longitude	-0.0170565651	0.978251134	-0.629971458	1.000000000	-0.964453635	0.0028407389
dropoff_latitude	0.0179120279	-0.985744429	0.642246478	-0.964453635	1.000000000	-0.0064213122
passenger_count	-0.0008454372	0.003721391	-0.004997712	0.002840739	-0.006421312	1.0000000000

If we observe the first row/column then, we can clearly conclude that *fare_amount* (our target variable) is very poorly correlated with all our predictors and hence we cannot get a good prediction out of multiple linear regression model.

Hence, we will try with a much complex prediction model, i.e. Random Forests, in the next section.

Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

In the coming sub-sections, you will see the output in R after running training data through a random forest model.

Output of Random Forest

```
Call:
 randomForest(formula = fare_amount ~ ., data = dsPredictors)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 1

          Mean of squared residuals: 20575.19
                    % Var explained: 30.56
```

As we can clearly see, even random forest is giving a poor accuracy, only 30.56%. Although this result was with 500 trees (which is by default). Hence, we will try with number of trees 1000 and 1500 as well to see if we get better accuracy.

Output of Random Forest (with ntree=1000)

```
Call:
 randomForest(formula = fare_amount ~ ., data = dsPredictors,      ntree = 
1000)
               Type of random forest: regression
                     Number of trees: 1000
No. of variables tried at each split: 1

          Mean of squared residuals: 20518.71
                    % Var explained: 30.75
```

Output of Random Forest (with ntree=1500)

```
Call:
 randomForest(formula = fare_amount ~ ., data = dsPredictors,      ntree = 
1500)
               Type of random forest: regression
                     Number of trees: 1500
No. of variables tried at each split: 1

          Mean of squared residuals: 20585.57
                    % Var explained: 30.52
```

Clearly, increasing the number of trees has not changed anything, the model still cannot explain more than 30% of the data.

2. Conclusion

1. Model Evaluation

Now that we have already passed our training data through the prediction models, we will now determine which model should be actually used to predict our target variables. Selection of model can be done using various parameters, but in this project, we have decided to stick to *accuracy* of predictions as a measure to choose.

a. Root Mean Squared Error (RMSE)

To determine the accuracy of predictions, we will be using the RMSE (Root Mean Squared Error) method. Below is the formula used to calculate accuracy percentage, code of which was written in R:

$$100 - \frac{RMSE * 100}{Mean(Test\ Values\ of\ Target\ Variable)}$$

Accuracy of our Models (R Output)

```
"Accuracy of Multiple Linear Regression is 35.732844553607"
```

```
"Accuracy of Random Forest is 30.8021895178643"
```

b. Model Selection

We can see from the above results that multiple linear regression is giving slightly better result. Hence, going forward to the next section, we will use multiple linear regression model to predict our target variable.

Prediction

1. Model Testing

Now that we have already trained our models in the previous section, we will use this trained model (random forest, as concluded in the previous section) to predict our target variable (fare_amount) in the test dataset provided to us.

Fig 4.1.a Test Dataset

	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
1	2015-01-27 13:08:24 UTC	-73.97332	40.76381	-73.98143	40.74384	1
2	2015-01-27 13:08:24 UTC	-73.98686	40.71938	-73.99889	40.73920	1
3	2011-10-08 11:53:44 UTC	-73.98252	40.75126	-73.97965	40.74614	1
4	2012-12-01 21:12:12 UTC	-73.98116	40.76781	-73.99045	40.75164	1
5	2012-12-01 21:12:12 UTC	-73.96605	40.78977	-73.98856	40.74443	1
6	2012-12-01 21:12:12 UTC	-73.96098	40.76555	-73.97918	40.74005	1
7	2011-10-06 12:10:20 UTC	-73.94901	40.77320	-73.95962	40.77089	1
8	2011-10-06 12:10:20 UTC	-73.77728	40.64664	-73.98508	40.75937	1
9	2011-10-06 12:10:20 UTC	-74.01410	40.70964	-73.99511	40.74137	1
10	2014-02-18 15:22:20 UTC	-73.96958	40.76552	-73.98069	40.77072	1
11	2014-02-18 15:22:20 UTC	-73.98937	40.74197	-73.99930	40.72253	1
12	2014-02-18 15:22:20 UTC	-74.00161	40.74089	-73.95639	40.76744	1
13	2010-03-29 20:20:32 UTC	-73.99120	40.73994	-73.99717	40.73527	1
14	2010-03-29 20:20:32 UTC	-73.98203	40.76272	-74.00187	40.76154	1
15	2011-10-06 03:59:12 UTC	-73.99246	40.72870	-73.98340	40.75015	1

We have run the above dataset against our random forest prediction model developed from the previous section to get Fig 4.1.c.

Fig 4.1.b R Code to predict fare_amount

```
ds_Test <- read.csv("C:/Users/riddh/OneDrive/Documents/Edwisor/Cab Fare Prediction/test.csv", sep = ",", header = T)

ds_Final <- ds_Test

ds_Final$fare_amount <- predict(mlrmodel, newdata=ds_Final[,2:6])
```

Fig 4.1.c Final Dataset with Target Variable

	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	fare_amount
1	2015-01-27 13:08:24 UTC	-73.97332	40.76381	-73.98143	40.74384	1	298.3332
2	2015-01-27 13:08:24 UTC	-73.98686	40.71938	-73.99889	40.73920	1	320.5683
3	2011-10-08 11:53:44 UTC	-73.98252	40.75126	-73.97965	40.74614	1	318.5210
4	2012-12-01 21:12:12 UTC	-73.98116	40.76781	-73.99045	40.75164	1	310.7863
5	2012-12-01 21:12:12 UTC	-73.96605	40.78977	-73.98856	40.74443	1	198.8777
6	2012-12-01 21:12:12 UTC	-73.96098	40.76555	-73.97918	40.74005	1	268.0087
7	2011-10-06 12:10:20 UTC	-73.94901	40.77320	-73.95962	40.77089	1	307.0011
8	2011-10-06 12:10:20 UTC	-73.77728	40.64664	-73.98508	40.75937	1	350.8743
9	2011-10-06 12:10:20 UTC	-74.01410	40.70964	-73.99511	40.74137	1	151.3052
10	2014-02-18 15:22:20 UTC	-73.96958	40.76552	-73.98069	40.77072	1	319.3669
11	2014-02-18 15:22:20 UTC	-73.98937	40.74197	-73.99930	40.72253	1	285.5138
12	2014-02-18 15:22:20 UTC	-74.00161	40.74089	-73.95639	40.76744	1	157.0114
13	2010-03-29 20:20:32 UTC	-73.99120	40.73994	-73.99717	40.73527	1	319.8817
14	2010-03-29 20:20:32 UTC	-73.98203	40.76272	-74.00187	40.76154	1	312.7466
15	2011-10-06 03:59:12 UTC	-73.99246	40.72870	-73.98340	40.75015	1	302.0007

Appendix

1. R Code

a. missingValueAnalysis.R

```
ds <- read.csv("C:/Users/riddh/OneDrive/Documents/Edwisor/Cab Fare Prediction/train_cab.csv",
sep = ",", header = T, na.strings = c(" ", "", "NA","0"))

ds_missingVal = data.frame(apply(ds,2,function(x){sum(is.na(x))}))
ds_missingVal$Columns = row.names(ds_missingVal)
names(ds_missingVal)[1] =  "Missing_percentage"
ds_missingVal$Missing_percentage = (ds_missingVal$Missing_percentage/nrow(ds)) * 100
ds_missingVal = ds_missingVal[order(-ds_missingVal$Missing_percentage),]
row.names(ds_missingVal) = NULL
ds_missingVal = ds_missingVal[,c(2,1)]
write.csv(ds_missingVal, "Missing_perc.csv", row.names = F)

ds_new <- na.omit(ds)
ds_new = ds_new[,c(2,1,3,4,5,6,7)]

for (i in 2:7){
  if(class(ds_new[,i]) == "factor")
  {
    ds_new[,i] <- as.numeric(ds_new[,i])
  }
}

write.csv(ds_new, "new_trainCab.csv", row.names = F)
```

b. outlierAnalysis.R

```
library(psych)
library(ggplot2)

drawBoxP <- function(df){
  for (i in 2:7){
    title_val <- paste("Boxplot (without Outliers) of ",colnames(df[i])) # paste("Boxplot without 
Outliers of ",colnames(df[i]))

    box_plot <- ggplot(df, aes(x='', y= df[,i])) +
      geom_boxplot(outlier.color = "red", outlier.shape = 19,
                   outlier.size = 1.5, outlier.stroke = 0.5) +
      labs(title=title_val,x=colnames(df[i]), y = "Frequency") +
      geom_jitter(shape=16, position=position_jitter(0.2)) #with jitter
    print(box_plot)
  }
}
#boxplot with outliers
drawBoxP(ds_new)
#removing outliers from passenger_count
unwanted_outliers <- boxplot(ds_new$passenger_count, plot = FALSE)$out
dsOutRemoved <- ds_new[-which(ds_new$passenger_count %in% unwanted_outliers ),]
drawBoxP(dsOutRemoved)

#effect of outliers

#with outliers

boxplot(ds_new$passenger_count, main="Boxplot for passenger_count with outliers",
        ylab="passenger_count")
hist(ds_new$passenger_count, main="Histogram for passenger_count with outliers",
     xlab="passenger_count")
#without outliers
boxplot(dsOutRemoved$passenger_count, main="Boxplot for passenger_count without outliers",
        ylab="passenger_count")
hist(dsOutRemoved$passenger_count, main="Histogram for passenger_count without outliers",
     xlab="passenger_count")
```

c. dataVisualization.R

```
library(psych)
library(ggplot2)

#Probability Density Funciton with histogram and normal fit
multi.hist(dsOutRemoved[,2:7], main = NULL, dcol = c("blue", "red"),
          dlty = c("solid", "solid"), bcol = "linen")


#histogram with mean line
for (i in 2:7){

  title_val <- paste("Histogram and Mean of ",colnames(dsOutRemoved[i]))
  hist_plot<-ggplot(dsOutRemoved, aes(x=dsOutRemoved[,i])) +
    geom_histogram(binwidth=0.5,color="darkblue", fill="lightblue",
                   linetype="solid")+
    labs(title=title_val,x=colnames(dsOutRemoved[i]), y = "Frequency")+
    theme_classic() +
    geom_vline(aes(xintercept=mean(dsOutRemoved[,i])),
               color="red", linetype="dashed", size=1)

  print(hist_plot)
}
```

d. mlr_rf.R

```
library(tidyverse)
library(randomForest)
library(caret)

dsPredictors <- dsOutRemoved[,2:7]
mlrmodel <- lm(fare_amount ~., data = dsPredictors)
summary(mlrmodel)
rfmodel <- randomForest(fare_amount ~ ., data = dsPredictors)
print(rfmodel)

trainIndex = createDataPartition(dsPredictors$fare_amount, p = .80, list = FALSE)
dsTest <- dsPredictors[-trainIndex,1:6]
predictionsRF_DT = predict(rfmodel, dsTest[,2:6])
predictionsMLR_DT = predict(mlrmodel, dsTest[,2:6])

AccuracyMLR<- 100 - RMSE(predictionsMLR_DT,
dsTest$fare_amount)*100/mean(dsTest$fare_amount)
print(paste("Accuracy of Multiple Linear Regression is",AccuracyMLR))

AccuracyRF <- 100 - RMSE(predictions_DT, dsTest$fare_amount)*100/mean(dsTest$fare_amount)
print(paste("Accuracy of Random Forest is",AccuracyRF))
```

e. predict.R

```
ds_Test <- read.csv("C:/Users/riddh/OneDrive/Documents/Edwisor/Cab Fare Prediction/test.csv",
sep = ",", header = T)
ds_Final <- ds_Test
ds_Final$fare_amount <- predict(mlrmodel, ds_Final[,2:6])
```

2. Instructions to Run the Code

To run the code, follow the below steps: -

- Please open the R Studio.
- Ensure that all the packages required in all the code snippets above are installed in the R Studio.
- Wherever 'csv' file location is mentioned, please change the location as per the presence of that file in the machine used to run the R script.
- The order in which these scripts must be run is pretty straightforward, when all the above steps are cross-checked, just the above scripts in alphabetical order, that is, see below -

```
a. missingValueAnalysis.R -> b. outlierAnalysis.R -> c. dataVisualization.R -> d. mlr_rf.R -> e. predict.R
```

References

www.geeksforgeeks.org

www.wikipedia.org

www.towardsdatascience.com

www.medium.com

www.data-flair.training

www.sthda.com

www.r-graph-gallery.com

www.machinelearningmastery.com

www.ucl.ac.uk