Chapter 1: Introduction

1.1 Problem Statement

The objective of this Case is to Predication of bike rental count on daily based on the environmental and seasonal settings.

1.2 Data

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	at
1	1/1/2011	1	0	1	0	6	0	2	0.344167	0
2	1/2/2011	1	0	1	0	0	0	2	0.363478	0
3	1/3/2011	1	0	1	0	1	1	1	0.196364	0
4	1/4/2011	1	0	1	0	2	1	1	0.2	0
5	1/5/2011	1	0	1	0	3	1	1	0.226957	0
6	1/6/2011	1	0	1	0	4	1	1	0.204348	0
7	1/7/2011	1	0	1	0	5	1	2	0.196522	0
8	1/8/2011	1	0	1	0	6	0	2	0.165	0
9	1/9/2011	1	0	1	0	0	0	1	0.138333	0

Table 1.1: Bike Count Sample Data

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.

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.

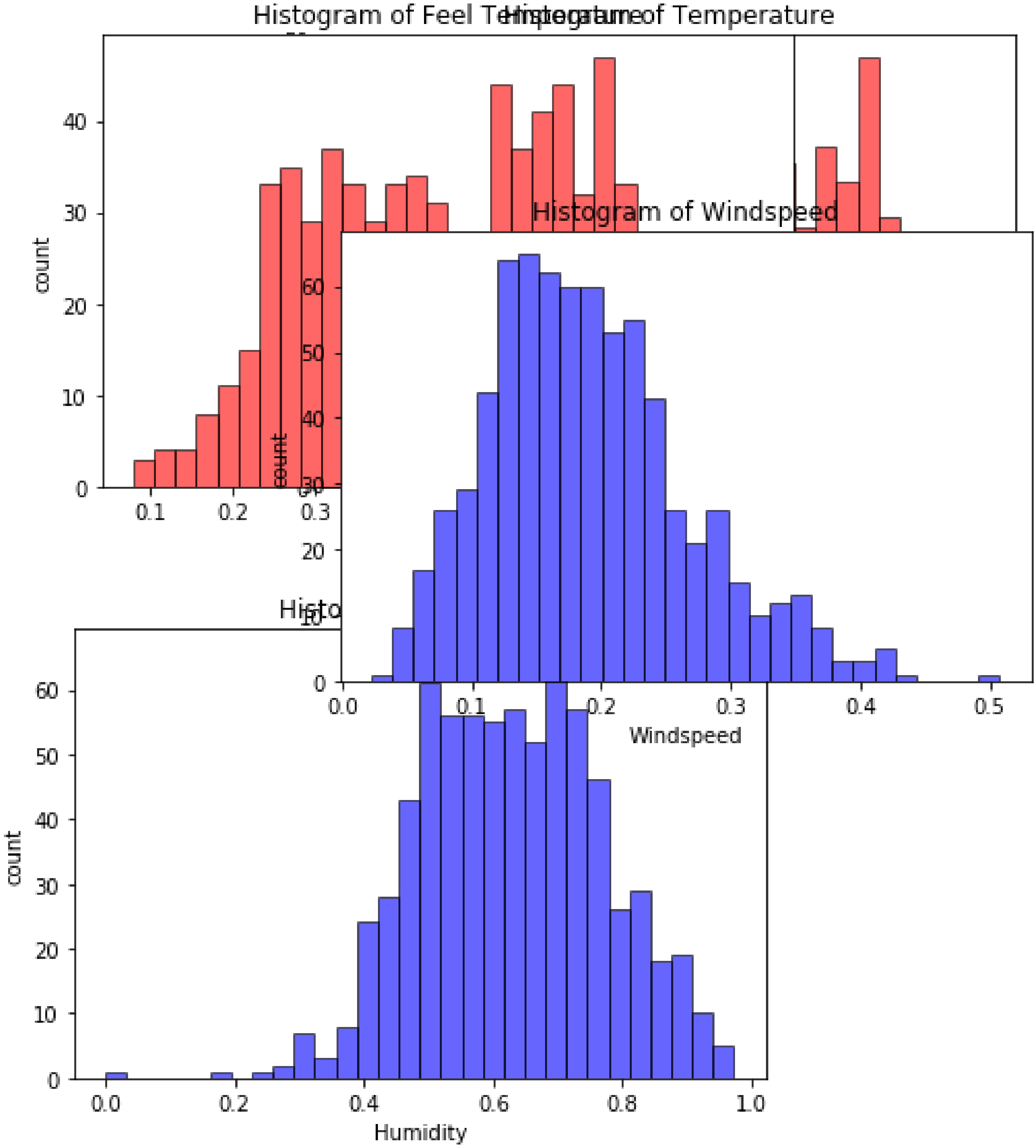
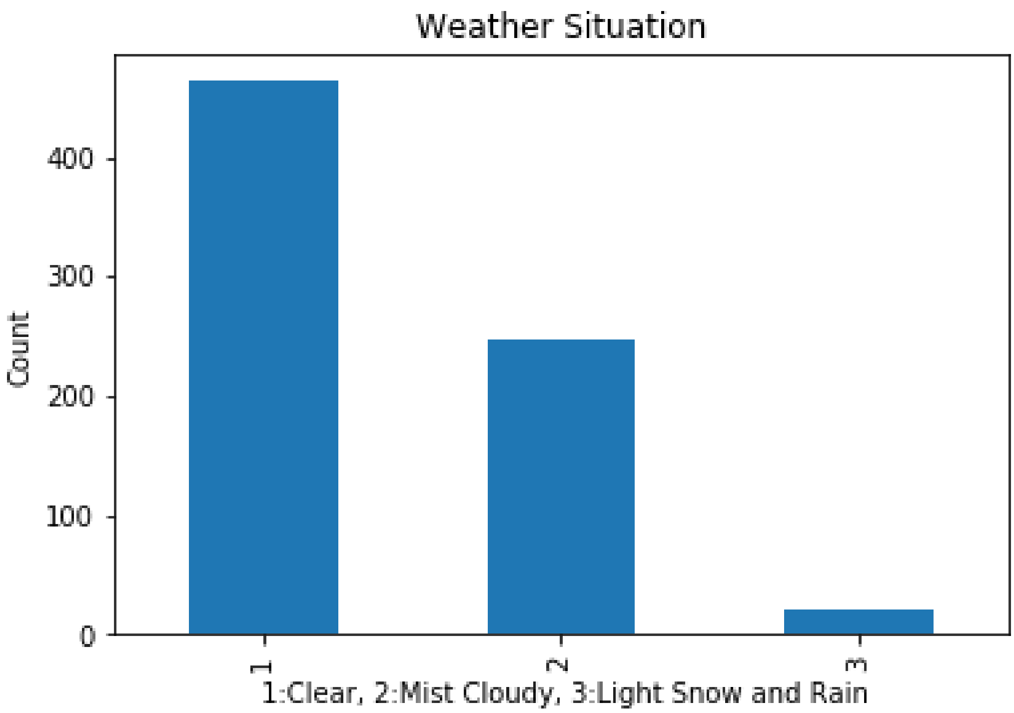
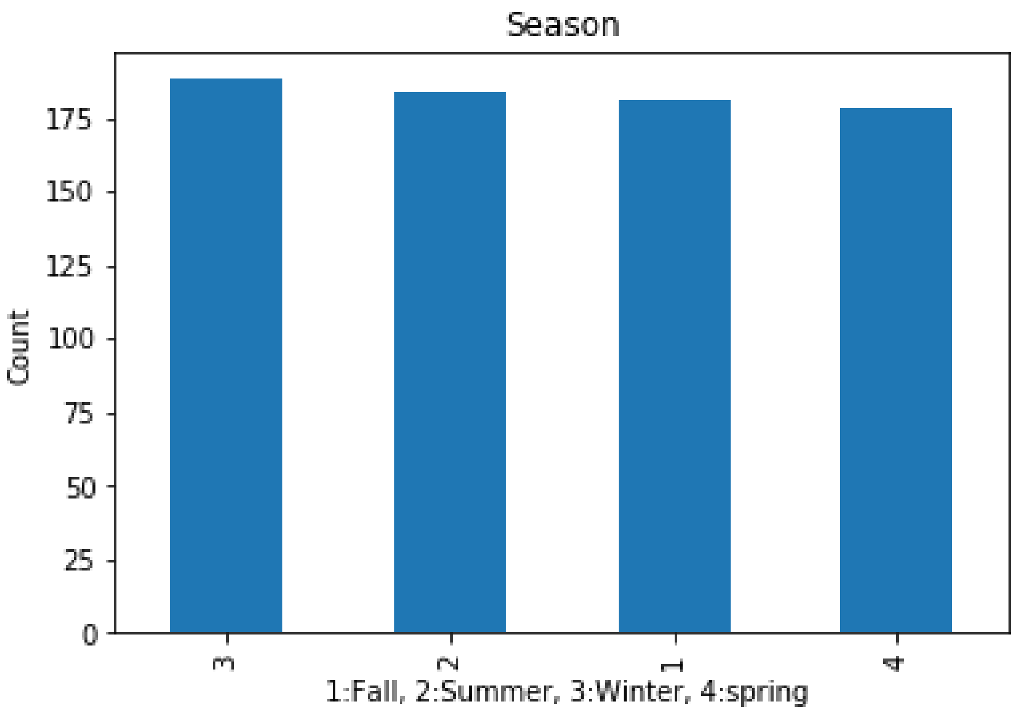
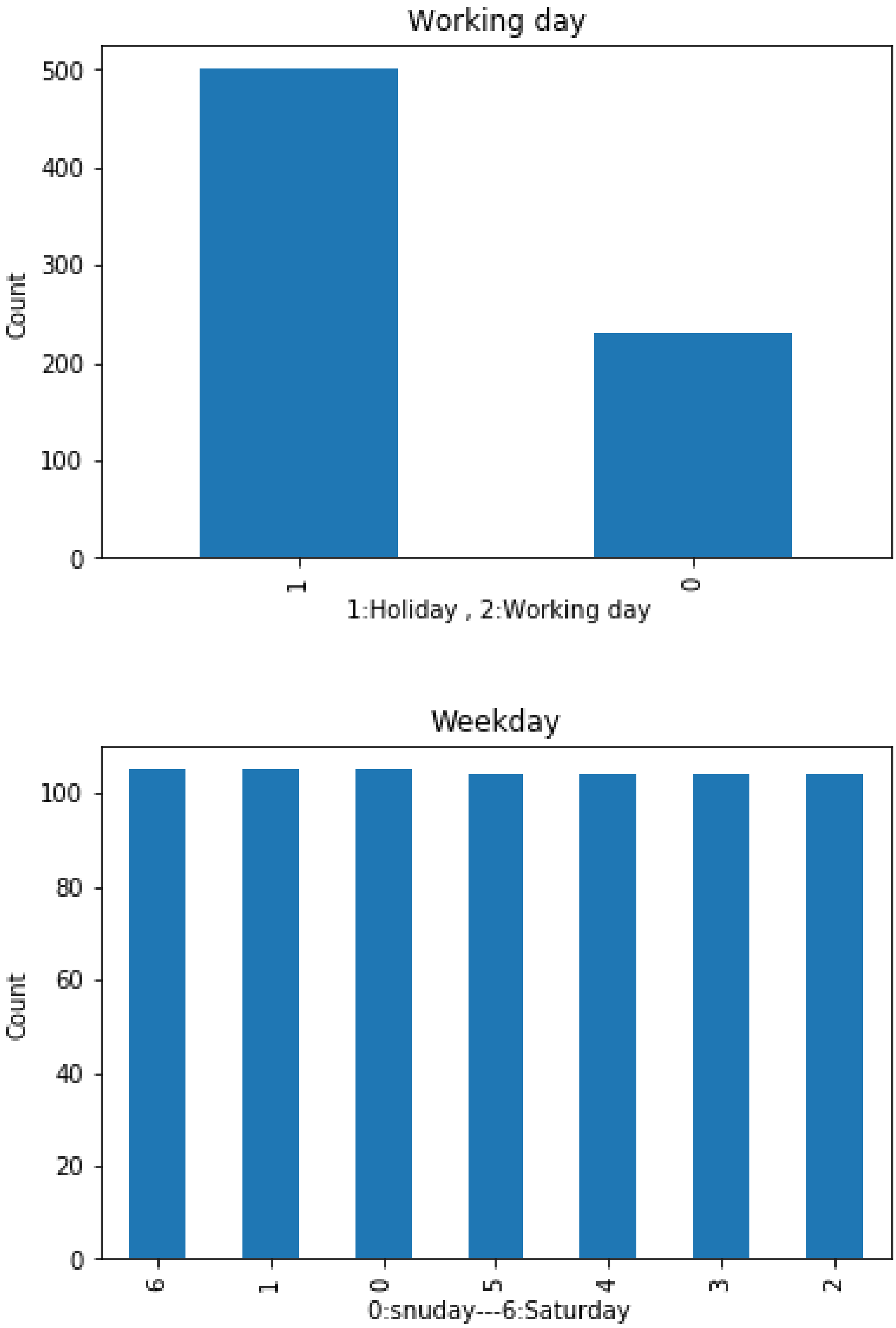


Fig 2.1: Distribution of continuous variables using Histograms

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



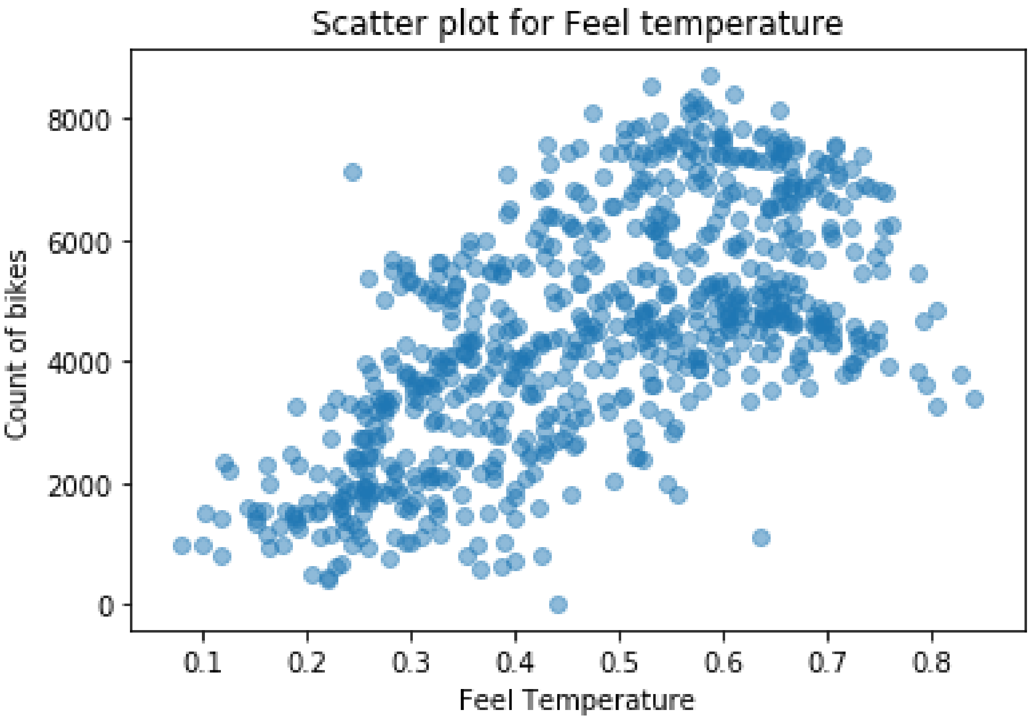


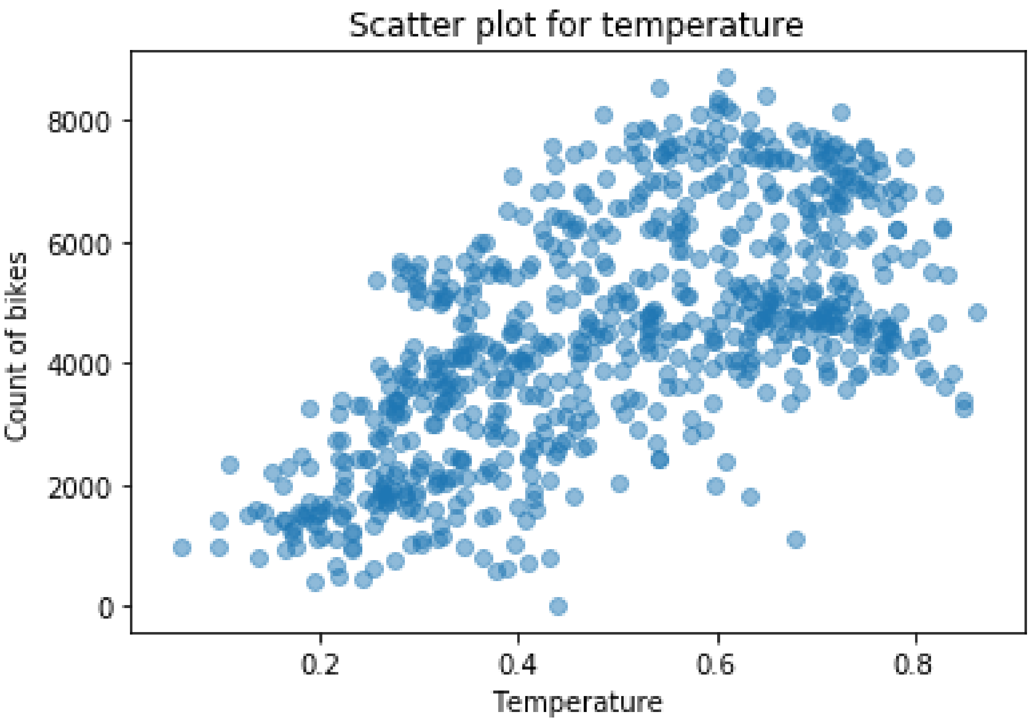


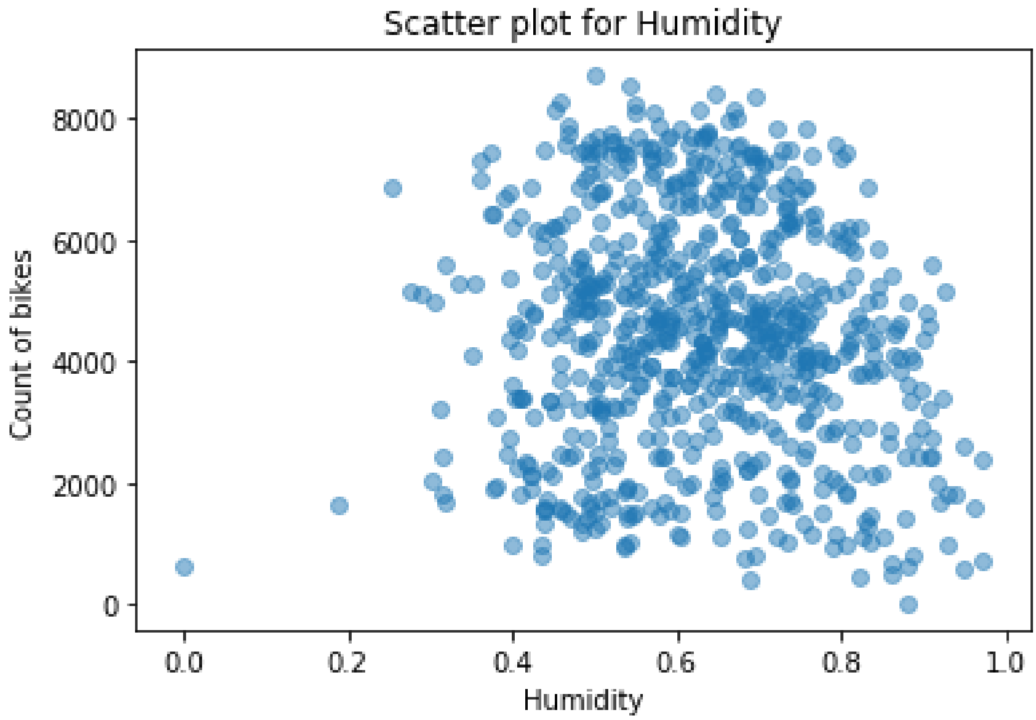
2.2: Fig Distribution of categorical variables using bar plots

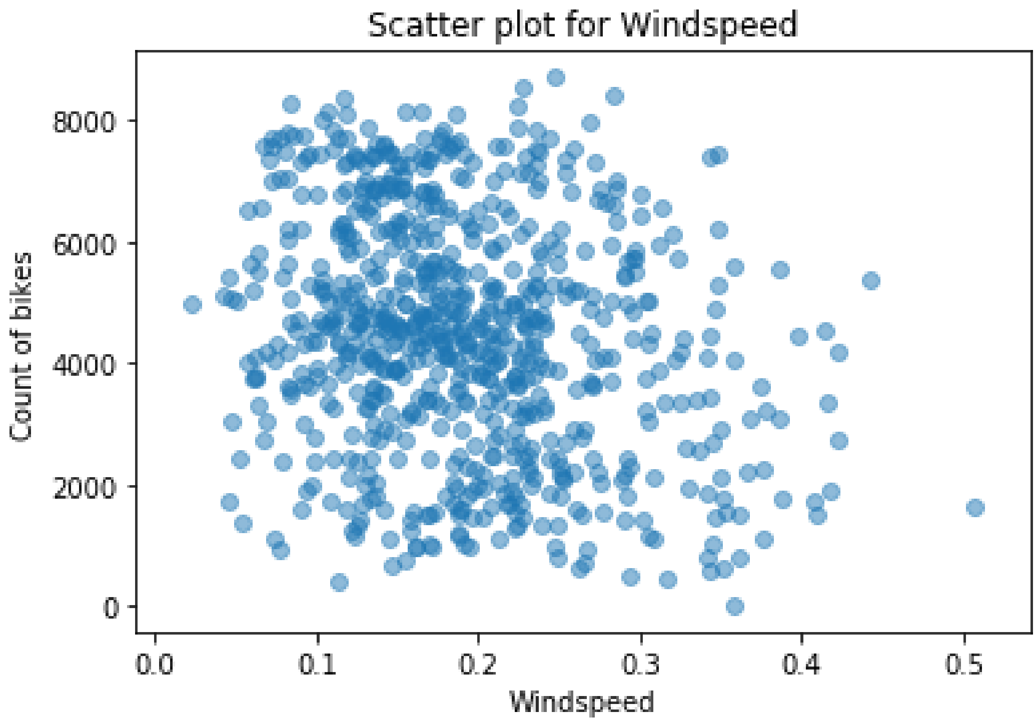
2.4 Relationship of Continuous variables against bike count

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.


Scatter plot for Feel temperature
Count of bikes
Feel Temperature
0
2000
4000
6000
8000
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


Scatter plot for temperature
Count of bikes
Temperature
0
2000
4000
6000
8000
0.2
0.4
0.6
0.8





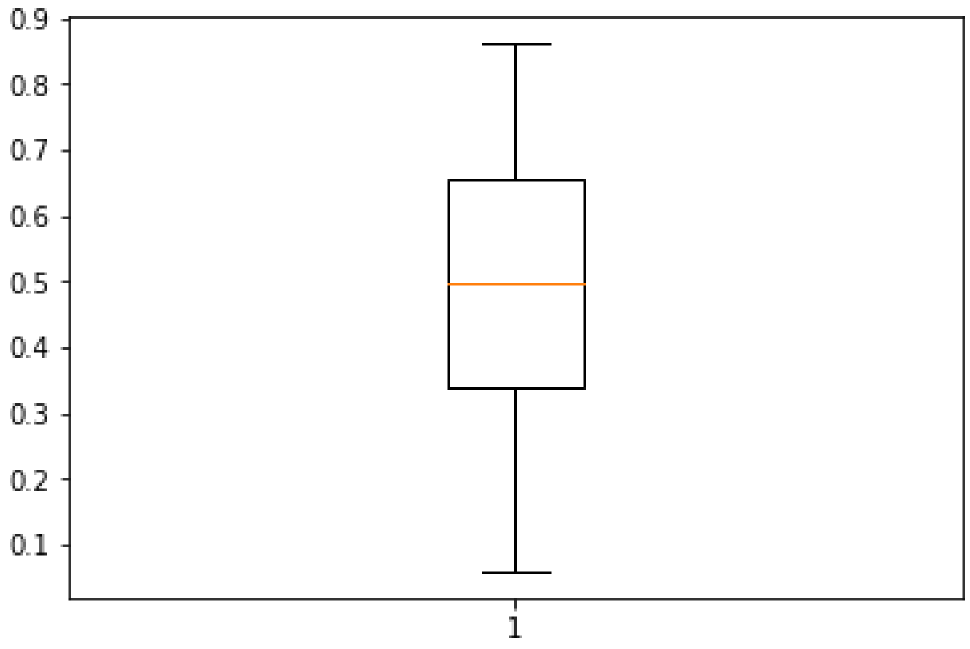
When plotting the count by weather related variables, it is found that temperature and wind speed affect the behaviour of renting a bike more dramatically. With the temperature grows the number of rental bikes grows. It seems that about 17 to 26 degree Celsius is the comfortable temperature for people to ride bikes, and the number goes down when temperature is more than 28. People feel uncomfortable to ride a bike in big wind, it seems that many people are not prefer to ride bikes when wind speed is more than 20 kilometres per hour, and only a few people rent bikes when wind speed is more than 30 kilometres.

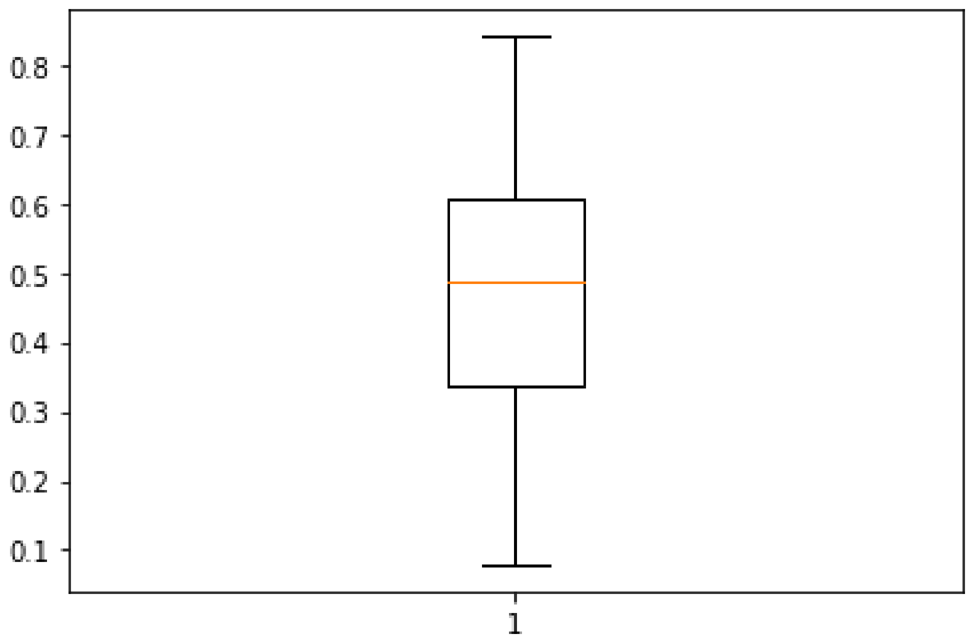
When the weather is very dry, which the relative humidity is less than 20%, people feel uncomfortable and not prefer to rent bikes. Weather seems an important factor, if there is fog, rain or snow, the number of rental bikes decrease a lot, especially snow. It is interested to find out that short term users are more affected by temperature. It seems that no matter cold or hot, working people insiste on renting bikes, while casual users prefer rent bikes in confortalbe weather.

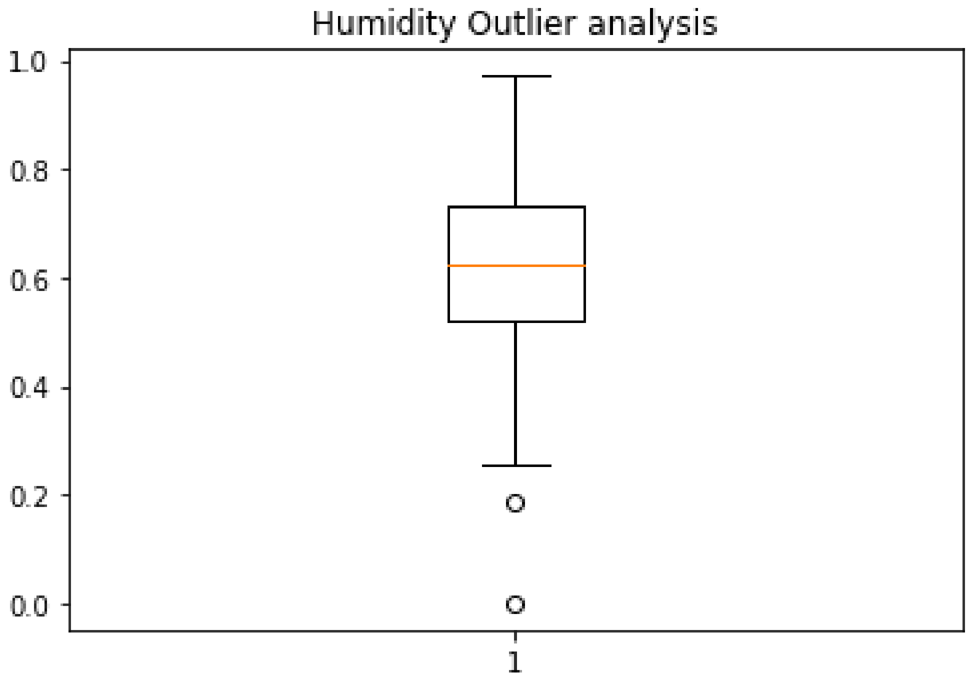
2.5: Detection of outliers:
Outliers are detected using boxplots. Below figure illustrates the boxplots for all the continuous variables.

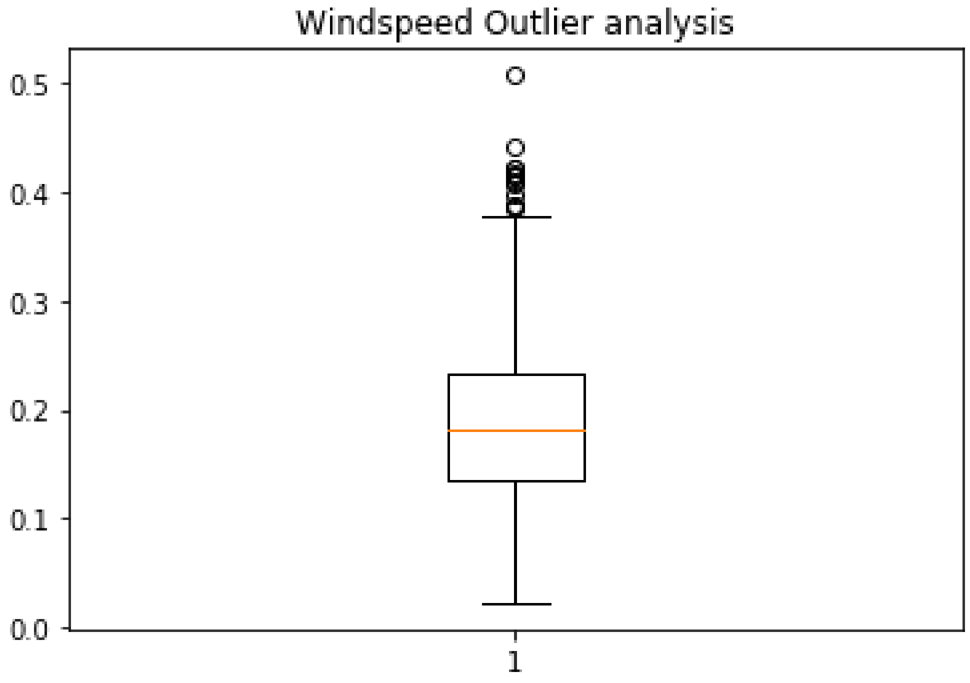
Temp Outlier analysis

atemp Outlier analysis


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1


0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1

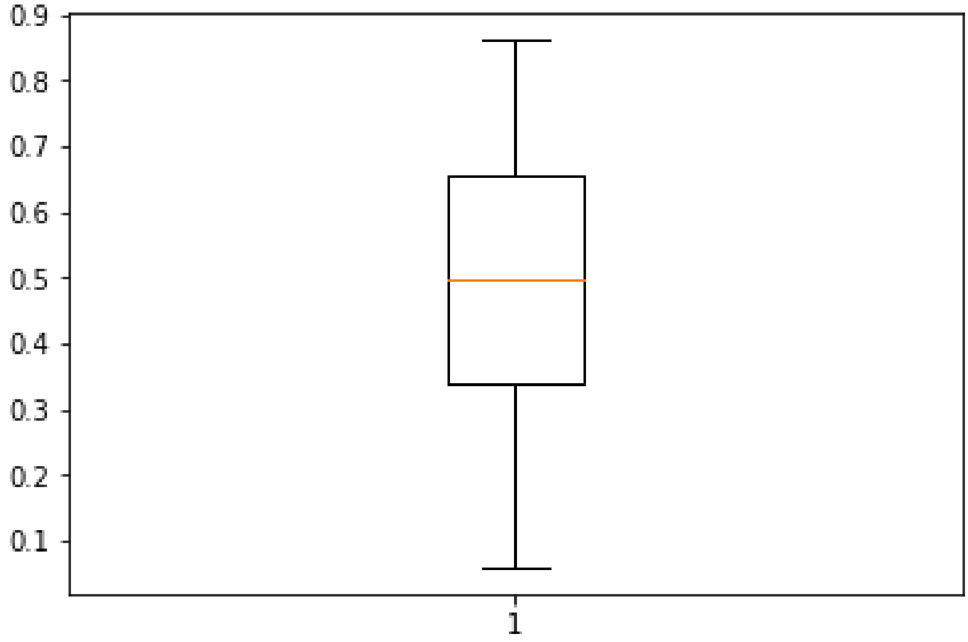


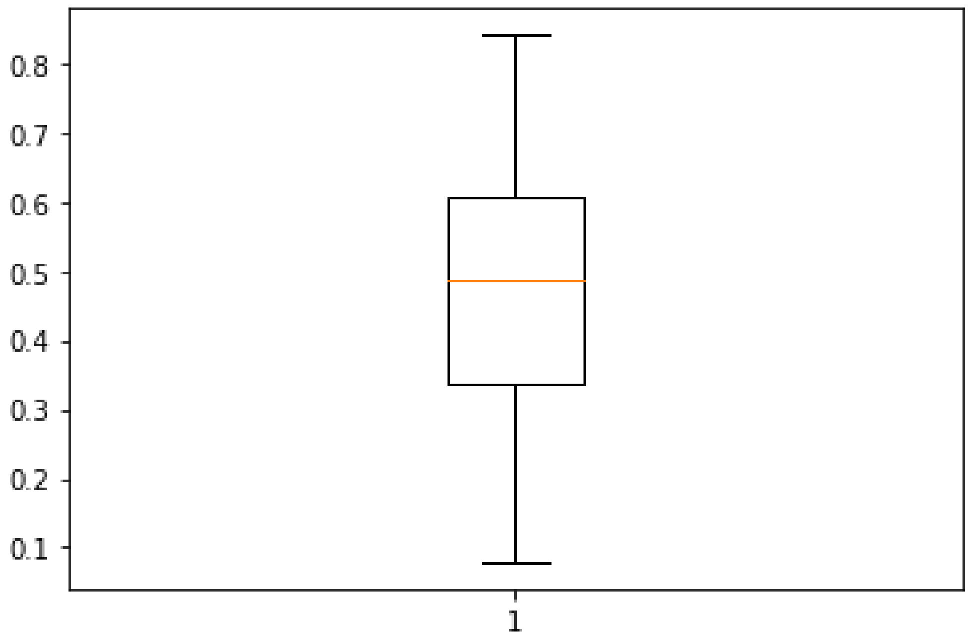


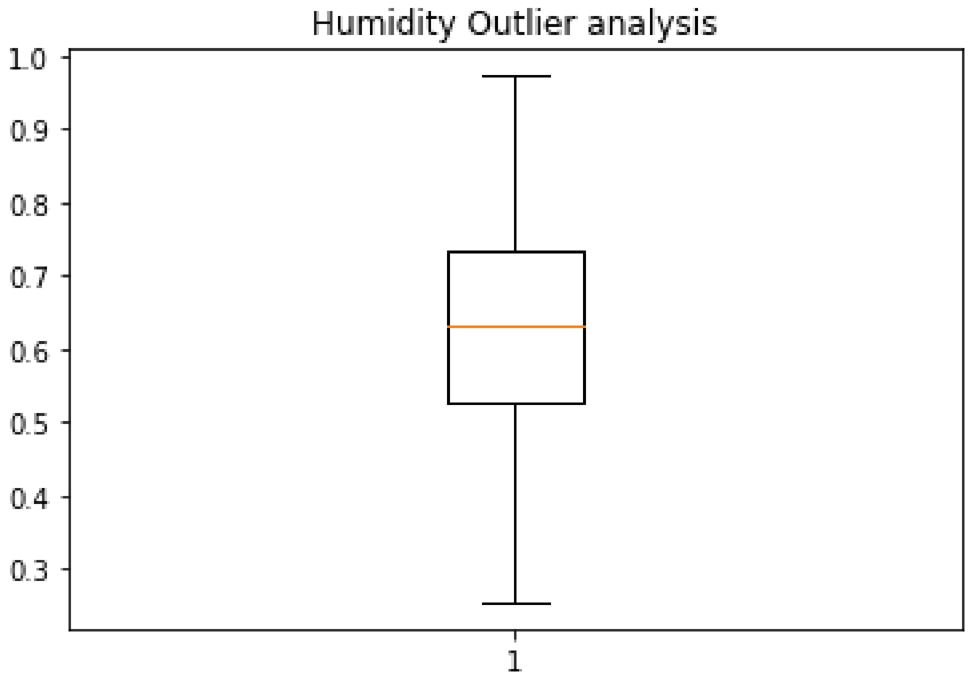
Outliers can be removed using the Boxplot stats method, wherein the Inter Quartile Range (IQR) is calculated and the minimum and maximum value are calculated for the variables. Any value ranging outside the minimum and maximum value are discarded. The boxplot of the continuous variables after removing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

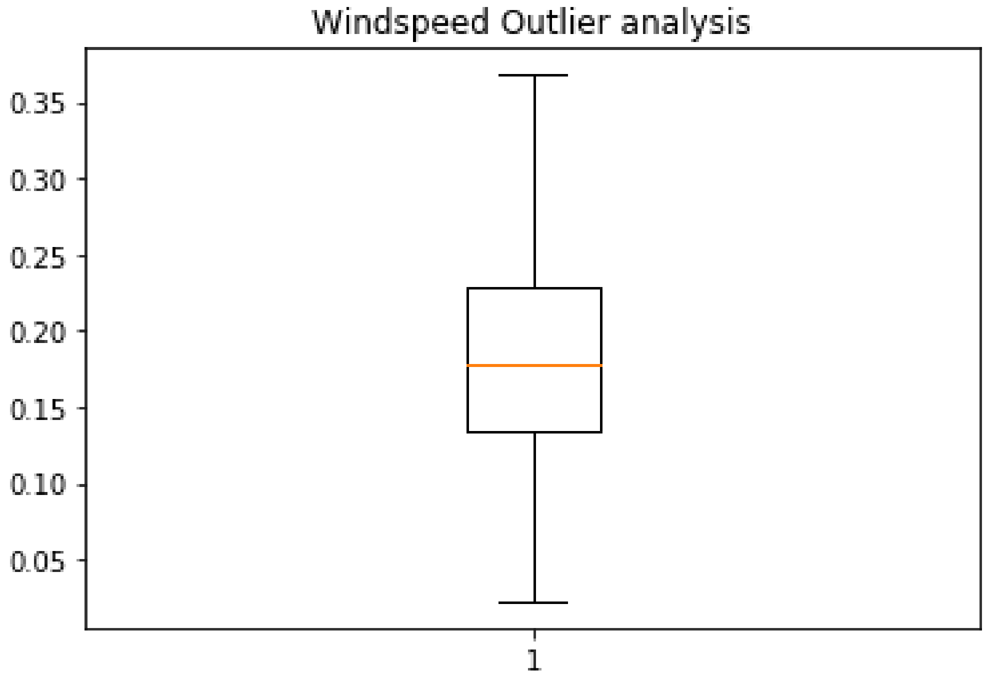
Temp Outlier analysis

atemp Outlier analysis

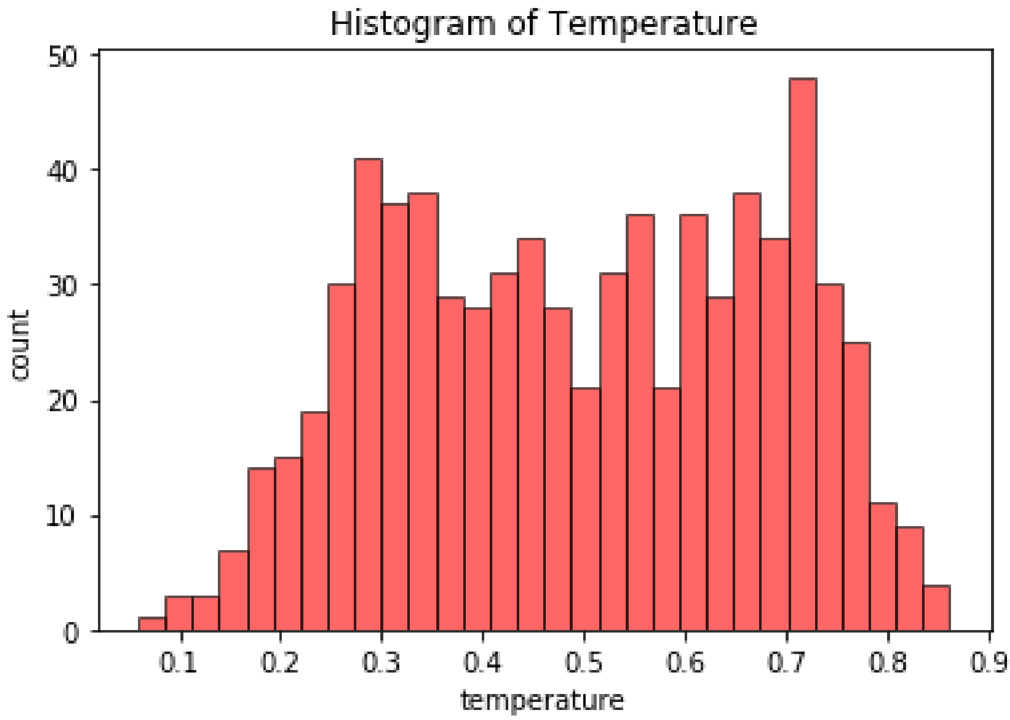


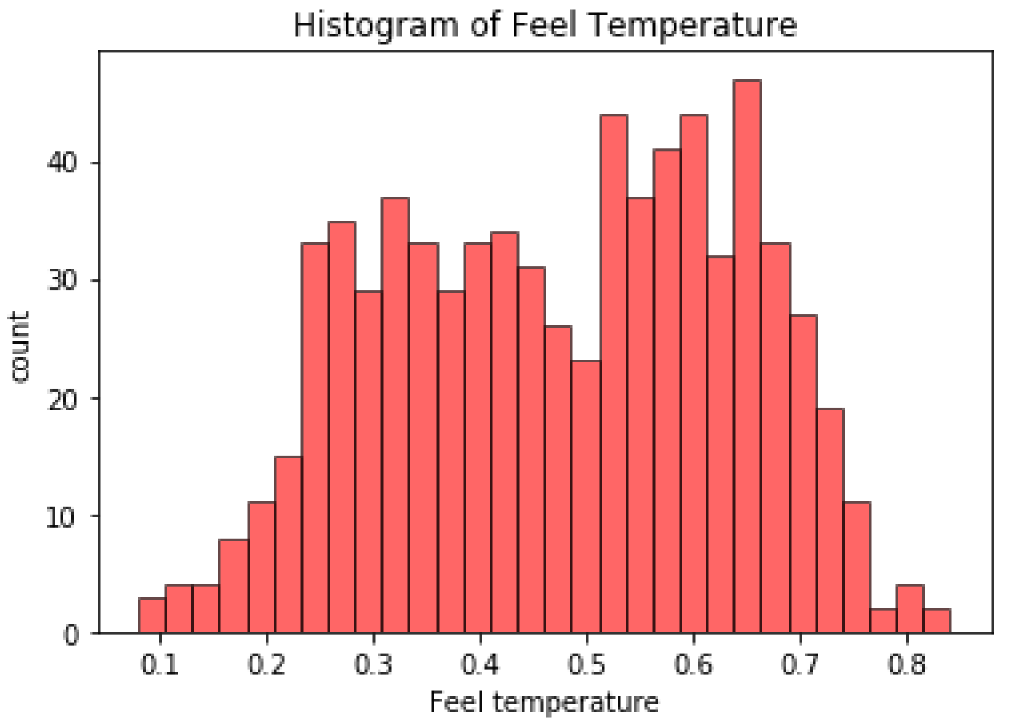


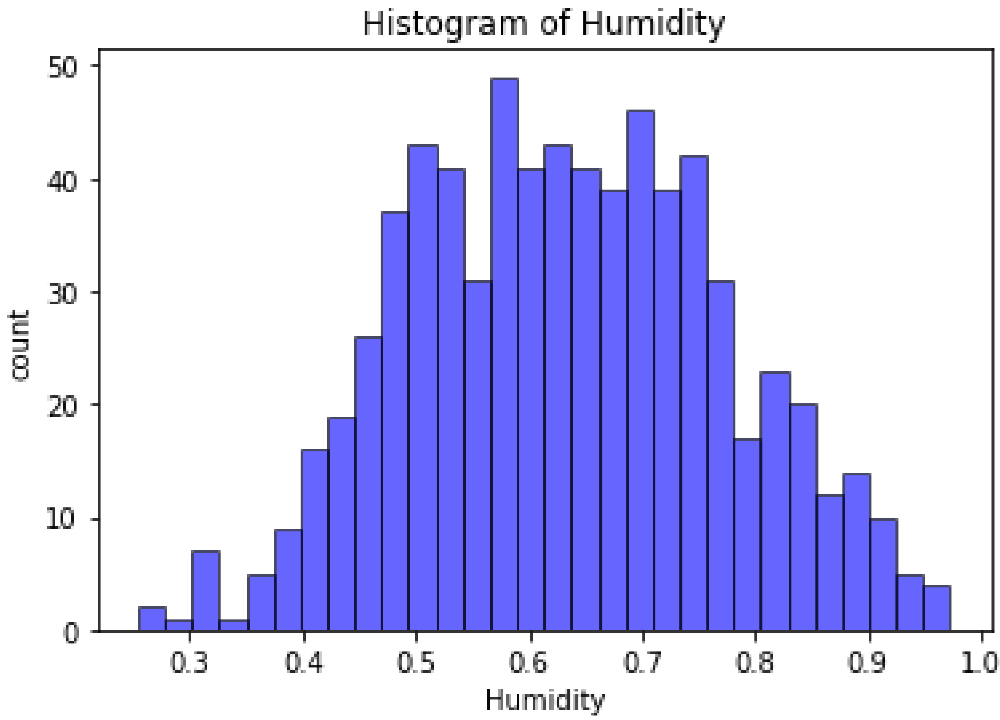


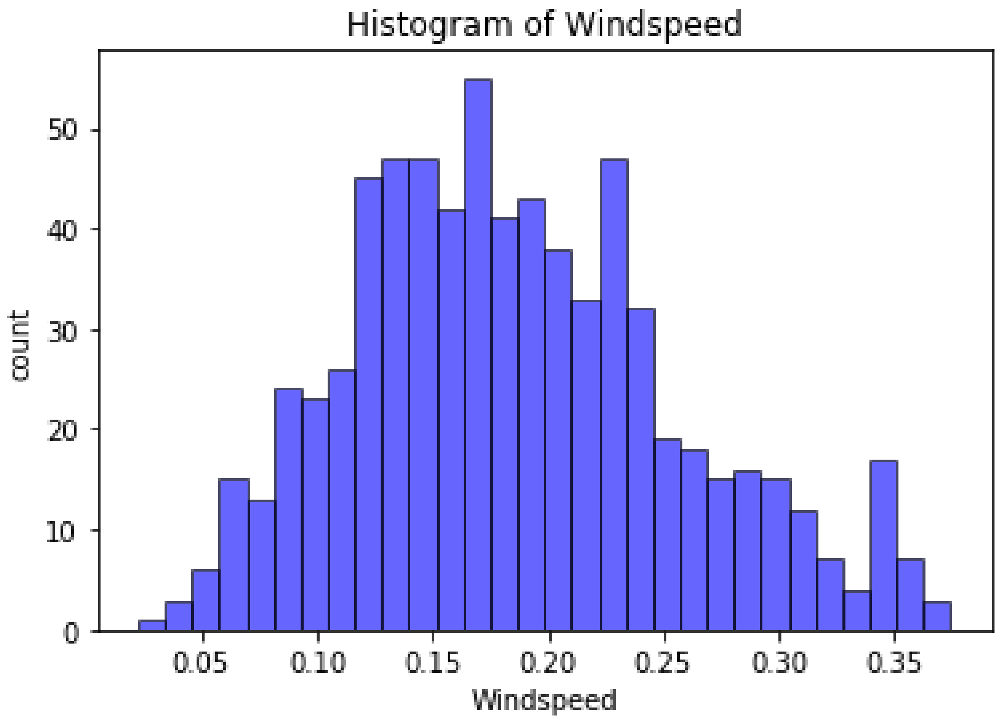


It can be observed from the distribution of Windspeed and humidity after removal of outliers, is that data is not skewed as much as before the removal of outliers. The figure shown below illustrates the distribution of continuous variables using histograms.


Histogram of Temperature
count
0
10
20
30
40
50
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
temperature

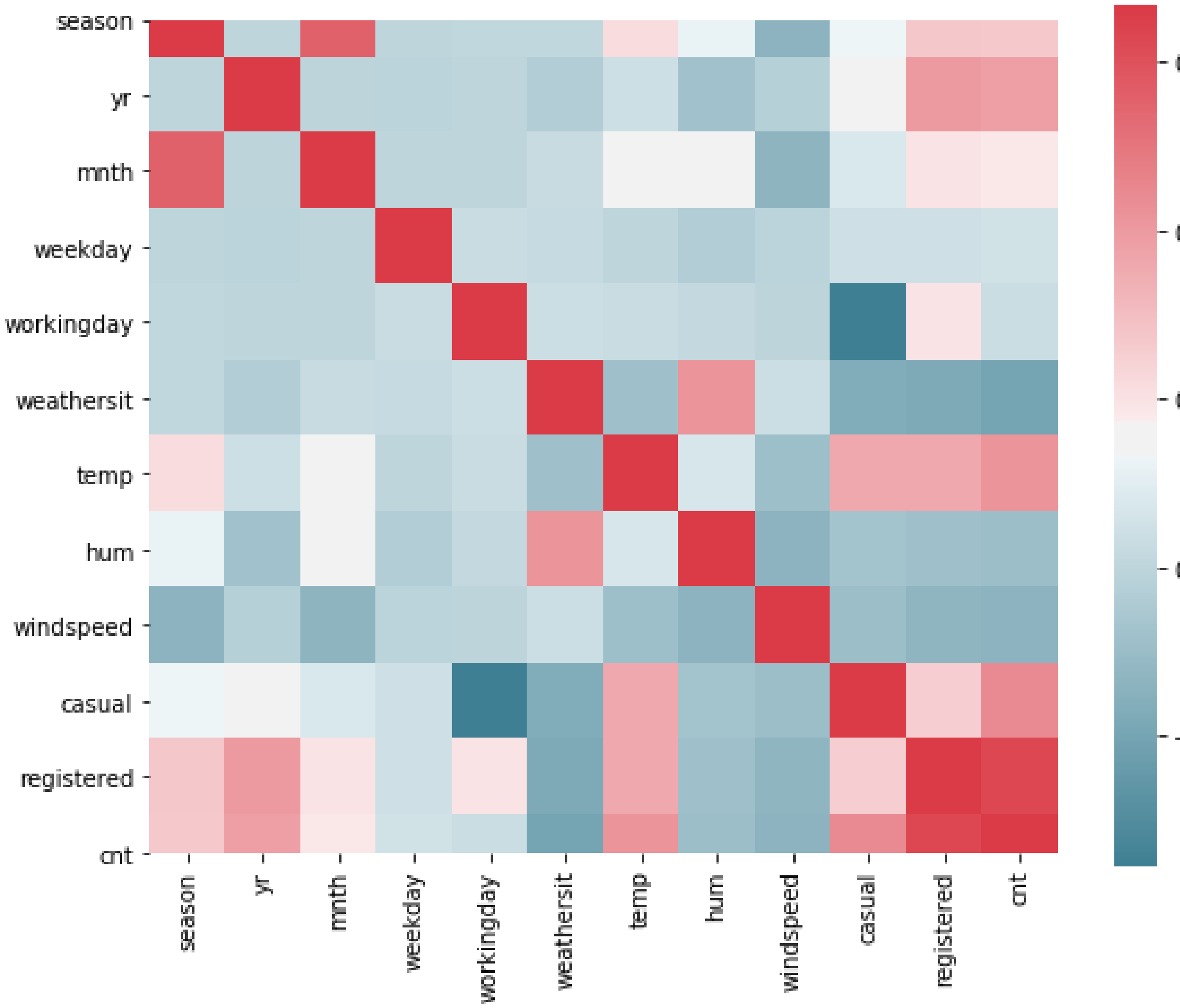

Histogram of Feel Temperature
count
0
10
20
30
40
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Feel temperature





3.3.6 Correlation

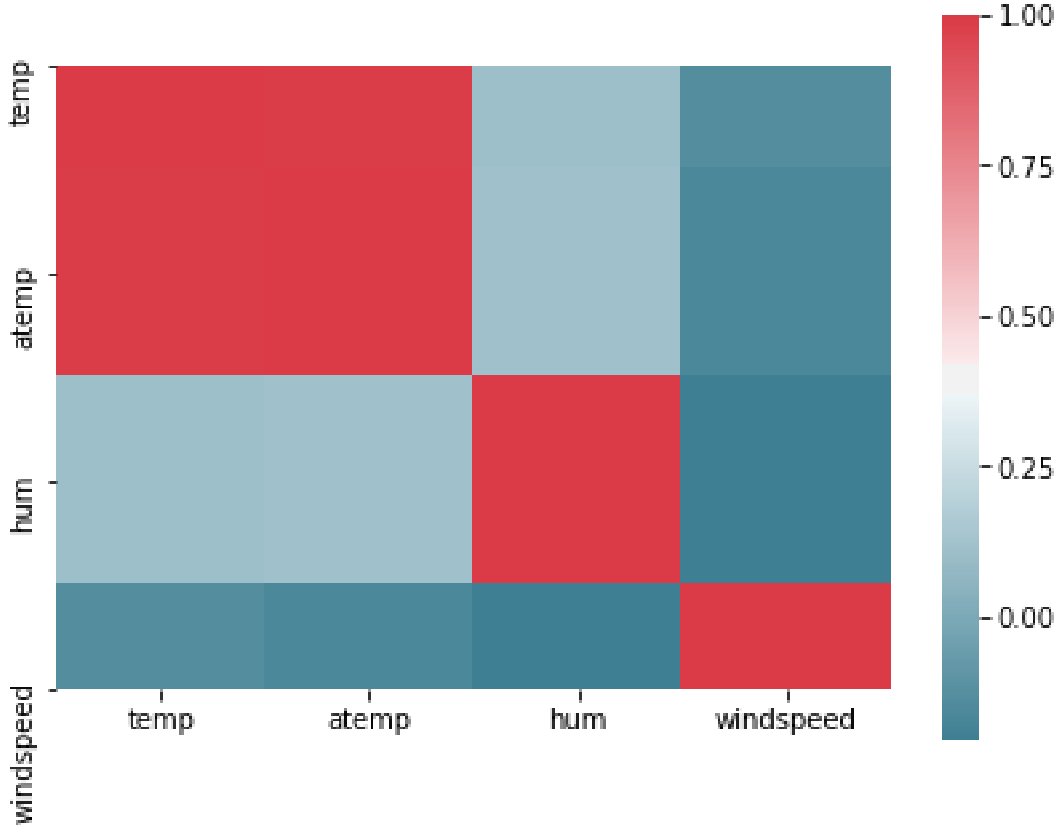
Before building predictive models, it is important to find out the relationship of all features with target variable (cnt). The correlation plot (more details in appendix A) below shows that cnt has a strong relationship with temperature, and relative strong relationship with hour, month, humidity, weather type. When building models, these features will get more attention.



Conclusion - About feature importance, time related variables (such as hour, weekday, month, and year), weather related variables (such as humidity, temperature, wind speed, rain, fog and snow) are all significant factors to affect people's behavior to rent a bike. All the variables should be included when building models. Specifically, variable temperature, hour, month, humidity, and weather type have relatively strong relationship with target variable cnt, so they should get more attention.

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.



As we can in above heatmap for continous variable , temp and atemp is highly correlated so we have removed to avoid multicollinearity in dataset.

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). And Mean absolute percentage error.

3.1 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

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

Residuals:
    Min      1Q  Median      3Q     Max 
-4014.3  -341.8    77.7   467.5  2900.0 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1521.86     271.45   5.606 3.28e-08 ***
season2       795.42     209.72   3.793 0.000166 ***
season3       960.31     252.49   3.803 0.000159 ***
season4      1639.81     207.96   7.885 1.72e-14 ***
yr1          2051.30      68.44  29.974  < 2e-16 ***
mnth2         195.05     171.97   1.134 0.257211    
mnth3         554.12     195.04   2.841 0.004664 ** 
mnth4         533.72     286.19   1.865 0.062728 .  
mnth5         885.32     309.63   2.859 0.004409 ** 
mnth6         636.14     325.81   1.953 0.051389 .  
mnth7         -24.72     363.78  -0.068 0.945838    
mnth8         246.58     357.38   0.690 0.490514    
mnth9         920.80     309.95   2.971 0.003101 ** 
mnth10        495.87     279.68   1.773 0.076789 .  
mnth11       -160.50     265.88  -0.604 0.546323    
mnth12       -162.47     210.49  -0.772 0.440512    
weekday1     -536.15     212.60  -2.522 0.011957 *  
weekday2     -467.51     234.45  -1.994 0.046642 *  
weekday3     -363.01     234.88  -1.546 0.122799    
weekday4     -357.59     234.41  -1.526 0.127708    
weekday5     -338.41     233.02  -1.452 0.146996    
weekday6      427.46     126.34   3.383 0.000768 ***
workingday1   738.50     200.38   3.686 0.000251 ***
weathersit2  -450.08      88.45  -5.088 4.98e-07 ***
weathersit3 -1960.75     215.77  -9.087  < 2e-16 ***
temp         4413.93     493.01   8.953  < 2e-16 ***
hum         -1500.11     333.95  -4.492 8.62e-06 ***
windspeed   -2748.98     504.16  -5.453 7.53e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 797.8 on 546 degrees of freedom
Multiple R-squared:  0.845,	Adjusted R-squared:  0.8373 
F-statistic: 110.2 on 27 and 546 DF,  p-value: < 2.2e-16
```

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.

3.2 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 574.218. The MAPE for this decision tree is 18.12%. Hence the accuracy for this model is 82.88%.

3.3 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 700. MAE for this model is

68. Using random forest, the MAPE was found to be 18.84%. Hence the accuracy is 82.16%.

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)

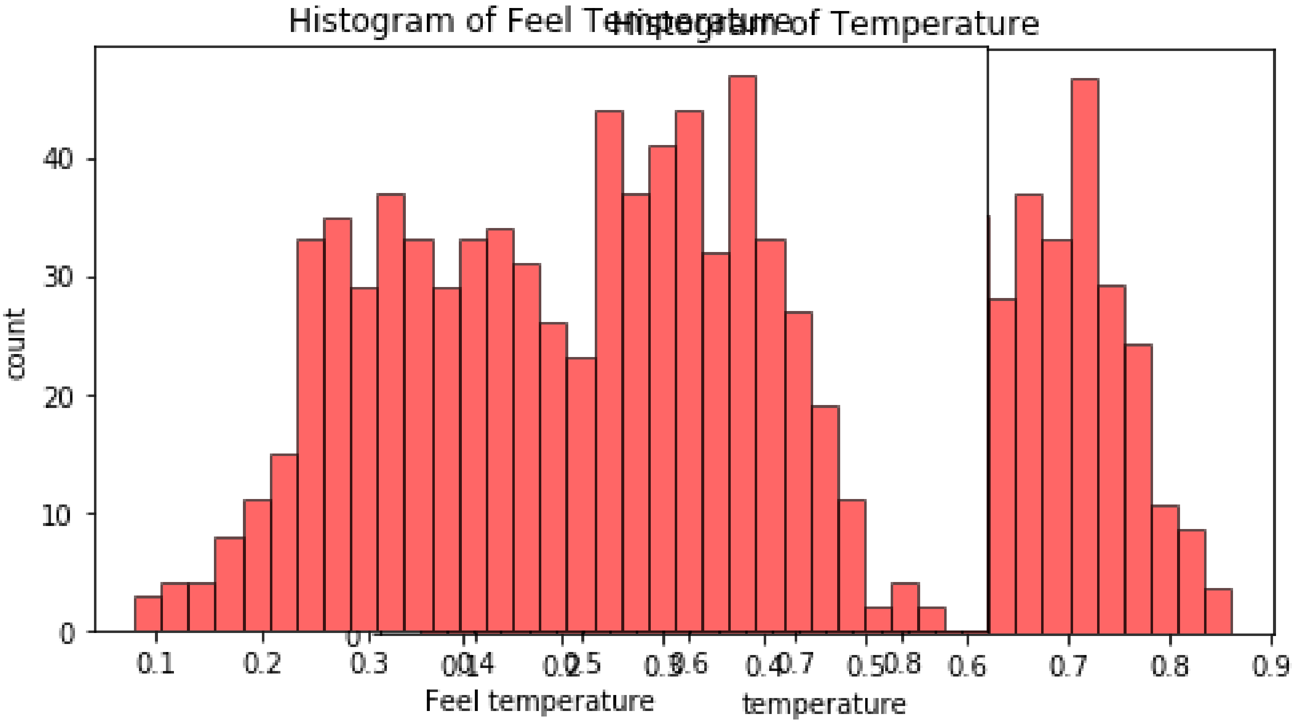
MAE is one of the error measures used to calculate the predictive performance of the model. We will apply this measure to our models that we have generated in the previous section.

```
MAE <- function (actual, pred)
{
  print(mean (abs (actual - pred)))
}
```

Linear Regression Model: MAE = 494 Decision Tree: MAE = 574.

Random Forest: MAE = 68

Based on the above error metrics, Random Forest is the better model for our

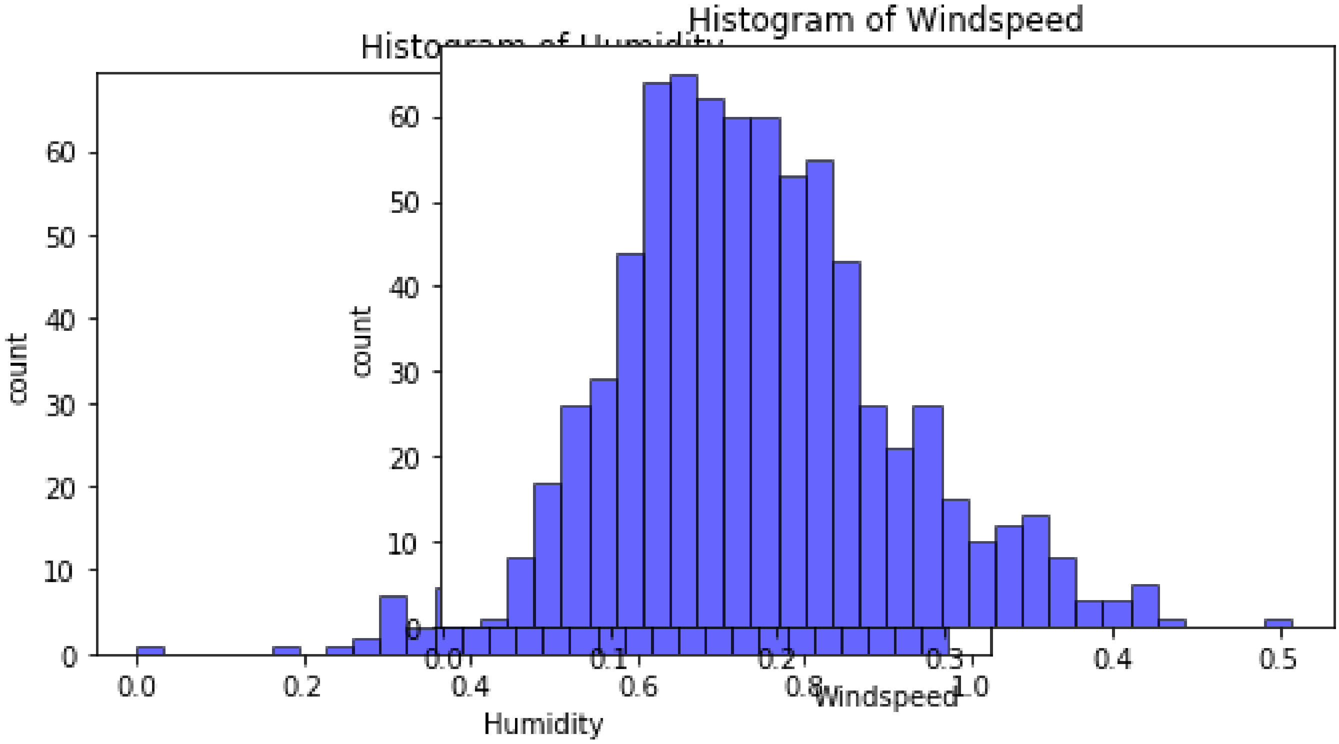


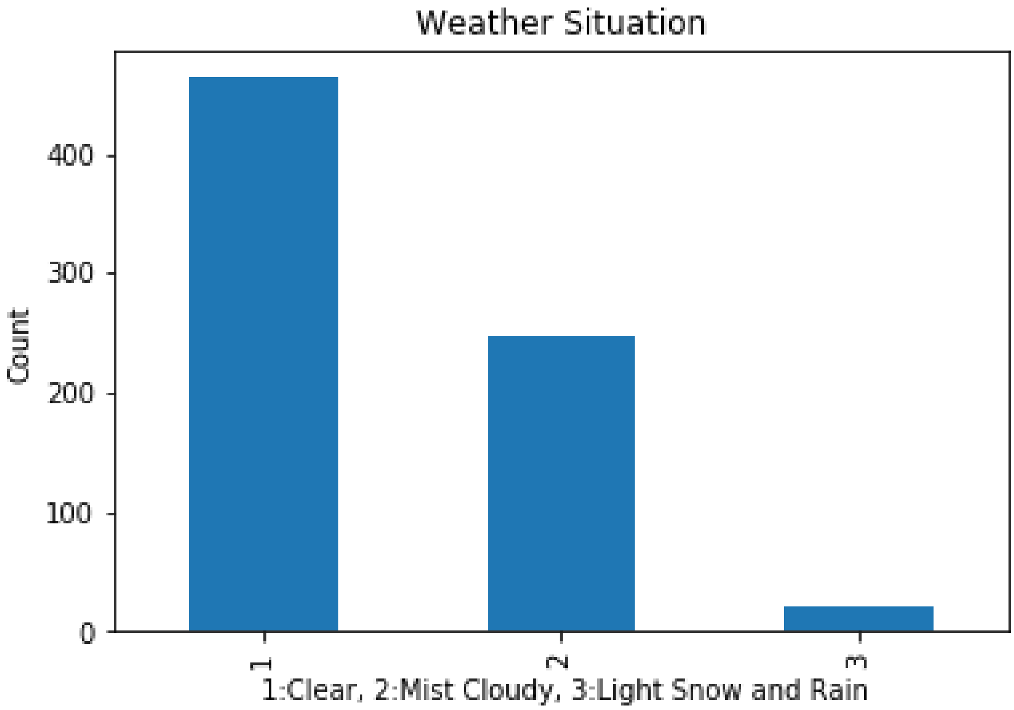
analysis. Hence Random Forest is chosen as the model for prediction of bike rental count.

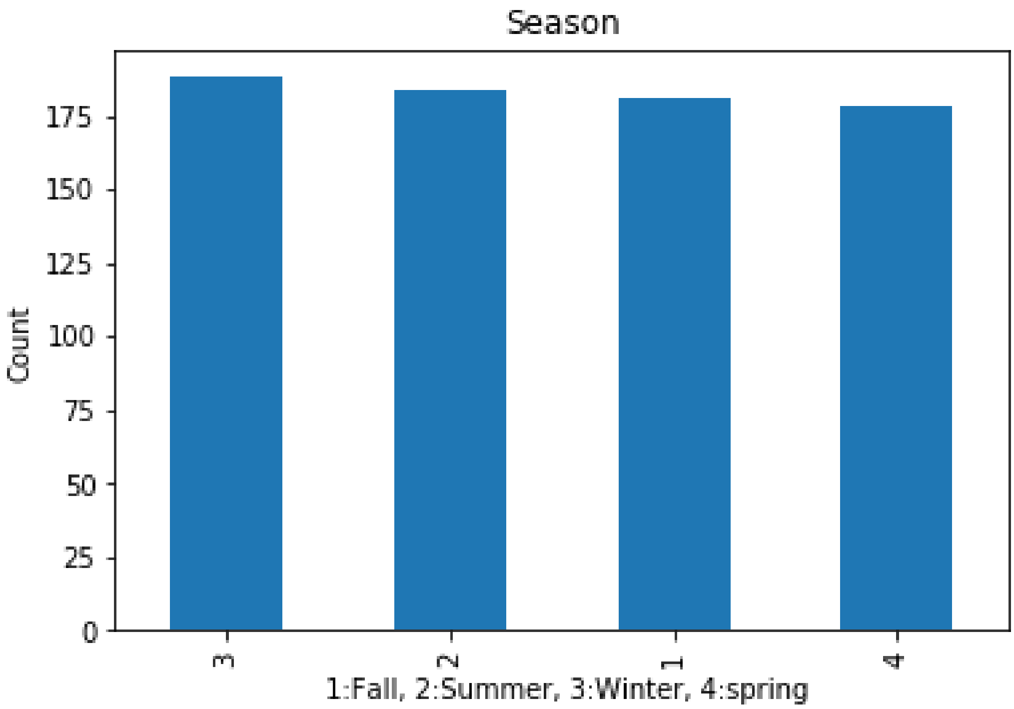
Chapter 5: Appendix

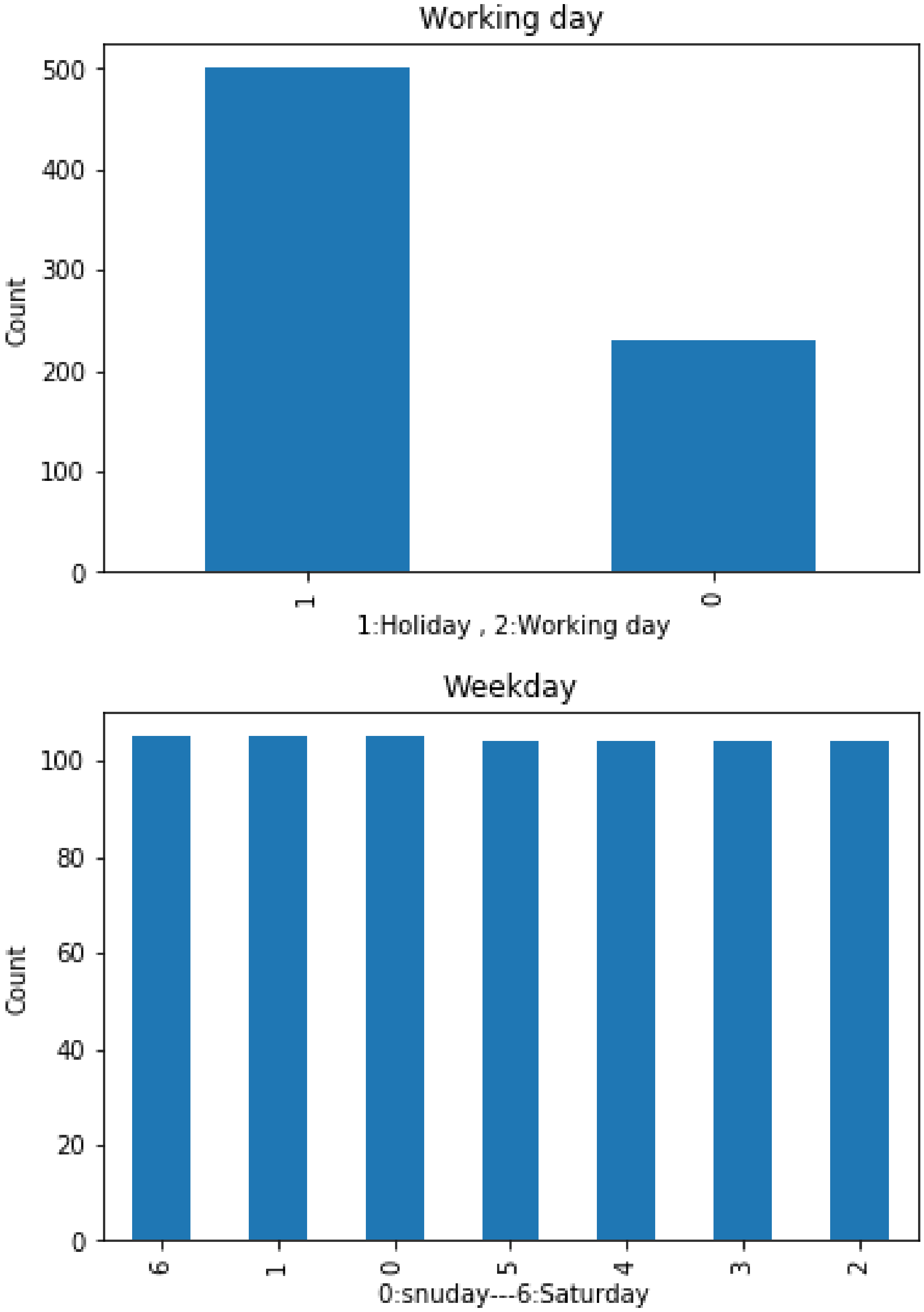
5.1 Figures

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

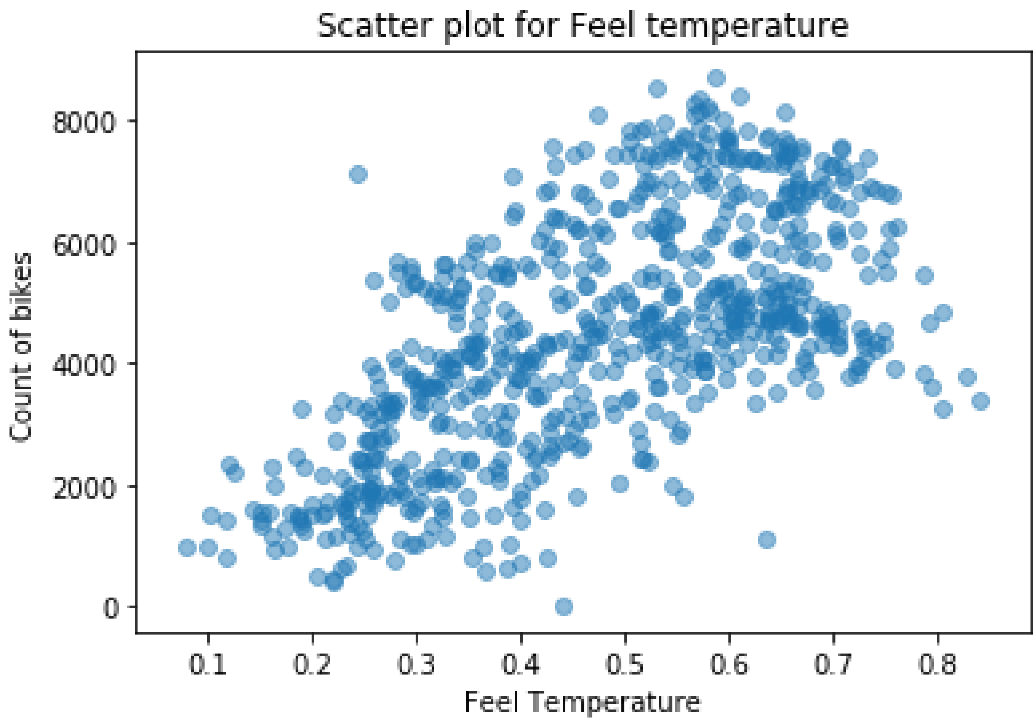

Histogram of Humidity
Histogram of Windspeed
count
Humidity
Windspeed

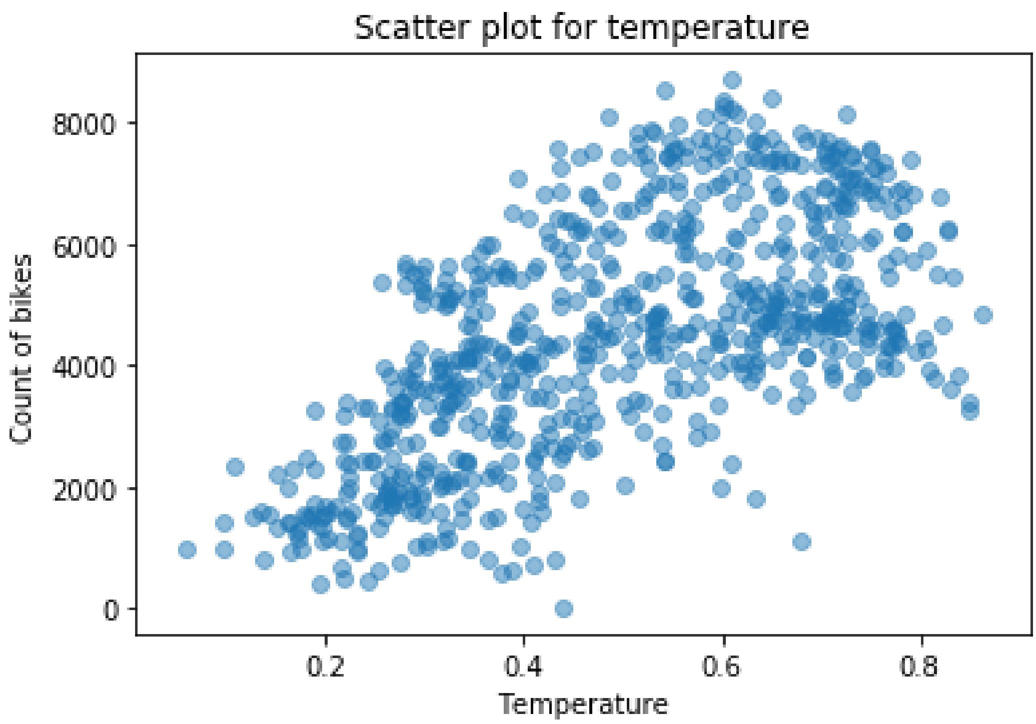

Weather Situation
400
300
200
100
0
Count
1
2
3
1:Clear, 2:Mist Cloudy, 3:Light Snow and Rain

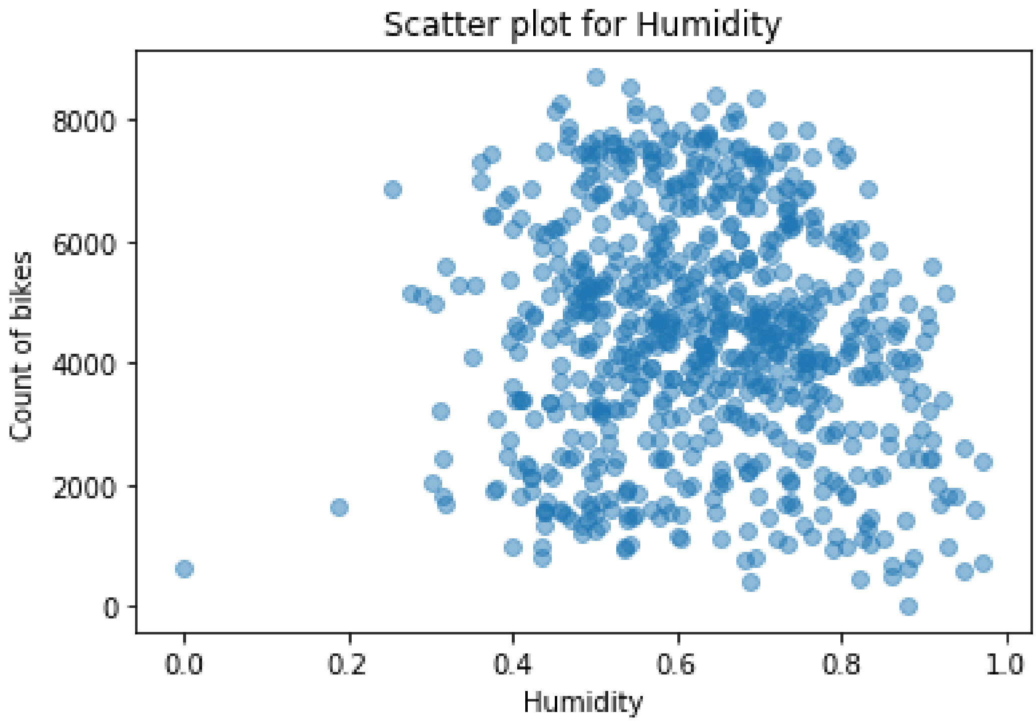

Season
175
150
125
100
75
50
25
0
Count
3
2
1
4
1:Fall, 2:Summer, 3:Winter, 4:spring

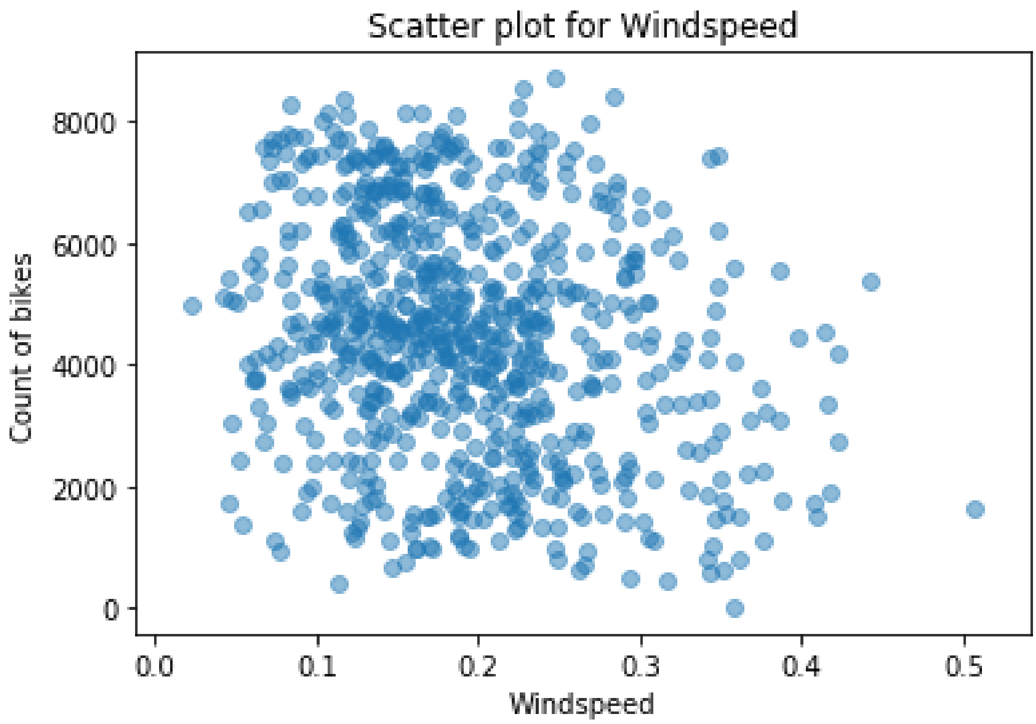


Distribution of categorical variables using bar plots


Scatter plot for Feel temperature
Count of bikes
Feel Temperature
0
2000
4000
6000
8000
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


Scatter plot for temperature
Count of bikes
Temperature
0
2000
4000
6000
8000
0.2
0.4
0.6
0.8

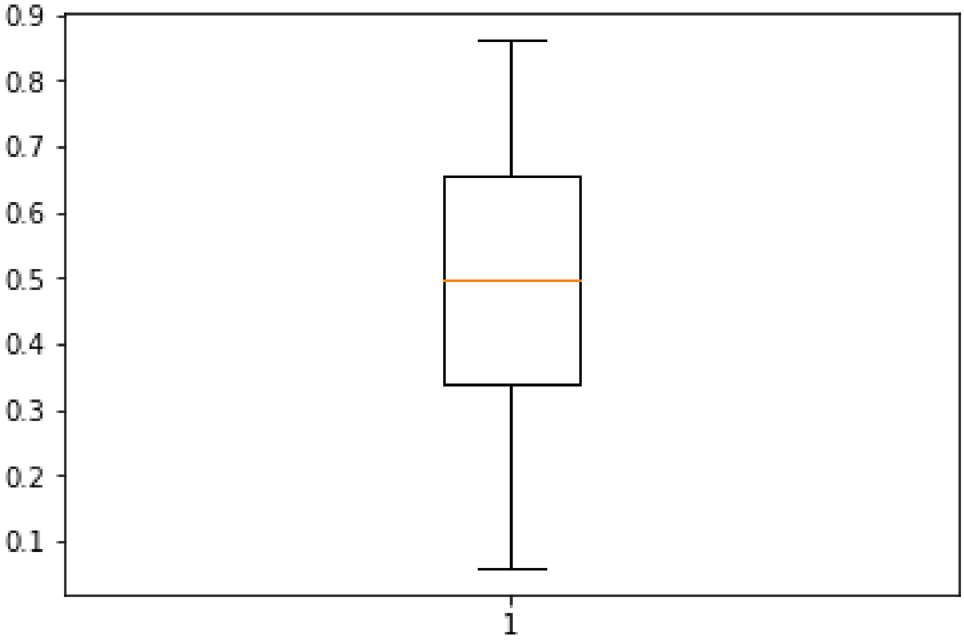


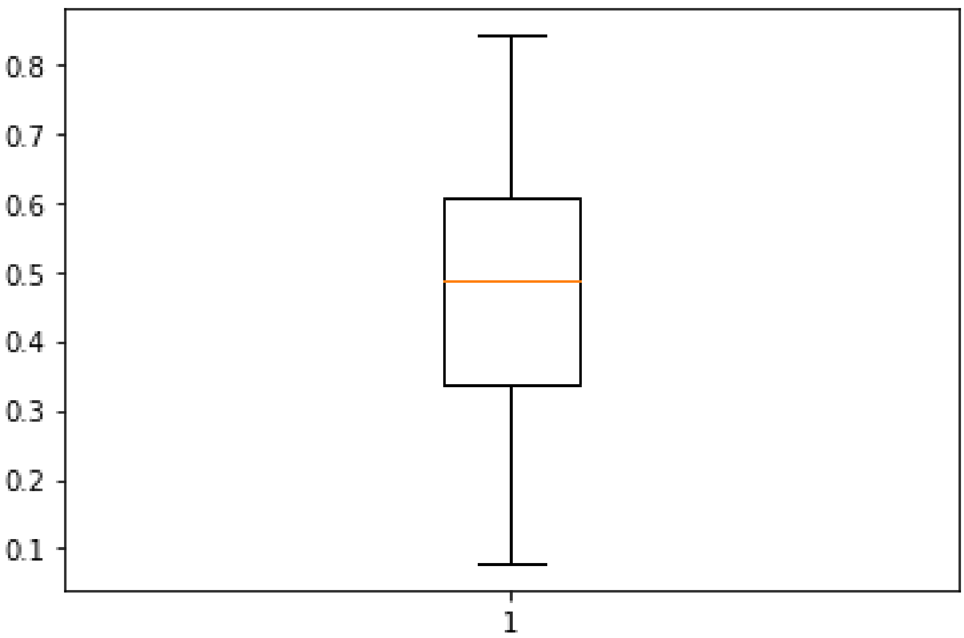


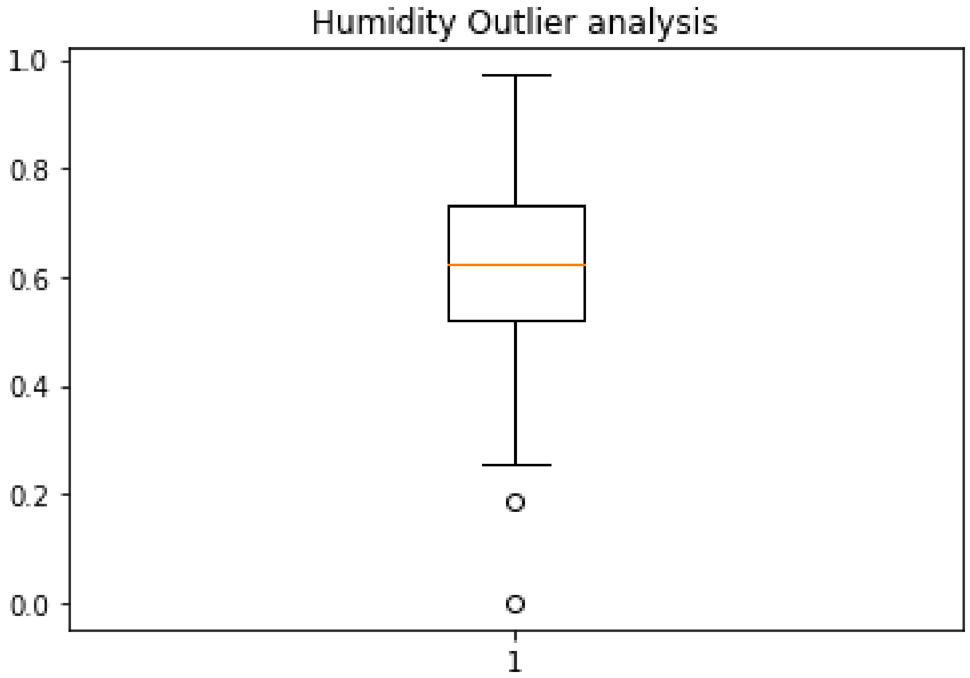
Scatter plot for continuous variables

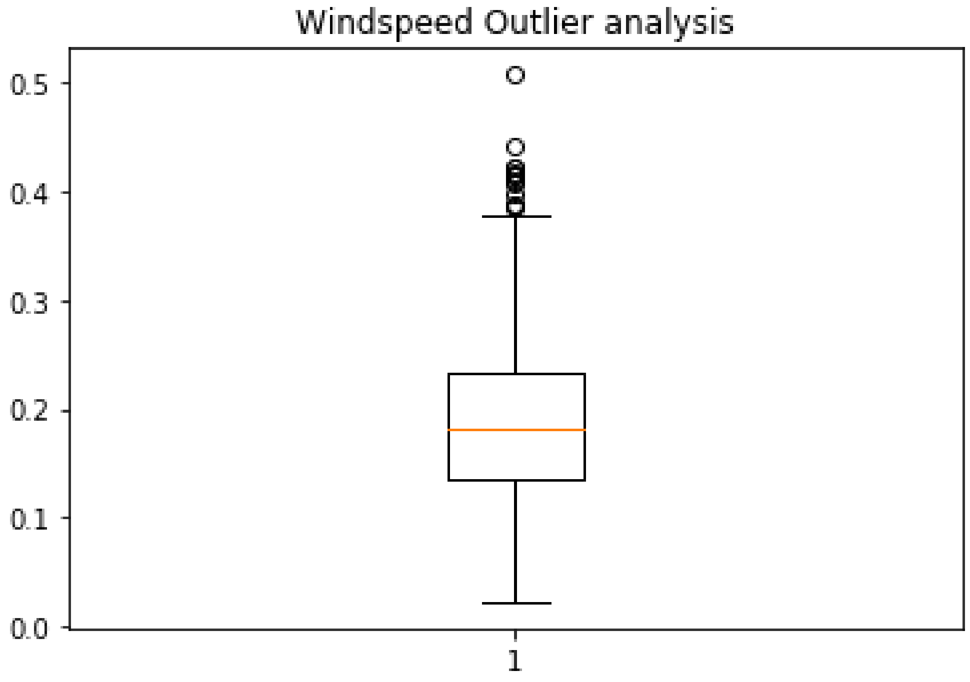
Temp Outlier analysis

atemp Outlier analysis





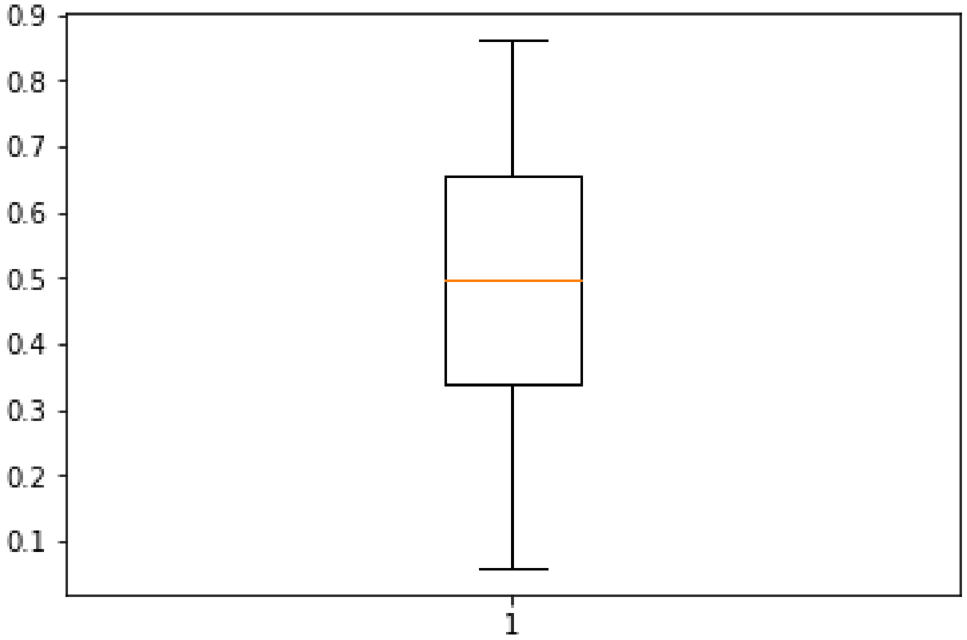


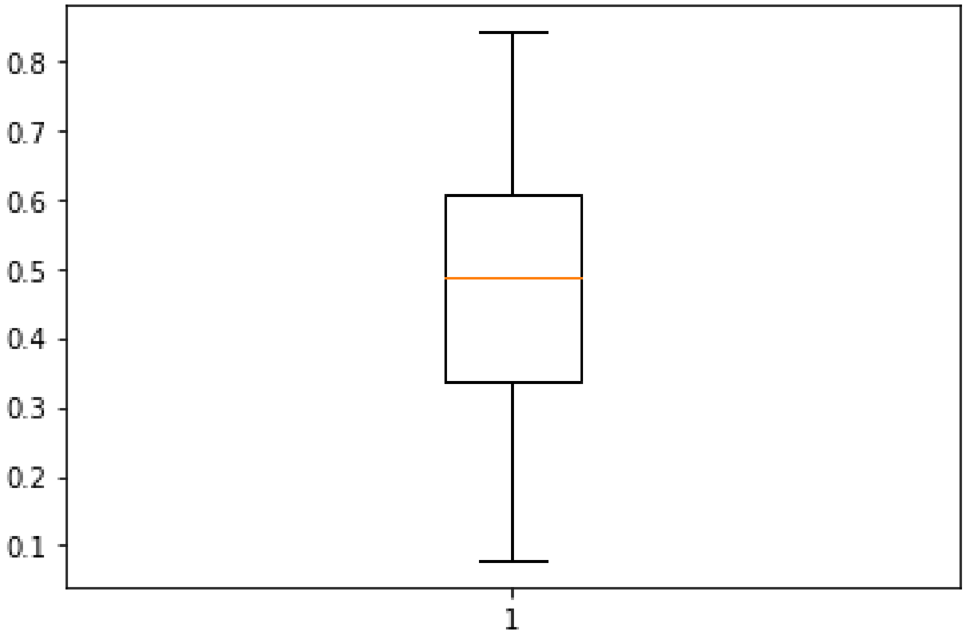


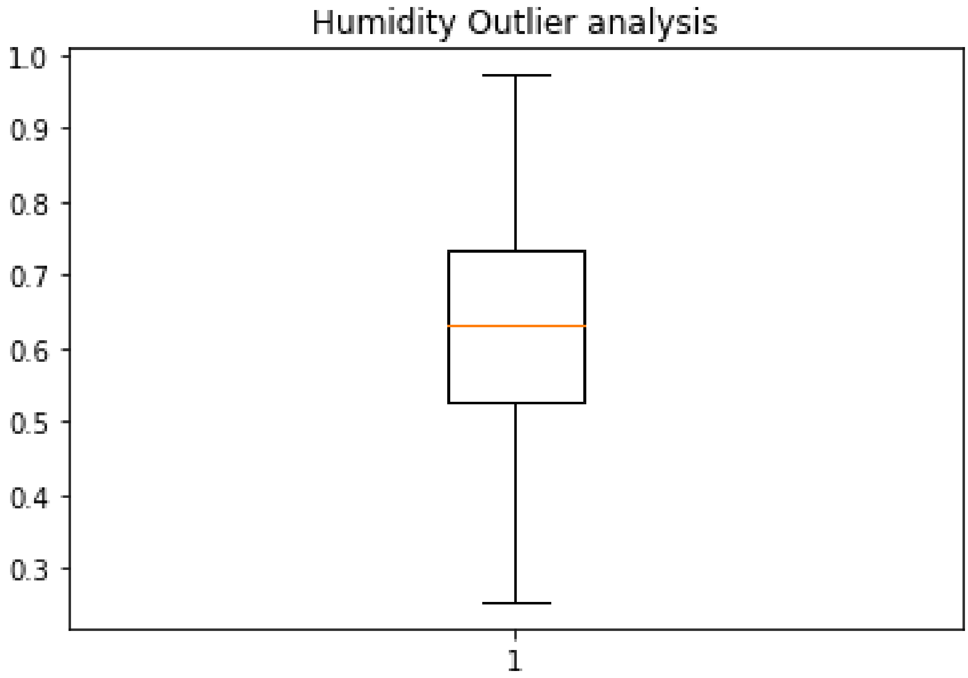
Boxplot of continuous variables

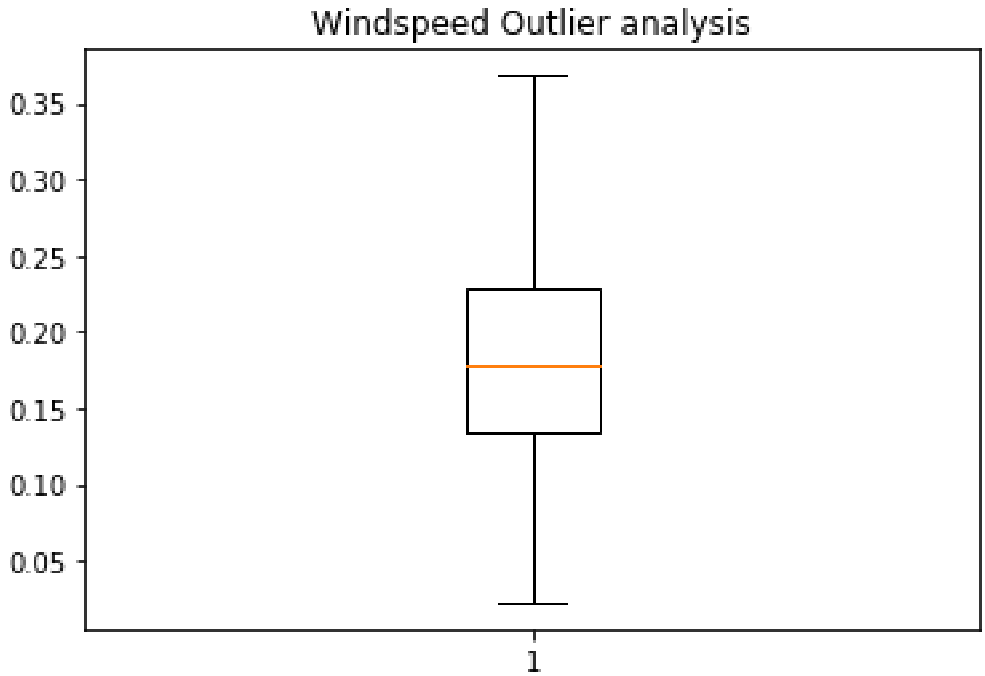
Temp Outlier analysis

atemp Outlier analysis

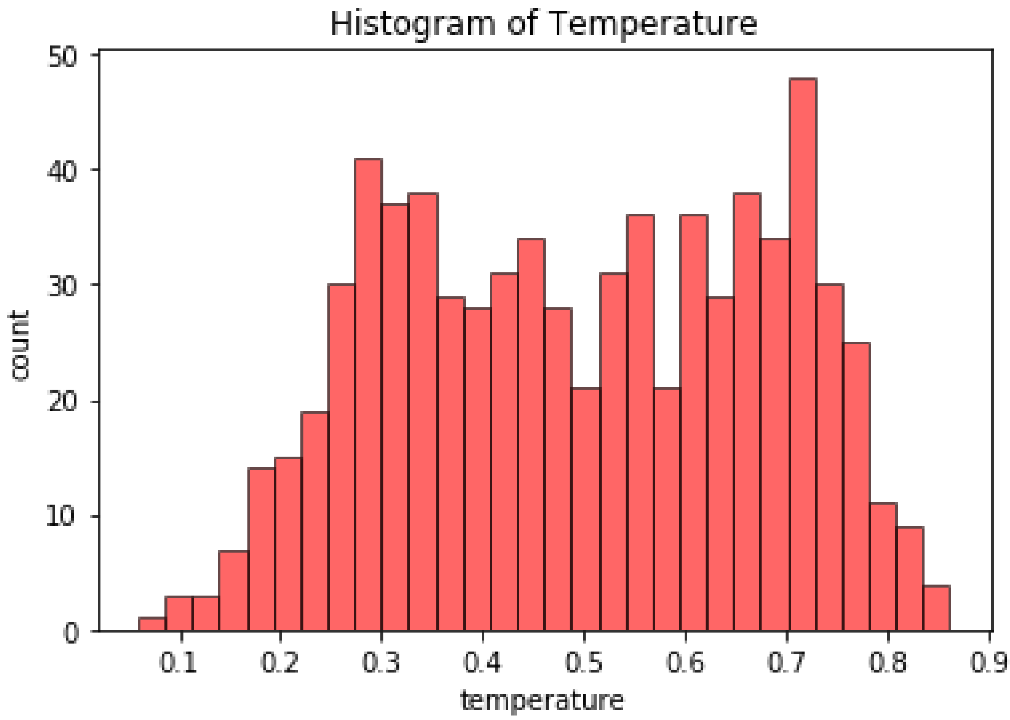


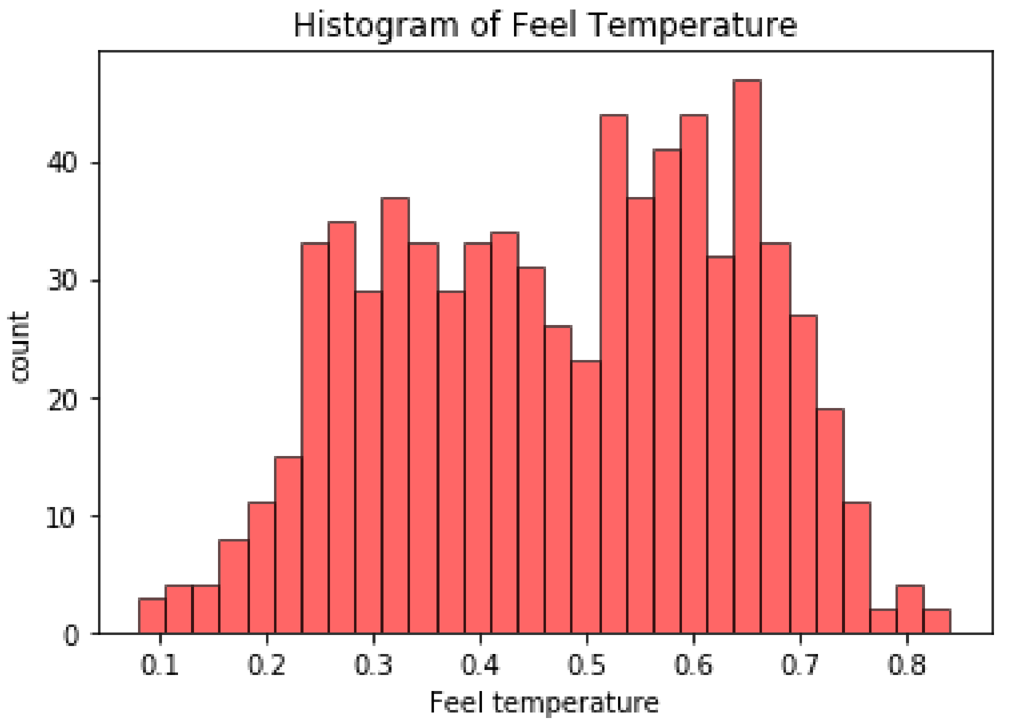


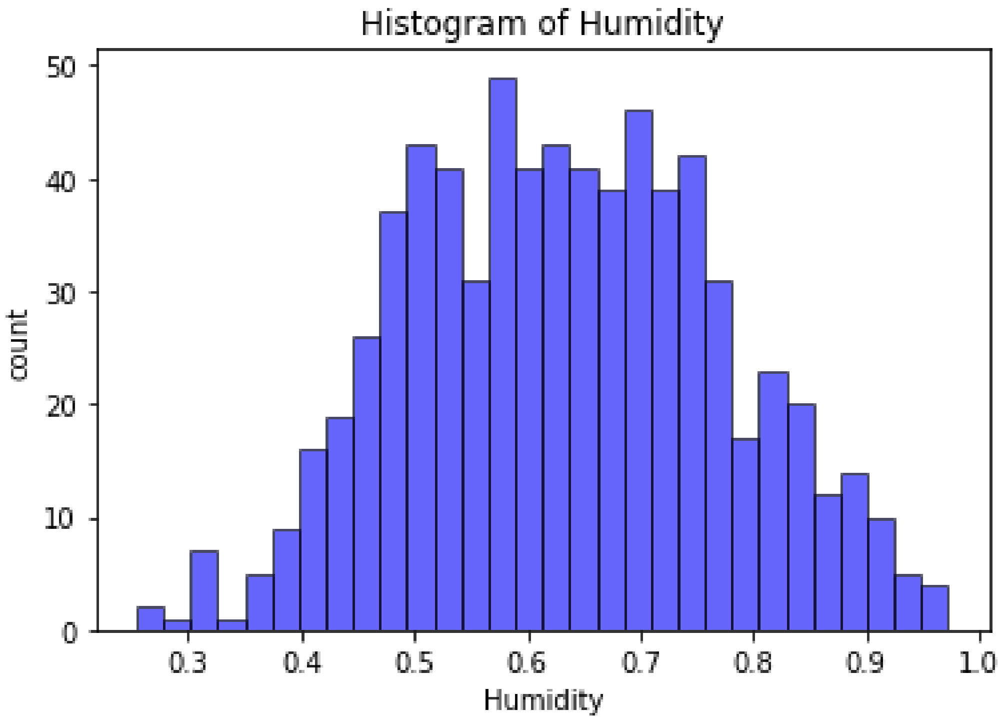

Humidity Outlier analysis
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
1

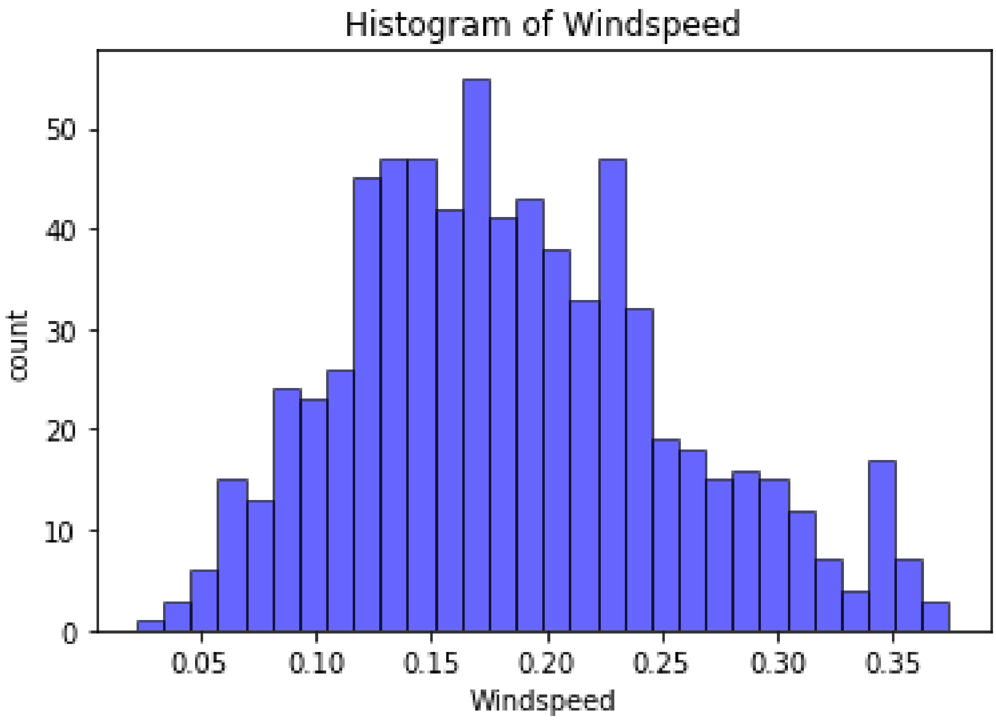

Windspeed Outlier analysis
0.35
0.30
0.25
0.20
0.15
0.10
0.05
1

Boxplot of continuous variables after removal of outliers

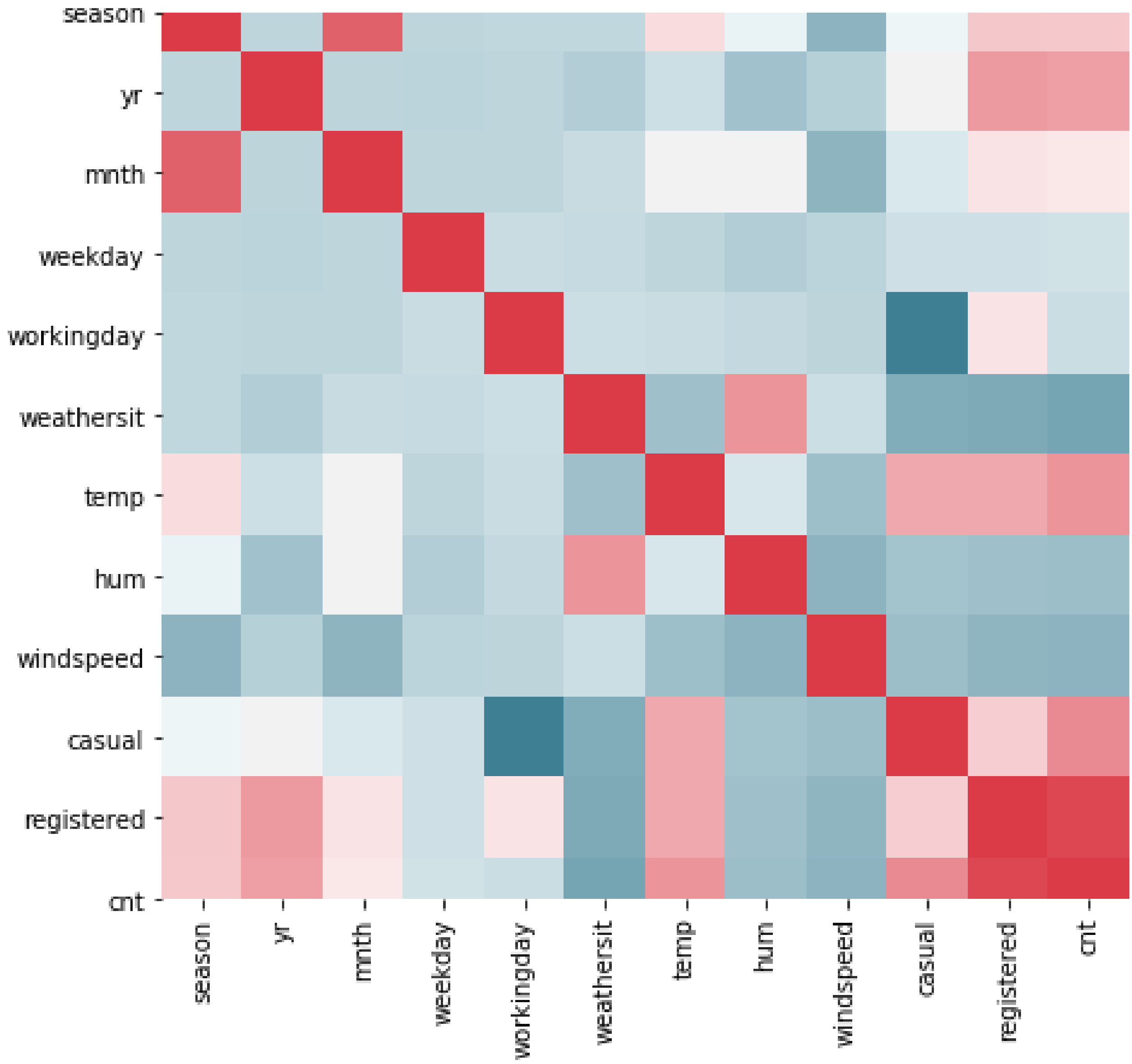

Histogram of Temperature
count
temperature
0
10
20
30
40
50
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9


Histogram of Feel Temperature
count
Feel temperature
0
10
20
30
40
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8





I Histogram of continuous variables after removal outlier.



Correlation plot of all the variables

Chapter 6: Python Code

```
####### Importing necessary libraries ######
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

```
import os
import numpy as np
import seaborn as sns
from scipy.stats import chi2_contingency
import statsmodels.api as sm
#from sklearn.cross_validation import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.tree import export_graphviz
from sklearn import tree

# setting the current working directory
os.chdir(r"C:\Users\Bhavesh\Desktop\Data scientist\Project")
dataset = pd.read_csv('day.csv', encoding = "ISO-8859-1",sep=',')
dataset.head(5)

# plotting the histogram for all continuous variable
num_bins = 30
plt.hist(dataset['temp'],  num_bins,facecolor='red', alpha=0.6, histtype='bar', ec='black')
plt.xlabel('temperature')
plt.ylabel('count')
plt.title(r'Histogram of Temperature')

num_bins = 30
plt.hist(dataset['hum'],  num_bins,facecolor='red', alpha=0.6,histtype='bar', ec='black')
plt.xlabel('Feel temperature')
plt.ylabel('count')
```

```
plt.title(r'Histogram of Feel Temperature')

num_bins = 30
plt.hist(dataset['hum'],  num_bins,facecolor='blue', alpha=0.6,histtype='bar', ec='black')
plt.xlabel('Humidity')
plt.ylabel('count')
plt.title(r'Histogram of Humidity')

num_bins = 30
plt.hist(dataset['windspeed'],  num_bins,facecolor='blue', alpha=0.6,histtype='bar', ec='black')
plt.xlabel('Windspeed')
plt.ylabel('count')
plt.title(r'Histogram of Windspeed')

# Plotting the bar graph for all categorical variable
#plt.bar(dataset['weathersit'] ,alpha=0.5)
dataset['weathersit'].value_counts().plot(kind='bar')
plt.title('Weather Situation')
plt.xlabel(' 1:Clear, 2:Mist Cloudy, 3:Light Snow and Rain')
plt.ylabel('Count')

#plt.bar(dataset['season'], dataset['cnt'], align='center', alpha=0.6)
dataset['season'].value_counts().plot(kind='bar')
plt.title('Season')
plt.xlabel(' 1:Fall, 2:Summer, 3:Winter, 4:spring')
plt.ylabel('Count')

dataset['workingday'].value_counts().plot(kind='bar')
plt.title('Working day')
```

```
plt.xlabel(' 1:Holiday , 2:Working day')
plt.ylabel('Count')

dataset['weekday'].value_counts().plot(kind='bar')
plt.title('Weekday')
plt.xlabel(' 0:snuday---6:Saturday')
plt.ylabel('Count')

# Plotting the scatter plot of conti variables
colors = (0,0,0)
plt.scatter(dataset['temp'], dataset['cnt'], alpha=0.5)
plt.title("Scatter plot for temperature")
plt.xlabel("Temperature")
plt.ylabel("Count of bikes")

plt.scatter(dataset['atemp'], dataset['cnt'], alpha=0.5)
plt.title("Scatter plot for Feel temperature")
plt.xlabel("Feel Temperature")
plt.ylabel("Count of bikes")

plt.scatter(dataset['hum'], dataset['cnt'], alpha=0.5)
plt.title("Scatter plot for Humidity")
plt.xlabel("Humidity")
plt.ylabel("Count of bikes")

plt.scatter(dataset['windspeed'], dataset['cnt'], alpha=0.5)
plt.title("Scatter plot for Windspeed")
plt.xlabel("Windspeed")
plt.ylabel("Count of bikes")

# BOX plot for conti variables
```

```
plt.boxplot(dataset['temp'])
plt.title("Windspeed Outlier analysis")

plt.boxplot(dataset['atemp'])
plt.title("Windspeed Outlier analysis")

plt.boxplot(dataset['windspeed'])
plt.title("Windspeed Outlier analysis")

plt.boxplot(dataset['hum'])
plt.title("Windspeed Outlier analysis")

# storing the conti and categ variables in an array and
Removing the observations lying below and beyong IQR
contivar = ["temp","atemp","hum","windspeed"]
catvar = ["season","yr","mnth","weekday","holiday","workingday","weathersit"]
for i in contivar:
    print(i)
    q75,q25 = np.percentile(dataset.loc[:,i], [75,25])
    iqr = q75 - q25
    min = q25 - (iqr*1.5)
    max = q75 + (iqr*1.5)
    print(min)
    print(max)
    dataset1 = dataset1.drop(dataset[dataset1.loc[:,i] < min].index)
    dataset1 = dataset1.drop(dataset[dataset1.loc[:,i] > max].index)

dataset1.shape

# Now again ploting the histogram for conti variables for validating skewness in
data
```

```
num_bins = 30

plt.hist(dataset1['hum'],  num_bins,facecolor='blue', alpha=0.6,histtype='bar', ec='black')

plt.xlabel('Humidity')

plt.ylabel('count')

plt.title(r'Histogram of Humidity')


num_bins = 30

plt.hist(dataset['windspeed'],  num_bins,facecolor='blue', alpha=0.6,histtype='bar', ec='black')

plt.xlabel('Windspeed')

plt.ylabel('count')

plt.title(r'Histogram of Windspeed')


# ploting the heapmaps

dataset_corr = dataset.loc[:,conti]

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

corr = dataset_corr.corr()

sns.heatmap(corr, mask=np.zeros_like(corr,dtype=np.bool),cmap=sns.diverging_palette(220,10,as_cmap=True),square=True,ax=ax)


dataset_corr = dataset.loc[:,catvar]

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

corr = dataset_corr.corr()

sns.heatmap(corr, mask=np.zeros_like(corr,dtype=np.bool),cmap=sns.diverging_palette(220,10,as_cmap=True),square=True,ax=ax)


# dropping the correlated variables.

dataset = dataset.drop(['atemp,holiday,instant'], axis=1)


# Linear regresssion model
```

```
train1, test1 = train_test_split(dataset,test_size=0.2)
model = sm.OLS(train1.iloc[:,11], train1.iloc[:,0:11].astype(float)).fit()
model.summary()
prediction_LR = model.predict(test.iloc[:,0:11])
prediction_LR
test.iloc[:,12]

def MAPE(y_true,y_pred):
    mape = np.mean(np.abs((y_true - y_pred)/ y_true))
    return mape

def MAE(y_true,y_pred):
    mae = np.mean(np.abs((y_true - y_pred)))
    return mae

MAPE(test.iloc[:,11],prediction_LR)
MAE(test.iloc[:,11],prediction_LR)

# Decision Tree Model
x = dataset2.values[:,0:11]
y = dataset2.values[:,11]
#train1, test1 = train_test_split(dataset2,test_size=0.2)
x_train , x_test, y_train, y_test =  train_test_split(x,y,test_size = 0.2)
#fit_DT  =
DecisionTreeRegressor(max_depth=2).fit(train1.iloc[:,0:11],train1.iloc[:,11])
fit_DT  = DecisionTreeRegressor(max_depth=2).fit(x_train, y_train)
#predict_DT = fit_DT.predict(test1.iloc[:,0:11])
predict_DT = fit_DT.predict(x_test)
MAPE(test1.iloc[:,11],predict_DT)
MAE(test1.iloc[:,11],predict_DT)
```

```
dotfile = open("pt3.dot", 'w')

df = tree.export_graphviz(fit_DT, out_file = dotfile, feature_names = x_train.columns)
```

```
# Random forest Model

train2, test2 = train_test_split(dataset3,test_size=0.2)

fit_RF = RandomForestRegressor(n_estimators=700).fit(train2.iloc[:,0:11],train2.iloc[:,11])

predict_RF = fit_RF.predict(test2.iloc[:,0:11])

MAPE(test2.iloc[:,11],predict_RF)

MAE(test2.iloc[:,11],predict_RF)
```

Chapter 7: R Code

```
#CHECK THE DISTRIBUTION OF CATEGORICAL DATA USING BAR

GRAPH

BAR1 = GGPLOT(DATA = DAY, AES(X = ACTUAL_SEASON)) + GEOM_BAR() + GGTITLE("COUNT OF SEASON")

BAR2 = GGPLOT(DATA = DAY, AES(X = ACTUAL_WEATHERSIT)) + GEOM_BAR() + GGTITLE("COUNT OF

WEATHER")

BAR3 = GGPLOT(DATA = DAY, AES(X = ACTUAL_HOLIDAY)) + GEOM_BAR() + GGTITLE("COUNT OF HOLIDAY") BAR4 = GGPLOT(DATA = DAY, AES(X = WORKINGDAY)) + GEOM_BAR() + GGTITLE("COUNT OF WORKING DAY")

GRIDEXTRA::GRID.ARRANGE(BAR1,BAR2,BAR3

,BAR4,NCOL=2) #CHECK THE DISTRIBUTION

OF NUMERICAL DATA USING HISTOGRAM
```

HIST1 = GGPLOT(DATA = DAY, AES(X =ACTUAL_TEMP)) + GGTITLE("DISTRIBUTION OF TEMPERATURE") +

GEOM_HISTOGRAM(BINS = 25)

HIST2 = GGPLOT(DATA = DAY, AES(X =ACTUAL_HUM)) + GGTITLE("DISTRIBUTION OF HUMIDITY") +

GEOM_HISTOGRAM(BINS = 25)

HIST3 = GGPLOT(DATA = DAY, AES(X =ACTUAL_FEEL_TEMP)) + GGTITLE("DISTRIBUTION OF FEEL TEMPERATURE") + GEOM_HISTOGRAM(BINS = 25)

HIST4 = GGPLOT(DATA = DAY, AES(X =ACTUAL_WINDSPEED)) + GGTITLE("DISTRIBUTION OF WINDSPEED") +

GEOM_HISTOGRAM(BINS = 25)

GRIDEXTRA::GRID.ARRANGE(HIST1,HIST2,HIST

3,HIST4,NCOL=2) #CHECK THE DISTRIBUTION

OF NUMERICAL DATA USING SCATTERPLOT

SCAT1 = GGPLOT(DATA = DAY, AES(X =ACTUAL_TEMP, Y = CNT)) + GGTITLE("DISTRIBUTION OF

TEMPERATURE") + GEOM_POINT() + XLAB("TEMPERATURE") + YLAB("BIKE COUNT")

SCAT2 = GGPLOT(DATA = DAY, AES(X =ACTUAL_HUM, Y = CNT)) + GGTITLE("DISTRIBUTION OF HUMIDITY") +

GEOM_POINT(COLOR="RED") + XLAB("HUMIDITY") + YLAB("BIKE COUNT")

SCAT3 = GGPLOT(DATA = DAY, AES(X =ACTUAL_FEEL_TEMP, Y = CNT)) + GGTITLE("DISTRIBUTION OF FEEL TEMPERATURE") + GEOM_POINT() + XLAB("FEEL TEMPERATURE") + YLAB("BIKE COUNT")

SCAT4 = GGPLOT(DATA = DAY, AES(X =ACTUAL_WINDSPEED, Y = CNT)) + GGTITLE("DISTRIBUTION OF

WINDSPEED") + GEOM_POINT(COLOR="RED") + XLAB("WINDSPEED") + YLAB("BIKE COUNT")

GRIDEXTRA::GRID.ARRANGE(SCAT1,SCAT2,SC

AT3,SCAT4,NCOL=2) #CHECK FOR OUTLIERS

IN DATA USING BOXPLOT

CNAMES = COLNAMES(DAY[,C("ACTUAL_TEMP","ACTUAL_FEEL_TEMP","ACTUAL_WINDSPEED","ACTUAL_HUM")]) 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)

#REMOVE OUTLIERS IN WINDSPEED

VAL = DAY[,19][DAY[,19] %IN%
BOXPLOT.STATS(DAY[,19])$OUT] DAY =
DAY[WHICH(!DAY[,19] %IN% VAL),]

#CHECK FOR MULTICOLLINEARITY USING VIF

DF =
DAY[,C("INSTANT","TEMP","ATEMP","HUM","WI
NDSPEED")] 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 DAY <-

SUBSET(DAY, SELECT = -

C(INSTANT,DTEDAY,ATEMP,CASUAL,REGISTERED,ACTUAL_TEMP,ACTUAL_FEEL_TEMP
,ACTUAL_WINDSPEED,AC
TUAL_HUM,ACTUAL_SEASON,ACTUAL_YR,ACTUAL_HOLIDAY,ACTUAL_WEATHERSIT))

# DECISION TREE        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[,-11])
#CREATE DATAFRAME FOR ACTUAL AND
PREDICTED VALUES
DF = DATA.FRAME("ACTUAL"=TEST[,11], "PRED"=DT_PREDICTIONS)
HEAD(DF)
#CALCULATE MAPE
REGR.EVAL(TRUES = TEST[,11], PREDS = DT_PREDICTIONS, STATS =
C("MAE","MAPE"))
```

#RANDOM FOREST TRAIN THE DATA USING

RANDOM FOREST

RF_MODEL = RANDOMFOREST(CNT~., DATA =

TRAIN, NTREE = 700) #PREDICT THE TEST

CASES

RF_PREDICTIONS = PREDICT(RF_MODEL,

TEST[,-11]) #CREATE DATAFRAME FOR

ACTUAL AND PREDICTED VALUES

DF =
CBIND(DF,RF_PREDICTION
S) HEAD(DF)

#CALCULATE MAPE

REGR.EVAL(TRUES = TEST[,11], PREDS = RF_PREDICTIONS, STATS =
C("MAE","MAPE"))

LINEAR REGRESSION TRAIN THE DATA USING

LINEAR REGRESSION

LR_MODEL = LM(FORMULA = CNT~.,

DATA = TRAIN) #CHECK THE SUMMARY

OF THE MODEL SUMMARY(LR_MODEL)

#PREDICT THE TEST CASES

LR_PREDICTIONS = PREDICT(LR_MODEL,

TEST[,-11]) #CREATE DATAFRAME FOR

ACTUAL AND PREDICTED VALUES

DF =
CBIND(DF,LR_PREDICTIONS
) HEAD(DF)

#CALCULATE MAPE

REGR.EVAL(TRUES = TEST[,11], PREDS = LR_PREDICTIONS, STATS =

C("MAE","MAPE")) #PREDICT A SAMPLE DATA

PREDICT(LR_MODEL,TEST[2,])