**Final Project, Part III Neural Networks**

1. Return to the data used for the midterm. I have renamed this data frame Data. To make things easier, prepare the data as follows.

a. Form a new data frame that contains only complete cases. Call this data frame Data2.

Table

Description automatically generated

b. Replace Gender with M = 1 and F = 0:

Data2$Gender <- ifelse(Data2$Gender=="M",1,0)

c. The Admissions director thought that whether the student lived east or west of the Connecticut River mattered in terms of retention and possibly in terms of grade point average.

Create a variable East where East = 1 if East.West.Out = “east”:

Data2$East <- ifelse(Data2$East.West.Out=="east", 1,0)

Do the same to create the variable Data2$West.

d. High schools were rated on a scale from 1 to 9, with 1 being the highest-ranking schools and 9 the lowest ranking schools. Create a new variable ERG\_low, which classifies schools ranked 5 – 9 as 1 (for low) and 0 (for high).

e. Remove the variable Retained.

f. Our target variable will be X1st\_yr\_GPA. Since some majors have a GPA requirement of 2.3, we will classify at risk students with X1st\_yr\_GPA = 1 if X1st\_yr\_GPA < 2.3. Otherwise, = 0.

g. Finally, replace all numeric variables with (x – min)/(max – min). They should now be scaled from 0 to 1. (In fact, all variables should have outcomes between 0 and 1.

2. You will need the package caret for this next step. Partition the data into training and testing.

set.seed(122)

inTrain <- createDataPartition(y=Data2$X1st\_yr\_GPA,p=.8,list=FALSE)

training <- Data2[inTrain,]

testing <- Data2[-inTrain,]

3. a. To begin, create a subset of the training data that contains only the variables Gender, VSAT, Housing, and HSGPA, and X1st\_yr\_GPA. Name this data frame training2. Do the same to create testing2.

b. For this next step, you will need the package neuralnet. Load this package.

c. Using the training2 data, make a neural network for the target variable. Let’s keep this simple and use only one hidden layer with 2 nodes. Insert your neural network. What was the error? How many steps did it take to converge? (If you don’t see the error on the printout of your network, you can find it nnA$result.matrix at the top.)



Chart, radar chart

Description automatically generated

The error was 54.852767. It took 24547 steps to converge.

d. For the first record in training2, calculate the output. Show your calculations using the activation function . If you can’t read the weights on your network, try nnA$weights. Chart

Description automatically generated with medium confidenceChart

Description automatically generated

e. Next, take your testing2 data frame and use it to test how well your network has done in predicting the “at risk” students. This code should work:



Text

Description automatically generated

Now, Compute the accuracy and error rate.

The accuracy of my neural network model is: 92.08791% (419/455 = 0.9208791 \* 100)

The error rate is 7.91209% (100%-92.08791%)

4. Now return to all the variables in the training data frame. Determine a neural network that seems to do a reasonable job in predicting the “at risk” students or at least the students who are “not at risk.” You can play around with the hidden layers (for example, you might try hidden = 4 or hidden = c(3,2)). Pick which variables you want in your network or use them all.

I decided to not include X\_1st\_year\_GPA as a variable because we used this variable to make the “at risk” variable.

1. Insert a copy of your final model (or best two models if you are having trouble picking).







1. What was the error for this model? Is it smaller than for the model from question 3?

I’m going to choose my 3rd model for my error rate. This model has hidden (4,2) while the other ones where either hidden(2), hidden(4,2) or hidden(2,1).

I noticed that when I made the first number bigger, the error rate would decrease but the steps would increase. When I made the second number bigger, the error rate increased, and the steps decreased.

This model is bigger than the one from question 3.

c. Show the confusion matrix and calculate the accuracy. Is the accuracy higher than in question 3?

Text

Description automatically generated

365/ 455 = 0.8021978

0.8021978\*100 = 80.21978% = Accuracy

100%-80.21978%= 19.78022% Error Rate

The accuracy is not higher than question 3.

5. Extra Credit. Replace the target variable X1st\_yr\_GPA with Retained. Partition the data into training and testing.

a. Create a neural network to predict whether students will be retained based on the training data.

b. Insert a plot of your network.

c. Test how well your network does based on the testing data. Insert a confusion matrix and calculate the accuracy.