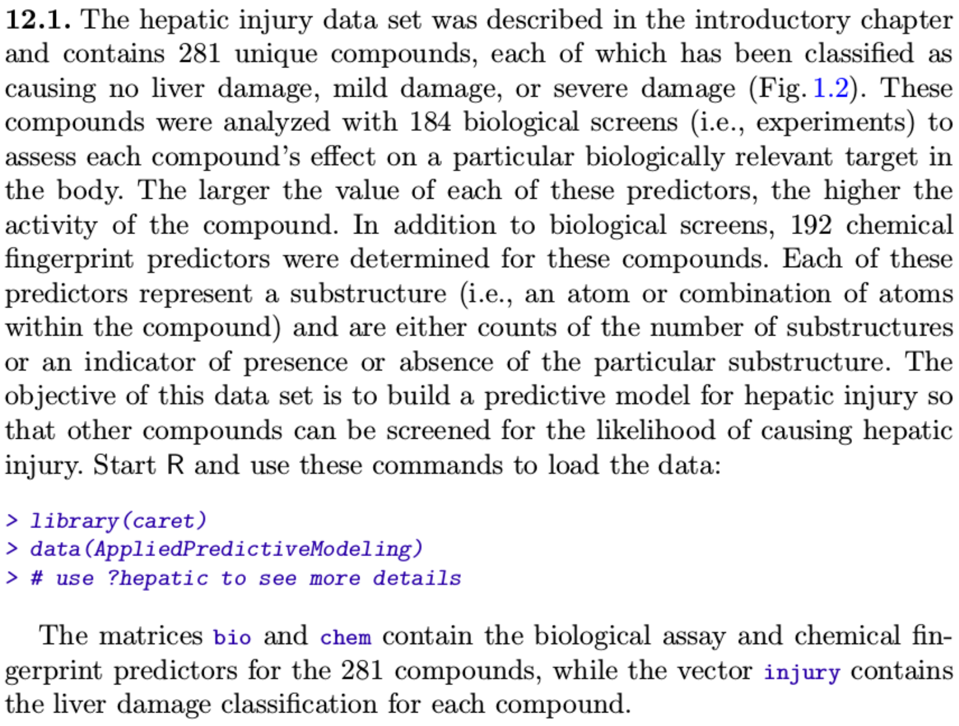
Michigan Technological University, Computer Science

MA 5790 Combined Section - Predictive Modeling
Assignment 5

Anil Silwal

December 1, 2018



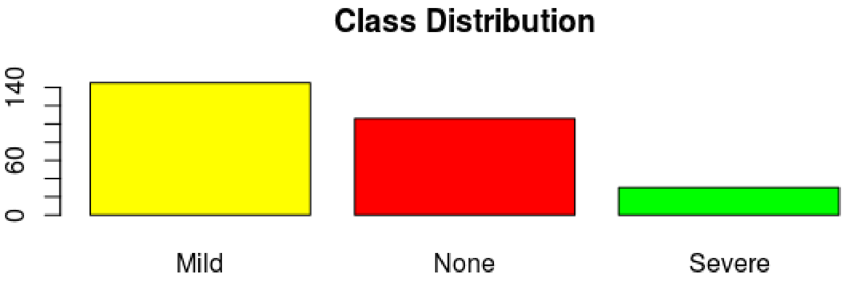
12.1. The hepatic injury data set was described in the introductory chapter and contains 281 unique compounds, each of which has been classified as causing no liver damage, mild damage, or severe damage (Fig. 1.2). These compounds were analyzed with 184 biological screens (i.e., experiments) to assess each compound's effect on a particular biologically relevant target in the body. The larger the value of each of these predictors, the higher the activity of the compound. In addition to biological screens, 192 chemical fingerprint predictors were determined for these compounds. Each of these predictors represent a substructure (i.e., an atom or combination of atoms within the compound) and are either counts of the number of substructures or an indicator of presence or absence of the particular substructure. The objective of this data set is to build a predictive model for hepatic injury so that other compounds can be screened for the likelihood of causing hepatic injury. Start R and use these commands to load the data:

```
> library(caret)
> data(AppliedPredictiveModeling)
> # use ?hepatic to see more details
```

The matrices `bio` and `chem` contain the biological assay and chemical fingerprint predictors for the 281 compounds, while the vector `injury` contains the liver damage classification for each compound.

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

Solution 12.1(a)



Given this classification imbalance in hepatic injury status, stratified random data splitting method would be the good choice to create a training and testing set.

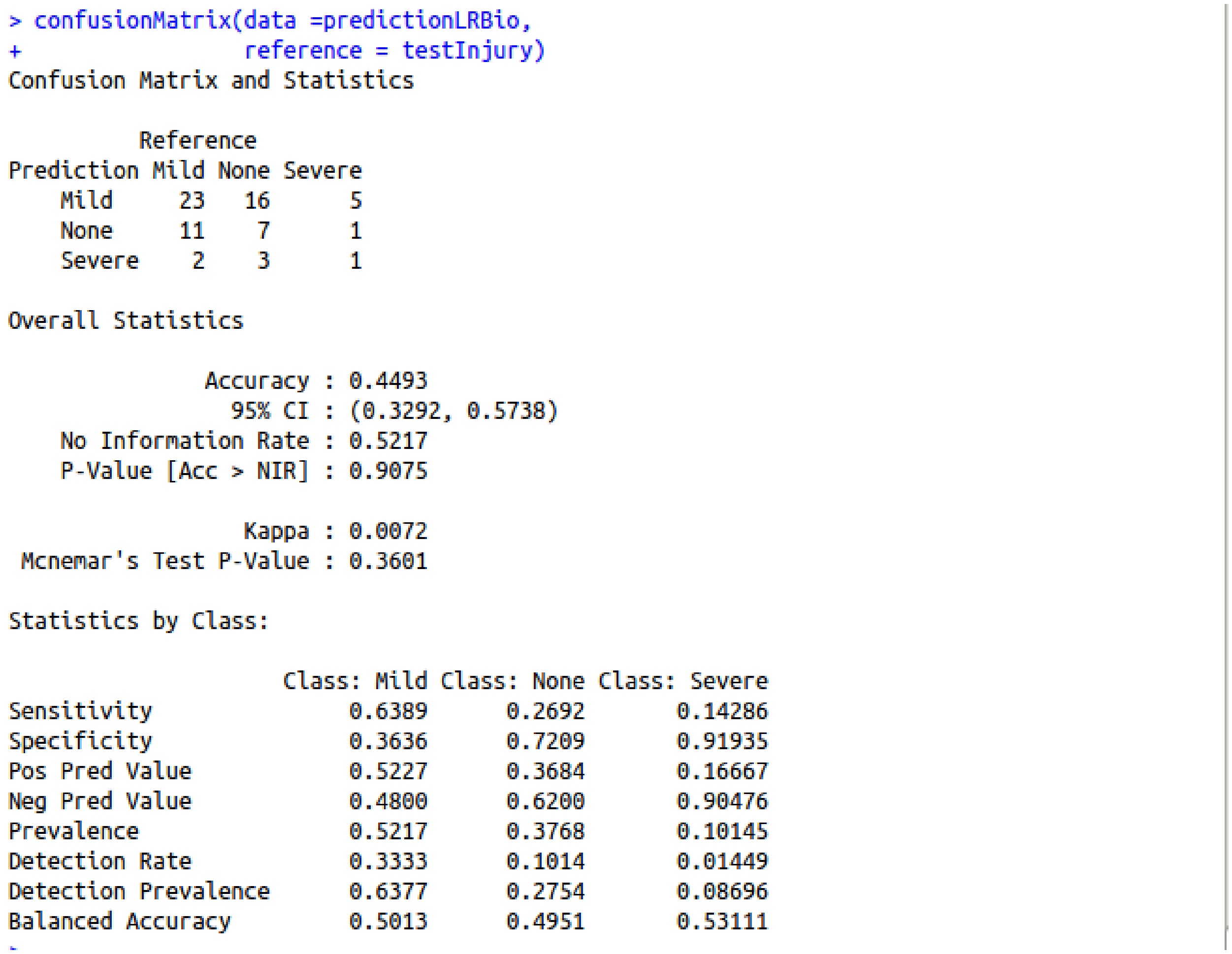
b. Which classification statistic would you choose to optimize for this exercise and why?

Solution 12.1(b)
For more than 2 classes, kappa and accuracy are the best classification statistic. In this exercise, we have 3-classes in response thus, I would choose accuracy as a classification statistic to optimize for this exercise.

c. Split the data into a training and a testing set, pre-process the data, and build models described in this chapter for the biological predictors and separately for the chemical fingerprint predictors. Which model has the best predictive ability for the biological predictors and what is the optimal performance? Which model has the best predictive ability for the chemical predictors and what is the optimal performance? Based on these results, which set of predictors contains the most information about hepatic toxicity?

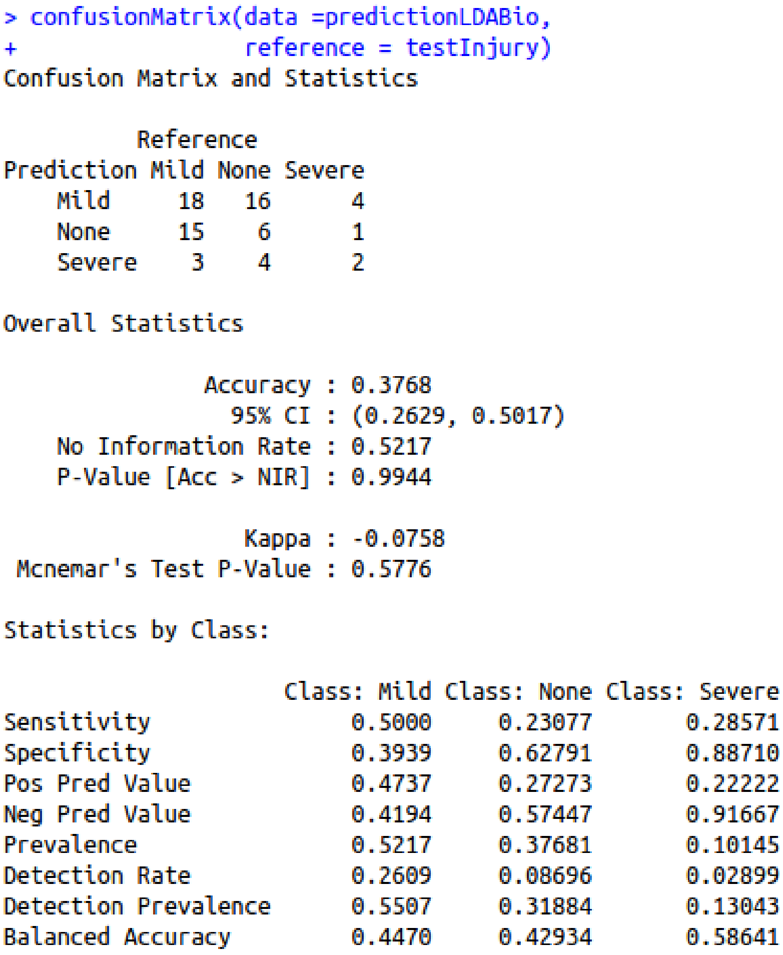
Solution:
I build the models described in given chapter for the biological screen predictor and chemical fingerprint predictor and output generated by each model are below:
first let's walk through output of model prediction for biological screen predictors:

i. **Logistic Regression Model Output for Biological Screen Predictor:**



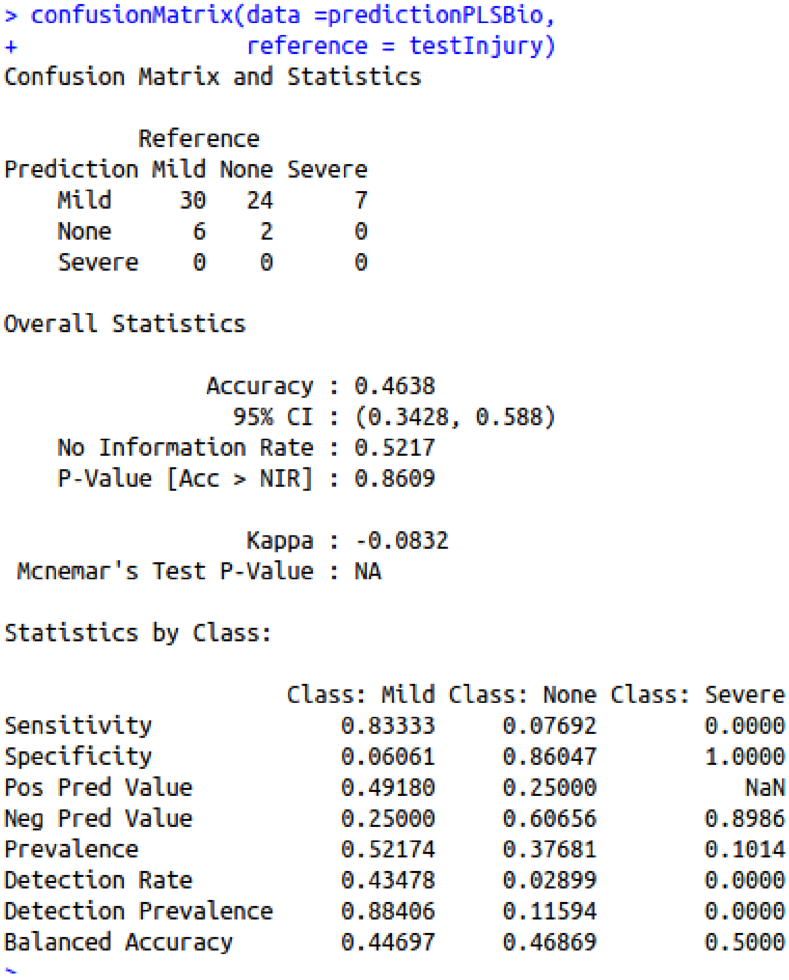
```
> confusionMatrix(data =predictionLRBio,
+                  reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     23   16      5
    None     11    7      1
    Severe    2    3      1

Overall Statistics

               Accuracy : 0.4493
                 95% CI : (0.3292, 0.5738)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.9075

                  Kappa : 0.0072
 Mcnemar's Test P-Value : 0.3601

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.6389      0.2692       0.14286
Specificity               0.3636      0.7209       0.91935
Pos Pred Value            0.5227      0.3684       0.16667
Neg Pred Value            0.4800      0.6200       0.90476
Prevalence                0.5217      0.3768       0.10145
Detection Rate            0.3333      0.1014       0.01449
Detection Prevalence      0.6377      0.2754       0.08696
Balanced Accuracy         0.5013      0.4951       0.53111
```

ii. **Linear Discriminant Analysis Output for Biological Screen Predictor:**



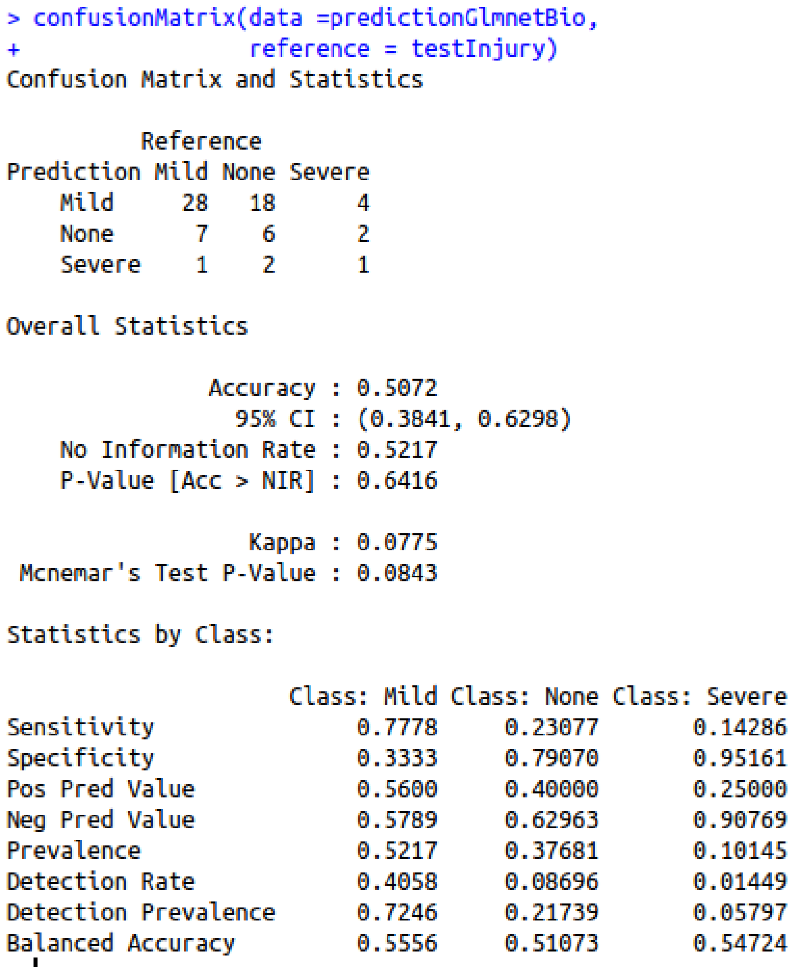
```
> confusionMatrix(data =predictionLDABio,
+                 reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     18   16      4
    None     15    6      1
    Severe    3    4      2

Overall Statistics
                                         
               Accuracy : 0.3768         
                 95% CI : (0.2629, 0.5017)
    No Information Rate : 0.5217         
    P-Value [Acc > NIR] : 0.9944         
                                         
                  Kappa : -0.0758        
 Mcnemar's Test P-Value : 0.5776         

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.5000     0.23077       0.28571
Specificity               0.3939     0.62791       0.88710
Pos Pred Value            0.4737     0.27273       0.22222
Neg Pred Value            0.4194     0.57447       0.91667
Prevalence                0.5217     0.37681       0.10145
Detection Rate            0.2609     0.08696       0.02899
Detection Prevalence      0.5507     0.31884       0.13043
Balanced Accuracy         0.4470     0.42934       0.58641
```

iii. **Partial Least Square Discriminant Analysis Output for Biological Screen Predictor:**



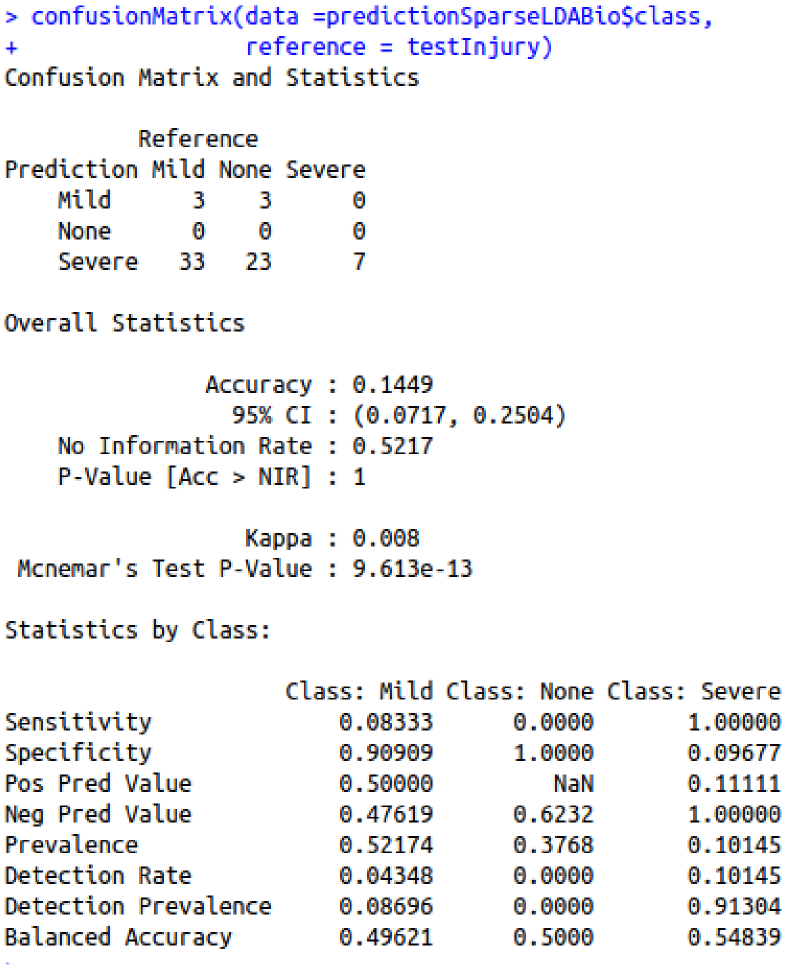
```
> confusionMatrix(data =predictionPLSBio,
+                 reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     30   24      7
    None      6    2      0
    Severe    0    0      0

Overall Statistics

               Accuracy : 0.4638
                 95% CI : (0.3428, 0.588)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.8609

                  Kappa : -0.0832
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity              0.83333     0.07692        0.0000
Specificity              0.06061     0.86047        1.0000
Pos Pred Value           0.49180     0.25000           NaN
Neg Pred Value           0.25000     0.60656        0.8986
Prevalence               0.52174     0.37681        0.1014
Detection Rate           0.43478     0.02899        0.0000
Detection Prevalence     0.88406     0.11594        0.0000
Balanced Accuracy        0.44697     0.46869        0.5000
```

iv. **Penalized Model for Logistic Regression for Biological Screen Predictor:**



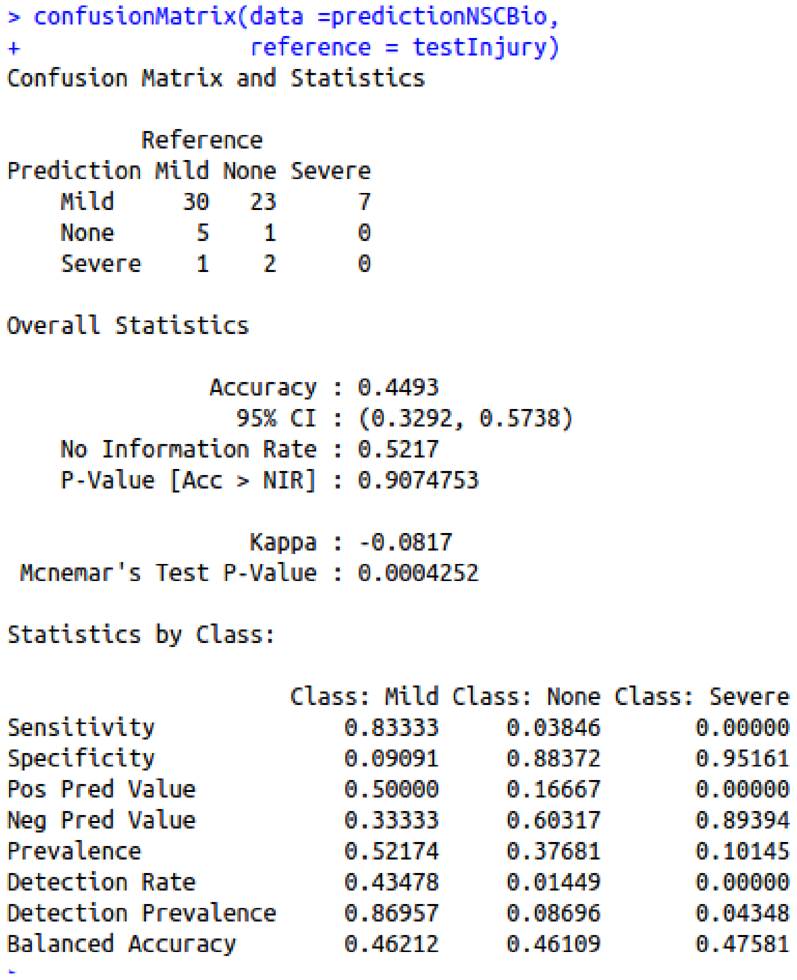
```
> confusionMatrix(data =predictionGlmnetBio,
+                 reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     28   18      4
    None      7    6      2
    Severe    1    2      1

Overall Statistics

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

                  Kappa : 0.0775
 Mcnemar's Test P-Value : 0.0843

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.7778     0.23077       0.14286
Specificity               0.3333     0.79070       0.95161
Pos Pred Value            0.5600     0.40000       0.25000
Neg Pred Value            0.5789     0.62963       0.90769
Prevalence                0.5217     0.37681       0.10145
Detection Rate            0.4058     0.08696       0.01449
Detection Prevalence      0.7246     0.21739       0.05797
Balanced Accuracy         0.5556     0.51073       0.54724
```

v. **Penalized Model for LDA Output for Biological Screen Predictor:**



```
> confusionMatrix(data =predictionSparseLDABio$class,
+                 reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild      3    3      0
    None      0    0      0
    Severe   33   23      7

Overall Statistics

               Accuracy : 0.1449
                 95% CI : (0.0717, 0.2504)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 1

                  Kappa : 0.008
 Mcnemar's Test P-Value : 9.613e-13

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity              0.08333      0.0000       1.00000
Specificity              0.90909      1.0000       0.09677
Pos Pred Value           0.50000         NaN       0.11111
Neg Pred Value           0.47619      0.6232       1.00000
Prevalence               0.52174      0.3768       0.10145
Detection Rate           0.04348      0.0000       0.10145
Detection Prevalence     0.08696      0.0000       0.91304
Balanced Accuracy        0.49621      0.5000       0.54839
```

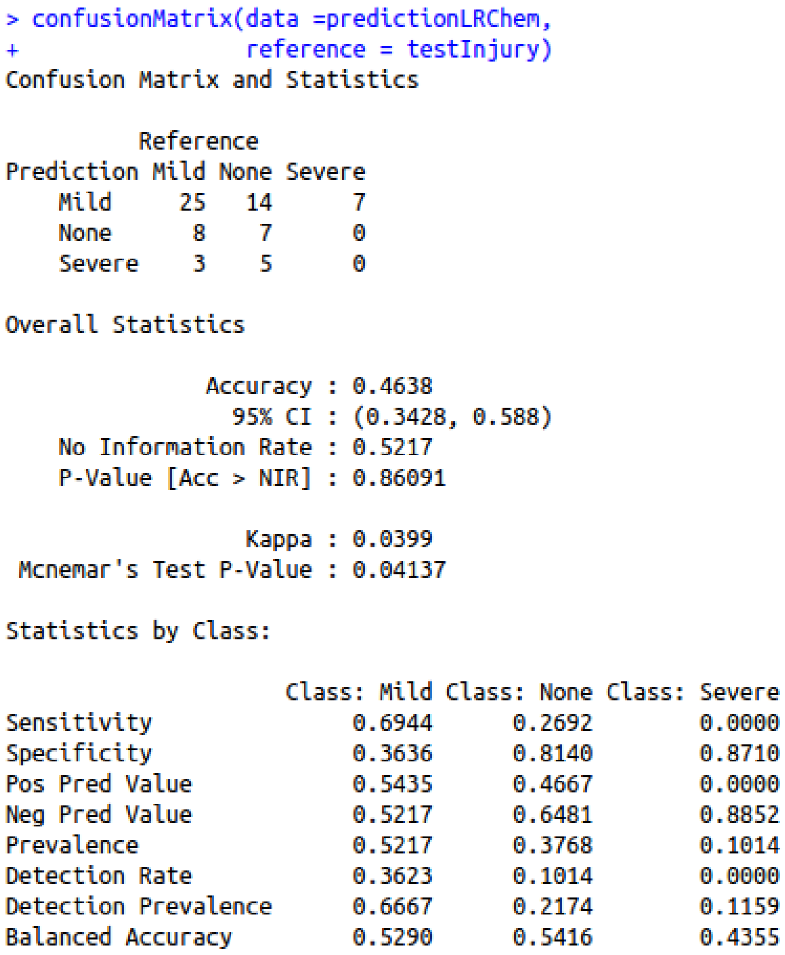
vi. **Nearest Shrinkage Centroids Output for Biological Screen Predictor:**



```
> confusionMatrix(data =predictionNSCBio,
+                 reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     30   23      7
    None      5    1      0
    Severe    1    2      0

Overall Statistics

               Accuracy : 0.4493
                 95% CI : (0.3292, 0.5738)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.9074753

                  Kappa : -0.0817
 Mcnemar's Test P-Value : 0.0004252

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity              0.83333     0.03846       0.00000
Specificity              0.09091     0.88372       0.95161
Pos Pred Value           0.50000     0.16667       0.00000
Neg Pred Value           0.33333     0.60317       0.89394
Prevalence               0.52174     0.37681       0.10145
Detection Rate           0.43478     0.01449       0.00000
Detection Prevalence     0.86957     0.08696       0.04348
Balanced Accuracy        0.46212     0.46109       0.47581
```

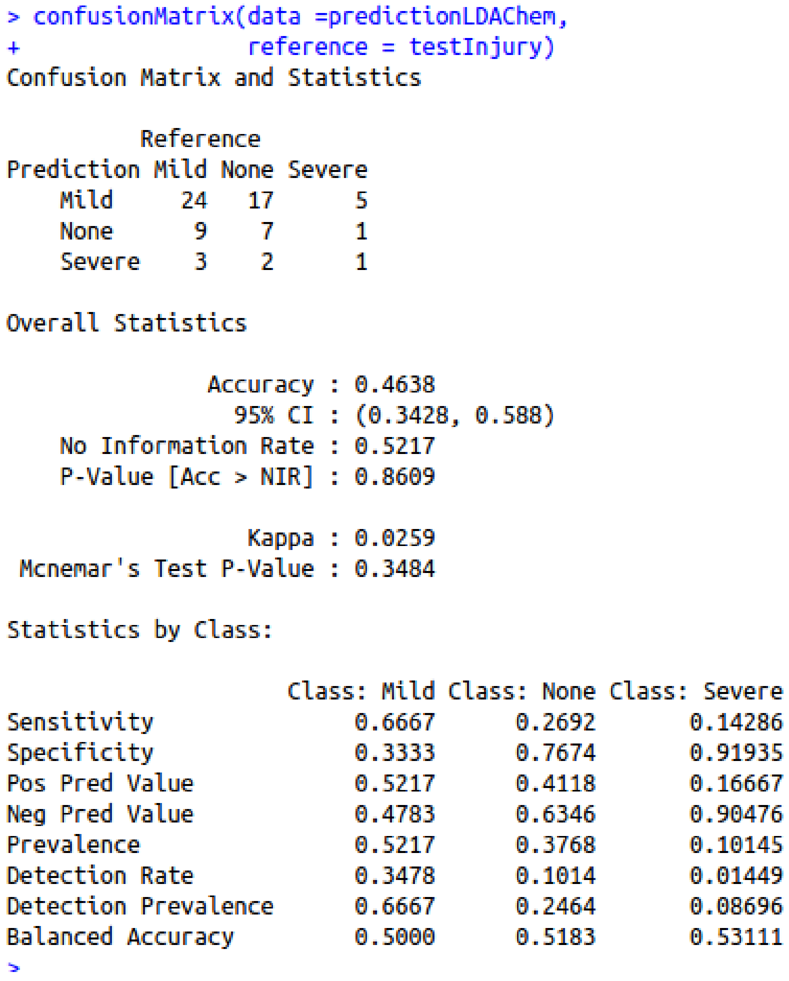
Now, Let’s walk through output of model prediction for chemical fingerprint predictors:

i. **Logistic Regression Model Output for Chemical Fingerprint Predictor:**



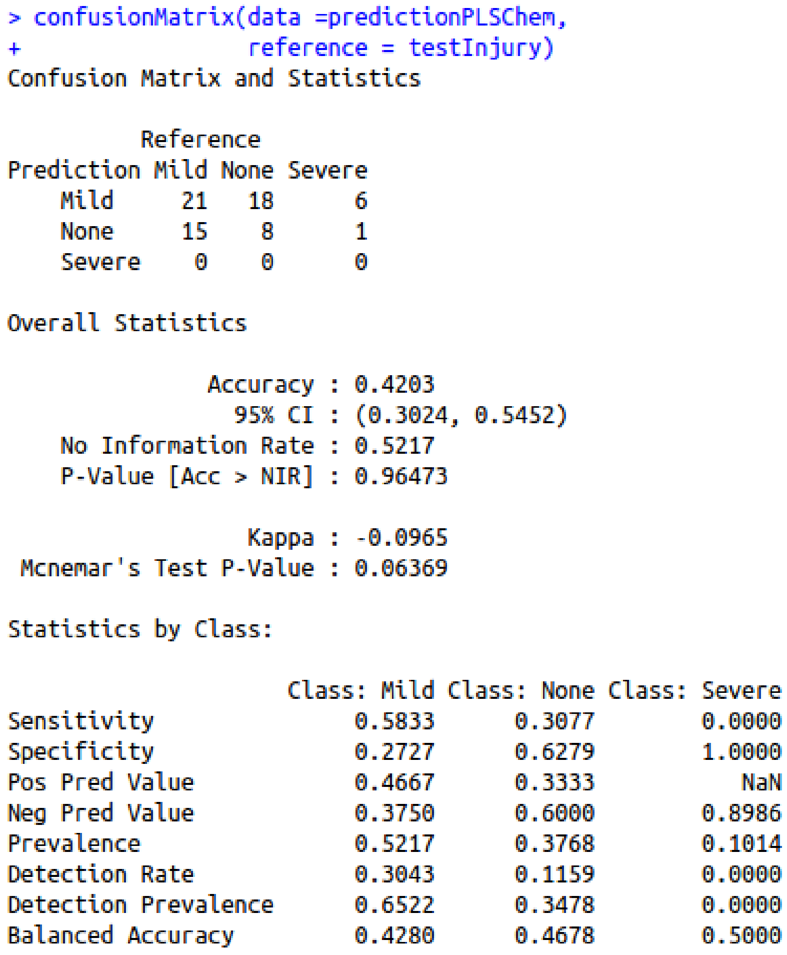
```
> confusionMatrix(data =predictionLRChem,
+                 reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     25   14      7
    None      8    7      0
    Severe    3    5      0

Overall Statistics

               Accuracy : 0.4638
                 95% CI : (0.3428, 0.588)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.86091

                  Kappa : 0.0399
 Mcnemar's Test P-Value : 0.04137

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.6944      0.2692        0.0000
Specificity               0.3636      0.8140        0.8710
Pos Pred Value            0.5435      0.4667        0.0000
Neg Pred Value            0.5217      0.6481        0.8852
Prevalence                0.5217      0.3768        0.1014
Detection Rate            0.3623      0.1014        0.0000
Detection Prevalence      0.6667      0.2174        0.1159
Balanced Accuracy         0.5290      0.5416        0.4355
```

ii. **Linear Discriminant Analysis Output for Chemical Fingerprint Predictor:**



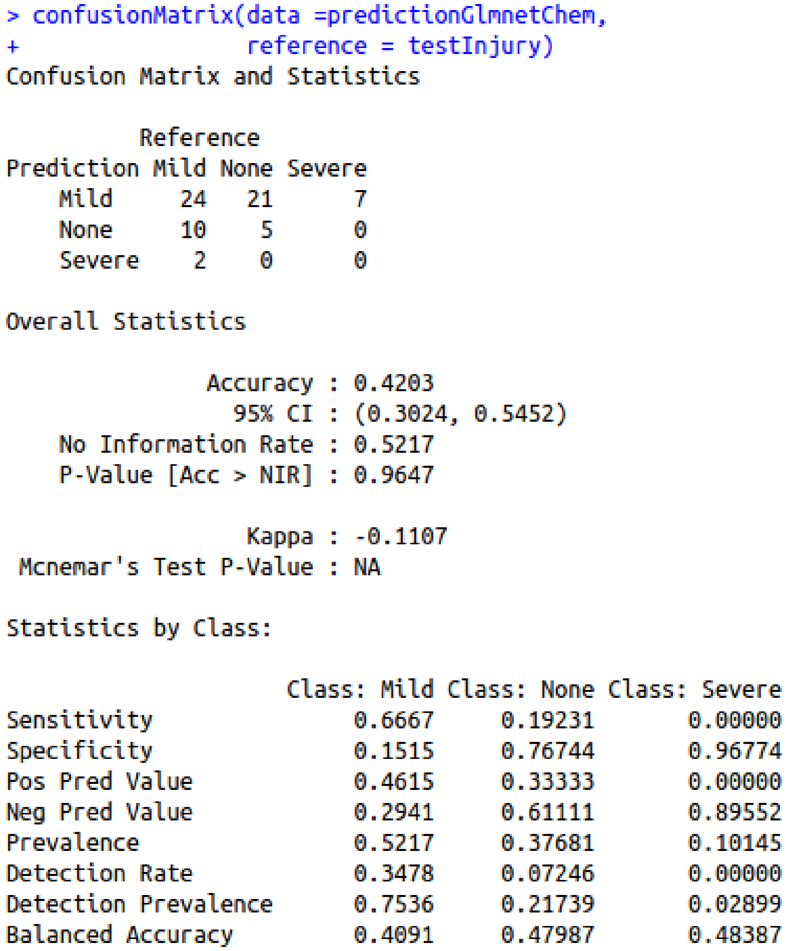
```
> confusionMatrix(data =predictionLDAChem,
+                 reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     24   17      5
    None      9    7      1
    Severe    3    2      1

Overall Statistics

               Accuracy : 0.4638
                 95% CI : (0.3428, 0.588)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.8609

                  Kappa : 0.0259
 Mcnemar's Test P-Value : 0.3484

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.6667      0.2692       0.14286
Specificity               0.3333      0.7674       0.91935
Pos Pred Value            0.5217      0.4118       0.16667
Neg Pred Value            0.4783      0.6346       0.90476
Prevalence                0.5217      0.3768       0.10145
Detection Rate            0.3478      0.1014       0.01449
Detection Prevalence      0.6667      0.2464       0.08696
Balanced Accuracy         0.5000      0.5183       0.53111
>
```

iii. **Partial Least Square Discriminant Analysis Output for Chemical Fingerprint Predictor:**



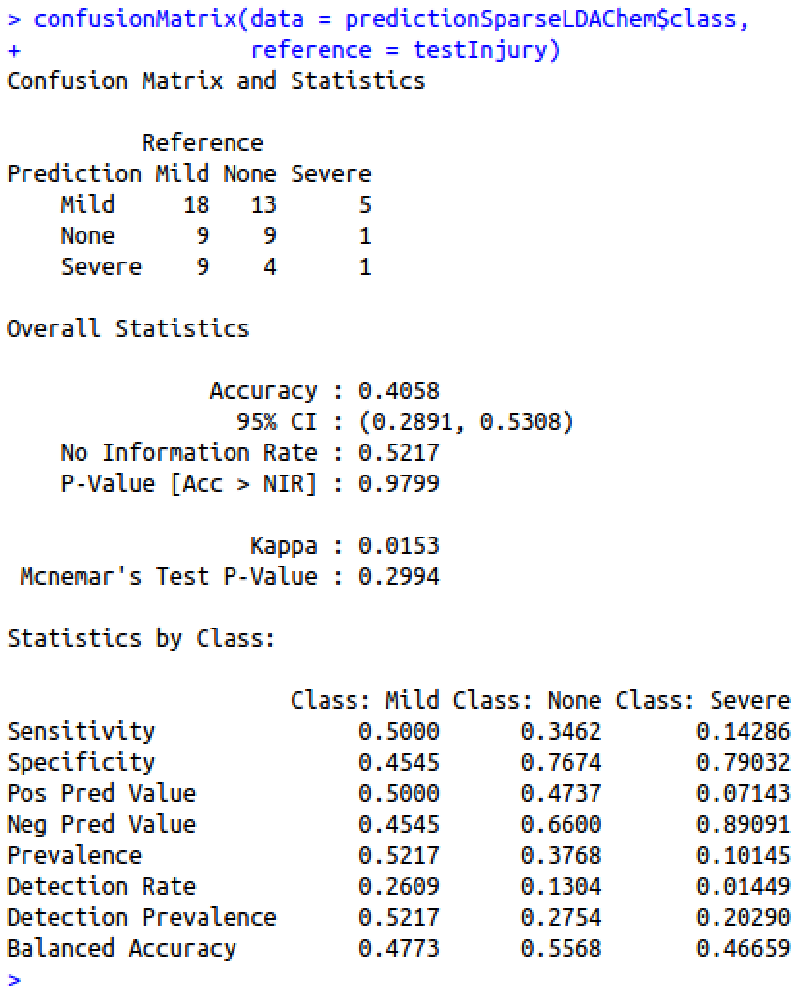
```
> confusionMatrix(data =predictionPLSChem,
+                 reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     21   18      6
    None     15    8      1
    Severe    0    0      0

Overall Statistics

               Accuracy : 0.4203
                 95% CI : (0.3024, 0.5452)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.96473

                  Kappa : -0.0965
 Mcnemar's Test P-Value : 0.06369

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.5833      0.3077        0.0000
Specificity               0.2727      0.6279        1.0000
Pos Pred Value            0.4667      0.3333           NaN
Neg Pred Value            0.3750      0.6000        0.8986
Prevalence                0.5217      0.3768        0.1014
Detection Rate            0.3043      0.1159        0.0000
Detection Prevalence      0.6522      0.3478        0.0000
Balanced Accuracy         0.4280      0.4678        0.5000
```

iv. **Penalized Model for Logistic Regression for Chemical Fingerprint Predictor:**



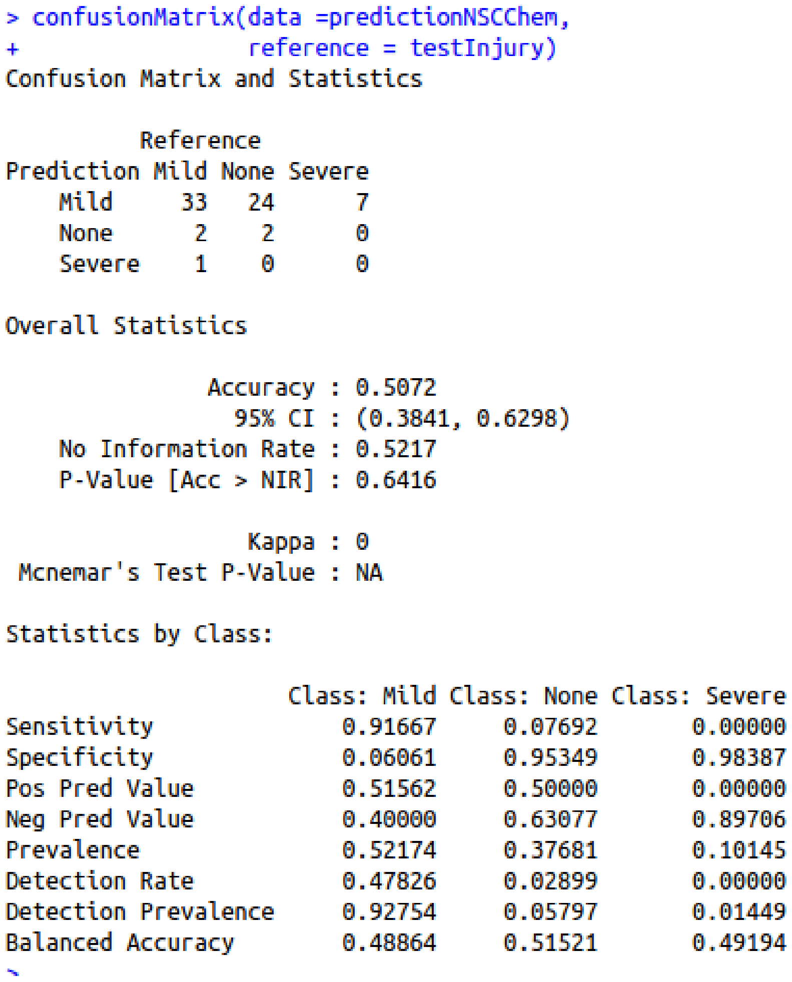
```
> confusionMatrix(data =predictionGlmnetChem,
+                 reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     24   21      7
    None     10    5      0
    Severe    2    0      0

Overall Statistics

               Accuracy : 0.4203
                 95% CI : (0.3024, 0.5452)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.9647

                  Kappa : -0.1107
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.6667     0.19231       0.00000
Specificity               0.1515     0.76744       0.96774
Pos Pred Value            0.4615     0.33333       0.00000
Neg Pred Value            0.2941     0.61111       0.89552
Prevalence                0.5217     0.37681       0.10145
Detection Rate            0.3478     0.07246       0.00000
Detection Prevalence      0.7536     0.21739       0.02899
Balanced Accuracy         0.4091     0.47987       0.48387
```

v. **Penalized Model for LDA Output for Chemical Fingerprint Predictor:**



```
> confusionMatrix(data = predictionSparseLDAChem$class,
+                 reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     18   13      5
    None      9    9      1
    Severe    9    4      1

Overall Statistics

               Accuracy : 0.4058
                 95% CI : (0.2891, 0.5308)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.9799

                  Kappa : 0.0153
 Mcnemar's Test P-Value : 0.2994

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.5000      0.3462       0.14286
Specificity               0.4545      0.7674       0.79032
Pos Pred Value            0.5000      0.4737       0.07143
Neg Pred Value            0.4545      0.6600       0.89091
Prevalence                0.5217      0.3768       0.10145
Detection Rate            0.2609      0.1304       0.01449
Detection Prevalence      0.5217      0.2754       0.20290
Balanced Accuracy         0.4773      0.5568       0.46659
>
```

vi. **Nearest Shrinkage Centroids Output for Chemical Fingerprint Predictor:**



```
> confusionMatrix(data =predictionNSCChem,
+                 reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     33   24      7
    None      2    2      0
    Severe    1    0      0

Overall Statistics

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

                  Kappa : 0
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity              0.91667     0.07692       0.00000
Specificity              0.06061     0.95349       0.98387
Pos Pred Value           0.51562     0.50000       0.00000
Neg Pred Value           0.40000     0.63077       0.89706
Prevalence               0.52174     0.37681       0.10145
Detection Rate           0.47826     0.02899       0.00000
Detection Prevalence     0.92754     0.05797       0.01449
Balanced Accuracy        0.48864     0.51521       0.49194
>
```

Solution 12.1(c)

Comparison table for different Model's Performance for predicting injury based on Biological Screen Predictors

#	Models	Bio(Accuracy)	Bio(Kappa)
1	Logistic Regression (LR)	0.4493	0.0072
2	Linear Discriminant Analysis(LDA)	0.3768	-0.0758
3	Partial Least Square Discriminant Analysis(PLS-DA)	0.4638	-0.0832
4	Penalized LR	0.5072	0.0775
5	Penalized LDA	0.1449	0.008
6	Nearest Shrinkage Centroids(NSC)	0.4493	-0.0817

This table shows that best model for predicting the hepatic injury based on biological screen predictor is: Penalized LR with accuracy **0.5072** and Kappa **0.0775**

Comparison table for different Model's Performance for predicting injury based on Chemical Fingerprints Predictors

#	Models	Chem(Accuracy)	Chem(Kappa)
1	Logistic Regression (LR)	0.4638	0.0399
2	Linear Discriminant Analysis(LDA)	0.4638	0.0259
3	Partial Least Square Discriminant Analysis(PLS-DA)	0.4203	-0.0965
4	Penalized LR	0.4203	-0.1107
5	Penalized LDA	0.4058	0.0153
6	Nearest Shrinkage Centroids(NSC)	0.5072	0.0

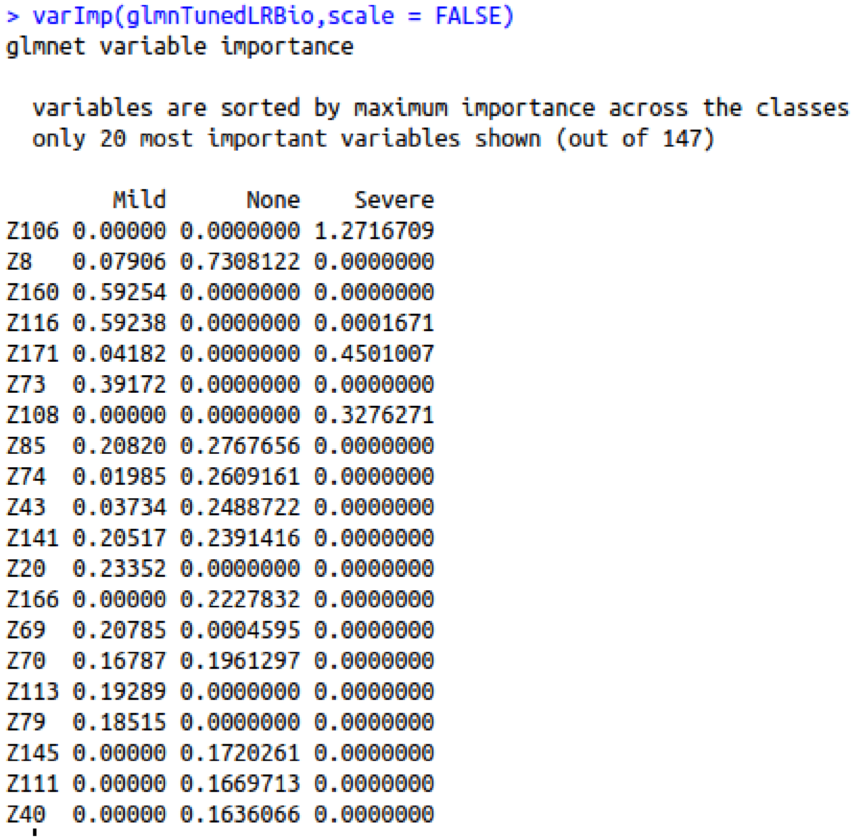
This table shows that best model for predicting the hepatic injury based on chemical fingerprints predictor is: Nearest Shrinkage Centroids with accuracy **0.5072**

d. For the optimal models for both the biological and chemical predictors,what are the top five important predictors?

Solution 12.1(d)

i. The top 5 important predictors for biological screen using optimal model(Penalized LR) are given below:

1. Z106
2. Z8
3. Z160
4. Z116
5. Z171



```
> varImp(glmnTunedLRBio,scale = FALSE)
glmnet variable importance

  variables are sorted by maximum importance across the classes
  only 20 most important variables shown (out of 147)

        Mild      None    Severe
Z106 0.00000 0.0000000 1.2716709
Z8   0.07906 0.7308122 0.0000000
Z160 0.59254 0.0000000 0.0000000
Z116 0.59238 0.0000000 0.0001671
Z171 0.04182 0.0000000 0.4501007
Z73  0.39172 0.0000000 0.0000000
Z108 0.00000 0.0000000 0.3276271
Z85  0.20820 0.2767656 0.0000000
Z74  0.01985 0.2609161 0.0000000
Z43  0.03734 0.2488722 0.0000000
Z141 0.20517 0.2391416 0.0000000
Z20  0.23352 0.0000000 0.0000000
Z166 0.00000 0.2227832 0.0000000
Z69  0.20785 0.0004595 0.0000000
Z70  0.16787 0.1961297 0.0000000
Z113 0.19289 0.0000000 0.0000000
Z79  0.18515 0.0000000 0.0000000
Z145 0.00000 0.1720261 0.0000000
Z111 0.00000 0.1669713 0.0000000
Z40  0.00000 0.1636066 0.0000000
```

ii. The top 5 important predictors for chemical Fingerprints using optimal model(Nearest Shrinkage Centroids) are given below:

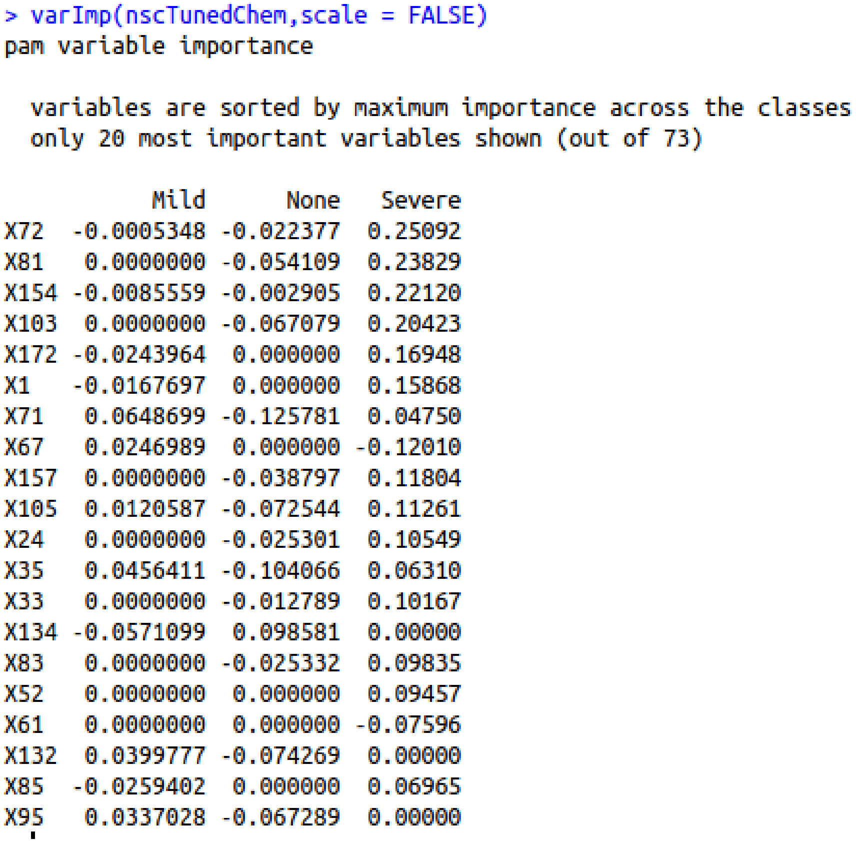
1. X72

2. X81

3. X154

4. X103

5. X172

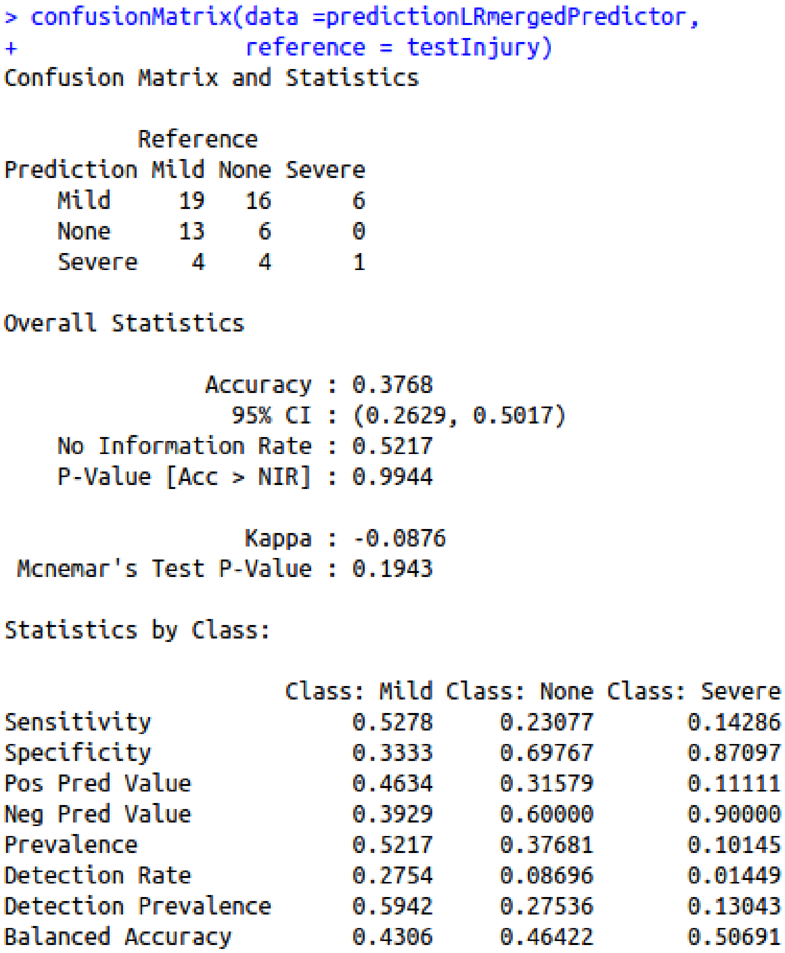


```
> varImp(nscTunedChem,scale = FALSE)
pam variable importance

  variables are sorted by maximum importance across the classes
  only 20 most important variables shown (out of 73)

          Mild      None   Severe
X72  -0.0005348 -0.022377  0.25092
X81   0.0000000 -0.054109  0.23829
X154 -0.0085559 -0.002905  0.22120
X103  0.0000000 -0.067079  0.20423
X172 -0.0243964  0.000000  0.16948
X1   -0.0167697  0.000000  0.15868
X71   0.0648699 -0.125781  0.04750
X67   0.0246989  0.000000 -0.12010
X157  0.0000000 -0.038797  0.11804
X105  0.0120587 -0.072544  0.11261
X24   0.0000000 -0.025301  0.10549
X35   0.0456411 -0.104066  0.06310
X33   0.0000000 -0.012789  0.10167
X134 -0.0571099  0.098581  0.00000
X83   0.0000000 -0.025332  0.09835
X52   0.0000000  0.000000  0.09457
X61   0.0000000  0.000000 -0.07596
X132  0.0399777 -0.074269  0.00000
X85  -0.0259402  0.000000  0.06965
X95   0.0337028 -0.067289  0.00000
```

e. Now combine the biological and chemical fingerprint predictors into one predictor set. Retrain the same set of predictive models you built from part (c). Which model yields best predictive performance? Is the model performance better than either of the best models from part (c)? What are the top five important predictors for the optimal model? How do these compare with the optimal predictors from each individual predictor set?

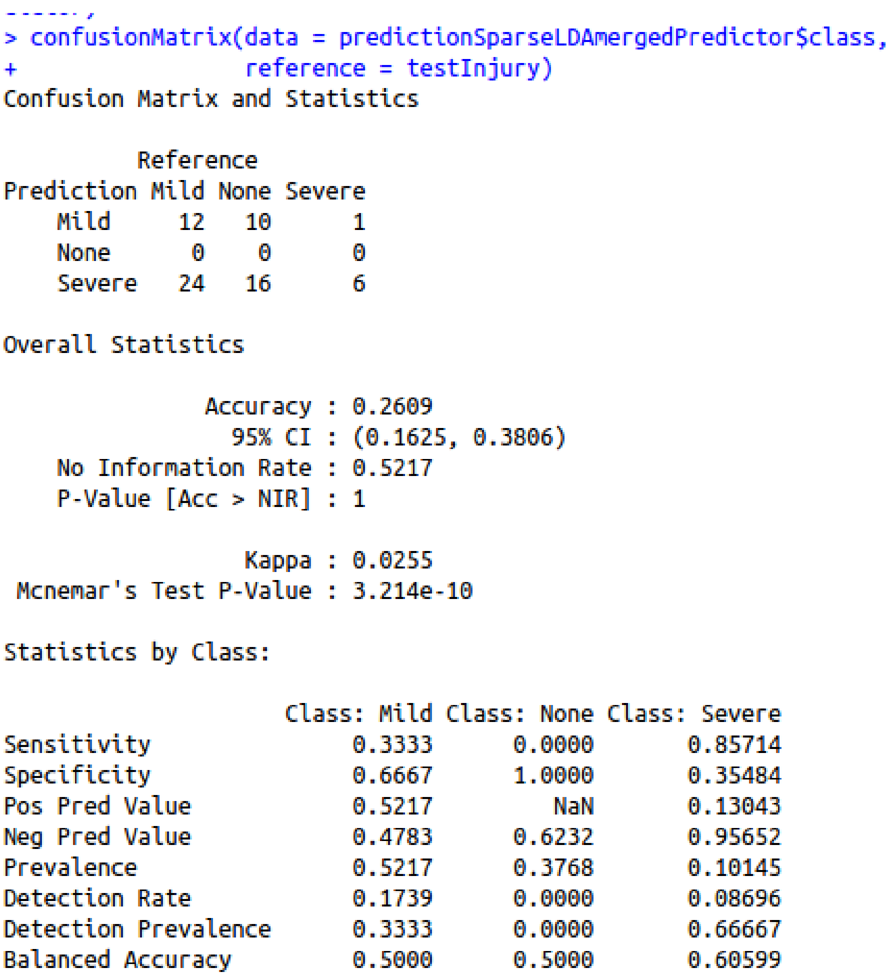
Solution 12.1(e)
I merged the filtered predictors of bio(147,out of 184) and chem(73,out of 192) data into one predictor set. Now, Let's walk through output of model prediction from merged predictors of biological screen and chemical fingerprint predictors:

i. **Logistic Regression Model Output for Merged Predictor:**



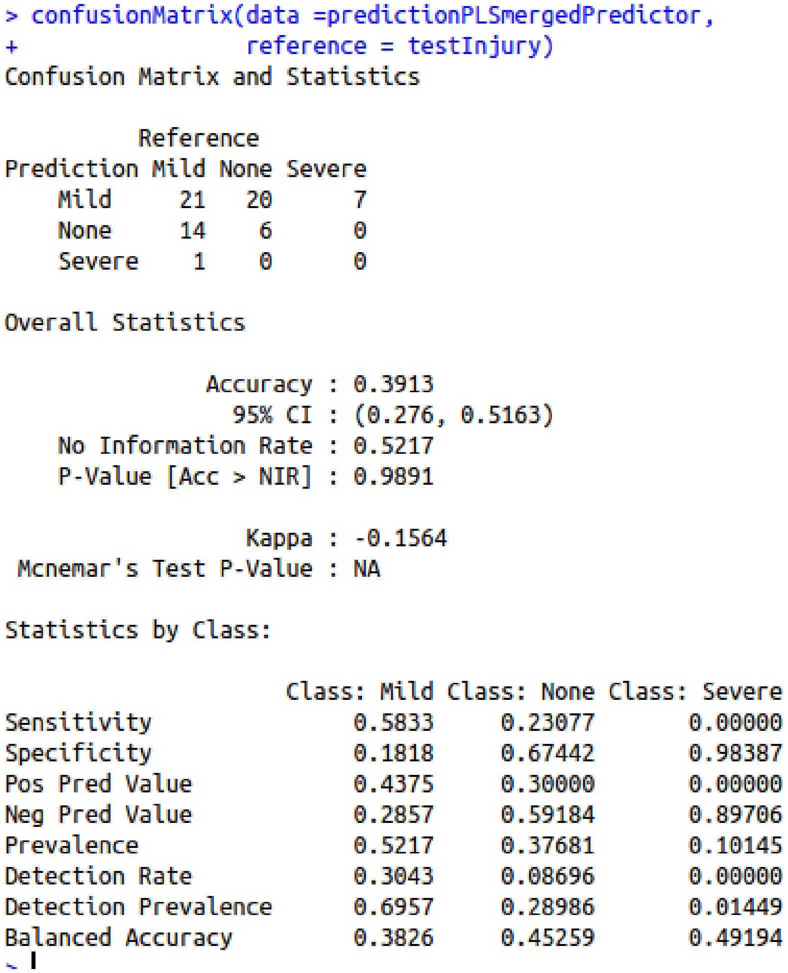
```
> confusionMatrix(data =predictionLRmergedPredictor,
+                  reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     19   16      6
    None     13    6      0
    Severe    4    4      1

Overall Statistics

               Accuracy : 0.3768
                 95% CI : (0.2629, 0.5017)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.9944

                  Kappa : -0.0876
 Mcnemar's Test P-Value : 0.1943

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.5278     0.23077       0.14286
Specificity               0.3333     0.69767       0.87097
Pos Pred Value            0.4634     0.31579       0.11111
Neg Pred Value            0.3929     0.60000       0.90000
Prevalence                0.5217     0.37681       0.10145
Detection Rate            0.2754     0.08696       0.01449
Detection Prevalence      0.5942     0.27536       0.13043
Balanced Accuracy         0.4306     0.46422       0.50691
```

ii. **Linear Discriminant Analysis Output for Merged Predictor:**



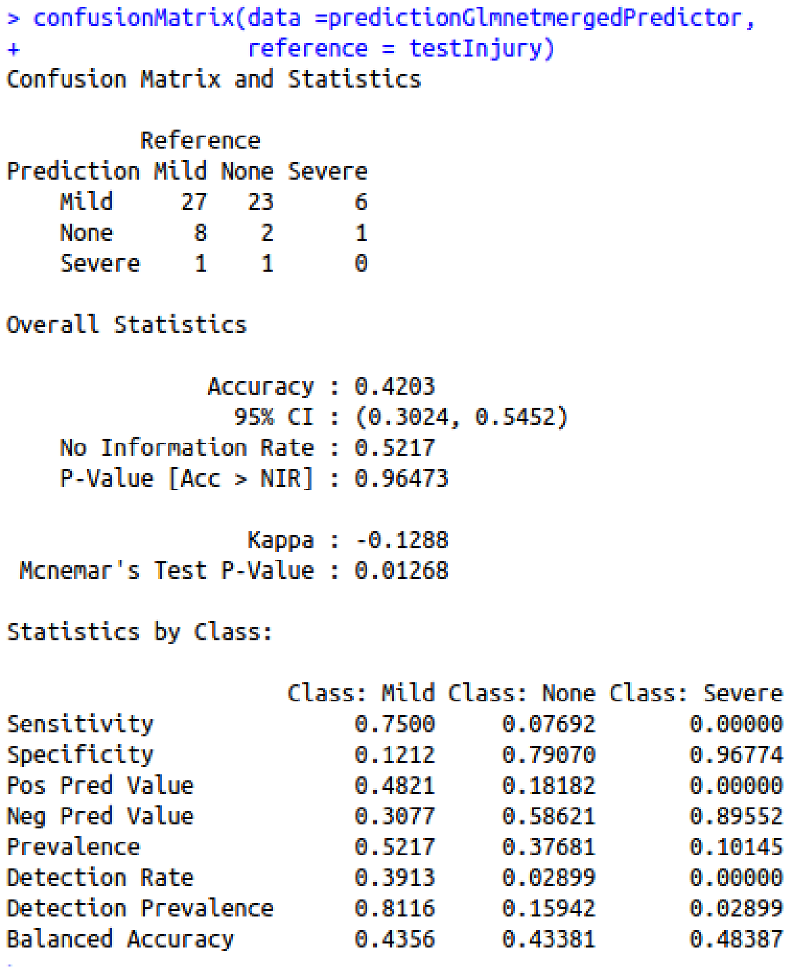
```
------,
> confusionMatrix(data = predictionSparseLDAmergedPredictor$class,
+                 reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     12   10      1
    None      0    0      0
    Severe   24   16      6

Overall Statistics

               Accuracy : 0.2609
                 95% CI : (0.1625, 0.3806)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0255
 Mcnemar's Test P-Value : 3.214e-10

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.3333      0.0000       0.85714
Specificity               0.6667      1.0000       0.35484
Pos Pred Value            0.5217         NaN       0.13043
Neg Pred Value            0.4783      0.6232       0.95652
Prevalence                0.5217      0.3768       0.10145
Detection Rate            0.1739      0.0000       0.08696
Detection Prevalence      0.3333      0.0000       0.66667
Balanced Accuracy         0.5000      0.5000       0.60599
```

iii. **Partial Least Square Discriminant Analysis Output for Merged Predictor:**



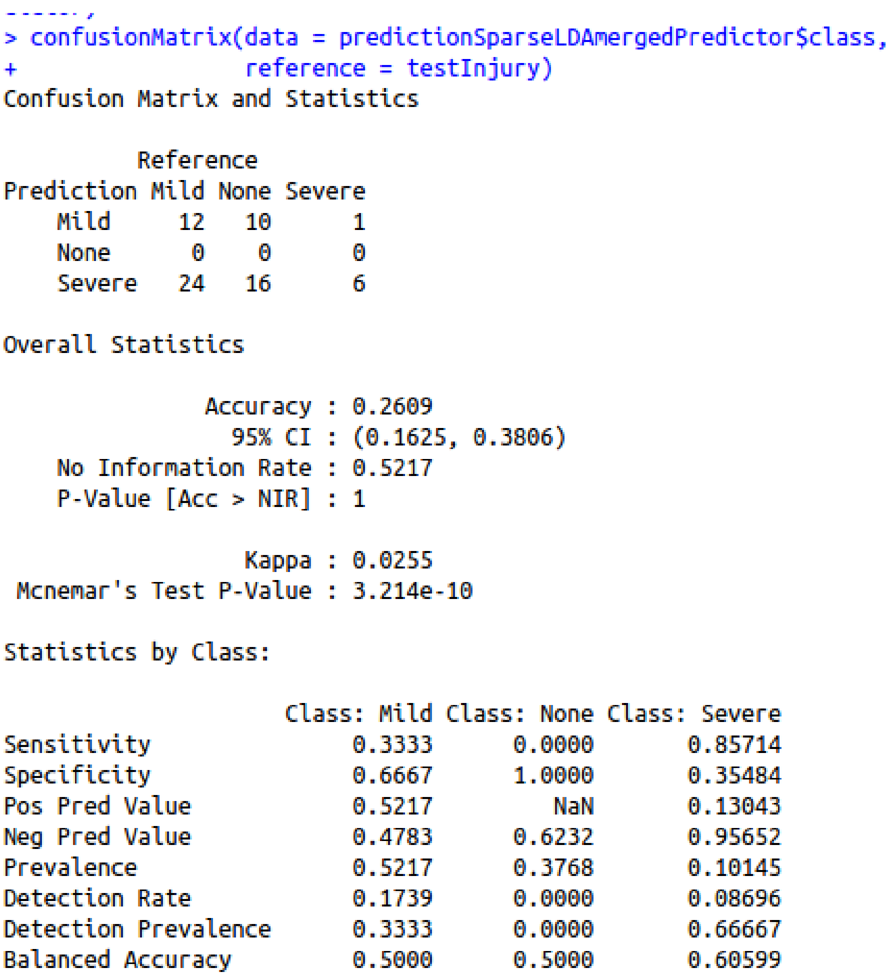
```
> confusionMatrix(data =predictionPLSmergedPredictor,
+                 reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     21   20      7
    None     14    6      0
    Severe    1    0      0

Overall Statistics

               Accuracy : 0.3913
                 95% CI : (0.276, 0.5163)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.9891

                  Kappa : -0.1564
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.5833     0.23077       0.00000
Specificity               0.1818     0.67442       0.98387
Pos Pred Value            0.4375     0.30000       0.00000
Neg Pred Value            0.2857     0.59184       0.89706
Prevalence                0.5217     0.37681       0.10145
Detection Rate            0.3043     0.08696       0.00000
Detection Prevalence      0.6957     0.28986       0.01449
Balanced Accuracy         0.3826     0.45259       0.49194
>
```

iv. **Penalized Model for Logistic Regression for Merged Predictor:**



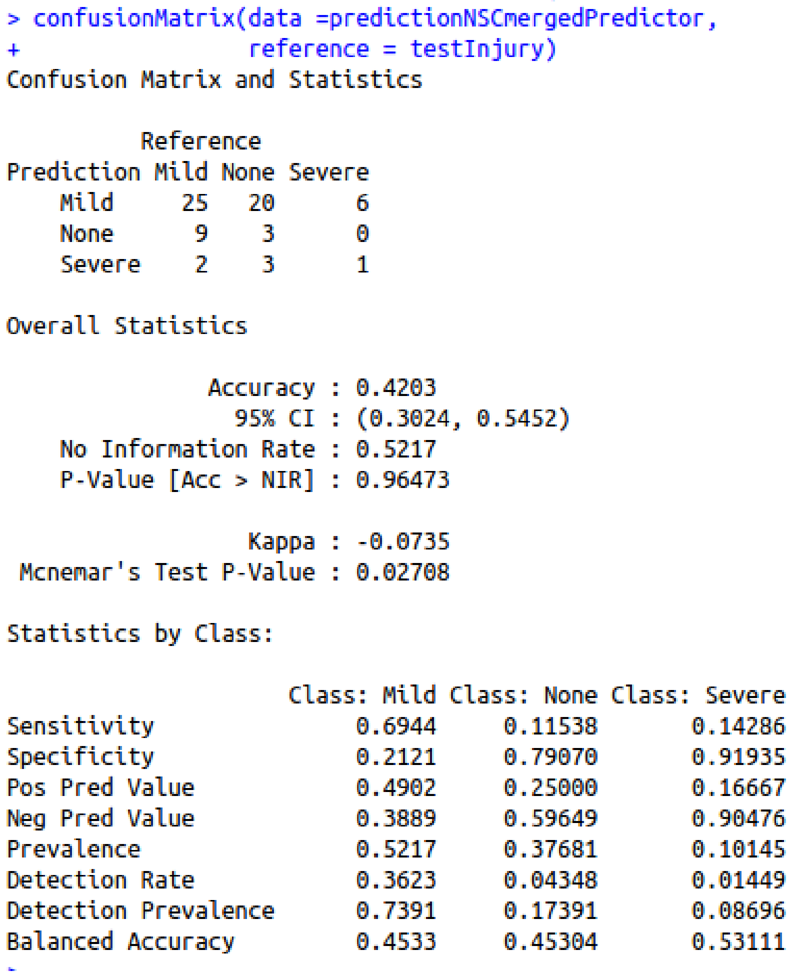
```
> confusionMatrix(data =predictionGlmnetmergedPredictor,
+                 reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     27   23      6
    None      8    2      1
    Severe    1    1      0

Overall Statistics

               Accuracy : 0.4203
                 95% CI : (0.3024, 0.5452)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.96473

                  Kappa : -0.1288
 Mcnemar's Test P-Value : 0.01268

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.7500     0.07692       0.00000
Specificity               0.1212     0.79070       0.96774
Pos Pred Value            0.4821     0.18182       0.00000
Neg Pred Value            0.3077     0.58621       0.89552
Prevalence                0.5217     0.37681       0.10145
Detection Rate            0.3913     0.02899       0.00000
Detection Prevalence      0.8116     0.15942       0.02899
Balanced Accuracy         0.4356     0.43381       0.48387
```

v. Penalized Model for LDA Output for Merged Predictor:



```
> confusionMatrix(data = predictionSparseLDAmergedPredictor$class,
+                  reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     12   10      1
    None      0    0      0
    Severe   24   16      6

Overall Statistics

               Accuracy : 0.2609
                 95% CI : (0.1625, 0.3806)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0255
 Mcnemar's Test P-Value : 3.214e-10

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.3333      0.0000       0.85714
Specificity               0.6667      1.0000       0.35484
Pos Pred Value            0.5217         NaN       0.13043
Neg Pred Value            0.4783      0.6232       0.95652
Prevalence                0.5217      0.3768       0.10145
Detection Rate            0.1739      0.0000       0.08696
Detection Prevalence      0.3333      0.0000       0.66667
Balanced Accuracy         0.5000      0.5000       0.60599
```

vi. **Nearest Shrinkage Centroids Output for Merged Predictor:**



```
> confusionMatrix(data =predictionNSCmergedPredictor,
+                 reference = testInjury)
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     25   20      6
    None      9    3      0
    Severe    2    3      1

Overall Statistics

               Accuracy : 0.4203
                 95% CI : (0.3024, 0.5452)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.96473

                  Kappa : -0.0735
 Mcnemar's Test P-Value : 0.02708

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.6944     0.11538       0.14286
Specificity               0.2121     0.79070       0.91935
Pos Pred Value            0.4902     0.25000       0.16667
Neg Pred Value            0.3889     0.59649       0.90476
Prevalence                0.5217     0.37681       0.10145
Detection Rate            0.3623     0.04348       0.01449
Detection Prevalence      0.7391     0.17391       0.08696
Balanced Accuracy         0.4533     0.45304       0.53111
```

Comparison table for different Model's Performance for predicting injury based on merged Biological Screens and Chemical Fingerprints Predictors

#	Models	Merged(Accuracy)	Merged(Kappa)
1	Logistic Regression (LR)	0.3768	-0.0876
2	Linear Discriminant Analysis(LDA)	0.2609	0.0255
3	Partial Least Square Discriminant Analysis(PLS-DA)	0.3913	-0.1564
4	Penalized LR	0.4203	-0.1288
5	Penalized LDA	0.2609	0.0255
6	Nearest Shrinkage Centroids(NSC)	0.4203	-0.0735

i. This table shows that best model for predicting the hepatic injury based on merged biological screen and chemical fingerprints predictor as: Penalized LR and Nearest Shrinkage Centroids with same accuracy **0.4203**
i.e

   1. Penalized LR with accuracy = 0.4203

   2. Nearest Shrinkage Centroids with accuracy = 0.4203

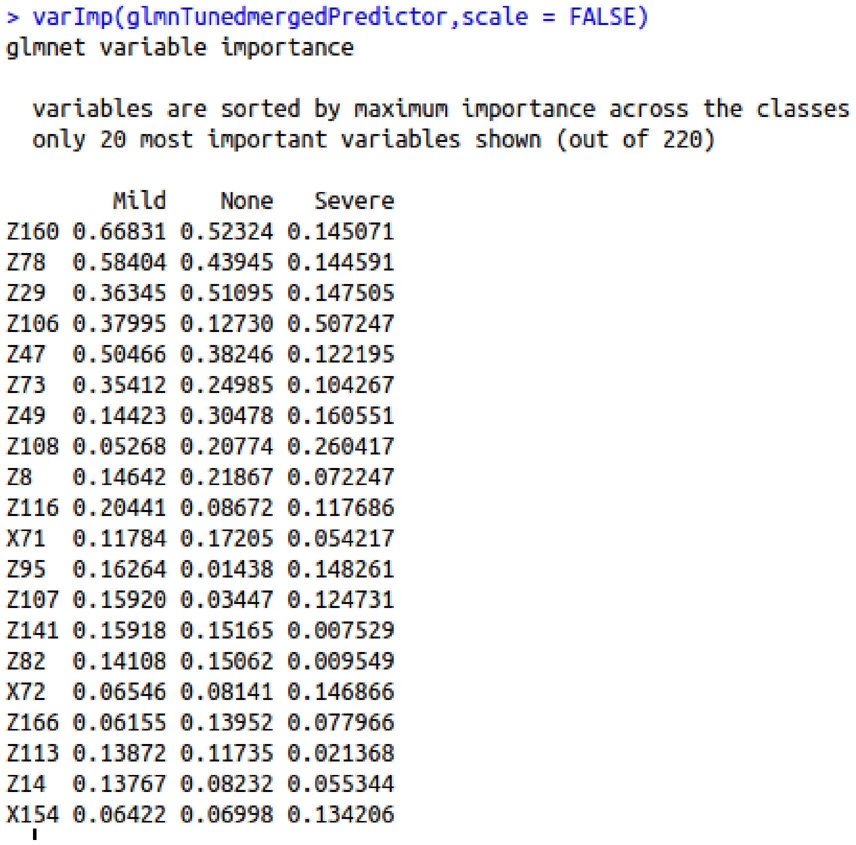
ii. This optimal model are same as those optimal model from individual biological and chemical predictors

   1. Optimal model for Bio predictor: Penalized LR with accuracy = 0.5072

   2. Optimal model for Chem predictor:Nearest Shrinkage Centroids with accuracy = 0.5072

iii. The top 5 important predictors for merged predictors using optimal model(Penalized LR) are given below:

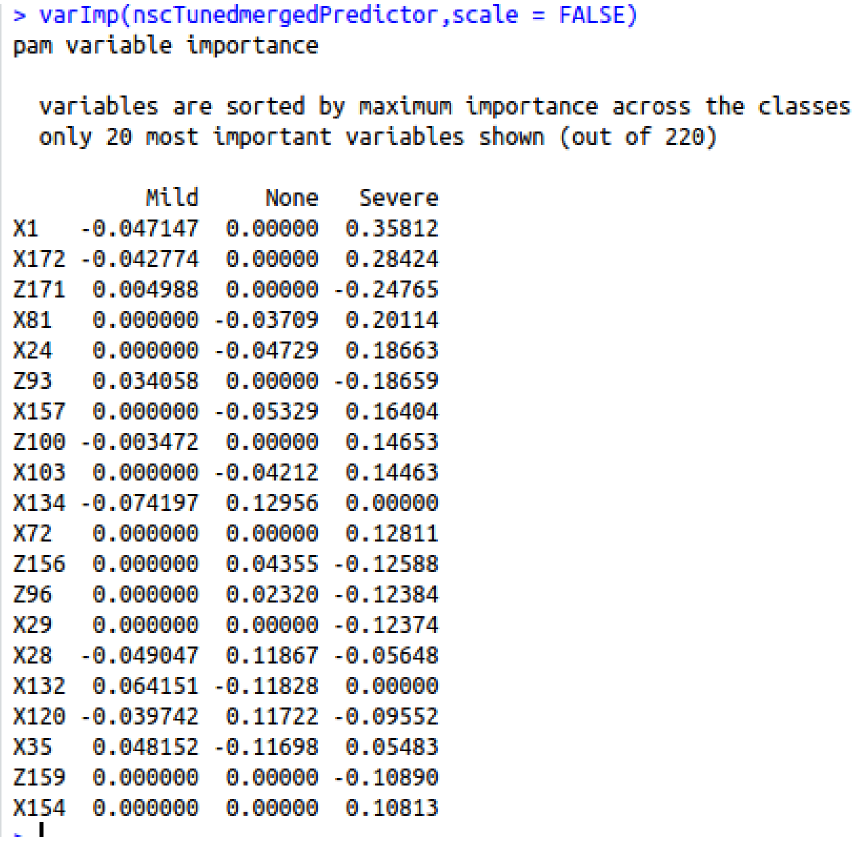
1. Z160
2. Z78
3. Z29
4. Z106
5. Z47



```
> varImp(glmnTunedmergedPredictor,scale = FALSE)
glmnet variable importance

  variables are sorted by maximum importance across the classes
  only 20 most important variables shown (out of 220)

        Mild    None   Severe
Z160 0.66831 0.52324 0.145071
Z78  0.58404 0.43945 0.144591
Z29  0.36345 0.51095 0.147505
Z106 0.37995 0.12730 0.507247
Z47  0.50466 0.38246 0.122195
Z73  0.35412 0.24985 0.104267
Z49  0.14423 0.30478 0.160551
Z108 0.05268 0.20774 0.260417
Z8   0.14642 0.21867 0.072247
Z116 0.20441 0.08672 0.117686
X71  0.11784 0.17205 0.054217
Z95  0.16264 0.01438 0.148261
Z107 0.15920 0.03447 0.124731
Z141 0.15918 0.15165 0.007529
Z82  0.14108 0.15062 0.009549
X72  0.06546 0.08141 0.146866
Z166 0.06155 0.13952 0.077966
Z113 0.13872 0.11735 0.021368
Z14  0.13767 0.08232 0.055344
X154 0.06422 0.06998 0.134206
```

The top 5 important predictors for merged predictors using optimal model (Nearest Shrinkage Centroids) are given below:

1. X1
2. X172
3. Z171
4. X81
5. X24



```
> varImp(nscTunedmergedPredictor,scale = FALSE)
pam variable importance

  variables are sorted by maximum importance across the classes
  only 20 most important variables shown (out of 220)

          Mild     None   Severe
X1   -0.047147  0.00000  0.35812
X172 -0.042774  0.00000  0.28424
Z171  0.004988  0.00000 -0.24765
X81   0.000000 -0.03709  0.20114
X24   0.000000 -0.04729  0.18663
Z93   0.034058  0.00000 -0.18659
X157  0.000000 -0.05329  0.16404
Z100 -0.003472  0.00000  0.14653
X103  0.000000 -0.04212  0.14463
X134 -0.074197  0.12956  0.00000
X72   0.000000  0.00000  0.12811
Z156  0.000000  0.04355 -0.12588
Z96   0.000000  0.02320 -0.12384
X29   0.000000  0.00000 -0.12374
X28  -0.049047  0.11867 -0.05648
X132  0.064151 -0.11828  0.00000
X120 -0.039742  0.11722 -0.09552
X35   0.048152 -0.11698  0.05483
Z159  0.000000  0.00000 -0.10890
X154  0.000000  0.00000  0.10813
```

Comparison table for different individual predictor set with top five important predictors of optimal Model

Predictors	Bio (Penalized - LR)	Mixed (Penalized - LR)	Chem(NSC)	Mixed(NSC)
1	Z106	Z160	X72	X1
2	Z8	Z78	X81	X172
3	Z160	Z29	X154	Z71
4	Z116	Z106	X103	X81
5	Z171	Z47	X172	X24

f. Which model (either model of individual biology or chemical fingerprints or the combined predictor model), if any, would you recommend using to predict compounds hepatic toxicity? Explain.

Solution 12.1(f)

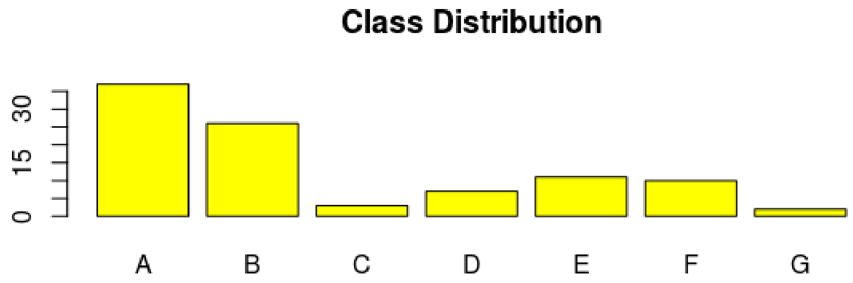
Based on the results, Penalized LR from bio predictor or Nearest Shrinkage Centroids from chem would be the best to predict the hepatic injury. Because both give same accuracy of 0.5072.

However,the combined predictor has two optimal models: Penalized LR and Nearest Shrinkage Centroids which give same accuracy of 0.4203 which is little lower i.e 0.5072 > 0.4203

12.2 In Exercise 4.4, we described a data set which contained 96 oil samples each from one of seven types of oils (pumpkin, sunflower, peanut, olive, soybean, rapeseed, and corn). Gas chromatography was performed on each sample and the percentage of each type of 7 fatty acids was determined. We would like to use these data to build a model that predicts the type of oil based on a samplefis fatty acid percentages.

a. Like the hepatic injury data, these data suffer from extreme imbalance.Given this imbalance, should the data be split into training and test sets?

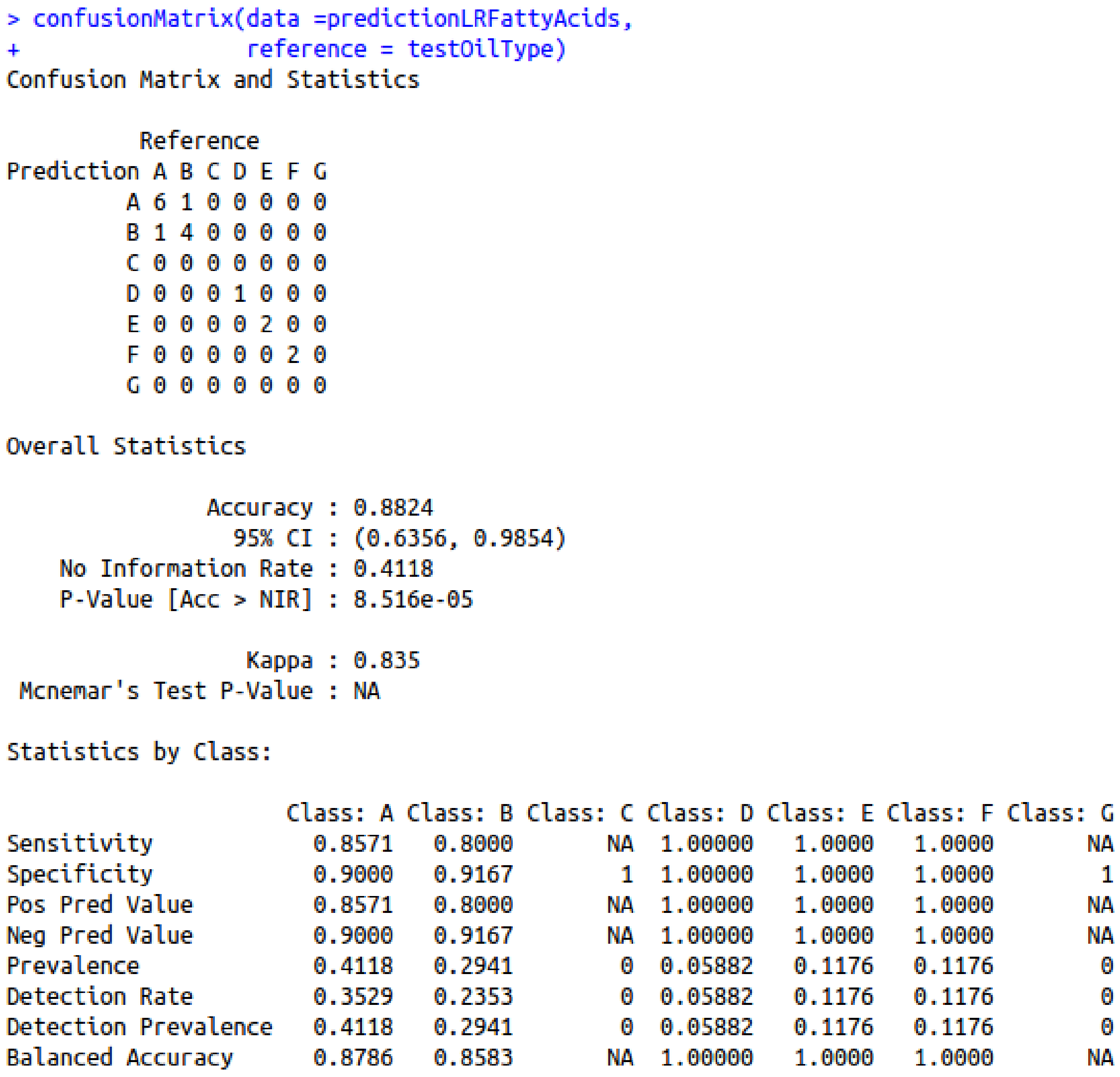
Solution 12.2(a)



Given this classification imbalance in oilType, stratified random data splitting method would be the good choice to create a training and testing set.

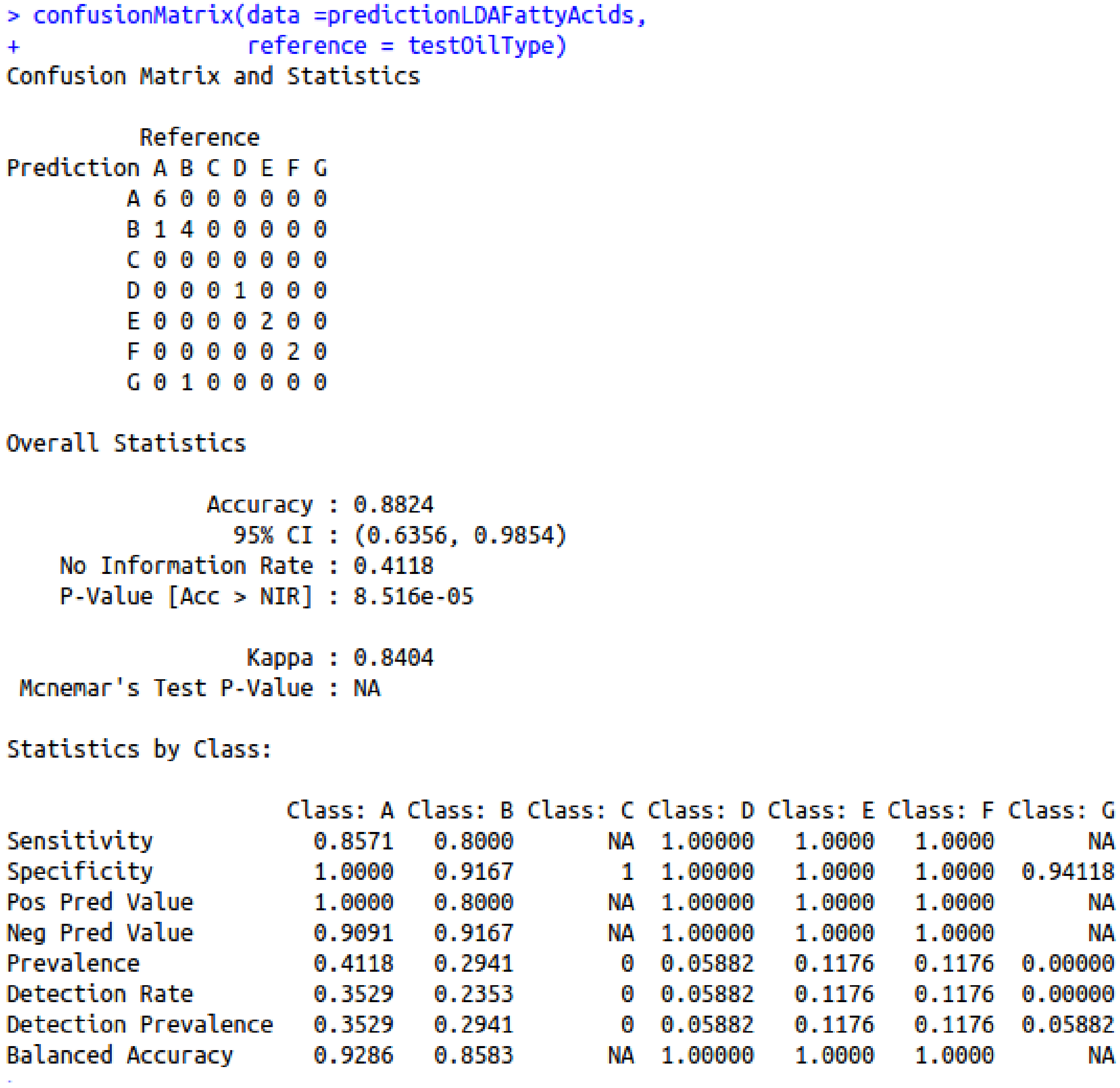
I build the models described in given chapter for the fatty acids predictor and output generated by each model are below:

i. **Logistic Regression Model Output for Fatty Acids Predictor:**



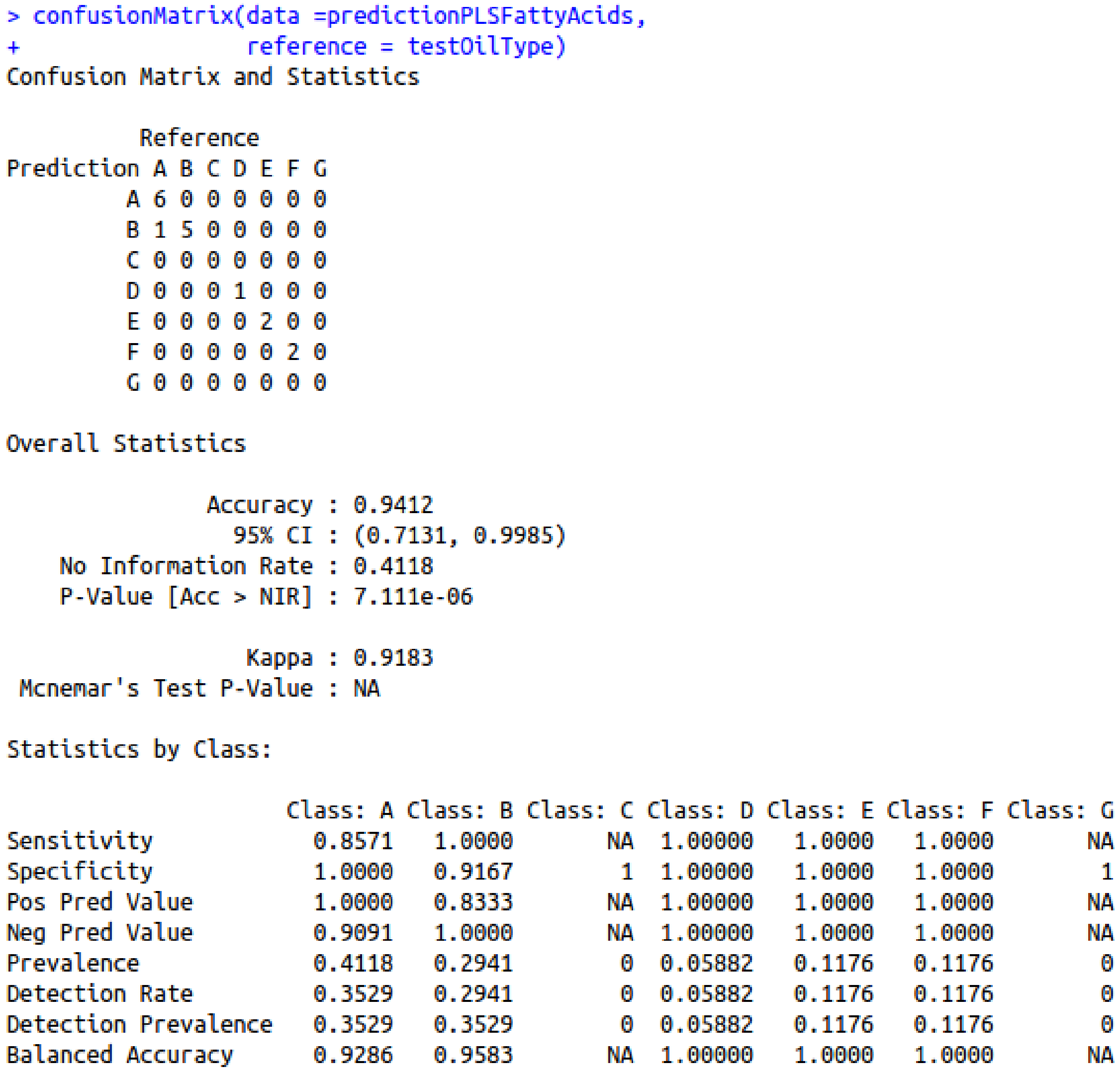
```
> confusionMatrix(data =predictionLRFattyAcids,
+                  reference = testOilType)
Confusion Matrix and Statistics

          Reference
Prediction A B C D E F G
         A 6 1 0 0 0 0 0
         B 1 4 0 0 0 0 0
         C 0 0 0 0 0 0 0
         D 0 0 0 1 0 0 0
         E 0 0 0 0 2 0 0
         F 0 0 0 0 0 2 0
         G 0 0 0 0 0 0 0

Overall Statistics

               Accuracy : 0.8824
                 95% CI : (0.6356, 0.9854)
    No Information Rate : 0.4118
    P-Value [Acc > NIR] : 8.516e-05

                  Kappa : 0.835
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E Class: F Class: G
Sensitivity            0.8571   0.8000       NA  1.00000   1.0000   1.0000       NA
Specificity            0.9000   0.9167        1  1.00000   1.0000   1.0000        1
Pos Pred Value         0.8571   0.8000       NA  1.00000   1.0000   1.0000       NA
Neg Pred Value         0.9000   0.9167       NA  1.00000   1.0000   1.0000       NA
Prevalence             0.4118   0.2941        0  0.05882   0.1176   0.1176        0
Detection Rate         0.3529   0.2353        0  0.05882   0.1176   0.1176        0
Detection Prevalence   0.4118   0.2941        0  0.05882   0.1176   0.1176        0
Balanced Accuracy      0.8786   0.8583       NA  1.00000   1.0000   1.0000       NA
```

ii. **Linear Discriminant Analysis Output for Fatty Acids Predictor:**



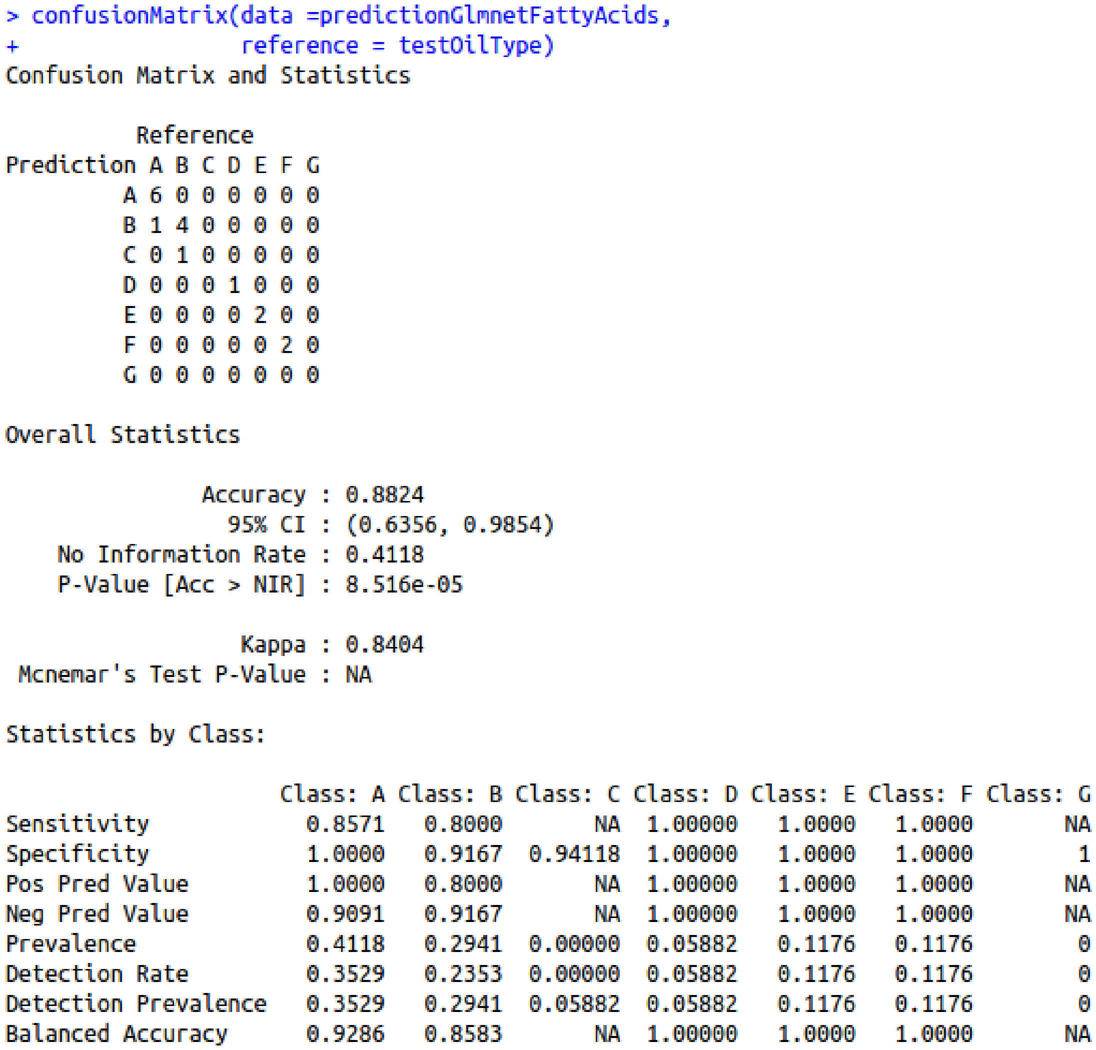
```
> confusionMatrix(data =predictionLDAFattyAcids,
+                 reference = testOilType)
Confusion Matrix and Statistics

          Reference
Prediction A B C D E F G
         A 6 0 0 0 0 0 0
         B 1 4 0 0 0 0 0
         C 0 0 0 0 0 0 0
         D 0 0 0 1 0 0 0
         E 0 0 0 0 2 0 0
         F 0 0 0 0 0 2 0
         G 0 1 0 0 0 0 0

Overall Statistics

               Accuracy : 0.8824
                 95% CI : (0.6356, 0.9854)
    No Information Rate : 0.4118
    P-Value [Acc > NIR] : 8.516e-05

                  Kappa : 0.8404
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E Class: F Class: G
Sensitivity            0.8571   0.8000       NA  1.00000   1.0000   1.0000       NA
Specificity            1.0000   0.9167        1  1.00000   1.0000   1.0000  0.94118
Pos Pred Value         1.0000   0.8000       NA  1.00000   1.0000   1.0000       NA
Neg Pred Value         0.9091   0.9167       NA  1.00000   1.0000   1.0000       NA
Prevalence             0.4118   0.2941        0  0.05882   0.1176   0.1176  0.00000
Detection Rate         0.3529   0.2353        0  0.05882   0.1176   0.1176  0.00000
Detection Prevalence   0.3529   0.2941        0  0.05882   0.1176   0.1176  0.05882
Balanced Accuracy      0.9286   0.8583       NA  1.00000   1.0000   1.0000       NA
```

iii. **Partial Least Square Discriminant Analysis Output for Fatty Acids Predictor:**



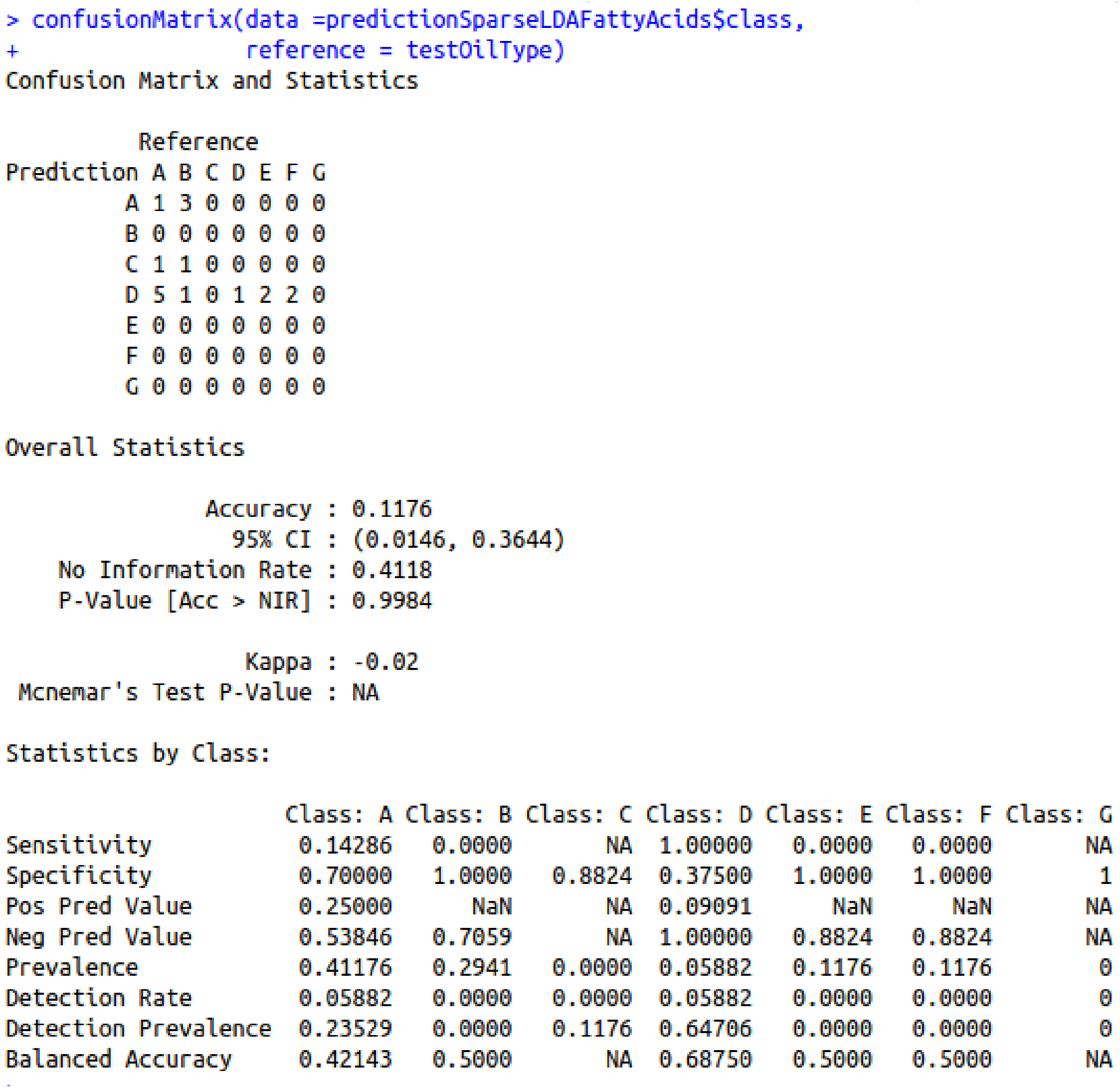
```
> confusionMatrix(data =predictionPLSFattyAcids,
+                 reference = testOilType)
Confusion Matrix and Statistics

          Reference
Prediction A B C D E F G
         A 6 0 0 0 0 0 0
         B 1 5 0 0 0 0 0
         C 0 0 0 0 0 0 0
         D 0 0 0 1 0 0 0
         E 0 0 0 0 2 0 0
         F 0 0 0 0 0 2 0
         G 0 0 0 0 0 0 0

Overall Statistics

               Accuracy : 0.9412
                 95% CI : (0.7131, 0.9985)
    No Information Rate : 0.4118
    P-Value [Acc > NIR] : 7.111e-06

                  Kappa : 0.9183
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E Class: F Class: G
Sensitivity            0.8571   1.0000       NA  1.00000   1.0000   1.0000       NA
Specificity            1.0000   0.9167        1  1.00000   1.0000   1.0000        1
Pos Pred Value         1.0000   0.8333       NA  1.00000   1.0000   1.0000       NA
Neg Pred Value         0.9091   1.0000       NA  1.00000   1.0000   1.0000       NA
Prevalence             0.4118   0.2941        0  0.05882   0.1176   0.1176        0
Detection Rate         0.3529   0.2941        0  0.05882   0.1176   0.1176        0
Detection Prevalence   0.3529   0.3529        0  0.05882   0.1176   0.1176        0
Balanced Accuracy      0.9286   0.9583       NA  1.00000   1.0000   1.0000       NA
```

iv. **Penalized Model for Logistic Regression for Fatty Acids Predictor:**



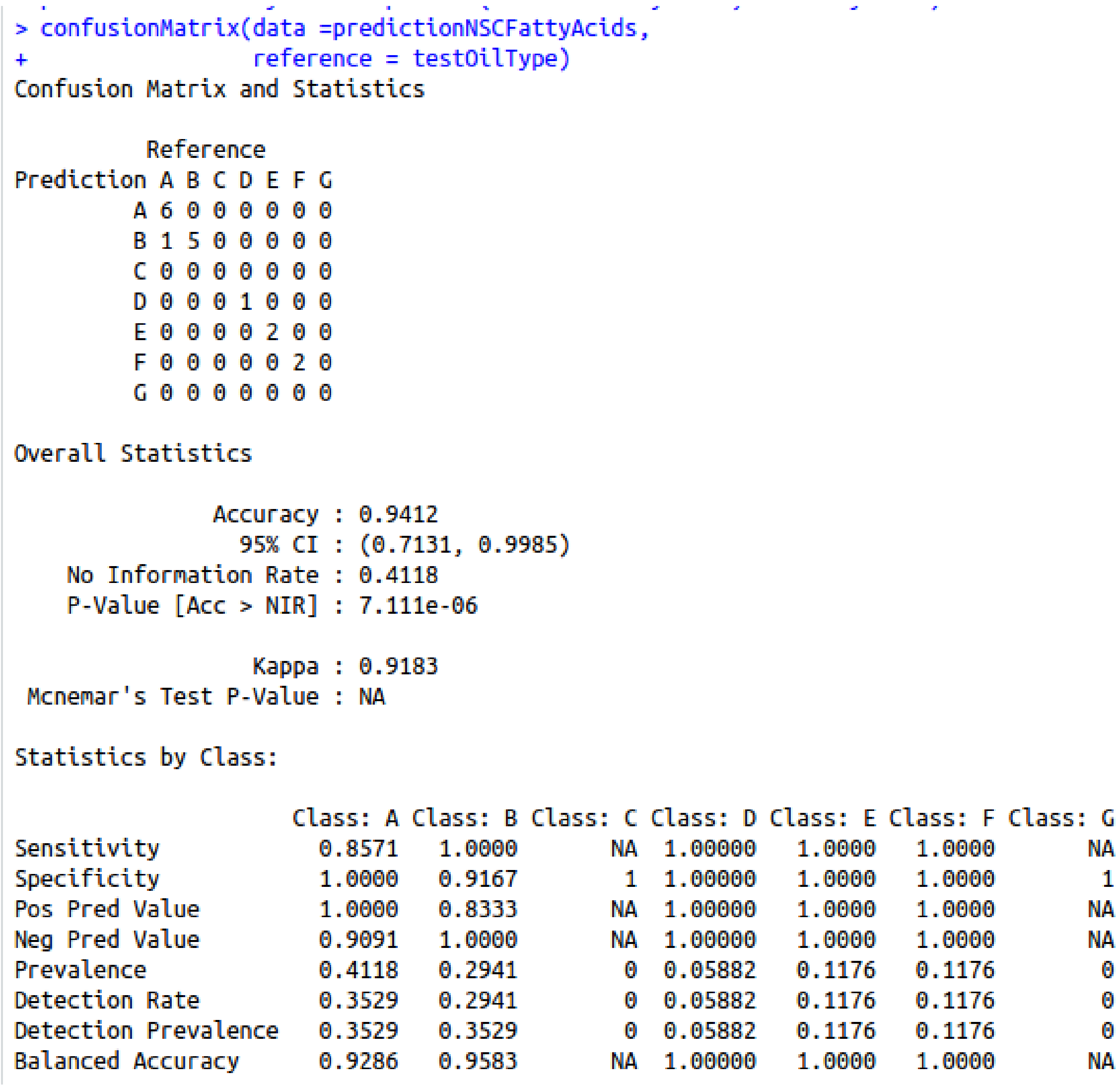
```
> confusionMatrix(data =predictionGlmnetFattyAcids,
+                 reference = testOilType)
Confusion Matrix and Statistics

          Reference
Prediction A B C D E F G
         A 6 0 0 0 0 0 0
         B 1 4 0 0 0 0 0
         C 0 1 0 0 0 0 0
         D 0 0 0 1 0 0 0
         E 0 0 0 0 2 0 0
         F 0 0 0 0 0 2 0
         G 0 0 0 0 0 0 0

Overall Statistics

               Accuracy : 0.8824
                 95% CI : (0.6356, 0.9854)
    No Information Rate : 0.4118
    P-Value [Acc > NIR] : 8.516e-05

                  Kappa : 0.8404
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E Class: F Class: G
Sensitivity            0.8571   0.8000       NA  1.00000   1.0000   1.0000       NA
Specificity            1.0000   0.9167  0.94118  1.00000   1.0000   1.0000        1
Pos Pred Value         1.0000   0.8000       NA  1.00000   1.0000   1.0000       NA
Neg Pred Value         0.9091   0.9167       NA  1.00000   1.0000   1.0000       NA
Prevalence             0.4118   0.2941  0.00000  0.05882   0.1176   0.1176        0
Detection Rate         0.3529   0.2353  0.00000  0.05882   0.1176   0.1176        0
Detection Prevalence   0.3529   0.2941  0.05882  0.05882   0.1176   0.1176        0
Balanced Accuracy      0.9286   0.8583       NA  1.00000   1.0000   1.0000       NA
```

v. **Penalized Model for LDA Output for Fatty Acids Predictor:**



```
> confusionMatrix(data =predictionSparseLDAFattyAcids$class,
+                 reference = testOilType)
Confusion Matrix and Statistics

          Reference
Prediction A B C D E F G
         A 1 3 0 0 0 0 0
         B 0 0 0 0 0 0 0
         C 1 1 0 0 0 0 0
         D 5 1 0 1 2 2 0
         E 0 0 0 0 0 0 0
         F 0 0 0 0 0 0 0
         G 0 0 0 0 0 0 0

Overall Statistics

               Accuracy : 0.1176
                 95% CI : (0.0146, 0.3644)
    No Information Rate : 0.4118
    P-Value [Acc > NIR] : 0.9984

                  Kappa : -0.02
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E Class: F Class: G
Sensitivity           0.14286   0.0000       NA  1.00000   0.0000   0.0000       NA
Specificity           0.70000   1.0000   0.8824  0.37500   1.0000   1.0000        1
Pos Pred Value        0.25000      NaN       NA  0.09091      NaN      NaN       NA
Neg Pred Value        0.53846   0.7059       NA  1.00000   0.8824   0.8824       NA
Prevalence            0.41176   0.2941   0.0000  0.05882   0.1176   0.1176        0
Detection Rate        0.05882   0.0000   0.0000  0.05882   0.0000   0.0000        0
Detection Prevalence  0.23529   0.0000   0.1176  0.64706   0.0000   0.0000        0
Balanced Accuracy     0.42143   0.5000       NA  0.68750   0.5000   0.5000       NA
```

vi. **Nearest Shrinkage Centroids Output for Fatty Acids Predictor:**



```
> confusionMatrix(data =predictionNSCFattyAcids,
+                 reference = testOilType)
Confusion Matrix and Statistics

          Reference
Prediction A B C D E F G
         A 6 0 0 0 0 0 0
         B 1 5 0 0 0 0 0
         C 0 0 0 0 0 0 0
         D 0 0 0 1 0 0 0
         E 0 0 0 0 2 0 0
         F 0 0 0 0 0 2 0
         G 0 0 0 0 0 0 0

Overall Statistics

               Accuracy : 0.9412
                 95% CI : (0.7131, 0.9985)
    No Information Rate : 0.4118
    P-Value [Acc > NIR] : 7.111e-06

                  Kappa : 0.9183
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E Class: F Class: G
Sensitivity            0.8571   1.0000       NA  1.00000   1.0000   1.0000       NA
Specificity            1.0000   0.9167        1  1.00000   1.0000   1.0000        1
Pos Pred Value         1.0000   0.8333       NA  1.00000   1.0000   1.0000       NA
Neg Pred Value         0.9091   1.0000       NA  1.00000   1.0000   1.0000       NA
Prevalence             0.4118   0.2941        0  0.05882   0.1176   0.1176        0
Detection Rate         0.3529   0.2941        0  0.05882   0.1176   0.1176        0
Detection Prevalence   0.3529   0.3529        0  0.05882   0.1176   0.1176        0
Balanced Accuracy      0.9286   0.9583       NA  1.00000   1.0000   1.0000       NA
```

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

Solution 12.2(b)
For more than 2 classes, kappa and accuracy are the best classification statistic. In this exercise, we have 7-classes in response thus, I would choose accuracy as a classification statistic to optimize for this exercise.

Comparison table for different Model's Performance for predicting oilType

#	Models	Accuracy	Kappa
1	Logistic Regression (LR)	0.8824	0.835
2	Linear Discriminant Analysis(LDA)	0.8824	0.8404
3	Partial Least Square Discriminant Analysis(PLS-DA)	0.9412	0.9183
4	Penalized LR	0.8824	0.8404
5	Penalized LDA	0.1176	-0.02
6	Nearest Shrinkage Centroids(NSC)	0.9412	0.9183

The comparison shows that both Partial Least Square Discriminant Analysis(PLS-DA) and Nearest Shrinkage Centroids(NSC) shows best accuracy or kappa value among all other models. So, the I would choose one of those statistic as a classification statistic to optimize for this exercise.

c. Of the models presented in this chapter, which performs best on these data? Which oil type does the model most accurately predict? Least accurately predict?

Solution 12.2(b)
The comparison shows that both Partial Least Square Discriminant Analysis(PLS-DA) and Nearest Shrinkage Centroids(NSC) shows best accuracy or kappa value among all other models
So, one of those models performs best on these data.

Analyzing the result,
The model accurately predicts: Class D, Class E, and Class F.
And, the model least accurately predicts: Class A and Class B
And, the model does not predict at all to: Class C and Class G

```
###################################
# question 12.1 for bio predictor
###################################
 library(caret)
library(AppliedPredictiveModeling)

data(hepatic)
# use ?hepatic to see more details


library(MASS)
set.seed(975)

barplot(table(injury),col=c(”yellow”,”red”,”green”), main=”Class Distribution”)

#----------------------------------------------------------------------
# Use the biological predictors:
#----------------------------------------------------------------------


#this gives Z114 predictor has zero-variance
nearZeroVar(bio)

#remove the Z114 predictor and then find the correlation between the predictors
noZVbio ¡- bio[,-114]

#remove the correlation between the predictors
highCorBio¡-findCorrelation(cor(noZVbio),cutoff = .75)
filteredCorBio ¡- noZVbio[,-highCorBio]

# splitting data into 75% and 25% based on injury response
set.seed(975)
trainingRows =  createDataPartition(injury, p = .75, list= FALSE)

trainBio ¡- filteredCorBio[ trainingRows, ]
testBio ¡- filteredCorBio[-trainingRows, ]


trainInjury ¡- injury[trainingRows]
testInjury ¡- injury[-trainingRows]


ctrl ¡- trainControl(summaryFunction = defaultSummary)

############ Logistic Regression Analysis #############
# logistic regression

library(caret)
set.seed(975)
lrBio ¡- train(x=trainBio,
               y = trainInjury,
               method = ”multinom”,
               metric = ”Accuracy”,
               trControl = ctrl)


predictionLRBio¡-predict(lrBio,testBio)

confusionMatrix(data =predictionLRBio,
                reference = testInjury)

#######################################################
############ Linear Discriminant Analysis #############
```

```
# LDA Analysis
library(MASS)
set.seed(975)

ldaBio ¡- train(x = trainBio,
                y = trainInjury,
                method = ”lda”,
                metric = ”Accuracy”,
                trControl = ctrl)

predictionLDABio ¡- predict(ldaBio,testBio)
confusionMatrix(data =predictionLDABio,
                reference = testInjury)
##########################################################################

############## Partial Least Squares Discriminant Analysis ###############
library(MASS)
set.seed(975)
plsBio ¡- train(x = trainBio,
                y = trainInjury,
                method = ”pls”,
                tuneGrid = expand.grid(.ncomp = 1:1),
                # preProc = c(”center”,”scale”),
                metric = ”Accuracy”,
                trControl = ctrl)

predictionPLSBio ¡-predict(plsBio,testBio)
confusionMatrix(data =predictionPLSBio,
                reference = testInjury)

#######################################################
########### Penalized Models ###########

########### Penalized Models for Logistic Regression ###########
glmnGrid ¡- expand.grid(.alpha = c(0, .1, .2, .4),
                        .lambda = seq(.01, .2, length = 10))
set.seed(975)

glmnTunedLRBio ¡- train(x=trainBio,
                      y =trainInjury,
                      method = ”glmnet”,
                      tuneGrid = glmnGrid,
                      # preProc = c(”center”, ”scale”),
                      metric = ”Accuracy”,
                      trControl = ctrl)

predictionGlmnetBio ¡-  predict(glmnTunedLRBio,testBio)
confusionMatrix(data =predictionGlmnetBio,
                reference = testInjury)


########### Penalized Models for LDA ###########
library(sparseLDA)
set.seed(975)
sparseLdaModelBio ¡- sda(x=trainBio,
                          y =trainInjury,
                          lambda = 0.01,
                          stop = -146)
## the ridge parameter called lambda.

predictionSparseLDABio ¡-  predict(sparseLdaModelBio,testBio)
confusionMatrix(data =predictionSparseLDABio$class,
                reference = testInjury)
```

```
#######################################################
########### Nearest Shrunken Centroids ###########

library(pamr)
nscGridBio ¡- data.frame(.threshold = seq(0,4, by=0.1))
set.seed(476)
nscTunedBio ¡- train(x = trainBio,
                     y = trainInjury,
                     method = ”pam”,
                     # preProc = c(”center”, ”scale”),
                     tuneGrid = nscGridBio,
                     metric = ”Accuracy”,
                     trControl = ctrl)

predictionNSCBio ¡-predict(nscTunedBio,testBio)
confusionMatrix(data =predictionNSCBio,
                reference = testInjury)



###################################
###################################

###################################
# question 12.1 for chem predictor
###################################
library(caret)
library(AppliedPredictiveModeling)

data(hepatic)
# use ?hepatic to see more details


library(MASS)
set.seed(975)

barplot(table(injury),col=c(”yellow”,”red”,”green”), main=”Class Distribution”)


set.seed(975)

#------------------------------------------------------------------------
# Use the Chemical predictors:
#------------------------------------------------------------------------


# this gives removes near-zero variance
# this is a categorical predictor and should remove near zero variance for this data
zv cols = nearZeroVar(chem)
noZVChem = chem[,-zv cols]


#remove the correlation between the predictors
highCorChem¡-findCorrelation(cor(noZVChem),cutoff = .75)
filteredCorChem ¡- noZVChem[,-highCorChem]



# splitting data into 75% and 25% based on injury response
set.seed(975)
trainingRows =  createDataPartition(injury, p = .75, list= FALSE)

trainChem ¡- filteredCorChem[trainingRows,]
testChem ¡- filteredCorChem[-trainingRows, ]
```

```
trainInjury ¡- injury[trainingRows]
testInjury ¡- injury[-trainingRows]


ctrl ¡- trainControl(summaryFunction = defaultSummary)

############ Logistic Regression Analysis #############
# logistic regression

library(caret)
set.seed(975)
lrChem ¡- train(x=trainChem,
               y = trainInjury,
               method = ”multinom”,
               metric = ”Accuracy”,
               trControl = ctrl)


predictionLRChem¡-predict(lrChem,testChem)

confusionMatrix(data =predictionLRChem,
                reference = testInjury)

#######################################################
############ Linear Discriminant Analysis #############

# LDA Analysis
library(MASS)
set.seed(975)

ldaChem ¡- train(x = trainChem,
                y = trainInjury,
                method = ”lda”,
                preProc = c(”center”,”scale”),
                metric = ”Accuracy”,
                trControl = ctrl)

predictionLDAChem ¡-predict(ldaChem,testChem)
confusionMatrix(data =predictionLDAChem,
                reference = testInjury)
##########################################################################

############## Partial Least Squares Discriminant Analysis ###############
library(MASS)
set.seed(975)
plsChem ¡- train(x = trainChem,
                y = trainInjury,
                method = ”pls”,
                tuneGrid = expand.grid(.ncomp = 1:1),
                preProc = c(”center”,”scale”),
                metric = ”Accuracy”,
                trControl = ctrl)

predictionPLSChem ¡-predict(plsChem,testChem)
confusionMatrix(data =predictionPLSChem,
                reference = testInjury)

#######################################################
########### Penalized Models ###########

########### Penalized Models for Logistic Regression ###########

glmnGrid ¡- expand.grid(.alpha = c(0, .1, .2, .4),
                        .lambda = seq(.01, .2, length = 10))
```

```
set.seed(975)
glmnTunedChem ¡- train(x=trainChem,
                      y =trainInjury,
                      method = ”glmnet”,
                      tuneGrid = glmnGrid,
                      preProc = c(”center”, ”scale”),
                      metric = ”Accuracy”,
                      trControl = ctrl)

predictionGlmnetChem ¡-  predict(glmnTunedChem,testChem)
confusionMatrix(data =predictionGlmnetChem,
                reference = testInjury)


########### Penalized Models for LDA ###########
library(sparseLDA)
set.seed(975)
sparseLdaModelChem ¡- sda(x=trainChem,
                      y =trainInjury,
                      lambda = 0.01,
                      stop = -73)
## the ridge parameter called lambda.

predictionSparseLDAChem ¡-  predict(sparseLdaModelChem,testChem)
confusionMatrix(data = predictionSparseLDAChem$class,
                reference = testInjury)

#######################################################
########### Nearest Shrunken Centroids ###########

library(pamr)

nscGridChem ¡- data.frame(.threshold = seq(0,4, by=0.1))
set.seed(975)
nscTunedChem ¡- train(x = trainChem,
                     y = trainInjury,
                     method = ”pam”,
                     preProc = c(”center”, ”scale”),
                     tuneGrid = nscGridChem,
                     metric = ”Accuracy”,
                     trControl = ctrl)

predictionNSCChem ¡-predict(nscTunedChem,testChem)
confusionMatrix(data =predictionNSCChem,
                reference = testInjury)



#



###################################
###################################

###################################
# question 12.1 for merged predictor
###################################

library(caret)
library(AppliedPredictiveModeling)

data(hepatic)
# use ?hepatic to see more details
```

```
library(MASS)
set.seed(975)

#this gives Z114 predictor has zero-variance
nearZeroVar(bio)

#remove the Z114 predictor and then find the correlation between the predictors
noZVbio ¡- bio[,-114]

#remove the correlation between the predictors
highCorBio¡-findCorrelation(cor(noZVbio),cutoff = .75)
filteredCorBio ¡- noZVbio[,-highCorBio]


# this gives removes near-zero variance
# this is a categorical predictor and should remove near zero variance for this data
zv cols = nearZeroVar(chem)
noZVChem = chem[,-zv cols]


#remove the correlation between the predictors
highCorChem¡-findCorrelation(cor(noZVChem),cutoff = .75)
filteredCorChem ¡- noZVChem[,-highCorChem]

mergedPredictor ¡-data.frame(filteredCorBio,filteredCorChem)

# splitting data into 75% and 25% based on injury response
set.seed(975)
trainingRows =  createDataPartition(injury, p = .75, list= FALSE)

trainmergedPredictor ¡- mergedPredictor[trainingRows,]
testmergedPredictor ¡- mergedPredictor[-trainingRows, ]

trainInjury ¡- injury[trainingRows]
testInjury ¡- injury[-trainingRows]


ctrl ¡- trainControl(summaryFunction = defaultSummary)

############ Logistic Regression Analysis #############
# logistic regression

library(caret)
set.seed(975)
lrmergedPredictor ¡- train(x=trainmergedPredictor,
                y = trainInjury,
                method = ”multinom”,
                metric = ”Accuracy”,
                trControl = ctrl)


predictionLRmergedPredictor¡-predict(lrmergedPredictor,testmergedPredictor)

confusionMatrix(data =predictionLRmergedPredictor,
                reference = testInjury)

#######################################################
############ Linear Discriminant Analysis #############

# LDA Analysis
library(MASS)
set.seed(975)

ldamergedPredictor ¡- train(x = trainmergedPredictor,
                 y = trainInjury,
                 method = ”lda”,
                 # preProc = c(”center”,”scale”),
                 metric = ”Accuracy”,
                 trControl = ctrl)

predictionLDAmergedPredictor ¡-predict(ldamergedPredictor,testmergedPredictor)
confusionMatrix(data =predictionLDAmergedPredictor,
                reference = testInjury)
###########################################################################

############### Partial Least Squares Discriminant Analysis ###############
library(MASS)
set.seed(975)
plsmergedPredictor ¡- train(x = trainmergedPredictor,
                 y = trainInjury,
                 method = ”pls”,
                 tuneGrid = expand.grid(.ncomp = 1:4),
                 # preProc = c(”center”,”scale”),
                 metric = ”Accuracy”,
                 trControl = ctrl)

predictionPLSmergedPredictor ¡-predict(plsmergedPredictor,testmergedPredictor)
confusionMatrix(data =predictionPLSmergedPredictor,
                reference = testInjury)

########################################################
########### Penalized Models ###########

########### Penalized Models for Logistic Regression ###########

glmnGrid ¡- expand.grid(.alpha = c(0, .1, .2, .4, .6, .8, 1),
                        .lambda = seq(.01, .2, length = 10))
set.seed(975)
glmnTunedmergedPredictor ¡- train(x=trainmergedPredictor,
                       y =trainInjury,
                       method = ”glmnet”,
                       tuneGrid = glmnGrid,
                       # preProc = c(”center”, ”scale”),
                       metric = ”Accuracy”,
                       trControl = ctrl)

varImp(glmnTunedmergedPredictor,scale = FALSE)
predictionGlmnetmergedPredictor ¡-  predict(glmnTunedmergedPredictor,testmergedPredictor)
confusionMatrix(data =predictionGlmnetmergedPredictor,
                reference = testInjury)



########### Penalized Models for LDA ###########
library(sparseLDA)
set.seed(975)
sparseLdaModelmergedPredictor ¡- sda(x=trainmergedPredictor,
                          y =trainInjury,
                          lambda = 0.01,
                          stop = -219)
## the ridge parameter called lambda.

predictionSparseLDAmergedPredictor ¡-  predict(sparseLdaModelmergedPredictor,testmergedPredictor)
confusionMatrix(data = predictionSparseLDAmergedPredictor$class,
                reference = testInjury)

########################################################
########### Nearest Shrunken Centroids ###########
```

```
library(pamr)

nscGridmergedPredictor ¡- data.frame(.threshold = seq(0,4, by=0.1))
set.seed(975)
nscTunedmergedPredictor ¡- train(x = trainmergedPredictor,
                      y = trainInjury,
                      method = ”pam”,
                      # preProc = c(”center”, ”scale”),
                      tuneGrid = nscGridmergedPredictor,
                      metric = ”Accuracy”,
                      trControl = ctrl)

varImp(nscTunedmergedPredictor,scale = FALSE)
predictionNSCmergedPredictor ¡-predict(nscTunedmergedPredictor,testmergedPredictor)
confusionMatrix(data =predictionNSCmergedPredictor,
                reference = testInjury)



###################################
###################################

###################################
# question 12.2 for fatty acid predictor
###################################
 library(caret)
library(AppliedPredictiveModeling)

data(oil)
# use ?hepatic to see more details


library(MASS)
set.seed(975)

barplot(table(oilType),col=c(”yellow”), main=”Class Distribution”)



#this gives 0 predictor with zero-variance
nearZeroVar(fattyAcids,saveMetrics =TRUE)

#remove the correlation between the predictors
highCorM¡-findCorrelation(cor(fattyAcids),cutoff = .75)
filteredCorFatty ¡- fattyAcids[,-highCorM]

# after removing the highly correlated predictor, we split the data using
# stratified random sampling

# splitting data into 80% and 20% based on oilType response

set.seed(975)
trainingRows =  createDataPartition(oilType, p = .80, list= FALSE)

trainFattyAcids ¡- filteredCorFatty[ trainingRows, ]
testFattyAcids ¡- filteredCorFatty[-trainingRows, ]

trainOilType ¡- oilType[trainingRows]
testOilType ¡- oilType[-trainingRows]

ctrl ¡- trainControl(summaryFunction = defaultSummary)

############ Logistic Regression Analysis #############
# logistic regression
```

```
library(caret)
set.seed(975)
lrFattyAcids ¡- train(x=trainFattyAcids,
               y = trainOilType,
               method = ”multinom”,
               metric = ”Accuracy”,
               trControl = ctrl)


predictionLRFattyAcids¡-predict(lrFattyAcids,testFattyAcids)

confusionMatrix(data =predictionLRFattyAcids,
                reference = testOilType)

#######################################################
############ Linear Discriminant Analysis #############

# LDA Analysis
library(MASS)
set.seed(975)


ldaFattyAcids ¡- train(x = trainFattyAcids,
                y = trainOilType,
                method = ”lda”,
                metric = ”Accuracy”,
                trControl = ctrl)

predictionLDAFattyAcids ¡-predict(ldaFattyAcids,testFattyAcids)
confusionMatrix(data =predictionLDAFattyAcids,
                reference = testOilType)
##########################################################################

############## Partial Least Squares Discriminant Analysis ###############
library(MASS)
set.seed(975)
plsFattyAcids ¡- train(x = trainFattyAcids,
                y = trainOilType,
                method = ”pls”,
                tuneGrid = expand.grid(.ncomp = 1:4),
                # preProc = c(”center”,”scale”),
                metric = ”Accuracy”,
                trControl = ctrl)

predictionPLSFattyAcids ¡-predict(plsFattyAcids,testFattyAcids)
confusionMatrix(data =predictionPLSFattyAcids,
                reference = testOilType)

#######################################################
########### Penalized Models ###########

########### Penalized Models for Logistic Regression ###########
# glmnGrid ¡- expand.grid(.alpha = c(0, .1, .2, .4),
#                          .lambda = seq(.01, .2, length = 10))
glmnGrid ¡- expand.grid(.alpha = c(0, .1, .2, .4, .6, .8, 1),
                        .lambda = seq(.01, .2, length = 10))
set.seed(476)

glmnTunedLRFattyAcids¡- train(x=trainFattyAcids,
                        y =trainOilType,
                        method = ”glmnet”,
                        tuneGrid = glmnGrid,
                        # preProc = c(”center”, ”scale”),
                        metric = ”Accuracy”,
                        trControl = ctrl)
```

```
predictionGlmnetFattyAcids ¡-  predict(glmnTunedLRFattyAcids,testFattyAcids)
confusionMatrix(data =predictionGlmnetFattyAcids,
                reference = testOilType)


########### Penalized Models for LDA ###########
library(sparseLDA)
set.seed(975)
sparseLdaModelFattyAcids ¡- sda(x=trainFattyAcids,
                         y =trainOilType,
                         lambda = 0.01,
                         stop = -7)
## the ridge parameter called lambda.

predictionSparseLDAFattyAcids ¡-  predict(sparseLdaModelFattyAcids,testFattyAcids)
confusionMatrix(data =predictionSparseLDAFattyAcids$class,
                reference = testOilType)



#######################################################
########### Nearest Shrunken Centroids ###########

library(pamr)
nscGridFattyAcids ¡- data.frame(.threshold = seq(0,4, by=0.1))
set.seed(975)
nscTunedFattyAcids ¡- train(x = trainFattyAcids,
                     y = trainOilType,
                     method = ”pam”,
                     # preProc = c(”center”, ”scale”),
                     tuneGrid = nscGridFattyAcids,
                     metric = ”Accuracy”,
                     trControl = ctrl)

predictionNSCFattyAcids ¡-predict(nscTunedFattyAcids,testFattyAcids)
confusionMatrix(data =predictionNSCFattyAcids,
                reference = testOilType)



###################################
###################################
```

References:

1. Applied Predictive Modeling : @authors Max Kuhn. Kjell Johnson
2. https://archive.ics.uci.edu/ml/index.php