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
```

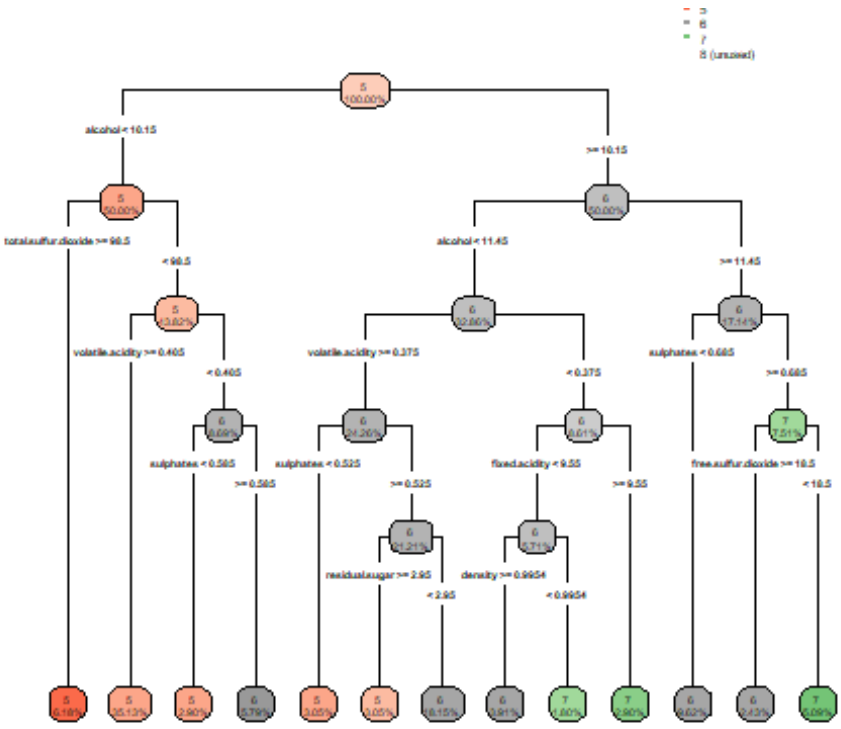
```
##
## Attaching package: 'ggplot2'
```

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

```
## Loading required package: lattice
```

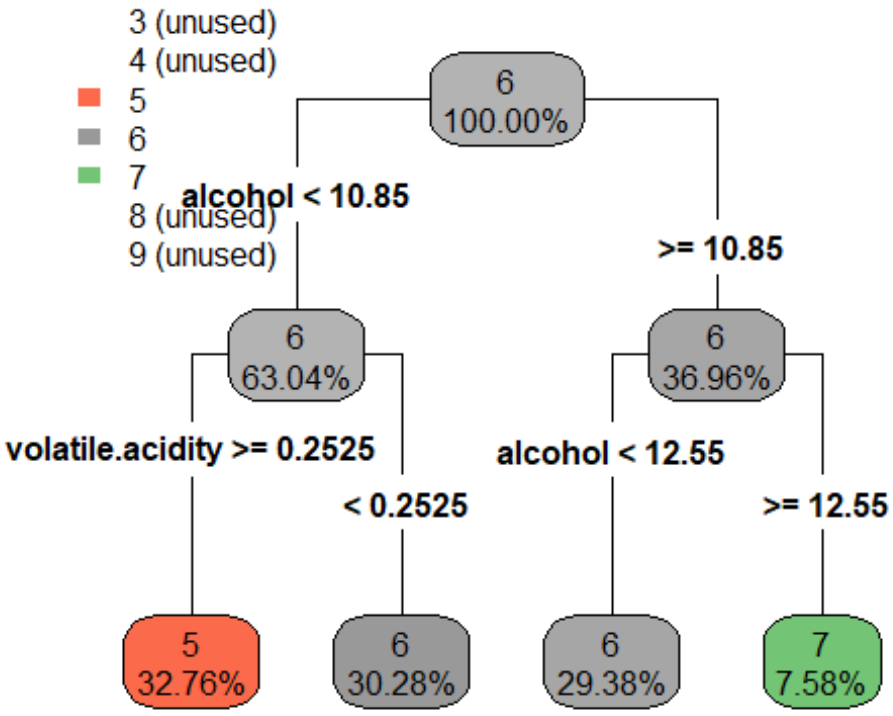
```
# Loading the Wine Quality sample dataset from the UCI Machine Learning
Repository
url_red = "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-
quality/winequality-red.csv"
url_white = "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-
quality/winequality-white.csv"
# 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)
```



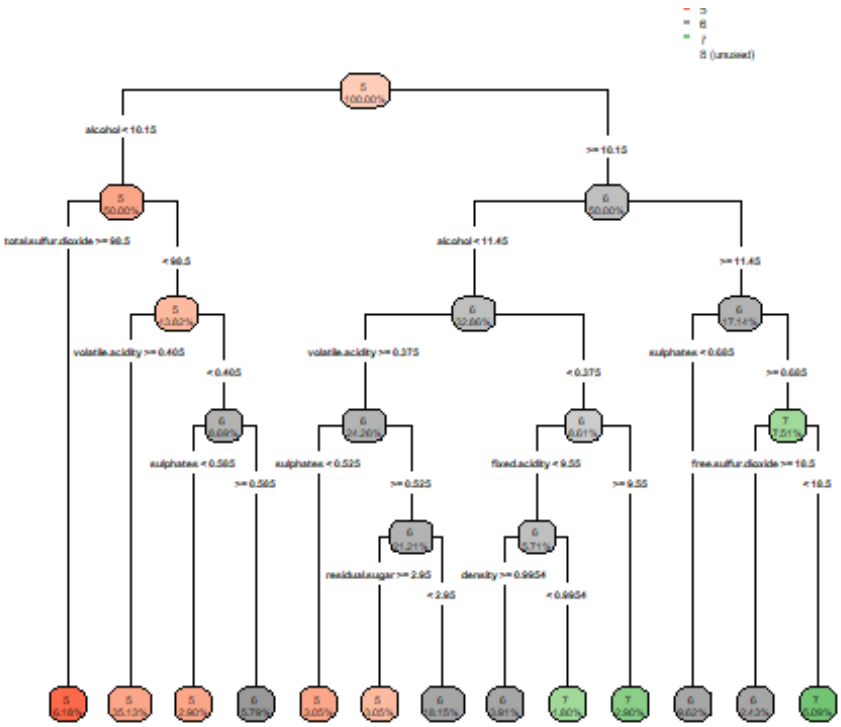
```
table(rpart_predict_red)

## 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)
```

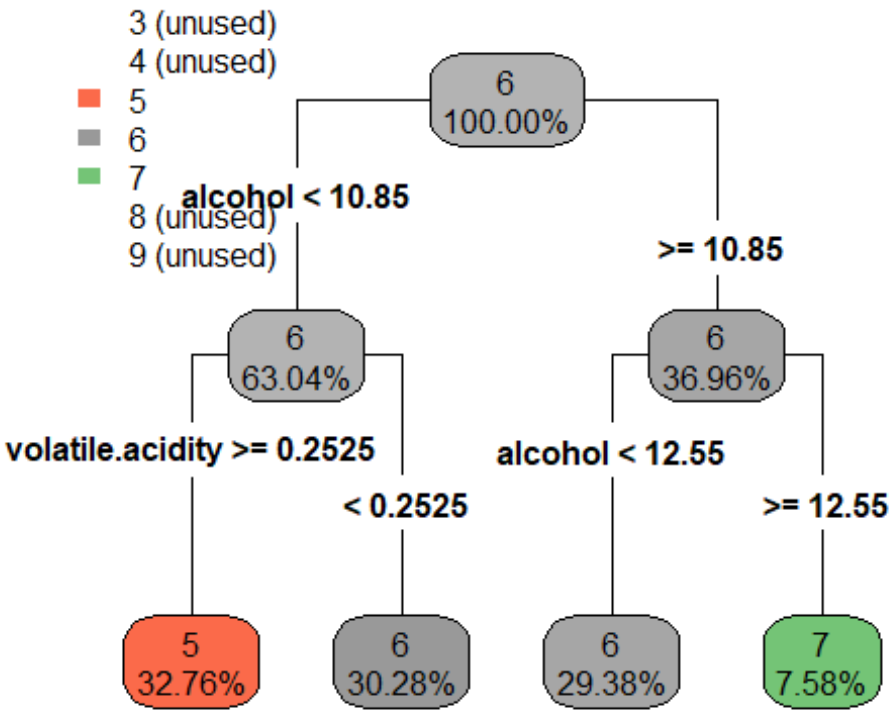



```
table(rpart_predict_white)

## 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
```