Decision trees and random forest

Objective: to predict wine quality rankings from its chemical properties.

Data set: Red wine dataset from https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv .

First, I prepared the environment, set the working directory and loaded the libraries using the following:

```
> ### Assignment:  Decision trees and random forest
>
> rm(list=ls()) #Clear the environment
> setwd("YOUR_PATH") #Set working directory for the assignment
> getwd() #Check working directory
[1] "YOUR_PATH"
>
>
> #Use decision trees and random forest to predict wine quality from its chemical properties
> #Dataset red wine from https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/w
inequality-red.csv
>
> #Load packages
>
> library(rpart)
> library (caret)
> library(randomForest)
```

Next, I loaded the red wine dataset from the previously downloaded csv file (https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv) using the following code:

```
#Load data from the CSV file
> redwine <- read.csv2("winequality-red.csv", header = TRUE, sep= ";", quote= "", stringsAsFactor
s= FALSE)
```

To make sure that the data was loaded correctly, I previewed the data and looked at the internal structure of the data frame:

```
> #Check
> View(redwine) #Preview data
> str(redwine) #Check internal structure of the data frame
'data.frame':      1599 obs. of  12 variables:
 $ X.fixed.acidity.       : chr  "7.4" "7.8" "7.8" "11.2" ...
 $ X.volatile.acidity.    : chr  "0.7" "0.88" "0.76" "0.28" ...
 $ X.citric.acid.         : chr  "0" "0" "0.04" "0.56" ...
 $ X.residual.sugar.      : chr  "1.9" "2.6" "2.3" "1.9" ...
 $ X.chlorides.           : chr  "0.076" "0.098" "0.092" "0.075" ...
 $ X.free.sulfur.dioxide. : chr  "11" "25" "15" "17" ...
 $ X.total.sulfur.dioxide.: chr  "34" "67" "54" "60" ...
 $ X.density.             : chr  "0.9978" "0.9968" "0.997" "0.998" ...
 $ X.pH.                  : chr  "3.51" "3.2" "3.26" "3.16" ...
 $ X.sulphates.           : chr  "0.56" "0.68" "0.65" "0.58" ...
 $ X.alcohol.             : chr  "9.4" "9.8" "9.8" "9.8" ...
 $ X.quality.             : int  5 5 5 6 5 5 5 7 7 5 ...
```

The data was loaded correctly. The data set contains 1599 observations of 12 variables. X.quality variable has int type, the rest of the variables loaded as char. First, for convenience purposes, I decided to change column names – delete X. in the beginning and delete the period in the end of each column name. I used the gsub() function to replace X. and .$ patterns:

```
> ###Change column names to easier to read
> #Delete X. in the beginning of column names
> names(redwine) <- gsub(pattern = "X.", replacement = "", names(redwine))
> #Delete the period in the end of column names
> names(redwine) <- gsub(pattern = ".$", replacement = "", names(redwine))
> #Check results
> str(redwine)
'data.frame':      1599 obs. of  12 variables:
 $ fixed.acidity       : chr  "7.4" "7.8" "7.8" "11.2" ...
 $ volatile.acidity    : chr  "0.7" "0.88" "0.76" "0.28" ...
 $ citric.acid         : chr  "0" "0" "0.04" "0.56" ...
 $ residual.sugar      : chr  "1.9" "2.6" "2.3" "1.9" ...
 $ chlorides           : chr  "0.076" "0.098" "0.092" "0.075" ...
 $ free.sulfur.dioxide : chr  "11" "25" "15" "17" ...
 $ total.sulfur.dioxide: chr  "34" "67" "54" "60" ...
 $ density             : chr  "0.9978" "0.9968" "0.997" "0.998" ...
 $ pH                  : chr  "3.51" "3.2" "3.26" "3.16" ...
 $ sulphates           : chr  "0.56" "0.68" "0.65" "0.58" ...
 $ alcohol             : chr  "9.4" "9.8" "9.8" "9.8" ...
 $ quality             : int  5 5 5 6 5 5 5 7 7 5 ...
```

As the above output shows, the column names were changed successfully. So, I proceeded to the data type conversions. I converted the output variable quality to a factor, and the rest of the variables to numeric using lapply(0 function:

```
> ###Data type conversions
> #Convert the response variable to factor
> redwine$quality <- as.factor(redwine$quality)
> #Convert all other columns to numeric
> redwine[1:11] <- lapply(redwine[1:11], as.numeric)
> #Check results
> str(redwine)
'data.frame':      1599 obs. of  12 variables:
 $ fixed.acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
 $ volatile.acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
 $ citric.acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
 $ residual.sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
 $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
 $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
 $ total.sulfur.dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
 $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
 $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
 $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
 $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
 $ quality             : Factor w/ 6 levels "3","4","5","6",..: 3 3 3 4 3 3 3 5 5 3 ...
```

The above conversion was successful, no warnings about any conversion errors (N/As etc.) . To make sure that there was no irregularities in the data that would affect the model, I displayed the summary statistics for all 12 variables in the data set:

```
> ###Summary statistics for all variables in the dataset
> summary(redwine)
 fixed.acidity   volatile.acidity  citric.acid    residual.sugar     chlorides       free.sulfur.dioxide total.sulfur.dioxide    density             pH
 Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900   Min.   :0.01200   Min.   : 1.00       Min.   :  6.00       Min.   :0.9901   Min.   :2.740
 1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900   1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 22.00       1st Qu.:0.9956   1st Qu.:3.210
 Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200   Median :0.07900   Median :14.00       Median : 38.00       Median :0.9968   Median :3.310
 Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539   Mean   :0.08747   Mean   :15.87       Mean   : 46.47       Mean   :0.9967   Mean   :3.311
 3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600   3rd Qu.:0.09000   3rd Qu.:21.00       3rd Qu.: 62.00       3rd Qu.:0.9978   3rd Qu.:3.400
 Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500   Max.   :0.61100   Max.   :72.00       Max.   :289.00       Max.   :1.0037   Max.   :4.010
   sulphates        alcohol      quality
 Min.   :0.3300   Min.   : 8.40   3: 10
 1st Qu.:0.5500   1st Qu.: 9.50   4: 53
 Median :0.6200   Median :10.20   5:681
 Mean   :0.6581   Mean   :10.42   6:638
 3rd Qu.:0.7300   3rd Qu.:11.10   7:199
 Max.   :2.0000   Max.   :14.90   8: 18
```

The above output shows that the target variable quality which contains wine expert ratings on a scale from 1 to 10 actually has only observations with levels 3 through 8. In addition the sample contains very few observations for both extremes and the bulk of observations is in the 5 – 6 range. For example, there is just 10 wine samples with the rating 3 in the whole dataset, and after splitting it in the training and testing sets the resulting number will not be sufficient to train the algorithm. So I decided to groups observation into three levels of quality –

low (for quality 3 and 4), medium (for quality 5 and 6), and high (for quality 7 and 8). I used nested ifelse() statements as follows:

```
> redwine$quality <- as.factor(with(redwine, ifelse(quality%in%c("3","4"), "low", (ifelse(quality
%in%c("5", "6"), "medium", "high")))))
```

```
> summary(redwine)
 fixed.acidity   volatile.acidity  citric.acid    residual.sugar     chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
 Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900   Min.   :0.01200   Min.   : 1.00       Min.   :  6.00       Min.   :0.9901
 1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900   1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 22.00       1st Qu.:0.9956
 Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200   Median :0.07900   Median :14.00       Median : 38.00       Median :0.9968
 Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539   Mean   :0.08747   Mean   :15.87       Mean   : 46.47       Mean   :0.9967
 3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600   3rd Qu.:0.09000   3rd Qu.:21.00       3rd Qu.: 62.00       3rd Qu.:0.9978
 Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500   Max.   :0.61100   Max.   :72.00       Max.   :289.00       Max.   :1.0037
       pH          sulphates         alcohol        quality
 Min.   :2.740   Min.   :0.3300   Min.   : 8.40   high  : 217
 1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50   low   :  63
 Median :3.310   Median :0.6200   Median :10.20   medium:1319
 Mean   :3.311   Mean   :0.6581   Mean   :10.42
 3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10
 Max.   :4.010   Max.   :2.0000   Max.   :14.90
> str(redwine)
'data.frame':   1599 obs. of  12 variables:
 $ fixed.acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
 $ volatile.acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
 $ citric.acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
 $ residual.sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
 $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
 $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
 $ total.sulfur.dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
 $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
 $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
 $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
 $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
 $ quality             : Factor w/ 3 levels "high","low","medium": 3 3 3 3 3 3 3 3 1 1 3 ...
>
```

As a result, the dataset contains 217 high quality wine samples, 63 low quality samples, and 1319 medium quality samples.

Next, I used the following code to split the data into training (80%) and testing (20%) sets:

```
> #Prepare training and testing datasets (80% training, 20% testing)
>
> set.seed(15)
> ind <- sample.int(n=nrow(redwine), size=floor(0.8*nrow(redwine)), replace = FALSE)
> redwine_train <-  redwine[ind, ] #training set
> redwine_test <- redwine[ -ind, ] #testing set
> #Prepare training and testing datasets (80% training, 20% testing)
>
> set.seed(15)
> ind <- sample.int(n=nrow(redwine), size=floor(0.8*nrow(redwine)), replace = FALSE)
> redwine_train <-  redwine[ind, ] #training set
> redwine_test <- redwine[ -ind, ] #testing set
```

The resulting training set contains 1279 observations of 12 variables and testing set contains 320 observations of 12 variables.

```
> #Check resulting datasets
```

```
> dim(redwine_train)
[1] 1279   12
> dim(redwine_test)
[1] 320  12
```

I compared proportion between the three levels of wine quality in the training and testing sets, and they were approximately equal:

```
> #Compare proportions between low, medium and high quality wines in testing and training data
> prop.table(table(redwine_train$quality))

     high       low    medium
0.1422987 0.0398749 0.8178264
> prop.table(table(redwine_test$quality))

    high      low   medium
0.109375 0.037500 0.853125
```

Next step is to build the classification model. I used rpart() function from the rpart package, and included all variables:

```
> #Build classification tree model
> winequality_rp <- rpart(quality ~ ., data = redwine_train)
>
> #Retrieve node details
> winequality_rp
n= 1279

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 1279 233 medium (0.142298671 0.039874902 0.817826427)
   2) alcohol>=11.55 201  95 medium (0.462686567 0.009950249 0.527363184)
     4) sulphates>=0.685 89  26 high (0.707865169 0.000000000 0.292134831)
       8) free.sulfur.dioxide< 18.5 62  12 high (0.806451613 0.000000000 0.193548387) *
       9) free.sulfur.dioxide>=18.5 27  13 medium (0.481481481 0.000000000 0.518518519)
        18) free.sulfur.dioxide>=27.5 11   2 high (0.818181818 0.000000000 0.181818182) *
        19) free.sulfur.dioxide< 27.5 16   4 medium (0.250000000 0.000000000 0.750000000) *
     5) sulphates< 0.685 112  32 medium (0.267857143 0.017857143 0.714285714)
      10) total.sulfur.dioxide< 15.5 29  13 high (0.551724138 0.034482759 0.413793103)
        20) sulphates>=0.585 14   2 high (0.857142857 0.000000000 0.142857143) *
        21) sulphates< 0.585 15   5 medium (0.266666667 0.066666667 0.666666667) *
      11) total.sulfur.dioxide>=15.5 83  15 medium (0.168674699 0.012048193 0.819277108)
        22) free.sulfur.dioxide>=31.5 9   2 high (0.777777778 0.000000000 0.222222222) *
        23) free.sulfur.dioxide< 31.5 74   8 medium (0.094594595 0.013513514 0.891891892) *
   3) alcohol< 11.55 1078 138 medium (0.082560297 0.045454545 0.871985158)
     6) volatile.acidity< 0.405 254  64 medium (0.236220472 0.015748031 0.748031496)
      12) alcohol>=10.45 118  51 medium (0.415254237 0.016949153 0.567796610)
        24) sulphates>=0.735 59  25 high (0.576271186 0.000000000 0.423728814)
          48) density>=0.99757 17   2 high (0.882352941 0.000000000 0.117647059) *
```

```
          49) density< 0.99757 42  19 medium (0.452380952 0.000000000 0.547619048)
            98) density< 0.9958 15   3 high (0.800000000 0.000000000 0.200000000) *
            99) density>=0.9958 27   7 medium (0.259259259 0.000000000 0.740740741) *
        25) sulphates< 0.735 59  17 medium (0.254237288 0.033898305 0.711864407) *
      13) alcohol< 10.45 136  13 medium (0.080882353 0.014705882 0.904411765) *
     7) volatile.acidity>=0.405 824  74 medium (0.035194175 0.054611650 0.910194175)
      14) volatile.acidity>=0.8375 65  19 medium (0.000000000 0.292307692 0.707692308)
        28) citric.acid< 0.085 29  13 low (0.000000000 0.551724138 0.448275862)
          56) citric.acid>=0.015 16   4 low (0.000000000 0.750000000 0.250000000) *
          57) citric.acid< 0.015 13   4 medium (0.000000000 0.307692308 0.692307692) *
        29) citric.acid>=0.085 36   3 medium (0.000000000 0.083333333 0.916666667) *
      15) volatile.acidity< 0.8375 759  55 medium (0.038208169 0.034255599 0.927536232) *
```

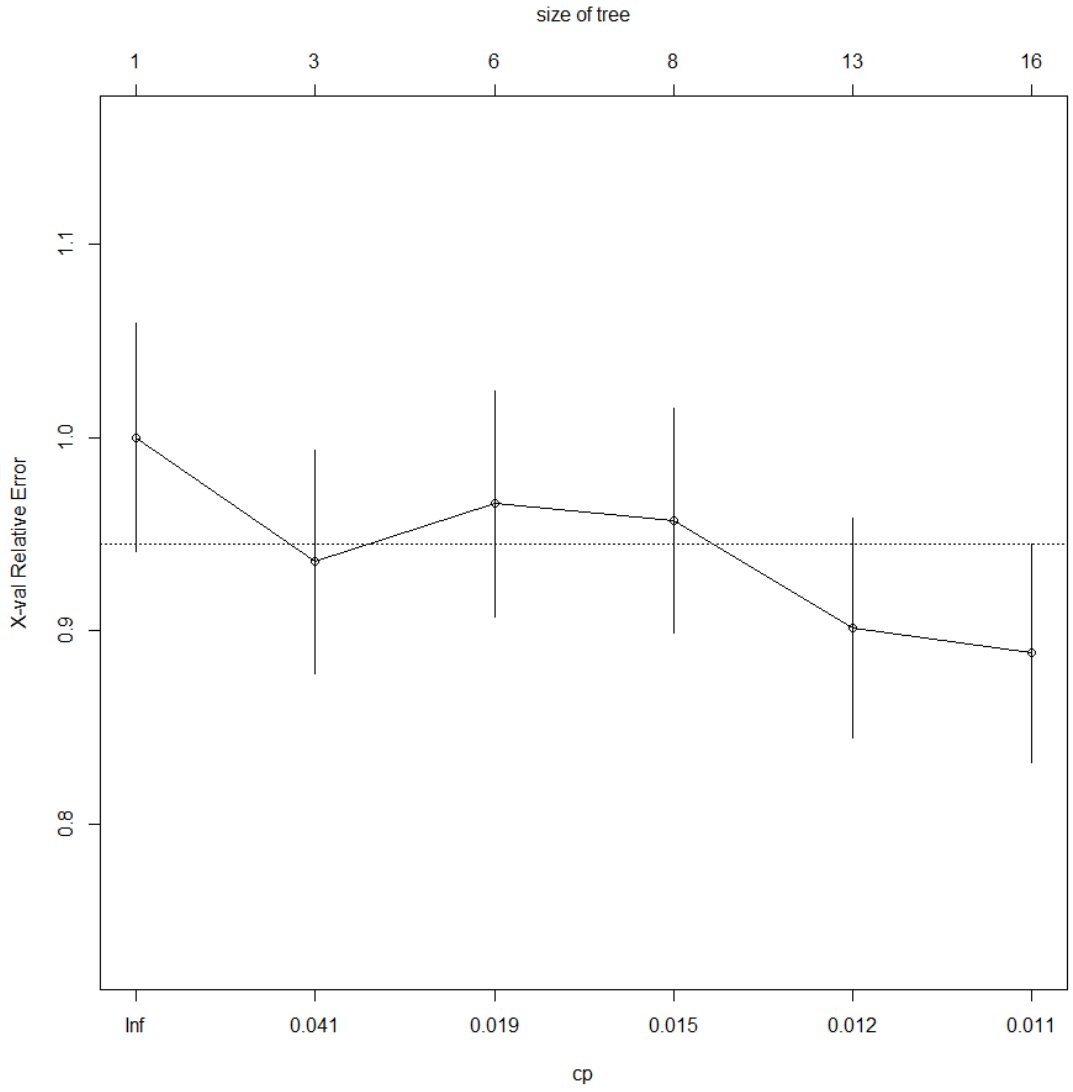
Then, I used printcp() to examine the complexityparameter:

```
> #Examine complexity
> printcp(winequality_rp)

Classification tree:
rpart(formula = quality ~ ., data = redwine_train)

Variables actually used in tree construction:
[1] alcohol              citric.acid          density              free.sulfur.dioxide  sulphates
total.sulfur.dioxide volatile.acidity

Root node error: 233/1279 = 0.18217

n= 1279

        CP nsplit rel error  xerror     xstd
1 0.079399      0   1.00000 1.00000 0.059245
2 0.021459      2   0.84120 0.93562 0.057716
3 0.017167      5   0.77682 0.96567 0.058441
4 0.012876      7   0.74249 0.95708 0.058236
5 0.011445     12   0.64807 0.90129 0.056860
6 0.010000     15   0.61373 0.88841 0.056532
```

The output above shows that the following seven variables were actually used in the tree construction: alcohol, citric.acid, density, free.sulfur.dioxide, sulphates, total.sulfur.dioxide and volatile.acidity. It also list the number od splits (nsplit), the relative error (xerror) and the standard error (xstand).

It is convenient to visually analyze the cost complexity parameters using plotcp(0 function:

```
> #Plot cost complexity parameters
> plotcp(winequality_rp)
```



The graph above shows that the minimum relative error estimated by a 10-fold classification (Xerror) is observed with the tree size of 16.

A detailed summary of the model can be displayed using the summary() command: I list the importance of the variables and characteristics of each node (how observations were split in each node):

```
> #Examine the model
> summary(winequality_rp)
Call:
rpart(formula = quality ~ ., data = redwine_train)
  n= 1279
```

```
          CP nsplit rel error    xerror       xstd
1 0.07939914      0 1.0000000 1.0000000 0.05924512
2 0.02145923      2 0.8412017 0.9356223 0.05771580
3 0.01716738      5 0.7768240 0.9656652 0.05844137
4 0.01287554      7 0.7424893 0.9570815 0.05823622
5 0.01144492     12 0.6480687 0.9012876 0.05686006
6 0.01000000     15 0.6137339 0.8884120 0.05653161

Variable importance
            alcohol           sulphates             density     volatile.acidity  free.sulfur.
dioxide total.sulfur.dioxide         citric.acid
                 23                  13                  13                  11
9                    8                   7
       fixed.acidity           chlorides                  pH       residual.sugar
                   6                   6                   4                   2

Node number 1: 1279 observations,    complexity param=0.07939914
  predicted class=medium  expected loss=0.1821736  P(node) =1
    class counts:   182    51  1046
   probabilities: 0.142 0.040 0.818
  left son=2 (201 obs) right son=3 (1078 obs)
  Primary splits:
      alcohol          < 11.55    to the right, improve=44.81302, (0 missing)
      sulphates        < 0.685    to the right, improve=26.88087, (0 missing)
      volatile.acidity < 0.3625   to the left,  improve=25.95614, (0 missing)
      citric.acid      < 0.315    to the right, improve=20.18783, (0 missing)
      density          < 0.99535  to the left,  improve=19.75534, (0 missing)
  Surrogate splits:
      density              < 0.993785 to the left,  agree=0.887, adj=0.279, (0 split)
      chlorides            < 0.0525   to the left,  agree=0.860, adj=0.109, (0 split)
      fixed.acidity        < 5.55     to the left,  agree=0.856, adj=0.085, (0 split)
      pH                   < 3.695    to the right, agree=0.850, adj=0.045, (0 split)
      total.sulfur.dioxide < 162.5    to the right, agree=0.845, adj=0.015, (0 split)

Node number 2: 201 observations,    complexity param=0.07939914
  predicted class=medium  expected loss=0.4726368  P(node) =0.157154
    class counts:    93     2   106
   probabilities: 0.463 0.010 0.527
  left son=4 (89 obs) right son=5 (112 obs)
  Primary splits:
      sulphates        < 0.685    to the right, improve=18.455050, (0 missing)
      pH               < 3.365    to the left,  improve= 9.083775, (0 missing)
      citric.acid      < 0.315    to the right, improve= 7.901565, (0 missing)
      fixed.acidity    < 7.85     to the right, improve= 7.333158, (0 missing)
      volatile.acidity < 0.335    to the left,  improve= 4.739165, (0 missing)
  Surrogate splits:
      fixed.acidity        < 8.05     to the right, agree=0.647, adj=0.202, (0 split)
      citric.acid          < 0.325    to the right, agree=0.632, adj=0.169, (0 split)
      volatile.acidity     < 0.345    to the left,  agree=0.627, adj=0.157, (0 split)
      alcohol              < 12.85    to the right, agree=0.612, adj=0.124, (0 split)
      total.sulfur.dioxide < 51       to the right, agree=0.607, adj=0.112, (0 split)

Node number 3: 1078 observations,    complexity param=0.01287554
  predicted class=medium  expected loss=0.1280148  P(node) =0.842846
    class counts:    89    49   940
   probabilities: 0.083 0.045 0.872
```

```
  left son=6 (254 obs) right son=7 (824 obs)
  Primary splits:
      volatile.acidity     < 0.405    to the left,  improve=13.244800, (0 missing)
      alcohol              < 10.45    to the right, improve=13.187220, (0 missing)
      sulphates            < 0.675    to the right, improve= 9.555997, (0 missing)
      total.sulfur.dioxide < 49.5     to the left,  improve= 6.863008, (0 missing)
      citric.acid          < 0.295    to the right, improve= 6.501941, (0 missing)
  Surrogate splits:
      citric.acid          < 0.395    to the right, agree=0.793, adj=0.122, (0 split)
      fixed.acidity        < 10.45    to the right, agree=0.773, adj=0.035, (0 split)
      free.sulfur.dioxide  < 3.5      to the left,  agree=0.768, adj=0.016, (0 split)
      total.sulfur.dioxide < 10.5     to the left,  agree=0.768, adj=0.016, (0 split)

Node number 4: 89 observations,    complexity param=0.01716738
  predicted class=high    expected loss=0.2921348  P(node) =0.06958561
    class counts:    63     0    26
   probabilities: 0.708 0.000 0.292
  left son=8 (62 obs) right son=9 (27 obs)
  Primary splits:
      free.sulfur.dioxide  < 18.5     to the left,  improve=3.972669, (0 missing)
      total.sulfur.dioxide < 56.5     to the left,  improve=3.750644, (0 missing)
      density              < 0.99533  to the left,  improve=3.280533, (0 missing)
      pH                   < 3.365    to the left,  improve=1.804469, (0 missing)
      alcohol              < 13.8     to the left,  improve=1.185295, (0 missing)
  Surrogate splits:
      total.sulfur.dioxide < 44       to the left,  agree=0.865, adj=0.556, (0 split)
      alcohol              < 11.65    to the right, agree=0.742, adj=0.148, (0 split)
      density              < 0.99176  to the right, agree=0.730, adj=0.111, (0 split)
      pH                   < 3.59     to the left,  agree=0.730, adj=0.111, (0 split)
      citric.acid          < 0.72     to the left,  agree=0.719, adj=0.074, (0 split)

Node number 5: 112 observations,    complexity param=0.02145923
  predicted class=medium  expected loss=0.2857143  P(node) =0.08756841
    class counts:    30     2    80
   probabilities: 0.268 0.018 0.714
  left son=10 (29 obs) right son=11 (83 obs)
  Primary splits:
      total.sulfur.dioxide < 15.5     to the left,  improve=6.697638, (0 missing)
      free.sulfur.dioxide  < 7.5      to the left,  improve=5.081589, (0 missing)
      pH                   < 3.275    to the left,  improve=4.984568, (0 missing)
      citric.acid          < 0.265    to the right, improve=4.472894, (0 missing)
      residual.sugar       < 3.9      to the right, improve=3.386447, (0 missing)
  Surrogate splits:
      free.sulfur.dioxide < 7.5      to the left,  agree=0.920, adj=0.690, (0 split)
      chlorides           < 0.1225   to the right, agree=0.768, adj=0.103, (0 split)
      residual.sugar      < 5.85     to the right, agree=0.750, adj=0.034, (0 split)

Node number 6: 254 observations,    complexity param=0.01287554
  predicted class=medium  expected loss=0.2519685  P(node) =0.1985927
    class counts:    60     4   190
   probabilities: 0.236 0.016 0.748
  left son=12 (118 obs) right son=13 (136 obs)
  Primary splits:
      alcohol              < 10.45    to the right, improve=14.223290, (0 missing)
      chlorides            < 0.0755   to the left,  improve= 8.651214, (0 missing)
      sulphates            < 0.715    to the right, improve= 7.886671, (0 missing)
      total.sulfur.dioxide < 49.5     to the left,  improve= 6.499420, (0 missing)
```

```
      density              < 0.99554  to the left,  improve= 6.368964, (0 missing)
  Surrogate splits:
      density              < 0.996785 to the left,  agree=0.669, adj=0.288, (0 split)
      chlorides            < 0.0705   to the left,  agree=0.650, adj=0.246, (0 split)
      sulphates            < 0.705    to the right, agree=0.618, adj=0.178, (0 split)
      citric.acid          < 0.315    to the right, agree=0.602, adj=0.144, (0 split)
      total.sulfur.dioxide < 47.5     to the left,  agree=0.594, adj=0.127, (0 split)

Node number 7: 824 observations,    complexity param=0.01144492
  predicted class=medium  expected loss=0.08980583  P(node) =0.6442533
    class counts:    29    45   750
   probabilities: 0.035 0.055 0.910
  left son=14 (65 obs) right son=15 (759 obs)
  Primary splits:
      volatile.acidity     < 0.8375   to the right, improve=6.968096, (0 missing)
      total.sulfur.dioxide < 14.5     to the left,  improve=4.245303, (0 missing)
      citric.acid          < 0.105    to the left,  improve=2.525045, (0 missing)
      free.sulfur.dioxide  < 6.5      to the left,  improve=2.250605, (0 missing)
      residual.sugar       < 1.55     to the left,  improve=2.026013, (0 missing)
  Surrogate splits:
      chlorides            < 0.048    to the left,  agree=0.925, adj=0.046, (0 split)
      fixed.acidity        < 5.15     to the left,  agree=0.924, adj=0.031, (0 split)
      total.sulfur.dioxide < 144.5    to the right, agree=0.924, adj=0.031, (0 split)
      pH                   < 3.73     to the right, agree=0.924, adj=0.031, (0 split)

Node number 8: 62 observations
  predicted class=high    expected loss=0.1935484  P(node) =0.04847537
    class counts:    50     0    12
   probabilities: 0.806 0.000 0.194

Node number 9: 27 observations,    complexity param=0.01716738
  predicted class=medium  expected loss=0.4814815  P(node) =0.02111024
    class counts:    13     0    14
   probabilities: 0.481 0.000 0.519
  left son=18 (11 obs) right son=19 (16 obs)
  Primary splits:
      free.sulfur.dioxide  < 27.5     to the right, improve=4.208754, (0 missing)
      pH                   < 3.37     to the left,  improve=3.333754, (0 missing)
      total.sulfur.dioxide < 56.5     to the left,  improve=3.114815, (0 missing)
      sulphates            < 0.755    to the right, improve=1.814815, (0 missing)
      density              < 0.995365 to the left,  improve=1.514449, (0 missing)
  Surrogate splits:
      pH               < 3.345    to the left,  agree=0.741, adj=0.364, (0 split)
      sulphates        < 0.81     to the right, agree=0.741, adj=0.364, (0 split)
      volatile.acidity < 0.415    to the right, agree=0.704, adj=0.273, (0 split)
      citric.acid      < 0.445    to the right, agree=0.704, adj=0.273, (0 split)
      fixed.acidity    < 9.15     to the right, agree=0.667, adj=0.182, (0 split)

Node number 10: 29 observations,    complexity param=0.02145923
  predicted class=high    expected loss=0.4482759  P(node) =0.02267396
    class counts:    16     1    12
   probabilities: 0.552 0.034 0.414
  left son=20 (14 obs) right son=21 (15 obs)
  Primary splits:
      sulphates            < 0.585    to the right, improve=4.543842, (0 missing)
      chlorides            < 0.0665   to the right, improve=2.484102, (0 missing)
      density              < 0.99413  to the left,  improve=1.656624, (0 missing)
```

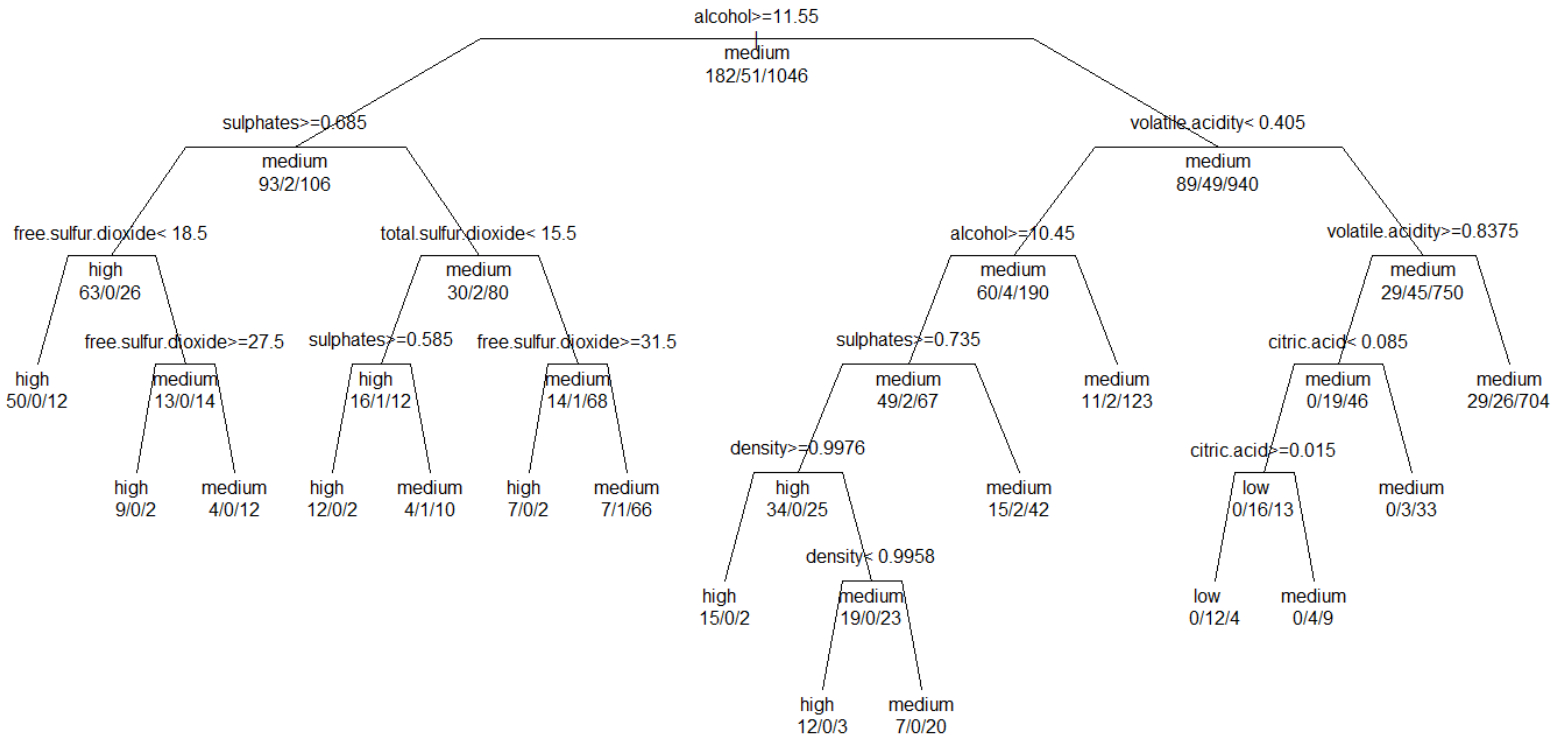
```
      total.sulfur.dioxide < 9        to the right, improve=1.497089, (0 missing)
      residual.sugar       < 3.9      to the right, improve=1.029557, (0 missing)
  Surrogate splits:
      chlorides        < 0.072    to the right, agree=0.759, adj=0.500, (0 split)
      residual.sugar   < 2.05     to the right, agree=0.724, adj=0.429, (0 split)
      volatile.acidity < 0.445    to the right, agree=0.655, adj=0.286, (0 split)
      density          < 0.995855 to the right, agree=0.655, adj=0.286, (0 split)
      fixed.acidity    < 8.25     to the right, agree=0.621, adj=0.214, (0 split)

Node number 11: 83 observations,    complexity param=0.02145923
  predicted class=medium  expected loss=0.1807229  P(node) =0.06489445
    class counts:    14     1    68
   probabilities: 0.169 0.012 0.819
  left son=22 (9 obs) right son=23 (74 obs)
  Primary splits:
      free.sulfur.dioxide  < 31.5     to the right, improve=7.345092, (0 missing)
      pH                   < 3.27     to the left,  improve=3.073070, (0 missing)
      total.sulfur.dioxide < 99.5     to the right, improve=2.408144, (0 missing)
      volatile.acidity     < 0.355    to the left,  improve=2.397998, (0 missing)
      residual.sugar       < 4.15     to the right, improve=1.999179, (0 missing)
  Surrogate splits:
      total.sulfur.dioxide < 99.5     to the right, agree=0.952, adj=0.556, (0 split)
      density              < 0.991555 to the left,  agree=0.928, adj=0.333, (0 split)
      pH                   < 3.035    to the left,  agree=0.928, adj=0.333, (0 split)
      chlorides            < 0.023    to the left,  agree=0.916, adj=0.222, (0 split)
      sulphates            < 0.395    to the left,  agree=0.916, adj=0.222, (0 split)

Node number 12: 118 observations,    complexity param=0.01287554
  predicted class=medium  expected loss=0.4322034  P(node) =0.09225958
    class counts:    49     2    67
   probabilities: 0.415 0.017 0.568
  left son=24 (59 obs) right son=25 (59 obs)
  Primary splits:
      sulphates            < 0.735    to the right, improve=5.542373, (0 missing)
      total.sulfur.dioxide < 50       to the left,  improve=3.812779, (0 missing)
      chlorides            < 0.0565   to the left,  improve=3.398901, (0 missing)
      pH                   < 3.265    to the left,  improve=2.145792, (0 missing)
      density              < 0.99757  to the right, improve=2.046632, (0 missing)
  Surrogate splits:
      total.sulfur.dioxide < 20.5     to the right, agree=0.644, adj=0.288, (0 split)
      volatile.acidity     < 0.335    to the left,  agree=0.619, adj=0.237, (0 split)
      free.sulfur.dioxide  < 7.5      to the right, agree=0.619, adj=0.237, (0 split)
      alcohol              < 11.05    to the left,  agree=0.619, adj=0.237, (0 split)
      density              < 0.99491  to the right, agree=0.610, adj=0.220, (0 split)

Node number 13: 136 observations
  predicted class=medium  expected loss=0.09558824  P(node) =0.1063331
    class counts:    11     2   123
   probabilities: 0.081 0.015 0.904

Node number 14: 65 observations,    complexity param=0.01144492
  predicted class=medium  expected loss=0.2923077  P(node) =0.05082095
    class counts:     0    19    46
   probabilities: 0.000 0.292 0.708
  left son=28 (29 obs) right son=29 (36 obs)
  Primary splits:
      citric.acid          < 0.085    to the left,  improve=7.047480, (0 missing)
```

```
      free.sulfur.dioxide  < 6.5      to the left,  improve=5.465641, (0 missing)
      total.sulfur.dioxide < 19.5     to the left,  improve=5.060156, (0 missing)
      pH                   < 3.28     to the right, improve=4.936752, (0 missing)
      fixed.acidity        < 7.65     to the left,  improve=3.798438, (0 missing)
  Surrogate splits:
      total.sulfur.dioxide < 34       to the left,  agree=0.769, adj=0.483, (0 split)
      fixed.acidity        < 7.65     to the left,  agree=0.723, adj=0.379, (0 split)
      free.sulfur.dioxide  < 6.5      to the left,  agree=0.692, adj=0.310, (0 split)
      pH                   < 3.38     to the right, agree=0.692, adj=0.310, (0 split)
      density              < 0.99671  to the left,  agree=0.662, adj=0.241, (0 split)

Node number 15: 759 observations
  predicted class=medium  expected loss=0.07246377  P(node) =0.5934324
    class counts:    29    26   704
   probabilities: 0.038 0.034 0.928

Node number 18: 11 observations
  predicted class=high    expected loss=0.1818182  P(node) =0.008600469
    class counts:     9     0     2
   probabilities: 0.818 0.000 0.182

Node number 19: 16 observations
  predicted class=medium  expected loss=0.25  P(node) =0.01250977
    class counts:     4     0    12
   probabilities: 0.250 0.000 0.750

Node number 20: 14 observations
  predicted class=high    expected loss=0.1428571  P(node) =0.01094605
    class counts:    12     0     2
   probabilities: 0.857 0.000 0.143

Node number 21: 15 observations
  predicted class=medium  expected loss=0.3333333  P(node) =0.01172791
    class counts:     4     1    10
   probabilities: 0.267 0.067 0.667

Node number 22: 9 observations
  predicted class=high    expected loss=0.2222222  P(node) =0.007036747
    class counts:     7     0     2
   probabilities: 0.778 0.000 0.222

Node number 23: 74 observations
  predicted class=medium  expected loss=0.1081081  P(node) =0.0578577
    class counts:     7     1    66
   probabilities: 0.095 0.014 0.892

Node number 24: 59 observations,    complexity param=0.01287554
  predicted class=high    expected loss=0.4237288  P(node) =0.04612979
    class counts:    34     0    25
   probabilities: 0.576 0.000 0.424
  left son=48 (17 obs) right son=49 (42 obs)
  Primary splits:
      density              < 0.99757  to the right, improve=4.474624, (0 missing)
      chlorides            < 0.061    to the left,  improve=2.523763, (0 missing)
      total.sulfur.dioxide < 56.5     to the left,  improve=2.405533, (0 missing)
      sulphates            < 0.765    to the left,  improve=1.651795, (0 missing)
      free.sulfur.dioxide  < 21.5     to the left,  improve=1.422582, (0 missing)
```

```
  Surrogate splits:
      residual.sugar      < 2.55     to the right, agree=0.847, adj=0.471, (0 split)
      fixed.acidity       < 9.95     to the right, agree=0.814, adj=0.353, (0 split)
      citric.acid         < 0.575    to the right, agree=0.763, adj=0.176, (0 split)
      free.sulfur.dioxide < 7.5      to the left,  agree=0.763, adj=0.176, (0 split)
      sulphates           < 0.745    to the left,  agree=0.763, adj=0.176, (0 split)

Node number 25: 59 observations
  predicted class=medium  expected loss=0.2881356  P(node) =0.04612979
    class counts:    15     2    42
   probabilities: 0.254 0.034 0.712

Node number 28: 29 observations,    complexity param=0.01144492
  predicted class=low     expected loss=0.4482759  P(node) =0.02267396
    class counts:     0    16    13
   probabilities: 0.000 0.552 0.448
  left son=56 (16 obs) right son=57 (13 obs)
  Primary splits:
      citric.acid         < 0.015    to the right, improve=2.806366, (0 missing)
      free.sulfur.dioxide < 5.5      to the left,  improve=1.881670, (0 missing)
      volatile.acidity    < 0.97     to the right, improve=1.609533, (0 missing)
      alcohol             < 10.95    to the right, improve=1.333716, (0 missing)
      fixed.acidity       < 6.65     to the right, improve=1.244828, (0 missing)
  Surrogate splits:
      chlorides        < 0.08     to the left,  agree=0.690, adj=0.308, (0 split)
      density          < 0.99558  to the right, agree=0.690, adj=0.308, (0 split)
      pH               < 3.545    to the left,  agree=0.690, adj=0.308, (0 split)
      fixed.acidity    < 6.45     to the right, agree=0.655, adj=0.231, (0 split)
      volatile.acidity < 1.1375   to the left,  agree=0.655, adj=0.231, (0 split)

Node number 29: 36 observations
  predicted class=medium  expected loss=0.08333333  P(node) =0.02814699
    class counts:     0     3    33
   probabilities: 0.000 0.083 0.917

Node number 48: 17 observations
  predicted class=high    expected loss=0.1176471  P(node) =0.01329163
    class counts:    15     0     2
   probabilities: 0.882 0.000 0.118

Node number 49: 42 observations,    complexity param=0.01287554
  predicted class=medium  expected loss=0.452381  P(node) =0.03283815
    class counts:    19     0    23
   probabilities: 0.452 0.000 0.548
  left son=98 (15 obs) right son=99 (27 obs)
  Primary splits:
      density     < 0.9958   to the left,  improve=5.639153, (0 missing)
      chlorides   < 0.061    to the left,  improve=3.530112, (0 missing)
      pH          < 3.39     to the right, improve=3.172024, (0 missing)
      citric.acid < 0.47     to the left,  improve=2.742857, (0 missing)
      alcohol     < 10.95    to the right, improve=2.484581, (0 missing)
  Surrogate splits:
      fixed.acidity    < 7.25     to the left,  agree=0.738, adj=0.267, (0 split)
      volatile.acidity < 0.355    to the right, agree=0.738, adj=0.267, (0 split)
      chlorides        < 0.061    to the left,  agree=0.738, adj=0.267, (0 split)
      pH               < 3.2      to the left,  agree=0.690, adj=0.133, (0 split)
      alcohol          < 10.95    to the right, agree=0.667, adj=0.067, (0 split)
```

```
Node number 56: 16 observations
  predicted class=low     expected loss=0.25  P(node) =0.01250977
    class counts:     0    12     4
   probabilities: 0.000 0.750 0.250

Node number 57: 13 observations
  predicted class=medium  expected loss=0.3076923  P(node) =0.01016419
    class counts:     0     4     9
   probabilities: 0.000 0.308 0.692

Node number 98: 15 observations
  predicted class=high    expected loss=0.2  P(node) =0.01172791
    class counts:    12     0     3
   probabilities: 0.800 0.000 0.200

Node number 99: 27 observations
  predicted class=medium  expected loss=0.2592593  P(node) =0.02111024
    class counts:     7     0    20
   probabilities: 0.259 0.000 0.741
```

It is more convenient to present this information by visualizing the tree. I used the uniform tree presentation and displayed all textual information on the tree:


alcohol>=11.55
medium
182/51/1046
sulphates>=0.685
medium
93/2/106
volatile.acidity< 0.405
medium
89/49/940
free.sulfur.dioxide< 18.5
high
63/0/26
total.sulfur.dioxide< 15.5
medium
30/2/80
alcohol>=10.45
medium
60/4/190
volatile.acidity>=0.8375
medium
29/45/750
high
50/0/12
free.sulfur.dioxide>=27.5
medium
13/0/14
sulphates>=0.585
high
16/1/12
free.sulfur.dioxide>=31.5
medium
14/1/68
sulphates>=0.735
medium
49/2/67
medium
11/2/123
citric.acid< 0.085
medium
0/19/46
medium
29/26/704
high
9/0/2
medium
4/0/12
high
12/0/2
medium
4/1/10
high
7/0/2
medium
7/1/66
density>=0.9976
high
34/0/25
medium
15/2/42
citric.acid>=0.015
low
0/16/13
medium
0/3/33
density< 0.9958
high
15/0/2
medium
19/0/23
low
0/12/4
medium
0/4/9
high
12/0/3
medium
7/0/20

Next step is to use the model to generate predictions on the testing dataset.

```
> ###Evaluate the model predictive performance
> #Generate predictions
> predictions <- predict(winequality_rp, redwine_test, type="class")
> table(redwine_test$quality, predictions)
        predictions
         high low medium
  high     17   0     18
  low       0   0     12
  medium   13   5    255
```

The classification table above shows that out of 320 test observation the model correctly predicted 17 high quality samples, and 255 medium quality samples. At the same time 18 high quality samples were classified as medium quality, 12 low quality samples were included in the medium category and 13 medium samples were classified as high quality, when 5 medium samples were included in the low-quality category.

A more detailed analysis of predictive performance of the classification model can be done using confusion matrix from the caret package:

```
> #generate confusion matrix using caret package
> confusionMatrix(table(predictions, redwine_test$quality))
Confusion Matrix and Statistics


predictions high low medium
     high     17   0     13
     low       0   0      5
     medium   18  12    255

Overall Statistics

               Accuracy : 0.85
                 95% CI : (0.8061, 0.8873)
    No Information Rate : 0.8531
    P-Value [Acc > NIR] : 0.6004

                  Kappa : 0.346
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: high Class: low Class: medium
Sensitivity              0.48571    0.00000        0.9341
```

```
Specificity             0.95439    0.98377        0.3617
Pos Pred Value          0.56667    0.00000        0.8947
Neg Pred Value          0.93793    0.96190        0.4857
Prevalence              0.10938    0.03750        0.8531
Detection Rate          0.05312    0.00000        0.7969
Detection Prevalence    0.09375    0.01562        0.8906
Balanced Accuracy       0.72005    0.49188        0.6479
```

The above output shows that the overall accuracy of the model is 85%. At the same time the low wine quality category has the lowest balanced accuracy of 49% ((sensitivity + specificity)/2) compared to 72% for high quality wines and about 65% for medium quality wines.

One way to improve the classification model would be to prune the tree. However, previously plotcp(winequality_rp) command demonstrated that the minimum error was obtained on the maximum size tree, so pruning based on minimum cross validation error will not work in this case.

In order to check, I proceeded to find the minimum cross-validation error:

```
> ###Pruning the tree
> #Find minimum cross-validation error
> min(winequality_rp$cptable[ ,"xerror"])
[1] 0.888412
```

And the record that had the minimum error:

```
> #Find the record with the minimum cross-validation error
> which.min(winequality_rp$cptable[ ,"xerror"])
6
```

I used the 6th record to estimate the cost complexity parameter and prune the tree based on it:

```
> #Get the cost complexity parameter of the record with min cross-validation error
> winequality_rp.cp <- winequality_rp$cptable[6, "CP"]
> #Prune the tree
> winequality_prune <- prune(winequality_rp, cp=winequality_rp.cp)
```

As the summary() command's sample output below shows, it resulted in the exactly the same model:

```
> #Examine the pruned model
> summary(winequality_prune)
Call:
rpart(formula = quality ~ ., data = redwine_train)
  n= 1279

          CP nsplit rel error    xerror       xstd
1 0.07939914      0 1.0000000 1.0000000 0.05924512
2 0.02145923      2 0.8412017 0.9356223 0.05771580
3 0.01716738      5 0.7768240 0.9656652 0.05844137
4 0.01287554      7 0.7424893 0.9570815 0.05823622
5 0.01144492     12 0.6480687 0.9012876 0.05686006
6 0.01000000     15 0.6137339 0.8884120 0.05653161

Variable importance
             alcohol            sulphates              density     volatile.acidity  free.sulfur.
dioxide total.sulfur.dioxide          citric.acid
                  23                   13                   13                   11
9                    8                    7
       fixed.acidity            chlorides                   pH       residual.sugar
                   6                    6                    4                    2

Node number 1: 1279 observations,    complexity param=0.07939914
  predicted class=medium  expected loss=0.1821736  P(node) =1
    class counts:   182    51  1046
   probabilities: 0.142 0.040 0.818
  left son=2 (201 obs) right son=3 (1078 obs)
  Primary splits:
      alcohol          < 11.55    to the right, improve=44.81302, (0 missing)
      sulphates        < 0.685    to the right, improve=26.88087, (0 missing)
      volatile.acidity < 0.3625   to the left,  improve=25.95614, (0 missing)
      citric.acid      < 0.315    to the right, improve=20.18783, (0 missing)
      density          < 0.99535  to the left,  improve=19.75534, (0 missing)
  Surrogate splits:
      density              < 0.993785 to the left,  agree=0.887, adj=0.279, (0 split)
      chlorides            < 0.0525   to the left,  agree=0.860, adj=0.109, (0 split)
      fixed.acidity        < 5.55     to the left,  agree=0.856, adj=0.085, (0 split)
      pH                   < 3.695    to the right, agree=0.850, adj=0.045, (0 split)
      total.sulfur.dioxide < 162.5    to the right, agree=0.845, adj=0.015, (0 split)

Node number 2: 201 observations,    complexity param=0.07939914
  predicted class=medium  expected loss=0.4726368  P(node) =0.157154
    class counts:    93     2   106
   probabilities: 0.463 0.010 0.527
  left son=4 (89 obs) right son=5 (112 obs)
  Primary splits:
      sulphates        < 0.685    to the right, improve=18.455050, (0 missing)
      pH               < 3.365    to the left,  improve= 9.083775, (0 missing)
      citric.acid      < 0.315    to the right, improve= 7.901565, (0 missing)
      fixed.acidity    < 7.85     to the right, improve= 7.333158, (0 missing)
      volatile.acidity < 0.335    to the left,  improve= 4.739165, (0 missing)
  Surrogate splits:
```

```
      fixed.acidity        < 8.05     to the right, agree=0.647, adj=0.202, (0 split)
      citric.acid          < 0.325    to the right, agree=0.632, adj=0.169, (0 split)
      volatile.acidity     < 0.345    to the left,  agree=0.627, adj=0.157, (0 split)
      alcohol              < 12.85    to the right, agree=0.612, adj=0.124, (0 split)
      total.sulfur.dioxide < 51       to the right, agree=0.607, adj=0.112, (0 split)
```

(the rest of the output is omitted for brevity)

Naturally, the tree visualization and confusion matrix resulted in the same output.

```
> #Visualize the prunned model
> #using the uniform tree presentation
> plot(winequality_prune, uniform = TRUE, branch = 0.6, margin = 0.1)
> text(winequality_prune, all=TRUE, use.n=TRUE)
> #Generate classification table
> predictions <- predict(winequality_prune, redwine_test, type = "class")
> table(redwine_test$quality, predictions)
        predictions
         high low medium
  high     17   0     18
  low       0   0     12
  medium   13   5    255
>
> #Grenrate confusion matrix
> confusionMatrix(table(predictions, redwine_test$quality))
Confusion Matrix and Statistics


predictions high low medium
     high     17   0     13
     low       0   0      5
     medium   18  12    255

Overall Statistics

               Accuracy : 0.85
                 95% CI : (0.8061, 0.8873)
    No Information Rate : 0.8531
    P-Value [Acc > NIR] : 0.6004

                  Kappa : 0.346
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: high Class: low Class: medium
Sensitivity              0.48571    0.00000        0.9341
Specificity              0.95439    0.98377        0.3617
Pos Pred Value           0.56667    0.00000        0.8947
Neg Pred Value           0.93793    0.96190        0.4857
Prevalence               0.10938    0.03750        0.8531
Detection Rate           0.05312    0.00000        0.7969
Detection Prevalence     0.09375    0.01562        0.8906
Balanced Accuracy        0.72005    0.49188        0.6479

>
```

In an effort to improve the predictive power, I tried to use the random forest approach. I fit the random forest classifier with the training set using the randomForest() command and displayed its results:

```
> #######Random Forest##############
```

```
> winequality_rf <- randomForest(quality ~ ., data = redwine_train, importance = T)
>
> #display model details
> winequality_rf

Call:
 randomForest(formula = quality ~ ., data = redwine_train, importance = T)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 13.29%
Confusion matrix:
       high low medium class.error
high     99   0     83  0.45604396
low       1   2     48  0.96078431
medium   34   4   1008  0.03632887
```

The estimated error rate for the random forest model shows some improvement over the previously constructed recursive partitioning model. However, the real test is to use the model on the previously unseen data. So, I used this model to generate predictions on the testing data set next:

```
> #Make predictions
> predictions_rf <- predict(winequality_rf, redwine_test)
>
> #Classification table
> table(predictions_rf, redwine_test$quality)

predictions_rf high low medium
        high     24   0     14
        low       0   0      0
        medium   11  12    259
```

The classification table above shows that 24 high quality samples and 259 medium samples were classified correctly, when overall 37 samples (11+12+14) were misclassified (or 11.56% of the testing dataset).

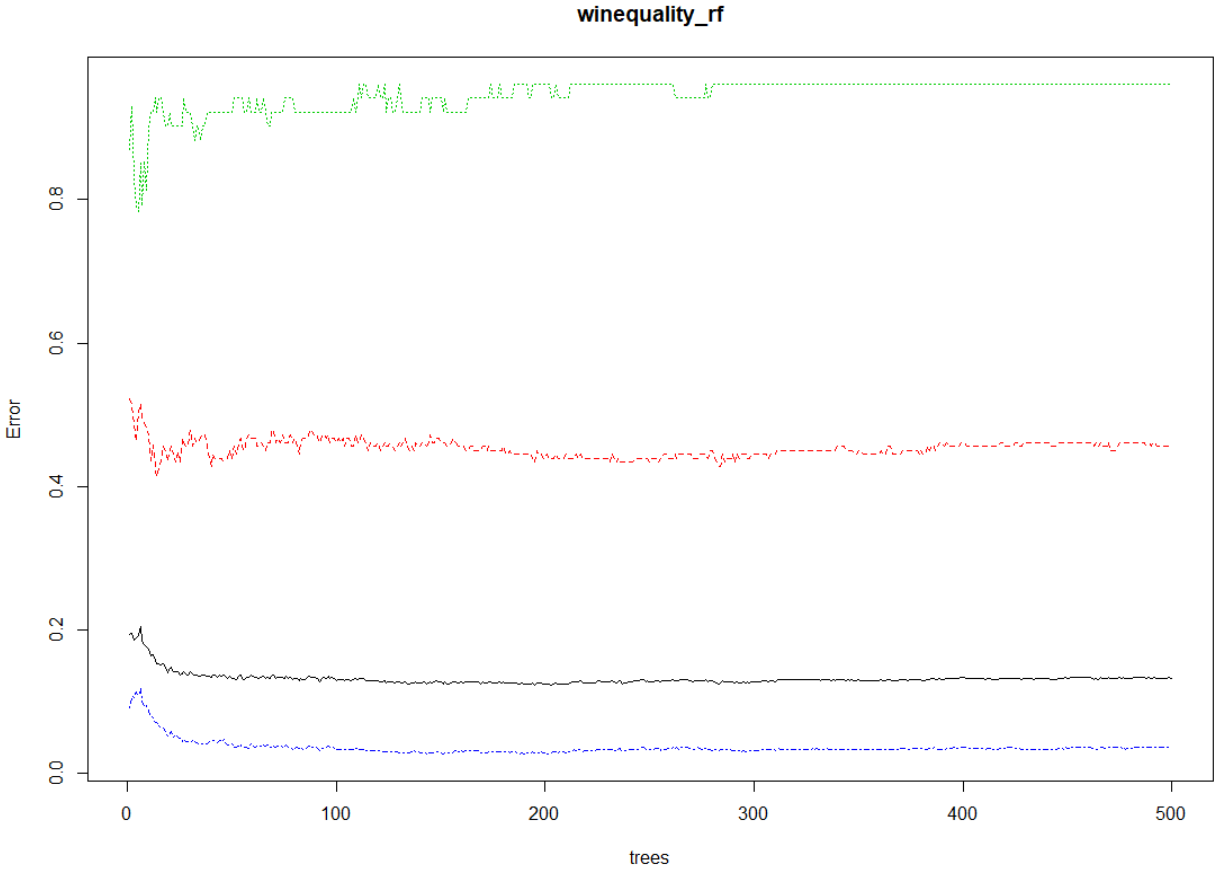
The below output of the confusion matrix shows that the random forest model has an improved overall accuracy of 88.44% (compared to 85% for the recursive partitioning tree). The balanced accuracy also improved for all three classes, however it remains low for the low quality wines.

```
> #Confusion matrix
```

```
> confusionMatrix(table(predictions_rf, redwine_test$quality))
Confusion Matrix and Statistics


predictions_rf high low medium
        high     24   0     14
        low       0   0      0
        medium   11  12    259

Overall Statistics

               Accuracy : 0.8844
                 95% CI : (0.8442, 0.9173)
    No Information Rate : 0.8531
    P-Value [Acc > NIR] : 0.06341

                  Kappa : 0.5084
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: high Class: low Class: medium
Sensitivity               0.6857     0.0000        0.9487
Specificity               0.9509     1.0000        0.5106
Pos Pred Value            0.6316        NaN        0.9184
Neg Pred Value            0.9610     0.9625        0.6316
Prevalence                0.1094     0.0375        0.8531
Detection Rate            0.0750     0.0000        0.8094
Detection Prevalence      0.1187     0.0000        0.8812
Balanced Accuracy         0.8183     0.5000        0.7297
```

The mean square error of the forest object is plotted below:

```
> #Plot the mean square error of the forest object
> plot(winequality_rf)
```

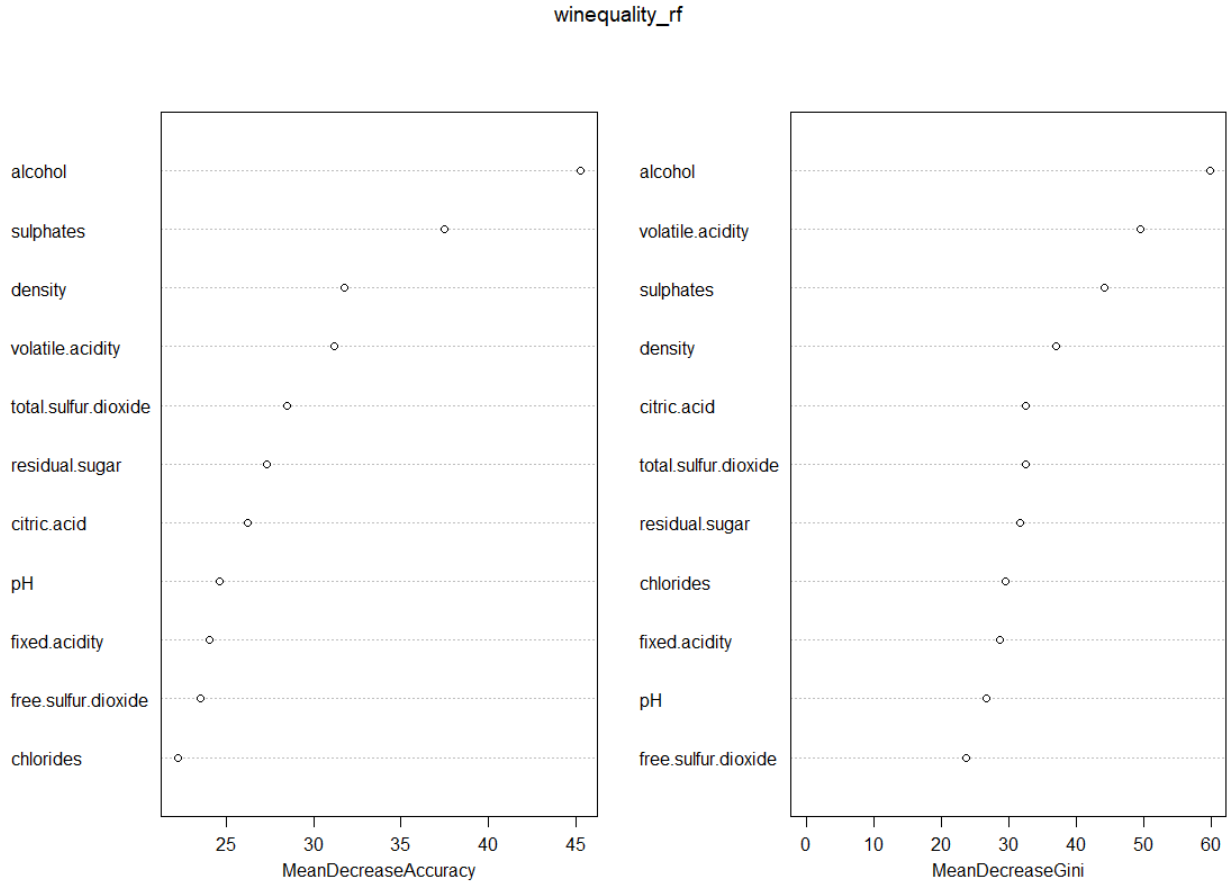


The randomForrest package allow us to analyze the most influential factors in the model using the importance() function:

```
> #Examine importance of each attribute
> importance(winequality_rf)
                        high       low   medium MeanDecreaseAccuracy MeanDecreaseGini
fixed.acidity       17.54848 -3.038072 18.67693             24.03245         28.55228
volatile.acidity    30.83032 19.299514 12.00160             31.16631         49.40767
citric.acid         22.00919  7.790991 15.85684             26.20611         32.51236
residual.sugar      19.42921  3.930668 21.47767             27.31235         31.62724
chlorides           13.77201 -3.212559 19.14726             22.21225         29.50768
free.sulfur.dioxide 13.79151  3.297954 19.71502             23.52889         23.60979
total.sulfur.dioxide 24.62445  5.058901 21.10894             28.49398         32.43242
density             23.29300 -2.489272 26.05187             31.76683         36.92529
pH                  17.51735  4.519130 18.66730             24.59749         26.68204
sulphates           42.25236  8.857301 17.37636             37.48310         44.07924
alcohol             45.45322  3.330081 27.25061             45.30108         59.74893
```

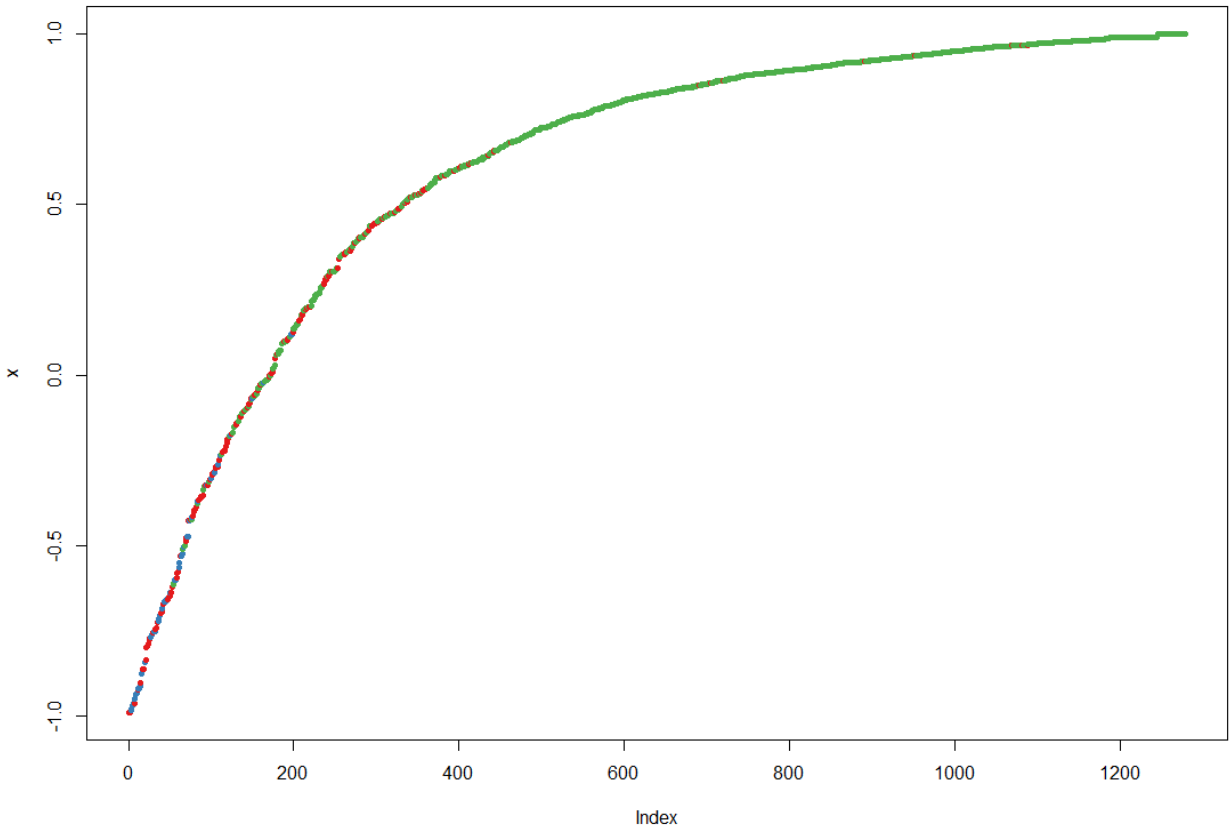
The top three most important for classification attributes are alcohol, volatile.acidity and sulphates as they have the highest mean decrease in Gini index. The variables' importance can be presented visually using varImpPlot() function:

```
> #Plot varaible importance
> varImpPlot(winequality_rf)
```



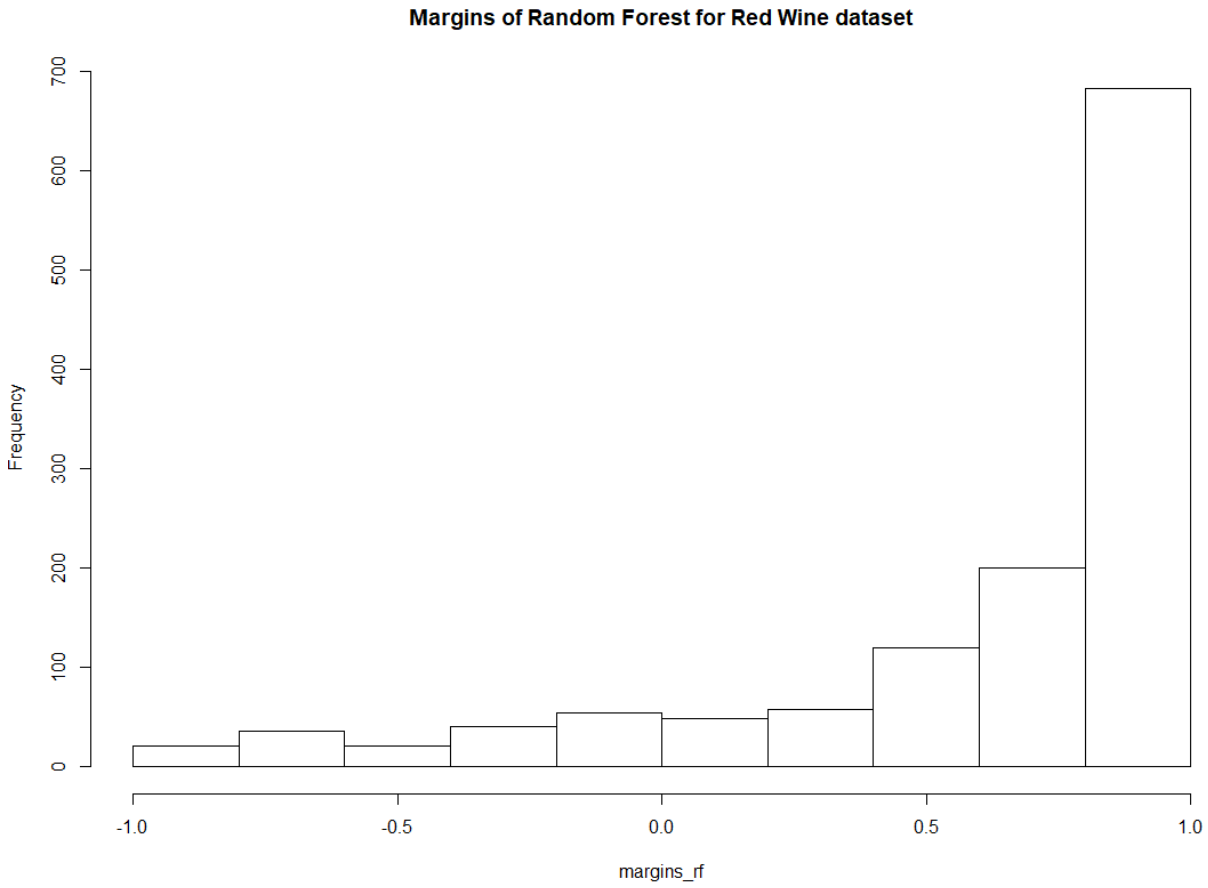
In order to assess certainty of the classification, we should look at the margins, which are calculated as the difference between the support of the correct class and the maximum support for the incorrect class. Correctly classified examples have positive margins and incorrectly classified examples have negative margins. The closer the margin is to one, the higher the degree of confidence for the classification. Low margins imply uncertain classification.

```
> #calculate margins and plot the margin cumulative distribution
> margins_rf <- margin(winequality_rf, redwine_train)
> plot(margins_rf)
```



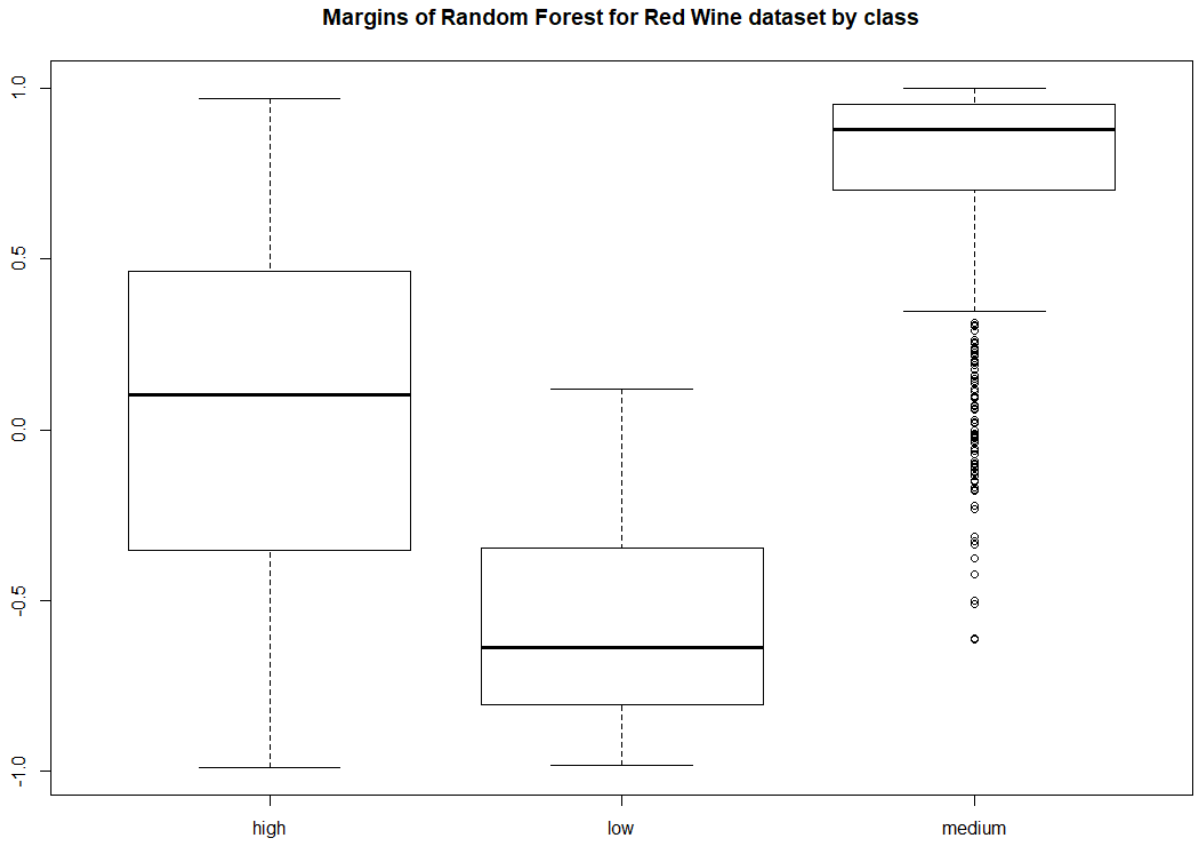
The above plot shows, for about of 400 training observations out of 1279 the margin was below 0.5 suggesting uncertain classifications. The histogram of margin distributions confirms that only about 700 training observations were classified reliably.

```
> #Visualize margin distribution
> hist(margins_rf, main="Margins of Random Forest for Red Wine dataset")
```



Visualizing margin by class of wine quality (see below) shows that the medium wine quality group had the margins closest to 1, but at the same time it had the highest number of outliers (in this case samples that could not be classified reliably). The low wine quality group had the lowest margins, followed by the high wine quality group.

```
> #Visualize margins by class
> boxplot(margins_rf ~ redwine_train$quality, main = "Margins of Random Forest for Red Wine datas
et by class")
```



Overall, the random forest classifier yielded more accurate predictions compared to the recursive partitioning tree. However, the predictive accuracy remained lower for the two extreme classes – low and high quality wines. It can partially be explained by how the quality ratings were collected for the data set as a median of at least three subjective evaluations made by wine experts. As a result, the ratings show evidence of an extremity avoidance bias, when a medium quality classification becomes a de-facto default classification.

In order to further improve the predictive power of the model, since no additional samples (observations) for the two extreme classes are available, it might be necessary to focus on the most important attributes for those classes separately. It might happen that the attributes that were not significant for the model as a whole might have more relative significance for the extreme classes. In this case, they need to intentionally be included in the model. In addition, fine tuning the parameters for the random forest algorithm (ex., number of trees, depth of trees and the number of features used for a split) can also be helpful.