Problem1.

Multinom:

setwd("C:/Users/Administrator/Desktop/machine learning")

library(nnet)

train <- read.table("zip\_train")

subtrain <- subset(train,train[,1]== 3|train[,1] == 5|train[,1] == 8) **#filter the data,get 3s,5s,8s**

test <- read.table("zip\_test")

subtest <- subset(test,test[,1] == 3|test[,1]==5|test[,1]==8)

mod1 <- multinom(V1~.,data = subtrain) **#call multinom function to fit the model**

predict1 <- predict(mod1,newdata= subtest) **#get predicted values**

table(subtest$V1,predict1)

predict1

3 5 8

3 147 10 9

5 11 137 12

8 6 9 151

mean(predict1 != subtest$V1) **# calculate the test error rate**

[1] 0.1158537

Knn:

library(class)

knn1 <- knn(subtrain,subtest,cl=subtrain$V1,k=1) **# k=1**

table(subtest$V1,knn1)

knn1

3 5 8

3 158 8 0

5 4 nn:inom:154 2

8 3 5 158

mean(knn1 != subtest$V1) **# calculate the test error rate of knn when k = 1**

[1] 0.04471545

knn2 <- knn(subtrain,subtest,cl=subtrain$V1,k=3) **# k=3**

table(subtest$V1,knn2)

knn2

3 5 8

3 157 9 0

5 3 157 0

8 2 3 161

mean(knn2 != subtest$V1) **# calculate the test error rate of knn when k = 3**

[1] 0.03455285

knn3 <- knn(subtrain,subtest,cl=subtrain$V1,k=5)

table(subtest$V1,knn3)

knn3

3 5 8

3 156 10 0

5 4 156 0

8 2 3 161

mean(knn3 != subtest$V1) **# calculate the test error rate of knn when k = 5**

[1] 0.03861789

knn4 <- knn(subtrain,subtest,cl=subtrain$V1,k=7)

table(subtest$V1,knn4)

knn4

3 5 8

3 157 9 0

5 2 157 1

8 1 3 162

mean(knn4 != subtest$V1) **# calculate the test error rate of knn when k = 7**

[1] 0.03252033

knn5 <- knn(subtrain,subtest,cl=subtrain$V1,k=15)

table(subtest$V1,knn5)

knn5

3 5 8

3 158 8 0

5 4 155 1

8 1 2 163

mean(knn5 != subtest$V1) **# calculate the test error rate of knn when k = 15**

[1] 0.03252033

**Comparing the Knn and multinom model,we can see that the overall performance of Knn function is much better than multinom model.**

Problem2.

SAheart <- read.csv("SAheart.data")

indictormatrix <- matrix(runif(924,0,0),nrow=462,ncol=2)  **#creat a matrix filled with 0**

for (i in 1:462){ **#use for loop to change the matrix into indictor matrix**

if (SAheart[i,11]==0){

indictormatrix[i,1]=1}else{

indictormatrix[i,2]=1}}

X <- SAheart[,1:10]

X <- X[,-6] **#for convenience,I delete the categorical variables here.**

X <- as.matrix(X)

X <- X[,-1]

TX <- t(X)

productX <- TX %\*% X

inverseproductX <- solve(productX)

Yhat = X %\*% inverseproductX %\*% TX %\*% indictormatrix

library(MASS)

lda.fit1 = lda(chd~sbp+tobacco+ldl+adiposity+typea+obesity+alcohol+age,

data=SAheart)

Ydf <- data.frame(Yhat)

Ydf$Y <- SAheart$chd

lda.fit2 = lda(Y~X1 + X2, data=Ydf)

SAheart<-SAheart[,-6]

SAheart<-SAheart[,-1]

ldaRes = predict(lda.fit1, SAheart)

table(SAheart$chd, ldaRes$class) **# this is the train error in the original case**

0 1

0 254 48

1. 79 81

ldaRes1 = predict(lda.fit2, Ydf)

table(Ydf$Y, ldaRes1$class) **# this is the train error using Yhat**

0 1

0 254 48

1 79 81

table(ldaRes1$class,ldaRes$class)

0 1

0 333 0

1. 0 129

**We can conclude that LDA using ˆY is identical to LDA in the original space**

Problem4

Easy: **# Easy situation**

test<-matrix(runif(11000,0,1),nrow=1000,ncol=11) **# produce random numbers**

train<-matrix(runif(1100,0,1),nrow=100,ncol=11)

test<-data.frame(test)

train<-data.frame(train)

for (i in 1:1000){

if (test[i,1] >0.5){

test[i,11] = 1}

else{

test[i,11] = 0

}}

for (i in 1:100){

if (train[i,1] >0.5){

train[i,11] = 1}

else{

train[i,11] = 0

}}  **# I set k=1,5,10,et. to draw the plot**

knn1 <- knn(train[,1:10],test[,1:10],train$X11,k=1)

knn5 <- knn(train[,1:10],test[,1:10],train$X11,k=5)

knn10 <- knn(train[,1:10],test[,1:10],train$X11,k=10)

knn18 <- knn(train[,1:10],test[,1:10],train$X11,k=18)

knn25 <- knn(train[,1:10],test[,1:10],train$X11,k=25)

knn32 <- knn(train[,1:10],test[,1:10],train$X11,k=32)

knn40 <- knn(train[,1:10],test[,1:10],train$X11,k=40)

knn45 <- knn(train[,1:10],test[,1:10],train$X11,k=45)

knn55 <- knn(train[,1:10],test[,1:10],train$X11,k=55)

knn60 <- knn(train[,1:10],test[,1:10],train$X11,k=60)

knn70 <- knn(train[,1:10],test[,1:10],train$X11,k=70)

table(test$X11,knnRes1)

mean(knnRes1 != test$X11)

forplot<-matrix(runif(22,0,0),nrow=11,ncol=2)

**# here I create a new dataframe to save the k value and test error**

forplot <- data.frame(forplot)

colnames(forplot)<-c("k","error\_rate")

forplot[1:11,1] <- c("1","5","10","18","25","32","40","45","55","60","70")

forplot[1:11,2] <- c(mean(knn1 != test$X11),mean(knn5 != test$X11),mean(knn10 !=

test$X11),mean(knn18 != test$X11),mean(knn25 != test$X11),mean(knn32 != test

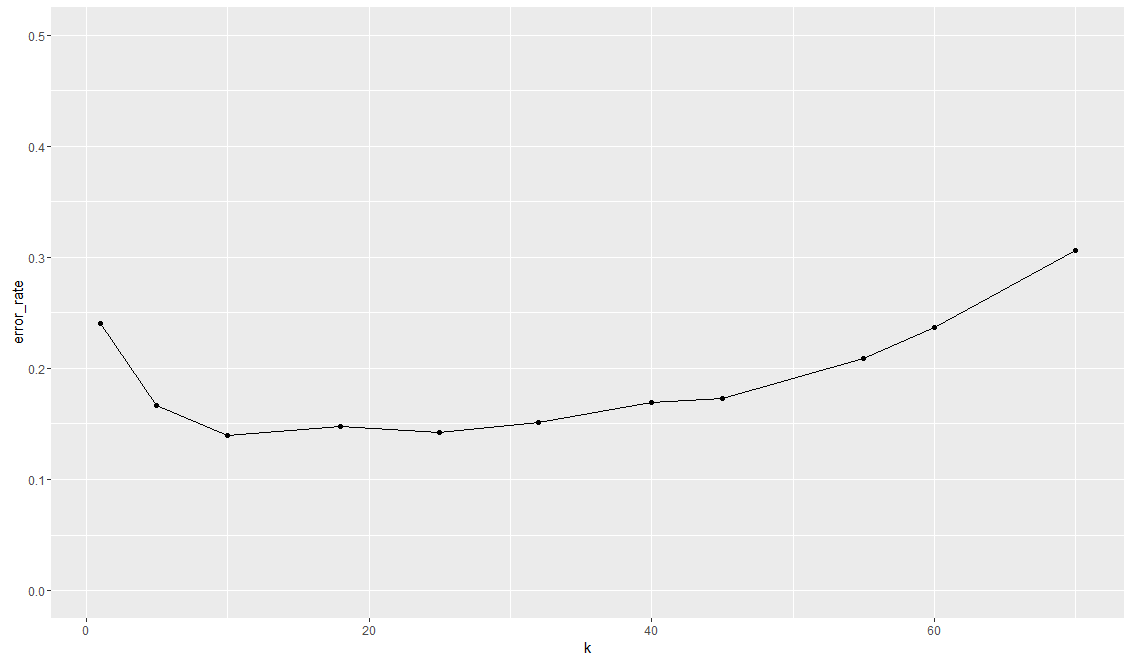
$X11),mean(knn40 != test$X11),mean(knn45 != test$X11),mean(knn55 != test

$X11),mean(knn60 != test$X11),mean(knn70 != test$X11))

forplot$k <- as.integer(forplot$k)

plot1 <- ggplot(data=forplot,mapping=aes(k,error\_rate))+geom\_line()+geom\_point()+ylim(0,0.5)

plot1



Difficult:

test<-matrix(runif(11000,0,1),nrow=1000,ncol=11)

train<-matrix(runif(1100,0,1),nrow=100,ncol=11)

test<-data.frame(test)

train<-data.frame(train)

for (i in 1:1000){

if ((test[i,1]-0.5)\*(test[i,2]-0.5)\*(test[i,3]-0.5) >0){

test[i,11] = 1}

else{

test[i,11] = 0

}}

for (i in 1:100){

if ((test[i,1]-0.5)\*(test[i,2]-0.5)\*(test[i,3]-0.5) >0){

train[i,11] = 1}

else{

train[i,11] = 0

}}

knn1 <- knn(train[,1:10],test[,1:10],train$X11,k=1)

knn5 <- knn(train[,1:10],test[,1:10],train$X11,k=5)

knn10 <- knn(train[,1:10],test[,1:10],train$X11,k=10)

knn18 <- knn(train[,1:10],test[,1:10],train$X11,k=18)

knn25 <- knn(train[,1:10],test[,1:10],train$X11,k=25)

knn32 <- knn(train[,1:10],test[,1:10],train$X11,k=32)

knn40 <- knn(train[,1:10],test[,1:10],train$X11,k=40)

knn45 <- knn(train[,1:10],test[,1:10],train$X11,k=45)

knn55 <- knn(train[,1:10],test[,1:10],train$X11,k=55)

knn60 <- knn(train[,1:10],test[,1:10],train$X11,k=60)

knn70 <- knn(train[,1:10],test[,1:10],train$X11,k=70)

table(test$X11,knnRes1)

mean(knnRes1 != test$X11)

forplot<-matrix(runif(22,0,0),nrow=11,ncol=2)

forplot <- data.frame(forplot)

colnames(forplot)<-c("k","error\_rate")

forplot[1:11,1] <- c("1","5","10","18","25","32","40","45","55","60","70")

forplot[1:11,2] <- c(mean(knn1 != test$X11),mean(knn5 != test$X11),mean(knn10 !=

test$X11),mean(knn18 != test$X11),mean(knn25 != test$X11),mean(knn32 != test

$X11),mean(knn40 != test$X11),mean(knn45 != test$X11),mean(knn55 != test

$X11),mean(knn60 != test$X11),mean(knn70 != test$X11))

forplot$k <- as.integer(forplot$k)

plot2 <- ggplot(data=forplot,mapping=aes(k,error\_rate))+geom\_line()+geom\_point()+ylim(0.4,0.6)

plot2

