|  |  |
| --- | --- |
| https://upload.wikimedia.org/wikipedia/commons/b/b2/Moscow_09-13_img19_rental_bikes.jpg  Bike rental prediction  Project report | Abstract  Bike rental prediction->The objective of this Case is to Predication of bike rental count on daily based on the environmental and seasonal settings.  Gagan Kohli  Data science |

**Contents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **Introduction** | | | 2 |
|  | 1.1 | Problem Statement . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 2 |
|  | 1.2 | Data. . . | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 2 |
| **2** | **Methodology** | | | **5** |
|  | 2.1 | **Data Analysis** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 5 |
|  |  | 2.1.1 | Univariate Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . | 6 |
|  |  | 2.1.2 | Bivariate Selection . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . | 10 |
|  | 2.2 | **Pre-processing Techniques** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 12 |
|  |  | 2.2.1 | Missing value analysis. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . | 12 |
|  |  | 2.2.2 | Outlier analysis. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 13 |
|  |  | 2.2.3 | Feature Selection. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . | 15 |
|  |  | 2.2.4 | Feature Scaling. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 16 |
| **3** | **Modelling** | |  | **19** |
|  |  | |  |  |
|  | 3.1 | Model Selection . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 19 |
|  | 3.2 | Decision Tree . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 20 |
|  | 3.3 | Random Forest . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 22 |
|  | 3.4 | Linear Regression . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 23 |
|  |  |  | |  |
| **117Appendix A - Extra Figures** | | | | **27** |
| **Appendix B - code** | | | | **29** |
|  |  | [Univariate Analysis (Fig:](#page24)  [2.2, 2.3 & 2.4)](#page6) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . | | 29 |
|  |  | [Bivariate Analysis (Fig:](#page24)  [2.5 &2.6)](#page7)  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . | | 30 |
|  |  | [Outlier Analysis (Fig:](#page24)  [2.7, 2.8, 2.9& 2.10)](#page19)  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . | | 32 |
|  |  | [Feature Selection (Fig:](#page24)  [2.11&2.12)](#page8) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . | | 33 |
|  |  | Decision Tree (Fig: 3.2.2&3.2.4) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . .  Random Forest (Fig:3.3.3) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Linear Regression (Fig:3.4.4) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 33  33  33 |
|  |  | Complete Python File . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . | | 34 |
|  |  | Complete R File . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 40 |
|  |  |  | |  |
| [**Reference**](#page30)**s** | | |  |  |

**Chapter 1**

**Introduction**

1. **Problem Statement**

The Bike Rental Data contains the daily count of rental bikes between the year 2011 and 2012 with corresponding weather and seasonal information. We would like to predict the daily count of rental count in order to automate the system based on given data.

1. **Data**

Our task is to build Regression model which will give the daily count of rental bikes based on weather, season and other given parameters. Given below is a sample of the data set that we are using to predict the count:

Table 1.1: Bike Rental Sample Data (Columns: 1-8)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| instant | dteday | season | yr | mnth | holiday | weekday |
| 1 | 1/1/2011 | 1 | 0 | 1 | 0 | 6 |
| 2 | 1/2/2011 | 1 | 0 | 1 | 0 | 0 |
| 3 | 1/3/2011 | 1 | 0 | 1 | 0 | 1 |
| 4 | 1/4/2011 | 1 | 0 | 1 | 0 | 2 |
| 5 | 1/5/2011 | 1 | 0 | 1 | 0 | 3 |

Table 1.2: Bike Rental Sample Data (Columns: 9-14)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| weathersit | temp | atemp | hum | windspeed | casual | registered | cnt |
| 2 | 0.344167 | 0.363625 | 0.805833 | 0.160446 | 331 | 654 | 985 |
| 2 | 0.363478 | 0.353739 | 0.696087 | 0.248539 | 131 | 670 | 801 |
| 1 | 0.196364 | 0.189405 | 0.437273 | 0.248309 | 120 | 1229 | 1349 |
| 1 | 0.2 | 0.212122 | 0.590435 | 0.160296 | 108 | 1454 | 1562 |
| 1 | 0.226957 | 0.22927 | 0.436957 | 0.1869 | 82 | 1518 | 1600 |

Below are the variables are used to predict the count of bike rentals

Table 1.3: Bike Rental Predictors

|  |  |
| --- | --- |
| s.no | Variables |
| 1 | dteday |
| 2 | season |
| 3 | yr |
| 4 | mnth |
| 5 | holiday |
| 6 | weekday |
| 7 | workingday |
| 8 | weathersit |
| 9 | temp |
| 10 | atemp |
| 11 | hum |
| 12 | windspeed |
| 13 | casual |
| 14 | registered |

The details of data attributes in the dataset are as follows –

* **instant:** Record index
* **dteday:** Date
* **season:** Season
  + 1:springer
  + 2:summer
  + 3:fall
  + 4:winter
* **yr:** Year
  + 0: 2011
  + 1:2012
* **mnth:** Month (1 to 12)
* **hr:** Hour (0 to 23)
* **holiday:** weather day is holiday or not (extracted from Holiday Schedule)
* **weekday:** Day of the week
* **workingday:** If day is neither weekend nor holiday is 1, otherwise is 0.
* **weathersit:** (extracted from Freemeteo)
  + 1: Clear, Few clouds, Partly cloudy, Partly cloudy
  + 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
  + 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
  + 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
* **temp:** Normalized temperature in Celsius. The values are derived via (t-t\_min)/(t\_max-t\_min), t\_min=-8, t\_max=+39 (only in hourly scale)
* **atemp:** Normalized feeling temperature in Celsius. The values are derived via (t-t\_min)/(t\_maxt\_min), t\_min=-16, t\_max=+50 (only in hourly scale)
* **hum:** Normalized humidity. The values are divided to 100 (max)
* **windspeed:** Normalized wind speed. The values are divided to 67 (max)
* **casual:** count of casual users
* **registered:** count of registered users
* **cnt:** count of total rental bikes including both casual and registered

**Chapter 2**

**Methodology**

1. **Pre Processing**

Any predictive modeling requires that we look at the data before we start modelling. However, in data mining terms *looking at data* refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as **Exploratory Data Analysis**. To start this process we will first try and make initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations look at all the distributions of the Numeric variables. Most analysis like regression, require the data to be normally distributed.

Starting with general analysis the data given to us includes 16 variables and 731 observations. Further

Analysing the data types for each column->

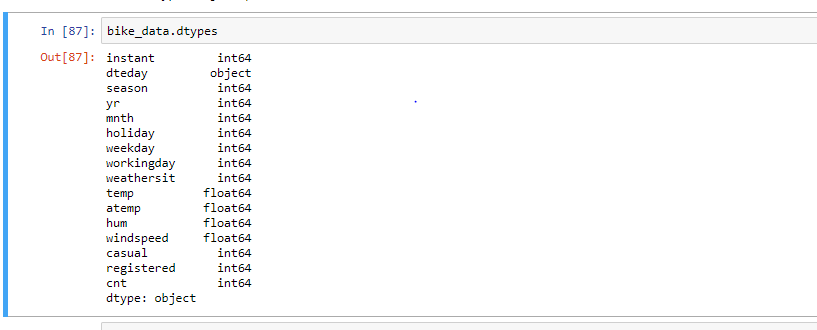


Figure 2.1 data types for variables

Further we also found basic statistical information for our data like median mean and other important information also by observation we can see the variable ”instance” is nothing but acting index for the data which could be easily dropped as its not adding any additional information to the model we will be building.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | instant | season | yr | mnth | holiday | weekday | workingday | weathersit | temp | atemp | hum | windspeed | casual | registered | cnt |
| **count** | 731.000000 | 731.000000 | 731.000000 | 731.000000 | 731.000000 | 731.000000 | 731.000000 | 731.000000 | 731.000000 | 731.000000 | 731.000000 | 731.000000 | 731.000000 | 731.000000 | 731.000000 |
| **mean** | 366.000000 | 2.496580 | 0.500684 | 6.519836 | 0.028728 | 2.997264 | 0.683995 | 1.395349 | 0.495385 | 0.474354 | 0.627894 | 0.190486 | 848.176471 | 3656.172367 | 4504.348837 |
| **std** | 211.165812 | 1.110807 | 0.500342 | 3.451913 | 0.167155 | 2.004787 | 0.465233 | 0.544894 | 0.183051 | 0.162961 | 0.142429 | 0.077498 | 686.622488 | 1560.256377 | 1937.211452 |
| **min** | 1.000000 | 1.000000 | 0.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 | 0.059130 | 0.079070 | 0.000000 | 0.022392 | 2.000000 | 20.000000 | 22.000000 |
| **25%** | 183.500000 | 2.000000 | 0.000000 | 4.000000 | 0.000000 | 1.000000 | 0.000000 | 1.000000 | 0.337083 | 0.337842 | 0.520000 | 0.134950 | 315.500000 | 2497.000000 | 3152.000000 |
| **50%** | 366.000000 | 3.000000 | 1.000000 | 7.000000 | 0.000000 | 3.000000 | 1.000000 | 1.000000 | 0.498333 | 0.486733 | 0.626667 | 0.180975 | 713.000000 | 3662.000000 | 4548.000000 |
| **75%** | 548.500000 | 3.000000 | 1.000000 | 10.000000 | 0.000000 | 5.000000 | 1.000000 | 2.000000 | 0.655417 | 0.608602 | 0.730209 | 0.233214 | 1096.000000 | 4776.500000 | 5956.000000 |
| **max** | 731.000000 | 4.000000 | 1.000000 | 12.000000 | 1.000000 | 6.000000 | 1.000000 | 3.000000 | 0.861667 | 0.840896 | 0.972500 | 0.507463 | 3410.000000 | 6946.000000 | 8714.000000 |

**2.1.1 Univariate Analysis**

In Figure 2.2 and 2.3 we have plotted the probability density functions numeric variables present in the data including target variable cnt..

1. Target variable cnt is normally distributed
2. Independent variables like ‘temp’,’atemp’, and ‘regestered’ data is distributed normally.
3. Independent variable ‘casual’ data is slightly skewed to the right so, there is chances of getting outliers.
4. Other Independent variable ‘hum’ data is slightly skewed to the left, here data is already in normalize form.

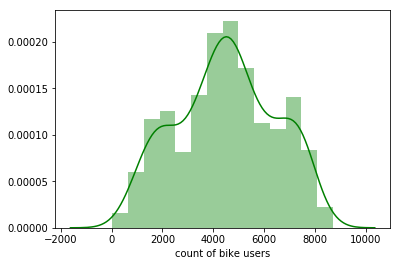
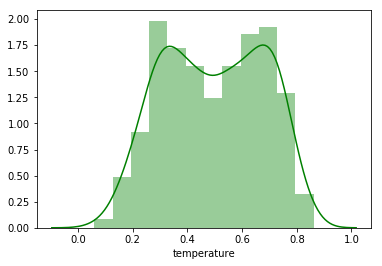
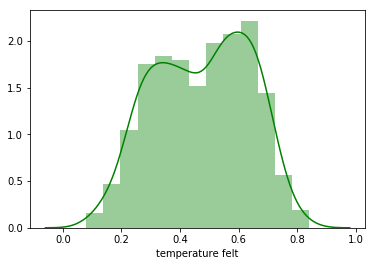
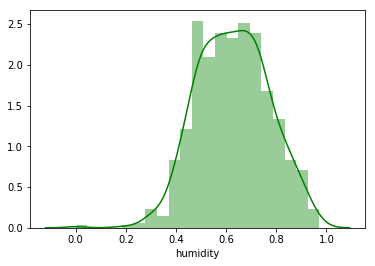
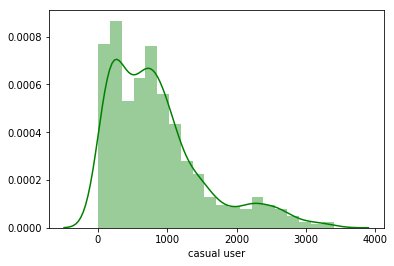
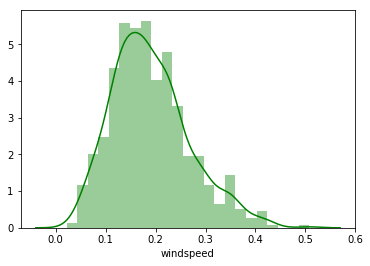


Figure 2.2 Distribution of target variable (CNT) (python code in Appendix B)









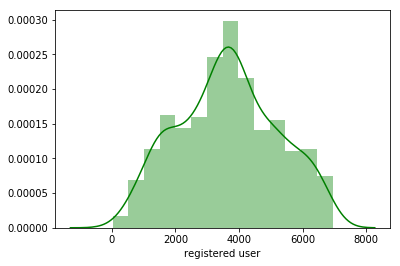


Figure 2.3 showing distribution of dependent variables (python code in Appendix B)

In fig 2.4 shows the bargraphs between all catagorical variables and there frequeancy

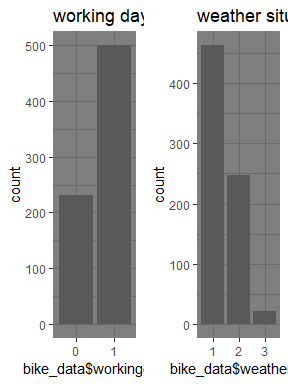
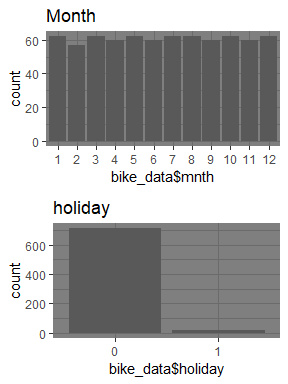
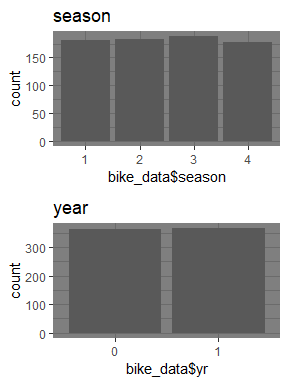
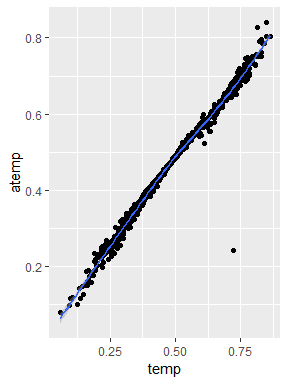
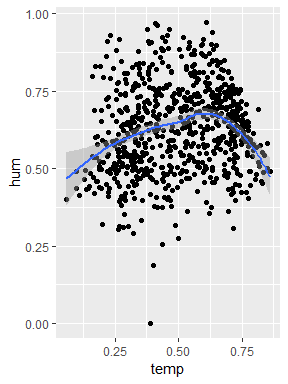


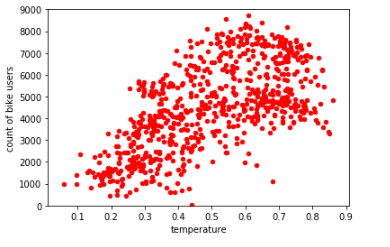
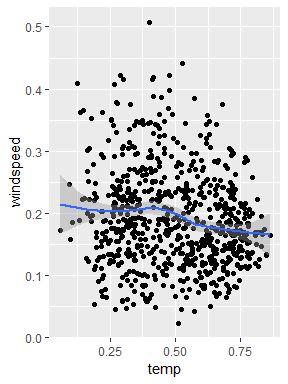
Figure 2.4 showing distribution of dependent catagorical variables (R in Appendix B)

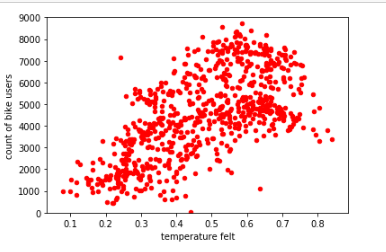
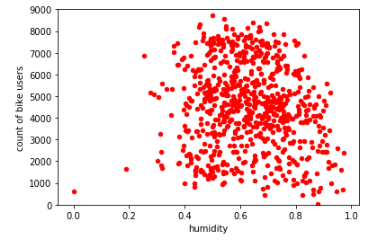
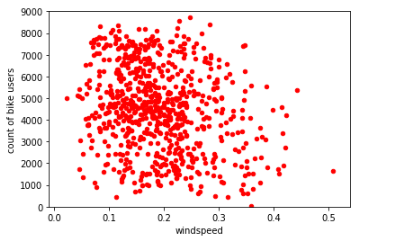
**2.1.2 Bivariate Analysis**

Below figures shows relationship between independent variables and also with numeric target variable using ggpair

1. Below ggpair graph is showing clearly that relationship between independent variables ‘temp’ and ‘atemp’ are very strong.
2. The relationship between ‘hum’ , ‘windspeed’ with target variable ‘cnt’ is less.

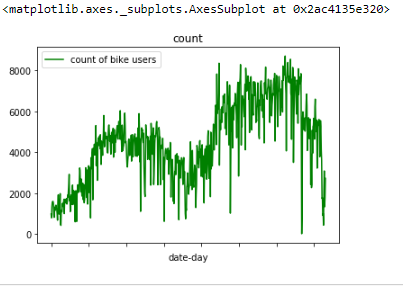
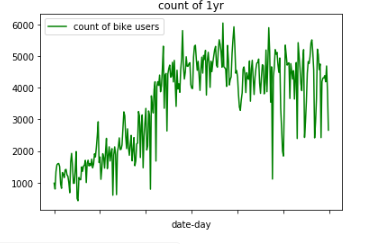
****

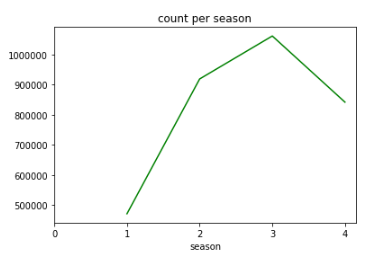
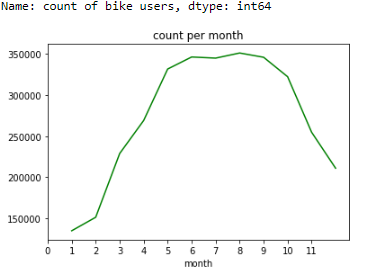
**Fig-2.5 shows relationship between numerical variables with “cnt” and among one selves**

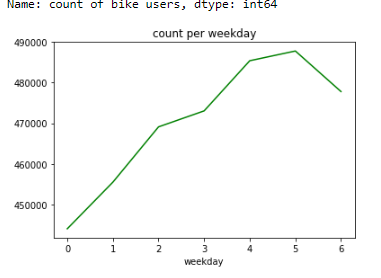


**Fig-2.5 shows relationship between numerical variables with “cnt” and among one selves**

Plotting graphs between our target variable and other variables we find the following patters in the data. The count of bike users varies with other variables.







**fig-2.6 shows relationship between categorical variables and count of bike user.**

**2.2 Pre-processing Techniques**

These include techniques like missing value analysis, outlier treatment, feature engineering, feature selection and feature scaling. These techniques are necessary before model building so the model could be correct and that of a good quality

**2.2.1 Missing Value Analysis**

Missing values in data is a common phenomenon in real world problems. Knowing how to handle missing values effectively is a required step to reduce bias and to produce powerful models.

Below table illustrate no missing value present in the data.

2.1 missing values

|  |  |  |
| --- | --- | --- |
| s.no | Variables | missing values |
| 1 | dteday | 0 |
| 2 | season | 0 |
| 3 | yr | 0 |
| 4 | mnth | 0 |
| 5 | holiday | 0 |
| 6 | weekday | 0 |
| 7 | workingday | 0 |
| 8 | weathersit | 0 |
| 9 | temp | 0 |
| 10 | atemp | 0 |
| 11 | hum | 0 |
| 12 | windspeed | 0 |
| 13 | casual | 0 |
| 14 | registered | 0 |

**2.2.2 Outlier Analysis**

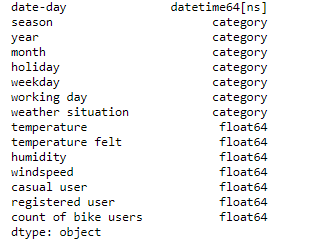
An outlier may also be explained as a piece of data or observation that deviates drastically from the given norm or average of the data set. An outlier may be caused simply by chance, but it may also indicate measurement error or that the given data set has a heavy-tailed distribution.

The Other steps of Pre-processing Technique is Outliers analysis , an outlier is an observation point that is distant from other observations. Outliers in data can distort predictions and affect the accuracy, if you don’t detect and handle them appropriately especially in regression models..

As we are observed in fig 2.2 the data is skewed so, there is chance of outlier in independent variable ‘casual’ , one of the best method to detect outliers is Boxplot

Boxplot :-  boxplot is a method for graphically depicting groups of numerical data through their [quartiles](https://en.wikipedia.org/wiki/Quartile). Box plots may also have lines extending vertically from the boxes (whiskers) indicating variability outside the upper and lower quartiles

Also for performing outlier analysis we first need to convert each variable to proper data format like categorical variables as factors and numeric variables to float. The table below shows the same



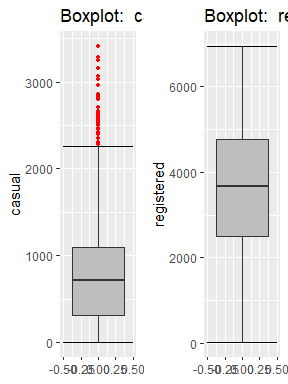
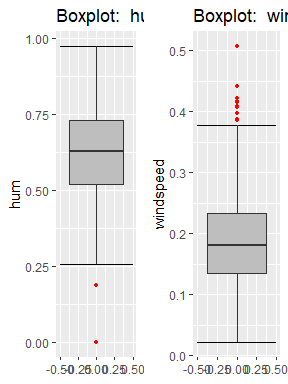
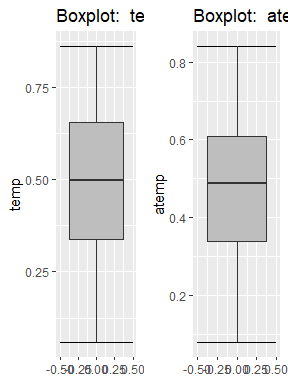


Fig 2.7 shows presence of Outliers in variable before removing Outliers

As can be seen that variables “windspeed “and “casual” consists of outliers which are to be handled before model building. Outliers can be treated in 2 ways one by deleting the outlier rows and other by deleting outlier observations and treating them as missing value.

KNN imputation is used in r while observations were deleted in python. After outlier treatment we can go for feature selection and feature scaling.

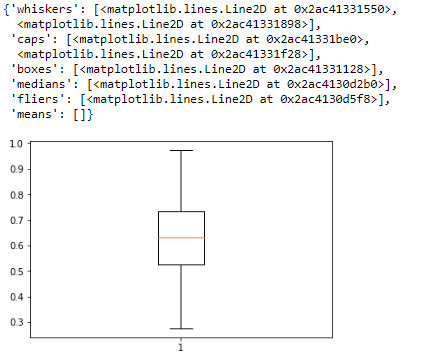
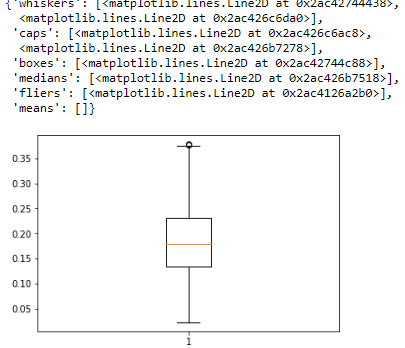


Fig 2.8 shows boxplot of variables after removing outliers after removing outliers

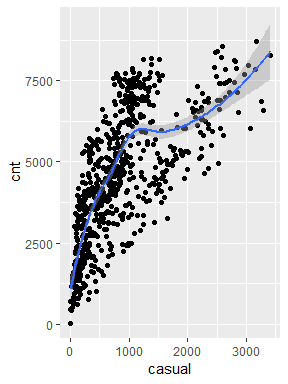


Figure 2.9 ‘casual’ Baxoplot and relation between ‘cnt’ and’ casual’ (R code in Appendix B)

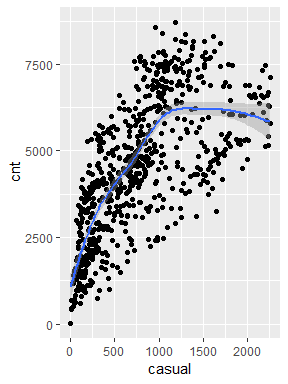


Figure 2.10 ‘casual’ Boxplot and relation between ‘casual’ and ‘cnt’ (R code in A)

**2.2.3 Features Selections**

Machine learning works on a simple rule – if you put garbage in, you will only get garbage to come out. By garbage here, I mean noise in data.

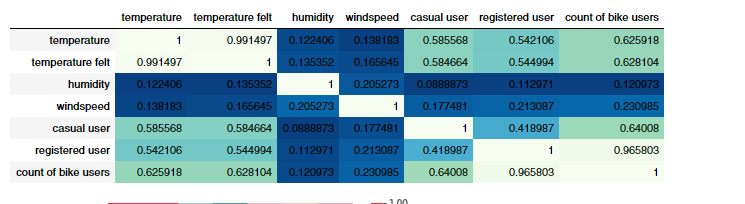
This becomes even more important when the number of features are very large. You need not use every feature at your disposal for creating an algorithm. You can assist your algorithm by feeding in only those features that are really important. I have myself witnessed feature subsets giving better results than complete set of feature for the same algorithm or – “Sometimes, less is better!”.

Corrgram : it help us visualize the data in correlation matrices. correlograms are implimented through the **corrgram(x, order = , panel=, lower.panel=, upper.panel=, text.panel=, diag.panel=)**

We should consider the selection of feature for model based on below criteria

1. The relationship between two independent variable should be less and
2. The relationship between Independent and Target variables should be high.

Below fig 2.9 illustrates that relationship between all numeric variables using Corrgram plot.



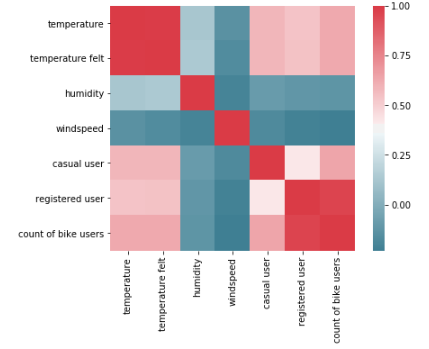
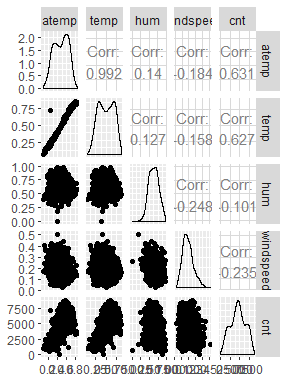


Figure 2.11 correlation plot of numeric variables (R code in Appendix B)

Color dark blue indicates there is strong positive relationship and if darkness is decreasing indicates relation between variables are decreasing.

Color dark Red indicates there is strong negative relationship and if darkness is decreasing indicates relationship between variables are decreasing



**Fig-2.12 correlation of numerical data along with scatterplot and distribution among them**

**2.4.1 Dimensionality Reduction for numeric variables**

Above Fig 2.11 is showing

There is strong relationship between independent variables ‘temp’ and ‘atemp’ so considering any one feature enough to predict the better.

And it is also showing there is almost no relationship between independent variable ‘hum’ and dependent variable ‘cnt’. so, ‘hum’ is not so important to predict.

Subsetting two independent features ‘atemp’ and ‘hum’ from actual dataset.

**2.2.4 Features Scaling**

The word “normalization” is used informally in statistics, and so the term normalized data can have multiple meanings. In most cases, when you normalize data you eliminate the units of measurement for data, enabling you to more easily compare data from different places. Some of the more common ways to normalize data include:

Transforming data using a [z-score](http://www.statisticshowto.com/probability-and-statistics/z-score/) or [t-score](http://www.statisticshowto.com/probability-and-statistics/t-distribution/t-score-formula/). This is usually called standardization. In the vast majority of cases, if a statistics textbook is talking about normalizing data, then this is the definition of “normalization” they are probably using.

[Rescaling data](http://www.statisticshowto.com/what-is-rescaling-data/) to have values between 0 and 1. This is usually called feature scaling. One possible formula to achieve this is.

[http://www.statisticshowto.com/wp-content/uploads/2015/11/normalize-data.png](http://www.statisticshowto.com/wp-content/uploads/2015/11/normalize-data.png)

In rental dataset numeric variables like ‘temp’ , ‘atem’ ,’hum’ and ‘ windspeed’ are in normalization form so , we have to Normalize two variables ‘casual’ and ‘registered’

After normalize ‘casual’ and ‘registered’ variables look like in table below where all values between 0 and 1

Table Normalization of ‘casual’ and ‘registered

|  |  |
| --- | --- |
| casual | registered |
| 0.037852113 | 0.09384926 |
| 0.034624413 | 0.17455963 |
| 0.025234742 | 0.21628646 |
| 0.042840376 | 0.21628646 |
| 0.019366197 | 0.12575801 |

**Chapter 3**

**Modelling**

**3.1 Model Selection**

In out earlier stage of analysis we have come to understand that few variables like ‘temp’ ,’casual,’registered ‘ are going to play key role in model development , for model development dependent variable may fall under below categories

1. Nominal
2. Ordinal
3. Interval
4. Ratio

In our case dependent variable is interval so, the predictive analysis that we can perform is **Regression** Analysis

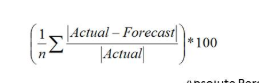
We will start our model building from Decision Tree.

**3.1.1 Evaluating Regression Model**

The main concept of looking at what is called **residuals** or difference between our predictions f(x[I,]) and actual outcomes y[i].

We are using two methods to evaluating performance of model

1. **MAPE**: (Mean Absolute Percent Error) measures the size of the error in percentage terms. It is calculated as the average of the unsigned percentage error.

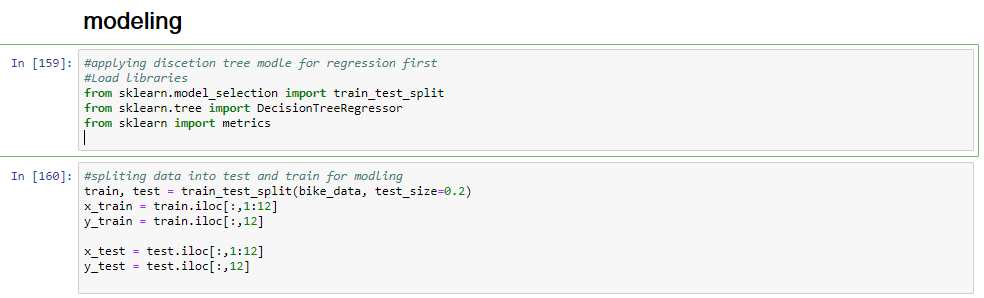


1. **RMSE:** (Root Mean Square Error) is a frequently used measure of the difference between values predicted by a model and the values actually observed from the environment that is being modelled.



After doing these pre processing we need to train our model so that we can predict the outcome in future. Here we need to split our data and then train our model.

**Splitting data:** we need to divide the data into train(80 percent) and test(20 percent)**.**



Model selection: we need to decide which model we need to use for our data. The target variable in our model is a continuous variable.So the models that we choose are Decision Tree and Random Forest, Linear Regression,OLS(python). The error metric chosen for the given problem statement is mean\_absolute\_error.

**3.2 Decision Tree**

A tree has many analogies in real life, and turns out that it has influenced a wide area of **machine learning**, covering both **classification and regression**. In decision analysis, 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.

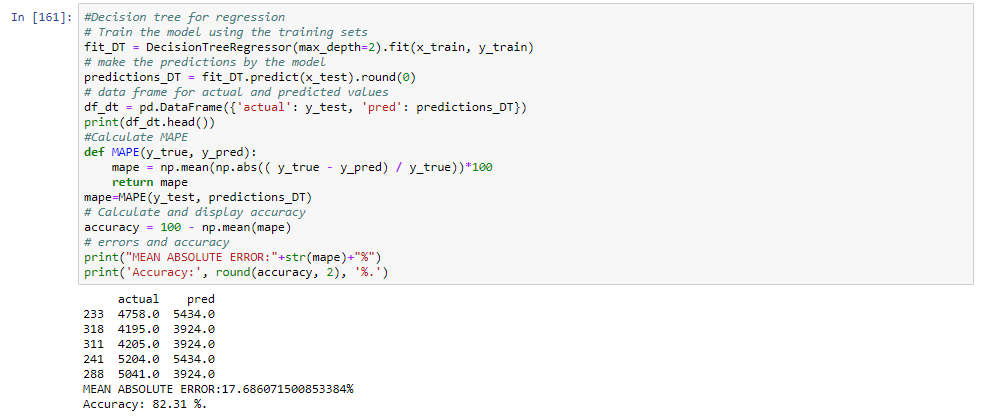


Figure 3.2.1 Decision Tree Algorithm

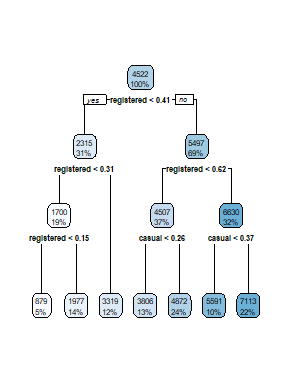


Figure 3.2.2 Graphical Representation of Decision tree

Look at the above figure 3.2 here decision tree is using only two predictors variables to predict the model , which is not very impressive here the model is overfitted and biased towards only two predictors i.e ‘casual’ and ‘registered’ .

**3.2.1 Evaluation of Decision Tree Model**

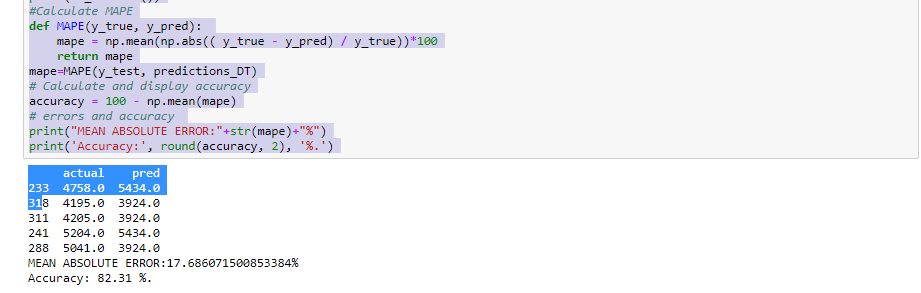


Figure 3.2.3 Evaluation of Decision Tree using MAPE and RMSE

Decision tree builds regression is in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

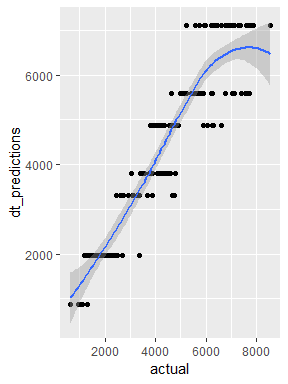


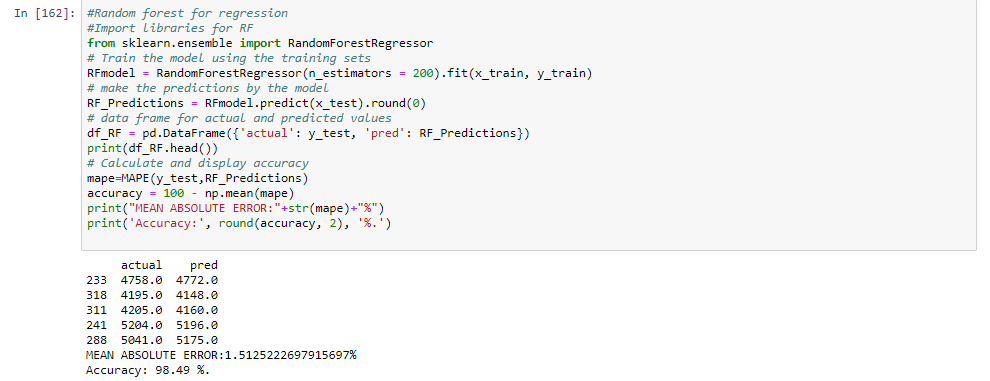
Fig-3.2.4 shows a curve between predicted and actual test data for target variable

|  |  |  |
| --- | --- | --- |
|  | Mae | Accuracy= (1-mae)\*100 |
| Python | 17.68% | 82.31% |
| R | 15.23% | 84.77% |
|  |  |  |

**3.3 Random Forest**

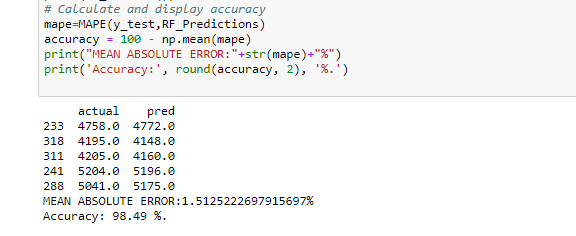
Random forests or random decision forests are an [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning) method for [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and other tasks, that operate by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) at training time and outputting the class that is the [mode](https://en.wikipedia.org/wiki/Mode_(statistics)) of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of [overfitting](https://en.wikipedia.org/wiki/Overfitting) to their [training set](https://en.wikipedia.org/wiki/Test_set).

Figure 3.3.1 Random Forest Implementation



**3.3.1 Evaluation of Random Forest**

Figure 3.2.2 Random Forest Evaluation



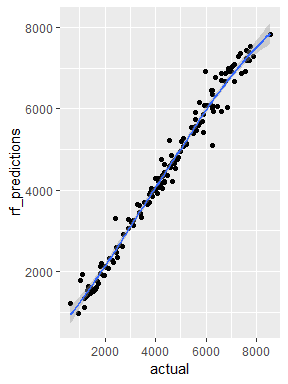


Fig-3.3.3 shows a curve between predicted and actual test data for target variable

|  |  |  |
| --- | --- | --- |
|  | Mae | Accuracy= (1-mae)\*100 |
| Python | 1.512% | 98.49% |
| R | 6.089% | 93.911% |
|  |  |  |

**3.4 Linear Regression**

[Multiple linear regression](http://www.statisticssolutions.com/academic-solutions/membership-resources/member-profile/data-analysis-plan-templates/data-analysis-plan-multiple-linear-regression/) is the most common form of linear regression analysis.  As a predictive analysis, the 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.

**VIF ( Variance Inflation factor )** : It quantifies the multicollinearity between the independent variables.

As Linear regression will work well if multicollinearity between the Independent variables are less.

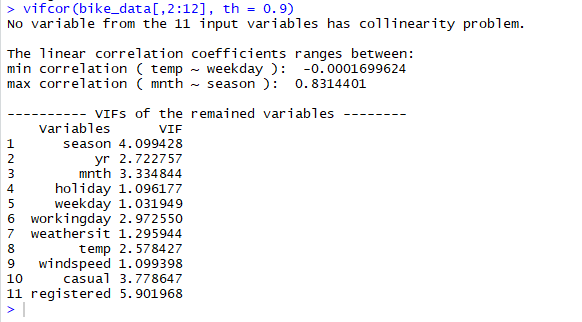
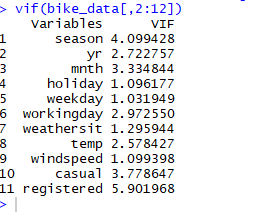


Figure 3.4.1 Multi collinearity between Independent variables

In the above figure it is showing there is strong correlation between no two independent variable, we need to consider all variable



Figure 3.4.2 Linear Regression Model

**3.4.2 Evaluation of Linear regression Model**

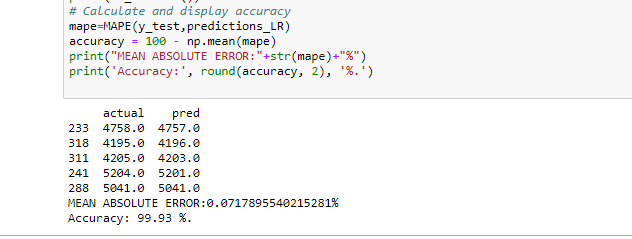


Figure 3.4.3 Evaluation of Regression Model

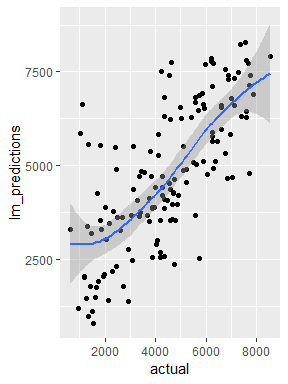


Fig-3.4.4 shows a curve between predicted and actual test data for target variable

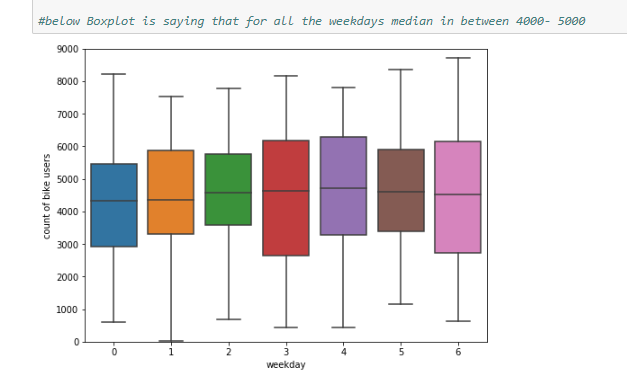
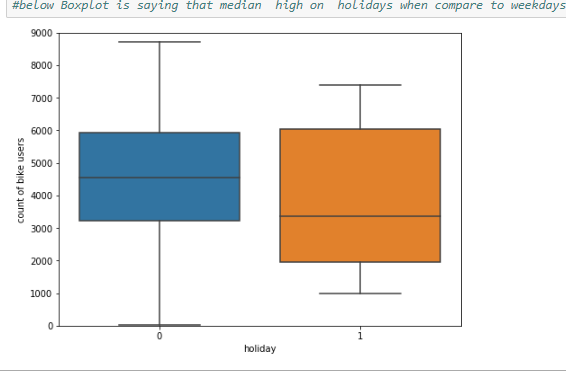
|  |  |  |
| --- | --- | --- |
|  | Mae | Accuracy= (1-mae)\*100 |
| Python | 0.071% | 99.93% |
| R | 3.75% | 96.25% |
|  |  |  |

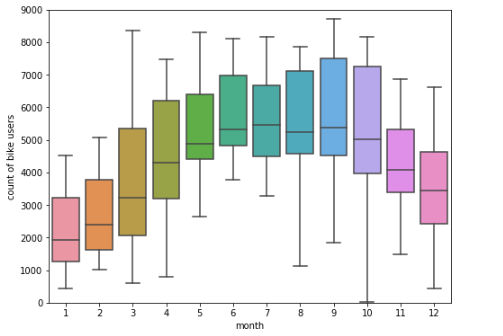
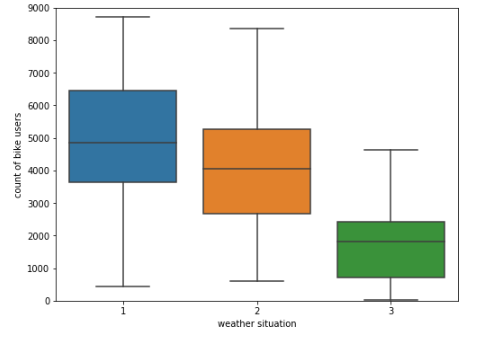
**Model Selection**

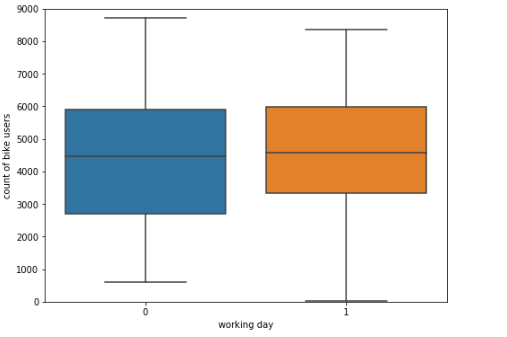
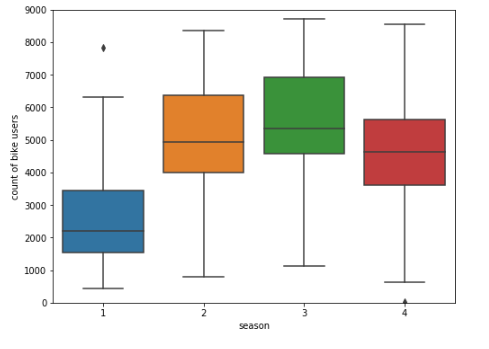
As we predicted counts for Bike Rental using three Models Decision Tree, Random Forest and Linear Regression as MAPE is high and RMSE is less for the Linear regression Model so conclusion is

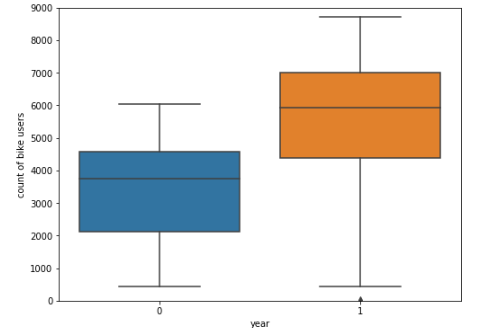
**Conclusion**: - For the Bike Rental Data Linear Regression Model is best model to predict the count.

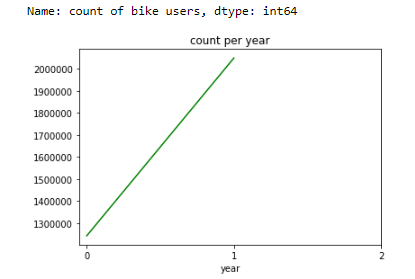
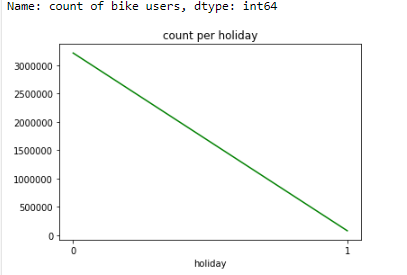
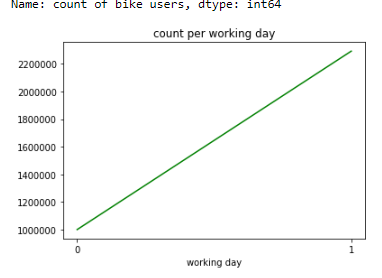
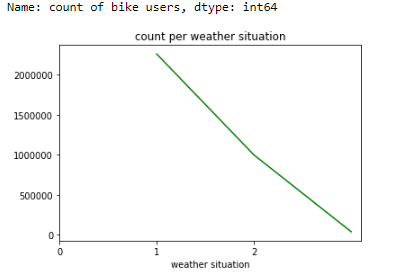
**Appendix A- Extra Figures**

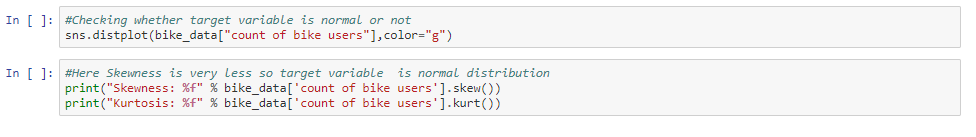
 



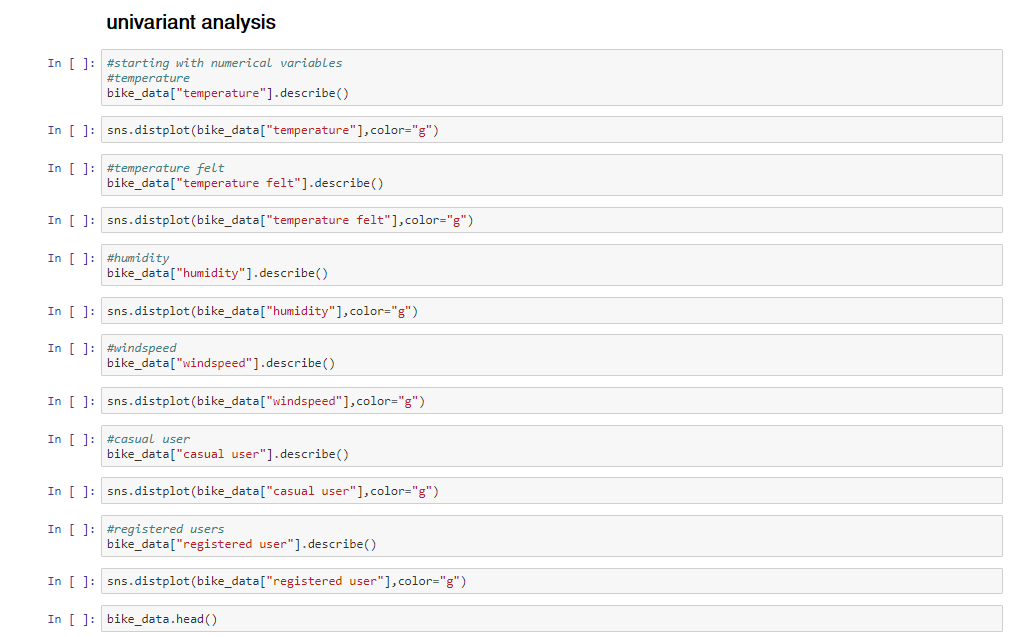
   

**Appendix B - Python Code**

[**Univariate Analysis (Fig:**](#page24) [**2.2, 2.3 & 2.4)**](#page6)



Python code for fig 2.2



Python code for fig 2.3

#Distribution of factor data using bar plot

bar1 = ggplot(data = bike\_data, aes(x = bike\_data$season)) + geom\_bar() + ggtitle("season") + theme\_dark()

bar2 = ggplot(data = bike\_data, aes(x = bike\_data$yr)) + geom\_bar() + ggtitle("year") + theme\_dark()

bar3 = ggplot(data = bike\_data, aes(x = bike\_data$mnth)) + geom\_bar() + ggtitle("Month") + theme\_dark()

bar4 = ggplot(data = bike\_data, aes(x = bike\_data$holiday)) + geom\_bar() + ggtitle("holiday") + theme\_dark()

bar5 = ggplot(data = bike\_data, aes(x = bike\_data$workingday)) + geom\_bar() + ggtitle("working day") + theme\_dark()

bar6 = ggplot(data = bike\_data, aes(x = bike\_data$weathersit)) + geom\_bar() + ggtitle("weather situation") + theme\_dark()

#making a grid

gridExtra::grid.arrange(bar1,bar2,ncol=1)

gridExtra::grid.arrange(bar3,bar4,ncol=1)

gridExtra::grid.arrange(bar5,bar6,ncol=2)

R code for fig 2.4

[**Bivariate Analysis (Fig:**](#page24) [**2.5 &2.6)**](#page7)



Python code for fig 2.5

#check the relationship between 'temp' and 'atemp' variable

ggplot(bike\_data, aes(x= temp,y=atemp)) +

geom\_point()+

geom\_smooth()

#This graph is saying that very strong relationship between 'temp' and 'atemp'

#check the relationship between 'temp' and 'hum' variable

ggplot(bike\_data, aes(x= temp,y=hum)) +

geom\_point()+

geom\_smooth()

# here it is showing Humidity is increses till temparature is 0.7 and it is decreasing gradually

#check the relationship between 'temp' and 'windspeed' variable

ggplot(bike\_data, aes(x= temp,y=windspeed)) +

geom\_point()+

geom\_smooth()

# it is showing that very less nagative correlation between temp and windspeed

#check the relationship between all numeric variable using pair plot

ggpairs(bike\_data[,c('atemp','temp','hum','windspeed','cnt')])

# that above plot stating that less nagative relationship between

# 'cnt'-'hum' and cnt-windspeed

# and there is strong positive relationship between

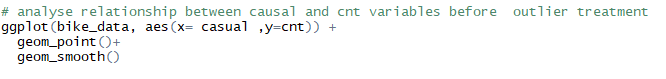
# temp- cnt and atemp-cnt

R code for fig 2.5

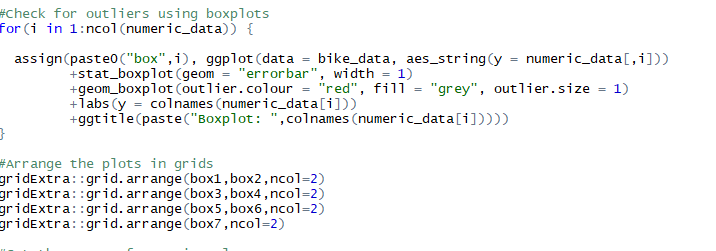


Python code for fig 2.6

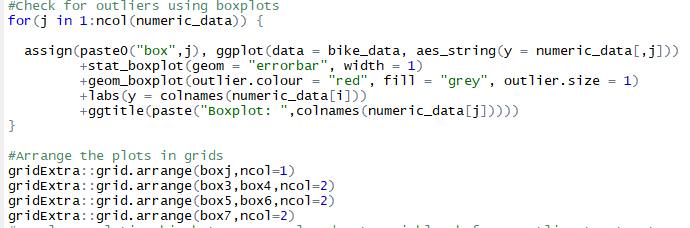
[**Outlier Analysis (Fig:**](#page24) [**2.7, 2.8, 2.9& 2.10)**](#page19)



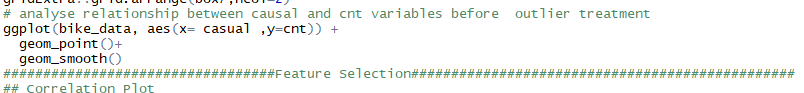
R code for fig 2.9



R code for fig 2.7



R code for fig 2.8

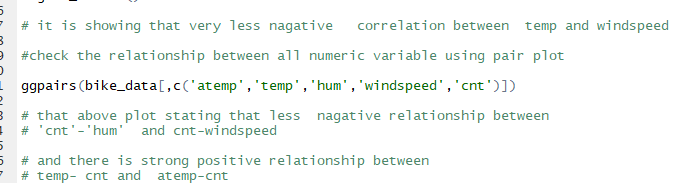


R code for fig 2.10

[**Feature Selection (Fig:**](#page24) [**2.11&2.12)**](#page8)



Python code for fig 2.11

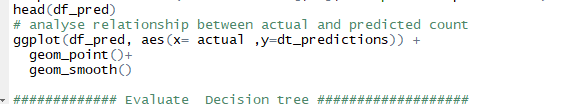


R code for fig 2.12

**Decision Tree(Fig: 3.2.2&3.2.4)**

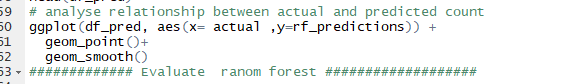


R code for fig 3.2.2



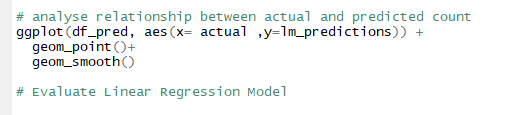
R code for fig 3.2.4

**Random Forest (Fig:3.3.3)**



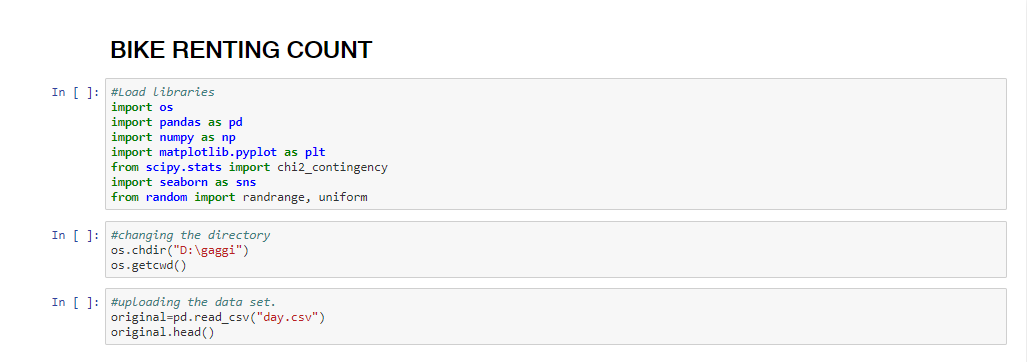
R code for fig 3.3.3

**Linear Regression (Fig:3.4.4)**

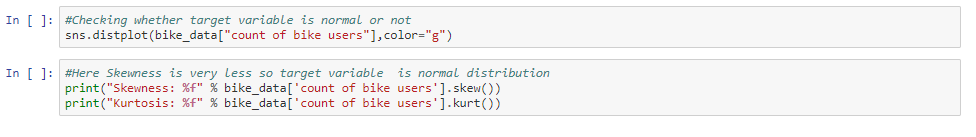


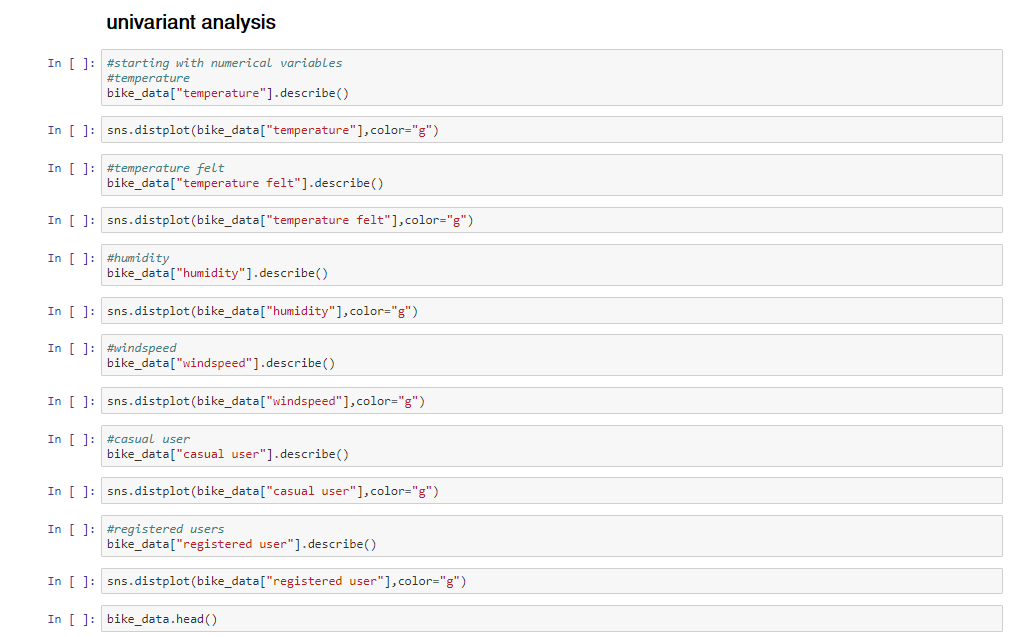
R code for fig 3.4.4

**Complete Python File**





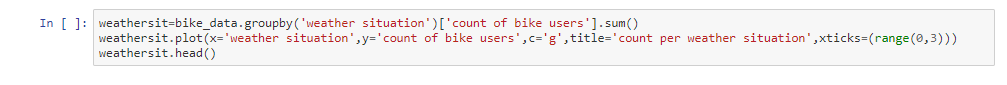


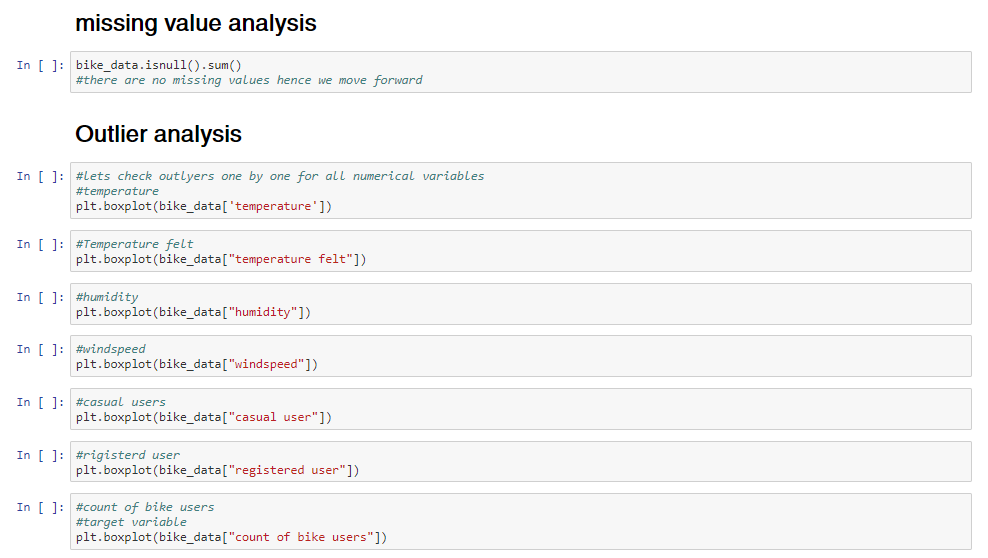


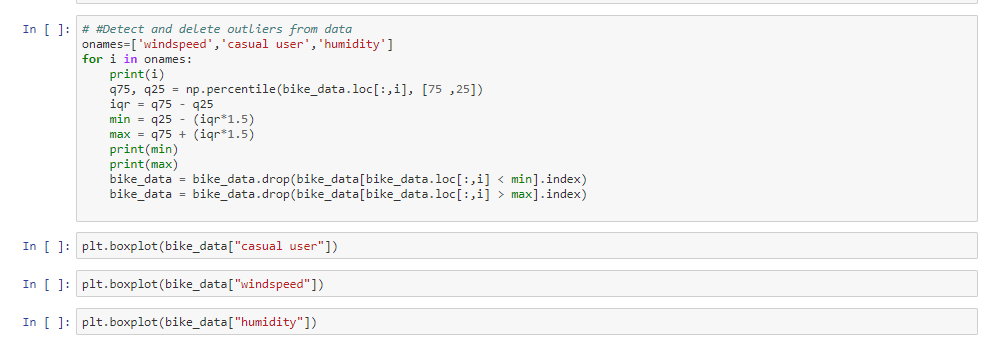






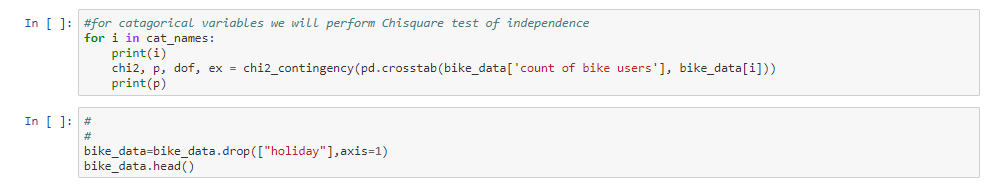


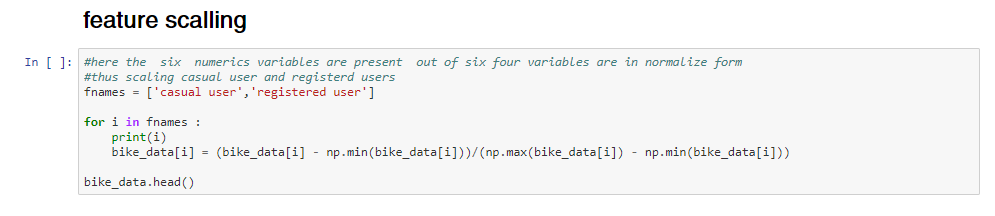


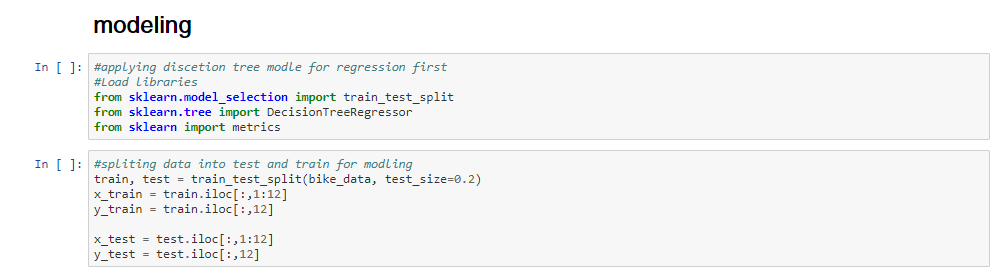


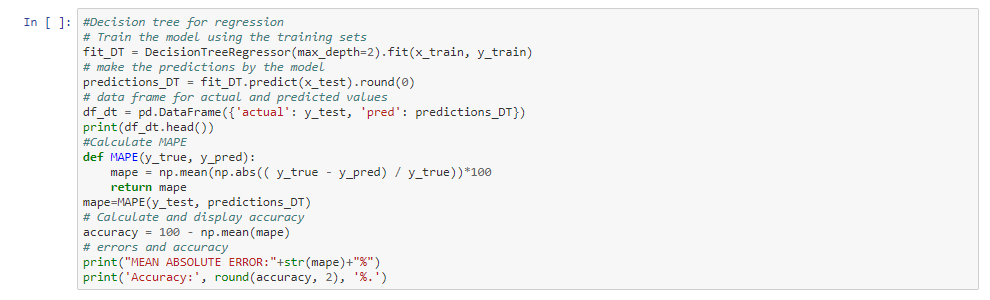


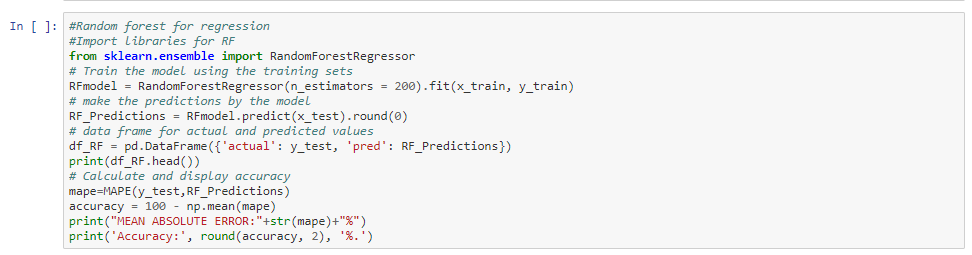














**Complete R File**

#Clean the environment

rm(list = ls())

#Seting the working directory

setwd("D:/gaggi")

#get Working directory

getwd()

#Loading the librarires which would be needed

libraries = c("dummies","caret","rpart.plot","plyr","dplyr", "ggplot2","rpart","dplyr","DMwR","randomForest","usdm","DataCombine")

lapply(X = libraries,require, character.only = TRUE)

rm(libraries)

#Read the csv file

bike\_data = read.csv("day.csv", header = T)

######Exploratory Data Analysis##################

# Summarizing data

colnames(bike\_data)

#Verify first five rows of data

head(bike\_data)

#target variable is 'cnt' and other variables are independent variable(or predictors)

summary(bike\_data$cnt)

#Verify summary of data

summary(bike\_data)

#It shows variables like 'mnth',holiday','weekday','weathersit' are

#catogical variabless and already encoded

#Nummeric vaiables like 'temp','atem','hum','windspeed' are

#standardized form

# data contains no missing values

# Outliers might be present in variables 'actual','registered','cnt'

#instant variable is jus an aditional index for the data so droping the same

bike\_data=subset(bike\_data, select=-c(instant))

#structure of data

str(bike\_data)

#############converting in proper datatype############################

bike\_data$season=as.factor(bike\_data$season)

bike\_data$mnth=as.factor(bike\_data$mnth)

bike\_data$yr=as.factor(bike\_data$yr)

bike\_data$holiday=as.factor(bike\_data$holiday)

bike\_data$weekday=as.factor(bike\_data$weekday)

bike\_data$workingday=as.factor(bike\_data$workingday)

bike\_data$weathersit=as.factor(bike\_data$weathersit)

bike\_data$dteday=as.Date.factor(bike\_data$dteday)

bike\_data$temp=as.numeric(bike\_data$temp)

bike\_data$atemp=as.numeric(bike\_data$atemp)

bike\_data$hum=as.numeric(bike\_data$hum)

bike\_data$windspeed=as.numeric(bike\_data$windspeed)

bike\_data$casual=as.numeric(bike\_data$casual)

bike\_data$registered=as.numeric(bike\_data$registered)

bike\_data$cnt=as.numeric(bike\_data$cnt)

str(bike\_data)

#######Missing Values Analysis###############################################

#checking for missing value

missing\_val = data.frame(apply(bike\_data,2,function(x){sum(is.na(x))}))

head(missing\_val,16)

#as seen earliear also there are no missing values in the data thus we need not perform missing value analysis

# Analyze variables by visualize

#Distribution of factor data using bar plot

bar1 = ggplot(data = bike\_data, aes(x = bike\_data$season)) + geom\_bar() + ggtitle("season") + theme\_dark()

bar2 = ggplot(data = bike\_data, aes(x = bike\_data$yr)) + geom\_bar() + ggtitle("year") + theme\_dark()

bar3 = ggplot(data = bike\_data, aes(x = bike\_data$mnth)) + geom\_bar() + ggtitle("Month") + theme\_dark()

bar4 = ggplot(data = bike\_data, aes(x = bike\_data$holiday)) + geom\_bar() + ggtitle("holiday") + theme\_dark()

bar5 = ggplot(data = bike\_data, aes(x = bike\_data$workingday)) + geom\_bar() + ggtitle("working day") + theme\_dark()

bar6 = ggplot(data = bike\_data, aes(x = bike\_data$weathersit)) + geom\_bar() + ggtitle("weather situation") + theme\_dark()

#making a grid

gridExtra::grid.arrange(bar1,bar2,ncol=1)

gridExtra::grid.arrange(bar3,bar4,ncol=1)

gridExtra::grid.arrange(bar5,bar6,ncol=2)

#Check the distribution of numerical data using histogram

hist1 = ggplot(data = bike\_data, aes(x =bike\_data$cnt)) + ggtitle("Distribution of : count of bike users") + geom\_histogram(bins = 25)

hist2 = ggplot(data = bike\_data, aes(x =bike\_data$registered)) + ggtitle("Distribution of: registered user") + geom\_histogram(bins = 25)

hist3 = ggplot(data = bike\_data, aes(x =bike\_data$casual)) + ggtitle("Distribution of: casual user") + geom\_histogram(bins = 25)

hist4 = ggplot(data = bike\_data, aes(x =bike\_data$windspeed)) + ggtitle("Distribution of : windspeed") + geom\_histogram(bins = 25)

hist5 = ggplot(data = bike\_data, aes(x =bike\_data$hum)) + ggtitle("Distribution of : humidity") + geom\_histogram(bins = 25)

hist6 = ggplot(data = bike\_data, aes(x =bike\_data$atemp)) + ggtitle("Distribution of : temprature felt") + geom\_histogram(bins = 25)

hist7 = ggplot(data = bike\_data, aes(x =bike\_data$temp)) + ggtitle("Distribution of : temperature") + geom\_histogram(bins = 25)

#making a grid

gridExtra::grid.arrange(hist1,hist2,ncol=1)

gridExtra::grid.arrange(hist3,hist4,ncol=1)

gridExtra::grid.arrange(hist5,hist6,hist7,ncol=2)

# \*\*\*bivariate relationship between numeric variables\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#check the relationship between 'temp' and 'atemp' variable

ggplot(bike\_data, aes(x= temp,y=atemp)) +

geom\_point()+

geom\_smooth()

#This graph is saying that very strong relationship between 'temp' and 'atemp'

#check the relationship between 'temp' and 'hum' variable

ggplot(bike\_data, aes(x= temp,y=hum)) +

geom\_point()+

geom\_smooth()

# here it is showing Humidity is increses till temparature is 0.7 and it is decreasing gradually

#check the relationship between 'temp' and 'windspeed' variable

ggplot(bike\_data, aes(x= temp,y=windspeed)) +

geom\_point()+

geom\_smooth()

# it is showing that very less nagative correlation between temp and windspeed

#check the relationship between all numeric variable using pair plot

ggpairs(bike\_data[,c('atemp','temp','hum','windspeed','cnt')])

# that above plot stating that less nagative relationship between

# 'cnt'-'hum' and cnt-windspeed

# and there is strong positive relationship between

# temp- cnt and atemp-cnt

# \*\*\*\*\*\*\*\*\*\*\*\*\*visualize the relationship between categorical variable\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#check relationship between season and holiday

var1= table(bike\_data$season,bike\_data$holiday)

var1

barplot(var1)

# here contgency table showing holiday=0 is same for almost all the seasons

#check relationship between season and weekday

var2= table(bike\_data$season,bike\_data$weekday)

barplot(var2)

#check relationship between season and weathersit

var3= table(bike\_data$weathersit,bike\_data$season)

var3

prop.table(var3,2)

barplot(var3)

#It is stating that in all the season whether 1 type is large numbers

##check relationship between holiday and weathersit

var4= table(bike\_data$weathersit,bike\_data$holiday)

var4

barplot(var4)

#to check in proportion

prop.table(var4,2)

# it it staing that holiday type '0' and weathersit type '1' almost covered 0.63%

###### Outlier Analysis########################################################

#####OUTLIER ANALYSIS

# here we will replace the outliers with Knn method.

#Get the data with only numeric columns

numeric\_index = sapply(bike\_data, is.numeric)

numeric\_data = bike\_data[,numeric\_index]

#Get the data with only factor columns

factor\_data = bike\_data[,!numeric\_index]

# analyse relationship between causal and cnt variables before outlier treatment

ggplot(bike\_data, aes(x= casual ,y=cnt)) +

geom\_point()+

geom\_smooth()

#Check for outliers using boxplots

for(i in 1:ncol(numeric\_data)) {

assign(paste0("box",i), ggplot(data = bike\_data, aes\_string(y = numeric\_data[,i]))

+stat\_boxplot(geom = "errorbar", width = 1)

+geom\_boxplot(outlier.colour = "red", fill = "grey", outlier.size = 1)

+labs(y = colnames(numeric\_data[i]))

+ggtitle(paste("Boxplot: ",colnames(numeric\_data[i]))))

#Arrange the plots in grids

gridExtra::grid.arrange(box1,box2,ncol=2)

gridExtra::grid.arrange(box3,box4,ncol=2)

gridExtra::grid.arrange(box5,box6,ncol=2)

gridExtra::grid.arrange(box7,ncol=2)

#Get the names of numeric columns

numeric\_columns = colnames(numeric\_data)

#Replacing all outlier data with NA

for(i in numeric\_columns){

val = bike\_data[,i][bike\_data[,i] %in% boxplot.stats(bike\_data[,i])$out]

print(paste(i,length(val)))

bike\_data[,i][bike\_data[,i] %in% val] = NA

}

#Check number of missing values

sapply(bike\_data,function(x){sum(is.na(x))})

#Get number of missing values after replacing outliers as NA

missing\_values\_out = data.frame(sapply(bike\_data,function(x){sum(is.na(x))}))

missing\_values\_out$Columns = row.names(missing\_values\_out)

row.names(missing\_values\_out) = NULL

names(missing\_values\_out)[1] = "Missing\_percentage"

missing\_values\_out$Missing\_percentage = ((missing\_values\_out$Missing\_percentage/nrow(bike\_data)) \*100)

missing\_values\_out = missing\_values\_out[,c(2,1)]

missing\_values\_out = missing\_values\_out[order(-missing\_values\_out$Missing\_percentage),]

missing\_values\_out

#Compute the NA values using KNN imputation

bike\_data[,9:15] = knnImputation(bike\_data[ ,9:15], k = 3)

#Check if any missing values

sum(is.na(bike\_data))

#Check for outliers using boxplots

for(j in 1:ncol(numeric\_data)) {

assign(paste0("box",j), ggplot(data = bike\_data, aes\_string(y = numeric\_data[,j]))

+stat\_boxplot(geom = "errorbar", width = 1)

+geom\_boxplot(outlier.colour = "red", fill = "grey", outlier.size = 1)

+labs(y = colnames(numeric\_data[i]))

+ggtitle(paste("Boxplot: ",colnames(numeric\_data[j]))))

}

#Arrange the plots in grids

gridExtra::grid.arrange(boxj,ncol=1)

gridExtra::grid.arrange(box3,box4,ncol=2)

gridExtra::grid.arrange(box5,box6,ncol=2)

gridExtra::grid.arrange(box7,ncol=2)

# analyse relationship between causal and cnt variables before outlier treatment

ggplot(bike\_data, aes(x= casual ,y=cnt)) +

geom\_point()+

geom\_smooth()

###############Feature Selection################################################

## Correlation Plot

corrgram(bike\_data[,numeric\_index], order = F,

upper.panel=panel.pie, text.panel=panel.txt, main = "Correlation Plot")

# correlation matrix stating 'temp' and 'atemp' having strong relationship

# and there is no relationship between 'hum' and 'cnt'

# dimensional reduction

bike\_data = subset(bike\_data,select=-c(atemp,hum))

#####################Feature Scaling################################################

#Normalisation

cnames = c("casual","registered")

for(i in cnames){

print(i)

bike\_data[,i] = (bike\_data[,i] - min(bike\_data[,i]))/

(max(bike\_data[,i] - min(bike\_data[,i])))

}

######################Model Development#######################################

head(bike\_data)

########DECISION TREE

#Splitting the data (80-20 percent)

set.seed(1)

train\_index = sample(1:nrow(bike\_data), 0.8\*nrow(bike\_data))

train = bike\_data[train\_index,]

test = bike\_data[-train\_index,]

#Build decsion tree using rpart

dt\_model = rpart(cnt ~ ., data = train, method = "anova")

# here we can try any method other than anova ,

#one of "anova", "poisson", "class" or "exp".

#If method is missing then the routine tries to make an intelligent guess.

#Ploting the tree

rpart.plot(dt\_model)

#Perdict for test cases

dt\_predictions = predict(dt\_model, test[,-13])

df3= data.frame((dt\_predictions))

#Create data frame for actual and predicted values

df\_pred = data.frame("actual"= test[,13], "dt\_pred"=dt\_predictions)

head(df\_pred)

# analyse relationship between actual and predicted count

ggplot(df\_pred, aes(x= actual ,y=dt\_predictions)) +

geom\_point()+

geom\_smooth()

############# Evaluate Decision tree ###################

#MAPE

#calculate MAPE

MAPE = function(y, yhat){

mean(abs((y - yhat)/y))

}

MAPE(test[,13], dt\_predictions)

#Error Rate: 0.1523074

#Accuracy:

#Evaluate Model using RMSE

RMSE <- function(y\_test,y\_predict) {

difference = y\_test - y\_predict

root\_mean\_square = sqrt(mean(difference^2))

return(root\_mean\_square)}

RMSE(test[,13], dt\_predictions)

#RMSE = 733.1856

########RANDOM FOREST

#Training the model using training data

rf\_model = randomForest(cnt~., data = train, ntree = 500)

#Predict the test cases

rf\_predictions = predict(rf\_model, test[,-13])

#Create dataframe for actual and predicted values

df\_pred = cbind(df\_pred,rf\_predictions)

head(df\_pred)

# analyse relationship between actual and predicted count

ggplot(df\_pred, aes(x= actual ,y=rf\_predictions)) +

geom\_point()+

geom\_smooth()

############# Evaluate ranom forest ###################

#MAPE

#calculate MAPE

MAPE(test[,13], rf\_predictions)

#0.06089973

#

RMSE(test[,13], rf\_predictions)

#RMSE = 293.857

#################### Develop Linear Regression Model ##########################

#check multicollearity

library(usdm)

#converting multilevel categorical variable into ineger again

bike\_data$season=as.integer(bike\_data$season)

bike\_data$mnth=as.integer(bike\_data$mnth)

bike\_data$yr=as.integer(bike\_data$yr)

bike\_data$holiday=as.integer(bike\_data$holiday)

bike\_data$weekday=as.integer(bike\_data$weekday)

bike\_data$workingday=as.integer(bike\_data$workingday)

bike\_data$weathersit=as.integer(bike\_data$weathersit)

vif(bike\_data[,2:12])

vifcor(bike\_data[,2:12], th = 0.9)

# develop Linear Regression model

#dividind data into test and train

train\_index = sample(1:nrow(bike\_data), 0.8 \* nrow(bike\_data))

train\_lr = bike\_data[train\_index,]

test\_lr = bike\_data[-train\_index,]

#run regression model

lm\_model = lm(cnt ~., data = train\_lr)

#Summary of the model

summary(lm\_model)

# observe the residuals and coefficients of the linear regression model

# Predict the Test data

#Predict

lm\_predictions = predict(lm\_model, test\_lr[,-13])

#Creating a new dataframe for actual and predicted values

df\_pred = cbind(df\_pred,lm\_predictions)

head(df\_pred)

# analyse relationship between actual and predicted count

ggplot(df\_pred, aes(x= actual ,y=lm\_predictions)) +

geom\_point()+

geom\_smooth()

# Evaluate Linear Regression Model

MAPE(test\_lr[,13], lm\_predictions)

#Error Rate: 6.416578e-16

#Accuracy: 99.9 + accuracy

RMSE(test\_lr[,13], lm\_predictions)

#RMSE = 2.327632e-12

# COnclusion For this Dataset Linear Regression is Accuracy is '99.9'

# and RMSE = 2.327632e-12

**References**

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical* *Learning*. Vol. 6. Springer.

Wickham, Hadley. 2009. *Ggplot2: Elegant Graphics for Data Analysis*. Springer Science & Business Media

https://stackoverflow.com

https://towardsdatascience.com/