# Index:-

**Introduction to the data**

1) Problem Statement 2

2) Data 2

**Methods applied in pre-processing**

1) Outliers 3

2) Missing Values Analysis 6

3) Feature Selection and Feature Scaling 6

**Modeling**

1) Model Selection 9

2) Linear Regression 9

**Model Evaluation**

1) MAPE 12

1. **INTRODUCTION TO THE DATA**

* **Problem Statement**

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

* **Data**

In this project our task is to build a regression model for prediction of bike rent count based on environmental and seasonal settings. We get 16 variables and 731 observations in this project. Given below few rows of the data.



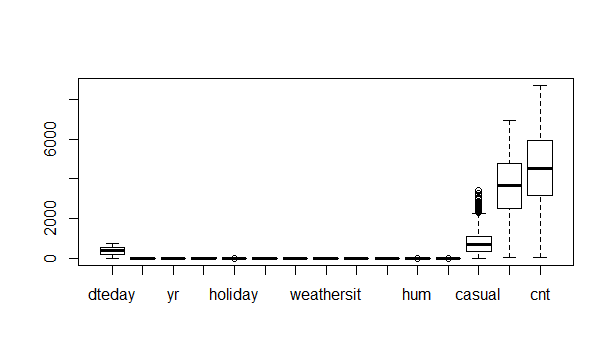
1. **METHODS APPLIED IN PRE-PROCESSING**

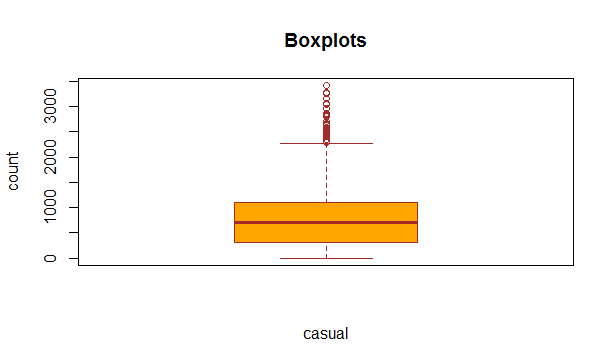
I have used several methods for pre-processing the data which are very important as machine takes proper data. So, we have to check make sure that our data should not have any missing values, we have to take care of outliers which are present in the data, and we also have to select variables which are important for model development. I have removed the instant column from the data because it contains index numbers only.

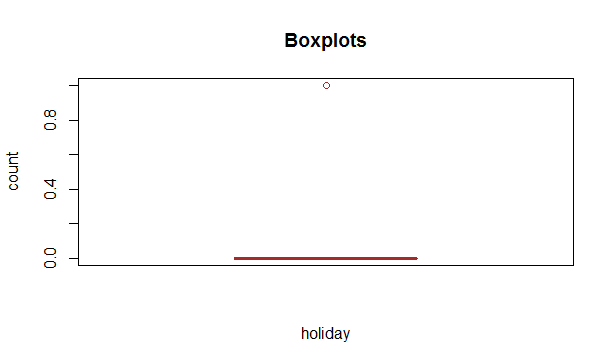
* **Outliers**

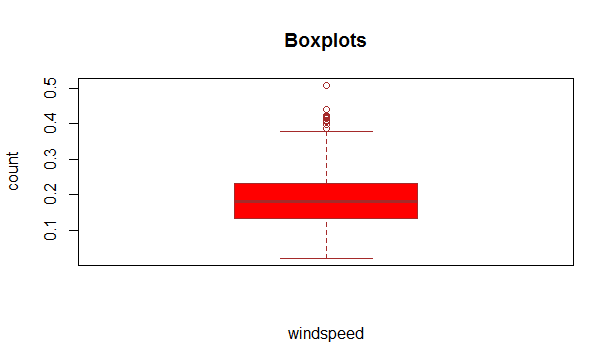
Except “dteday” variable all are numeric variables so I have checked outliers for all the variables. There are outliers found in “4” variables i.e., “holiday”,” hum”, “windspeed” and “casual”, but, in variable holiday there are only two distinct values “0” and “1” and “1” appears only once and rest of the time it appears “0”, and “hum” variable also don’t have the outliers so it is outlier for machine but actually it is not. “windspeed” and “casual” variables have real outliers. So there is a need to remove these outliers.

If there is not much difference between max-min value then that is not an outlier. Below, are some visualizations of boxplot :









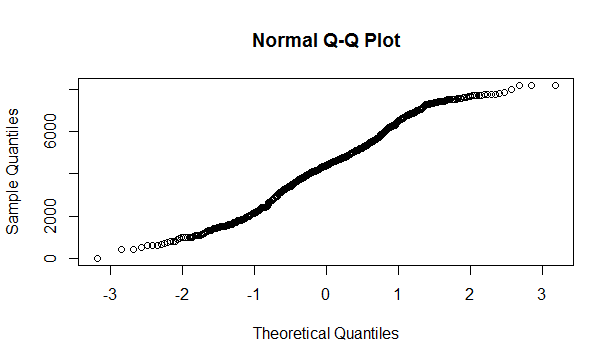
These are some pictures of outliers; we can see that there is no outlier in any variable except variables “windspeed” and “casual” having some outliers.

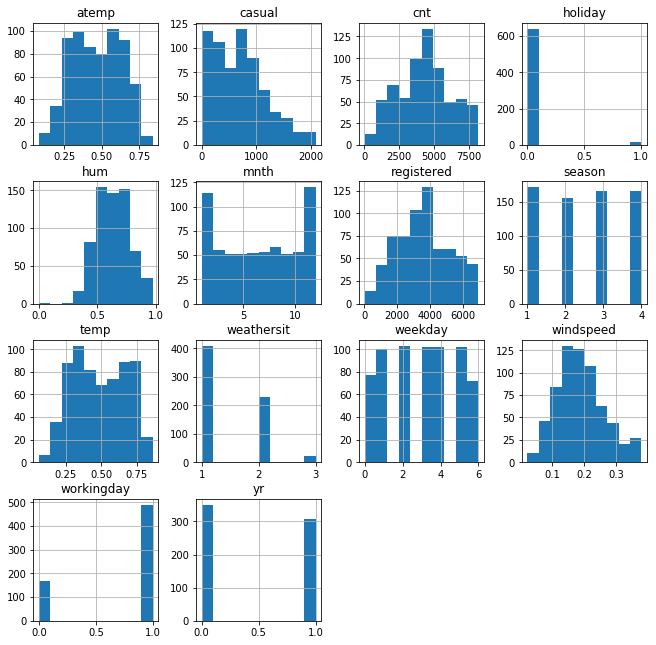
After using certain codes we have removed outliers, initially we have 731 observations but now we have 674 observations.

* **Missing value analysis**

I have checked the whole data by using some functions like in R “sum(is.na())” similarly in python “.isnull().sum() “. In this data we don’t have any missing values.

* **Feature selection and feature scaling**

For **feature scaling**, initially we have checked is the data normally distributed or not, we see that, the data is not perfectly normally distributed, but somewhat a straight line can be seen in the fig below. We can see in the below Q-Q plot and histogram:

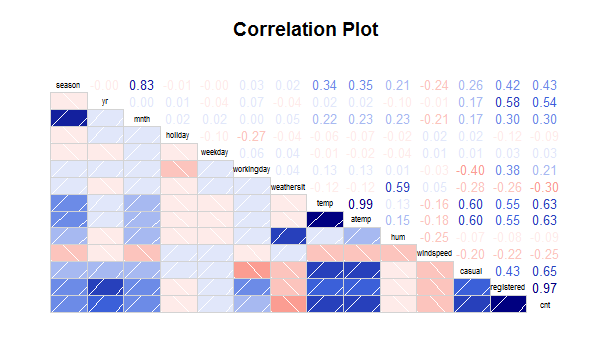


Since our data is not perfectly normally distributed so I used min-max normalization method for feature scaling. The formula is:

(X - Min(X))/ (Max(X) - Min(X))

Where “X” is a general data point and “Min(X)” is minimum value of the variable and “max(X)” is the maximum value of the variable.

After feature scaling, we have to select the important variables for modeling our data, so, now I have done **feature selection** by using “corrgram” library in R and “matplotlib “ in python. Have a look of plot and see the correlation between the different variables in data.



We can see that “season” and “mnth” variables are highly correlated similarly “temp” and “atemp” variables gives almost same information so we have to remove one of them and we can also see that variable “holiday”, “weekday” and “hum” don’t give much information regarding out target variable “cnt”.

So we will remove the variables which are not giving much information (which are less correlated with target variables) and also remove those independent variables which are highly correlated among themselves.

**III. MODELING**

* **Model Selection**

Our data contains target variable as continuous variable or we can say that, there are cardinals in the dependent variable, so we will go for regression model and choose **linear regression** as our model because we have almost all numeric variables.

* **Linear Regression**

Before putting the train data in the model let us first calculate the correlation between the independent variables:-

|  |
| --- |
| vifcor(n\_data[,-9],th=0.8)  No variable from the 8 input variables has collinearity problem.  The linear correlation coefficients ranges between:  min correlation ( yr ~ season ): 0.004034129  max correlation ( casual ~ atemp ): 0.5940725  ---------- VIFs of the remained variables --------  Variables VIF  1 season 1.585412  2 yr 2.639544  3 workingday 2.888214  4 weathersit 1.294908  5 atemp 2.522389  6 windspeed 1.088579  7 casual 3.523729  8 registered 6.224966 |
|  |
| |  | | --- | |  | |

So, there is no problem in collinearity in the dataset.

After putting the train data in the model, the summary of the data.

**Model summary in R:**

Residuals:

Min 1Q Median 3Q Max

-7.617e-11 -3.700e-14 2.780e-13 5.960e-13 1.158e-11

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.200e+01 8.288e-13 2.654e+13 < 2e-16 \*\*\*

season -2.326e-12 5.966e-13 -3.899e+00 0.000109 \*\*\*

yr -5.213e-12 5.710e-13 -9.130e+00 < 2e-16 \*\*\*

workingday -3.114e-12 6.745e-13 -4.616e+00 4.91e-06 \*\*\*

weathersit 1.277e-12 7.296e-13 1.750e+00 0.080698 .

atemp 1.256e-12 1.320e-12 9.520e-01 0.341633

windspeed 7.695e-13 9.221e-13 8.350e-01 0.404371

casual 2.256e+03 1.441e-12 1.565e+15 < 2e-16 \*\*\*

registered 6.926e+03 1.893e-12 3.659e+15 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.162e-12 on 530 degrees of freedom

Multiple R-squared: 1, Adjusted R-squared: 1

F-statistic: 1.317e+31 on 8 and 530 DF, p-value: < 2.2e-16

The values we get after applying the linear regression are very impressive as we can see that the value of R-squared and multiple R-squared is 1, and values of F-statistics and p-value<0.05 are also impressive and, we can reject the null hypothesis that target variable is independent of the predictor variables.

**Model summary in Python:**

model.summary()

|  |  |  |  |
| --- | --- | --- | --- |
| **Dep. Variable:** | y | **R-squared (uncentered):** | 1.000 |
| **Model:** | OLS | **Adj. R-squared (uncentered):** | 1.000 |
| **Method:** | Least Squares | **F-statistic:** | 1.629e+32 |
| **Date:** | Thu, 07 Nov 2019 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 22:34:03 | **Log-Likelihood:** | 18754. |
| **No. Observations:** | 549 | **AIC:** | -3.749e+04 |
| **Df Residuals:** | 540 | **BIC:** | -3.745e+04 |
| **Df Model:** | 9 |  |  |
| **Covariance Type:** | nonrobust |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **coef** | **std err** | **t** | **P>|t|** | **[0.025** | **0.975]** |
| **x1** | -5.655e-16 | 4.95e-17 | -11.422 | 0.000 | -6.63e-16 | -4.68e-16 |
| **x2** | 4.545e-16 | 4.67e-17 | 9.732 | 0.000 | 3.63e-16 | 5.46e-16 |
| **x3** | -4.163e-17 | 9.75e-17 | -0.427 | 0.670 | -2.33e-16 | 1.5e-16 |
| **x4** | 2.255e-16 | 4.63e-17 | 4.866 | 0.000 | 1.34e-16 | 3.17e-16 |
| **x5** | 2.602e-16 | 5.91e-17 | 4.399 | 0.000 | 1.44e-16 | 3.76e-16 |
| **x6** | 0.1539 | 8.6e-17 | 1.79e+15 | 0.000 | 0.154 | 0.154 |
| **x7** | 0.4724 | 7.47e-17 | 6.33e+15 | 0.000 | 0.472 | 0.472 |
| **x8** | -1.457e-16 | 1.06e-16 | -1.380 | 0.168 | -3.53e-16 | 6.17e-17 |
| **x9** | 1.943e-16 | 7.71e-17 | 2.520 | 0.012 | 4.28e-17 | 3.46e-16 |
| **x10** | 0.4440 | 6.23e-17 | 7.12e+15 | 0.000 | 0.444 | 0.444 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Omnibus:** | 13.041 | **Durbin-Watson:** | 1.804 |
| **Prob(Omnibus):** | 0.001 | **Jarque-Bera (JB):** | 7.171 |
| **Skew:** | -0.044 | **Prob(JB):** | 0.0277 |
| **Kurtosis:** | 2.447 | **Cond. No.** | 1.54e+16 |

**IV. Model Evaluation**

Now we have some more models from which we can develop models but we get very impressive values from linear Regression model so we will stick to it for some time. Now we will test our data and if we get bad accuracy from this model then we will try other model.

* **MAPE(Mean Absolute Percentage Error)**

Now we will predict the performance of our model using MAPE.

mape <- mean(abs((test[,9]-prediction\_lm ))/test[,9])

> mape

[1] 9.45714e-16