Lab 4: Multivariate normal and LDA, QDA

Zhanrui Cai

2022-09-14

## Exercise 1 Classification Trees

Bronchopulmonary Dysplasia (BPD) Study (Biostatistics Casebook, pp 104-119)

Training samples consist of all infants at the Stanford Medical Center between 1962 and 1973 who were diagnosed with respiratory distress syndrome (RDS) and received ventilatory assistance for at least 24 hours (except one infant with incomplete records). Most of these babies were born prematurely and had underdeveloped lungs. Some breathing assistance involving elevated levels of oxygen was needed to keep the babies alive. Bronchiopulmonary dysplasia (BPD) is deterioration of lung tissue (scarring) in infants exposed to a high level of oxygen. A panel of physicians reviewed each case to determine if BPD was present.

Infants who did not survive for at least 72 hours were excluded from the analysis because there was not enough time for BPD to develop. One additional infant was excluded due to incomplete records. This reduced the sample from 299 to 248 babies, including 78 with BPD and 170 without BPD.

General background variables: Sex (0=female, 1=male) YOB: year of birth (coded from 62 to 73) APGAR: one minute APGAR score (0 to 10 with 10 as the most healthy) GEST: gestational age (weeks × 10) BWT: birth weight (grams) AGSYM: age at onset of RDS (hours × 10) RDS: severity of initial X-ray for RDS (0 to 5=most severe) Therapy variables: AGVEN: Age at onset of ventilation (hours) VENTL: total hours on the ventilator LOWO2: hours of exposure to (21-39%) levels of oxygen MEDO2: hours of exposure to (40-79%) levels of oxygen HIO2: hours of exposure to (80-100%) levels of oxygen INTUB: hours of endotracheal intubation Response variable: BPD (coded 1=yes, 2=no)

1. Attach the rpart and rpart.plot packages.

set.seed(123)  
library(rpart)  
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.1.3

1. Construct a classification tree. First convert categorical variables like sex and rds to factors. (There is no need to convert categorical variables into zero-one variables, because it would not change the tree that is produced. Read the data into a data frame and create factors.)

bpdr <- read.csv("bpd.csv", header=T)  
bpdr$sex <- as.factor(bpdr$sex)  
head(bpdr)

## id sex yob apgar gest bwt agsym rds agven intub ventl lowo2 medo2 hio2 bpd  
## 1 5 1 64 8 400 3203 10 1 58 41 40 79 26 31 2  
## 2 1 0 62 <NA> 370 2948 0 0 30 88 77 102 89 0 2  
## 3 2 0 62 <NA> 360 2480 3 1 20 209 204 7 233 1 2  
## 4 6 1 64 <NA> 320 1860 0 2 1 653 644 0 625 156 1  
## 5 7 1 64 9 NA 1701 5 2 49 40 40 0 12 79 2  
## 6 8 0 64 <NA> 310 1417 5 5 17 103 102 0 3 119 2

1. Create a factor to distinguish the two populations with labels “BPD” and “No BPD”.

bpdr$y[bpdr$bpd==1]<-'BPD'  
bpdr$y[bpdr$bpd==2]<-'No BPD'  
bpdr$y<-as.factor(bpdr$y)  
  
head(bpdr)

## id sex yob apgar gest bwt agsym rds agven intub ventl lowo2 medo2 hio2 bpd  
## 1 5 1 64 8 400 3203 10 1 58 41 40 79 26 31 2  
## 2 1 0 62 <NA> 370 2948 0 0 30 88 77 102 89 0 2  
## 3 2 0 62 <NA> 360 2480 3 1 20 209 204 7 233 1 2  
## 4 6 1 64 <NA> 320 1860 0 2 1 653 644 0 625 156 1  
## 5 7 1 64 9 NA 1701 5 2 49 40 40 0 12 79 2  
## 6 8 0 64 <NA> 310 1417 5 5 17 103 102 0 3 119 2  
## y  
## 1 No BPD  
## 2 No BPD  
## 3 No BPD  
## 4 BPD  
## 5 No BPD  
## 6 No BPD

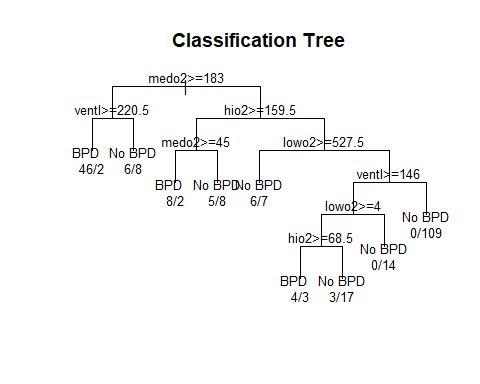
1. Train the decision tree using the function rpart(). Print out the summary of the tree. Which variable did the tree choose to split first? Medo2

bpd.rp <- rpart(y ~ sex+yob+gest+bwt+agsym+agven+intub+ventl+lowo2+medo2+hio2+rds, data=bpdr, cp=0.0001)  
  
summary(bpd.rp)

## Call:  
## rpart(formula = y ~ sex + yob + gest + bwt + agsym + agven +   
## intub + ventl + lowo2 + medo2 + hio2 + rds, data = bpdr,   
## cp = 1e-04)  
## n= 248   
##   
## CP nsplit rel error xerror xstd  
## 1 0.538461538 0 1.0000000 1.0000000 0.09374569  
## 2 0.038461538 1 0.4615385 0.5897436 0.07847551  
## 3 0.025641026 3 0.3846154 0.5769231 0.07780972  
## 4 0.003205128 4 0.3589744 0.5769231 0.07780972  
## 5 0.000100000 8 0.3461538 0.5897436 0.07847551  
##   
## Variable importance  
## medo2 intub ventl lowo2 hio2 agven yob   
## 31 19 19 13 10 6 1   
##   
## Node number 1: 248 observations, complexity param=0.5384615  
## predicted class=No BPD expected loss=0.3145161 P(node) =1  
## class counts: 78 170  
## probabilities: 0.315 0.685   
## left son=2 (62 obs) right son=3 (186 obs)  
## Primary splits:  
## medo2 < 183 to the right, improve=45.43011, (0 missing)  
## ventl < 220.5 to the right, improve=43.28307, (1 missing)  
## intub < 271 to the right, improve=39.85744, (0 missing)  
## lowo2 < 437.5 to the right, improve=29.52134, (0 missing)  
## hio2 < 153 to the right, improve=17.09879, (0 missing)  
## Surrogate splits:  
## intub < 286.5 to the right, agree=0.871, adj=0.484, (0 split)  
## ventl < 301.5 to the right, agree=0.863, adj=0.452, (0 split)  
## lowo2 < 429 to the right, agree=0.831, adj=0.323, (0 split)  
## agven < 64.5 to the right, agree=0.786, adj=0.145, (0 split)  
## hio2 < 305 to the right, agree=0.774, adj=0.097, (0 split)  
##   
## Node number 2: 62 observations, complexity param=0.02564103  
## predicted class=BPD expected loss=0.1612903 P(node) =0.25  
## class counts: 52 10  
## probabilities: 0.839 0.161   
## left son=4 (48 obs) right son=5 (14 obs)  
## Primary splits:  
## ventl < 220.5 to the right, improve=6.034381, (1 missing)  
## intub < 221.5 to the right, improve=4.680002, (0 missing)  
## hio2 < 45.5 to the right, improve=4.466501, (0 missing)  
## medo2 < 443.5 to the right, improve=4.181601, (0 missing)  
## lowo2 < 434 to the right, improve=1.743506, (0 missing)  
## Surrogate splits:  
## intub < 221.5 to the right, agree=0.984, adj=0.929, (1 split)  
## medo2 < 243 to the right, agree=0.820, adj=0.214, (0 split)  
## agven < 129 to the left, agree=0.803, adj=0.143, (0 split)  
## lowo2 < 1928 to the left, agree=0.787, adj=0.071, (0 split)  
## hio2 < 0.5 to the right, agree=0.787, adj=0.071, (0 split)  
##   
## Node number 3: 186 observations, complexity param=0.03846154  
## predicted class=No BPD expected loss=0.1397849 P(node) =0.75  
## class counts: 26 160  
## probabilities: 0.140 0.860   
## left son=6 (23 obs) right son=7 (163 obs)  
## Primary splits:  
## hio2 < 159.5 to the right, improve=9.500455, (0 missing)  
## ventl < 223.5 to the right, improve=5.863258, (0 missing)  
## intub < 229 to the right, improve=4.550538, (0 missing)  
## lowo2 < 321 to the right, improve=3.948499, (0 missing)  
## yob < 68.5 to the left, improve=2.688830, (0 missing)  
##   
## Node number 4: 48 observations  
## predicted class=BPD expected loss=0.04166667 P(node) =0.1935484  
## class counts: 46 2  
## probabilities: 0.958 0.042   
##   
## Node number 5: 14 observations  
## predicted class=No BPD expected loss=0.4285714 P(node) =0.05645161  
## class counts: 6 8  
## probabilities: 0.429 0.571   
##   
## Node number 6: 23 observations, complexity param=0.03846154  
## predicted class=BPD expected loss=0.4347826 P(node) =0.09274194  
## class counts: 13 10  
## probabilities: 0.565 0.435   
## left son=12 (10 obs) right son=13 (13 obs)  
## Primary splits:  
## medo2 < 45 to the right, improve=1.950502, (0 missing)  
## rds splits as --LLR-R, improve=1.950502, (0 missing)  
## agven < 32.5 to the right, improve=1.715062, (0 missing)  
## lowo2 < 4.5 to the right, improve=1.715062, (0 missing)  
## yob < 68.5 to the left, improve=1.590062, (0 missing)  
## Surrogate splits:  
## agven < 37.5 to the right, agree=0.826, adj=0.6, (0 split)  
## lowo2 < 17.5 to the right, agree=0.783, adj=0.5, (0 split)  
## intub < 172 to the left, agree=0.739, adj=0.4, (0 split)  
## ventl < 169.5 to the left, agree=0.739, adj=0.4, (0 split)  
## yob < 65.5 to the left, agree=0.696, adj=0.3, (0 split)  
##   
## Node number 7: 163 observations, complexity param=0.003205128  
## predicted class=No BPD expected loss=0.0797546 P(node) =0.6572581  
## class counts: 13 150  
## probabilities: 0.080 0.920   
## left son=14 (13 obs) right son=15 (150 obs)  
## Primary splits:  
## lowo2 < 527.5 to the right, improve=4.1181750, (0 missing)  
## ventl < 250.5 to the right, improve=1.9012610, (0 missing)  
## intub < 433.5 to the right, improve=1.7798600, (0 missing)  
## medo2 < 82.5 to the right, improve=1.6898960, (0 missing)  
## rds splits as RRRLRLL, improve=0.5579593, (0 missing)  
## Surrogate splits:  
## intub < 390 to the right, agree=0.945, adj=0.308, (0 split)  
## ventl < 288.5 to the right, agree=0.939, adj=0.231, (0 split)  
##   
## Node number 12: 10 observations  
## predicted class=BPD expected loss=0.2 P(node) =0.04032258  
## class counts: 8 2  
## probabilities: 0.800 0.200   
##   
## Node number 13: 13 observations  
## predicted class=No BPD expected loss=0.3846154 P(node) =0.05241935  
## class counts: 5 8  
## probabilities: 0.385 0.615   
##   
## Node number 14: 13 observations  
## predicted class=No BPD expected loss=0.4615385 P(node) =0.05241935  
## class counts: 6 7  
## probabilities: 0.462 0.538   
##   
## Node number 15: 150 observations, complexity param=0.003205128  
## predicted class=No BPD expected loss=0.04666667 P(node) =0.6048387  
## class counts: 7 143  
## probabilities: 0.047 0.953   
## left son=30 (41 obs) right son=31 (109 obs)  
## Primary splits:  
## ventl < 146 to the right, improve=1.7369110, (0 missing)  
## intub < 148.5 to the right, improve=1.4771010, (0 missing)  
## medo2 < 158 to the right, improve=1.2132460, (0 missing)  
## rds splits as RRRRRLR, improve=0.5901655, (0 missing)  
## agven < 4.5 to the left, improve=0.4336516, (0 missing)  
## Surrogate splits:  
## intub < 146 to the right, agree=0.953, adj=0.829, (0 split)  
## lowo2 < 206 to the right, agree=0.753, adj=0.098, (0 split)  
## agven < 5.5 to the left, agree=0.747, adj=0.073, (0 split)  
## medo2 < 133 to the right, agree=0.747, adj=0.073, (0 split)  
## hio2 < 144.5 to the right, agree=0.733, adj=0.024, (0 split)  
##   
## Node number 30: 41 observations, complexity param=0.003205128  
## predicted class=No BPD expected loss=0.1707317 P(node) =0.1653226  
## class counts: 7 34  
## probabilities: 0.171 0.829   
## left son=60 (27 obs) right son=61 (14 obs)  
## Primary splits:  
## lowo2 < 4 to the right, improve=1.2393860, (0 missing)  
## medo2 < 81.5 to the right, improve=1.2393860, (0 missing)  
## agven < 31 to the right, improve=1.1223610, (0 missing)  
## ventl < 155 to the left, improve=1.1223610, (0 missing)  
## rds splits as R-RLRLR, improve=0.8971124, (0 missing)  
## Surrogate splits:  
## hio2 < 82.5 to the left, agree=0.805, adj=0.429, (0 split)  
## bwt < 1078 to the right, agree=0.707, adj=0.143, (0 split)  
## agven < 12.5 to the right, agree=0.707, adj=0.143, (0 split)  
## yob < 67.5 to the right, agree=0.683, adj=0.071, (0 split)  
## gest < 275 to the right, agree=0.683, adj=0.071, (0 split)  
##   
## Node number 31: 109 observations  
## predicted class=No BPD expected loss=0 P(node) =0.4395161  
## class counts: 0 109  
## probabilities: 0.000 1.000   
##   
## Node number 60: 27 observations, complexity param=0.003205128  
## predicted class=No BPD expected loss=0.2592593 P(node) =0.108871  
## class counts: 7 20  
## probabilities: 0.259 0.741   
## left son=120 (7 obs) right son=121 (20 obs)  
## Primary splits:  
## hio2 < 68.5 to the right, improve=1.8417990, (0 missing)  
## medo2 < 76 to the right, improve=1.5282650, (0 missing)  
## lowo2 < 99 to the left, improve=1.4158250, (0 missing)  
## rds splits as R-RLRLL, improve=1.4158250, (0 missing)  
## ventl < 161.5 to the left, improve=0.9259259, (0 missing)  
## Surrogate splits:  
## yob < 68.5 to the left, agree=0.815, adj=0.286, (0 split)  
## gest < 390 to the right, agree=0.815, adj=0.286, (0 split)  
## bwt < 2976.5 to the right, agree=0.815, adj=0.286, (0 split)  
## ventl < 150 to the left, agree=0.815, adj=0.286, (0 split)  
## medo2 < 170.5 to the right, agree=0.815, adj=0.286, (0 split)  
##   
## Node number 61: 14 observations  
## predicted class=No BPD expected loss=0 P(node) =0.05645161  
## class counts: 0 14  
## probabilities: 0.000 1.000   
##   
## Node number 120: 7 observations  
## predicted class=BPD expected loss=0.4285714 P(node) =0.02822581  
## class counts: 4 3  
## probabilities: 0.571 0.429   
##   
## Node number 121: 20 observations  
## predicted class=No BPD expected loss=0.15 P(node) =0.08064516  
## class counts: 3 17  
## probabilities: 0.150 0.850

1. Using function plot() to display the tree. The arguments of the par( ) function enable all of the text for the node labels to be displayed, otherwise only part of the label can be seen. How many terminal nodes does the tree have? 9 terminal nodes

par(mfrow=c(1,1), xpd=NA)  
plot(bpd.rp,uniform=T)  
text(bpd.rp, use.n=TRUE, cex=0.8)  
title("Classification Tree")



1. Estimate the predicted probabilities of the two classes using the predict() function. Print the estimated probabilities for the first 10 samples using the head() function.

bpd.prob <- predict(bpd.rp)  
head(bpd.prob, 10)

## BPD No BPD  
## 1 0.0000000 1.00000000  
## 2 0.0000000 1.00000000  
## 3 0.4285714 0.57142857  
## 4 0.9583333 0.04166667  
## 5 0.0000000 1.00000000  
## 6 0.0000000 1.00000000  
## 7 0.0000000 1.00000000  
## 8 0.4285714 0.57142857  
## 9 0.0000000 1.00000000  
## 10 0.0000000 1.00000000

##### Get AUC  
  
library(pROC)

## Warning: package 'pROC' was built under R version 4.1.3

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

roc.curve = roc(bpdr$y, bpd.prob[,1])

## Setting levels: control = BPD, case = No BPD

## Setting direction: controls > cases

auc(roc.curve)

## Area under the curve: 0.9617

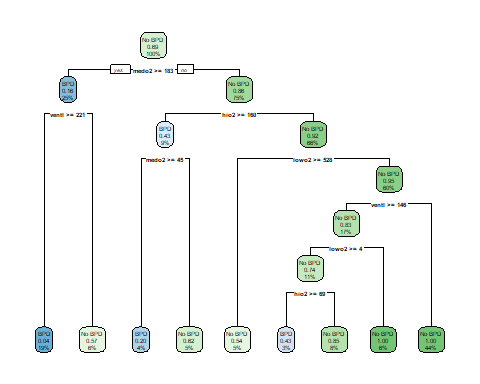
1. Make a table of the confusion matrix. The rows are the truth. What is the true positive rate? What is the false discovery rate? What is the prediction accuracy rate? 58/65 is the true positive rate. 7/65 is the false discovery rate. 221/248 is the prediction accuracy rate.

bpd.class <- rep( "Predicted BPD", nrow(bpd.prob))  
bpd.class[ bpd.prob[ ,2] > 0.5] <- "Predicted No BPD"  
table(bpdr$y, bpd.class)

## bpd.class  
## Predicted BPD Predicted No BPD  
## BPD 58 20  
## No BPD 7 163

1. Display the tree using the function rpart.plot().

rpart.plot(bpd.rp)



1. Based on 7, what is the predicted probability of having Bronchopulmonary Dysplasia if medo2>183 and ventl<220.5? What is the predicted probability of not having Bronchopulmonary Dysplasia if medo2<183 and hio2>159.5 and medo2<45?
2. 19 percent b).62
3. Based on 7, what is the percentage of samples satisfy that medo2>183? What is the percentage of samples that satisfy medo2<183 and hio2>160? 25 percent 9 percent ## Exercise 2 Out-of-sample prediction

In this exercise, we will learn how to perform training and testing on different datasets, i.e., out-of-sample prediction.

1. Create sub-datasets of bpd: bpd1 and bpd2, whose response is equal to 1 and 2, respectively.

bpd1 <- subset(bpdr, bpdr$bpd == 1)  
bpd1$rds <- as.factor(bpd1$rds)  
bpd2 <- subset(bpdr, bpdr$bpd == 2)  
bpd2$rds <- as.factor(bpd2$rds)

1. What is the sample size of bpd1 and bpd2? Report them as n1 and n2, respectively.

n1 <- dim(bpd1)[1]  
n2 <- dim(bpd2)[1]

1. Draw a random subset of bpd1 of size n1/2, name as bpd1.train. Name the remaining subset of bpd1 as bpd1.test.

sample1 <- sample(nrow(bpd1), size = n1/2, replace = F)  
bpd1.train <- bpd1[sample1, ]  
bpd1.test <- bpd1[-sample1, ]

1. Draw a random subset of bpd2 of size n2/2, name as bpd2.train. Name the remaining subset of bpd2 as bpd2.test.

sample2 <- sample(nrow(bpd2), size = n2/2, replace = F)  
bpd2.train <- bpd2[sample2, ]  
bpd2.test <- bpd2[-sample2, ]

1. Use the function rbind() to combine bpd1.train. and bpd2.train, name it as training.set.

training.set <- rbind(bpd1.train, bpd2.train)

1. Use the function rbind() to combine bpd1.test. and bpd2.test, name it as testing.set.

testing.set <- rbind(bpd1.test, bpd2.test)

1. Use the training.set to train the classification tree (with the function rpart). Predict the probabilities of the testing.set.

bpd.tree <- rpart(y ~ sex+yob+gest+bwt+agsym+agven+intub+ventl+lowo2+medo2+hio2+rds, data=training.set, cp=0.0001)

1. What is the prediction accuracy?

bpd.prob <- predict(bpd.tree, testing.set)  
  
y = as.numeric(testing.set$y)-1  
mean((bpd.prob[,2]>0.5)==y)

## [1] 0.8145161

1. What is the AUC?

library(pROC)  
roc.curve = roc(testing.set$y, bpd.prob[,1])

## Setting levels: control = BPD, case = No BPD

## Setting direction: controls > cases

auc(roc.curve)

## Area under the curve: 0.7814

1. Write a “for loop” to repeat the process in 1-8 for 20 times, report the mean of prediction accuracy.

accuracy\_all = rep(0, 20)  
auc\_all = rep(0, 20)  
  
for(ii in 1:20){  
   
 #### Fill in the code here  
   
 accuracy\_all[ii] = mean((bpd.prob[,2]>0.5)==y)  
 auc\_all[ii] = auc(roc.curve)  
}

## Exercise 3 Random Forests

We will use the same dataset as in exercise 1. We will use the R package randomForest.

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.1.3

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

bpdr1 <- na.omit(bpdr)  
  
bpd31 <- subset(bpdr1, bpdr1$bpd == 1)  
bpd31$rds <- as.factor(bpd31$rds)  
bpd32 <- subset(bpdr1, bpdr1$bpd == 2)  
bpd32$rds <- as.factor(bpd32$rds)  
num1 <- dim(bpd31)[1]  
num2 <- dim(bpd32)[1]  
sample31 <- sample(nrow(bpd31), size = num1/2, replace = F)  
bpd31.train <- bpd31[sample31, ]  
bpd31.test <- bpd31[-sample31, ]  
sample32 <- sample(nrow(bpd32), size = num2/2, replace = F)  
bpd32.train <- bpd32[sample32, ]  
bpd32.test <- bpd32[-sample32, ]  
training.set1 <- rbind(bpd31.train, bpd32.train)  
testing.set1 <- rbind(bpd31.test, bpd32.test)

1. Train the random forest model using the function randomForest(). Set the number of trees to be 50.

#### Fit random forest model, omit the NA missing values in the data  
rf1 = randomForest(y ~ sex+yob+gest+bwt+agsym+agven+intub+ventl+lowo2+medo2+hio2+rds, ntree = 50,  
 data=bpdr1, na.action=na.omit)

1. Estimate the predicted probabilities of the two classes using the predict() function. Print the estimated probabilities for the first 10 samples using the head() function.

bpd.rp1 <- predict(rf1, type = "prob", data = testing.set1)  
head(bpd.rp1, 10)

## BPD No BPD  
## 1 0.00000000 1.00000000  
## 7 0.00000000 1.00000000  
## 8 0.72727273 0.27272727  
## 11 0.05263158 0.94736842  
## 16 0.00000000 1.00000000  
## 24 0.16666667 0.83333333  
## 25 0.90909091 0.09090909  
## 26 0.21052632 0.78947368  
## 28 0.21052632 0.78947368  
## 29 1.00000000 0.00000000

y = as.numeric(bpdr1$y)-1  
mean((bpd.rp1[,2]>0.5)==y)

## [1] 0.8375635

##### Get AUC  
library(pROC)  
roc.curve = roc(bpdr1$y, bpd.rp1[,1])

## Setting levels: control = BPD, case = No BPD

## Setting direction: controls > cases

auc(roc.curve)

## Area under the curve: 0.9132

1. Make a table of the confusion matrix. The rows are the truth. What is the true positive rate? What is the false discovery rate? What is the prediction accuracy rate? 41/51 is the true positive rate. 10/51 is the false discovery rate and 174/197 is the prediction accuracy rate.

bpd.class1 <- rep( "Predicted BPD", nrow(bpd.rp1))  
bpd.class1[ bpd.rp1[ ,2] > 0.5] <- "Predicted No BPD"  
table(bpdr1$y, bpd.class1)

## bpd.class1  
## Predicted BPD Predicted No BPD  
## BPD 38 16  
## No BPD 16 127

1. Write a for loop as in exercise 2.9, but use random forest as the prediction model. Report the mean of prediction accuracy. Is it higher or lower compared to the single classification tree?

accuracy\_all1 = rep(0, 20)  
auc\_all1 = rep(0, 20)  
  
for(ii in 1:20){  
   
 #### Fill in the code here  
   
 accuracy\_all1[ii] = mean((bpd.rp1[,2]>0.5)==y)  
 auc\_all1[ii] = auc(roc.curve)  
}

1. Change the number of trees in (4) to be 25. Does the prediction accuracy increase?

rf1 = randomForest(y ~ sex+yob+gest+bwt+agsym+agven+intub+ventl+lowo2+medo2+hio2+rds, ntree = 25,  
 data=bpdr1, na.action=na.omit)