

Stat 648: Assignment 1 Solutions

Problem 1:

(2.1) (5 points)

$$\begin{aligned} \min_k \|\mathbf{t}_k - \hat{\mathbf{y}}\| &= \min_k \sqrt{\sum_{j=1}^K (t_{kj} - \hat{y}_j)^2} = \min_k \left(\sum_{j=1}^K (t_{kj} - \hat{y}_j)^2 \right) = \min_k \left(\sum_{j=1, j \neq k}^K \hat{y}_j^2 + (1 - \hat{y}_k)^2 \right) \\ &= \min_k \left(\sum_{j=1}^K \hat{y}_j^2 + 1 - 2\hat{y}_k \right) = \min_k (-2\hat{y}_k) = \max_k \hat{y}_k \end{aligned}$$

Thus, classifying to the largest element of $\hat{\mathbf{y}}$ amounts to choosing the closest target.

(2.2) (5 points) We have two classifications, Blue and Orange, so that the Bayes decision boundary consists of all \mathbf{x} that satisfy $\Pr(\text{Blue}|\mathbf{X} = \mathbf{x}) = \Pr(\text{Orange}|\mathbf{X} = \mathbf{x})$.

Now, $\Pr(\text{Blue}|\mathbf{X} = \mathbf{x}) = \frac{f(\mathbf{x}|\text{Blue})\Pr(\text{Blue})}{f(\mathbf{x})}$ with a similar result for Orange. Since $\Pr(\text{Orange}) = \Pr(\text{Blue})$, we arrive at a Bayes decision boundary that consists of all \mathbf{x} satisfying $f(\mathbf{x}|\text{Blue}) = f(\mathbf{x}|\text{Orange})$. With $\mathbf{m}_{k, \text{Orange}}$ and $\mathbf{m}_{k, \text{Blue}}$, $k = 1, \dots, 10$, generated from $N_2((1, 0)^T, \mathbf{I})$ (and assumed known), the decision boundary is all \mathbf{x} such that $\frac{1}{10} \sum_{k=1}^{10} g(\mathbf{x}|\mathbf{m}_{k, \text{Orange}}, \mathbf{I}/5) = \frac{1}{10} \sum_{k=1}^{10} g(\mathbf{x}|\mathbf{m}_{k, \text{Blue}}, \mathbf{I}/5)$, where g represents the bivariate normal density.

(2.7) (15 points)

(a) $f(\mathbf{x}_0) = \mathbf{x}_0^T \hat{\beta} = \mathbf{x}_0^T (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{y} = \sum_{i=1}^N (\mathbf{x}_0^T (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}')_i y_i$

So, for linear regression, ℓ_i is the i^{th} element of $\mathbf{x}_0^T (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'$.

For k-nearest neighbors, by equation (2.8), $f(\mathbf{x}_0) = \frac{1}{k} \sum_{i|\mathbf{x}_i \in N_k(\mathbf{x}_0)} y_i$. Thus, $\ell_i = \frac{1}{k} I_{\mathbf{x}_i \in N_k(\mathbf{x}_0)}$.

(b) Let \hat{f} , f and f_i represent $\hat{f}(\mathbf{x}_0)$, $f(\mathbf{x}_0)$, and $f(\mathbf{x}_i)$, respectively.

$$\begin{aligned} E_{\mathcal{Y}|\mathcal{X}} (f - \hat{f})^2 &= E_{\mathcal{Y}|\mathcal{X}} (f - E_{\mathcal{Y}|\mathcal{X}} \hat{f} + E_{\mathcal{Y}|\mathcal{X}} \hat{f} - \hat{f})^2 \\ &= E_{\mathcal{Y}|\mathcal{X}} (f - E_{\mathcal{Y}|\mathcal{X}} \hat{f})^2 + E_{\mathcal{Y}|\mathcal{X}} (E_{\mathcal{Y}|\mathcal{X}} \hat{f} - \hat{f})^2 \\ &\quad + 2E_{\mathcal{Y}|\mathcal{X}} ((f - E_{\mathcal{Y}|\mathcal{X}} \hat{f}) (E_{\mathcal{Y}|\mathcal{X}} \hat{f} - \hat{f})) \\ &= (f - E_{\mathcal{Y}|\mathcal{X}} \hat{f})^2 + E_{\mathcal{Y}|\mathcal{X}} (E_{\mathcal{Y}|\mathcal{X}} \hat{f} - \hat{f})^2 + 0 \\ &= (\text{Bias}_{\mathcal{Y}|\mathcal{X}} \hat{f})^2 + \text{Var}_{\mathcal{Y}|\mathcal{X}} \hat{f} \\ &= \left(f - \sum_{i=1}^N \ell_i(\mathbf{x}_0; \mathcal{X}) f_i \right)^2 + \sum_{i=1}^N \ell_i^2(\mathbf{x}_0; \mathcal{X}) \sigma^2 \end{aligned}$$

$$\begin{aligned}
(c) \quad \mathbb{E}_{\mathcal{Y}, \mathcal{X}} (f - \hat{f})^2 &= \mathbb{E}_{\mathcal{X}} \mathbb{E}_{\mathcal{Y}|\mathcal{X}} (f - \hat{f})^2 \\
&= \mathbb{E}_{\mathcal{X}} \left((\text{Bias}_{\mathcal{Y}|\mathcal{X}} \hat{f})^2 + \text{Var}_{\mathcal{Y}|\mathcal{X}} \hat{f} \right) \\
&= \mathbb{E}_{\mathcal{X}} \left(f^2 - 2f\mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f} + (\mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f})^2 \right) - \text{Var}_{\mathcal{X}} \mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f} + \text{Var}_{\mathcal{Y}, \mathcal{X}} \hat{f} \\
&= f^2 - 2f\mathbb{E}_{\mathcal{Y}, \mathcal{X}} \hat{f} + \mathbb{E}_{\mathcal{X}} \left((\mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f})^2 \right) - \text{Var}_{\mathcal{X}} \mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f} + \text{Var}_{\mathcal{Y}, \mathcal{X}} \hat{f} \\
&= f^2 - 2f\mathbb{E}_{\mathcal{Y}, \mathcal{X}} \hat{f} + (\mathbb{E}_{\mathcal{X}} \mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f})^2 + \text{Var}_{\mathcal{Y}, \mathcal{X}} \hat{f} \\
&= f^2 - 2f\mathbb{E}_{\mathcal{Y}, \mathcal{X}} \hat{f} + (\mathbb{E}_{\mathcal{Y}, \mathcal{X}} \hat{f})^2 + \text{Var}_{\mathcal{Y}, \mathcal{X}} \hat{f} \\
&= (f - \mathbb{E}_{\mathcal{Y}, \mathcal{X}} \hat{f})^2 + \text{Var}_{\mathcal{Y}, \mathcal{X}} \hat{f} \\
&= (\text{Bias}_{\mathcal{Y}, \mathcal{X}} \hat{f})^2 + \text{Var}_{\mathcal{Y}, \mathcal{X}} \hat{f}
\end{aligned}$$

(d)

$$\begin{aligned}
\mathbb{E}_{\mathcal{X}} \left((\text{Bias}_{\mathcal{Y}|\mathcal{X}} \hat{f})^2 \right) - (\text{Bias}_{\mathcal{Y}, \mathcal{X}} \hat{f})^2 &= \mathbb{E}_{\mathcal{X}} \left((f - \mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f})^2 \right) - (f - \mathbb{E}_{\mathcal{Y}, \mathcal{X}} \hat{f})^2 \\
&= \mathbb{E}_{\mathcal{X}} (\mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f})^2 - (\mathbb{E}_{\mathcal{Y}, \mathcal{X}} \hat{f})^2 \\
&= \mathbb{E}_{\mathcal{X}} (\mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f})^2 - (\mathbb{E}_{\mathcal{X}} \mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f})^2 \\
&= \text{Var}_{\mathcal{X}} \mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f} \geq 0
\end{aligned}$$

So, the expected value of the squared conditional bias from (b) is larger than the squared bias from (c) by the amount $\text{Var}_{\mathcal{X}} \mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f}$.

$$\begin{aligned}
\mathbb{E}_{\mathcal{X}} (\text{Var}_{\mathcal{Y}|\mathcal{X}} \hat{f}) - \text{Var}_{\mathcal{Y}, \mathcal{X}} \hat{f} &= -\mathbb{E}_{\mathcal{X}} (\mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f})^2 + (\mathbb{E}_{\mathcal{Y}, \mathcal{X}} \hat{f})^2 \\
&= -\text{Var}_{\mathcal{X}} \mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f} \leq 0
\end{aligned}$$

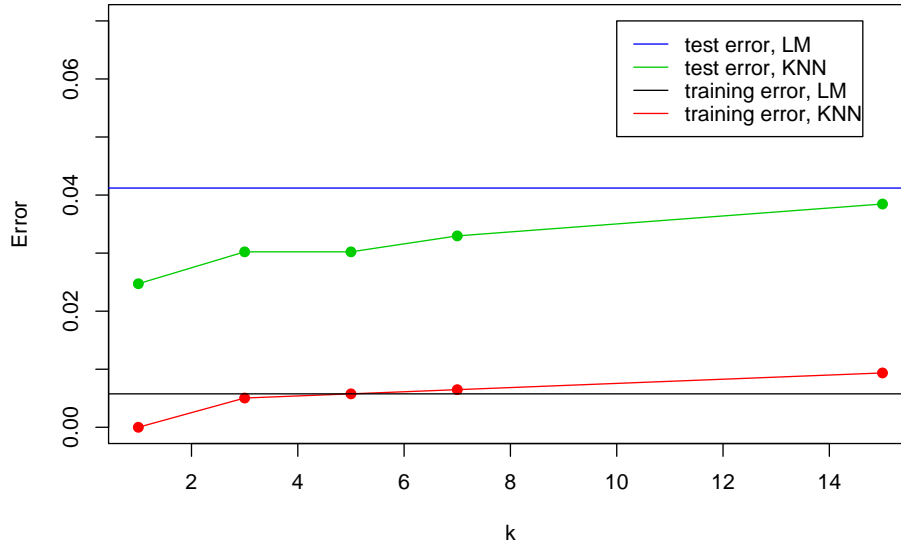
So, the expected value of the conditional variance (from (b)) is smaller than the variance from (c) by the amount $\text{Var}_{\mathcal{X}} \mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f}$.

This, by the way, produces

$$\begin{aligned}
\mathbb{E}_{\mathcal{X}} \text{MSE}_{\mathcal{Y}|\mathcal{X}} \hat{f} &= \mathbb{E}_{\mathcal{X}} \left((f - \mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f})^2 + \text{Var}_{\mathcal{Y}|\mathcal{X}} \hat{f} \right) \\
&= (f - \mathbb{E}_{\mathcal{Y}, \mathcal{X}} \hat{f})^2 + \text{Var}_{\mathcal{X}} \mathbb{E}_{\mathcal{Y}|\mathcal{X}} \hat{f} + \mathbb{E}_{\mathcal{X}} \text{Var}_{\mathcal{Y}|\mathcal{X}} \hat{f} \\
&= (f - \mathbb{E}_{\mathcal{Y}, \mathcal{X}} \hat{f})^2 + \text{Var}_{\mathcal{Y}, \mathcal{X}} \hat{f} \\
&= \text{MSE}_{\mathcal{Y}, \mathcal{X}} \hat{f}
\end{aligned}$$

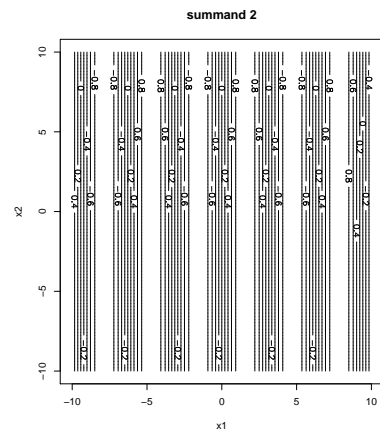
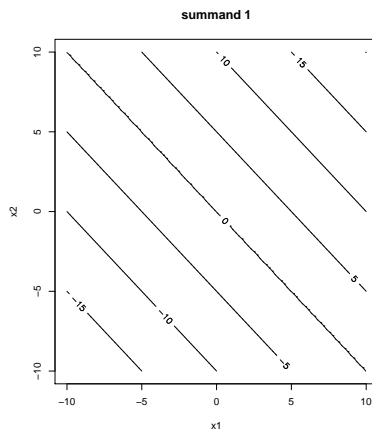
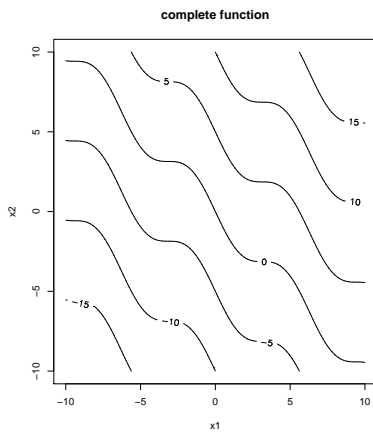
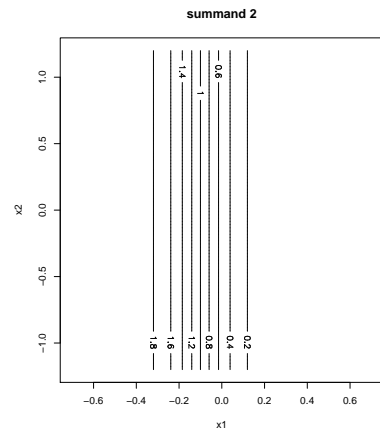
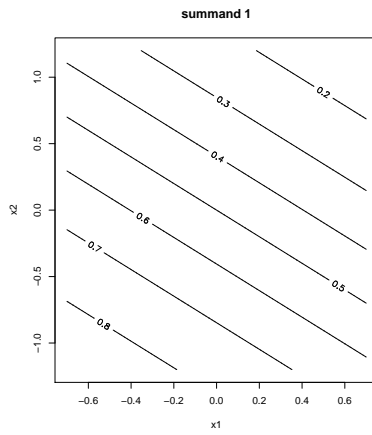
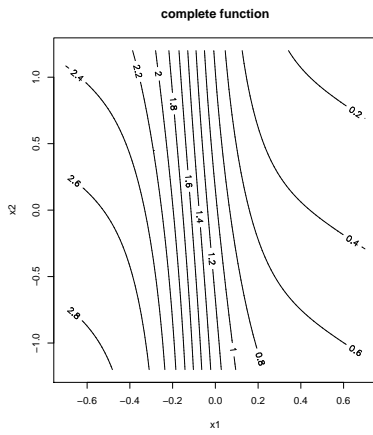
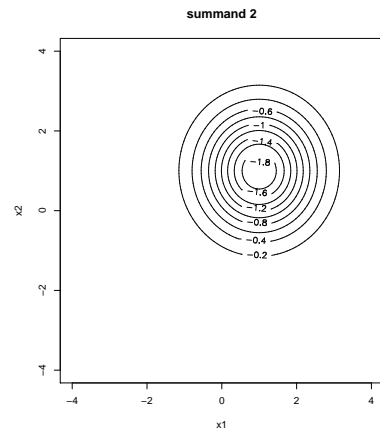
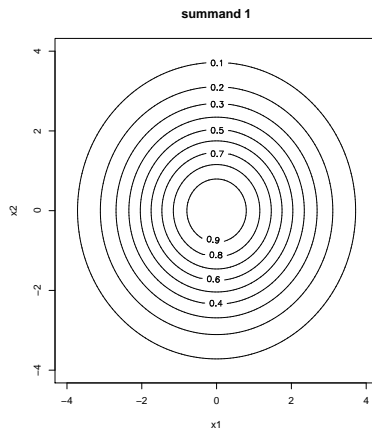
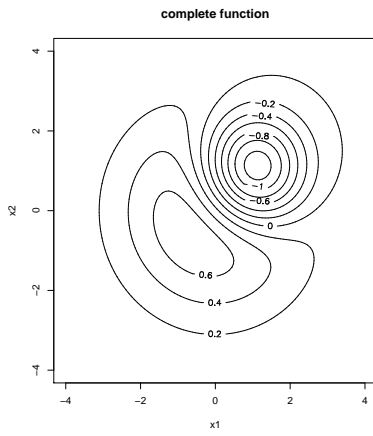
and in some sense, averaging across training inputs moves penalty from “bias” to “variance.”

(2.8) (15 points) The linear regression method has a higher training error than the KNN method when $k = 1, 3$ and a higher test error than the KNN method for all values of k . KNN with $k = 1$ provides the lowest test error (and, obviously, training error) for all methods.



| | Training | Test |
|-------------------|----------|---------|
| linear regression | 0.00576 | 0.04121 |
| $k = 1$ | 0.00000 | 0.02473 |
| $k = 3$ | 0.00504 | 0.03022 |
| $k = 5$ | 0.00576 | 0.03022 |
| $k = 7$ | 0.00648 | 0.03297 |
| $k = 15$ | 0.00936 | 0.03846 |

Problem 2: (5 points)



Problem 3: Regression table assignment (25 points)

sample code

```
####Normalizing the x's and centering the y's for the training set then applying to the test set
p=8
T=cov(prostatetrain[,1:8])
V=sqrt(diag(T)[1:p])
M=apply(prostatetrain[,1:9],2,mean)

for(i in 1:8){
  prostate[,i]=(prostate[,i]-M[i])/V[i]}

prostate[,9]=prostate[,9]-M[9]

prostatetrain=prostate[prostate[,10]=="TRUE",]
prostatetest=prostate[prostate[,10]=="FALSE",]

#####Least Squares#####
L=lm(lpsa~lcavol+lweight+age+lbph+svi+lcp+gleason+pgg45-1,prostatetrain)

v=matrix(L$coefficients[1:8],nrow=1)
predictlm=v%*%t(prostatetest[,1:8])
residlm=prostatetest[,9]-predictlm
testerrlm=mean(residlm^2)
selm=sd(c(residlm^2))/sqrt(27)

#####Best Subset#####
library(leaps)
l2=leaps(prostatetrain[,1:8],prostatetrain[,9],method="r2",int=FALSE,nbest=1)
keep=(1:8)[l2$which[2,]]
xnam=names(prostatetrain)[keep]
fmla=as.formula(paste("lpsa ~ ", paste(xnam, collapse= "+"),"-1"))
L2=lm(fmla,prostatetrain)
fit=L2$coefficients%*%t(prostatetest[,keep])
residbs=prostatetest[,9]-fit
testerrorbs=mean(residbs^2)
sebs=sd(c(residbs^2))/sqrt(27)

#####Ridge Regression#####
library(MASS)
d=svd(prostatetrain[,1:8])$d
f=function(lam){sum(d^2/(d^2+lam))-5}
lamb=uniroot(f,c(0,1000))$root
L2=lm.ridge(lpsa~lcavol+lweight+age+lbph+svi+lcp+gleason+pgg45-1,prostatetrain,lambda=lamb)
fit2=L2$coef%*%t(prostatetest[,1:8])
residR=prostatetest[,9]-fit2
testerrorR=mean(residR^2)
seR=sd(c(residR^2))/sqrt(27)

#####Lasso#####
library(lasso2)
L1=l1ce(lpsa~lcavol+lweight+age+lbph+svi+lcp+gleason+pgg45,prostatetrain,bound=0.5,trace=TRUE,absolute.t=FALSE)
fitL=L1$coef[2:9]%*%t(prostatetest[,1:8])
residL=prostatetest[,9]-fitL
testerrorL=mean(residL^2)
seL=sd(c(residL^2))/sqrt(27)
```

```

#####Principle Components Regression and Partial Least Squares#####
library(pls)
##Cross Validation for PCR
PredErrorPCR=matrix(nrow=9,ncol=10)
for(i in 1:10){
index=(1:70)[prostatetrain[,11]==i]
CVapply=prostatetrain[-index,]
CVtest=prostatetrain[index,]
PredErrorPCR[1,i]=mean(CVtest[,9]^2)
pc=pcr(lpsa~lcavol+lweight+age+lbph+svi+lcp+gleason+pgg45,8,data=CVapply,validation="none")
for(j in 1:8){
fitpc=pc$coef[,j]*%t(CVtest[,1:8])
residpc=CVtest[,9]-fitpc
PredErrorPCR[(j+1),i]=mean(residpc^2)}}

##Plot for PCR
SE=apply(PredErrorPCR,1,sd)/sqrt(10)
PCRerror=apply(PredErrorPCR,1,mean)
plot(seq(0,8,,100),seq(.4,1.8,,100),xlab="Number of Directions",
ylab="CV Error",type="n",main="Principal Components Regression")
points(0:8,PCRerror,pch=19)
lines(0:8,PCRerror)
points(0:8,PCRerror+SE,pch=3,col=4)
points(0:8,PCRerror-SE,pch=3,col=4)
for(i in 1:9){
lines(rep(i-1,2),c((PCRerror+SE)[i],(PCRerror-SE)[i]),col=4)}
abline(h=min(PCRerror+SE),col=3,lty=2)
abline(v=7,col=3,lty=2)

##Finding Errors
pc=pcr(lpsa ~lcavol+lweight+age+lbph+svi+lcp+gleason+pgg45 ,7, data=prostatetrain, validation="none")
pl=pls(lpsa ~lcavol+lweight+age+lbph+svi+lcp+gleason+pgg45 ,2, data=prostatetrain, validation="none")
fitpc=pc$coef[,7]*%t(prostatetest[,1:8])
fitpl=pl$coef[,2]*%t(prostatetest[,1:8])
residPCR=prostatetest[,9]-fitpc
residPLS=prostatetest[,9]-fitpl
testerrorPCR=mean(residPCR^2)
sePCR=sd(c(residPCR^2))/sqrt(27)
testerrorPLS=mean(residPLS^2)
sePLS=sd(c(residPLS^2))/sqrt(27)

```

Sample code for (2.8) and the contour plots of Problem 2

```
##Exercise 2.8 of HTF2
##ziptrain23 and ziptest23 represent the reduced data sets with 2's and 3's

####Training Data, k nearest neighbors
##Each column of the matrix called Cavg gives the mean of
##the labels (2 and 3s) for the k nearest neighbors
ks=c(3,5,7,15)
Cavg=matrix(nrow=1389,ncol=4)
neigh=nn(ziptrain23,p=14)$nn.idx
for(i in 1:4){
  k=ks[i]
  C=matrix(nrow=1389,ncol=k)
  C[,1]=ziptrain23[,257]
  for(j in 2:k){
    C[,j]=ziptrain23[neigh[, (j-1)],257]}
  Cavg[,i]=apply(C,1,mean)}

##If the average is >2.5 assign a 2, else a 3
Class=round(Cavg)

##compute the error rate
apply(abs(Class-ziptrain23[,257]),2,sum)/1389

###linear model applied to training data
linfit=lm(ziptrain23[,257]~ziptrain23[,1:256])
linfitc=linfit$coefficients

###Classifying the training data and computing the error based on the linear model
Clintrain=ifelse(linfit$fitted.values>2.5,3,2)
lmerrortrain=sum(abs(Clintrain-ziptrain23[,257]))/1389

###Classifying the test set data computing the error based on the linear model
Clintest=rep(0,364)
for(i in 1:364){
  Clintest[i]=ifelse((linfitc[1]+sum(linfitc[2:257]*ziptest23[i,1:256]))>2.5,3,2)}
lmerrortest=sum(abs(Clintest-ziptest23[,257]))/364

#####Test Data, k nearest neighbors
##Each column of the matrix called Cavg2 gives the mean of the
##labels (2 and 3s) for the k nearest neighbors
ks2=c(1,3,5,7,15)
Cavg2=matrix(nrow=364,ncol=5)
for(m in 1:364){
  ziptestC=rbind(ziptest23[m,],ziptrain23)
  neigh=as.matrix(nn(ziptestC,p=15)$nn.idx[1,])
  for(i in 1:5){
    k=ks2[i]
    C=ziptestC[c(neigh)[1:k],257]
    Cavg2[m,i]=mean(C)}}

##If the average is >2.5 assign a 2, else a 3
Class2=round(Cavg2)

##compute the error rate
testerror=apply(abs(Class2-ziptest23[,257]),2,sum)/364
```

```

#####PROBLEM 2#####
#Part i)
f1=function(x1,x2){exp(-(1/6)*(x1^2+x2^2))}
f2=function(x1,x2){-2*exp(-.5*((x1-1)^2+(x2-1)^2)}
f=function(x1,x2){f1(x1,x2)+f2(x1,x2)}
x1=seq(-4,4,,100)
x2=seq(-4,4,,100)
par(mfrow=c(1,3))
z=outer(x1,x2,"f")
contour(x1,x2,z,xlab="x1",ylab="x2",main="complete function")
z=outer(x1,x2,"f1")
contour(x1,x2,z,xlab="x1",ylab="x2",main="summand 1")
z=outer(x1,x2,"f2")
contour(x1,x2,z,xlab="x1",ylab="x2",main="summand 2")

#Part ii)
f1=function(x1,x2){1/(1+exp(x1+x2))}
f2=function(x1,x2){2/(1+exp(1+10*x1))}
f=function(x1,x2){f1(x1,x2)+f2(x1,x2)}
x1=seq(-.7,.7,,100)
x2=seq(-1.2,1.2,,100)
par(mfrow=c(1,3))
z=outer(x1,x2,"f")
contour(x1,x2,z,xlab="x1",ylab="x2",main="complete function",nlevels=15)
z=outer(x1,x2,"f1")
contour(x1,x2,z,xlab="x1",ylab="x2",main="summand 1")
z=outer(x1,x2,"f2")
contour(x1,x2,z,xlab="x1",ylab="x2",main="summand 2")

#Part iii)
f1=function(x1,x2){x1+x2}
f2=function(x1,x2){sin(x1)}
f=function(x1,x2){f1(x1,x2)+f2(x1,x2)}
x1=seq(-10,10,,100)
x2=seq(-10,10,,100)
par(mfrow=c(1,3))
z=outer(x1,x2,"f")
contour(x1,x2,z,xlab="x1",ylab="x2",main="complete function")
z=outer(x1,x2,"f1")
contour(x1,x2,z,xlab="x1",ylab="x2",main="summand 1")
z=outer(x1,x2,"f2")
contour(x1,x2,z,xlab="x1",ylab="x2",main="summand 2")

```