Predicting Bike Rental Count

*Raunak Agrawal*

*4 October 2018*

**Contents**

[**1 Introduction** **2**](#_Toc111177)

[1.1 Problem Statement 2](#_Toc111178)

[1.2 Data 2](#_Toc111179)

[**2 Methodology** **4**](#_Toc111180)

[2.1 Pre Processing 4](#_Toc111181)

[2.1.1 Univariate Analysis 4](#_Toc111182)

2.1.2 Missing Value Analysis……………………………………………………………………………………………………………………………………..5

2.1.3 Outlier Analysis………………………………………………………………………………………………………………………………………………..7

[2.1.4 Feature Selection and Dimension Reduction………………………………………………………………………………………………….10](#_Toc111183)

[2.2 Modeling 12](#_Toc111184)

[2.2.1 Model Selection 12](#_Toc111185)

[2.2.2 Multiple Linear Regression 12](#_Toc111186)

[2.2.3 Regression Trees 14](#_Toc111187)

[2.2.4 Random Forest Regression 14](#_Toc111188)

[**3 Conclusion** **15**](#_Toc111189)

[3.1 Model Evaluation 15](#_Toc111190)

[3.1.1 Mean Absolute Percentage Error (MAPE) 15](#_Toc111191)

[3.2 Model Selection 16](#_Toc111193)

3.3 Sample Testing…………………………………………………………………………………………………………………………………………………..17

[**Appendix - R Code** 18](#_Toc111195)

[Complete R File 19](#_Toc111202)

**Chapter 1**

# Introduction

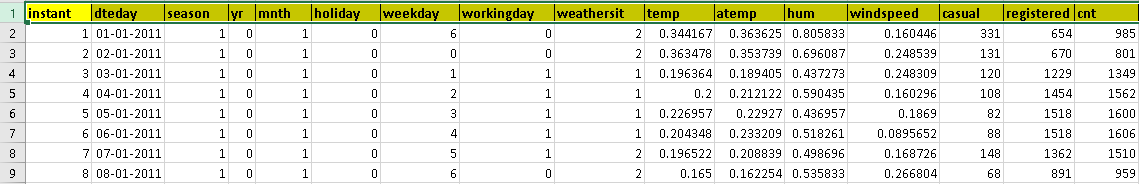
## 1.1 Problem Statement

The objective of this Case is to Predict the Bike Rental count based on the environmental and seasonal settings.

## 1.2 Data

Our task is to build regression models which will predict the count of Bike Rentals based on certain environmental conditions.  
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: 1-16)



**Chapter 2**

# Methodology

## 2.1 Pre Processing

Any predictive modeling requires that we look at the data before we start modeling. 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**.   
Before we explore the data let us look at the data attributes provided to us.

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)

**holiday**: weather day is holiday or not (extracted fromHoliday Schedule)

**weekday**: Day of the week

**workingday**: If day is neither weekend nor holiday is 1, otherwise is 0.

**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

**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  
  
  
Based on the above details we can now identify the **Predictor** (Input) and **Target** (output) variables.

Table 1.5: **Predictor Variables**

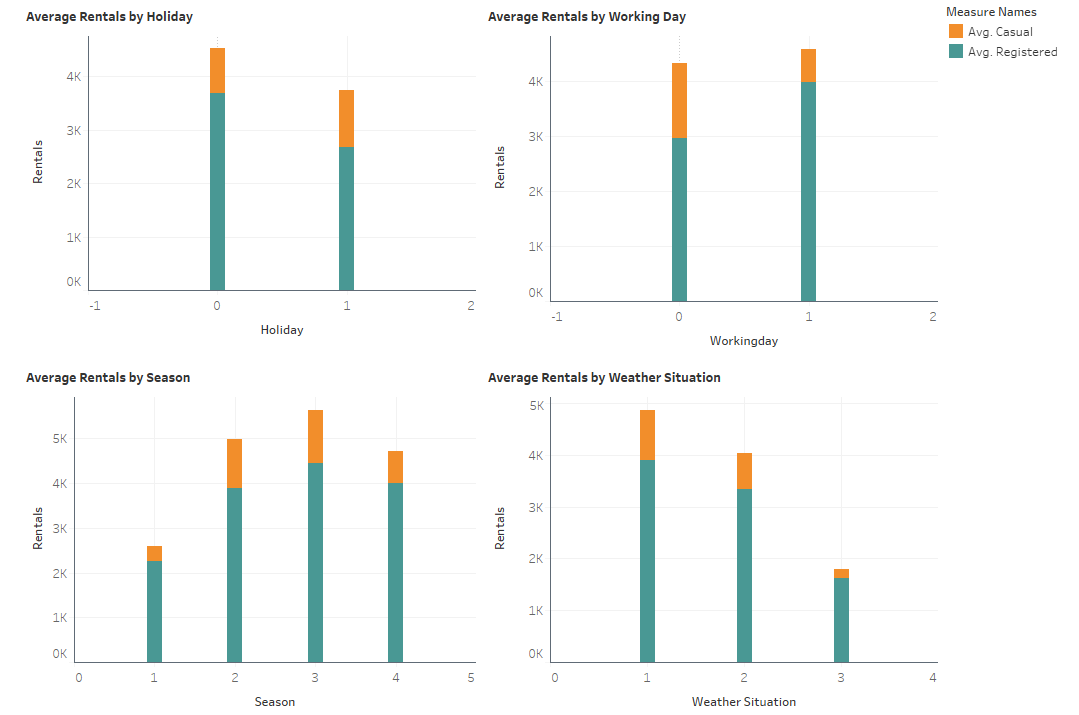
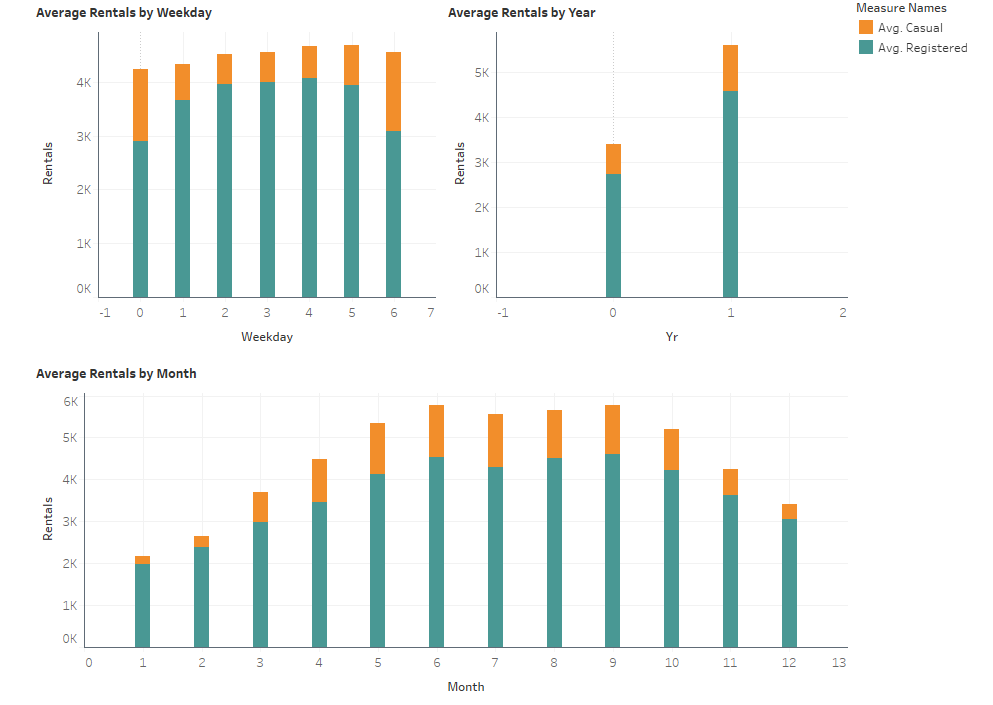
|  |  |
| --- | --- |
|  | Predictor |
|  | season |
|  | yr |
|  | mnth |
|  | holiday |
|  | weekday |
|  | workingday |
|  | weathersit |
|  | temp |
|  | atemp |
|  | hum |
|  | windspeed |

Table 1.6: **Target Variable**

|  |  |
| --- | --- |
|  | Target |
|  | Cnt |

The category of the variables can be defined as follows:  
  
**Categorical**: yr, mnth, holiday, weekday, workingday, weathersit  
**Continuous**: temp, atemp, hum, windspeed, cnt

We will use the above data attributes to draw inferences from the graphs of various predictor variables in the dataset.

To start this process we will first try and look at all the graphs of Average Rental Counts by each variable. The below graphs have been generated in Tableau.  
  
  
  
  
From the above graphs following observations can be made:

1. Average Count across the days of the week did not vary much. Number of Casual Users were higher on Day 0 and Day 6.
2. Average Count increased drastically on Year 1(2012) as compared to Year 0(2011)
3. Average Count was highest during Months 6-9.
4. Average Count was higher on a non holiday as compared to a holiday
5. Average Count was highest during Season 3 (Fall)
6. Average Count was highest at Weather Situation 1, and it decreased gradually further.

Now we look at some environmental parameters :



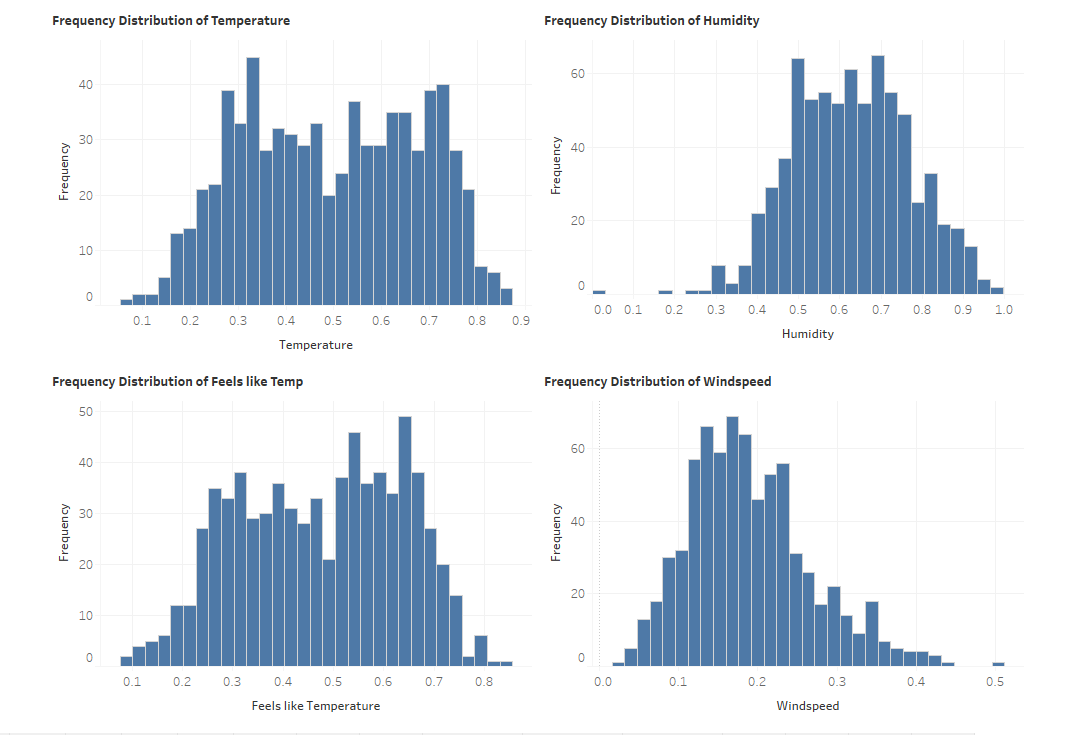
Below observations were made from the above graphs, which were generated in Tableau:

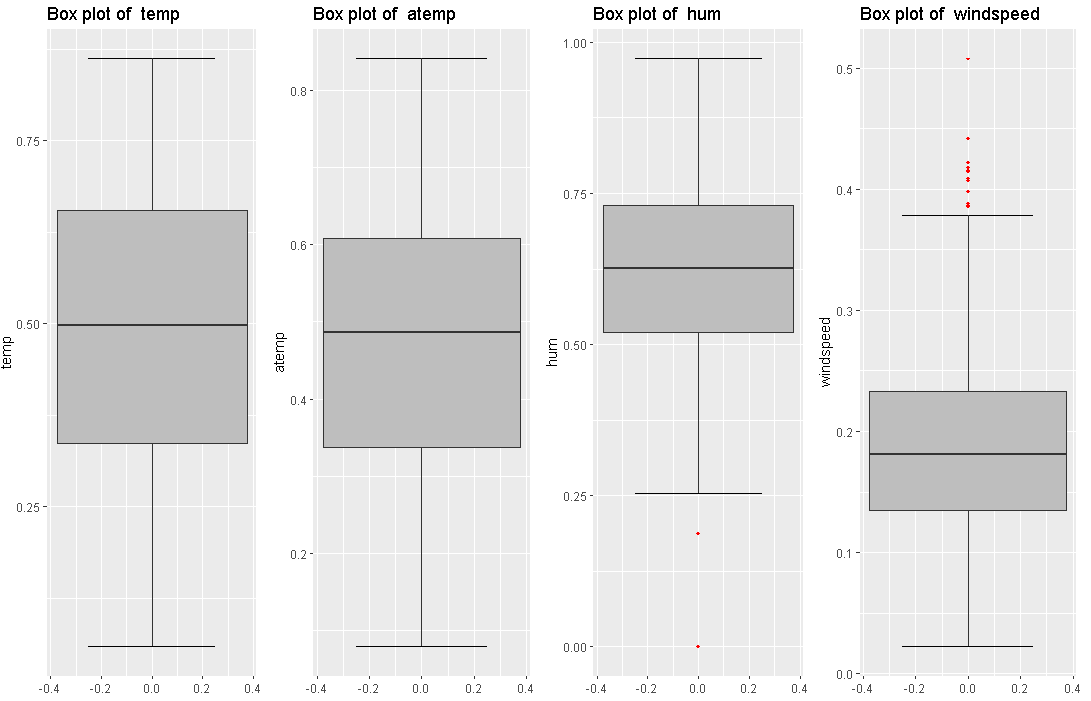
1. Average Count increased with increase in temperature and feels like temperature
2. Temperature and Feels like Temperature showed the same pattern hence they are highly correlated
3. Average Count decreased with increase in humidity

### 2.1.1 Univariate Analysis

At this stage, we explore the distribution of variables one by one. Method to perform uni-variate analysis will depend on whether the variable type is categorical or continuous. Let’s look at these methods and statistical measures for categorical and continuous variables individually:

Continuous Variables:- In case of continuous variables, we need to understand the central tendency and spread of the variable. These are measured using various statistical metrics visualization methods.   
  
  
We have drawn histogram plots and box plots of the continuous variables to understand their spread.



The above graphs indicate a normal distribution of the continuous variables in our data, with very few exceptions for ‘Hum’ and ‘Windspeed’  
  
We now create box plots for all the continuous variables to understand better:  


From the above boxplots it can be understood that the values of the continuous variables are normally distributed. Only hum, and windspeed have few outliers.

### 2.1.2 Missing Value Analysis

Missing data in the training data set can reduce the power / fit of a model or can lead to a biased model because we have not analysed the behavior and relationship with other variables correctly. It can lead to wrong prediction or classification.

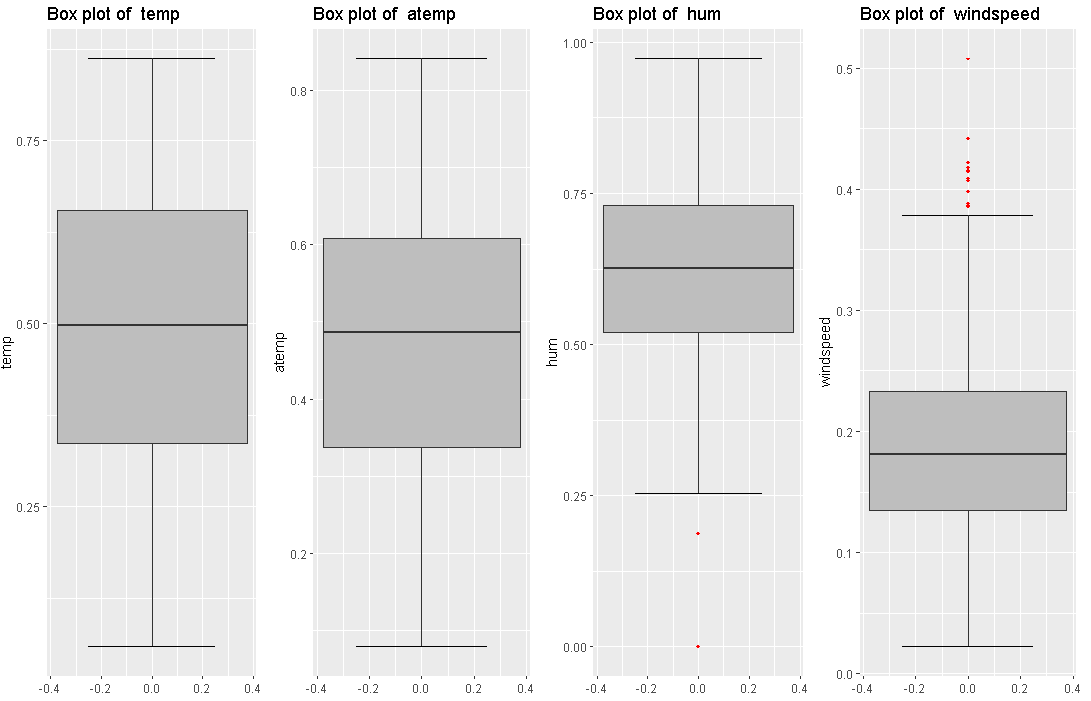
In the given dataset we could not find any missing values

### 2.1.3 Outlier Analysis

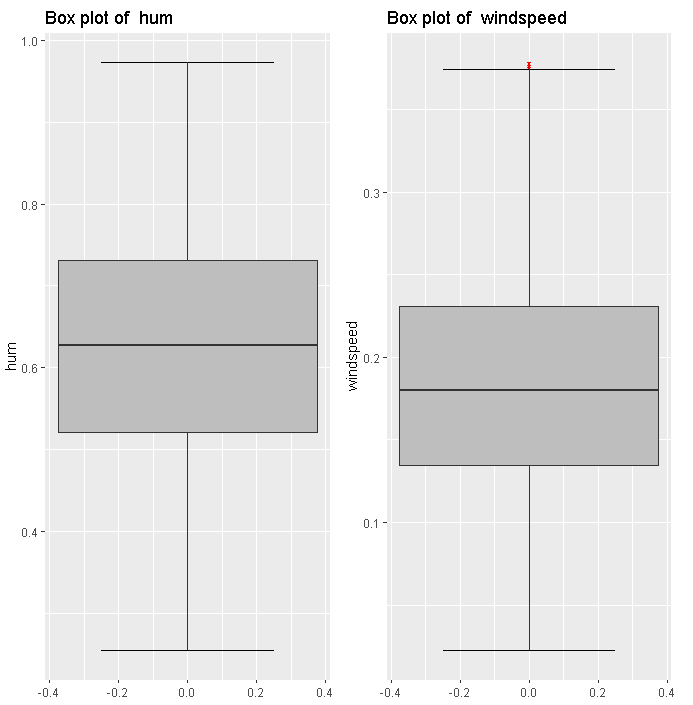
We can clearly observe from the probability/frequency distributions that the variables are not much skewed, except for few exceptions like ‘hum’ and ‘windspeed’ .The skew in these distributions can be most likely explained by the presence of outliers and extreme values in the data. We can see the effect of the skew in the histogram figures in Chapter 2.1.1. This is clearly the effect of outliers and extreme values.

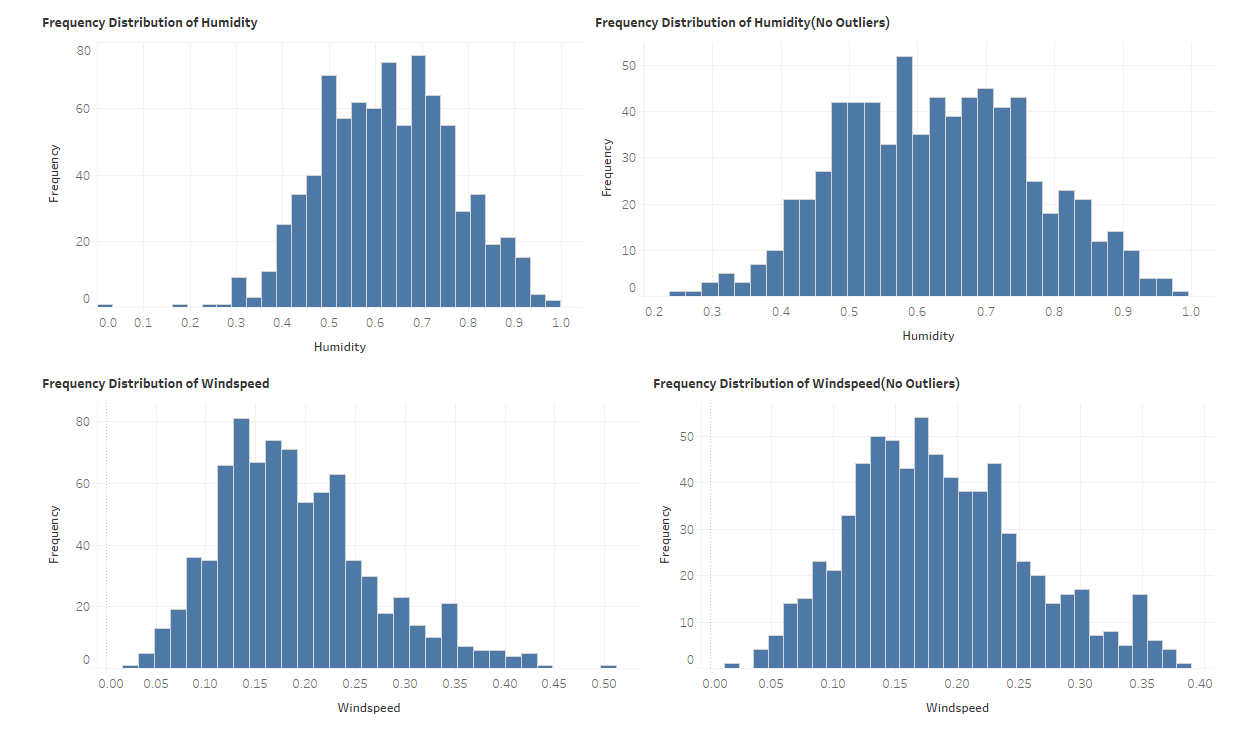
One of the other steps of **pre-processing** apart from checking for normality is the presence of outliers. In this case we use a classic approach of removing outliers, We visualize the outliers using *boxplots*.

In below figure we have plotted the boxplots of the 4 predictor variables. As we can see, we have few outliers and extreme values in hum and windspeed.



Even though the data we received was normalised, as mentioned , yet there were few outliers identified. These outliers do not look like data which may have been erroneously entered, hence we decided to impute the Outliers using KNN imputation method.   
After performing outlier Analysis and imputing outliers using *KNN Imputation Method*, We plot the Predictor boxplots again and compare the differences .

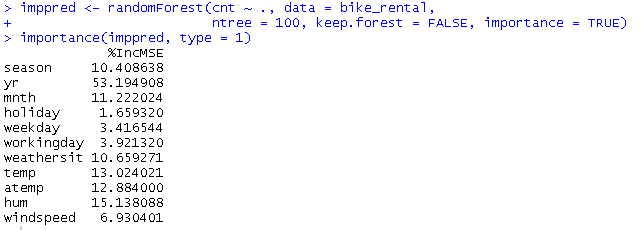
  
  
From the above Boxplots we can easily see that ‘hum’ does not contain any more outliers and ‘windspeed’ has 1 oulier very close to the maximum value, hence it can be ignored.

Let us now compare the histograms of ‘hum’ and ‘windspeed’ with and without outliers.  
  


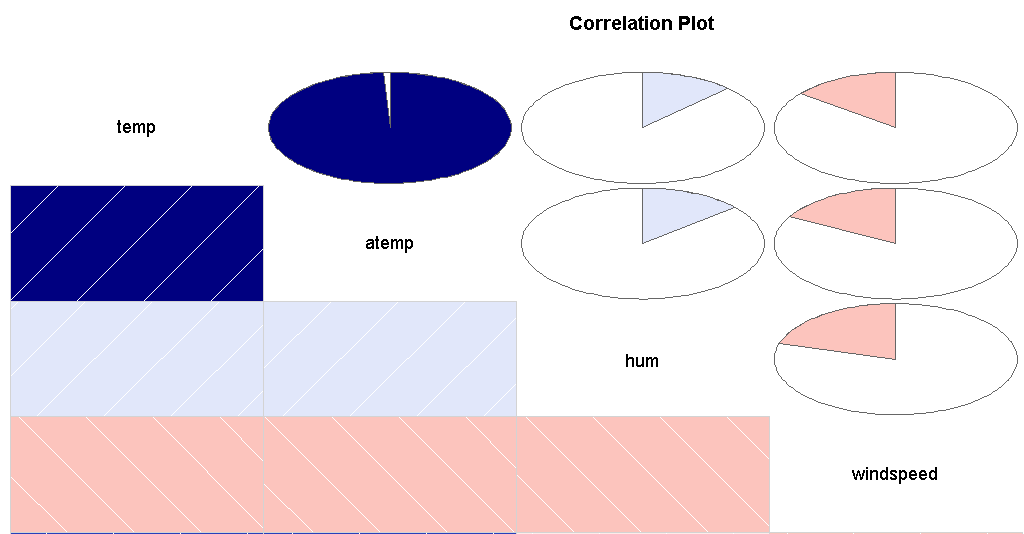
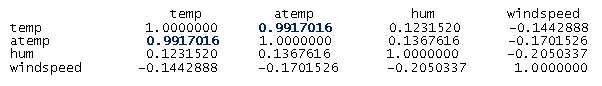
As can be seen from the above comparison, the skewness has been removed in hum and windspeed, because we replaced the Outliers with KNN imputated values.

### 2.1.4 Feature Selection and Dimension Reduction

Before performing any type of modeling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class prediction.  
  
Another step of Exploratory Data Analysis is to look for highly correlated variables in the data. Below we have used Random Forest to perform feature selection.



We can see that ‘yr’ has the highest prediction power followed by humidity and temp.

We also generated the below correlation plot to check interdependency of Continuous Variables.  
  
  
  
  
  
The above plot and values tells that temp and atemp are highly correlated.  
Finally we have dropped below variables from the dataset:  
**instant**: It is nothing but the index and has no role in Total Count  
**dteday**: Month and Year are already present as different variables, hence we do not need exact date.  
**atemp**: Because temp and atemp are highly correlated and atemp has lower prediction power than temp  
**casual**: This is a part of what we are going to predict  
**registered**: This is a part of what we are going to predict.

## 2.2 Modeling

### 2.2.1 Model Selection

The dependent variable can fall in either of the four categories:

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

If the dependent variable, in our case *Count*  is Nominal the only predictive analysis that we can perform is **Classification**, and if the dependent variable is Interval or Ratio the normal method is to do a **Regression** analysis, or classification after binning.

You always start your model building from the most simplest to more complex. Therefore we use Multiple Linear Regression.

### 2.2.2 Multiple Linear Regression

Call:

lm(formula = cnt ~ ., data = train)

Residuals:

Min 1Q Median 3Q Max

-3545.2 -480.7 58.8 551.1 2519.9

Coefficients:

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

(Intercept) 1437.31 267.26 5.378 1.10e-07 \*\*\*

season 525.52 62.91 8.353 5.04e-16 \*\*\*

yr 2057.77 75.41 27.289 < 2e-16 \*\*\*

mnth -43.83 19.65 -2.230 0.0261 \*

holiday -338.61 235.07 -1.440 0.1503

weekday 55.15 18.37 3.002 0.0028 \*\*

workingday 165.89 81.52 2.035 0.0423 \*

weathersit -623.56 92.88 -6.713 4.59e-11 \*\*\*

temp 5126.44 225.10 22.774 < 2e-16 \*\*\*

hum -855.69 376.56 -2.272 0.0234 \*

windspeed -2213.52 560.11 -3.952 8.72e-05 \*\*\*

---

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

Residual standard error: 893.5 on 573 degrees of freedom

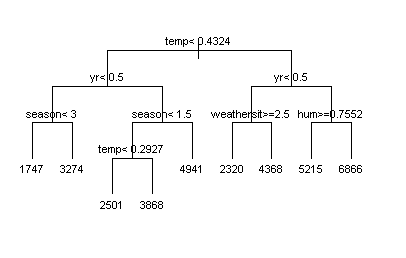
Multiple R-squared: 0.7871, Adjusted R-squared: 0.7833

F-statistic: 211.8 on 10 and 573 DF, p-value: < 2.2e-16

As you can see the *Adjusted R-squared* value, we can explain about 78% of the data using our multiple linear regression model. This is quite average. Also P value for holiday is >0.05 which means it is not very significant for the target prediction.

### 2.2.3 Regression Trees

Now we will try and use a different regression model to predict our *Cnt* target variable. We will use a regression tree to predict the values of our target variable.



Regression Tree for Cnt

### 2.2.4 Random Forest Lastly we used Random Forest Regression to predict the Count of Rentals

Length Class Mode

call 5 -none- call

type 1 -none- character

predicted 584 -none- numeric

mse 500 -none- numeric

rsq 500 -none- numeric

oob.times 584 -none- numeric

importance 20 -none- numeric

importanceSD 10 -none- numeric

localImportance 0 -none- NULL

proximity 0 -none- NULL

ntree 1 -none- numeric

mtry 1 -none- numeric

forest 11 -none- list

coefs 0 -none- NULL

y 584 -none- numeric

test 0 -none- NULL

inbag 0 -none- NULL

terms 3 terms call

**Chapter 3**

# Conclusion

## 3.1 Model Evaluation

Now that we have a few models for predicting the target variable, we need to decide which one to choose. There are several criteria that exist for evaluating and comparing models. We can compare the models using any of the following criteria:

1. Predictive Performance
2. Interpretability
3. Computational Efficiency

In our case of Wine Data, the latter two, *Interpretability* and *Computation Efficiency*, do not hold much significance. Therefore we will use *Predictive performance* as the criteria to compare and evaluate models.

Predictive performance can be measured by comparing Predictions of the models with real values of the target variables, and calculating some average error measure.

### 3.1.1 Mean Absolute Percentage Error (MAPE)

MAPE is one of the error measures used to calculate the predictive performance of the model. We will apply this measure to our models that we have generated in the previous section.

We defined a MAPE function as follows:

MAPE = function(y, yhat){ mean(abs((y - yhat)/y))}

Using the above function we calculated Percentage errors for all the 3 models we have used  
  
1) Multiple Linear Regression  
 #Error: 14.4% #Accuracy: 85.6%

2) Decision Tree Regression  
 #Error : 16.9% #Accuracy:83.1%

3) Random Forest Regression  
 #Error: 12.3% #Accuracy: 87.7%

## 3.2 Model Selection

We can see that all 3 models performed comparatively same.  
The priority of selecting Prediction Model for the Bike Rental dataset would be  
  
Random Forest > Linear Regression > Decision Tree

## 3.3 Sample Testing

## We picked the below sample data to test our model

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **season** | **yr** | **mnth** | **holiday** | **weekday** | **workingday** | **weathersit** | **temp** | **hum** | **windspeed** | **cnt** |
| 4 | 1 | 9 | 0 | 5 | 1 | 2 | 0.619167 | 0.69 | 0.164179 | 7415 |

The Random Forest Model Predicted the count as 6836

The Mean Absolute Error Percentage for this sample data was 8 %

# Appendix 1- R Code

### Complete R File

|  |
| --- |
| rm(list=ls(all=T))  Load Libraries  x = c("ggplot2", "corrgram", "DMwR", "caret", "randomForest", "unbalanced", "C50", "dummies", "e1071", "Information", "MASS", "rpart", "gbm", "ROSE", 'sampling', 'DataCombine','inTrees','usdm')  install.packages(x)  lapply(x, require, character.only = TRUE)  ## Read the data  bike\_rental = read.csv("day.csv", header = T, na.strings = c(" ", "", "NA"))  ########################## Explore the data ##########################  str(bike\_rental)  #Remove dteday , instant column  bike\_rental= subset(bike\_rental, select = -c(dteday,instant, casual,registered))  # Casual and Registered columns are also removed because that is what we are going to predict.  # Generate histograms for continuous variables  hist(bike\_rental$temp)  hist(bike\_rental$atemp)  hist(bike\_rental$hum)  hist(bike\_rental$windspeed)  ##############Missing Values Analysis##################  #Check for null fields  sum(is.na(bike\_rental))  ###############Outlier Analysis###############  # ## BoxPlots - Distribution and Outlier Check  #First we will convert non continuous variables to Factor  bike\_rental[1:7] <- sapply(bike\_rental[1:7] , as.factor)  #select only numeric index  numeric\_index = sapply(bike\_rental,is.numeric)  numeric\_index  numeric\_data =bike\_rental[,numeric\_index]  cnames = colnames(numeric\_data)  #Generate Box Plots for Numeric variables. The same code has been used for Univariate Analysis  for (i in 1:length(cnames))  {  assign(paste0("gn",i), ggplot(aes\_string(y = (cnames[i])), data = subset(bike\_rental))+  stat\_boxplot(geom = "errorbar", width = 0.5) +  geom\_boxplot(outlier.colour="red", fill = "grey" ,outlier.shape=18,  outlier.size=1, notch=FALSE) +  theme(legend.position="bottom")+  labs(y=cnames[i])+  ggtitle(paste("Box plot of ",cnames[i])))  }  ## Plotting plots together  gridExtra::grid.arrange(gn1,gn2,gn3,gn4,ncol=4)  # #Replace all outliers with NA and impute  # #create NA on 'hum' and 'windspeed'  df=bike\_rental  #bike\_rental=df  cnames= c( "hum" , "windspeed")  for(i in cnames){  val = bike\_rental[,i][bike\_rental[,i] %in% boxplot.stats(bike\_rental[,i])$out]  #print(length(val))  bike\_rental[,i][bike\_rental[,i] %in% val] = NA  }  #Check for NA in the dataset.  sum(is.na(bike\_rental))  #We get count of NA=15 , hence there were a total of 15 outliers  # Temporarily change variable types to numeric because KNN Imputation works on numeric data #only  bike\_rental[1:7] <- sapply(bike\_rental[1:7] , as.numeric)  #Apply KNN Imputation  bike\_rental = knnImputation(bike\_rental, k =5)  #Check for NA in the dataset again.  sum(is.na(bike\_rental))  #We get count of NA=0  #Convert categorical variables back to Factor  bike\_rental[1:7] <- sapply(bike\_rental[1:7] , as.factor)  #################Feature Selection##################  # Use Random Forest to determine prediction power of each variable  variable\_power <- randomForest(cnt ~ ., data = bike\_rental,  ntree = 100, keep.forest = FALSE, importance = TRUE)  #Display output of Random Forest  importance(variable\_power, type = 1)  # Correlation Plot  corrgram(bike\_rental[,numeric\_index], order = F,  upper.panel=panel.pie, text.panel=panel.txt, main = "Correlation Plot")  #Correlation Values  sub=data.frame(bike\_rental$temp,bike\_rental$atemp,bike\_rental$hum,bike\_rental$windspeed)  cor(sub)  # Dimension Reduction  bike\_rental= subset(bike\_rental, select = -c(atemp))    ###############Feature Scaling################  #Normality check  hist(bike\_rental$temp)  hist(bike\_rental$atemp)  hist(bike\_rental$hum)  hist(bike\_rental$windspeed)  #As we have already removed any outliers, the dataset is already normalised  #################Model Development###############  #Clean the environment  library(DataCombine)  rmExcept("bike\_rental")  #Define MAPE function  MAPE = function(y, yhat){ mean(abs((y - yhat)/y))}  #Divide the data into train and test using Random Sampling  set.seed(123)  train\_index =sample(1:nrow(bike\_rental), 0.80 \* nrow(bike\_rental))  train = bike\_rental[train\_index,]  test = bike\_rental[-train\_index,]  ######### MULTIPLE LINEAR REGRESSION ###############  #We convert all factor variables to numeric, as expected by the model.  bike\_rental[1:7] <- sapply(bike\_rental[1:7] , as.numeric)  #run regression model  lm\_model = lm(cnt ~., data = train)  #Summary of the model  summary(lm\_model)  #Predict  predictions\_LR = predict(lm\_model, test[,1:10])  #Calculate MAPE  MAPE(predictions\_LR,test[,11])  #Error: 14.4%  #Accuracy: 85.6%  ############# DECISION TREE REGRESSION ################  bike\_rental[1:7] <- sapply(bike\_rental[1:7] , as.factor)  #Generating Decision Tree  rtmodel <- rpart(cnt ~ ., data = train)  plot(rtmodel, uniform = T, branch = 1, margin = 0.1,cex =0.7)  text(rtmodel, cex = 0.7)  #Apply the Decision Tree Regression through rpart  fit = rpart(cnt ~ ., data = train, method = "anova")  #Predict for new test cases  predictions\_DT = predict(fit, test[,-11])  summary(predictions\_DT)  MAPE( predictions\_DT,test[,11])  #Error Rate: 16.9%  #Accuracy:83.1%  #############RANDOM FOREST REGRESSION################  RF\_model = randomForest(cnt ~ ., train, importance = TRUE, ntree = 500)  summary(RF\_model)  #Predict test data using random forest model  RF\_Predictions= predict(RF\_model, test[,-11])  MAPE(RF\_Predictions,test[,11])  #Error: 12.3%  #Accuracy: 87.7%  #### SAMPLE INPUT AND OUTPUT#######  #Select 1 random row from dataset  sample\_index =sample(1:nrow(bike\_rental), 0.002 \* nrow(bike\_rental))  test\_sample = bike\_rental[sample\_index,]  #View Sample  test\_sample  #Cnt for above sample is 7415  #Predict Output Count using the model  RF\_Predictions\_sample= predict(RF\_model, test\_sample[,-11])  # View Predicted Value of Cnt  RF\_Predictions\_sample  #Predicted value is 6836.77  #Calculate Mean Percentage Error  MAPE(RF\_Predictions\_sample,test\_sample[,11])  #Error: 8.4% |

|  |
| --- |
|  |

|  |
| --- |
| }  **return**(df)  }  outliereff <- function(varlist, df) { for (i in varlist) { total = **length**(df[, i])  **par**(mfrow = **c**(2, 2), oma = **c**(0, 0, 3,  0)) **boxplot**(df[, i], main = "With Outliers") **hist**(df[, i], main = "With Outliers", xlab = NA, ylab = NA, prob = TRUE)  df <- **outtona**(i, df)  **boxplot**(df[, i], main = "Without outliers") **hist**(df[, i], main = "Without outliers", xlab = NA, ylab = NA, prob = TRUE)  out <- **sum**(**is.na**(df[, i])) per <- **round**((out)/total \* 100, 1) **title**(**paste**("Effect of", out, "(", per, "%)", "Outliers on", **colnames**(df)[i], sep = " "), outer = TRUE)  }  }  **outliereff**(4, d1.r) d1.on <- **outtona**(1:11, d1.r) d1.f <- d1.on[**complete.cases**(d1.on), ] d2.on <- **outtona**(1:11, d2.r) d2.f <- d2.on[**complete.cases**(d2.on), ] **qboxp**(d1.f) **qboxp**(d2.f)  imppred <- **randomForest**(quality ~ ., data = d1, ntree = 100, keep.forest = FALSE, importance = TRUE)  **importance**(imppred, type = 1)  imppred <- **randomForest**(quality ~ ., data = d2, ntree = 100, keep.forest = FALSE, importance = TRUE)  **importance**(imppred, type = 1) **symnum**(**cor**(d1.r)) **symnum**(**cor**(d2.r))  lrmodel.red <- **lm**(Quality ~ ., data = d1.r)  **summary**(lrmodel.red)  **kable**(**anova**(lrmodel.red), booktabs = T)  lrmodel.red2 <- **update**(lrmodel.red, . ~ . - `Citric Acid` -  `Residual Sugar`) **summary**(lrmodel.red2)  rtmodel.red <- **rpart**(Quality ~ ., data = d1.r) **plot**(rtmodel.red, uniform = T, branch = 1, margin = 0.05, cex = 0.9)  **text**(rtmodel.red, cex = 0.7)  lrm.pred.red <- **predict**(lrmodel.red, d1.r) rt.pred.red <- **predict**(rtmodel.red, d1.r) mae.lrm.red <- **mean**(**abs**(lrm.pred.red - d1.r[,  12])) mae.rt.red <- **mean**(**abs**(rt.pred.red - d1.r[, 12])) |

|  |
| --- |
| mae.lrm.red mae.rt.red  (mse.lrm.red <- **mean**((lrm.pred.red - d1.r[, 12])^2)) (mse.rt.red <- **mean**((rt.pred.red - d1.r[, 12])^2))  **multi.hist**(d2.r, main = NA, dcol = **c**("blue", "red"), dlty = **c**("solid", "solid"), bcol = "grey95")  **qboxp**(d2.r) **outliereff**(5, d1.r) **outliereff**(10, d1.r) **outliereff**(1, d2.r) **outliereff**(5, d2.r) **outliereff**(6, d2.r) **outliereff**(8, d2.r) **outliereff**(10, d2.r) |