Coursework 8

Fortunat Mutunda

March 30, 2016

```
## Loading required package: xtable

## Warning: Failed to load RGtk2 dynamic library, attempting to install it.

## Please install GTK+ from http://r.research.att.com/libs/GTK_2.24.17-X11.pkg

## If the package still does not load, please ensure that GTK+ is installed and that it is on your PATH

## IN ANY CASE, RESTART R BEFORE TRYING TO LOAD THE PACKAGE AGAIN

## Rattle: A free graphical interface for data mining with R.
## Version 4.1.0 Copyright (c) 2006-2015 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

1. Read - http://www.r2d3.us/visual-intro-to-machine-learning-part-1/ What is the quality of the classifier? Can you understand when it works well and when not?

This is a tricky and

2. Use this small data example and build a decision tree (manually, explaining all steps/choices).

ord.	Outlook	Temp	Humidity	Windy	Play
1.	Sunny	Hot	High	FALSE	No
2.	Sunny	Hot	High	TRUE	No
3.	Overcast	Hot	High	FALSE	Yes
4.	Rainy	Mild	High	FALSE	Yes
5.	Rainy	Cool	Normal	FALSE	Yes
6.	Rainy	Cool	Normal	TRUE	No
7.	Overcast	Cool	Normal	TRUE	Yes
8.	Sunny	Mild	High	FALSE	No
9.	Sunny	Cool	Normal	FALSE	Yes
10.	Rainy	Mild	Normal	FALSE	Yes
11.	Sunny	Mild	Normal	TRUE	Yes
12.	Overcast	Mild	High	TRUE	Yes
13.	Overcast	Hot	Normal	FALSE	Yes
14.	Rainy	Mild	High	TRUE	No
15.	Overcast	Cool	High	FALSE	No

```
Providing  that there is mild, overcast, high humidity and high wind weather - should one play tennis o
```

- Calculate entropy

```r
playTennis = entropy(c(6,9), unit = "log2")

windy = (
 ((6/15) * entropy(c(3,3), unit = "log2" )) +
    ((9/15) * entropy(c(6,3), unit = "log2" ))
)

ewindy = playTennis - windy


humidity = (
 ((8/15) * entropy(c(3,5), unit = "log2" )) +
    ((7/15) * entropy(c(6,1), unit = "log2" ))
)

ehumidity = playTennis - humidity

temp = (
 ((4/15) * entropy(c(2,2), unit = "log2" )) +
    ((6/15) * entropy(c(4,2), unit = "log2" )) +
   ((5/15) * entropy(c(3,2), unit = "log2" ))
)

etemp  = playTennis - temp
outlook = (
 ((5/15) * entropy(c(2,3), unit = "log2" )) +
    ((5/15) * entropy(c(4,2), unit = "log2" )) +
   ((5/15) * entropy(c(3,2), unit = "log2" ))
)

eoutlook = playTennis - outlook

ewindy
```

```
## [1] 0.01997309
```

```r
ehumidity
```

```
## [1] 0.1858052
```

```r
etemp
```

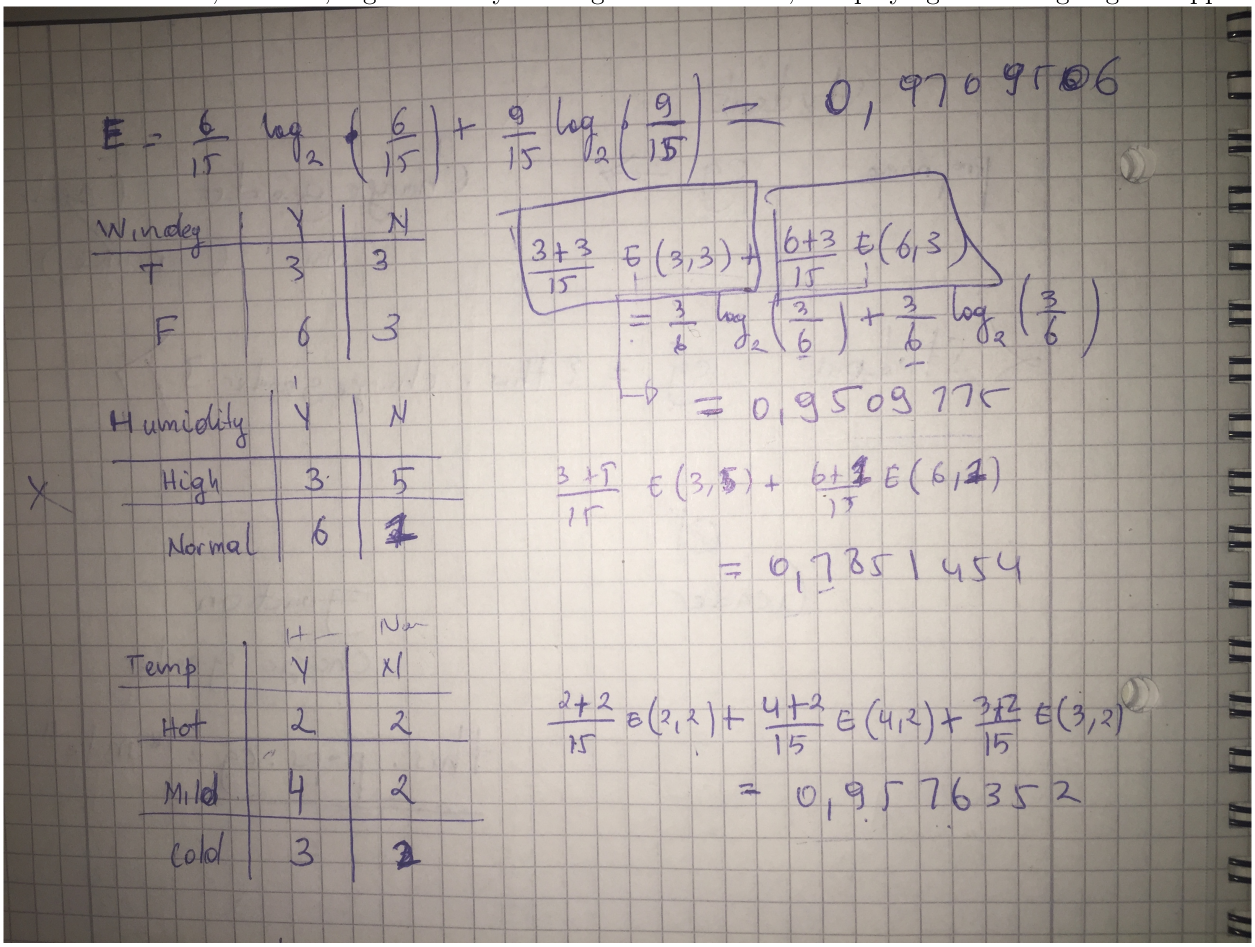
```
## [1] 0.0133154
```

```r
eoutlook
```

```
## [1] 0.01755159
```

This task was more trickier than expected. Basicaly I found two ways of doing it. One is just to go with all the data one by one and try to see which one should be the root. It takes time, depending on wheter one is fast at doing that. Or there is another way which is the best solution. To calculate the entropy. Basically

I would have to calculate the entropy and then the gain. But doing so I have written a small script to ulstrate how I did, and I have aslo joined a screenshot below for the tree and my calculations. Providing that there is mild, overcast, high humidity and high wind weather, Yes playing tennis is going to happen.



$$E = \frac{6}{15}\log_2\left(\frac{6}{15}\right) + \frac{9}{15}\log_2\left(\frac{9}{15}\right) = 0,9709506$$

Windey	Y	N
T	3	3
F	6	3

$$\boxed{\frac{3+3}{15}E(3,3)} + \boxed{\frac{6+3}{15}E(6,3)}$$

$$= \frac{3}{6}\log_2\left(\frac{3}{6}\right) + \frac{3}{6}\log_2\left(\frac{3}{6}\right)$$

$$= 0,9509775$$

Humidity	Y	N
High	3	5
Normal	6	1

$$\frac{3+5}{15}E(3,5) + \frac{6+1}{15}E(6,1)$$

$$= 0,7851454$$

Temp	Y	N
Hot	2	2
Mild	4	2
Cold	3	2

$$\frac{2+2}{15}E(2,2) + \frac{4+2}{15}E(4,2) + \frac{3+2}{15}E(3,2)$$

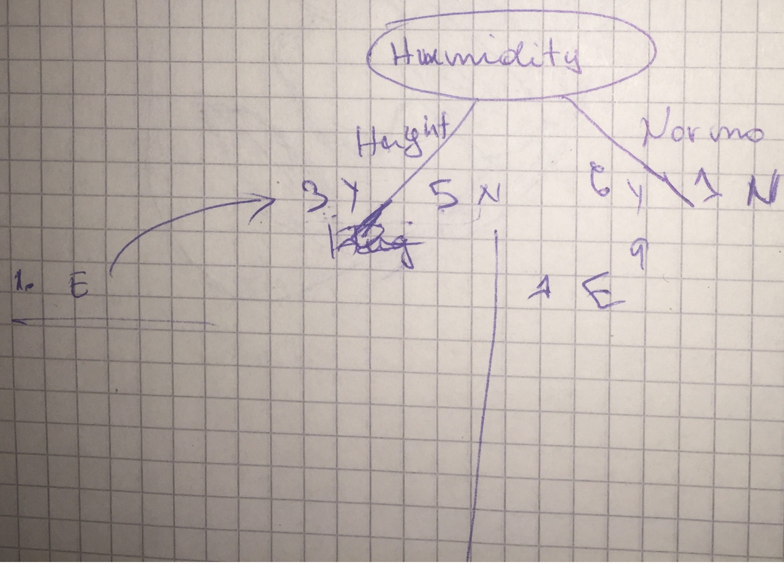
$$= 0,9576352$$

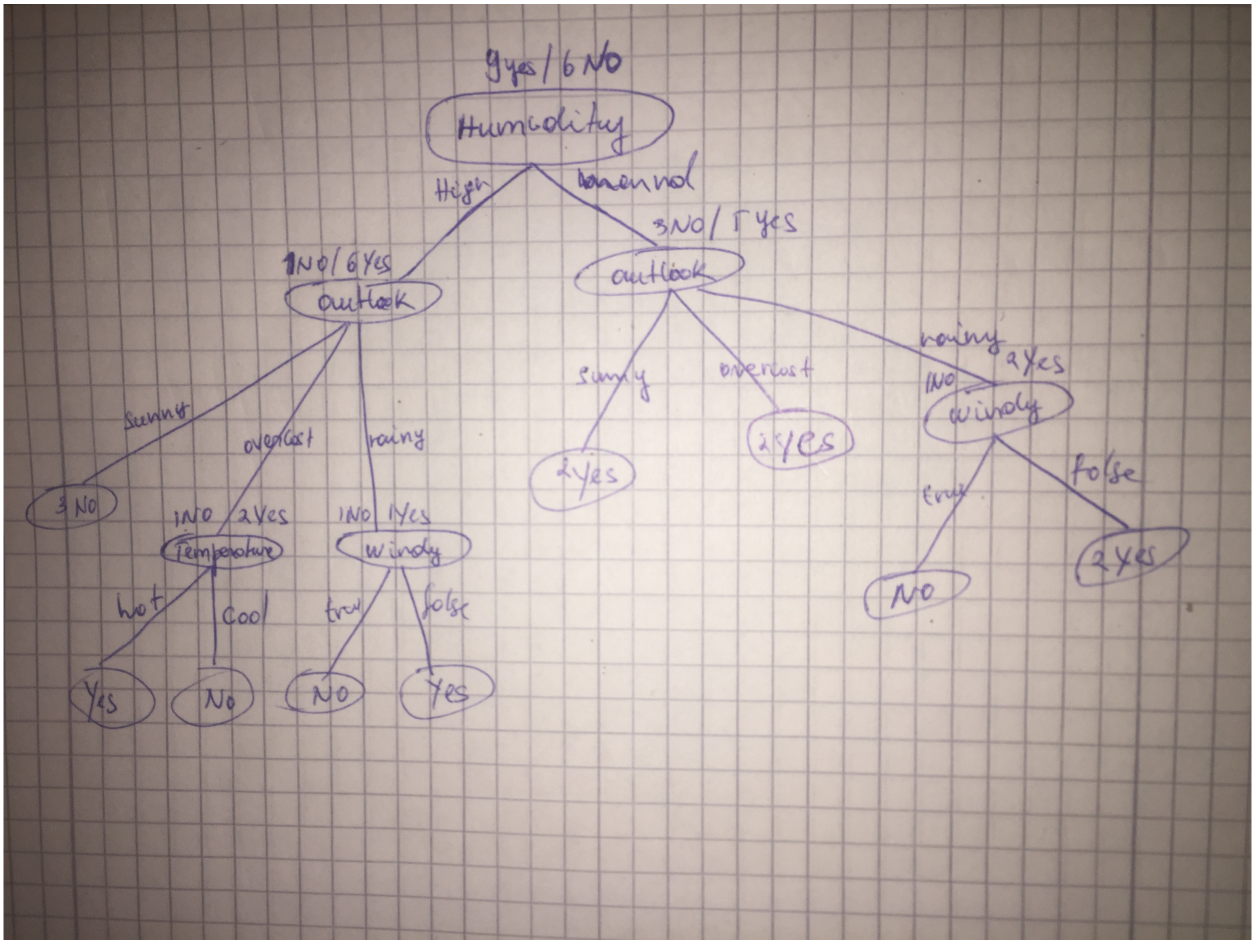
Outlook	Y	N
Sunny	2	3
Overcast	4	1
Rainy	3	2

$$\frac{2+3}{15}\boxed{E(2,3)} + \frac{4+1}{15}E(4,1) + \frac{3+2}{15}E(3,2)$$

$$= \frac{2}{10}$$

$$= 0,953399$$

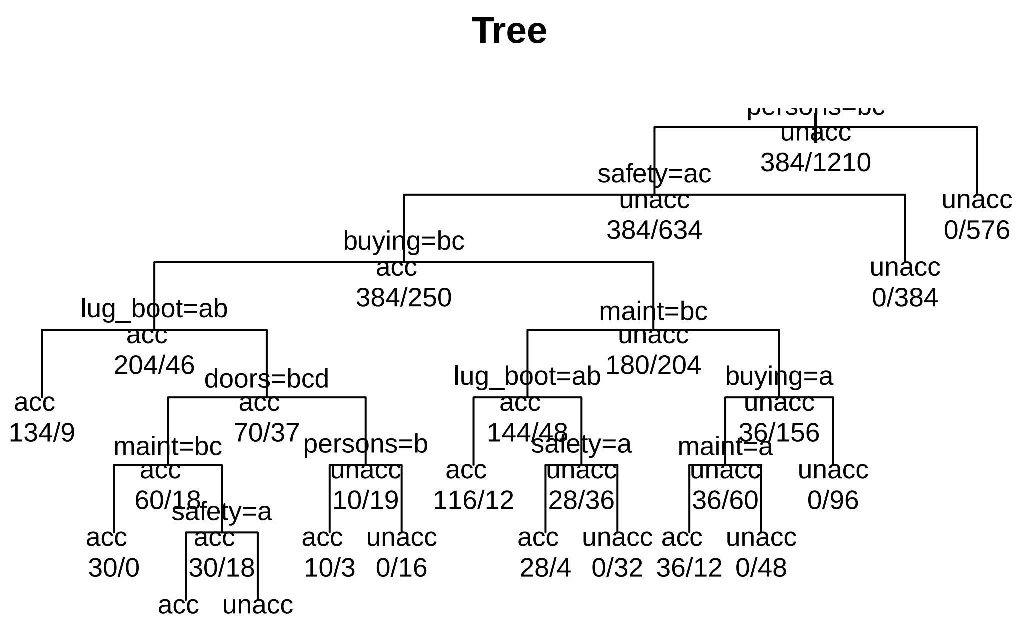




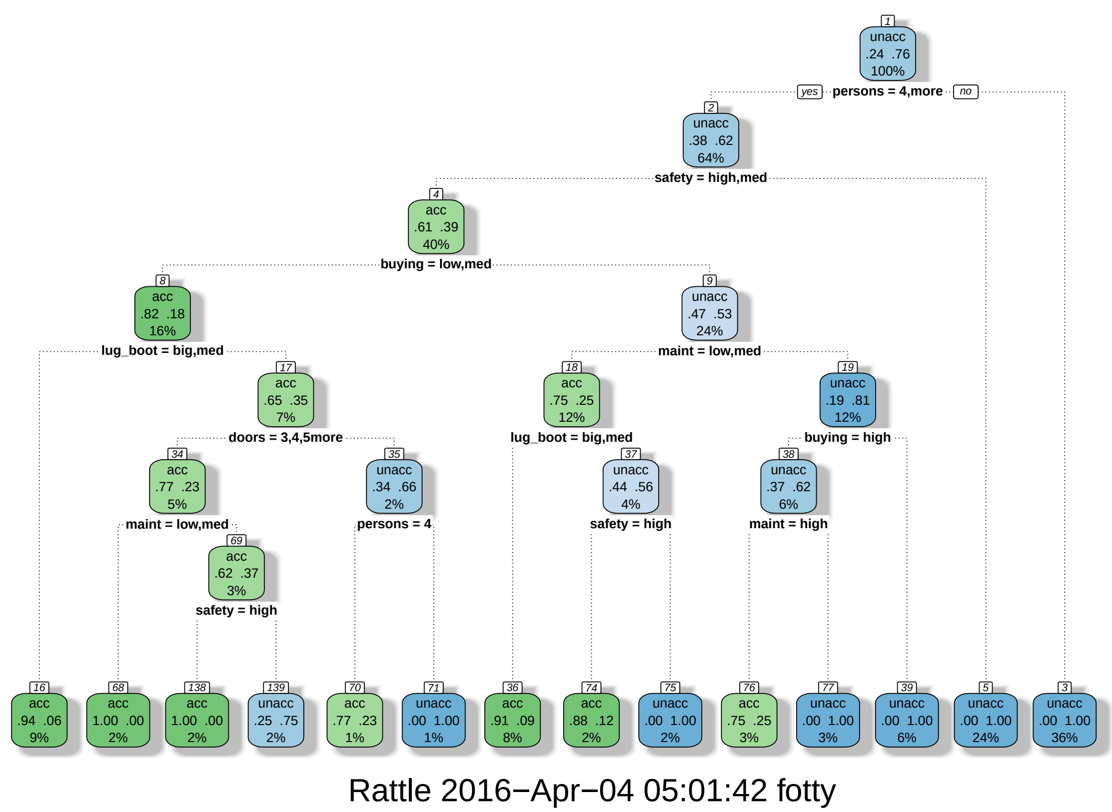
3. Use the Cars data set and apply decision trees for classification. Describe the tree. (you can use R, or Weka (install Weka from here), or python...). Compare the decision tree approach to the association rules derived from the same data.

    - To make your life easier, we recommend you remove observations with two infrequent classes - good and v-good. You can get the resulting dataset here
    - in R, you can use library `rpart` to build the trees and `rpart.plot` to visualize them

```
carss = read.csv("car.data.txt", header = T, sep = ",")
emc2 = class ~  buying + maint + doors + persons + lug_boot + safety
flash = rpart(emc2, method="class", data=carss)
plot(flash, uniform=TRUE, main="Tree")
text(flash, use.n=TRUE, all=TRUE, cex=.8)
```

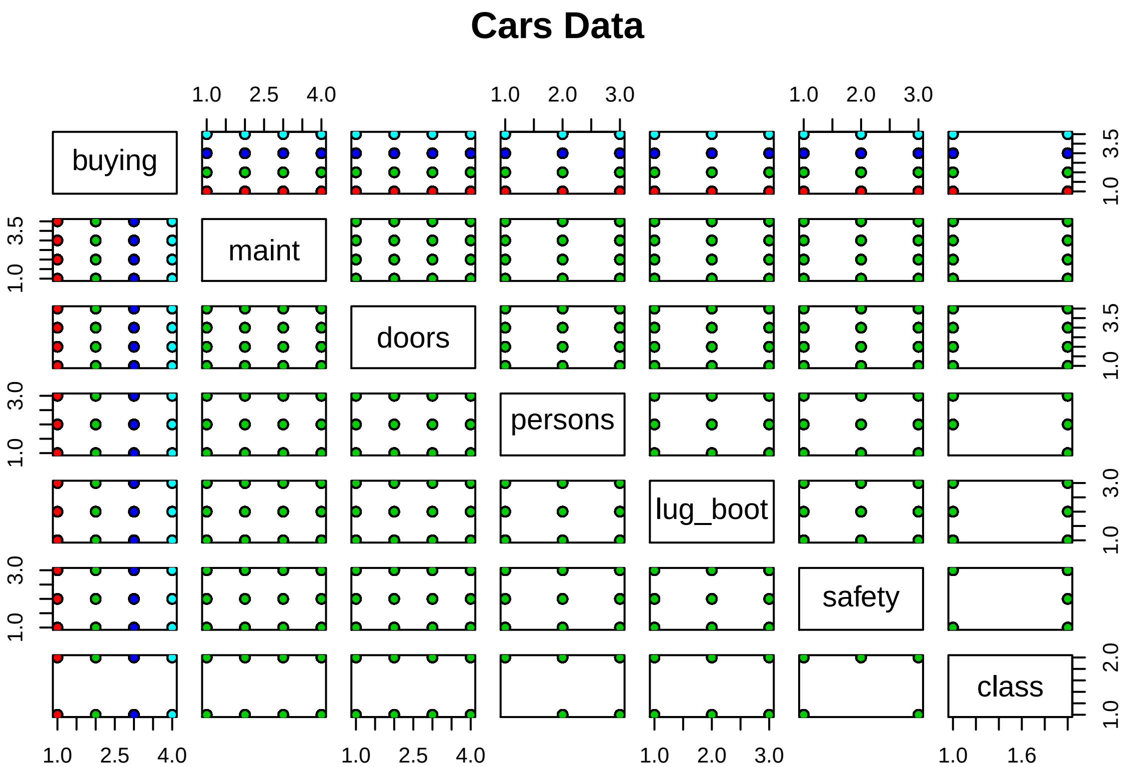



```r
fancyRpartPlot(flash)
```



4. Use the same cars data set. Apply decision trees and Naive Bayes classifiers on the same data. Can you confirm that one method is better than the other in some way? Perform 10-fold cross-validation. Provide final results as 2x2 tables of TP. FP, FN, TN and some measures - accuracy, precision, recall.

```r
pairs(carss, main = "Cars Data ",pch = 21, bg =c("red","green3","blue","cyan","magenta","yellow","orang
```




```r
data(carss)
```

```
## Warning in data(carss): data set 'carss' not found
```

```r
summary(carss)
```

```
##    buying      maint       doors      persons     lug_boot     safety
##  high :432   high :419   2    :407   2   :576   big  :512   high:481
##  low  :347   low  :360   3    :399   4   :510   med  :527   low :576
##  med  :383   med  :383   4    :394   more:508   small:555   med :537
##  vhigh:432   vhigh:432   5more:394
##    class
##  acc  : 384
##  unacc:1210
##
##
```

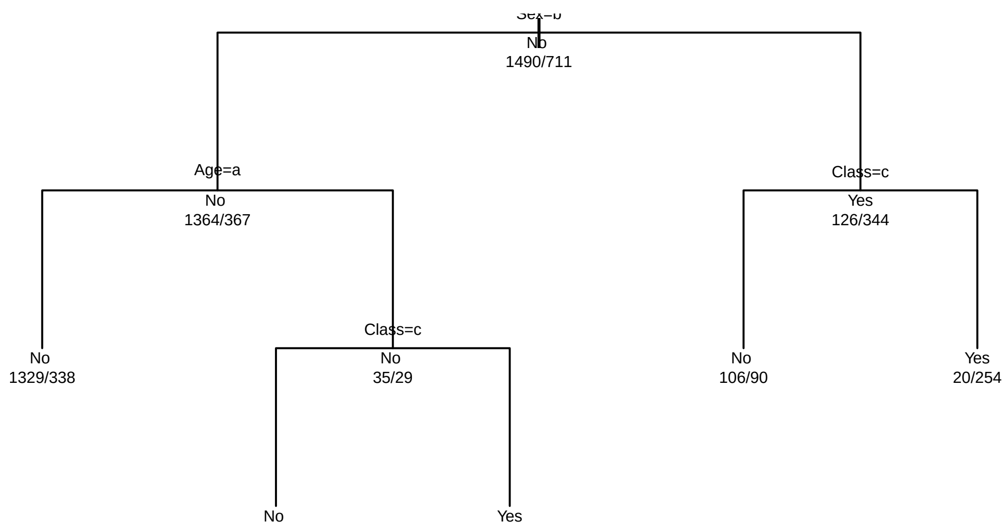
```r
classifier<-naiveBayes(carss[,1:7], carss[,7])
table(predict(classifier, carss[,-1]), carss[,1])
```

```
##
##         high low med vhigh
##   acc    108  89 115    72
##   unacc  324 258 268   360
```

This simple case study shows that a Naïve Bayes classifier makes few mistakes in a dataset that, although simple, is not linearly separable, as shown in the scatterplots and by a look at the confusion matrix, where all misclassifications are between unacc and acc.

5. Use the Titanic data set - compare your classifiers learned from Titanic data - decision trees, Bayes rules, association rules - and try to characterise the rules observed in data using these approaches. How can they be interpreted against each other?

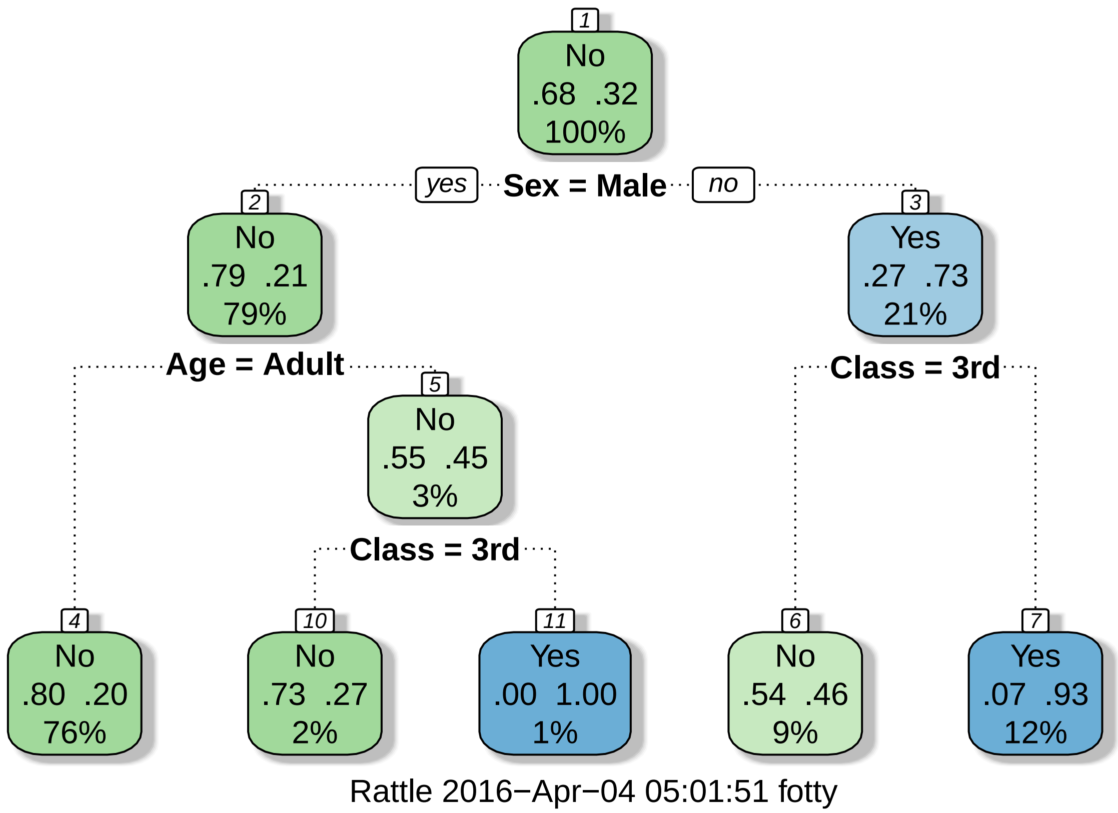
```
titanic <- read.table( "titanic.txt", sep = ',' , header = TRUE)
tfit <- rpart(Survived ~ Class + Age + Sex, data = titanic)
plot(tfit, uniform=TRUE)
text(tfit,use.n=TRUE, all=TRUE, cex=.5)
```




```
length(titanic$Sex[titanic$Sex=="Female" & titanic$Survived =="Yes"])
```

```
## [1] 344
```

```
tnaive <- naiveBayes(Survived ~ Class + Age + Sex, data = titanic)
fancyRpartPlot(tfit)
```



6. (Bonus 1p) How to detect and avoid overfitting? What is the good (optimal?) size of the decision tree classifiers? Use the above Cars data, and for comparison use one of the two data sets - the Mushroom (LINK) or the Connect 4 (LINK).