Week 6 Assignment

Problem 12.1a Given the classification imbalance in hepatic injury status, describe how you would create a training and testing set

- I would use a stratified random sampling technique because that would be more representative of the proportions in the overall response than to use a simple random sampling technique.

Problem 12.1b Which classification statistic would you choose to optimize for this exercise and why?

- I believe Kappa is a good statistic to use to judge the models here. It is more informative than simply using the accuracy rate, as the accuracy rate simply looks at all classes as equal. The highest Kappa value will be considered the best model. Also since there are more than two outcomes, ROC becomes a little less interpretable, which is why we may choose not to use this metric.

Problem 12.1c Pre-process the data, split the data into a training and a testing set, and build models described in this chapter for the biological predictors. Using each model to predict on the testing set, which model has the best predictive ability for the biological predictors and what is the optimal performance?

- Linear Discriminant Analysis

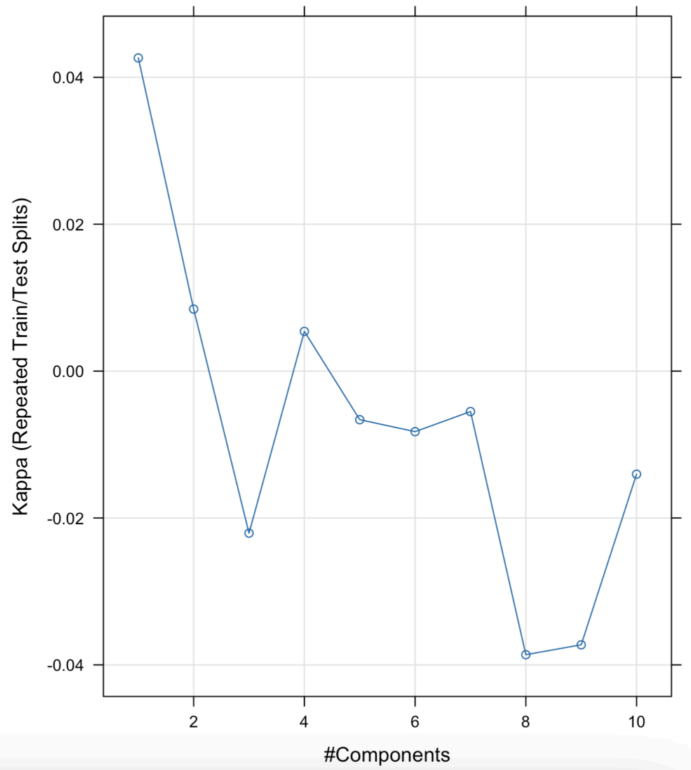
```
Linear Discriminant Analysis

225 samples
 91 predictor
  3 classes: 'Mild', 'None', 'Severe'

Pre-processing: centered (91), scaled (91)
Resampling: Repeated Train/Test Splits Estimated (10 reps, 75%)
Summary of sample sizes: 169, 169, 169, 169, 169, 169, ...
Resampling results:

  Accuracy   Kappa
  0.4446429  0.08415325
```

```
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     16    9      3
    None     11   10      1
    Severe    2    2      2

Overall Statistics

               Accuracy : 0.5
                 95% CI : (0.3634, 0.6366)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.6562

                  Kappa : 0.1413

 Mcnemar's Test P-Value : 0.8653

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.5517      0.4762       0.33333
Specificity               0.5556      0.6571       0.92000
Pos Pred Value            0.5714      0.4545       0.33333
Neg Pred Value            0.5357      0.6765       0.92000
Prevalence                0.5179      0.3750       0.10714
Detection Rate            0.2857      0.1786       0.03571
Detection Prevalence      0.5000      0.3929       0.10714
Balanced Accuracy         0.5536      0.5667       0.62667
```

-

- Partial Least Squares Discriminant Analysis

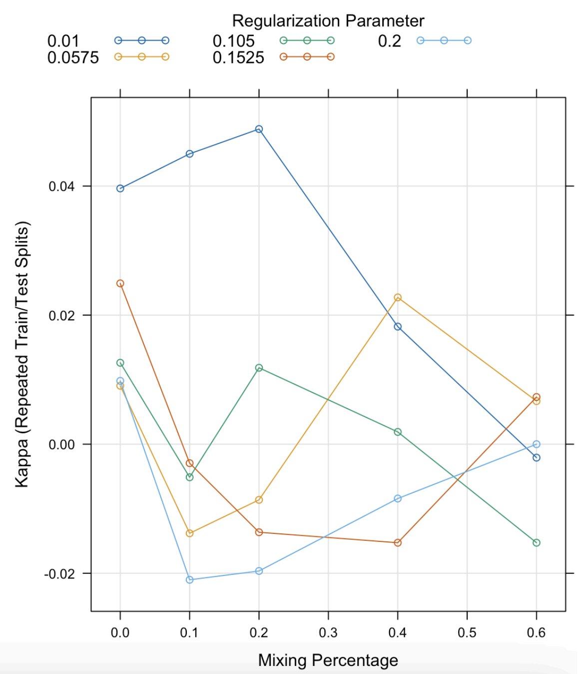
```
Partial Least Squares

225 samples
 91 predictor
  3 classes: 'Mild', 'None', 'Severe'

Pre-processing: centered (91), scaled (91)
Resampling: Repeated Train/Test Splits Estimated (10 reps, 75%)
Summary of sample sizes: 169, 169, 169, 169, 169, 169, ...
Resampling results across tuning parameters:

  ncomp  Accuracy   Kappa
   1     0.5035714   0.042641245
   2     0.4839286   0.008444541
   3     0.4589286  -0.022063946
   4     0.4696429   0.005403703
   5     0.4642857  -0.006616683
   6     0.4607143  -0.008240888
   7     0.4589286  -0.005506676
   8     0.4357143  -0.038615293
   9     0.4339286  -0.037265952
  10     0.4464286  -0.014036123

Kappa was used to select the optimal model using the largest value.
The final value used for the model was ncomp = 1.
```



```
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     26   17      6
    None      3    4      0
    Severe    0    0      0

Overall Statistics

               Accuracy : 0.5357
                 95% CI : (0.3974, 0.6701)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.4475

                  Kappa : 0.0714

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.8966     0.19048        0.0000
Specificity               0.1481     0.91429        1.0000
Pos Pred Value            0.5306     0.57143           NaN
Neg Pred Value            0.5714     0.65306        0.8929
Prevalence                0.5179     0.37500        0.1071
Detection Rate            0.4643     0.07143        0.0000
Detection Prevalence      0.8750     0.12500        0.0000
Balanced Accuracy         0.5223     0.55238        0.5000
```

- Penalized Model

```
glmnet

225 samples
 91 predictor
  3 classes: 'Mild', 'None', 'Severe'

Pre-processing: centered (91), scaled (91)
Resampling: Repeated Train/Test Splits Estimated (10 reps, 75%)
Summary of sample sizes: 169, 169, 169, 169, 169, 169, ...
Resampling results across tuning parameters:

  alpha  lambda  Accuracy   Kappa
  0.0    0.0100  0.4517857   0.039628481
  0.0    0.0575  0.4589286   0.009050681
  0.0    0.1050  0.4714286   0.012617514
  0.0    0.1525  0.4821429   0.024918133
  0.0    0.2000  0.4785714   0.009809834
  0.1    0.0100  0.4589286   0.044996563
  0.1    0.0575  0.4535714  -0.013809616
  0.1    0.1050  0.4714286  -0.005126396
  0.1    0.1525  0.4803571  -0.002940312
  0.1    0.2000  0.4767857  -0.021012823
  0.2    0.0100  0.4642857   0.048835536
  0.2    0.0575  0.4642857  -0.008623359
  0.2    0.1050  0.4875000   0.011839575
  0.2    0.1525  0.4839286  -0.013643231
  0.2    0.2000  0.4875000  -0.019646290
  0.4    0.0100  0.4482143   0.018221615
  0.4    0.0575  0.4892857   0.022744838
  0.4    0.1050  0.4946429   0.001897383
  0.4    0.1525  0.4982143  -0.015268890
  0.4    0.2000  0.5107143  -0.008420766
  0.6    0.0100  0.4428571  -0.002084610
  0.6    0.0575  0.4892857   0.006673787
  0.6    0.1050  0.4982143  -0.015268890
  0.6    0.1525  0.5196429   0.007308085
  0.6    0.2000  0.5178571   0.000000000

Kappa was used to select the optimal model using the largest value.
The final values used for the model were alpha = 0.2 and lambda = 0.01.
```



```
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     18    9      4
    None     11   11      1
    Severe    0    1      1

Overall Statistics

               Accuracy : 0.5357
                 95% CI : (0.3974, 0.6701)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.4475

                  Kappa : 0.1642

 Mcnemar's Test P-Value : 0.2407

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.6207      0.5238       0.16667
Specificity               0.5185      0.6571       0.98000
Pos Pred Value            0.5806      0.4783       0.50000
Neg Pred Value            0.5600      0.6970       0.90741
Prevalence                0.5179      0.3750       0.10714
Detection Rate            0.3214      0.1964       0.01786
Detection Prevalence      0.5536      0.4107       0.03571
Balanced Accuracy         0.5696      0.5905       0.57333
```

Model	Tuning parameter	Kappa
Linear Discriminant Analysis	N/A	0.1413
Partial Least Squares Discriminant Analysis	Components = 1	0.0714
Penalized	Alpha = 0.2, Lambda = 0.01	0.1642

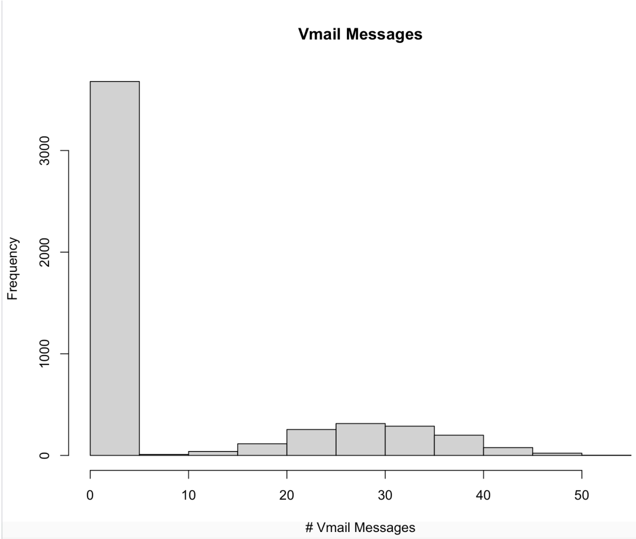
- The penalized model with an alpha of 0.2 and a lambda of 0.01 was shown to be the highest performing model.

Problem 12.1d For the optimal model for the biological predictors, what are the top five important predictors?

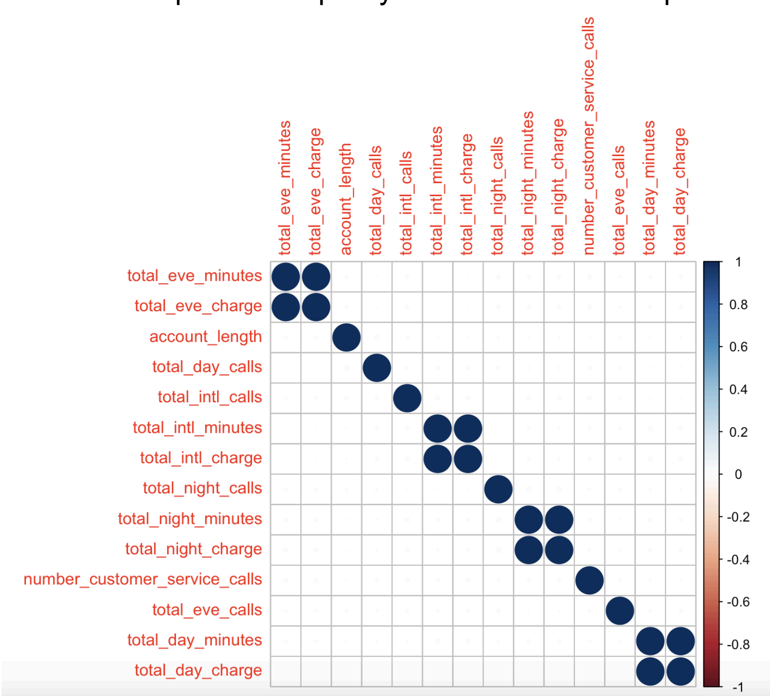
- The top 5 most important biological predictors in the highest performing penalized model model was shown to be Z167, Z38, Z42, Z100, and Z71.

```
glmnet variable importance

  variables are sorted by maximum importance across the classes
  only 20 most important variables shown (out of 91)

          Mild  None Severe
Z167 100.00000  2.29 66.425
Z38   98.71615 61.65  5.785
Z42    0.00000 67.04 96.107
Z100   7.90491 43.57 82.759
Z71   39.34312 74.64  4.015
Z116  70.36695 35.27  3.810
Z36    0.00000 70.25 42.607
Z121  14.26399 66.12 20.576
Z59   28.98512  3.88 64.150
Z82   62.56418 22.62  8.657
Z145   3.12327 60.47 26.064
Z149  53.48731 60.28  0.000
Z128   4.75792 56.86 20.818
Z126  55.88285 12.47 12.124
Z15    0.05943 55.35 24.262
Z147  54.00999  0.00 28.040
Z132  51.60417 27.95  0.000
Z7     3.30901 16.71 51.304
Z176  40.24023 51.15  0.000
Z44    0.18084 50.58 19.889
```

Problem 12.3a Explore the data by visualizing the relationship between the predictors and the outcome. Are there important features of the predictor data themselves, such as between-predictor correlations or degenerate distributions?



- Only the number of voicemail messages was shown to be a low variance predictor, the histogram shows that the frequency is concentrated around zero. This will not add much to the predictive quality of the model so this predictor will be removed



- Some predictors do appear to be highly correlated in this dataset. The Correlation plot shows that:
    - Total evening minutes is associated with total eve charge
    - Total international minutes is associated with total international charge
    - Total night minutes is associated with total night charge
    - Total day minutes is associated with total day charge

Problem 12.3b What criteria should be used to evaluate the effectiveness of the models?

- The area under the ROC curve would be a good statistic to use to judge the models. The area under the ROC curve will maximize sensitivity and specificity, both of which we care about here.

Problem 12.3c Split the data into training set and test set using random splitting (80% and 20%). Fit models covered in this chapter to the training set and tune them via resampling. Which model has the best performance?

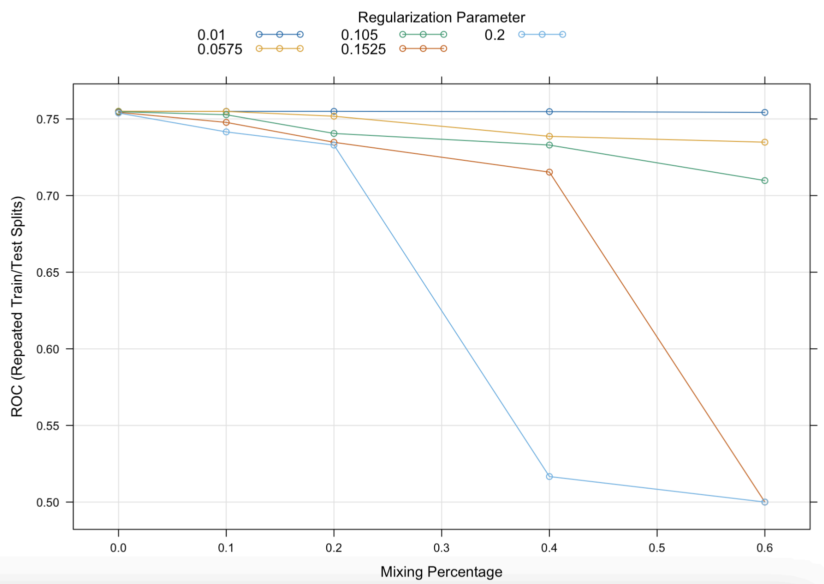
- Logistic Regression model

```
Generalized Linear Model

4001 samples
  14 predictor
   2 classes: 'yes', 'no'

Pre-processing: centered (11), scaled (11), ignore (3)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 3002, 3002, 3002, 3002, 3002, 3002, ...
Resampling results:

  ROC        Sens       Spec
  0.7569928  0.1582979  0.9797203
```

```
Confusion Matrix and Statistics

          Reference
Prediction yes  no
       yes  18  19
       no  123 839

               Accuracy : 0.8579
                 95% CI : (0.8346, 0.8789)
    No Information Rate : 0.8589
    P-Value [Acc > NIR] : 0.5584

                  Kappa : 0.1525

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.12766
            Specificity : 0.97786
         Pos Pred Value : 0.48649
         Neg Pred Value : 0.87214
             Prevalence : 0.14114
         Detection Rate : 0.01802
   Detection Prevalence : 0.03704
      Balanced Accuracy : 0.55276

       'Positive' Class : yes
```

- Penalized model

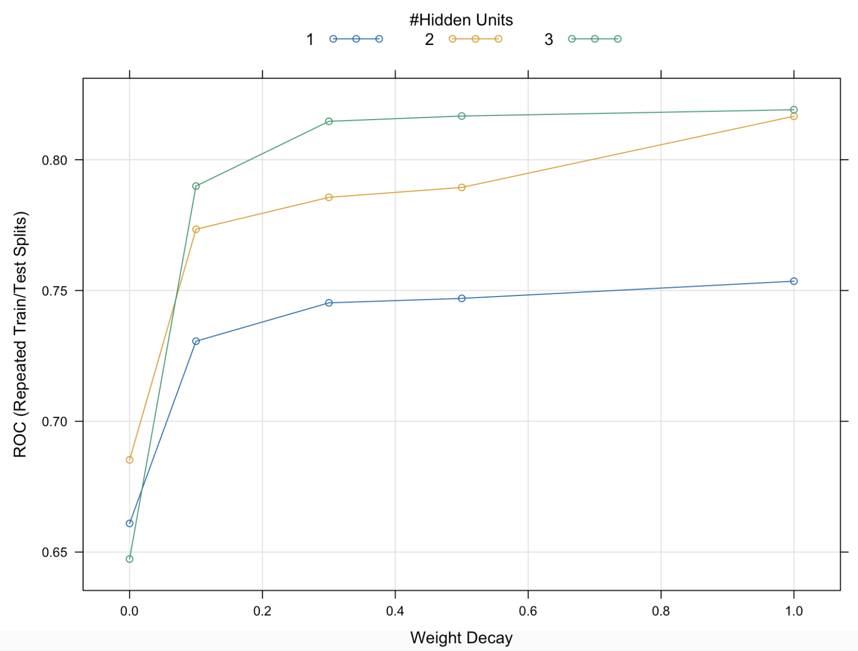
```
glmnet

4001 samples
  14 predictor
   2 classes: 'yes', 'no'

Pre-processing: centered (11), scaled (11), ignore (3)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 3002, 3002, 3002, 3002, 3002, 3002, ...
Resampling results across tuning parameters:

  alpha  lambda  ROC        Sens          Spec
  0.0    0.0100  0.7548930  0.0578723404  0.9897902
  0.0    0.0575  0.7550470  0.0161702128  0.9984149
  0.0    0.1050  0.7547534  0.0056737589  1.0000000
  0.0    0.1525  0.7543266  0.0039716312  1.0000000
  0.0    0.2000  0.7538994  0.0019858156  1.0000000
  0.1    0.0100  0.7549515  0.0550354610  0.9904429
  0.1    0.0575  0.7549349  0.0121985816  0.9991608
  0.1    0.1050  0.7527726  0.0045390071  1.0000000
  0.1    0.1525  0.7477664  0.0000000000  1.0000000
  0.1    0.2000  0.7415533  0.0000000000  1.0000000
  0.2    0.0100  0.7549617  0.0530496454  0.9910023
  0.2    0.0575  0.7517992  0.0079432624  0.9994872
  0.2    0.1050  0.7405512  0.0002836879  1.0000000
  0.2    0.1525  0.7347817  0.0000000000  1.0000000
  0.2    0.2000  0.7329399  0.0000000000  1.0000000
  0.4    0.0100  0.7547868  0.0470921986  0.9924942
  0.4    0.0575  0.7386677  0.0056737589  0.9998601
  0.4    0.1050  0.7329749  0.0000000000  1.0000000
  0.4    0.1525  0.7152967  0.0000000000  1.0000000
  0.4    0.2000  0.5166475  0.0000000000  1.0000000
  0.6    0.0100  0.7542859  0.0417021277  0.9932401
  0.6    0.0575  0.7348348  0.0008510638  1.0000000
  0.6    0.1050  0.7098031  0.0000000000  1.0000000
  0.6    0.1525  0.5000000  0.0000000000  1.0000000
  0.6    0.2000  0.5000000  0.0000000000  1.0000000

ROC was used to select the optimal model using the largest value.
The final values used for the model were alpha = 0 and lambda = 0.0575.
```



```
Confusion Matrix and Statistics

          Reference
Prediction yes  no
       yes   2   2
       no  139 856

               Accuracy : 0.8589
                 95% CI : (0.8357, 0.8799)
    No Information Rate : 0.8589
    P-Value [Acc > NIR] : 0.5224

                  Kappa : 0.02

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.014184
            Specificity : 0.997669
         Pos Pred Value : 0.500000
         Neg Pred Value : 0.860302
             Prevalence : 0.141141
         Detection Rate : 0.002002
   Detection Prevalence : 0.004004
      Balanced Accuracy : 0.505927

       'Positive' Class : yes
```

- Neural Net model

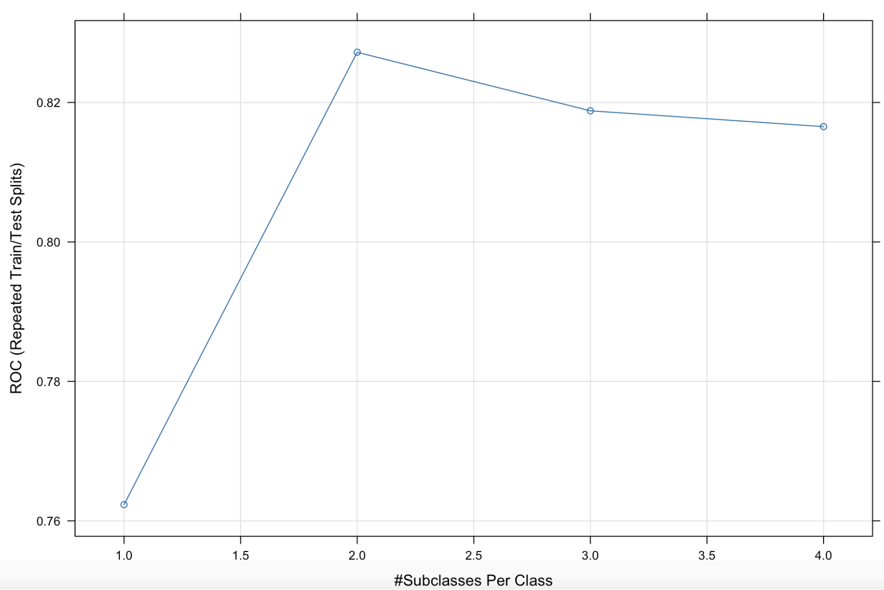
```
Neural Network

4001 samples
  14 predictor
   2 classes: 'yes', 'no'

Pre-processing: centered (11), scaled (11), spatial sign transformation (11), ignore (3)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 3002, 3002, 3002, 3002, 3002, 3002, ...
Resampling results across tuning parameters:

  size  decay  ROC        Sens        Spec
  1     0.0    0.6609511  0.02666667  0.9977622
  1     0.1    0.7306211  0.14212766  0.9823310
  1     0.3    0.7452853  0.09900709  0.9846620
  1     0.5    0.7469900  0.08482270  0.9875991
  1     1.0    0.7535754  0.05134752  0.9937995
  2     0.0    0.6852687  0.11659574  0.9740793
  2     0.1    0.7734282  0.30269504  0.9671795
  2     0.3    0.7856192  0.29106383  0.9733800
  2     0.5    0.7893997  0.25248227  0.9806061
  2     1.0    0.8166179  0.24822695  0.9888578
  3     0.0    0.6473379  0.19375887  0.9517949
  3     0.1    0.7899284  0.37049645  0.9627972
  3     0.3    0.8146810  0.37787234  0.9717949
  3     0.5    0.8166982  0.34609929  0.9773427
  3     1.0    0.8191139  0.28028369  0.9867133

ROC was used to select the optimal model using the largest value.
The final values used for the model were size = 3 and decay = 1.
```



```
Confusion Matrix and Statistics

          Reference
Prediction yes  no
       yes  44  10
       no   97 848

               Accuracy : 0.8929
                 95% CI : (0.872, 0.9114)
    No Information Rate : 0.8589
    P-Value [Acc > NIR] : 0.0008284

                  Kappa : 0.4048

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.31206
            Specificity : 0.98834
         Pos Pred Value : 0.81481
         Neg Pred Value : 0.89735
             Prevalence : 0.14114
         Detection Rate : 0.04404
   Detection Prevalence : 0.05405
      Balanced Accuracy : 0.65020

       'Positive' Class : yes
```

- Mixed Discriminant Analysis

```
Mixture Discriminant Analysis

4001 samples
  14 predictor
   2 classes: 'yes', 'no'

No pre-processing
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 3002, 3002, 3002, 3002, 3002, 3002, ...
Resampling results across tuning parameters:

  subclasses  ROC        Sens       Spec
  1           0.7623251  0.1520567  0.9778555
  2           0.8272049  0.4252482  0.9765967
  3           0.8188054  0.4184397  0.9686247
  4           0.8165392  0.4592908  0.9569697

ROC was used to select the optimal model using the largest value.
The final value used for the model was subclasses = 2.
```



```
Confusion Matrix and Statistics

          Reference
Prediction yes  no
       yes  43  16
       no   98 842

               Accuracy : 0.8859
                 95% CI : (0.8645, 0.9049)
    No Information Rate : 0.8589
    P-Value [Acc > NIR] : 0.006838

                  Kappa : 0.3782

 Mcnemar's Test P-Value : 3.291e-14

            Sensitivity : 0.30496
            Specificity : 0.98135
         Pos Pred Value : 0.72881
         Neg Pred Value : 0.89574
             Prevalence : 0.14114
         Detection Rate : 0.04304
   Detection Prevalence : 0.05906
      Balanced Accuracy : 0.64316

       'Positive' Class : yes
```

Model	Tuning Parameters	ROC
Logistic Regression	N/A	0.736
Penalized model	Alpha = 0, lambda = 0.0575	0.7647
Neural Net	Size = 3, decay = 1	0.7984
Mixed Discriminant Analysis	Subclasses = 2	0.7913

- The Neural Net model with size = 3 and decay = 1 performed the best of any of the models tested here as it had the highest area under the ROC curve.

R Code

```
#Assignment Week 6
#Cody Rorick

library(caret)
library(AppliedPredictiveModeling)
data(hepatic)

#Problem 12.1c Pre-process the data, split the data into a training and a testing set, and build
models described
#in this chapter for the biological predictors. Using each model to predict on the testing set,
which
#model has the best predictive ability for the biological predictors and what is the optimal
#performance?
low.variability.columns <- nearZeroVar(bio)
bio <- bio[,-low.variability.columns]
correlations <- cor(bio)
library(corrplot)
corrplot(correlations, order = "hclust")
highly.correlated.columns <- findCorrelation(correlations, cutoff = .85)
bio <- bio[,-highly.correlated.columns]
set.seed(100)
trainRows <- createDataPartition(injury, p = .80, list = FALSE)
trainResp <- injury[trainRows]
trainPred <- bio[trainRows,]
testResp <- injury[-trainRows]
testPred <- bio[-trainRows,]

### Linear Discriminant Analysis
ctrl <- trainControl(method = "LGOCV", number = 10, classProbs = TRUE, savePredictions =
TRUE)
LDAFull <- train(trainPred, y = trainResp, method = "lda", preProc = c("center","scale"), metric =
"Kappa", trControl = ctrl)
LDAFull
confusionMatrix(data = predict(LDAFull, testPred), reference = testResp) #average over
10*.25*1000 observations

#### Partial Least Squares Discriminant Analysis
set.seed(100)
ctrl <- trainControl(method = "LGOCV", number = 10, classProbs = TRUE, savePredictions =
TRUE)
plsFit <- train(x = trainPred, y = trainResp, method = "pls",tuneGrid = expand.grid(.ncomp =
1:10), preProc = c("center","scale"), metric = "Kappa", trControl = ctrl)
plsFit
```

```
plot(plsFit)
confusionMatrix(data = predict(plsFit, testPred), reference = testResp)

###Penalized glmnet model
library(glmnet)
ctrl <- trainControl(method = "LGOCV", number = 10, classProbs = TRUE,savePredictions =
TRUE)
glmnGrid <- expand.grid(.alpha = c(0, .1, .2, .4, .6),.lambda = seq(.01, .2, length = 5))
set.seed(100)
glmnTuned <- train(trainPred, y = trainResp, method = "glmnet", tuneGrid = glmnGrid, preProc =
c("center", "scale"), metric = "Kappa", trControl = ctrl)
glmnTuned
plot(glmnTuned)
confusionMatrix(data = predict(glmnTuned, testPred), reference = testResp)

#Problem 12.1d For the optimal model for the biological predictors, what are the top five
important predictors?
varImp(glmnTuned)


#Problem 12.3a Explore the data by visualizing the relationship between the predictors and the
outcome. Are
#there important features of the predictor data themselves, such as between-predictor
correlations
#or degenerate distributions?
library(modeldata)
data("mlc_churn")
low.variability.columns <- nearZeroVar(mlc_churn)
hist(unlist(mlc_churn[,6]), main = 'Vmail Messages', xlab = '# Vmail Messages')
mlc_churn <- mlc_churn[,-low.variability.columns]
correlations <- cor(mlc_churn[,-c(1,3,4,5,19)])
corrplot(correlations, order = "hclust")
highly.correlated.columns <- findCorrelation(correlations, cutoff = .85)
mlc_churn <- mlc_churn[,-highly.correlated.columns]

#Problem 12.3c Split the data into training set and test set using random splitting (80% and
20%). Fit models
#covered in this chapter to the training set and tune them via resampling. Which model has the
best performance?
set.seed(100)
trainRows <- createDataPartition(mlc_churn$churn, p = .80, list = FALSE)
trainResp <- mlc_churn$churn[trainRows]
trainPred <- mlc_churn[trainRows,-c(15)]
testResp <- mlc_churn$churn[-trainRows]
```

```
testPred <- mlc_churn[-trainRows,-c(15)]

### Logistic Regression
set.seed(100)
ctrl <- trainControl(method = "LGOCV", summaryFunction = twoClassSummary, classProbs =
TRUE, savePredictions = TRUE)
lrFull <- train(trainPred, y = trainResp, method = "glm",preProc = c("center", "scale"), metric =
"ROC", trControl = ctrl)
lrFull
confusionMatrix(data = predict(lrFull, testPred), reference = testResp)
library(pROC)
rocCurve <- roc(response = testResp, predictor = predict(lrFull, testPred, type = 'prob')[,2])
auc(rocCurve)

### Penalized glmnet model
library(glmnet)
ctrl <- trainControl(method = "LGOCV",summaryFunction = twoClassSummary,classProbs =
TRUE,savePredictions = TRUE)
glmnGrid <- expand.grid(.alpha = c(0, .1, .2, .4, .6),.lambda = seq(.01, .2, length = 5))
set.seed(100)
glmnTuned <- train(trainPred, y = trainResp, method = "glmnet",tuneGrid = glmnGrid,preProc =
c("center", "scale"),metric = "ROC",trControl = ctrl)
glmnTuned
plot(glmnTuned)
confusionMatrix(data = predict(glmnTuned, testPred), reference = testResp)
rocCurve <- roc(response = testResp, predictor = predict(glmnTuned, testPred, type =
'prob')[,1])
auc(rocCurve)

### Neural Net Model
nnetGrid <- expand.grid(.size = 1:3, .decay = c(0, .1, .3, .5, 1))
maxSize <- max(nnetGrid$.size)
numWts <- (15 * (14 + 1) + (15+1)*2) ## 14 is the number of predictors; 2 is the number of
classes; 15 is size*decay
ctrl <- trainControl(method = 'LGOCV', summaryFunction = twoClassSummary,classProbs =
TRUE)
nnetFit <- train(x = trainPred, y = trainResp,method = "nnet",metric = "ROC",preProc =
c("center", "scale", "spatialSign"),tuneGrid = nnetGrid,trace = FALSE,maxit = 2000,MaxNWts =
numWts,trControl = ctrl)
nnetFit
plot(nnetFit)
confusionMatrix(data = predict(nnetFit, testPred), reference = testResp)
rocCurve <- roc(response = testResp, predictor = predict(nnetFit, testPred, type = 'prob')[,1])
auc(rocCurve)
```

```
### Mixed Discriminant Analysis
library(mda)
ctrl <- trainControl(method = "LGOCV", summaryFunction = twoClassSummary, classProbs =
TRUE, savePredictions = TRUE)
set.seed(100)
mdaFit <- train(x = trainPred, y = trainResp,method = "mda",metric = "ROC", tuneGrid =
expand.grid(.subclasses = 1:4),trControl = ctrl)
mdaFit
plot(mdaFit)
confusionMatrix(data = predict(mdaFit, testPred), reference = testResp)
rocCurve <- roc(response = testResp, predictor = predict(mdaFit, testPred, type = 'prob')[,1])
auc(rocCurve)
```