### Phần code:

##10.2 The human leukocyte antigen system (HLA) is a group of genes on chromosome 6 that are linked to the immune system. The data set hla in the gap library contains HLA data on six markers for 177 schizophrenia patients and 94 matched healthy controls. How can we use these six markers to distinguish between the patients and their controls?

#install.packages('gap')

library(gap)

data = hla[, 2:8]

library(rpart)

names(data)

tree1 <- rpart(y ~. , data = data)

tree1

printcp(tree1)

best1 <- tree1$cptable[which.min(tree1$cptable[, "xerror"]), "CP"]

best1

# Tạo cây được cắt tỉa

prunned\_tree <- prune(tree1, cp= best1)

#install.packages('rpart.plot')

library(rpart.plot)

prp(prunned\_tree,

faclen = 0, # sử dụng tên đầy đủ cho nhãn

extra = 1, # hiển thị số lượng quan sát cho mỗi nút

roundint = F, # Không làm tròn

digits = 5) # hiển thị tối đa 5 chữ số thập phân có ý nghĩa

# 10.4 The data set iris has four measurements on three different species of 150 irises. Figure 3.21 shows the parallel coordinate plot of this data. Describe the factor loadings from a discriminant analysis and try to reconcile these with what you see in this figure.

data(iris)

library(MASS)

ld <- lda(Species ~ ., data = iris)

ld$scaling

#Tải trọng yếu tố trong phân tích phân biệt giúp định lượng mức độ quan trọng của từng biến độc lập (dự báo) trong việc phân biệt giữa các nhóm [1]. Trong bối cảnh của hình ảnh, hiển thị tập dữ liệu Iris, tải trọng yếu tố giúp xác định phép đo hoa nào (Chiều dài đài hoa, Chiều rộng đài hoa, Chiều dài cánh hoa, Chiều rộng cánh hoa) hữu ích nhất trong việc phân tách các loài hoa Iris.

#Tải trọng yếu tố cho phân biệt tuyến tính đầu tiên (LD1) đều dương ngoại trừ Chiều rộng đài hoa, có giá trị âm rất nhỏ. Điều này cho thấy Chiều dài đài hoa, Chiều dài cánh hoa và Chiều rộng cánh hoa đều góp phần tích cực vào LD1, trong khi Chiều rộng đài hoa có đóng góp tiêu cực không đáng kể.

#Tải trọng yếu tố cho phân biệt tuyến tính thứ hai (LD2) là dương cho Chiều dài cánh hoa và âm cho Chiều dài đài hoa và Chiều rộng đài hoa. Điều này cho thấy Chiều dài cánh hoa góp phần tích cực vào LD2, trong khi Chiều dài đài hoa và Chiều rộng đài hoa góp phần tiêu cực.

#10.6 The promotergene data set in the kernlab is another example of a discrimination problem in a large data set. This data set contains 106 gene sequences, classified as being either promoters or not. The explanatory variables are the lists of DNA base values preceding the gene of interest. The base values take the values a, c, g, or t corresponding to adenine, cytosine, guanine, or thymine, respectively. Use the base values to find a rule for discriminating between promoters and non-promoters. Additional information on this data can be found on the UCI website

library(kernlab)

data(promotergene)

library(rpart)

tree2 <- rpart(Class ~ ., data = promotergene)

tree2

printcp(tree2)

best2 <- tree2$cptable[which.min(tree2$cptable[, "xerror"]), "CP"]

best2

# Tạo cây được cắt tỉa

prunned\_tree2 <- prune(tree2, cp= best2)

library(rpart.plot)

prp(prunned\_tree2,

faclen = 0, # sử dụng tên đầy đủ cho nhãn

extra = 1, # hiển thị số lượng quan sát cho mỗi nút

roundint = F, # Không làm tròn

digits = 5) # hiển thị tối đa 5 chữ số thập phân có ý nghĩa

# 10.7 The spam data set in the kernlab library was gathered in an attempt to distinguish emails that are advertisements (spam) from non-spam. There are 4601 email messages collected by Hewlett-Packard and of these, 1813 are classified as spam and 2788 are non-spam. There are 57 explanatory variables that measure the frequency of various words such as “free” or “credit.” Other explanatory variables count the frequency of different punctuation symbols or the length of strings of all capital letters. See the help file for a detailed explanation of the variables in this data set. Can you find a good rule to distinguish spam from non-spam?

library(kernlab)

data(spam)

library(rpart)

tree3 <- rpart(type ~ ., data = spam)

tree3

printcp(tree3)

best3 <- tree3$cptable[which.min(tree3$cptable[, "xerror"]), "CP"]

best3

# Tạo cây được cắt tỉa

prunned\_tree3 <- prune(tree3, cp= best3)

library(rpart.plot)

prp(prunned\_tree3,

faclen = 0, # sử dụng tên đầy đủ cho nhãn

extra = 1, # hiển thị số lượng quan sát cho mỗi nút

roundint = F, # Không làm tròn

digits = 5) # hiển thị tối đa 5 chữ số thập phân có ý nghĩa

# 10.12 The data frame stagec in the rpart library summarizes observations made on 146 men treated for Stage C prostate cancer. There are eight measurements made on each patient including whether or not the cancer has progressed. Use regression tree methods to indicate important prognostic measures on whether the cancer will progress from this data.

library(rpart)

data(stagec)

tree4 <- rpart(pgstat ~ ., data = stagec)

tree4

printcp(tree4)

# Tạo cây được cắt tỉa

prunned\_tree4 <- prune(tree4, cp= best4)

prp(prunned\_tree4,

faclen = 0, # sử dụng tên đầy đủ cho nhãn

extra = 1, # hiển thị số lượng quan sát cho mỗi nút

roundint = F, # Không làm tròn

digits = 5) # hiển thị tối đa 5 chữ số thập phân có ý nghĩa

# 10.13 Kyphosis is a severe curvature of the spine. The kyphosis data frame in the rpart library presents history of 81 children who had corrective spinal surgery and whether this resulted in kyphosis. The data frame includes three additional variables indicating the extent of the surgery and the child’s age in months.

#(a) Use regression trees to identify risk factors for kyphosis.

#(b) Similarly, use logistic regression to build a linear model of thelog-odds risk of kyphosis.

#(c) Discuss these two different approaches to modeling kyphosis.Which model is more intuitive? Which is easier to explain and use?

# a)

library(rpart)

data(kyphosis)

tree5 <- rpart(Kyphosis ~ ., data = kyphosis)

tree5

printcp(tree5)

best5 <- tree5$cptable[which.min(tree5$cptable[, "xerror"]), "CP"]

best5

# Tạo cây được cắt tỉa

prunned\_tree5 <- prune(tree5, cp= best5)

library(rpart.plot)

prp(prunned\_tree5,

faclen = 0, # sử dụng tên đầy đủ cho nhãn

extra = 1, # hiển thị số lượng quan sát cho mỗi nút

roundint = F, # Không làm tròn

digits = 5) # hiển thị tối đa 5 chữ số thập phân có ý nghĩa

# b)

library(nnet)

kyphosis$class <- as.factor(kyphosis$Kyphosis)

kyphosis$rClass <- relevel(kyphosis$Kyphosis, ref = 2)

kyphosislogit <- multinom(rClass ~ ., data = kyphosis)

print(ws <- summary(kyphosislogit), digits = 4)

# Kiểm định các biến có ý nghĩa trong mô hình 0

tratio <- ws$coefficients/ws$standard.errors

print(2 \* (1 - pt(abs(tratio), df = ws$edf)), digits = 4)

ws$fitted.values # xác suất ươc tính mỗi loại rượu thuộc nhóm i

# Về mặt trực quan và dễ giải thích:

# Cây hồi quy: Nói chung được coi là trực quan hơn, đặc biệt đối với đối tượng không chuyên về kỹ thuật. Quy trình ra quyết định rõ ràng và dễ theo dõi.

# Hồi quy Logistic: Ít trực quan hơn, đặc biệt là việc giải thích hệ số. Cần có một số hiểu biết về thống kê và cách hoạt động của hồi quy logistic.

#Về mặt dễ sử dụng:

# Cây hồi quy: Tương đối dễ sử dụng với các hàm tích hợp trong thư viện R như rpart. Tuy nhiên, việc điều chỉnh và cắt tỉa cây có thể đòi hỏi một số thử nghiệm.

# Hồi quy Logistic: Cũng tương đối dễ sử dụng với các hàm như glm trong R. Việc giải thích có thể khó khăn hơn so với cây hồi quy.

#10.14 The data set mammoexp in the TH.data library describes a survey of 412 women’s attitudes to and experiences with mammography. Among the data are the respondent’s perceived benefits of mammography and whether a close female relative had been diagnosed with breast cancer. A suitable response to these measurements is the variable ME that measures the time since the woman had her last mammography. The categorical responses to this variable are never, within a year, and over a year ago. Use a regression tree approach to model the time since last mammography.

library(TH.data)

data(mammoexp)

tree6 <- rpart(ME ~ ., data = mammoexp)

tree6

printcp(tree6)

best6 <- tree6$cptable[which.min(tree6$cptable[, "xerror"]), "CP"]

best6

# Tạo cây được cắt tỉa

prunned\_tree6 <- prune(tree6, cp= best6)

library(rpart.plot)

prp(prunned\_tree6,

faclen = 0, # sử dụng tên đầy đủ cho nhãn

extra = 1, # hiển thị số lượng quan sát cho mỗi nút

roundint = F, # Không làm tròn

digits = 5) # hiển thị tối đa 5 chữ số thập phân có ý nghĩa

# 10.16 Regression trees are not limited to classifying individuals into discrete groups. These methods will also estimate cut points in continuous measurements and thereby create groups of data. Build a regression tree to model the mpg values in the car data introduced in Table 9.1. Show that the most important classification is the number of cylinders and, to a smaller degree, the displacement

library(MASS)

data(mtcars)

tree7 <- rpart(mpg ~ ., data = mtcars)

tree7

printcp(tree7)

best7 <- tree7$cptable[which.min(tree7$cptable[, "xerror"]), "CP"]

best7

# Tạo cây được cắt tỉa

prunned\_tree7 <- prune(tree7, cp= best7)

library(rpart.plot)

prp(prunned\_tree7,

faclen = 0, # sử dụng tên đầy đủ cho nhãn

extra = 1, # hiển thị số lượng quan sát cho mỗi nút

roundint = F, # Không làm tròn

digits = 5) # hiển thị tối đa 5 chữ số thập phân có ý nghĩa

### Phần chạy chương trình:

> library(gap)

Error: package or namespace load failed for ‘gap’ in readRDS(nsInfoFilePath):

error reading from connection

In addition: Warning messages:

1: package ‘gap’ was built under R version 4.1.3

2: In readRDS(nsInfoFilePath) : error reading the file

> data = hla[, 2:8]

> library(rpart)

> names(data)

[1] "y" "DQR.a1" "DQR.a2" "DQA.a1" "DQA.a2" "DQB.a1" "DQB.a2"

> tree1 <- rpart(y ~. , data = data)

> tree1

n= 271

node), split, n, deviance, yval

\* denotes terminal node

1) root 271 61.3948300 0.34686350

2) DQA.a1< 5.5 207 40.8502400 0.27053140

4) DQR.a2< 12.5 110 12.2181800 0.12727270

8) DQR.a1>=1.5 87 4.7126440 0.05747126 \*

9) DQR.a1< 1.5 23 5.4782610 0.39130430

18) DQB.a2>=2.5 15 2.9333330 0.26666670 \*

19) DQB.a2< 2.5 8 1.8750000 0.62500000 \*

5) DQR.a2>=12.5 97 23.8144300 0.43298970

10) DQR.a1< 6.5 35 5.6000000 0.20000000

20) DQR.a1>=2.5 15 0.0000000 0.00000000 \*

21) DQR.a1< 2.5 20 4.5500000 0.35000000 \*

11) DQR.a1>=6.5 62 15.2419400 0.56451610

22) DQA.a2>=3.5 29 6.8275860 0.37931030

44) DQR.a2>=15.5 21 3.8095240 0.23809520

88) DQR.a2< 21.5 14 0.9285714 0.07142857 \*

89) DQR.a2>=21.5 7 1.7142860 0.57142860 \*

45) DQR.a2< 15.5 8 1.5000000 0.75000000 \*

23) DQA.a2< 3.5 33 6.5454550 0.72727270

46) DQR.a1>=21.5 8 1.5000000 0.25000000 \*

47) DQR.a1< 21.5 25 2.6400000 0.88000000 \*

3) DQA.a1>=5.5 64 15.4375000 0.59375000

6) DQA.a1>=7.5 44 10.6363600 0.40909090

12) DQB.a2< 2.5 21 3.8095240 0.23809520 \*

13) DQB.a2>=2.5 23 5.6521740 0.56521740 \*

7) DQA.a1< 7.5 20 0.0000000 1.00000000 \*

> printcp(tree1)

Regression tree:

rpart(formula = y ~ ., data = data)

Variables actually used in tree construction:

[1] DQA.a1 DQA.a2 DQB.a2 DQR.a1 DQR.a2

Root node error: 61.395/271 = 0.22655

n= 271

CP nsplit rel error xerror xstd

1 0.083184 0 1.00000 1.00480 0.039329

2 0.078470 1 0.91682 1.01619 0.049349

3 0.078201 2 0.83835 0.97963 0.052286

4 0.048416 3 0.76015 0.77507 0.050811

5 0.034810 4 0.71173 0.77181 0.057755

6 0.033020 6 0.64211 0.79324 0.061638

7 0.024726 7 0.60909 0.76650 0.062008

8 0.019133 8 0.58436 0.76011 0.064147

9 0.019003 9 0.56523 0.76532 0.065879

10 0.017102 10 0.54623 0.75274 0.066127

11 0.010912 11 0.52912 0.70882 0.066520

12 0.010000 12 0.51821 0.68988 0.065026

> best1 <- tree1$cptable[which.min(tree1$cptable[, "xerror"]), "CP"]

> best1

[1] 0.01

> # Tạo cây được cắt tỉa

> prunned\_tree <- prune(tree1, cp= best1)

> install.packages('rpart.plot')

Error in install.packages : Updating loaded packages

> #install.packages('rpart.plot')

> library(rpart.plot)

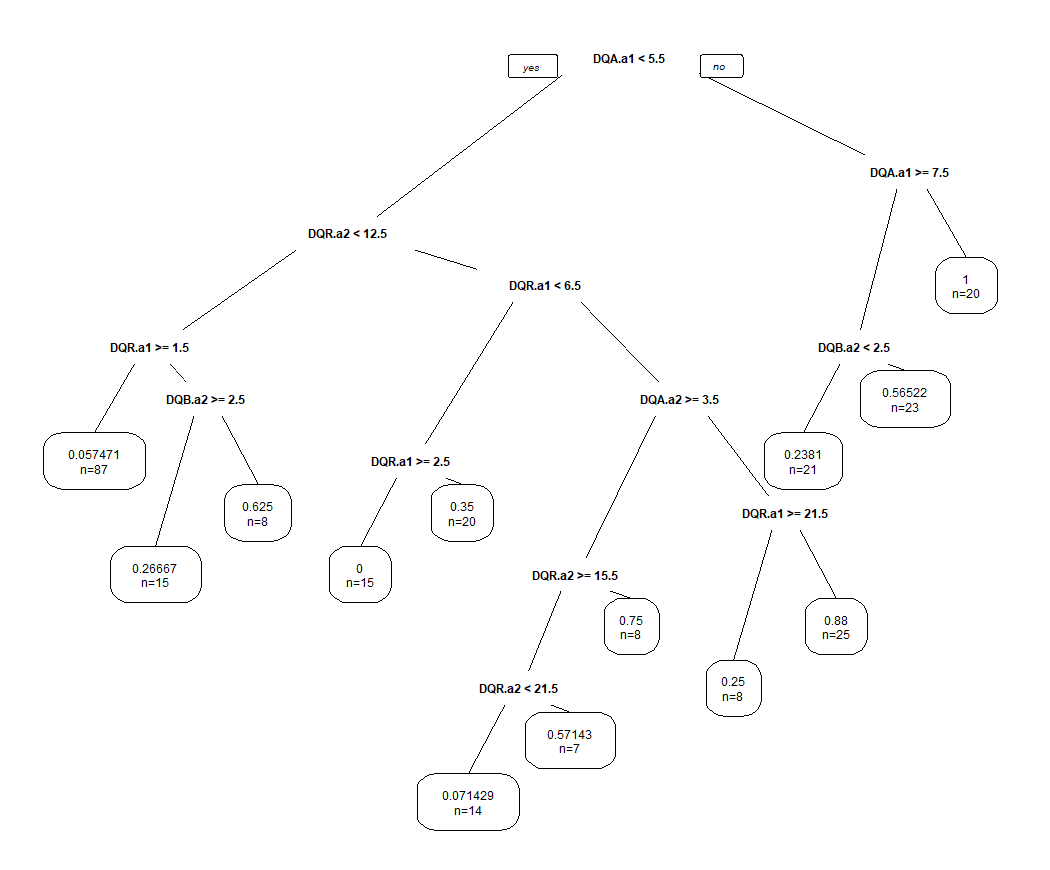
> prp(prunned\_tree,

+ faclen = 0, # sử dụng tên đầy đủ cho nhãn

+ extra = 1, # hiển thị số lượng quan sát cho mỗi nút

+ roundint = F, # Không làm tròn

+ digits = 5) # hiển thị tối đa 5 chữ số thập phân có ý nghĩa



#10.6 The promotergene data set in the kernlab is another example of a discrimination problem in a large data set. This data set contains 106 gene sequences, classified as being either promoters or not. The explanatory variables are the lists of DNA base values preceding the gene of interest. The base values take the values a, c, g, or t corresponding to adenine, cytosine, guanine, or thymine, respectively. Use the base values to find a rule for discriminating between promoters and non-promoters. Additional information on this data can be found on the UCI website

> library(kernlab)

> data(promotergene)

> library(rpart)

> tree2 <- rpart(Class ~ ., data = promotergene)

> tree2

n= 106

node), split, n, loss, yval, (yprob)

\* denotes terminal node

1) root 106 53 + (0.50000000 0.50000000)

2) V16=t 54 11 + (0.79629630 0.20370370)

4) V40=a,t 41 3 + (0.92682927 0.07317073) \*

5) V40=c,g 13 5 - (0.38461538 0.61538462) \*

3) V16=a,c,g 52 10 - (0.19230769 0.80769231)

6) V18=g 19 9 + (0.52631579 0.47368421) \*

7) V18=a,c,t 33 0 - (0.00000000 1.00000000) \*

> printcp(tree2)

Classification tree:

rpart(formula = Class ~ ., data = promotergene)

Variables actually used in tree construction:

[1] V16 V18 V40

Root node error: 53/106 = 0.5

n= 106

CP nsplit rel error xerror xstd

1 0.603774 0 1.00000 1.22642 0.094606

2 0.056604 1 0.39623 0.64151 0.090673

3 0.018868 2 0.33962 0.64151 0.090673

4 0.010000 3 0.32075 0.62264 0.089948

> best2 <- tree2$cptable[which.min(tree2$cptable[, "xerror"]), "CP"]

> best2

[1] 0.01

> # Tạo cây được cắt tỉa

> prunned\_tree2 <- prune(tree2, cp= best2)

> library(rpart.plot)

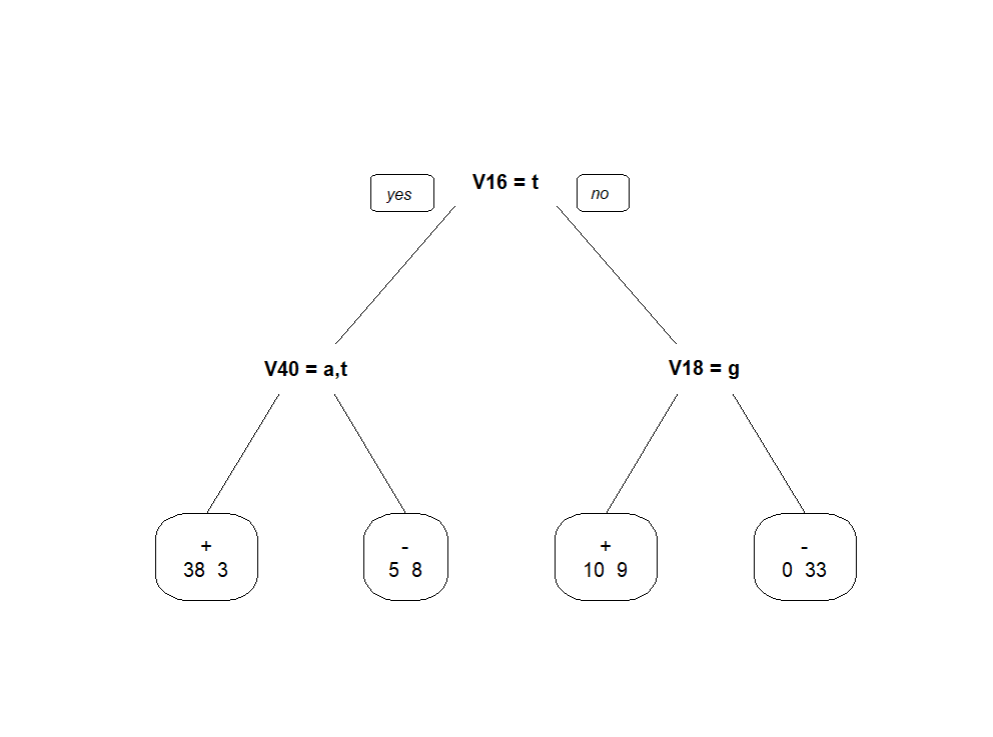
> prp(prunned\_tree2,

+ faclen = 0, # sử dụng tên đầy đủ cho nhãn

+ extra = 1, # hiển thị số lượng quan sát cho mỗi nút

+ roundint = F, # Không làm tròn

+ digits = 5) # hiển thị tối đa 5 chữ số thập phân có ý nghĩa



> # 10.7 The spam data set in the kernlab library was gathered in an attempt to distinguish emails that are advertisements (spam) from non-spam. There are 4601 email messages collected by Hewlett-Packard and of these, 1813 are classified as spam and 2788 are non-spam. There are 57 explanatory variables that measure the frequency of various words such as “free” or “credit.” Other explanatory variables count the frequency of different punctuation symbols or the length of strings of all capital letters. See the help file for a detailed explanation of the variables in this data set. Can you find a good rule to distinguish spam from non-spam?

> library(kernlab)

> data(spam)

> library(rpart)

> tree3 <- rpart(type ~ ., data = spam)

> tree3

n= 4601

node), split, n, loss, yval, (yprob)

\* denotes terminal node

1) root 4601 1813 nonspam (0.60595523 0.39404477)

2) charDollar< 0.0555 3471 816 nonspam (0.76490925 0.23509075)

4) remove< 0.055 3141 516 nonspam (0.83572111 0.16427889)

8) charExclamation< 0.378 2737 275 nonspam (0.89952503 0.10047497) \*

9) charExclamation>=0.378 404 163 spam (0.40346535 0.59653465)

18) capitalTotal< 55.5 182 52 nonspam (0.71428571 0.28571429)

36) free< 0.845 161 32 nonspam (0.80124224 0.19875776) \*

37) free>=0.845 21 1 spam (0.04761905 0.95238095) \*

19) capitalTotal>=55.5 222 33 spam (0.14864865 0.85135135) \*

5) remove>=0.055 330 30 spam (0.09090909 0.90909091) \*

3) charDollar>=0.0555 1130 133 spam (0.11769912 0.88230088)

6) hp>=0.4 70 7 nonspam (0.90000000 0.10000000) \*

7) hp< 0.4 1060 70 spam (0.06603774 0.93396226) \*

> printcp(tree3)

Classification tree:

rpart(formula = type ~ ., data = spam)

Variables actually used in tree construction:

[1] capitalTotal charDollar charExclamation free hp remove

Root node error: 1813/4601 = 0.39404

n= 4601

CP nsplit rel error xerror xstd

1 0.476558 0 1.00000 1.00000 0.018282

2 0.148924 1 0.52344 0.54550 0.015369

3 0.043023 2 0.37452 0.47711 0.014618

4 0.030888 4 0.28847 0.33149 0.012608

5 0.010480 5 0.25758 0.29564 0.012003

6 0.010000 6 0.24710 0.27910 0.011705

> best3 <- tree3$cptable[which.min(tree3$cptable[, "xerror"]), "CP"]

> best3

[1] 0.01

> # Tạo cây được cắt tỉa

> prunned\_tree3 <- prune(tree3, cp= best3)

> library(rpart.plot)

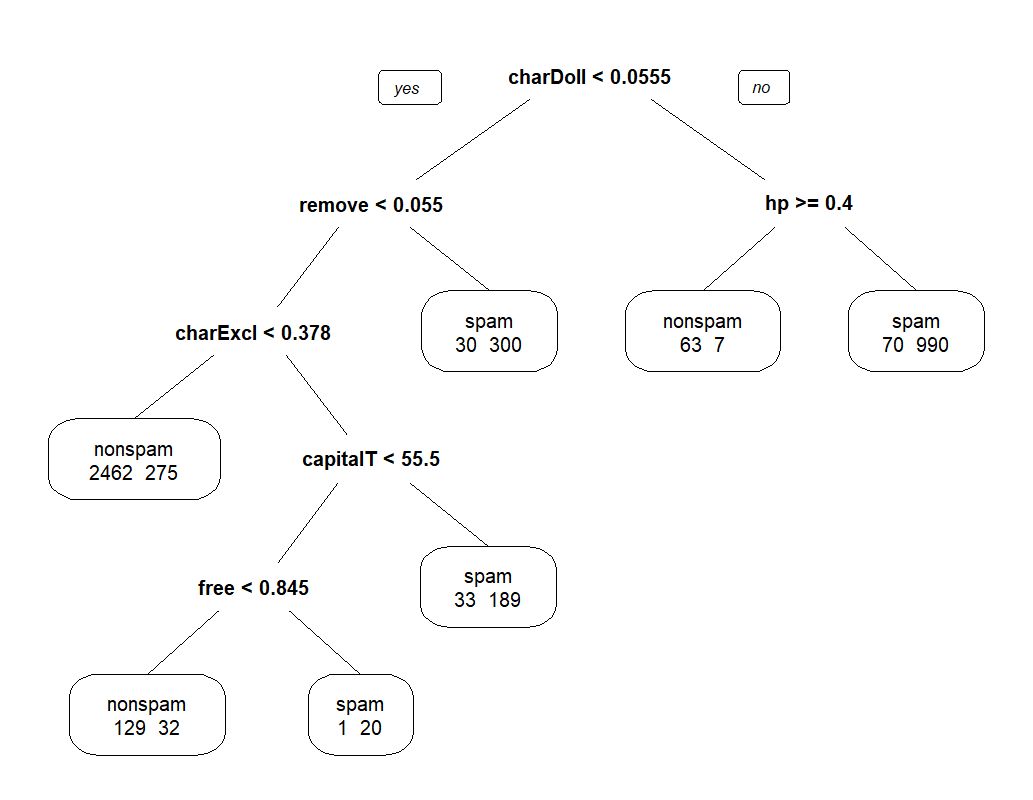
> prp(prunned\_tree3,

+ faclen = 0, # sử dụng tên đầy đủ cho nhãn

+ extra = 1, # hiển thị số lượng quan sát cho mỗi nút

+ roundint = F, # Không làm tròn

+ digits = 5) # hiển thị tối đa 5 chữ số thập phân có ý nghĩa



> # 10.12 The data frame stagec in the rpart library summarizes observations made on 146 men treated for Stage C prostate cancer. There are eight measurements made on each patient including whether or not the cancer has progressed. Use regression tree methods to indicate important prognostic measures on whether the cancer will progress from this data.

> library(rpart)

> data(stagec)

> tree4 <- rpart(pgstat ~ ., data = stagec)

> tree4

n= 146

node), split, n, deviance, yval

\* denotes terminal node

1) root 146 34.027400 0.36986300

2) pgtime>=4.45 102 14.166670 0.16666670

4) g2< 13.2 61 3.737705 0.06557377

8) gleason< 6.5 45 0.000000 0.00000000 \*

9) gleason>=6.5 16 3.000000 0.25000000 \*

5) g2>=13.2 41 8.878049 0.31707320

10) pgtime>=8.1 13 0.000000 0.00000000 \*

11) pgtime< 8.1 28 6.964286 0.46428570

22) g2>=17.6 12 2.250000 0.25000000 \*

23) g2< 17.6 16 3.750000 0.62500000 \*

3) pgtime< 4.45 44 5.886364 0.84090910

6) pgtime>=3.05 17 3.529412 0.70588240 \*

7) pgtime< 3.05 27 1.851852 0.92592590 \*

> printcp(tree4)

Regression tree:

rpart(formula = pgstat ~ ., data = stagec)

Variables actually used in tree construction:

[1] g2 gleason pgtime

Root node error: 34.027/146 = 0.23306

n= 146

CP nsplit rel error xerror xstd

1 0.410680 0 1.00000 1.01720 0.045574

2 0.050910 1 0.58932 0.71639 0.097092

3 0.028339 3 0.48750 0.75612 0.104451

4 0.021680 4 0.45916 0.77418 0.105884

5 0.014844 5 0.43748 0.80540 0.107960

6 0.010000 6 0.42264 0.78575 0.103671

> # Tạo cây được cắt tỉa

> prunned\_tree4 <- prune(tree4, cp= best4)

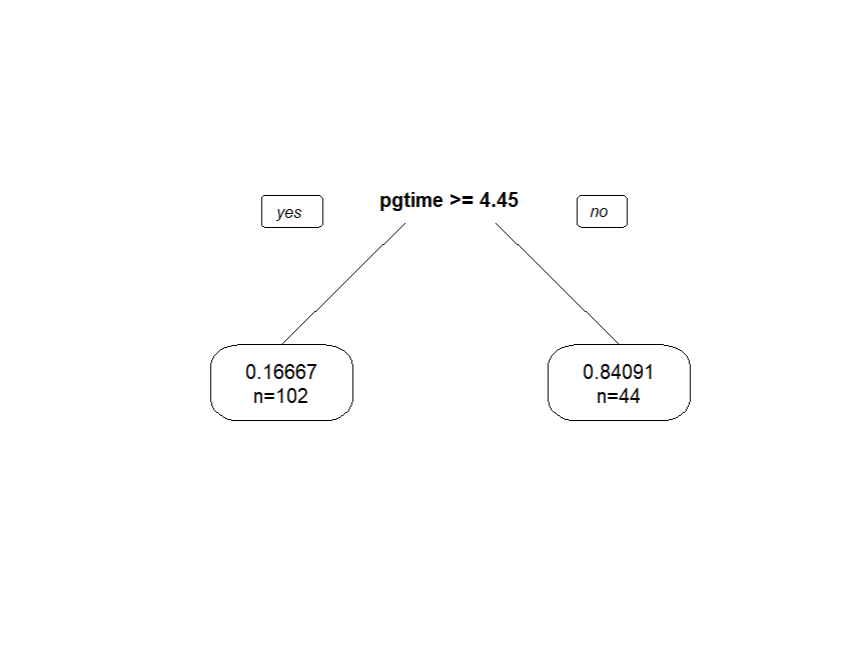
> prp(prunned\_tree4,

+ faclen = 0, # sử dụng tên đầy đủ cho nhãn

+ extra = 1, # hiển thị số lượng quan sát cho mỗi nút

+ roundint = F, # Không làm tròn

+ digits = 5) # hiển thị tối đa 5 chữ số thập phân có ý nghĩa



# 10.13 Kyphosis is a severe curvature of the spine. The kyphosis data frame in the rpart library presents history of 81 children who had corrective spinal surgery and whether this resulted in kyphosis. The data frame includes three additional variables indicating the extent of the surgery and the child’s age in months.

> #(a) Use regression trees to identify risk factors for kyphosis.

> #(b) Similarly, use logistic regression to build a linear model of thelog-odds risk of kyphosis.

> #(c) Discuss these two different approaches to modeling kyphosis.Which model is more intuitive? Which is easier to explain and use?

> # a)

> library(rpart)

> data(kyphosis)

> tree5 <- rpart(Kyphosis ~ ., data = kyphosis)

> tree5

n= 81

node), split, n, loss, yval, (yprob)

\* denotes terminal node

1) root 81 17 absent (0.79012346 0.20987654)

2) Start>=8.5 62 6 absent (0.90322581 0.09677419)

4) Start>=14.5 29 0 absent (1.00000000 0.00000000) \*

5) Start< 14.5 33 6 absent (0.81818182 0.18181818)

10) Age< 55 12 0 absent (1.00000000 0.00000000) \*

11) Age>=55 21 6 absent (0.71428571 0.28571429)

22) Age>=111 14 2 absent (0.85714286 0.14285714) \*

23) Age< 111 7 3 present (0.42857143 0.57142857) \*

3) Start< 8.5 19 8 present (0.42105263 0.57894737) \*

> printcp(tree5)

Classification tree:

rpart(formula = Kyphosis ~ ., data = kyphosis)

Variables actually used in tree construction:

[1] Age Start

Root node error: 17/81 = 0.20988

n= 81

CP nsplit rel error xerror xstd

1 0.176471 0 1.00000 1.0000 0.21559

2 0.019608 1 0.82353 1.1176 0.22433

3 0.010000 4 0.76471 1.1176 0.22433

> best5 <- tree5$cptable[which.min(tree5$cptable[, "xerror"]), "CP"]

> best5

[1] 0.1764706

> # Tạo cây được cắt tỉa

> prunned\_tree5 <- prune(tree5, cp= best5)

> library(rpart.plot)

> prp(prunned\_tree5,

+ faclen = 0, # sử dụng tên đầy đủ cho nhãn

+ extra = 1, # hiển thị số lượng quan sát cho mỗi nút

+ roundint = F, # Không làm tròn

+ digits = 5) # hiển thị tối đa 5 chữ số thập phân có ý nghĩa

> # b)

> library(nnet)

> kyphosis$class <- as.factor(kyphosis$Kyphosis)

> kyphosis$rClass <- relevel(kyphosis$Kyphosis, ref = 2)

> kyphosislogit <- multinom(rClass ~ ., data = kyphosis)

# weights: 7 (6 variable)

initial value 56.144922

iter 10 value 0.179818

iter 20 value 0.002084

iter 30 value 0.001728

iter 40 value 0.001045

iter 50 value 0.000742

iter 60 value 0.000690

iter 70 value 0.000616

iter 80 value 0.000564

iter 90 value 0.000541

iter 100 value 0.000502

final value 0.000502

stopped after 100 iterations

> print(ws <- summary(kyphosislogit), digits = 4)

Call:

multinom(formula = rClass ~ ., data = kyphosis)

Coefficients:

Values Std. Err.

(Intercept) 7.771689 166.4184

Kyphosispresent -12.300162 73.9920

Age 0.006211 0.9895

Number 0.454269 28.6584

Start 0.245392 10.1257

classpresent -12.300162 73.9920

Residual Deviance: 0.001003053

AIC: 10.001

> # Kiểm định các biến có ý nghĩa trong mô hình 0

> tratio <- ws$coefficients/ws$standard.errors

> print(2 \* (1 - pt(abs(tratio), df = ws$edf)), digits = 4)

(Intercept) Kyphosispresent Age Number Start classpresent

0.9646 0.8745 0.9952 0.9880 0.9816 0.8745

> ws$fitted.values # xác suất ươc tính mỗi loại rượu thuộc nhóm i

[,1]

1 9.999797e-01

2 9.999987e-01

3 2.283920e-06

4 9.999664e-01

5 9.999983e-01

6 9.999967e-01

7 9.999982e-01

8 9.999983e-01

9 9.999983e-01

10 2.056549e-05

11 2.460558e-05

12 9.999992e-01

13 9.999762e-01

14 9.999964e-01

15 9.999995e-01

16 9.999979e-01

17 9.999996e-01

18 9.999994e-01

19 9.999995e-01

20 9.999936e-01

21 9.999971e-01

22 4.911510e-06

23 6.623467e-06

24 9.999639e-01

25 2.118496e-06

26 9.999983e-01

27 9.999765e-01

28 9.999981e-01

29 9.999979e-01

30 9.999987e-01

31 9.999982e-01

32 9.999947e-01

33 9.999992e-01

34 9.999989e-01

35 9.999988e-01

36 9.999988e-01

37 9.999882e-01

38 2.867769e-06

39 9.999973e-01

40 1.592826e-05

41 9.579136e-07

42 9.999964e-01

43 9.999986e-01

44 9.999633e-01

45 9.999988e-01

46 5.295726e-06

47 9.999993e-01

48 9.999992e-01

49 8.499018e-07

50 9.999982e-01

51 9.999975e-01

52 9.999975e-01

53 4.771165e-05

54 9.999974e-01

55 9.999993e-01

56 9.999993e-01

57 9.999956e-01

58 7.146680e-06

59 9.999986e-01

60 9.999976e-01

61 8.665356e-07

62 1.708007e-05

63 9.999676e-01

64 9.999990e-01

65 9.999994e-01

66 9.999947e-01

67 9.999992e-01

68 9.999989e-01

69 9.999959e-01

70 9.999992e-01

71 9.999995e-01

72 9.999984e-01

73 9.999992e-01

74 9.999984e-01

75 9.999975e-01

76 9.999994e-01

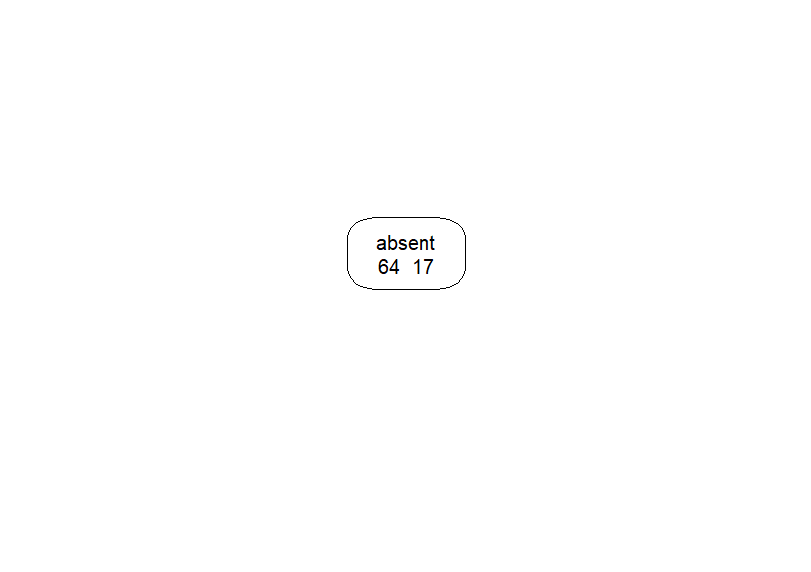
77 1.236492e-05

78 9.999994e-01

79 9.999967e-01

80 6.685638e-06

81 9.999977e-01



# Về mặt trực quan và dễ giải thích:

# Cây hồi quy: Nói chung được coi là trực quan hơn, đặc biệt đối với đối tượng không chuyên về kỹ thuật. Quy trình ra quyết định rõ ràng và dễ theo dõi.

# Hồi quy Logistic: Ít trực quan hơn, đặc biệt là việc giải thích hệ số. Cần có một số hiểu biết về thống kê và cách hoạt động của hồi quy logistic.

#Về mặt dễ sử dụng:

# Cây hồi quy: Tương đối dễ sử dụng với các hàm tích hợp trong thư viện R như rpart. Tuy nhiên, việc điều chỉnh và cắt tỉa cây có thể đòi hỏi một số thử nghiệm.

# Hồi quy Logistic: Cũng tương đối dễ sử dụng với các hàm như glm trong R. Việc giải thích có thể khó khăn hơn so với cây hồi quy.

#10.14 The data set mammoexp in the TH.data library describes a survey of 412 women’s attitudes to and experiences with mammography. Among the data are the respondent’s perceived benefits of mammography and whether a close female relative had been diagnosed with breast cancer. A suitable response to these measurements is the variable ME that measures the time since the woman had her last mammography. The categorical responses to this variable are never, within a year, and over a year ago. Use a regression tree approach to model the time since last mammography.

> library(TH.data)

> data(mammoexp)

> tree6 <- rpart(ME ~ ., data = mammoexp)

> tree6

n= 412

node), split, n, loss, yval, (yprob)

\* denotes terminal node

1) root 412 178 Never (0.56796117 0.25242718 0.17961165)

2) SYMPT=Strongly Agree,Agree 113 18 Never (0.84070796 0.05309735 0.10619469) \*

3) SYMPT=Disagree,Strongly Disagree 299 160 Never (0.46488294 0.32775920 0.20735786)

6) PB>=9.5 51 16 Never (0.68627451 0.13725490 0.17647059) \*

7) PB< 9.5 248 144 Never (0.41935484 0.36693548 0.21370968)

14) HIST=No 216 117 Never (0.45833333 0.33796296 0.20370370)

28) DECT=Not likely,Somewhat likely 46 17 Never (0.63043478 0.13043478 0.23913043) \*

29) DECT=Very likely 170 100 Never (0.41176471 0.39411765 0.19411765)

58) BSE=No 15 3 Never (0.80000000 0.20000000 0.00000000) \*

59) BSE=Yes 155 91 Within a Year (0.37419355 0.41290323 0.21290323)

118) SYMPT=Strongly Agree,Agree,Disagree 69 38 Never (0.44927536 0.37681159 0.17391304) \*

119) SYMPT=Strongly Disagree 86 48 Within a Year (0.31395349 0.44186047 0.24418605) \*

15) HIST=Yes 32 14 Within a Year (0.15625000 0.56250000 0.28125000) \*

> printcp(tree6)

Classification tree:

rpart(formula = ME ~ ., data = mammoexp)

Variables actually used in tree construction:

[1] BSE DECT HIST PB SYMPT

Root node error: 178/412 = 0.43204

n= 412

CP nsplit rel error xerror xstd

1 0.024345 0 1.00000 1.0000 0.056487

2 0.016854 3 0.92697 1.0449 0.056747

3 0.010000 6 0.86517 1.0337 0.056690

> best6 <- tree6$cptable[which.min(tree6$cptable[, "xerror"]), "CP"]

> best6

[1] 0.02434457

> # Tạo cây được cắt tỉa

> prunned\_tree6 <- prune(tree6, cp= best6)

> library(rpart.plot)

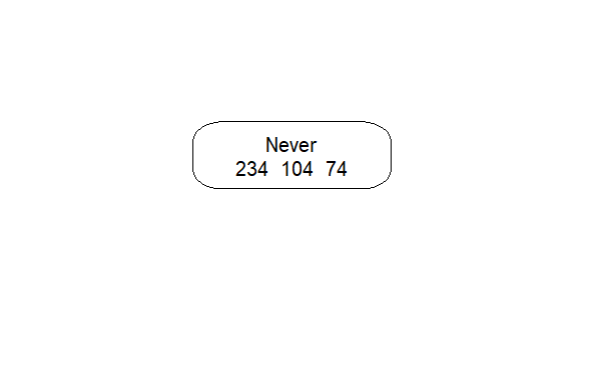
> prp(prunned\_tree6,

+ faclen = 0, # sử dụng tên đầy đủ cho nhãn

+ extra = 1, # hiển thị số lượng quan sát cho mỗi nút

+ roundint = F, # Không làm tròn

+ digits = 5) # hiển thị tối đa 5 chữ số thập phân có ý nghĩa



> # 10.16 Regression trees are not limited to classifying individuals into discrete groups. These methods will also estimate cut points in continuous measurements and thereby create groups of data. Build a regression tree to model the mpg values in the car data introduced in Table 9.1. Show that the most important classification is the number of cylinders and, to a smaller degree, the displacement

>

> # 10.16 Regression trees are not limited to classifying individuals into discrete groups. These methods will also estimate cut points in continuous measurements and thereby create groups of data. Build a regression tree to model the mpg values in the car data introduced in Table 9.1. Show that the most important classification is the number of cylinders and, to a smaller degree, the displacement

> library(MASS)

> data(mtcars)

> tree7 <- rpart(mpg ~ ., data = mtcars)

> tree7

n= 32

node), split, n, deviance, yval

\* denotes terminal node

1) root 32 1126.04700 20.09062

2) cyl>=5 21 198.47240 16.64762

4) hp>=192.5 7 28.82857 13.41429 \*

5) hp< 192.5 14 59.87214 18.26429 \*

3) cyl< 5 11 203.38550 26.66364 \*

> printcp(tree7)

Regression tree:

rpart(formula = mpg ~ ., data = mtcars)

Variables actually used in tree construction:

[1] cyl hp

Root node error: 1126/32 = 35.189

n= 32

CP nsplit rel error xerror xstd

1 0.643125 0 1.00000 1.05407 0.25121

2 0.097484 1 0.35687 0.70721 0.18020

3 0.010000 2 0.25939 0.57555 0.12780

> best7 <- tree7$cptable[which.min(tree7$cptable[, "xerror"]), "CP"]

> best7

[1] 0.01

> # Tạo cây được cắt tỉa

> prunned\_tree7 <- prune(tree7, cp= best7)

> library(rpart.plot)

> prp(prunned\_tree7,

+ faclen = 0, # sử dụng tên đầy đủ cho nhãn

+ extra = 1, # hiển thị số lượng quan sát cho mỗi nút

+ roundint = F, # Không làm tròn

+ digits = 5) # hiển thị tối đa 5 chữ số thập phân có ý nghĩa

