DataMining

Packages to include

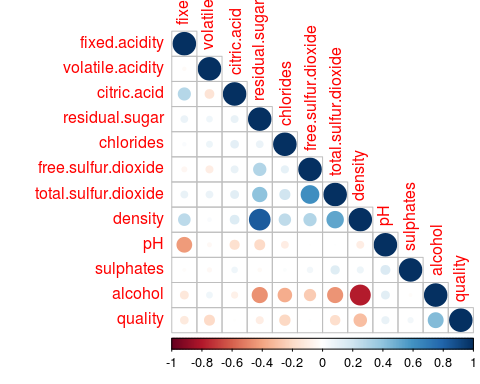
setwd("/home/roshan/workspace/R Studio/DataMining/")  
library(party)  
library(corrplot)  
library(caTools)  
library(ggplot2)

Import Data Set

data <- read.csv(file = "wineQualityWhites.csv", sep = ",", header = TRUE)

Feature Selection using Correlation Plots

cr <- cor(data[c(-1)])  
corrplot(cr, type = "lower")



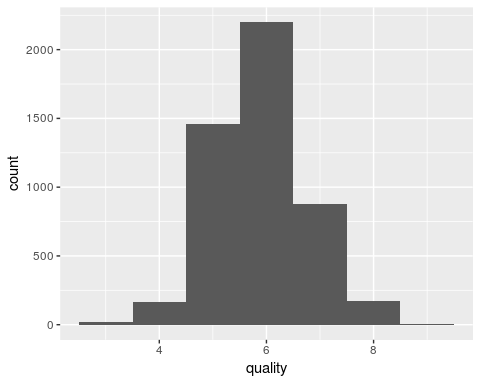
Find Quality Levels

data$quality.factor <- as.factor(data$quality)  
  
levels(data$quality.factor)

## [1] "3" "4" "5" "6" "7" "8" "9"

Histogram of Quality

ggplot(data = data, aes(x = quality)) +  
 geom\_histogram(binwidth = 1)



Add Binary Ordered Quality Attribute

for(i in 1:nrow(data)){  
 if(data$quality[[i]] < 6){  
 data$quality.order[[i]] <- "Bad"  
 }else{  
 data$quality.order[[i]] <- "Good"  
 }   
}

Count

i <- j <- 0  
for(q in data$quality.order){  
 if(q == "Bad"){  
 i <- i+1   
 }else{  
 j <- j+1  
 }  
}  
print(i)

## [1] 1640

print(j)

## [1] 3258

data$quality.order <- factor(factor(data$quality.order), levels = c("Bad", "Good"))

Convert Quality To Binary Attribute

for(i in 1:nrow(data)){  
 if(data$quality.order[[i]] == "Bad"){  
 data$quality.num[[i]] <- 0  
 }else{  
 data$quality.num[[i]] <- 1  
 }   
}

Train and Testing Data Sets

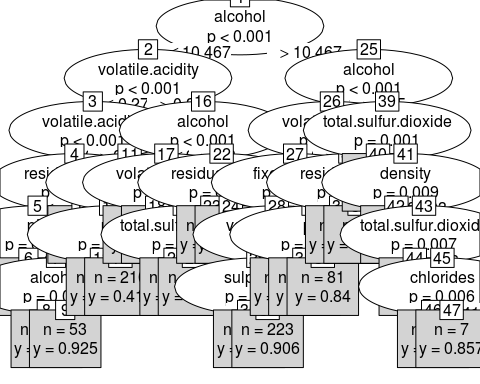
value <- sample.split(data$X, SplitRatio = 0.7)  
train.data <- subset(data, value == TRUE)  
test.data <- subset(data, value == FALSE)  
  
write.csv(train.data, file = "train\_data.csv")  
write.csv(test.data, file = "test\_data.csv")

Loading Existing training and testing data sets

train.data <- read.csv(file = "train\_data.csv", sep = ",", header = TRUE)   
test.data <- read.csv(file = "test\_data.csv", sep = ",", header = TRUE)

Fitting a Decision Tree Model and Plotting the tree

d.tree <- ctree(quality.num ~   
 fixed.acidity   
 + volatile.acidity   
 + residual.sugar   
 + chlorides   
 + total.sulfur.dioxide   
 + density  
 + pH  
 + sulphates  
 + alcohol  
 , data = train.data)  
  
  
plot(d.tree, type = "simple")



Predictions for the Test Data Set

prob <- predict(d.tree, newdata = test.data, type = "response")  
  
classes <- c()  
for (i in prob) {  
 if(i >= 0.5) {  
 classes <- c(classes,1)  
 } else {  
 classes <- c(classes,0)  
 }   
}

Plotting Confussion Matrix and finding the Accuracy

conf.matrix <- table(Actual = test.data[,16], Predicted = classes)  
print(conf.matrix)

## Predicted  
## Actual 0 1  
## Bad 276 238  
## Good 148 808

acc <- (conf.matrix[1,1]+conf.matrix[2,2])/length(test.data[,16])  
acc <- round(acc, digits = 4)  
print(paste("Accuracy = ",acc))

## [1] "Accuracy = 0.7374"