```
---
title: "Ex13"
author: "Chathrua Gunasekara"
output: word_document
---
1.a
```{r,echo=FALSE}
library("caret")
library("AppliedPredictiveModeling")
data(hepatic)
barplot(table(injury))


#biological predictors

nzro <- nearZeroVar(bio)
length(nzro)
dim(bio[,-nzro])
filteredbio <- bio[,-nzro]
filteredbio <- filteredbio[,-findCorrelation(cor(filteredbio))]
comboInfo <- findLinearCombos(filteredbio)
comboInfo$remove
# No linear combinations
preProcValues <- preProcess(filteredbio, method = c("center", "scale"))
Transformedbio <- predict(preProcValues, filteredbio)
trainIndex <- createDataPartition(injury, p = .8,list = FALSE,times = 1)

trainX<-Transformedbio[trainIndex,]
trainY<-factor(injury[trainIndex])
testX<-Transformedbio[-trainIndex,]
testY<-factor(injury[-trainIndex])

#--------------------Non-LDA

mdaModel <- train(trainX,y = trainY,method = "mda",metric = "Kappa",tuneGrid = expand.grid(.subclasses=1:5))
mdaModel
mdaPred <-predict(mdaModel,testX)
confusionMatrix(mdaPred,testY)
plsImp <- varImp(mdaModel,scale=FALSE)
```
```{r,echo=FALSE}
#------------------NN
nnetGrid <-expand.grid(.size=1:10,.decay=c(0,0.1,1,2))
maxSize <- max(nnetGrid$.size)
numWts <- 1*(maxSize * (length(trainX) + 1) + maxSize + 1)
set.seed(476)
nnetModel <-
train(x=trainX,y=trainY,method="nnet",metric="Kappa",tuneGrid=nnetGrid,preProc="spatialSign",trace=FALSE,maxit=200,MaxN

nnetModel
confusionMatrix(predict(nnetModel,testX),testY)

```
```{r,echo=FALSE}

#----fda----

fdaModel <- train(trainX,y = trainY,method = "fda",metric = "Kappa")
fdaModel
fdaPred <-predict(fdaModel,testX)
confusionMatrix(fdaPred,testY)
plsImp <- varImp(fdaModel,scale=FALSE)
```
```{r,echo=FALSE}
#----SVM---
library(e1071)
library(kernlab)
library(klaR)
sigmaRange <- sigest(as.matrix(trainX))
svmRGrid <- expand.grid(.sigma = sigmaRange[1],.C = 2^(seq(-4, 4)))

svmRModel <- train(trainX, trainY,method = "svmRadial",metric = "Kappa",tuneGrid = svmRGrid,fit = FALSE)
svmRModel
svmPred<-predict(svmRModel,testX)
confusionMatrix(svmPred,testY)
```
```{r,echo=FALSE}
#-----Knn-----
knnFit <- train(trainX, trainY,method = "knn",metric = "Kappa",tuneGrid = data.frame(.k = c(4*(0:5)+1,20*(1:5)+1,50*
(2:9)+1)))
```
```

```
confusionMatrix(knn(trainX,testX,trainY,k=13),testY)
```
```{r,echo=FALSE}
#----Naive Bayes---
NBFit <- train(trainX, trainY,method = "nb",metric = "Kappa")
NBPred<-predict(NBFit,testX)
confusionMatrix(NBPred,testY)
```

```{r,echo=FALSE}
##########################chem###############


library(AppliedPredictiveModeling)
data(hepatic)

nzro <- nearZeroVar(chem)
length(nzro)
filteredChem <- chem[,-nzro]
filteredChem <- filteredChem[,-findCorrelation(cor(filteredChem))]
comboInfo <- findLinearCombos(filteredChem)
comboInfo$remove
filteredChem <- filteredChem[,-comboInfo$remove]
# No linear combinations
preProcValues <- preProcess(filteredChem, method = c("center", "scale"))
TransformedChem<- predict(preProcValues, filteredChem)
trainIndex <- createDataPartition(injury, p = .8,list = FALSE,times = 1)

trainX<-TransformedChem[trainIndex,]
trainY<-factor(injury[trainIndex])
testX<-TransformedChem[-trainIndex,]
testY<-factor(injury[-trainIndex])

```


```{r,echo=FALSE}
#-------------------------------MDA
train<-cbind(trainY,trainX)
mdaModel <- mda(trainY~.,data=train)
 mdaModel
 mdaPred <-predict(mdaModel,testX)
 confusionMatrix(mdaPred,testY)
 mdaModel <- train(trainX,y = trainY,method = "mda",metric = "Kappa",tuneGrid = expand.grid(.subclasses=1:10))
 mdaModel
 mdaPred <-predict(mdaModel,testX)
 confusionMatrix(mdaPred,testY)
 plsImp <- varImp(mdaModel,scale=FALSE)
 plsImp
```

```{r,echo=FALSE}
#-----------------NN
nnetGrid <-expand.grid(.size=1:10,.decay=c(0,0.1,1,2))
maxSize <- max(nnetGrid$.size)
numWts <- 1*(maxSize * (length(trainX) + 1) + maxSize + 1)
set.seed(476)
nnetModel <-
train(x=trainX,y=trainY,method="nnet",metric="Kappa",tuneGrid=nnetGrid,preProc="spatialSign",trace=FALSE,maxit=200,MaxN

nnetModel
confusionMatrix(predict(nnetModel,testX),testY)

```
```{r,echo=FALSE}

#----fda----

fdaModel <- train(trainX,y = trainY,method = "fda",metric = "Kappa")
fdaModel
fdaPred <-predict(fdaModel,testX)
confusionMatrix(fdaPred,testY)
plsImp <- varImp(fdaModel,scale=FALSE)
```
```{r,echo=FALSE}
#----SVM---
library(e1071)
library(kernlab)
library(klaR)
sigmaRange <- sigest(as.matrix(trainX))
```

```
svmRGrid <- expand.grid(.sigma = sigmaRange[1],.C = 2^(seq(-4, 4)))

svmRModel <- train(trainX, trainY,method = "svmRadial",metric = "Kappa",tuneGrid = svmRGrid,fit = FALSE)
svmRModel
svmPred<-predict(svmRModel,testX)
confusionMatrix(svmPred,testY)
```
```{r,echo=FALSE}
#-----Knn-----
knnFit <- train(trainX, trainY,method = "knn",metric = "Kappa",tuneGrid = data.frame(.k = c(4*(0:5)+1,20*(1:5)+1,50*
(2:9)+1)))

confusionMatrix(knn(trainX,testX,trainY,k=13),testY)
```
```{r,echo=FALSE}
#----Naive Bayes---
NBFit <- train(trainX, trainY,method = "nb",metric = "Kappa")
NBPred<-predict(NBFit,testX)
confusionMatrix(NBPred,testY)

```
```{r,echo=FALSE}

#-----------------combination-----


data(hepatic)
dataset <- cbind(bio,chem)

nzro <- nearZeroVar(dataset)
length(nzro)
filtereddataset <- dataset[,-nzro]
filtereddataset <- filtereddataset[,-findCorrelation(cor(filtereddataset))]
comboInfo <- findLinearCombos(filtereddataset)
comboInfo$remove
filtereddataset <- filtereddataset[,-comboInfo$remove]
# No linear combinations
preProcValues <- preProcess(filtereddataset, method = c("center", "scale"))
Transformed<- predict(preProcValues, filtereddataset)
trainIndex <- createDataPartition(injury, p = .8,list = FALSE,times = 1)

trainX<-Transformed[trainIndex,]
trainY<-factor(injury[trainIndex])
testX<-Transformed[-trainIndex,]
testY<-factor(injury[-trainIndex])

train<-cbind(trainY,trainX)
```
```{r,echo=FALSE}
#---------#MDA---

 mdaModel <- mda(trainY~.,data=train)
 mdaModel
 mdaPred <-predict(mdaModel,testX)
 confusionMatrix(mdaPred,testY)
```
```{r,echo=FALSE}
#-----------------NN
nnetGrid <-expand.grid(.size=1:10,.decay=c(0,0.1,1,2))
maxSize <- max(nnetGrid$.size)
numWts <- 1*(maxSize * (length(trainX) + 1) + maxSize + 1)
set.seed(476)
nnetModel <-
train(x=trainX,y=trainY,method="nnet",metric="Kappa",tuneGrid=nnetGrid,preProc="spatialSign",trace=FALSE,maxit=200,MaxN

nnetModel
confusionMatrix(predict(nnetModel,testX),testY)
```
```{r,echo=FALSE}


#----fda----

fdaModel <- train(trainX,y = trainY,method = "fda",metric = "Kappa")
fdaModel
fdaPred <-predict(fdaModel,testX)
confusionMatrix(fdaPred,testY)
plsImp <- varImp(fdaModel,scale=FALSE)
```
```{r,echo=FALSE}
#----SVM---
```

```
library(e1071)
library(kernlab)
library(klaR)
sigmaRange <- sigest(as.matrix(trainX))
svmRGrid <- expand.grid(.sigma = sigmaRange[1],.C = 2^(seq(-4, 4)))

svmRModel <- train(trainX, trainY,method = "svmRadial",metric = "Kappa",tuneGrid = svmRGrid,fit = FALSE)
svmRModel
svmPred<-predict(svmRModel,testX)
confusionMatrix(svmPred,testY)
```
```{r,echo=FALSE}
#-----Knn-----
knnFit <- train(trainX, trainY,method = "knn",metric = "Kappa",tuneGrid = data.frame(.k = c(4*(0:5)+1,20*(1:5)+1,50*
(2:9)+1)))

confusionMatrix(knn(trainX,testX,trainY,k=13),testY)
```
```{r,echo=FALSE}
#----Naive Bayes---
NBFit <- train(trainX, trainY,method = "nb",metric = "Kappa")
NBPred<-predict(NBFit,testX)
confusionMatrix(NBPred,testY)


```

2.a. Becuase the extream class imbalance the data set should be split using stratified sampling.
```{r,echo=TRUE}


data(oil)
barplot(table(oilType))
library(corrplot)
corrplot(cor(fattyAcids), order = "hclust")
fattyAcids <- fattyAcids[,-findCorrelation(cor(fattyAcids))]

trainIndex <- createDataPartition(oilType, p = 0.75,list = FALSE,times = 1)
trainX <-fattyAcids[trainIndex,]
trainY <- as.factor(oilType[trainIndex])
testX<-fattyAcids[-trainIndex,]
testY <-as.factor(oilType[-trainIndex])
```
```{r,echo=FALSE}
###Non-LDA-----------
library(mda)
mdaModel <- train(trainX,y = trainY,method = "mda",metric = "Kappa",tuneGrid = expand.grid(.subclasses=1:10))
mdaModel
mdaPred <-predict(mdaModel,testX)
confusionMatrix(mdaPred,testY)
plsImp <- varImp(mdaModel,scale=FALSE)
plsImp
```
```{r,echo=FALSE}
#------------------NN
nnetGrid <-expand.grid(.size=1:10,.decay=c(0,0.1,1,2))
maxSize <- max(nnetGrid$.size)
numWts <- 1*(maxSize * (length(trainX) + 1) + maxSize + 1)
set.seed(476)
nnetModel <-
train(x=trainX,y=trainY,method="nnet",metric="Kappa",tuneGrid=nnetGrid,preProc="spatialSign",trace=FALSE,maxit=200,MaxN

nnetModel
confusionMatrix(predict(nnetModel,testX),testY)
```
```{r,echo=FALSE}


#----fda----

fdaModel <- train(trainX,y = trainY,method = "fda",metric = "Kappa")
fdaModel
fdaPred <-predict(fdaModel,testX)
confusionMatrix(fdaPred,testY)
plsImp <- varImp(fdaModel,scale=FALSE)
plsImp
```
```{r,echo=FALSE}
#----SVM---

sigmaRange <- sigest(as.matrix(trainX))
```

```
svmRGrid <- expand.grid(.sigma = sigmaRange[1],.C = 2^(seq(-4, 4)))

svmRModel <- train(trainX, trainY,method = "svmRadial",metric = "Kappa",tuneGrid = svmRGrid,fit = FALSE)
svmRModel
svmPred<-predict(svmRModel,testX)
confusionMatrix(svmPred,testY)
```
```{r,echo=FALSE}
#-----Knn-----
knnFit <- train(trainX, trainY,method = "knn",metric = "Kappa",tuneGrid = data.frame(.k = c(4*(0:5)+1,20*(1:5)+1,50*
(2:9)+1)))

confusionMatrix(knn(trainX,testX,trainY,k=1),testY)
```
```{r,echo=FALSE}
#----Naive Bayes---


classifier<-naiveBayes(trainX, trainY)
confusionMatrix(predict(classifier, testX),testY)

```
```
```