Practical 5

Classification Models

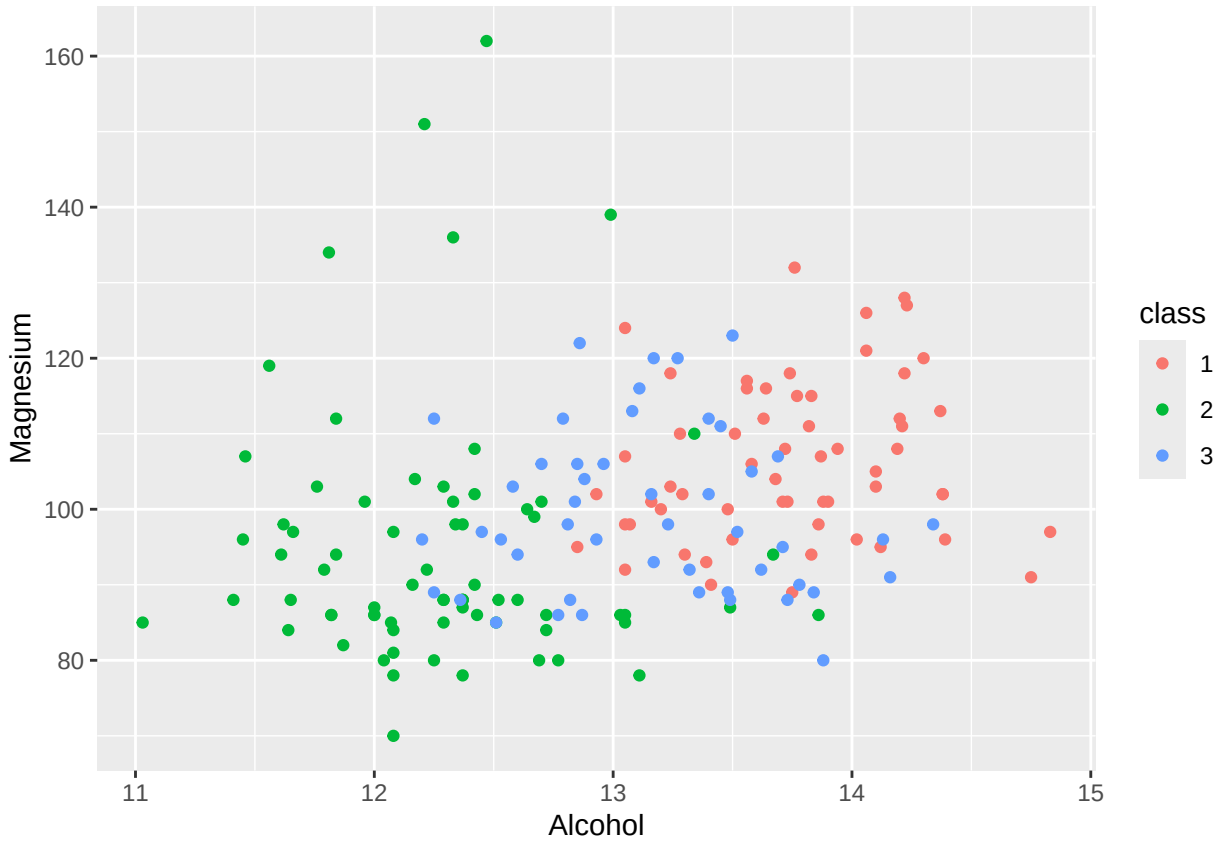
Joan Navarro Bellido

PART 1: A classification case study

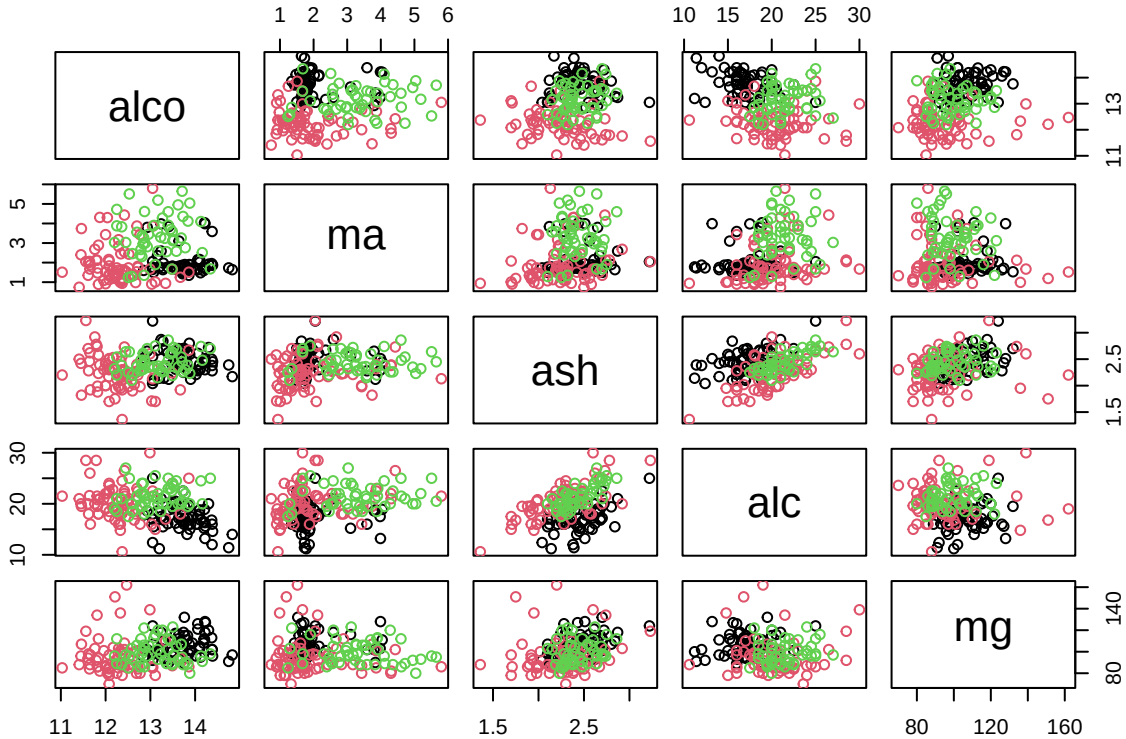
```r
# Load the dataset
wine <- read.table("./dataset/wine.txt", header = FALSE, sep = ",")

# Rename columns
colnames(wine) <- c("class", "alco", "ma", "ash", "alc", "mg", "tp", "flav", "noflav",
                    "proa", "col", "hue", "od", "prol")

# Convert class to a factor
wine$class <- as.factor(wine$class)

# Check structure
str(wine)
```

```
## 'data.frame':    178 obs. of  14 variables:
##  $ class : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ alco  : num  14.2 13.2 13.2 14.4 13.2 ...
##  $ ma    : num  1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
##  $ ash   : num  2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
##  $ alc   : num  15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
##  $ mg    : int  127 100 101 113 118 112 96 121 97 98 ...
##  $ tp    : num  2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
##  $ flav  : num  3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
##  $ noflav: num  0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
##  $ proa  : num  2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
##  $ col   : num  5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
##  $ hue   : num  1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
##  $ od    : num  3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
##  $ prol  : int  1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...
```

```r
library(ggplot2)

# Scatterplot of Alcohol vs Magnesium colored by wine class
qplot(alco, mg, data = wine, color = class,
      xlab = "Alcohol", ylab = "Magnesium")
```

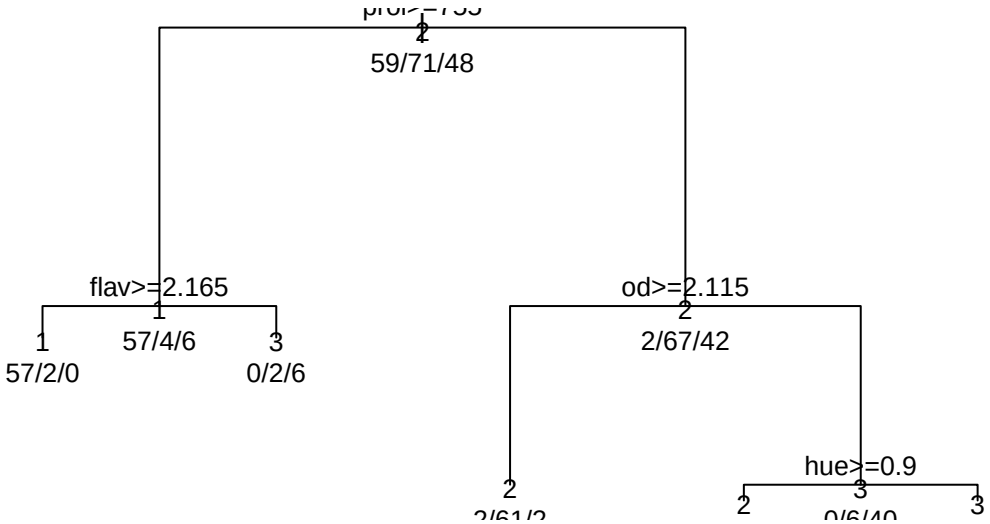



```
pairs(wine[, 2:6], col = wine$class)
```




```
# Install and load the rpart package
if (!require("rpart")) {
```

```
  install.packages("rpart", dependencies = TRUE)
  library("rpart")
}

# Build the decision tree model
model <- rpart(class ~ ., data = wine, method = "class")

# Plot the decision tree
plot(model, main = "Classification Tree for Wine Dataset")
text(model, use.n = TRUE, all = TRUE, cex = 0.8)
```




```
printcp(model)
```

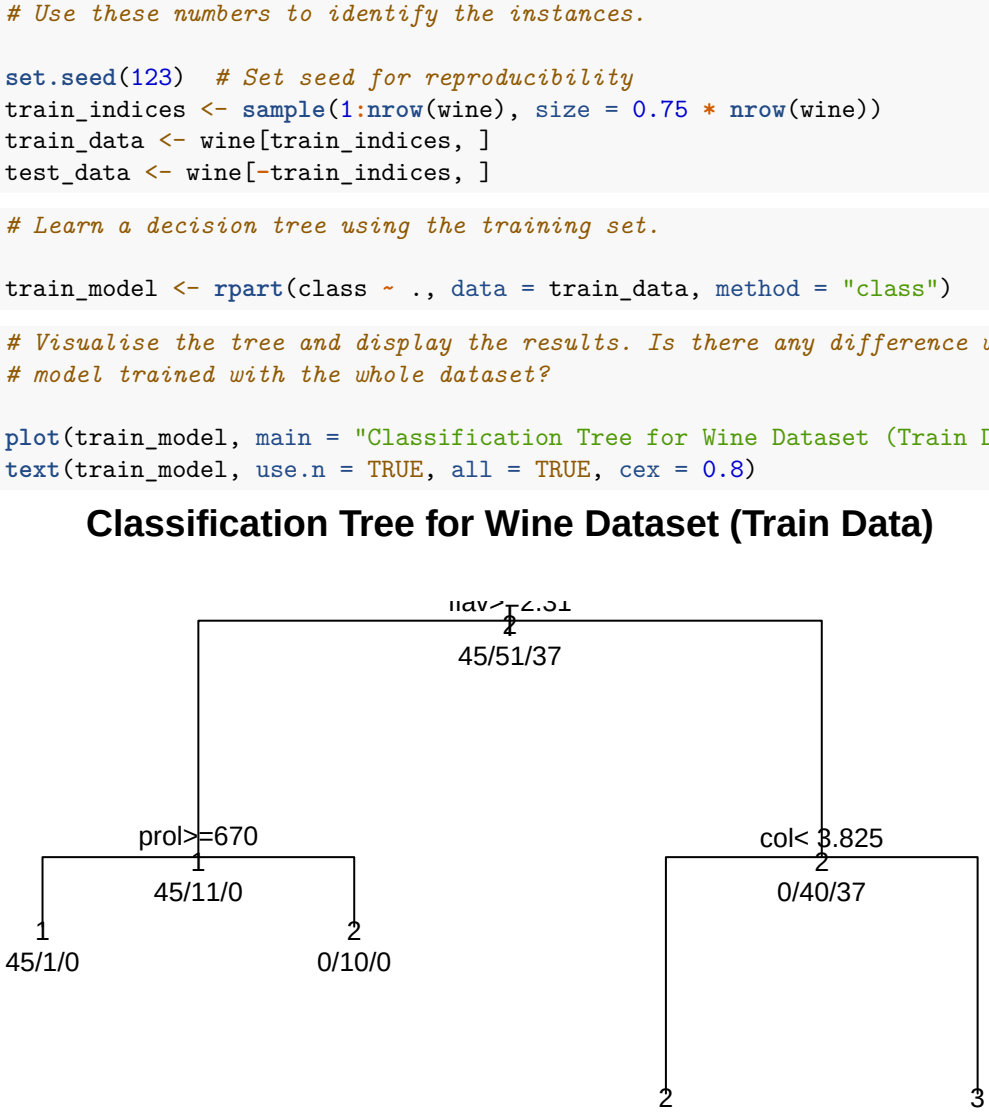
```
##
## Classification tree:
## rpart(formula = class ~ ., data = wine, method = "class")
##
## Variables actually used in tree construction:
## [1] flav hue  od   prol
##
## Root node error: 107/178 = 0.60112
##
## n= 178
##
##         CP nsplit rel error  xerror     xstd
## 1 0.495327      0   1.00000 1.00000 0.061056
## 2 0.317757      1   0.50467 0.48598 0.056701
## 3 0.056075      2   0.18692 0.33645 0.050084
## 4 0.028037      3   0.13084 0.28037 0.046676
## 5 0.010000      4   0.10280 0.21495 0.041825
```

PART 2: Exercices

```r
# Randomly split the dataset into 75% train and 25% test.
# Note: It can be done by using the sample function to generate
# integers belonging to the interval [1..size(data set)].
# Use these numbers to identify the instances.

set.seed(123)  # Set seed for reproducibility
train_indices <- sample(1:nrow(wine), size = 0.75 * nrow(wine))
train_data <- wine[train_indices, ]
test_data <- wine[-train_indices, ]
```

```r
# Learn a decision tree using the training set.

train_model <- rpart(class ~ ., data = train_data, method = "class")
```

```r
# Visualise the tree and display the results. Is there any difference with respect to the
# model trained with the whole dataset?

plot(train_model, main = "Classification Tree for Wine Dataset (Train Data)")
text(train_model, use.n = TRUE, all = TRUE, cex = 0.8)
```




```r
# Use the model to predict the class label for the test set by using the "predict"
# function. Repeat the predictions but now using the parameter type="class" (use a
# different variable to keep the new results). What are the differences?

# Predict probabilities
pred_prob <- predict(train_model, newdata = test_data)

# Predict class labels
pred_class <- predict(train_model, newdata = test_data, type = "class")
```

```r
# Calculate the performance of the model when it is applied to the test set by
# displaying a table that shows the predicted classes versus the real classes.
```

```r
table(Predicted = pred_class, Actual = test_data$class)
```

```
##           Actual
## Predicted  1  2  3
##         1 13  3  0
##         2  0 16  0
##         3  1  1 11
```

```r
# Try some other methods or parameters of the rpart
# package to see whether you can still improve the results further.
# You can also compare with other packages and classification techniques.

# Example: Prune the tree using the optimal CP value
optimal_cp <- train_model$cptable[which.min(train_model$cptable[, "xerror"]), "CP"]
pruned_model <- prune(train_model, cp = optimal_cp)

# Plot the pruned tree
plot(pruned_model, main = "Pruned Classification Tree")
text(pruned_model, use.n = TRUE, all = TRUE, cex = 0.8)
```