Martha Foye, Dave Hiltbrand, and Allen Phelps

Dr. Cuixian Chen

STT 592

4 October 2016

**Group Report 4**

In search of the best classification method for the professional diagnosis of Posttraumatic Stress Disorder as modeled by the t-scores of the Anxious Arousal and Defensive Avoidance clinical scales of the Trauma Symptom Inventory from the blast dataset, the k-Nearest Neighbors (k-NN) algorithm was considered as a non-parametric alternative to Logistic Regression as well as Linear and Quadratic Discriminant Analysis. Since the constant, k, is data-driven, trial and error was used at first to zero in on k=20, other test values including 1, 5, 10, and finally 30, where a decrease in overall classification accuracy was observed. Rather than continue to guess and check which value between 20 and 30 marked the initial decrease, 5-fold cross validation was programmed in a loop to make the determination computationally. R confirmed that 20 tied with 22 for the optimal value of k between 1 and 30; thus, 20 was selected to run the k-NN algorithm on the selected subset of data.

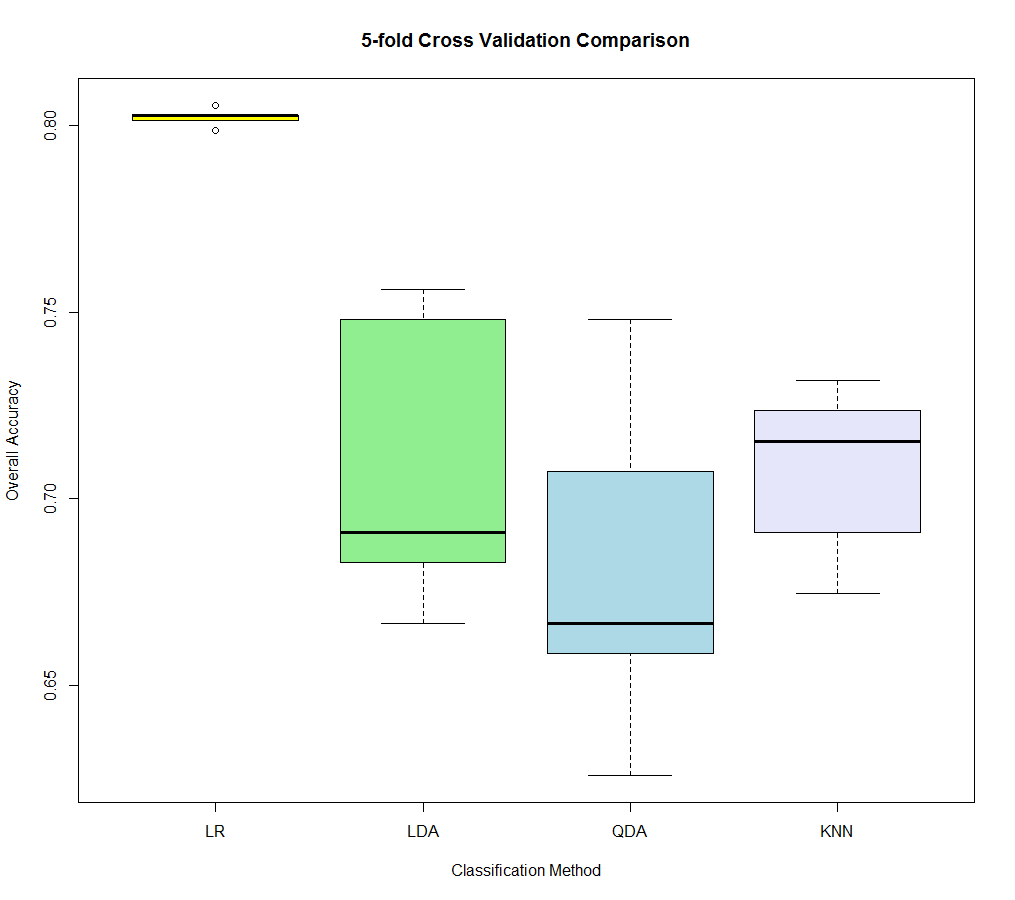
In keeping with the data mining procedure developed thus far, 20-Nearest Neighbors was initially run on the full selection of data with the following results:

k FALSE TRUE

FALSE 169 67

TRUE 106 273

That is 71.86992% overall classification accuracy; however, rather than continue the usual procedure by training an arbitrary 80% of the data to test the remaining 20%, 5-fold cross validation was also programmed in a loop to select these proportions randomly. The same was repeated for the prior three classification methods toward the end of comparison. The distributions of the resultant randomized folds in terms of overall accuracy by classification method can be visually compared with side-by-side boxplots:



The randomization also resulted in a different conclusion than before in that Logistic Regression performed better than both Linear and Quadratic Discriminant Analysis as originally expected since the groups—professionally diagnosed with Posttraumatic Stress Disorder or not—are not well-separated by the given predictors. The randomization of alternative cross validation method, Leave One Out Cross Validation (LOOCV), resulted in much of the same. The only difference is LOOCV treats each individual observation as a fold, so the overall classification accuracy assigned per iteration is binary; therefore, side-by-side boxplots are not very informative in this case. Rather, the mean number of correct classifications per method is provided for comparison of Logistic Regression, Linear and Quadratic Discriminant Analysis, and k-Nearest Neighbors, respectively:

0.8033903 0.7089431 0.6813008 0.6959350

**Appendix: R Code**

#Allen Phelps

dat=read.csv("stt592dat.csv", header=T)

attach(dat)

dat=data.frame(ptsd,aa,da,aa\*da)

dat=dat[complete.cases(dat),]

library(boot)

library(MASS)

#Logistic Regression LOO CV

y=glm(ptsd~aa\*da,data=dat,family=binomial)

error=cv.glm(dat,y)

lr1out=1-error$delta;lr1out

#Logistic Regression K-fold CV

set.seed(1)

accuracy=rep(0,5)

for (i in 1:5)

{

y=glm(ptsd~poly(aa\*da,i),data=dat,family=binomial)

accuracy[i]=1-cv.glm(dat,y,K=5)$delta[1]

}

lr5out=accuracy;lr5out

#LDA K-fold CV

nfolds=5

foldi=sample(rep(1:nfolds,length.out=dim(dat)[1]))

table(foldi)

lda5out=NULL

for(k in 1:nfolds)

{

testi=which(foldi==k)

train=dat[-testi,]

test=dat[testi,]

l=lda(ptsd~aa\*da,data=dat)

lpred=predict(l,test[,2:4])

lclass=lpred$class

table(lclass,test[,1])

accuracy=mean(lclass==test[,1])

lda5out=c(lda5out,accuracy)

}

print(lda5out)

mean(lda5out)

boxplot(lda5out,col='green')

#LDA LOO CV

n=dim(dat)[1]

foldi=sample(rep(1:n,length.out=n))

table(foldi)

lda1out=NULL

for(k in 1:n)

{

testi=which(foldi==k)

train=dat[-testi,]

test=dat[testi,]

l=lda(ptsd~aa\*da,data=dat)

lpred=predict(l,test[,2:4])

lclass=lpred$class

table(lclass,test[,1])

accuracy=mean(lclass==test[,1])

lda1out=c(lda1out,accuracy)

}

print(lda1out)

mean(lda1out)

boxplot(lda1out,col='green')

#QDA K-fold CV

nfolds=5

foldi=sample(rep(1:nfolds,length.out=dim(dat)[1]))

table(foldi)

qda5out=NULL

for(k in 1:nfolds)

{

testi=which(foldi==k)

train=dat[-testi,]

test=dat[testi,]

q=qda(ptsd~aa\*da,data=dat)

qpred=predict(q,test[,2:4])

qclass=qpred$class

table(qclass,test[,1])

accuracy=mean(qclass==test[,1])

qda5out=c(qda5out,accuracy)

}

print(qda5out)

mean(qda5out)

boxplot(qda5out,col='green')

#QDA LOO CV

n=dim(dat)[1]

foldi=sample(rep(1:n,length.out=n))

table(foldi)

qda1out=NULL

for(k in 1:n)

{

testi=which(foldi==k)

train=dat[-testi,]

test=dat[testi,]

q=qda(ptsd~aa\*da,data=dat)

qpred=predict(q,test[,2:4])

qclass=qpred$class

table(qclass,test[,1])

accuracy=mean(qclass==test[,1])

qda1out=c(qda1out,accuracy)

}

print(qda1out)

mean(qda1out)

boxplot(qda1out,col='green')

#Dave Hiltbrand

dat=read.csv("stt592dat.csv", header=T)

attach(dat)

names(dat)

dat=data.frame(chartn,ptsd,dat[,19:31])

dat=dat[complete.cases(dat),]

dim(dat)

library(class)

## k-Fold CV to determine best K for KNN

n=dim(dat)[1]; m=dim(dat)[2]; print(c(m,n))

n\_fold<-5;

rep(1:n\_fold, length.out = n)

folds\_i <- sample(rep(1:n\_fold, length.out = n))

table(folds\_i)

Final.OUT=NULL

for (i in 1:30)

{

OUT.KNN=NULL

for (j in 1:n\_fold)

{

test.ID <- which(folds\_i == j)

train\_X <- dat[-test.ID, c("aa","da")]

train\_Y <- dat[-test.ID, 2]

test\_X <- dat[test.ID, c("aa","da")]

test\_Y <- dat[test.ID, 2]

knn.pred=knn(train\_X, test\_X, train\_Y, k=i)

Accuracy=mean(knn.pred==test\_Y)

OUT.KNN=c(OUT.KNN, Accuracy)

}

Final.OUT=rbind(Final.OUT, OUT.KNN)

}

print(Final.OUT)

apply(Final.OUT, 1, mean)

rowMeans(Final.OUT)

max(rowMeans(Final.OUT))

boxplot(t(Final.OUT), col=rainbow(10))

## KNN

set.seed(1)

x=cbind(dat$aa,dat$da)

k=knn(x,x,dat$ptsd,k=20,prob=T)

table(k,dat$ptsd)

mean(k==dat$ptsd)

##k-fold:5 CV for KNN

n\_fold<-5;

rep(1:n\_fold, length.out = n)

folds\_i <- sample(rep(1:n\_fold, length.out = n))

table(folds\_i)

OUT.KNN=NULL

for (j in 1:n\_fold)

{

test.ID <- which(folds\_i == j)

train\_X <- dat[-test.ID, c("aa","da")]

train\_Y <- dat[-test.ID, 2]

test\_X <- dat[test.ID, c("aa","da")]

test\_Y <- dat[test.ID, 2]

knn.pred=knn(train\_X, test\_X, train\_Y, k=20)

table(knn.pred,test\_Y)

Accuracy=mean(knn.pred==test\_Y)

OUT.KNN=c(OUT.KNN, Accuracy)

}

print(OUT.KNN)

mean(OUT.KNN) ##overall accuracy

boxplot(OUT.KNN,col="orange")

## LOO CV for KNN

folds\_i <- sample(rep(1:n, length.out = n))

table(folds\_i)

OUT.KNN1=NULL

for (j in 1:n)

{

test.ID <- which(folds\_i == j)

train\_X <- dat[-test.ID, c("aa","da")]

train\_Y <- dat[-test.ID, 2]

test\_X <- dat[test.ID, c("aa","da")]

test\_Y <- dat[test.ID, 2]

knn.pred=knn(train\_X, test\_X, train\_Y, k=20)

table(knn.pred,test\_Y)

Accuracy=mean(knn.pred==test\_Y)

OUT.KNN1=c(OUT.KNN1, Accuracy)

}

print(OUT.KNN1)

mean(OUT.KNN1) ##overall accuracy

boxplot(OUT.KNN1,col="orange")

boxplot(lr5out,lda5out,qda5out,OUT.KNN,col=c('yellow','lightgreen','lightblue','lavender'),main='5-fold Cross Validation Comparison',names=c('LR','LDA','QDA','KNN'),xlab='Classification Method',ylab='Overall Accuracy')

out1=c(mean(lr1out),mean(lda1out),mean(qda1out),mean(OUT.KNN1))

out1