Prediction of Bike Rental Count based on Environmental and Seasonal settings

Alfazal M

Table of Contents

1.Introduction

Problem Statement

The objective of the project is to predict the count of the booking of bikes on rent based on environmental and seasonal settings. We will use various data predicting models by using the data we have. And find out which is best suitable model and use that model for prediction. By making a prediction method it will be beneficial for bike renting service since they can be prepared for renting those no. of bikes. it will be useful especially in peak period of bike rent.

Data

Our aim is to develop a model to predict the count of bike rent(‘cnt’). Given below is the sample of data. Using that data we will develop a model

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

Given below are expansion of abbreviated column names:
dteday: Date
yr: Year
mnth: Month
weathersit: Weather Situation
temp: Temperature
atemp: Actual Temperature
hum: Humidity
cnt: Count

Our target variable is ‘cnt’. Other variables which predict the count variable are:
1.instant
2.dteday

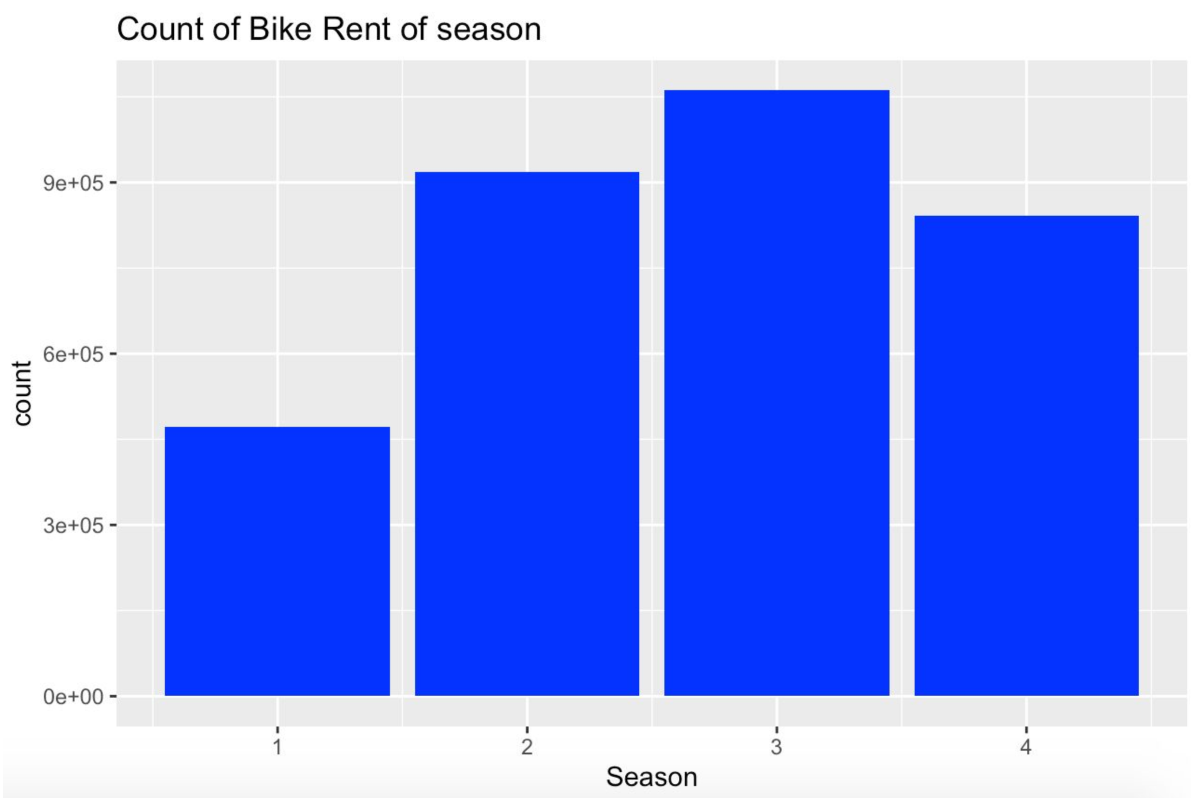
3.season
4.yr
5.mnth
6.holiday
7.weekday
8.workingday
9.weathersit: Weather Situation
10.temp
11.atemp
12.hum: Humidity
13.windspeed
14.casual
15.registered
The ‘cnt’ variable is not dependent on 'instant’ and ‘dteday’ variables. cnt variable is sum of casual and registered variables. So those variables(instant,dteday,casual,registered) need not be considered.

2.Methodology

In methodology data processing, model development and deploying are employed
Data Preprocessing: It includes missing value analysis,outlier analysis,feature selection and feature scaling
Modeling: Various models are deployed on given data and the best model is chosen.

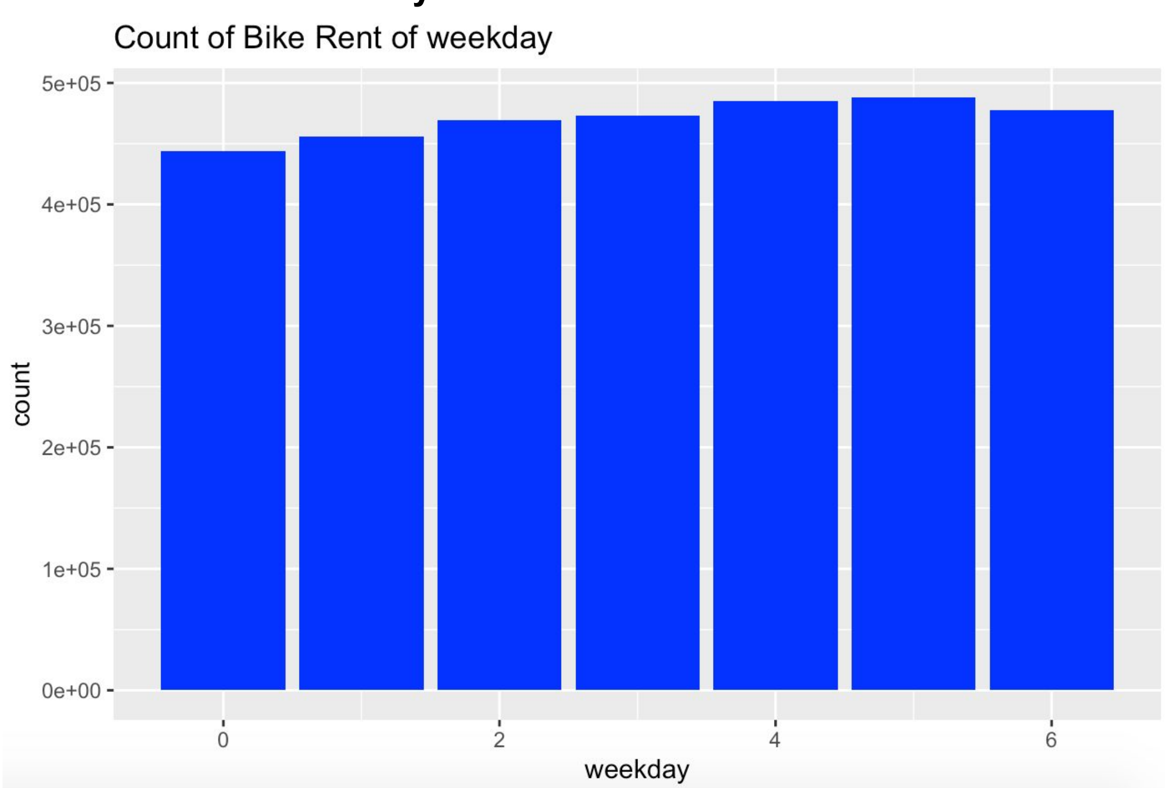
Data Understanding

Plotting graphs will help to understand data. Cnt vs categorical variables are plotted to understand where ‘cnt’ maximum.
Count of Bike VS Season



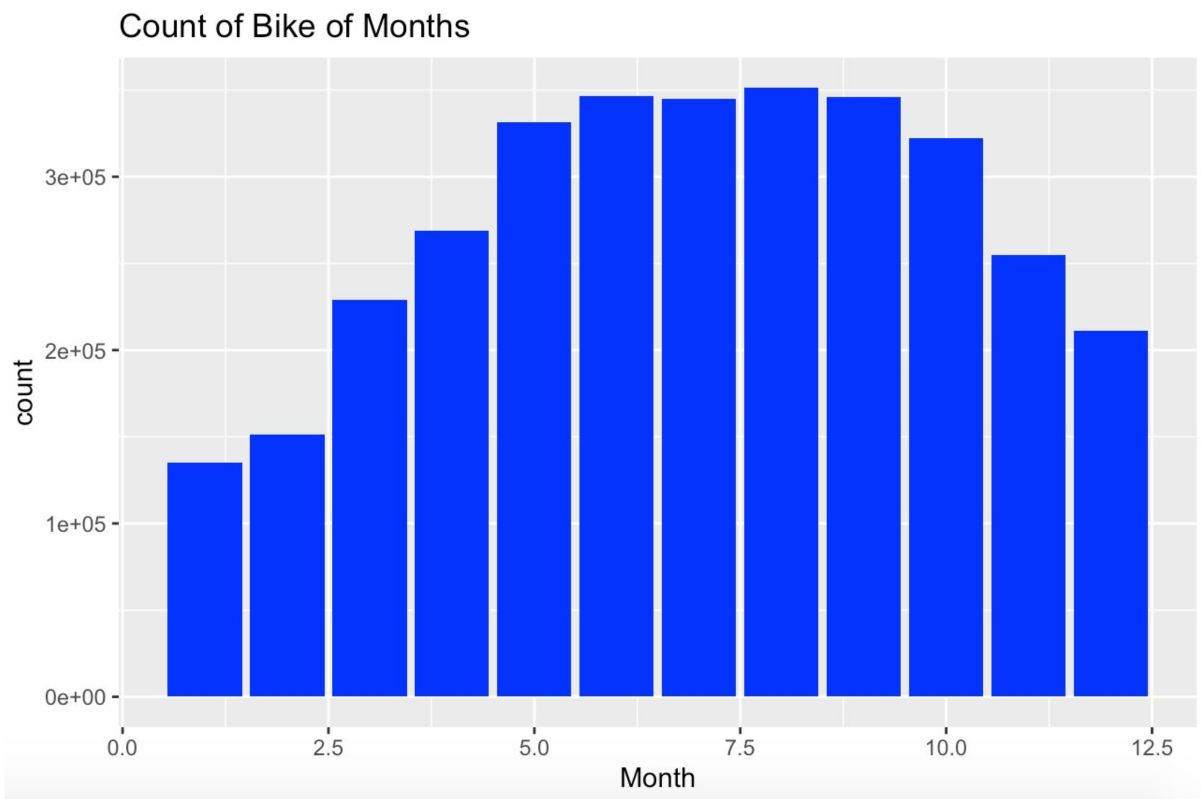
It is observed in season3 the bike rent count is maximum

Count of Bike vs Weekday



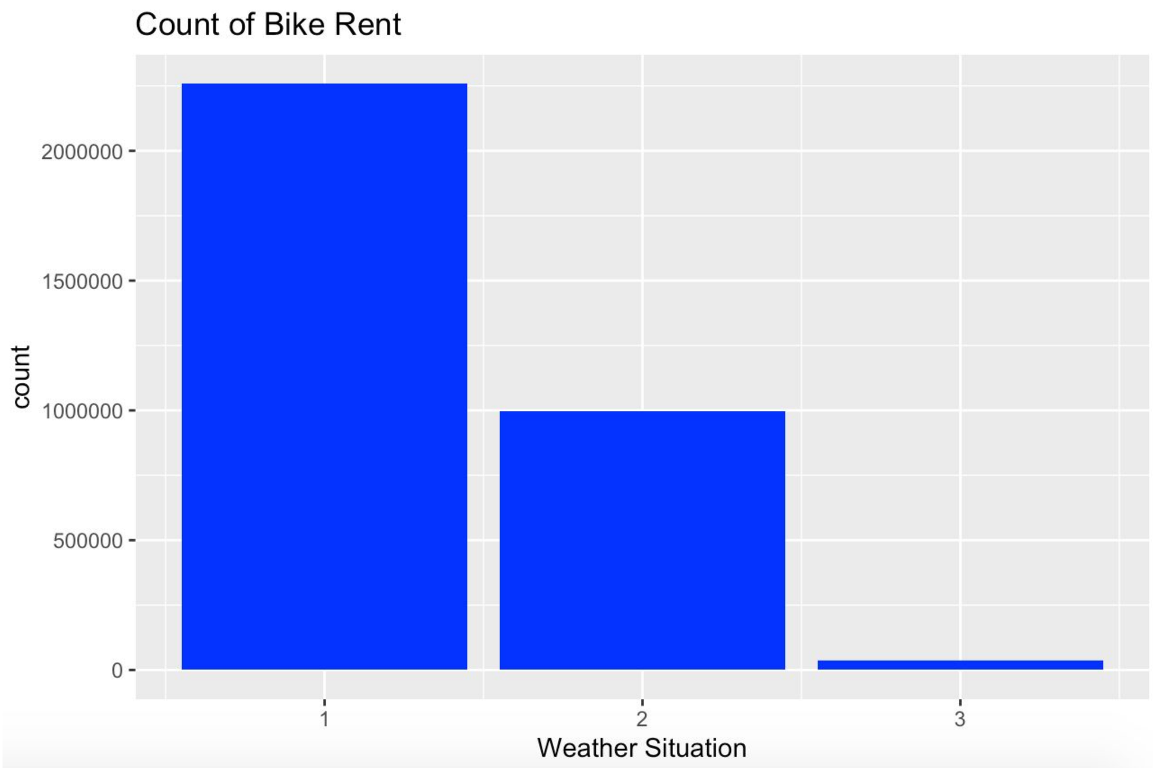
There is no much difference in count in weekdays 5,6&6 but count is maximum in weekday 5.

Count of Bike vs Months



In month 8 the bike rent count is maximum.

Count of Bike Rent vs Weather Situation



In weather situation 1 the count is maximum.

Data Preprocessing

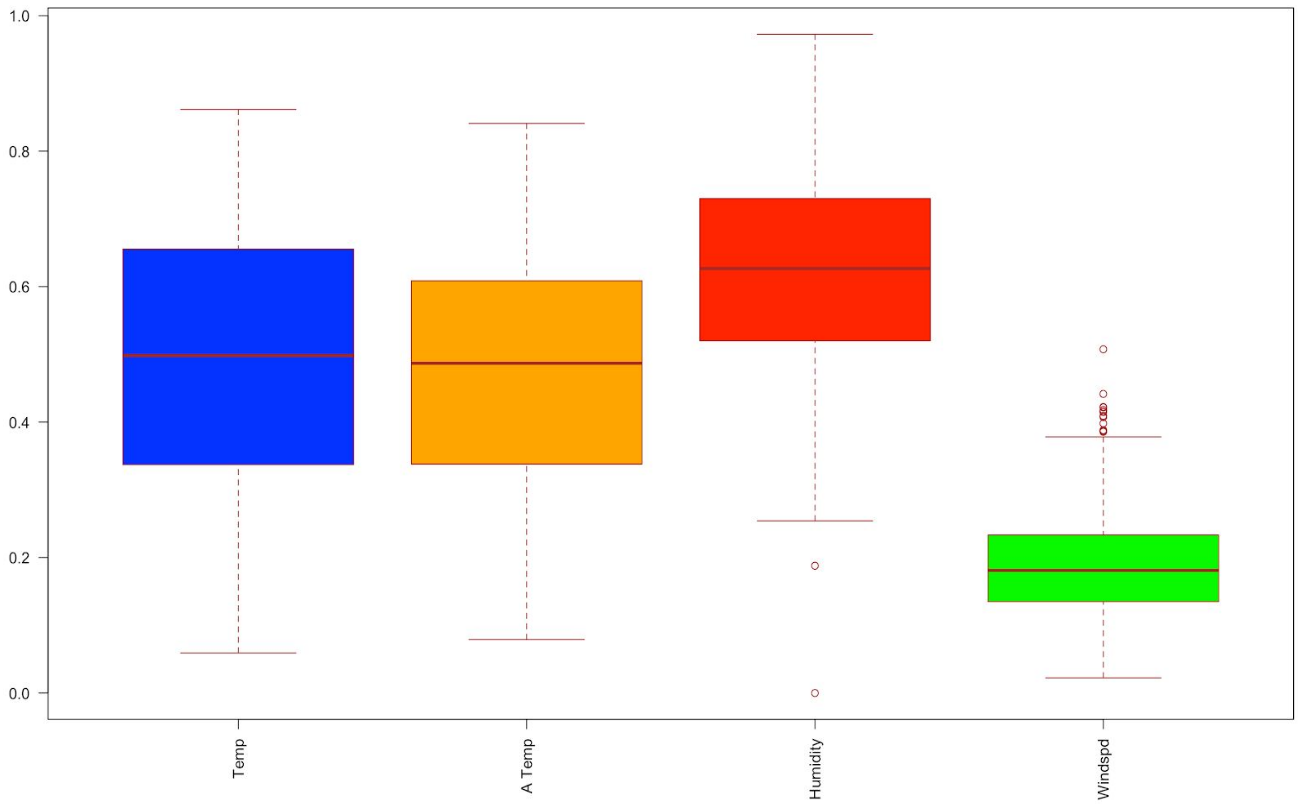
We are developing various models using the sample data so the data should be without any missing value and the data should be refined. For that we are doing data preprocessing. Also we can drop some variables which helps to develop the model easily.

Missing Value Analysis

We are checking whether there is any missing value in the dataset. In the dataset it is found there is no missing value in the data.

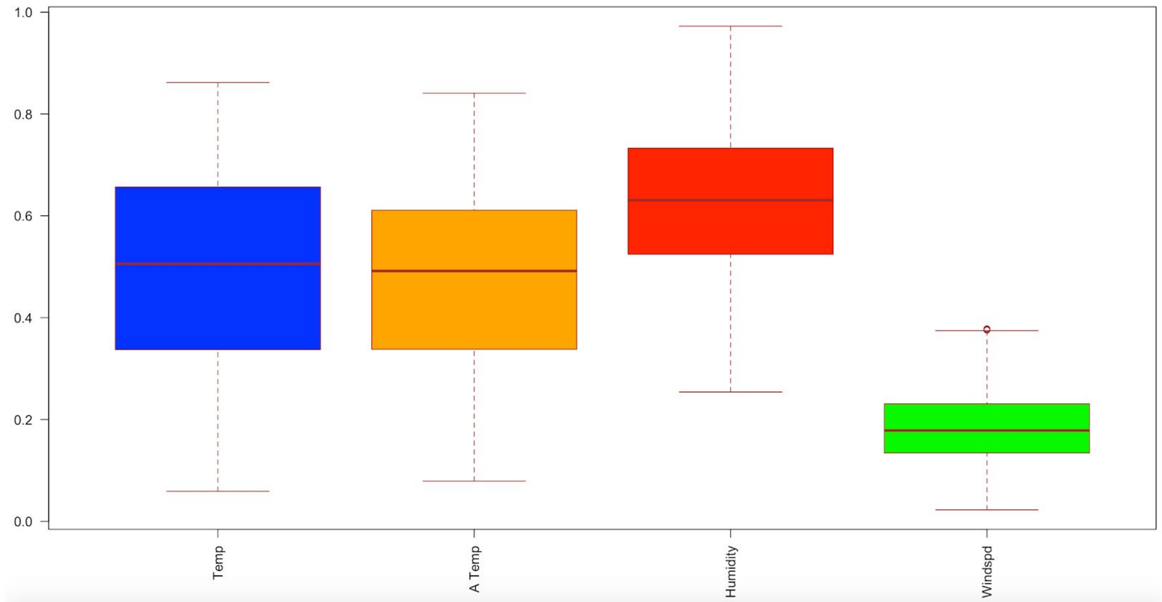
Outlier Analysis

Outliers are those data points which stand outside the overall pattern and distribution of data. Outliers are included in the data because of error of sensors we use to detect the observation, error in entering observations...etc. Outliers are detected in this project using box plots of numeric variables. The box plot of numeric variables are given below.



It is found from boxplot there are outliers in Humidity and Wind Speed variables.

Outliers are removed in humidity windspeed let’s see the box plot after removal of outliers

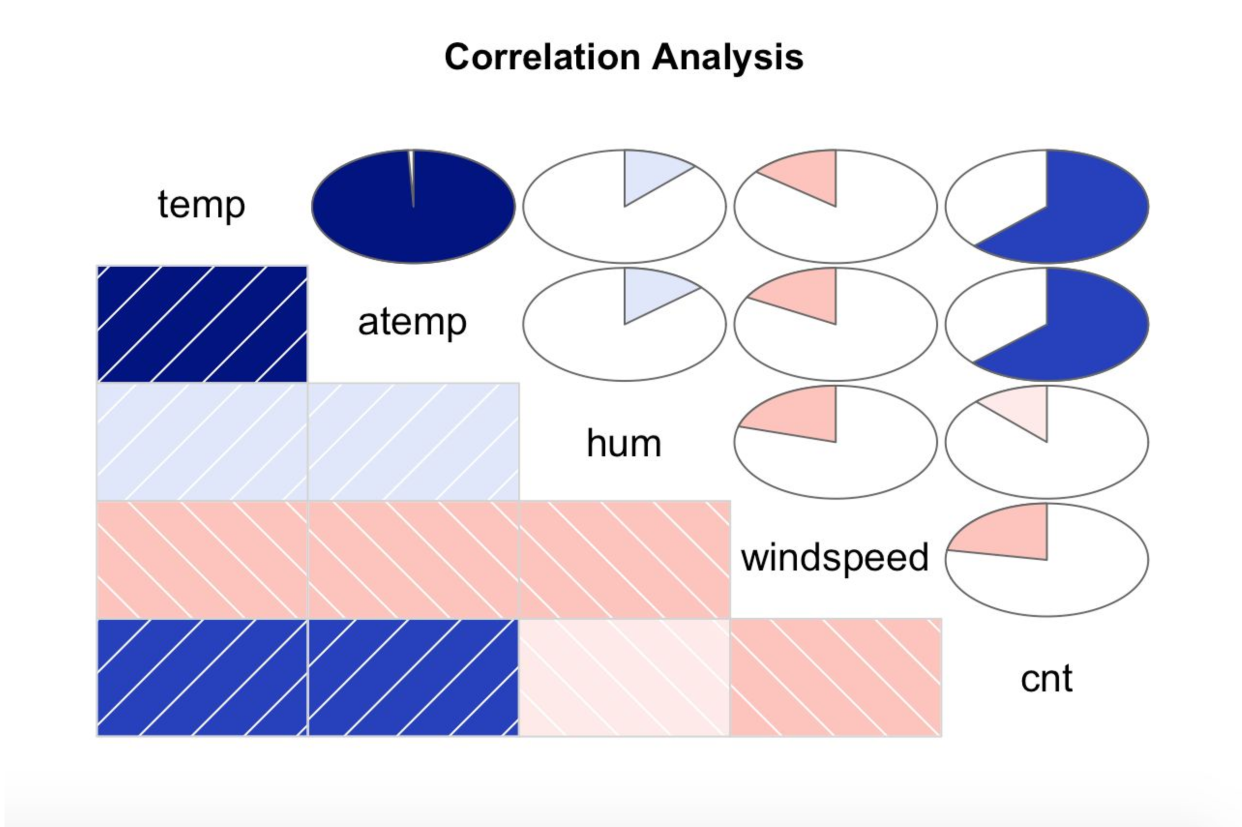


Outliers are removed in humidity and windspeed.
Outliers values are imputed using KNN imputation in R and mean of the corresponding variables in python

Feature Selection

There will be some redundant variables which doesn’t contribute significantly to predict the target variables. These variables should be removed because it will hamper the accuracy of modeling. Also some variables will be highly correlated to each other so one among them is enough need to be considered in model developing. Correlation plot is plotted for numeric variables to check collinearity and anova test is done for categorical variables to check whether target variable has dependency over each categorical variable.

Correlation Analysis



It is found ‘temp’ variable and ‘atemp’ variables are highly correlated each othe from pie chart. So ‘atemp’ is not considered for modeling and prediction.
Anova test is done for categorical variables.

Anova Test Summary

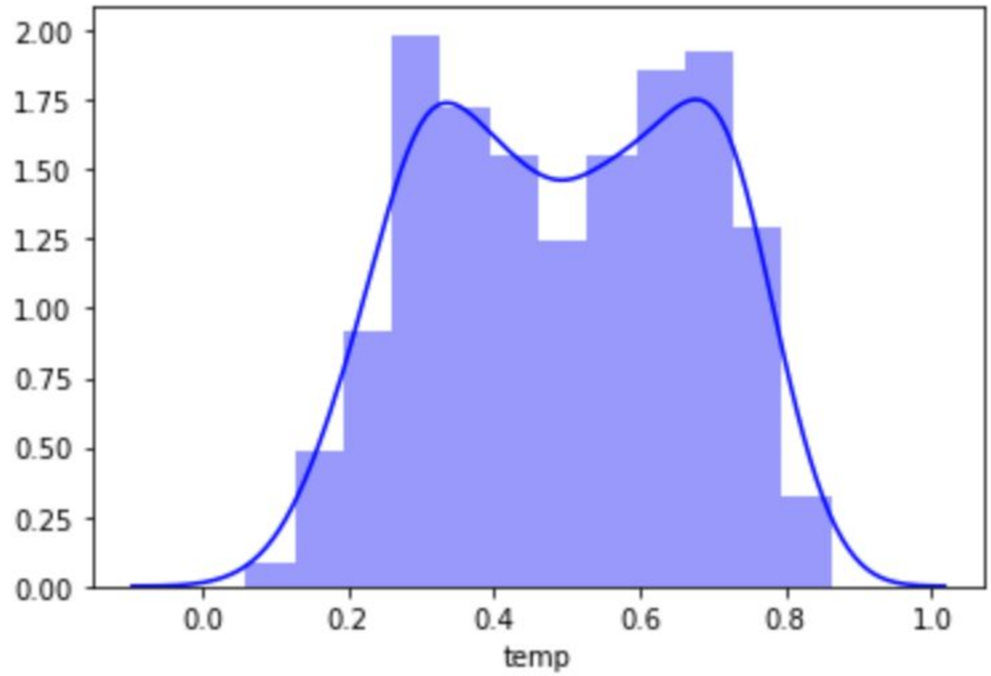
```
                sum_sq     df           F        PR(>F)
season    4.517974e+08    1.0  143.967653  2.133997e-30
Residual  2.287738e+09  729.0         NaN           NaN
                sum_sq     df           F        PR(>F)
yr        8.798289e+08    1.0  344.890586  2.483540e-63
Residual  1.859706e+09  729.0         NaN           NaN
                sum_sq     df          F        PR(>F)
mnth      2.147445e+08    1.0  62.004625  1.243112e-14
Residual  2.524791e+09  729.0        NaN           NaN
                sum_sq     df         F    PR(>F)
holiday   1.279749e+07    1.0  3.421441  0.064759
Residual  2.726738e+09  729.0       NaN       NaN
                sum_sq     df         F    PR(>F)
weekday   1.246109e+07    1.0  3.331091  0.068391
Residual  2.727074e+09  729.0       NaN       NaN
                  sum_sq     df         F    PR(>F)
workingday  1.024604e+07    1.0  2.736742  0.098495
Residual    2.729289e+09  729.0       NaN       NaN
                  sum_sq     df          F        PR(>F)
weathersit  2.422888e+08    1.0  70.729298  2.150976e-16
Residual    2.497247e+09  729.0        NaN           NaN
```

It is found P values are higher than 0.05 for weekday,working day and holiday So null hypothesis is accepted (Target variable doesn’t have dependancy over those variables). So those variables are dropped. It will help to reduce the dimension of data.

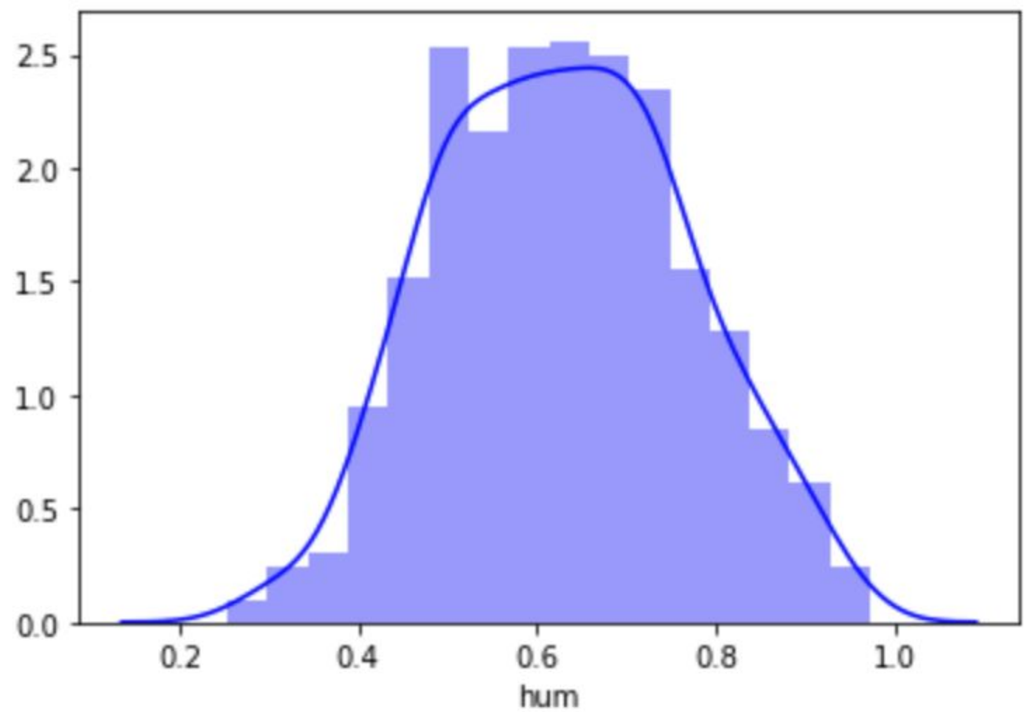
Feature Scaling

In this dataset it is found dataset is approximately symmetric So need of feature scaling. Plot of variables which show dataset is symmetric is given below.

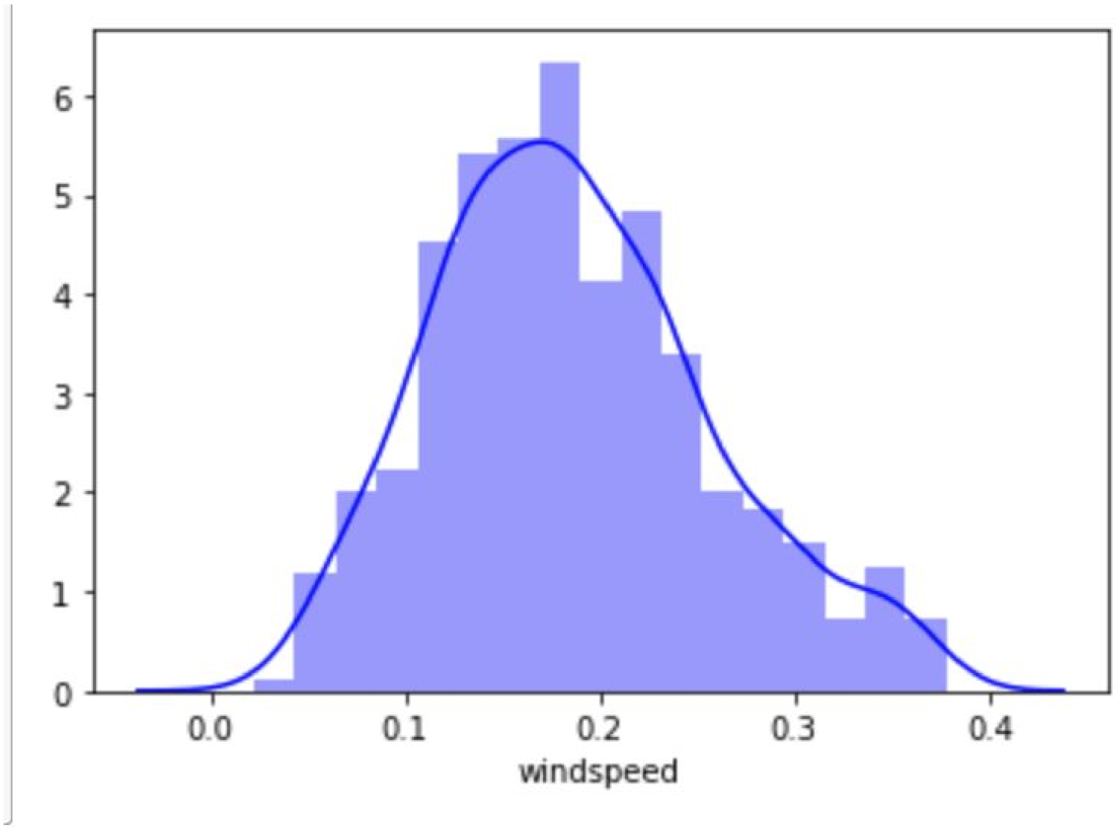
1. temp



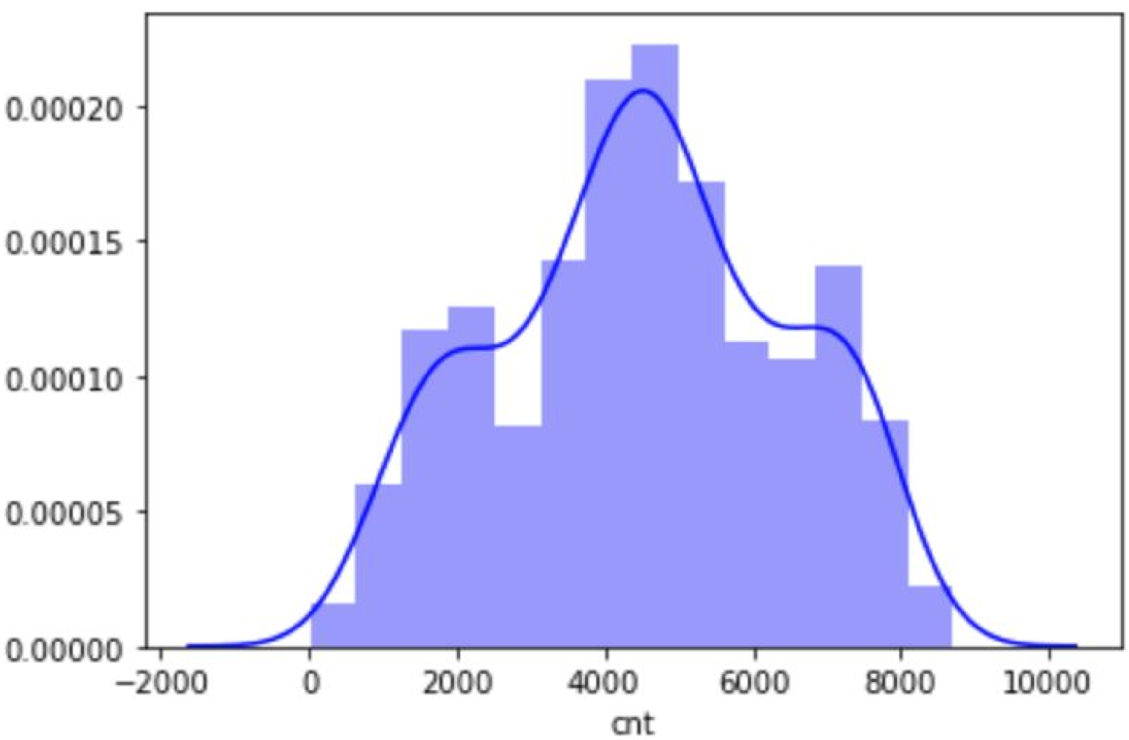
2.’hum’



3.’windspeed’


6
5
4
3
2
1
0
0.0
0.1
0.2
0.3
0.4
windspeed

4.'cnt'


0.00020
0.00015
0.00010
0.00005
0.00000
−2000
0
2000
4000
6000
8000
10000
cnt

Modeling

There are many machine learning algorithms to predict the target variable. Since the target variable is numeric we are trying to predict the target variables using Decision Tree,Random Forest and Linear Regression models. These models are developed using 80 percent of the dataset (Train Data) and we will test it on the rest of data. And the models are compared. Since we are using some datas to model the data it is called supervised learning.

Decision Tree

Decision tree is a supervised machine learning algorithm which uses binary rules to predict target variable. Decision tree model is deployed in R and Python
Decision Tree Rules

```
1) root 584 2146661000 4519.848
  2) temp< 0.4329165 245  541603500 3148.122
    4) yr0>=0.5 125  135210300 2297.920
      8) season4< 0.5 82   30974330 1720.610 *
      9) season4>=0.5 43   24789660 3398.837 *
    5) yr0< 0.5 120  221917300 4033.750
     10) temp< 0.2804165 32   26119320 2691.844 *
     11) temp>=0.2804165 88  117221500 4521.716
       22) season1>=0.5 37   19791990 3880.676 *
       23) season1< 0.5 51   71194240 4986.784
         46) hum>=0.765417 8   14086640 3193.000 *
         47) hum< 0.765417 43   26577220 5320.512 *
  3) temp>=0.4329165 339  810887700 5511.212
    6) yr0>=0.5 161  117805300 4278.484
     12) weathersit3>=0.5 8    1031700 2320.500 *
     13) weathersit3< 0.5 153   84500320 4380.863 *
    7) yr0< 0.5 178  227131700 6626.208
     14) hum>=0.7710415 20   46987470 4965.450 *
     15) hum< 0.7710415 158  117999300 6836.430 *
```

The plot shows the splitting of trees. For example splitting occurs in terms of temperature (Second line in the plot) data is split which has temperature value greater than and less than 0.4329165 are splitted.

Then those datas with temperature values less than specified values are splitted further in terms of year (Third line). Decision tree model is developed in R and in Python.
MAPE = Mean Absolute Percentage Error
R Square Value
MAE = Mean Absolute Error
Decision Tree in R Summary

MAPE = 26.76604
MAE = 796.4505777
RSquare = 0.8171842

Decision Tree in Python Summary

MAPE= 36.94809301452646
R Square = 0.808987445722944

Random Forest

Random forest is another supervised machine learning algorithm which selects random observations and uses multiple decision trees to predict the target variable. The no. of trees is set as 500 in random forest. Random forest model is deployed in R and Python as well.

Random Forest in R Summary

MAPE = 19.51173
MAE = 547.4096596
RSquare = 0.9167251

Random Forest in Python Summary

MAPE = 20.668671141459097
RSquare = 0.9418756962002987

Linear Regression

Linear Regression method is used to predict target variable using one or more than one input variable. Linear regression finds out a linear relationship with input variables and target variables.

Linear Regression in R

```
Residuals:
    Min      1Q  Median      3Q     Max 
-3785.9  -341.1    80.1   478.4  3009.1 

Coefficients: (4 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3703.71     423.35   8.749  < 2e-16 ***
season1     -1510.66     209.71  -7.204 1.89e-12 ***
season2      -676.02     247.37  -2.733  0.00648 ** 
season3      -891.28     223.48  -3.988 7.54e-05 ***
season4           NA         NA      NA       NA    
yr0         -1985.04      67.98 -29.202  < 2e-16 ***
yr1               NA         NA      NA       NA    
mnth1         -67.10     213.31  -0.315  0.75323    
mnth2         155.98     218.21   0.715  0.47501    
mnth3         552.05     213.77   2.582  0.01006 *  
mnth4         517.92     287.64   1.801  0.07231 .  
mnth5         691.88     295.56   2.341  0.01958 *  
mnth6         460.13     302.60   1.521  0.12893    
mnth7          65.81     320.64   0.205  0.83745    
mnth8         622.76     307.36   2.026  0.04322 *  
mnth9        1147.34     249.16   4.605 5.11e-06 ***
mnth10        422.97     185.05   2.286  0.02264 *  
mnth11       -106.23     173.18  -0.613  0.53985    
mnth12            NA         NA      NA       NA    
weathersit1  1728.99     230.10   7.514 2.27e-13 ***
weathersit2  1342.96     208.92   6.428 2.76e-10 ***
weathersit3       NA         NA      NA       NA    
temp         4652.79     475.01   9.795  < 2e-16 ***
hum         -1770.41     355.38  -4.982 8.40e-07 ***
windspeed   -2848.63     517.12  -5.509 5.51e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 802.9 on 563 degrees of freedom
Multiple R-squared:  0.8309,	Adjusted R-squared:  0.8249 
F-statistic: 138.3 on 20 and 563 DF,  p-value: < 2.2e-16
```

Linear Regression in R Summary

MAPE = 21.11433
MAE = 642.7422879
RSquare= 0.86049

Linear Regression in Python Summary

MAPE = 18.783428096440577
RSquare = 0.9185237800664934

3.Conclusion

We have deployed various models (Decision Tree, Random Forest and Linear Regression) on our data. We have splitted data into train and test data. We applied those models on data and compared test data prediction with actual data values. From these we measured average error values. By looking at error values we can select a model for bike rent prediction.

MAPE (Mean Absolute Percentage Error)

Mean absolute percentage error is calculated using taking the mean of absolute value of predicted value and actual value and multiplied with 100.
Let's see the MAPE values for various models.

Method	Decision Tree	Random Forest	Linear Regression
MAPE in R	26.76604	19.51173	36.94809301
MAPE in Python	36.94809301	20.66867114	18.7834281

It is found MAPE value is least in Random Forest method in R and least in Linear Regression in Python.

R Square

R square shows the strength of relation between predicted value and actual value. The higher the R Square value the higher the accuracy of the model. Basically we are taking the square of the correlation of predicted value and actual value to compute R Square value.
Let's see the R square values of various models.

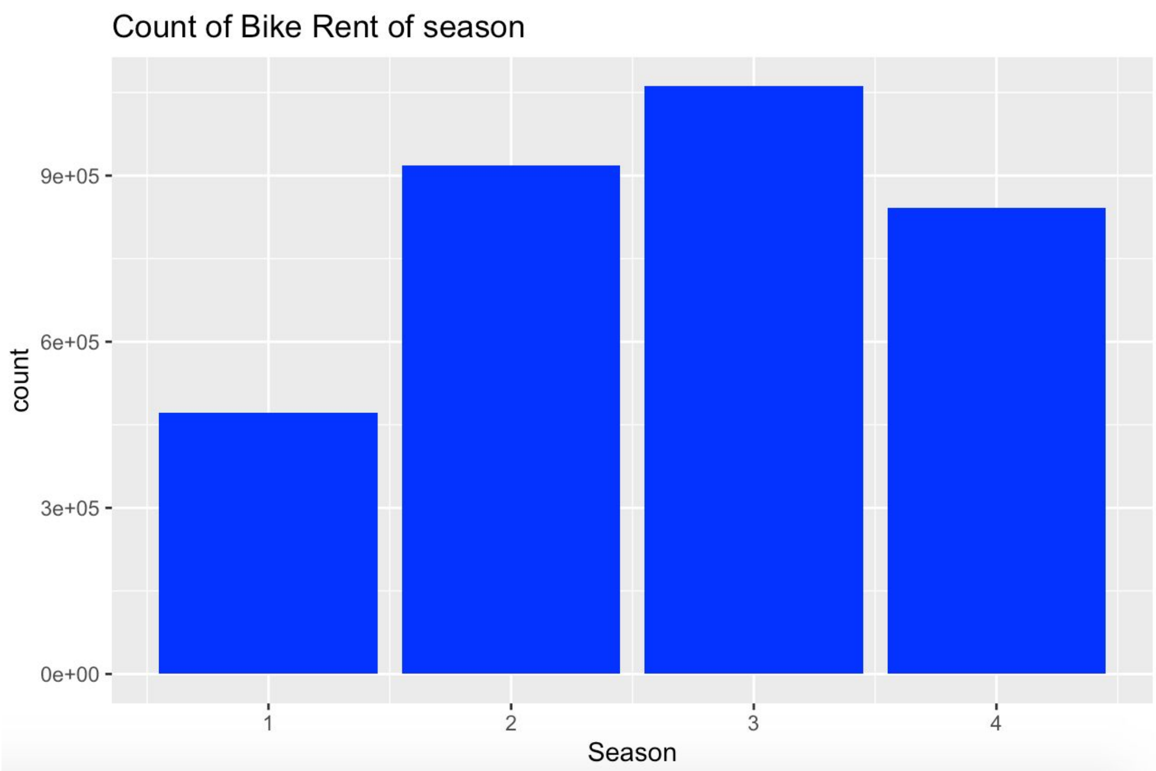
Method	Decision Tree	Random Forest	Linear Regression
RSquare in R	0.8171842	0.9167251	0.86049
RSquare in Python	0.8089874457	0.9418756962	0.9185237801

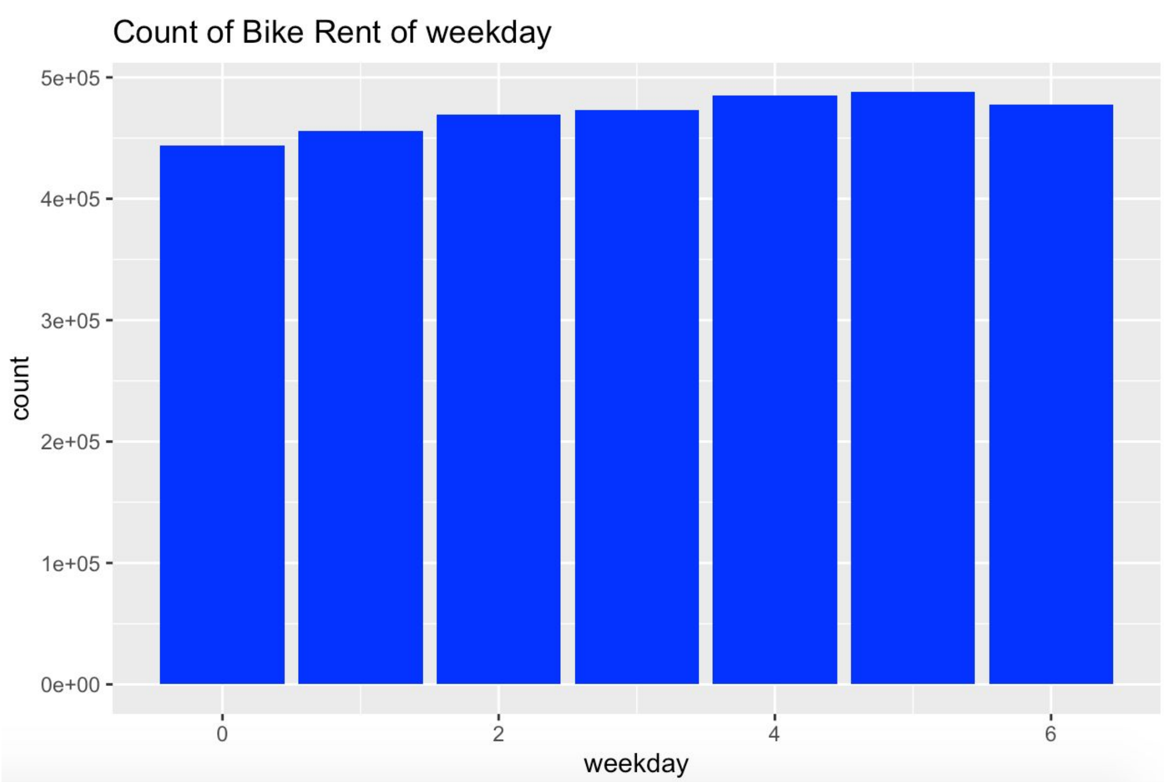
R square is highest for Random Forest in R and Python

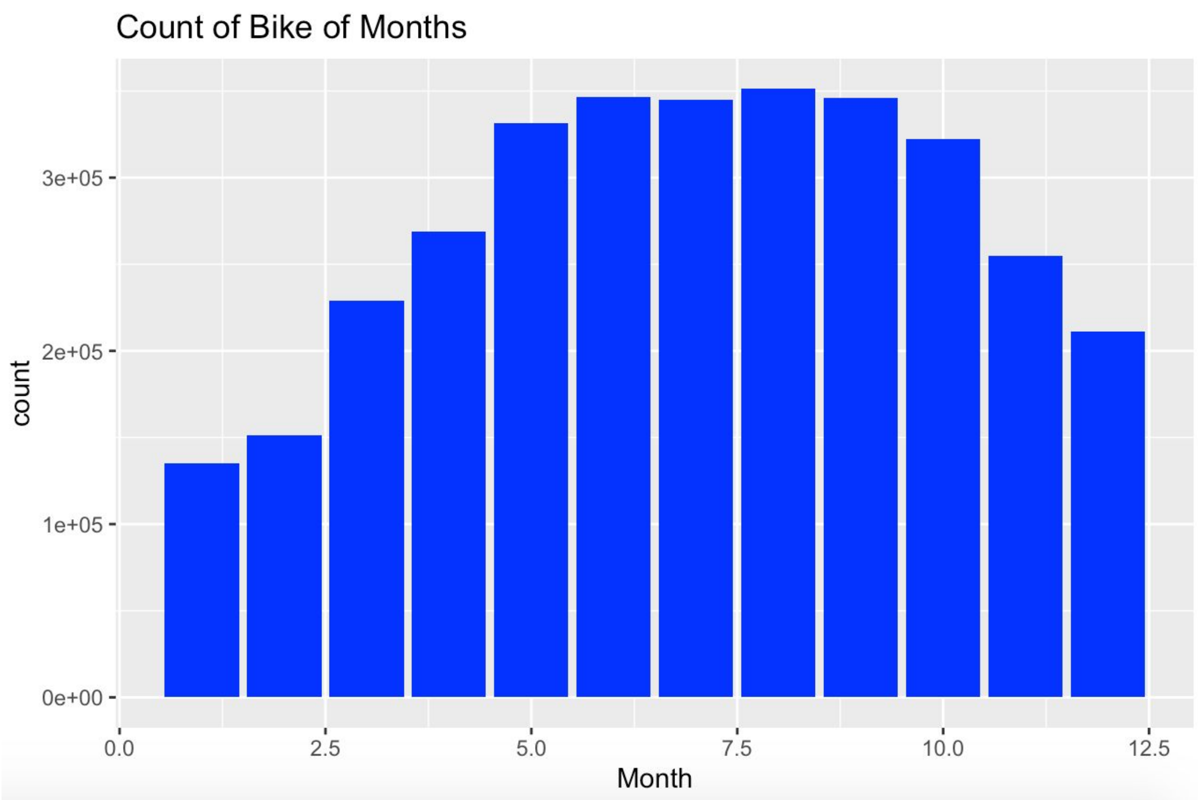
Model selection

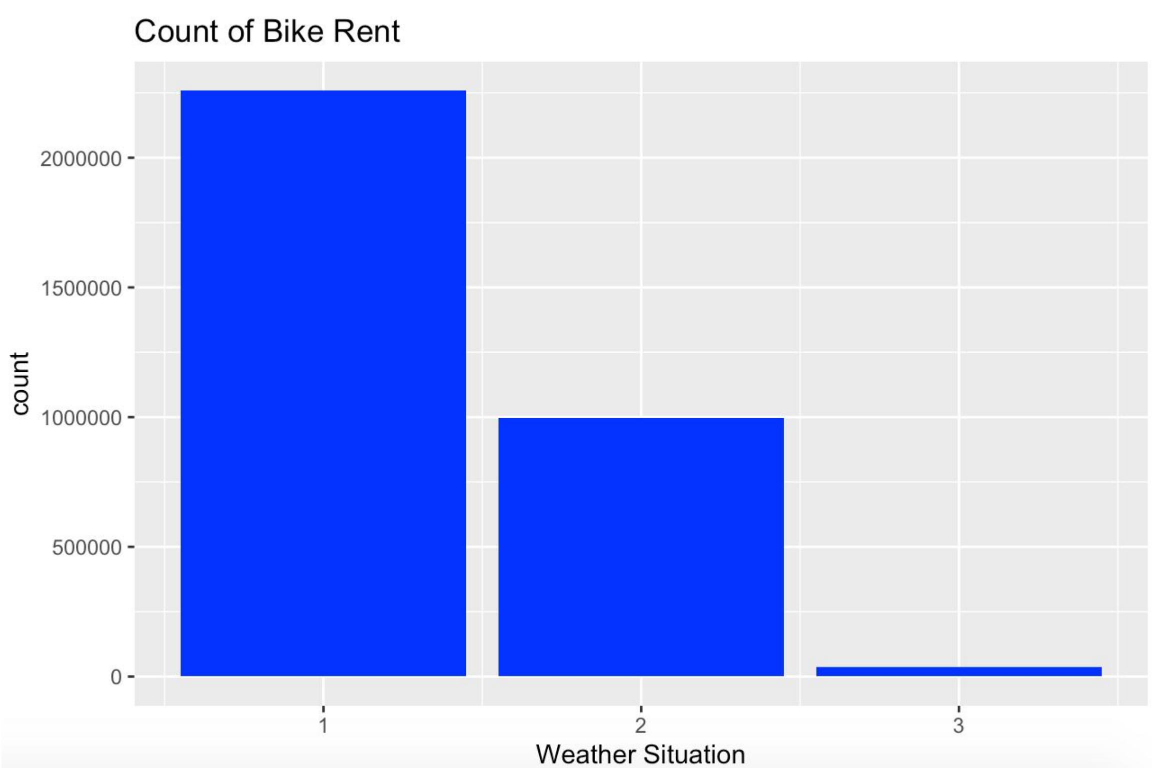
MAPE is least in Random Forest in and R. But MAPE is least in Linear Regression in Python So we can use both Random Forest and Linear Regression while considering MAPE alone. But considering R Square value R Square value is higher for Random Forest in R and Python. So Random Forest is the best suitable method.

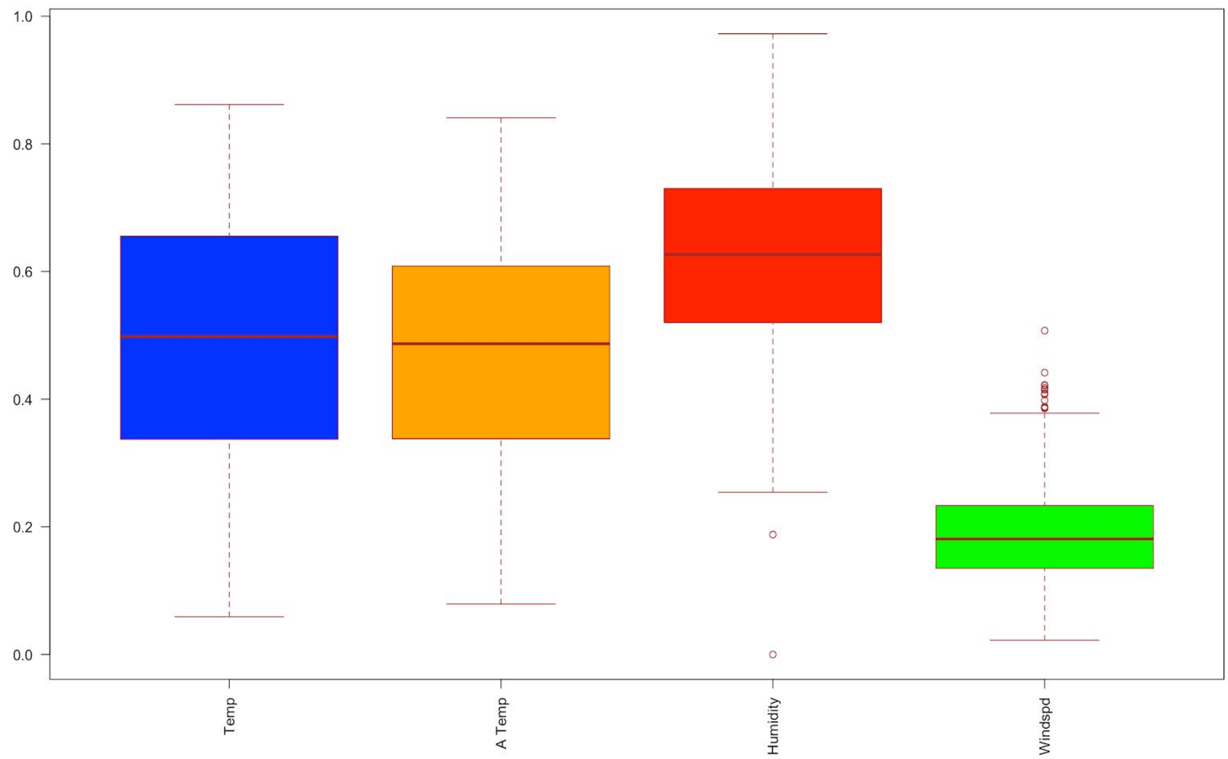
Appendix

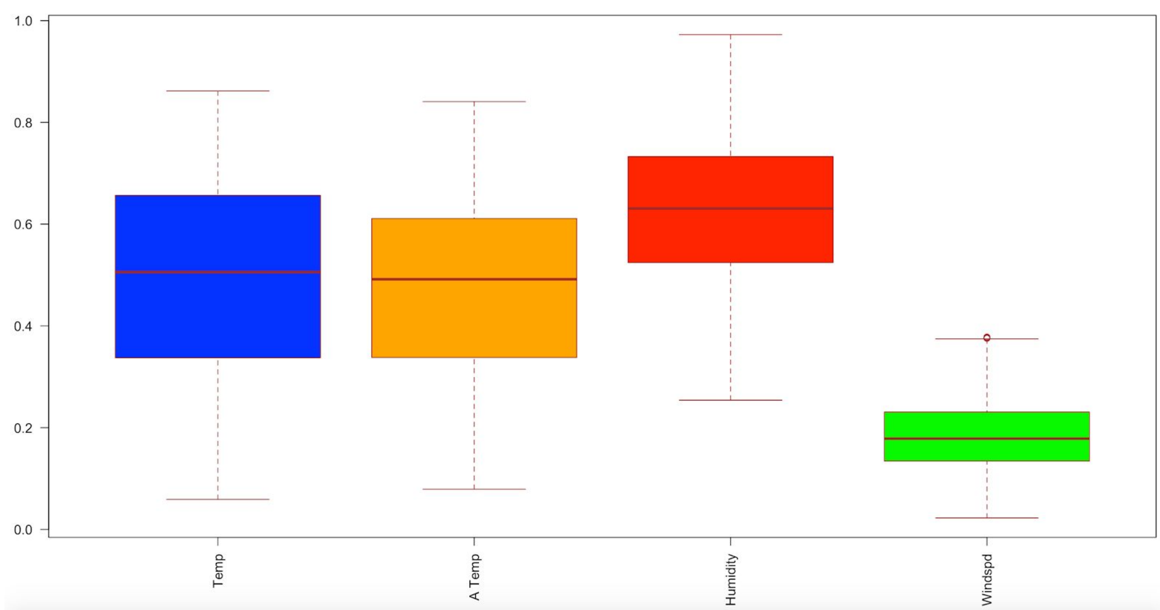

Count of Bike Rent of season
count
9e+05
6e+05
3e+05
0e+00
1
2
3
4
Season

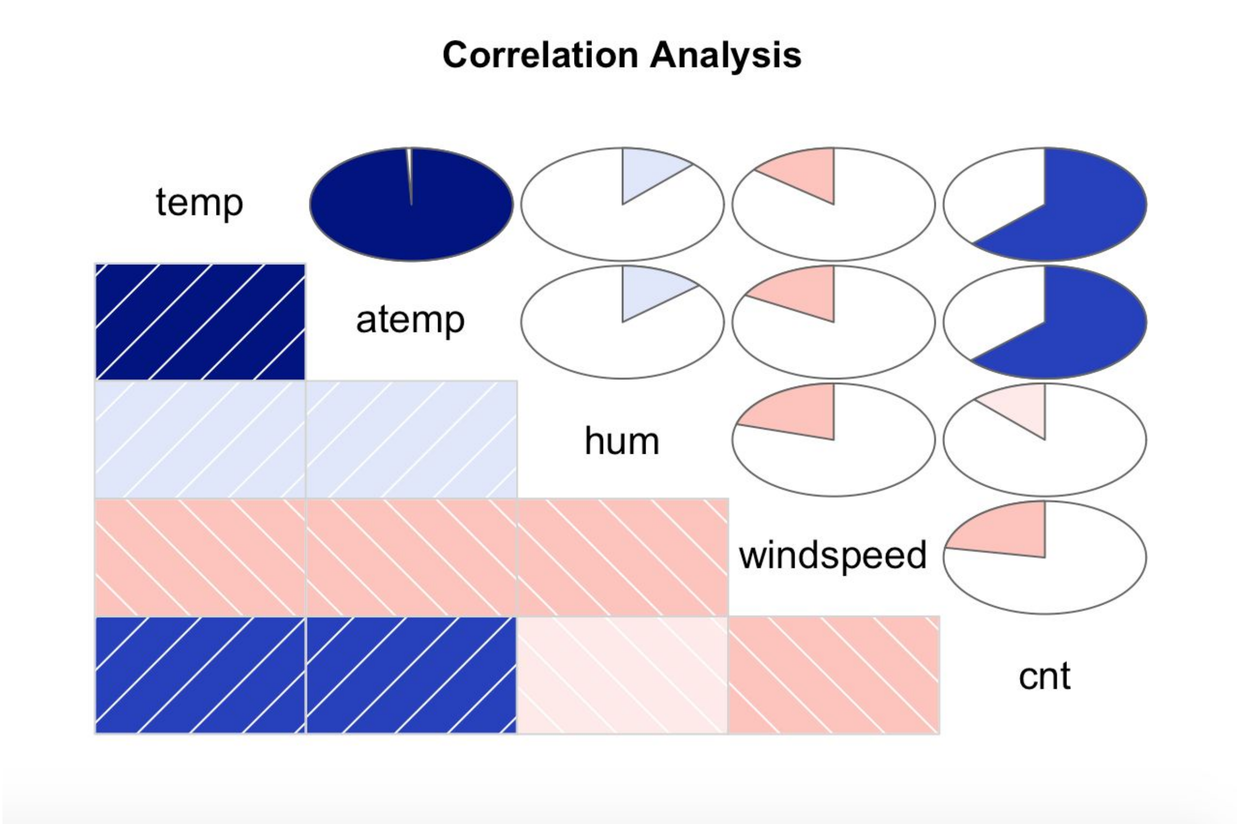

Count of Bike Rent of weekday
count
5e+05
4e+05
3e+05
2e+05
1e+05
0e+00
0
2
4
6
weekday

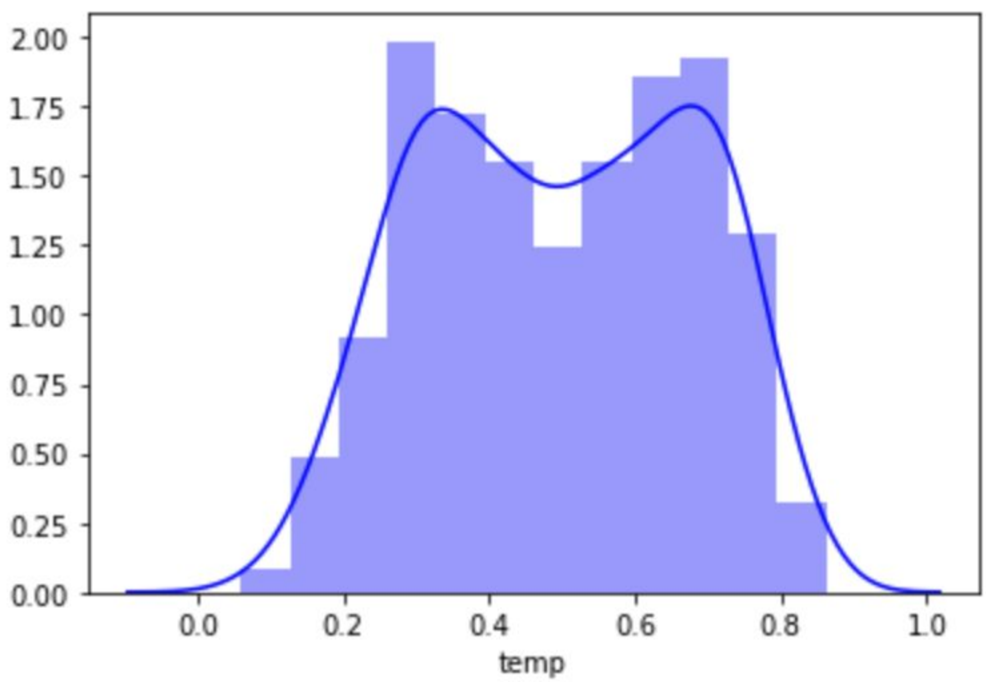

Count of Bike of Months
count
3e+05
2e+05
1e+05
0e+00
0.0
2.5
5.0
7.5
10.0
12.5
Month

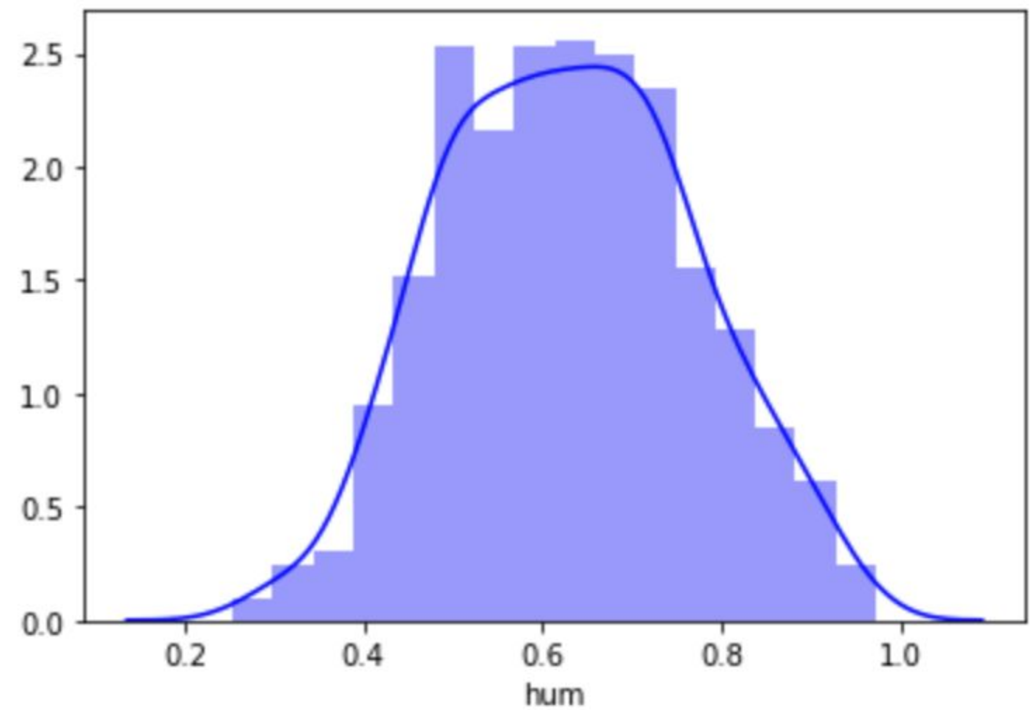

Count of Bike Rent
count
2000000
1500000
1000000
500000
0
1
2
3
Weather Situation

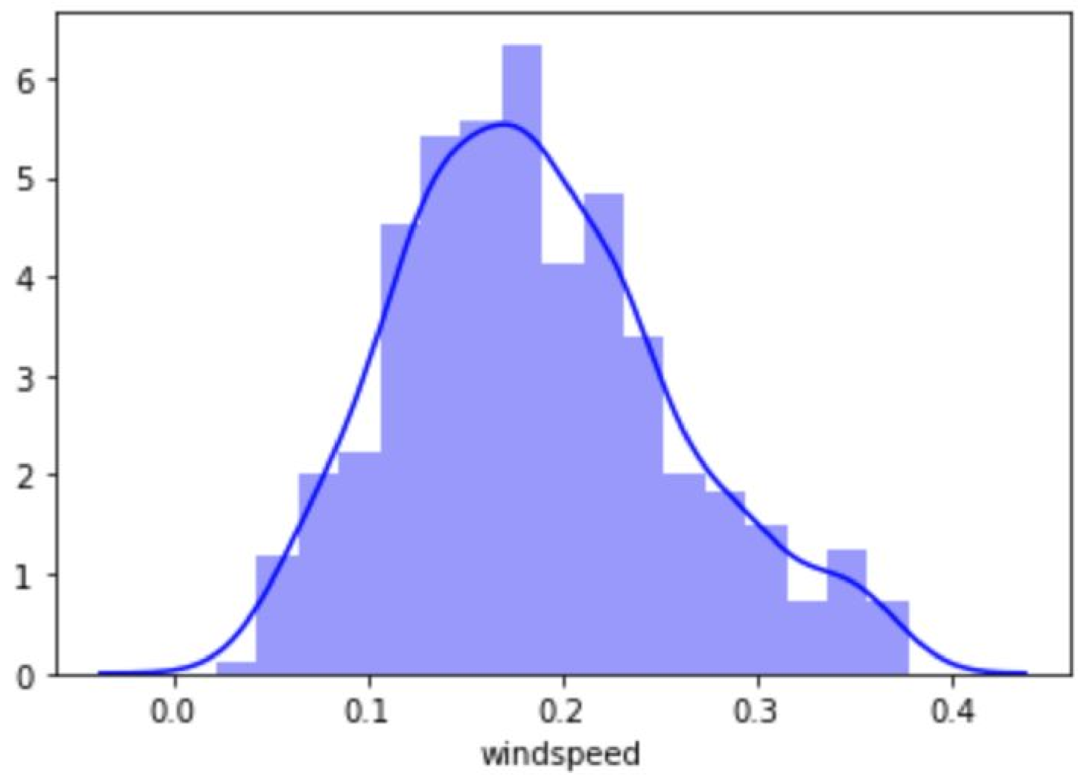

1.0
0.8
0.6
0.4
0.2
0.0
Temp
A Temp
Humidity
Windspd

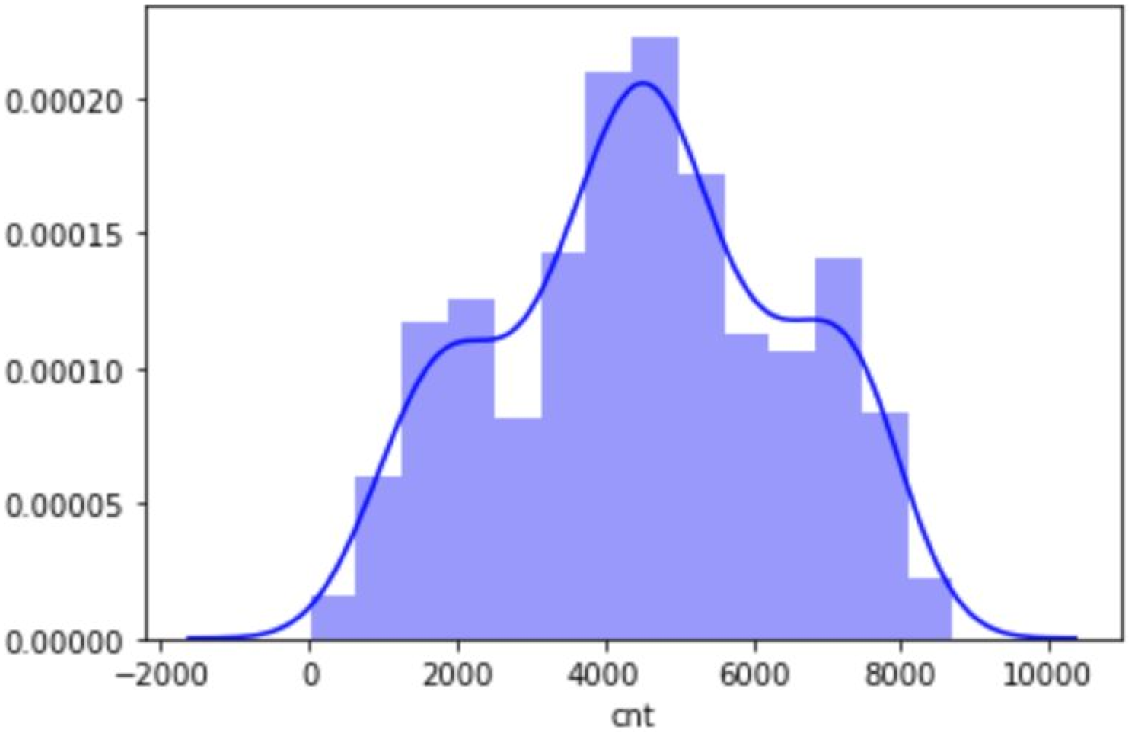

1.0
0.8
0.6
0.4
0.2
0.0
Temp
A Temp
Humidity
Windspd


Correlation Analysis
temp
atemp
hum
windspeed
cnt


2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
0.8
1.0
temp


2.5
2.0
1.5
1.0
0.5
0.0
0.2
0.4
0.6
0.8
1.0
hum


6
5
4
3
2
1
0
0.0
0.1
0.2
0.3
0.4
windspeed



R Code

```
#loading csv file
day=read.csv("day.csv",header = T)

#Missing Value analysis
sum(is.na(day))
#There is no missing value in the data(day.csv file)
#outlier Analysis
library(ggplot2)
boxplot(day$temp, day$atemp, day$hum, day$windspeed,
        names = c("Temperature", "ATemp", "Humidity", "Windspeed"),
        las = 2,
        col = c("blue","orange","red","green"),
        border = "brown",
        horizontal = FALSE,notch = FALSE
)
```

```
#There are outliers in windspeed and humidity
#outliers are saved in outlier vectors
outliers_w=boxplot(day$windspeed, plot=FALSE)$out
outliers_h=boxplot(day$hum, plot=FALSE)$out
day2=day
day2 = day2[-which(day2$windspeed %in% outliers_w),]
day2 = day2[-which(day2$hum %in% outliers_h),]
#box plot without outliers
boxplot(day2$temp, day2$atemp, day2$hum, day2$windspeed,
        names = c("Temp", "A Temp", "Humidity", "Windspd"),
        las = 2,
        col = c("blue","orange","red","green"),
        border = "brown",
        horizontal = FALSE,notch = FALSE
)
#putting NA in outlier values
day2=day
day2[,'windspeed'][day2[,'windspeed'] %in% outliers_w] = NA
day2[,'hum'][day2[,'hum'] %in% outliers_h] = NA
sum(is.na(day2))
#imputing outlier values using KNN imputation

library(DMwR)
library(rpart)
day3 = subset(day2, select = -c(instant, dteday, casual, registered))
day3 = knnImputation(day3, k = 5)
day2$windspeed=day3$windspeed
day2$hum=day3$hum
sum(is.na(day2))

#=========Feature selection=======
#correlational analysis
```

```
library(corrgram)
numeric=c('temp', 'atemp', 'hum', 'windspeed', 'cnt')
corrgram(day2[,numeric],order=FALSE,upper.panel = panel.pie,
        text.panel = panel.txt,
        main= "Correlation Analysis")
#from the pie plot it is observed temp and atemp is highly correlated
#atemp is removed as part of feature selection
day2 = subset(day2, select=-atemp)

#anova test for categorical variables
categ = c('season', 'yr', 'mnth', 'holiday', 'weekday', 'workingday',
'weathersit')
for(i in categ){
  print(i)
  print(summary(aov(formula = cnt~day2[,i],day2))
  )
}
# The p values for  workingday,weekday and holiday variables are
greater than 0.05
# The variables with p values greater than 0.05 are eliminated
day2 = subset(day2, select=-c(holiday,weekday,workingday))
#==============Feature Scaling=============
library(propagate)
numeric2=c('temp', 'hum', 'windspeed', 'cnt')
for(i in numeric2){
  print(i)
  print(skewness(day2[,i]))
}
#data is approximatey symmetric

#checking normality
hist(day2$temp)
```

```
hist(day2$hum)
hist(day2$windspeed)
hist(day2$cnt)
 # the distribution is approximately symmetric and normally distributed
scaling not recquired

#################Modeling################

####Decision tree################
# mape = 26.76604
# mae = 796.4505777
# rsquare = 0.8171842

library(rpart)
library(MASS)
#######droping few columns for decision tree#####
day2= subset(day2, select = -c(instant, dteday, casual, registered))

categ2= c("season","yr","mnth","weathersit")

library(dummies)

day3 = dummy.data.frame(day2, categ2)


train_index=sample(1:nrow(day3),0.8*nrow(day3))
train= day3[train_index,]
test= day3[-train_index,]
###rpart for regression###

model_dt = rpart(cnt ~ .,data=train,method="anova")
# test values without target variable
```

```
test2= subset(test, select=-cnt)
# prediction using decision tree
predictions_dt = predict(model_dt,test2)

#####calculate mape#####
mape = function(y,y1){mean(abs((y-y1)/y))}*100
mape(test$cnt,predictions_dt)

regr.eval(test$cnt,predictions_dt,stats = c('mape','mae') )
cor(test$cnt,predictions_dt)^2


#========Random Forest=======
# MAPE = 19.51173
# MAE = 547.4096596
# Rsquare = 0.9167251


library(randomForest)
model_rf = randomForest(cnt~., train, ntree = 500, importance =
TRUE)

# Prediction using random forest
predictions_rf = predict(model_rf, test2)

regr.eval(test$cnt,predictions_rf,stats = c('mape','mae') )
cor(test$cnt,predictions_rf)^2

#==========Linear Regression=======
# MAPE = 21.11433
# MAE = 642.7422879
# Rsquare = 0.86049
```

```
model_lr = lm(cnt~., train)
# Predictions using linear regression
predictions_lr = predict(model_lr, test2)

regr.eval(test$cnt,predictions_lr,stats = c('mape','mae') )
cor(test$cnt,predictions_lr)^2
```

Choosing Model#####

```
 # Random forest method is best suitable since MAPE,MAe is least in
Random Forest method
# x is the sample input
x=
data.frame("season"=1,"yr"=0,"mnth"=2,"weathersit"=2,"temp"=0.173,"
hum"=0.796,"windspeed"=0.1323,"cnt"=NA)
# creating dataframe using values in x for deploying in random forest
model we have created
day4=day2
bind = rbind(day4,x)
bind = dummy.data.frame(bind, categ2)
input = bind[-(1:nrow(day4)), -25]
output = predict(model_rf, input)
# output is 1439
#bar chart of season vs count
ggplot(day, aes(x = season, y = cnt))+
  labs(title = "Count of Bike Rent of season ", x = "Season", y =
"count")+
  geom_bar(stat = "identity", fill = "blue")
#bar chart of weekday vs count
ggplot(day, aes(x = weekday, y = cnt))+
  labs(title = "Count of Bike Rent of weekday ", x = "weekday", y =
"count")+
```

```
  geom_bar(stat = "identity", fill = "blue")
#bar chart of year vs count
ggplot(day, aes(x = yr, y = cnt))+
  labs(title = "Count of Bike Rent of year ", x = "Year", y = "count")+
  geom_bar(stat = "identity", fill = "blue")
#bar chart of Month vs count
ggplot(day, aes(x = mnth, y = cnt))+
  labs(title = "Count of Bike of Months ", x = "Month", y = "count")+
  geom_bar(stat = "identity", fill = "blue")
#bar chart of Weather Situation vs count
ggplot(day, aes(x = weathersit, y = cnt))+
  labs(title = "Count of Bike Rent ", x = "Weather Situation", y =
"count")+
  geom_bar(stat = "identity", fill = "blue")

hist(day$temp,main = "Frequency of Temperature",xlab =
"Temperature")
hist(day$atemp,main = "Frequency of Actual Temperature",xlab = "
Actual Temperature")
hist(day$hum,main = "Frequency of Humidity",xlab = "Humidity")
hist(day$windspeed,main = "Frequency of Windspeed",xlab = "Wind
Speed")
model_dt
summary(model_lr)
```