**Prediction of Bike Rental Count**

**Arnav Chowdhury**

**20-07-2019**

Contents

[Chapter 1: Introduction 3](#_Toc14517134)

[Problem Statement 3](#_Toc14517135)

[Hypothesis 3](#_Toc14517136)

[Data 3](#_Toc14517137)

[Chapter 2: Methodology 5](#_Toc14517138)

[Pre-Processing 5](#_Toc14517139)

[Distribution of continuous variables 5](#_Toc14517140)

[Distribution of categorical variables 6](#_Toc14517141)

[Relationship of Continuous variables against bike count 7](#_Toc14517142)

[Detection of outliers 8](#_Toc14517143)

[Feature Selection 11](#_Toc14517144)

[Chapter 3: Modelling 12](#_Toc14517145)

[Model Selection 12](#_Toc14517146)

[Multiple Linear Regression 12](#_Toc14517147)

[Decision Tree 13](#_Toc14517148)

[Random Forest 13](#_Toc14517149)

[Chapter 4: Conclusion 14](#_Toc14517150)

[Mean Absolute Percentage Error (MAPE) 14](#_Toc14517151)

[Chapter 5: Appendix 15](#_Toc14517152)

[Figures 15](#_Toc14517153)

[Chapter 6: Python code 20](#_Toc14517154)

[Linear Regression 21](#_Toc14517155)

[Decision tree 22](#_Toc14517156)

[Random Forest 23](#_Toc14517157)

# Chapter 1: Introduction

## Problem Statement

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

## Hypothesis

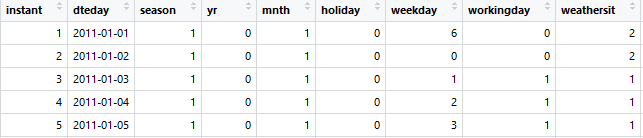
A list of Hypothesis which may affect bike ride count(cnt) are:

1. Season: Type of season where bike rides are mostly taken.
2. Weather: Bike rides may be high during a particular season.
3. Weekday/weekends: Bike rides are normally more on Weekdays than Weekends.
4. Temperature: Type of temperature where bikes rides are more common.
5. Humidity: Environmental conditions affecting bike rides.
6. Windspeed: Environmental conditions affecting bike rides.

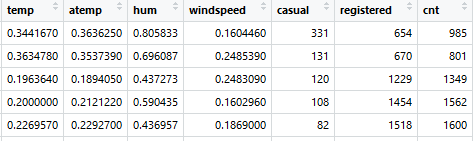
## Data

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)**



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



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

|  |  |
| --- | --- |
| **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

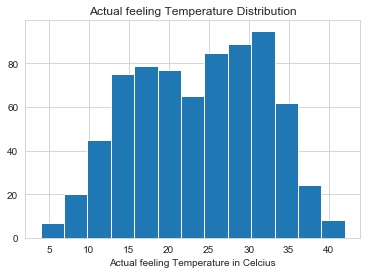
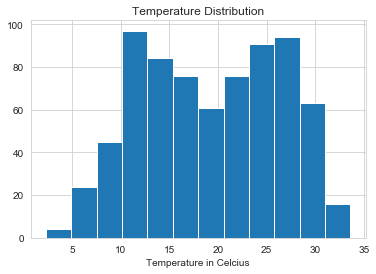
## Pre-Processing

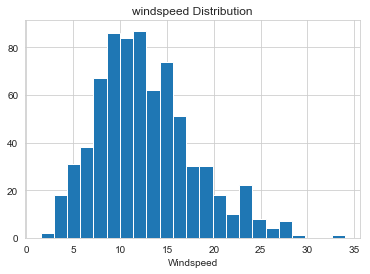
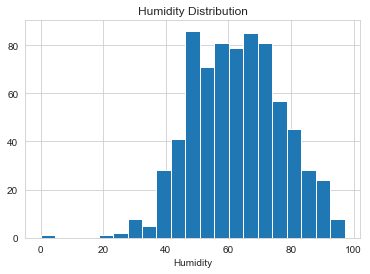
A predictive model requires that we look at the data before we start to create a model. However, in data mining, looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is known as Exploratory Data Analysis. To start this process we will first look at the distribution of variables. Most analysis like regression require the data to be normally distributed. We can visualize this by looking at the distribution of different predictor variables.

## 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.

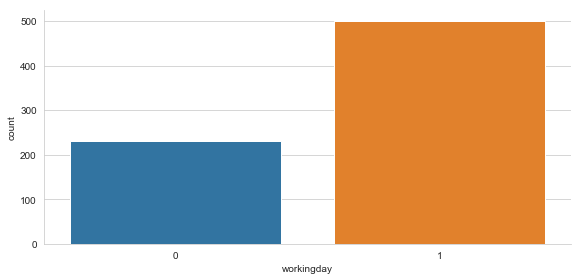
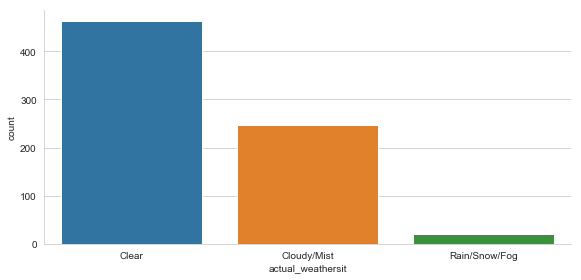
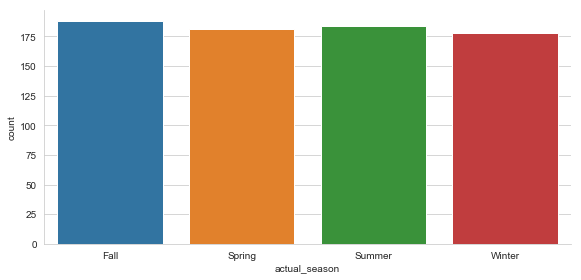




**Fig 2.1: Distribution of continuous variables using Histograms**

## Distribution of categorical variables

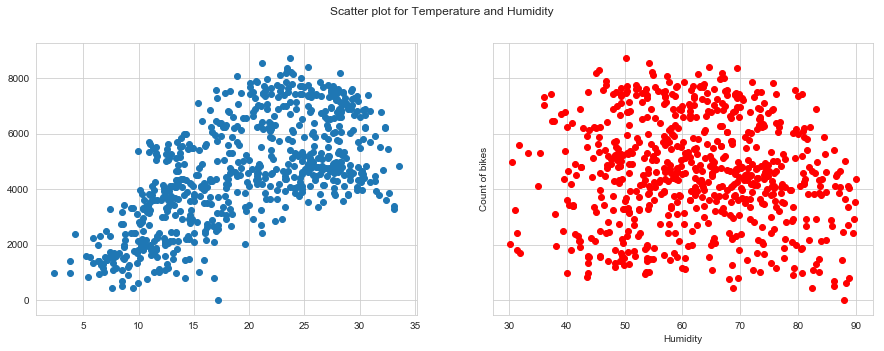
The distribution of categorical variables is as shown in the below figure:

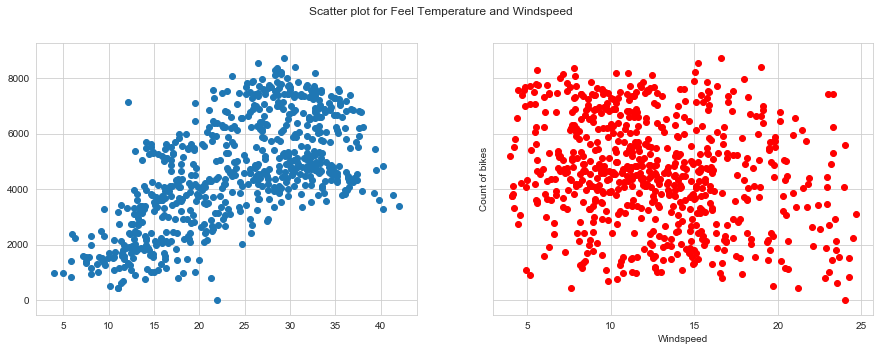


**Fig 2.2: Distribution of categorical variables using bar plots**

## 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.

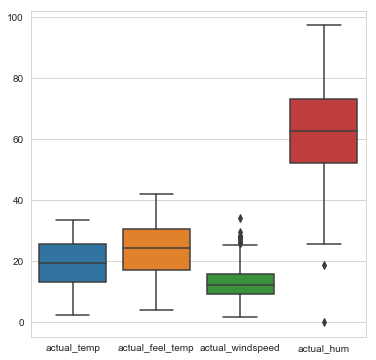




**Fig 2.3: Scatter plot for continuous variables**

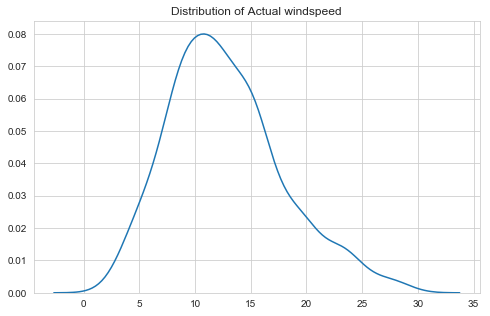
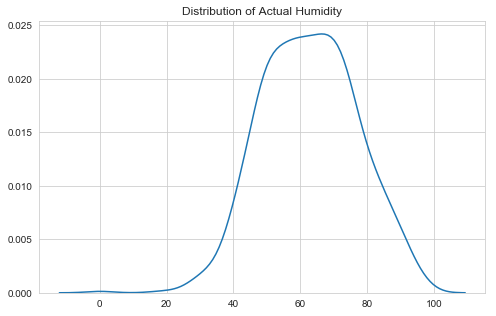
## Detection of outliers

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



**Fig 2.4: Boxplot of continuous variables**

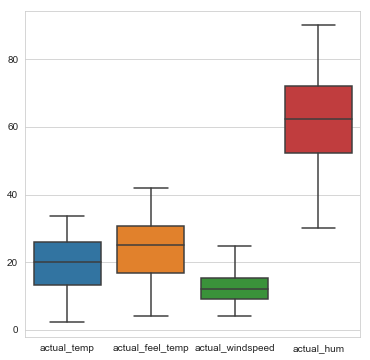
Outliers can be removed using the Boxplot stats method, wherein the Inter Quartile Range (IQR) is calculated and the minimum and maximum value are calculated for the variables. Any value ranging outside the minimum and maximum value are discarded. However, on a more detailed analysis of Windspeed and Humidity using a depth plot we were able to find the actual values which were affecting the model prediction.



We were able to find 5 rows below 30 and 19 values above 90 for Humidity. We considered them as outliers and removed them from our data. Also for Windspeed

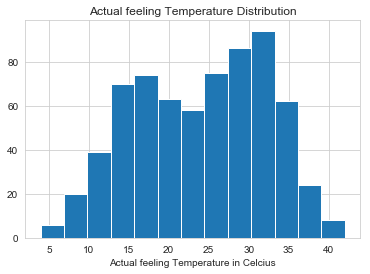
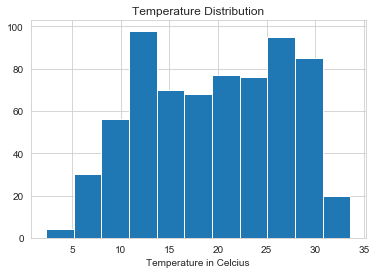
we found 12 values less than 4 and 16 values more than 25. Since collectively it forms less than 10% of the dataset we removed them for better model performance.

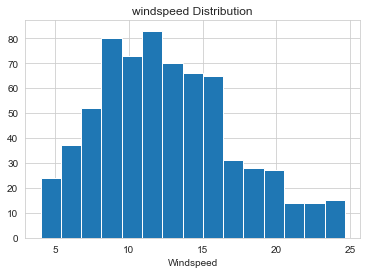
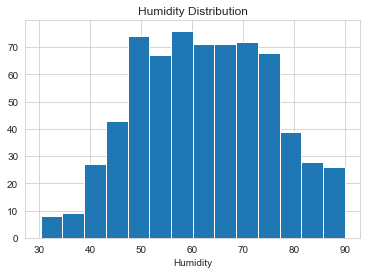
The boxplot of the continuous variables after removing the outliers is shown in the below figure:



**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 the data is not skewed as much as before the removal of outliers. The figure shown below illustrates the distribution of continuous variables using histograms.



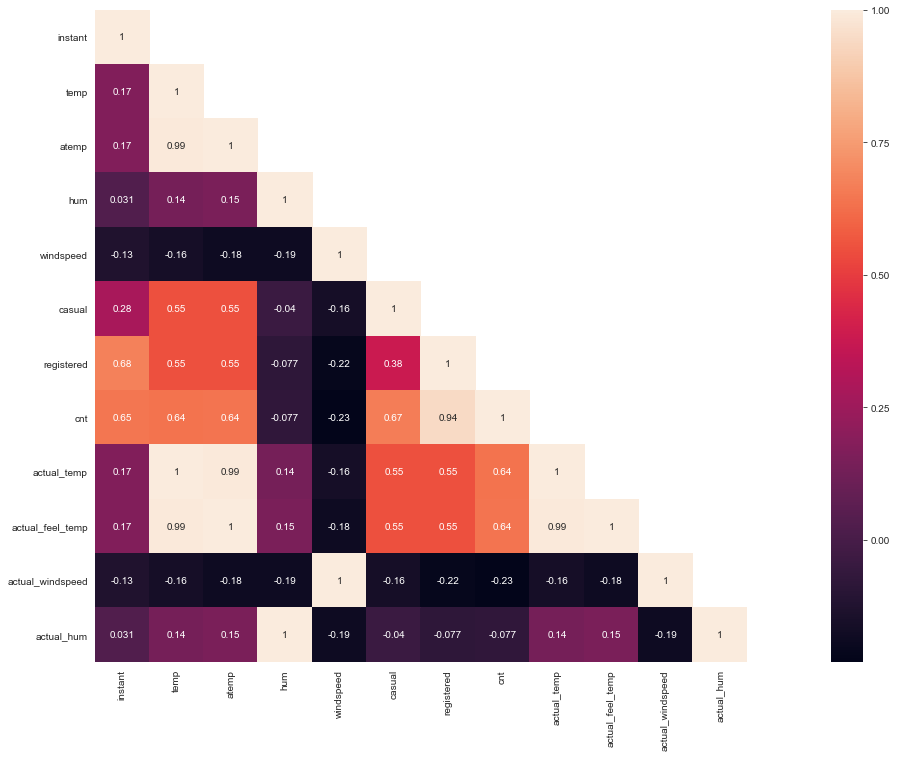


**Fig 2.6: Distribution of numerical data using histograms after removal of outliers**

## 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

## Model Selection

The dependent variable in our model is a continuous variable i.e., Count of bike rentals whereas the independent variables are a mix of continuous and categorical values. 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).

## 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.

Based on the strength of the relationships between some of the features and the target variable, we can expect the model to perform relatively well.

As you can see the Adjusted R-squared value, we can explain 87.41% of the data using our multiple linear regression model. By looking at the F-statistic and combined p-value we can

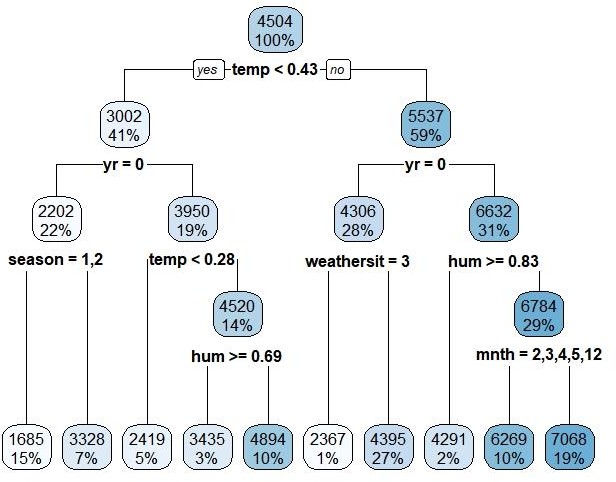
reject the null hypothesis that target variable does not depend on any of the predictor variables. This model explains the data very well and is considered to be good.

Even after removing the non-significant variables, the accuracy, Adjusted R-squared and F- statistic do not change by much, hence the accuracy of this model is chosen to be final.

MAPE of this multiple linear regression model is 11.48%. Hence the accuracy of this model is 88.52%. This model performs very well for this test data.

## 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 13.26%. Hence the accuracy for this model is 86.74%.

## 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 9.14%. Hence the accuracy is 90.86%.

# 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.

## Mean Absolute Percentage Error (MAPE)

MAPE 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 = np.mean(np.abs((y\_actual - y\_pred)/y\_actual))\*100**

Linear Regression Model: **MAPE = 11.48**

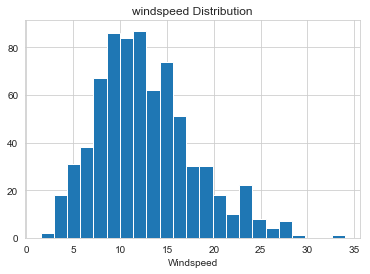
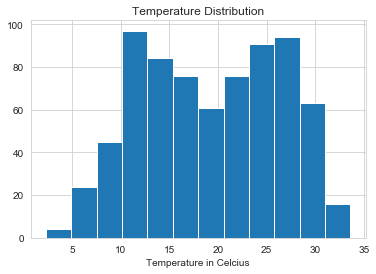
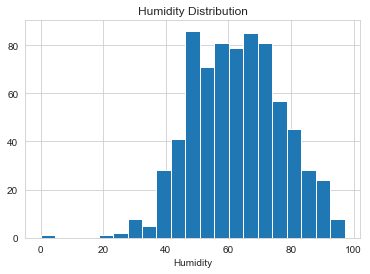
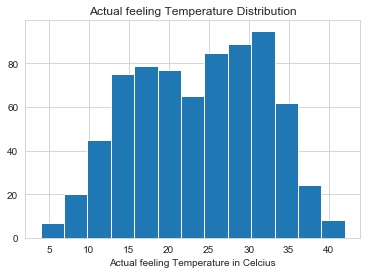
Decision Tree: **MAPE = 13.26**

Random Forest: **MAPE = 9.14**

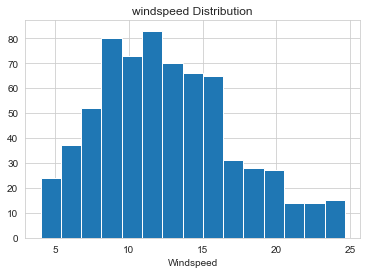
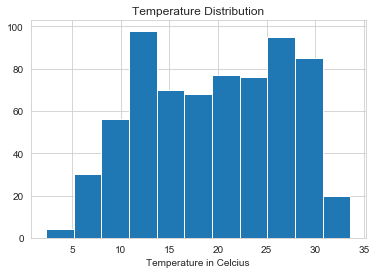
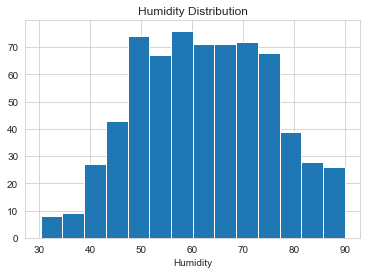
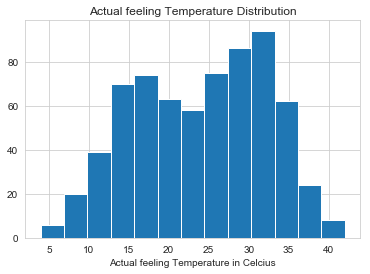
Based on the above error metrics, Random Forest is the better model for our analysis. Hence Random Forest is chosen as the model for prediction of bike rental count.

# Chapter 5: Appendix

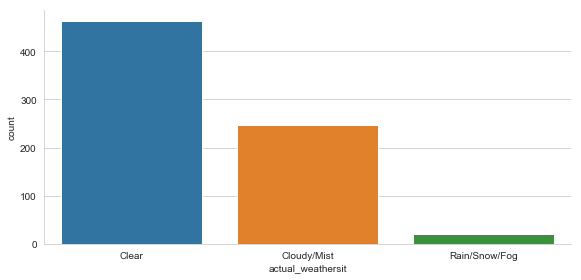
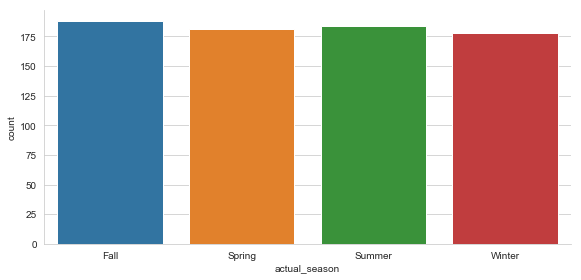
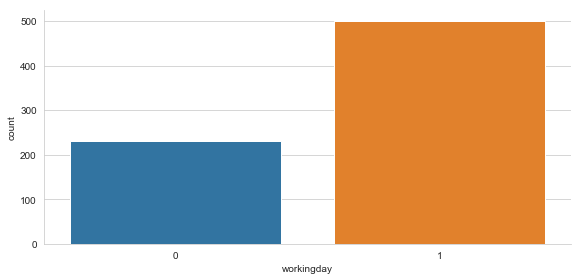
## Figures



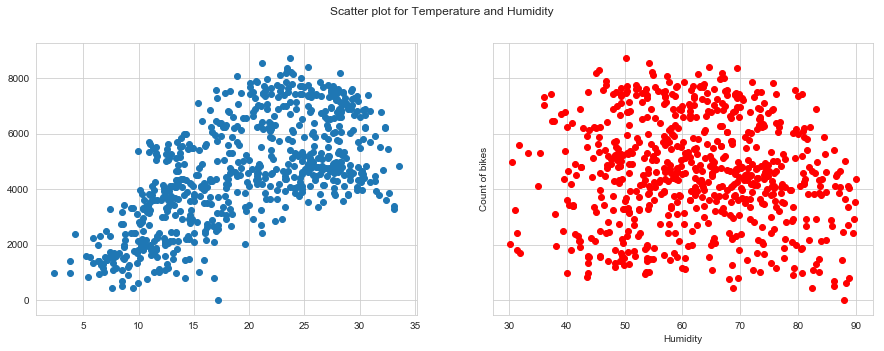
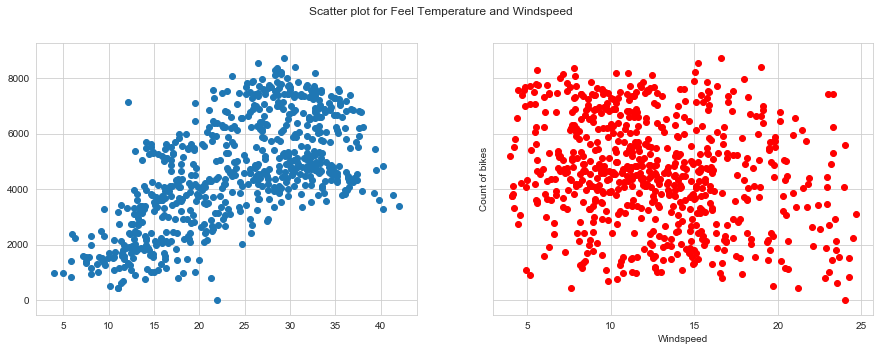
**Fig 5.1: Distribution of continuous variables using Histograms**



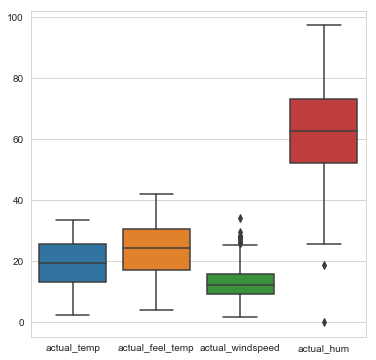
**Fig 5.2: Distribution of numerical data using histograms after removal of outliers**



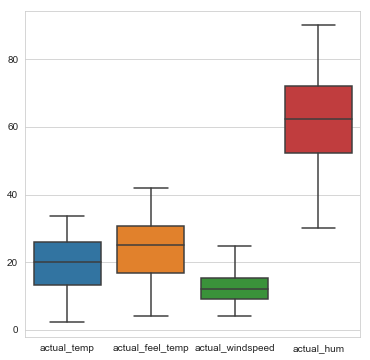
**Fig 5.3: Distribution of categorical variables using bar plots**



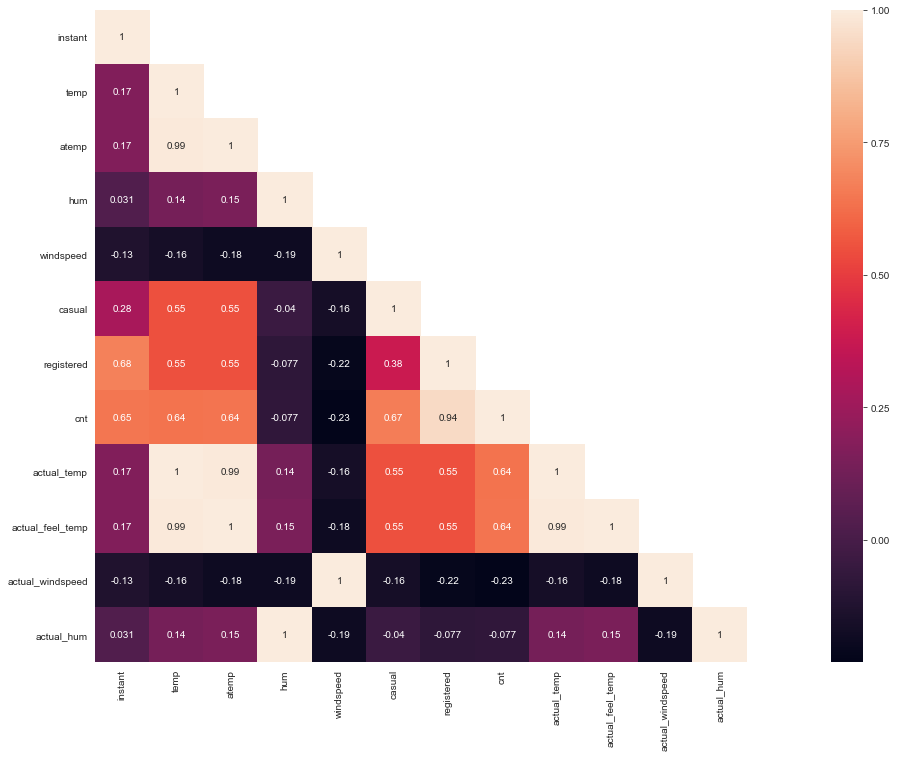
**Fig 5.4: Scatter plot for continuous variables**



**Fig 5.5: Boxplot of continuous variables**



**Fig 5.6: Boxplot of continuous variables after removal of outliers**



**Fig 5.7: Correlation plot of all the variables**

# Chapter 6: Python code

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Exploratory Data Analysis\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Check the bar graph of categorical Data using factorplot

sns.set\_style("whitegrid")

sns.factorplot(data=df, x='actual\_season', kind= 'count',size=4,aspect=2)

sns.set\_style("whitegrid")

sns.factorplot(data=df, x='actual\_weathersit', kind= 'count',size=4,aspect=2)

sns.set\_style("whitegrid")

sns.factorplot(data=df, x='workingday', kind= 'count',size=4,aspect=2)

#Check the distribution of numerical data using histogram

plt.hist(data=df, x='actual\_temp', bins='auto', label='Temperature')

plt.xlabel('Temperature in Celcius')

plt.title("Temperature Distribution")

plt.hist(data=df, x='actual\_feel\_temp', bins='auto', label='Temperature')

plt.xlabel('Actual feeling Temperature in Celcius')

plt.title("Actual feeling Temperature Distribution")

plt.hist(data=df, x='actual\_hum', bins='auto', label='Temperature')

plt.xlabel('Humidity')

plt.title("Humidity Distribution")

plt.hist(data=df, x='actual\_windspeed', bins='auto', label='Windspeed')

plt.xlabel('Windspeed')

plt.title("windspeed Distribution")

#Check for outliers in data using boxplot

sns.boxplot(data=df[['actual\_temp','actual\_feel\_temp','actual\_windspeed','actual\_hum']])

fig=plt.gcf()

fig.set\_size\_inches(6,6)

plt.figure(figsize=(8,5))

sns.kdeplot(df['actual\_hum'].values).set\_title("Distribution of Actual Humidity")

df.loc[df['actual\_hum']<30].shape

df.loc[df['actual\_hum']>90].shape

df=df.loc[df['actual\_hum'].between(30,90)]

plt.figure(figsize=(8,5))

sns.kdeplot(df['actual\_windspeed'].values).set\_title("Distribution of Actual windspeed")

df.loc[df['actual\_windspeed']<4].shape

df.loc[df['actual\_windspeed']>25].shape

df=df.loc[df['actual\_windspeed'].between(4,25)]

#Check for outliers in data using boxplot after removing outliers

sns.boxplot(data=df[['actual\_temp','actual\_feel\_temp','actual\_windspeed','actual\_hum']])

fig=plt.gcf()

fig.set\_size\_inches(6,6)

#Check for collinearity using corelation matrix.

cor\_mat= df[:].corr()

mask = np.array(cor\_mat)

mask[np.tril\_indices\_from(mask)] = False

fig=plt.gcf()

fig.set\_size\_inches(30,12)

sns.heatmap(data=cor\_mat,mask=mask,square=True,annot=True,cbar=True)

#Check the distribution of Temperature and Humdity against Bike rental count using scatter plot

fig, axs = plt.subplots(1,2, figsize=(15, 5), sharey=True)

axs[0].scatter(data=df, x='actual\_temp', y='cnt')

axs[1].scatter(data=df, x='actual\_hum', y='cnt', color = 'red')

fig.suptitle('Scatter plot for Temperature and Humidity')

plt.xlabel("Humidity")

plt.ylabel("Count of bikes")

#Check the distribution of Feel Temperature and Windspeed against Bike rental count using scatter plot

fig, axs = plt.subplots(1,2, figsize=(15, 5), sharey=True)

axs[0].scatter(data=df, x='actual\_feel\_temp', y='cnt')

axs[1].scatter(data=df, x='actual\_windspeed', y='cnt', color = 'red')

fig.suptitle('Scatter plot for Feel Temperature and Windspeed')

plt.xlabel("Windspeed")

plt.ylabel("Count of bikes")

df = df.drop(columns=['holiday','instant','dteday','atemp','casual','registered','actual\_temp','actual\_feel\_temp',

'actual\_windspeed','actual\_hum','actual\_season','actual\_yr','actual\_holiday','actual\_weathersit'])

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Modelling\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### Linear Regression

#import libraries for Linear regression

import statsmodels.api as sm

from sklearn.metrics import mean\_squared\_error

#Divide data into train and test

from sklearn.model\_selection import train\_test\_split

train,test = train\_test\_split(df, test\_size = 0.2, random\_state = 123)

#Train the model

lr\_model = sm.OLS(train.iloc[:,9].astype(float), train.iloc[:,0:9].astype(float)).fit()

#Check the summary of model

lr\_model.summary()

#Predict the results of test data

lr\_predictions = lr\_model.predict(test.iloc[:,0:9])

##Create a dataframe for actual values and predicted values

df\_lr = pd.DataFrame({'actual': test.iloc[:,9], 'pred': lr\_predictions})

df\_lr.head()

#Function for Mean Absolute Percentage Error

def MAPE(y\_actual,y\_pred):

mape = np.mean(np.abs((y\_actual - y\_pred)/y\_actual))\*100

return mape

#Calclulate MAPE

MAPE(test.iloc[:,9],lr\_predictions)

#Create continuous data. Save target variable first

train\_lr = train[['cnt','temp','hum','windspeed']]

test\_lr = test[['cnt','temp','hum','windspeed']]

#Create dummies for categorical variables

cat\_names = ["season", "yr", "mnth", "weekday", "workingday", "weathersit"]

for i in cat\_names:

temp1 = pd.get\_dummies(train[i], prefix = i)

temp2 = pd.get\_dummies(test[i], prefix = i)

train\_lr = train\_lr.join(temp1)

test\_lr = test\_lr.join(temp2)

#Train the model

lr\_model = sm.OLS(train\_lr.iloc[:,0].astype(float), train\_lr.iloc[:,1:34].astype(float)).fit()

#summary of model

lr\_model.summary()

#Predict the results of test data

lr\_predictions = lr\_model.predict(test\_lr.iloc[:,1:34])

#Create a dataframe for actual values and predicted values

df\_lr = pd.DataFrame({'actual': test\_lr.iloc[:,0], 'pred': lr\_predictions})

df\_lr.head()

#Calclulate MAPE

MAPE(test\_lr.iloc[:,0],lr\_predictions)

### Decision tree

#Import Libraries for decision tree

from sklearn.tree import DecisionTreeRegressor

#Train the model

dt\_model = DecisionTreeRegressor(random\_state=123).fit(train.iloc[:,0:9], train.iloc[:,9])

#Predict the results of test data

dt\_predictions = dt\_model.predict(test.iloc[:,0:9])

df\_dt = pd.DataFrame({'actual': test.iloc[:,9], 'pred': dt\_predictions})

df\_dt.head()

#Calculate MAPE for decision tree

MAPE(test.iloc[:,9],dt\_predictions)

### Random Forest

#Import library for RandomForestRegressor

from sklearn.ensemble import RandomForestRegressor

#Train the model

rf\_model = RandomForestRegressor(n\_estimators=500,random\_state=123).fit(train.iloc[:,0:9], train.iloc[:,9])

#Predict the results of test data

rf\_predictions = rf\_model.predict(test.iloc[:,0:9])

#Create a dataframe for actual values and predicted values

df\_rf = pd.DataFrame({'actual': test.iloc[:,9], 'pred': rf\_predictions})

df\_rf.head()

#Calculate MAPE

MAPE(test.iloc[:,9],rf\_predictions)

#MAPE: 9.14%

#Accuracy:90.86%