BSAN 450 Assignment 19

1) In this problem we will review using neural network classification on the loan acceptance data that was used in the most recent video. This problem will cover the R commands to create a neural network. 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 variable Response needs to be changed into factor variable.

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

set.seed(1)

train=sample(5000,4000)

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

loan$Response=factor(loan$Response)

Preparing the data:

The continuous variables need to be transformed so that they are all between 0 and 1. This is done via the max-min transformation. The following R commands define new transformed variables. The first part of this code creates a function that does the transformation. This function is then used to create the new variables: Age\_mn, Exp\_mn, Inc\_mn, Fam\_mn, CCAve\_mn, and Mort\_mn. Execute these commands.

minmax = function(x){

x=(x-min(x))/(max(x)-min(x))

return(x)

}

loan$Age\_mn=minmax(loan$Age)

loan$Exp\_mn=minmax(loan$Exp)

loan$Inc\_mn=minmax(loan$Inc)

loan$Fam\_mn=minmax(loan$Fam)

loan$CCAve\_mn=minmax(loan$CCAve)

loan$Mort\_mn=minmax(loan$Mort)

The categorical variable Educ needs to be converted into two indicator variable with values 0 and 1. The following R commands create the two indicator variables: Educug and Educgr. Execute there commands.

loan$Educug[loan$Educ==1]=1

loan$Educug[loan$Educ!=1]=0

loan$Educgr[loan$Educ==2]=1

loan$Educgr[loan$Educ!=2]=0

In order to make some of the next R commands simpler, the following R command creates a new data frame that only includes the variable we are predicting: Response, and the other variables that will be used as inputs. One way to do this is to exclude the following variables because they have been transformed: Age, Exp, Inc, Fam, CCAve, Mort, and Educ. The following command creates this new data frame.

loan.newdat=as.data.frame(loan[,-c(1,2,3,4,5,6,7)])

Next create the training set and the test sets by executing the following R commands.

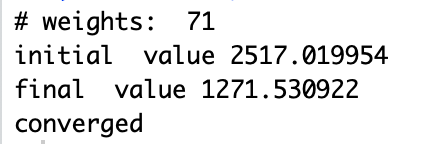
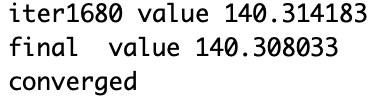
loantrain=loan.newdat[train,]

loantest=loan.newdat[test,]

The commands to fit the neural network for this data are the following: **Note that I had to execute the second command twice because on the first time it converged without much of a reduction in the value.**

library(nnet)

net.data=nnet(Response ~ .,data=loantrain,size=5,maxit=10000)

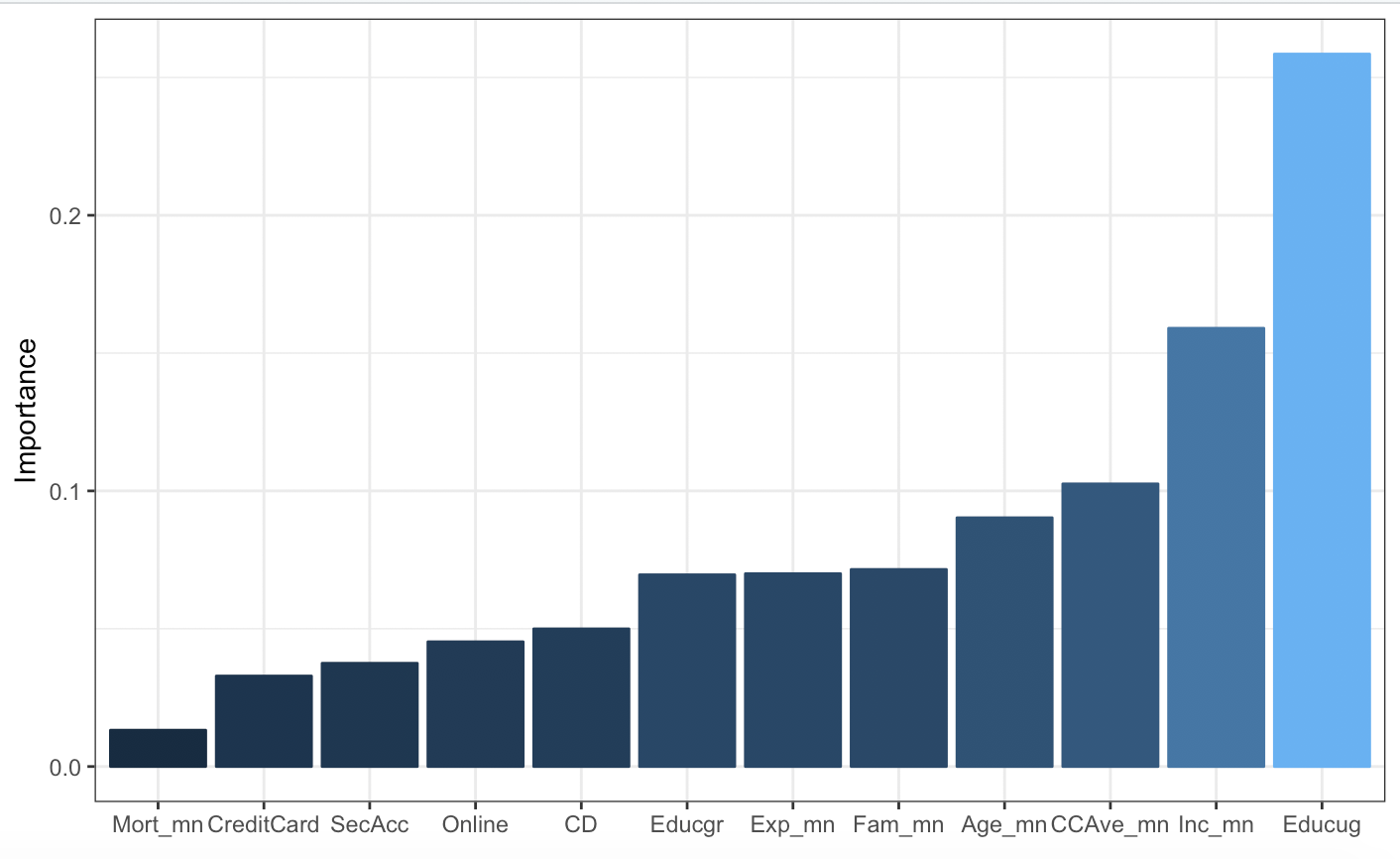
The following command will plot a graph of the importance of the input variables. The win.graph command opens a new window. Copy this graph.

library(NeuralNetTools)

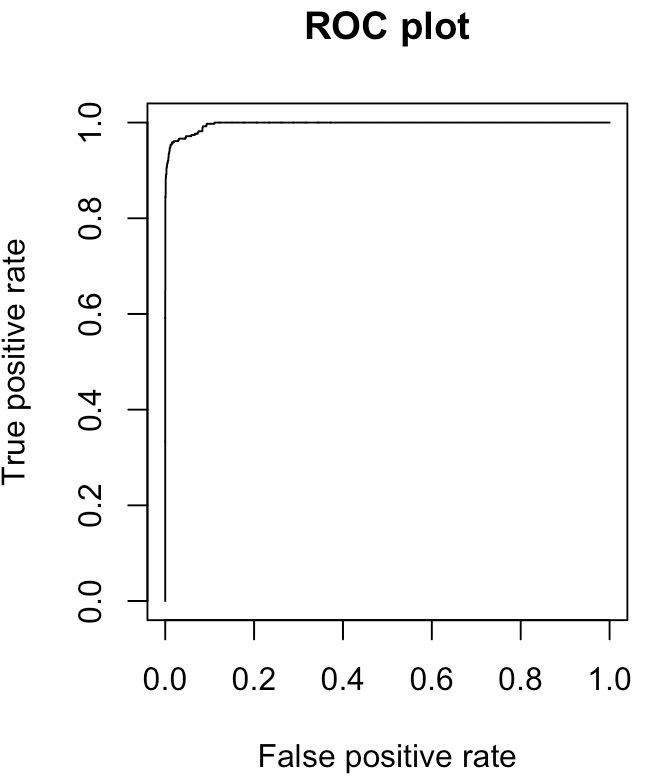
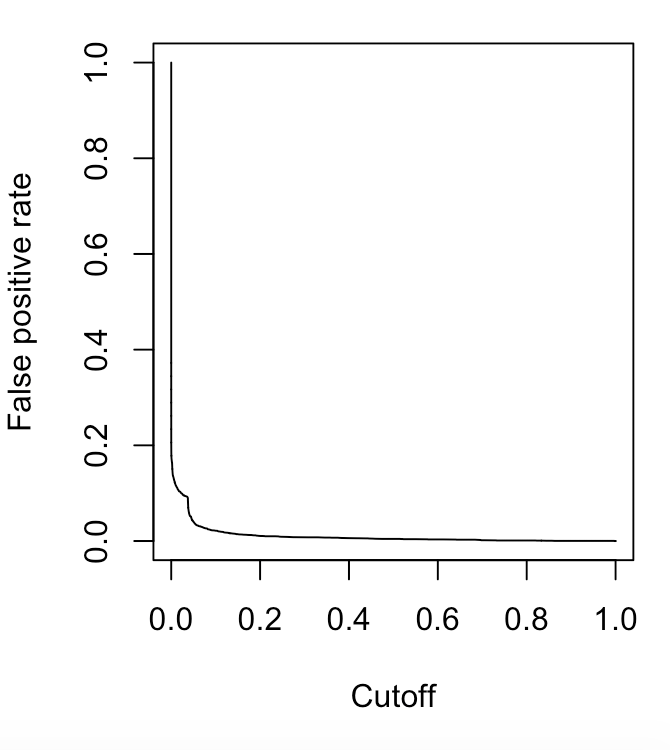
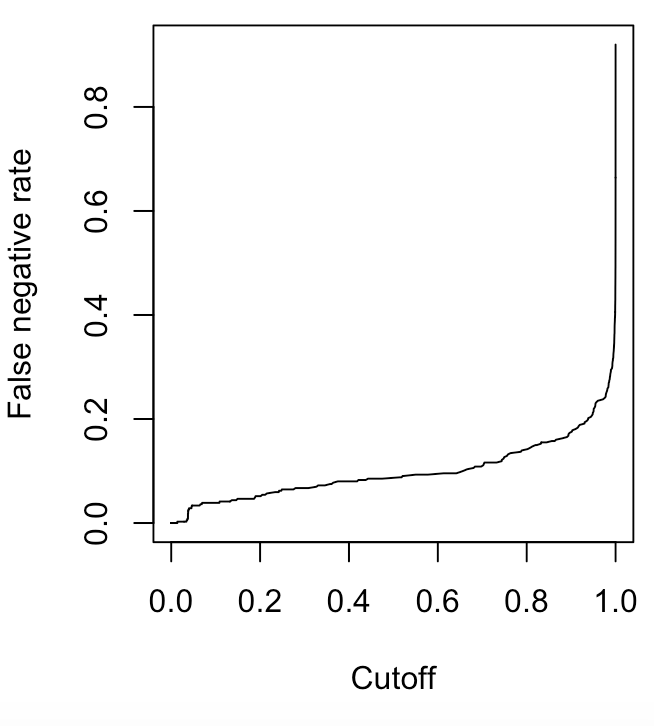
gar=garson(net.data)

win.graph(width=8.875, height = 3.5,pointsize=8)

plot(gar)



1b) The following R commands will plot an ROC plot, a plot of the false positive error rate versus the cutoff, and a plot of the false negative rate versus the cutoff. Execute these commands.

1c) The following R commands print the confusion matrix for the training data and for the test data with a cutoff of .5. Execute these commands and compute the overall error, the false positive rate, and the false negative rate.

prob = predict(net.data,loantrain)

pred=ifelse(prob >.5,1,0)

table(pred,loantrain$Response)

prob = predict(net.data,loantest)

pred=ifelse(prob>.5,1,0)

Overall error = (16+34)/(3597+16+34+353) =.0125

False pos = (16)/(3597+16) = .004428

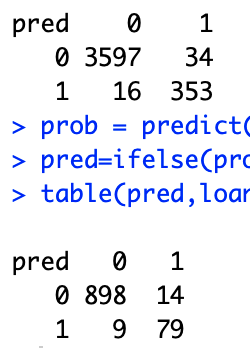
False neg = (34)/(34+353) = .0879

Overall error = (9+14)/(898+9+`4+79) = .0232

False pos = (9)/(898+9) = .0099

False neg = (14)/(`4+79) = .169

table(pred,loantest$Response)



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

Read the data into R Studio and create the indexes for the training set and the test set.

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

set.seed(1)

train=sample(200,150)

test=(c(1:200)[-train])

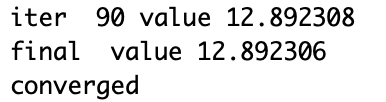
a) There are 3 continuous variables in the data set: AGE, SYS, and HRA. Use the max-min transformation to create 3 new variables whose values range from 0 to 1.

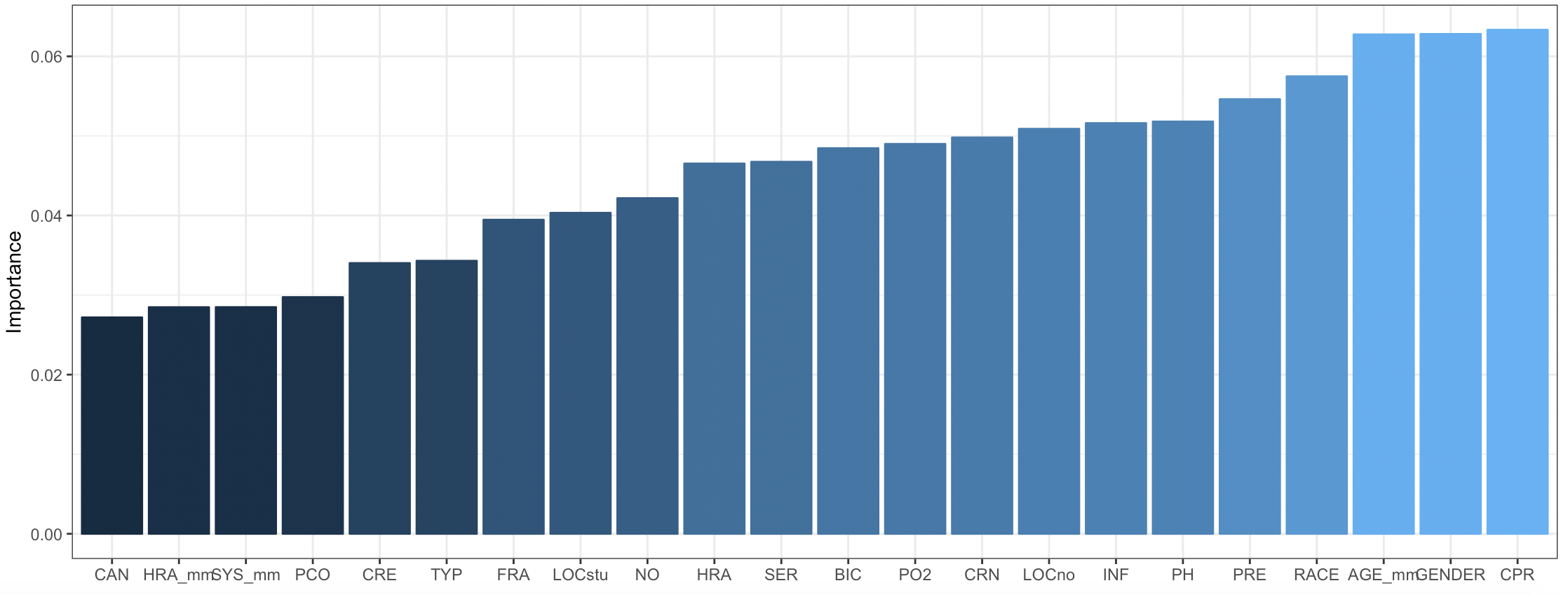
Previously we created 2 categories for the variable LOC, patients who when they were admitted were not in a coma or a stupor and patients when they were admitted were either in a coma or stupor. Thus we need to create a new indicator variable to replace the categorical variable LOC. The following R commands creates a variable called NO that is a variable that is 1 if the patient when admitted is not in a coma or stupor and otherwise equals 0.

icu$NO[icu$LOC!="no"]=1

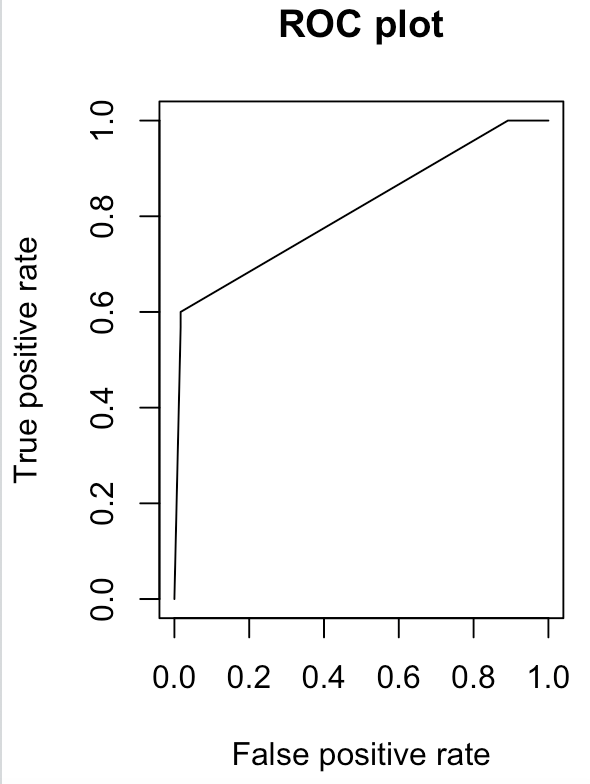
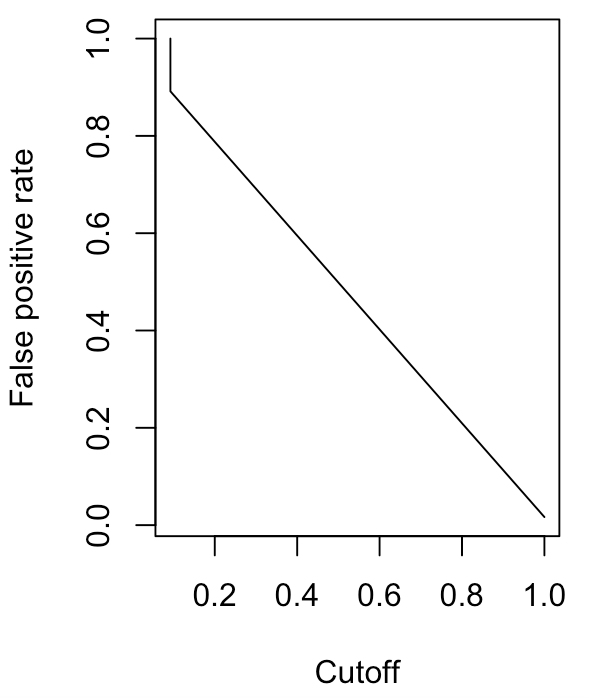
icu$NO[icu$LOC=="no"]=0

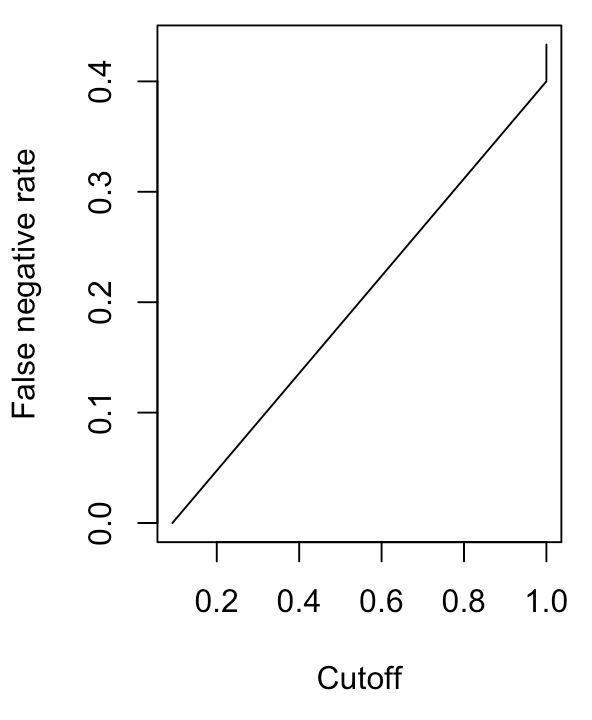
Estimate a neural network with 5 hidden layers for the training data to predict the variable STA with the inputs. Modify the R commands in problem 1. Produce a plot of the importance of the input variables.





b) Plot an ROC plot, a plot of the false positive error rate versus the cutoff, and a plot of the false negative rate versus the cutoff. Execute these commands.



c) Print the confusion matrix for the training data and for the test data with a cutoff of .5. Compute the overall error, the false positive rate, and the false negative rate.

Overall error = (2+12)/(118+2+12+18) =.09333

False pos = (2)/(2+118) = .0167

False neg = (12)/(12+18) = .4

Overall error = (1+6)/(39+1+6+4) = .14

False pos = (1)/(39+1) = .025

False neg = (6)/(6+4) = .6

