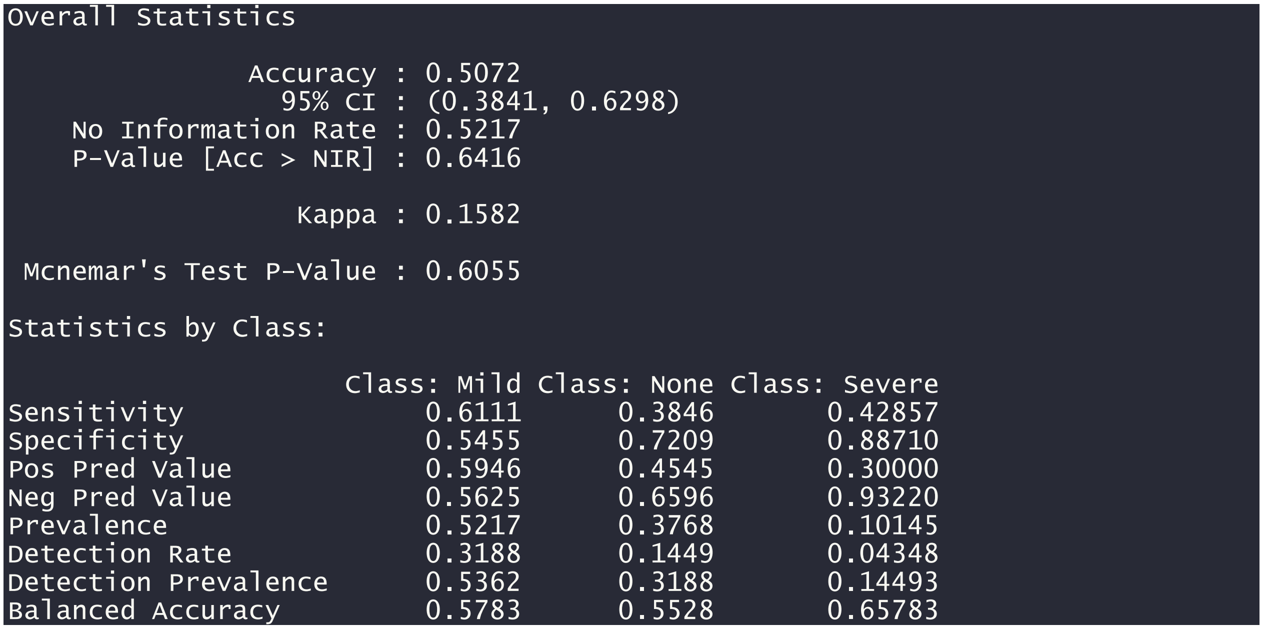
Homework – 6

a) Which model has the best predictive ability for the biological predictors and what is the optimal performance?

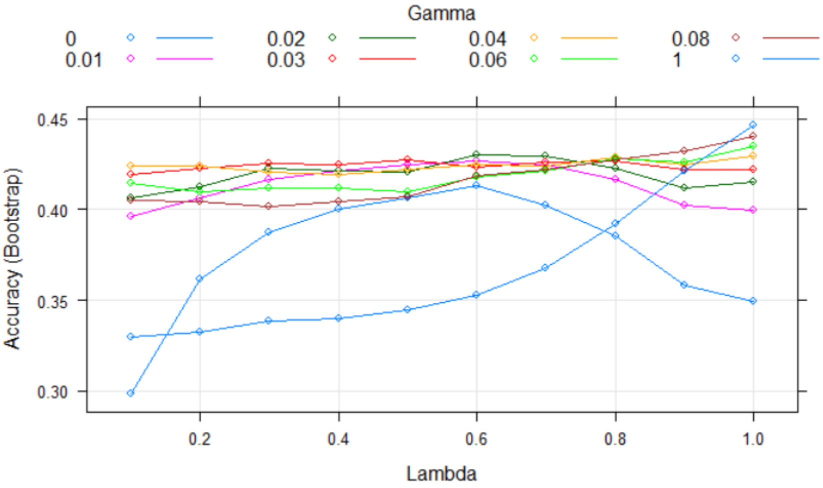
Answer:

MDA:

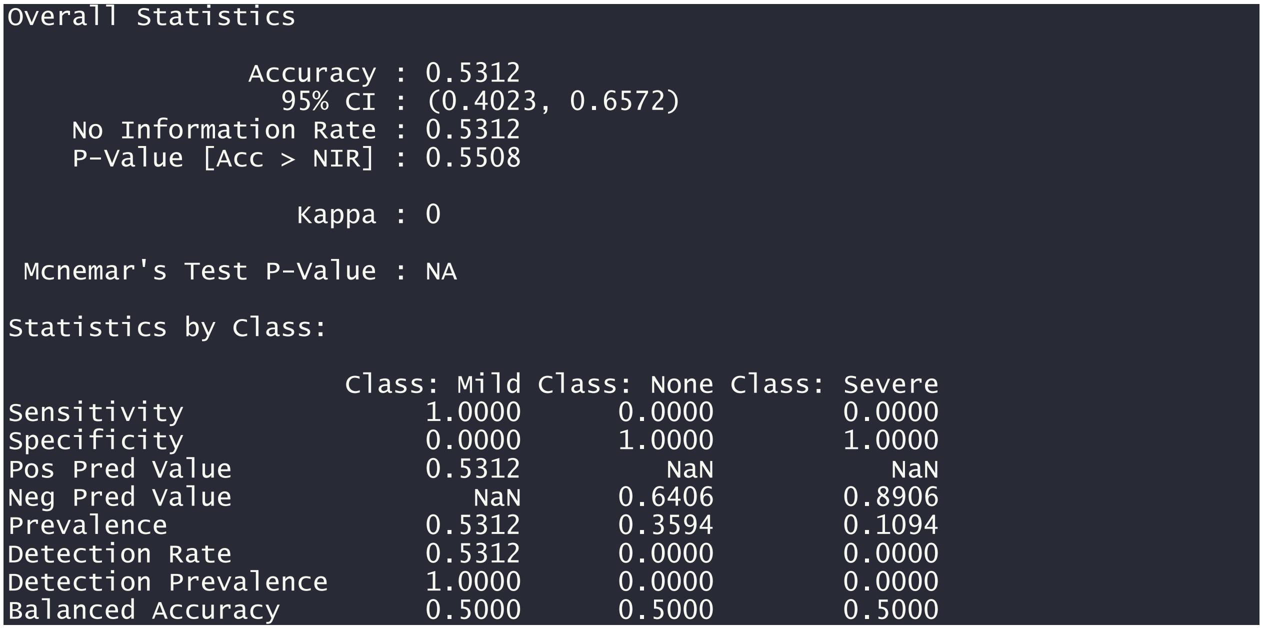


```
Overall Statistics

               Accuracy : 0.5072
                 95% CI : (0.3841, 0.6298)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.6416

                  Kappa : 0.1582

 Mcnemar's Test P-Value : 0.6055

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.6111      0.3846       0.42857
Specificity               0.5455      0.7209       0.88710
Pos Pred Value            0.5946      0.4545       0.30000
Neg Pred Value            0.5625      0.6596       0.93220
Prevalence                0.5217      0.3768       0.10145
Detection Rate            0.3188      0.1449       0.04348
Detection Prevalence      0.5362      0.3188       0.14493
Balanced Accuracy         0.5783      0.5528       0.65783
```

Plot of MDA:

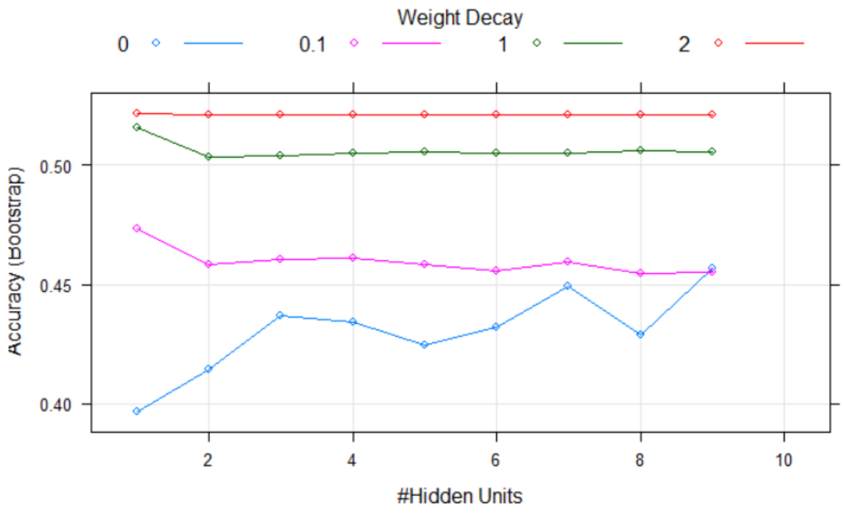


Neural Network:

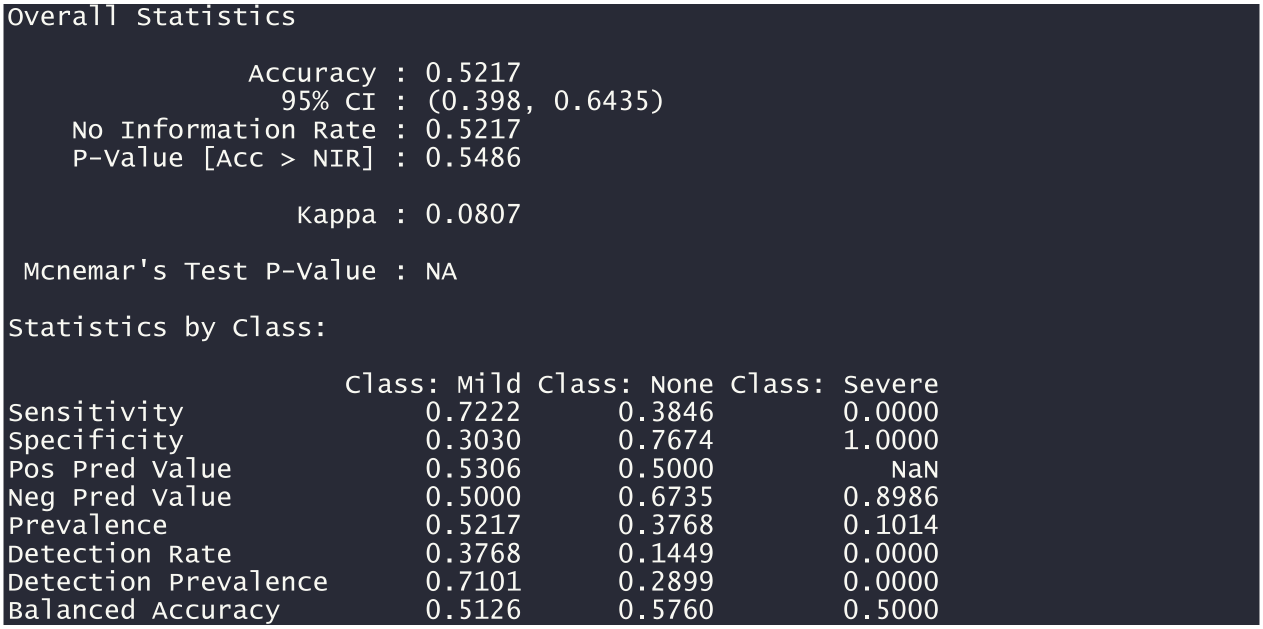


```
Overall Statistics

               Accuracy : 0.5312
                 95% CI : (0.4023, 0.6572)
    No Information Rate : 0.5312
    P-Value [Acc > NIR] : 0.5508

                  Kappa : 0

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               1.0000      0.0000        0.0000
Specificity               0.0000      1.0000        1.0000
Pos Pred Value            0.5312         NaN           NaN
Neg Pred Value               NaN      0.6406        0.8906
Prevalence                0.5312      0.3594        0.1094
Detection Rate            0.5312      0.0000        0.0000
Detection Prevalence      1.0000      0.0000        0.0000
Balanced Accuracy         0.5000      0.5000        0.5000
```

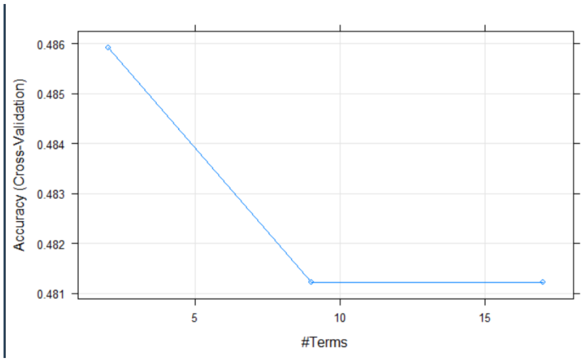
Plot of Neural Network:



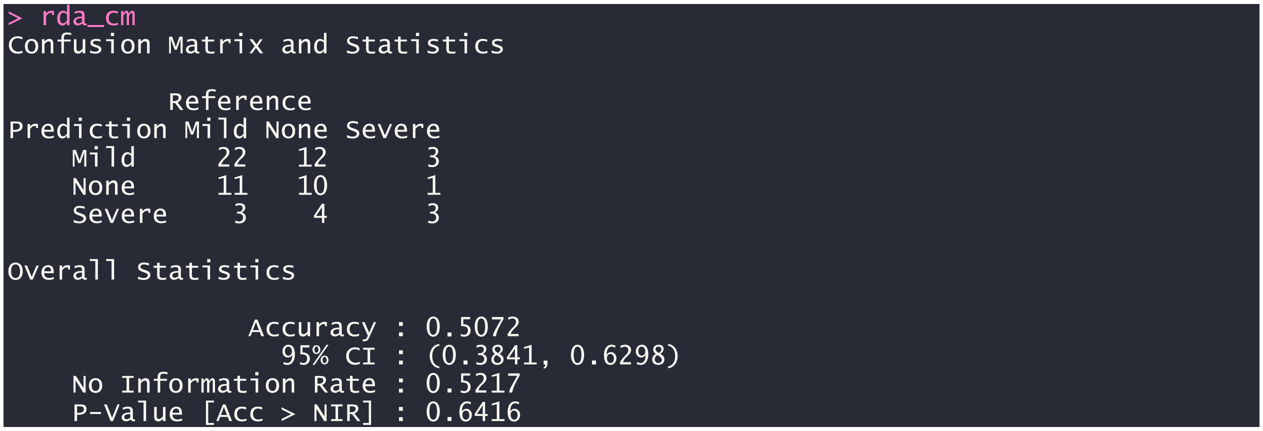
FDA:



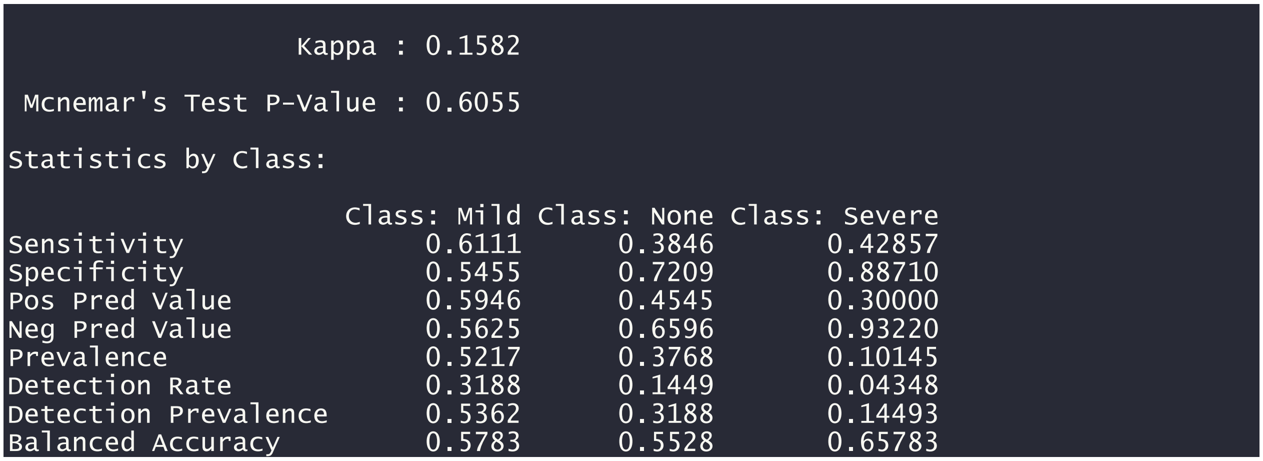
```
Overall Statistics

               Accuracy : 0.5217
                 95% CI : (0.398, 0.6435)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.5486

                  Kappa : 0.0807

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.7222      0.3846        0.0000
Specificity               0.3030      0.7674        1.0000
Pos Pred Value            0.5306      0.5000           NaN
Neg Pred Value            0.5000      0.6735        0.8986
Prevalence                0.5217      0.3768        0.1014
Detection Rate            0.3768      0.1449        0.0000
Detection Prevalence      0.7101      0.2899        0.0000
Balanced Accuracy         0.5126      0.5760        0.5000
```



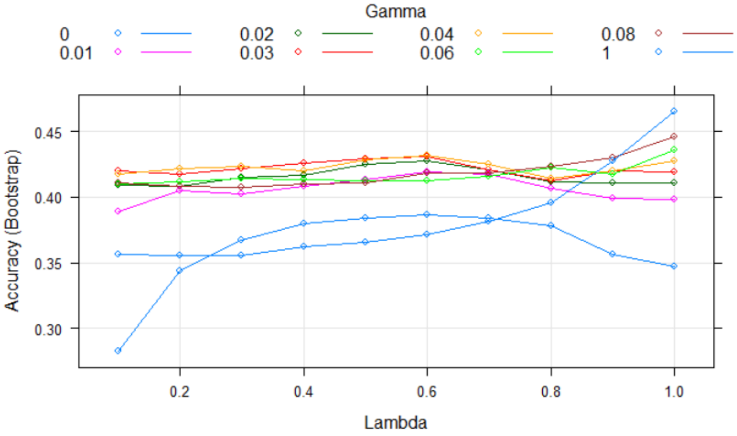
RDA:



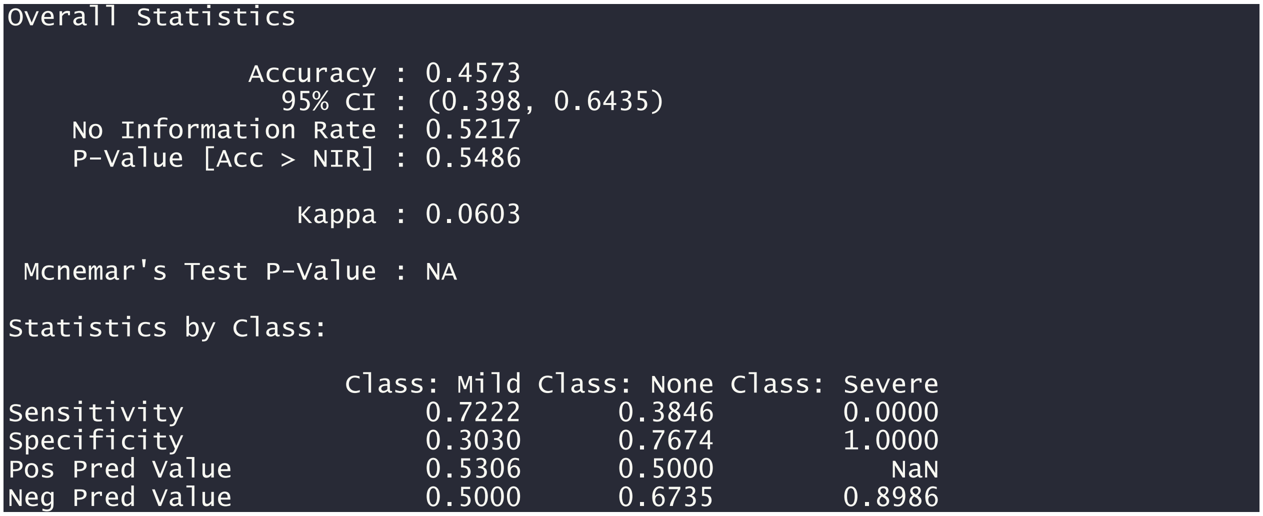
```
> rda_cm
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     22   12      3
    None     11   10      1
    Severe    3    4      3

Overall Statistics

               Accuracy : 0.5072
                 95% CI : (0.3841, 0.6298)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.6416
```



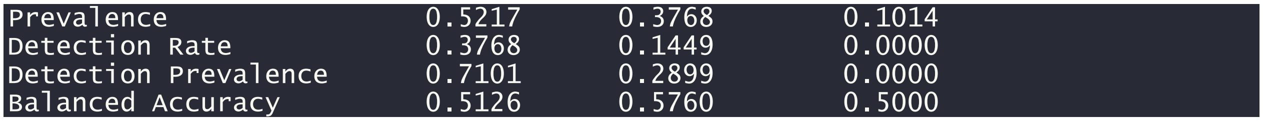
```
                   Kappa : 0.1582

 Mcnemar's Test P-Value : 0.6055

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.6111      0.3846       0.42857
Specificity               0.5455      0.7209       0.88710
Pos Pred Value            0.5946      0.4545       0.30000
Neg Pred Value            0.5625      0.6596       0.93220
Prevalence                0.5217      0.3768       0.10145
Detection Rate            0.3188      0.1449       0.04348
Detection Prevalence      0.5362      0.3188       0.14493
Balanced Accuracy         0.5783      0.5528       0.65783
```



SVM:



```
Overall Statistics

               Accuracy : 0.4573
                 95% CI : (0.398, 0.6435)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.5486

                  Kappa : 0.0603

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.7222      0.3846        0.0000
Specificity               0.3030      0.7674        1.0000
Pos Pred Value            0.5306      0.5000           NaN
Neg Pred Value            0.5000      0.6735        0.8986
```



```
Prevalence                 0.5217      0.3768       0.1014
Detection Rate             0.3768      0.1449       0.0000
Detection Prevalence       0.7101      0.2899       0.0000
Balanced Accuracy          0.5126      0.5760       0.5000
```

Model	**Accuracy**
MDA	0.5072
FDA	0.5217
Neural Network	0.5312
SVM	0.4573
RDA	0.5072

From the above given models, the best model with the highest accuracy is Neural Network with the highest accuracy of 0.5312.

b) Does the nonlinear structure of these models help to improve the classification performance?
Previously, linear models had the below accuracy:

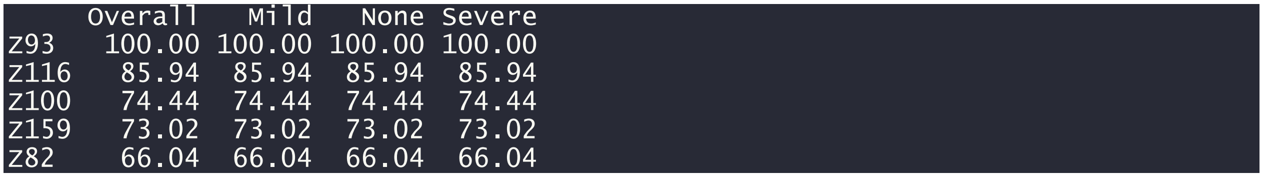
Model	**Accuracy**
Logistic Regression	0.5072
Linear Discriminant Analysis	0.5072
Partial Least Square Discriminant Analysis	0.5362
Penalized Model	0.5652
Nearest Shrunken Centroids	0.5217

The best accuracy is of Penalized Model of 0.5652. However, in Non Linear Models, we observe that the best accuracy if of Neural Network, 0.5312. Hence, we can say that Linear Model performs better and Non linear Model does not improve the classification performance.

c) For the optimal models for the biological predictors, what are the top five important predictors?

The Best model that we see is Neural Network.

Top 5 important predictors are:



```
     Overall   Mild   None Severe
Z93   100.00 100.00 100.00 100.00
Z116   85.94  85.94  85.94  85.94
Z100   74.44  74.44  74.44  74.44
Z159   73.02  73.02  73.02  73.02
Z82    66.04  66.04  66.04  66.04
```

R CODE:

```
install.packages(c("glmnet", "pamr", "rms", "sparseLDA", "subselect", "kernlab"))

#12.1

library(caret)
library(AppliedPredictiveModeling)
data(hepatic)
library(MASS)
set.seed(1)
#barplot(table(injury), main="Imbalanced Class Distribution")

#PreProcess the data
#---------------------------
set.seed(1)
trainingRows = createDataPartition(injury, p = .75, list= FALSE)
trainBio <- bio[ trainingRows, ]
testBio <- bio[-trainingRows, ]
trainInjury <- injury[trainingRows]
```

```
testInjury <- injury[-trainingRows]

pp <- preProcess(trainBio, method = c("BoxCox","center","scale"))
trainBio <- predict(pp, trainBio)
testBio <- predict(pp, testBio)

nz <- nearZeroVar(trainBio)
trainBio <- trainBio[-nz]
testBio <- testBio[-nz]

hc <- cor(trainBio)
hc_p <- findCorrelation(hc)

trainBio <- trainBio[,-hc_p]
testBio <- testBio[,-hc_p]
#------------------------

#Model building

###MDA####

set.seed(1)

ctrl <- trainControl(summaryFunction = defaultSummary)

mdaFit <- train(x = trainBio,
        y = trainInjury,
        method = "mda",
        metric = "Accuracy",
        tuneGrid = expand.grid(.subclasses = 1:4),
         trControl = ctrl)
mdaFit

summary(mdaFit)

plot(mdaFit)

predictionmda<-predict(mdaFit,testBio)

confusionMatrix(data = predictionmda, reference = testInjury)

##RDA

set.seed(1)

library(klaR)

rdaGrid<-expand.grid(.gamma=c(0,.01,.02,.03,.04,.06,.08,1),
            .lambda=seq(.1,1,length=10))

rda_clf<-train(x = trainBio,y = trainInjury,method="rda",
        tuneGrid=rdaGrid,preProcess=c("center","scale"),
        metric="Accuracy",
        trControl = ctrl)
rda_clf
rda_pred<-predict(rda_clf,testBio)
rda_cm<-confusionMatrix(data=rda_pred,reference=testInjury)
rda_cm

summary(rda_clf)
```

```
plot(rda_clf)


## QDA

qda_fit<-train(trainBio,trainInjury,method="qda",
          preProcess=c("center","scale"),metric="Accuracy",
          trControl = ctrl)

qda_pred<-predict(qda_fit,testBio)
qda_cm<-confusionMatrix(data=qda_pred,reference=testInjury)
qda_cm


############### Neural Networks ############


library(nnet)

set.seed(1)

nnetGrid <- expand.grid(.size = 1:10, .decay = c(0, .1, 1, 2))
maxSize <- max(nnetGrid$.size)
numWts <- (maxSize * (98 + 1) + (maxSize+1)*2)

nnetFit <- train(x = trainBio,
           y = trainInjury,
           method = "nnet",
           metric = "Accuracy",
          preProc = c("center", "scale", "spatialSign"),
          tuneGrid = nnetGrid,
          trace = FALSE,
          maxit = 2000,
          MaxNWts = numWts,
          trControl = ctrl)
nnetFit

nn_pred<-predict(nnetFit,testBio)
nn_cm<-confusionMatrix(data=nn_pred,reference=testInjury)
nn_cm

plot(nnetFit)

important=varImp(nnetFit)
plot(important, top = 5, scales = list(y = list(cex = .95)))


########## Flexible Discriminant Analysis ############

library(earth)

fda_grid<-expand.grid(.degree=1:4,.nprune=2:38)

fda_clf<-train(trainBio,trainInjury,
         method="fda",preProc=c("BoxCox","center","scale"),
         metric="Accuracy",
         trControl = trainControl(method = "cv", number = 3))
fda_clf
plot(fda_clf)
```

```
fda_pred<-predict(fda_clf,testBio)
fda_cm<-confusionMatrix(data=fda_pred,reference=testInjury)
fda_cm




############# Support Vector Machines #########
set.seed(1)
library(kernlab)

sigmaRangeReduced <- sigest(as.matrix(trainBio))

## Given a range of values for the "sigma" inverse width parameter
## in the Gaussian Radial Basis kernel for use with SVM.
## The estimation is based on the data to be used.

svmRGridReduced <- expand.grid(.sigma = sigmaRangeReduced[1],
                               .C = 2^(seq(-4, 6)))

svm_clf<-train(x=trainBio,y=trainInjury,
               method="svmRadial",tuneGrid=svmRGridReduced,
               preProc=c("center","scale"),
               fit=FALSE,
               metric="Accuracy",
               trControl = trainControl(method = "cv", number = 3))
svm_clf
svm_clf$finalModel
plot(svm_clf)
svm_pred<-predict(svm_clf,testBio,type = "raw")
```

```
svm_cm<-confusionMatrix(data=svm_pred,reference=testInjury)
svm_cm

plot(svm_clf)
```