**Predicting Bike rental count**

*Abhay Sharma*

*15 Aug 2018*

Table of Contents

[**1** **Introduction** 3](#_Toc522491017)

[**1.1** **Problem Statement** 3](#_Toc522491018)

[**1.2** **Data Understanding** 3](#_Toc522491019)

[**2** **Methodology** 5](#_Toc522491020)

[**2.1** **Data Preprocessing** 5](#_Toc522491021)

[**2.1.1** **Missing Value Analysis** 5](#_Toc522491022)

[**2.1.2** **Outlier Analysis** 6](#_Toc522491023)

[**2.2** **Data Visualization** 9](#_Toc522491024)

[**2.3** **Feature Selection** 16](#_Toc522491025)

[**3** **Modelling** 18](#_Toc522491026)

[**3.1** **Linear Regression** 18](#_Toc522491027)

[**3.2** **Decision Tree Regression** 20](#_Toc522491028)

[**4** **Conclusion** 21](#_Toc522491029)

[**4.1** **Model Evaluation** 21](#_Toc522491030)

[**4.2** **Model Selection** 21](#_Toc522491031)

[**5** **Codes** 22](#_Toc522491032)

[**5.1** **R Code** 22](#_Toc522491033)

[**5.2** **Python Code** 26](#_Toc522491034)

# **Introduction**

## **Problem Statement**

The objective of this Case is to Predication of bike rental count on daily based on the

environmental and seasonal settings. Here we have to predict the count of bike rental on daily basis, we have other variables to predict the bike renting on daily basis and it is a regression task there can be many factors which will be used for this i.e., environment, weather and season.

## **Data Understanding**

Our tasks is to build regression model which will predict further target variable values which is Bike Count. Given below are the some.

Table 1.1 (Bike Renting Data)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| instant | dteday | season | yr | mnth | holiday | weekday | workingday |
| 1 | 1/1/2011 | 1 | 0 | 1 | 0 | 6 | 0 |
| 2 | 1/2/2011 | 1 | 0 | 1 | 0 | 0 | 0 |
| 3 | 1/3/2011 | 1 | 0 | 1 | 0 | 1 | 1 |
| 4 | 1/4/2011 | 1 | 0 | 1 | 0 | 2 | 1 |

Table 2.2 (Bike Renting Data)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| weathersit | temp | atemp | hum | windspeed | casual | registered | cnt |
| 2 | 0.344167 | 0.363625 | 0.805833 | 0.160446 | 331 | 654 | 985 |
| 2 | 0.363478 | 0.353739 | 0.696087 | 0.248539 | 131 | 670 | 801 |
| 1 | 0.196364 | 0.189405 | 0.437273 | 0.248309 | 120 | 1229 | 1349 |
| 1 | 0.2 | 0.212122 | 0.590435 | 0.160296 | 108 | 1454 | 1562 |

Lets see what variables tells:-

instant: Record index

dteday: Date

season: Season (1:spring, 2:summer, 3:fall, 4:winter)

yr: Year (0: 2011, 1:2012)

mnth: Month (1 to 12)

hr: Hour (0 to 23)

holiday: weather day is holiday or not (extracted from Holiday Schedule)

weekday: Day of the week

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

weathersit: (extracted from Freemeteo)

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

temp: Normalized temperature in Celsius. The values are derived via

(t-t\_min)/(t\_max-t\_min),

t\_min=-8, t\_max=+39 (only in hourly scale)

atemp: Normalized feeling temperature in Celsius. The values are derived via

(t-t\_min)/(t\_maxt\_

min), t\_min=-16, t\_max=+50 (only in hourly scale)

hum: Normalized humidity. The values are divided to 100 (max)

windspeed: Normalized wind speed. The values are divided to 67 (max)

casual: count of casual users

registered: count of registered users

cnt: count of total rental bikes including both casual and registered

From these data we found out that out of 16 variables there is one target variable ‘cnt’.

Rest all are predictor variables except some variables which may be drop in further data- preprocessing.



# **Methodology**

## **Data Preprocessing**

Data in real world is dirty it of no use until unless we apply data preprocessing on it. In other

words, Pre- processing refers to the transformations applied to your data before feeding

it to the algorithm. It’s a data mining technique which that involves transforming raw data into an understandable format or we can say that it prepares raw data to further processing. There are so many things that we do in data preprocessing like data cleaning, data integration, data transformation, or data reduction.

### **Missing Value Analysis**

Missing Values Analysis is use to fill NULL values in data with some imputation techniques

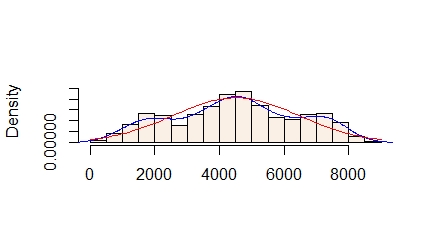
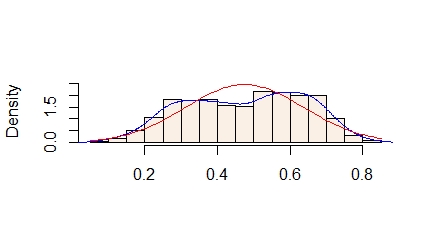
But here in our Bike Renting Data we don’t have any null Value. By the way Our data doesn’t contain missing value. Our Data is fit.



### **Outlier Analysis**

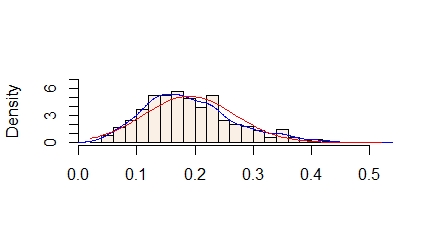
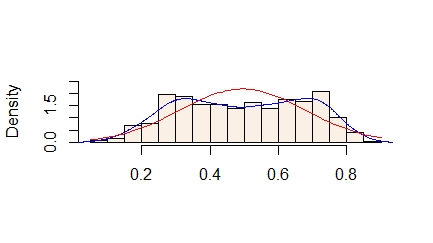
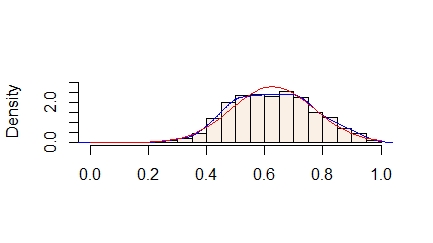
Outlier are some values which cause deviation in central tendencies or they usually show skewness in histogram or outlier value in boxplot. We cannot see anything in figure 2.1

Before Drawing Histogram we have to break ‘dteday’ column to new ‘date’ column.

Then we will also plot histogram for date also to if there any examine any extra value. 

cnt

Atemp



Windspeed

Temp

hum

Figure 1: Probability Density function on Bike Renting Data



We can see that there is some skewness in *hum* and *windspeed* histogram, Let’s build their Boxplot.

We can see that there are some outliers in Data of *hum* and *windspeed* .We will remove those values as it is normalized data and for better accuracy.

Code:



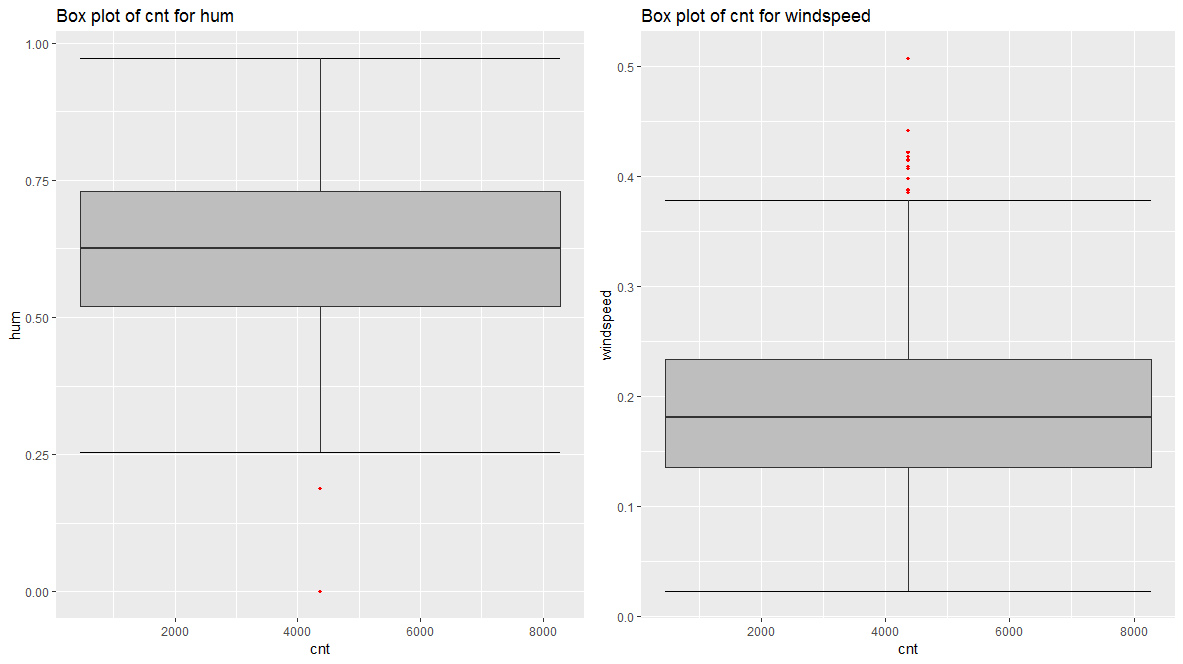


Figure 2: Box Plot of hum and windspeed (See R Code in Appendix)

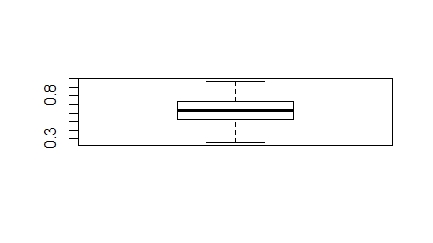
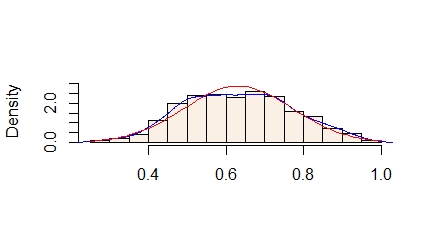


Figure 3: *hum* after Outlier Analysis

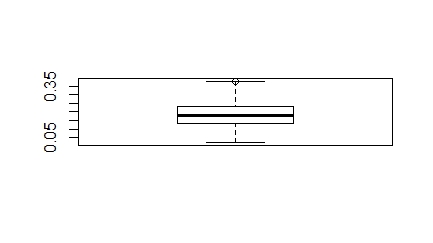
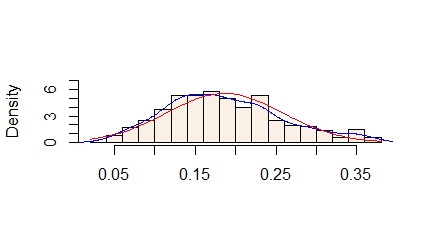


Figure 4: *windspeed* after outlier analysis

We used *box.stats* method to identify and replace outliers in *hum* and *windspeed* with mean.

Also its normalized data there is not too much to do with outliers.

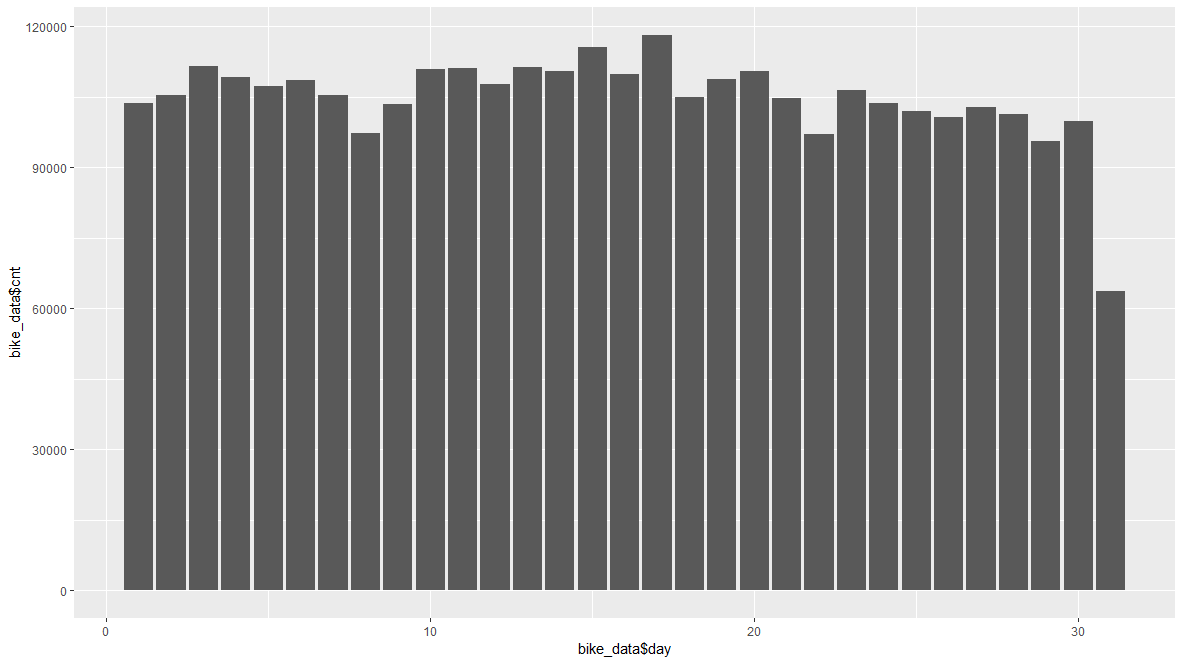


Outliers is the important steps of data preprocessing we have to perform them to make values accurate.

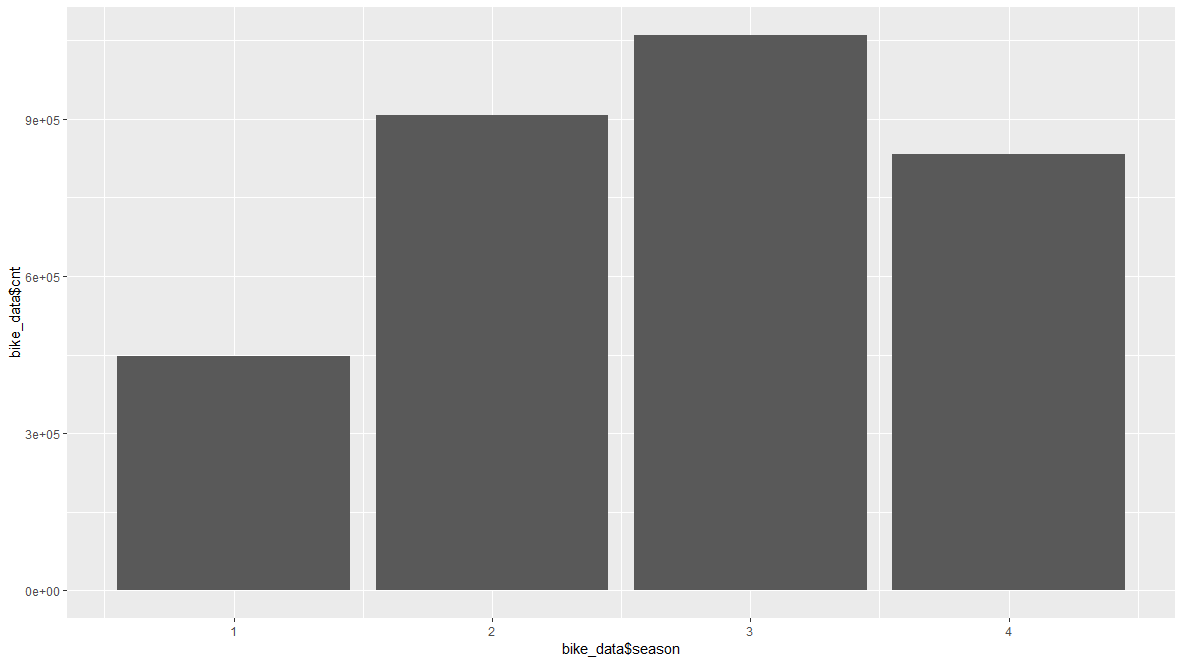
## **Data Visualization**

Data Visualization is important concept it will help us to understand data , and will tell us answer of various questions also it will show relation between variables.

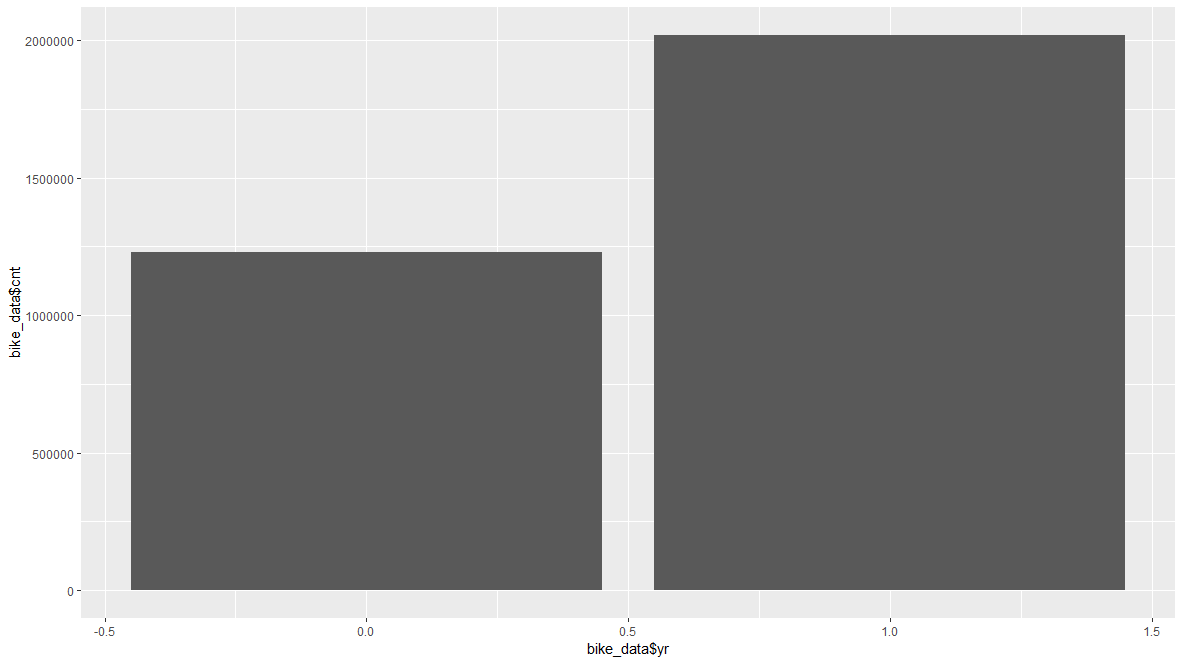
* Relation between count and days, it showing peoples are renting bike less in last days of month, or this graph cannot give us fruitful result. There is no satisfying result.



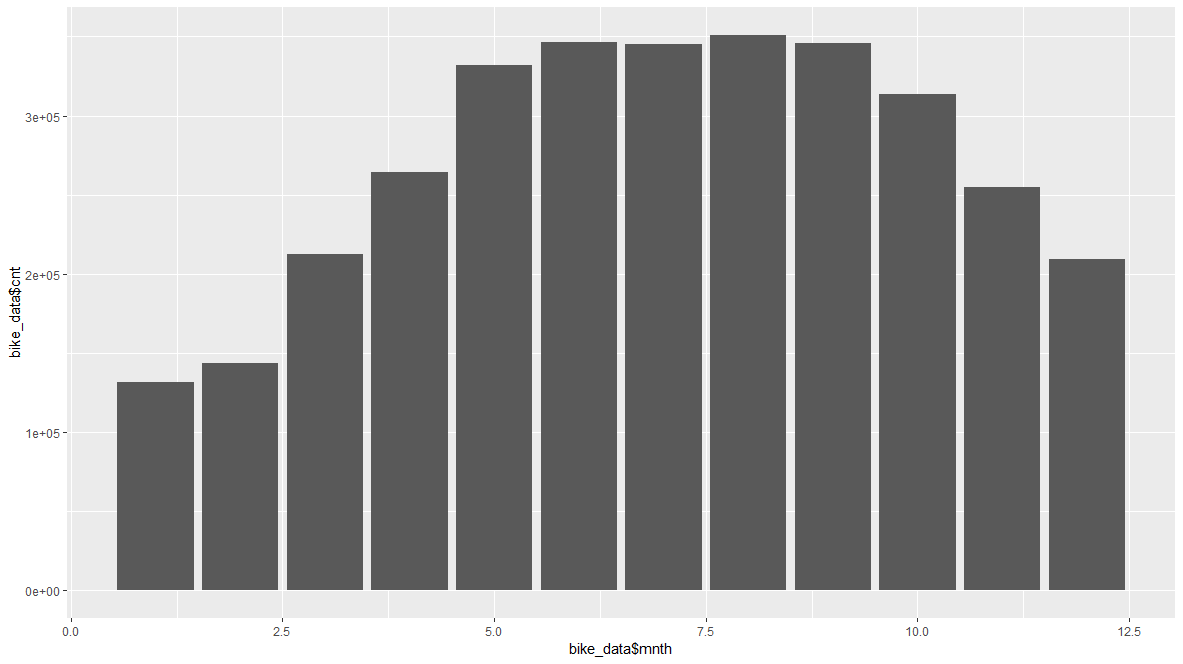
* See most of the peoples rent bike on fall(3) rather than spring(1), These are useful results and this would be important to modelling and prediction values.



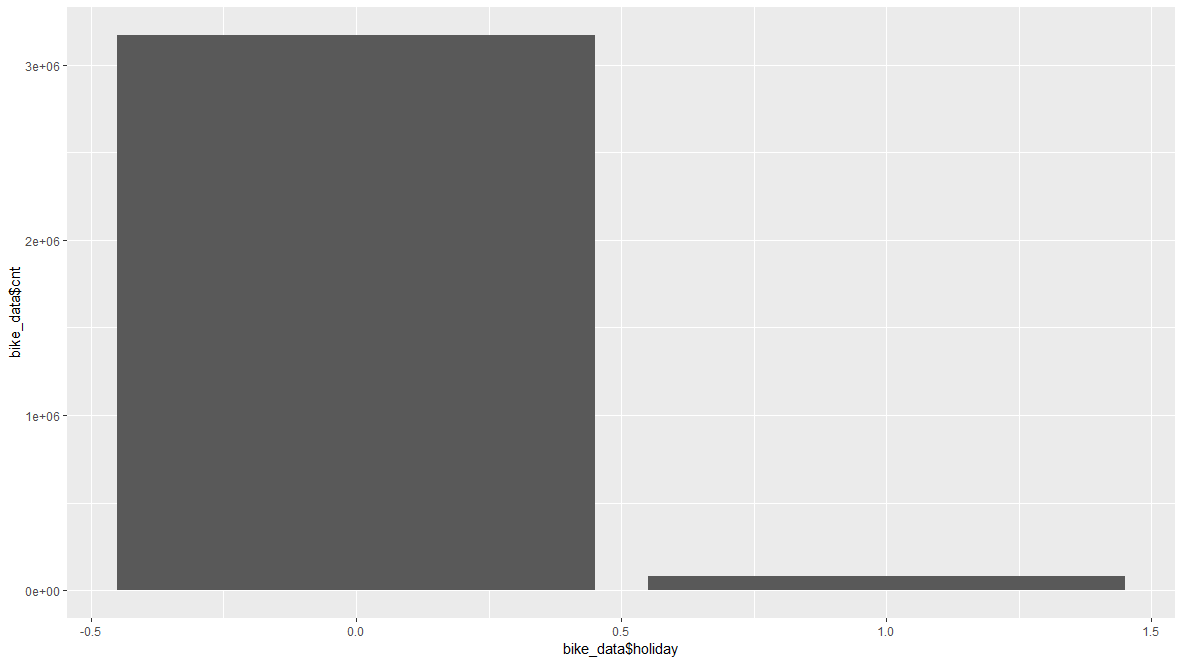
* Relation with Bike Rent and year, Most people hire bike in 2012 year rather than 2011 year, we can assume that season or environment may be more good in 2012 than 2011 or we can say that it is usual growth by increasing consumers.



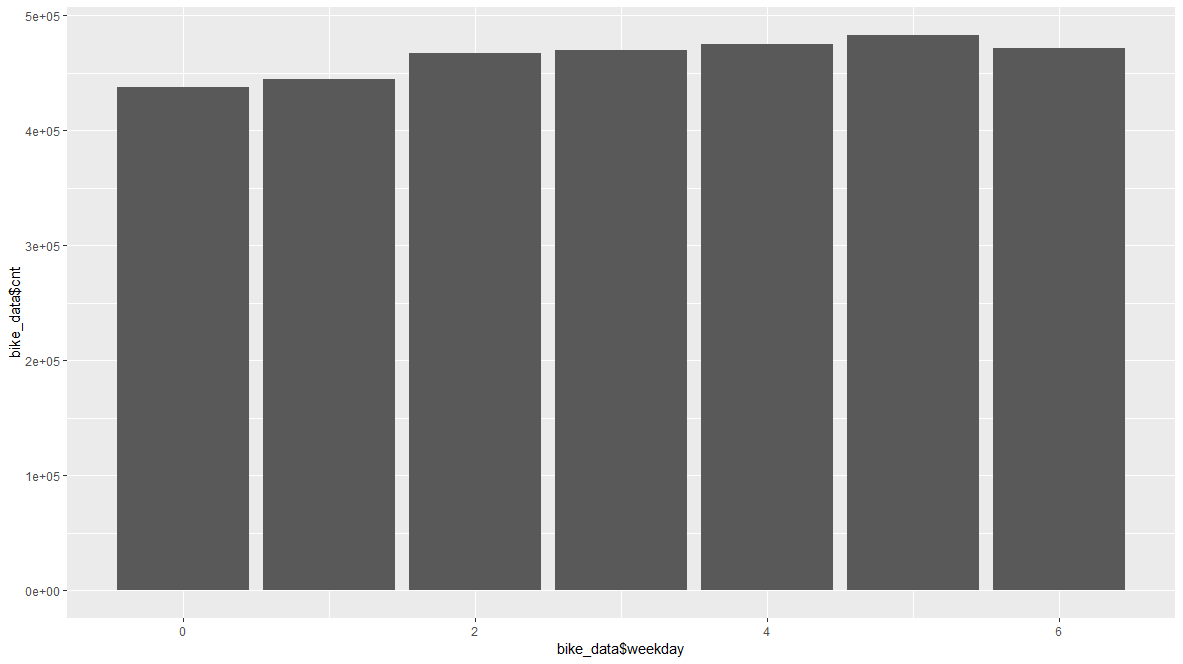
* We can see season summer(2) and fall(3) directly related to most of months with largest Bike rent 5 to 9 months. This is a valid relation between data and can be used in modelling.



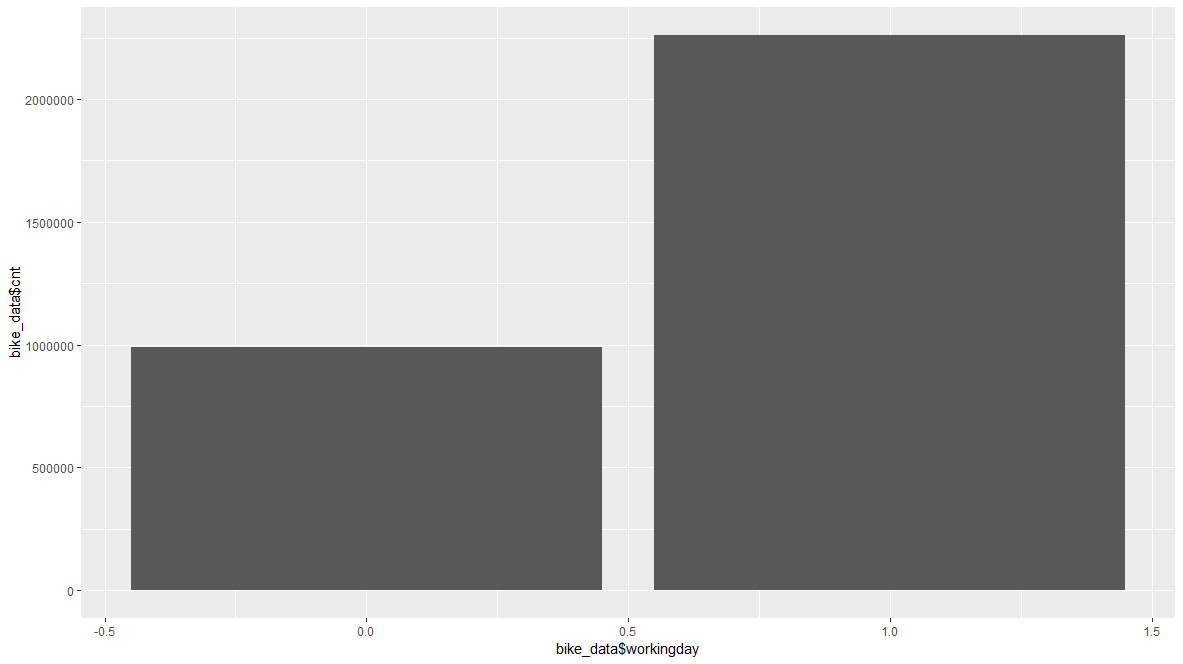
* No one rent bike on holiday, only 1 % of cases are there when on holiday people hired bike. These values are important for modelling and we can also say that some of these values are outliers but its useful.



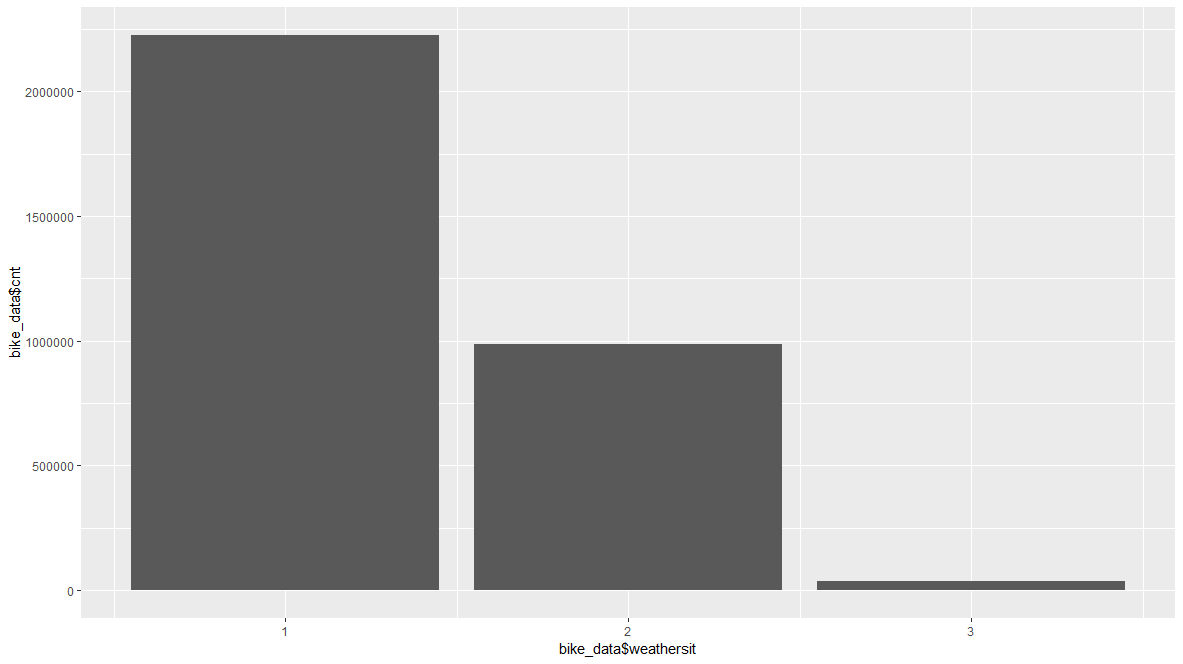
* Relation between weekdays and count, not so few rents on Sunday. This is common result and be used further not so impactfully.



* Working Day and count, bike are more hired on working day. Peoples less hired bike in holiday , most of them would be on long ride after renting.



* No one hired bike on heavy rain or snow fall weather and less for light rain and snow fall, Peoples hired bike on clear weather too much. This variable I useful too much for predictive model.



Code:

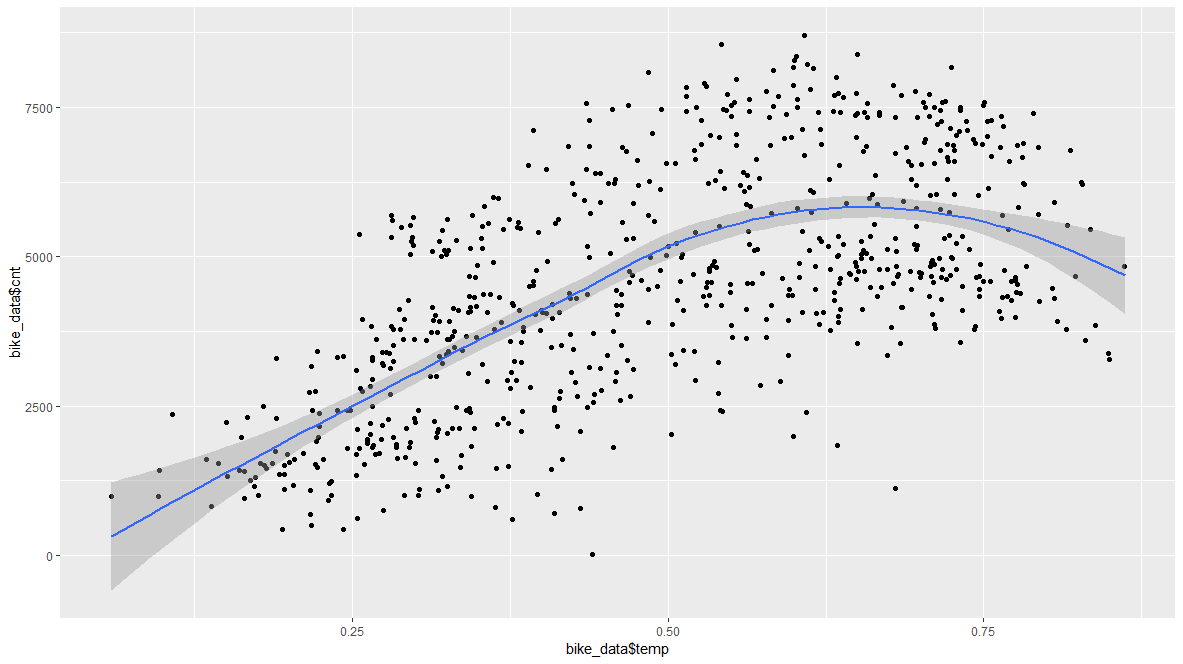


Some More visualization with continuous variable and cnt

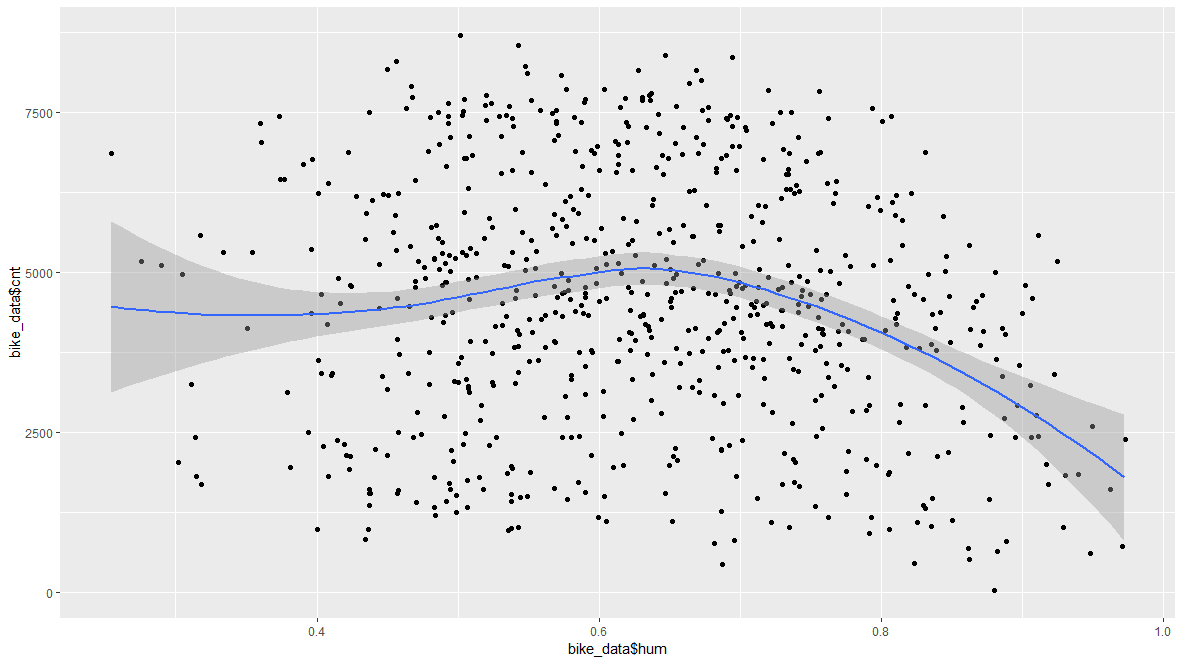
Code:



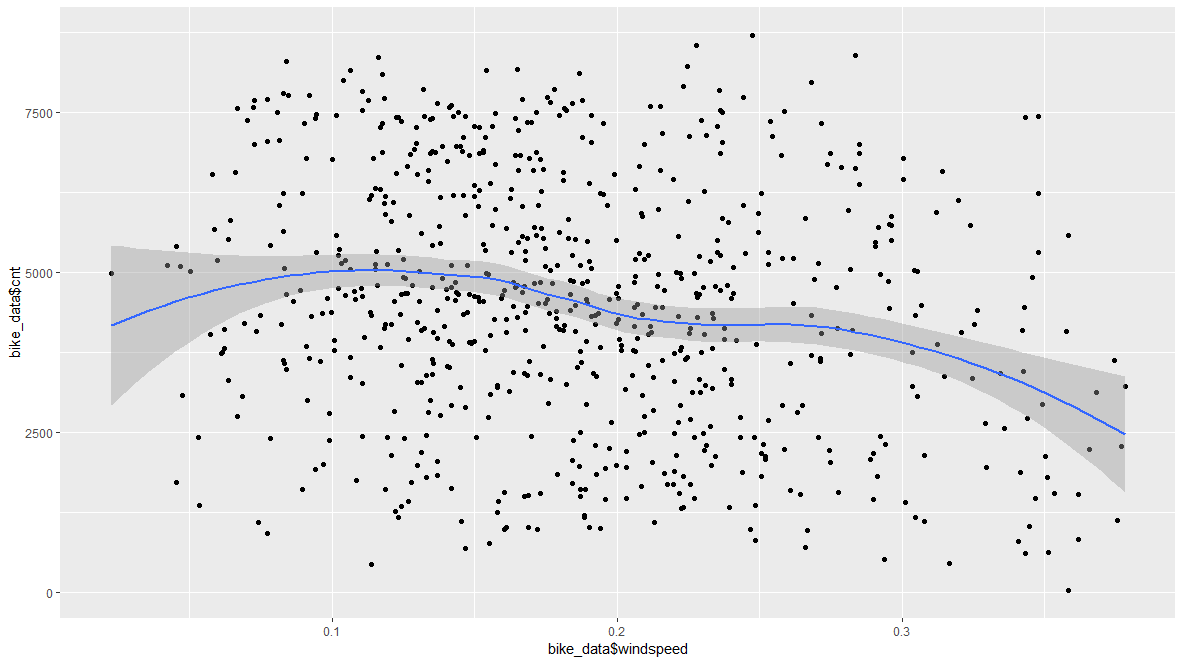
* Relation between count and cnt, its showing parabolic lines peoples don’t rent bike on high temperature days. Graph line increase but reached at a peak point where renting rate is high and after increase in temp also decreased in cnt.



* Humidity is directly proportional to count sometimes but little bit when there in excess humidity peoples seems to less interested in renting bike.



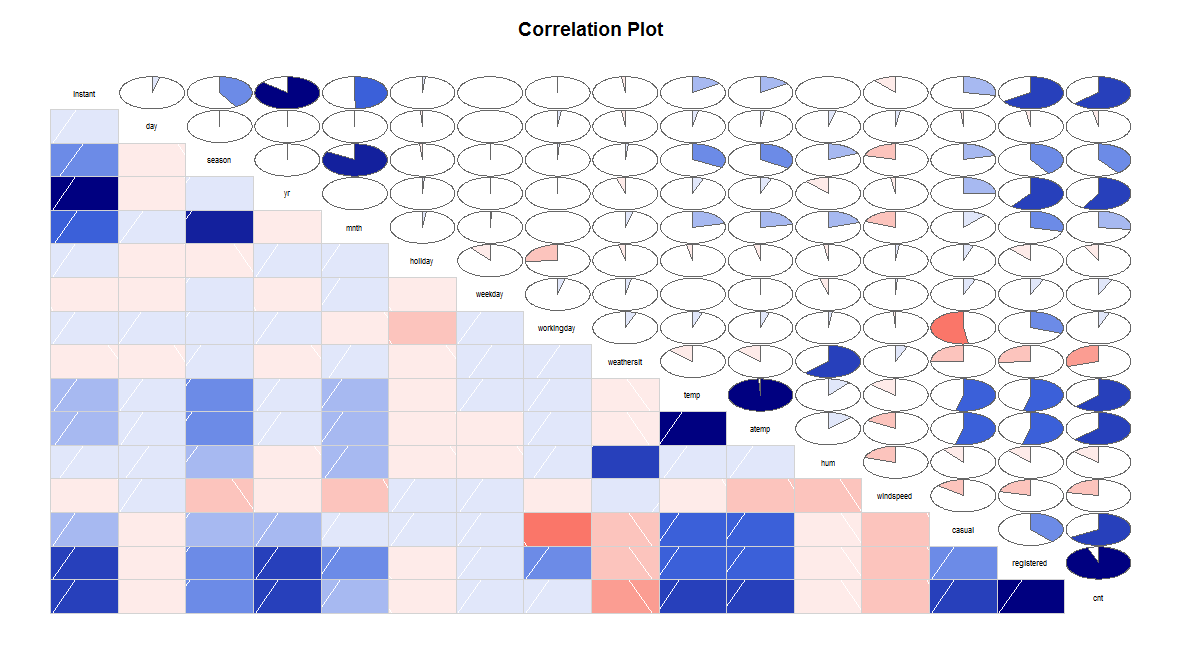
* Windspeed also directly proportional to count , graph is same as humidity .



## **Feature Selection**

Before performing any type of modeling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class prediction. There are several methods of doing that. Below will build correlation graph and see vif.

* Correlation Grapgh



We can see that cnt is highly correlated with casual and registered and temp and atemp are also highly correlated. Lets do another test.

Here we will drop *atemp* it is highly correlated and *instant* as it is nothing just as a index.

With that we will also drop *day* variable it is also no fruitful to our data because of no relation.

It will decrease features of our model due to high so many levels.

We will also drop two most highly correlated variables which are *casual* and *registered*.

If we wouldn’t remove those variable , they will show same *cnt* variable in prediction values.



By above result we need to remove *casual*, *registered* and *atemp* variable as they are highly correlation, hence will give no good prediction model. Instance column is also of no use.



Its look like our data is fit, move to modelling now.

# **Modelling**

As our problem statement suggest that we have to do prediction by using our useful predictor variable.

Our problem is regression problem, we have to predict count of daily rent of bike.

For regression we have to use linear regression or decision tree regression algorithms.

## **Linear Regression**

Linear Regression is a regression algorithm (but you probably figured that out). This algorithm’s principle is to find a linear relation within your data (see below). Once the linear relation is found, predicting a new value is done with respect to this relation.

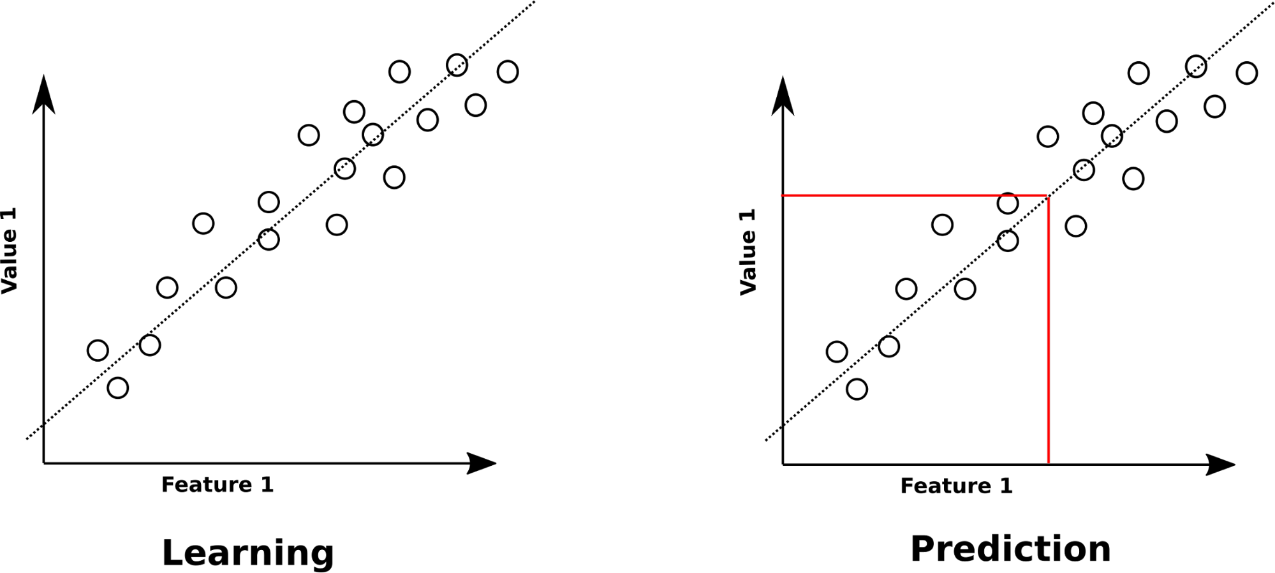
**Main advantages :**

• very simple algorithm

• doesn’t take a lot of memory

• quite fast

• easy to explain

[](https://recast.ai/blog/wp-content/uploads/2017/02/image12.png)

**Main drawbacks:**

• requires the data to be linearly spread (see « Polynomial Regression » if you think you need a polynomial fitting)

• is unstable in case features are redundant, i.e. if there is multicollinearity (note that, in that case you should have a look to « Elastic-Net or « Ridge-Regression »).

As we will create model first look for the code below.



We got R-squared value around **80%** approx.

Now, We will Check for accuracy of model below.



Here our percentage is **17.57%**, it means our accuracy is **82.43%.** We got decent accuracy.

Our values are stored in ‘predicions\_LR’ .

## **Decision Tree Regression**

The Decision Tree algorithm is a classification and regression algorithm. It subdivides learning data into regions having similar features. Descending the tree allows the prediction of the class or value of the new input data point.

**Main advantages:**

• quite simple

• easy to communicate about

• easy to maintain

• few parameters are required and they are quite intuitive

• prediction is quite fast

**Main drawbacks:**

• can take a lot of memory (the more features you have, the deeper and larger your decision tree is likely to be)

• Naturally overfits a lot (it generates high-variance models, it suffers less from that if the branches are pruned, though)

• Not capable of being incrementally improved

Let’s build our decision Tree



Above we can see our decision tree, how it divides data within range. Now we move to predictions.



We are getting **18.71%** of error in our prediction values or we are getting **81.39%** of accuracy in our model.

Let’s store these value in csv file.

# **Conclusion**

## **Model Evaluation**

Our model was of regression model, we used only two machine learning algorithm.

Which is Linear and decision tree regression.

* Linear Regression:

In Linear Regression we got **83% Accuracy** and **17% Error** by mean absolute percentage error.

* Decision Tree:

In Decision Tree, we got **18.71%** of error in our prediction values or we are getting **81.39%** of accuracy in our model.

## **Model Selection**

By this we can see that we can select either of two models they are almost same in means of accuracy and fits to our model.

# **Codes**

## **R Code**

# Remove Lists

rm(list = ls())

# Set Working Directory

setwd("D:/Bike Renting")

# Get Working Directory

getwd()

# Load Data

bike\_data <- read.csv("day.csv", header = T)

# Load Library

library(tidyr)

library(plyr)

library(psych)

library(ggplot2)

library(rpart)

library(usdm)

library(rpart)

library(MASS)

library(MLmetrics)

library(DMwR)

# Explore Data

str(bike\_data)

# Missing Value Analysis

sum(is.na(bike\_data))

#Multi Histogram for finding out Outlier

multi.hist(bike\_data$temp, main = NA, dcol = c("blue", "red"),

dlty = c("solid", "solid"), bcol = "linen")

multi.hist(bike\_data$atemp, main = NA, dcol = c("blue", "red"),

dlty = c("solid", "solid"), bcol = "linen")

multi.hist(bike\_data$hum, main = NA, dcol = c("blue", "red"), # Outlier

dlty = c("solid", "solid"), bcol = "linen")

multi.hist(bike\_data$windspeed, main = NA, dcol = c("blue", "red"), # Outlier

dlty = c("solid", "solid"), bcol = "linen")

multi.hist(bike\_data$cnt, main = NA, dcol = c("blue", "red"),

dlty = c("solid", "solid"), bcol = "linen")

# We can see some skewness in 'hum' and 'windspeed'

#

# # Split 'dteday' column into new Day column

bike\_data <- separate(data = bike\_data,col = "dteday",into = c("year","month","day"))

bike\_data$year <- NULL

bike\_data$month <- NULL

## Convert day to int from

bike\_data$day <- as.numeric(as.character(bike\_data$day))

#

# Plot BoxPlot For them and lets go with Outlier Analysis

numeric\_index = sapply(bike\_data,is.numeric)

numeric\_data = bike\_data[,numeric\_index]

cnames = colnames(numeric\_data)

for (i in 1:length(cnames))

{

assign(paste0("gn",i), ggplot(aes\_string(y = (cnames[i]), x = "cnt"), data = subset(bike\_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=cnames[i],x="cnt")+

ggtitle(paste("Box plot of cnt for",cnames[i])))

}

#gridExtra::grid.arrange(gn10,gn11,gn12,ncol=3)

#gridExtra::grid.arrange(gn13,gn14,gn16,ncol=3)

gridExtra::grid.arrange(gn11,gn12,ncol=2)

# hum and windspeed have outliers, Lets remove those values

outlier\_val <- c('hum' , 'windspeed')

#

for(i in outlier\_val){

val = bike\_data[,i][bike\_data[,i] %in% boxplot.stats(bike\_data[,i])$out]

#print(length(val))

bike\_data[,i][bike\_data[,i] %in% val] = NA

}

#

bike\_data$hum[is.na(bike\_data$hum)] = mean(bike\_data$hum, na.rm = T)

bike\_data$windspeed[is.na(bike\_data$windspeed)] = mean(bike\_data$windspeed, na.rm = T)

#

##

boxplot(bike\_data$hum)

multi.hist(bike\_data$hum, main = NA, dcol = c("blue", "red"),

dlty = c("solid", "solid"), bcol = "linen")

boxplot(bike\_data$windspeed)

multi.hist(bike\_data$windspeed, main = NA, dcol = c("blue", "red"),

dlty = c("solid", "solid"), bcol = "linen")

#Visualization

##Int value

ggplot(bike\_data, aes(x = bike\_data$day, y = bike\_data$cnt )) + geom\_col(show.legend = TRUE)

ggplot(bike\_data, aes(x = bike\_data$season, y = bike\_data$cnt )) + geom\_col(show.legend = TRUE)

ggplot(bike\_data, aes(x = bike\_data$yr, y = bike\_data$cnt )) + geom\_col(show.legend = TRUE )

ggplot(bike\_data, aes(x = bike\_data$mnth, y = bike\_data$cnt )) + geom\_col(show.legend = TRUE )

ggplot(bike\_data, aes(x = bike\_data$holiday, y = bike\_data$cnt )) + geom\_col(show.legend = TRUE )

ggplot(bike\_data, aes(x = bike\_data$weekday, y = bike\_data$cnt )) + geom\_col(show.legend = TRUE )

ggplot(bike\_data, aes(x = bike\_data$workingday, y = bike\_data$cnt )) + geom\_col(show.legend = TRUE )

ggplot(bike\_data, aes(x = bike\_data$weathersit, y = bike\_data$cnt )) + geom\_col(show.legend = TRUE )

##Numeric Value

ggplot(bike\_data, aes(x = bike\_data$temp, y = bike\_data$cnt )) + geom\_jitter() + geom\_smooth(method = loess, formula = y ~ x)

ggplot(bike\_data, aes(x = bike\_data$atemp, y = bike\_data$cnt )) + geom\_jitter() + geom\_smooth(method = loess, formula = y ~ x)

ggplot(bike\_data, aes(x = bike\_data$hum, y = bike\_data$cnt )) + geom\_jitter() + geom\_smooth(method = loess, formula = y ~ x)

ggplot(bike\_data, aes(x = bike\_data$windspeed, y = bike\_data$cnt )) + geom\_jitter() + geom\_smooth(method = loess, formula = y ~ x)

# Correlation graph

library(corrgram)

corrgram(bike\_data[,numeric\_index], order = F,

upper.panel=panel.pie, text.panel=panel.txt, main = "Correlation Plot")

vifcor(bike\_data[,-16], th = 0.9)

# Lets make towards Model

# Drop some nouse variable

#df <- bike\_data[, -c(1,11,14:15)]

df <- bike\_data[, -c(1:2,11,14:15)]

vifcor(df[-11], th = 0.9)

vif(df)

# Linear Regression

lm\_model = lm(cnt ~., data = df)

summary(lm\_model)

# p 0.05 Accept NULL Hypothesis and can say that this variable not relevant to us

predictions\_LR = predict(lm\_model, df[,1:10])

library(MLmetrics)

MAPE(df[,11], predictions\_LR)

# 83%

write.csv(predictions\_LR, file = 'D:/Bike Renting/countpredictLR.csv')

# Decision Tree Regression

n = nrow(df)

trainIndex = sample(1:n, size = round(0.8\*n), replace=FALSE)

train = df[trainIndex ,]

test = df[-trainIndex ,]

# r part for decision tree regreession

fit = rpart(cnt ~ ., data = train, method = 'anova')

# predictions = predict(fit, test[,-12])

predictions\_TR = predict(fit, test[,-11])

library(MLmetrics)

MAPE(test[,11], predictions\_TR)

# 81%

write.csv(predictions\_TR, file = 'D:/Bike Renting/countpredictTR.csv')

## **Python Code**

import os

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

from sklearn.cross\_validation import train\_test\_split

import statsmodels.api as sm

#os.chdir("D:\bike")

path = r"D:\bike"

#Load Data

bike\_data = pd.read\_csv("day.csv")

bike\_data.head(5)

# Check Missing Values

pd.DataFrame(bike\_data.isnull().sum())

# Data summary

bike\_data.describe()

# Data Info

bike\_data.info()

**Data Preprocessing**

# BoxPlot

sns.boxplot(data=bike\_data[['temp',

'atemp', 'hum', 'windspeed']])

fig=plt.gcf()

#Boxplot

sns.boxplot(data=bike\_data[['cnt',

'casual']])

# Set for outliers

outlier\_val = ['hum', 'windspeed']

cnames = ['instant','dteday','season','yr','mnth', 'holiday','weekday','workingday','weathersit','temp'

'atemp','hum','windspeed','casual','registered','cnt']

#Detect and delete outliers from data

for i in outlier\_val:

print(i)

q75, q25 = np.percentile(bike\_data.loc[:,i], [75 ,25])

iqr = q75 - q25

min = q25 - (iqr\*1.5)

max = q75 + (iqr\*1.5)

print(min)

print(max)

bike\_data = bike\_data.drop(bike\_data[bike\_data.loc[:,i] < min].index)

bike\_data = bike\_data.drop(bike\_data[bike\_data.loc[:,i] > max].index)

**Data Visualization**

#Bar Graph

sns.factorplot(x="season",y="cnt",data=bike\_data,kind='bar',size=5,aspect=1.5)

sns.factorplot(x="yr",y="cnt",data=bike\_data,kind='bar',size=5,aspect=1.5) sns.factorplot(x="mnth",y="cnt",data=bike\_data,kind='bar',size=5,aspect=1.5) sns.factorplot(x="holiday",y="cnt",data=bike\_data,kind='bar',size=5,aspect=1.5) sns.factorplot(x="weekday",y="cnt",data=bike\_data,kind='bar',size=5,aspect=1.5) sns.factorplot(x="workingday",y="cnt",data=bike\_data,kind='bar',size=5,aspect=1.5) sns.factorplot(x="weathersit",y="cnt",data=bike\_data,kind='bar',size=5,aspect=1.5)

# Scatter Plot

plt.scatter(x="temp",y="cnt",data=bike\_data,color='#ff4125')

plt.scatter(x="atemp",y="cnt",data=bike\_data,color='#ff4125') plt.scatter(x="hum",y="cnt",data=bike\_data,color='#ff4125') plt.scatter(x="windspeed",y="cnt",data=bike\_data,color='#ff4125')

**Feature Engineering**

#hist of cnnt

%matplotlib inline

plt.hist(bike\_data['cnt'], bins='auto')

# Correlation Graph

df\_corr = bike\_data.loc[:,cnames]

#Set the width and hieght of the plot

f, ax = plt.subplots(figsize=(7, 5))

#Generate correlation matrix

corr = df\_corr.corr()

#Plot using seaborn library

sns.heatmap(corr, mask=np.zeros\_like(corr, dtype=np.bool), cmap=sns.diverging\_palette(220, 10, as\_cmap=True),

square=True, ax=ax)

# Drop use less variable which can be used to predict

bike\_data = bike\_data.drop(['instant','dteday','atemp','casual','registered'], axis=1)

**Model Development**

# Linear Regresssion

train, test = train\_test\_split(bike\_data, test\_size=0.2)

model = sm.OLS(bike\_data.iloc[:,10], bike\_data.iloc[:,0:10]).fit()

model.summary()

predictions\_LR = model.predict(test.iloc[:,0:10])

def MAPE(y\_true, y\_pred):

mape = np.mean(np.abs((y\_true - y\_pred) / y\_true))\*100

return mape

#Calculate MAPE

MAPE(test.iloc[:,10], predictions\_LR)

19.694792307271566

# Here we are getting 81% accuracy

# Decision Tree

from sklearn.tree import DecisionTreeRegressor

dtrain, dtest = train\_test\_split(bike\_data, test\_size=0.2)

fit\_DT = DecisionTreeRegressor(max\_depth=2).fit(bike\_data.iloc[:,0:10], bike\_data.iloc[:,10])

fit\_DT

def MAPE(y\_true, y\_pred):

mape = np.mean(np.abs((y\_true - y\_pred) / y\_true))\*100

return mape

MAPE(dtest.iloc[:,10], predictions\_DT)

# Store LR prediction value

***Because In python we are not getting good accuracy of decision tree , as in Linear Regression ,But linear regression always preferred in industry for regression problem***

predictions\_LR.to\_csv('predictions\_LR.csv', sep=',', encoding='utf-8', index= False)

