Bike Rental Project

Pavan Kumar Nagula

1st Aug 2019

Contents:

1. Introduction
    1.1 Problem Statement
    1.2 Data

2. Data Pre-Processing
    2.1 Missing Value Analysis
    2.2 Data Distribution
    2.2 Outlier Analysis
    2.4 Feature Selection
        2.4.1 Correlation Analysis

3. Model Deployment
    3.1 KNN Imputation Method
    3.2 Random Forest Regression
    3.3 Linear Regression

4. Conclusion
    4.1 Model Evaluation
    4.2 Model Selection

Appendix A – Extra Figures

Appendix B – R Code

Chapter 1

Introduction

1.1 Problem Statement

Prediction of bike rental count on daily based on the environment and seasonal settings. The main aim of this project is to predict the count of the bike rentals on daily basis based on the environment and seasonal conditions. So, by predicting the count we can figure out on which day or season the bike rentals will be higher, which helps the company to increase its bikes count on that particular season and also it helps in increasing the sales.

1.2 Data

The dataset contains about two years historical data related to the bike rental company in which we have 15 independent variables and 1 dependent variable. Basically, the data is all about to describe when the count of bike rentals increased like during the holidays or weekend or during the particular season. Based on that we can predict the count of bike rentals using the historical data. In this project as the target variable is a continuous variable, we need to build regression models based on which we can predict the count of the bike rentals. Below table will show the sample of the data set which we are going to use for predicting the count of the bike rentals.

Table 1: Sample Data set of Bike Rental data

Instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	Cnt
1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600

Table 2: Below table contains list of continuous/numeric variables which are used for predicting the target variable cnt.

Numeric Variables:

Numeric Variables
1. Temp
2. Atemp
3. Hum
4. Windspeed

Chapter 2

Data Pre-Processing

For any predictive modelling before passing the data into the model, we need to pre-process the data. Here data pre-processing means we need to check the distribution of the numeric variables which is mainly required for regression models, need to check if any missing values present in the data set & need to detect outliers using box plots and replace them with the missing value analysis. And, we need to select only particular features which provides meaningful information about the target variable, and also delete the variables or features which doesn’t provide much information about the target variable. And check if there is any correlation between the variables and drop those variables which are correlated with each other from the dataset.

2.1 Missing Value Analysis:

In the missing value analysis, we should check if any missing values like NA or blank values present in the data set and to replace them, we use mean method or median method or KNN method. Mostly missing values are generated due to human errors or due to optional box in questionnaire, etc… We can determine how much missing percentage a variable in the dataset contains based on the missing value percentage table. And we can drop that variable which contains the missing value percentage greater than 30%. In the mean method it will calculates the mean of the data excluding the NA values and replaces the missing values with the mean value. Median method is just as similar as mean method in this it will replaces the NA values with the median value. And coming to KNN imputation method it will replaces the NA value with its nearest neighbour of that particular variable by using the Euclidean method or Manhattan method.

But firstly, before applying any of those method check if there are any missing values present in the dataset. In this project as per our dataset values we couldn’t find any missing values in any of the variables in the dataset. I have mentioned a table below which show if any missing values are present in the dataset.

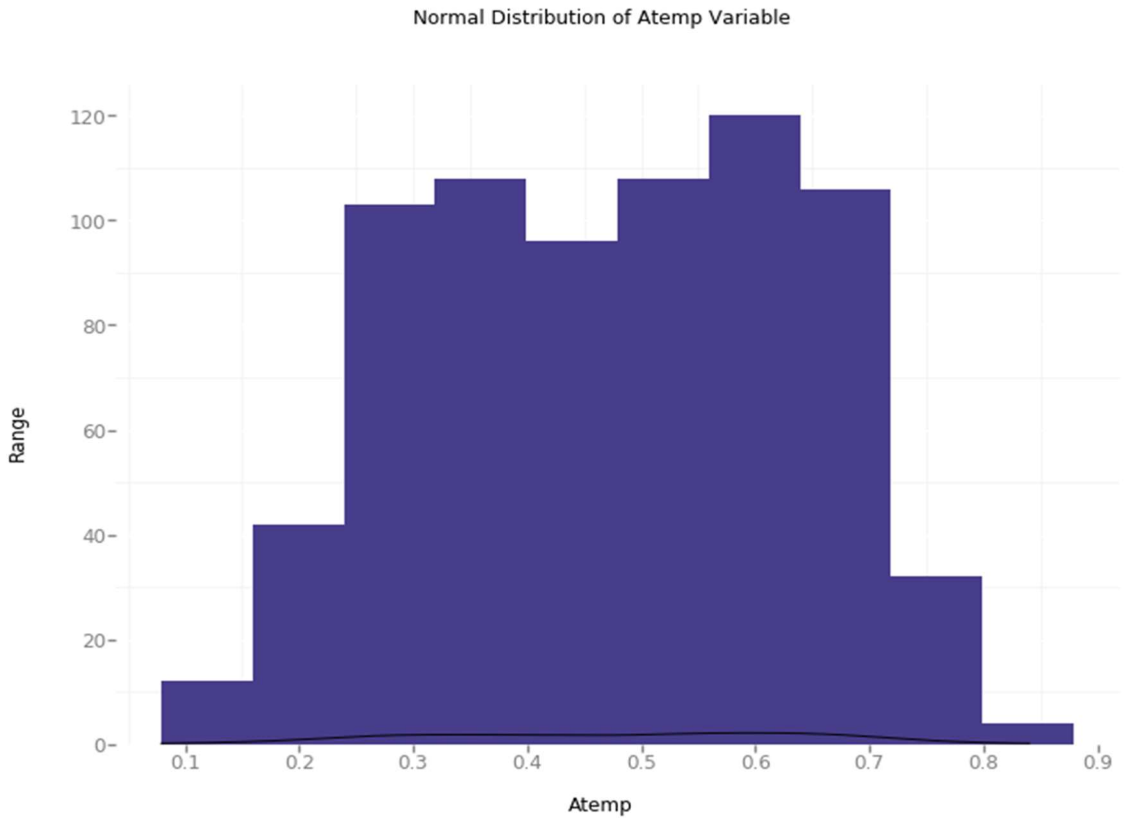
Table 3: Below is the table which shows that there are no missing values in the dataset.

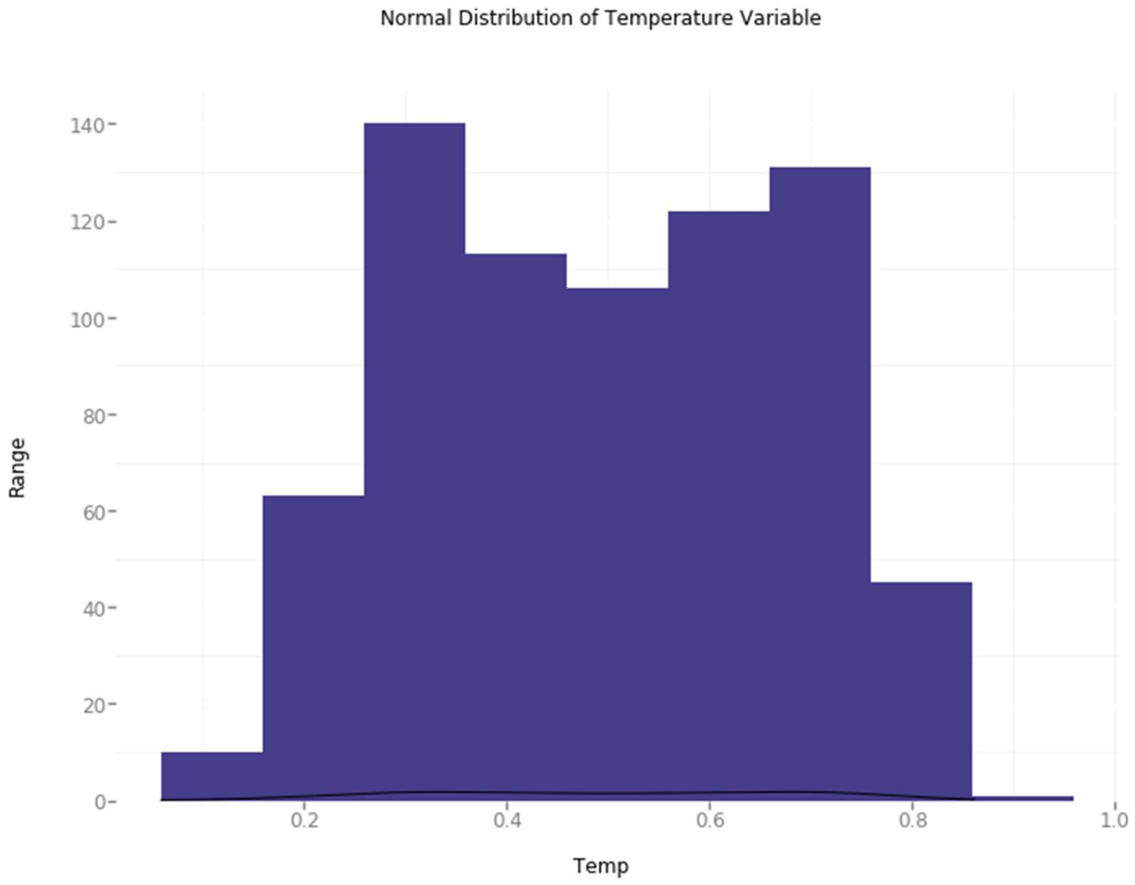
Variable	Missing values
instant	False
dteday	False
season	False
yr	False
mnth	False
holiday	False
weekday	False
workingday	False
weathersit	False
temp	False
atemp	False
hum	False
windspeed	False
casual	False
registered	False
cnt	False

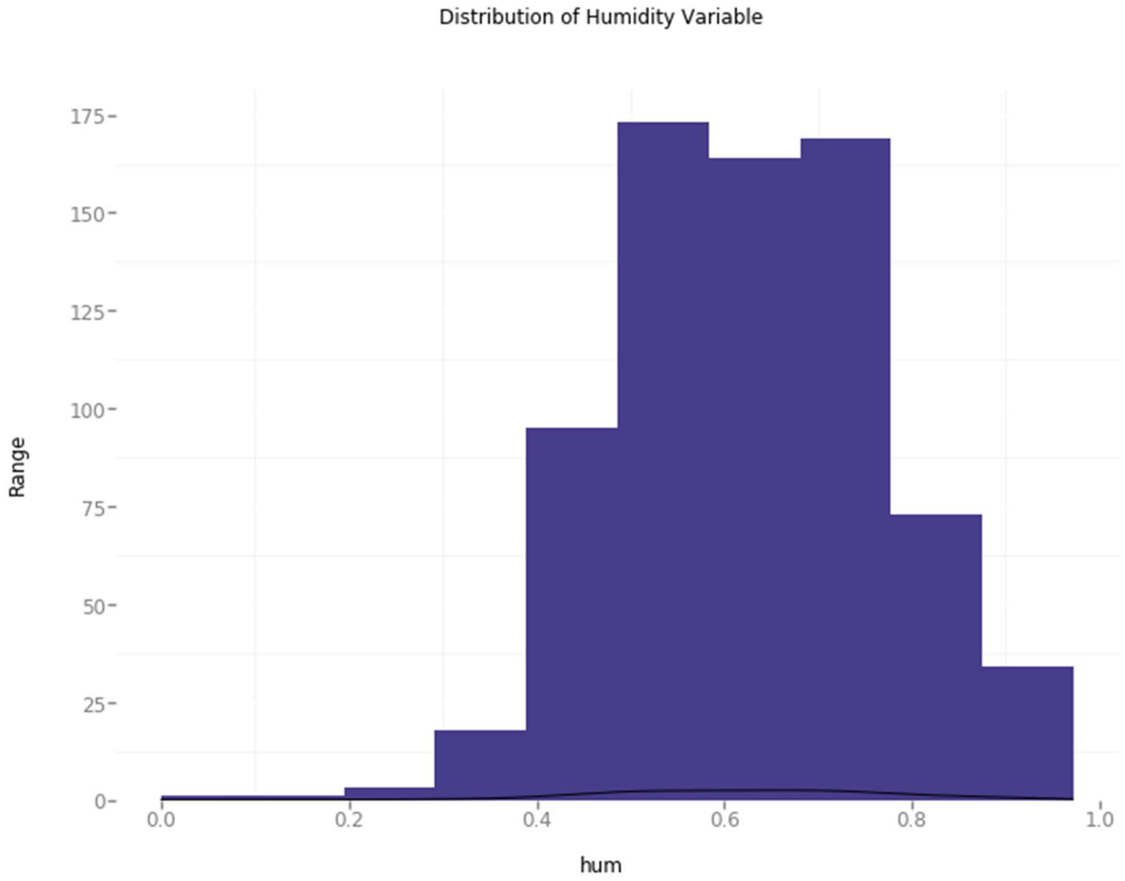
2.2 Data Distribution:

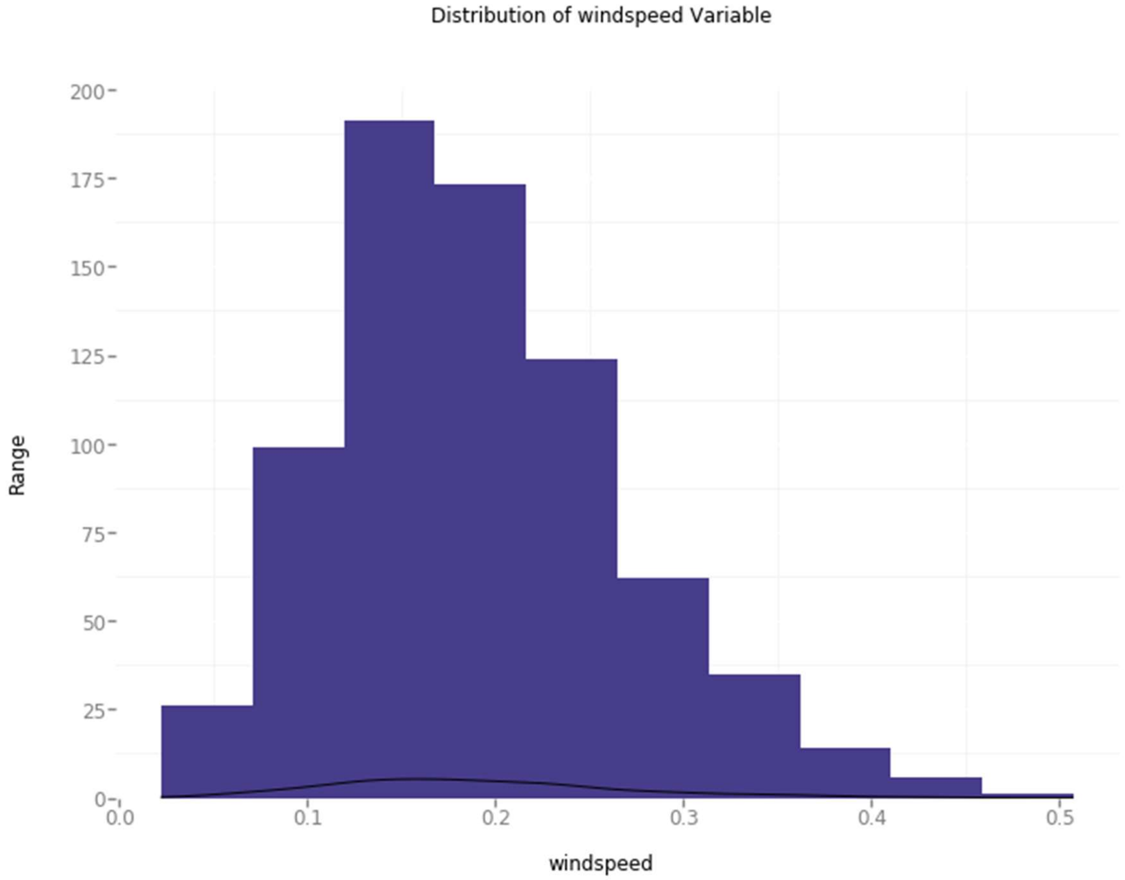
For the regression models we need to check the distribution of the variables before we pass the data into the model. Here the data is normally distributed in the Temp and Atemp variables. But there are skews observed in the Humidity and Windspeed variables. These skews are mostly occurred due to the presence of outliers in the data. So, below I have mentioned box plots for the detection of outliers in each variable and found outliers in Windspeed variable and Humidity variable and I have plotted the histogram plots of each variable from which we can know the distribution of the numeric variables like if the variable is normally distributed or not.

Distribution plots of numeric variables:

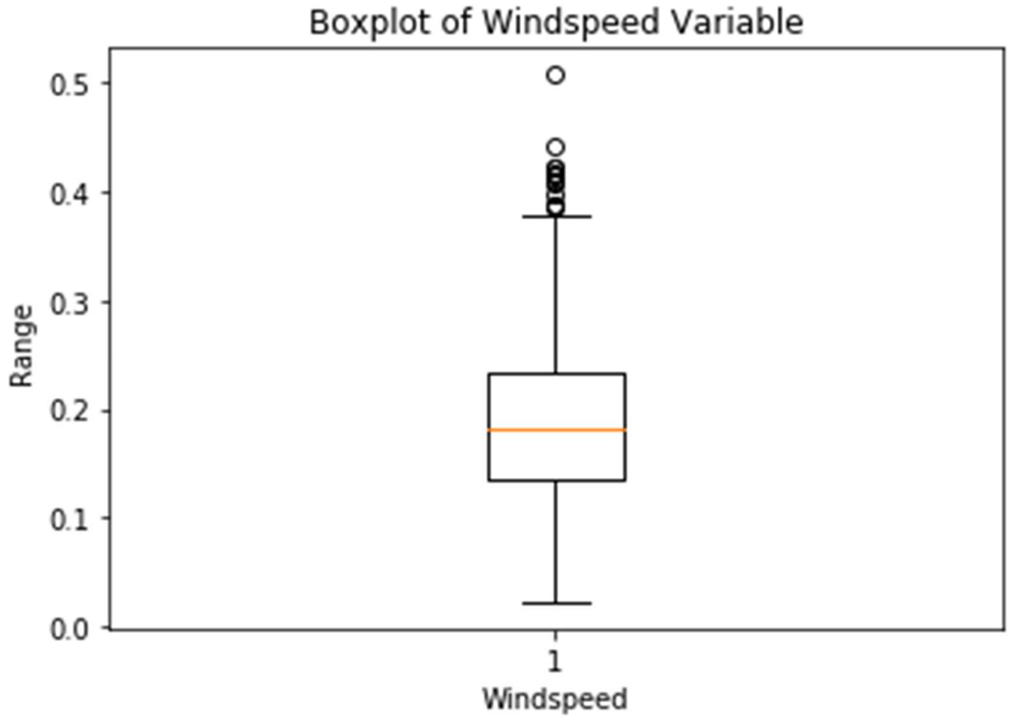


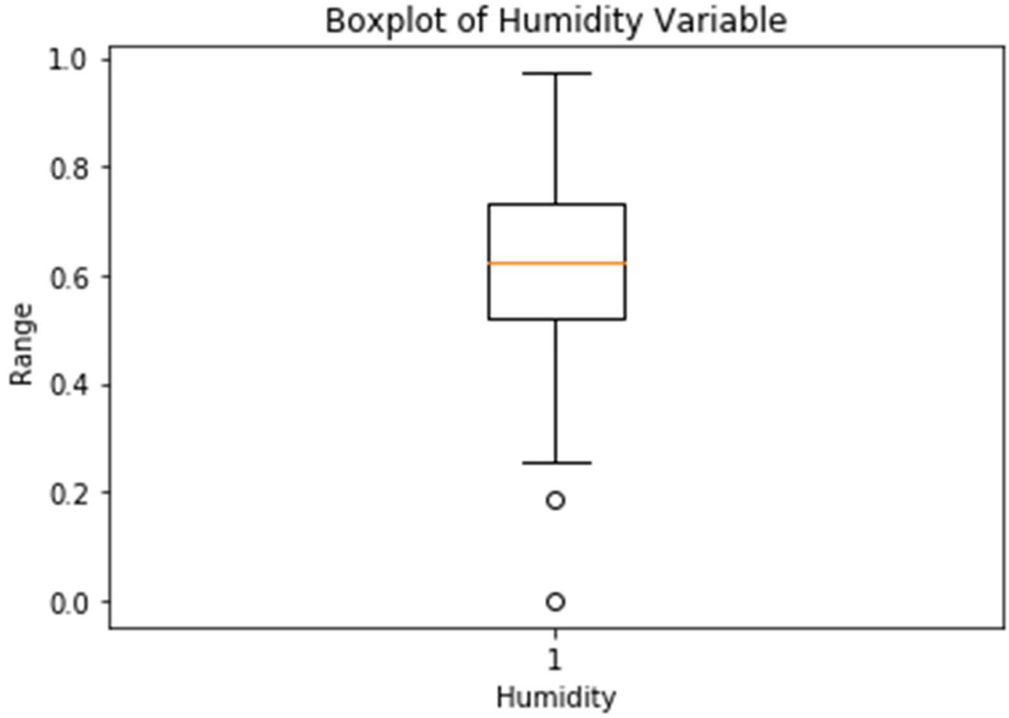


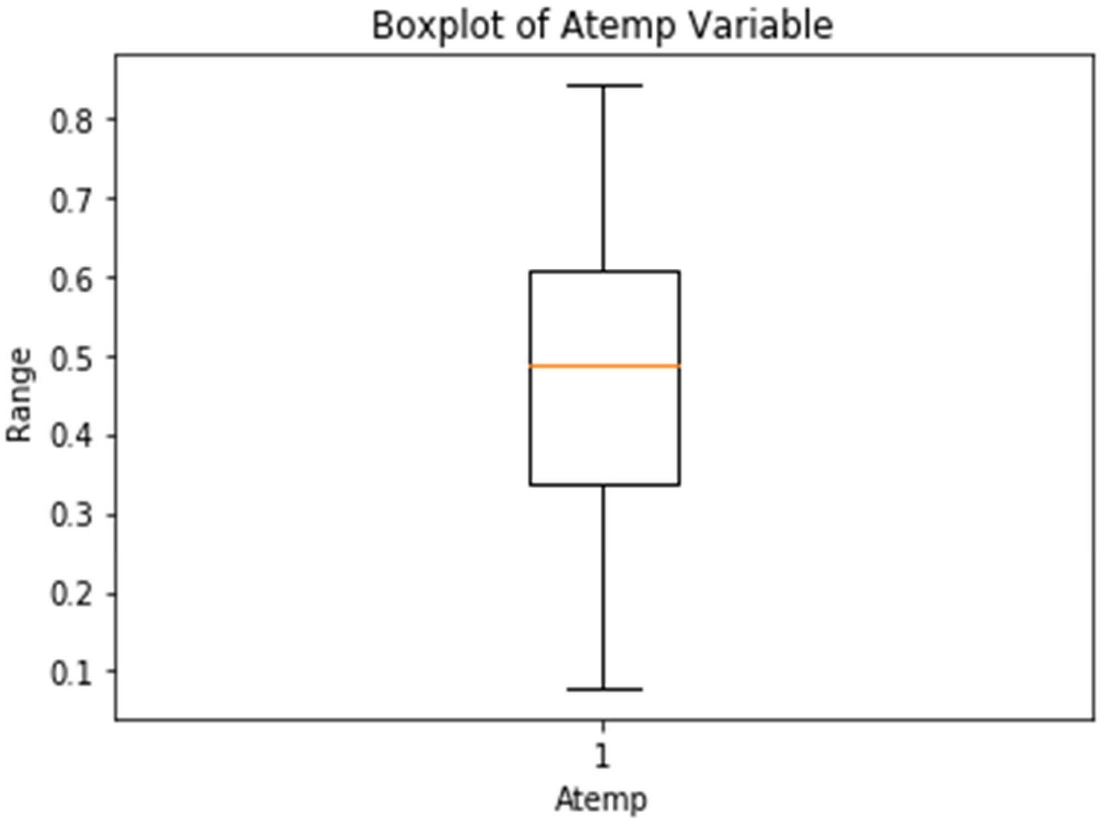

Distribution of Humidity Variable
Range
175
150
125
100
75
50
25
0
0.0
0.2
0.4
0.6
0.8
1.0
hum

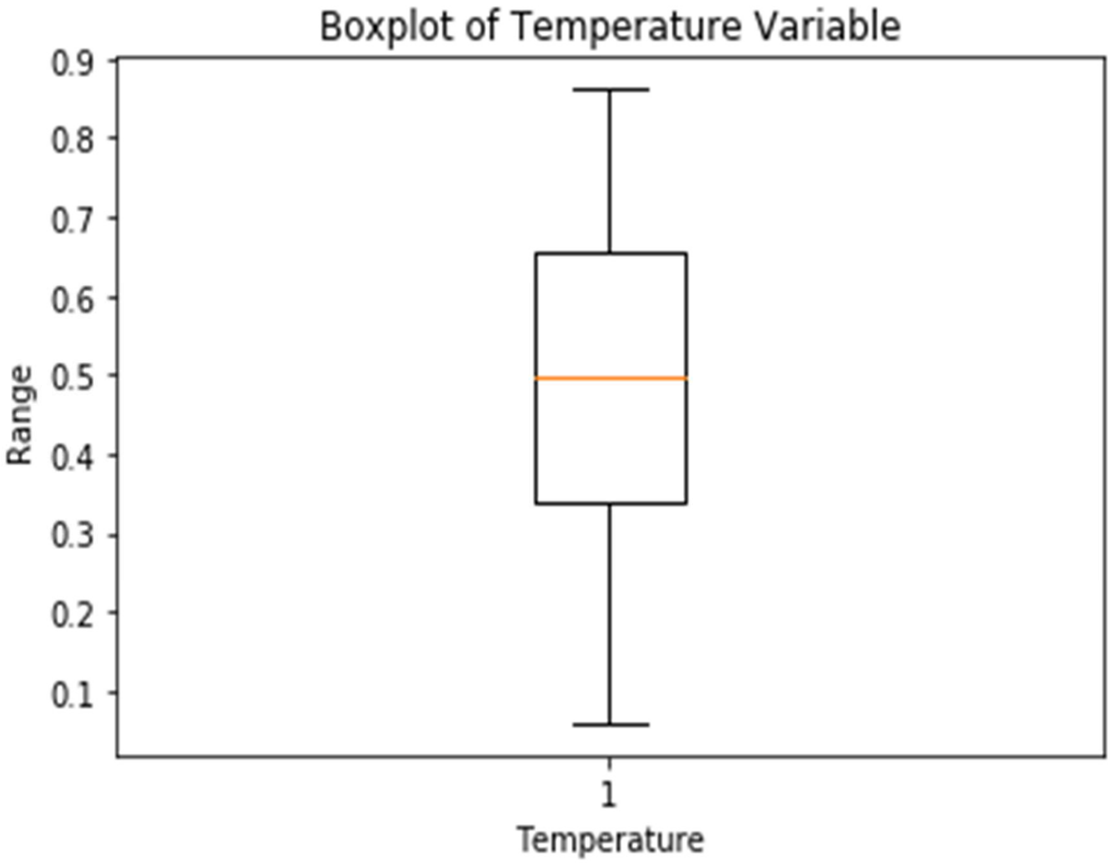

Distribution of windspeed Variable
Range
200
175
150
125
100
75
50
25
0
0.0
0.1
0.2
0.3
0.4
0.5
windspeed

Box Plots of each numeric variables:






Boxplot of Atemp Variable
Range
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
1
Atemp

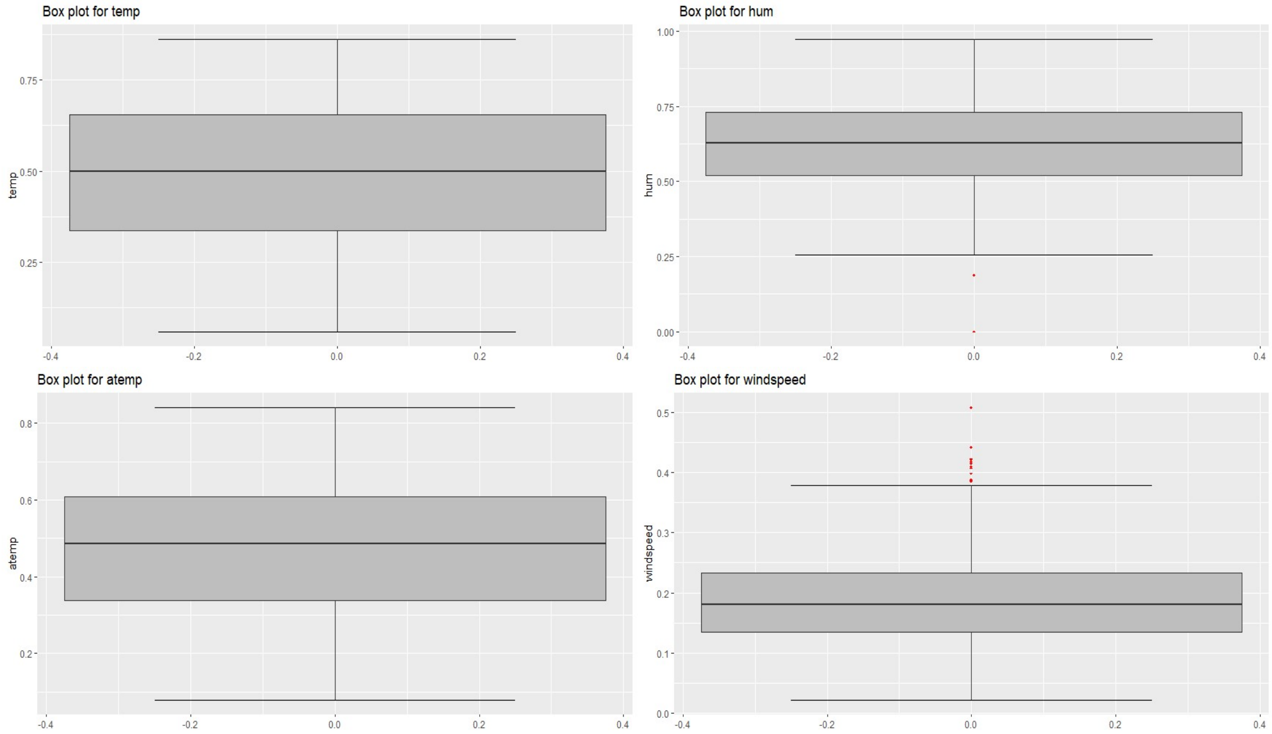

Boxplot of Temperature Variable
Range
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Temperature

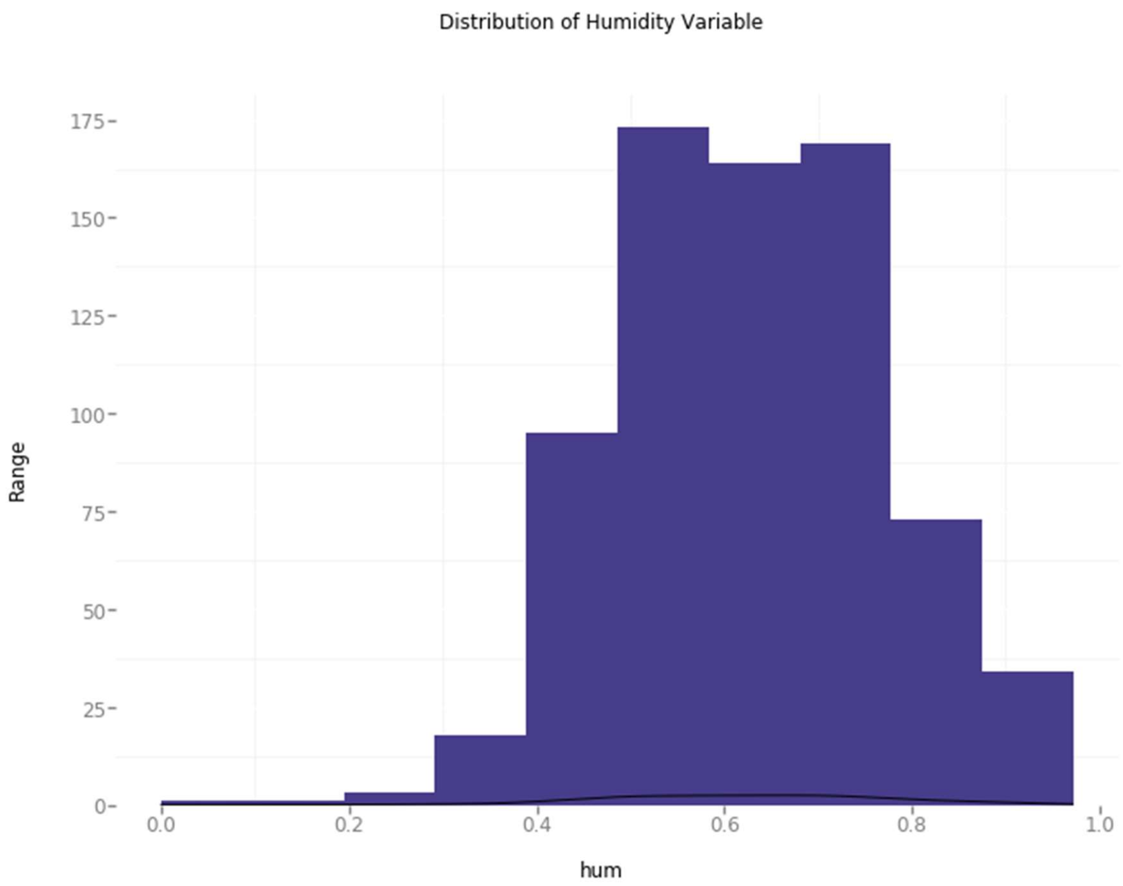
2.3 Outlier Analysis:

In the outlier analysis we used box plots for plotting the outliers in the numeric variables. From the below box plot graphs, we can say that in windspeed variable there are negative outliers and in the humidity variable there are positive outliers present in the data. So, after detecting the outliers I have deleted them. After that I have again plotted the box plots of the numeric variables and there are no outliers present in the dataset. As I said earlier due to the presence of outliers in the data the windspeed and hum variables are skewed, so after the removal of outliers again I have plotted the distribution graphs to check the normal distribution of the variables as from the below graphs the data is now normally distributed after the removal of outliers.

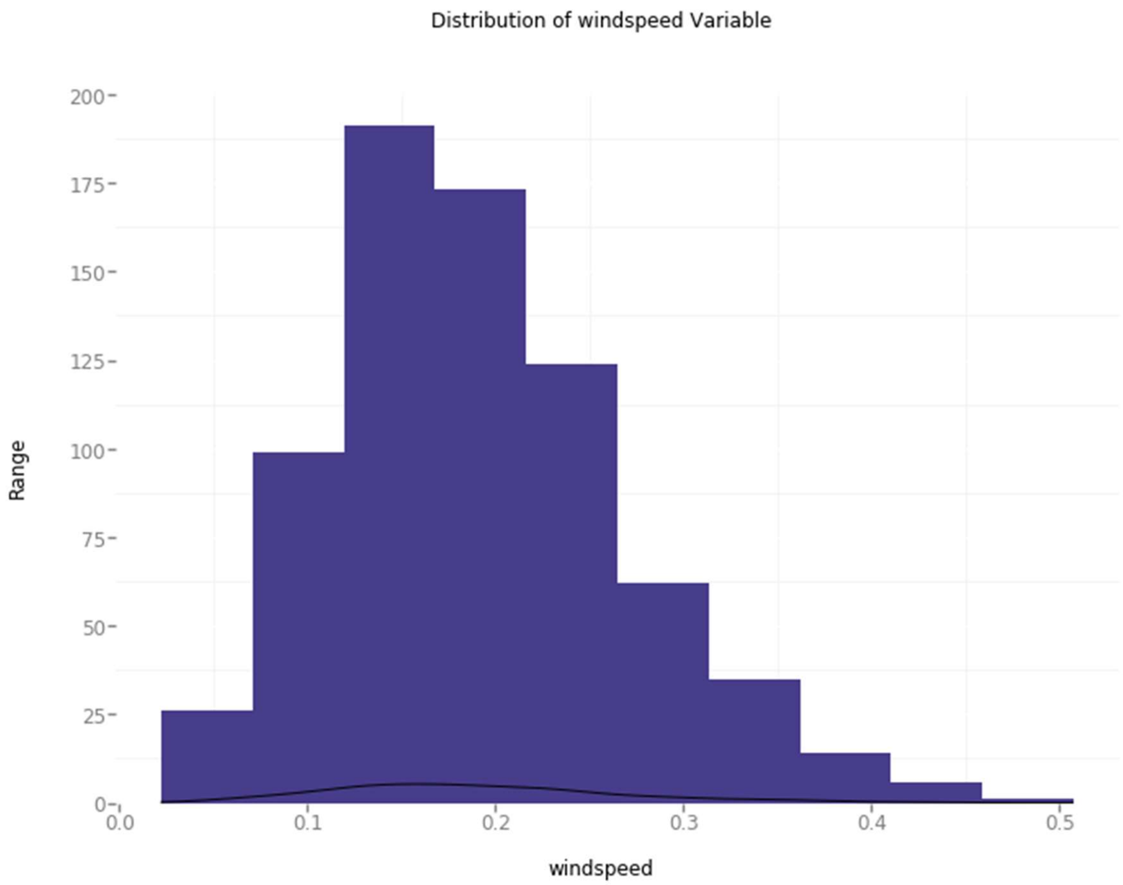
Below graphs shows how the distribution varies from the skewed to normal distribution after the removal of outliers.

With outliers:



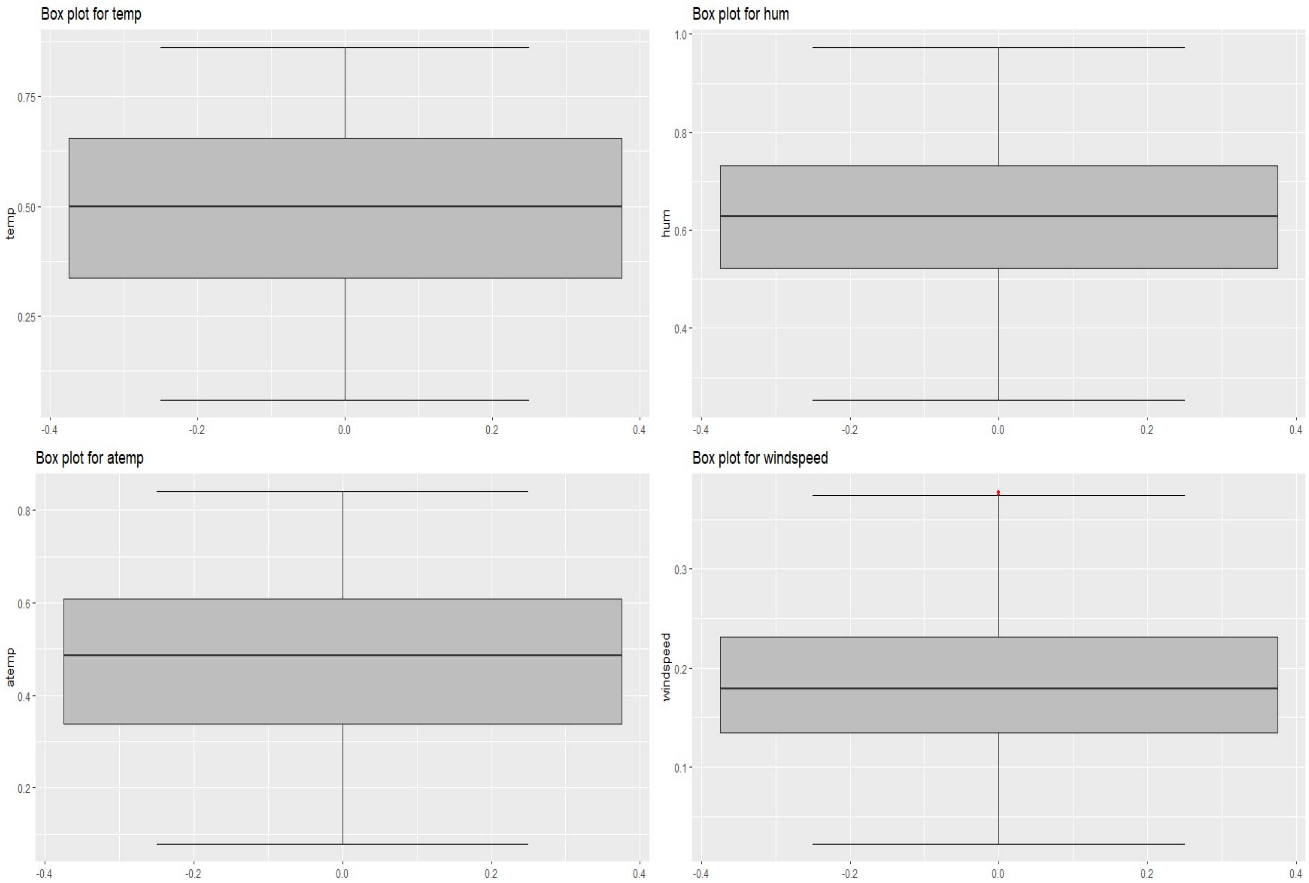


Distribution of Humidity Variable

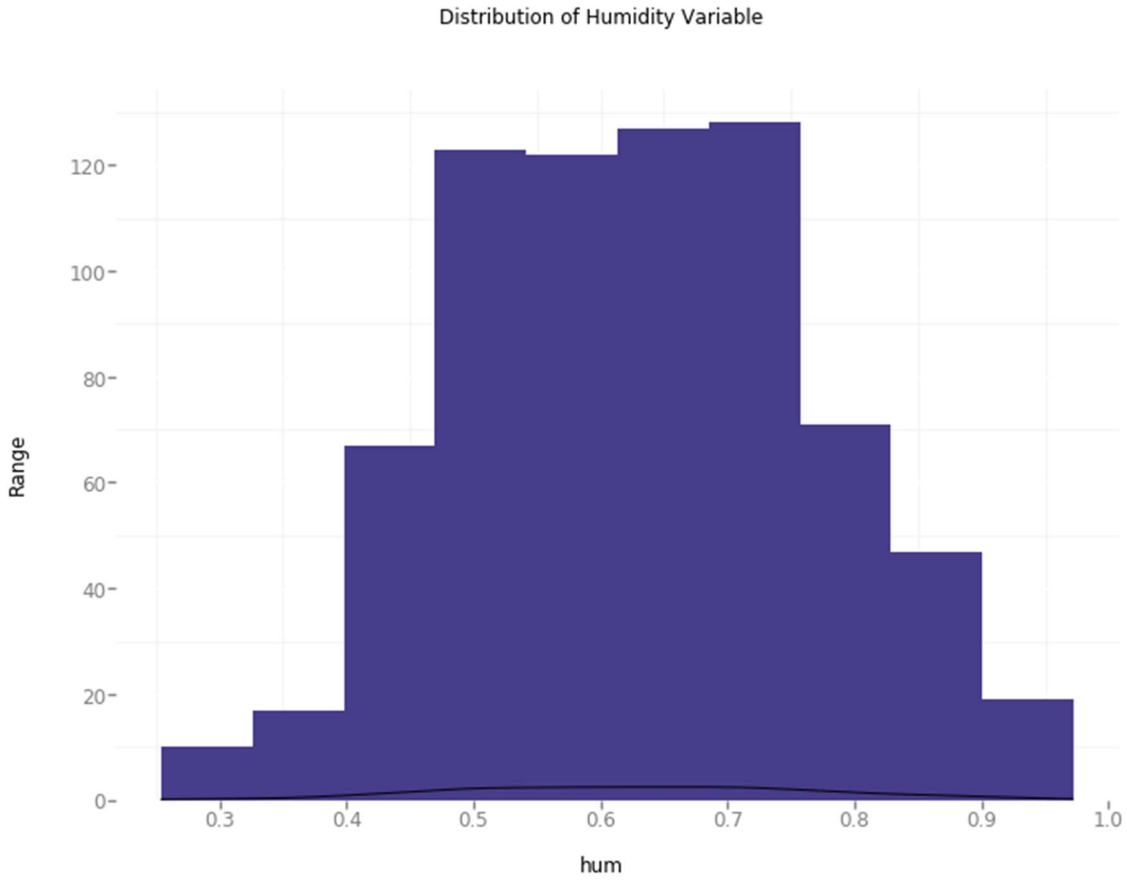


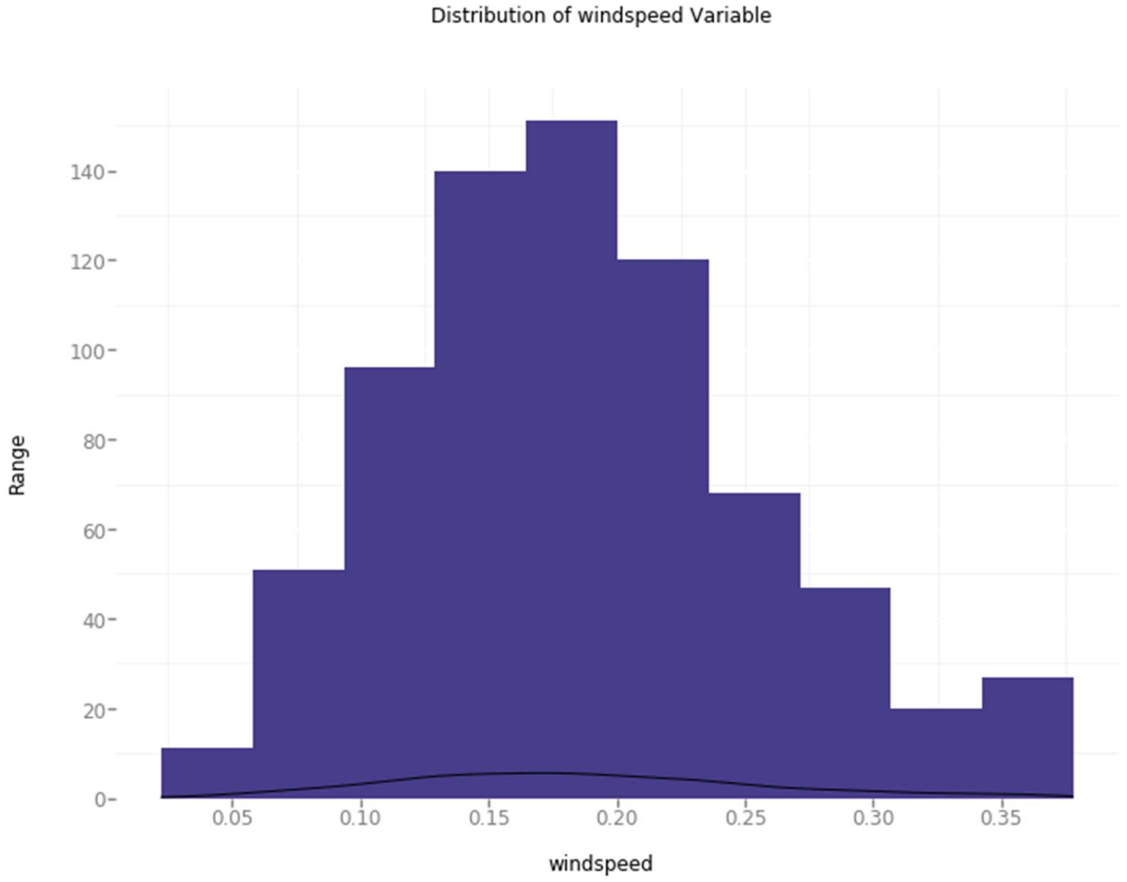
Distribution of windspeed Variable

Without Outliers:



As we can see from below graphs the skewed distribution of both humidity and windspeed is changed to normal distribution because of detecting and deleting those outliers in them.


Distribution of Humidity Variable

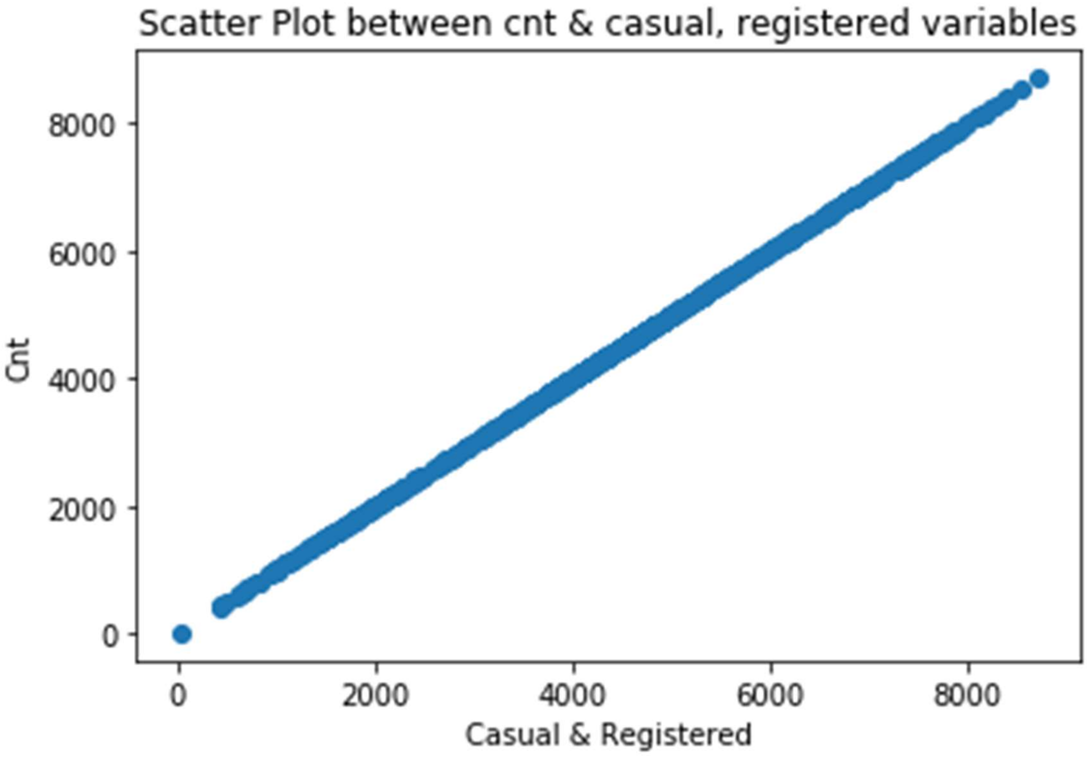

Distribution of windspeed Variable

2.4 Feature Selection

After the removal of all the missing values and outliers from the dataset we should not pass the data directly into the model, before that we need to check that if all the variables possess the important or valuable information about the target variable. There is a possibility that some of the variables in our dataset doesn't possess much information about the target variable. So, we can remove such type of variables in order to run the model smoothly.

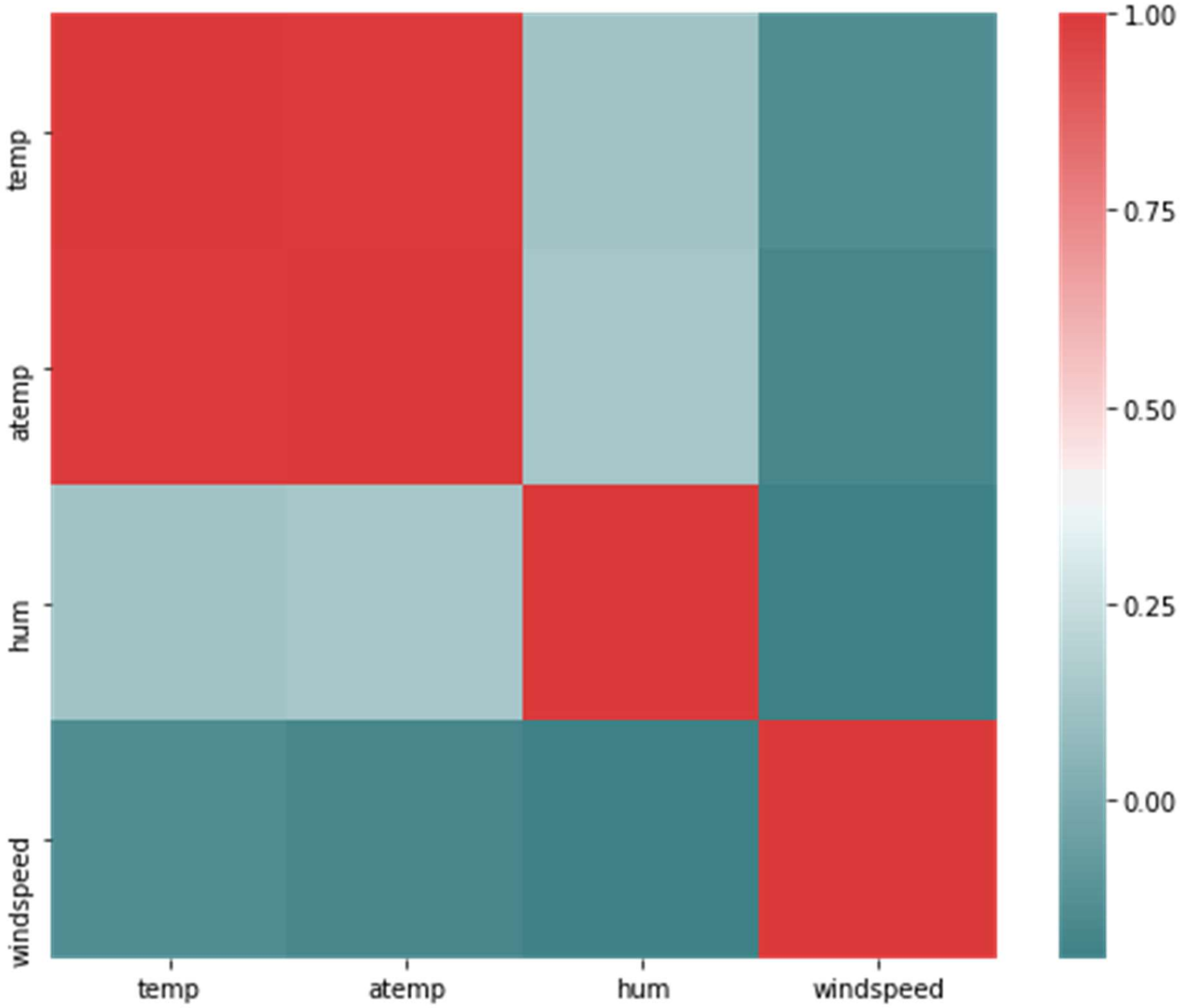
Firstly, in our dataset we can drop the instant variable as it doesn't contain much information about the target variable and there are already month, year and weekday columns so we can drop the dteday variable from the dataset.

Secondly, I have plotted a scatter plot between the casual, registered variables & cnt variable. As from the below scatter plot I can say that both the casual & registered variables are linearly correlated with the cnt variable. So, we can drop those two variables.



2.4.1 Correlation Analysis:

In the Correlation Analysis I have passed all the numeric variables into a variable and generated a correlation matrix using the corr() function. From that matrix I have plotted the correlation graph to check if any of the variables are correlated to each other. From the below graph we can say that temp variable and atemp variable are correlated to each other, So to avoid the multicollinearity I have to drop the atemp variable from the dataset.



Chapter 3

Model Deployment

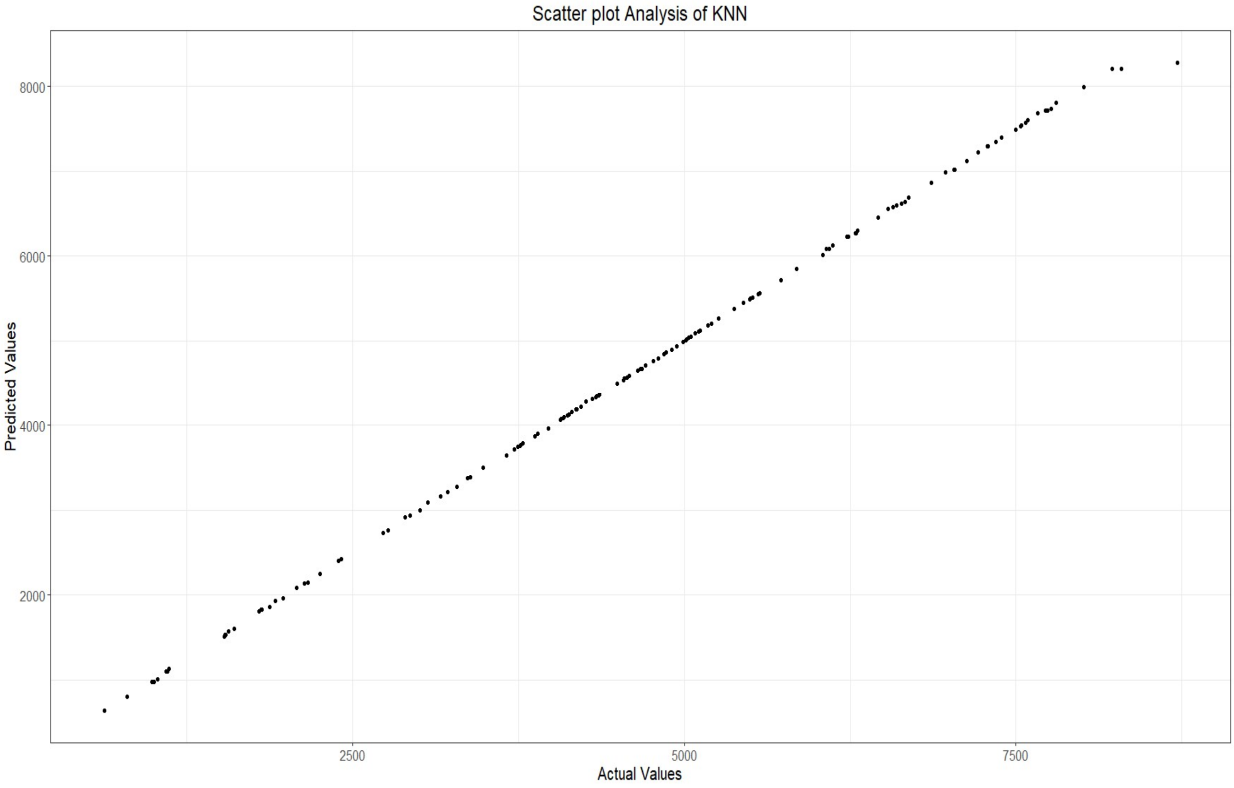
In the Model deployment, as the target variable is continuous variable, we need to go for regression model analysis. So, for regression model we have KNN Imputation method, Random forest regression, Linear regression model & Decision Tree regression. for predicting the target variable. Before deploying any model, we have to divide the dataset into train and test datasets. As the target variable is continuous, we need to divide the dataset by using simple random sampling technique. In this sampling technique I have divided the 80% data into train data and remaining 20% data into test data.

R Code: For splitting the data into train and test data.

```
train_index = sample(1:nrow(bike_rental_data), 0.8 * nrow(bike_rental_data))
train = bike_rental_data[train_index,]
test = bike_rental_data[-train_index,]
```

3.1 KNN Imputation:

In the KNN imputation method, I have used the knn.reg function as our target variable is continuous. The internal functionality of the KNN imputation is that it will not store any rules or pattern to predict the target variable it will just takes the test dataset values and calculates the distance between the test data values and train data values and selects the observations whose distance is lesser with the test data and imputes the target variable values by mean method. In this function we need to pass the value to the K which is the no. of neighbours to select. We can use either Euclidean method or Manhattan method for the distance calculation in this method. The KNN method gave the most accurate values when compared with the other models in this project, below I have mentioned the scatter plot between the actual values and predicted values, As we can see they both are linearly correlated with each other.



3.2 Random Forest Regression

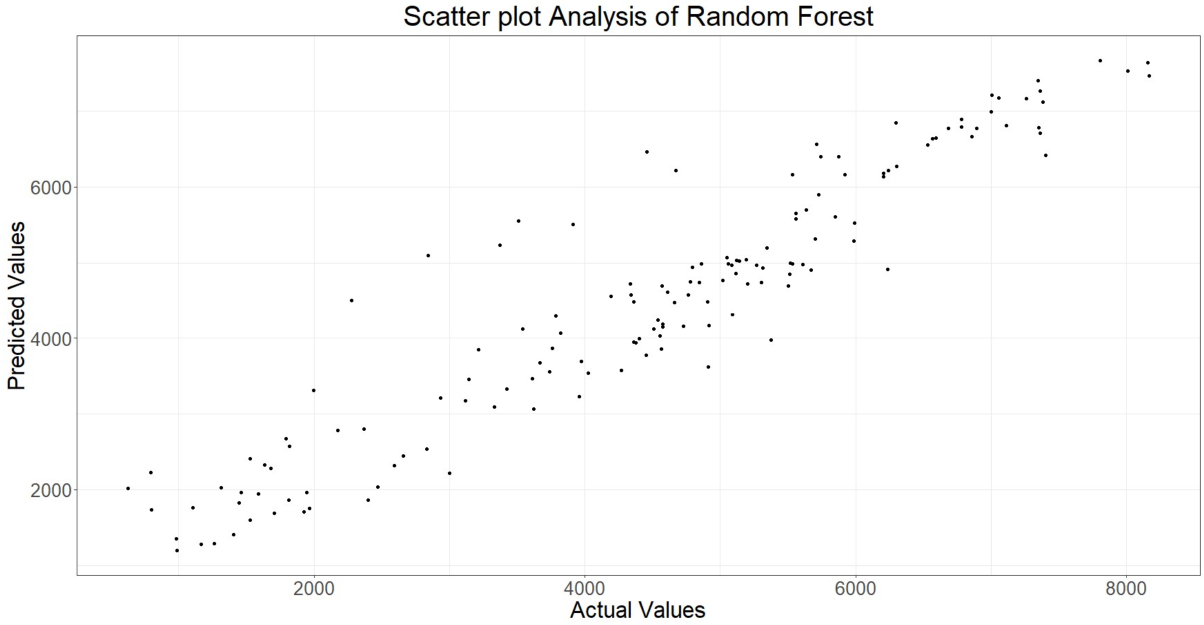
In the Random Forest Regression, I have used ‘randomForest’ function to predict the target variable in which we need to enter the number of trees, based on the number of trees we pass, the MAPE or accuracy changes, I have checked by passing 500 trees and 700 trees but there isn’t much of difference in the accuracy, it’s showing mostly similar percentages. So, I have passed 500 trees as the input for ntree in this project. After passing the train dataset into the randomForest function, store the model pattern obtained from it into other variable and use that model pattern to predict the test dataset target variable by using the predict function. Below I have mentioned scatter plot which I have plotted between the test dataset target values and predicted values.

```
rf_model = randomForest(cnt~., data = train, ntree = 500)

rf_predictions = predict(rf_model, test[,-10])

summary(rf_predictions)
```

```
  Min.    1st Qu.   Median   Mean   3rd Qu.   Max.
  1194     3271      4617     4482    5673     7669
```



3.3 Linear Regression

In the Linear regression model, the internal functionality of this method is it will calculate the coefficients of each independent variables from the past or historical data and uses those coefficients on the test data set to predict the target variable. Linear regression model also used for describing the relationship between any two variables. But before we apply the dataset to this model we need to check if the data is normally distributed. As our dataset is normally distributed after the removal of outliers, we can pass the dataset to this model. We can evaluate this model by using the parameters like R-squared, and Adjusted R-squared which should be greater than 80% and we can determine the variable importance from this model. Below I have mentioned the results obtained from the Linear Regression model.

```
summary(lr_model)

Call:
lm(formula = cnt ~ ., data = train)

Residuals:
    Min       1Q      Median      3Q       Max
  -4155.3   -445.4     52.2      535.7    3021.0



Coefficients:
             Estimate    Std.Error   t value      Pr(>|t|)
(Intercept)   1458.70     265.35     5.497       5.86e-08 ***
season         520.22      61.83    8.414         3.29e-16 ***
yr           1990.67       74.96    26.555       < 2e-16 ***
mnth           -38.54      19.13   -2.015        0.044391 *
holiday       -720.74     226.65   -3.180        0.001554 **
weekday         76.88      18.64    4.125        4.27e-05 ***
workingday      84.08      82.15    1.024        0.306500
weathersit    -605.17      94.84   -6.381        3.68e-10 ***
temp          5140.84     224.10   22.940        < 2e-16 ***
hum           -922.62     378.11   -2.440        0.014993 *
windspeed   -2130.21      568.38   -3.748        0.000197 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 880.7 on 562 degrees of freedom
Multiple R-squared:  0.7938,      Adjusted R-squared:  0.7901
F-statistic: 216.3 on 10 and 562 DF,   p-value: < 2.2e-16
```

Chapter 4

Conclusion

4.1 Model Evaluation:

I have evaluated the regression model using the error metrics which are MAPE, MSE, RMSE values, and also based on the accuracy of the models and the plots between the predicted values and actual values.

MAPE (Mean Absolute Percentage Error):

It is one of the error metrics method to predict the performance of the regression models. It is calculated based on the formula mean of actual value minus predicted value to the actual value.

$$\mathrm{M} = \frac{1}{n} \sum \left| \frac{At - Pt}{At} \right|$$

At = Actual Value

Pt = Predicted Value

Below I Have mentioned the error metrics of each model by calculating it.

1. KNN Imputation Model

MAPE: 0.45%

Accuracy: 99.55%

2. Random Forest Regression

MAPE: 16.71%

Accuracy: 83.29%

3. Linear Regression Model

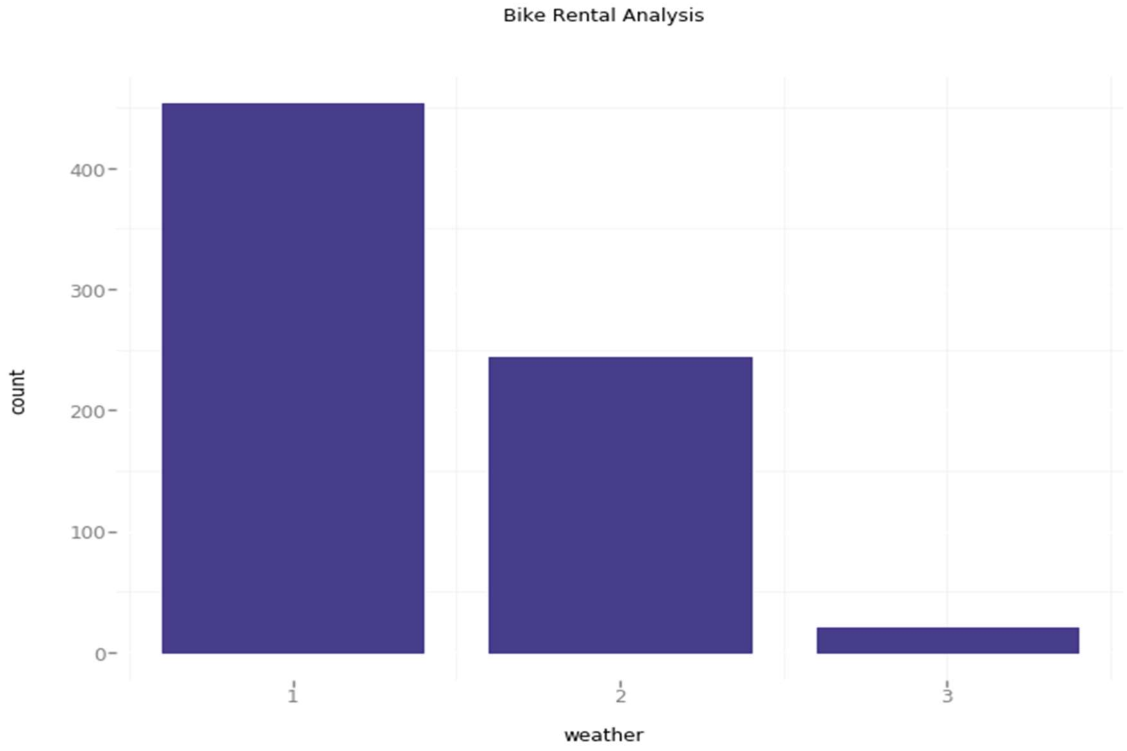
MAPE: 20.2 %

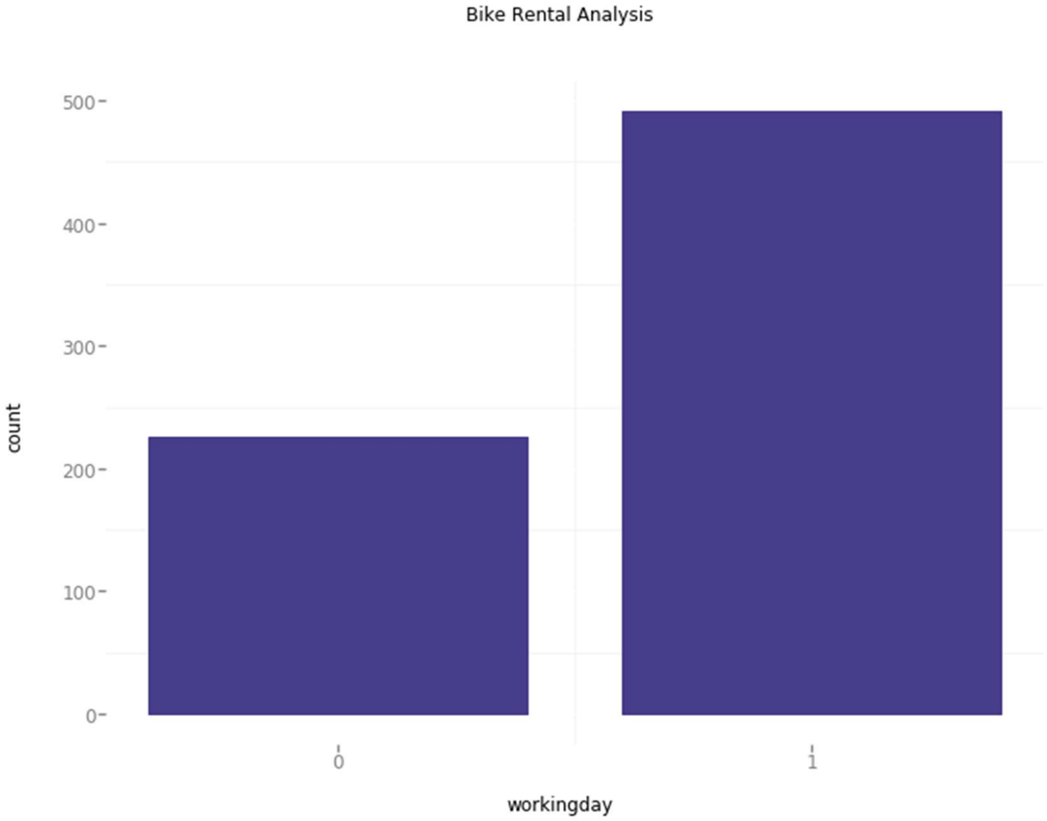
Accuracy: 79.8%

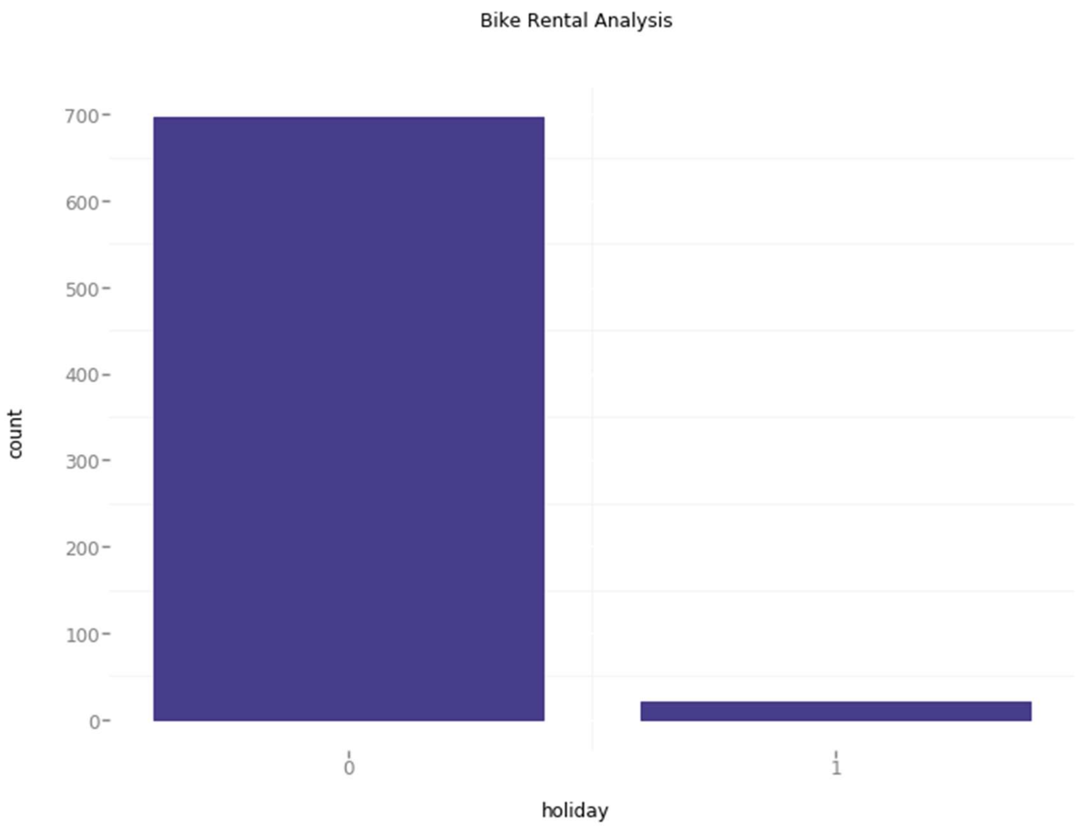
4.2 Model Selection

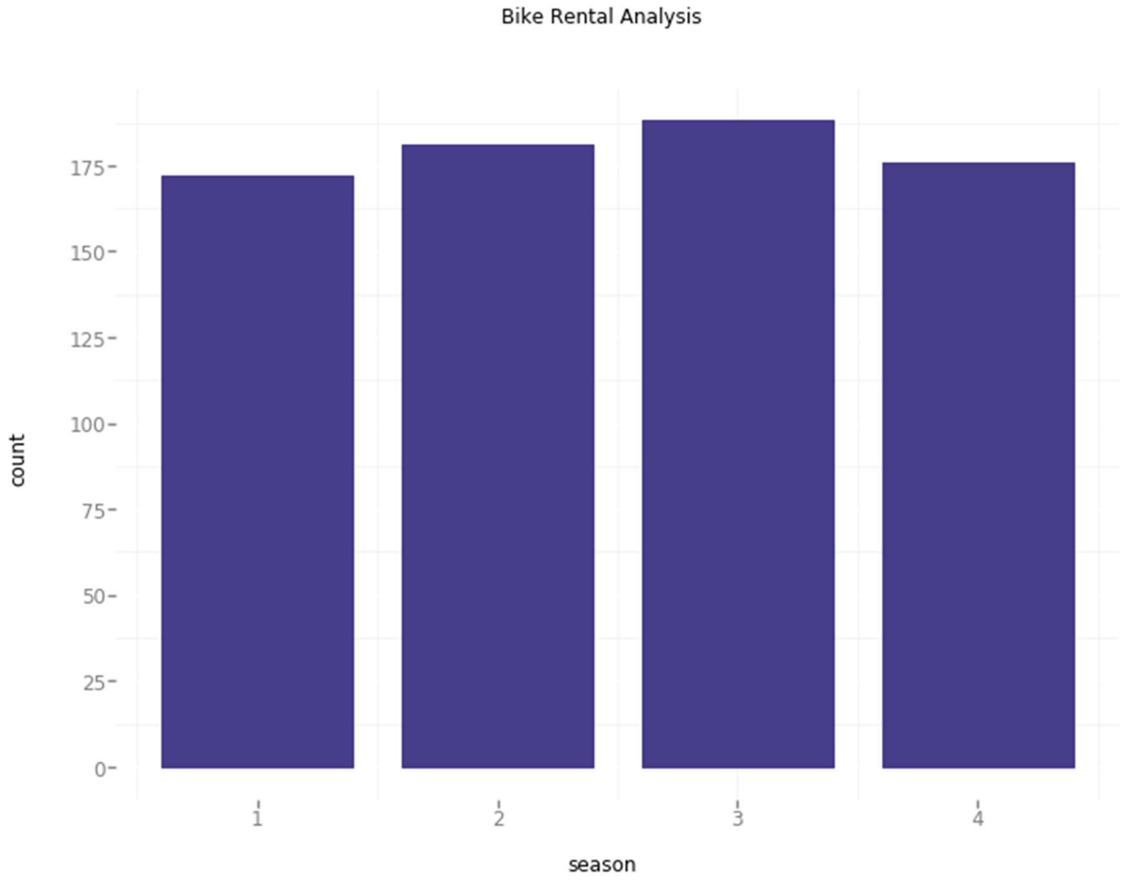
Based on the MAPE percentage and accuracy values we can choose the KNN Imputation Model for the prediction of bike rentals on daily basis and environmental situations.

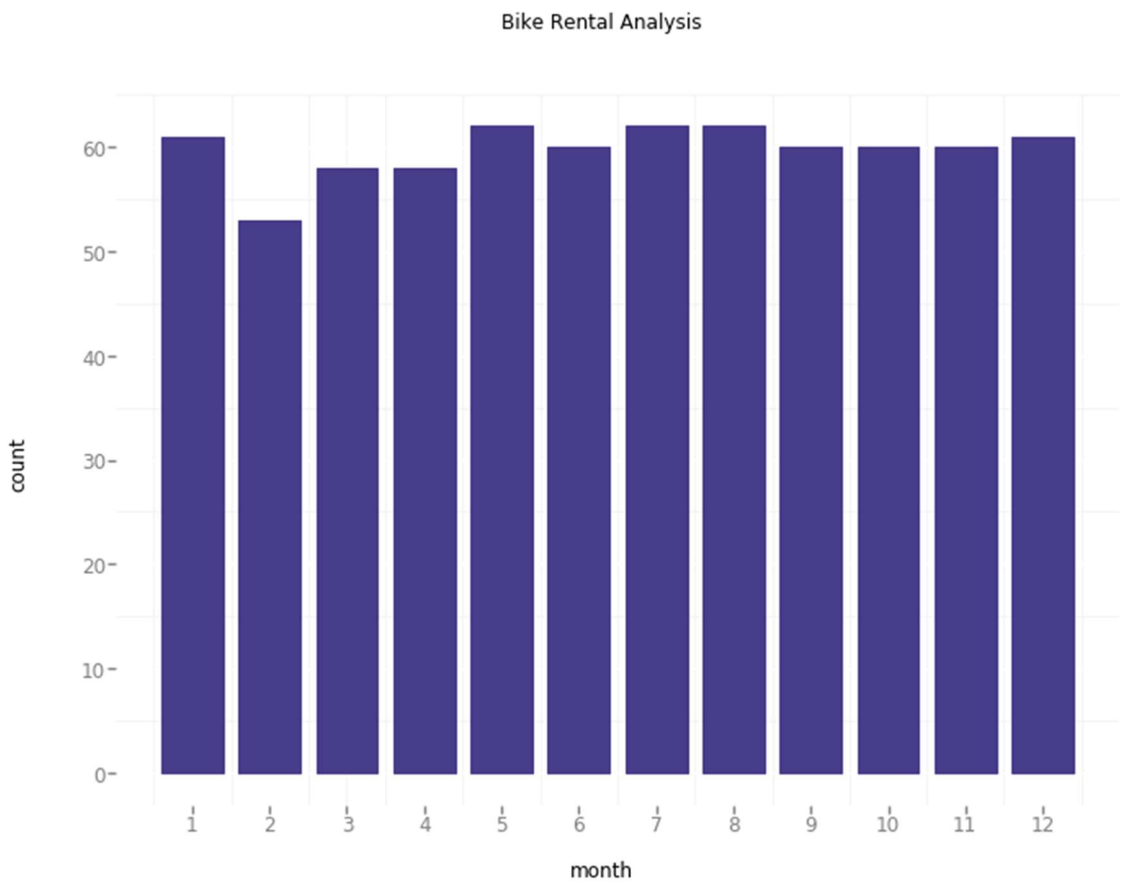
Appendix A – Extra Figures

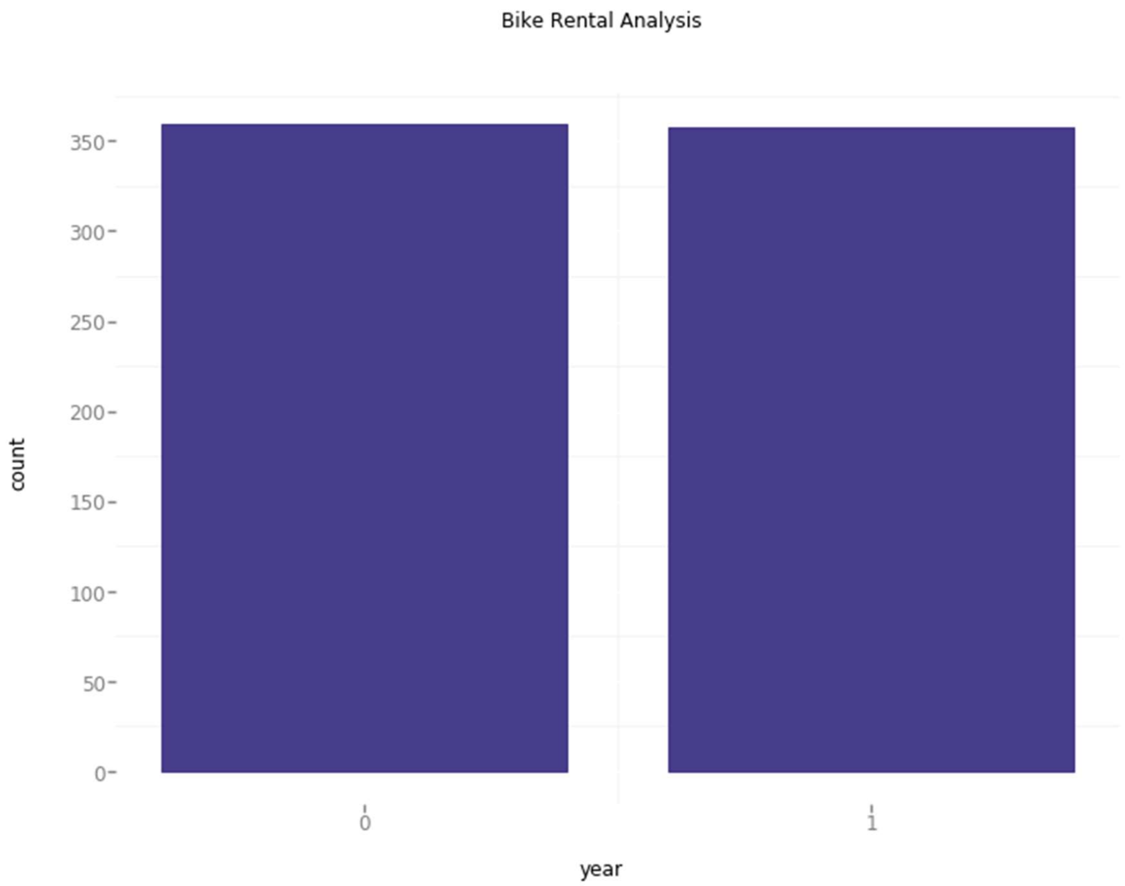





Bike Rental Analysis
700
600
500
400
300
200
100
0
count
0
1
holiday


Bike Rental Analysis
175
150
125
100
75
50
25
0
count
1
2
3
4
season


Bike Rental Analysis
count
0
10
20
30
40
50
60
1
2
3
4
5
6
7
8
9
10
11
12
month


Bike Rental Analysis
count
0
50
100
150
200
250
300
350
0
1
year

Appendix B – R CODE

```
rm(list=ls())

#Setting directory
setwd("C:/Users/npava/Desktop/Bike Rental Project/R")

#Loading Libraries
libraries = c("plyr","dplyr",
"ggplot2","gplots","rpart","dplyr","DMwR","randomForest","usdm","corrgram",
"DataCombine","FNN",)
lapply(libraries, require, character.only = TRUE)
rm(libraries)

#Reading the csv file
bike_rental_data = read.csv("day.csv", header = T, sep = ',', na.strings =
c(" ", "", "NA"))

###########Data exploration#########

head(bike_rental_data)

names(bike_rental_data)

str(bike_rental_data)


#################### MISSING VALUES ####################
missing_values = sapply(bike_rental_data, function(x){sum(is.na(x))})

missing_values
#There are no missing values in the data set which can be confirmed after
running the above line of codes.

################### Data Distribution ##################

#As the target variable is continuous variable we are going to use
regression models for predictions
#So before we pass the data we need to check the Normal distribution of
Data

#Distribution of Temp Variable
ggplot(bike_rental_data, aes_string(x = bike_rental_data$temp)) +
   geom_histogram(fill="DarkSlateBlue", colour = "black", binwidth = 0.1) +
geom_density() +
   theme_bw() + xlab("Temp") + ylab("Range") + ggtitle("Normal Distribution
of Temperature Variable") +
   theme(text=element_text(size=10))

#Distribution of Atemp Variable
ggplot(bike_rental_data, aes_string(x = bike_rental_data$atemp)) +
   geom_histogram(fill="DarkSlateBlue", colour = "black", binwidth = 0.1) +
geom_density() +
   theme_bw() + xlab("Atemp") + ylab("Range") + ggtitle("Normal
Distribution of Atemp Variable") +
   theme(text=element_text(size=10))

#Distribution of Humidity Variable
ggplot(bike_rental_data, aes_string(x = bike_rental_data$hum)) +
   geom_histogram(fill="DarkSlateBlue", colour = "black", binwidth = 0.1) +
geom_density() +
   theme_bw() + xlab("Humidity") + ylab("Range") + ggtitle("Normal
Distribution of Humidity Variable") +
   theme(text=element_text(size=10))

#Distribution of Windspeed Variable
ggplot(bike_rental_data, aes_string(x = bike_rental_data$windspeed)) +
   geom_histogram(fill="DarkSlateBlue", colour = "black", binwidth = 0.1) +
geom_density() +
   theme_bw() + xlab("Windspeed") + ylab("Range") + ggtitle("Normal
Distribution of Windspeed Variable") +
   theme(text=element_text(size=10))

###################### Outlier Analysis ######################

#I have loaded the continuous or numeric varibales column names into
numeric_var for detecting the outliers
numeric_var =
colnames(bike_rental_data[,c("temp","atemp","hum","windspeed")])

for (i in 1:length(numeric_var))
{
  assign(paste0("gn",i), ggplot(aes_string(y = numeric_var[i]), data =
bike_rental_data)+
           stat_boxplot(geom = "errorbar", width = 0.5) +
           geom_boxplot(outlier.colour="red", fill = "grey"
,outlier.shape=18,
                        outlier.size=1, notch=FALSE) +
           theme(legend.position="bottom")+
           labs(y=numeric_var[i])+
           ggtitle(paste("Box plot of",numeric_var[i])))
}

gridExtra::grid.arrange(gn1,gn3,gn2,gn4,ncol=2)


for (i in numeric_var) {
   print(i)
   val = bike_rental_data[,i][bike_rental_data[,i] %in%
boxplot.stats(bike_rental_data[,i])$out]
   print(length(val))
   bike_rental_data = bike_rental_data[which(!bike_rental_data[,i] %in%
val),]
}

####################### feature selection ################################

#Check for multicollinearity using VIF
dataframe_numeric = bike_rental_data[,c("temp","atemp","hum","windspeed")]
vifcor(dataframe_numeric)
#The atemp variable is highly correlated with the temp which is also
plotted using the correlation plot below.

#Check for collinearity using corelation graph
corrgram(bike_rental_data, order = F, upper.panel=panel.pie,
text.panel=panel.txt, main = "Correlation Plot")

#Removing the correlated variables from the dataset
```

```
bike_rental_data <- subset(bike_rental_data, select = -
c(instant,dteday,atemp,casual,registered))

rmExcept(keepers = "bike_rental_data")

head(bike_rental_data)

######################### Model Deployment ###############################

#Before passing the dataset into model we need to divide the data into
train and test datasets
#Dividing the dataset into train and test datsets
set.seed(1234)
train_index = sample(1:nrow(bike_rental_data), 0.8 *
nrow(bike_rental_data)) # dividing 80% of data into train dataset
train = bike_rental_data[train_index,]
test = bike_rental_data[-train_index,]

# KNN IMPUTATION

#MAPE = 0.45%
#MAE = 9.20
#RMSE = 26.59
#ACCURACY = 99.55%

KNN_Predictions = knn.reg(train[, 1:11], test[, 1:11], train$cnt, k = 3)


#For calculating MAPE percentage
MAPE = function(Actual_val, Predicted_val){
   print(mean(abs((Actual_val - Predicted_val)/Actual_val)) * 100)
}

MAPE(test[,11], KNN_Predictions$pred)

regr.eval(trues = test[,11], preds = KNN_Predictions$pred, stats =
c("mae","mse","rmse","mape"))

#Scatterplot between the Actual values and predicted values
ggplot(test, aes_string(x = test[,11], y = KNN_Predictions$pred)) +
   geom_point() +
   theme_bw()+ ylab("Predicted Values") + xlab("Actual Values") +
ggtitle("Scatter plot Analysis of KNN") +
   theme(text=element_text(size=15)) + theme(plot.title =
element_text(hjust = 0.5))


##Random Forest Model

#MAPE: 16.71%
#MAE: 502
#RMSE: 675
#Accuracy: 83.29%

#Train the data using random forest
rf_model = randomForest(cnt~., data = train, ntree = 500)

#Predict the test cases
rf_predictions = predict(rf_model, test[,-11])

#Calculate MAPE
```

```
regr.eval(trues = test[,11], preds = rf_predictions, stats =
c("mae","mse","rmse","mape"))

MAPE(test[,11], rf_predictions)

#Scatterplot between the Actual values and predicted values from Random
Forest Model
ggplot(test, aes_string(x = test[,11], y = rf_predictions)) +
   geom_point() +
   theme_bw()+ ylab("Predicted Values") + xlab("Actual Values") +
ggtitle("Scatter plot Analysis of Random Forest") +
   theme(text=element_text(size=15)) + theme(plot.title =
element_text(hjust = 0.5))


#LINEAR REGRESSION
#MAPE: 20.2%
#RMSE: 883
#MAE: 691
#Accuracy: 79.8%
#Adjusted R squared: 0.7887
#F-statistic: 214.6

#Train the data using linear regression
lr_model = lm(formula = cnt~., data = train)

#Check the summary of the model
summary(lr_model)

#Predict the test cases
lr_predictions = predict(lr_model, test[,-11])

#Create dataframe for actual and predicted values
df = data.frame("actual"=test[,11], "pred"=lr_predictions)
df = cbind(df,lr_predictions)
head(df)

#Calculate MAPE
regr.eval(trues = test[,11], preds = lr_predictions, stats =
c("mae","mse","rmse","mape"))
MAPE(test[,11], lr_predictions)

#Scatterplot between the Actual values and predicted values from Linear
Regression Model
ggplot(test, aes_string(x = test[,11], y = lr_predictions)) +
   geom_point() +
   theme_bw()+ ylab("Predicted Values") + xlab("Actual Values") +
ggtitle("Scatter plot Analysis of Linear Regression") +
   theme(text=element_text(size=15)) + theme(plot.title =
element_text(hjust = 0.5))


#Predict a sample data
predict(rf_model,test[2,])
test[2,11]


#################### Extra graphs  ###################

head(bike_rental_data)
ggplot(bike_rental_data, aes_string(x = bike_rental_data$weathersit)) +
   geom_bar(stat="count",fill =  "DarkSlateBlue") + theme_bw() +
   xlab("weather") + ylab('Count') +
   ggtitle("Bike Renatal Analysis") +  theme(text=element_text(size=15))


head(bike_rental_data)
ggplot(bike_rental_data, aes_string(x = bike_rental_data$workingday)) +
   geom_bar(stat="count",fill =  "DarkSlateBlue") + theme_bw() +
   xlab("workingday") + ylab('Count') +
   ggtitle("Bike Renatal Analysis") +  theme(text=element_text(size=15))


head(bike_rental_data)
ggplot(bike_rental_data, aes_string(x = bike_rental_data$holiday)) +
   geom_bar(stat="count",fill =  "DarkSlateBlue") + theme_bw() +
   xlab("holiday") + ylab('Count') +
   ggtitle("Bike Renatal Analysis") +  theme(text=element_text(size=15))


head(bike_rental_data)
ggplot(bike_rental_data, aes_string(x = bike_rental_data$season)) +
   geom_bar(stat="count",fill =  "DarkSlateBlue") + theme_bw() +
   xlab("season") + ylab('Count') +
   ggtitle("Bike Renatal Analysis") +  theme(text=element_text(size=15))


head(bike_rental_data)
ggplot(bike_rental_data, aes_string(x = bike_rental_data$month)) +
   geom_bar(stat="count",fill =  "DarkSlateBlue") + theme_bw() +
   xlab("month") + ylab('Count') +
   ggtitle("Bike Renatal Analysis") +  theme(text=element_text(size=15))


head(bike_rental_data)
ggplot(bike_rental_data, aes_string(x = bike_rental_data$yr)) +
   geom_bar(stat="count",fill =  "DarkSlateBlue") + theme_bw() +
   xlab("year") + ylab('Count') +
   ggtitle("Bike Renatal Analysis") +  theme(text=element_text(size=15))
```