# Predicting Bike Rental Count

Vinod Pawar

# Contents

# Chapter 1: Introduction

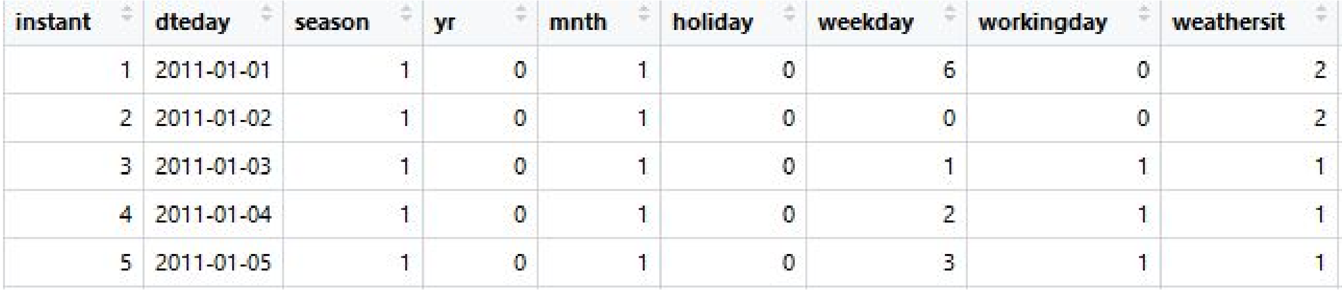
## 1.1 Problem Statement

The project aim is to predict the count of bike rentals based on the various season and environmental factors. 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

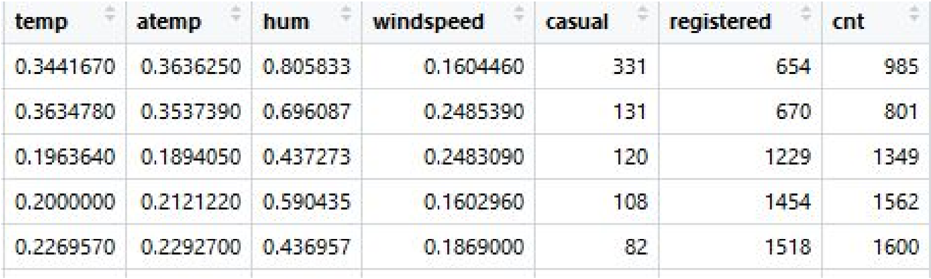
The goal is to build regression models which will predict the number of bikes used based on the environmental and season behavior. Given below is a sample of the data set that we are using to predict the number of bikes:

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



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

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



| temp | atemp | hum | windspeed | casual | registered | cnt |
|---|---|---|---|---|---|---|
| 0.3441670 | 0.3636250 | 0.805833 | 0.1604460 | 331 | 654 | 985 |
| 0.3634780 | 0.3537390 | 0.696087 | 0.2485390 | 131 | 670 | 801 |
| 0.1963640 | 0.1894050 | 0.437273 | 0.2483090 | 120 | 1229 | 1349 |
| 0.2000000 | 0.2121220 | 0.590435 | 0.1602960 | 108 | 1454 | 1562 |
| 0.2269570 | 0.2292700 | 0.436957 | 0.1869000 | 82 | 1518 | 1600 |

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:

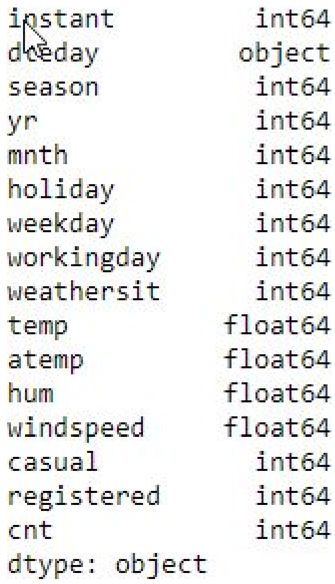
| Sl.No | Variables |
|---|---|
| 1 | Instant |
| 2 | Dteday |
| 3 | Season |
| 4 | Yr |
| 5 | Month |
| 6 | Holiday |
| 7 | Weekday |
| 8 | Workingday |
| 9 | Weathersit |
| 10 | Temp |
| 11 | Atemp |
| 12 | Hum |
| 13 | windspeed |

Table 1.3: Predictor variables

# Chapter 2: Methodology

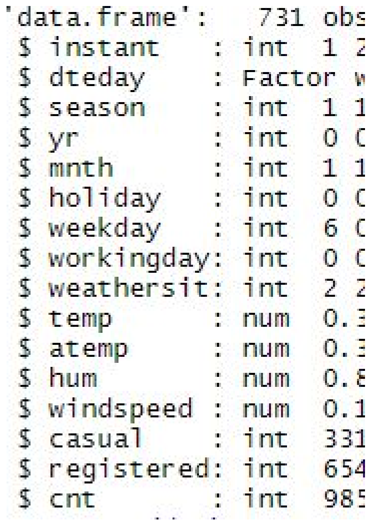
## 2.1 Pre-Processing

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. Figure 2.1.1 shows actual data formats and Figure 2.1.2 shows the formatted data.



```
instant          int64
dteday          object
season           int64
yr               int64
mnth             int64
holiday          int64
weekday          int64
workingday       int64
weathersit       int64
temp           float64
atemp          float64
hum            float64
windspeed      float64
casual           int64
registered       int64
cnt              int64
dtype: object
```

2.1.1



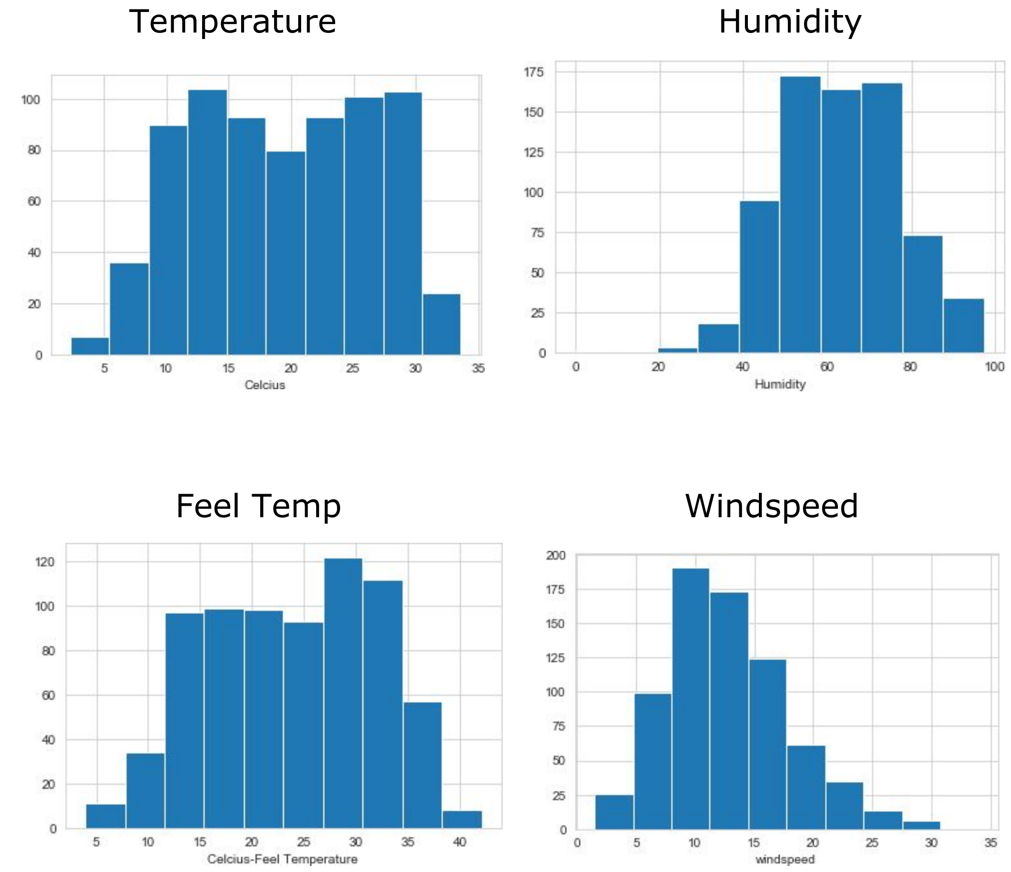
```
'data.frame':   731 obs
 $ instant   : int  1 2
 $ dteday    : Factor w
 $ season    : int  1 1
 $ yr        : int  0 0
 $ mnth      : int  1 1
 $ holiday   : int  0 0
 $ weekday   : int  6 0
 $ workingday: int  0 0
 $ weathersit: int  2 2
 $ temp      : num  0.3
 $ atemp     : num  0.3
 $ hum       : num  0.8
 $ windspeed : num  0.1
 $ casual    : int  331
 $ registered: int  654
 $ cnt       : int  985
```

2.1.2

## 2.2 Distribution of continuous variables

It can be observed from the below histograms is that temperature and feel temperature are normally distributed, whereas the variables windspeed and humidity are slightly skewed.

The skewness is likely because of the presence of outliers and extreme data in those variables.



Temperature

Humidity

Feel Temp

Windspeed

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:

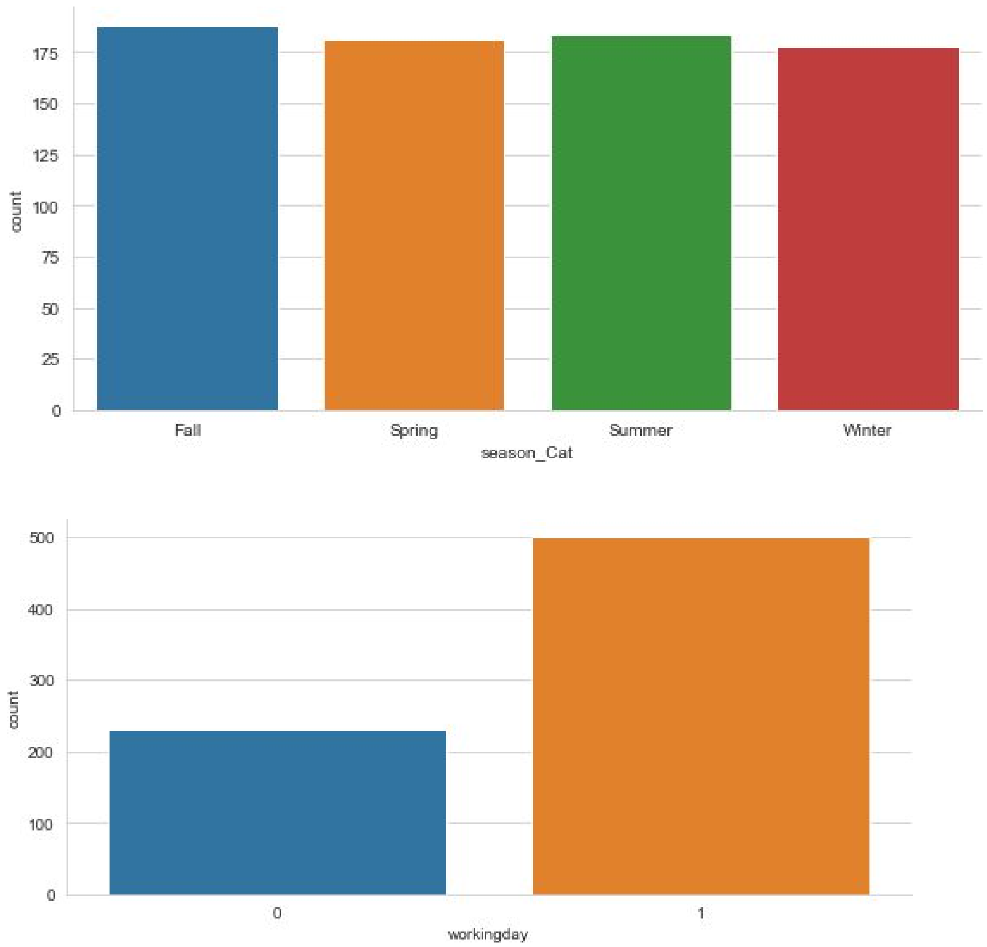


Fig 2.2: Distribution of categorical variables using bar plots

## 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.

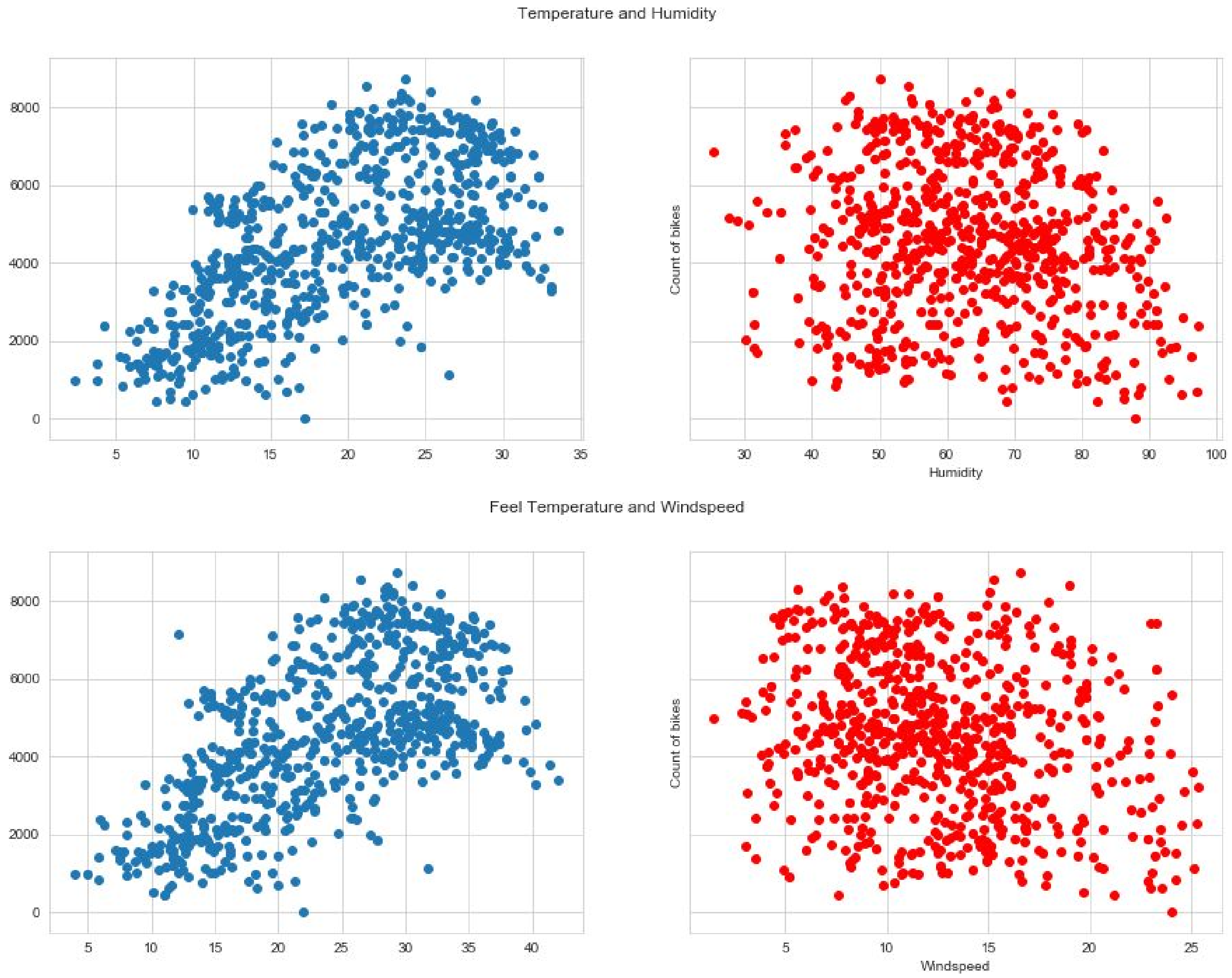


Fig 2.3: Scatter plot for continuous variables

### 2.5 : Detection of outliers:

Outliers are detected using boxplots. Below figure illustrates the boxplots for all the continuous variables.

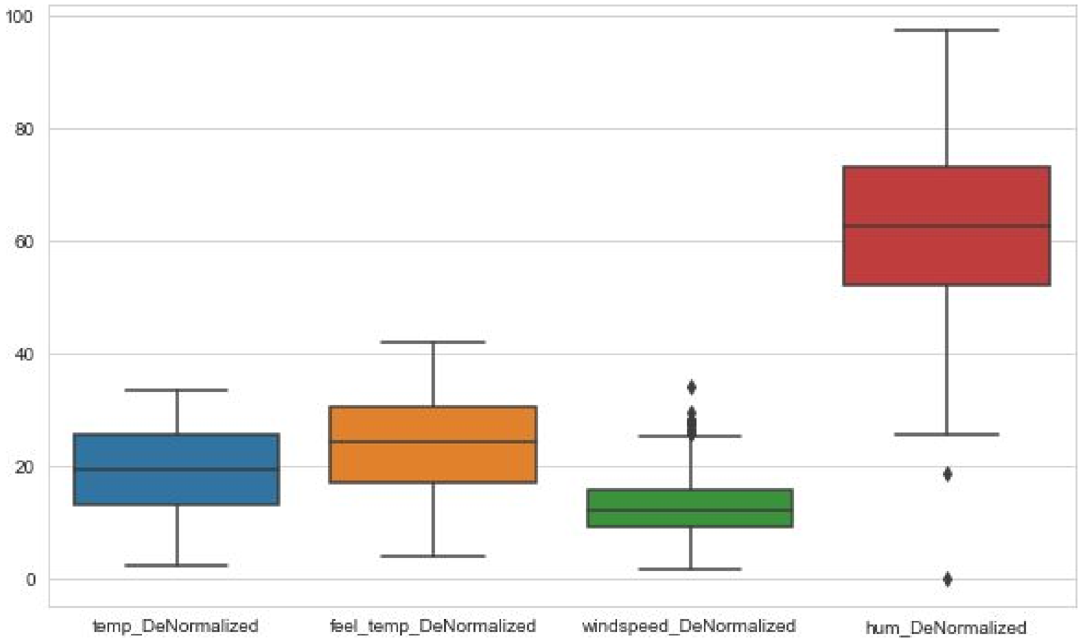


Fig 2.4: Boxplot of continuous variables

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:

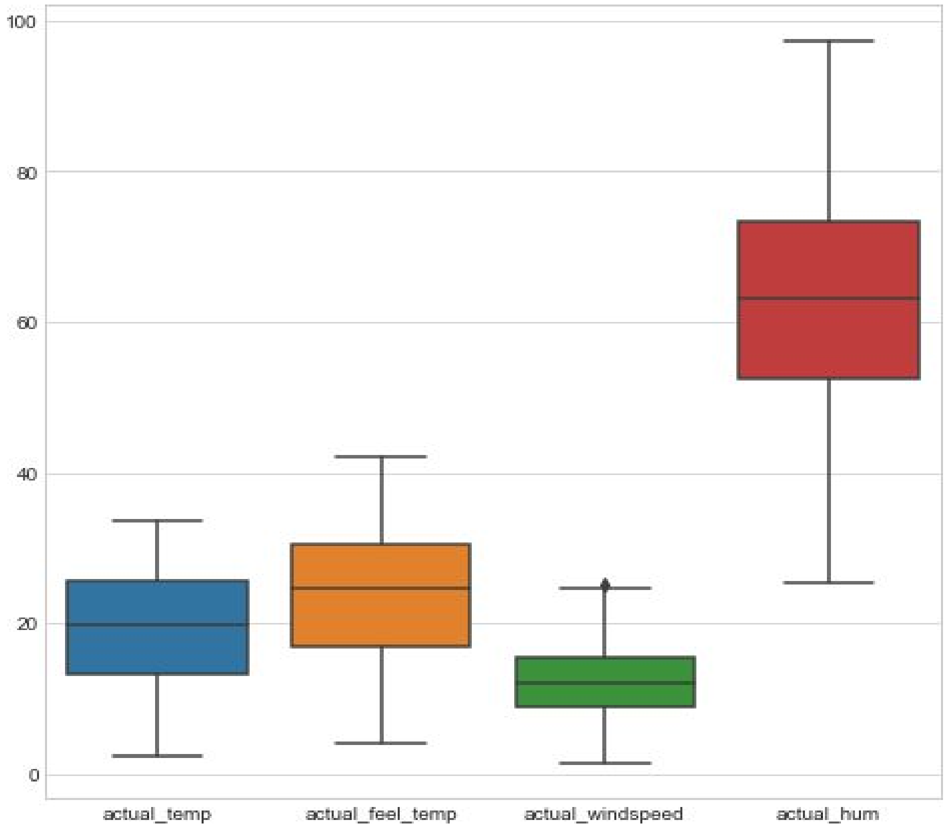


Fig 2.5: Boxplot of continuous variables after removal of outliers

It can be observed from the distribution of Windspeed and humidity after removal of outliers, is that data is not skewed as much as before the removal of outliers. The figure shown below illustrates the distribution of continuous variables using histograms.

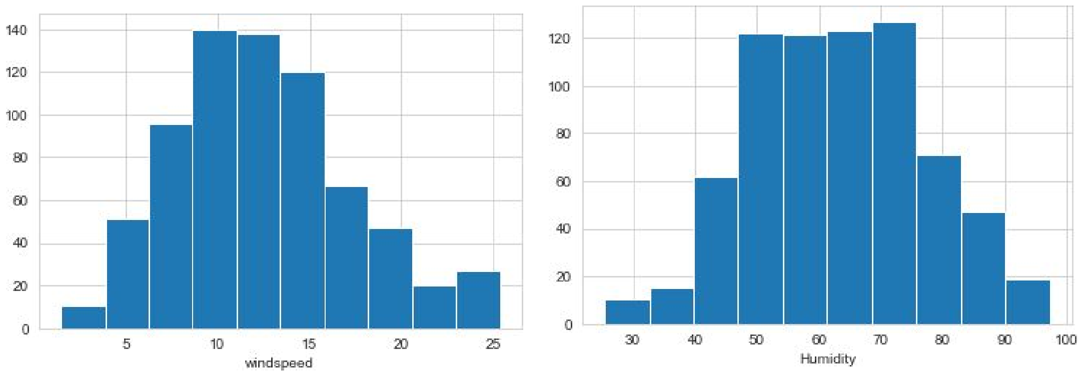


Fig 2.6: Distribution of numerical data using histograms after removal of outliers

## 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.

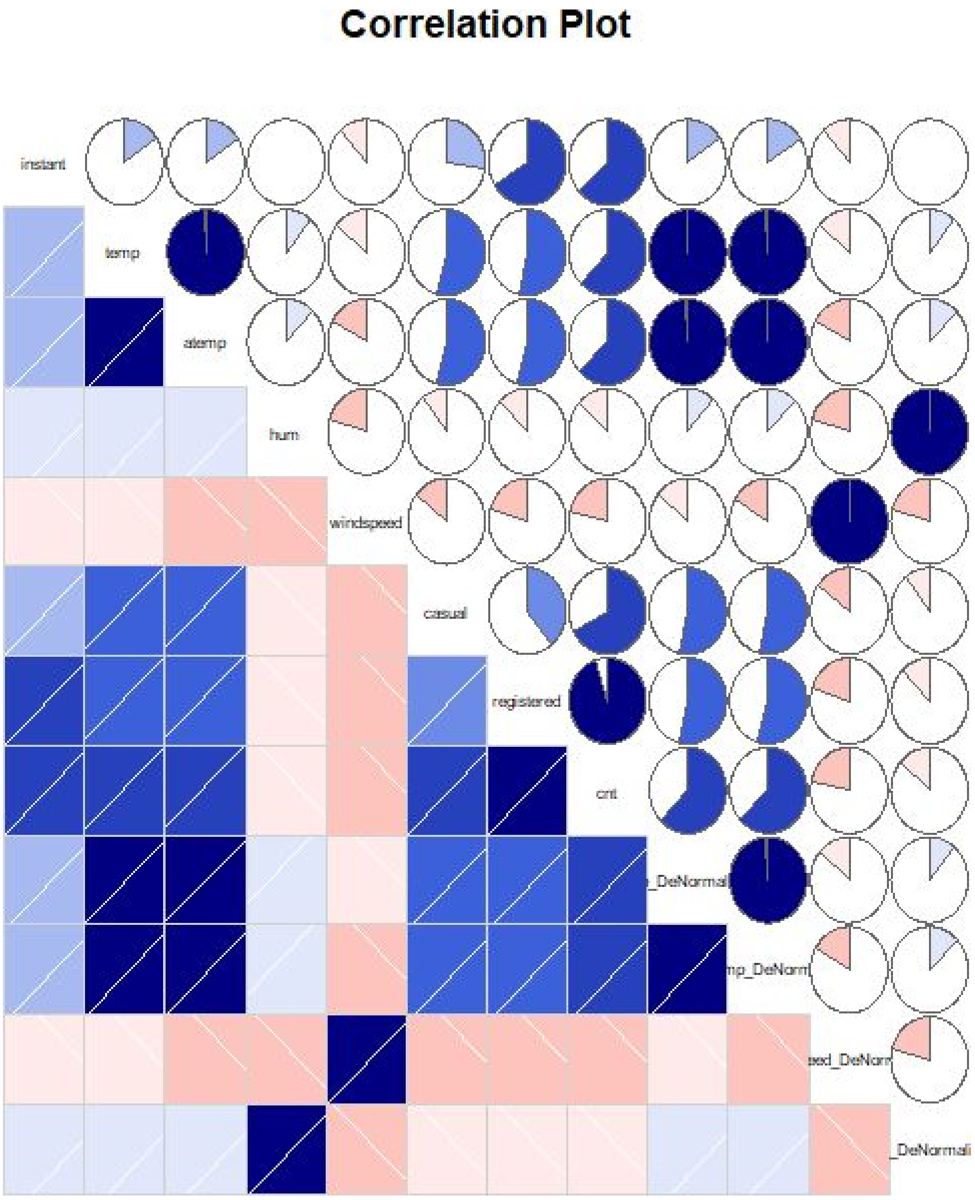


Fig 2.7: Correlation plot of all the variables

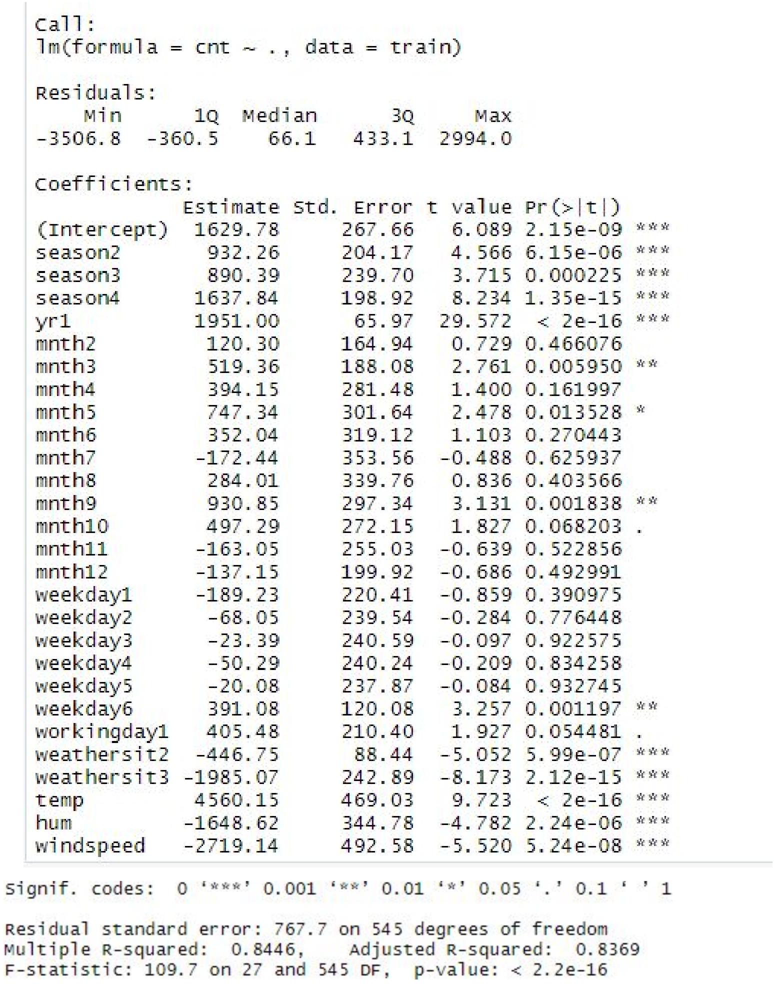
# Chapter 3: Modelling

## 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 Percentage Error (MAPE).

## 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.



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

Residuals:
    Min      1Q  Median      3Q     Max 
-3506.8  -360.5    66.1   433.1  2994.0 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1629.78     267.66   6.089 2.15e-09 ***
season2       932.26     204.17   4.566 6.15e-06 ***
season3       890.39     239.70   3.715 0.000225 ***
season4      1637.84     198.92   8.234 1.35e-15 ***
yr1          1951.00      65.97  29.572  < 2e-16 ***
mnth2         120.30     164.94   0.729 0.466076    
mnth3         519.36     188.08   2.761 0.005950 ** 
mnth4         394.15     281.48   1.400 0.161997    
mnth5         747.34     301.64   2.478 0.013528 *  
mnth6         352.04     319.12   1.103 0.270443    
mnth7        -172.44     353.56  -0.488 0.625937    
mnth8         284.01     339.76   0.836 0.403566    
mnth9         930.85     297.34   3.131 0.001838 ** 
mnth10        497.29     272.15   1.827 0.068203 .  
mnth11       -163.05     255.03  -0.639 0.522856    
mnth12       -137.15     199.92  -0.686 0.492991    
weekday1     -189.23     220.41  -0.859 0.390975    
weekday2      -68.05     239.54  -0.284 0.776448    
weekday3      -23.39     240.59  -0.097 0.922575    
weekday4      -50.29     240.24  -0.209 0.834258    
weekday5      -20.08     237.87  -0.084 0.932745    
weekday6      391.08     120.08   3.257 0.001197 ** 
workingday1   405.48     210.40   1.927 0.054481 .  
weathersit2  -446.75      88.44  -5.052 5.99e-07 ***
weathersit3 -1985.07     242.89  -8.173 2.12e-15 ***
temp         4560.15     469.03   9.723  < 2e-16 ***
hum         -1648.62     344.78  -4.782 2.24e-06 ***
windspeed   -2719.14     492.58  -5.520 5.24e-08 ***
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 767.7 on 545 degrees of freedom
Multiple R-squared:  0.8446,	Adjusted R-squared:  0.8369 
F-statistic: 109.7 on 27 and 545 DF,  p-value: < 2.2e-16
```

As you can see the Adjusted R-squared value, we can explain 83.69% 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 can 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.

MAPE of this multiple linear regression model is 16.31%. Hence the accuracy of this model is 83.69%. 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. The MAPE for this decision tree is 28.31%. Hence the accuracy for this model is 71.69%.

### 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. Using random forest, the MAPE was found to be 22.39%. Hence the accuracy is 77.61%.

# 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 Percentage 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.

```
MAPE = function(actual, pred){
  print(mean(abs((actual - pred)/actual)) * 100)
}
```

**Linear Regression Model : MAPE = 12.17**

**Decision Tree: MAPE = 28.31**
**Random Forest: MAPE = 22.39**

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

**Chapter 6: R code**

```
#Clean the environment
rm(list = ls())

#rm(day2)

#Set working directory
setwd("G:/RPrac")

#Load the libraries
libraries = c("plyr","dplyr",
  "ggplot2","rpart","dplyr","DMwR","randomForest","usdm","corrgram","DataCombine
  ")
lapply(X = libraries,require, character.only = TRUE)
rm(libraries)

#sapply(day, function(x) sum(length(which(is.na(x)))))

#Fetch csv file data into R
day = read.csv(file = "day.csv", header = T, sep = ",")
#day2= read.csv(file = "day.csv", header = T, sep = ",")

#Converting the data to its required format for data exploration
day$dteday = as.character(day$dteday)
day$season = as.factor(day$season)
day$yr = as.factor(day$yr)
day$mnth = as.factor(day$mnth)
day$holiday = as.factor(day$holiday)
day$weekday = as.factor(as.character(day$weekday))
day$workingday = as.factor(as.character(day$workingday))
day$weathersit = as.factor(day$weathersit)


##DATA EXPLORATION##
#Check the distribution of categorical Data using bar graph
day$season_Factored = factor(x = day$season, levels = c(1,2,3,4), labels =
  c("Spring","Summer","Fall","Winter"))
day$yr_Factored = factor(x = day$yr, levels = c(0,1), labels = c("2011","2012"))
day$holiday_Factored = factor(x = day$holiday, levels = c(0,1), labels = c("Working
  day","Holiday"))
day$weathersit_Factored = factor(x = day$weathersit, levels = c(1,2,3,4),
                              labels = c("Clear","Cloudy/Mist","Rain/Snow/Fog","Heavy
  Rain/Snow/Fog"))

Season_bar = ggplot(data = day, aes(x = season_Factored))+ geom_bar()+
  ggtitle("Count of Season" )
Weather_bar = ggplot(data = day, aes(x = weathersit_Factored)) + geom_bar() +
  ggtitle("Count of Weather")
Holoday_bar = ggplot(data = day, aes(x = holiday_Factored)) + geom_bar() +
  ggtitle("Count of Holiday")
WDay_bar = ggplot(data = day, aes(x = workingday)) + geom_bar() +
  ggtitle("Count of Working day")

gridExtra::grid.arrange(Season_bar,Weather_bar)
gridExtra::grid.arrange(Holoday_bar,WDay_bar)

#Data is normalised.Converting data to Check the distribution of numerical data
  using histogram.
day$temp_DeNormalized <- day$temp*39
day$feel_temp_DeNormalized <- day$atemp*50
day$windspeed_DeNormalized <- day$windspeed*67
day$hum_DeNormalized = day$hum * 100

Temp_hist = ggplot(data = day, aes(x =temp_DeNormalized)) +
  ggtitle("Temperature Distribution") + geom_histogram(bins = 50)
Hum_hist = ggplot(data = day, aes(x =hum_DeNormalized)) + ggtitle("Humidity
  Distribution") + geom_histogram(bins = 25)
Feel_hist = ggplot(data = day, aes(x =feel_temp_DeNormalized)) + ggtitle("Feel
  Temperature Distribution") + geom_histogram(bins = 25)
Wind_hist = ggplot(data = day, aes(x =windspeed_DeNormalized)) +
  ggtitle("Windspeed Distribution") + geom_histogram(bins = 25)

gridExtra::grid.arrange(Temp_hist,Hum_hist,Feel_hist,Wind_hist)

#Check the distribution of numerical data using scatterplot
scat1 = ggplot(data = day, aes(x =temp_DeNormalized, y = cnt)) +
  ggtitle("Distribution of Temperature") + geom_point() + xlab("Temperature") +
  ylab("Bike COunt")
scat2 = ggplot(data = day, aes(x =hum_DeNormalized, y = cnt)) +
  ggtitle("Distribution of Humidity") + geom_point(color="red") + xlab("Humidity") +
  ylab("Bike COunt")
scat3 = ggplot(data = day, aes(x =feel_temp_DeNormalized, y = cnt)) +
  ggtitle("Distribution of Feel Temperature") + geom_point() + xlab("Feel
  Temperature") + ylab("Bike COunt")
scat4 = ggplot(data = day, aes(x =windspeed_DeNormalized, y = cnt)) +
  ggtitle("Distribution of Windspeed") + geom_point(color="red") +
  xlab("Windspeed") + ylab("Bike COunt")
gridExtra::grid.arrange(scat1,scat2,scat3,scat4,ncol=2)


#Check for outliers in data using boxplot
cnames =
  colnames(day[,c("temp_DeNormalized","feel_temp_DeNormalized","windspeed_De
  Normalized","hum_DeNormalized")])
for (i in 1:length(cnames))
{
  assign(paste0("gn",i), ggplot(aes_string(y = cnames[i]), data = day)+
         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])+
          ggtitle(paste("Box plot for",cnames[i])))
}


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

str(day)
#Remove outliers in Windspeed and humidity
val = day[,23][day[,23] %in% boxplot.stats(day[,23])$out]
day = day[which(!day[,23] %in% val),]

val = day[,24][day[,24] %in% boxplot.stats(day[,24])$out]
day = day[which(!day[,24] %in% val),]

#Check for multicollinearity using VIF
df = day[,c("instant","temp","atemp","hum","windspeed")]
vifcor(df)

#Check for collinearity using corelation graph
corrgram(day, order = F, upper.panel=panel.pie, text.panel=panel.txt, main =
  "Correlation Plot")


#Remove the unwanted variables
df2=day
day <- subset(day, select =
  -c(holiday,instant,dteday,atemp,casual,registered,temp_DeNormalized,feel_temp_D
  eNormalized,windspeed_DeNormalized,

  hum_DeNormalized,season_Factored,yr_Factored,holiday_Factored,weathersit_Fact
  ored))

rmExcept(keepers = "day")
########################################DECISION
  TREE########################################
#MAPE: 28.31%
#Accuracy: 71.69%

#Divide the data into train and test
set.seed(123)
train_index = sample(1:nrow(day), 0.8 * nrow(day))
train = day[train_index,]
test = day[-train_index,]

#rpart for regression
dt_model = rpart(cnt ~ ., data = train, method = "anova")
```

```
#Predict the test cases
dt_predictions = predict(dt_model, test[,-10])

#Create dataframe for actual and predicted values
df = data.frame("actual"=test[,10], "pred"=dt_predictions)
head(df)


#calculate MAPE
MAPE = function(actual, pred){
  print(mean(abs((actual - pred)/actual)) * 100)
}
MAPE(test[,10], dt_predictions)

## RANDOM FOREST ##
#MAPE: 22.39%
#Accuracy: 77.61%

#Train the data using random forest
rf_model = randomForest(cnt~., data = train, ntree = 500)

#Predict the test cases
rf_predictions = predict(rf_model, test[,-10])

#Create dataframe for actual and predicted values
df = cbind(df,rf_predictions)
```