Martha Foye, Allen Phelps, and Dave Hiltbrand

Dr. Cuixian Chen

STT 592

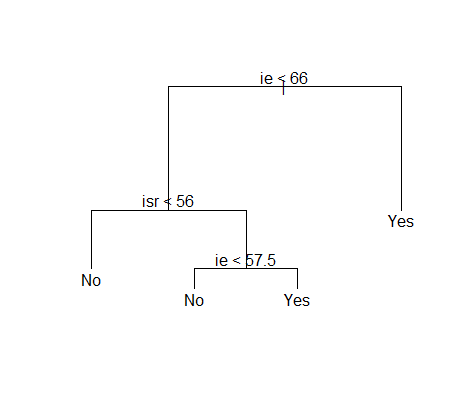
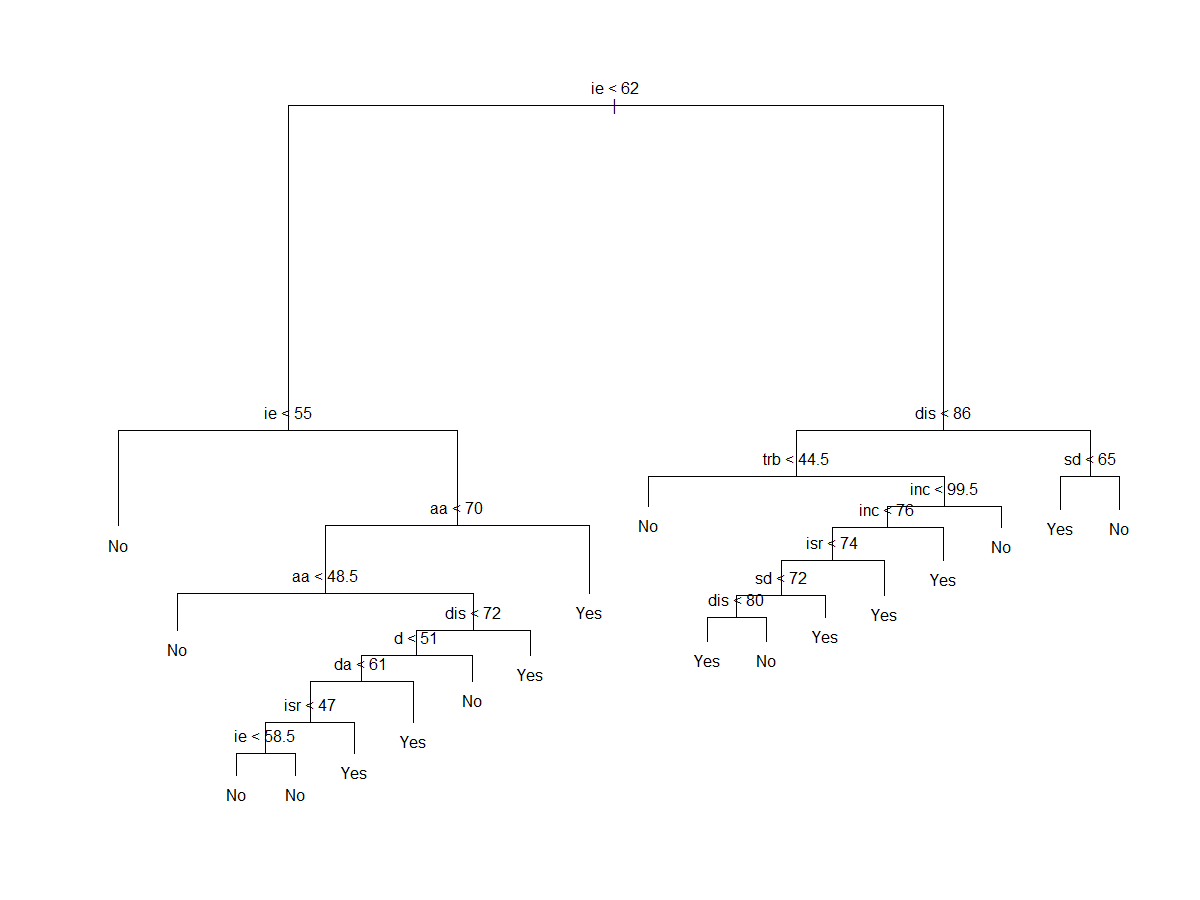
25 October 2016

**Group Report 5**

In researching our data set we sought new types of modeling to better understand methods to predict PTSD within the individuals of our study. Decision tree modeling has a key advantage in that it allows a proper visual representation of the procedure showcasing the step by step determination used to calculate the overall accuracy and variance concluded from a data set. We used the 13 clinical scales found within the TSI to predict whether an individual would be diagnosed with PTSD by a psychological professional. For both the full model and pruned tree we applied the validation set approach to training and testing data but randomly splitting the 615 complete observations into 2 groups. Our decision tree proved to have moderate accuracy with a mean of **0.6612378** and a standard deviation of **0.4740616** with 18 terminal nodes as depicted in Figure 1. A common problem with decision trees is that they tend to yield high variance so we engaged the cross validation method of pruning through best fit to eliminate any variables that were causing unneeded variance while maintaining the lowest bias possible. Our result pruned the original 18 terminal nodes we used with all of predicators down to the best 4 as depicted in Figure 2. We found a slight improvement in our accuracy with a mean of **0.6807818** and standard deviation of **0.4669348**.

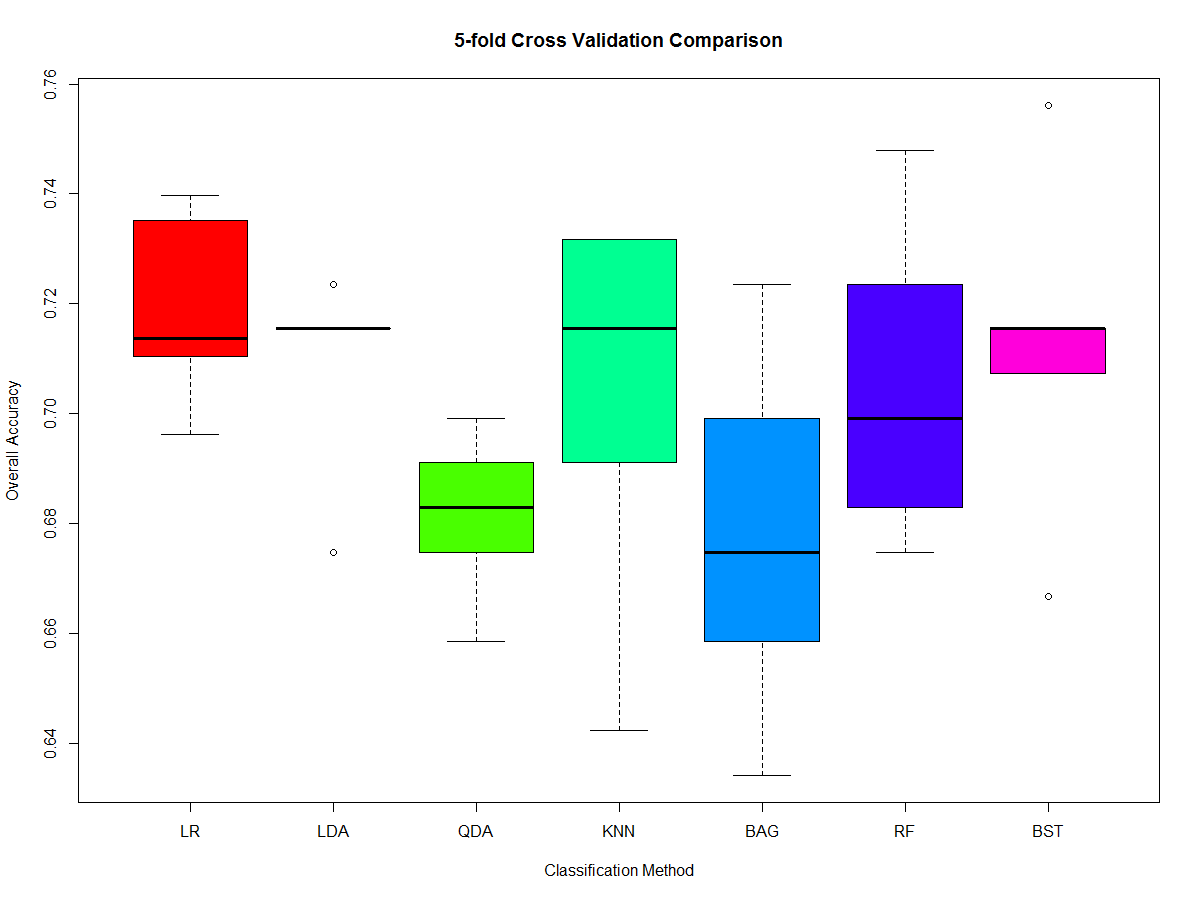
Following our results from the decision trees we further sought their potential by using bagging and random forest methods. Both of the methods were set up using K-Fold Cross Validation with K=5 random groups. Between the 2 models created random forest provided more desirable results with an accuracy mean of **0.7056911** and standard deviation of **0.03009229** while bagging left us with accuracy mean of **0.6780488** of and standard deviation of **0.03477927**. Bagging builds on the idea of decision trees to reduce variance by averaging the results of multiple trees bootstrapped from the original data set. so it should be no surprise that it resulted in better results than the unpruned decision tree. Random forest takes bagging a step further and helps offset the strength of any single predictor which again proved conclusive with the aforementioned results. Boosting was the final method implored by again taking bagging a step further except in this time the model learned slowly by building each sequential bootstrapped tree from the previous. Boosting proved to provide to best results between the 3 methods with an accuracy mean of **0.7121951** and standard deviation of **0.03180099**. One item of note is that bagging provided the lowest standard deviation of the 3 methods due to each bootstrapped tree using the original data set with m=p, predictors.

To maintain consistency with our previous LDA, QDA, KNN, and Logistic Regression models we ran Leave On Out Cross Validation on Bagging, Random Forest, and Boosting methods as well but the results were not as strong as that of K-Fold Cross Validation, K=5, as can be seen in Table-2. Comparing all 7 methods in the side-by-side boxplot of Figures 3 and 4 it is clear to see that Logistic Regression yields the highest overall accuracy and QDA delivers the lowest standard deviation.

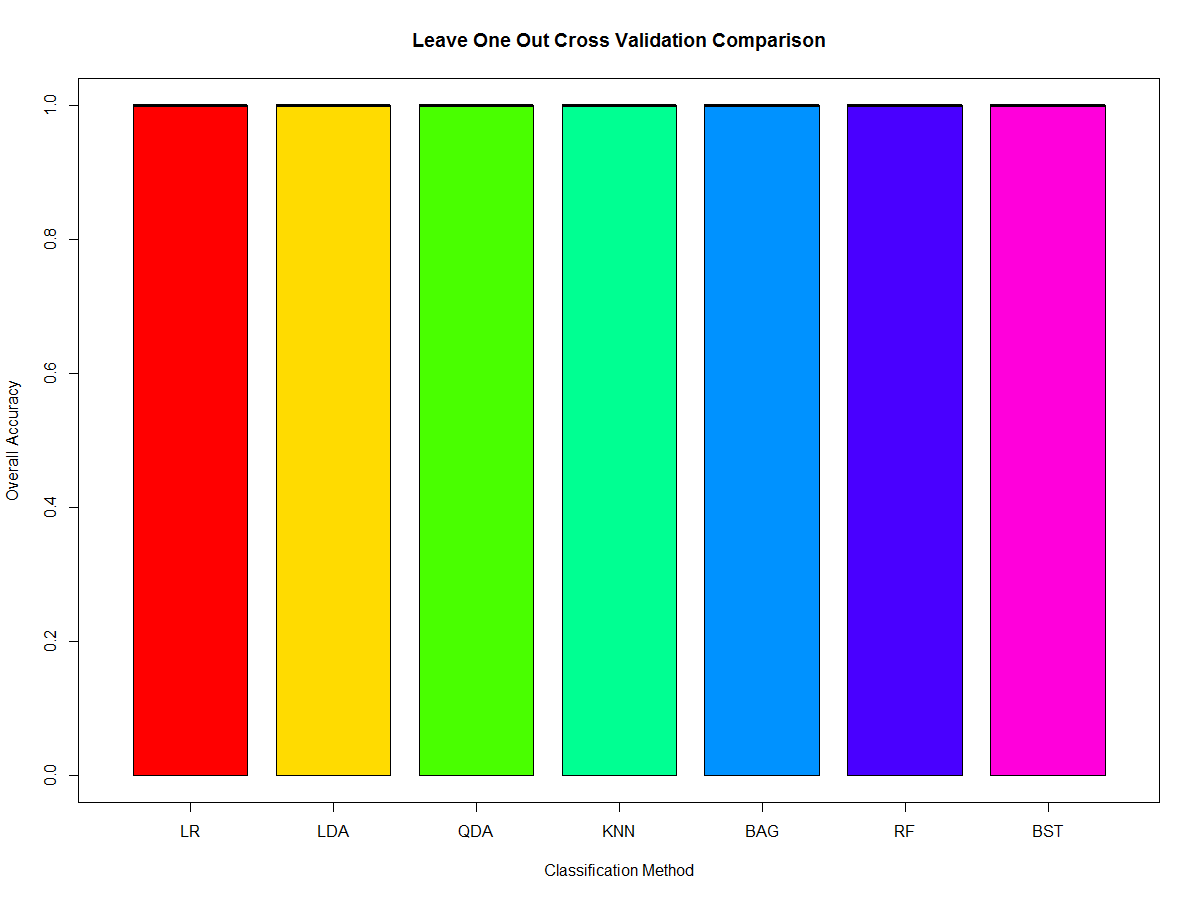
[[1]](#endnote-2)

**Figure 1**. The full *Decision Tree* based on the clinical scales scores of the Trauma Symptom Inventory, or TSI, scale. The first split is based on the Intrusive Experience, ie, being less than 62. (Clinical scale abbreviation descriptions found in Appendix 2: Clinical Scale Descriptions)

Figure 2. Pruned decision tree based on the clinical scales scores of the Trauma Symptom Inventory, or TSI, scale.



**Figure 3**. The Boxplot displays the prediction accuracy, from 0 to 1, for all seven classifier applying 5-fold CV: Logistic Regression, LDA, QDA, K-NN, Bagging, Random Forest, and Boosting.



**Figure 4.** The Boxplot displays the prediction accuracy, from 0 to 1, for all seven classifier applying Leave One Out Cross Validation (LOOCV): Logistic Regression, LDA, QDA, K-NN, Bagging, Random Forest, and Boosting.

|  |  |  |
| --- | --- | --- |
| Classification Method (5-fold CV) | Mean | SD |
| Logistic Regression | 0.7190422 | 0.01809779 |
| Linear Discriminant Analysis | 0.7089431 | 0.01941030 |
| Quadratic Discriminant Analysis | 0.6813008 | 0.01563852 |
| K Nearest Neighbors | 0.7024390 | 0.03752189 |
| Bagging | 0.6780488 | 0.03477927 |
| Random Forest | 0.7056911 | 0.03009229 |
| Boosting | 0.7121951 | 0.0318009 |

|  |  |  |
| --- | --- | --- |
| Classification Method (LOOCV) | Mean | SD |
| Logistic Regression | 0.6294581 | 0.4829513 |
| Linear Discriminant Analysis | 0.7089431 | 0.4546195 |
| Quadratic Discriminant Analysis | 0.6813008 | 0.4663514 |
| K Nearest Neighbors | 0.7056911 | 0.4561025 |
| Bagging | 0.6829268 | 0.465715 |
| Random Forest | 0.697561 | 0.4596882 |
| Boosting | 0.699187 | 0.4589848 |

**Appendix 1: R Code**

install.packages('gbm')

install.packages('randomForest')

install.packages('tree')

library(boot)

library(class)

library(gbm)

library(MASS)

library(randomForest)

library(tree)

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

names(dat)

ptsd=ifelse(dat[,13]==TRUE,'Yes','No')

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

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

#Classification Decision Tree

dim(dat)/2

set.seed(1)

traindat=sample(1:nrow(dat),308)

testdat=dat[-traindat,]

ytest=dat[,1][-traindat]

ytree=tree(ptsd~.,dat,subset=traindat)

summary(ytree)

plot(ytree)

text(ytree,pretty=0)

ypred=predict(ytree,testdat,type='class')

table(ypred,ytest)

mean(ypred==ytest)

sd(ypred==ytest)

set.seed(1)

cvytree=cv.tree(ytree,FUN=prune.misclass)

cvytree

yprune=prune.misclass(ytree,best=4)

plot(yprune)

text(yprune,pretty=0)

ypred=predict(yprune,testdat,type='class')

table(ypred,ytest)

mean(ypred==ytest)

sd(ypred==ytest)

#Bagging LOOCV

n=dim(dat)[1]

set.seed(1)

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

table(foldi)

bag1out=NULL

for(k in 1:n)

{

testi=which(foldi==k)

train=dat[-testi,]

test=dat[testi,]

ybag=randomForest(ptsd~.,data=train,mtry=13,ntree=500,importance=T)

bagpred=predict(ybag,test)

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

bag1out=c(bag1out,accuracy)

}

print(bag1out)

mean(bag1out) #0.6829268

sd(bag1out) #0.465715

boxplot(bag1out,col='green')

#Bagging 5-fold CV

nfolds=5

set.seed(1)

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

table(foldi)

bag5out=NULL

for(k in 1:nfolds)

{

testi=which(foldi==k)

train=dat[-testi,]

test=dat[testi,]

ybag=randomForest(ptsd~.,data=train,mtry=13,ntree=500,importance=T)

bagpred=predict(ybag,test)

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

bag5out=c(bag5out,accuracy)

}

print(bag5out)

mean(bag5out) #0.6780488

sd(bag5out) #0.03477927

boxplot(bag5out,col='green')

#Random Forest LOOCV

n=dim(dat)[1]

set.seed(1)

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

table(foldi)

rf1out=NULL

for(k in 1:n)

{

testi=which(foldi==k)

train=dat[-testi,]

test=dat[testi,]

yrf=randomForest(ptsd~.,data=train,mtry=4,ntree=500,importance=T)

rfpred=predict(yrf,test)

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

rf1out=c(rf1out,accuracy)

}

print(rf1out)

mean(rf1out) #0.697561

sd(rf1out) #0.4596882

boxplot(rf1out,col='green')

#Random Forest 5-fold CV

nfolds=5

set.seed(1)

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

table(foldi)

rf5out=NULL

for(k in 1:nfolds)

{

testi=which(foldi==k)

train=dat[-testi,]

test=dat[testi,]

yrf=randomForest(ptsd~.,data=train,mtry=4,ntree=500,importance=T)

rfpred=predict(yrf,test)

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

rf5out=c(rf5out,accuracy)

}

print(rf5out)

mean(rf5out) #0.7056911

sd(rf5out) #0.03009229

boxplot(rf5out,col='green')

#Boosting LOOCV

n=dim(dat)[1]

set.seed(1)

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

table(foldi)

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

names(dat)

ptsd=ifelse(dat[,13]==TRUE,1,0)

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

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

boost1out=NULL

for(k in 1:n)

{

testi=which(foldi==k)

train=dat[-testi,]

test=dat[testi,]

yboost=gbm(ptsd~.,data=train,distribution='bernoulli',interaction.depth=2,n.trees=1000,shrinkage=0.01)

boostpred=predict(yboost,test,n.trees=1000,type='response')

boostpred=factor(ifelse(boostpred<=0.5,0,1))

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

boost1out=c(boost1out,accuracy)

}

print(boost1out)

mean(boost1out) #0.699187

sd(boost1out) #0.4589848

boxplot(boost1out,col='green')

#Boosting 5-fold CV

nfolds=5

set.seed(1)

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

table(foldi)

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

names(dat)

ptsd=ifelse(dat[,13]==TRUE,1,0)

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

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

boost5out=NULL

for(k in 1:nfolds)

{

testi=which(foldi==k)

train=dat[-testi,]

test=dat[testi,]

yboost=gbm(ptsd~.,data=train,distribution='bernoulli',interaction.depth=2,n.trees=1000,shrinkage=0.01)

boostpred=predict(yboost,test,n.trees=1000,type='response')

boostpred=factor(ifelse(boostpred<=0.5,0,1))

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

boost5out=c(boost5out,accuracy)

}

print(boost5out)

mean(boost5out) #0.7121951

sd(boost5out) #0.03180099

boxplot(boost5out,col='green')

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

names(dat)

dat=data.frame(dat[,13],dat[,22],dat[,26],dat[,22]\*dat[,26])

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

names(dat)=c('ptsd','aa','da','aada')

#Logistic Regression LOOCV

n=dim(dat)[1]

set.seed(1)

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

table(foldi)

lr1out=NULL

for(k in 1:n)

{

testi=which(foldi==k)

train=dat[-testi,]

test=dat[!train,]

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

ptrain=predict(ytrain,test[,2:4],type='response')

ptrain=ifelse(ptrain<=0.5,FALSE,TRUE)

accuracy=(ptrain==test[,1])

lr1out=c(lr1out,accuracy)

}

print(lr1out)

mean(lr1out) #0.6294581

sd(lr1out) #0.4829513

boxplot(lr1out,col='green')

#Logistic Regression 5-fold CV

nfolds=5

set.seed(1)

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

table(foldi)

lr5out=NULL

for(k in 1:nfolds)

{

testi=which(foldi==k)

train=dat[-testi,]

test=dat[!train,]

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

ptrain=predict(ytrain,test[,2:4],type='response')

ptrain=ifelse(ptrain<=0.5,FALSE,TRUE)

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

lr5out=c(lr5out,accuracy)

}

print(lr5out)

mean(lr5out) #0.7190422

sd(lr5out) #0.01809779

boxplot(lr5out,col='green')

#LDA LOOCV

n=dim(dat)[1]

set.seed(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) #0.7089431

sd(lda1out) #0.4546195

boxplot(lda1out,col='green')

#LDA 5-fold CV

nfolds=5

set.seed(1)

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) #0.7089431

sd(lda5out) #0.0194103

boxplot(lda5out,col='green')

#QDA LOOCV

n=dim(dat)[1]

set.seed(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) #0.6813008

sd(qda1out) #0.4663514

boxplot(qda1out,col='green')

#QDA 5-fold CV

nfolds=5

set.seed(1)

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) #0.6813008

sd(qda5out) #0.01563852

boxplot(qda5out,col='green')

#KNN LOOCV

set.seed(1)

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, 1]

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

test\_Y <- dat[test.ID, 1]

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) #0.7056911

sd(OUT.KNN1) #0.4561025

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

#KNN 5-fold CV

n\_fold<-5;

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

set.seed(1)

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, 1]

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

test\_Y <- dat[test.ID, 1]

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) #0.702439

sd(OUT.KNN) #0.03752189

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

boxplot(lr5out,lda5out,qda5out,OUT.KNN,bag5out,rf5out,boost5out,col=c(rainbow(7)),

main='5-fold Cross Validation Comparison',names=c('LR','LDA','QDA','KNN','BAG','RF','BST'),

xlab='Classification Method',ylab='Overall Accuracy')

boxplot(lr1out,lda1out,qda1out,OUT.KNN1,bag1out,rf1out,boost1out,col=c(rainbow(7)),

main='Leave One Out Cross Validation Comparison',names=c('LR','LDA','QDA','KNN','BAG','RF','BST'),

xlab='Classification Method',ylab='Overall Accuracy')

**Appendix 2**: Clinical Scales Descriptions

The clinical scales are:

* *Anxious Arousal* (AA) (symptoms of anxiety, including those associated with posttraumatic hyperarousal);
* *Depression* (D) (depressive symptomatology, both in terms of mood state and depressive cognitive distortions);
* *Anger/Irritability* (AI) (angry or irritable affect, as well as associated angry cognitions and behavior);
* *Intrusive Experiences* (IE) (intrusive symptoms associated with posttraumatic stress, such as flashbacks, nightmares, and intrusive thoughts);
* *Defensive Avoidance* (DA) (posttraumatic avoidance, both cognitive and behavioral);
* *Dissociation* (DIS) (dissociative symptomatology, such as depersonalization, out-of-body experiences, and psychic numbing);
* *Sexual Concerns* (SC) (sexual distress, such as sexual dissatisfaction, sexual dysfunction, and unwanted sexual thoughts or feelings);
* *Dysfunctional Sexual Behavior* (DSB) (sexual behavior that is in some way dysfunctional, either because of its indiscriminate quality, its potential for self-harm, or its inappropriate use to accomplish non-sexual goals);
* *Impaired Self-reference* (ISR) (problems in the "self" domain, such as identity confusion, self-other disturbance, and a relative lack of self-support); and
* *Tension Reduction Behavior* (TRB) (the respondent's tendency to turn to external methods of reducing internal tension or distress, such as self-mutilation, angry outbursts, and suicide threats).

1. [↑](#endnote-ref-2)