```
# Al Pakrosnis
# Homework 6
# Prof. Dale Embers
# STAT385 Sp25

setwd("~/Desktop/stat385/")

library(rpart)
library(rpart.plot)
library(e1071)
library(ISLR2)

# Q1 (a)
df <- read.table("homework6/Hemophilia-dat.txt", header = FALSE)
summary(df)
head(df)

colnames(df) <- c("group","AHF activity","AHF-like antigen")

set.seed(2024325)

train <- sample(1:dim(df)[1],60)

# (b)
tree <- rpart(group ~ ., data=df[train,],method="class", control=rpart.control(minsplit =
3, minbucket = 2, cp=0))

# (c)
prp(tree,type=2,extra=1) # plot here needs to be included

# (d)
# If a person had AHF activity -.14 and AHF-like antigen activity of .064, based on the
constructed tree, their predicted group would be group one as
# using these two criteria you move once to the left from the seed then again to the left
and thus you're in group one.

# (e)
predictions <- predict(tree, df[-train,], type="class")
table(df[-train,"group"],predictions)
# predictions
#   1 2
# 1 2 4
# 2 1 8

# (f)
nbmodel <- naiveBayes(group ~ ., data = df, subset=train)
nbpred <- predict(nbmodel, df[-train,])
table(df[-train,"group"],nbpred)
# nbpred
#   1 2
# 1 3 3
# 2 1 8

# Based on the two provided tables the naive bayes is marginally better at predicting than
the tree.

# (g)
px1<-seq(min(df[,2]),max(df[,2]),length=100)
px2<-seq(min(df[,3]),max(df[,3]),length=100)
xgrid<-expand.grid('AHF activity'=px1,'AHF-like antigen'=px2)
treepredict<- predict(tree,xgrid,type="class")
plot(xgrid, col = as.numeric(treepredict), pch = 20, cex = .2, main="Tree Method") # plot
here needs to be included
```

```
# (h)
nbpred <- predict(nbmodel, xgrid)
plot(xgrid, col = as.numeric(nbpred), pch = 20, cex = .2, main="NB method") # plot here
needs to be included

# The naive bayes method provides a region that is bulbous in shape, but both methods
generally separate a similar area in the plot. The classification tree
# plot is cut up into squares, which makes sense as the tree takes each leaf's area in
divides it up, thus it shows up as squares.


# Q2 (a)

set.seed(2024325)

ndf <- Default

train2 <- sample(1:nrow(ndf), .7*nrow(ndf))

# (b)

tree2 <- rpart(default~., data=ndf[train2,])
prp(tree2,type=2,extra=1) # plot here needs to be included

# (c)
printcp(tree2)
# Classification tree:
#   rpart(formula = default ~ ., data = ndf[train2, ])
#
# Variables actually used in tree construction:
#   [1] balance income
#
# Root node error: 236/7000 = 0.033714
#
# n= 7000
#
#         CP nsplit rel error  xerror     xstd
# 1 0.182203      0   1.00000 1.00000 0.063988
# 2 0.014831      1   0.81780 0.85593 0.059348
# 3 0.014124      3   0.78814 0.85593 0.059348
# 4 0.010000      6   0.74576 0.85169 0.059205

plotcp(tree2) # plot here needs to be included

# (d)
tree2_pruned <-prune(tree2, cp=0.0142)
par(mfrow= c(2,1))
prp(tree2,type=2,extra=1)
prp(tree2_pruned,type=2,extra=1) # plot here needs to be included

# (e)
# Dude ion even kno tbh... like you're pruning it so you're expecting it to be smaller
which it is, that's how one could've expected the pruned tree based on the results
# from part c and the pruning cp.

# (f)
pred <-predict(tree2, ndf[-train2,], type="class")
pred_pruned <-predict(tree2_pruned, ndf[-train2,], type="class")

table(ndf[-train2,"default"], pred)
#   pred
#       No  Yes
# No  2888   15
```

```
# Yes   69   28

err1 <- (15+69)/(2888+15+69+28)
err1
# 0.028


table(ndf[-train2,"default"], pred_pruned)
#  pred_pruned
#       No  Yes
# No  2896    7
# Yes   75   22


err2 <- (7+75)/(2896+7+75+22)
err2
# 0.02733333
```