Submitted By:

Tejasvi Rastogi

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Description** | **Page No** |
| 1. | 1-Introduction | 3 |
| 2. | 1.1Problem Statement | 4 |
| 3. | 1.2 Data | 5-6 |
| 4. | 2-Methodology | 7 |
| 5. | 2.1 Data Analysis | 8 |
| 6. | 2.1.1-Pre-Processing |  |
| 7. | 2.1.2 Univariate Analysis |  |
| 8. | 2.1.3 Bivariate Analysis |  |
| 9. | 2.2 Processing Techniques |  |
| 10. |  |  |

**1.1 Problem Statement:**

The objective of this Case is to predication of bike rental count on daily based on the environmental and seasonal setting. 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 counting order to automate the system.

**1.2 Data:**

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **instant** | **dteday** | **season** | **yr** | **mnth** | **holiday** | **weekday** | **workingday** | **weathersit** |
| 1 | 1/1/2011 | 1 | 0 | 1 | 0 | 6 | 0 | 2 |
| 2 | 1/2/2011 | 1 | 0 | 1 | 0 | 0 | 0 | 2 |
| 3 | 1/3/2011 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 4 | 1/4/2011 | 1 | 0 | 1 | 0 | 2 | 1 | 1 |
| 5 | 1/5/2011 | 1 | 0 | 1 | 0 | 3 | 1 | 1 |
| 6 | 1/6/2011 | 1 | 0 | 1 | 0 | 4 | 1 | 1 |
| 7 | 1/7/2011 | 1 | 0 | 1 | 0 | 5 | 1 | 2 |

**Table 1- Bike Rental Sample Data Column (1-8)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| temp | atemp | hum | windspeed | casual | registered | cnt |
| 0.344167 | 0.363625 | 0.805833 | 0.160446 | 331 | 654 | 985 |
| 0.363478 | 0.353739 | 0.696087 | 0.248539 | 131 | 670 | 801 |
| 0.196364 | 0.189405 | 0.437273 | 0.248309 | 120 | 1229 | 1349 |
| 0.2 | 0.212122 | 0.590435 | 0.160296 | 108 | 1454 | 1562 |
| 0.226957 | 0.22927 | 0.436957 | 0.1869 | 82 | 1518 | 1600 |
| 0.204348 | 0.233209 | 0.518261 | 0.089565 | 88 | 1518 | 1606 |
| 0.196522 | 0.208839 | 0.498696 | 0.168726 | 148 | 1362 | 1510 |

**Table 2- Bike Rental Sample Data Column (9-15)**

In the given data set we have 15 variables, 14 are independent and 1 is dependent variable. Below are the variables which we are going to used to predict the count of bike rentals.

|  |  |
| --- | --- |
| 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 |

**Table-3 Bike Rental Predictor**

**2.1 Data Analysis:**

Data analysis is a process of inspecting, [cleansing](https://en.wikipedia.org/wiki/Data_cleansing), [transforming](https://en.wikipedia.org/wiki/Data_transformation), and [modeling](https://en.wikipedia.org/wiki/Data_modeling) [data](https://en.wikipedia.org/wiki/Data) with the goal of discovering useful information, informing conclusions, and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. In today's business world, data analysis plays a role in making decisions more scientific and helping businesses operate more effectively.

**2.1.1 Pre-Processing:**

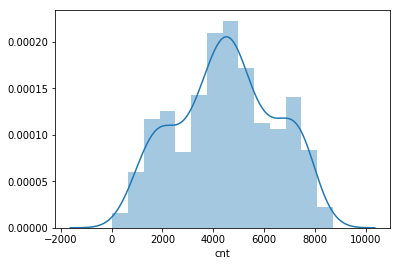
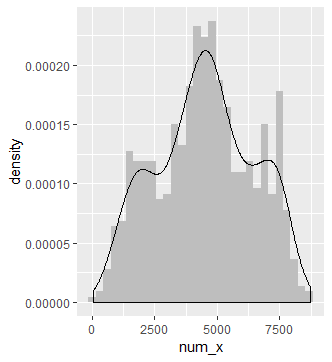
Data preprocessing is an important step in the [data mining](https://en.wikipedia.org/wiki/Data_mining) process. The phrase ["garbage in, garbage out"](https://en.wikipedia.org/wiki/GIGO) is particularly applicable to [data mining](https://en.wikipedia.org/wiki/Data_mining) and [machine learning](https://en.wikipedia.org/wiki/Machine_learning) projects. Data-gathering methods are often loosely controlled, resulting in [out-of-range](https://en.wikipedia.org/w/index.php?title=Range_error&action=edit&redlink=1) values (e.g., Income: −100), impossible data combinations (e.g., Sex: Male, Pregnant: Yes), [missing values](https://en.wikipedia.org/wiki/Missing_values), etc. Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and [quality of data](https://en.wikipedia.org/wiki/Data_quality) is first and foremost before running an analysis. Often, data preprocessing is the most important phase of a [machine learning](https://en.wikipedia.org/wiki/Machine_learning) project, especially in [computational biology](https://en.wikipedia.org/wiki/Computational_biology). If there is much irrelevant and redundant information present or noisy and unreliable data, then [knowledge discovery](https://en.wikipedia.org/wiki/Knowledge_discovery) during the training phase is more difficult. Data preparation and filtering steps can take considerable amount of processing time. Data preprocessing includes [cleaning](https://en.wikipedia.org/wiki/Data_cleaning), [Instance selection](https://en.wikipedia.org/wiki/Instance_selection), [normalization](https://en.wikipedia.org/wiki/Data_normalization), [transformation](https://en.wikipedia.org/wiki/Data_transformation), [feature extraction](https://en.wikipedia.org/wiki/Feature_extraction) and [selection](https://en.wikipedia.org/wiki/Feature_selection), etc. The product of data preprocessing is the final [training set](https://en.wikipedia.org/wiki/Training_set)

**2.1.2 Univariate Analysis:**

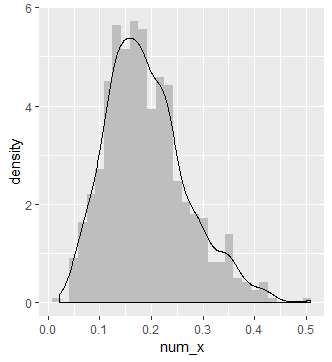
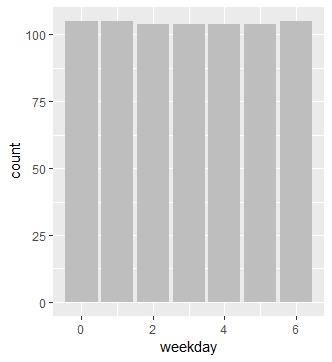
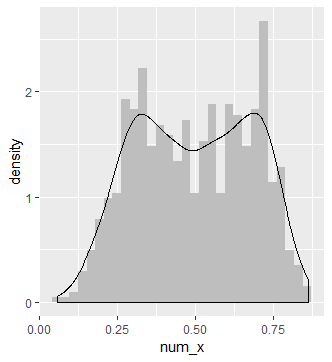
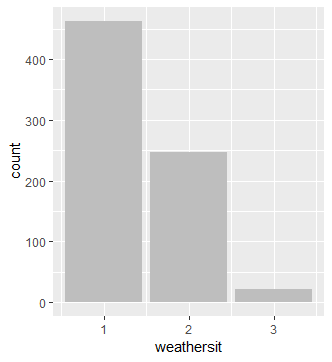
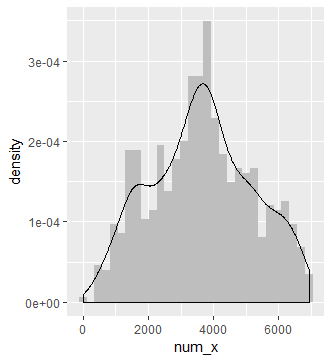
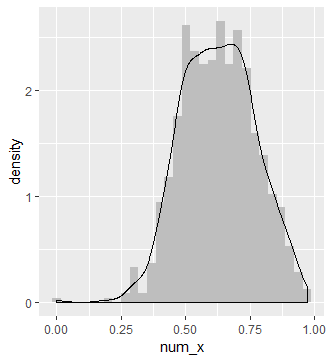
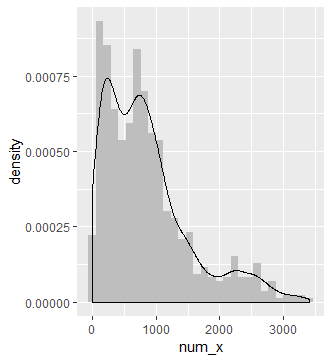
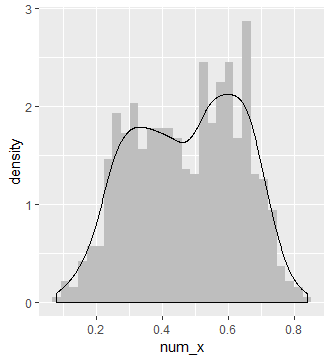
In the following Figure we have plotted the probability density functions numeric variables present in the data including target variable cnt.

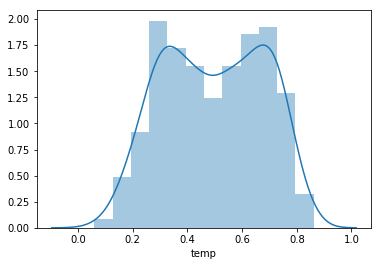
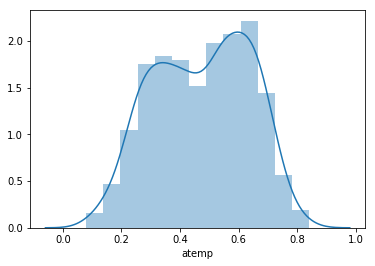
1. Target variable cnt is normally distributed.
2. Independent variables like ‘temp’,’atemp’, and ‘registered’ data are distributed normally.
3. Independent variable ‘casual’ data is slightly skewed to the right so, there are chances of getting outliers.
4. Other Independent variable ‘hum’ data is slightly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

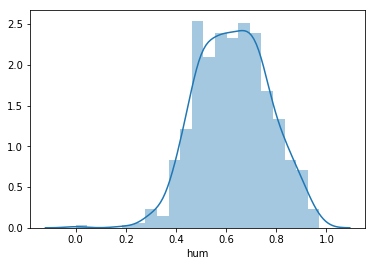
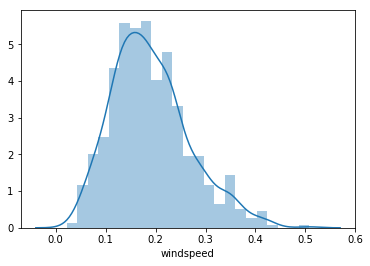
**Figure-Distribution of target Variable (CNT) in R and Python**

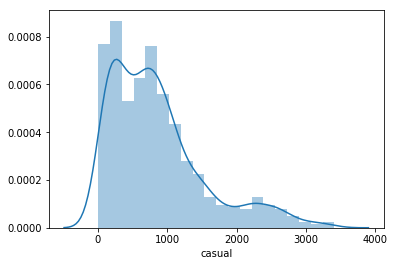
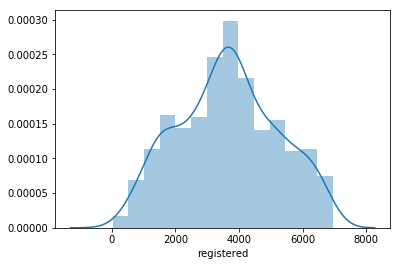
 

**Figure-showing distribution of dependent variables in both R and Python**

**2.1.3 Bivariate Analysis**

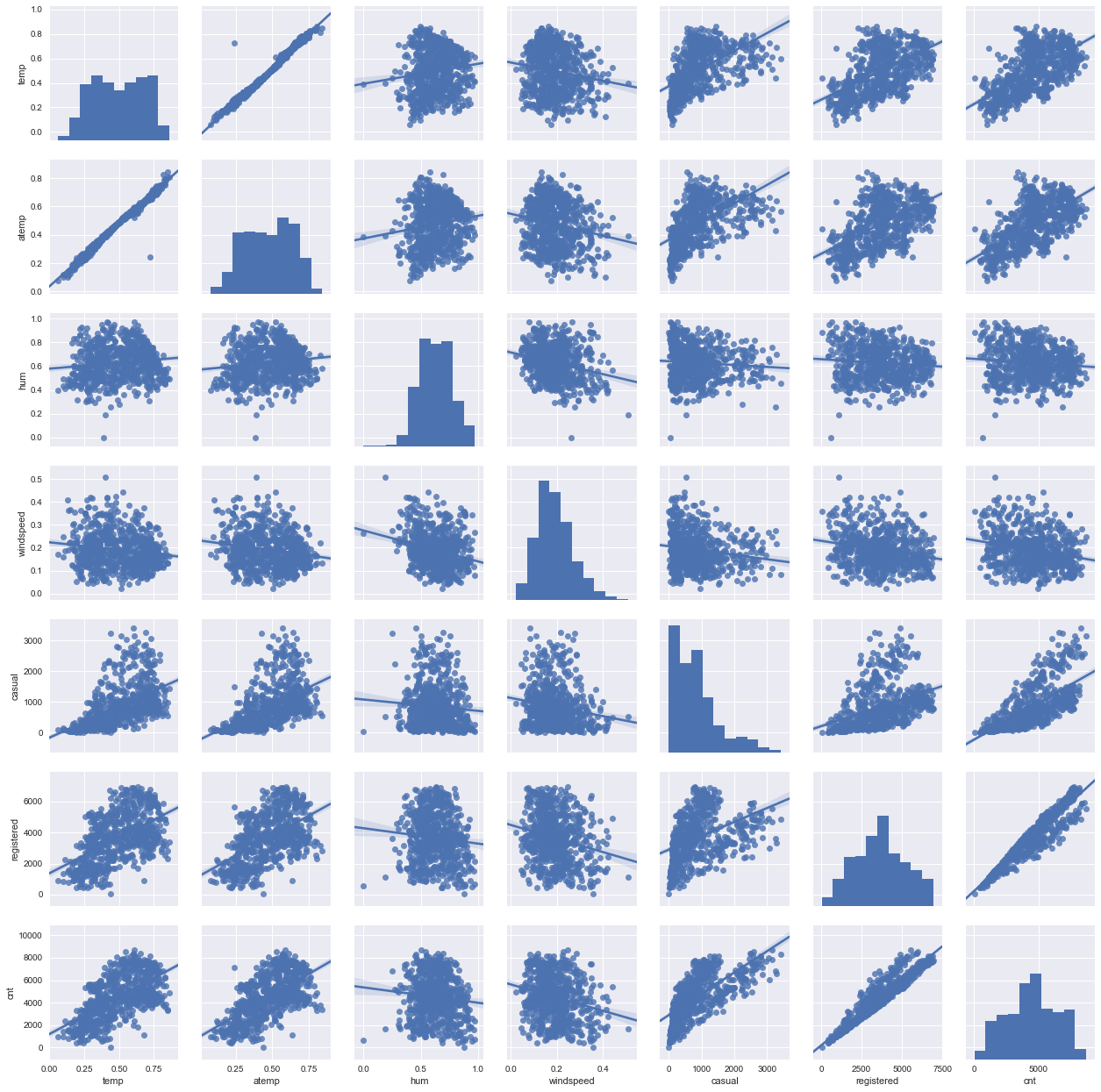
Bivariate analysis is one of the simplest forms of [quantitative (statistical) analysis](https://en.wikipedia.org/wiki/Statistics). It involves the analysis of two [variables](https://en.wikipedia.org/wiki/Dependent_and_independent_variables) (often denoted as *X*, *Y*), for the purpose of determining the empirical relationship between them. Bivariate analysis can be helpful in testing simple [hypotheses](https://en.wikipedia.org/wiki/Hypotheses) of [association](https://en.wikipedia.org/wiki/Association_(statistics)). Bivariate analysis can help determine to what extent it becomes easier to know and predict a value for one variable (possibly a [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable)) if we know the value of the other variable (possibly the [independent variable](https://en.wikipedia.org/wiki/Independent_variable)) (see also [correlation](https://en.wikipedia.org/wiki/Correlation" \o "Correlation)and [simple linear regression](https://en.wikipedia.org/wiki/Simple_linear_regression)). Bivariate analysis can be contrasted with [univariate analysis](https://en.wikipedia.org/wiki/Univariate_analysis" \o "Univariate analysis) in which only one variable is analysed. Like univariate analysis, bivariate analysis can be [descriptive](https://en.wikipedia.org/wiki/Descriptive_statistics) or [inferential](https://en.wikipedia.org/wiki/Inferential_statistics). It is the analysis of the relationship between the two variables. Bivariate analysis is a simple (two variable) special case of [multivariate analysis](https://en.wikipedia.org/wiki/Multivariate_analysis) (where multiple relations between multiple variables are examined simultaneously).

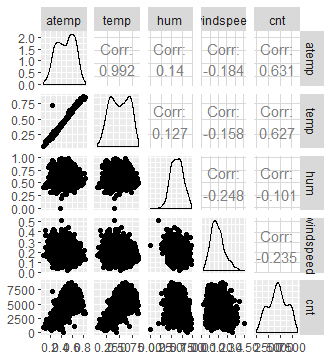
Ggpair function built upon [ggplot2](http://ggplot2.org/), [GGally](https://github.com/ggobi/ggally) provides templates for combining plots into a matrix through the ggpairs function. Such a matrix of plots can be useful for quickly exploring therelationships between multiple columns of data in a data frame.The lower and upper arguments to the ggpairs function specifiesthe type of plot or data in each position of the lower or upper diagonalof the matrix, respectively.For continuous X and Y data, one can specify the smooth option toinclude a regression line.

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

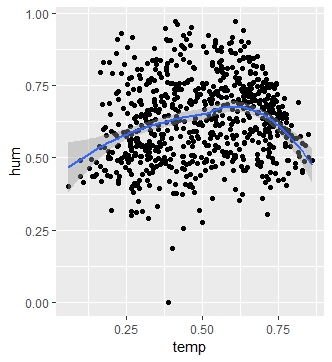
1. Below ggpair graph is showing clearly that relationship between independent variables ‘temp’ and ‘atemp’ are very strong.
2. The relationship between ‘hum’ , ‘windspeed’ with target variable ‘cnt’ is less.

**Figure-Relationship between numeric variables in both R and Python:**

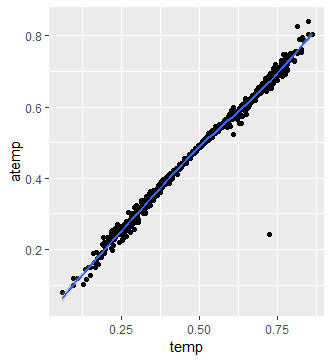




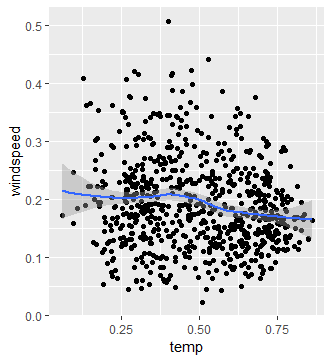
**Relationship plot of all the variables:**



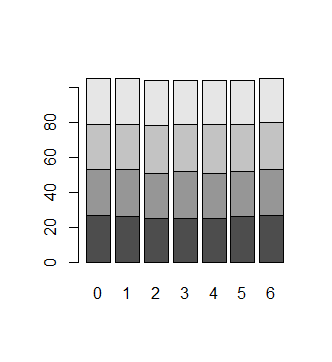
**Figure: Relationship between ‘temp’ & ‘hum’**



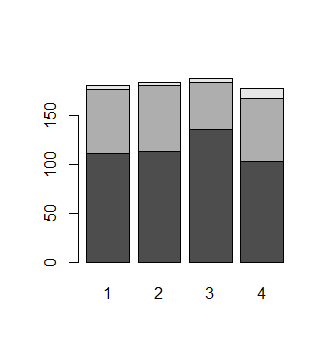
**Figure: Relationship between ‘temp’ & ‘atemp’**



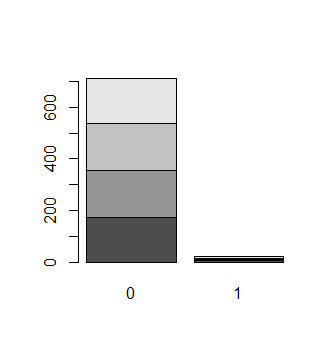
**Figure: Relationship between ‘temp’ & ‘wind’**



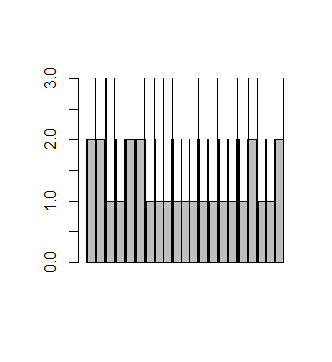
**Figure: Relationship between ‘season’ & ‘weekday’**



**Figure: Relationship between ‘season’ & ‘weathersit’**



**Figure: Relationship between ‘season’ & ‘holiday’**



**Figure: Relationship between ‘holiday’ & ‘weathersit’**

**2.2.1 MISSING VALUE ANALYSIS:**

The concept of missing values is important to understand in order to successfully [manage](http://www.statisticssolutions.com/academic-solutions/resources/dissertation-resources/data-entry-and-management/multiple-imputation-for-missing-data/) data.  If the missing values are not handled properly by the researcher, then he/she may end up drawing an inaccurate inference about the data.  Due to improper handling, the result obtained by the researcher will differ from ones where the missing values are present. Item non-response occurs when the respondent does not respond to certain questions due to stress, fatigue or lack of knowledge. The respondent may not respond because some questions are sensitive. This lack of answers would be considered missing values. 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.

|  |  |  |
| --- | --- | --- |
| 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:**

In [statistics](https://en.wikipedia.org/wiki/Statistics), an outlier is a [data point](https://en.wikipedia.org/wiki/Data_point) that differs significantly from other observations. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the [data set](https://en.wikipedia.org/wiki/Data_set). An outlier can cause serious problems in statistical analyses. Outliers can occur by chance in any distribution, but they often indicate either [measurement error](https://en.wikipedia.org/wiki/Measurement_error) or that the population has a [heavy-tailed distribution](https://en.wikipedia.org/wiki/Heavy-tailed_distribution). In the former case one wishes to discard them or use statistics that are [robust](https://en.wikipedia.org/wiki/Robust_statistics) to outliers, while in the latter case they indicate that the distribution has high [skewness](https://en.wikipedia.org/wiki/Skewness" \o "Skewness) and that one should be very cautious in using tools or intuitions that assume a [normal distribution](https://en.wikipedia.org/wiki/Normal_distribution). A frequent cause of outliers is a mixture of two distributions, which may be two distinct sub-populations, or may indicate 'correct trial' versus 'measurement error; this is modeled by a [mixture model](https://en.wikipedia.org/wiki/Mixture_model).

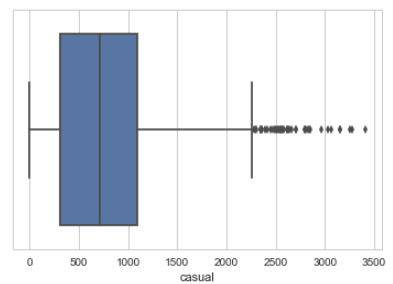
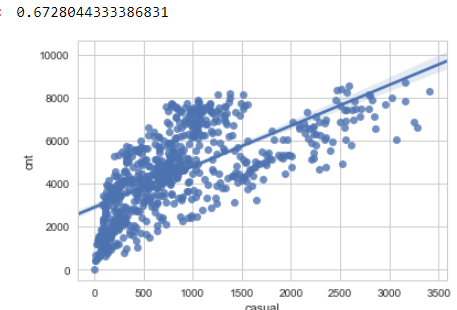
The Other steps of Preprocessing Technique is Outliers analysis, an outlier is an observation point that is distant from other observations. Outliers in data can distort predictions and affect the accuracy, if you don’t detect and handle them appropriately especially in regression models.

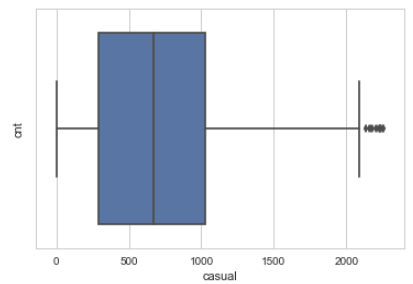
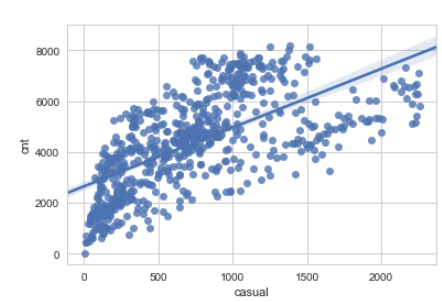
As we are observed in fig 2.2 the data is skewed so, there is chance of outlier in independent variable ‘casual’, one of the best method to detect outliers is Boxplot.

Fig 2.4 shows presence of Outliers in variable ‘casual’ and relationship between ‘casual’ and ‘cnt’ before removing Outliers.

Fig 2.5 shows boxplot of ‘casual’ after removing outliers and relationship between ‘casual’ and ‘cnt’ after removing outliers.

Figure 2.4‘casual’ Baxoplot and relation between ‘cnt’ and’ casual’

Since there is significant difference between Pearson coefficient correlation between before and after outlier detection for ‘casual’ and ‘cnt’ and losing nearly 40 observation so, we are not going to treat the outliers.

**2.2.3 Features Selections:**

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning) and [statistics](https://en.wikipedia.org/wiki/Statistics), **feature selection**, also known as **variable selection**, **attribute selection** or **variable subset selection**, is the process of selecting a subset of relevant [features](https://en.wikipedia.org/wiki/Feature_(machine_learning)) (variables, predictors) for use in model construction. Feature selection techniques are used for four reasons:

* Simplification of models to make them easier to interpret by researchers/users,
* shorter training times,
* to avoid the [curse of dimensionality](https://en.wikipedia.org/wiki/Curse_of_dimensionality),
* enhanced generalization by reducing [overfitting](https://en.wikipedia.org/wiki/Overfitting) (formally, reduction of [variance](https://en.wikipedia.org/wiki/Bias-variance_tradeoff))

The central premise when using a feature selection technique is that the data contains some features that are either *redundant* or *irrelevant*, and can thus be removed without incurring much loss of information. *Redundant* and *irrelevant* are two distinct notions, since one relevant feature may be redundant in the presence of another relevant feature with which it is strongly correlated.

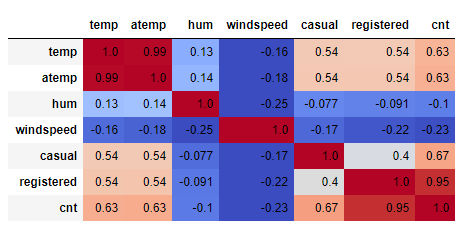
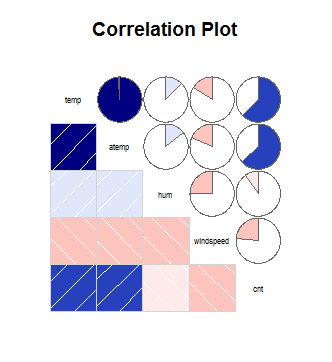
Machine learning works on a simple rule – if you put garbage in, you will only get garbage to come out. By garbage here, I mean noise in data.

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!”.

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 figure illustrates that relationship between all numeric variables using Corrgram plot in both R and Python.



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

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

**2.2.4 Features Scaling:**

Feature scaling is a method used to standardize the range of independent variables or features of data. In [data processing](https://en.wikipedia.org/wiki/Data_processing), it is also known as data normalization and is generally performed during the data preprocessing step.

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.

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

**Table Normalization of ‘casual’ and ‘registered**

**2.3 Dimensionality Reduction for numeric variables:**

Above figure is showing, there is strong relationship between independent variables ‘temp’ and ‘atemp’ so considering any one features 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.

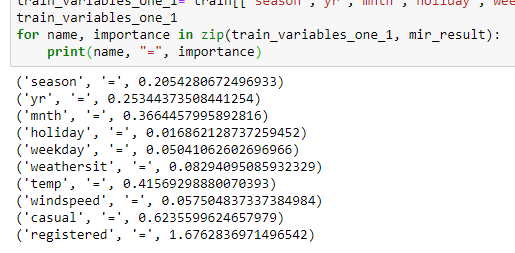
Subsetting two independent features ‘atemp’ and ‘hum’ from actual dataset.

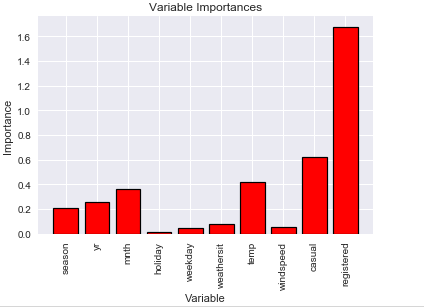
**2.3.1 Dimensional Reduction using Random Forest Variable Importance:**

There are several methods to check the relation between categorical variable , but here using Random Forest to get the importance of variables .

The below figure shows that variables ‘season’ ‘windspeed’ , ‘weekday ‘ ,‘weathersit’ and ‘holiday’ are less importance in predict the ‘cnt’ of Rental Bikes .

So these variable are removing while performaning Random Forest Model.





**Figure Variable Imporatnce Graph**

**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

* Nominal
* Ordinal
* Interval
* Ratio

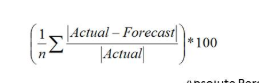
In our case dependent variable is interval so, the predictive analysis that we can perform is **Regression** Analysis. We will start our model building from Decision Tree

**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.



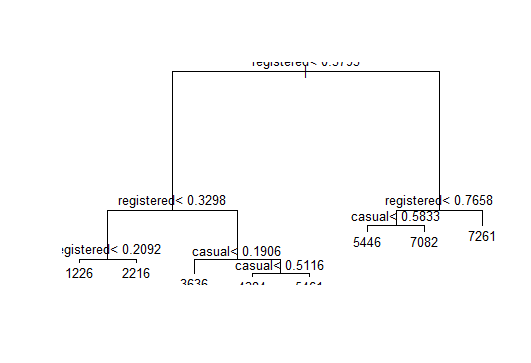
1. **RMSE :**(Root Mean Square Error) is a frequently used measure of the difference between values predicted by a model and the values actually observed from the environment that is being modelled.



**3.2 Decision Tree**

A **decision tree** is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. A decision tree is a flowchart-like structure in which each internal node represents a “test” on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules. Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods. Tree based methods empower predictive models with high accuracy, stability and ease of interpretation. Unlike linear models, they map non-linear relationships quite well. They are adaptable at solving any kind of problem at hand (classification or regression). Decision Tree algorithms are referred to as **CART** **(Classification and Regression Trees)**. 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.

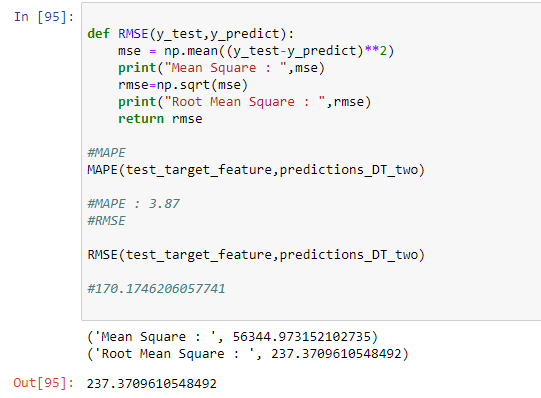
**Graphical representation of Decision Tree:**



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**

In the below figure Evaluation of Decision Tree using MAPE and RMSE



In Figure Model Accuracy is 100- 3.8 = 96.2 which is nearly 96.2% it is quite good but RMSE is 237 which is very high so it’s clearly stating that our Decision Tree Model is Over fitted and it working well for training data but won’t predict good for new set of data. To overcome this over fit we have to tune the model using Random Forest.

**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 operates 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" \o "Overfitting) to their [training set](https://en.wikipedia.org/wiki/Test_set). The first algorithm for random decision forests was created by [Tin Kam Ho](https://en.wikipedia.org/wiki/Tin_Kam_Ho)using the [random subspace method](https://en.wikipedia.org/wiki/Random_subspace_method), which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg. An extension of the algorithm was developed by [Leo Breiman](https://en.wikipedia.org/wiki/Leo_Breiman) and Adele Cutler, who registered "Random Forests" as a [trademark](https://en.wikipedia.org/wiki/Trademark) (as of 2019, owned by [Minitab, Inc.](https://en.wikipedia.org/wiki/Minitab)). The extension combines Breiman's "[bagging](https://en.wikipedia.org/wiki/Bootstrap_aggregating)" idea and random selection of features, introduced first by Ho and later independently by Amit and [Geman](https://en.wikipedia.org/wiki/Donald_Geman" \o "Donald Geman) in order to construct a collection of decision trees with controlled variance. 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" \o "Overfitting) to their [training set](https://en.wikipedia.org/wiki/Test_set).

Random forest functions in below way

1. Draws a bootstrap sample from training data.
2. For each sample grow a decision tree and at each node of the tree
3. Ramdomly draws a subset of mtry variable and p total of features that are available
4. Picks the best variable and best split from the subset of mtry variable
5. Continues until the tree is fully grown.

As we saw in section 3.2 Decision tree is overfitting and its accuracy MAPE and RMSE is also poor in order to improve the performance of the model developing model using Random Forest.



Figure Random Forest Implementation

Mtry : Number of variables to split at each node i.e. 7.

Node size: size of each node is 10

Our Random Forest model is looking quite good where it utilized maximum variables to predict the count values.

**3.3.1 Evaluation of Random Forest:**

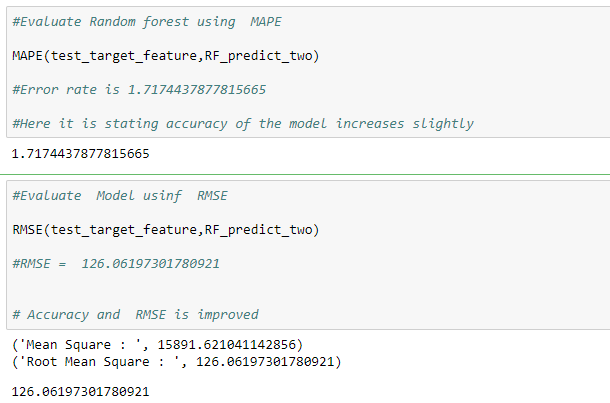


Figure shows Random Forest model performs dramatically better than Decision tree on both training and test data and well also improve the Accuracy (MAPE = 1.71) and decrease the RMSE (126) of the model which is quite impressive.

Using Linear Regression we will predict the ‘cnt’ values and compare with Random Forest.

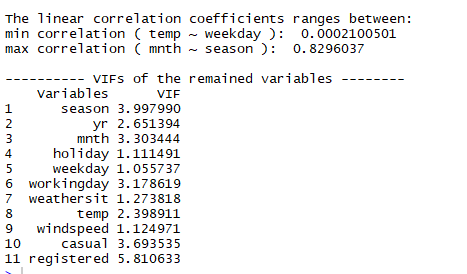
**3.4 Linear Regression**

[Multiple linear regressions](http://www.statisticssolutions.com/academic-solutions/membership-resources/member-profile/data-analysis-plan-templates/data-analysis-plan-multiple-linear-regression/) are 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 is less.

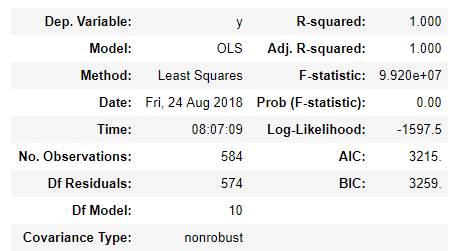
**Figure Multi collinearity between Independent variables**



In the above figure it is showing there is strong correlation between two independent variable ‘mnth’ and ‘season’ so , it is enough to consider any one variable.

**Figure -Multiple Linear Regression Model**

.



Here:

Residual standard error: 3.231e-12 on 576 degrees of freedom

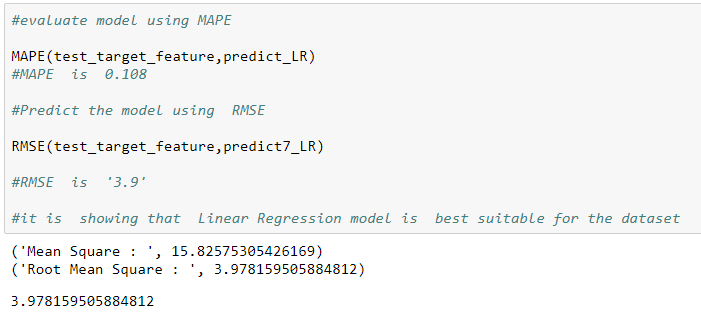
Multiple R-squared: 1

Adjusted R-squared: 1

Here residual Standard error is quite less so the distance between predicted values f(x[I,]) and actual values f(x) are very less so this model is predicted almost accurate values.

And Multiple R-Square value is 1 so, we can explain about 100 % of the data using our multiple linear regression models. This is very impressive.

**3.4.1 Evaluation of Linear regression Model:**



**Figure- Evaluation of Regression Model**

From above figure it is clearly showing that Model Accuracy is 99.9 % and RMSE is nearly equal to 3.9.

**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.

R-Code

#Remove all object stored

rm(list=ls())

#Set Working directory

setwd("D:/ Edwisor Assignment/Edwisor Project/Project")

#Get Working directory

getwd()

#Install Library

x = c("ggplot2", "corrgram", "DMwR", "caret", "randomForest", "unbalanced", "C50", "dummies", "e1071", "Information",

"MASS", "rpart", "gbm", "ROSE", 'sampling', 'DataCombine', 'inTrees')

#install.packages(x)

lapply(x, require, character.only = TRUE)

for ( i in x ) {

print(i)

library("ggplot2")

}

install.packages(c("dplyr","plyr","reshape","ggplot2","data.table","corrgram"))

install.packages("GGally","DataCombine")

# Install libraries

library("dplyr")

library("plyr")

library("ggplot2")

library("data.table")

library("GGally")

# Load Day.csv file

df=read.csv("D:/Edwisor Assignment/Edwisor Project/day.csv", header=T)

#View Day.csv file

View(df)

#summary of data

summary(df)

#View structure of the given data

str(df)

#Create a function uniquevalue for the distribution of numeric values

uniquevalue\_numeric= function(num\_x)

{

ggplot(df)+geom\_histogram(aes(x=num\_x,y=..density..),

fill="grey")+

geom\_density(aes(x=num\_x,y=..density..))

}

#Visualize the distribution of target variable 'cnt'

uniquevalue\_numeric(df$cnt)

#Visualize the distribution of independent variable 'temp'

uniquevalue\_numeric(df$temp)

#Visualize the distribution of independent variable 'atemp'

uniquevalue\_numeric(df$atemp)

#Visualize the distribution of independent variable 'hum'

uniquevalue\_numeric(df$hum)

#Visualize the distribution of independent variable 'windspeed'

uniquevalue\_numeric(df$windspeed)

#Visualize the distribution of independent variable 'casual'

uniquevalue\_numeric(df$casual)

#Visualize the distribution of independent variable 'registered'

uniquevalue\_numeric(df$registered)

#Visulaize the categorical variable 'mnth' with target variable 'cnt'

ggplot(df, aes(x=as.factor(mnth), y=cnt),fill="grey") +

stat\_summary(fun.y="mean", geom="bar")

ggplot(df)+

geom\_histogram(aes(x=cnt,y=..density..),

fill= "grey")+

geom\_density(aes(x=cnt,y=..density..))

#Visulaize the categorical variable 'holiday'

ggplot(df) +

geom\_bar(aes(x=holiday),fill="grey")

#Now Visulaize the categorical variable 'weekday'

ggplot(df) +

geom\_bar(aes(x=weekday),fill="grey")

#Visulaize the categorical variable 'weathersit'

ggplot(df) +

geom\_bar(aes(x=weathersit),fill="grey")

# Letsee the dependencies or relationship between 'temp' and 'atemp'

ggplot(df, aes(x= temp,y=atemp)) +

geom\_point()+

geom\_smooth()

#\* Graph represent both are highly dependent or highly related to each other

# Letsee the dependencies or relationship between 'temp' and 'hum'

ggplot(df, aes(x= temp,y=hum)) +

geom\_point()+

geom\_smooth()

#graph shows humidity is increas till temprature is 0.7 and it is decreasing

#gradually

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

ggplot(df,aes(x=temp,y=windspeed))+

geom\_point()+

geom\_smooth()

#check the relationship between all numeric variable using pair plot

ggpairs(df[,c('atemp','temp','hum','windspeed','cnt')])

#check relationship between season and holiday

rel\_mnth\_holi= table(df$season,df$holiday)

rel\_mnth\_holi

barplot(rel\_mnth\_holi)

#check relationship between season and weekday

rels\_cats\_2 <- table(df$season,df$weekday)

barplot(rels\_cats\_2)

#check relationship between season and weathersit

rels\_cats\_3 <- table(df$weathersit,df$season)

rels\_cats\_3

prop.table(rels\_cats\_3,2)

barplot(rels\_cats\_3)

#check relationship between holiday and weathersit

rels\_cats\_4 <- table(df$weathersit,df$holiday)

rels\_cats\_4

barplot(df$weathersit,df$holiday)

#to check in proportion

prop.table(rels\_cats\_4,2)

barplot(rels\_cats\_4)

#--------------------------------Missing Value Analysis----------------

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

missing\_val$columns=row.names(missing\_val)

missing\_val

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

missing\_val$Missing\_percentage = (missing\_val$Missing\_percentage/nrow(df)) \* 100

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

row.names(missing\_val) = NULL

missing\_val = missing\_val[,c(2,1)]

# there is no missing value in our data set---

#-----------------------Outlier Analysis--------------------------------

# detect outliers in 'actual' , 'registered' and 'cnt' variables

ggplot(data = df, aes(x = "", y = casual)) +

geom\_boxplot()

# boxplot for Registered variable

ggplot(data = df, aes(x = "", y = registered)) +

geom\_boxplot()

# boxplot for cnt variable

ggplot(data = df, aes(x = "", y = cnt)) +

geom\_boxplot()

# it is showing that there is no outliers in cnt variable

#--------------------Handling Outlier-----------------

# analyse relationship between causal and cnt variables before outlier treatment

ggplot(df, aes(x= casual,y=cnt)) +

geom\_point()+

geom\_smooth()

df\_out = df

#Remove outliers using boxplot method

val = df\_out$casual[df\_out$casual %in% boxplot.stats(df\_out$casual)$out]

df\_out = df\_out[which(!df\_out$casual %in% val),]

# boxplot for casual variable

ggplot(data = df\_out, aes(x = "", y = casual)) +

geom\_boxplot()

# verify the relationship after outliers

ggplot(df\_out, aes(x= casual,y=cnt)) +

geom\_point()+

geom\_smooth()

cor(df\_out$casual,df\_out$cnt)

cor(df\_out$casual,df\_out$cnt)

#------------------------------Feature Selection or dimension reduction-----------------

library(corrgram)

# verify correleation between Numeric variable

corrgram(df[,c('temp','atemp','hum','windspeed','cnt')], order = F,

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

# dimensional reduction

df\_features = subset(df,select=-c(atemp,hum))

#----------------------------Normality check------------------------

#Normalisation

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

for(i in cnames){

print(i)

df\_features[,i] = (df\_features[,i] - min(df\_features[,i]))/

(max(df\_features[,i] - min(df\_features[,i])))

}

df$casual

#-----------------------------Model Development----------------------

library(DataCombine)

feature\_train\_columns=c("season" ,"yr" ,"mnth","holiday","weekday","workingday","weathersit","temp","windspeed","casual","registered","cnt")

#Divide data into train and test

library(caret)

install.packages("caret")

library(rpart)

library(MASS)

train.index = createDataPartition(df\_features$cnt, p = .80, list = FALSE)

train = df\_features[ train.index,]

test = df\_features[-train.index,]

train\_feature = train[,c("season" ,"yr" ,"mnth","holiday","weekday","workingday","weathersit","temp","windspeed","casual","registered","cnt")]

train\_feature

test\_features = test[,c("season" ,"yr" ,"mnth","holiday","weekday","workingday","weathersit","temp","windspeed","casual","registered","cnt")]

test\_features

#------------------------------------develope Decision tree model---------------------

#rpart for regression

fit = rpart(cnt ~ ., data = train\_feature, method = "anova")

#Predict for new test cases

predictions\_DT = predict(fit, test\_features[,-12])

print(fit)

# plotting decision tree

par(cex= 0.8)

plot(fit)

text(fit)

#------------------------------ Evaluate Decision Tree-----------------------------

#calculate MAPE

MAPE = function(y, yhat){

mean(abs((y - yhat)/y))\*100

}

MAPE(test\_features[,12], predictions\_DT)

#Error Rate: 11.68

#Accuracy: 88.32

#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\_features[,12], predictions\_DT)

#RMSE->490.8318

#-----------------------------Random Forest----------------------

install.packages("randomForest")

library(party)

library(randomForest)

Rental\_rf=randomForest(cnt ~ . , data = train\_feature)

Rental\_rf

plot(Rental\_rf)

#--------------------------------Evaluate Random Forest--------------------

#Predict for new test cases

predictions\_DT\_two = predict(Rental\_rf, test\_features[,-12])

#calculate MAPE

MAPE = function(y, yhat){

mean(abs((y - yhat)/y))\*100

}

MAPE(test\_features[,12], predictions\_DT\_two)

#Error Rate:4.19

#Accuracy: 95.81

RMSE(test\_features[,12], predictions\_DT\_two)

#RMSE=190.2036

#------------------------Parameter Tuning for random forest---------------------

Rental\_rf\_2=randomForest(cnt ~ . , data = train\_feature,mtry =7,ntree=500 ,nodesize =10 ,importance =TRUE)

Rental\_rf\_2

#Predict for new test cases

predictions\_RF\_two = predict(Rental\_rf\_2, test\_features[,-12])

#calculate MAPE

MAPE = function(y, yhat){

mean(abs((y - yhat)/y))\*100

}

MAPE(test\_features[,12], predictions\_RF\_two)

#Error Rate: 2.008418

#Accuracy: 97.99

RMSE(test\_features[,12], predictions\_RF\_two)

#RMSE = 92.70644

# check Variable Importance

varimp <- importance(Rental\_rf\_2)

varimp

# sort variable

sort\_var <- names(sort(varimp[,1],decreasing =T))

sort\_var

# draw varimp plot

varImpPlot(Rental\_rf\_2,type = 2)

#------------------------------Tuning Random Forest Dimensional Reduction--------------

#remove four variables which is contributing less

train\_feature\_two = train[,c("yr" ,"mnth","weekday","workingday","temp","casual","registered","cnt")]

test\_features\_two = test[,c("yr" ,"mnth","weekday","workingday","temp","casual","registered","cnt")]

# Develop Random Forest Model

Rental\_rf\_3=randomForest(cnt ~ . , data = train\_feature\_two,mtry =7,ntree=500 ,nodesize =10 ,importance =TRUE)

Rental\_rf\_3

#Predict for new test cases

predictions\_RF\_three = predict(Rental\_rf\_3, test\_features\_two[,-8])

#calculate MAPE

MAPE = function(y, yhat){

mean(abs((y - yhat)/y))\*100

}

MAPE(test\_features\_two[,8], predictions\_RF\_three)

#Error Rate: 1.650

#Accuracy: 98.35

RMSE(test\_features\_two[,8], predictions\_RF\_three)

#RMSE = 74.13

#---------------------------------Develope Linear Regression Model------------------

install.packages('usdm')

library(usdm)

vif(train\_feature[,-12])

vifcor(train\_feature[,-12], th = 0.9)

# Correleation between two variables is 'season' and 'mnth' is 0.82 so, removing one variable from the model

train\_feature\_three = train[,c("yr" ,"mnth","holiday","weekday","workingday","weathersit","temp","windspeed","casual","registered","cnt")]

test\_features\_three = test[,c("yr" ,"mnth","holiday","weekday","workingday","weathersit","temp","windspeed","casual","registered","cnt")]

# develop Linear Regression model

#run regression model

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

#Summary of the model

summary(lm\_model)

# observe the residuals and coefficients of the linear regression model

# Predict the Test data

#Predict

predictions\_LR = predict(lm\_model, test\_features\_three[,-11])

# Evaluate Linear Regression Model

#calculate MAPE

MAPE = function(y, yhat){

mean(abs((y - yhat)/y))\*100

}

MAPE(test\_features\_three[,11], predictions\_LR)

#Error Rate: 5.958383e-15

#Accuracy: 94.041

RMSE(test\_features\_three[,11], predictions\_LR)

#RMSE = 3.474482e-13

# COnclusion For this Dataset Linear Regression is Accuracy is '94.41'

# and RMSE = 6.678262e-13