BIKE RENTAL ANALYSIS

ADITYA JOSHI

02-08-2019

Contents

1. Introduction

1.1 Problem Statement

1.2 Data

2. Methodology

2.1 Pre-Processing

2.2 Distribution of continuous variables

2.3 Distribution of categorical variables

2.4 Relationship of Continuous variables against bike count

2.5 Detection of outliers

2.6 Feature Selection

3. Modelling

3.1 Model Selection

3.2 Multiple Linear Regressions

3.3 Decision Tree

3.4 Random Forest

4. Conclusion

4.1 MAPE

5. R code

Chapter 1: Introduction

1.1 Problem Statement

The aim of this project is to predict the count of bike rentals based on the seasonal and environmental settings. By predicting the count, it would be possible to help accommodate in managing the number of bikes required on a daily basis, and being prepared for high demand of bikes during peak periods

1.2 Data

The goal is to build regression models which will predict the number of bikes used based on the environmental and season behaviour. Given below is a sample of the data set that we are using to predict the number of bikes:

Table 1.1: Bike Count Sample Data (Columns: 1-9)

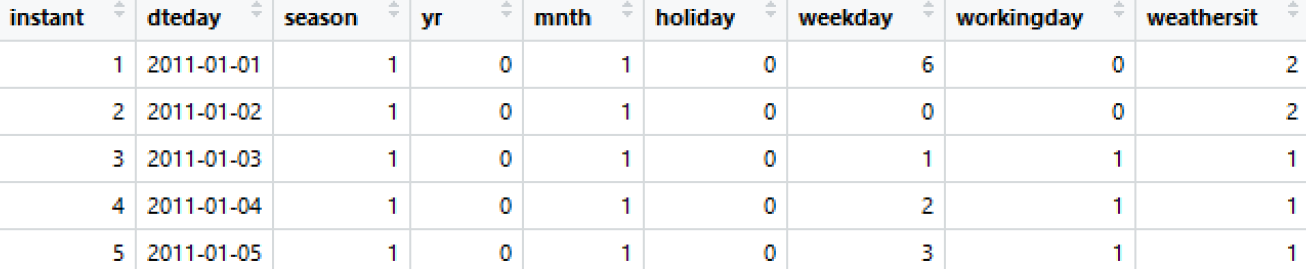
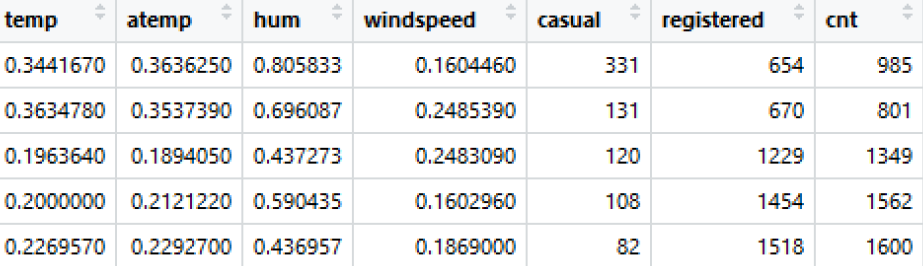


Table 1.2: Bike Count Sample Data (Columns: 10-16)



As you can see in the table below we have the following 13 variables, using which we have to correctly predict the count of bikes:

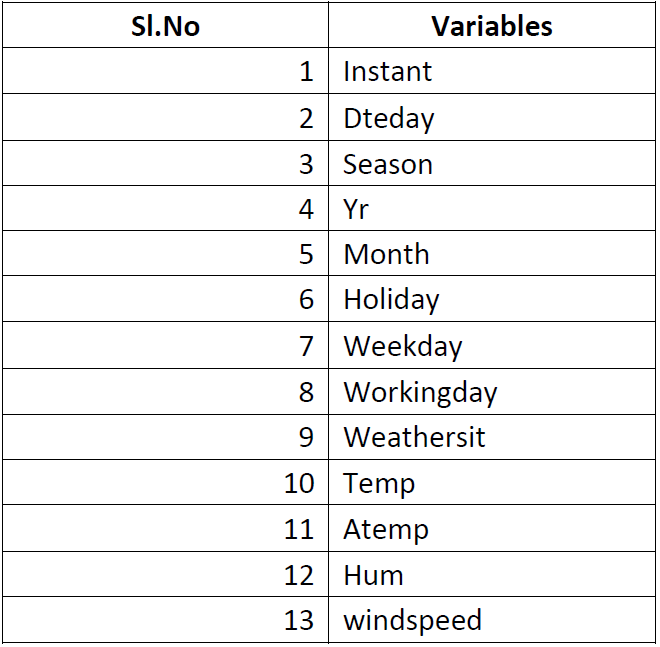


TABLE 1.3 : PREDICTOR VARIABLES

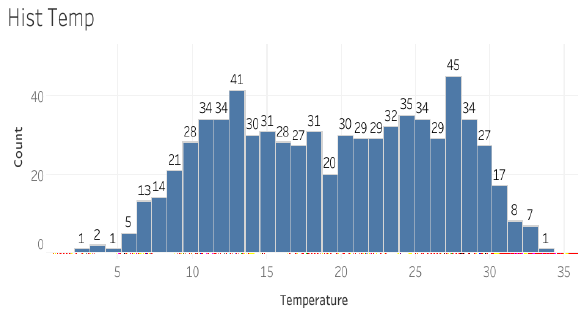
Chapter 2: Methodology

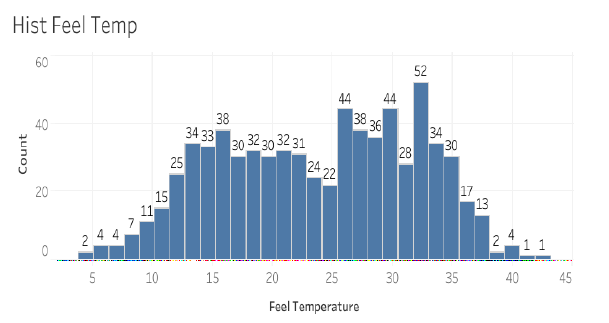
**2.1 Pre-Processing**

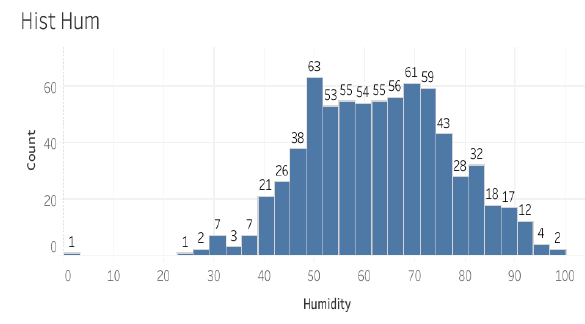
A predictive model requires that we look at the data before we start to create a model. However, in data mining, looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is known as Exploratory Data Analysis.

**2.2 Distribution of continuous variables**

It can be observed from the below histograms is that temperature and feel temperature are normally distributed, where as the variables windspeed and humidity are slightly skewed. The skewness is likely because of the presence of outliers and extreme data in those variables.







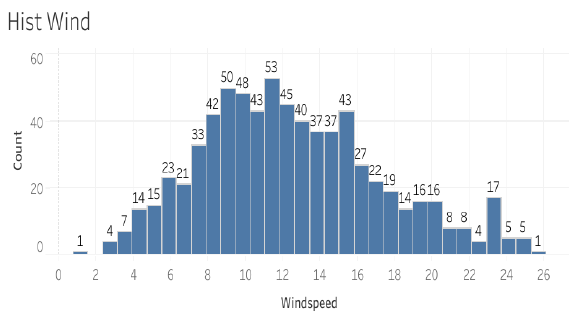
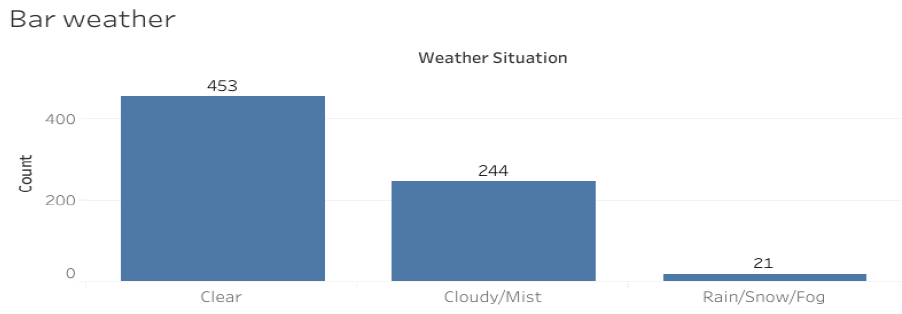
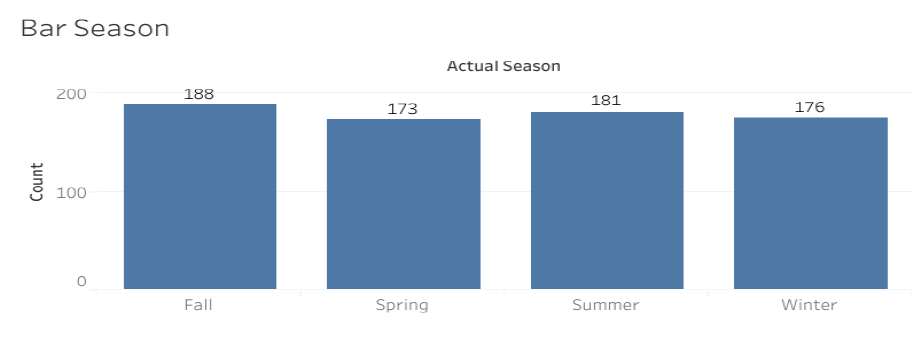


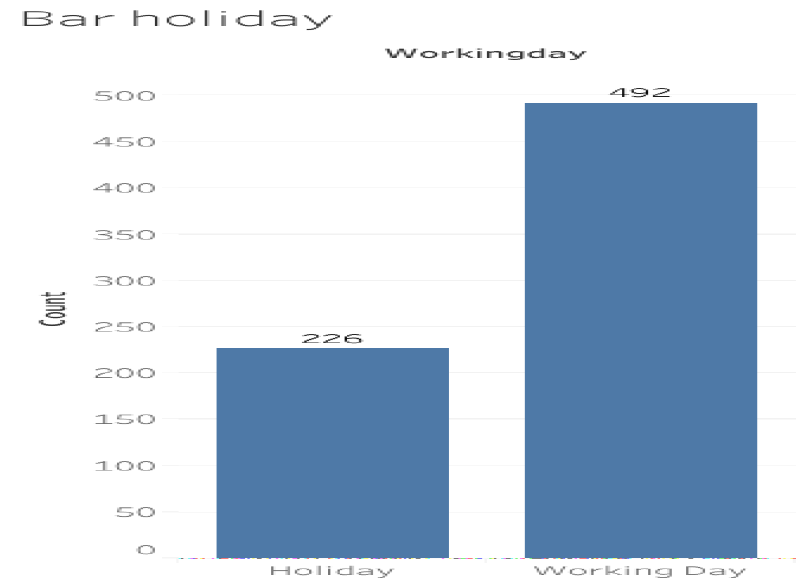
Fig 2.1: Distribution of continuous variables using Histograms

**2.3 Distribution of categorical variables**

The distribution of categorical variables is as shown in the below figure:

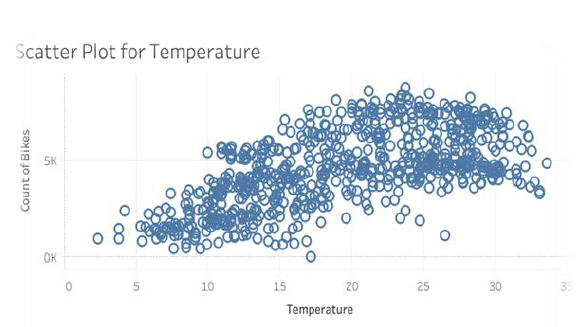


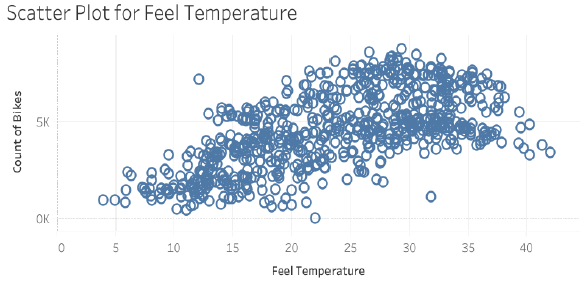


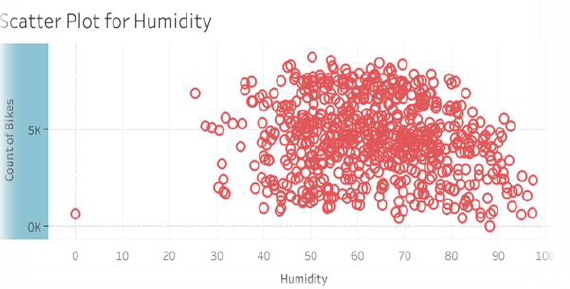


**2.4 Relationship of Continuous variables against bike count**

The below figure shows the relationship between continuous variables and the target variable using scatter plot. It can be observed that there exists a linear positive relationship between the variables temperature and feel temperature with the bike rental count. There also exists a negative linear relationship between the variable’s humidity and windspeed with the bike rental count.







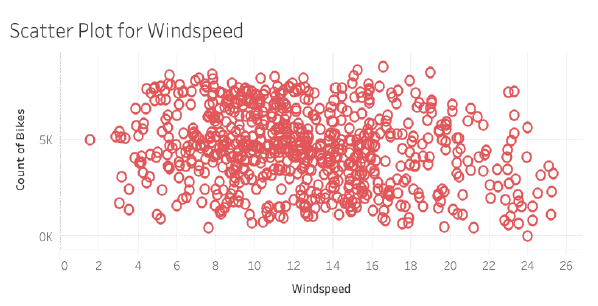
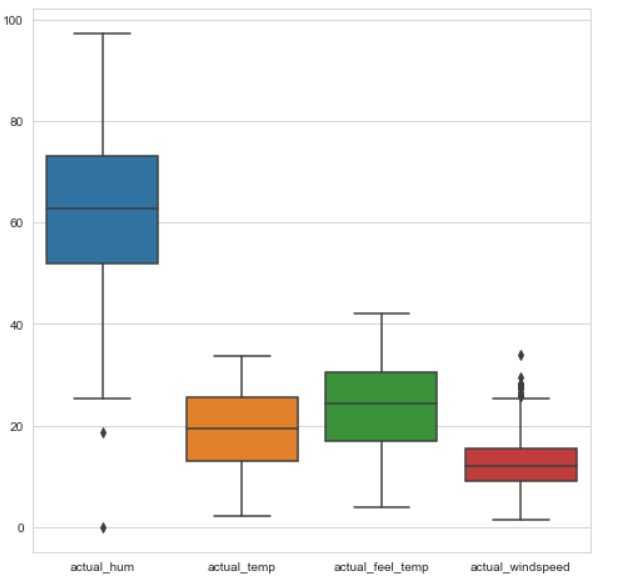
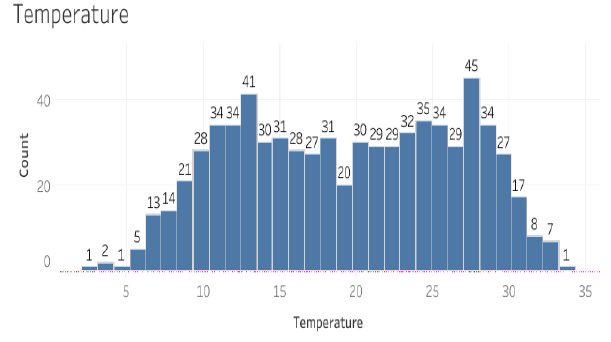
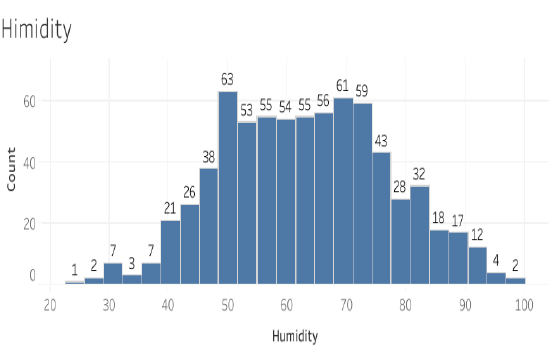


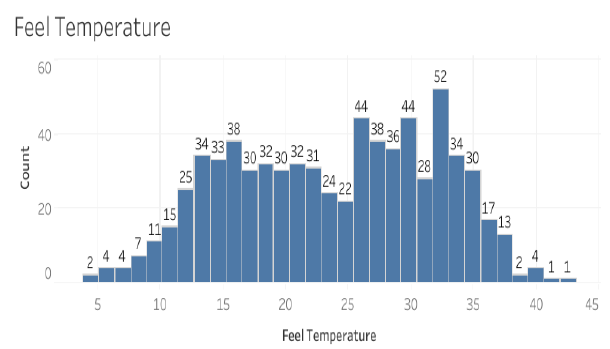
Fig 2.3: Scatter plot for continuous variables

Outliers can be removed using the Boxplot stats method, wherein the Inter Quartile Range (IQR) is calculated and the minimum and maximum value are calculated for the variables. Any value ranging outside the minimum and maximum value are discarded. The boxplot of the continuous variables after removing the outliers is shown in the below figure:









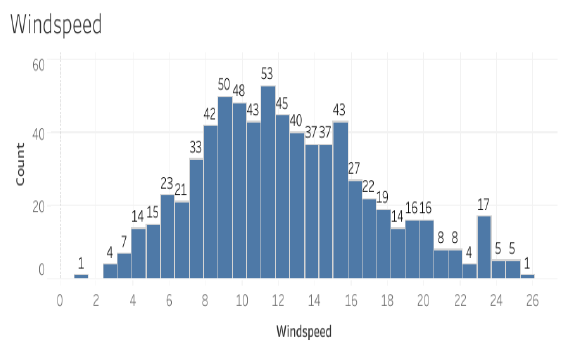


Fig 2.6: Distribution of numerical data using histograms after removal of outliers

**2.6: Feature Selection**

Feature Selection reduces the complexity of a model and makes it easier to interpret. It also reduces overfitting. 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.

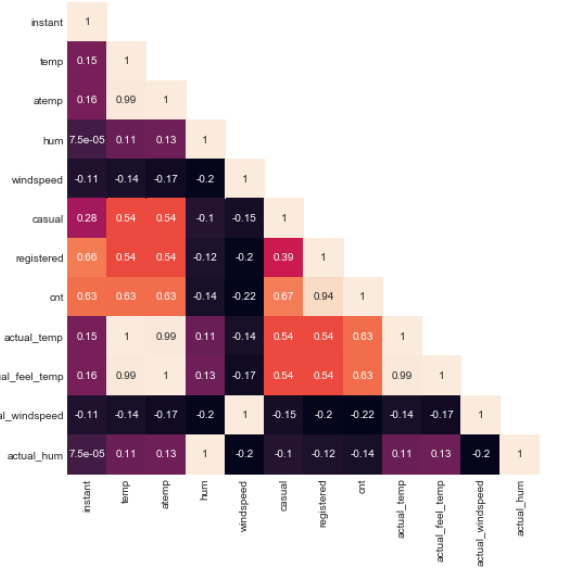


Fig 2.7: Correlation plot of all the variables

**Chapter 3: Modelling**

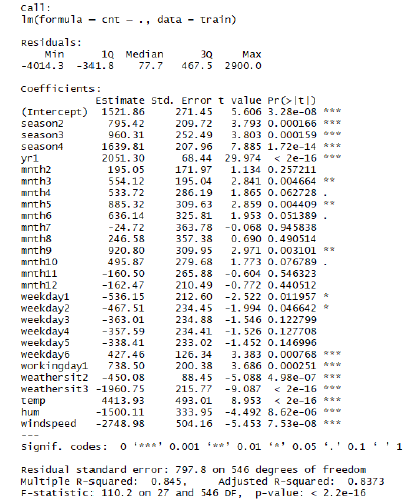
**3.1 Model Selection**

The dependent variable in our model is a continuous variable i.e., Count of bike rentals. Hence the models that we choose are Linear Regression, Decision Tree

and Random Forest. The error metric chosen for the problem statement is Mean Absolute Error (MAE).

**3.2 Multiple Linear Regression**

Multiple linear regression is the most common form of linear regression analysis. Multiple linear regression is used to explain the relationship between one continuous dependent variable and two or more independent variables. The independent variables can be continuous or categorical.



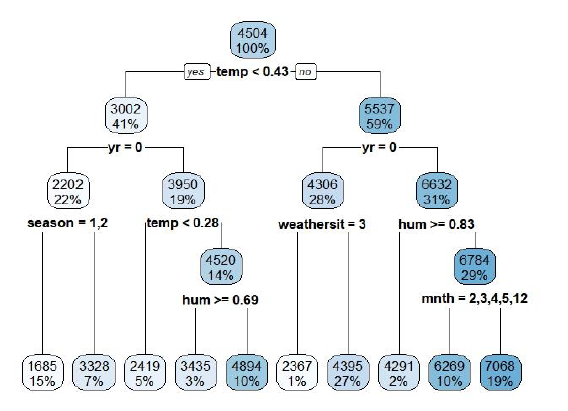
As you can see the Adjusted R-squared value, we can explain 83.73% of the data using our multiple linear regression model. By looking at the F-statistic and combined p-value we can

reject the null hypothesis that target variable does not depend on any of the predictor variables. This model explains the data very well and is considered to be good.

Even after removing the non-significant variables, the accuracy, Adjusted R-squared and F-statistic do not change by much, hence the accuracy of this model is chosen to be final. Mean Absolute Error (MAE) is calculated and found to be 494. MAPE of this multiple linear regression model is 12.17%. Hence the accuracy of this model is 87.83%. This model performs very well for this test data.

**3.3 Decision Tree:**

A decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions.



Using decision tree, we can predict the value of bike count. MAE for this model is 684. The MAPE for this decision tree is 17.47%. Hence the accuracy for this model is 82.53%.

**3.4 Random Forest:**

Using Classification for prediction analysis in this case is not normal, though it can be done. The number of decision trees used for prediction in the forest is 500. MAE for this model is 392. Using random forest, the MAPE was found to be 10.68%. Hence the accuracy is 89.32%.

**Chapter 4: Conclusion**

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 count prediction Data, 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 Predictions of the models with real values of the target variables, and calculating some average error measure.

**4.1 Mean Absolute Error (MAE)**

MAE is one of the error measures used to calculate the predictive performance of the model. We will apply this measure to our models that we have generated in the previous section.

MAE <- function (actual, pred)

{

print(mean (abs (actual - pred)))

}

**Linear Regression Model: MAE = 494**

**Decision Tree: MAE = 684.**

**Random Forest: MAE = 392**

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

**Chapter 5: R code**

**#Cleaning environment**

rm(list = ls())

**#Setting work directory**

setwd("C:/Users/aditya joshi/Edwisor/Bike Rental Analysis")

**#Loading librarires**

libraries = c("plyr","dplyr", "ggplot2","rpart","dplyr","DMwR","randomForest","usdm","corrgram","DataCombine")

lapply(X = libraries,require, character.only = TRUE)

rm(libraries)

**#Read the csv file**

data = read.csv(file = "day.csv", header = T, sep = ",", na.strings = c(" ", "", "NA"))

**########################################EXPLORING THE DATA#########################################**

**#First few rows**

head(data)

**#data Dimensions**

dim(data)

**#names of cloumns**

names(data)

**#Structures of variables**

str(data)

**########################################FEATURE ENGINEERING########################################**

**#Creating columns**

data$actual\_temp <- data$temp\*39

data$actual\_feel\_temp <- data$atemp\*50

data$actual\_windspeed <- data$windspeed\*67

data$actual\_hum = data$hum \* 100

data$actual\_season = factor(x = data$season, levels = c(1,2,3,4), labels = c("Spring","Summer","Fall","Winter"))

data$actual\_yr = factor(x = data$yr, levels = c(0,1), labels = c("2011","2012"))

data$actual\_holiday = factor(x = data$holiday, levels = c(0,1), labels = c("Working day","Holiday"))

data$actual\_weathersit = factor(x = data$weathersit, levels = c(1,2,3,4),

labels = c("Clear","Cloudy/Mist","Rain/Snow/Fog","Heavy Rain/Snow/Fog"))

data$weathersit = as.factor(data$weathersit)

data$season = as.factor(data$season)

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

data$mnth = as.factor(data$mnth)

data$weekday = as.factor(as.character(data$weekday))

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

data$yr = as.factor(data$yr)

data$holiday = as.factor(data$holiday)

**########################################MISSING VALUES########################################**

missing\_values = sapply(data, function(x){sum(is.na(x))})

**########################################EXPLORE USING GRAPHS########################################**

**#Checking distribution of categorical Data using bar graph**

graph1 = ggplot(data = data, aes(x = actual\_season)) + geom\_bar() + ggtitle("Count of Season")

graph2 = ggplot(data = data, aes(x = actual\_weathersit)) + geom\_bar() + ggtitle("Count of Weather")

graph3 = ggplot(data = data, aes(x = actual\_holiday)) + geom\_bar() + ggtitle("Count of Holiday")

graph4 = ggplot(data = data, aes(x = workingday)) + geom\_bar() + ggtitle("Count of Working day")

**# ## Plot plots together**

gridExtra::grid.arrange(graph1,graph2,graph3,graph4,ncol=2)

**#Checking distribution of numerical data using histogram**

gram1 = ggplot(data = data, aes(x =actual\_temp)) + ggtitle("Distribution of Temperature") + geom\_histogram(bins = 25)

gram2 = ggplot(data = data, aes(x =actual\_hum)) + ggtitle("Distribution of Humidity") + geom\_histogram(bins = 25)

gram3 = ggplot(data = data, aes(x =actual\_feel\_temp)) + ggtitle("Distribution of Feel Temperature") + geom\_histogram(bins = 25)

gram4 = ggplot(data = data, aes(x =actual\_windspeed)) + ggtitle("Distribution of Windspeed") + geom\_histogram(bins = 25)

**# ## Plot plots together**

gridExtra::grid.arrange(gram1,gram2,gram3,gram4,ncol=2)

**#Checking distribution of numerical data using scatterplot**

scan1 = ggplot(data = data, aes(x =actual\_temp, y = cnt)) + ggtitle("Distribution of Temperature") + geom\_point() + xlab("Temperature") + ylab("Bike COunt")

scan2 = ggplot(data = data, aes(x =actual\_hum, y = cnt)) + ggtitle("Distribution of Humidity") + geom\_point(color="red") + xlab("Humidity") + ylab("Bike COunt")

scan3 = ggplot(data = data, aes(x =actual\_feel\_temp, y = cnt)) + ggtitle("Distribution of Feel Temperature") + geom\_point() + xlab("Feel Temperature") + ylab("Bike COunt")

scan4 = ggplot(data = data, aes(x =actual\_windspeed, y = cnt)) + ggtitle("Distribution of Windspeed") + geom\_point(color="red") + xlab("Windspeed") + ylab("Bike COunt")

**# ## Plot plots together**

gridExtra::grid.arrange(scan,scan2,scan3,scan4,ncol=2)

**#Checking outliers in data using boxplot**

columnnames = colnames(day[,c("actual\_temp","actual\_feel\_temp","actual\_windspeed","actual\_hum")])

for (i in 1:length(columnnames))

{

assign(paste0("gn",i), ggplot(aes\_string(y = columnnames[i]), data = 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=columnnames[i])+

ggtitle(paste("Box plot for",columnnames[i])))

}

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

**#Removing outliers in Windspeed**

val = data[,19][data,19] %in% boxplot.stats(data[,19])$out]

data = data[which(!data[,19] %in% val),]

**#Checking multicollinearity using VIF**

df1 = data[,c("instant","temp","atemp","hum","windspeed")]

vifcor(df1)

**#Checking for collinearity using corelation graph**

corrgram(data, order = F, upper.panel=panel.pie, text.panel=panel.txt, main = "Correlation Plot")

**#Removing unwanted variables**

data <- subset(data, select = -c(holiday,instant,dteday,atemp,casual,registered,actual\_temp,actual\_feel\_temp,actual\_windspeed,

actual\_hum,actual\_season,actual\_yr,actual\_holiday,actual\_weathersit))

rmExcept(keepers = "data")

########################################DECISION TREE########################################

#MAPE: 17.47%

#MAE: 684

#RMSE: 864.8

#Accuracy: 82.53%

#Dividing data into train and test

set.seed(123)

train\_index = sample(1:nrow(data), 0.8 \* nrow(data))

train = data[train\_index,]

test = data[-train\_index,]

**#rparting regression**

dx1\_model = rpart(cnt ~ ., data = train, method = "anova")

**#Predicting test cases**

dx1\_predictions = predict(dx1\_model, test[,-10])

**#Creating dataframe for actual and predicted values**

df1 = data.frame("actual"=test[,10], "pred"=dx1\_predictions)

head(df1)

#calculating MAPE

regr.eval(trues = test[,10], preds = dx1\_predictions, stats = c("mae","mse","rmse","mape"))

**#calculating MAPE**

MAPE = function(actual, pred){

print(mean(abs((actual - pred)/actual)) \* 100)

}

MAPE(test[,10], dx1\_predictions)

**########################################RANDOM FOREST########################################**

**#MAPE: 10.68%**

**#MAE: 392**

**#RMSE: 535**

**#Accuracy: 89.32%**

**#Training data using random forest**

rx1\_model = randomForest(cnt~., data = train, ntree = 500)

**#Predicting test cases**

rx1\_predictions = predict(rx1\_model, test[,-10])

**#Creating dataframe for actual and predicted values**

df1 = cbind(df1,rx1\_predictions)

head(df1)

**#Calculating MAPE**

regr.eval(trues = test[,10], preds = rx1\_predictions, stats = c("mae","mse","rmse","mape"))

MAPE(test[,10], rx1\_predictions)

**########################################LINEAR REGRESSION########################################**

**#MAPE: 12.17%**

**#RMSE: 673**

**#Accuracy: 87.83%**

**#MAE: 494**

**#Adjusted R squared: 0.8373**

**#F-statistic: 110.2**

**#Trainning data using linear regression**

lx1\_model = lm(formula = cnt~., data = train)

**#Checking summary of the model**

summary(lx1\_model)

**#Predicting test cases**

lx1\_predictions = predict(lx1\_model, test[,-10])

**#Creating dataframe for actual and predicted values**

df1 = cbind(df1,lrx1\_predictions)

head(df1)

**#Calculating MAPE**

regr.eval(trues = test[,10], preds = lx1\_predictions, stats = c("mae","mse","rmse","mape"))

MAPE(test[,10], lx1\_predictions)

**#Ploting graph for actual vs predicted values**

plot(test$cnt,type="l",lty=2,col="green")

lines(lx1\_predictions,col="blue")

**#Predicting sample data**

predict(lx1\_model,test[2,])