**Bike Renting Count Prediction**Name : Nishkarsh Bansal

**Contents :**

**Topics Name Page No.**

**Chapter 1: Introduction …………………………………………………………………3** 1.1 Problem Statement…………………………………………………………………………….3  
 1.2 Overview of the statement………………………………………………………………..3

**Chapter 2: Visualization ………………………………………………………………..4** 2.1 Visualizing and Interpreting the raw data ………………………………………….5  
 2.2 Summary of visualizing the raw data …………………………………………………8

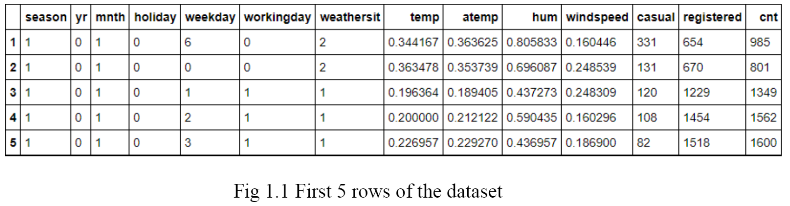
**Chapter 3: Outlier Analysis…………………………………………………………….9** 3.1 Outlier Analysis of the raw data …………………………………………………**……..9**

**Chapter 4: Feature Selection ………………………………………………………….12** 4.1 Correlation Analysis…………………………………………………………………………….12  
 4.2 ANOVA test………………………………………………………………………………………….13  
 4.3 VIF test **…………………………………………………………………………………………….**13 4.4Summary of the feature selection………………………………………………………..14

**Chapter 5: Model Development and Conclusion……………………………16** 5.1 Linear Regression Model…………………………………………………………………**….**165.2 Decision Tree………………………………………………………………………………………17  
 5.3 Random Forest Model…………………………………………………………………………18  
 5.4 Conclusion……………………………………………………………………………………………19

**Chapter 1: Introduction**

* 1. Problem Statement

The objective of this case is to predict the number of bikes rent count based on the environment and seasonal setting i.e. the total number of people renting the bike on the daily basis.  
  
  
1.2 Overview of the dataset 

As we can see , there are seven categorical variables and 7 numeric variables .The dependent variable or the target variable is cnt, i.e. the count of people renting the bikes.  
  
While checking, it is seen that there are no duplicate rows or empty values present throughout the dataset. Also the numerical variables are scaled by normalizing the data ,so only two variable we need to normalize and also no need of imputing missing values as no missing values present.

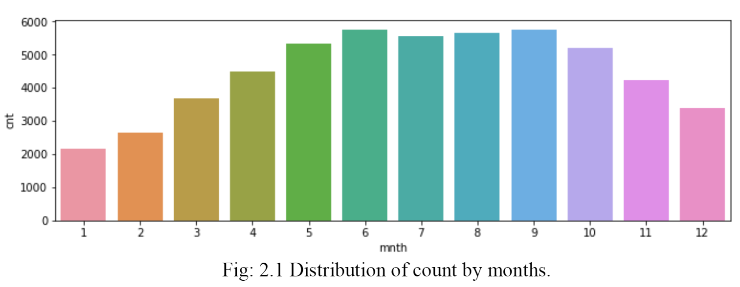
The details of data attributes in the dataset are as follows

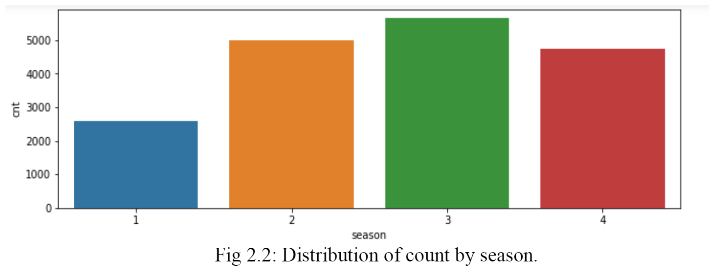
seasons: Seasons(1: spring , 2: summer, 3:fall, 4:winter)  
yr: Year(0:2011, 1:2012)  
mnth: Month ( 1 to 12) (Jan to December)

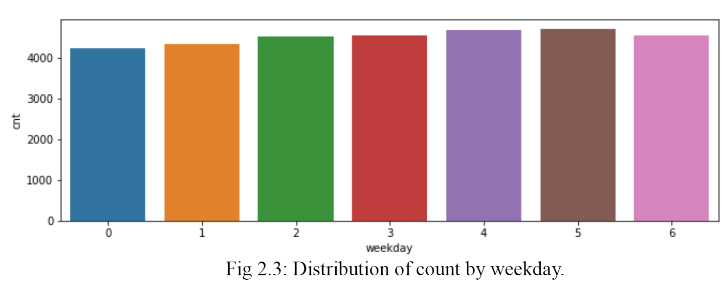
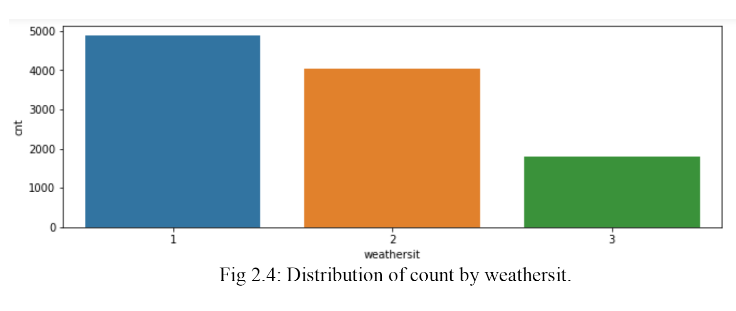
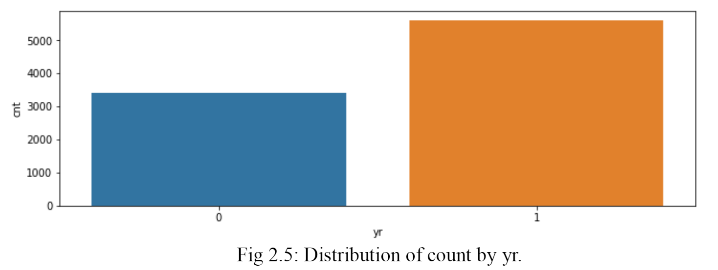
holiday: whether day is holiday or not(extracted from Holiday Schedule)  
weekday : Day of the week.  
workingday :If day is neither nor holiday is 1, otherwise is 0  
weathersit : (extracte from freemeteo)  
1.Clear,Few clouds, Partly cloudy  
2.Mist+Cloudy, Mist +Broken clouds , Mist +Few clouds, Mist.  
3.Light snow/Heavy snow, Heavy rain + thunderstorm + scattered clouds , light rain + scattered clouds.  
temp : Normalized temperature in Celsius  
the values are derived via (t-tmin)/tmax-tmin).  
atemp : Normalized feeling temperature in celcius  
hum: Normalized Humidity.The values are divided by 100.  
windspeed :Normalized wind speed.The values are divided to 67.   
casual: count of casual users.  
registered : count of registered users.  
count : count of tota rent bikes including both casual and registered.

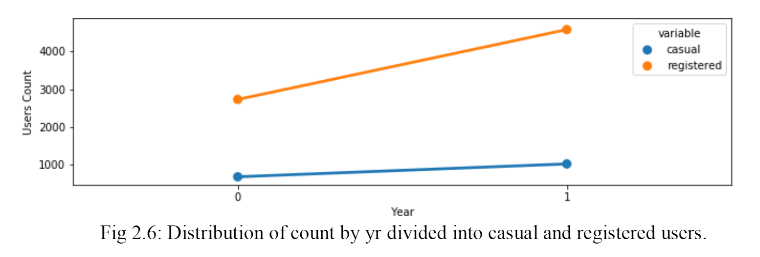
**Chapter 2 : Visualization**

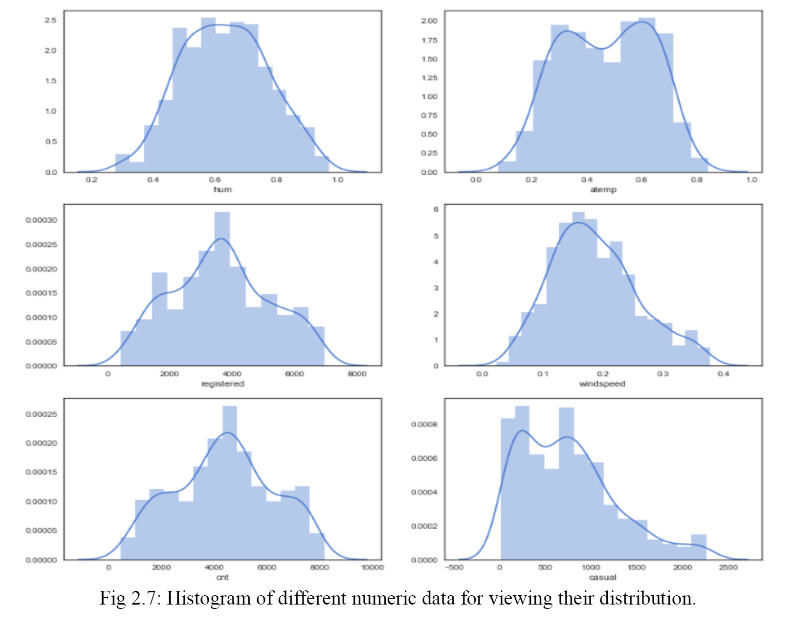
2.1 Visualizing the raw data

To understand what the raw data is, what it looks like and to understand the parameters , we need to look at various graph plots.







2.2 Summary of Visualizing the Raw Data

i)Fig 2.2 shows that the rent of bikes is lowest in spring season (season 1 ) and highest in fall (season 3) this can be concluded ffrom the fact that that months (1,2,3,12) , (3,4,5,6) , (6,7,8,9) and (9,10,11,12) are seasons 1,2 ,3,4 respectively.

ii) Fig 2.3 shows that the weekdays doesn’t affect the rent count much and the mean count is almost same throughout the days.

iii) Fig 2.4 suggests that weather situation 1 is completely suitable for bikers , while weather situation 3 completely disrupts the bike renting.

iv)Fig 2.5 shows that there is incremental in the bike renting from year 2010 to 2011.

v)Although the amount of total rent bikers has increased over the the year as shown in fig 2.5 , amount of casual users have not increased much but count of registered users have increased a lot (increased sharply).

vi) Fig 2.7 clearly shows that the histograms of various numerical variable a good normal distribution except temp and casual which is slightly left skewed.

Chapter 3 : Pre-processing

To check the outliers present in the data visually

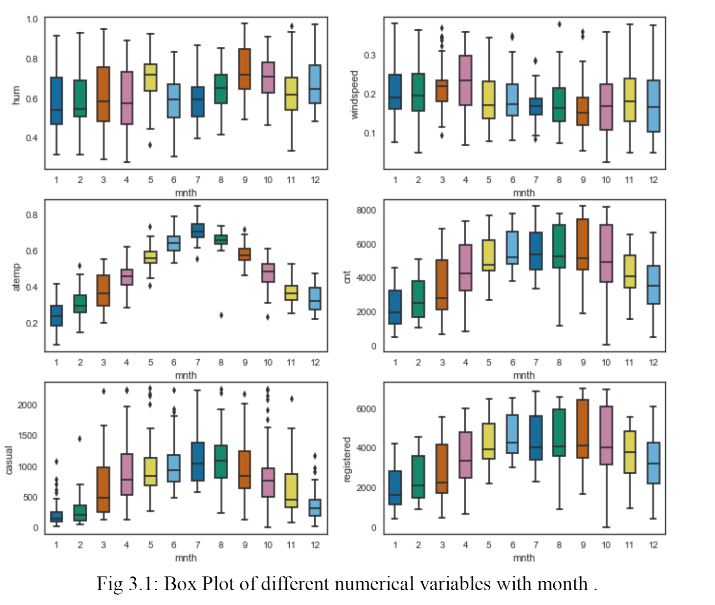


Fig 3.1 shows the number of outliers present in the data , now we need to remove these outliers and replace these outliers with NA and replace these missing values using KNN .

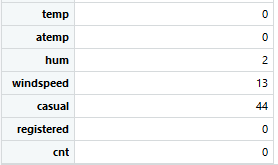
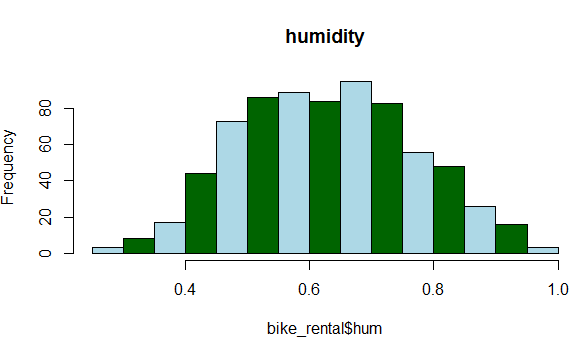
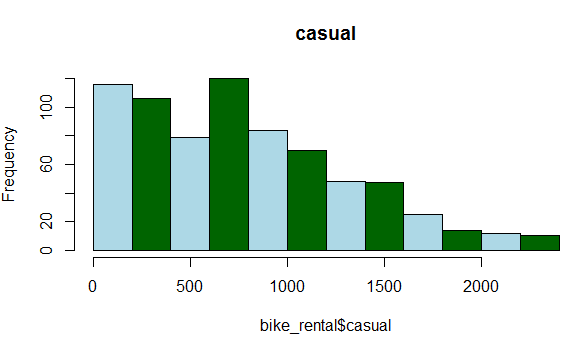
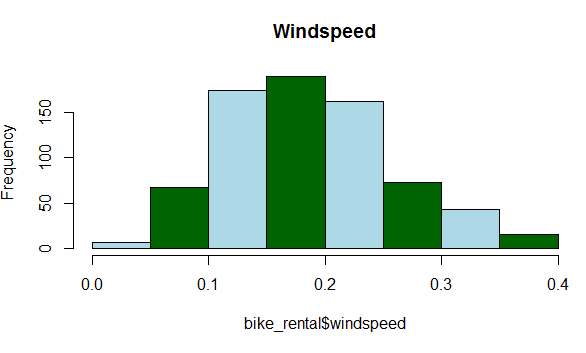


Fig 3.2 :No. of outliers present

In fig: 3.2 we can clearly see the number of outliers present in the given dataset.

Fig 3.3 shows the histogram of variables after removing outliers and impute these outliers with KNN.





**Chapter 4 : Feature Selection**

For selecting the features and understand the inter-relationship between the variables we have

performed two tests

**4.1 Correlation analysis :**Fig 4.1 shows the correlation analysis table where we can deduce that temp and atemp are highly correlated , also registered and our target variable are highly correlated . And increase in humidity and windspeed have high impact on cnt

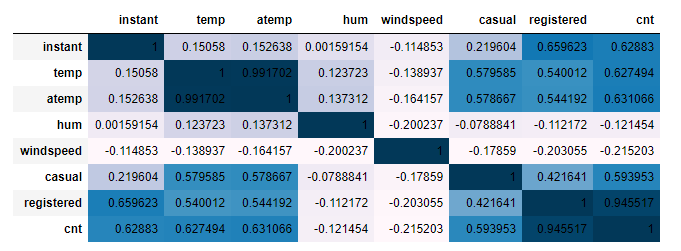


Fig 4.1 Correlation Values

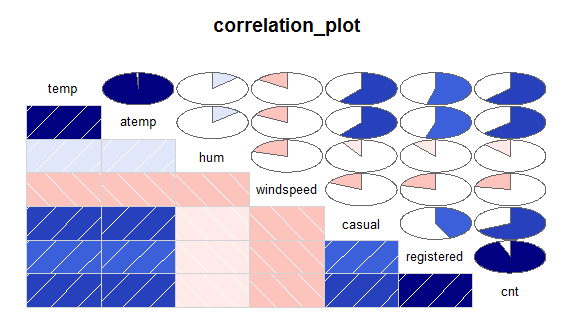


Fig 4.2 Correlation Plot

**4.2 ANOVA test :** ANOVA is performed with numeric and categorical variables combined .The Pr(>F) value suggests the probability of group means are influencing our target variables .Over here weekday variable follows null hypothesis so we need to reject weekday variable as it is not dependent on target variable cnt.

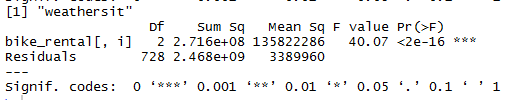
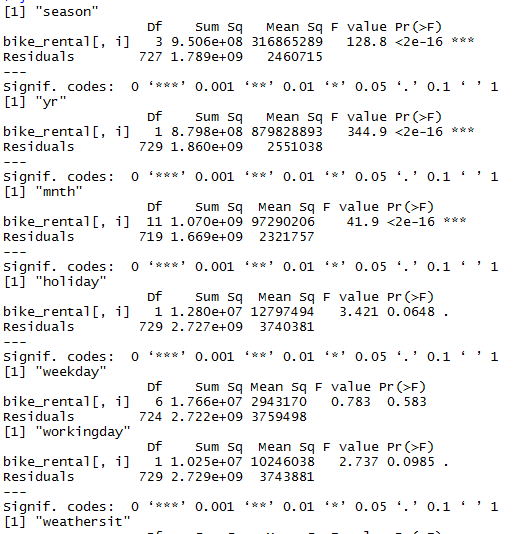
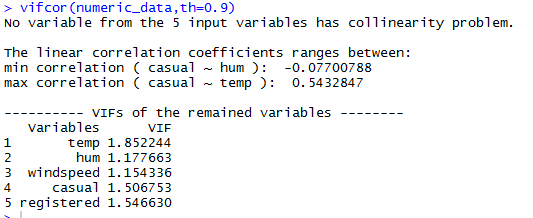


Fig 4.2 ANOVA TEST

Variable weekday has been removed from the dataset.

**4.3 Variance Inflation Factor(VIF) :** We have performed VIF test using function VIF which is used to check whether variables have multicolllinearity .



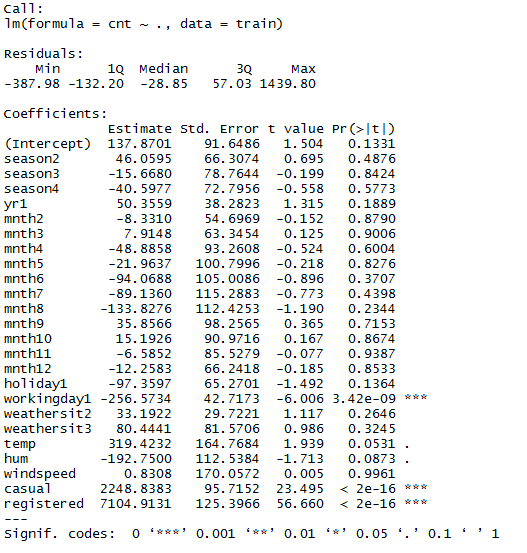
**4.3 Summary of Feature Selection :**

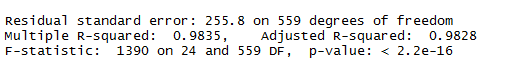
Conclusions from the various test has been drawn that temp and atemp are highly correlated so need to drop atleast one variable .From the ANOVA test we have found that variable ‘weekday’ have p-value greater than 0.05(significance value) so need to drop variable weekday.

**Chapter 5 :Model Development and Conclusion**

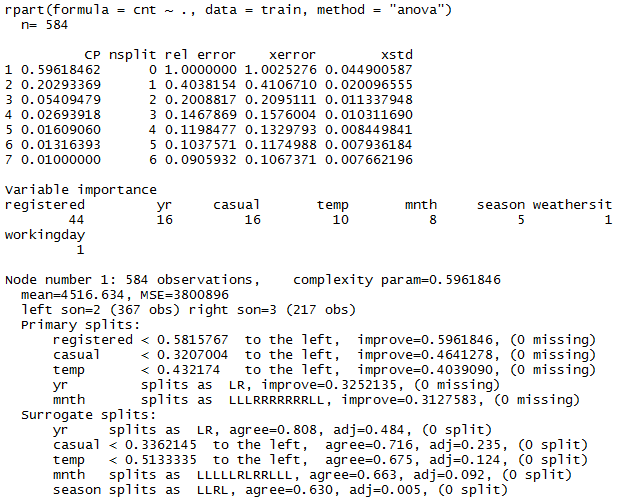
Model Selection :After preprocessing of data we must proceed with model development.For employee absenteeism project, we want to find what changes should company bring to reduce absenteeism problem and also what is the loss in year 2011 per month, if same trend follow.So we need to find the importance of each variable with respect to target variable to suggest the changes for the company and also predict the result of year 2011 loss due to absenteeism.For this we can choose following regression model :

I) Multiple Linear Regression.  
 II)Decesion Tree Regression.  
 III) Random Forest Regression.

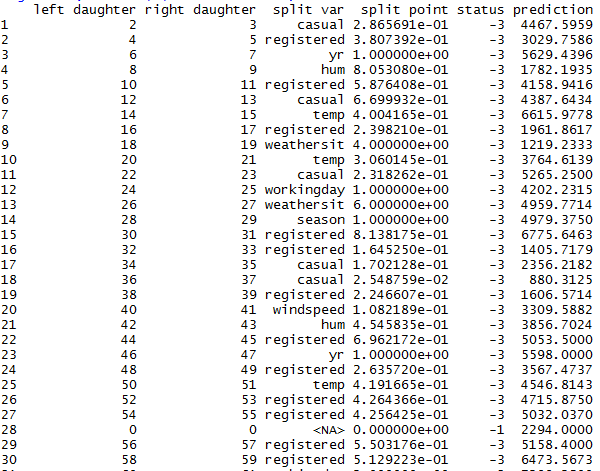
**5.1 Multiple Linear Regression : **

****

Multiple R-squared : 0.9835 .  
 Adjusted R-squared:0.9828 .  
 MAPE : 3.5 %

5.2 Decesion T ree :  MAPE :13.1%

Problem with Decesion tree is that overfitting of the data occur and also mape is high as compared to Multiple Linear Regression.

5.3 Random Forest Model :   
  
  
MAPE : 5.4%   
mape for random forest is lower as compared to mape for decesion tree regressor

**5.4: Model Selection :** As it can be clearly seen from the MAPE result ,Multiple Linear Regression model is performing best for Bike Renting dataset.So we select Multiple Linear Regression for the prediction of the model.

**5.5 Conclusion :**As we can clearly see from our model development result, adjusted R-square value is good so actual variation in target variable is explained by our linear regression model , also MAPE value is 3.5% which is really a good number as per indusytry standards.So Linear Regression Model is best suited for our model development.

**Complete Python Code**

import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import chi2\_contingency

import seaborn as sns

#Upload CSV data file

os.getcwd()

#Upload the CSV file

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

bike\_rental.shape

bike\_rental.columns

#Changing the data type of variables

cat\_names=["season","yr","mnth","holiday","weekday","workingday","weathersit"]

for i in cat\_names:

print(i)

bike\_rental.loc[:,i]=bike\_rental.loc[:,i].astype(str)

#Get column names

def get\_cnames(data):

all\_cnames=[]

num\_cnames=[]

cat\_cnames=[]

for i in data.columns:

all\_cnames.append(str(i))

if(data[i].dtype=="object"):

cat\_cnames.append(str(i))

else:

num\_cnames.append(str(i))

cnames=[all\_cnames,num\_cnames,cat\_cnames]

return(cnames)

#get cnames cnames[0]-all

#cnames[1]-numeric cnames

#cnames[2]-categorical cnames

cnames=get\_cnames(bike\_rental)

#get rows and columns

rows=bike\_rental.shape[0]

columns=bike\_rental.shape[1]

#Missing\_Value\_Analysis

missing\_value=pd.DataFrame(bike\_rental.isnull().sum())

missing\_value

#No missing value is present in the data.

#Load the library

import seaborn as sn

#Visualizing the Raw Data

fig,(ax1,ax2,ax3,ax4, ax5, ax6)= plt.subplots(nrows=6)

fig.set\_size\_inches(13,25)

cnt\_by\_mnth = pd.DataFrame(bike\_rental.groupby("mnth")["cnt"].mean()).reset\_index()

sn.barplot(data=cnt\_by\_mnth,x="mnth",y="cnt",ax=ax1)

cnt\_by\_season = pd.DataFrame(bike\_rental.groupby(["season"])["cnt"].mean()).reset\_index()

sn.barplot(data=cnt\_by\_season,x="season",y="cnt",ax=ax2)

cnt\_by\_weekday = pd.DataFrame(bike\_rental.groupby("weekday")["cnt"].mean()).reset\_index()

sn.barplot(data=cnt\_by\_weekday,x="weekday",y="cnt",ax=ax3)

cnt\_by\_weathersit = pd.DataFrame(bike\_rental.groupby("weathersit")["cnt"].mean()).reset\_index()

sn.barplot(data=cnt\_by\_weathersit,x="weathersit",y="cnt",ax=ax4)

cnt\_by\_yr = pd.DataFrame(bike\_rental.groupby("yr")["cnt"].mean()).reset\_index()

sn.barplot(data=cnt\_by\_yr,x="yr",y="cnt",ax=ax5)

transformed = pd.melt(bike\_rental[["yr","casual","registered"]], id\_vars=['yr'], value\_vars=['casual', 'registered'])

cnt\_by\_user= pd.DataFrame(transformed.groupby(["yr","variable"],sort=True)["value"].mean()).reset\_index()

sn.pointplot(x=cnt\_by\_user["yr"], y=cnt\_by\_user["value"],hue=cnt\_by\_user["variable"],hue\_order=["casual","registered"], data=transformed, join=True,ax=ax6)

ax6.set(xlabel='Year', ylabel='Users Count',label='big')

#Histograms before removing outliers

a4\_dims = (14, 14)

fig, axes = plt.subplots(nrows=3,ncols=2,figsize=a4\_dims)

sn.set(color\_codes=True)

sn.set(style="white", palette="muted")

sn.distplot(bike\_rental['hum'], ax=axes[0][0])

sn.distplot(bike\_rental['atemp'], ax=axes[0][1])

sn.distplot(bike\_rental['registered'], ax=axes[1][0])

sn.distplot(bike\_rental['windspeed'], ax=axes[1][1])

sn.distplot(bike\_rental['cnt'], ax=axes[2][0])

sn.distplot(bike\_rental['casual'], ax=axes[2][1])

#PLotting Scatter Plot

#Visualizing distribution

ax8 =bike\_rental.plot(kind='scatter',x='instant', y='registered')

ax9 = bike\_rental.plot(kind='scatter',x='instant', y='casual')

ax10 = bike\_rental.plot(kind='scatter',x='instant', y='cnt')

#Outlier Analysis

out\_data=bike\_rental.loc[:,cnames[1]]

out\_data.drop(["cnt"], axis = 1, inplace = True )

onames=out\_data.columns

onames

for i in onames:

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

iqr=q75-q25

min=q25 - (1.5\*iqr)

max=q75 + (1.5\*iqr)

bike\_rental.loc[bike\_rental[i]<min,i]=np.nan

bike\_rental.loc[bike\_rental[i]>max,i]=np.nan

# #Calculating missing value after outlier analysis

missing\_val = pd.DataFrame(bike\_rental.isnull().sum())

missing\_val

#Impute missing values using median

bike\_rental['hum']=bike\_rental['hum'].fillna(bike\_rental['hum'].median())

bike\_rental['windspeed']=bike\_rental['windspeed'].fillna(bike\_rental['windspeed'].median())

bike\_rental['casual']=bike\_rental['casual'].fillna(bike\_rental['casual'].median())

#Histograms After Outlier Analysis

a4\_dims = (14, 14)

fig, axes = plt.subplots(nrows=3,ncols=2,figsize=a4\_dims)

sn.set(color\_codes=True)

sn.set(style="white", palette="muted")

sn.distplot(bike\_rental['hum'], ax=axes[0][0])

sn.distplot(bike\_rental['atemp'], ax=axes[0][1])

sn.distplot(bike\_rental['registered'], ax=axes[1][0])

sn.distplot(bike\_rental['windspeed'], ax=axes[1][1])

sn.distplot(bike\_rental['cnt'], ax=axes[2][0])

sn.distplot(bike\_rental['casual'], ax=axes[2][1])

#Feature Selection

#Correlation Analysis

cor=cnames[1]

bike\_rental\_corr=bike\_rental.loc[:,cor]

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

corr=bike\_rental\_corr.corr()

import seaborn as sns

sns.heatmap(corr,mask=np.zeros\_like(corr,dtype=np.bool),cmap=sns.diverging\_palette(220,50,as\_cmap=True),square=True,ax=ax)

#Feature Selection Using Anova Test

#ANOVA Analysis

import statsmodels.api as sm

from statsmodels.formula.api import ols

cw\_lm=ols('cnt ~ C(yr)+C(holiday)+C(workingday)+ C(mnth)+C(weekday)+ C(weathersit)+C(season)', data=bike\_rental).fit()

print(sm.stats.anova\_lm(cw\_lm, typ=2))

#Dimensionality Reduction

bike\_rental=bike\_rental.drop(['weekday','instant','dteday','atemp'],axis=1)

#Feature Scaling

fea\_name=['casual','registered']

for i in fea\_name:

print(i)

bike\_rental[i]=(bike\_rental[i]-np.min(bike\_rental[i]))/(np.max(bike\_rental[i])-np.min(bike\_rental[i]))

#Error metrix

from sklearn.metrics import mean\_squared\_error

from math import sqrt

def MAPE(y,yhat):

print(np.mean(np.abs((y - yhat) / y))\*100)

#Model Development

##Linear Regression Model

X=bike\_rental.iloc[:,:-1].values

Y=bike\_rental.iloc[:,11].values

#Splitting the data in train and testing

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,Y\_train,Y\_test=train\_test\_split(X,Y,test\_size=0.2)

#Multiple Linear Regression:

from sklearn.linear\_model import LinearRegression

lm\_model=LinearRegression()

lm\_model.fit(X\_train,Y\_train)

lm\_predict=lm\_model.predict(X\_test)

MAPE(Y\_test,lm\_predict)

#Decision Tree Regressor

from sklearn.tree import DecisionTreeRegressor

DT\_model=DecisionTreeRegressor()

DT\_model.fit(X\_train,Y\_train)

DT\_predict=DT\_model.predict(X\_test)

MAPE(Y\_test,DT\_predict)

#Random Forest Regressor

from sklearn.ensemble import RandomForestRegressor

RF\_model=RandomForestRegressor()

RF\_model.fit(X\_train,Y\_train)

RF\_predict=RF\_model.predict(X\_test)

MAPE(Y\_test,RF\_predict)

**Complete R Code**

rm(list=ls())

getwd()

#Loading Required Libraries

x = c("ggplot2", "corrgram", "DMwR", "caret", "randomForest", "unbalanced", "C50", "dummies", "e1071", "Information",

"MASS", "rpart", "gbm", "ROSE", 'sampling', 'DataCombine', 'inTrees',"usdm","scales","psych","gplots")

install.packages(x)

lapply(x, require, character.only = TRUE)

#Loading CSV file

bike\_rental = read.csv("day.csv", header = T, na.strings = c(" ", "", "NA"))

#Converting the required numerical variables to factor variables:

catnames=c("season","yr","mnth","holiday","weekday","workingday","weathersit")

for(i in catnames){

print(i)

bike\_rental[,i]=as.factor(bike\_rental[,i])

}

str(bike\_rental)

bike\_rental$dteday=NULL

bike\_rental$instant=NULL

#Missing Value Analysis

sum(is.na(bike\_rental))

##No missing values are present in the given data set.

#Outlier Analysis

class(bike\_rental)

num\_index=sapply(bike\_rental, is.numeric)

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

num\_cnames=colnames(numeric\_data)

hist(bike\_rental$temp,main = "Temperature",col = (c("lightblue","darkgreen")))

hist(bike\_rental$atemp,main = "ATemperature",col = (c("lightblue","darkgreen")))

hist(bike\_rental$hum,main = "Humidity",col = (c("lightblue","darkgreen")))

hist(bike\_rental$windspeed,main = "Windspeed",col = (c("lightblue","darkgreen")))

hist(bike\_rental$casual,main = "Casual",col = (c("lightblue","darkgreen")))

hist(bike\_rental$registered,main = "Registered",col = (c("lightblue","darkgreen")))

hist(bike\_rental$cnt,main = "Count",col = (c("lightblue","darkgreen")))

num\_cnames=num\_cnames[num\_cnames!="cnt"]

num\_cnames

for (i in 1:length(num\_cnames))

{

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

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=num\_cnames[i],x="cnt")+

ggtitle(paste("Box plot of count for",num\_cnames[i])))

}

# ## Plotting plots together

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

gridExtra::grid.arrange(gn4,gn5,ncol=2)

gridExtra::grid.arrange(gn6,gn3,ncol=2)

for(i in num\_cnames){

print(i)

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

print(length(val))

}

#Replace all outliers with NA and impute

for(i in num\_cnames){

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

print(length(val))

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

}

sum(is.na(bike\_rental))

bike\_rental = knnImputation(bike\_rental, k = 7)

sum(is.na(bike\_rental))

#Plotting Bar Plot together

install.packages("scales")

library("ggplot2")

library(scales)

library("psych")

library("gplots")

#Plot of month vs count

ggplot(bike\_rental, aes\_string(x = bike\_rental$mnth,y=bike\_rental$cnt)) +

geom\_bar(stat="identity",fill = "DarkSlateBlue") + theme\_bw() +

xlab("Month") + ylab('Count') +

ggtitle("Distribution of count by months") + theme(text=element\_text(size=12))

#PLot of seasons vs count

ggplot(bike\_rental, aes\_string(x = bike\_rental$season,y=bike\_rental$cnt)) +

geom\_bar(stat="identity",fill = "DarkSlateBlue") + theme\_bw() +

xlab("Seasons") + ylab('Count') +

ggtitle("Distribution of count by seasons") + theme(text=element\_text(size=12))

#Plot of weekday vs count

ggplot(bike\_rental, aes\_string(x = bike\_rental$weekday,y=bike\_rental$cnt)) +

geom\_bar(stat="identity",fill = "DarkSlateBlue") + theme\_bw() +

xlab("Weekday") + ylab('Count') +

ggtitle("Distribution of count by weekday") + theme(text=element\_text(size=12))

#Plot of weathersit vs count

ggplot(bike\_rental, aes\_string(x = bike\_rental$weathersit,y=bike\_rental$cnt)) +

geom\_bar(stat="identity",fill = "DarkSlateBlue") + theme\_bw() +

xlab("Weathersit") + ylab('Count') +

ggtitle("Distribution of count by weathersit") + theme(text=element\_text(size=12))

#Plot of yr vs count

ggplot(bike\_rental, aes\_string(x = bike\_rental$yr,y=bike\_rental$cnt)) +

geom\_bar(stat="identity",fill = "DarkSlateBlue") + theme\_bw() +

xlab("Year") + ylab('Count') +

ggtitle("Distribution of count by Year") + theme(text=element\_text(size=12))

#Feature Selection

#Correlation Plot

corrgram(bike\_rental,upper.panel = panel.pie,text.panel = panel.txt,main="correlation\_plot")

#As temp and atemp are highly correlated so one variable need to be remove, so atemp was removed

#ANOVA test

for (i in catnames) {

print(i)

print(summary(aov(bike\_rental$cnt~bike\_rental[,i],bike\_rental)))

}

bike\_rental=subset(bike\_rental,select=-c(weekday,atemp))

#Checking from variance inflation factor

numeric\_data=subset(numeric\_data,select=-c(atemp,cnt))

library(usdm)

vif(numeric\_data)

vifcor(numeric\_data,th=0.9)

#From the Variance inflation factor we can clearly see that no variable is having multicollinearity issues

#Feature Scaling

fea\_names=c("casual","registered")

for (i in fea\_names) {

bike\_rental[,i]=(bike\_rental[,i]-min(bike\_rental[,i]))/(max(bike\_rental[,i])-min(bike\_rental[,i]))

}

#Plotting Histogram after preprocessing of Data

hist(bike\_rental$temp,main = "Temperature",col = (c("lightblue","darkgreen")))

hist(bike\_rental$hum,main = "Humidity",col = (c("lightblue","darkgreen")))

hist(bike\_rental$windspeed,main = "Windspeed",col = (c("lightblue","darkgreen")))

hist(bike\_rental$casual,main = "Casual",col = (c("lightblue","darkgreen")))

hist(bike\_rental$registered,main = "Registered",col = (c("lightblue","darkgreen")))

hist(bike\_rental$cnt,main = "Count",col = (c("lightblue","darkgreen")))

#Model Development

set.seed(123)

X\_index=sample(1:nrow(bike\_rental),0.8\*nrow(bike\_rental))

X\_train=bike\_rental[X\_index,-12]

X\_test=bike\_rental[-X\_index,-12]

Y\_train=bike\_rental[X\_index,12]

Y\_test=bike\_rental[-X\_index,12]

train=bike\_rental[X\_index,]

test=bike\_rental[-X\_index,]

#Calculate MAPE

MAPE = function(y, yhat){

mean(abs((y - yhat)/y))

}

###########################Multiple Linear Regression###################

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

summary(lm\_model)

#Predict for new test cases

cat\_index=sapply(bike\_rental, is.factor)

cat\_data=bike\_rental[,cat\_index]

cat\_cnames=colnames(cat\_data)

cat\_cnames

for (i in cat\_cnames) {

lm\_model$xlevels[[i]]=union(lm\_model$xlevels[[i]],levels(X\_test[[i]]))

}

lm\_predict=predict(lm\_model,newdata = X\_test)

lm\_predict

MAPE(lm\_predict,Y\_test)

######################Decesion Tree Regression#####################

DT\_model=rpart(cnt ~.,data=train,method = "anova")

#Predict for new test cases

DT\_predict=predict(DT\_model,X\_test)

summary(DT\_model)

MAPE(DT\_predict,Y\_test)

#####################Random Forest###############

RF\_model=randomForest(x=X\_train,y=Y\_train,ntree = 100)

#Predict for new test cases

RF\_predict=predict(RF\_model,X\_test)

getTree(RF\_model,1,labelVar = TRUE)

summary(RF\_model)

MAPE(RF\_predict,Y\_test)