##############KNN###################################

#k=1

library(class)

knn.pred=knn(traindata, testdata, usedata[(1:(n/2)),m],k = 1)

testtable=table(knn.pred ,usedata[15082:30162,m])

traintable=table(knn.pred ,usedata[1:15081,m])

knn.misrate.test=(testtable[1,2]+testtable[2,1])/n

knn.misrate.train=(traintable[1,2]+traintable[2,1])/n

knn.misrate.test

knn.misrate.train

> knn.misrate.test

[1] 0.1451495

> knn.misrate.train

[1] 0.1911677

So for k=1, 14.5% of the observations in the test set are incorrectly predicted. Of course, it may be that K = 1 results in an overly flexible fit to the data. Below, we repeat the analysis using K = 3.

#k=3

knn.pred=knn(traindata, testdata, usedata[(1:(n/2)),m],k = 3)

testtable=table(knn.pred ,usedata[15082:30162,m])

traintable=table(knn.pred ,usedata[1:15081,m])

knn.misrate.test=(testtable[1,2]+testtable[2,1])/n

knn.misrate.train=(traintable[1,2]+traintable[2,1])/n> knn.misrate.test

knn.misrate.test

knn.misrate.train

[1] 0.1267157

> knn.misrate.train

[1] 0.1693522

The results have improved and the mis-classsified rate of train set and test set have decreased both. We will try K=5 next step.

#k=5

knn.pred=knn(traindata, testdata, usedata[(1:(n/2)),m],k = 5)

testtable=table(knn.pred ,usedata[15082:30162,m])

traintable=table(knn.pred ,usedata[1:15081,m])

knn.misrate.test=(testtable[1,2]+testtable[2,1])/n

knn.misrate.train=(traintable[1,2]+traintable[2,1])/n

knn.misrate.test

knn.misrate.train

> knn.misrate.test

[1] 0.118394

> knn.misrate.train

[1] 0.1571845

The result of k=5 even becomes better. But does it mean the misrate keeps on decreasing with increasing K further? We will try K=10.

#k=10

knn.pred=knn(traindata, testdata, usedata[(1:(n/2)),m],k = 10)

testtable=table(knn.pred ,usedata[15082:30162,m])

traintable=table(knn.pred ,usedata[1:15081,m])

knn.misrate.test=(testtable[1,2]+testtable[2,1])/n

knn.misrate.train=(traintable[1,2]+traintable[2,1])/n

knn.misrate.test

knn.misrate.train

[1] 0.1107354

> knn.misrate.train

[1] 0.1456137

#k=15

library(class)

knn.pred=knn(traindata, testdata, usedata[(1:(n/2)),m],k = 15)

testtable=table(knn.pred ,usedata[15082:30162,m])

traintable=table(knn.pred ,usedata[1:15081,m])

knn.misrate.test=(testtable[1,2]+testtable[2,1])/n

knn.misrate.train=(traintable[1,2]+traintable[2,1])/n

knn.misrate.test

knn.misrate.train

> knn.misrate.test

[1] 0.1072542

> knn.misrate.train

[1] 0.1400769

#k=20

knn.pred=knn(traindata, testdata, usedata[(1:(n/2)),m],k = 20)

testtable=table(knn.pred ,usedata[15082:30162,m])

traintable=table(knn.pred ,usedata[1:15081,m])

knn.misrate.test=(testtable[1,2]+testtable[2,1])/n

knn.misrate.train=(traintable[1,2]+traintable[2,1])/n

knn.misrate.test

knn.misrate.train

> knn.misrate.test

[1] 0.1069558

> knn.misrate.train

[1] 0.1373914

#k=25

library(class)

knn.pred=knn(traindata, testdata, usedata[(1:(n/2)),m],k = 25)

testtable=table(knn.pred ,usedata[15082:30162,m])

traintable=table(knn.pred ,usedata[1:15081,m])

knn.misrate.test=(testtable[1,2]+testtable[2,1])/n

knn.misrate.train=(traintable[1,2]+traintable[2,1])/n

knn.misrate.test

knn.misrate.train

> knn.misrate.test

[1] 0.1071547

> knn.misrate.train

[1] 0.1353359

#k=30

knn.pred=knn(traindata, testdata, usedata[(1:(n/2)),m],k = 30)

testtable=table(knn.pred ,usedata[15082:30162,m])

traintable=table(knn.pred ,usedata[1:15081,m])

knn.misrate.test=(testtable[1,2]+testtable[2,1])/n

knn.misrate.train=(traintable[1,2]+traintable[2,1])/n

knn.misrate.test

knn.misrate.train

> knn.misrate.test

[1] 0.1073536

> knn.misrate.train

[1] 0.1341423

In order to find the optimal K that minimize mis-classified rate of the test data, we try every integer from 26 to 29 as K.

misrate.test=rep(0,4)

misrate.train=rep(0,4)

for(i in 1:4)

{

knn.pred=knn(traindata, testdata, usedata[(1:(n/2)),m],k = i+25)

testtable=table(knn.pred ,usedata[15082:30162,m])

traintable=table(knn.pred ,usedata[1:15081,m])

knn.misrate.test=(testtable[1,2]+testtable[2,1])/n

knn.misrate.train=(traintable[1,2]+traintable[2,1])/n

misrate.test[i]=knn.misrate.test

misrate.train[i]=knn.misrate.train

}

> misrate.test

[1] 0.1069226 0.1067568 0.1070221 0.1072210

> misrate.train

[1] 0.1352364 0.1348054 0.1344738 0.1342749

When K=27, we get the min test error of 13.67%. And the train error is 13.48%.