Alexandrea Stylianou
STATS 667
Homework 4

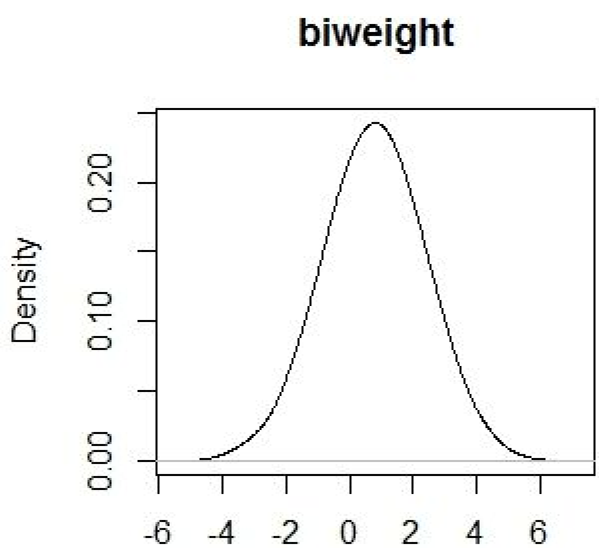
```
# Read in simulated.csv
###########################################
#Question 1a
setwd("D:/Fall 2016/STAT 667")
sim_data <- read.csv("simulated.csv")

X <- sim_data$x1
y<-sim_data$y
X2<-sim_data$x2
#return( result1 )
Xi<- X[38]

bikern<-function(X){ #bikern<-function(u)
  result1 <- 0
  ind1 <- ifelse( X <=1 ,1, 0)
  Xi<- X[38]
  K <- ( ( 15/16 )*( 1 - X^2 )^2 )*ind1
  return(K)
  n<-length(X)
  h<-1
  for(Xi in X)
  {
    Ki <- K((x-Xi)/h)/(n*h)
  }
  return (Ki)

}

plot(X,bikern( abs(X) ),
     main="Biweight",
     type="l")
```




```
##########################################################################
#Question 1 b

#[b] Restrict your attention to the x1 feature vector and the class y. Use kernel discriminant analysis
#(Gaussian kernel, default bandwidth selector) to predict the class labels. Report the number of
#misclassification errors in each class.
#kda
meh<-kda(X,y)

g.of.x1<-log(meh$x.group/meh$x.group.estimate) + log(.5/.5)
pred.class<-ifelse(g.of.x1>0.5, "0", "1")
table(pred.class)
#pred.class<---------------predicting class labels
#0   1
#28 111


yhat <- ifelse((pred.class)>0.5, 1, 0)
tab<-table(meh$x.group,meh$x.group.estimate)
tab[row(tab)!=col(tab)]
#[1] 28 39<-------missclassification errors in each class
```

```
#[c] Now restrict your attention to the x2 feature vector and the class y. Use kernel discriminant
analysis
#(Gaussian kernel, default bandwidth selector) to predict the class labels. Report the number of
#misclassification errors in each class.
meh_2<-kda(X2,y)


g.of.x2<-log(meh_2$x.group/meh_2$x.group.estimate) + log(.5/.5)
pred.class2<-ifelse(g.of.x2>0.5, "0", "1")
table(pred.class2)
#pred.class2<------------------------------predicting class labels
#0   1
#17 119

yhat2 <- ifelse((pred.class2)>0.5, 1, 0)
tab2<-table(meh_2$x.group,meh_2$x.group.estimate)
tab2[row(tab2)!=col(tab2)]
#[1] 17 36<-------missclassification errors in each class



#[d] Construct a naïve Bayes classifier considering both features x1 and x2. Report the number
of
#misclassification errors in each class.

b<-naiveBayes(y ~ X+X2, data = sim_data)
b_class<-predict(b,sim_data,type="raw")

yhat3 <- ifelse((X)>0.5, 1, 0)
tab<-table(yhat3, y)
tab[row(tab)!=col(tab)]
#[1] 65 50<----------missclassification errors in each class


######################################################################

#Problem 2: [10 points]
#For the heartdisease.csv data,
#[a] Construct a Naïve Bayes classifier (use a Gaussian kernel, default bandwidth selector) to
predict chd
#using the variables age, adiposity, and alcohol. Report the overall misclassification rate.
```

```
heart_data<-read.csv("heartdisease.csv")
age<-heart_data$age
adiposity<-heart_data$adiposity
alcohol<-heart_data$alcohol
chd<-heart_data$chd

kernel.age<-sm.density(age,eval.points=c(alcohol,age,adiposity,chd), display="none" )
kernel.ad<-sm.density(adiposity,eval.points=c(alcohol,age,adiposity,chd), display="none" )
kernel.alc<-sm.density(alcohol,eval.points=c(alcohol,age,adiposity,chd), display="none" )
kernel.chd<-sm.density(chd,eval.points=c(alcohol,age,adiposity,chd), display="none")

bayes<-naiveBayes(kernel.chd$estimate ~ kernel.age$estimate + kernel.ad$estimate +
kernel.alc$estimate , data = heart_data)


heart_class<-predict(bayes,heart_data,type="raw")

yhat2 <- ifelse((heart_class)>0.5, 1)
tab<-table(yhat2, chd)

length(yhat2)
length(chd)
1 - sum(yhat2==chd)/length(chd)


#[b] Compare the misclassification error using Naïve Bayes classifier with the error rate
associated with
#a logistic regression model predicting chd using age, adiposity, and alcohol.
c<-naiveBayes(chd ~ alcohol + adiposity + age, data=heart_data)
c_phat<-predict(c,heart_data,type="raw")

bayes_mis<-ifelse(c_phat>0.5, 1, 0)
length(bayes_mis) #[1] 924

tab<-table(bayes_mis, chd)
tab[row(tab)!=col(tab)]
#54 95
1 - sum(bayes_mis==chd)/length(bayes_mis)
#0.5

##################################################################################
#########
```

```
logreg1<-glm(chd ~ alcohol + adiposity + age, data=heart_data)
logreg1.phat<-predict(logreg1, type="response")

### Prediction algorithm
logreg.class<-ifelse(logreg1.phat>0.5, 1, 0)
length(logreg.class) #[1] 462

tab<-table(logreg.class, chd)
tab[row(tab)!=col(tab)]
#[1] 54 95
1 - sum(logreg.class==chd)/length(logreg.class)
#0.3225108


#Question3
#Again, for the heartdisease.csv data:
#  [a] using age, adiposity, and alcohol, list the 5 nearest neighbors to observation 84
heart_data<-read.csv("heartdisease.csv")
x<-heart_data[,-1]

#Report the distance for each of these 5 nearest neighbors.

obs84 = sqrt((x[,1]-x[84,1])^2+(x[,2]-x[84,2])^2+(x[,3]-x[84,3])^2)
which(match(obs84,sort(obs84)[2:6])>0)
obs84[which(match(obs84,sort(obs84)[2:6])>0)]
#4.551099 4.007855 3.042384 3.225291 2.876821


#[b] Using the five nearest neighbors identified in part (a), what is the predicted class for this observation
#using the knn rule?
y[which(match(obs84,sort(obs84)[2:6])>0)]
#-11.28108 -11.45652 -13.84040 -10.44620 -13.75662

#[c] Use k-nearest neighbors with k = 5 to classify all observations in the heartdisease dataset. Report
#the number of misclassification errors in each class.

library(class)
yhat_3 = y
for(i in 1:length(y)){
  yhat_3[i]=knn(x[-i,],x[i,],y[-i],5)
}
```

```
table(yhat_3,y)
#tab[row(tab)!=col(tab)]
#zero misclassifications


#For the dataset in hmwk4.csv, the class to be predicted is stored in the first column (type), and
the explanatory
#variables are all binary variables and are stored in the remaining columns (2-222). Each row
represents a
#different subject. For this dataset, do the following:
#  [a] Read the data into the R programming environment. Assume that the frequencies of
matches….

class_data_set <- read.csv("hmwk4.csv")
y_1=class_data_set[,1]
x_1=class_data_set[,-1]
h = matrix(,length(y_1),length(y_1))
for(i in 1:length(y_1)) {
  for(j in 1:length(y_1)) {
    a=0
    b=0
    c=0
    d=0
    for(k in 1:length(x_1[1,])) {
          if {
        if(x_1[i,k]==1) {
          b=b+1
        } else {
          c=c+1
        }
    h[i,j] = 1 - (a+d)/(a+d+2*(b+c))
  }
}


#[b] For each subject, identify its ~5 nearest neighbors by using observations with a rank ≤5.5,
and obtain the
#predicted class. List the row numbers that are nearest neighbors for subject 30 and 65.
library(nnet)
pred_class=unique(y_1)
yhat_5=rep(0,length(y_1))
```

```
for(i in 1:length(y_1)) {
  temp=h[i,-i]
  vector1=vector('numeric')
  temp_val=sort(temp)[1:5]
  unique_val=unique(temp_val)

  if(i==30 || i==65) {
    print(vector1)
  }
  freq = rep(0,length(pred_class))

}
yhat_5=factor(yhat_5,labels=levels(y_1))

#[c] Produce a cross-tabulation of the predicted class versus the true class (type). Report the misclassification
#error rate.
table(y,yhat_5)
```