BSAN 450 Assignment 16

1) We consider again the loan acceptance data that has been analyzed in earlier assignments. To review, 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 which equals 1 if the customer accepted the loan and 0 if the customer did not accept the loan.

We will use the k nearest neighbor classification method to classify the customers in the test set. As before we divide the data into a training set of 4,000 customers and a test set of 1,000 customers. The following R commands read the data and create the row numbers for the training and test sets as in the previous assignments.

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

set.seed(1)

train=sample(5000,4000)

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

The following R commands set up the data set. The first command changes the variable Response to a factor variable. The second set of commands scales the continuous variables (Age, Exp, Inc, Fam, CCAve, and Mort) so that their mean is zero and their standard deviation is one. The third set of commands creates a new indicator type variable in place of the categorical variable Educ. Note that the categorical variables SecAcc, CD, Online, and CreditCard are already set up as indicator variables with 1 or 0 values. This is what is required to get the correct values when using the Euclidean distance measure.

# change the variable Response to a factor, the Y must be a factor in the knn function.

loan$Response=factor(loan$Response)

# scale the variables that are on a continuous scale

num.vars = c(1,2,3,4,5,7)

loan[num.vars] = lapply(loan[num.vars], scale)

# create 3 variables for the categorical variable Educ.

loan$e1 =0

loan$e2 =0

loan$e3 = 0

loan$e1[loan$Educ ==1] = 1/sqrt(2)

loan$e2[loan$Educ==2] = 1/sqrt(2)

loan$e3[loan$Educ==3] = 1/sqrt(2)

The next commands select the input variables that will be used in the nearest neighbor algorithm. We need to exclude the variable Response (which is the 8th variable in the dataframe) and the variable Educ (which is the 6th variable in the dataframe). These commands also create the training and test sets. Note that we need to have the input variables in one place and the Y variable in a separate place.

# select the variables that will be used as the inputs

x.vars= c(1,2,3,4,5,7,9,10,11,12,13,14,15)

x=loan[,x.vars]

# create the training and test sets

xtrain =x[train,]

xtest = x[test,]

ytrain = loan$Response[train]

ytest = loan$Response[test]

The next set of commands computes the overall error, false positive error and false negative error for different values of K.

library(class)

# use leave one out cross validation to determine the value of K. o.error is the overall error, fp.rate is

# the false positive error and fn.rate is the false negative error. Try k = 1 to 10

set.seed(1)

o.error=rep(NA,10)

fp.rate=rep(NA,10)

fn.rate=rep(NA,10)

for(i in 1:10){

knn.pred=knn.cv(xtrain,ytrain,k=i)

ttt =table(knn.pred,ytrain)

o.error[i]=(ttt[1,2]+ttt[2,1])/(ttt[1,1]+ttt[1,2]+ttt[2,1]+ttt[2,2])

fp.rate[i]= (ttt[2,1])/(ttt[2,1]+ttt[1,1])

fn.rate[i]=(ttt[1,2])/(ttt[1,2]+ttt[2,2])

}

o.error

fp.rate

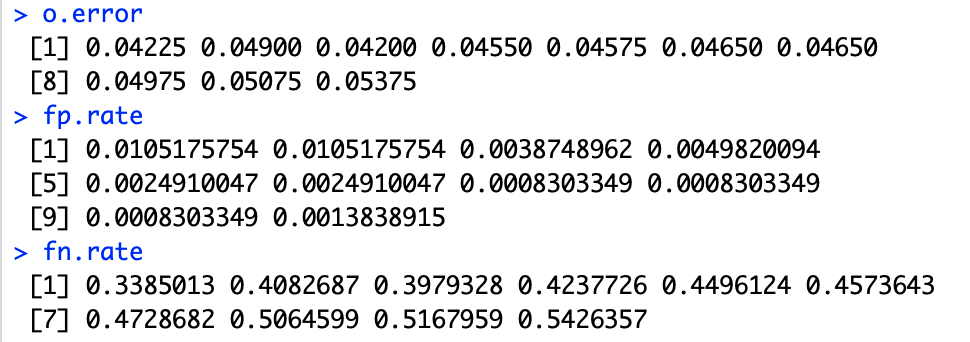
fn.rate

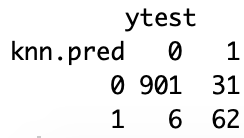
# compute the confusion matrix for the test data

knn.pred=knn(xtrain,xtest,ytrain,k=3)

table(knn.pred,ytest)

Execute these commands to reproduce the results in the video.





2) The data for this example came from the Global Longitudinal Study of Osteoporosis in Women. This data was taken from the book Applied Logistic Regression: Third Edition by Hosmer, Lemeshow, and Surdivant. The variable of interest is whether or not the woman had a fracture in the first year after a medical exam. The data set has 500 women, 125 had a fracture during the first year of follow up and 375 did not have a fracture during the first year. Five independent variables were thought to be important are:

PRIORFRAC = 1 if the woman has a history of prior fractures, and 0 if she does not have a history of prior fractures

AGE = the age of the woman at enrollment,

WEIGHT = the weight of the woman at enrollment,

HEIGHT = the height of the woman at enrollment

PREMENO = 1 if the woman experienced menopause before age 45 and otherwise 0

MOMFRAC = 1 if the woman’s mother had a fracture and otherwise 0

ARMASSIST = 1 if the woman need assistance in walking and otherwise 0

SMOKE = 1 if the woman is a smoker and otherwise 0

RATERISK a self reported risk of fracture: 1 = less than others of the same age, 2 = same as others of the same age, and 3 = greater than others of the same age.

FRACTURE = 1 is the woman had a fracture in the first year and otherwise 0.

FRACTURE is the dependent variable.

Read the data into R Studio and create a training set of 400 women and a test set of 100 women.

glow=read.csv("glow500.csv")

set.seed(1)

train=sample(500,400)

test=(c(1:500)[-train])

a) We will use the K nearest neighbor classification for this data. The first thing we need to do is to prepare the data.

i) Change the variable FRACTURE into a factor variable.

ii) The continuous variables are: AGE, WEIGHT and HEIGHT. Rescale there variables by subtracting their mean and dividing by their standard deviation.

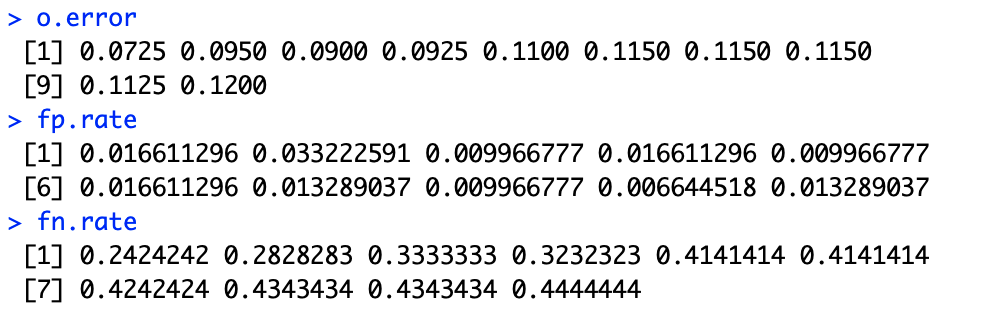
iiii) The following variables are categorical variables with 2 categories: PRIORFRAC, PREMENO, MOMFRAC, ARMASSIST, and SMOKE. Do you need to create any new variables associated with these categorical variables? Why or why not?

**No new variables are not needed since for the current variable can either be 1 or 0. It is where there is more than two that you need new variables.**

iv) The variable RATERISK is a categorical variable with 3 categories, create a set of three new indicator type variables that can be used instead or the variable RATERISK when computing the Euclidean distance calculations.

By following the example in problem 1, select the variables that will be used as inputs in the nearest neighbor classification and set up the training set inputs, the training set classification of the variable FRACTURE, and the test set inputs.

Modify the R commands that were used in problem 1to calculate the overall error, the false positive error and the false negative error for this problem. Compute these errors based on leave one out cross validation on the training data. Based on the output select what you believe to be an appropriate value of K.



**K=3**

b) For the value of K you selected in part a, calculate the confusion matrix for the test data and calculate the overall error, the false positive error and the false negative error for the test data.

