MA4790

Homework 5

Ian Boulis

12.1

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

This data is very similar in classification to my group project data, so a strategy similar to the one we used on our data would be effective. Stratified random sampling would be able to deal with the imbalances and split the training and testing set evenly according to the predictors.

B. Which Classification statistic would you choose to optimize for this exercise and why?

Since this is not a numerical data set, the best statistic to optimize for it would be accuracy.

C. Split the data into a testing a training set and fit the models from the chapter to them.

```
Generalized Linear Model

281 samples
102 predictors
  2 classes: 'yes', 'none'

Pre-processing: centered (102), scaled (102)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 281, 281, 281, 281, 281, 281, ...
Resampling results:

  Accuracy   Kappa
  0.5264184  0.0117997
```

```
Linear Discriminant Analysis

281 samples
102 predictors
  2 classes: 'yes', 'none'

Pre-processing: centered (102), scaled (102)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 212, 212, 212, 212, 212, 212, ...
Resampling results:

  Accuracy   Kappa
  0.5524638  0.03389064
```

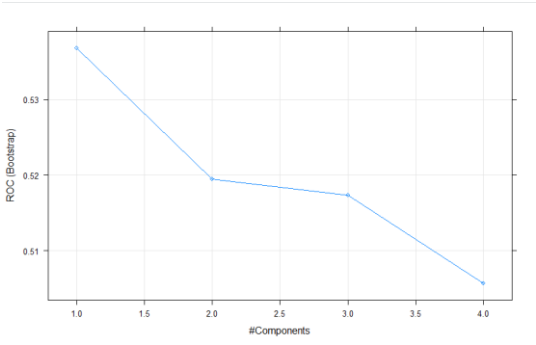
```
Partial Least Squares

281 samples
102 predictors
  2 classes: 'yes', 'none'

Pre-processing: centered (102), scaled (102)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 281, 281, 281, 281, 281, 281, ...
Resampling results across tuning parameters:

  ncomp  ROC        Sens       Spec
  1      0.5368276  0.8681982  0.1888075
  2      0.5194621  0.8105182  0.2416225
  3      0.5173082  0.7738291  0.2788836
  4      0.5056225  0.7236948  0.3071026

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



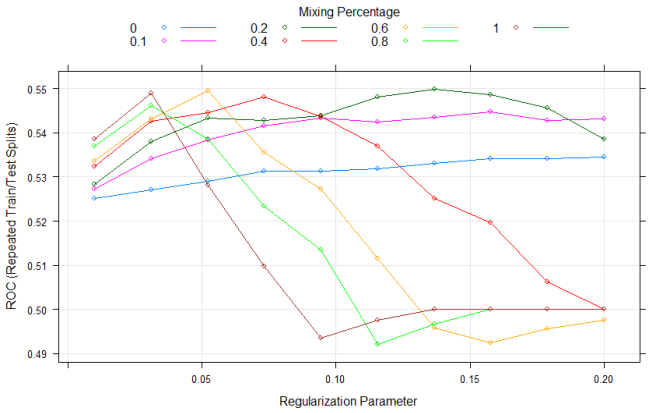
```
glmnet

281 samples
102 predictors
  2 classes: 'yes', 'none'

Pre-processing: centered (102), scaled (102)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 212, 212, 212, 212, 212, 212, ...
Resampling results across tuning parameters:

  alpha  lambda      ROC        Sens       Spec
  0.0    0.01000000  0.5251878  0.6837209  0.370769231
  0.0    0.03111111  0.5270125  0.7302326  0.330769231
  0.0    0.05222222  0.5290877  0.7534884  0.310769231
  0.0    0.07333333  0.5312701  0.7758140  0.296923077
  0.0    0.09444444  0.5312701  0.7916279  0.283076923
  0.0    0.11555556  0.5317710  0.8139535  0.264615385
  0.0    0.13666667  0.5331664  0.8232558  0.258461538
  0.0    0.15777778  0.5341682  0.8353488  0.250769231
  0.0    0.17888889  0.5340966  0.8465116  0.236923077
  0.0    0.20000000  0.5345259  0.8567442  0.220000000
  0.1    0.01000000  0.5273345  0.6976744  0.364615385
  0.1    0.03111111  0.5341682  0.7525581  0.333846154
  0.1    0.05222222  0.5383542  0.7906977  0.290769231
  0.1    0.07333333  0.5415742  0.8232558  0.266153846
  0.1    0.09444444  0.5433631  0.8530233  0.236923077
  0.1    0.11555556  0.5425224  0.8753488  0.220000000
  0.1    0.13666667  0.5435242  0.8874419  0.190769231
  0.1    0.15777778  0.5448301  0.8948837  0.176923077
  0.1    0.17888889  0.5428801  0.8995349  0.164615385
  0.1    0.20000000  0.5431843  0.9097674  0.152307692
  0.2    0.01000000  0.5283721  0.7013953  0.356923077
  0.2    0.03111111  0.5380322  0.7683721  0.307692308
  0.2    0.05222222  0.5434347  0.8195349  0.270769231
  0.2    0.07333333  0.5428444  0.8688372  0.232307692
  0.2    0.09444444  0.5438104  0.8920930  0.192307692
  0.2    0.11555556  0.5480680  0.9060465  0.170769231
  0.2    0.13666667  0.5499463  0.9144186  0.147692308
  0.2    0.15777778  0.5486762  0.9302326  0.115384615
  0.2    0.17888889  0.5457245  0.9432558  0.090769231
  0.2    0.20000000  0.5385689  0.9553488  0.056923077
  0.4    0.01000000  0.5323077  0.7162791  0.352307692
  0.4    0.03111111  0.5426655  0.8139535  0.273846154
  0.4    0.05222222  0.5445617  0.8855814  0.207692308
  0.4    0.07333333  0.5481574  0.9106977  0.170769231
  0.4    0.09444444  0.5437030  0.9330233  0.115384615
  0.4    0.11555556  0.5369052  0.9562791  0.060000000
  0.4    0.13666667  0.5250984  0.9832558  0.029230769
  0.4    0.15777778  0.5197138  0.9897674  0.004615385
  0.4    0.17888889  0.5062612  0.9953488  0.001538462
  0.4    0.20000000  0.4999642  0.9972093  0.000000000
  0.6    0.01000000  0.5337030  0.7283721  0.350769231
  0.6    0.03111111  0.5431664  0.8586047  0.247692308
  0.6    0.05222222  0.5494991  0.9088372  0.167692308
  0.6    0.07333333  0.5355277  0.9441860  0.092307692
  0.6    0.09444444  0.5272987  0.9767442  0.032307692
  0.6    0.11555556  0.5115921  0.9925581  0.003076923
  0.6    0.13666667  0.4958140  0.9972093  0.000000000
  0.6    0.15777778  0.4924508  0.9990698  0.000000000
  0.6    0.17888889  0.4955993  1.0000000  0.000000000
  0.6    0.20000000  0.4974776  1.0000000  0.000000000
  0.8    0.01000000  0.5370304  0.7432558  0.333846154
  0.8    0.03111111  0.5461360  0.8865116  0.212307692
  0.8    0.05222222  0.5385689  0.9376744  0.115384615
  0.8    0.07333333  0.5234347  0.9786047  0.032307692
  0.8    0.09444444  0.5135242  0.9944186  0.001538462
  0.8    0.11555556  0.4920572  0.9990698  0.000000000
  0.8    0.13666667  0.4966726  1.0000000  0.000000000
  0.8    0.15777778  0.5000000  1.0000000  0.000000000
  0.8    0.17888889  0.5000000  1.0000000  0.000000000
  0.8    0.20000000  0.5000000  1.0000000  0.000000000
  1.0    0.01000000  0.5385331  0.7562791  0.318461538
  1.0    0.03111111  0.5489803  0.9023256  0.178461538
  1.0    0.05222222  0.5281038  0.9534884  0.063076923
  1.0    0.07333333  0.5098032  0.9916279  0.003076923
  1.0    0.09444444  0.4934347  0.9990698  0.000000000
  1.0    0.11555556  0.4974776  1.0000000  0.000000000
  1.0    0.13666667  0.5000000  1.0000000  0.000000000
  1.0    0.15777778  0.5000000  1.0000000  0.000000000
  1.0    0.17888889  0.5000000  1.0000000  0.000000000
  1.0    0.20000000  0.5000000  1.0000000  0.000000000

ROC was used to select the optimal model using the largest value.
The final values used for the model were alpha = 0.2 and lambda = 0.1366667.
```



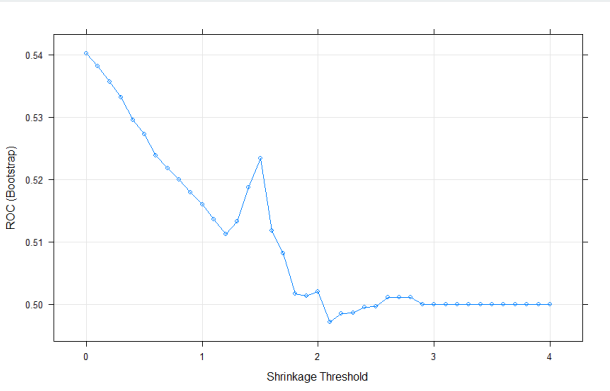
```
Nearest Shrunken Centroids

281 samples
102 predictors
  2 classes: 'yes', 'none'

Pre-processing: centered (102), scaled (102)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 281, 281, 281, 281, 281, 281, ...
Resampling results across tuning parameters:

  threshold  ROC        Sens       Spec
  0.0        0.5402133  0.8367906  0.263601340
  0.1        0.5381870  0.8405494  0.252668177
  0.2        0.5356590  0.8490078  0.241201131
  0.3        0.5331103  0.8556621  0.227095065
  0.4        0.5294808  0.8654770  0.215691253
  0.5        0.5272218  0.8747519  0.211502361
  0.6        0.5238256  0.8866149  0.190658734
  0.7        0.5218102  0.8990402  0.164001041
  0.8        0.5199318  0.9115123  0.144168736
  0.9        0.5179055  0.9215565  0.123717487
  1.0        0.5159564  0.9324508  0.101111253
  1.1        0.5136590  0.9418170  0.085847080
  1.2        0.5112015  0.9510296  0.069105020
  1.3        0.5133070  0.9619958  0.050616832
  1.4        0.5186862  0.9706002  0.036038752
  1.5        0.5233880  0.9808161  0.020207906
  1.6        0.5117495  0.9872492  0.013523726
  1.7        0.5081909  0.9907246  0.010391363
  1.8        0.5016766  0.9918841  0.008340081
  1.9        0.5013062  0.9947826  0.004210526
  2.0        0.5020751  0.9959420  0.003157895
  2.1        0.4971326  0.9971014  0.003157895
  2.2        0.4984833  0.9994203  0.001052632
  2.3        0.4986159  1.0000000  0.000000000
  2.4        0.4995542  1.0000000  0.000000000
  2.5        0.4997023  1.0000000  0.000000000
  2.6        0.5011745  1.0000000  0.000000000
  2.7        0.5011823  1.0000000  0.000000000
  2.8        0.5011823  1.0000000  0.000000000
  2.9        0.5000000  1.0000000  0.000000000
  3.0        0.5000000  1.0000000  0.000000000
  3.1        0.5000000  1.0000000  0.000000000
  3.2        0.5000000  1.0000000  0.000000000
  3.3        0.5000000  1.0000000  0.000000000
  3.4        0.5000000  1.0000000  0.000000000
  3.5        0.5000000  1.0000000  0.000000000
  3.6        0.5000000  1.0000000  0.000000000
  3.7        0.5000000  1.0000000  0.000000000
  3.8        0.5000000  1.0000000  0.000000000
  3.9        0.5000000  1.0000000  0.000000000
  4.0        0.5000000  1.0000000  0.000000000

ROC was used to select the optimal model using the largest value.
The final value used for the model was threshold = 0.
```



Model	Statistic	Parameters
Logistic Regression	Accuracy = 0.5264184	Kappa = 0.0117997
Linear Discriminant Analysis	Accuracy = 0.5524638	Kappa = 0.03389064
Partial Least Squares Dis. Analysis	ROC = 0.5368276	Ncomp = 1
Penalized Models	ROC = 0.5499463	Alpha = 0.2, Lambda = 0.13366667
Nearest Shrunken Centroid	ROC = 0.5402133	Threshold = 0

D. For the optimal model, what are the top five important predictors

I believe that the LDA model was the best model for this data set. The results from the varImp function are:

```
     Importance
Z130     100.00
Z64       97.89
Z118      81.62
Z48       75.01
Z40       72.96
```

12.3

A. Explore the data, are there important features of the predictor data themselves such as between predictor correlations or degenerate distributions?

Most of the distributions seem too be quite varied. In the response variable there were a considerable amount of “yes” over “no”. There were some predictor correlations, however, in a set this big there are bound to be some correlations between every aspect of the data.

B. What criteria should be used to evaluate the effectiveness of the models?

With this data set all the models can be made to produce a ROC value. So, the best criteria to use for comparing the effectiveness of the different models would be ROC.

C. Split the data into a testing a training set and fit the models from the chapter to them.

```
Generalized Linear Model

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

Pre-processing: centered (14), scaled (14)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 4001, 4001, 4001, 4001, 4001, 4001, ...
Resampling results:

  ROC        Sens       Spec
  0.7599604  0.9827956  0.09860941
```

```
Linear Discriminant Analysis

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

Pre-processing: centered (14), scaled (14)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 4001, 4001, 4001, 4001, 4001, 4001, ...
Resampling results:

  ROC        Sens       Spec
  0.7690299  0.9804818  0.09965254
```

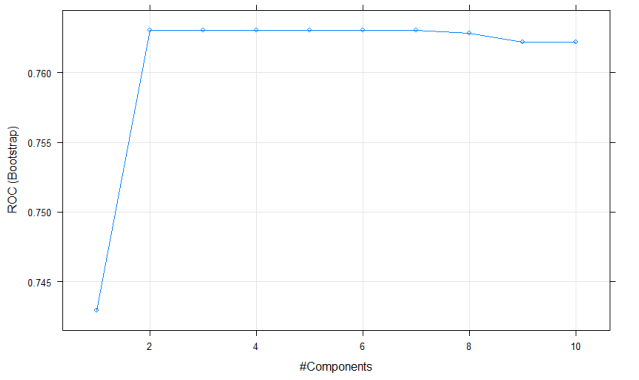
```
Partial Least Squares

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

Pre-processing: centered (14), scaled (14)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 4001, 4001, 4001, 4001, 4001, 4001, ...
Resampling results across tuning parameters:

  ncomp  ROC        Sens       Spec
   1     0.7429193  0.9997819  0.01122525
   2     0.7630224  0.9976483  0.02254283
   3     0.7630454  0.9977426  0.02314713
   4     0.7630556  0.9977743  0.02314713
   5     0.7630542  0.9977743  0.02314713
   6     0.7630482  0.9977743  0.02314713
   7     0.7630318  0.9977731  0.02314713
   8     0.7628558  0.9977096  0.02333852
   9     0.7621776  0.9976154  0.02314328
  10     0.7621727  0.9975847  0.02294816

ROC was used to select the optimal model using the largest value.
The final value used for the model was ncomp = 4.
```



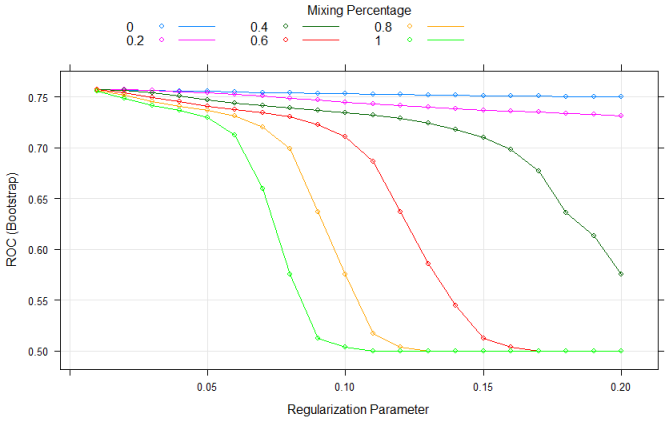
```
glmnet

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

Pre-processing: centered (14), scaled (14)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 4001, 4001, 4001, 4001, 4001, 4001, ...
Resampling results across tuning parameters:

  alpha  lambda  ROC        Sens       Spec
  0.0    0.01    0.7572109  0.9881572  0.0702804615
  0.0    0.02    0.7567389  0.9914235  0.0589441324
  0.0    0.03    0.7562621  0.9934882  0.0470412018
  0.0    0.04    0.7557958  0.9949788  0.0398451658
  0.0    0.05    0.7553366  0.9962813  0.0329932872
  0.0    0.06    0.7548877  0.9973956  0.0289286869
  0.0    0.07    0.7544167  0.9980292  0.0240003632
  0.0    0.08    0.7539969  0.9987600  0.0185452145
  0.0    0.09    0.7535781  0.9990145  0.0158571956
  0.0    0.10    0.7531650  0.9993344  0.0131917614
  0.0    0.11    0.7527680  0.9994903  0.0122334369
  0.0    0.12    0.7523769  0.9996494  0.0093693766
  0.0    0.13    0.7520116  0.9997781  0.0074601317
  0.0    0.14    0.7516643  0.9998729  0.0059185125
  0.0    0.15    0.7513168  0.9999047  0.0043552604
  0.0    0.16    0.7510337  0.9999373  0.0039715475
  0.0    0.17    0.7507204  1.0000000  0.0037810713
  0.0    0.18    0.7504112  1.0000000  0.0031946549
  0.0    0.19    0.7501043  1.0000000  0.0031946549
  0.0    0.20    0.7498208  1.0000000  0.0031946549
  0.2    0.01    0.7574169  0.9897777  0.0648923977
  0.2    0.02    0.7568992  0.9933595  0.0474792178
  0.2    0.03    0.7560661  0.9959284  0.0349044463
  0.2    0.04    0.7549582  0.9978062  0.0247537752
  0.2    0.05    0.7536924  0.9985718  0.0185444546
  0.2    0.06    0.7522295  0.9990780  0.0124193684
  0.2    0.07    0.7505432  0.9995235  0.0091543986
  0.2    0.08    0.7486990  0.9996182  0.0064826541
  0.2    0.09    0.7468172  1.0000000  0.0043530494
  0.2    0.10    0.7448574  1.0000000  0.0024722080
  0.2    0.11    0.7431757  1.0000000  0.0014935047
  0.2    0.12    0.7417186  1.0000000  0.0005751716
  0.2    0.13    0.7401902  1.0000000  0.0003846954
  0.2    0.14    0.7385130  1.0000000  0.0001895735
  0.2    0.15    0.7371104  1.0000000  0.0000000000
  0.2    0.16    0.7359771  1.0000000  0.0000000000
  0.2    0.17    0.7349457  1.0000000  0.0000000000
  0.2    0.18    0.7337705  1.0000000  0.0000000000
  0.2    0.19    0.7326255  1.0000000  0.0000000000
  0.2    0.20    0.7314602  1.0000000  0.0000000000
  0.4    0.01    0.7574166  0.9909546  0.0571956448
  0.4    0.02    0.7560663  0.9950393  0.0393271293
  0.4    0.03    0.7537835  0.9976162  0.0249498353
  0.4    0.04    0.7506596  0.9987012  0.0151026650
  0.4    0.05    0.7471437  0.9994276  0.0085748099
  0.4    0.06    0.7442247  0.9997122  0.0053276642
  0.4    0.07    0.7412631  1.0000000  0.0026393562
  0.4    0.08    0.7389174  1.0000000  0.0011416439
  0.4    0.09    0.7367288  1.0000000  0.0003846954
  0.4    0.10    0.7345721  1.0000000  0.0000000000
  0.4    0.11    0.7321529  1.0000000  0.0000000000
  0.4    0.12    0.7289891  1.0000000  0.0000000000
  0.4    0.13    0.7244721  1.0000000  0.0000000000
  0.4    0.14    0.7182296  1.0000000  0.0000000000
  0.4    0.15    0.7098818  1.0000000  0.0000000000
  0.4    0.16    0.6982592  1.0000000  0.0000000000
  0.4    0.17    0.6769708  1.0000000  0.0000000000
  0.4    0.18    0.6362535  1.0000000  0.0000000000
  0.4    0.19    0.6132415  1.0000000  0.0000000000
  0.4    0.20    0.5757392  1.0000000  0.0000000000
  0.6    0.01    0.7571838  0.9919027  0.0524363655
  0.6    0.02    0.7543588  0.9965978  0.0323897103
  0.6    0.03    0.7497066  0.9985739  0.0176037764
  0.6    0.04    0.7451476  0.9994596  0.0080022579
  0.6    0.05    0.7409874  0.9998717  0.0039837789
  0.6    0.06    0.7378892  1.0000000  0.0013321201
  0.6    0.07    0.7346243  1.0000000  0.0003846954
  0.6    0.08    0.7303117  1.0000000  0.0000000000
  0.6    0.09    0.7229714  1.0000000  0.0000000000
  0.6    0.10    0.7111013  1.0000000  0.0000000000
  0.6    0.11    0.6864195  1.0000000  0.0000000000
  0.6    0.12    0.6366621  1.0000000  0.0000000000
  0.6    0.13    0.5854938  1.0000000  0.0000000000
  0.6    0.14    0.5445528  1.0000000  0.0000000000
  0.6    0.15    0.5125700  1.0000000  0.0000000000
  0.6    0.16    0.5040707  1.0000000  0.0000000000
  0.6    0.17    0.5000000  1.0000000  0.0000000000
  0.6    0.18    0.5000000  1.0000000  0.0000000000
  0.6    0.19    0.5000000  1.0000000  0.0000000000
  0.6    0.20    0.5000000  1.0000000  0.0000000000
  0.8    0.01    0.7566209  0.9927915  0.0478647513
  0.8    0.02    0.7517481  0.9975233  0.0274367109
  0.8    0.03    0.7456189  0.9990464  0.0112472956
  0.8    0.04    0.7405387  0.9998081  0.0045712541
  0.8    0.05    0.7363982  1.0000000  0.0013321201
  0.8    0.06    0.7310258  1.0000000  0.0001951220
  0.8    0.07    0.7204931  1.0000000  0.0000000000
  0.8    0.08    0.6992028  1.0000000  0.0000000000
  0.8    0.09    0.6369670  1.0000000  0.0000000000
  0.8    0.10    0.5756960  1.0000000  0.0000000000
  0.8    0.11    0.5172284  1.0000000  0.0000000000
  0.8    0.12    0.5040707  1.0000000  0.0000000000
  0.8    0.13    0.5000000  1.0000000  0.0000000000
  0.8    0.14    0.5000000  1.0000000  0.0000000000
  0.8    0.15    0.5000000  1.0000000  0.0000000000
  0.8    0.16    0.5000000  1.0000000  0.0000000000
  0.8    0.17    0.5000000  1.0000000  0.0000000000
  0.8    0.18    0.5000000  1.0000000  0.0000000000
  0.8    0.19    0.5000000  1.0000000  0.0000000000
  0.8    0.20    0.5000000  1.0000000  0.0000000000
  1.0    0.01    0.7559170  0.9935554  0.0434816273
  1.0    0.02    0.7487092  0.9983498  0.0205124671
  1.0    0.03    0.7418393  0.9994596  0.0066506441
  1.0    0.04    0.7367547  1.0000000  0.0016909294
  1.0    0.05    0.7294569  1.0000000  0.0001951220
  1.0    0.06    0.7122662  1.0000000  0.0000000000
  1.0    0.07    0.6595150  1.0000000  0.0000000000
  1.0    0.08    0.5757158  1.0000000  0.0000000000
  1.0    0.09    0.5125700  1.0000000  0.0000000000
  1.0    0.10    0.5040707  1.0000000  0.0000000000
  1.0    0.11    0.5000000  1.0000000  0.0000000000
  1.0    0.12    0.5000000  1.0000000  0.0000000000
  1.0    0.13    0.5000000  1.0000000  0.0000000000
  1.0    0.14    0.5000000  1.0000000  0.0000000000
  1.0    0.15    0.5000000  1.0000000  0.0000000000
  1.0    0.16    0.5000000  1.0000000  0.0000000000
  1.0    0.17    0.5000000  1.0000000  0.0000000000
  1.0    0.18    0.5000000  1.0000000  0.0000000000
  1.0    0.19    0.5000000  1.0000000  0.0000000000
  1.0    0.20    0.5000000  1.0000000  0.0000000000

ROC 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.
```



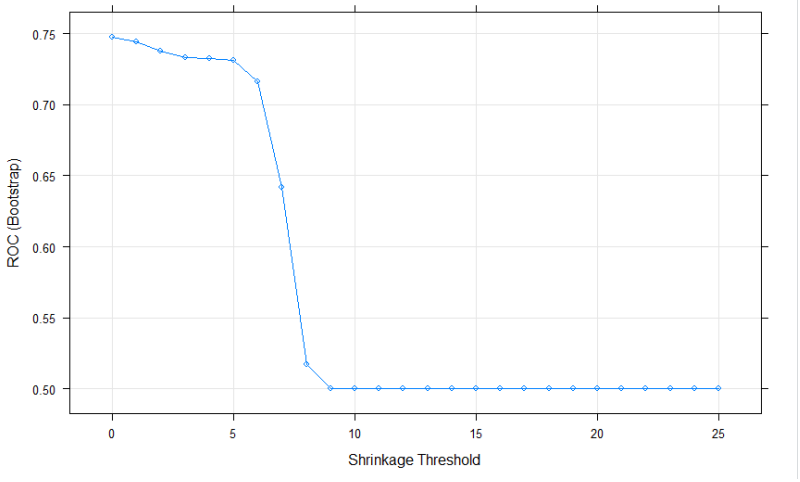
```
Nearest Shrunken Centroids

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

Pre-processing: centered (14), scaled (14)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 4001, 4001, 4001, 4001, 4001, 4001, ...
Resampling results across tuning parameters:

  threshold  ROC        Sens  Spec
   0         0.7477012  1     0
   1         0.7441349  1     0
   2         0.7374817  1     0
   3         0.7331592  1     0
   4         0.7327072  1     0
   5         0.7311192  1     0
   6         0.7163305  1     0
   7         0.6420583  1     0
   8         0.5171627  1     0
   9         0.5000000  1     0
  10         0.5000000  1     0
  11         0.5000000  1     0
  12         0.5000000  1     0
  13         0.5000000  1     0
  14         0.5000000  1     0
  15         0.5000000  1     0
  16         0.5000000  1     0
  17         0.5000000  1     0
  18         0.5000000  1     0
  19         0.5000000  1     0
  20         0.5000000  1     0
  21         0.5000000  1     0
  22         0.5000000  1     0
  23         0.5000000  1     0
  24         0.5000000  1     0
  25         0.5000000  1     0

ROC was used to select the optimal model using the largest value.
The final value used for the model was threshold = 0.
```



Model	ROC Value	Parameters
Logistic Regression	0.7599604	
Linear Discriminant Analysis	0.7690299	
Partial Least Squares Dis. Analysis	0.7630556	Ncomp = 4
Penalized Models	0.7574169	Alpha = 0.2, Lambda = 0.01
Nearest Shrunken Centroid	0.7477012	Threshold = 0

```
##Chapter 12##
##12.1##
##C##
library(caret)
library(glmnet)
library(pamr)
library(AppliedPredictiveModeling)
data(hepatic)
nzv = nearZeroVar(bio)
bio = bio[,-nzv]
damage = as.character( injury )
damage[ damage=="Mild" ] = "yes"
damage[ damage=="Severe" ] = "yes"
damage[ damage=="None" ] = "none"
table( damage )
partition = createDataPartition(damage, p=4/5, list=FALSE)
damage <- factor(damage, levels=c("yes","none"))
ctrlLR <- trainControl(method="boot", 25)
##Logistic Regression##
LR <- train(bio, damage, method="glm", preProc=c("center","scale"), trControl=ctrlLR)
LR
confusionMatrix(LR$pred$pred, LR$pred$obs)
LRcon <- confusionMatrix(data = LR$pred$pred, reference = LR$pred$obs)
##Linear Discriminant Analysis##
ctrlLDA <- trainControl(method="LGOCV")
LDA <- train(bio, damage, method="lda", preProc=c("center","scale"), trControl=ctrlLDA)
LDA
##PArtial Least Squares Discriminant Analysis
ctrlPLSDA <- trainControl(summaryFunction = twoClassSummary,
                          classProbs = TRUE)
```

```
PLSDA <- train(bio, damage, method = "pls", tuneGrid = expand.grid(.ncomp = 1:4), preProc = c("center","scale"), metric = "ROC", trControl = ctrlPLSDA)

PLSDA

plot(PLSDA)

##Penalized Model##

ctrlPM <- trainControl(method = "LGOCV", summaryFunction = twoClassSummary, classProbs = TRUE)

PMGrid <- expand.grid(.alpha = c(0, .1, .2, .4, .6, .8, 1), .lambda = seq(.01, .2, length = 10))

PM <- train(bio, damage, method = "glmnet", tuneGrid = PMGrid, preProc = c("center", "scale"), metric = "ROC", trControl = ctrlPM)

PM

plot(PM)

##Nearest Shrunken Centroids##

nscGrid <- data.frame(.threshold = seq(0,4, by=0.1))

NSC <- train(bio, damage, method = "pam", preProc = c("center", "scale"), tuneGrid = nscGrid, metric = "ROC", trControl = ctrlPLSDA)

NSC

plot(NSC)

##D##

varImp(LDA)


##12.3##

install.packages("modeldata")

library(modeldata)

data(mlc_churn)

library(caret)

library(AppliedPredictiveModeling)

library(pROC)

churn <- mlc_churn

table(churn$churn)

plot(churn)
```

```
train <- churn[,-c(1,3,4,5)]

test <- churn[,c(1,3,4,5)]

y <- train[,16]

y <- data.frame(y)

x <- train[,-16]

x <- x[,-nearZeroVar(x)]

partition <- createDataPartition(churn$churn, p=4/5, list=FALSE)

trP <- x[partition,]

trR <- y[partition]



##Logistic Regression##

ctrl=trainControl(summaryFunction=twoClassSummary, classProbs=TRUE)

LR <- train(trP, trR, method="glm", preProc=c("center","scale"), trControl=ctrl)

LR

confusionMatrix(LR$pred$pred, LR$pred$obs)

##Linear Discriminant Analysis##

LDA <- train(trP, trR, method="lda", preProc=c("center","scale"),metric="ROC", trControl=ctrl)

LDA

plot(LDA)

##Partial LEast Squares Discriminant Analysis##

PLSDA <- train(trP, trR, method="pls", tuneGrid=expand.grid(.ncomp=1:10),
preProc=c("center","scale"), metric="ROC", trControl=ctrl)

PLSDA

plot(PLSDA)

##Penalized##

grid = expand.grid(.alpha=c(0, 0.2, 0.4, 0.6, 0.8, 1.0), .lambda=seq( 0.01, 0.2, length=20))

PM <- train(trP, trR, method="glmnet", tuneGrid=grid, preProc=c("center","scale"),
metric="ROC", trControl=ctrl)
```

```
PM

plot(PM)

##Nearest Shrunken Centroids##

NGrid = expand.grid(.threshold=0:25)

NSC <- train(trP, trR, method="pam", tuneGrid=NGrid, preProc=c("center","scale"),
metric="ROC", trControl=ctrl)

NSC

plot(NSC)
```