**Project: Bike Rental Daily Count Prediction**

Contents

[Problem Statement 2](#_Toc32601981)

[Data 2](#_Toc32601982)

[Outlier Analysis 3](#_Toc32601983)

[Correlation: 9](#_Toc32601984)

[Models Used 9](#_Toc32601985)

[Performance Management 9](#_Toc32601986)

[R and Python Code 11](#_Toc32601987)

[Conclusion 11](#_Toc32601988)

[Input and Output Sample 11](#_Toc32601989)

# Problem Statement

The objective of this Case is Predication of bike rental count on daily based on the

Environmental and seasonal settings.

# Data

Predictor Variables:

1. instant: Record index
2. dteday: Date
3. season: Season (1:springer, 2:summer, 3:fall, 4:winter)
4. yr: Year (0: 2011, 1:2012)
5. mnth: Month (1 to 12)
6. hr: Hour (0 to 23)
7. holiday: weather day is holiday or not (extracted fromHoliday Schedule)
8. weekday: Day of the week
9. workingday: If day is neither weekend nor holiday is 1, otherwise is 0.
10. weathersit: (extracted fromFreemeteo)

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

1. 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)

1. atemp: Normalized feeling temperature in Celsius. The values are derived via

(t-t\_min)/(t\_maxt\_

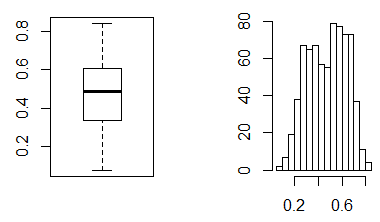
min), t\_min=-16, t\_max=+50 (only in hourly scale)

1. hum: Normalized humidity. The values are divided to 100 (max)
2. windspeed: Normalized wind speed. The values are divided to 67 (max)
3. casual: count of casual users
4. registered: count of registered users
5. cnt: count of total rental bikes including both casual and registered

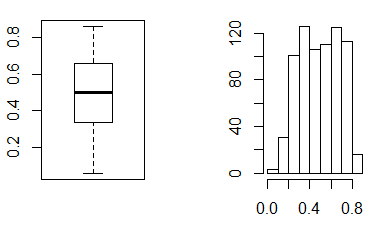
# Outlier Analysis

**Outliers** are data points that are far from other data points. In other words, they're unusual values in a dataset. **Outliers** are problematic for many statistical analyses because they can cause tests to either miss significant findings or distort real results. Below we have used box plot analysis to detect the outliers in the dependent variables.

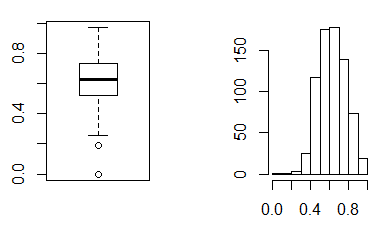
1. dataset$atemp (Normalized temperature):



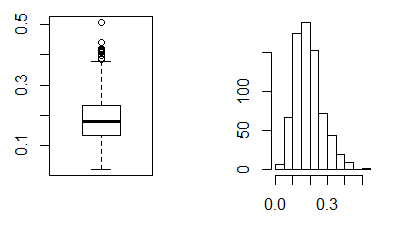
1. dataset$temp (Normalized feeling temperature) :



1. dataset$hum(Normalized Humility) :



1. dataset$windspeed :



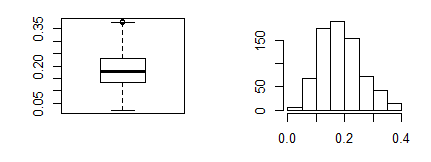
Outliers identified: 13

nPropotion (%) of outliers: 1.8

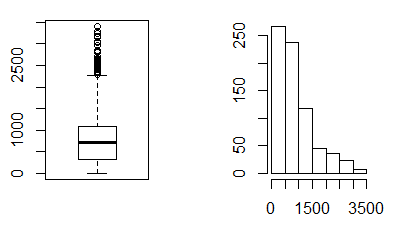
nMean of the outliers: 0.42

nMean without removing outliers: 0.19

nMean if we remove outliers: 0.19 n



1. dataset$casual(Casual Users):



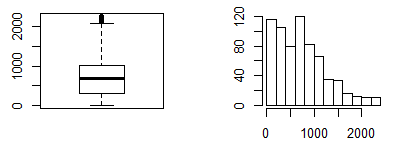
Outliers identified: 44

nPropotion (%) of outliers: 6.4

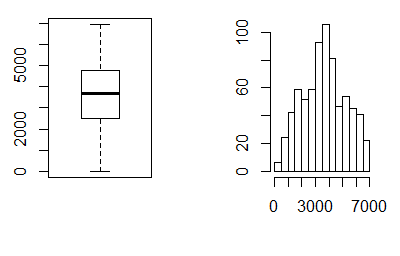
nMean of the outliers: 2661.95

nMean without removing outliers: 848.18

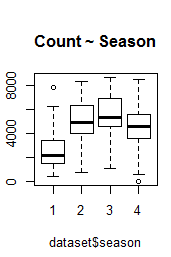
nMean if we remove outliers: 732.01 n



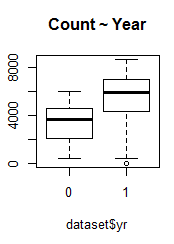
1. dataset$registered (registered users) :



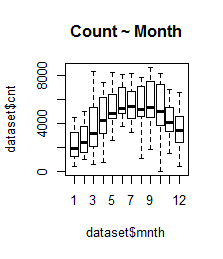
1. dataset$season :



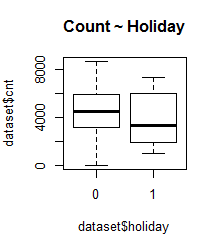
1. dataset$yr(Year) :



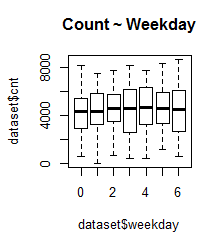
1. dataset$mnth :



1. dataset$holiday :



1. dataset$weekday :



Based on the above data we see that there are outliers in 2 of the data columns / variables:

1. Wind speed
2. Casual users

We remove these outliers and replace them with NA values. These NA values are then replaced using the KNN Imputation method. :

cnames = colnames(numeric\_data)

for(i in cnames){

val = outliers[,i][outliers[,i] %in% boxplot.stats(outliers[,i])$out]

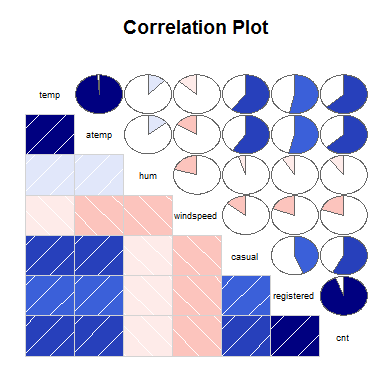
print(length(val))

outliers[,i][outliers[,i] %in% val] = NA

}

cleaned<- kNN(outliers, k = 5)

# Correlation:



# Models Used

1. Multiple Linear Regression
2. Decision Tree Regression
3. Random Forest Regression
4. SVM Regression

# Performance Management

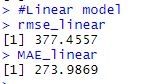
Metrics used for determining the performance management of the models –

1. RMSE - Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit.RMSE can be interpreted as the standard deviation of the unexplained variance, and has the useful property of being in the same units as the response variable.
2. Mean Absolute Error (MAE) is another loss function used for regression models. MAE is the sum of absolute differences between our target and predicted variables. So it measures the average magnitude of errors in a set of predictions, without considering their directions.

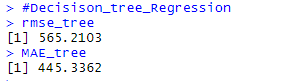
Lower values of RMSE and MAE indicate better fit.

Below are the RMSE and MAE of the 4 models tested to check for the best fit.

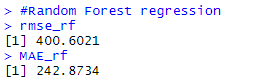
* Multiple Linear Regression:



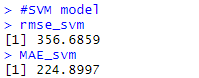
* Decision tree Regression :



* Random Forest regression ( ntree = 500 ) :



* SVM Regression :



# R and Python Code

Please find the R code and Python code in the respective R and Python files attached. Both the files contain the required code to source out all the inferences in the report

# Conclusion

Based on the above results it is clear that the RMSE and MAE is lowest for Support vector machine regression, therefore we can infer that SVM regression is the best fit for the prediction.

# Input and Output Sample

Please check the attached samples of Input and Output Data. In the Output Data, the last column (predicted\_count) is the required predicted count derived from the SVM regression model