winequality

Shiva Sankar Modala

2023-03-28

# Installing and loading the necessary packages  
library(rpart)  
#install.packages("rpart.plot")  
# Package to create the binary decision tree  
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.2.3

library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

library(caret)

## Loading required package: ggplot2

##   
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':  
##   
## margin

## Loading required package: lattice

# Loading the Wine Quality sample dataset from the UCI Machine Learning Repository  
url\_red = "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv"  
url\_white = "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv"  
# Preparing the table  
RedWine <- read.table(file=url\_red, header=TRUE, sep=";",stringsAsFactors=TRUE)  
WhiteWine <- read.table(file=url\_white, header=TRUE, sep=";",stringsAsFactors=TRUE)

#redwine  
set.seed(1)

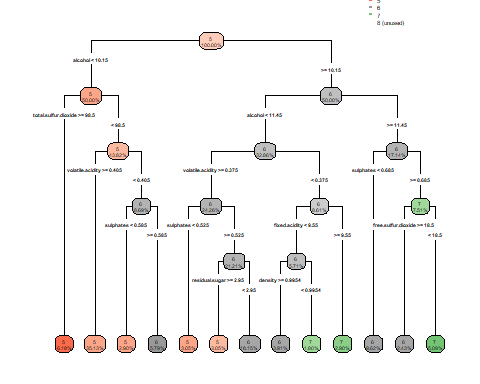
# Create an 80/20 test-train split of each wine dataframe  
index <- createDataPartition(RedWine$quality,p=0.2,list=FALSE)

# Separating the data based on the test and train data.  
test\_red <-RedWine[index,]  
train\_red <-RedWine[-index,]

train\_red$quality <- factor(train\_red$quality)  
test\_red$quality <- factor(test\_red$quality)

# Use the rpart package to induce a decision tree of both the red and white wines  
rpart\_tree\_red = rpart(quality~., data = train\_red)  
# targeting the quality output variable  
rpart\_predict\_red <- predict(rpart\_tree\_red, test\_red, type = "class")

# Visualizing the tree using the rpart.plot library  
rpart.plot(rpart\_tree\_red, digits = 4, fallen.leaves = TRUE, type = 4, extra = 100)



table(rpart\_predict\_red)

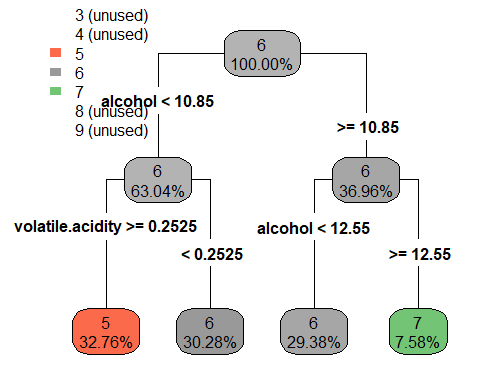
## rpart\_predict\_red  
## 3 4 5 6 7 8   
## 0 0 158 134 29 0

# Using the caret package confusionMatrix method to determine the decision tree accuracy on the test set  
decision\_tree\_red\_cm<-confusionMatrix(data = rpart\_predict\_red, reference = test\_red$quality)

#First split was done at “alcohol < 11” for White wine dataset  
#First split was done at “alcohol < 9.5” for Red wine dataset  
#Sulphates was taken into consideration in Red Wine Dataset. On the other hand its absent in White Wine Dataset.  
#Total Sulfur Dioxide was taken into consideration in Red Wine Dataset and its absent in White Wine Dataset.  
#Free Sulfur Dioxide was taken into consideration in White Wine Dataset and its absent in Red Wine Dataset.

#white wine  
set.seed(1)  
index <- createDataPartition(WhiteWine$quality,p=0.3,list=FALSE)  
test\_white <-WhiteWine[index,]  
train\_white <-WhiteWine[-index,]

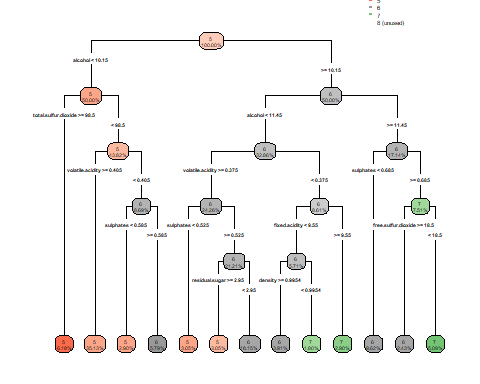
train\_white$quality <- factor(train\_white$quality)  
test\_white$quality <- factor(test\_white$quality)  
rpart\_tree\_white = rpart(quality~., data = train\_white)  
rpart\_predict\_white <- predict(rpart\_tree\_white, test\_white, type = "class")  
rpart.plot(rpart\_tree\_white, digits = 4, fallen.leaves = TRUE, type = 4, extra = 100)



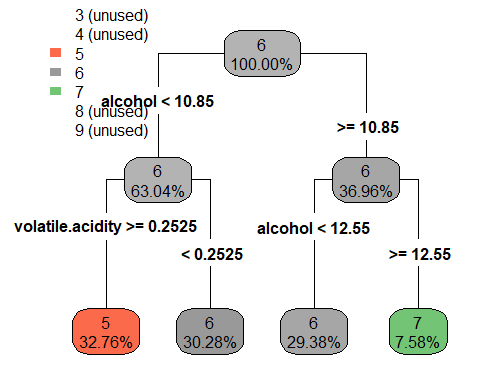
table(rpart\_predict\_white)

## rpart\_predict\_white  
## 3 4 5 6 7 8 9   
## 0 0 487 888 95 0 0

# Using the caret package confusionMatrix method to determine the decision tree accuracy on the test set  
decision\_tree\_white\_cm<-confusionMatrix(rpart\_predict\_white, test\_white$quality)  
# Using the rpart package to induce a decision tree of both the red and white wines  
rpart.plot(rpart\_tree\_red, digits = 4, fallen.leaves = TRUE, type = 4, extra = 100)



# Using the rpart package to induce a decision tree of both the red and white wines  
rpart.plot(rpart\_tree\_white, digits = 4, fallen.leaves = TRUE, type = 4, extra = 100)



varImp(rpart\_tree\_red)

## Overall  
## alcohol 99.52523  
## chlorides 4.26621  
## citric.acid 25.23650  
## density 43.89424  
## fixed.acidity 46.27422  
## free.sulfur.dioxide 12.96934  
## pH 17.60606  
## residual.sugar 20.44688  
## sulphates 104.79208  
## total.sulfur.dioxide 76.46514  
## volatile.acidity 103.15831

varImp(rpart\_tree\_white)

## Overall  
## alcohol 187.18188  
## chlorides 86.82763  
## citric.acid 17.63781  
## density 100.60980  
## free.sulfur.dioxide 26.76760  
## total.sulfur.dioxide 42.57525  
## volatile.acidity 133.60637  
## fixed.acidity 0.00000  
## residual.sugar 0.00000  
## pH 0.00000  
## sulphates 0.00000

#randomforest  
random\_forest\_red <- randomForest(quality~., data = train\_red)  
randomforestred\_predict <- predict(object = random\_forest\_red, newdata = test\_red)  
randomforest\_red\_cm<-confusionMatrix(data = randomforestred\_predict, reference = test\_red$quality)

random\_forest\_white <- randomForest(quality~., data = train\_white)

randomforestwhite\_predict <- predict(object = random\_forest\_white, newdata = test\_white)  
randomforest\_white\_cm<-confusionMatrix(data = randomforestwhite\_predict, reference = test\_white$quality)

#Comparision  
print("Comparision of accuracy between red wine decision tree vs randomforest: for the Red Wine Decision Tree")

## [1] "Comparision of accuracy between red wine decision tree vs randomforest: for the Red Wine Decision Tree"

decision\_tree\_red\_cm$overall["Accuracy"]

## Accuracy   
## 0.6105919

print("Red Wine Random Forest")

## [1] "Red Wine Random Forest"

randomforest\_red\_cm$overall["Accuracy"]

## Accuracy   
## 0.7102804

print("Comparision of accuracy between white wine decision tree vs randomforest: White Wine Decision Tree")

## [1] "Comparision of accuracy between white wine decision tree vs randomforest: White Wine Decision Tree"

decision\_tree\_white\_cm$overall["Accuracy"]

## Accuracy   
## 0.4986395

print("White Wine Random Forest")

## [1] "White Wine Random Forest"

randomforest\_white\_cm$overall["Accuracy"]

## Accuracy   
## 0.670068

# For White Wine Dataset Random Forest returned an accuracy of 69.4% (+-2)   
# For Red Wine Dataset Random Forest returned an accuracy of 71.9% (+-2)

# The Accuracy increased from 52% to 69% in Random Forest Classifier in White Wine Dataset  
# The Accuracy increased from 53% to 71% in Random Forest Classifier in Red Wine Dataset