**데이터마이닝이론**

**S T A 6 6 0 0**

**김현중 교수님**

**Homework 9**

2018314030

응용통계학과 김단아

(R or Python) Modify your program for Assignment #7 to do followings. For this assignment, use ‘titanic.csv’ file for categorical variables.

1. Prompt the user whether to run regression or classification.

2. If classification is chosen, prompt the user to choose (i) LDA and (ii) QDA, (iii) RDA, (iv) Logistic regression, (v) Naïve Bayes, or (vi) 1-level decision tree. However, if the data has more than two classes, do not prompt (iv), (v) and (vi).

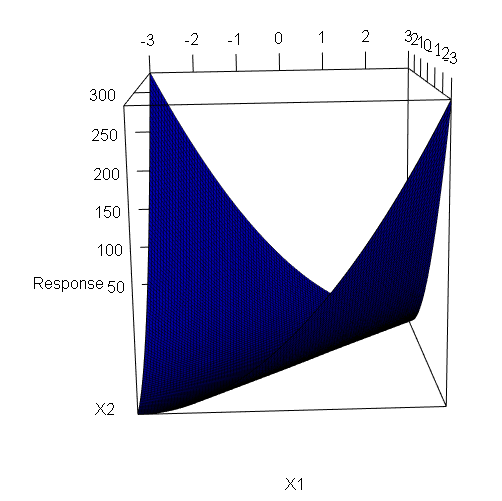
3. Make your program to implement (vi) 1-level decision tree only for two classes:

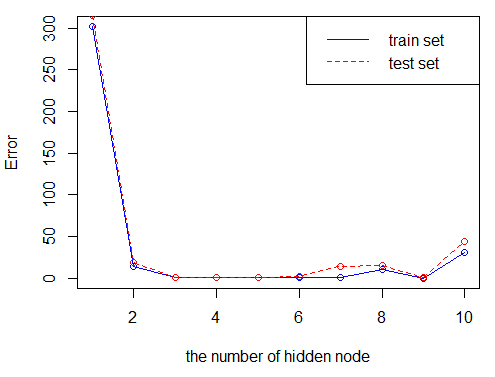
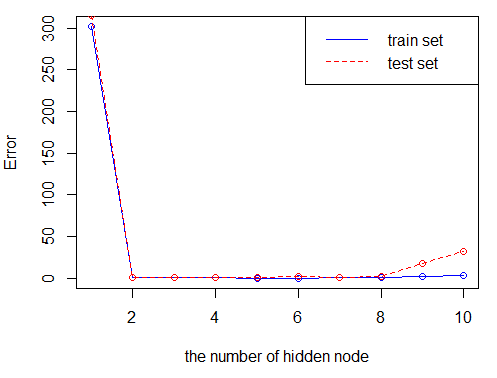
a. Find CART splitting rule, then split the current node into two subnodes. (Categorical variables should be considered in this assignment)

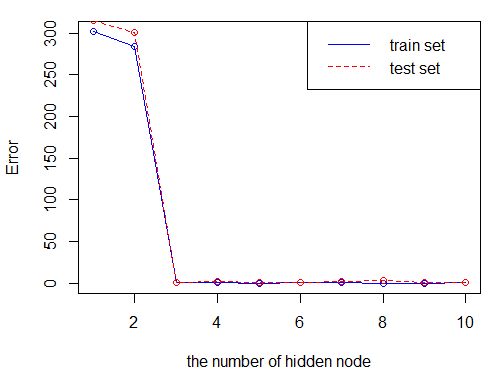
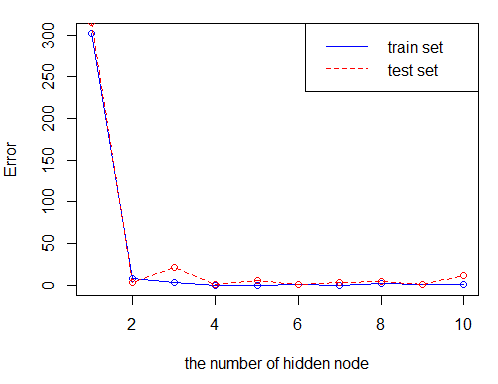
b. Print out the 1-level tree information and number of observations from each class.

4. Perform (i)-(vi) methods depending on the choice by the user.

|  |
| --- |
| **(R code)**  # Generate a training sample of size 1000  # train set 1000  set.seed(1234)  a1 = matrix(c(2, 2),nrow=2)  a2 = matrix(c(3, -3),nrow=2)  X1 = rnorm(1000, 0, 1)  X2 = rnorm(1000, 0, 1)  X = rbind(X1, X2)  Z = rnorm(1000, 0, 1)  sigmoid = function(x){1/(1+exp(-x))}  Y = c(sigmoid(t(a1)%\*%X) + (t(a2)%\*%X)^2 + 0.3\*Z)  train = data.frame(X1,X2,Y)  head(train)  # test set 1000  set.seed(5678)  a1 = matrix(c(2, 2),nrow=2)  a2 = matrix(c(3, -3),nrow=2)  X1 = rnorm(1000, 0, 1)  X2 = rnorm(1000, 0, 1)  X = rbind(X1, X2)  Z = rnorm(1000, 0, 1)  sigmoid = function(x){1/(1+exp(-x))}  Y = c(sigmoid(t(a1)%\*%X) + (t(a2)%\*%X)^2 + 0.3\*Z)  test = data.frame(X1,X2,Y)  head(test)  ## a. Plot the surface of responses  #install.packages('rgl')  require(rgl)  X1 = seq(from=-3, to=3, length.out = 100)  X2 = seq(from=-3, to=3, length.out = 100)  a1 = matrix(c(2, 2),nrow=2)  a2 = matrix(c(3, -3),nrow=2)  Y = function(x,y) {  X = rbind(x, y)  sigmoid(t(a1)%\*%X) + (t(a2)%\*%X)^2  }  Z <- outer(X1, X2, Y)  persp3d(X1, X2, Z, theta=50, phi=25, expand=0.75, col='blue',  ticktype="detailed", xlab="X1", ylab="X2", zlab="Response",axes=TRUE)  surface3d(X1, X2, Z, back = "lines")  surface3d(X1, X2, Z, front = "lines")  ## Neural network analysis using nnet  #install.packages('nnet')  library(nnet)  train\_error = c()  test\_error = c()  # hidden nodes from 1 to 10  for (i in 1:10) {  nn <- nnet(Y~., data=train, size=i, linout = TRUE)  print(summary(nn))    #### validation  # train  pred <- predict(nn, train[,-3])  train\_error[i] = sum((train$Y-pred)^2)/length(pred)    # test  pred\_t <- predict(nn, test[,-3])  test\_error[i] = sum((test$Y-pred\_t)^2)/length(pred\_t)  }  ## plot the error curves  plot(train\_error, type = 'o', col='blue', ylab='Error', xlab='the number of hidden node')  lines(test\_error, col='red', type='o', lty = 2)  legend('topright', c('train set', 'test set'), col = c('blue', 'red'), lty = c(1,2)) |







**(Python code)**

