DTWineQualityCART.R

ai

Mon Jun 5 18:33:58 2017

```r
# Reference for data source (
# @misc{Lichman:2013 ,
# author = "M. Lichman",
# year = "2013",
# title = "{UCI} Machine Learning Repository",
# url = "http://archive.ics.uci.edu/ml",
# institution = "University of California, Irvine, School of Information and Computer Sciences" })

# Decision Trees
# Source of Data Set:- UCI Repository - Wine Quality Data(https://archive.ics.uci.edu/ml/datasets/wine+

# required libraries
# # The rpart package can be installed via the install.packages("rpart") and
# # loaded with the library(rpart) command.
library(rpart) #recursive and partitioning trees

# # The plotly package can be installed via the install.packages("plotly") and
# # loaded with the library(plotly) command.
library(plotly) #data visualization
```

```
## Loading required package: ggplot2

##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```r
# # The rpart.plot package can be installed via the install.packages("rpart.plot") and
# # loaded with the library(rpart.plot) command.
library(rpart.plot)

# # The rattle package can be installed via the install.packages("rattle") and
# # loaded with the library(rattle) command.
library(rattle)
```

```
## Rattle: A free graphical interface for data mining with R.
## Version 5.0.8 Copyright (c) 2006-2017 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```r
# # The RColorBrewer package can be installed via the install.packages("RColorBrewer") and
# # loaded with the library(RColorBrewer) command.
library(RColorBrewer)


# # The RWeka package can be installed via the install.packages("RWeka") and
# # loaded with the library(RWeka) command.
library(RWeka)


# Exploring and preparing the data
# Step 2: Exploring and preparing the data
# Read the csv file into a data frame titled WineData.
WineData <- read.table("winequality-red.csv", sep=";", header=TRUE)



WineData$quality <- ifelse(WineData$quality < 5, 'bad', ifelse(WineData$quality > 6,'good','normal'))
WineData$quality <- as.factor(WineData$quality)
str(WineData$quality)
```

```
##  Factor w/ 3 levels "bad","good","normal": 3 3 3 3 3 3 3 2 2 3 ...
```

```r
# Data preparation - creating random training and test datasets
# Create random sample
# Divide the data into a training set and a test set randomly with ratio 80:20

set.seed(123)
train_sample <- sample(nrow(WineData), 0.8 * nrow(WineData))
WineData_train <- WineData[train_sample, ]
WineData_test <- WineData[-train_sample, ]
# Check whether data set fairly even split
prop.table(table(WineData_train$quality))
```

```
##
##        bad       good     normal
## 0.03909304 0.13995309 0.82095387
```

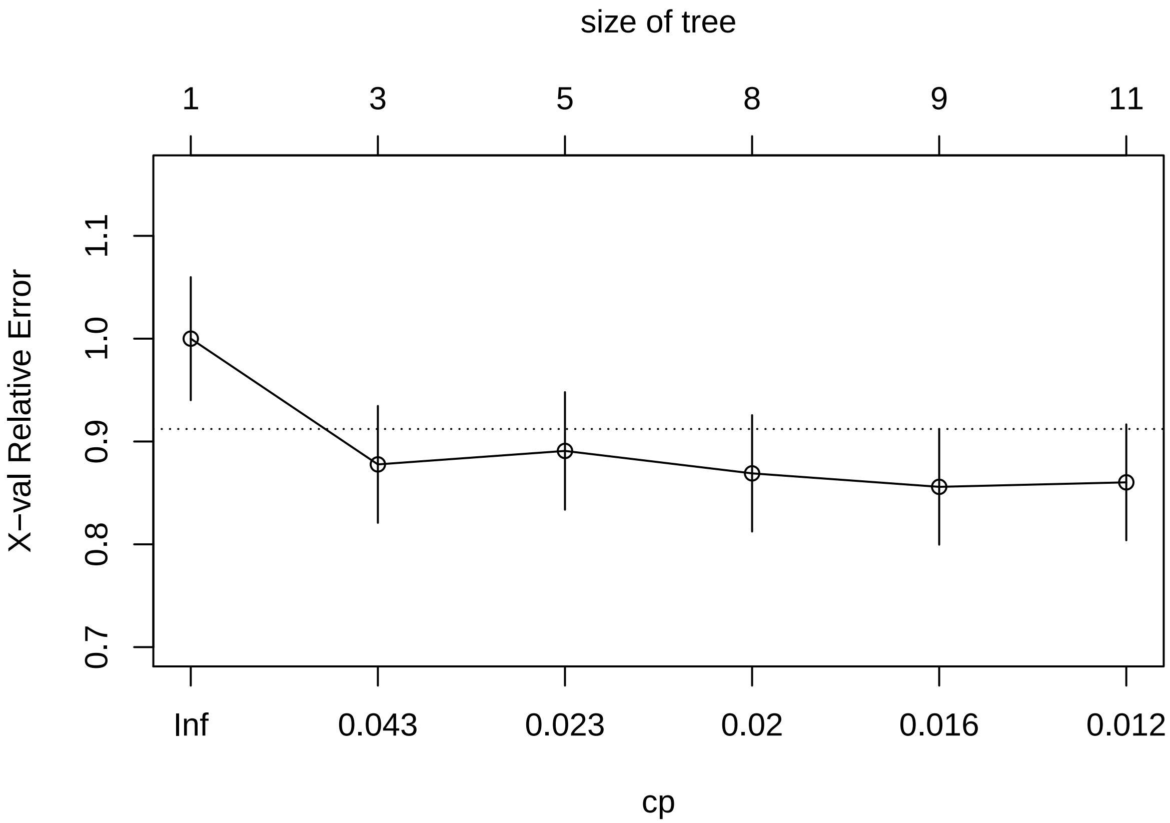
```r
prop.table(table(WineData_test$quality))
```

```
##
##      bad     good   normal
## 0.040625 0.118750 0.840625
```

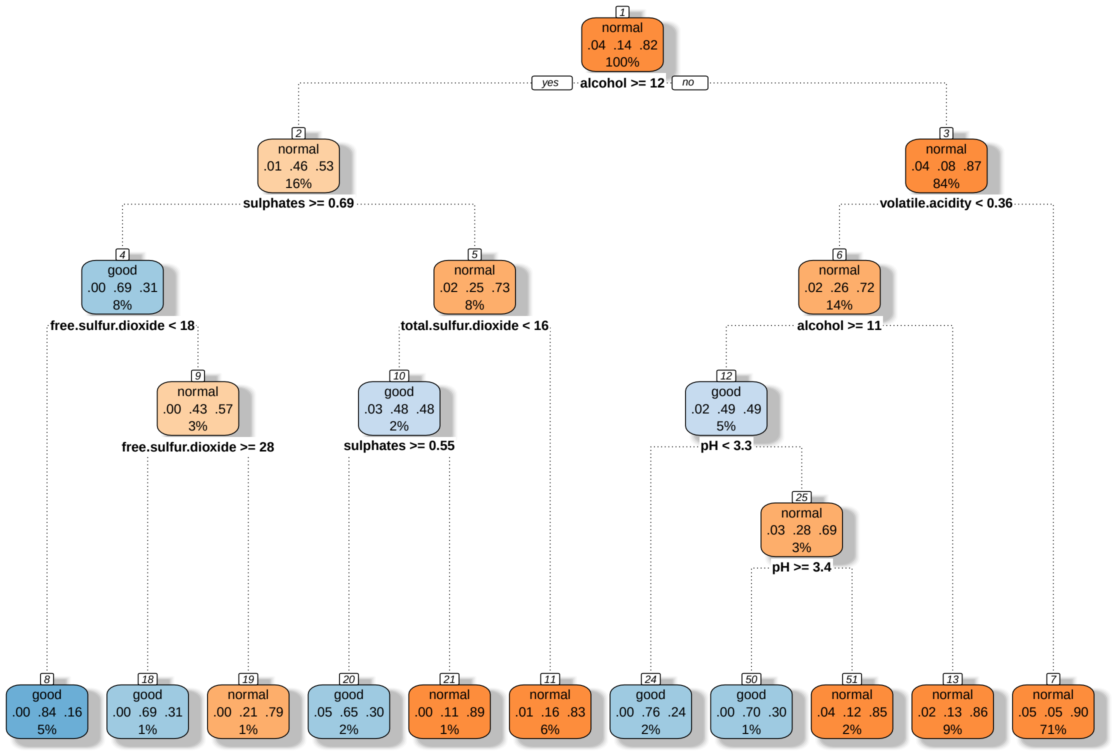
```r
# Train model - Regression Tree
# Build the model with recursive partitioning trees
WineData_model <- rpart(quality ~. , data = WineData_train)

summary(WineData_model)
```

```
## Call:
## rpart(formula = quality ~ ., data = WineData_train)
##   n= 1279
##
##           CP nsplit rel error    xerror       xstd
## 1 0.07860262      0 1.0000000 1.0000000 0.05987446
## 2 0.02401747      2 0.8427948 0.8777293 0.05683772
## 3 0.02183406      4 0.7947598 0.8908297 0.05718058
## 4 0.01746725      7 0.7292576 0.8689956 0.05660668
## 5 0.01528384      8 0.7117904 0.8558952 0.05625634
## 6 0.01000000     10 0.6812227 0.8602620 0.05637363
## 
## Variable importance
##              alcohol              density            sulphates 
##                   27                   12                   11 
##  free.sulfur.dioxide total.sulfur.dioxide     volatile.acidity 
##                   11                    9                    7 
##                   pH        fixed.acidity            chlorides 
##                    7                    7                    4 
##          citric.acid       residual.sugar 
##                    4                    1 
## 
## Node number 1: 1279 observations,    complexity param=0.07860262
##   predicted class=normal  expected loss=0.1790461  P(node) =1
##     class counts:    50   179  1050
##    probabilities: 0.039 0.140 0.821 
##   left son=2 (200 obs) right son=3 (1079 obs)
##   Primary splits:
##       alcohol          < 11.55    to the right, improve=44.55296, (0 missing)
##       sulphates        < 0.675    to the right, improve=23.61681, (0 missing)
##       volatile.acidity < 0.3625   to the left,  improve=21.54164, (0 missing)
##       density          < 0.99537  to the left,  improve=18.29519, (0 missing)
##       citric.acid      < 0.315    to the right, improve=16.76058, (0 missing)
##   Surrogate splits:
##       density          < 0.994185 to the left,  agree=0.890, adj=0.295, (0 split)
##       fixed.acidity    < 5.65     to the left,  agree=0.858, adj=0.090, (0 split)
##       chlorides        < 0.0525   to the left,  agree=0.856, adj=0.080, (0 split)
##       pH               < 3.695    to the right, agree=0.849, adj=0.035, (0 split)
##       volatile.acidity < 0.14     to the left,  agree=0.845, adj=0.010, (0 split)
## 
## Node number 2: 200 observations,    complexity param=0.07860262
##   predicted class=normal  expected loss=0.47  P(node) =0.1563722
##     class counts:     2    92   106
##    probabilities: 0.010 0.460 0.530 
##   left son=4 (96 obs) right son=5 (104 obs)
##   Primary splits:
##       sulphates        < 0.685    to the right, improve=18.306920, (0 missing)
##       pH               < 3.365    to the left,  improve= 4.384906, (0 missing)
##       fixed.acidity    < 7.85     to the right, improve= 4.328485, (0 missing)
##       citric.acid      < 0.315    to the right, improve= 4.013838, (0 missing)
##       volatile.acidity < 0.575    to the left,  improve= 3.174624, (0 missing)
##   Surrogate splits:
##       fixed.acidity        < 7.85     to the right, agree=0.630, adj=0.229, (0 split)
##       citric.acid          < 0.305    to the right, agree=0.615, adj=0.198, (0 split)
##       free.sulfur.dioxide  < 11.5     to the right, agree=0.610, adj=0.187, (0 split)
##       density              < 0.9939   to the right, agree=0.605, adj=0.177, (0 split)
##       total.sulfur.dioxide < 33.5     to the right, agree=0.600, adj=0.167, (0 split)
## 
## Node number 3: 1079 observations,    complexity param=0.02183406
##   predicted class=normal  expected loss=0.1251158  P(node) =0.8436278
##     class counts:    48    87   944
##    probabilities: 0.044 0.081 0.875 
##   left son=6 (176 obs) right son=7 (903 obs)
##   Primary splits:
##       volatile.acidity     < 0.3625   to the left,  improve=11.969700, (0 missing)
##       alcohol              < 10.45    to the right, improve=10.809070, (0 missing)
##       sulphates            < 0.675    to the right, improve= 7.165121, (0 missing)
##       citric.acid          < 0.315    to the right, improve= 5.652792, (0 missing)
##       total.sulfur.dioxide < 49.5     to the left,  improve= 5.632499, (0 missing)
##   Surrogate splits:
##       residual.sugar      < 11.95    to the right, agree=0.839, adj=0.011, (0 split)
##       citric.acid         < 0.71     to the right, agree=0.838, adj=0.006, (0 split)
##       free.sulfur.dioxide < 1.5      to the left,  agree=0.838, adj=0.006, (0 split)
##       pH                  < 2.885    to the left,  agree=0.838, adj=0.006, (0 split)
## 
## Node number 4: 96 observations,    complexity param=0.02401747
##   predicted class=good    expected loss=0.3125  P(node) =0.07505864
##     class counts:     0    66    30
##    probabilities: 0.000 0.688 0.312 
##   left son=8 (61 obs) right son=9 (35 obs)
##   Primary splits:
##       free.sulfur.dioxide  < 18.5     to the left,  improve=7.385831, (0 missing)
##       total.sulfur.dioxide < 56.5     to the left,  improve=4.438406, (0 missing)
##       fixed.acidity        < 11.35    to the left,  improve=2.437801, (0 missing)
##       density              < 0.99533  to the left,  improve=1.493016, (0 missing)
##       pH                   < 3.365    to the left,  improve=1.484192, (0 missing)
##   Surrogate splits:
##       total.sulfur.dioxide < 41.5     to the left,  agree=0.854, adj=0.600, (0 split)
##       alcohol              < 13.8     to the left,  agree=0.688, adj=0.143, (0 split)
##       citric.acid          < 0.72     to the left,  agree=0.667, adj=0.086, (0 split)
##       density              < 0.99176  to the right, agree=0.667, adj=0.086, (0 split)
##       pH                   < 3.59     to the left,  agree=0.667, adj=0.086, (0 split)
## 
## Node number 5: 104 observations,    complexity param=0.01528384
##   predicted class=normal  expected loss=0.2692308  P(node) =0.08131353
##     class counts:     2    26    76
##    probabilities: 0.019 0.250 0.731 
##   left son=10 (29 obs) right son=11 (75 obs)
##   Primary splits:
##       total.sulfur.dioxide < 15.5     to the left,  improve=4.661468, (0 missing)
##       residual.sugar       < 4.8      to the right, improve=3.318910, (0 missing)
##       free.sulfur.dioxide  < 7.5      to the left,  improve=2.763960, (0 missing)
##       pH                   < 3.275    to the left,  improve=2.598077, (0 missing)
##       sulphates            < 0.615    to the right, improve=2.366774, (0 missing)
##   Surrogate splits:
##       free.sulfur.dioxide < 7.5      to the left,  agree=0.885, adj=0.586, (0 split)
##       chlorides           < 0.0925   to the right, agree=0.760, adj=0.138, (0 split)
##       residual.sugar      < 5.85     to the right, agree=0.750, adj=0.103, (0 split)
## 
## Node number 6: 176 observations,    complexity param=0.02183406
##   predicted class=normal  expected loss=0.2784091  P(node) =0.1376075
##     class counts:     3    46   127
##    probabilities: 0.017 0.261 0.722 
##   left son=12 (65 obs) right son=13 (111 obs)
##   Primary splits:
##       alcohol       < 10.75    to the right, improve=10.915280, (0 missing)
##       chlorides     < 0.0755   to the left,  improve= 5.518480, (0 missing)
##       sulphates     < 0.815    to the right, improve= 4.741189, (0 missing)
##       fixed.acidity < 11.65    to the right, improve= 4.446591, (0 missing)
##       pH            < 3.265    to the left,  improve= 4.407279, (0 missing)
##   Surrogate splits:
##       density              < 0.99574  to the left,  agree=0.778, adj=0.400, (0 split)
##       total.sulfur.dioxide < 11.5     to the left,  agree=0.688, adj=0.154, (0 split)
##       chlorides            < 0.0665   to the left,  agree=0.682, adj=0.138, (0 split)
##       citric.acid          < 0.625    to the right, agree=0.648, adj=0.046, (0 split)
##       residual.sugar       < 1.3      to the left,  agree=0.642, adj=0.031, (0 split)
##
## Node number 7: 903 observations
##   predicted class=normal  expected loss=0.0952381  P(node) =0.7060203
##     class counts:    45    41   817
##    probabilities: 0.050 0.045 0.905
##
## Node number 8: 61 observations
##   predicted class=good    expected loss=0.1639344  P(node) =0.04769351
##     class counts:     0    51    10
##    probabilities: 0.000 0.836 0.164
##
## Node number 9: 35 observations,    complexity param=0.02401747
##   predicted class=normal  expected loss=0.4285714  P(node) =0.02736513
##     class counts:     0    15    20
##    probabilities: 0.000 0.429 0.571
##   left son=18 (16 obs) right son=19 (19 obs)
##   Primary splits:
##       free.sulfur.dioxide  < 27.5     to the right, improve=3.952068, (0 missing)
##       pH                   < 3.36     to the left,  improve=2.742857, (0 missing)
##       total.sulfur.dioxide < 56.5     to the left,  improve=2.274436, (0 missing)
##       residual.sugar       < 2.35     to the right, improve=1.542857, (0 missing)
##       volatile.acidity     < 0.375    to the right, improve=1.376190, (0 missing)
##   Surrogate splits:
##       volatile.acidity     < 0.415    to the right, agree=0.743, adj=0.438, (0 split)
##       pH                   < 3.36     to the left,  agree=0.714, adj=0.375, (0 split)
##       sulphates            < 0.755    to the right, agree=0.686, adj=0.313, (0 split)
##       total.sulfur.dioxide < 73       to the right, agree=0.657, adj=0.250, (0 split)
##       chlorides            < 0.0755   to the left,  agree=0.629, adj=0.188, (0 split)
##
## Node number 10: 29 observations,    complexity param=0.01528384
##   predicted class=good    expected loss=0.5172414  P(node) =0.02267396
##     class counts:     1    14    14
##    probabilities: 0.034 0.483 0.483
##   left son=20 (20 obs) right son=21 (9 obs)
##   Primary splits:
##       sulphates            < 0.545    to the right, improve=3.9704980, (0 missing)
##       chlorides            < 0.0665   to the right, improve=2.2935140, (0 missing)
##       total.sulfur.dioxide < 9.5      to the right, improve=1.4006570, (0 missing)
##       density              < 0.99413  to the left,  improve=1.2482760, (0 missing)
##       pH                   < 3.375    to the left,  improve=0.8638603, (0 missing)
##   Surrogate splits:
##       fixed.acidity  < 9.7      to the left,  agree=0.759, adj=0.222, (0 split)
##       citric.acid    < 0.545    to the left,  agree=0.759, adj=0.222, (0 split)
##       residual.sugar < 1.675    to the right, agree=0.759, adj=0.222, (0 split)
##       chlorides      < 0.064    to the right, agree=0.759, adj=0.222, (0 split)
##       density        < 0.99209  to the right, agree=0.759, adj=0.222, (0 split)
## 
## Node number 11: 75 observations
##   predicted class=normal  expected loss=0.1733333  P(node) =0.05863956
##     class counts:     1    12    62
##    probabilities: 0.013 0.160 0.827 
## 
## Node number 12: 65 observations,    complexity param=0.02183406
##   predicted class=good    expected loss=0.5076923  P(node) =0.05082095
##     class counts:     1    32    32
##    probabilities: 0.015 0.492 0.492 
##   left son=24 (29 obs) right son=25 (36 obs)
##   Primary splits:
##       pH                   < 3.265    to the left,  improve=7.022900, (0 missing)
##       free.sulfur.dioxide  < 6.5      to the left,  improve=4.716923, (0 missing)
##       total.sulfur.dioxide < 28.5     to the left,  improve=3.108802, (0 missing)
##       fixed.acidity        < 10.45    to the right, improve=2.919500, (0 missing)
##       sulphates            < 0.815    to the right, improve=2.210538, (0 missing)
##   Surrogate splits:
##       fixed.acidity        < 8.7      to the right, agree=0.862, adj=0.690, (0 split)
##       total.sulfur.dioxide < 28.5     to the left,  agree=0.800, adj=0.552, (0 split)
##       free.sulfur.dioxide  < 6.5      to the left,  agree=0.754, adj=0.448, (0 split)
##       density              < 0.99669  to the right, agree=0.738, adj=0.414, (0 split)
##       citric.acid          < 0.525    to the right, agree=0.677, adj=0.276, (0 split)
## 
## Node number 13: 111 observations
##   predicted class=normal  expected loss=0.1441441  P(node) =0.08678655
##     class counts:     2    14    95
##    probabilities: 0.018 0.126 0.856 
## 
## Node number 18: 16 observations
##   predicted class=good    expected loss=0.3125  P(node) =0.01250977
##     class counts:     0    11     5
##    probabilities: 0.000 0.688 0.312 
## 
## Node number 19: 19 observations
##   predicted class=normal  expected loss=0.2105263  P(node) =0.01485536
##     class counts:     0     4    15
##    probabilities: 0.000 0.211 0.789 
## 
## Node number 20: 20 observations
##   predicted class=good    expected loss=0.35  P(node) =0.01563722
##     class counts:     1    13     6
##    probabilities: 0.050 0.650 0.300 
## 
## Node number 21: 9 observations
##   predicted class=normal  expected loss=0.1111111  P(node) =0.007036747
##     class counts:     0     1     8
##    probabilities: 0.000 0.111 0.889 
## 
## Node number 24: 29 observations
##   predicted class=good    expected loss=0.2413793  P(node) =0.02267396
##     class counts:     0    22     7
##    probabilities: 0.000 0.759 0.241 
## 
## Node number 25: 36 observations,    complexity param=0.01746725
##   predicted class=normal  expected loss=0.3055556  P(node) =0.02814699
##     class counts:     1    10    25
##    probabilities: 0.028 0.278 0.694 
##   left son=50 (10 obs) right son=51 (26 obs)
##   Primary splits:
##       pH             < 3.395    to the right, improve=4.633333, (0 missing)
##       residual.sugar < 1.85     to the right, improve=3.083333, (0 missing)
##       chlorides      < 0.075    to the left,  improve=2.713333, (0 missing)
##       citric.acid    < 0.375    to the left,  improve=2.333333, (0 missing)
##       fixed.acidity  < 8.95     to the left,  improve=1.488506, (0 missing)
##   Surrogate splits:
##       volatile.acidity    < 0.355    to the right, agree=0.806, adj=0.3, (0 split)
##       free.sulfur.dioxide < 37.5     to the right, agree=0.806, adj=0.3, (0 split)
##       fixed.acidity       < 7.25     to the left,  agree=0.750, adj=0.1, (0 split)
##       chlorides           < 0.0555   to the left,  agree=0.750, adj=0.1, (0 split)
## 
## Node number 50: 10 observations
##   predicted class=good    expected loss=0.3  P(node) =0.007818608
##     class counts:     0     7     3
##    probabilities: 0.000 0.700 0.300 
## 
## Node number 51: 26 observations
##   predicted class=normal  expected loss=0.1538462  P(node) =0.02032838
##     class counts:     1     3    22
##    probabilities: 0.038 0.115 0.846
```

```r
# plot the cost complexity parameters
plotcp(WineData_model)
```

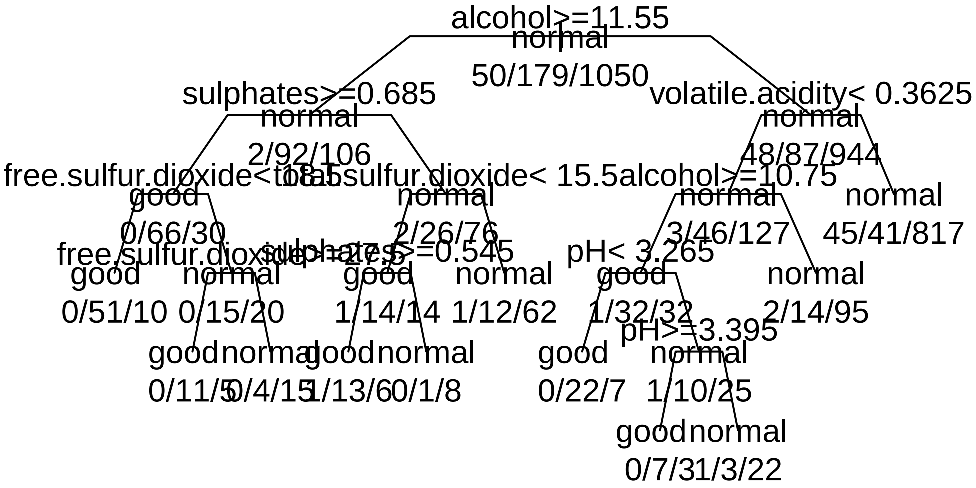


```
# Visualizing Decision Trees
fancyRpartPlot(WineData_model)
```



Rattle 2017−Jun−05 18:34:00 ai

```
# Visualize the classification tree
plot(WineData_model, uniform=TRUE, branch=0.6, margin=0.1)
text(WineData_model, all=TRUE, use.n = TRUE)
```



```
# Model Evaluation using test data
WineData_predict <- predict(WineData_model, WineData_test, type="class")

# Use the table function to generate a classification table for testing dataset
table(WineData_test$quality, WineData_predict)
```

```
##         WineData_predict
##          bad good normal
##   bad      0    0     13
##   good     0   15     23
##   normal   0   13    256
```

```
# Accuracy : Measures of performance
library(caret)
```

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

```
confusionMatrix(table(WineData_predict, WineData_test$quality))
```

```
## Confusion Matrix and Statistics
##
##
## WineData_predict bad good normal
##           bad      0    0      0
##           good     0   15     13
##           normal  13   23    256
##
## Overall Statistics
##
##                Accuracy : 0.8469
##                  95% CI : (0.8027, 0.8845)
##     No Information Rate : 0.8406
##     P-Value [Acc > NIR] : 0.4159
##
##                   Kappa : 0.3119
##  Mcnemar's Test P-Value : NA
```

```
##
## Statistics by Class:
##
##                      Class: bad Class: good Class: normal
## Sensitivity             0.00000     0.39474        0.9517
## Specificity             1.00000     0.95390        0.2941
## Pos Pred Value              NaN     0.53571        0.8767
## Neg Pred Value          0.95937     0.92123        0.5357
## Prevalence              0.04063     0.11875        0.8406
## Detection Rate          0.00000     0.04688        0.8000
## Detection Prevalence    0.00000     0.08750        0.9125
## Balanced Accuracy       0.50000     0.67432        0.6229
```

```r
# Pruning a recursive partitioning tree
# Find minimum cross-validation error of the classification tree model
min(WineData_model$cptable[,"xerror"])
```

```
## [1] 0.8558952
```

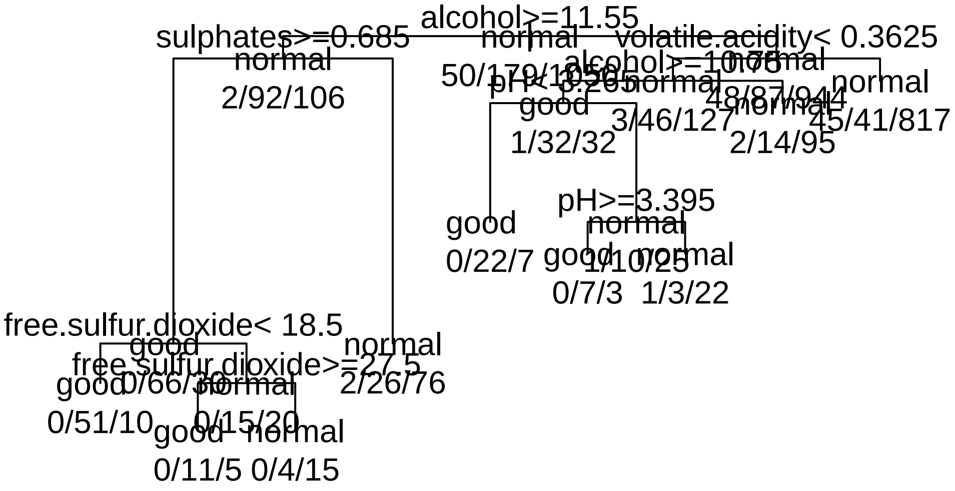
```r
# Locate the record with the minimum cross-validation errors
which.min(WineData_model$cptable[,"xerror"])
```

```
## 5
## 5
```

```r
# Get the cost complexity parameter of the record with the minimum cross-validation errors
WineData_model_CP = WineData_model$cptable[5, "CP"]
WineData_model_CP
```

```
## [1] 0.01528384
```

```r
# Prune the tree by setting the cp parameter to the CP value of the record with minimum cross-validatio
prune_tree = prune(WineData_model, cp=WineData_model_CP)

# Visualize the classification tree by using the plot and text function
plot(prune_tree, margin=0.1)
text(prune_tree, all=TRUE, use.n=TRUE)
```



```r
# Generate a classification table based on the pruned classification tree model
predictions = predict(prune_tree, WineData_test, type="class")
table(WineData_test$quality, predictions)
```

```
##         predictions
##          bad good normal
##   bad      0    0     13
##   good     0   10     28
##   normal   0   13    256
```

```
# Generate confusion matrix
confusionMatrix(table(predictions, WineData_test$quality))
```

```
## Confusion Matrix and Statistics
##
##
## predictions bad good normal
##      bad      0    0      0
##      good     0   10     13
##      normal  13   28    256
##
## Overall Statistics
##
##                Accuracy : 0.8312
##                  95% CI : (0.7856, 0.8706)
##     No Information Rate : 0.8406
##     P-Value [Acc > NIR] : 0.7078
##
##                   Kappa : 0.2012
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: bad Class: good Class: normal
## Sensitivity             0.00000     0.26316        0.9517
## Specificity             1.00000     0.95390        0.1961
## Pos Pred Value              NaN     0.43478        0.8620
## Neg Pred Value          0.95937     0.90572        0.4348
## Prevalence              0.04063     0.11875        0.8406
## Detection Rate          0.00000     0.03125        0.8000
## Detection Prevalence    0.00000     0.07187        0.9281
## Balanced Accuracy       0.50000     0.60853        0.5739
```

```
# # Model Improvement using M5P from RWeka
# # Build Model
# WineData_model_M5P <- M5P(quality ~. , data= WineData_train)

# # Model Evaluation using test data
# WineData_predict_M5P <- predict(WineData_model_M5P, WineData_test)

# WineData_model_M5P

# MAE(WineData_test$quality, WineData_predict_M5P)
```