Prediction of Bike Rental Count

Aditya Kumar Ghosh

12 -7-2019

Contents

Appendix B

Chapter 1

Introduction

1.1 Problem Statement

The objective of this Case is to Predication of bike rental count on daily based on the environmental and seasonal settings.Aim is to predict the count of bike rented daily based on data set provided .We would like to predict the number of bikes that can be rented on particular data based on environmental and seasonal settings.

1.2 Data

Our task is to build a regression model to predict the count of bikes rented any particular day based on environmental and seasonal settings Given below is a sample of the data set that we are using to predict the count of bike rented :

Table 1.1 bike rented dataset (Columns 1-9)

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit
1	2011-01-01	1	0	1	0	6	0	2
2	2011-01-02	1	0	1	0	0	0	2
3	2011-01-03	1	0	1	0	1	1	1
4	2011-01-04	1	0	1	0	2	1	1
5	2011-01-05	1	0	1	0	3	1	1
6	2011-01-06	1	0	1	0	4	1	1
7	2011-01-07	1	0	1	0	5	1	2

Table 1.2 bike rented dataset (Columns 10-16)

temp	atemp	hum	windspeed	casual	registered	cnt
0.3441670	0.3636250	0.805833	0.1604460	331	654	985
0.3634780	0.3537390	0.696087	0.2485390	131	670	801
0.1963640	0.1894050	0.437273	0.2483090	120	1229	1349
0.2000000	0.2121220	0.590435	0.1602960	108	1454	1562
0.2269570	0.2292700	0.436957	0.1869000	82	1518	1600
0.2043480	0.2332090	0.518261	0.0895652	88	1518	1606
0.1965220	0.2088390	0.498696	0.1687260	148	1362	1510

As you can see in the table below we have the following 16 variables, using which we have to correctly predict the count of bike rented our target variable is *cnt*

Table 1.3 Predictor Variables

S.No	**Predictor**
1	instant
2	dteday
3	season
4	yr
5	mnth
6	holiday
7	weekday
8	workingday
9	weathersit
10	temp
11	atemp
12	hum
13	windspeed
14	casual
15	registered

Chapter 2

Methodology

2.1 Pre Processing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing.

Data goes through a series of steps during preprocessing:

- Data Cleaning: Data is cleansed through processes such as filling in missing values, smoothing noisy data, or resolving inconsistencies in the data.
- Data Integration: Data with different representations are put together and conflicts within the data are resolved.
- Data Transformation: Data is normalized, aggregated and generalized.
- Data Reduction: This step aims to present a reduced representation of the data in a data warehouse.
- Data Discretization: Involves the reduction of a number of values of a continuous attribute by dividing the range of attribute intervals.

2.1.1 Feature Engineering

2.1.1.1 Extracting Day From date

Date columns usually provide valuable information about the model target,here we try to create new feature day from date.

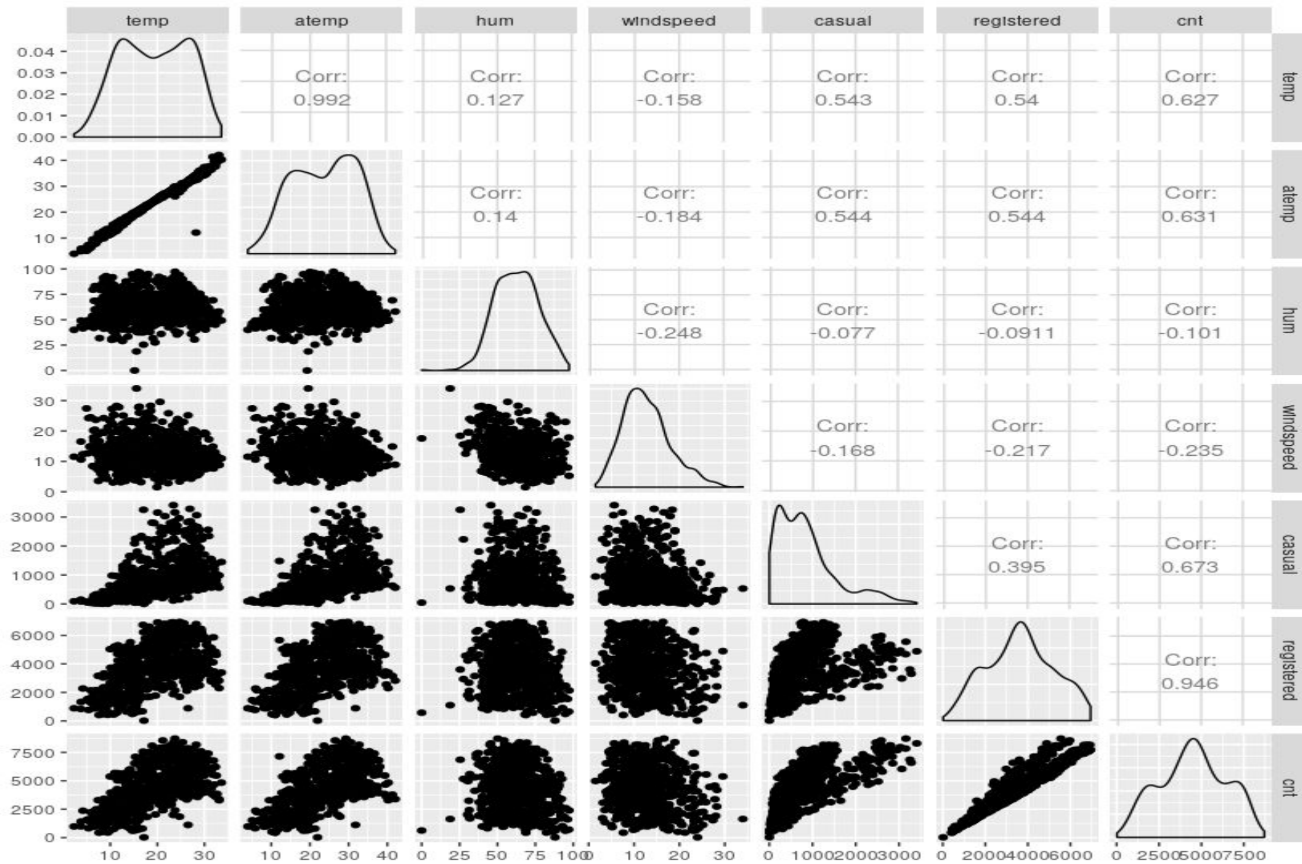
Table 1.4 Extracting day from date column

Year	Day
2011-01-01	1
2011-01-02	2

2.1.2 Distribution of continuous variables

Data visualization is an important part of any data analysis. It helps us to recognize relations between variables and also to find which variables are significant or which variables can affect the predicted variable.

Figure 1 : Pair plot and correlation value of continuous variables (R code)

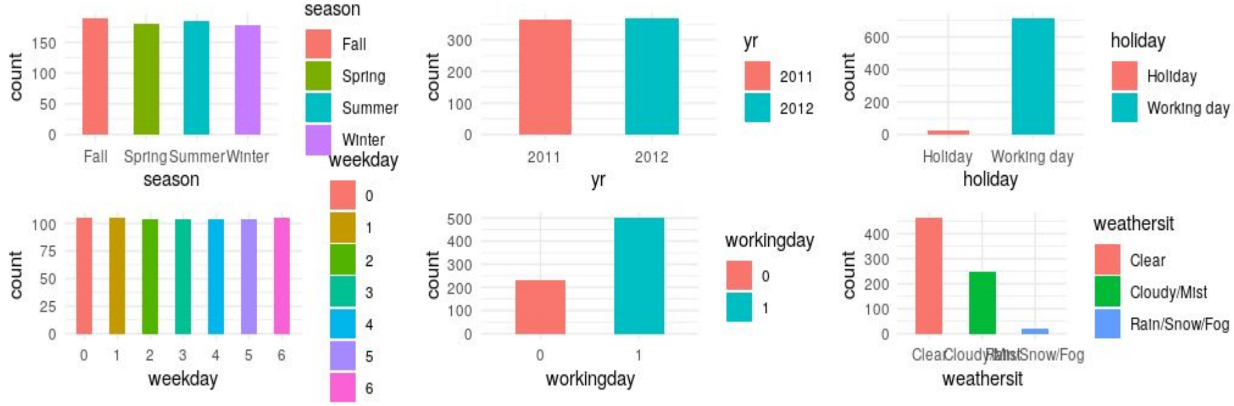


Observation

1. temp and atemp highly positively correlated
2. cnt and registered is highly positively correlated
3. cnt and casual is positively correlated
4. casual is right skewed
5. atemp is shows a moderate correlation toward cnt
6. temp is shows a moderate correlation toward cnt

2.1.3 Distribution of categorical variables

Figure 2 :Distribution of categorical variable (R code)



Observation

1. Number of holiday was less than working day
2. There are very few observations of Rain/snow/fog

2.1.4 Data summarisation with respect to target variable

Figure 3: Effect of categorical variable with respect to target variable(R code)

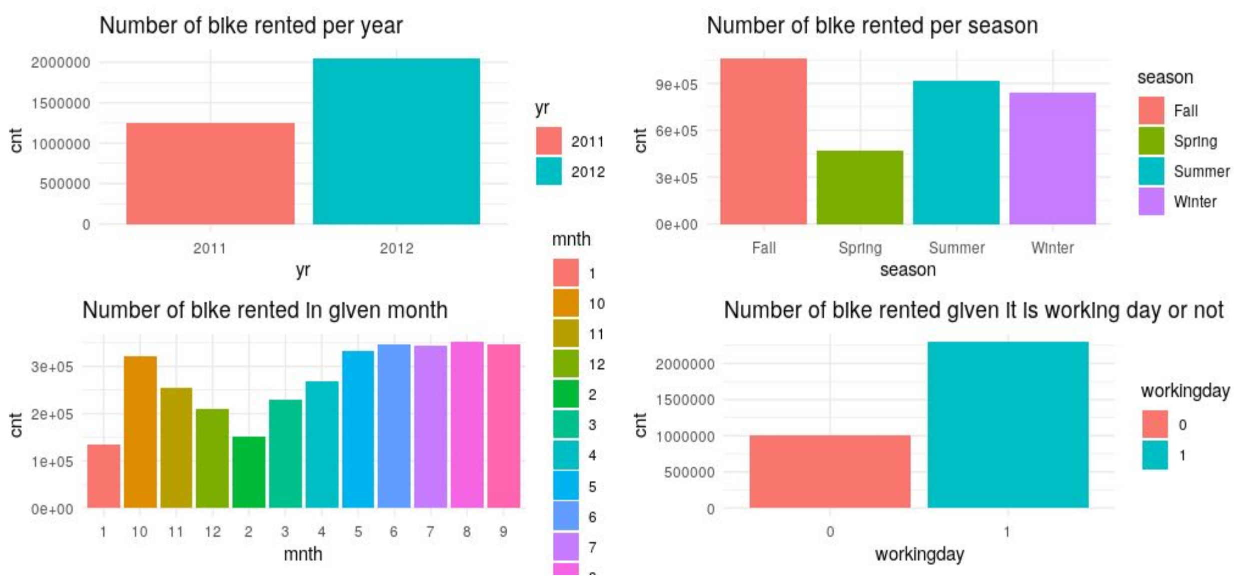
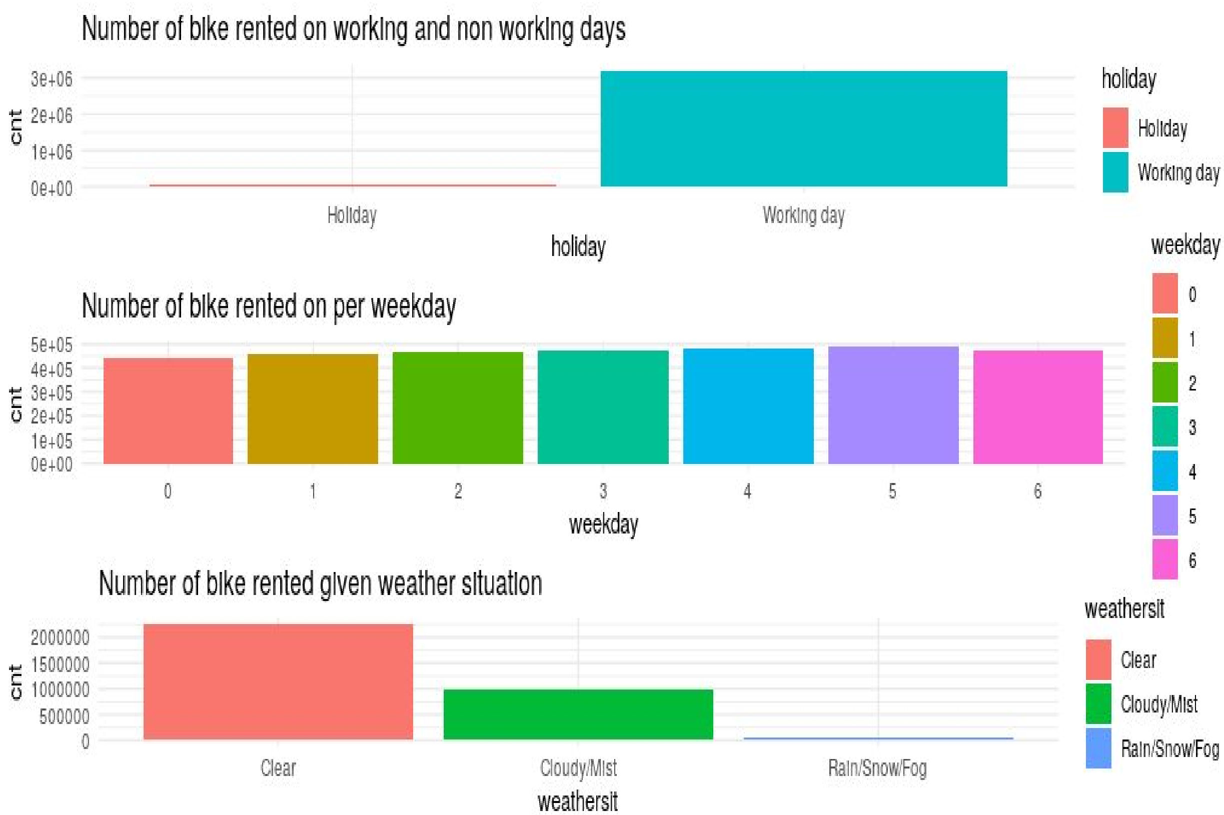


Figure 4: Effect of categorical variable with respect to target variable ([R code](#))



Observations

1. On year 2012 more user rented bike 2011
2. On Fall season more number of people rented bike
3. On month 8 or August most no of bike where rented amount =351194
4. On month 1 or january least number of bike where rented amount = 134933
5. On working day most number of bike where rented amount =3214244
6. On weekday 5 or friday most number of bike were rented amount = 487790
7. On weekday 0 or sunday least number of bike where rented amount =444027
8. On clear weather most number of bike was rented amount =2257952
9. On Rain/snow/fog least number of bike was rented amount =37869
10. On holiday very few number of bike where rented compared to working day

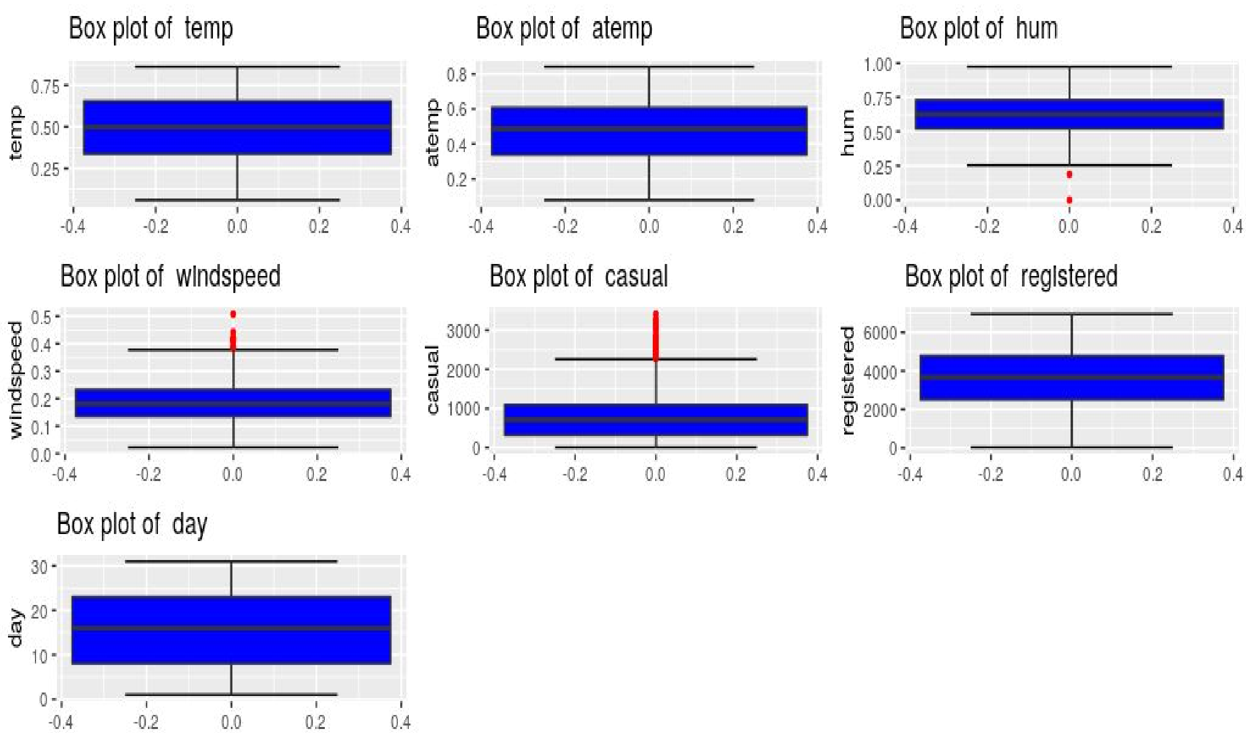
2.1.5 Outlier Analysis

In statistics, an outlier is an observation point that is distant from other observations.
We can clearly observe from these probability distributions that most of the variables are skewed , for example *casual ,hum,wind speed* .The skew in these distributions can be most likely explained by the presence of outliers and extreme values in the data

In descriptive statistics, a box plot is a method for graphically depicting groups of numerical data through their quartiles. Box plots may also have lines extending vertically from the boxes (whiskers) indicating variability outside the upper and lower quartiles, hence the terms box-and-whisker plot and box-and-whisker diagrams. Outliers may be plotted as individual points

Any observations outside of upper fence and lower fence is treated as an outlier , we can either remove the outlier or impute values using mean ,mode ,median or knn imputation ,removing outliers can make data set small as whole row is removed this won't have any effect if we have large dataset but if you have small dataset removing outliers can make already small dataset more small

Figure 5 : Outlier Analysis of continuous variables ([R code]())



2.1.6 Feature Selection

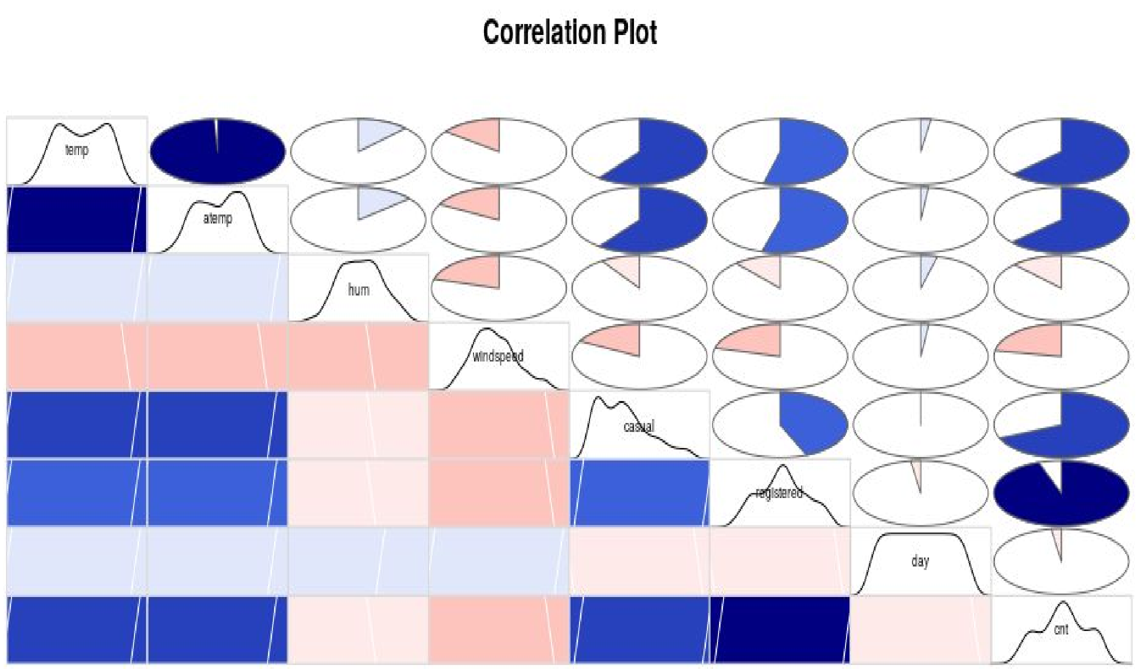
Feature Selection is one of the core concepts in machine learning which hugely impacts the performance of your model. The data features that you use to train your machine learning models have a huge influence on the performance you can achieve.

Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in.

Benefits are of feature selection first Reduces Overfitting Less redundant data means less opportunity to make decisions based on noise.Improves Accuracy Less misleading data means modeling accuracy improves.Reduces Training Time fewer data points reduce algorithm complexity and algorithms train faster.

Features are selected based on their scores in various statistical tests for their correlation with the outcome variable. Correlation plot is used to find out if there is any multicollinearity between variables. The highly collinear variables are dropped and then the model is executed.

Figure 6: Correlation plot between numeric variables ([R code]())



Chapter 3: Modelling

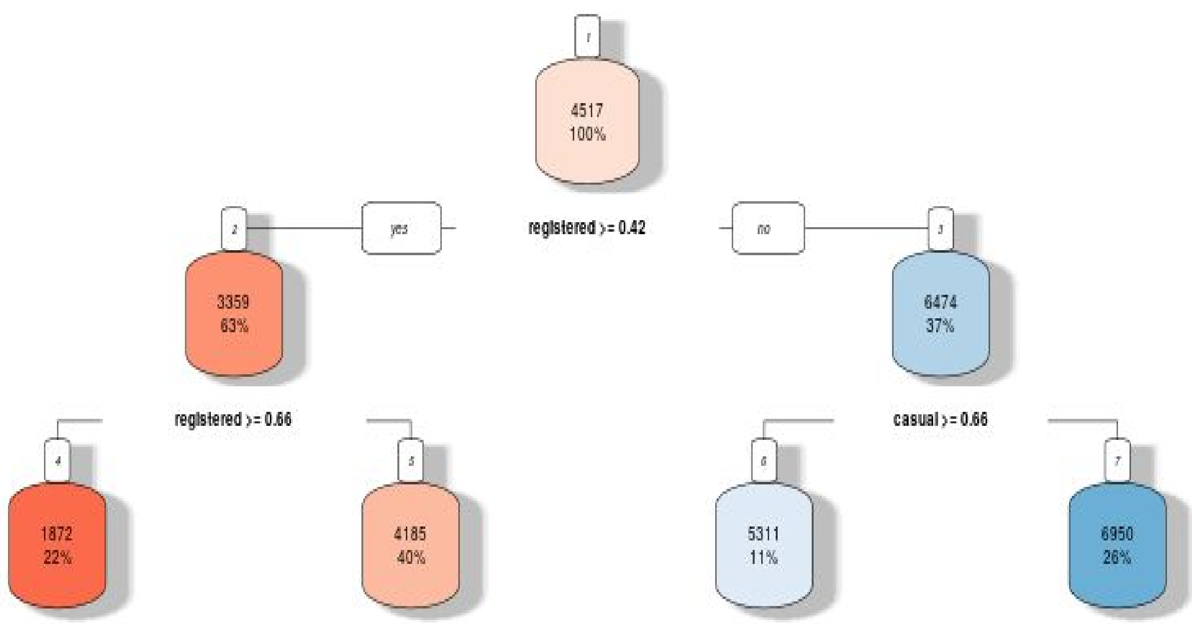
3.1 Model Selection

During analysis of dataset we have come to know that ***regression*** model will be most suited for modeling as our target variable is continuous , had our continuous variable been categorical we could go for classification Algorithm

3.2 Decision Tree

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

Figure 7: Decision tree for bike rental count



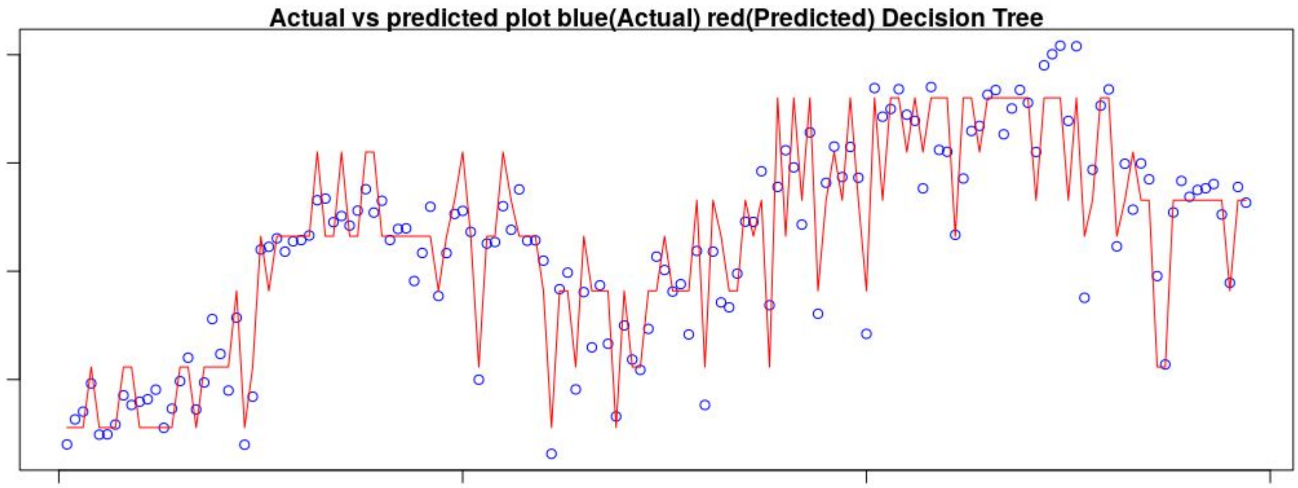


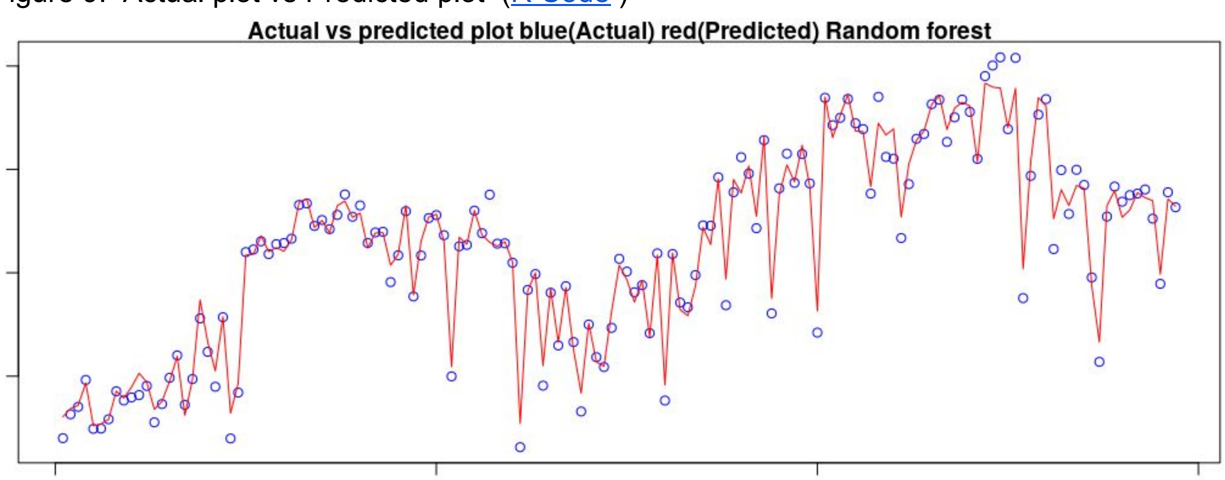
Figure 8: Actual plot vs Predicted plot(R Code)

Using decision tree we are able to predict the count of bike rented ,(Mean absolute percentage error) MAPE is 10.6% hence Accuracy is

3.3 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. Random decision forests correct for decision trees' habit of overfitting to their training set.

Figure 9: Actual plot vs Predicted plot (R Code)



Using Random Forest MAPE score is 4.7% and accuracy is 95.7%

3.4 Linear Regression

In statistics, linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression.

Table 1.5 Linear Regression model summary

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

Residuals:
    Min      1Q  Median      3Q     Max
-366.10 -115.94  -30.12   53.73 1683.73

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  9510.495    113.752  83.607  < 2e-16 ***
season        -17.286     18.679  -0.925  0.35514
yr             36.612     35.009   1.046  0.29610
mnth           -2.295      5.299  -0.433  0.66514
holiday       -10.719     62.474  -0.172  0.86383
weekday        16.442      5.208   3.157  0.00168 **
workingday   -207.639     40.353  -5.146 3.68e-07 ***
weathersit     30.018     26.517   1.132  0.25810
temp         -101.954     72.572  -1.405  0.16061
hum            40.251     73.128   0.550  0.58225
windspeed       1.991     55.729   0.036  0.97151
casual      -2334.992     83.257 -28.046  < 2e-16 ***
registered  -7095.168    114.014 -62.231  < 2e-16 ***
day            19.038     35.313   0.539  0.59002
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 246.5 on 570 degrees of freedom
Multiple R-squared:  0.9844,   Adjusted R-squared:  0.984
F-statistic:  2766 on 13 and 570 DF,  p-value: < 2.2e-16
```

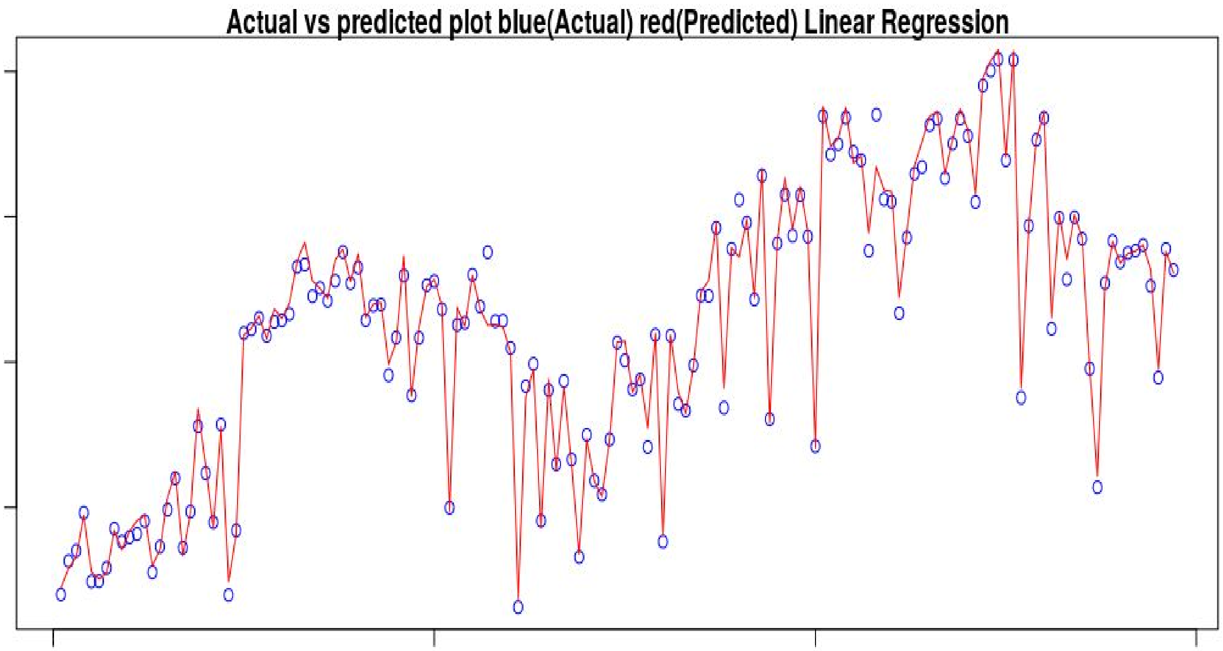


Figure 10: Actual plot vs Predicted plot ([R Code](#))

Using linear regression we are able to get accuracy of 97% and MAPE score of 3.07%
And Multiple R-squared: 98.4% , which means our independent variable are able to explain 98% variance in dependent variable which is quite good .

Chapter 4

Conclusion

4.1 Model Evaluation

Now that we have a few models for predicting the target variable, we need to decide which one to choose. There are several criteria that exist for evaluating and comparing models. We can compare the models using any of the following criteria:

1. Predictive Performance
2. Interpretability
3. Computational Efficiency

In our case of bike rental Data, the latter two, *Interpretability* and *Computation Efficiency,* do not hold much significance. Therefore we will use *Predictive performance* as the criteria to compare and evaluate models. Predictive performance can be measured by comparing the Predictions of the models with real values of the target variables, and calculating some average error measure.

4.1.1 MAPE (Mean Absolute Percentage error)

Measure accuracy as a percentage of error

Mape = $1/n \sum_{i=1}^{n} (\mid actual - predicted \mid)/actual$

Decision Tree MAPE :10.6%
Random Forest MAPE: 4.7%
Linear regression MAPE: 3.07%

4.12 Model Selection

Based on the above error metrics,Linear regression is the better model for our analysis. Hence Linear regression is chosen as the model for prediction of bike rental count

Chapter 5 : Appendix A

Figure 1 : Pair plot and correlation value of continuous variables ([R code]())

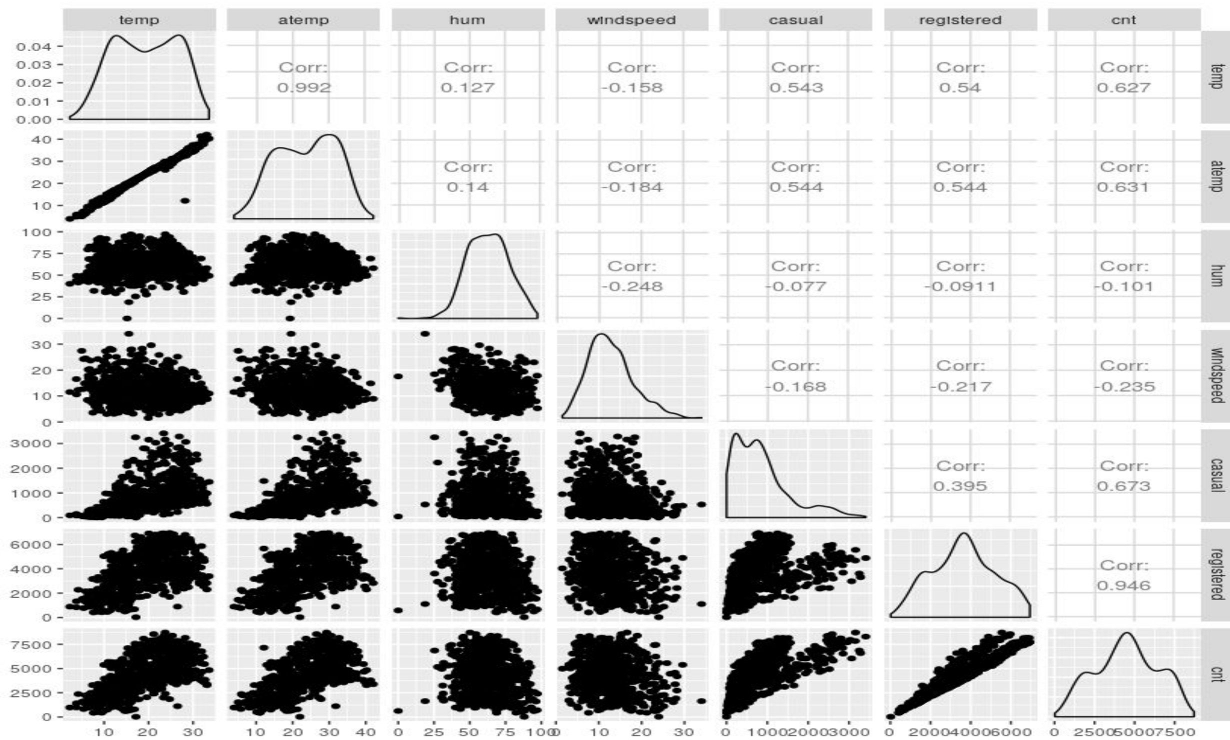


Figure 2 :Distribution of categorical variable ([R code]())

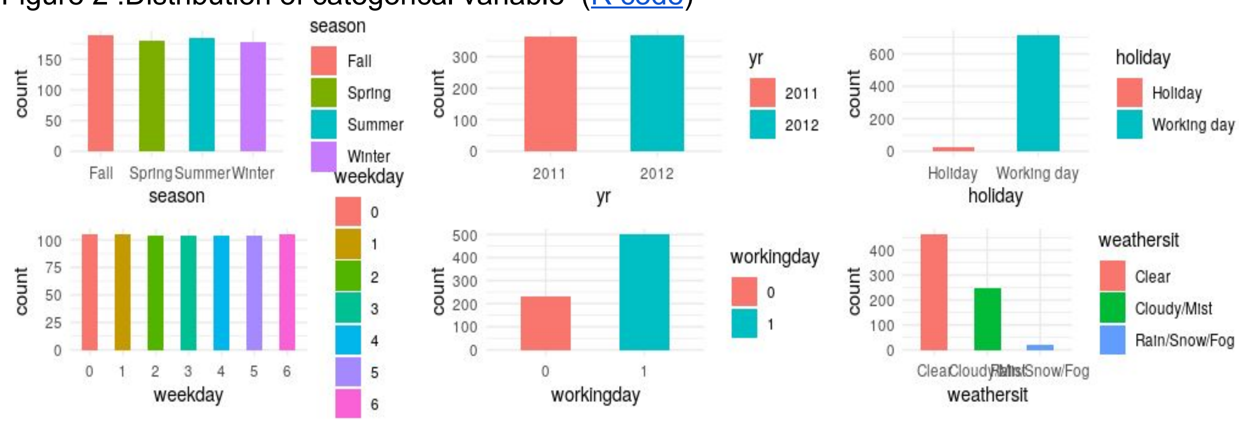


Figure 3: Effect of categorical variable with respect to target variable(R code)

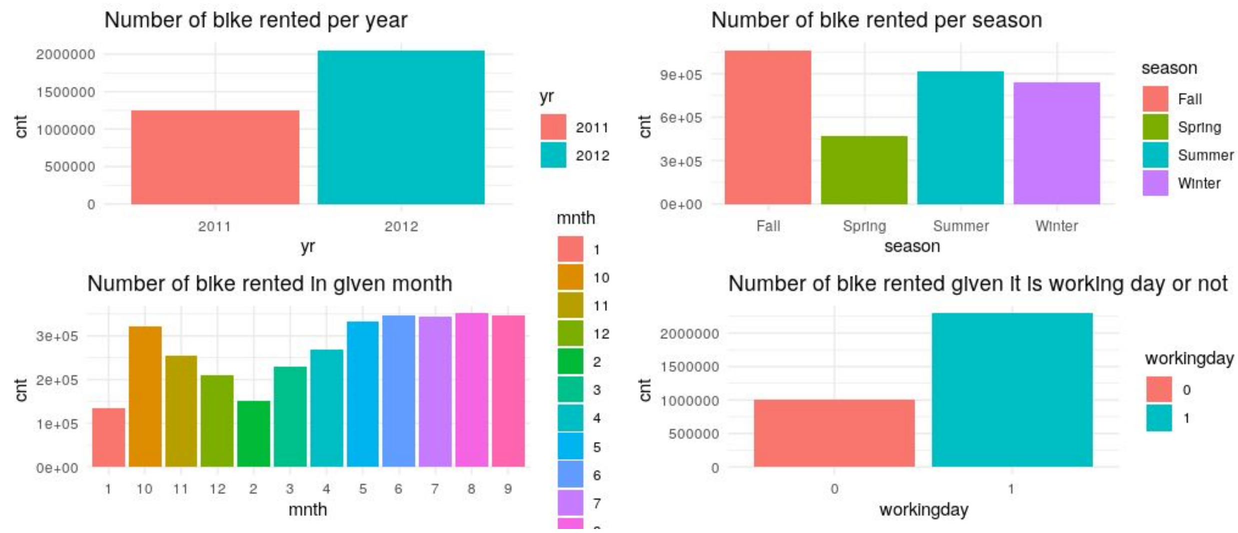


Figure 4: Effect of categorical variable with respect to target variable (R code)

Figure 5 : Outlier Analysis of continuous variables ([R code]())

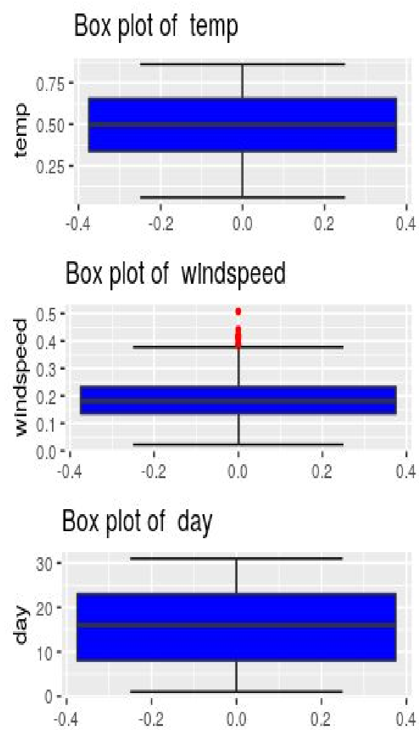


Figure 6: Correlation plot between numeric variables ([R code]())

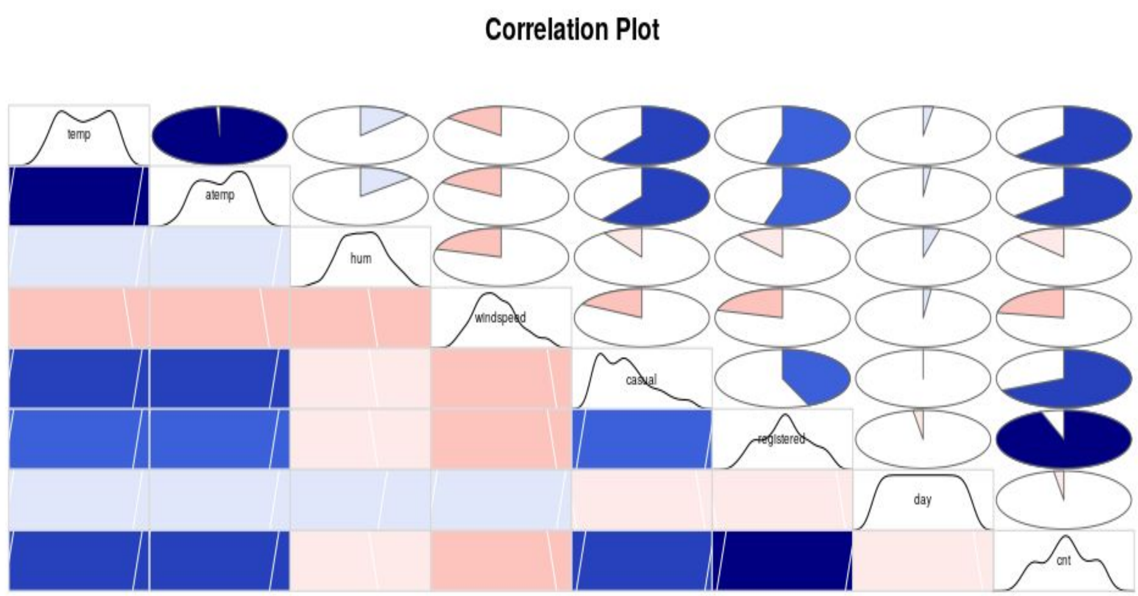
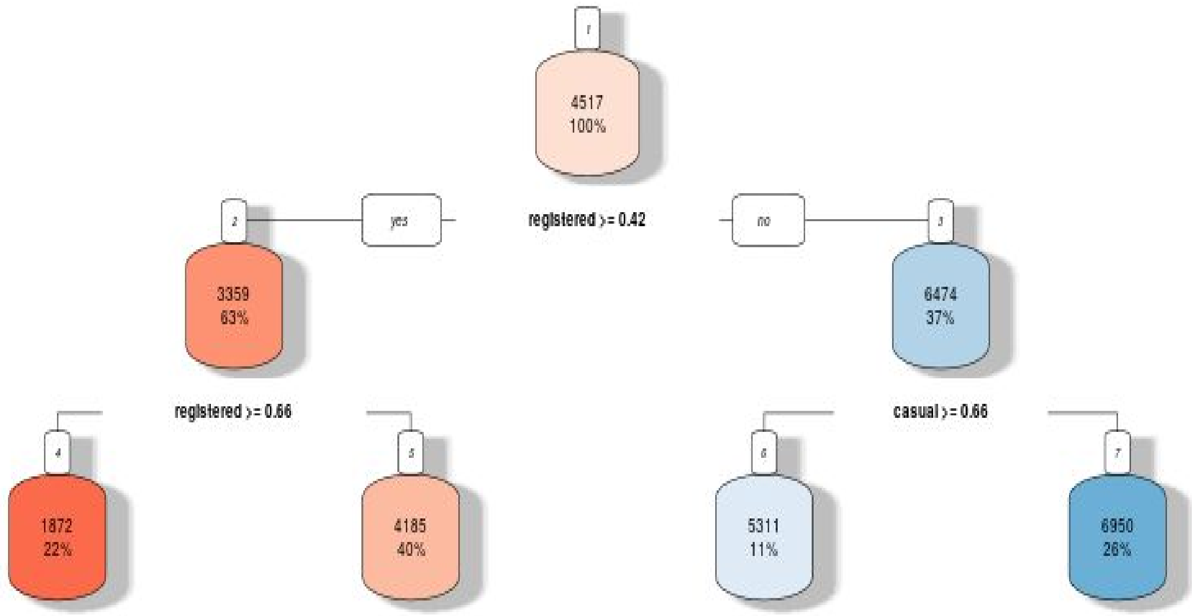


Figure 7: Decision tree for bike rental count



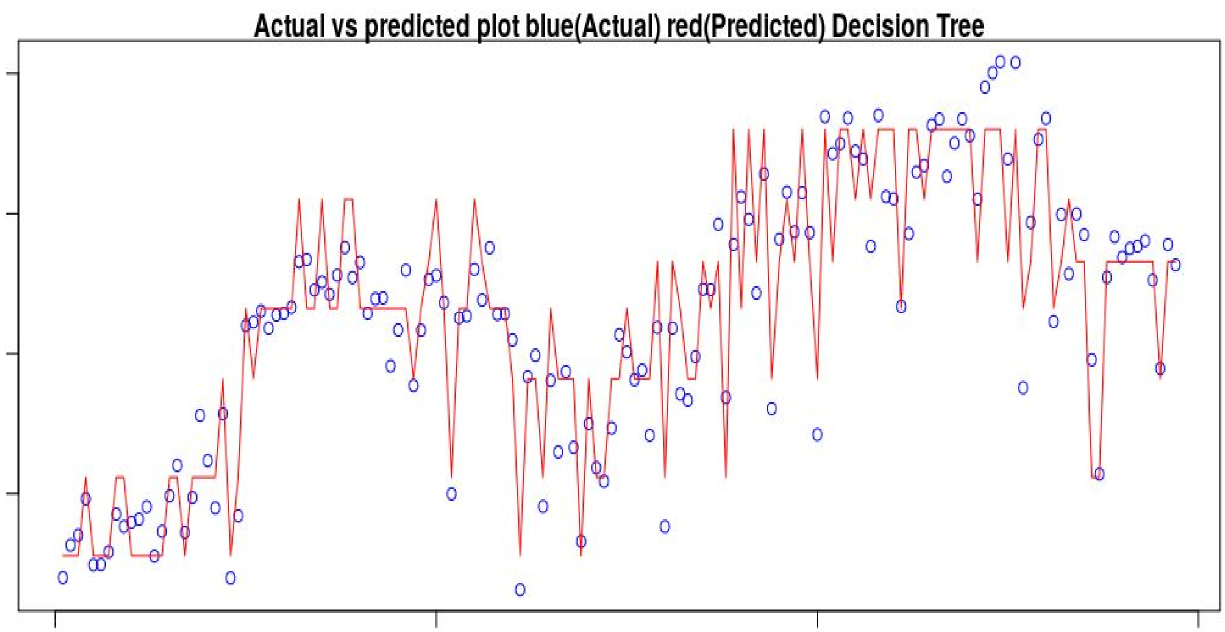


Figure 8: Actual plot vs Predicted plot([R Code]( ))

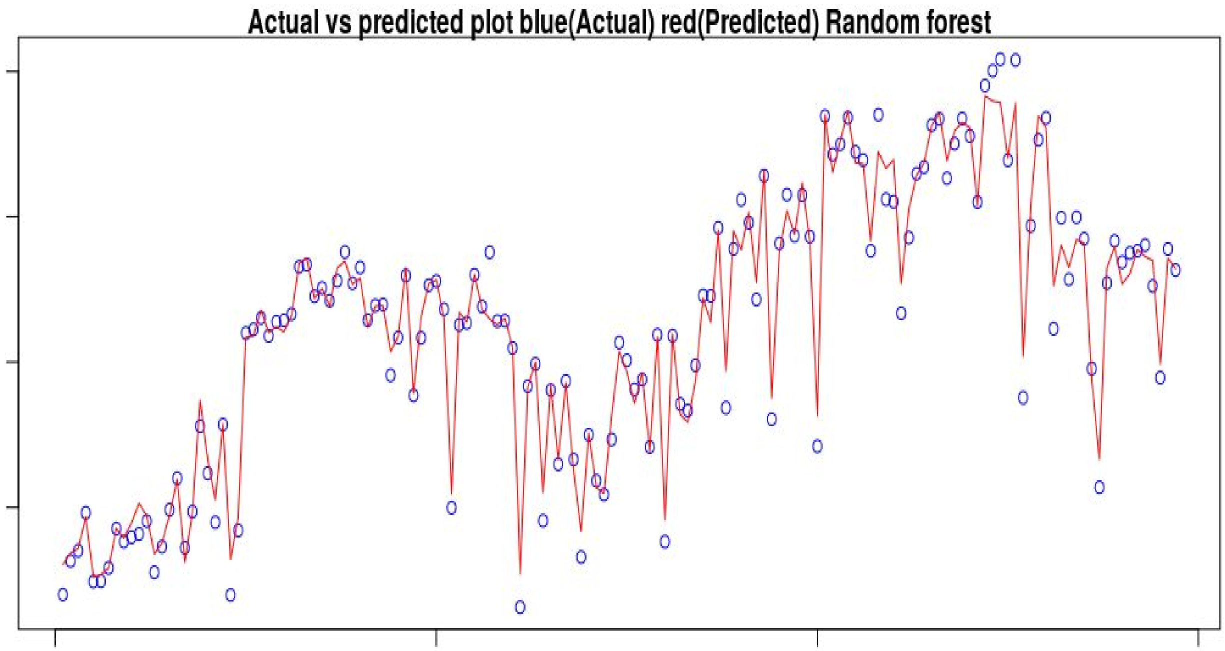


Figure 9: Actual plot vs Predicted plot ([R Code](#))

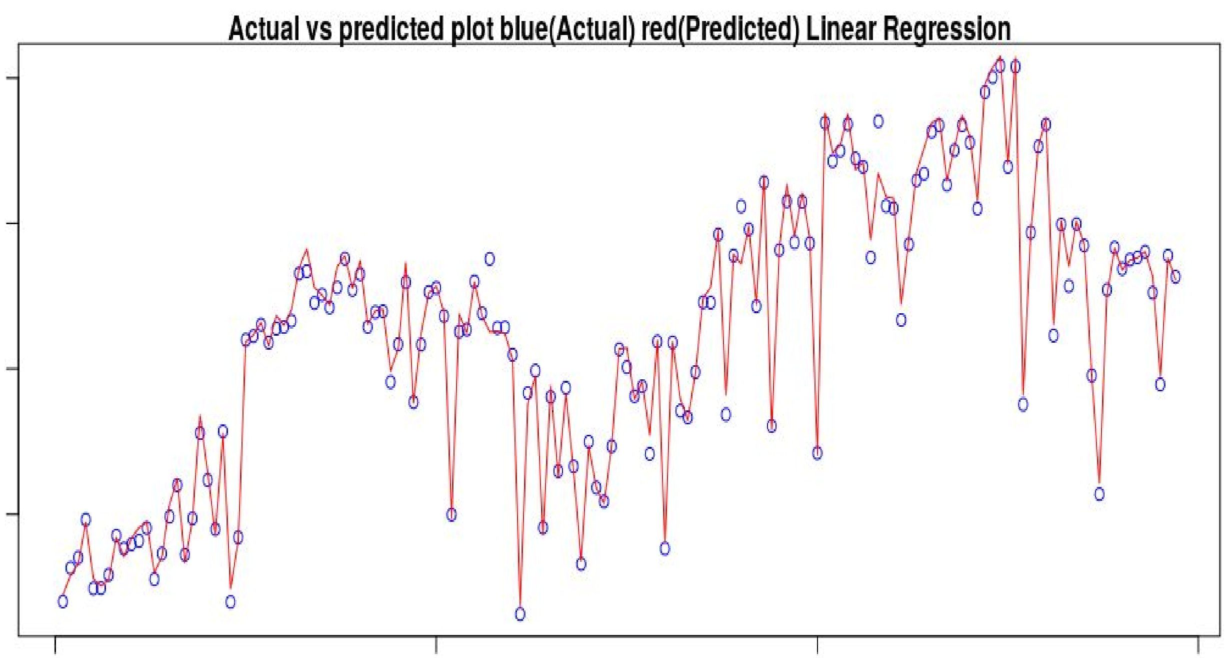


Figure 10: Actual plot vs Predicted plot ([R Code](#))

Appendix B - R Code

1. Pair plot and correlation value of continuous variables (Figure 1)

```
ggpairs(temp[continous])
```

2. Distribution of categorical variable (Figure 2) (Figure 3)

```
j=1
for( i in category){
assign(paste0("gn",j) ,ggplot(data =temp ,aes_string(x=i ,fill=i) )+
  geom_bar(stat ="count",width = 0.5 ) +
  theme_minimal()
)
j=j+1
}
gridExtra::grid.arrange(gn1,gn2,gn4,gn5,gn6,gn7,nrow=3 ,ncol=3)
gridExtra::grid.arrange(gn3 ,nrow=2 ,ncol=2)
```

3. Effect of categorical variable with respect to target variable (Figure 4)

```
cntByYear=temp %>%
  group_by(yr) %>%
  summarise(cnt=sum(cnt))

assign(paste0("tp",1),ggplot(data =cntByYear ,aes(x=yr ,y=cnt,fill=yr))+
  geom_col()+
  theme_minimal()+
  ggtitle("Number of bike rented per year ")
)

preSeasonCnt =temp %>%
  group_by(season) %>%
  summarise(cnt =sum(cnt))

assign(paste0("tp",2),ggplot(data =preSeasonCnt ,aes(x=season ,y=cnt,fill=season))+
  geom_col()+
```

```
  theme_minimal()+
  ggtitle("Number of bike rented per season ")
)

preMnthCnt =temp %>%
  group_by(mnth) %>%
  summarise(cnt =sum(cnt))

assign(paste0("tp",3),ggplot(data =preMnthCnt ,aes(x=mnth ,y=cnt,fill=mnth))+
  geom_col()+
  theme_minimal()+
  ggtitle("Number of bike rented in given month  ")
)

holidayCnt =temp %>%
  group_by(holiday) %>%
  summarise(cnt =sum(cnt))

assign(paste0("tp",4),ggplot(data =holidayCnt ,aes(x=holiday ,y=cnt,fill=holiday))+
  geom_col()+
  theme_minimal()+
  ggtitle("Number of bike rented on working and non working days  ")
)


weekdayCnt =temp %>%
  group_by(weekday) %>%
  summarise(cnt =sum(cnt))

assign(paste0("tp",5),ggplot(data =weekdayCnt  ,aes(x=weekday ,y=cnt,fill=weekday))+
  geom_col()+
  theme_minimal()+
  ggtitle("Number of bike rented on per weekday ")
)


weathersitCnt =temp %>%
  group_by(weathersit) %>%
  summarise(cnt =sum(cnt))

assign(paste0("tp",6),ggplot(data =weathersitCnt  ,aes(x=weathersit
,y=cnt,fill=weathersit))+
```

```
  geom_col()+
  theme_minimal()+
  ggtitle("Number of bike rented given weather situation  ")
)


workingdayCnt =temp %>%
  group_by(workingday) %>%
  summarise(cnt =sum(cnt))

assign(paste0("tp",7),ggplot(data =workingdayCnt  ,aes(x=workingday
,y=cnt,fill=workingday))+
  geom_col()+
  theme_minimal()+
  ggtitle("Number of bike rented given it is working day or not  ")
)

gridExtra::grid.arrange(tp1,tp2 ,tp3,tp7 ,ncol=2,nrow=2)
gridExtra::grid.arrange(tp4,tp5,tp6)
```

4. Outlier Analysis of continuous variables [(Figure 5)]

```
#creating box plot for numeric variables

for(i in 1:length(numeric_col)){
assign(paste0("bp",i),ggplot(data =data ,aes_string(y=numeric_col[i])) +
  stat_boxplot(geom = "errorbar", width = 0.5) +
  geom_boxplot( notch = FALSE , outlier.size=1 ,notchwidth = .2,outlier.colour = "red"
,fill="blue")+
    labs(y=numeric_col[i])+
    ggtitle(paste("Box plot of ",numeric_col[i]))
)
}

#ploting boxplot
gridExtra::grid.arrange(bp1 ,bp2,bp3,bp4,bp5,bp6,bp7 ,ncol=3, nrow=3)
```

5. Correlation plot between numeric variables [(Figure 6)]

```
corrgram(numericData ,upper.panel=panel.pie ,diag.panel=panel.density,text.panel = panel.txt ,main="Correlation Plot")
```

6. Actual plot vs Predicted plot

6.1 Decision Tree (Figure 8)

```
plot(test[,13],type = 'p',col="blue" ,main = "Actual vs predicted plot blue(Actual) red(Predicted) Decision Tree" )
lines(prediction_DT ,col='red')
```

6.2 Random Forest (Figure 9)

```
plot(test[,13],type = 'p',col="blue" ,main = "Actual vs predicted plot blue(Actual) red(Predicted) Random forest", )
lines(pred_y ,col='red')
```

6.3 Linear Regression (Figure 10)

```
plot(test[,13],type = 'p',col="blue" ,main = "Actual vs predicted plot blue(Actual) red(Predicted) Linear Regression", )
lines(y_pred ,col='red')
```

7. Complete R File

```
setwd('/home/aditya/code_pen/edwiser_pro_r')
getwd()
rm(list = ls())
data = read.csv("day.csv")
head(data)
#droping instant variable as it  Record index
data$instant=NULL
head(data)

 #data frame description
str(data)
summary(data)

#data visualisation

temp =data
```

```
categorical_var
=c("season","yr","mnth","holiday","weekday","workingday","weathersit")
#converting all categorical variable to character type
for ( i in categorical_var){
  temp[,i] =as.character(temp[,i] )

}
# converting dteday to Date format
temp$dteday =as.Date(temp$dteday)
rownames(temp) =temp$dteday
# droping dteday after setting it as index
temp$dteday =NULL

temp$season[temp$season %in%  1]="Spring"
temp$season[temp$season %in%  2]="Summer"
temp$season[temp$season %in%  3]="Fall"
temp$season[temp$season %in%  4]="Winter"

temp$yr[temp$yr %in% 0]="2011"
temp$yr[temp$yr %in% 1]="2012"

temp$holiday[temp$holiday  %in% 0]="Working day"
temp$holiday[temp$holiday  %in% 1]="Holiday"

temp$weathersit[temp$weathersit %in% 1]="Clear"
temp$weathersit[temp$weathersit %in% 2]="Cloudy/Mist"
temp$weathersit[temp$weathersit %in% 3]="Rain/Snow/Fog"
temp$weathersit[temp$weathersit %in% 4]="Heavy/Rain/Snow/Fog"

temp$temp =temp$temp *39
temp$atemp =temp$atemp *50
temp$windspeed =temp$windspeed *67
temp$hum =temp$hum *100

head(temp)

#pair plot for numeric variable analysis
continous=c('temp', 'atemp', 'hum', 'windspeed','casual','registered', 'cnt')
par(mar=c(1,1,1,1))
library("ggplot2")                 # Load ggplot2 package
library("GGally")                  # Load GGally package
```

```
  ggpairs(temp[continous])

# Observation
#1. temp and atemp highly posetively correlated
#2. cnt and registered is highly positevely  correlated
#3. cnt and casual is positevely correlated
#4. casual is right skewed
#5. atemp is shows a moderate corelation toward cnt
#6. temp is shows a moderate corelation toward cnt

category =c('season', 'yr', 'mnth', 'holiday', 'weekday', 'workingday',
          'weathersit')

j=1
for( i in category){
assign(paste0("gn",j) ,ggplot(data =temp ,aes_string(x=i ,fill=i) )+
  geom_bar(stat ="count",width = 0.5 ) +
  theme_minimal()
)
j=j+1
}
gridExtra::grid.arrange(gn1,gn2,gn4,gn5,gn6,gn7,nrow=3 ,ncol=3)
gridExtra::grid.arrange(gn3 ,nrow=2 ,ncol=2)

# Observation
#1. number of holiday was less then working day
#2. there are very  few observation of Rain/snow/fog



library(dplyr)
#data summarisation with respect to target variable

cntByYear=temp %>%
  group_by(yr) %>%
  summarise(cnt=sum(cnt))

assign(paste0("tp",1),ggplot(data =cntByYear ,aes(x=yr ,y=cnt,fill=yr))+
  geom_col()+
  theme_minimal()+
  ggtitle("Number of bike rented per year ")
)
```

```
preSeasonCnt =temp %>%
  group_by(season) %>%
  summarise(cnt =sum(cnt))

assign(paste0("tp",2),ggplot(data =preSeasonCnt ,aes(x=season ,y=cnt,fill=season))+
  geom_col()+
  theme_minimal()+
  ggtitle("Number of bike rented per season ")
)

preMnthCnt =temp %>%
  group_by(mnth) %>%
  summarise(cnt =sum(cnt))

assign(paste0("tp",3),ggplot(data =preMnthCnt ,aes(x=mnth ,y=cnt,fill=mnth))+
  geom_col()+
  theme_minimal()+
  ggtitle("Number of bike rented in given month  ")
)

holidayCnt =temp %>%
  group_by(holiday) %>%
  summarise(cnt =sum(cnt))

assign(paste0("tp",4),ggplot(data =holidayCnt ,aes(x=holiday ,y=cnt,fill=holiday))+
  geom_col()+
  theme_minimal()+
  ggtitle("Number of bike rented on working and non working days  ")
)


weekdayCnt =temp %>%
  group_by(weekday) %>%
  summarise(cnt =sum(cnt))

assign(paste0("tp",5),ggplot(data =weekdayCnt  ,aes(x=weekday ,y=cnt,fill=weekday))+
  geom_col()+
  theme_minimal()+
  ggtitle("Number of bike rented on per weekday ")
)
```

```
weathersitCnt =temp %>%
  group_by(weathersit) %>%
  summarise(cnt =sum(cnt))

assign(paste0("tp",6),ggplot(data =weathersitCnt  ,aes(x=weathersit
,y=cnt,fill=weathersit))+
  geom_col()+
  theme_minimal()+
  ggtitle("Number of bike rented given weather situation  ")
)


workingdayCnt =temp %>%
  group_by(workingday) %>%
  summarise(cnt =sum(cnt))

assign(paste0("tp",7),ggplot(data =workingdayCnt  ,aes(x=workingday
,y=cnt,fill=workingday))+
  geom_col()+
  theme_minimal()+
  ggtitle("Number of bike rented given it is working day or not  ")
)

gridExtra::grid.arrange(tp1,tp2 ,tp3,tp7 ,ncol=2,nrow=2)
gridExtra::grid.arrange(tp4,tp5,tp6)
# Observations
#1. On year 2012  more user rented bike  2011
#2. On Fall season more number of people rented bike
#3. On month 8 or August most no of bike where rented  amount =351194
#4. On month 1 or january least number of bike where rented amoumt = 134933
#5. On working day most number of bike where rented amount =3214244
#6. On weekday 5  or friday most number of bike were reneted amount = 487790
#7. On weekday 0 or sunday least number of bike where rented  amount =444027
#8. On clear weather most number of bike was rented amount =2257952
#9. On Rain/snow/fog   least number of bike was rented amount =37869
#10. On holiday very few number of bike where rented compared to working day


#Exploratory data analysis

#dteday
```

```
data$dteday  = as.Date(data$dteday)
# date type can be split into day ,year ,months , weekday  .we all ready have year ,
months , weekdays
#creating dummy day variable
data$day =NA

#now we will extract day from date
for(i in 1:dim(data)[1]){
  data$day[i] = unclass(as.POSIXlt(data$dteday[i]))$mday
}

#setting date as index
rownames(data) =data$dteday
data$dteday=NULL

#season: Season (1:springer, 2:summer, 3:fall, 4:winter)
data$season = as.character(data$season)

#yr: Year (0: 2011, 1:2012)
data$yr = as.character(data$yr)

#mnth: Month (1 to 12)
data$mnth =as.character(data$mnth)

#holiday: weather day is holiday or not (extracted fromHoliday Schedule)
data$holiday =as.character(data$holiday)

#weekday: Day of the week [0 -6]
data$weekday =as.character(data$weekday)

#workingday: If day is neither weekend nor holiday is 1, otherwise is 0.

data$workingday =as.character(data$workingday)

#weathersit: (extracted fromFreemeteo)
#1: Clear, Few clouds, Partly cloudy, Partly cloudy
#2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
#3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered
#clouds
#4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
```

```
# After checking the weather situation it was found that there is no category 4 in data
data$weathersit =as.character(data$weathersit)
# day of month
data$day = as.numeric(data$day)

str(data)
temp =data

#missing value analysis
missing_col = data.frame(apply(data ,2 ,function(x){sum(is.na(x))}))
colnames(missing_col)[1] ="percentage"
missing_col$percentage =(missing_col$percentage /nrow(data))
missing_col$percentage  =missing_col[order(-missing_col$percentage),]
#zero missing value


# outlier analysis
# on continous variable
numeric_col=c('temp', 'atemp', 'hum', 'windspeed', 'casual', 'registered',
          'day')


#creating box plot for numeric variables
for(i in 1:length(numeric_col)){
assign(paste0("bp",i),ggplot(data =data ,aes_string(y=numeric_col[i])) +
  stat_boxplot(geom = "errorbar", width = 0.5) +
  geom_boxplot( notch = FALSE , outlier.size=1 ,notchwidth = .2,outlier.colour = "red"
,fill="blue")+
    labs(y=numeric_col[i])+
    ggtitle(paste("Box plot of ",numeric_col[i]))
)
}

#ploting boxplot
gridExtra::grid.arrange(bp1 ,bp2,bp3,bp4,bp5,bp6,bp7 ,ncol=3, nrow=3)
        # We need to remove outlier from casual ,hum,wind speed



#removing outlier from numerical variables
for(i  in numeric_col ){
  print(i)
```

```
#fetching outlier values
val=data[,i][data[,i] %in% boxplot.stats(data[,i])$out]
print(length(val))
# filling outliers with na
data[,i][data[,i] %in% val]=NA

}

# missing value after outlier removal
colSums(is.na(data))

#knn imputing value as mean and median are giving poor results
library(DMwR)

#coverting categorical value to number, not as data is all ready  encoded
categorical_var
=c("season","yr","mnth","holiday","weekday","workingday","weathersit")

for(i in categorical_var){
  data[,i]=as.numeric(data[,i])
}
#knn imputation
data=knnImputation(data ,k=5)
# imputated 0 missing value
colSums(is.na(data))

# correlation plot
numeric_col=c('temp', 'atemp', 'hum', 'windspeed', 'casual', 'registered',
            'day' ,'cnt')
numericData=data[numeric_col]
library(corrgram)
#correlation plot
corrgram(numericData ,upper.panel=panel.pie ,diag.panel=panel.density,text.panel =
panel.txt ,main="Correlation Plot")
# temp is positevely correlated with atemp we should drop atemp
# removing atemp  from data
data$atemp =NULL

colnames(data)

#Feature scaling
#Normality check
```

```
# ploting distribution plot

  j=1
  for( i in numeric_col ){
  assign(paste0("dp",j),ggplot(numericData , aes_string(x=i))+
    geom_density(fill="palegreen3")+
    theme_minimal()+
    ggtitle(paste("Distribution plot of ", i)) )
    j=j+1
  }

gridExtra::grid.arrange(dp1,dp2,dp3 ,dp4 ,dp5 ,dp6,dp7 ,dp8 ,ncol=3,nrow=3)
#most of the distribution are not normal

#humidity and wind speed is some what normal
numeric_var =c('temp' ,'hum','windspeed','casual','registered'  ,'day')
#normalising data

temp =data
#normalising data
for(i in numeric_var ){
  print(i)
 data[,i]=(data[,i] -max(data[,i])) / (min(data[,i]) - max(data[,i]))
}


#Model development

# Decision tree
# Random forest
# Linear regression




library(DataCombine)
library(caret)
set.seed(123)

#Simple random sampling
train.index =sample(1:nrow(data) ,.8 *nrow(data))
train = data[train.index,]
```

```
test =data[-train.index,]

#dimension of test train data
dim(train)
dim(test)

##Decision tree for classification
#Develop Model on training data
library(rpart)
library(MASS)
#Train Decision tree
dtmodel=  rpart(cnt~., data =train ,method ='anova')
#Prediction on test data
prediction_DT =predict(dtmodel ,test[-13])
library("rpart.plot")
rpart.plot(dtmodel,box.palette="RdBu", shadow.col="gray", nn=TRUE)
#MAPE function
MAPE =function (act ,pred){
   mean(abs((act-pred)/act)) *100
}
#Test MAPE =10.6%
#Accuracy 89.4%
#MAPE score on test data
MAPE(test[,13] ,prediction_DT)

#Model evalution with  multiple error metric
regr.eval(trues = test[,13], preds = prediction_DT, stats = c("mae","mse","rmse","mape" ))
#mae         mse         rmse         mape
#4.116334e+02 2.668836e+05 5.166077e+02 1.061794e-01

#actual vs predicted plot
plot(test[,13],type = 'p',col="blue" ,main = "Actual vs predicted plot blue(Actual)
red(Predicted) Decision Tree" )
lines(prediction_DT ,col='red')

######################### Random forest ###################################

library(randomForest)
#train  Random Forest
rf_model =randomForest(cnt~. , train , importance=TRUE ,ntree=50)
#predict test data
pred_y =predict(rf_model , test[-13])
```

```
#MAPE test data 4.7%   ntree =50  , increasing ntree doesnt improve the model
#inceasing trees may over train model
#accuracy test 95.3%
#MAPE score on test data
MAPE(test[,13],  pred_y)

#model evalution with  multiple error metric
regr.eval(test[,13] , preds = pred_y, stats = c("mae","mse","rmse","mape"))
#mae         mse        rmse        mape
#1.749346e+02 6.959832e+04 2.638149e+02 4.797810e-02

#actual vs predicted plot
plot(test[,13],type = 'p',col="blue" ,main = "Actual vs predicted plot blue(Actual)
red(Predicted) Random forest", )
lines(pred_y ,col='red')

###################### Linear Regression ########################
  #MAPE test =3.07%
#accuracy on test data 97%
#training linear regression
linearmodel =lm(cnt~.,data =train )
#model summary
summary(linearmodel)
#prediction on test data
y_pred = predict(linearmodel ,test[-13])

#MAPE score on test data
MAPE(test[,13],y_pred)

#model evalution with  multiple error metric
regr.eval(trues = test[,13], preds = y_pred, stats = c("mae","mse","rmse","mape"))
#mae         mse        rmse        mape
#1.227498e+02 5.044234e+04 2.245937e+02 3.070828e-02

#actual vs predicted plot
plot(test[,13],type = 'p',col="blue" ,main = "Actual vs predicted plot blue(Actual)
red(Predicted) Linear Regression", )
lines(y_pred ,col='red')


# Overall Linear regression is best model compared to others
# Linear regression gives best accuracy and low error rate
```

Accuracy =97% in test data
MAPE test =3.07%

References

1.https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e
2.https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba

3.https://towardsdatascience.com/analyze-the-data-through-data-visualization-using-seaborn-255e1cd3948e