PREDICTION OF BIKE RENTAL COUNT

ARJUN PRAKASH

23-10-2019

1. Introduction

1.1 Problem Statement

The Project aims to predict the count of bike rentals based on seasonal and environmental settings. By the count prediction it would help to accommodate the bikes required on the daily basis and the preparation during the peak periods where the demands of the bikes are high.

1.2 Data Set

The goal is to build the regression models which will predict the number of bikes used based on the environmental and seasonal behaviors. Below is the 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	1	1/1/2011	1	0	1	0	6	0	2
2	2	1/2/2011	1	0	1	0	0	0	2
3	3	1/3/2011	1	0	1	0	1	1	1
4	4	1/4/2011	1	0	1	0	2	1	1
5	5	1/5/2011	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

In the below Table we have the following 13 variables, using which we have to predict the count of bikes :

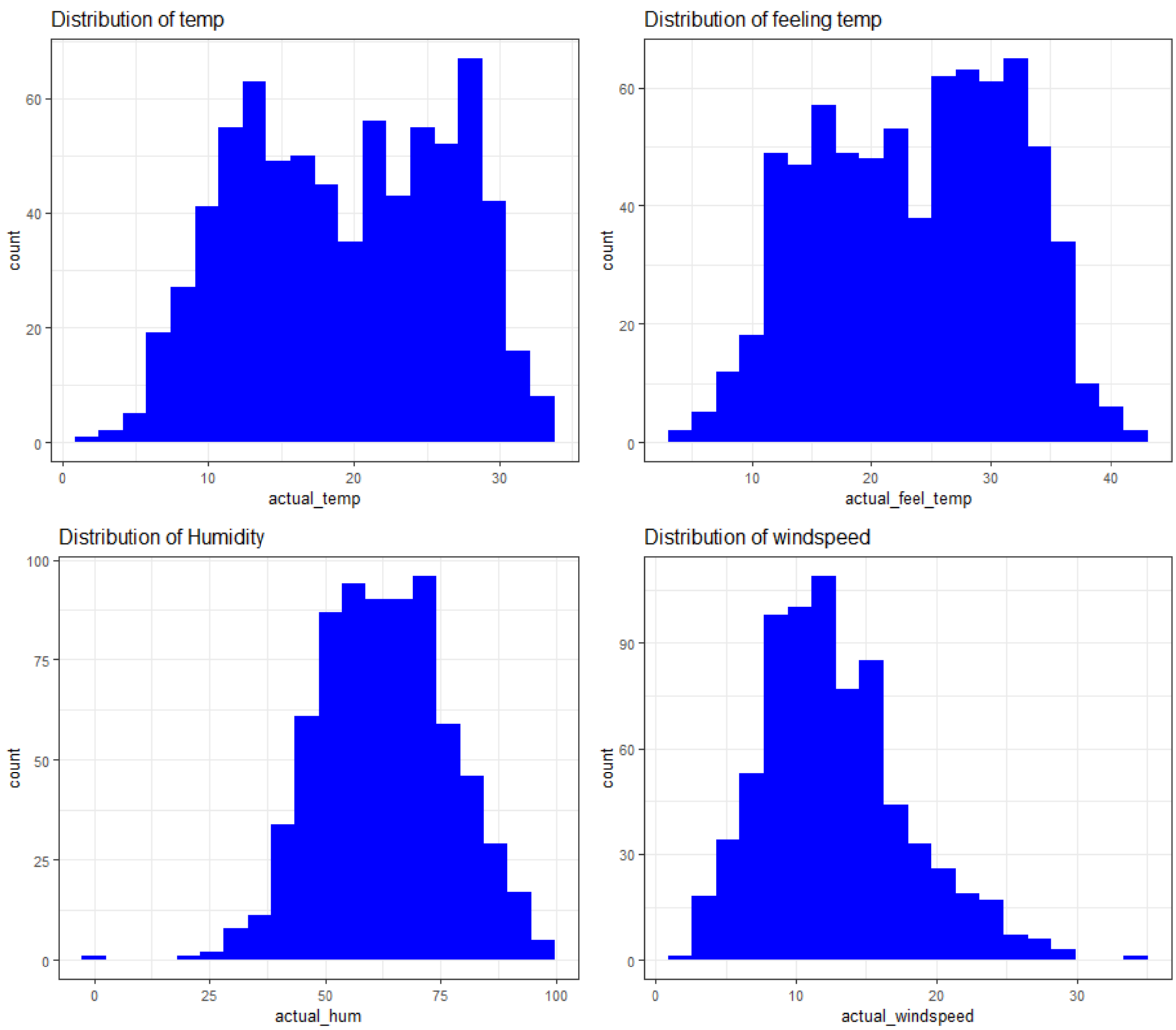
Sl.No	Variables
1	Instant
2	Dteday
3	Season
4	Yr
5	Month
6	Holiday
7	Weekday
8	Workingday
9	Weathersit
10	Temp
11	Atemp
12	Hum
13	windspeed

Table 1.3: Predictor variables

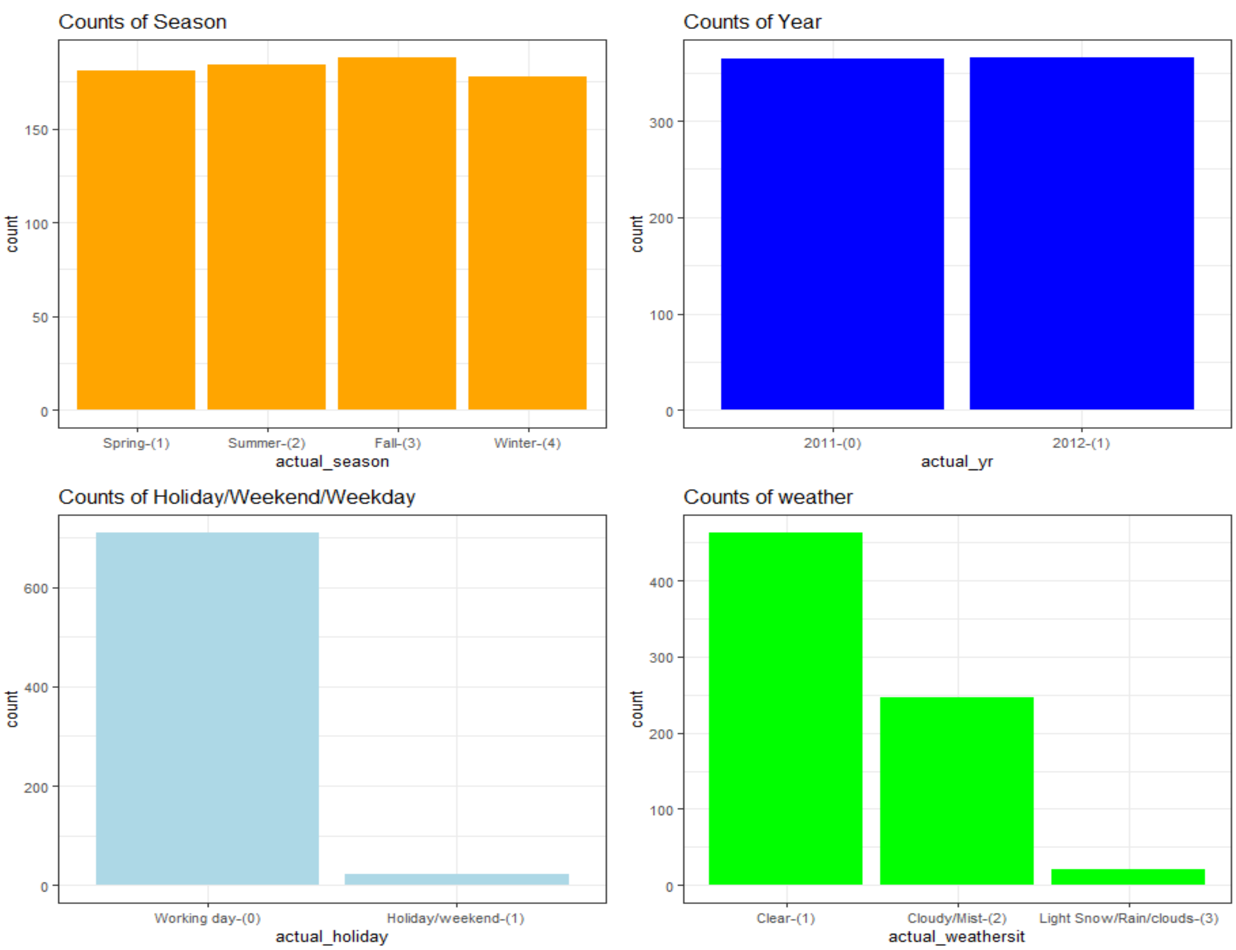
2. Methodology

2.1 Data Exploration

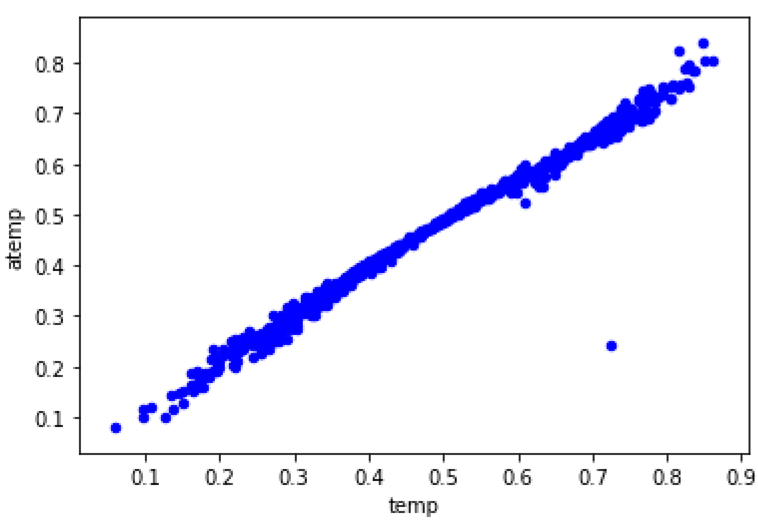
2.1.1 Distribution Of Continuous and Categorical Variables using Univariate Analysis



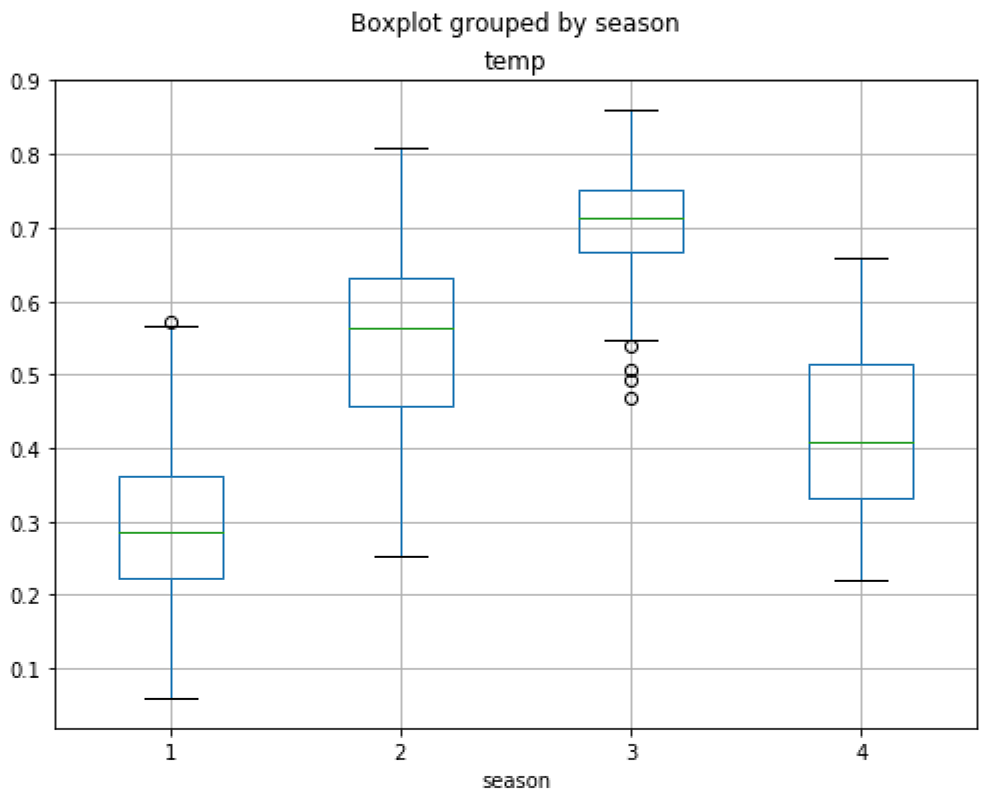
2.1.2 Distribution Of Continuous and Categorical Variables using Univariate Analysis



2.1.3 Distribution of Continuous Variables Bivariate Analysis



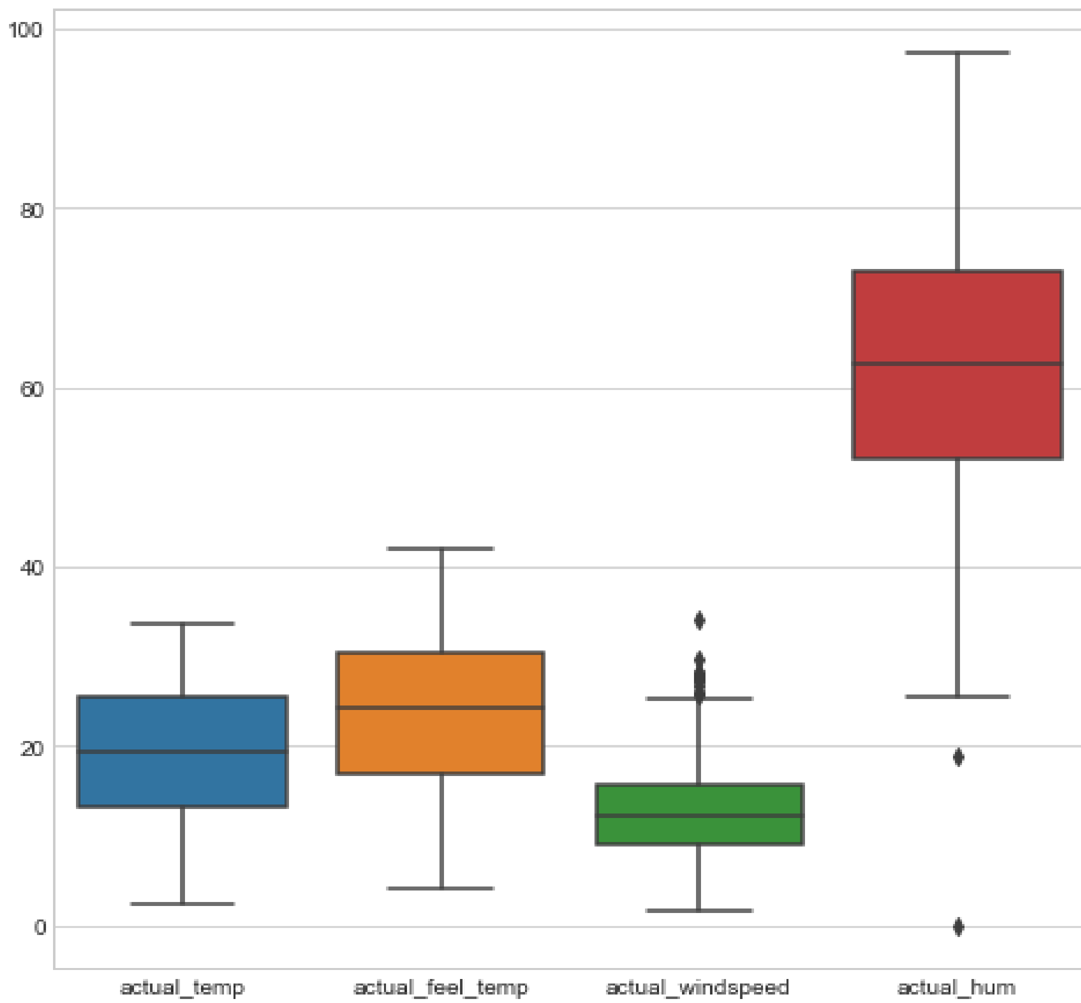
2.1.4 Distribution of Categorical and Continous Variables Bivariate Analysis



2.2 Pre-Processing

2.2.1 Detection of Outliers

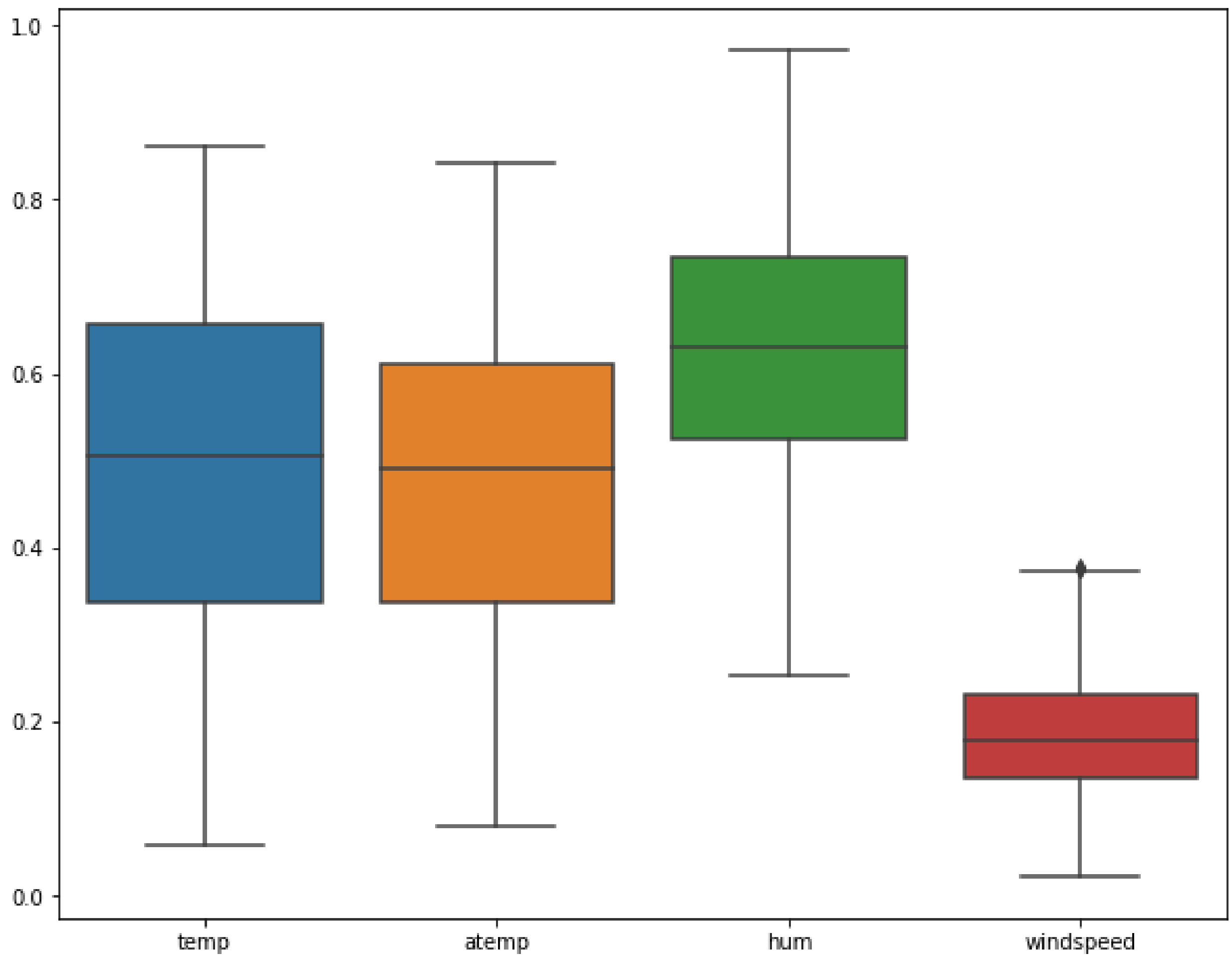
Outliers: An outlier is a data point in a data set that is distant from all other observations. A data point that lies outside the overall distribution of the dataset Outliers are detected using boxplots. Below figure illustrates the boxplots for all the continuous variables.



2.2.2 Removal of Outliers

Outliers can be removed using the Boxplot stats method, wherein the Inter Quartile Range (IQR) is calculated and the minimum and maximum values 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:



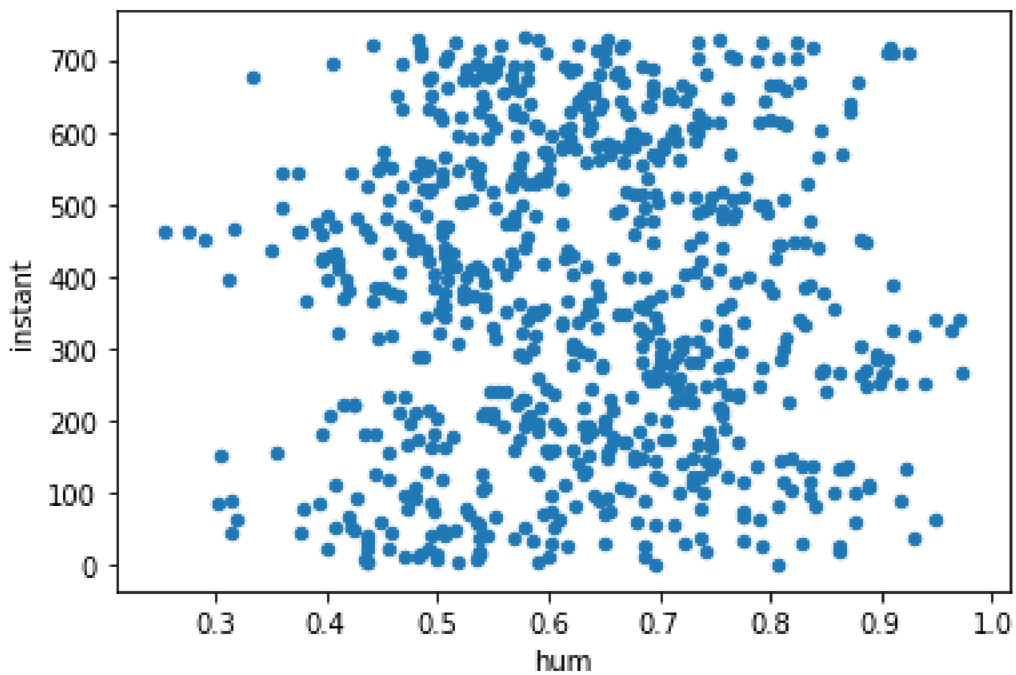
Boxplot of continuous variables after removal of outliers

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.

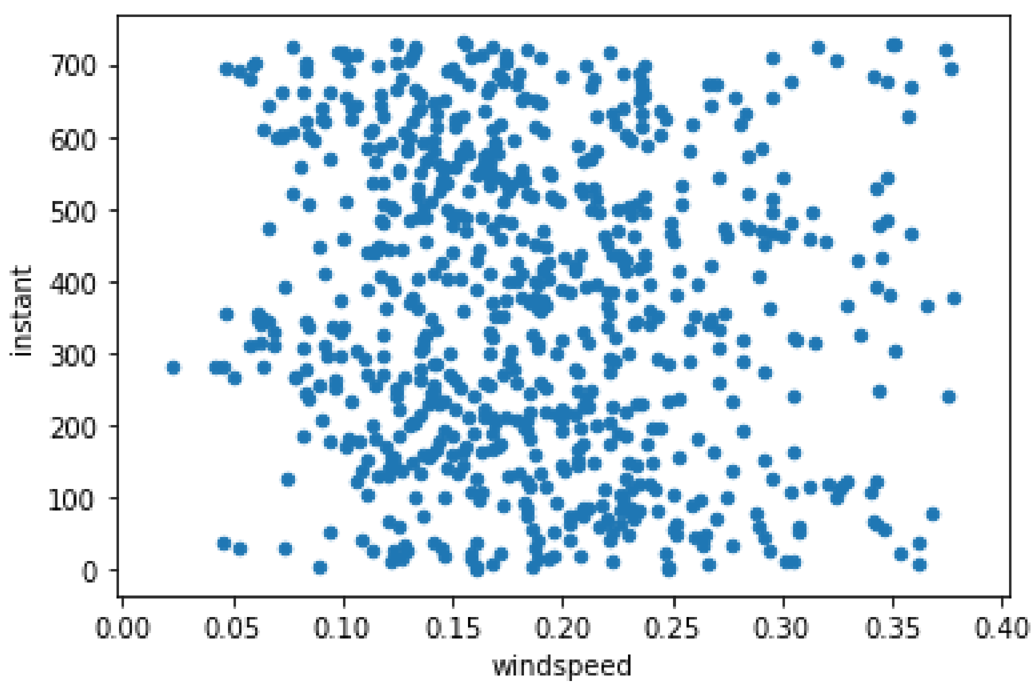
The figure shown below illustrates the distribution of continuous variables using

Scatterplots after removal of outliers

Humidity



Windspeed



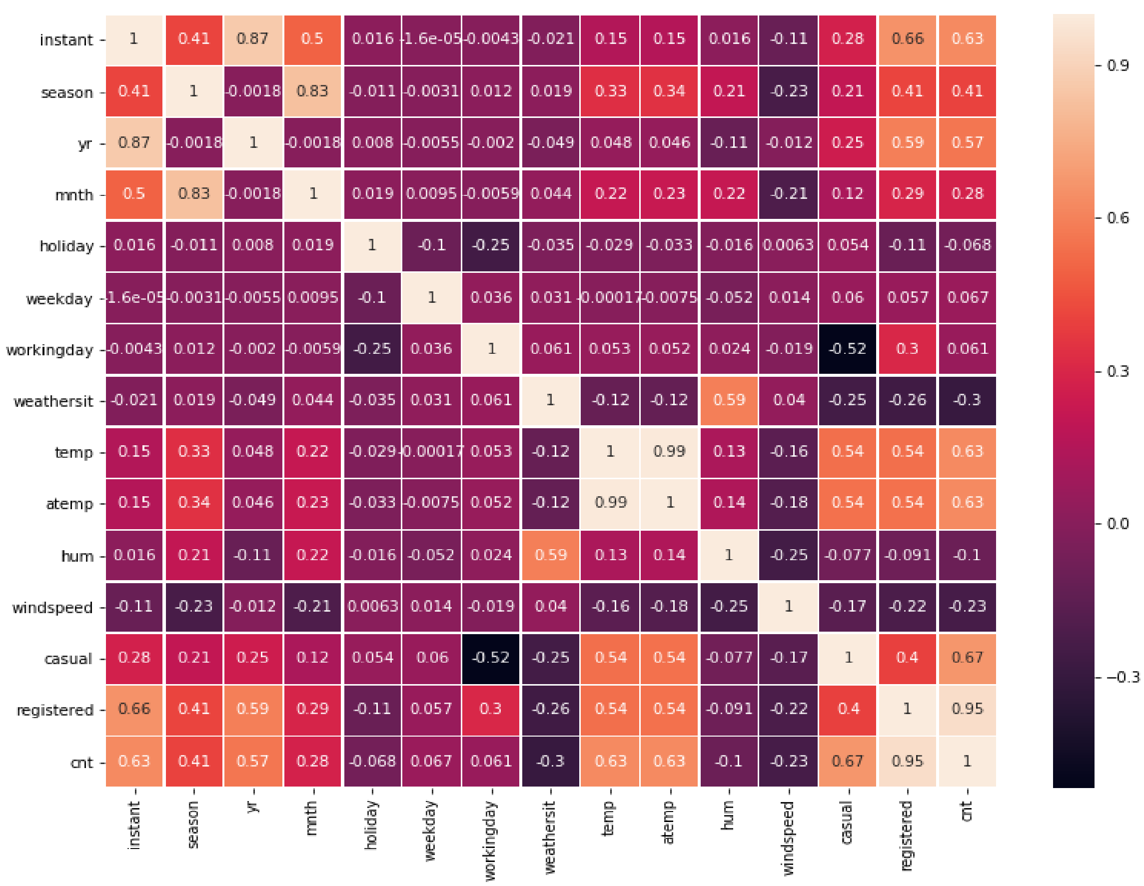
2.2.3 Detection of multicollinearity using VIF Method and Correlation Graph

Multicollinearity (also collinearity) is a phenomenon in which two or more predictor variables (Independent variables) in a regression model are highly correlated i.e one can be linearly predicted from the others with a substantial degree of accuracy.

	VIF_Factor	Features
0	46.4	Intercept
1	63.3	Temp
2	63.9	Atemp
3	1.1	Hum
4	1.1	Windspeed

From the above we can understand that "temp" and "atemp" have a high Variance inflation factor (VIF), they have almost same variance within the dataset.

Below is the correlation graph on all the variable to check how the variable or features are correlated between each other.



2.3 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.

From point 2.2.3 we have noticed the collinearity between the variables or features, from that we can remove some of the features.

Feature selection from Random Forest regressor model using feature_importances_

```
[('season', 0.11),
 ('yr', 0.3),
 ('mnth', 0.03),
 ('weekday', 0.0),
 ('weathersit', 0.01),
 ('temp', 0.0),
 ('hum', 0.02),
 ('windspeed', 0.43)]
```

3. Modelling

3.1 Model Building

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

Linear Regression

Decision Tree Regressor

Random Forest Regressor

The error metric chosen for the problem statement are,

Mean Absolute Error (MAE)

Mean Absolute Percentage Error (MAPE)

Mean Squared Error (MSE)

Root Mean Squared Error (RMSE).

3.1.1 Linear Regression model

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.

Linear Regression Model OLS (Ordinary Least Squares)

Summary:

OLS Regression Results

Dep. Variable:	cnt	R-squared (uncentered):	0.966
Model:	OLS	Adj. R-squared (uncentered):	0.966
Method:	Least Squares	F-statistic:	1611.
Date:	Fri, 25 Oct 2019	Prob (F-statistic):	0.00
Time:	16:29:10	Log-Likelihood:	-4717.4
No. Observations:	573	AIC:	9455.
Df Residuals:	563	BIC:	9498.
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
season	612.4684	71.135	8.610	0.000	472.745	752.191
yr	2156.4156	76.055	28.353	0.000	2007.029	2305.802
mnth	-55.5089	22.495	-2.468	0.014	-99.693	-11.325
holiday	-289.0369	242.120	-1.194	0.233	-764.605	186.531
weekday	109.5736	18.837	5.817	0.000	72.574	146.573
workingday	205.6906	84.076	2.446	0.015	40.550	370.831
weathersit	-734.6931	96.591	-7.606	0.000	-924.415	-544.971
temp	5291.5533	220.588	23.988	0.000	4858.277	5724.829
hum	420.5327	312.126	1.347	0.178	-192.540	1033.606
windspeed	-664.6212	467.121	-1.423	0.155	-1582.134	252.892

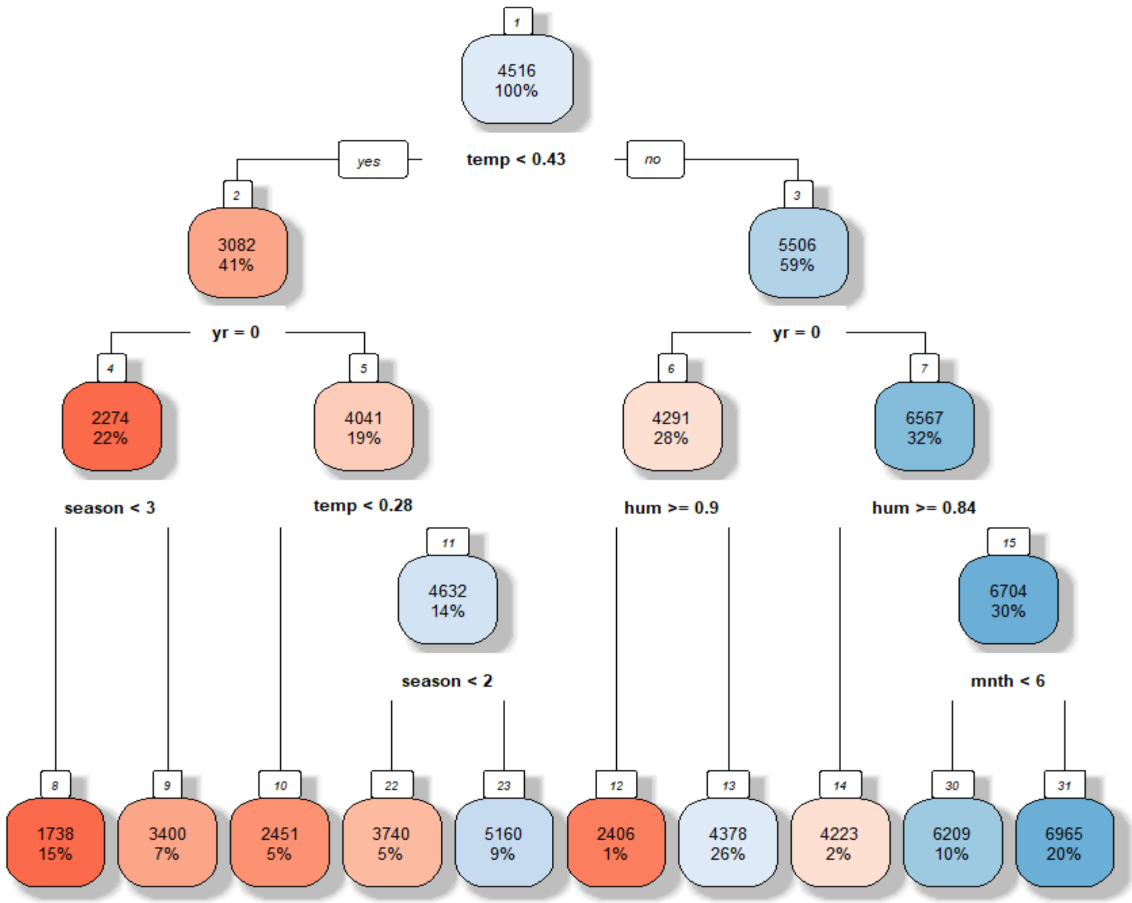
Omnibus:	68.588	Durbin-Watson:	2.015
Prob(Omnibus):	0.000	Jarque-Bera (JB):	134.378
Skew:	-0.705	Prob(JB):	6.61e-30
Kurtosis:	4.908	Cond. No.	105.

Mean absolute percentage error(MAPE) - 18.998,
Mean absolute error (MAE) - 649.937
Mean squared error (MSE) - 729478.496
Root mean squared error (RMSE) - 854.095
Model Accuracy - 81.002 %

3.1.2 Decision Tree:

A Decision Tree can be used to visually and explicitly represent decision making.

Using the decision tree, we can predict the model count



Mean absolute percentage error (MAPE) - 19.631
Mean absolute error (MAE) - 630.974
Mean squared error (MSE) - 787495.337
Root mean squared error (RMSE) - 887.409
Model Accuracy -80.369 %

3.1.3 Random Forest:

Using Classification for prediction analysis in this case.

Mean absolute percentage error (MAPE) - 15.579
Mean absolute error (MAE) - 513.547
Mean squared error (MSE) - 571939.005
Root mean squared error (RMSE) - 756.266
Model Accuracy - 84.421 %

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.

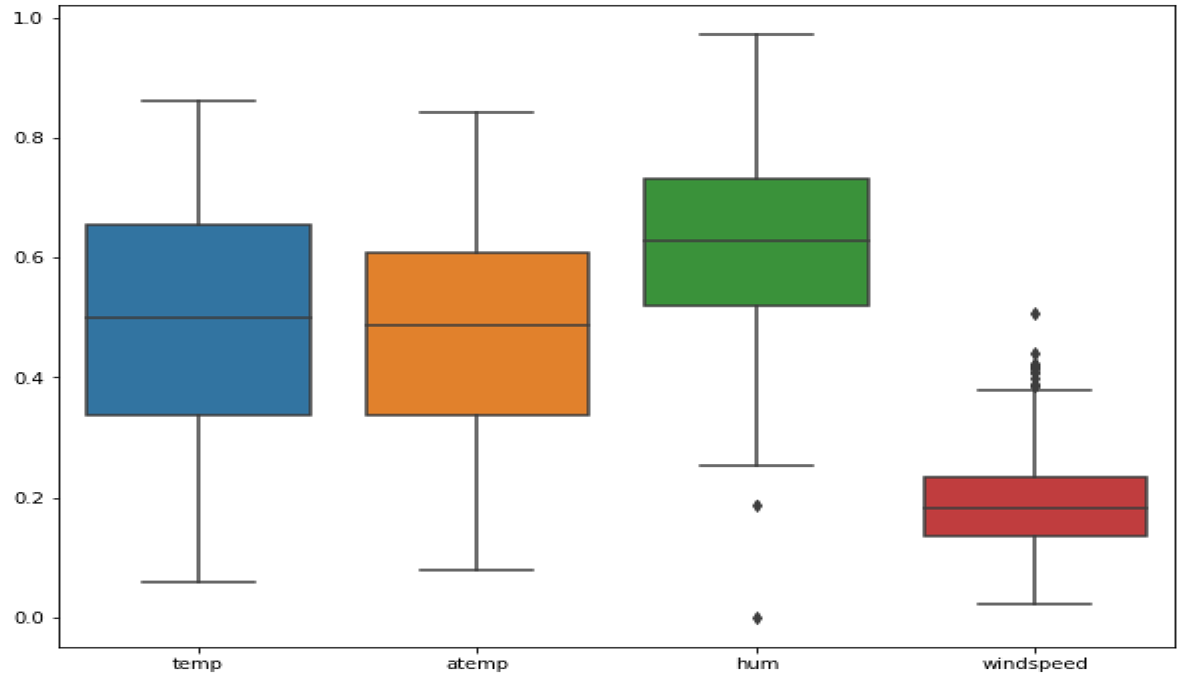
Predictive performance can be measured by comparing Predictions of the models with real values of the target variables and calculating some average error metrices like MAPE, MAE, MSE, RMSE.

	LINEAR REGRESSION MODEL	**DECISION TREE**	**RANDOM FOREST**
MAPE (Mean Absolute Percentage Error)	18.998	19.631	15.579
MAE (Mean Absolute Error)	649.937	630.974	513.547
MSE (Mean Squared Error)	729478.496	787495.337	571939.005
RMSE (Root Mean Squared Error)	854.095	887.409	756.266
ACCURACY	81.002 %	80.369 %	84.421 %

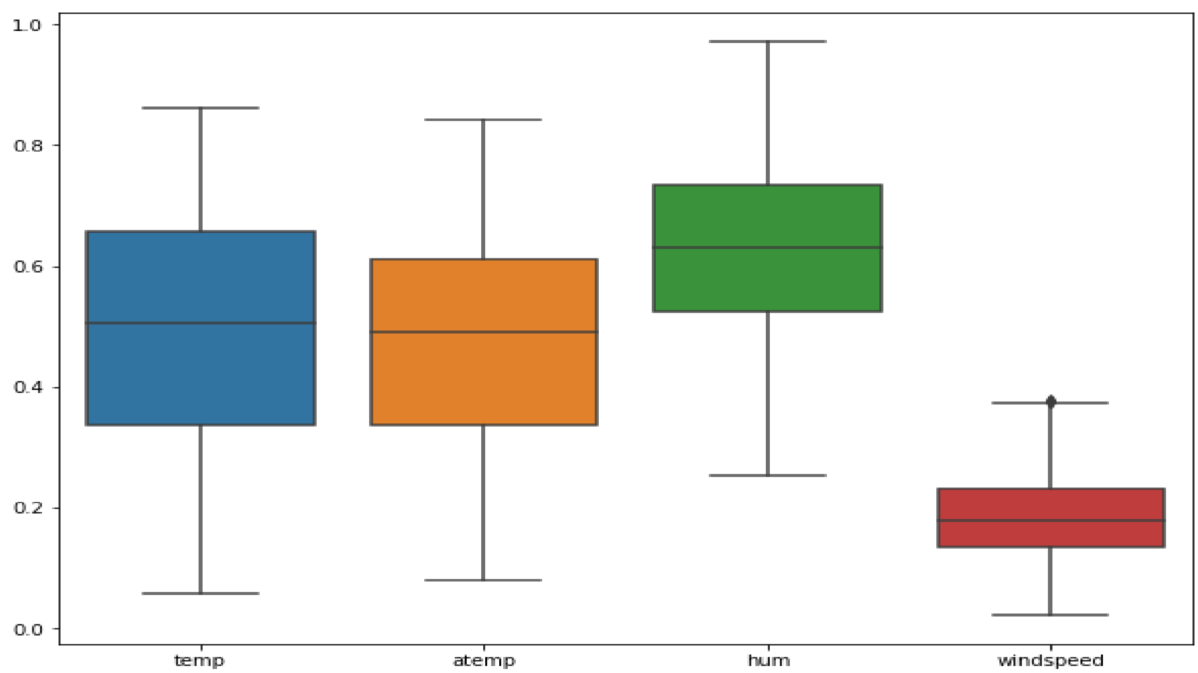
Random Forest is getting less Error Metrics compared to other models. Comparing to other models Random Forest and all the other metrics error scores are less if we are to look at the bigger picture the error scores should have been more lesser.

5. Appendix A

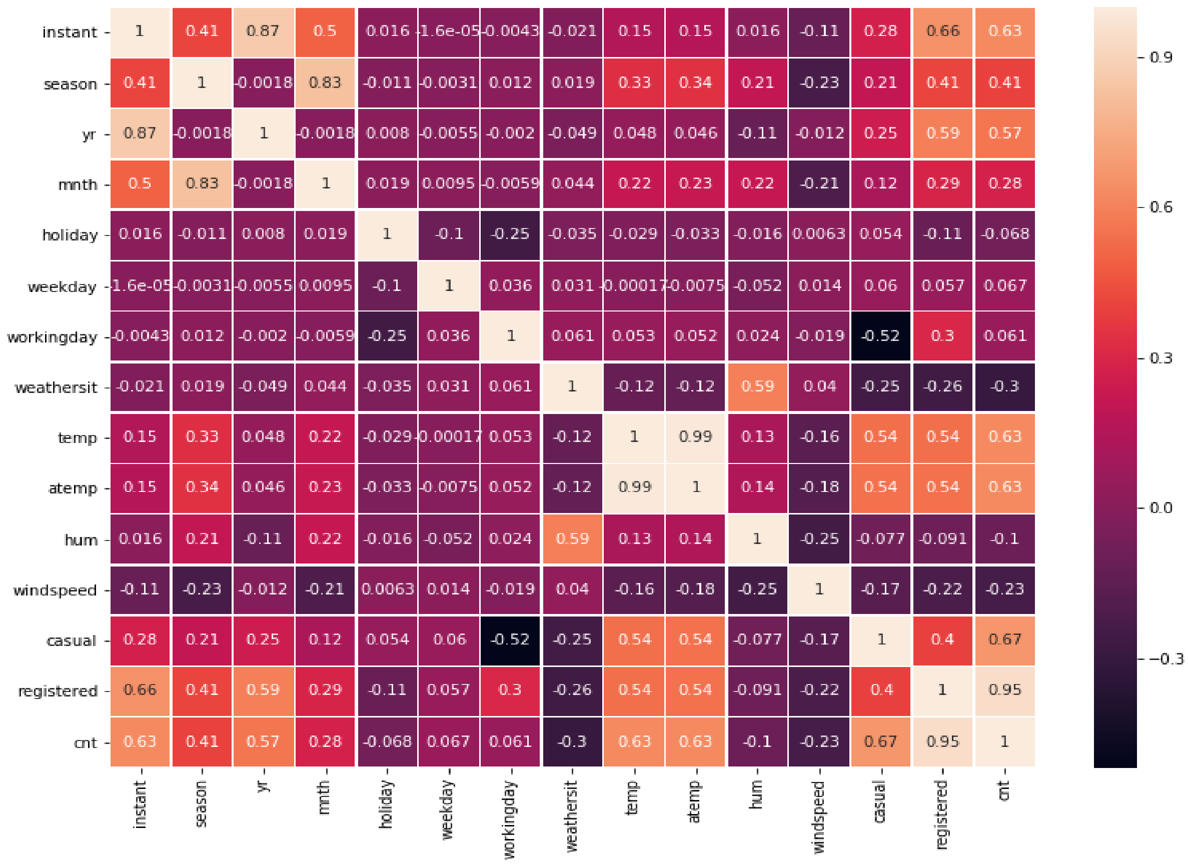
5.1Figures



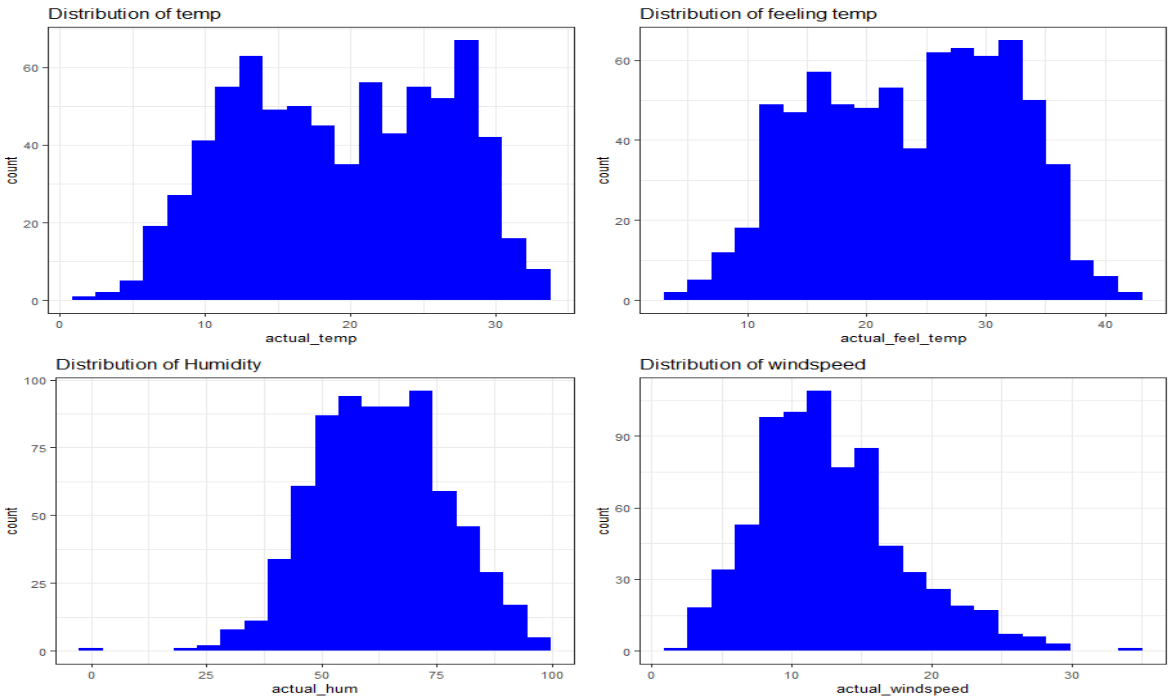
OUTLIER DETECTION



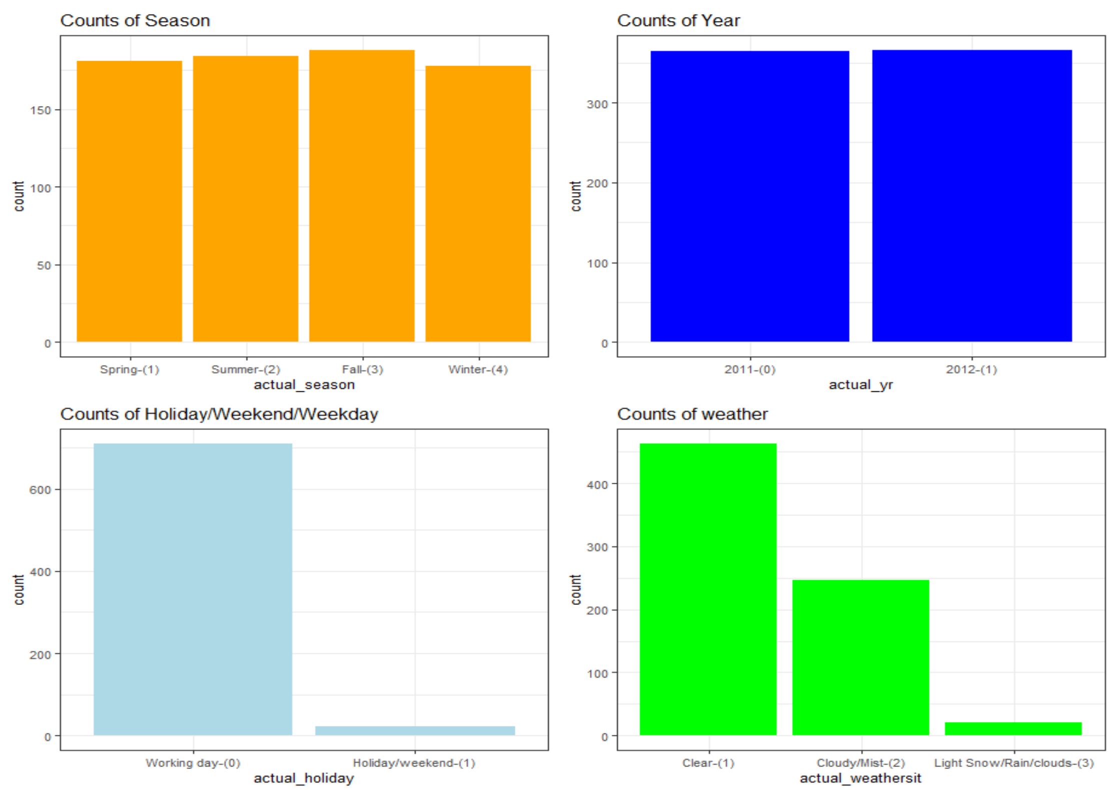
After DROPPING THE OUTLIERS



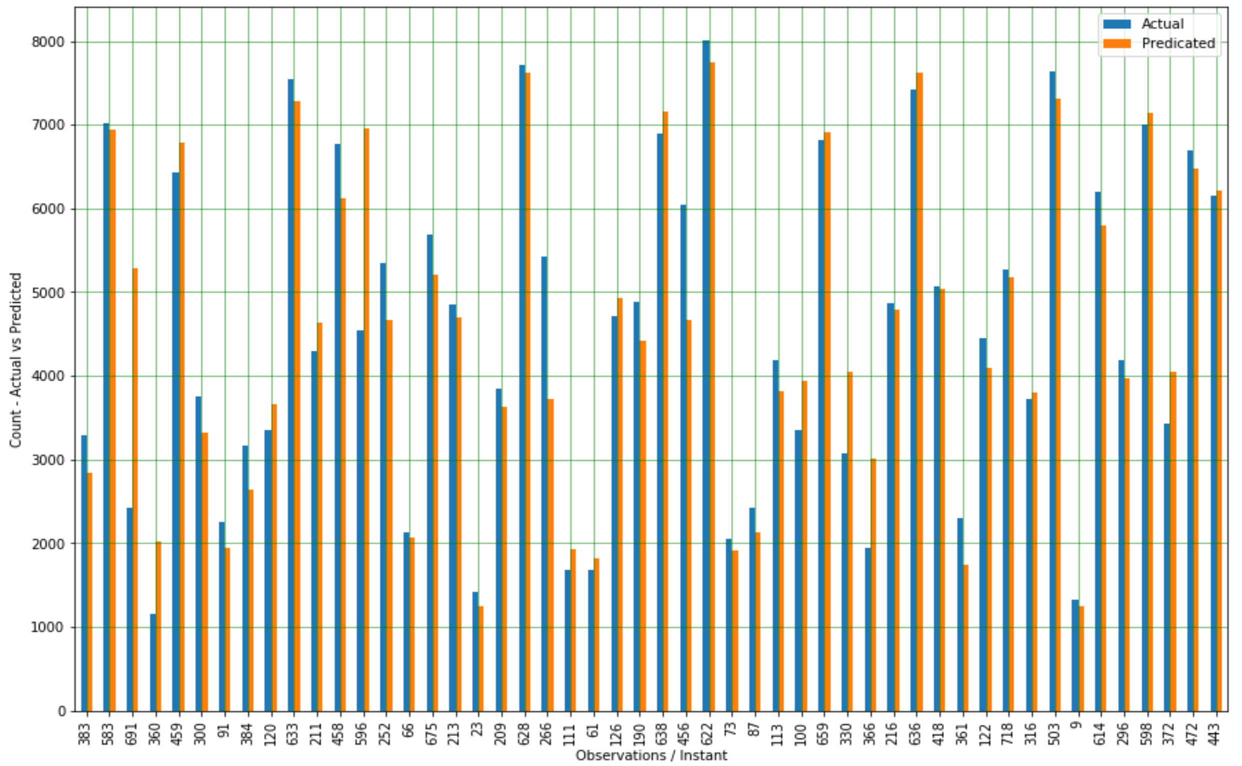
CORRELATION OF ALL PLOTS



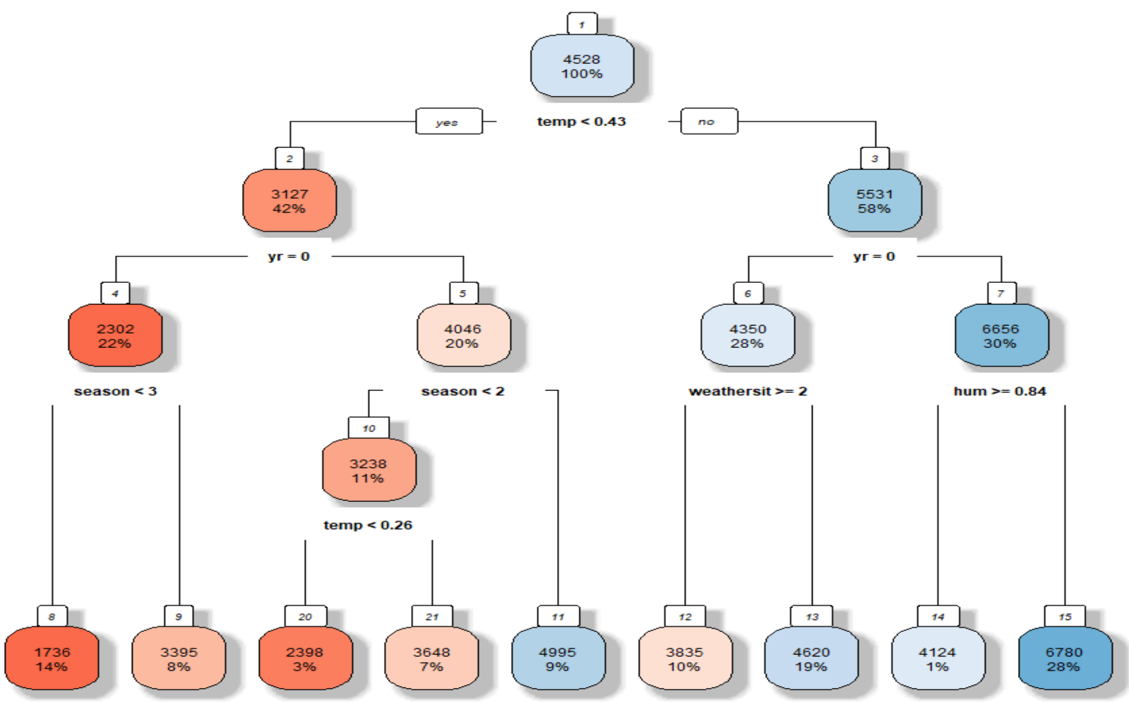
Continuous and Categorical Variables using Univariate Analysis



Continuous and Categorical Variables using Univariate Analysis



Linear Regression Model Output



DECISION TREE MODEL

7. APPENDIX

7.1 R CODE

```
#Clear the environment
rm(list = ls())
rm(library)
#Load the Libraries
library(DataCombine)
library(ggplot2)
library(gridExtra)
library(caret)
library(usdm)
library(corrgram)
library(DMwR)
library(corrplot)
library(rpart)
library(randomForest)
library(rpart.plot)

#Set Working directory
setwd("C:/Users/rnp/Desktop/Arjun/Data Science/Project2")
# Load the bike_df.cvs files
bike_df = read.csv("day.csv", header = T)
#****Explore the bike_df.csv file****#
#display the dataset
head(bike_df)
#view the dimension
dim(bike_df)
```

```
#view summary
summary(bike_df)


#view the structure of dataset
str(bike_df)
org_bike_df <- bike_df
dim(org_bike_df)

#****** Feature engineering*****#


#adding new columns by converting the categorical columns with actual values
#and, normalized continous values to actual values w.r.t given in problem statement

#Create new columns and merge to exsiting dataset
bike_df$actual_temp <-bike_df$temp*39
bike_df$actual_feel_temp <-bike_df$atemp*50
bike_df$actual_hum = bike_df$hum * 100
bike_df$actual_windspeed <-bike_df$windspeed*67

bike_df$actual_season = factor(x = bike_df$season, levels = c(1,2,3,4), labels = c("Spring-(1)","Summer-(2)","Fall-(3)","Winter-(4)"))
bike_df$actual_yr = factor(x = bike_df$yr,levels = c(0,1),labels = c("2011-(0)","2012-(1)"))
bike_df$actual_holiday = factor(x = bike_df$holiday,levels = c(0,1),labels = c("Working day-(0)","Holiday/weekend-(1)"))
bike_df$actual_weathersit = factor(x = bike_df$weathersit,levels = c(1,2,3,4),labels = c("Clear-(1)","Cloudy/Mist-(2)","Light Snow/Rain/clouds-(3)","Heavy Rain/Snow/Fog(4)"))

#check the structure of the dataset after adding more columns
```

```
str(bike_df)

#Univariate analysis to see how the data is distributed



# Continous variable
hist_plot1 = ggplot(bike_df, aes(actual_temp))+theme_bw()+geom_histogram(fill='blue',
bins = 20)+ggtitle("Distribution of temp")+theme(text = element_text(size = 10))
hist_plot2 = ggplot(bike_df,
aes(actual_feel_temp))+theme_bw()+geom_histogram(fill='blue', bins =
20)+ggtitle("Distribution of feeling temp")+theme(text = element_text(size = 10))
hist_plot3 = ggplot(bike_df, aes(actual_hum))+theme_bw()+geom_histogram(fill='blue',
bins = 20)+ggtitle("Distribution of Humidity")+theme(text = element_text(size = 10))
hist_plot4 = ggplot(bike_df,
aes(actual_windspeed))+theme_bw()+geom_histogram(fill='blue', bins =
20)+ggtitle("Distribution of windspeed")+theme(text = element_text(size = 10))

#Plot the Histogram graph together for continous variable
gridExtra::grid.arrange(hist_plot1, hist_plot2, hist_plot3, hist_plot4,ncol=2)
#Categorical variable
bar_plot1 = ggplot(bike_df,
aes(actual_season))+theme_bw()+geom_bar(fill='orange')+ggtitle("Counts of
Season")+theme(text = element_text(size = 10))
bar_plot2 = ggplot(bike_df,
aes(actual_yr))+theme_bw()+geom_bar(fill='blue')+ggtitle("Counts of Year")+theme(text =
element_text(size = 10))
bar_plot3 = ggplot(bike_df,
aes(actual_holiday))+theme_bw()+geom_bar(fill='lightblue')+ggtitle("Counts of
Holiday/Weekend/Weekday")+theme(text = element_text(size = 10))
```

```
bar_plot4 = ggplot(bike_df,
aes(actual_weathersit))+theme_bw()+geom_bar(fill='green')+ggtitle("Counts of
weather")+theme(text = element_text(size = 10))

#Plot the Histogram graph together for continous variable
gridExtra::grid.arrange(bar_plot1,bar_plot2, bar_plot3, bar_plot4,ncol=2)



#*** Bivariate analysis*****#
#Continous variable
bike_df$actual_temp <- as.factor(bike_df$actual_temp)
bike_df$actual_feel_temp <- as.factor(bike_df$actual_feel_temp)
scatter_plot = ggplot(bike_df, aes(x=actual_temp,
y=actual_feel_temp))+geom_point()+ggtitle("Distibution of Temp and Atemp")
plot(scatter_plot)

#### we can observer temp and atemp has a positve linear relation.
#### Correlation give us idea about Linear relpationship b/w 2 continous variables
bike_df$actual_temp = as.numeric(bike_df$actual_temp)
bike_df$actual_feel_temp = as.numeric(bike_df$actual_feel_temp)



### Finding the correlation between temp and atemp
cor(bike_df$actual_temp, bike_df$actual_feel_temp)
### correlation - 0.9917016

#Continous and Categorical
bike_df$actual_temp <- as.integer(bike_df$actual_temp)
bike_df$actual_feel_temp = as.integer(bike_df$actual_feel_temp)
box_plot = ggplot(bike_df, aes(x=actual_season, y=actual_temp))+geom_boxplot()
```

```
plot(box_plot)

#****** PRE-PROCESSION *************#
#Finding the missing values in dataset
missing_value<-data.frame(missing_value=apply(bike_df,2,function(x){sum(is.na(x))}))
missing_value

#### NO missing values found

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

#Detect multicollinearity###

####We have already verifed that a  relation between "temp" and "atemp" during bivariate
analysis.
####both are strongly correlated to each other
####Now, will see if collinearity existence bewtween Continuous variable over target  using
VIF method
vif_df <- bike_df[,c('temp','atemp','hum','windspeed')]
vif(vif_df)
### VIF values
#   Variables       VIF
# 1      temp 62.969819
# 2     atemp 63.632351
# 3       hum  1.079267
# 4 windspeed  1.126768
###
```

From the above we can understand that "temp" and "atemp" have a high Variance inflation factor(VIF),
they have almost same variance within the dataset. So, we might need to drop one of the feature before
moving to model buliding otherwise will end-up buliding a model with high multicolinearity

#******** Outlier detection and removal ********#

```
box_plot1 = ggplot(aes_string(y = bike_df$temp), data = bike_df)+stat_boxplot(geom =
"errorbar", width = 0.5) +
  geom_boxplot(outlier.colour="red", fill = "grey" ,outlier.shape=10,outlier.size=1,
notch=FALSE) +
  theme(legend.position="bottom")+labs(y=bike_df$temp)+ggtitle(paste("Box plot for
temp"))

box_plot2 = ggplot(aes_string(y = bike_df$atemp), data = bike_df)+stat_boxplot(geom =
"errorbar", width = 0.5) +
  geom_boxplot(outlier.colour="red", fill = "blue" ,outlier.shape=10,outlier.size=1,
notch=FALSE) +
  theme(legend.position="bottom")+labs(y=bike_df$atemp)+ggtitle(paste("Box plot for
atemp"))

box_plot3 = ggplot(aes_string(y = bike_df$hum), data = bike_df)+stat_boxplot(geom =
"errorbar", width = 0.5) +
  geom_boxplot(outlier.colour="red", fill = "green" ,outlier.shape=10,outlier.size=1,
notch=FALSE) +
  theme(legend.position="bottom")+labs(y=bike_df$hum)+ggtitle(paste("Box plot for hum"))
```

```
box_plot4 = ggplot(aes_string(y = bike_df$windspeed), data = bike_df)+stat_boxplot(geom
= "errorbar", width = 0.5) +
  geom_boxplot(outlier.colour="red", fill = "orange" ,outlier.shape=10,outlier.size=1,
notch=FALSE) +
  theme(legend.position="bottom")+labs(y=bike_df$windspeed)+ggtitle(paste("Box plot for
windspeed"))

gridExtra::grid.arrange(box_plot1,box_plot2,box_plot3,box_plot4, ncol=2, top='Outlier for
continous variable')
```

```
### as you refer from the BOX PLOT generated, we can observe OUTLIERS in features "hum"
and "windspeed"
 #Removing the outlier from feature "hum"
    #get the outlier values
    hum_outliers <- boxplot(bike_df$hum, plot=FALSE)$out
    hum_outliers

    #display the outliers
    bike_df[which(bike_df$hum %in% hum_outliers),]

    #drop those outliers
    bike_df <- bike_df[-which(bike_df$hum %in% hum_outliers),]
    #Removing the outlier from feature "windspeed"
    #get the outlier values
    win_outliers <- boxplot(bike_df$windspeed, plot=FALSE)$out
    #display the outliers
    bike_df[which(bike_df$windspeed %in% win_outliers),]
   #drop those outliers
```

```
bike_df <- bike_df[-which(bike_df$windspeed %in% win_outliers),]
dim(bike_df)



box_plot3 = ggplot(aes_string(y = bike_df$hum), data = bike_df)+stat_boxplot(geom =
"errorbar", width = 0.5) +geom_boxplot(outlier.colour="red", fill = "green"
,outlier.shape=10,outlier.size=1,
notch=FALSE) +  theme(legend.position="bottom")+labs(y=bike_df$hum)+ggtitle(paste("Box
plot for hum"))

box_plot4 = ggplot(aes_string(y = bike_df$windspeed), data = bike_df)+stat_boxplot(geom
= "errorbar", width = 0.5) +
geom_boxplot(outlier.colour="red", fill = "orange" ,outlier.shape=10,outlier.size=1,
notch=FALSE) +
theme(legend.position="bottom")+labs(y=bike_df$windspeed)+ggtitle(paste("Box plot for
windspeed"))

gridExtra::grid.arrange(box_plot3,box_plot4, ncol=2, top='Box plot after Outlier removal')
#********* MODEL BULIDING *********#
colnames(bike_df)
#Drop the columns/features which are not needed

bike_df <- subset(bike_df,select = -c(instant, dteday, atemp, casual, registered,
actual_temp, actual_feel_temp, actual_hum, actual_windspeed, actual_season, actual_yr,
actual_holiday, actual_weathersit))
colnames(bike_df)

#**** Liner regression model *****#
# divide data into train and test
train_index = sample(1:nrow(bike_df), 0.8 * nrow(bike_df))
```

```
        train <- bike_df[train_index,]
        test <- bike_df[-train_index,]
#Inovke linear regression model
lr_model = lm(cnt ~., data = train)
#Summary of the model
summary(lr_model)
#prediction of test data
pred_lr = predict(lr_model, test[,-11])
#display actual vs predicted values
temp_df = data.frame("actual"=test[,11], "pred"=pred_lr)
head(temp_df)
#Calculate MAPE
MAPE = function(actual, pred){
 return(mean(abs((actual - pred)/actual)) * 100)
        }
Mape <- MAPE(test[,11], pred_lr)
print(Mape)
```