BSAN 450 Assignment 17

1) In this problem we will review using a classification tree on the loan acceptance data that was used in the most recent video. This problem will cover the R commands to create a classification tree. This data is taken from Shmueli et at (2010). The data set contains information of 5000 loan applications. The response is whether or not an offered loan had been accepted on an earlier occasion. The explanatory variables are:

Age = age of the customer

Exp = professional experience in years

Inc = income of the customer

Fam = family size of the customer

CCAve = average monthly credit card spending

Educ = three categories of education level: 1 = undergraduate, 2 = graduate, 3 = professional

Mort = size of mortgage

SecAcc = 1 if the customer has a securities account and otherwise = 0

CD = 1 if the customer has a CD account and otherwise = 0

Online = 1 if the customer has an online account and otherwise = 0

CreditCard = 1 if the customer has a credit card and otherwise = 0

The name of the response variable is Response.

a) Read the data into R Studio and create the training and test sets. Note that these are the same training and test sets that were used earlier. The variables Response, Educ, SecAcc, CD and Online need to be changed into factor variables because they are all categorical variables that are codes using numeric values. If they are not recoded, they will be treated incorrectly.

loan=read.csv("LoanAccept.csv")

# change the type of the variables that are categorical.

loan$Response=factor(loan$Response)

loan$Educ=factor(loan$Educ)

loan$SecAcc=factor(loan$SecAcc)

loan$CD=factor(loan$CD)

loan$Online=factor(loan$Online)

set.seed(1)

train=sample(5000,4000)

test=(c(1:5000)[-train])

loantrain=loan[train,]

loantest=loan[test,]

The next commands construct the classification tree to classify the variable Response. The first expression is just like we used in multiple regression – it designates the dependent variable and the potential independent variables (in this case all the inputs). The second expression specifies the data frame. The third expression specifies that we want to construct a classification tree. Execute these commands.

library(rpart)

library(rpart.plot)

m1=rpart(Response~.,data=loantrain,method="class")

The next commands print out results of fitting the classification tree. The first command prints a table with a summary, the second plots the values in this table and the third plots the classification tree. Execute these commands.

printcp(m1)

plotcp(m1)

rpart.plot(m1,type=3,extra=1)

Consider the output from the command printcp(m1):

Classification tree:

rpart(formula = Response ~ ., data = loantrain, method = "class")

Variables actually used in tree construction:

[1] CCAve CD Educ Fam Inc

Root node error: 387/4000 = 0.09675

n= 4000

CP nsplit rel error xerror xstd

1 0.328165 0 1.00000 1.00000 0.048311

2 0.139535 2 0.34367 0.34884 0.029512

3 0.018088 3 0.20413 0.20930 0.023019

4 0.014212 5 0.16796 0.20672 0.022880

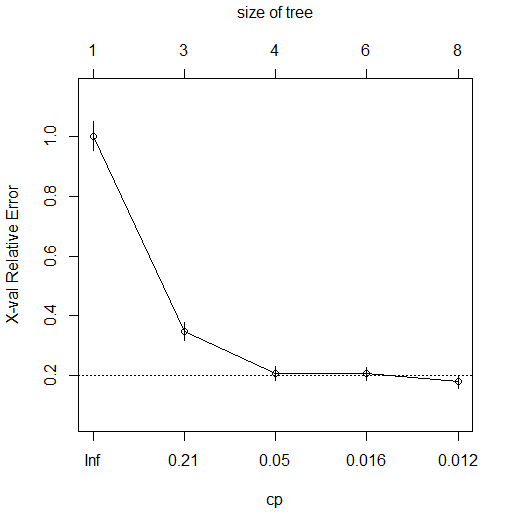
5 0.010000 7 0.13953 0.18088 0.021429

The column labeled xerror is the cross validation error for different subsets of the tree. This column gives the cross validation error for different subsets of the tree. These cross validation errors are scaled to that the first row = 1.000. What we are looking for is a low cross validation error. The column laveled xstd gives the standard errors or the cross validation errors. This column can be used to get an idea of how much the values in the xerror column could reasonably vary.

If we look down the xerror column we see that the values decrease until they level out with the third value onward. Thus, the tree corresponding to the third row is the tree that gives us nearly the lowest cross validation error.

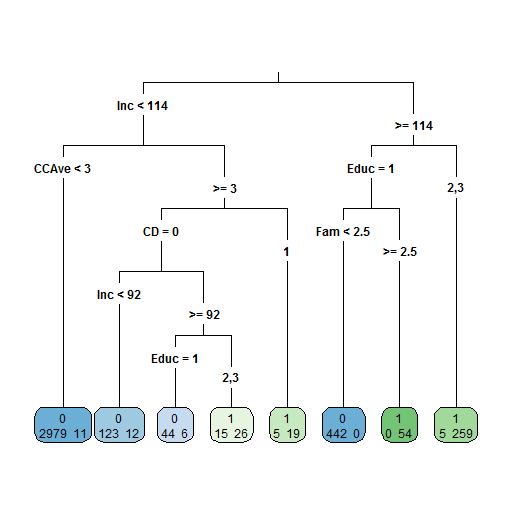
We want to prune the tree so that the pruned tree corresponds to the third row in this table. The value that is needed to prune the tree is listed in the column under CP, this value is 0.018988. This is the value that we will need in a subsequent R command.

The output from the command plotcp(m1) is below:



This is a plot of the values in the table. The values of the cross validation error are plotted versus the values of cp. The little vertical lines around the circles represent the amount of variation that could be reasonably expected. From the plot we see that the appropriate value of cp corresponds to the third row of the table.

The output from the command rpart.plot(m1,type=3,extra=1):

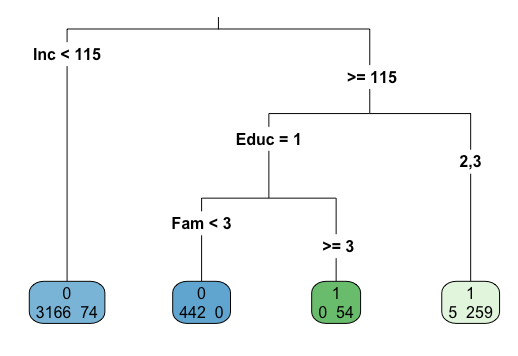


This is a plot of the classification tree. Starting at the top, the first split is based on whether Inc > 114 or Inc >= 114. The other splits are given in the tree. At the bottom of the tree the circles give the classification at the top and the number of customers in the training set that did not accept the loan to the left with the number of customers who accepted the loan to the right. For example, in the first circle at the far left, 2979 customers did not accept the loan and 11 customers did accept the loan. Thus customers in this branch would be classified as not accepting the loan (or 0). The customers in this branch had Inc < 114 and CCAve < 3.

a) The following R command uses the value of cp = .018088 to produce the pruned tree. The second command plots the pruned tree. Note that the pruned tree is labeled m2 and the unpruned tree is labeled m1 in these commands. Execute these commands. What is the difference between the pruned and the unpruned tree?

m2=prune(m1,cp=.018088)

rpart.plot(m2,type = 3,extra=1)



**The full tree gives more insight into the data that is in the branch inc <115; however, the impact that the other variables had was not that significant. Also since there are less variables there are far fewer branches.**

b) We can compute probabilities for each branch of the pruned tree. The probability for the branch is the number of customers who accepted the loan divided by the number of customers in the branch. Using the pruned tree, verify that the following probabilities are correct.

P(Accept loan | Inc < 114) = .0228

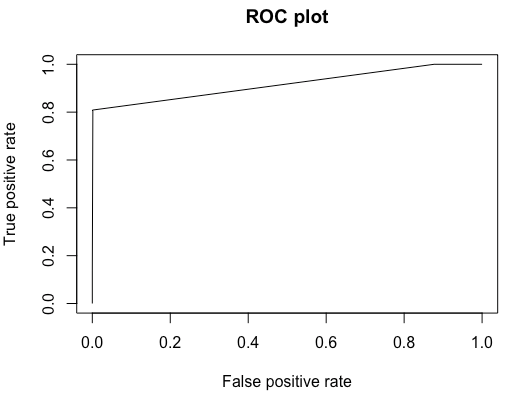
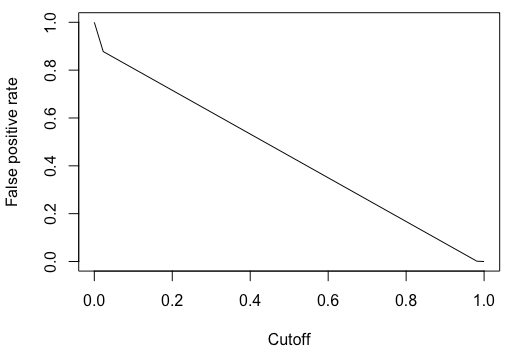
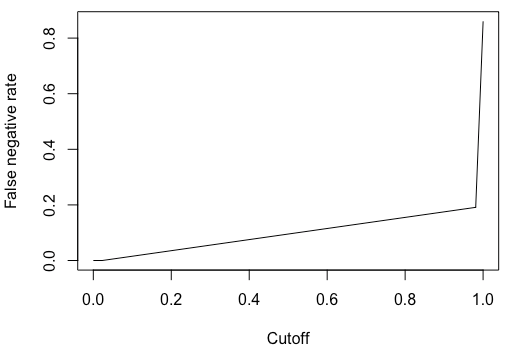
P(Accept loan | Inc > 114 and Educ = 1 and Fam < 2.5) = .0000

P(Accept loan | Inc > 114 and Educ = 1 and Fam >= 2.5) = 1.000

P(Accept loan | Inc > 114 and Educ = 2,3) = .9811

c) In a classification tree, the probabilities can be used to plot an ROC plot, a plot of the false positive error versus the cutoff, and a plot of the false negative error versus the cutoff. The R commands to do this follow. Note that these commands are almost the same as we have had before except for the command: pred=prediction(prob[,2],loantrain$Response). The expression prob[,2] is different because of the way the probabilities are stored in classification trees. Execute these commands to produce these plots.

Notice that the plots are a combination of three straight lines. This is because there are exactly 4 possible probabilities (those computed in part b) for the pruned tree.

2) This data consists of a sample of 200 subjects who were part of a much larger study on survival of patients following admission to an adult intensive care unit (ICU). The major goal of this study was to develop a logistic regression model to predict the probability of survival to hospital discharge of these patients and to study risk factors associated with ICU mortality. This data was taken from Hosmer, Lemeshow, and Sturdivant.

The variables in this data set are as follows.

STA: Vital status 1 if lived until discharge and 0 is died prior to discharge

AGE: Patient’s age in years.

GENDER: 1 if Female and 0 if Male

RACE: 1 if White, 2 if Black, and 3 if Other

SER: 1 if surgical service and 0 if medical service when admitted to ICU

CAN: 1 if cancer is part of the presenting problem otherwise 0

CRN: 1 if a history of chronic renal failure otherwise 0

INF: 1 if infection is probable at ICU admission otherwise 0

CPR: 1 if had CPR prior to ICU admission otherwise 0

SYS: systolic blood pressure at ICU admission

HRA: heart rate at ICU admission beats per minute

PRE: 1 if had been previously admitted to an ICU in the prior 6 months 0 otherwise

TYP: 1 if emergency admission 0 if elective admission

FRA: 1 if a long bone, multiple, neck, single area or hip fracture 0 otherwise

PO2: PO2 from initial blood gases 1 if less than or equal to 60 0 if greater than 60

PH: PH from initial blood gases1 if 1 if less than or equal to 7.25 0 if greater than 7.25

PCO: PCO2 from initial blood gases 1 if greater than or equal to 45 0 if less than 45

BIC: Bicarbonate from initial blood gases 1 if less than or equal to 18 0 if greater than 18

CRE: creatinine from initial blood gases 1 if greater than or equal to 2.0 and 0 is less than 2.0

LOC: level of consciousness at ICU admission “no” = no coma or stupor, “stu” = deep stupor, “com” = coma

a) Read the data into R Studio and recode the variable LOC as we did when we analyzed this data with logistic regression.

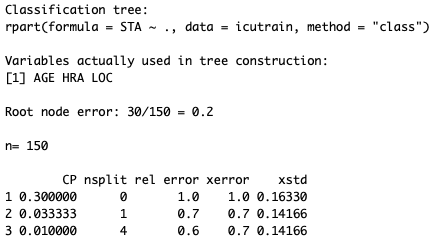
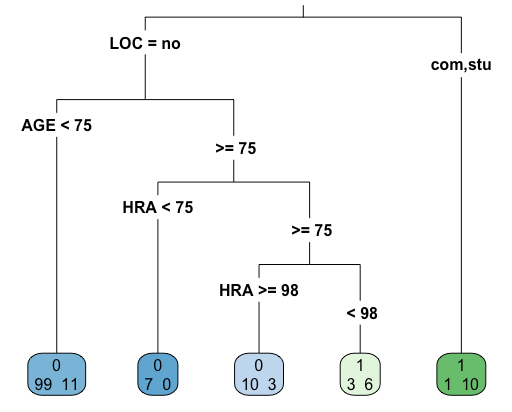
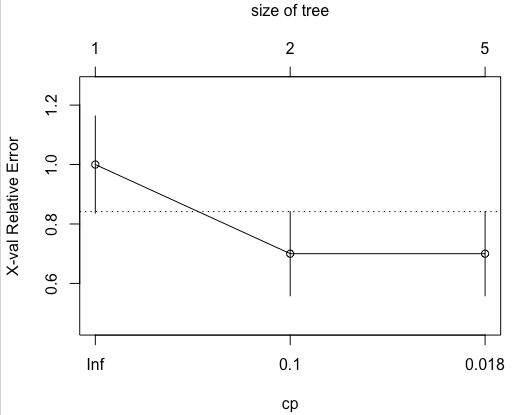
icu=read.csv("icu.csv")

library(car)

icu$LOC1=icu$LOC

icu$LOC1=recode(icu$LOC1, "'stu'='com'")

a) In R construct the full classification tree with STA as the response. Plot the full tree.

b) For the full tree print and plot the summary including the cross validation error. What value of cp should be used to prune the full tree?

**Cp = .033333 because after two entries in the table the xerror and xstd level off at 0.7 and 0.14166 respectively.**

c) Using the value of cp you determined in part b, prune the full tree. (Note, that when I did this I had to round the value of cp in the table slightly up to get the pruned tree). Print out the result.

