Bike Rental

Kapil Bhatt

25 April 2019

Contents:

Chapter 1

Introduction

1.1 Problem Statement

The objective of this project is to forecast the count of bike rentals on a given day based on the environmental conditions. Predicting the count will help the rental company to manage the number of riders on a given day with different conditions.

1.2 Data

Our task here is to build the regression model in order to predict the count of bike rentals in a given day based on the predictors we had. Given below is the sample of data that we are using to predict the bike rentals.

Table1.1 Bike rental samples column 1 to 10.

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957

Table 1.2 Bike rental samples column 11 to 16

atemp	hum	windspeed	casual	registered	cnt
0.363625	0.805833	0.160446	331	654	985
0.353739	0.696087	0.248539	131	670	801
0.189405	0.437273	0.248309	120	1229	1349
0.212122	0.590435	0.160296	108	1454	1562
0.229270	0.436957	0.186900	82	1518	1600

As you can see in the table below we have the following 13 variables (two of them are cut out because there sum is in count). We have to use these variables to predict the bike rental count and do other analysis:

S.no	Predictor
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 : Number of Predictor variables

Chapter 2

Methodology

Preprocessing

Any predictive modelling requires to do exploratory analysis of data, make it clean and suitable to create various machine learning models. This step contains cleaning of data, creating visualization like graphs and plots to gain more insights. The data which is given to us is having numerical variables which are normally distributed. So that's the plus point for us because in most cases like regression requires data to be normally distributed.

2.1 Missing Value Analysis

In this analysis we found out the missing values in our data set and then how to deal with it. Fortunately this dataset didn't have any missing values. If it have we can remove or impute them according to the problem statement and number of missing values.

2.2 Univariate Analysis (Distribution of Numeric variables)

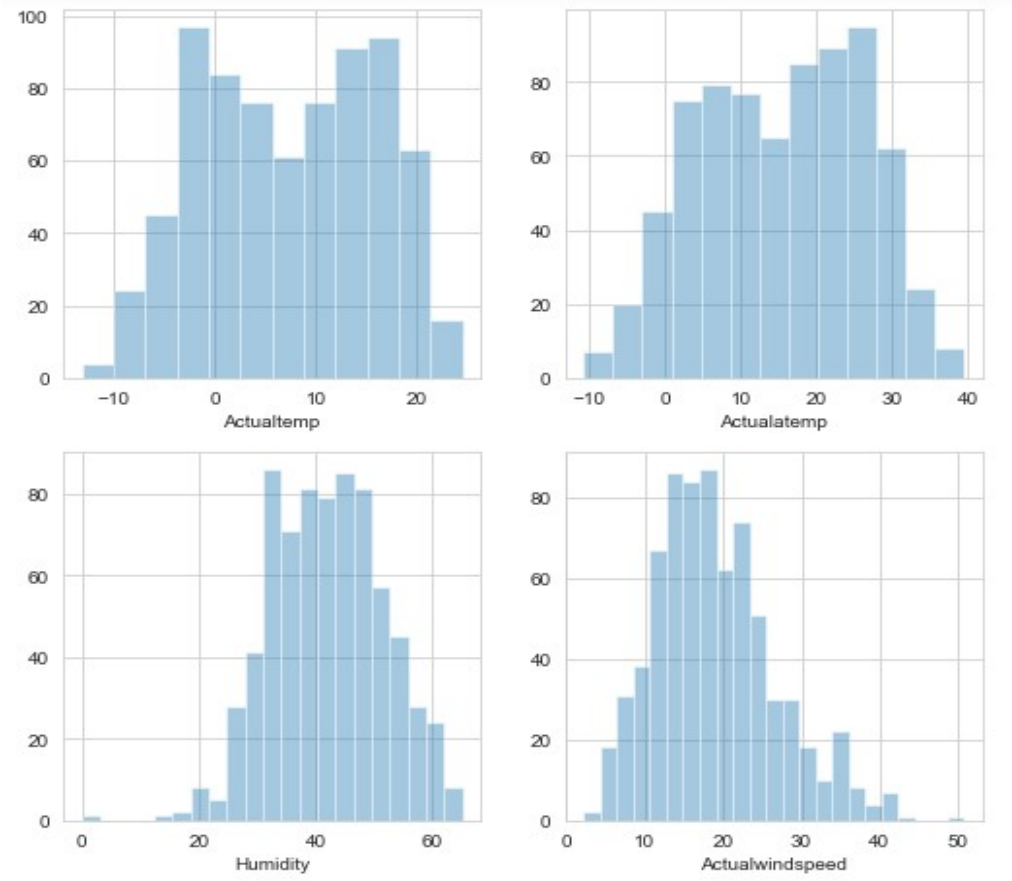


Fig 2.1: Histogram of the Numeric Variables.

As you can see above the actual temp and feeling temp are almost following normal distribution while the humidity and windspeed are slightly skewed, humidity -left skew and windspeed - right skew. This is maybe due to presence of data in extreme ends or outliers.

2.3 Bivariate Analysis (Distribution of Categorical variables with Count)

This analysis is done by plotting bar graphs of categorical variables with respect to the numerical variables.

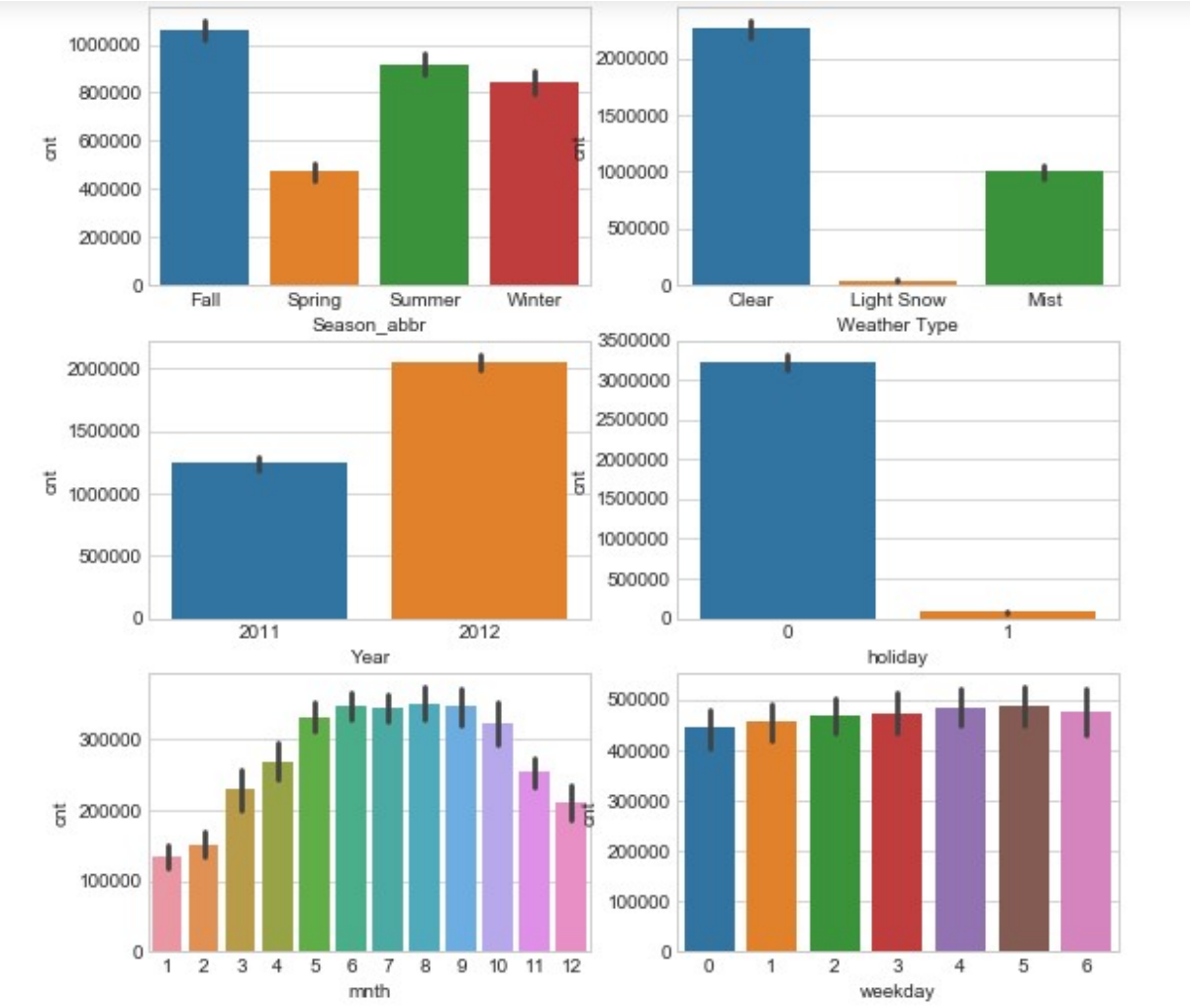


Fig 2.2 :Bar chart for different categorical variables

The bar charts here tells about how the different categories is creating effect on the count of the bike rentals. Let's take a look at first bar graph of Seasons vs. count as you can see that fall

have the highest number of rentals count then all others while Spring have the least. Also from holiday bar graph we get that the people do more bike rentals on workdays than on holidays. Also the bike rentals in 2012 are much larger than the rentals in 2011 this implies the bike rental business is making a growth.

2.4 Bivariate Analysis (Distribution of Continuous variables with count)

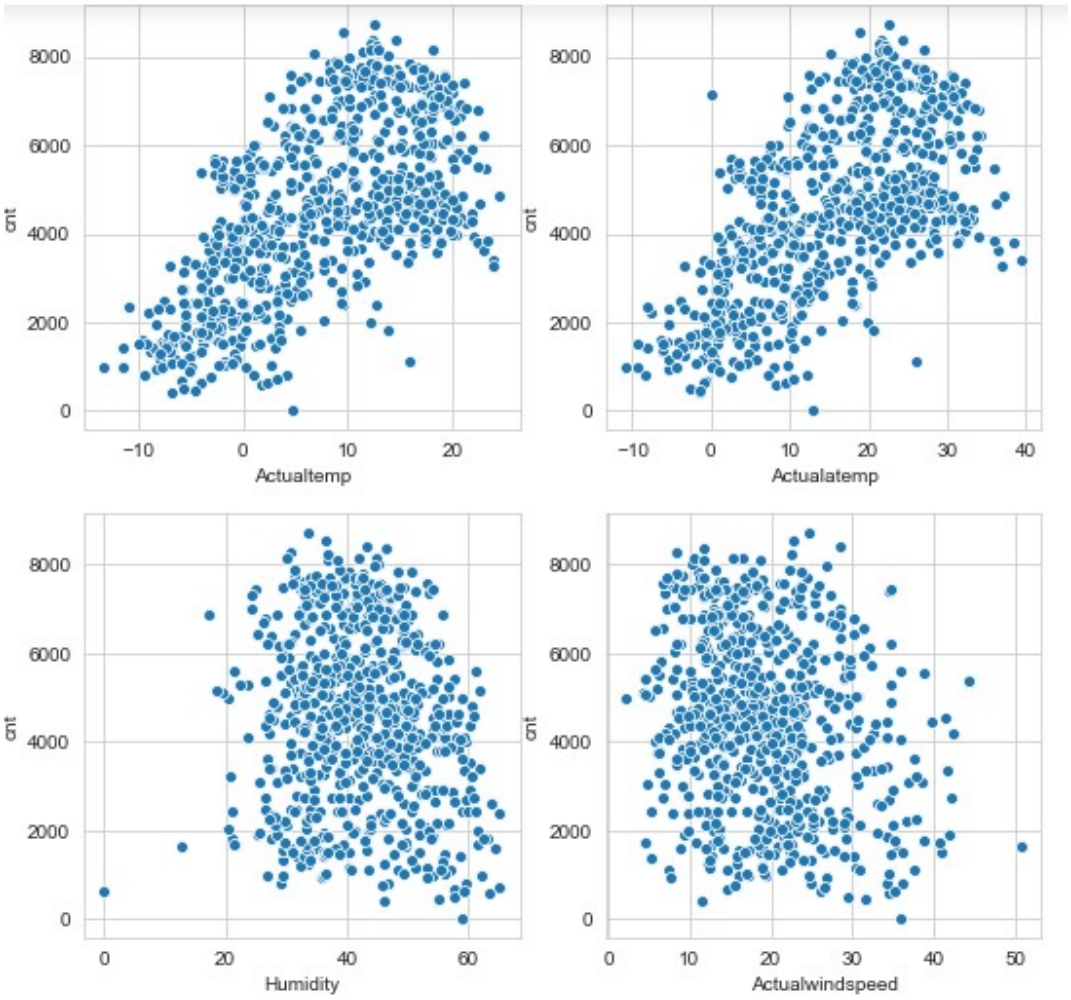


Fig 2.3 Scatterplot of continuous Variables

In this analysis we created scatterplots of all four numerical variables. The temperature and feeling temperature are showing positive strong relationship with count while Humidity and windspeed have weak relationship with the count.

2.5 Outlier Analysis

Outliers are the values that are distant from the main observations. They are detected by box-plots.

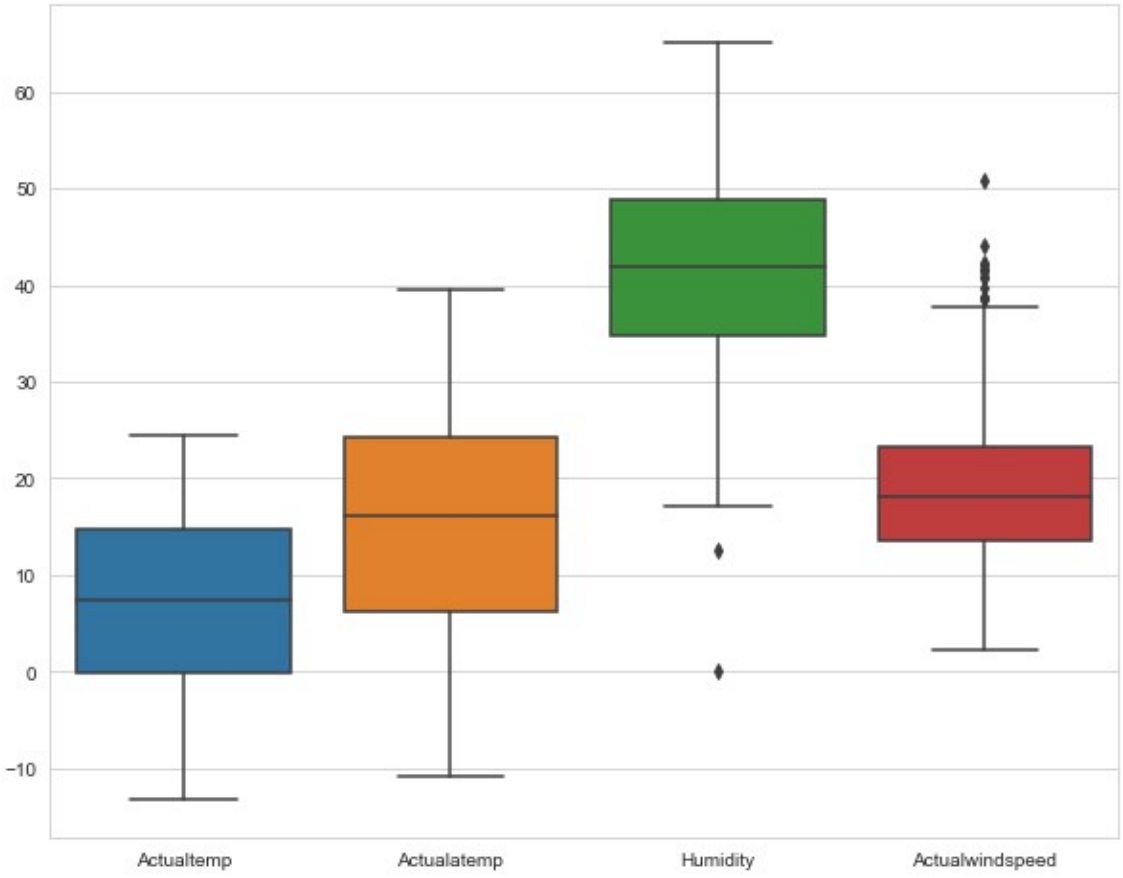


Fig 2.4 Boxplot (without Outlier removal) of continuous variables.

As you can see here windspeed and Humidity have few outliers. They can be removed by creating a function which calculates the interquartile range and minimum and maximum value of the variable. The values which lie beyond the minimum and maximum value is removed as they are outliers.

The box plot after outlier removal is shown in the next page:

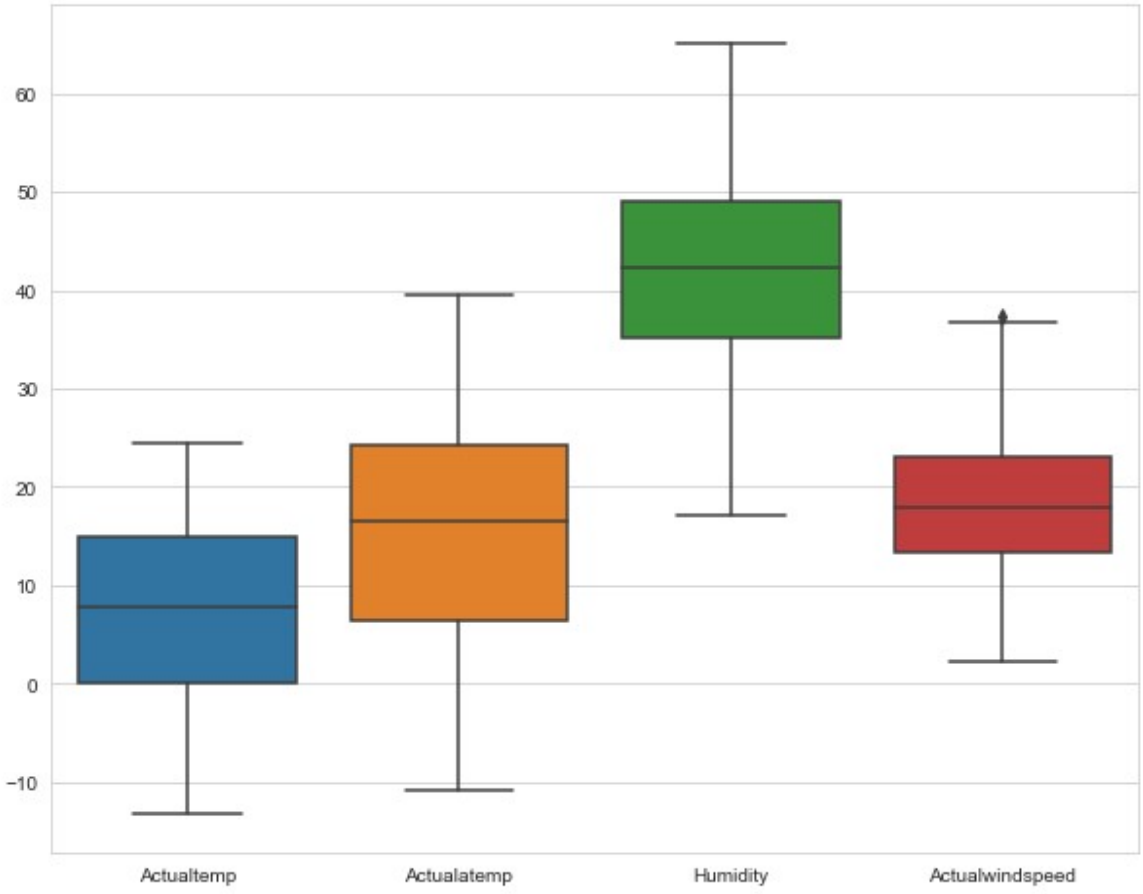


Fig 2.5: Box plot (after outlier removal) of continuous variables.

2.6 Feature Selection:

Feature selection is the most important step of data preprocessing. It is the process where you manually or automatically select the features which have contributed most to your prediction variable. It also reduces overfitting, increases accuracy and save some time to train the model because the data points became few after that.

In this data we uses the correlation scores to check the relationship between independent variables and then remove those variables which have strong relationships among them.

we use the correlation matrix, we can also use the correlation plot.

	temp	atemp	hum	windspeed	casual	registered	cnt
temp	1.000000	0.991738	0.114191	-0.140169	0.539714	0.538095	0.625892
atemp	0.991738	1.000000	0.126587	-0.166038	0.540234	0.541977	0.629204
hum	0.114191	0.126587	1.000000	-0.204496	-0.101439	-0.124701	-0.136621
windspeed	-0.140169	-0.166038	-0.204496	1.000000	-0.146178	-0.203677	-0.216193
casual	0.539714	0.540234	-0.101439	-0.146178	1.000000	0.389848	0.670547
registered	0.538095	0.541977	-0.124701	-0.203677	0.389848	1.000000	0.944581
cnt	0.625892	0.629204	-0.136621	-0.216193	0.670547	0.944581	1.000000

Table 2.1 : Correlation matrix of continuous variables

There is a strong relationship between atemp and temp variable because of their correlation score is too high. so they both are contributing same thing to the dependent variable. Therefore we can remove atemp from the dataset.

The relationship is further confirmed by the scatterplot between these two variables which is below:

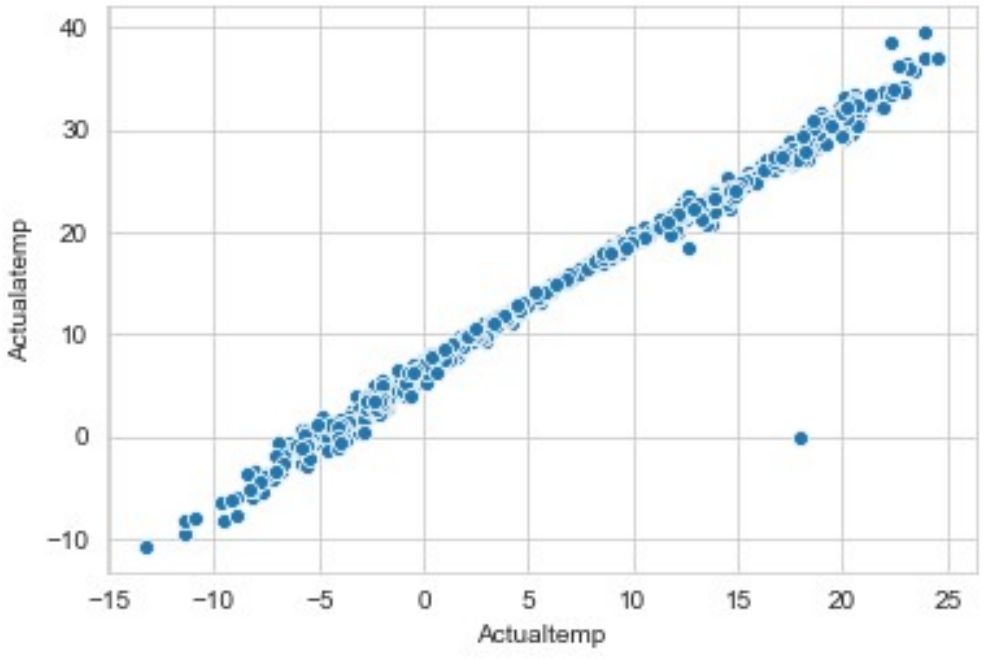


Fig 2.6 : Scatterplot between temp and feeling temp.

In the scatterplot we can see that the temperature have a positively strong relationship with the feeling temperature i.e. atemp**.**

Chi-Square Test:

The Pearson's chi-squared statistical hypothesis is an example of a test for independence between categorical variables. Its mainly done on classification problems for feature selection. I just added a little bit of code snippet of chi square test between the categorical variables of the data. This test based on the null hypothesis or alternate hypothesis. null hypothesis says that the two variables are independent while alternate hypothesis says that they are dependent.

That's the end of data preprocessing. We didn't did feature scaling as the data is normalized by various methods in the data set, few of them simply normalized with the min max scaling and the remaining ones are scaled by the max of them.

Chapter 3

Modeling

3.1 Model Selection

In the problem we need to predict the count of bike rentals which is a continuous variable. So it falls under the forecasting problems. The models we use here are of regressions such as Linear Regression, Decision Tree Regressor and RandomForest. There are many error metrics are available but in this model we will compare the models in terms of R square and mean average percentile error (MAPE).

3.2 Multiple Linear Regression

OLS Regression Results

Dep. Variable:	cnt	**R-squared:**	0.835
Model:	OLS	**Adj. R-squared:**	0.826
Method:	Least Squares	**F-statistic:**	101.9
Date:	Sat, 20 Apr 2019	**Prob (F-statistic):**	2.34e-193
Time:	23:55:08	**Log-Likelihood:**	-4617.7
No. Observations:	573	**AIC:**	9291.
Df Residuals:	545	**BIC:**	9413.
Df Model:	27		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
hum	-1786.8118	352.359	-5.071	0.000	-2478.960	-1094.664
windspeed	-2820.2686	495.436	-5.692	0.000	-3793.467	-1847.070
temp	4645.1269	471.538	9.851	0.000	3718.872	5571.382

FIG 3.1 Multiple Regression Results.

Multiple linear regression tells the relationship between one continuous variables to two or more continuous or categorical independent variables. In the above model R2 Square is 0.835 means we can explain 83.5% of the data from input variables and by looking the p value we can say that this model explained the data well.

3.2 Decision Trees

Decision tree is used for both classification and regression problems. It uses a tree like model where each branch connected with "and" and multiple branches are connected with "or". There are two methods of splitting the tree one is by Information gain and other is by Gini Index. Here the R2_SQUARED value comes out to be 0.876 which means it explained 87.6% of the data from input variables which is better than Linear Regression.

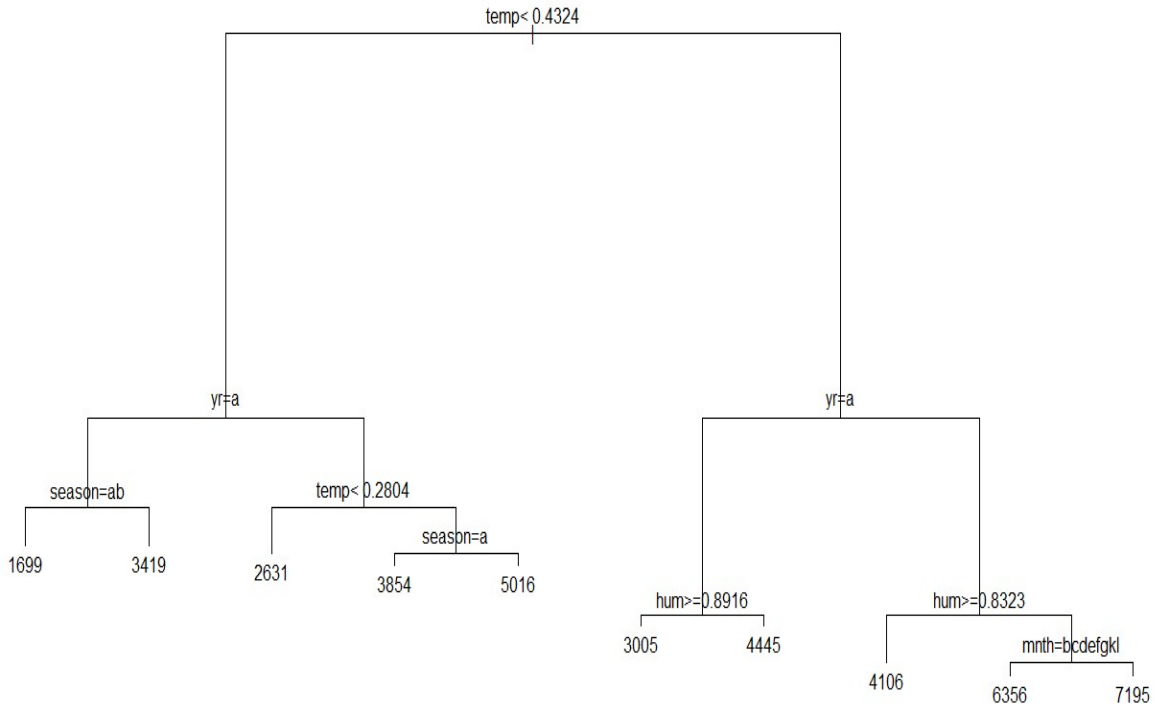


Fig 3.2 Plot of the Decision Tree model.

3.3 Random Forest

Random forest use number of decision trees and then calculate the result based on mean and mode. In classification it uses mode of class output by individual trees while in regression it uses the mean to get the results. It is the combination of weak methods and it feed error from one decision tree to another to get the better results. In predictions a sample is iterated all the trees and the mean vote of all the trees is considered. In our model the number of decision tree we use to create random forest model is 100 and we get R square value of 0.92 which is high as 92.2% of the data is explained by the input variables.

Chapter 4

Conclusion

4.1 Model Evaluation

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 rental the last two don't hold much significance. So we stick to the predictive performance or evaluation of our models using mean absolute percentile error (MAPE).

Predictive performance can be measured by comparing Predictions of the models with real values of the target variables, and calculating some average error measure.

Mean Absolute Percentile Error (MAPE)

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 build.

The function is given by

```
def MAPE(y_true,y_pred):
    mape = np.mean((np.abs(y_true-y_pred)/y_true)*100)
    return mape
```

So based on the above metrics following are the results:

Linear Regression : MAPE - 16.26% -> ACCURACY = 84.74%

Decision Tree : MAPE - 18.05% -> ACCURACY = 81.95%

Random Forest : MAPE - 15.27% -> ACCURACY = 84.73%

Based on the above results we can conclude that Random Forest is the better model for our Analysis, So we choose that model.

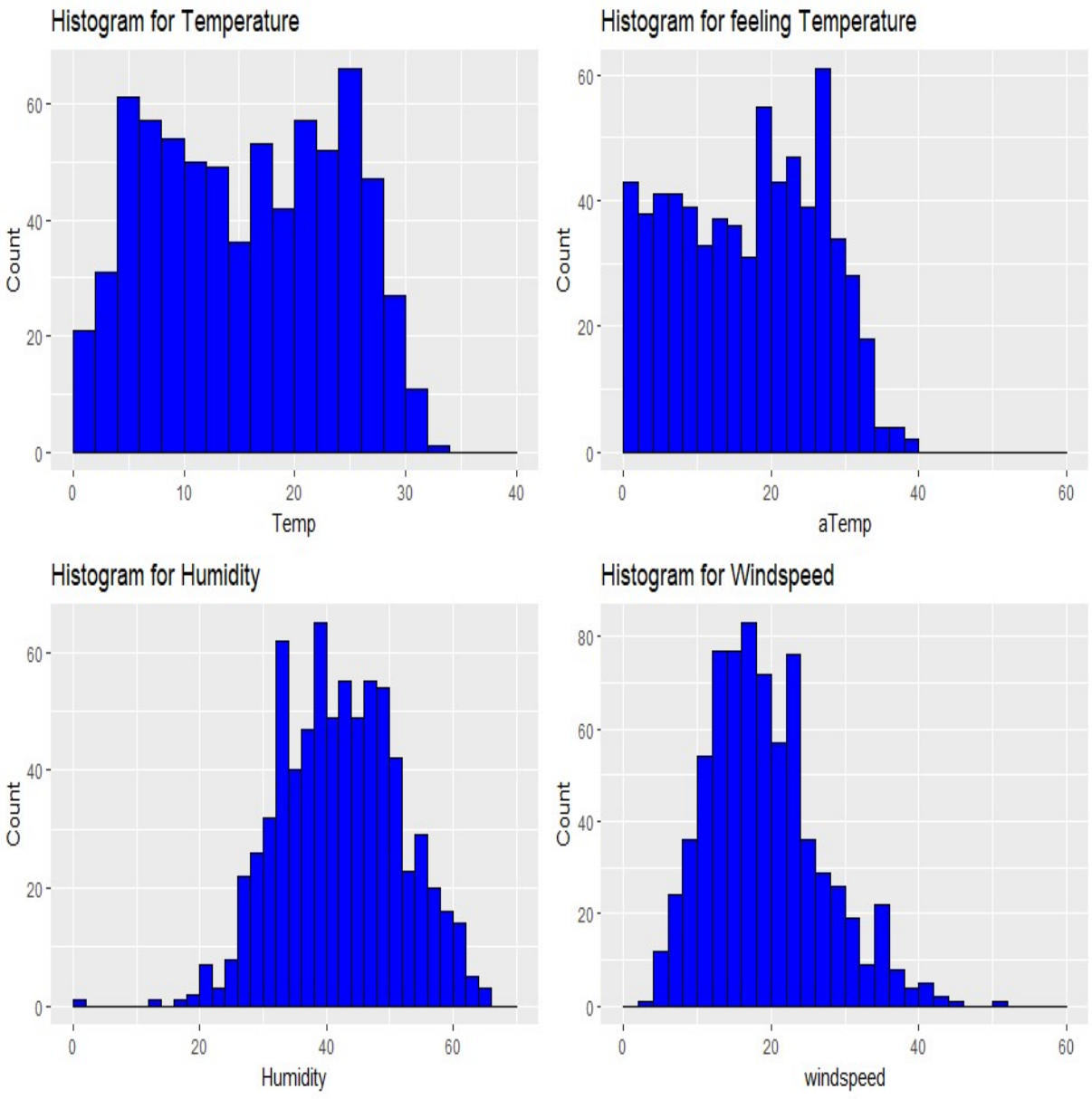
Chapter 5 :

APPENDIX

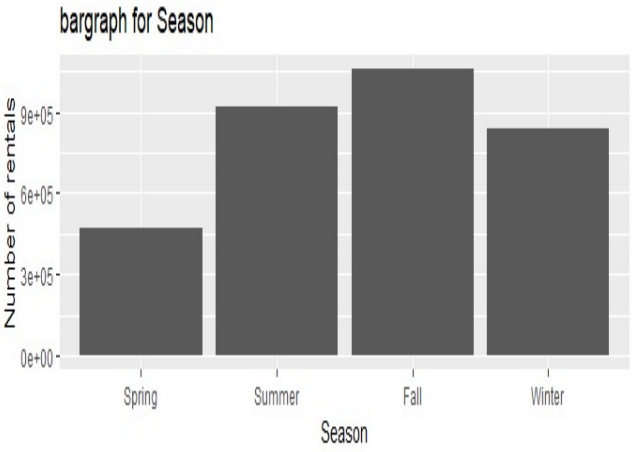
5.1 Extra Figures (visualization generated in R)

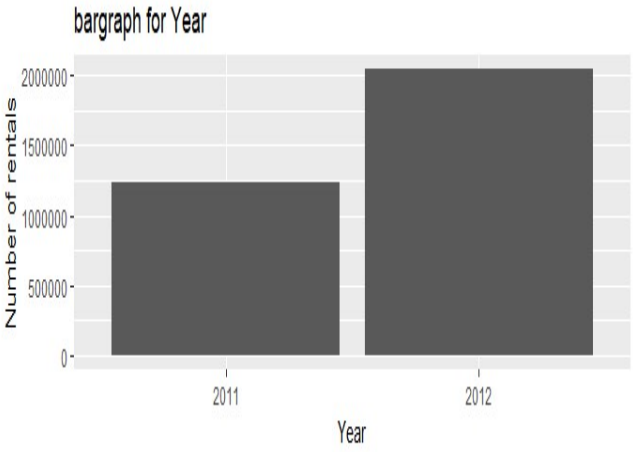
We use the visualization generated in python in the lessons of this report. Now below are the same visualization generated in R.

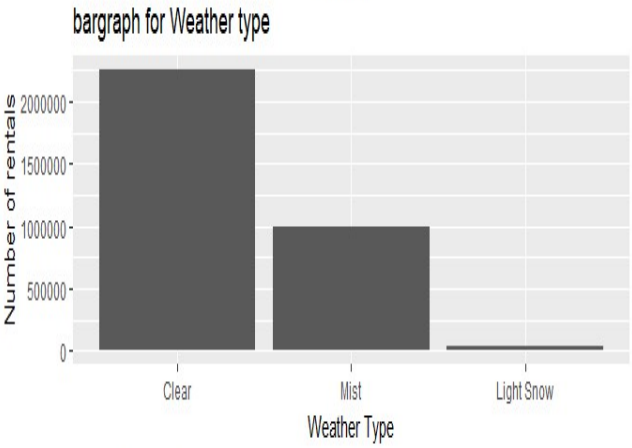
Histogram of continuous variables :

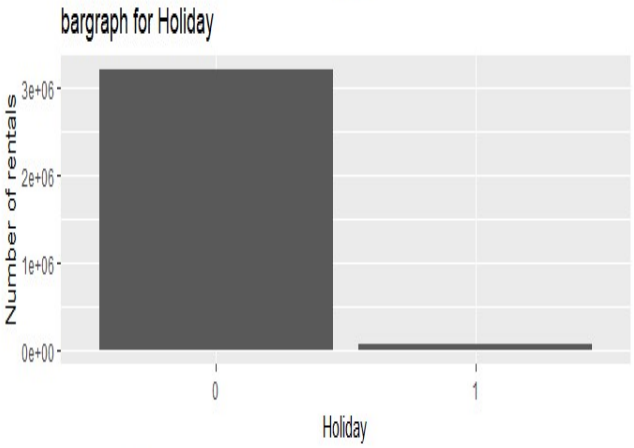


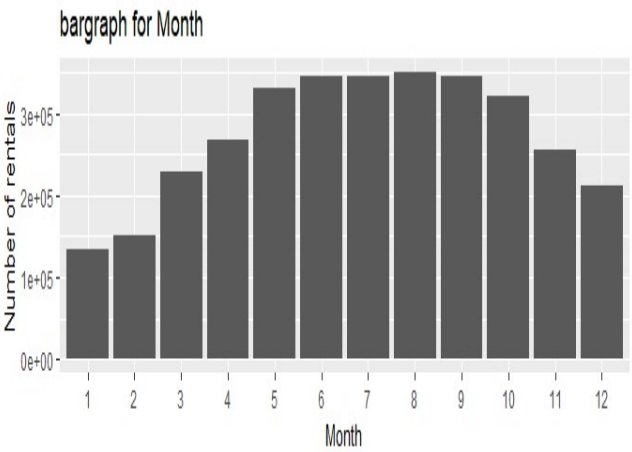
Bar Graph of Categorical Variables with Count :

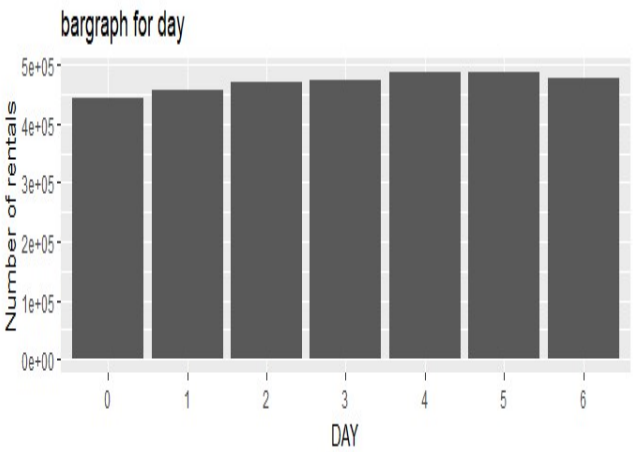




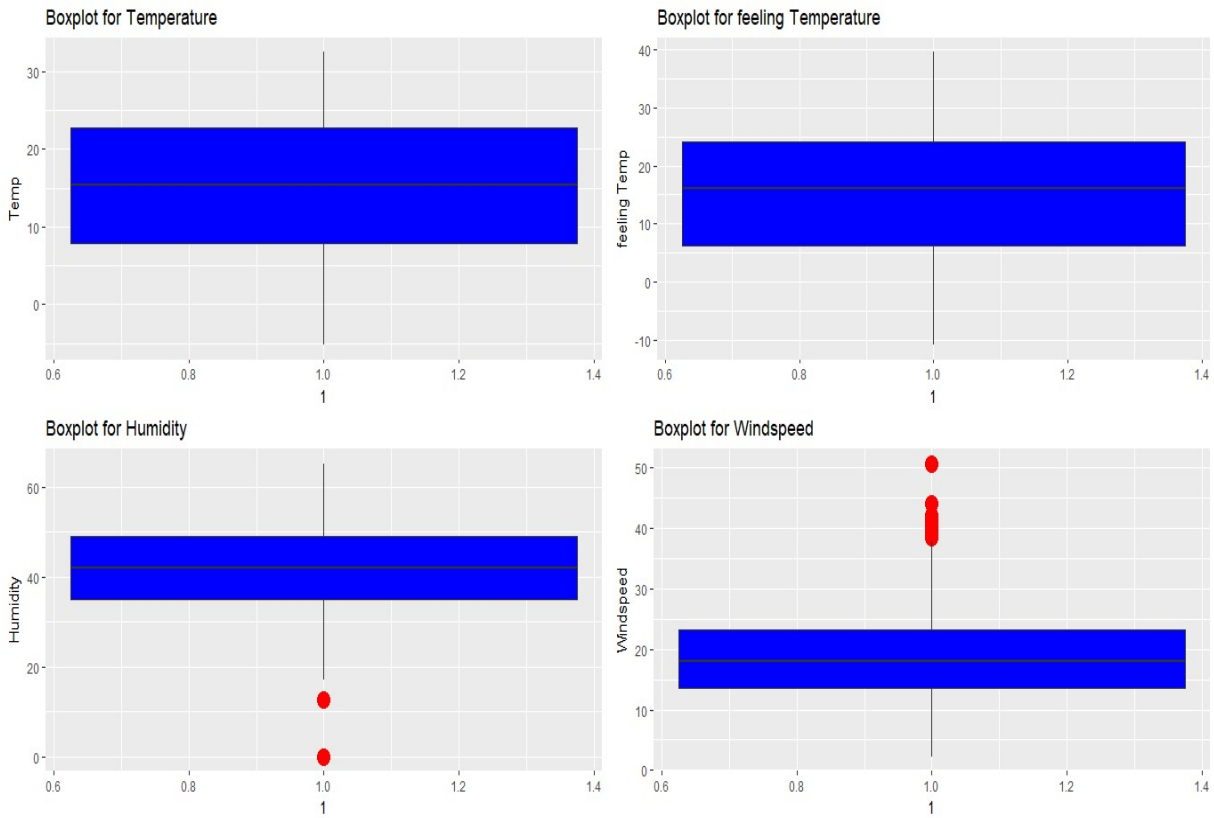




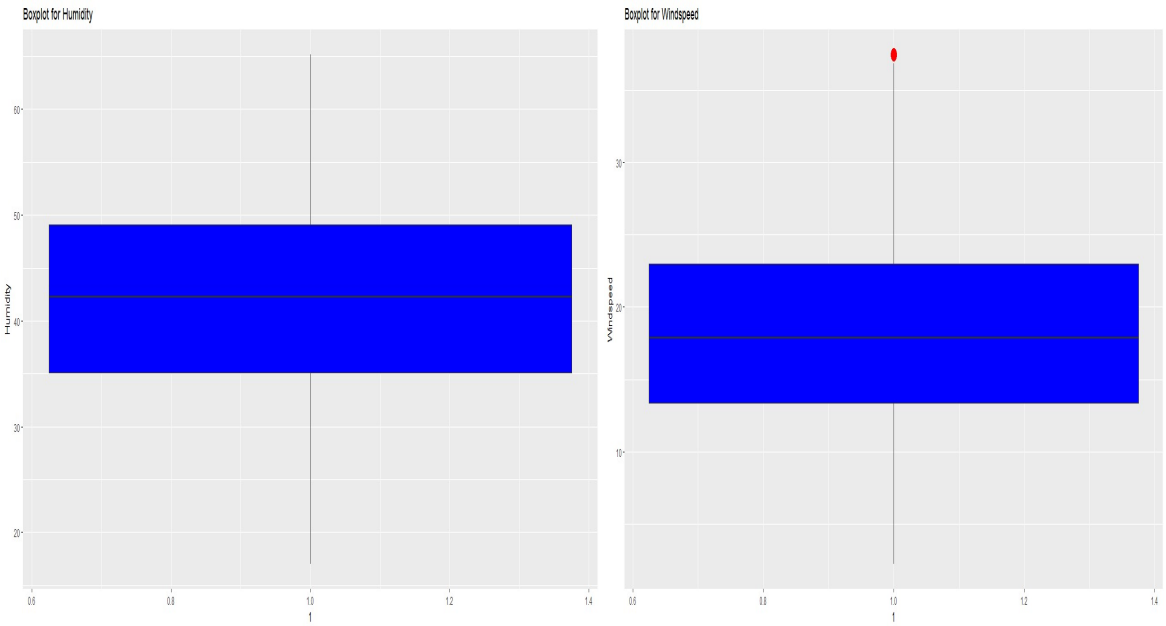




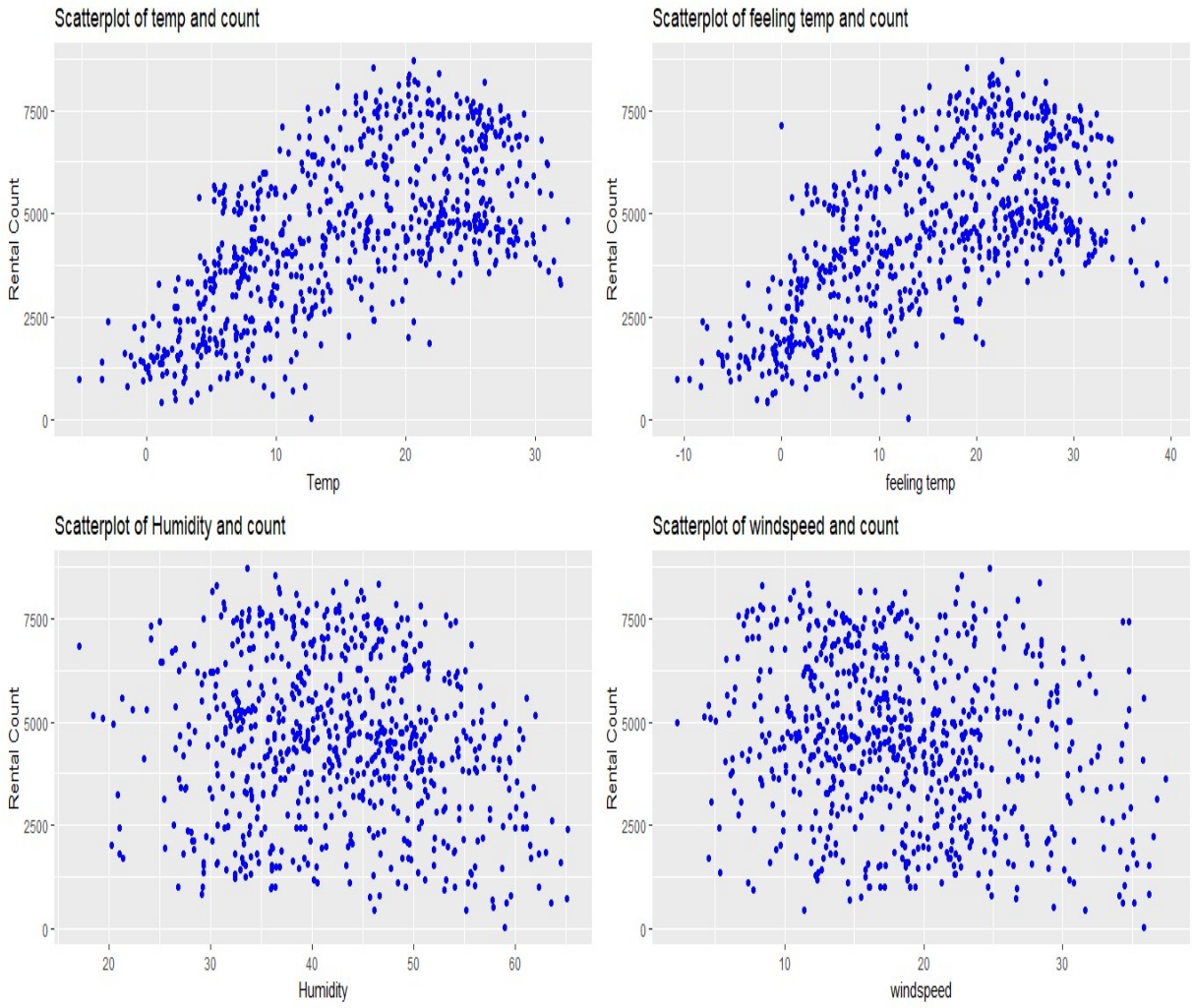
Box Plots Before Outlier Removal :



Box Plots after Outlier Removal:



Scatterplots of Continuous Variables with Count:



Chapter 6

Python Code

#!/usr/bin/env python

 # coding: utf-8

import os

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

from scipy.stats import chi2_contingency **#Importing Libraries for Preprocessing**

os.chdir("C:\\Users\\I B BHATT\\Desktop\\machine learning\\edwisor-projects") **#change working directory**

os.getcwd()

#Read the data.

bike_rental = pd.read_csv("day.csv")

print (bike_rental.head())

print (bike_rental.shape)

#Datatypes of Variables

bike_rental.dtypes

#Changing numerical datatype of category variables to Categorical data type

for i in range(2,9):

  bike_rental.iloc[:,i] = bike_rental.iloc[:,i].astype('category')

bike_rental.dtypes

 # Denormalizing the data to analyze it better

bike_rental.set_index('instant',inplace = True)

bike_rental['Actualtemp'] = bike_rental['temp']*47-16

```
bike_rental['Actualatemp'] = bike_rental['atemp']*66-16

bike_rental['Humidity'] = bike_rental['hum']*67

bike_rental['Actualwindspeed'] = bike_rental['windspeed']*100

bike_rental['Season_abbr']=bike_rental['season'].replace([1,2,3,4],['Spring','Summer','Fall','Win
ter'])

bike_rental['Year'] = bike_rental['yr'].replace([0,1],[2011,2012])

bike_rental['Weather Type'] = bike_rental['weathersit'].replace([1,2,3,4],['Clear','Mist','Light
Snow','Heavy Rain'])

bike_rental.head()
```

#changing new categorical variables data type

```
for i in range(-3,0):
    bike_rental.iloc[:,i] = bike_rental.iloc[:,i].astype('category')
```

#Missing Value Analysis

```
bike_rental.isnull().sum()
```

#Plotting Histogram for univariate analysis by using seaborn library

```
sns.set_style("whitegrid")

plt.figure(figsize = (10,10))

plt.subplot(2,2,1)

sns.distplot(bike_rental['Actualtemp'],kde= False)

plt.subplot(2,2,2)

sns.distplot(bike_rental['Actualatemp'],kde= False)

plt.subplot(2,2,3)

sns.distplot(bike_rental['Humidity'],kde= False)

plt.subplot(2,2,4)

sns.distplot(bike_rental['Actualwindspeed'],kde= False)
```

#Plotting bar graph of categorical variable with matplotlib (just an example)

```
season = bike_rental.groupby('Season_abbr').cnt.sum()
```

```
#year = bike_rental.groupby('Year').cnt.sum()

#Weather = bike_rental.groupby('Weather Type').cnt.sum()

bar = plt.figure()

plot1 = bar.add_subplot(1,1,1)

plot1.set_xlabel('Season_abbr')

plot1.set_ylabel('Sum of count')

plot1.set_title("Season_abbr impact")

season.plot(kind='bar',figsize = (10,10))
```

#Plotting bar graphs using seaborn library

```
sns.set_style("whitegrid")

plt.figure(figsize = (20,20))

plt.subplot(3,2,1)

sns.barplot("Season_abbr","cnt",estimator =sum, data=bike_rental) #estimator is mean by default

plt.subplot(3,2,2)

sns.barplot("Weather Type","cnt",estimator =sum, data=bike_rental)

plt.subplot(3,2,3)

sns.barplot("Year","cnt",estimator =sum, data=bike_rental)

plt.subplot(3,2,4)

sns.barplot("holiday","cnt",estimator =sum, data=bike_rental)

plt.subplot(3,2,5)

sns.barplot("mnth","cnt",estimator =sum, data=bike_rental)

plt.subplot(3,2,6)

sns.barplot("weekday","cnt",estimator =sum, data=bike_rental)
```

#Outlier Analysis using Box plots

```
plt.figure(figsize = (10,8))
```

```
sns.boxplot(data=bike_rental.iloc[:,-7:-3])
```

#Outlier Removal from Humidity

```
q_25,q_75 = np.percentile(bike_rental['Humidity'],[25,75])
iqr = q_75 -q_25
max_out = q_75 + (iqr*1.5)
min_out = q_25 - (iqr*1.5)
bike_rental = bike_rental.drop(bike_rental[bike_rental.Humidity>max_out].index)
bike_rental = bike_rental.drop(bike_rental[bike_rental.Humidity<min_out].index)
```

#Outlier Removal from windspeed

```
q_25,q_75 = np.percentile(bike_rental['Actualwindspeed'],[25,75])
iqr = q_75 -q_25
max_out = q_75+ (iqr*1.5)
min_out = q_25 - (iqr*1.5)
bike_rental = bike_rental.drop(bike_rental[bike_rental.Actualwindspeed>max_out].index)
bike_rental = bike_rental.drop(bike_rental[bike_rental.Actualwindspeed<min_out].index)
```

#boxplot after outlier removal

```
plt.figure(figsize = (10,8))
sns.boxplot(data=bike_rental.iloc[:,-7:-3])
```

#Scatter plot of continuous variable with count using sns,scatterplot

```
plt.figure(figsize = (8.8,8.8))
plt.subplot(2,2,1)
sns.scatterplot(bike_rental['Actualtemp'],bike_rental['cnt'])
plt.subplot(2,2,2)
sns.scatterplot(bike_rental['Actualatemp'],bike_rental['cnt'])
plt.subplot(2,2,3)
```

```
sns.scatterplot(bike_rental['Humidity'],bike_rental['cnt'])

plt.subplot(2,2,4)

sns.scatterplot(bike_rental['Actualwindspeed'],bike_rental['cnt'])
```

#Feature Selection Correlation Analysis

```
bike_rental.corr()
```

#scatter plot after finding higher correlation

```
sns.scatterplot(bike_rental['Actualtemp'],bike_rental['Actualatemp'])
```

#chi square test- just an example. got holiday highly dependent here

```
cat_names = ["season", "yr","holiday", "mnth", "weekday", "workingday", "weathersit"]

for i in cat_names:
   print (i)
   for j in cat_names:
      chi2,p,dof,ex = chi2_contingency(pd.crosstab(bike_rental[j],bike_rental[i]))
      print (j +" " + str(p))
```

#dropping of variables after feature selection

```
bike_rental=bike_rental.drop(['atemp','dteday','casual','registered','Actualatemp','Actualtemp','
Humidity','Actualwindspeed','holiday','Season_abbr','Year','Weather Type'],axis= 1)
```

 # ## Modelling

#Linear Regression

```
import statsmodels.api as sm

from sklearn.model_selection import train_test_split

from sklearn.metrics import r2_score

bike_rental_v2 = bike_rental[['cnt','hum','windspeed','temp']]
```

#creating dummy variables for categories for better analysis

```
cat_names = ["season", "yr", "mnth", "weekday", "workingday", "weathersit"]

for i in cat_names:
```

```
    temp = pd.get_dummies(bike_rental[i], prefix = i)
    bike_rental_v2 = bike_rental_v2.join(temp)
#splitting into train test
train,test = train_test_split(bike_rental_v2,test_size = 0.2, random_state = 0 )
#model
lm_model = sm.OLS(train.iloc[:,0].astype('float'),train.iloc[:,1:35].astype('float')).fit()
lm_model.summary()
#creating mape error function
def MAPE(y_true,y_pred):
    mape = np.mean((np.abs(y_true-y_pred)/y_true)*100)
    return mape
#Predictions
predict_lm = lm_model.predict(test.iloc[:,1:35])
MAPE(test.iloc[:,0],predict_lm)
#16.26 - MAPE
#83.3- adjusted R Squared
#Decision Tree Classifier
from sklearn.tree import DecisionTreeRegressor
model_dtr = DecisionTreeRegressor(max_depth =5,random_state =
0).fit(train.iloc[:,1:35],train.iloc[:,0])
#Predictions
predict_dtr = model_dtr.predict(test.iloc[:,1:35])
#R square function
r2_score(test.iloc[:,0],predict_dtr)
MAPE(test.iloc[:,0],predict_dtr)
```

#18.05 Mape

#0.8768 R squared

#random forest regressor

```
from sklearn.ensemble import RandomForestRegressor

model_rfr = RandomForestRegressor(n_estimators = 100,random_state =
0).fit(train.iloc[:,1:35],train.iloc[:,0])
```

#Predictions

```
predict_rfr = model_rfr.predict(test.iloc[:,1:35])

r2_score(test.iloc[:,0],predict_rfr)

MAPE(test.iloc[:,0],predict_rfr)
```

#15.27 Mape #0.92223 Rsquared