Ex13

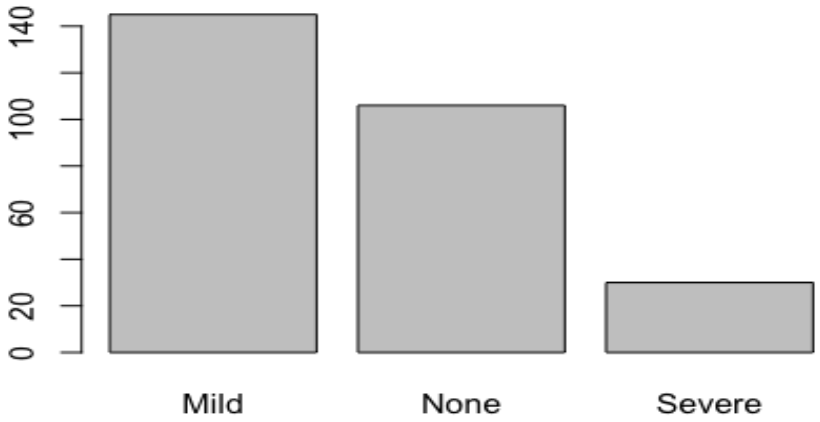
Chathrua Gunasekara

1.a Preprocessing steps done in chapter 12 are repeated in the same way in this exercise too.

Preprocessing done on both bio and chem and combined data sets.

i. Remove nearzero variance predictors
ii. Remove high correlated predictors
iii. Remove linear combination predictors
iv. Splitting data set using stratified sampling

Following diagram illustrates the class distribution in predictor variable.



1. Mixture Discriminant Analysis

```
225 samples
 96 predictor
  3 classes: 'Mild', 'None', 'Severe'

Pre-processing : Center and Scale
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...
Resampling results across tuning parameters:

  subclasses  Accuracy   Kappa       Accuracy SD  Kappa SD
  1           0.4145599  0.05659400  0.05389300   0.06594883
  2           0.4239917  0.06274809  0.03231601   0.04932816
  3           0.4361365  0.07152077  0.03558633   0.05913592
  4           0.3832163  0.01625659  0.02123588   0.02585894
  5           0.4129537  0.05560859  0.06633504   0.08922207
```

```
Kappa was used to select the optimal model using  the largest value.
The final value used for the model was subclasses = 3.

Confusion Matrix and Statistics



          Reference
Prediction Mild None Severe
    Mild     19   12      3
    None      6    9      2
    Severe    4    0      1

Overall Statistics for Testing set

               Accuracy : 0.5179
                 95% CI : (0.3803, 0.6534)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.5537

                  Kappa : 0.1424
 Mcnemar's Test P-Value : 0.2464

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.6552      0.4286       0.16667
Specificity               0.4444      0.7714       0.92000
```

2. Neural Network

```
225 samples
 96 predictor
  3 classes: 'Mild', 'None', 'Severe'

Pre-processing: spatial sign transformation, scaled, centered
Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...



Resampling results across tuning parameters:

  size  decay  Accuracy   Kappa          Accuracy SD  Kappa SD
   1    0.0    0.3912201  -0.0005849393  0.07975896   0.05673328
   1    0.1    0.4407795  -0.0262458622  0.04372911   0.07310105
   1    1.0    0.4600638  -0.0362919794  0.04409184   0.06066712
   1    2.0    0.4921698  -0.0068195868  0.07045113   0.01971180
   2    0.0    0.3856094  -0.0004402377  0.08293128   0.07808454
```

```
   5    0.0    0.4096316  -0.0014802390  0.05190740   0.07650497
   5    0.1    0.4270451   0.0019751218  0.04324519   0.06628027
   5    1.0    0.4633821  -0.0181128791  0.04928725   0.07184842
   5    2.0    0.4947387  -0.0061781221  0.06236068   0.01782982
   6    0.0    0.4095587   0.0010321192  0.04626995   0.05163436
   6    0.1    0.4321282   0.0098848259  0.04536148   0.07110344
   6    1.0    0.4618823  -0.0210948650  0.04883736   0.07177771
   6    2.0    0.4947387  -0.0061781221  0.06236068   0.01782982
   7    0.0    0.4185346   0.0122778107  0.05351421   0.08608156
   7    0.1    0.4320596   0.0091119355  0.04101778   0.06640802
   7    1.0    0.4628187  -0.0189470605  0.04884592   0.07115908
   7    2.0    0.4947387  -0.0061781221  0.06236068   0.01782982
   8    0.0    0.4288824   0.0154870583  0.04395671   0.06638126
   8    0.1    0.4280180   0.0038667622  0.04409914   0.06890253
   8    1.0    0.4623368  -0.0200885878  0.04870877   0.07126549
   8    2.0    0.4947387  -0.0061781221  0.06236068   0.01782982
   9    0.0    0.4200026  -0.0016920521  0.04874879   0.07801793
   9    0.1    0.4280546   0.0062862751  0.04730466   0.07316046
   9    1.0    0.4617518  -0.0207221916  0.04745530   0.07010182
   9    2.0    0.4947387  -0.0061781221  0.06236068   0.01782982

Kappa was used to select the optimal model using  the largest value.
The final values used for the model were size = 8 and decay = 0.

Confusion Matrix and Statistics Testing set

          Reference
Prediction Mild None Severe
    Mild     20   13      5
    None      7    7      1
    Severe    2    1      0

Overall Statistics

               Accuracy : 0.4821
                 95% CI : (0.3466, 0.6197)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.7482

                  Kappa : 0.0453


Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.6897      0.3333       0.00000
Specificity               0.3333      0.7714       0.94000
```

3. Flexible Discriminant Analysis

```
  225 samples
   96 predictor
    3 classes: 'Mild', 'None', 'Severe'

  Pre-processing : Center and Scale
  Resampling: Bootstrapped (25 reps)

  Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

  Resampling results across tuning parameters:

    nprune  Accuracy   Kappa        Accuracy SD  Kappa SD
     2      0.4861072  0.006071507  0.05023715   0.05087418
    35      0.4402642  0.046216816  0.04940299   0.08463509
    69      0.4361744  0.048250463  0.05527767   0.08147733

  Tuning parameter 'degree' was held constant at a value of 1
  Kappa was used to select the optimal model using  the largest value.
  The final values used for the model were degree = 1 and nprune = 69.
```

Confusion Matrix and Statistics **Testing set**

Prediction	Reference Mild	None	Severe
Mild	25	16	5
None	1	4	1
Severe	3	1	0

Overall Statistics

Accuracy	0.5179
95% CI	(0.3803, 0.6534)
No Information Rate	0.5179
P-Value [Acc > NIR]	0.553730
Kappa	0.0847
Mcnemar's Test P-Value	0.003289

Statistics by Class:

	Class: Mild	Class: None	Class: Severe
Sensitivity	0.8621	0.19048	0.00000
Specificity	0.2222	0.94286	0.92000

4. Support Vector Machines with Radial Basis Function Kernel

```
225 samples
 96 predictor
  3 classes: 'Mild', 'None', 'Severe'

Pre-processing : Center and Scale
Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

Resampling results across tuning parameters:

  C        Accuracy   Kappa         Accuracy SD  Kappa SD
   0.0625  0.5121283   0.000000000  0.04002546   0.000000000
   0.1250  0.5121283   0.000000000  0.04002546   0.000000000
   0.2500  0.5097409   0.000987425  0.03592039   0.008526046
   0.5000  0.4977915  -0.009749648  0.03509011   0.028841709
   1.0000  0.4932844   0.005611551  0.04567346   0.076477763
   2.0000  0.4863447   0.012146986  0.04611067   0.097487928
   4.0000  0.4837670   0.035061047  0.03573951   0.073209728
   8.0000  0.4860284   0.063532606  0.03820134   0.071332936
  16.0000  0.4700797   0.048854218  0.04206646   0.076239954

Tuning parameter 'sigma' was held constant at a value of 0.002492319
Kappa was used to select the optimal model using  the largest value.
The final values used for the model were sigma = 0.002492319 and C = 8.
```

Confusion Matrix and Statistics **Testing set**

```
          Reference
Prediction Mild None Severe
    Mild     22   13      5
    None      6    8      1
    Severe    1    0      0

Overall Statistics

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

                  Kappa : 0.1202
 Mcnemar's Test P-Value : 0.1003

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.7586      0.3810       0.00000
Specificity               0.3333      0.8000       0.98000
```

5. k-Nearest Neighbors

```
225 samples
 96 predictor
  3 classes: 'Mild', 'None', 'Severe'
Pre-processing : Center and Scale
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...
Resampling results across tuning parameters:
```

k	Accuracy	Kappa	Accuracy SD	Kappa SD
1	0.4798202	0.096591515	0.05429008	0.085075079
5	0.4644326	0.068348020	0.05141708	0.068566556
9	0.4749089	0.055596568	0.05450278	0.076484795
13	0.5054816	0.077005867	0.06561231	0.080430756
17	0.5148153	0.079358231	0.06213484	0.069301861
251	0.5256963	0.000000000	0.04019658	0.000000000
301	0.5256963	0.000000000	0.04019658	0.000000000
351	0.5256963	0.000000000	0.04019658	0.000000000
401	0.5256963	0.000000000	0.04019658	0.000000000
451	0.5256963	0.000000000	0.04019658	0.000000000

```
Kappa was used to select the optimal model using  the largest value.
The final value used for the model was k = 13.
```

Confusion Matrix and Statistics **Testing set**

Prediction	Reference Mild	None	Severe
Mild	27	16	5
None	2	5	0
Severe	0	0	1

```
Overall Statistics

               Accuracy : 0.5893
                 95% CI : (0.4498, 0.719)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.1747
```

```
                  Kappa : 0.1904
   Mcnemar's Test P-Value : NA

  Statistics by Class:

                       Class: Mild Class: None Class: Severe
  Sensitivity               0.9310     0.23810       0.16667
  Specificity               0.2222     0.94286       1.00000
```

6. Naive Bayes

```
225 samples

 96 predictor

  3 classes: 'Mild', 'None', 'Severe'

Pre-processing : Center and Scale

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

Resampling results across tuning parameters:

  usekernel  Accuracy   Kappa       Accuracy SD  Kappa SD

  FALSE            NaN         NaN         NA             NA

   TRUE      0.2618047  0.02044771  0.1022174    0.04616507


Tuning parameter 'fL' was held constant at a value of 0

Kappa was used to select the optimal model using  the largest value.

The final values used for the model were fL = 0 and usekernel = TRUE.


  Confusion Matrix and Statistics Testing set

            Reference
  Prediction Mild None Severe
      Mild      2    1      1
      None      3    4      0
      Severe   24   16      5

  Overall Statistics

                 Accuracy : 0.1964
                   95% CI : (0.1023, 0.3243)
      No Information Rate : 0.5179
      P-Value [Acc > NIR] : 1

                    Kappa : 0.0319
```

```
   Mcnemar's Test P-Value : 2.614e-08


Statistics by Class:

                     Class: Mild Class: None Class: Severe
  Sensitivity              0.06897     0.19048       0.83333
  Specificity              0.92593     0.91429       0.20000
```

From Ex 12 :

FOR Testing set:

LIEAR MODEL	Accuracy	Kappa
Logistic Reg (averaged)	0.5833	0.02
LDA	0.5179	0.102
PLSDA	0.5893	0.04
NSC	0.625	0.07

NON LIEAR MODEL	Accuracy	Kappa
MDA	***0.5179***	***0.1424***
NNet	0.4821	0.0453
FDA	0.5179	0.0847
SVM	0.5357	0.1202
KNN	0.5893	0.1904
Naïve Bayes	0.1964	0.0319

Best Models for Biological predictors is MDA model. Yes it does do a better job than all of the Linear models from chapter 12 for the biological data.

Chemical Predictors

(Same preprocessing has been done as the Ex 12 and biological predictors)

1. Mixture Discriminant Analysis

```
225 samples
105 predictors
  3 classes: 'Mild', 'None', 'Severe'

Pre-processing : Center and Scale
Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

Resampling results across tuning parameters:

  subclasses  Accuracy   Kappa      Accuracy SD  Kappa SD
   1          0.4768575  0.1407148  0.04461386   0.06389823
   2          0.4809074  0.1380243  0.04511030   0.06184408
   3          0.4790951  0.1497179  0.04332225   0.06019336
   4          0.4930027  0.1261193  0.02066666   0.06079511
   5          0.4935065  0.1365727          NA           NA

Kappa was used to select the optimal model using  the largest value.
The final value used for the model was subclasses = 3.
```

Confusion Matrix and Statistics **Testing set**

```
          Reference
Prediction Mild None Severe
    Mild     19   12      6
    None      7    4      0
    Severe    3    5      0

Overall Statistics

               Accuracy : 0.4107
                 95% CI : (0.281, 0.5502)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.95910

                  Kappa : -0.0359
 Mcnemar's Test P-Value : 0.06249



Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.6552     0.19048        0.0000
Specificity               0.3333     0.80000        0.8400
```

2. Neural Network

```
  225 samples
  105 predictors
    3 classes: 'Mild', 'None', 'Severe'

  Pre-processing: spatial sign transformation, scaled, centered
  Resampling: Bootstrapped (25 reps)

  Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

  Resampling results across tuning parameters:

    size  decay  Accuracy   Kappa          Accuracy SD  Kappa SD
     1    0.0    0.4362578   0.0643256215  0.09454080   0.086041032
     1    0.1    0.5397015   0.1535189601  0.05579580   0.090870709
     1    1.0    0.5319810   0.1108969983  0.05090369   0.075931592
     1    2.0    0.5029356   0.0000000000  0.07226298   0.000000000
     2    0.0    0.4656351   0.1232237996  0.07902498   0.083677834
     2    0.1    0.5220552   0.1522592843  0.05154941   0.080289755
     2    1.0    0.5276783   0.1028643224  0.05041837   0.088945147
     2    2.0    0.5029356   0.0000000000  0.07226298   0.000000000
     3    0.0    0.4837104   0.1179377366  0.05773742   0.083631879
     3    0.1    0.5289381   0.1632786716  0.05074972   0.086573708
     3    1.0    0.5262686   0.1016492490  0.05092069   0.088742193
     3    2.0    0.5038245  -0.0011502054  0.06931202   0.005751027
     4    0.0    0.4652746   0.0940609344  0.06512006   0.089810182
     4    0.1    0.5360214   0.1756546192  0.05396036   0.088262583
     4    1.0    0.5276264   0.1036408713  0.05074551   0.088828979
     4    2.0    0.5038245  -0.0011502054  0.06931202   0.005751027
     5    0.0    0.4855815   0.1182206895  0.04985448   0.080637664
     6    0.1    0.5326444   0.1667321278  0.04741452   0.079481441
     6    1.0    0.5276311   0.1037629926  0.05048460   0.087904638
     6    2.0    0.5042690  -0.0007327586  0.06789757   0.003663793
     7    0.0    0.4965067   0.1325307242  0.05740434   0.087491481

  Kappa was used to select the optimal model using  the largest value.
  The final values used for the model were size = 4 and decay = 0.1.

  Confusion Matrix and Statistics Testing set

            Reference
  Prediction Mild None Severe
      Mild     20    7      6
      None      5   10      0
      Severe    4    4      0



Overall Statistics for Testing set

                  Accuracy : 0.5357
                    95% CI : (0.3974, 0.6701)
```

```
     No Information Rate : 0.5179
     P-Value [Acc > NIR] : 0.4475

                   Kappa : 0.1982
  Mcnemar's Test P-Value : 0.1924

  Statistics by Class:

                       Class: Mild Class: None Class: Severe
  Sensitivity               0.6897      0.4762        0.0000
  Specificity               0.5185      0.8571        0.8400
  Pos Pred Value            0.6061      0.6667        0.0000
  Neg Pred Value            0.6087      0.7317        0.8750
  Prevalence                0.5179      0.3750        0.1071
  Detection Rate            0.3571      0.1786        0.0000
  Detection Prevalence      0.5893      0.2679        0.1429
  Balanced Accuracy         0.6041      0.6667        0.4200
```

3. Flexible Discriminant Analysis

```
  225 samples
  105 predictors
    3 classes: 'Mild', 'None', 'Severe'

  Pre-processing : Center and Scale
  Resampling: Bootstrapped (25 reps)

  Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

  Resampling results across tuning parameters:

    nprune  Accuracy   Kappa       Accuracy SD  Kappa SD
     2      0.5015368  0.03112774  0.04447235   0.06885291
    37      0.4923910  0.12518750  0.05192218   0.06916835
    72      0.4839631  0.12240884  0.05371793   0.07259106

  Tuning parameter 'degree' was held constant at a value of 1
  Kappa was used to select the optimal model using  the largest value.
  The final values used for the model were degree = 1 and nprune = 37.


Confusion Matrix and Statistics for Testing set

            Reference
  Prediction Mild None Severe
      Mild     18   10      5
      None      7   10      1
      Severe    4    1      0

  Overall Statistics
```

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

                  Kappa : 0.1146
 Mcnemar's Test P-Value : 0.8871

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.6207      0.4762       0.00000
Specificity               0.4444      0.7714       0.90000
```

4.Support Vector Machines with Radial Basis Function Kernel

```
225 samples
105 predictors
  3 classes: 'Mild', 'None', 'Severe'

Pre-processing : Center and Scale
Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

Resampling results across tuning parameters:

  C        Accuracy   Kappa        Accuracy SD  Kappa SD
   0.0625  0.5145799  0.000000000  0.03848358   0.00000000
   0.1250  0.5145799  0.000000000  0.03848358   0.00000000
   0.2500  0.5136192  0.001083215  0.03693514   0.00978591
   0.5000  0.5183040  0.035416932  0.03409514   0.04878818
   1.0000  0.5396650  0.111770968  0.03903374   0.07316135
   2.0000  0.5581065  0.171586409  0.05365454   0.09837376
   4.0000  0.5638349  0.201461792  0.04848978   0.08507142
   8.0000  0.5527656  0.195408110  0.04879124   0.08445991
  16.0000  0.5491559  0.194837858  0.05111048   0.08646477

Tuning parameter 'sigma' was held constant at a value of 0.002809725
Kappa was used to select the optimal model using  the largest value.
The final values used for the model were sigma = 0.002809725 and C = 4.
```

Confusion Matrix and Statistics **Testing set**

```
          Reference
Prediction Mild None Severe
    Mild     23   13      6
    None      6    8      0
    Severe    0    0      0
```

```
Overall Statistics

               Accuracy : 0.5536
                 95% CI : (0.4147, 0.6866)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.3448

                  Kappa : 0.1379
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.7931      0.3810        0.0000
Specificity               0.2963      0.8286        1.0000
```

Confusion Matrix and Statistics **Testing set**

```
          Reference
Prediction Mild None Severe
    Mild     19    9      3
    None     10   12      2
    Severe    0    0      1

Overall Statistics

               Accuracy : 0.5714
                 95% CI : (0.4322, 0.7029)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.2524

                  Kappa : 0.2218
Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.6552      0.5714       0.16667
Specificity               0.5556      0.6571       1.00000
```

5. k-Nearest Neighbors

225 samples

105 predictors

3 classes: 'Mild', 'None', 'Severe'

Pre-processing : `Center and Scale`

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

Resampling results across tuning parameters:

k Accuracy Kappa Accuracy SD Kappa SD

```
 3 0.5454926  0.2131093798 0.04922864  0.085427893
 5 0.4854528  0.1079461141 0.04748937  0.069487519
 9 0.4861334  0.0839134159 0.05149011  0.089374177
13 0.4918352  0.0866274922 0.04852894  0.080135955
17 0.4907008  0.0854200474 0.06183916  0.092388082
21 0.4873642  0.0756337978 0.05701704  0.084778743
```

Kappa was used to select the optimal model using the largest value.

The final value used for the model was k = 3

```
Confusion Matrix and Statistics Testing set

          Reference
Prediction Mild None Severe
    Mild     23   12      3
    None      5    5      2
    Severe    1    4      1

Overall Statistics

               Accuracy : 0.5179
                 95% CI : (0.3803, 0.6534)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.5537

                  Kappa : 0.134
Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.7931     0.23810       0.16667
Specificity               0.4444     0.80000       0.90000
```

6. Naive Bayes

225 samples

105 predictors

3 classes: 'Mild', 'None', 'Severe'

Pre-processing : `Center and Scale`

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

Resampling results across tuning parameters:

usekernel	Accuracy	Kappa	Accuracy SD	Kappa SD
FALSE	NaN	NaN	NA	NA
TRUE	0.5012535	0.06762174	0.05173594	0.07834722

Tuning parameter 'fL' was held constant at a value of 0

Kappa was used to select the optimal model using the largest value.

The final values used for the model were fL = 0 and usekernel = TRUE.

Confusion Matrix and Statistics **`Testing set`**

Prediction	Reference: Mild	None	Severe
Mild	25	15	5
None	4	5	1
Severe	0	1	0

Overall Statistic

Accuracy : 0.5357

Kappa : 0.0985

For Testing set:

LINEAR	Accuracy	Kappa
LDA	0.5179	0.102
PLSDA	0.5357	0.145
NSC	0.51	0

Non Linear Model	Accuracy	Kappa
MDA	0.4107	-0.012
NNEt	0.5389	0.1982
FDA	0.5	0.1146
SVM	0.5536	0.1379
KNN	0.5179	0.134
Naïve Bayes	0.5357	0.0985

Non Lenear model Neural network is better for the Chemical predictor comparing all the models linear and non linear.

Combined Predictors(BIO+CHEM)

1. Non linear Discrimination Analysis

```
mda(formula = trainY ~ ., data = train)

  Dimension: 8
  Pre Process: Center and Scale
  Percent Between-Group Variance Explained:
      v1     v2     v3     v4     v5     v6     v7     v8
   45.62  70.54  79.58  87.30  92.60  96.98  99.22 100.00


  Deviance: 0

  Confusion Matrix and Statistics

            Reference
  Prediction Mild None Severe
      Mild     11   11      3
      None     11    4      1
      Severe    7    6      2

  Overall Statistics for Testing set

                 Accuracy : 0.3036
                   95% CI : (0.1878, 0.441)
      No Information Rate : 0.5179
      P-Value [Acc > NIR] : 0.9996

                    Kappa : -0.1003
   Mcnemar's Test P-Value : 0.1597

  Statistics by Class:

                       Class: Mild Class: None Class: Severe
  Sensitivity               0.3793     0.19048       0.33333
  Specificity               0.4815     0.65714       0.74000
```

2. Neural Network

```
  225 samples
  202 predictors
    3 classes: 'Mild', 'None', 'Severe'

  Pre-processing: spatial sign transformation, scaled, centered
  Resampling: Bootstrapped (25 reps)

  Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

  Resampling results across tuning parameters:
```

```
  size  decay  Accuracy   Kappa       Accuracy SD  Kappa SD
  1     0.0    0.4537836  0.07471607  0.08057553   0.06684354
  1     0.1    0.5133845  0.12088479  0.06281646   0.10210564
  2     0.0    0.4428505  0.08085378  0.06288195   0.07227809
  2     0.1    0.5023176  0.11916586  0.05634693   0.09123446
  3     0.0    0.4676187  0.08745088  0.04241189   0.07151448
  3     0.1    0.4956529  0.10897236  0.05547206   0.08962320
  4     0.0    0.4686464  0.09977135  0.04895218   0.07394145


Kappa was used to select the optimal model using  the largest value.
The final values used for the model were size = 1 and decay = 0.1.

Confusion Matrix and Statistics for Testing set

          Reference
Prediction Mild None Severe
    Mild     17   16      5
    None     12    5      1
    Severe    0    0      0

Overall Statistics

               Accuracy : 0.3929
                 95% CI : (0.265, 0.5325)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.97778

                  Kappa : -0.1498
 Mcnemar's Test P-Value : 0.08689

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.5862     0.23810        0.0000
Specificity               0.2222     0.62857        1.0000
```

3.Flexible Discriminant Analysis

```
225 samples
202 predictors
  3 classes: 'Mild', 'None', 'Severe'

Pre-processing : Center and Scale
Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...
```

```
Resampling results across tuning parameters:

  nprune  Accuracy   Kappa       Accuracy SD  Kappa SD
   2      0.4850801  0.01561390  0.04515211   0.06310917
  38      0.4773120  0.11601226  0.05975337   0.09750591
  74      0.4413228  0.08431081  0.07197247   0.10911220

Tuning parameter 'degree' was held constant at a value of 1
Kappa was used to select the optimal model using  the largest value.
The final values used for the model were degree = 1 and nprune = 38.
```

Confusion Matrix and Statistics **Testing set**

```
          Reference
Prediction Mild None Severe
    Mild     22   14      2
    None      6    6      2
    Severe    1    1      2

Overall Statistics

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

                  Kappa : 0.1515
 Mcnemar's Test P-Value : 0.2762

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.7586      0.2857       0.33333
Specificity               0.4074      0.7714       0.96000
```

4.Support Vector Machines with Radial Basis Function Kernel

```
225 samples
202 predictors
  3 classes: 'Mild', 'None', 'Severe'

Pre-processing : Center and Scale
Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

Resampling results across tuning parameters:

  C       Accuracy   Kappa        Accuracy SD  Kappa SD
   0.0625  0.5122456  0.000000000  0.03794856   0.00000000
   0.1250  0.5122456  0.000000000  0.03794856   0.00000000
   0.2500  0.5119341  0.005702462  0.04175516   0.02721642
```

```
    0.5000  0.5147571  0.040719794  0.03682783   0.06075267
    1.0000  0.5228223  0.078864123  0.03592325   0.06688309
    2.0000  0.5374330  0.128709971  0.03744246   0.06212469
    4.0000  0.5374355  0.147896508  0.04652034   0.08207168
    8.0000  0.5232891  0.138126080  0.04664973   0.07857214
   16.0000  0.5107618  0.123936561  0.05396726   0.09679151

Tuning parameter 'sigma' was held constant at a value of 0.001278265
Kappa was used to select the optimal model using  the largest value.
The final values used for the model were sigma = 0.001278265 and C = 4.

Confusion Matrix and Statistics for testing set

          Reference
Prediction Mild None Severe
    Mild     20   17      6
    None      8    4      0
    Severe    1    0      0

Overall Statistics

               Accuracy : 0.4286
                 95% CI : (0.2971, 0.5678)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.9294

                  Kappa : -0.0987
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.6897     0.19048       0.00000
Specificity               0.1481     0.77143       0.98000
```

5. k-Nearest Neighbors

225 samples

202 predictors

3 classes: 'Mild', 'None', 'Severe'

pre-processing : Center and Scale

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

Resampling results across tuning parameters:

k	Accuracy	Kappa	Accuracy SD	Kappa SD
1	0.4604128	0.1026969919	0.04571765	0.077625979
5	0.4770297	0.1049465821	0.06632193	0.077520150
9	0.4769611	0.0710456864	0.07567141	0.095280731
13	0.4775598	0.0496226638	0.07097258	0.077295030
17	0.4853737	0.0475059905	0.06182266	0.062205010
21	0.4850623	0.0382689613	0.06228243	0.079176958
41	0.4952677	0.0311927608	0.05836922	0.081458116

Kappa was used to select the optimal model using the largest value.

The final value used for the model was k = 5.

```
Confusion Matrix and Statistics for Testing set
          Reference
Prediction Mild None Severe
    Mild     18   12      4
    None      9    8      2
    Severe    2    1      0

Overall Statistics

               Accuracy : 0.4643
                 95% CI : (0.3299, 0.6026)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.8254

                  Kappa : 0.0306
 Mcnemar's Test P-Value : 0.6989
```

Statistics by Class:

	Class: Mild	Class: None	Class: Severe
Sensitivity	0.6207	0.3810	0.00000
Specificity	0.4074	0.6857	0.94000

6.Naive Bayes

225 samples

202 predictors

3 classes: 'Mild', 'None', 'Severe'

Pre-processing : Center and Scale

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

Resampling results across tuning parameters:

usekernel	Accuracy	Kappa	Accuracy SD	Kappa SD
FALSE	NaN	NaN	NA	NA
TRUE	0.4643774	0.08191371	0.08246112	0.07702208

Tuning parameter 'fL' was held constant at a value of 0

Kappa was used to select the optimal model using the largest value.

The final values used for the model were fL = 0 and usekernel = TRUE.

Confusion Matrix and Statistics **Testing set**

Prediction	Reference Mild	None	Severe
Mild	22	18	5
None	4	2	0
Severe	3	1	1

Overall Statistics

Accuracy : 0.4464
95% CI : (0.3134, 0.5853)
No Information Rate : 0.5179
P-Value [Acc > NIR] : 0.88561

Kappa : -0.0364

```
Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.7586     0.09524       0.16667
Specificity               0.1481     0.88571       0.92000
```

Testing set:

Linear	Accuracy	Kappa	Sensitivity	Specificity
LDA	0.3571	0.03	0.412	0.695
PLSDA	0.5357	0.09	0.4562	0.6298
NSC	0.4643	0.09	0	0.905

Non Linear Models	Accuracy	Kappa
MDA	0.3069	0.032
NNet	0.3929	0.0132
FDA	0.5357	0.1515
SVM	0.4268	0.031
KNN	0.4643	0.036
Naïve Bayes	0.4464	0.03

Both Linear and Non-linear models discussed in here do NOT do better on the combined data set. Only F DA shows somewhat better performance.

b)

20 most important variables shown (out of 96)
for **Biological Data**

	Mild	None	Severe
Z15	0.6013	0.6355	0.6355
Z100	0.6015	0.6108	0.6108
Z116	0.5990	0.5515	0.5990
Z59	0.5803	0.5434	0.5803
Z44	0.5801	0.5578	0.5801
Z56	0.5700	0.5782	0.5782
Z167	0.5756	0.5441	0.5756
Z64	0.5695	0.5695	0.4603
Z34	0.5658	0.5658	0.5241
Z121	0.5368	0.5581	0.5581
Z18	0.5580	0.5580	0.5090
Z101	0.5571	0.5461	0.5571
Z7	0.5548	0.5424	0.5548
Z46	0.5523	0.5523	0.4343
Z11	0.5514	0.5272	0.5514
Z71	0.5498	0.5498	0.4652
Z50	0.5487	0.5487	0.5208
Z42	0.5477	0.5477	0.4682
Z53	0.5016	0.5453	0.5453
Z48	0.5450	0.5450	0.4447

20 most important variables shown (out of 96)
for **Chemical Predictors**

	Mild	None	Severe
X139	0.6694	0.6870	0.6870
X145	0.6566	0.6804	0.6804
X1	0.6386	0.6711	0.6711
X133	0.5903	0.6701	0.6701
X132	0.6307	0.6672	0.6672
X144	0.6471	0.6600	0.6600
X101	0.6228	0.6576	0.6576
X35	0.5867	0.6544	0.6544
X138	0.6480	0.6346	0.6480
X81	0.6221	0.6471	0.6471
X150	0.6386	0.5990	0.6386
X120	0.5744	0.6348	0.6348
X171	0.6060	0.6297	0.6297
X103	0.5997	0.6260	0.6260
X127	0.6058	0.6240	0.6240
X24	0.5961	0.6208	0.6208
X28	0.5894	0.6196	0.6196
X62	0.5824	0.6137	0.6137
X142	0.6128	0.5953	0.6128
X23	0.6124	0.5738	0.6124

c) Combined Predictors

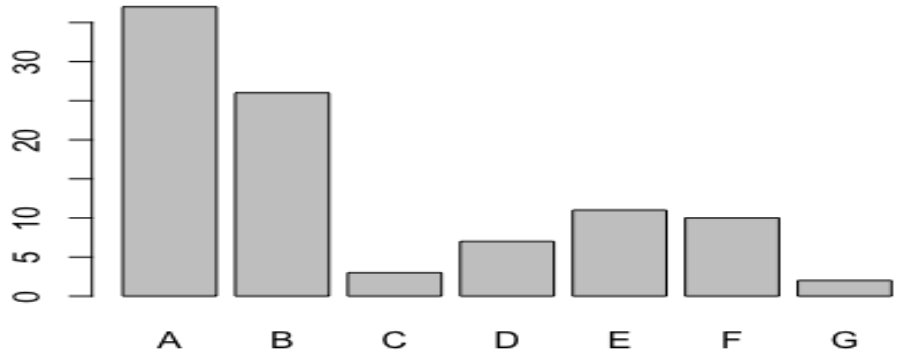
only 20 most important variables shown (out of 202) **BIO+CHEM Combined**

	Mild	None	Severe
X1	0.6640	0.6757	0.6757
X172	0.6652	0.6373	0.6652
X139	0.6426	0.6608	0.6608
X150	0.6573	0.5880	0.6573
X142	0.6518	0.6162	0.6518
X132	0.6403	0.6458	0.6458
X138	0.6406	0.6272	0.6406
X141	0.6404	0.6071	0.6404
X28	0.6160	0.6370	0.6370
X24	0.5907	0.6331	0.6331
X120	0.5916	0.6331	0.6331
X144	0.6306	0.6284	0.6306
X151	0.6304	0.5716	0.6304
Z15	0.5675	0.6265	0.6265
X171	0.5729	0.6243	0.6243
X133	0.5445	0.6225	0.6225
X145	0.6036	0.6196	0.6196
X123	0.6144	0.6154	0.6154
Z40	0.5671	0.6125	0.6125
X85	0.6058	0.6110	0.61

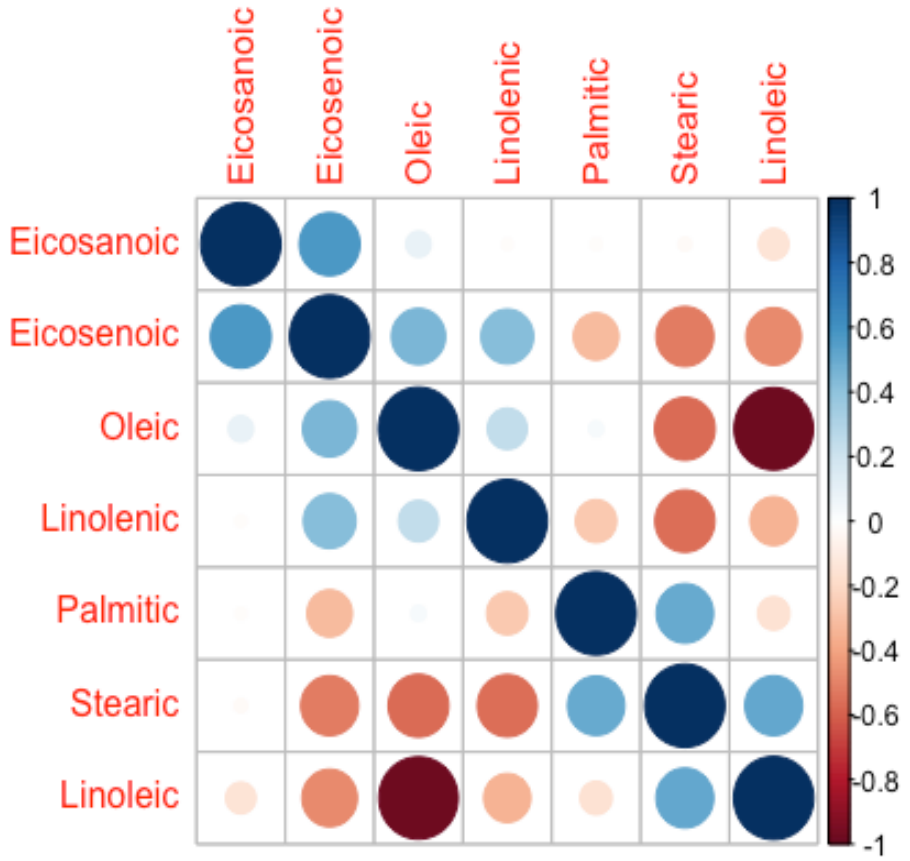
d) **Biological data with Non-Linear model(SVM)** performs best out of other cases consider in this exercise.

2.a.

Because the class imbalance the data set should be split using stratified sampling.




```
Based on the Correlation plot there are some correlated predictors which were removed.
```



1. Mixture Discriminant Analysis

```
74 samples
 6 predictor
 7 classes: 'A', 'B', 'C', 'D', 'E', 'F', 'G'

Pre-processing : Center and Scale
Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 74, 74, 74, 74, 74, 74, ...

Resampling results across tuning parameters:

  subclasses  Accuracy   Kappa      Accuracy SD  Kappa SD
  1           0.9268027  0.9000571  0.04613226   0.06230732
  2           0.9136905  0.8766354  0.09978009   0.14252050

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

Confusion Matrix and Statistics **Testing set**

```
          Reference
Prediction A B C D E F G
         A 9 0 0 0 0 0 0
         B 0 6 0 0 0 0 0
         C 0 0 1 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 1

Overall Statistics for Testing set :

               Accuracy : 1
                 95% CI : (0.8456, 1)
    No Information Rate : 0.4091
    P-Value [Acc > NIR] : 2.884e-09

                  Kappa : 1
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E Class: F
Sensitivity            1.0000   1.0000  1.00000  1.00000  1.00000  1.00000
Specificity            1.0000   1.0000  1.00000  1.00000  1.00000  1.00000
                     Class: G
Sensitivity           1.00000
```

```
Specificity           1.00000


variables are sorted by maximum importance across the classes
           A    B C D E      F G
Stearic    1 1.00 1 1 1 1.0000 1
Palmitic   1 1.00 1 1 1 1.0000 1
Linolenic  1 1.00 1 1 1 1.0000 1
Oleic      1 1.00 1 1 1 1.0000 1
Eicosanoic 1 0.95 1 1 1 1.0000 1
Eicosenoic 1 1.00 1 1 1 0.8542 1
```

2. Neural Network

```
74 samples
 6 predictor
 7 classes: 'A', 'B', 'C', 'D', 'E', 'F', 'G'

Pre-processing: spatial sign transformation, scaled, centered
Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 74, 74, 74, 74, 74, 74, ...

Resampling results across tuning parameters:

  size  decay  Accuracy   Kappa       Accuracy SD  Kappa SD
   1    0.0    0.6987881  0.57791892  0.11334438   0.14718124
   1    0.1    0.6366933  0.47999634  0.13745402   0.15611758
   3    0.0    0.8280413  0.76343216  0.10899117   0.14861201
   3    0.1    0.9036922  0.86530845  0.09538117   0.13467173
   3    1.0    0.6731553  0.50692249  0.11053172   0.14404744
   3    2.0    0.5259045  0.27195591  0.16456447   0.22225626
   4    0.0    0.8634232  0.81252625  0.10719723   0.14398761
   4    0.1    0.9088823  0.87356650  0.09228015   0.12790248
   4    1.0    0.6886408  0.53435636  0.11699580   0.15801568
   4    2.0    0.5291096  0.27813048  0.16321931   0.21659432
   5    0.0    0.8754391  0.82796214  0.09683868   0.13396598
   5    0.1    0.9148953  0.88264541  0.08742802   0.11942961
   5    1.0    0.6948350  0.54428195  0.11903585   0.16212303
   5    2.0    0.5470168  0.30760730  0.16391903   0.21663562
     Kappa was used to select the optimal model using  the largest value.
The final values used for the model were size = 5 and decay = 0.1.

Confusion Matrix and Statistics Testing set

          Reference
Prediction A B C D E F G
         A 9 0 0 0 0 0 0
         B 0 6 0 0 0 0 1
         C 0 0 1 0 0 0 0
         D 0 0 0 0 0 0 0
         E 0 0 0 0 2 0 0
         F 0 0 0 1 0 2 0
         G 0 0 0 0 0 0 0
```

```
Overall Statistics for Testing set

               Accuracy : 0.9091
                 95% CI : (0.7084, 0.9888)
    No Information Rate : 0.4091
    P-Value [Acc > NIR] : 1.485e-06

                  Kappa : 0.8743
 Mcnemar's Test P-Value : NA



Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E Class: F
Sensitivity            1.0000   1.0000  1.00000  0.00000  1.00000  1.00000
Specificity            1.0000   0.9375  1.00000  1.00000  1.00000  0.95000

                     Class: G
Sensitivity           0.00000
Specificity           1.00000
```

3. Flexible Discriminant Analysis

```
74 samples
 6 predictor
 7 classes: 'A', 'B', 'C', 'D', 'E', 'F', 'G'

Pre-processing : Center and Scale

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 74, 74, 74, 74, 74, 74, ...

Resampling results across tuning parameters:

  nprune  Accuracy   Kappa      Accuracy SD  Kappa SD
   2      0.5788969  0.3919771  0.12983654   0.17813099
   7      0.9212303  0.8937645  0.04500279   0.06016160
  13      0.9237493  0.8969303  0.04157087   0.05550361

Tuning parameter 'degree' was held constant at a value of 1
Kappa was used to select the optimal model using  the largest value.
The final values used for the model were degree = 1 and nprune = 13.
```

Confusion Matrix and Statistics **Testing set**

```
          Reference
Prediction A B C D E F G
         A 9 0 0 0 0 0 0
         B 0 6 0 0 0 0 0
         C 0 0 1 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 1
```

```
Overall Statistics
               Accuracy : 1
                  Kappa : 1
Statistics by Class:
                     Class: A Class: B Class: C Class: D Class: E Class: F Class: G
Sensitivity            1.0000   1.0000  1.00000  1.00000  1.00000  1.00000  1.00000
Specificity            1.0000   1.0000  1.00000  1.00000  1.00000  1.00000  1.00000
```

fda variable importance

```
           Overall
Palmitic    100.00
Oleic        93.04
Linolenic    84.83
Stearic      74.44
Eicosenoic   28.78
Eicosanoic    0.00
```

5. Support Vector Machines with Radial Basis Function Kernel

```
74 samples
 6 predictor
 7 classes: 'A', 'B', 'C', 'D', 'E', 'F', 'G'

Pre-processing : Center and Scale
Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 74, 74, 74, 74, 74, 74, ...

Resampling results across tuning parameters:

  C        Accuracy   Kappa       Accuracy SD  Kappa SD
   0.0625  0.3675221  0.00000000  0.06691789   0.00000000
   0.1250  0.3884587  0.03673462  0.08308983   0.08448864
   0.2500  0.6864940  0.54756831  0.14405452   0.18809785
   0.5000  0.8072887  0.73374931  0.11593993   0.15174910
   1.0000  0.8937349  0.85520193  0.09786982   0.13207147
   2.0000  0.9219243  0.89510608  0.07700160   0.10067752
   4.0000  0.9315190  0.90754581  0.07822848   0.10274981
   8.0000  0.9400412  0.90573022  0.07800019   0.10220908
  16.0000  0.9315797  0.90795720  0.07901996   0.10368099

Tuning parameter 'sigma' was held constant at a value of 0.033386
Kappa was used to select the optimal model using  the largest value.
The final values used for the model were sigma = 0.033386 and C = 8.
```

Confusion Matrix and Statistics **Testing set**

```
          Reference
Prediction A B C D E F G
         A 9 0 0 0 0 0 0
         B 0 6 0 0 0 0 0
         C 0 0 1 0 0 0 0
         D 0 0 0 0 0 0 0
         E 0 0 0 0 2 0 0
         F 0 0 0 0 0 2 0
         G 0 0 0 1 0 0 1

Overall Statistics

               Accuracy : 0.9545
                 95% CI : (0.7716, 0.9988)
    No Information Rate : 0.4091
    P-Value [Acc > NIR] : 9.454e-08

                  Kappa : 0.9382
 Mcnemar's Test P-Value : NA
```

```
Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E Class: F
Sensitivity            1.0000   1.0000  1.00000  0.00000  1.00000  1.00000
Specificity            1.0000   1.0000  1.00000  1.00000  1.00000  1.00000

                     Class: G
Sensitivity           1.00000
Specificity           0.95238
```

6. k-Nearest Neighbors

```
74 samples
 6 predictor
 7 classes: 'A', 'B', 'C', 'D', 'E', 'F', 'G'

Pre-processing : Center and Scale

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 74, 74, 74, 74, 74, 74, ...

Resampling results across tuning parameters:

  k    Accuracy   Kappa         Accuracy SD  Kappa SD
    3  0.9437413   0.921563597  0.03115882   0.04654870
    5  0.8974869   0.859998705  0.06600801   0.08908393
    9  0.8526078   0.800567045  0.07350034   0.09467525
   13  0.7864498   0.710750040  0.11679834   0.15233482
   17  0.7380869   0.640733236  0.08764073   0.11421086
   21  0.6747135   0.541849740  0.10598175   0.14161552
   41  0.5024954   0.283891081  0.12272963   0.13262620
   61  0.3530650   0.001767019  0.06639894   0.01560056
   81  0.3461684  -0.002319588  0.06692049   0.01037351
  101  0.3522029   0.002561380  0.07035510   0.02483029

  401  0.3547891   0.009154930  0.06847409   0.04094209
  451  0.3461684  -0.009472656  0.06692049   0.04236301

Kappa was used to select the optimal model using  the largest value.
The final value used for the model was k = 3.
```

Confusion Matrix and Statistics **Testing set**

```
          Reference
Prediction A B C D E F G
         A 9 0 0 0 0 0 1
         B 0 6 0 0 0 0 0
         C 0 0 1 0 0 0 0
         D 0 0 0 0 0 0 0
         E 0 0 0 0 2 0 0
         F 0 0 0 1 0 2 0
         G 0 0 0 0 0 0 0
```

```
Overall Statistics for Testing set :

               Accuracy : 0.9091
                 95% CI : (0.7084, 0.9888)
    No Information Rate : 0.4091
    P-Value [Acc > NIR] : 1.485e-06

                  Kappa : 0.8732
 Mcnemar's Test P-Value : NA

 Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E Class: F
 Sensitivity           1.0000   1.0000  1.00000  0.00000  1.00000  1.00000
 Specificity           0.9231   1.0000  1.00000  1.00000  1.00000  0.95000

                     Class: G
 Sensitivity          0.00000
 Specificity          1.00000
```

7.Naive Bayes Classifier for Discrete Predictors

```
 Call:
 naiveBayes.default(x = trainX, y = trainY)

 A-priori probabilities:
 trainY
         A          B          C          D          E          F
 0.37837838 0.27027027 0.02702703 0.08108108 0.12162162 0.10810811
         G
 0.01351351
   Conditional probabilities:


       Palmitic
 trainY     [,1]       [,2]
     A 10.95714 1.36474894
     B  6.29000 0.36259300
     C  9.65000 0.07071068
     D 11.90000 1.56588633
     E 10.41111 0.69362173
     F  5.11250 0.40510140
     G 10.00000         NA
```

```
      Stearic
trainY     [,1]       [,2]
     A 5.335714 0.58004743
     B 4.050000 0.40457905
     C 3.350000 0.07071068
     D 2.783333 0.14719601
     E 3.988889 0.26193723
     F 1.925000 0.20528726
     G 2.300000         NA

      Oleic
trainY     [,1]     [,2]
     A 33.38929 4.391434
     B 26.25000 1.883865
     C 58.50000 1.131371
     D 73.90000 3.055487
     E 25.81111 2.010873
     F 58.87500 4.089272
     G 36.90000       NA

      Linolenic
trainY     [,1]       [,2]
     A 1.014286 1.00764275
     B 0.635000 0.51633832
     C 0.150000 0.07071068
     D 0.700000 0.08944272
     E 6.766667 0.79056942
     F 8.312500 0.99058064
     G 2.200000         NA

      Eicosanoic
trainY      [,1]      [,2]
     A 0.4142857 0.2731358
     B 0.3550000 0.5835238
     C 1.5000000 0.0000000
     D 0.1500000 0.1224745
     E 0.3111111 0.2204793
     F 0.4375000 0.2924649
     G 0.5000000        NA

      Eicosenoic
trainY      [,1]       [,2]
     A 0.1821429 0.14920424
     B 0.2000000 0.17770466
     C 1.5000000 0.42426407
     D 0.1333333 0.08164966
     E 0.2444444 0.26977357
     F 1.0000000 0.65246784
     G 0.5000000         NA
```

Confusion Matrix and Statistics **Testing set**

```
          Reference
Prediction A B C D E F G
         A 9 0 0 1 0 0 1
         B 0 6 0 0 0 0 0
         C 0 0 1 0 0 0 0
         D 0 0 0 0 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 for Testing set :

```
               Accuracy : 0.9091
                 95% CI : (0.7084, 0.9888)
    No Information Rate : 0.4091
    P-Value [Acc > NIR] : 1.485e-06

                  Kappa : 0.8706
 Mcnemar's Test P-Value : NA
```

Statistics by Class:

	Class: A	Class: B	Class: C	Class: D	Class: E	Class: F	Class: G
Sensitivity	1.0000	1.0000	1.00000	0.00000	1.00000	1.00000	0.00000
Specificity	0.8462	1.0000	1.00000	1.00000	1.00000	1.00000	1.00000

LINEAR MODELS	Kappa	Accuracy
LDA	1	1
PLSDA	0.7413	0.8182
Penalised Models	0.8764	0.9091
NSC	0.9391	0.9545

Non – LINEAR MODELS	Kappa	Accuracy
MDA	1	1
NNet	0.8743	0.9091
FDA	1	1
SVM	0.9382	0.9545
KNN	0.8732	0.9091
Naïve Bayes	0.8706	0.9888

a).Based on Kappa and accuracy MDA and FDA are best models for this dataset from Non linear models. LDA from Linear models have similar performance. Two of the Non Linear models were able to do perfect classification and one of the linear models also achieved the same. Also performance difference in other models is very close. So I assume this is LINEAR BOUNDARY.

b).

Best predice Oil type : A

Least accurate oil type : G

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


#biological predictors

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

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

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

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

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

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

#----fda----

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

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

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

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


library(AppliedPredictiveModeling)
data(hepatic)

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

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

```


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

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

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

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

#----fda----

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

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

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

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

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

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


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

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

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

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

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

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


#----fda----

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

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

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

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


```

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


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

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

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


#----fda----

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

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

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

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

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


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

```
```
```