Homework Assignment for Chapter 13

13.1 Hepatic Injury

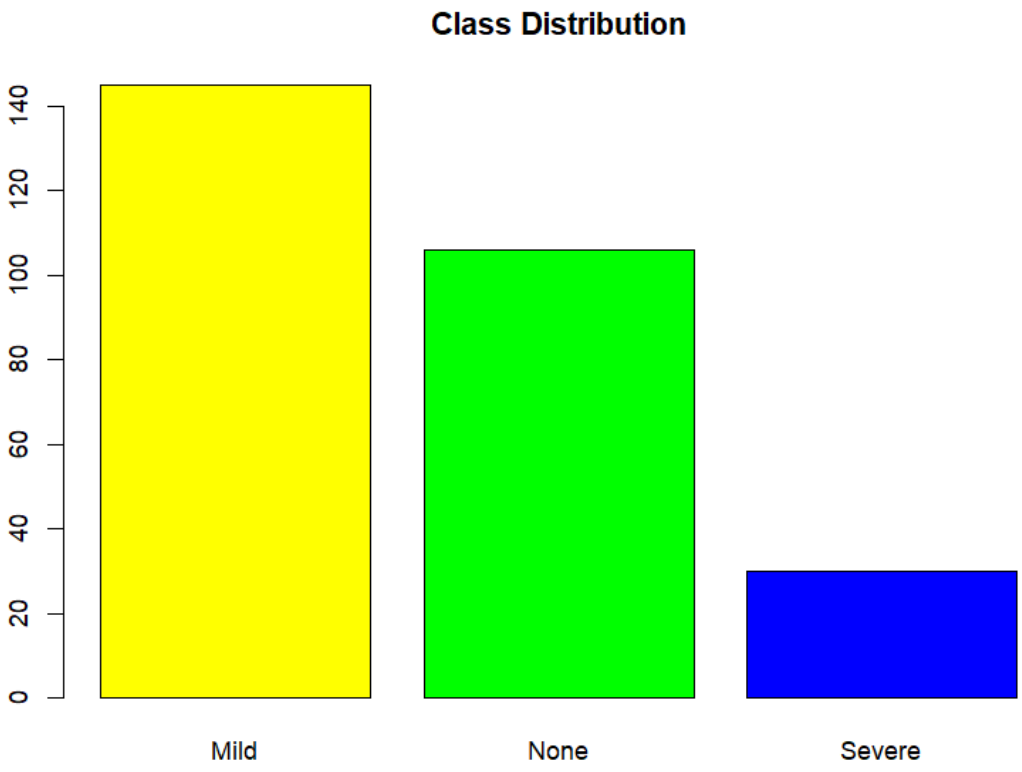


Fig 1: Distribution of Difference classes in the response variable injury

a) Model building

MDA

```
Mixture Discriminant Analysis

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

No pre-processing
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 169, 169, 169, 169, 169, 169, ...
Resampling results across tuning parameters:

  subclasses  logLoss    AUC        prAUC      Accuracy   Kappa       Mean_F1    Mean_Sensitivity
   1           4.178933  0.5376305  0.3495797  0.4035714  0.00803808  0.3477111  0.3529611
   2           5.961177  0.5429546  0.3587080  0.4214286  0.03755265  0.3817567  0.3688342
   3           7.266982  0.5597702  0.3671220  0.4450000  0.05735752  0.3953140  0.3676190
   4           8.231726  0.5635855  0.3708688  0.4450000  0.06064775  0.3999887  0.3811823
   5           9.267994  0.5602360  0.3627392  0.4457143  0.05788837  0.3977113  0.3821675
   6          10.265879  0.5678326  0.3589256  0.4414286  0.04318430  0.3941713  0.3790695
   7          11.079285  0.5662708  0.3478225  0.4428571  0.04885807  0.4002747  0.3716913
   8          12.140675  0.5517320  0.3333661  0.4371429  0.03142476  0.3904543  0.3652107
   9          12.605192  0.5693813  0.3252889  0.4528571  0.05298365  0.4100283  0.3770772
  10          13.320107  0.5609963  0.3172712  0.4421429  0.04028470  0.3940754  0.3680679
  11          13.663628  0.5674444  0.3172243  0.4514286  0.04937839  0.4197958  0.3833826
  12          14.407416  0.5614268  0.2988344  0.4428571  0.03636062  0.4154443  0.3761138
  13          15.234782  0.5544155  0.2831353  0.4364286  0.02097973  0.3872224  0.3589491
  14          15.708051  0.5470113  0.2780329  0.4371429  0.02896333  0.3884731  0.3586973
  Mean_Specificity  Mean_Pos_Pred_Value  Mean_Neg_Pred_Value  Mean_Precision  Mean_Recall
  0.6677220         0.3477920            0.6677495            0.3477920       0.3529611
  0.6788092         0.3633894            0.6780216            0.3633894       0.3688342
  0.6869474         0.3639282            0.6860365            0.3639282       0.3676190
  0.6862674         0.3797372            0.6859527            0.3797372       0.3811823
  0.6845982         0.3789535            0.6848840            0.3789535       0.3821675
  0.6789093         0.3739619            0.6794499            0.3739619       0.3790695
  0.6817115         0.3680451            0.6821630            0.3680451       0.3716913
  0.6752720         0.3588369            0.6758429            0.3588369       0.3652107
  0.6834004         0.3803254            0.6834082            0.3803254       0.3770772
  0.6790688         0.3673439            0.6783030            0.3673439       0.3680679
  0.6808296         0.3851673            0.6809530            0.3851673       0.3833826
  0.6759252         0.3777984            0.6765182            0.3777984       0.3761138
  0.6717150         0.3593790            0.6719557            0.3593790       0.3589491
  0.6748289         0.3627629            0.6749317            0.3627629       0.3586973
  Mean_Detection_Rate  Mean_Balanced_Accuracy
  0.1345238            0.5103416
```

```
  0.1349238             0.5103416
  0.1404762             0.5238217
  0.1483333             0.5272832
  0.1483333             0.5337248
  0.1485714             0.5333829
  0.1471429             0.5289894
  0.1476190             0.5267014
  0.1457143             0.5202413
  0.1509524             0.5302388
  0.1473810             0.5235683
  0.1504762             0.5321061
  0.1476190             0.5260195
  0.1454762             0.5153320
  0.1457143             0.5167631

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was subclasses = 9.
```

```
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     14   11      4
    None     12    9      0
    Severe    3    1      2

Overall Statistics

               Accuracy : 0.4464
                 95% CI : (0.3134, 0.5853)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.8856

                  Kappa : 0.0451

 Mcnemar's Test P-Value : 0.7563

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.4828      0.4286       0.33333
Specificity               0.4444      0.6571       0.92000
Pos Pred Value            0.4828      0.4286       0.33333
Neg Pred Value            0.4444      0.6571       0.92000
Prevalence                0.5179      0.3750       0.10714
Detection Rate            0.2500      0.1607       0.03571
Detection Prevalence      0.5179      0.3750       0.10714
Balanced Accuracy         0.4636      0.5429       0.62667
```

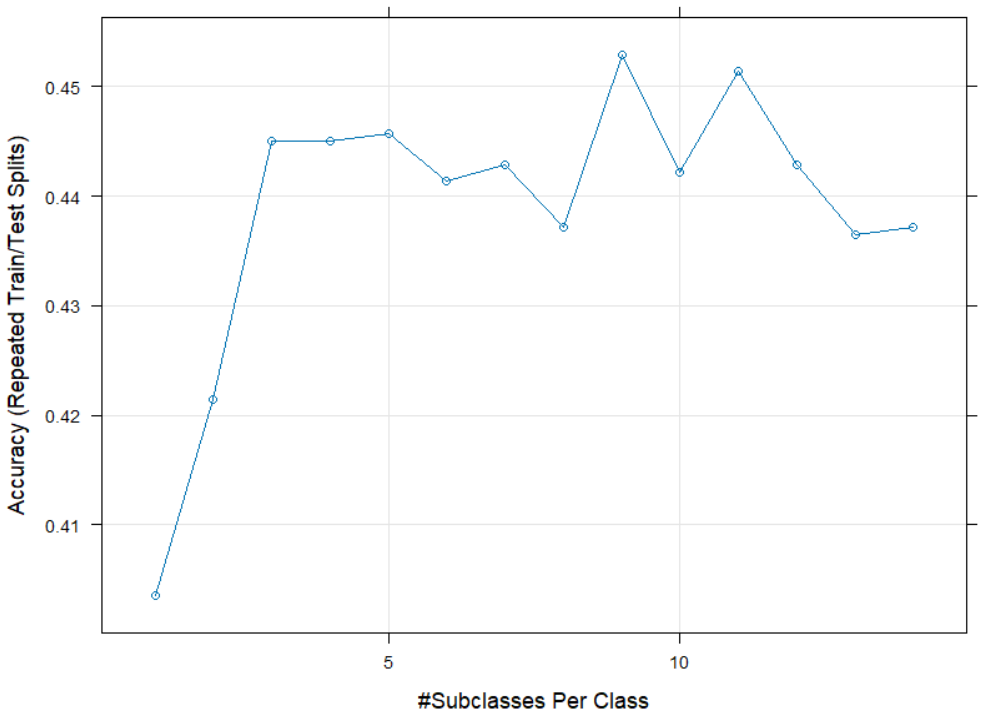


Fig 2: Plot of Accuracy vs Subclasses for MDA model

RDA

```
Regularized Discriminant Analysis

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

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...
Resampling results across tuning parameters:

  gamma  lambda  Accuracy   Kappa
  0.1    0.01    0.4874741  0.05031715
  0.1    0.02    0.4868943  0.05191302
  0.1    0.03    0.4884157  0.05904163
  0.1    0.04    0.4850267  0.05626032
  0.1    0.05    0.4829684  0.05444879
  0.1    0.06    0.4809693  0.05216068
  0.1    0.07    0.4796206  0.05089113
  0.1    0.08    0.4797085  0.05179773
  0.1    0.09    0.4778953  0.04958311
  0.1    0.10    0.4750367  0.04660817
  0.2    0.01    0.4808991  0.06328720
  0.2    0.02    0.4822783  0.06719870
  0.2    0.03    0.4804187  0.06504344
  0.2    0.04    0.4800186  0.06752597
  0.2    0.05    0.4801165  0.06808543
  0.2    0.06    0.4782244  0.06639986
  0.2    0.07    0.4777063  0.06754982
  0.2    0.08    0.4767251  0.06590818
  0.2    0.09    0.4785982  0.06991458
  0.2    0.10    0.4761319  0.06687623
  0.3    0.01    0.4765511  0.07184971
  0.3    0.02    0.4761650  0.07162001
  0.3    0.03    0.4733846  0.06854029
  0.3    0.04    0.4699619  0.06340166
  0.3    0.05    0.4684744  0.06092653
  0.3    0.06    0.4689290  0.06149693
  0.3    0.07    0.4669791  0.06043329
  0.3    0.08    0.4670166  0.06148844
  0.3    0.09    0.4659894  0.06104998
  0.3    0.10    0.4669284  0.06460421
  0.4    0.01    0.4677126  0.06715168
  0.4    0.02    0.4653623  0.06442657
  0.4    0.03    0.4649470  0.06473755
  0.4    0.04    0.4639436  0.06431922
  0.4    0.05    0.4634011  0.06423774
```

```
  0.9    0.05    0.4338284  0.05131360
  0.9    0.06    0.4338284  0.05131360
  0.9    0.07    0.4328438  0.04923561
  0.9    0.08    0.4323787  0.04813720
  0.9    0.09    0.4333396  0.04961283
  0.9    0.10    0.4328004  0.04858869
  1.0    0.01    0.4335286  0.04451921
  1.0    0.02    0.4335286  0.04451921
  1.0    0.03    0.4335286  0.04451921
  1.0    0.04    0.4335286  0.04451921
  1.0    0.05    0.4340414  0.04504590
  1.0    0.06    0.4340414  0.04504590
  1.0    0.07    0.4340414  0.04504590
  1.0    0.08    0.4340414  0.04504590
  1.0    0.09    0.4354856  0.04652657
  1.0    0.10    0.4354856  0.04652657

Accuracy was used to select the optimal model using the largest value.
The final values used for the model were gamma = 0.1 and lambda = 0.03.
```

```
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     19   12      5
    None      8    9      0
    Severe    2    0      1

Overall Statistics

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

                  Kappa : 0.1194

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.6552       0.4286        0.16667
Specificity               0.3704       0.7714        0.96000
Pos Pred Value            0.5278       0.5294        0.33333
Neg Pred Value            0.5000       0.6923        0.90566
Prevalence                0.5179       0.3750        0.10714
Detection Rate            0.3393       0.1607        0.01786
Detection Prevalence      0.6429       0.3036        0.05357
Balanced Accuracy         0.5128       0.6000        0.56333
```

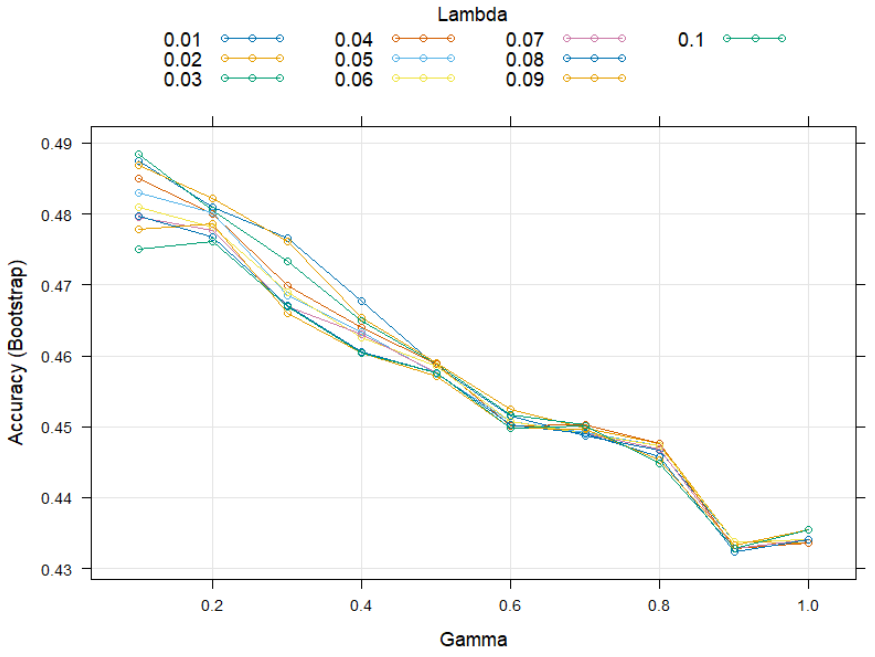


Fig 3: Accuracy vs Gamma for RDA model

Neural Networks

```
Neural Network

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

Pre-processing: centered (96), scaled (96), spatial sign transformation (96)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 169, 169, 169, 169, 169, 169, ...
Resampling results across tuning parameters:
```

size	decay	logLoss	AUC	prAUC	Accuracy	Kappa	Mean_F1	Mean_Sensitivity
1	0.0	6.9067257	0.5146908	0.2130704	0.4085714	0.015590316	0.3044242	0.3452436
1	0.1	1.4213475	0.5219499	0.3453776	0.4764286	0.033731933	NaN	0.3488779
1	1.0	0.9479833	0.5209672	0.3530608	0.5178571	0.000000000	NaN	0.3333333
1	2.0	0.9489150	0.5241853	0.3358213	0.5178571	0.000000000	NaN	0.3333333
2	0.0	11.3579795	0.5401471	0.2758850	0.4342857	0.073753866	0.3748754	0.3751724
2	0.1	1.5610775	0.5609116	0.3610982	0.4635714	0.058987106	0.4116880	0.3659880
2	1.0	0.9459389	0.5600953	0.3677187	0.5178571	0.001709760	NaN	0.3340339
2	2.0	0.9485471	0.5216771	0.3333431	0.5178571	0.000000000	NaN	0.3333333
3	0.0	15.0001059	0.5243451	0.2894801	0.4292857	0.030423121	0.3732720	0.3520307
3	0.1	1.6036631	0.5561929	0.3564728	0.4650000	0.063372105	0.4083882	0.3668966
3	1.0	0.9457598	0.5599702	0.3652539	0.5178571	0.001709760	NaN	0.3340339
3	2.0	0.9483343	0.5217013	0.3327701	0.5178571	0.000000000	NaN	0.3333333
4	0.0	14.0627456	0.5531144	0.3082265	0.4692857	0.078292832	0.4026298	0.3821894
4	0.1	1.5837189	0.5614894	0.3614097	0.4592857	0.051165150	0.3833732	0.3623536
4	1.0	0.9457460	0.5603142	0.3651098	0.5192857	0.005043185	NaN	0.3353038
4	2.0	0.9482002	0.5224848	0.3333829	0.5178571	0.000000000	NaN	0.3333333
5	0.0	13.9984631	0.5419934	0.3043854	0.4571429	0.073064915	0.4016557	0.3773618
5	0.1	1.5927907	0.5603283	0.3599704	0.4585714	0.046366378	0.3793465	0.3617187
5	1.0	0.9457042	0.5599158	0.3651410	0.5200000	0.006727395	NaN	0.3359387
5	2.0	0.9481112	0.5215531	0.3322930	0.5178571	0.000000000	NaN	0.3333333
6	0.0	14.4905680	0.5544497	0.3017605	0.4507143	0.064592687	0.3794914	0.3723481
6	0.1	1.5640626	0.5662920	0.3656767	0.4771429	0.079852674	0.3950074	0.3785988
6	1.0	0.9457180	0.5594470	0.3652778	0.5200000	0.006727395	NaN	0.3359387
6	2.0	0.9480494	0.5209716	0.3328669	0.5178571	0.000000000	NaN	0.3333333
7	0.0	14.1970160	0.5608607	0.3093591	0.4500000	0.069983536	0.4040049	0.3896771
7	0.1	1.5801650	0.5627807	0.3605709	0.4657143	0.065260520	0.3952149	0.3703558
7	1.0	0.9457266	0.5593762	0.3655169	0.5200000	0.006727395	NaN	0.3359387
7	2.0	0.9480056	0.5179086	0.3308895	0.5178571	0.000000000	NaN	0.3333333
8	0.0	13.9653341	0.5644837	0.3129434	0.4450000	0.051565325	0.3873478	0.3691954
8	0.1	1.5905984	0.5615815	0.3606004	0.4642857	0.056999157	0.3889334	0.3689217
8	1.0	0.9457425	0.5599985	0.3657543	0.5200000	0.006727395	NaN	0.3359387
8	2.0	0.9479734	0.5175584	0.3307302	0.5178571	0.000000000	NaN	0.3333333
9	0.0	13.6280599	0.5653923	0.3188494	0.4578571	0.064592732	0.4016033	0.3825835
9	0.1	1.6061619	0.5650530	0.3632408	0.4657143	0.061713769	0.3890032	0.3687794

```
  0.1530952             0.5227733
  0.1730952             0.5017048
  0.1726190             0.5000000
  0.1523810             0.5341391
  0.1528571             0.5215846
  0.1733333             0.5022692
  0.1726190             0.5000000
  0.1502381             0.5303801
  0.1590476             0.5355985
  0.1733333             0.5022692
  0.1726190             0.5000000
  0.1500000             0.5392463
  0.1552381             0.5291088
  0.1733333             0.5022692
  0.1726190             0.5000000
  0.1483333             0.5260242
  0.1547619             0.5269356
  0.1733333             0.5022692
  0.1726190             0.5000000
  0.1526190             0.5347810
  0.1552381             0.5278698
  0.1733333             0.5022692
  0.1726190             0.5000000
  0.1492857             0.5293807
  0.1559524             0.5297055
  0.1733333             0.5022692
  0.1726190             0.5000000

Accuracy was used to select the optimal model using the largest value.
The final values used for the model were size = 5 and decay = 1.
```

```
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     27   20      6
    None      2    1      0
    Severe    0    0      0

Overall Statistics

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

                  Kappa : -0.0208

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity              0.93103     0.04762        0.0000
Specificity              0.03704     0.94286        1.0000
Pos Pred Value           0.50943     0.33333           NaN
Neg Pred Value           0.33333     0.62264        0.8929
Prevalence               0.51786     0.37500        0.1071
Detection Rate           0.48214     0.01786        0.0000
Detection Prevalence     0.94643     0.05357        0.0000
Balanced Accuracy        0.48404     0.49524        0.5000
```

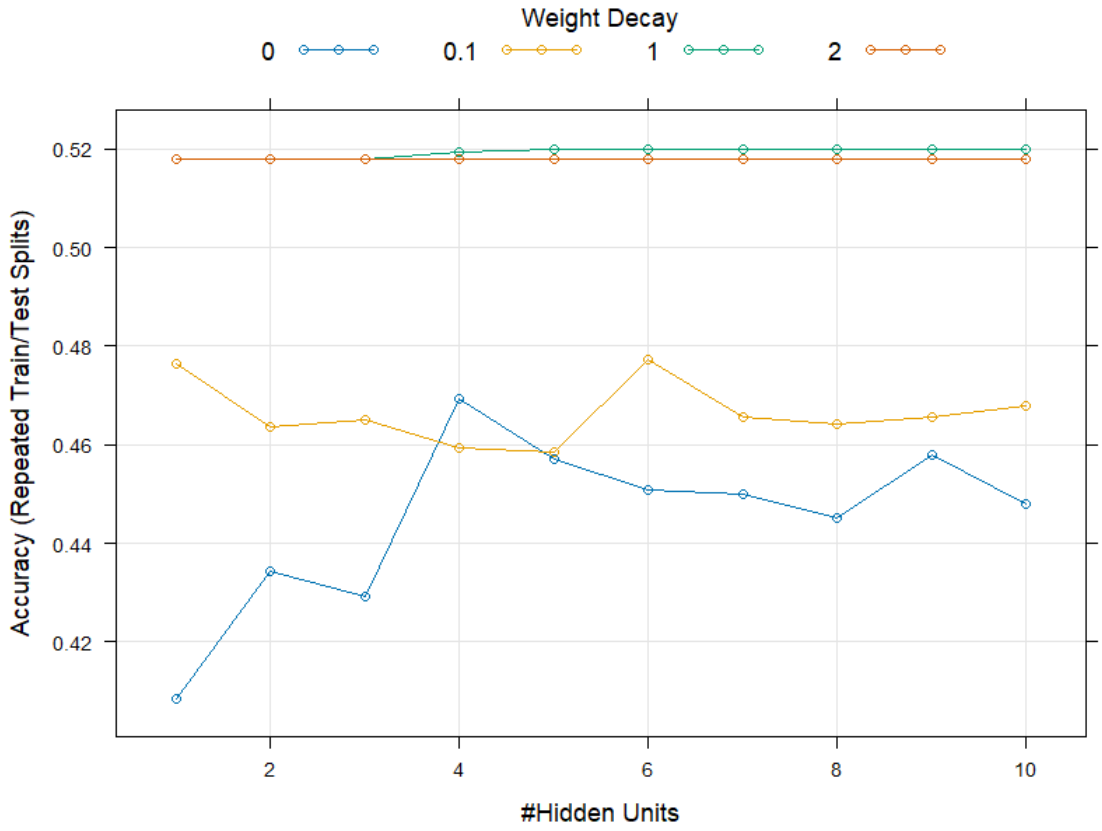


Fig 4: Accuracy vs Number of Hidden Units for Neural Networks

FDA

```
Flexible Discriminant Analysis

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

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 204, 203, 201, 202, 203, 202, ...
Resampling results across tuning parameters:

  degree  nprune  Accuracy   Kappa
  1        2      0.5075216   0.004884179
  1        3      0.5118695   0.016206664
  1        4      0.5162173   0.044165306
  1        5      0.5120671   0.046545197
  1        6      0.4944782   0.031727485
  1        7      0.4901303   0.026468028
  1        8      0.4901303   0.026468028
  1        9      0.4857825   0.022579589
  1       10      0.4903280   0.035177854
  1       11      0.4946758   0.040171469
  1       12      0.4859801   0.029673108
  1       13      0.4859801   0.029673108
  1       14      0.4859801   0.029673108
  1       15      0.4859801   0.029673108
  1       16      0.4859801   0.029673108
  1       17      0.4859801   0.029673108
  1       18      0.4859801   0.029673108
  1       19      0.4859801   0.029673108
  1       20      0.4859801   0.029673108
  1       21      0.4859801   0.029673108
  1       22      0.4859801   0.029673108
  1       23      0.4859801   0.029673108
  1       24      0.4859801   0.029673108
  1       25      0.4859801   0.029673108
  1       26      0.4859801   0.029673108
  1       27      0.4859801   0.029673108
  1       28      0.4859801   0.029673108
  1       29      0.4859801   0.029673108
  1       30      0.4859801   0.029673108
  1       31      0.4859801   0.029673108
  1       32      0.4859801   0.029673108
  1       33      0.4859801   0.029673108
  1       34      0.4859801   0.029673108
```

```
  2        10       0.4670078   0.005718410
  2        11       0.4806442   0.032643610
  2        12       0.4806442   0.032643610
  2        13       0.4806442   0.028159764
  2        14       0.4851896   0.040090981
  2        15       0.4851896   0.040090981
  2        16       0.4988260   0.062523694
  2        17       0.4988260   0.062523694
  2        18       0.4988260   0.062523694
  2        19       0.4988260   0.062523694
  2        20       0.4988260   0.062523694
  2        21       0.4988260   0.062523694
  2        22       0.4988260   0.062523694
  2        23       0.4988260   0.062523694
  2        24       0.4988260   0.062523694
  2        25       0.4988260   0.062523694
  2        26       0.4988260   0.062523694
  2        27       0.4988260   0.062523694
  2        28       0.4988260   0.062523694
  2        29       0.4988260   0.062523694
  2        30       0.4988260   0.062523694
  2        31       0.4988260   0.062523694
  2        32       0.4988260   0.062523694
  2        33       0.4988260   0.062523694
  2        34       0.4988260   0.062523694
  2        35       0.4988260   0.062523694
  2        36       0.4988260   0.062523694
  2        37       0.4988260   0.062523694
  2        38       0.4988260   0.062523694

Accuracy was used to select the optimal model using the largest value.
The final values used for the model were degree = 1 and nprune = 4.

Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     24   17      4
    None      2    1      0
    Severe    3    3      2

Overall Statistics

               Accuracy : 0.4821
                 95% CI : (0.3466, 0.6197)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.74822

                  Kappa : 0.0558

 Mcnemar's Test P-Value : 0.00183

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.8276     0.04762       0.33333
Specificity               0.2222     0.94286       0.88000
Pos Pred Value            0.5333     0.33333       0.25000
Neg Pred Value            0.5455     0.62264       0.91667
Prevalence                0.5179     0.37500       0.10714
Detection Rate            0.4286     0.01786       0.03571
Detection Prevalence      0.8036     0.05357       0.14286
Balanced Accuracy         0.5249     0.49524       0.60667
```

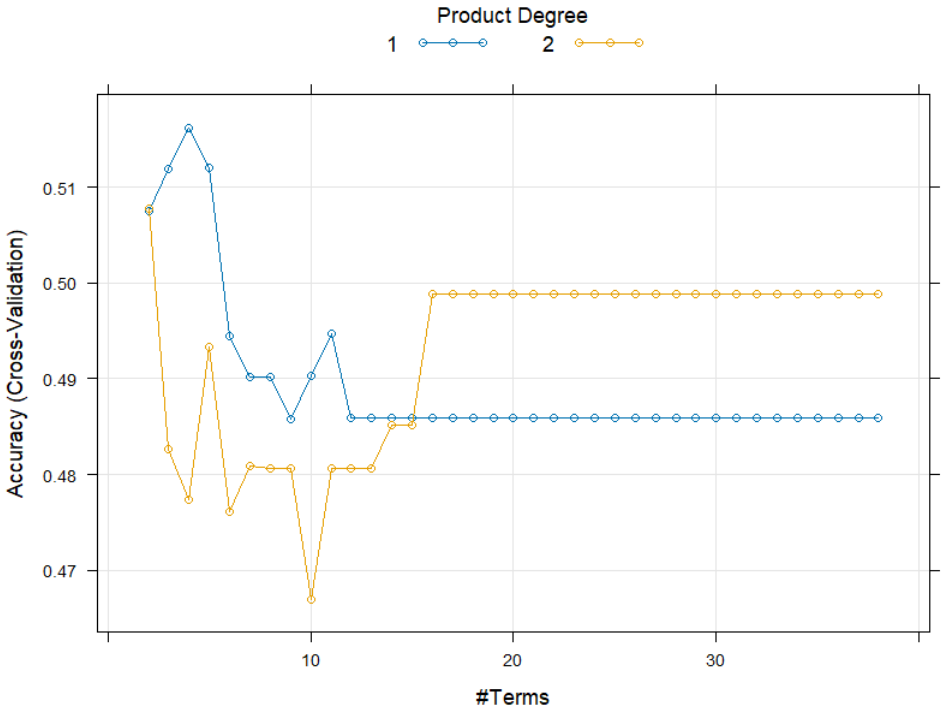


Fig 5: Accuracy vs Number of Terms for FDA model

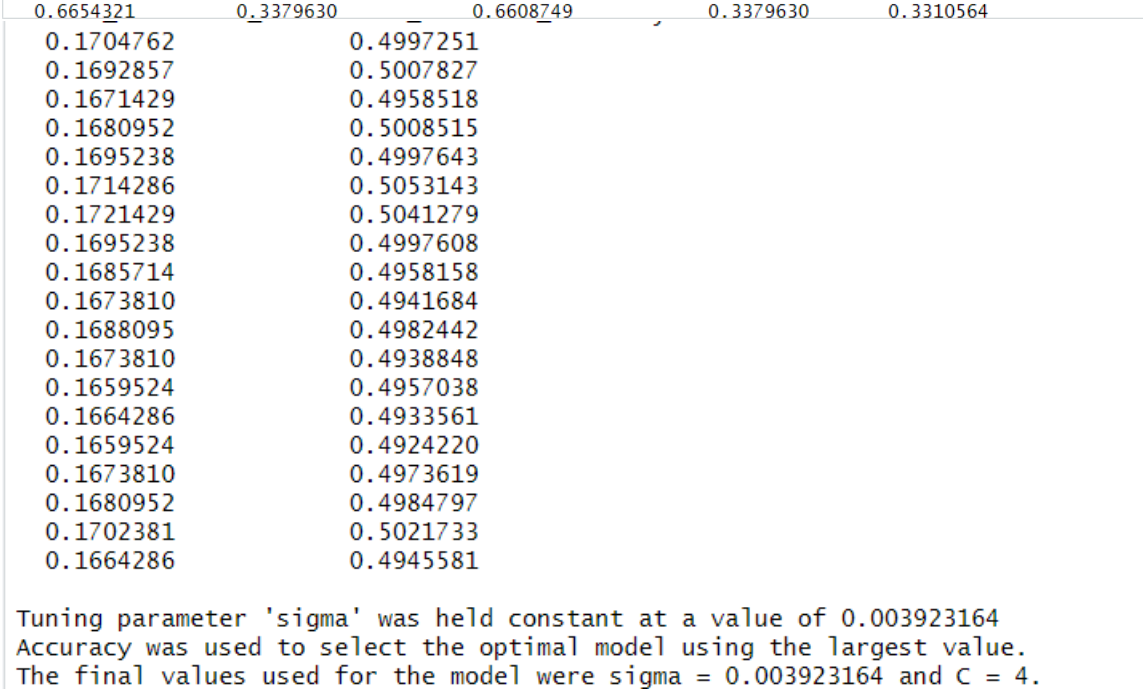
SVM

```
Support Vector Machines with Radial Basis Function Kernel

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

Pre-processing: centered (96), scaled (96)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 169, 169, 169, 169, 169, 169, ...
Resampling results across tuning parameters:

  C           logLoss    AUC        prAUC      Accuracy   Kappa          Mean_F1    Mean_Sensitivity
     0.0625   0.9451827  0.5487008  0.3530637  0.5114286  -0.0004991687        NaN  0.3330487
     0.1250   0.9475734  0.5462783  0.3567186  0.5078571   0.0031918851        NaN  0.3339026
     0.2500   0.9475976  0.5547264  0.3579220  0.5014286  -0.0113130144        NaN  0.3287137
     0.5000   0.9510538  0.5464115  0.3558365  0.5042857   0.0034489606  0.3886207  0.3342419
     1.0000   0.9483945  0.5396770  0.3495982  0.5085714   0.0005879288        NaN  0.3326108
     2.0000   0.9470089  0.5522489  0.3603225  0.5142857   0.0159662973        NaN  0.3390914
     4.0000   0.9460979  0.5546404  0.3527642  0.5164286   0.0123684544        NaN  0.3374932
     8.0000   0.9508793  0.5412411  0.3435207  0.5085714  -0.0002286653        NaN  0.3326108
    16.0000   0.9459449  0.5479181  0.3484454  0.5057143  -0.0123423802        NaN  0.3288451
    32.0000   0.9455865  0.5527670  0.3541242  0.5021429  -0.0158950304        NaN  0.3265463
    64.0000   0.9507604  0.5376813  0.3419671  0.5064286  -0.0038100361        NaN  0.3310564
   128.0000   0.9485094  0.5456791  0.3493259  0.5021429  -0.0175304205        NaN  0.3265463
   256.0000   0.9501055  0.5459662  0.3466640  0.4978571  -0.0114460868        NaN  0.3285167
   512.0000   0.9466037  0.5552995  0.3542447  0.4992857  -0.0188461116        NaN  0.3261084
  1024.0000   0.9519703  0.5424939  0.3436692  0.4978571  -0.0208656997        NaN  0.3246634
  2048.0000   0.9474419  0.5418183  0.3443299  0.5021429  -0.0069294694        NaN  0.3302244
  4096.0000   0.9490168  0.5441076  0.3508953  0.5042857  -0.0031066229        NaN  0.3314286
  8192.0000   0.9487912  0.5467959  0.3455668  0.5107143   0.0069171950        NaN  0.3353914
 16384.0000   0.9493023  0.5361636  0.3420038  0.4992857  -0.0151819113        NaN  0.3271593
 Mean_Specificity  Mean_Pos_Pred_Value  Mean_Neg_Pred_Value  Mean_Precision  Mean_Recall
 0.6664014         0.3190788            0.6411652            0.3190788       0.3330487
 0.6676628         0.3376378            0.6338724            0.3376378       0.3339026
 0.6629898               NaN            0.6135092                  NaN       0.3287137
 0.6674610         0.4558620            0.6655773            0.4558620       0.3342419
 0.6669178         0.4097222            0.6297972            0.4097222       0.3326108
 0.6715372         0.5092593            0.7010024            0.5092593       0.3390914
 0.6707626               NaN            0.6518247                  NaN       0.3374932
 0.6669108         0.2818930            0.6299098            0.2818930       0.3326108
 0.6627866               NaN            0.6331339                  NaN       0.3288451
 0.6617905         0.3189300            0.6325303            0.3189300       0.3265463
 0.6654321         0.3379630            0.6608749            0.3379630       0.3310564
```



```
  0.1704762             0.4997251
  0.1692857             0.5007827
  0.1671429             0.4958518
  0.1680952             0.5008515
  0.1695238             0.4997643
  0.1714286             0.5053143
  0.1721429             0.5041279
  0.1695238             0.4997608
  0.1685714             0.4958158
  0.1673810             0.4941684
  0.1688095             0.4982442
  0.1673810             0.4938848
  0.1659524             0.4957038
  0.1664286             0.4933561
  0.1659524             0.4924220
  0.1673810             0.4973619
  0.1680952             0.4984797
  0.1702381             0.5021733
  0.1664286             0.4945581

Tuning parameter 'sigma' was held constant at a value of 0.003923164
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.003923164 and C = 4.
```

```
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     23   15      6
    None      6    6      0
    Severe    0    0      0

Overall Statistics

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

                  Kappa : 0.0597

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.7931      0.2857        0.0000
Specificity               0.2222      0.8286        1.0000
Pos Pred Value            0.5227      0.5000           NaN
Neg Pred Value            0.5000      0.6591        0.8929
Prevalence                0.5179      0.3750        0.1071
Detection Rate            0.4107      0.1071        0.0000
Detection Prevalence      0.7857      0.2143        0.0000
Balanced Accuracy         0.5077      0.5571        0.5000
```

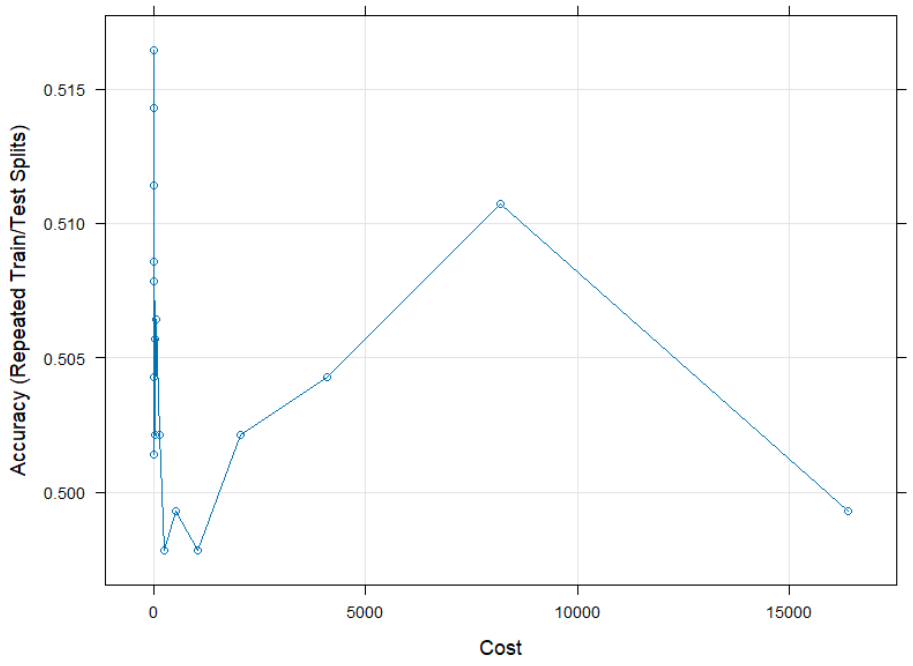


Fig 6: Accuracy vs Cost for SVM model

KNN

```
k-Nearest Neighbors

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

Pre-processing: centered (96), scaled (96)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 169, 169, 169, 169, 169, 169, ...
Resampling results across tuning parameters:
```

k	logLoss	AUC	prAUC	Accuracy	Kappa	Mean_F1	Mean_Sensitivity
1	18.9963270	0.5310711	0.1976138	0.4500000	0.053717583	0.3959582	0.3803503
2	11.3365627	0.5363018	0.2793809	0.4450000	0.048486713	0.3880853	0.3646196
3	7.3426198	0.5453810	0.3242071	0.4671429	0.066964085	0.4000421	0.3703996
4	5.0532838	0.5479730	0.3243801	0.4628571	0.046260806	0.3925813	0.3516037
5	3.5530166	0.5464189	0.3224836	0.4514286	0.023511925	0.4005005	0.3398577
6	2.6265778	0.5506113	0.3295840	0.4528571	0.015766640	0.3760155	0.3328517
7	2.0626004	0.5540827	0.3349943	0.4692857	0.036992421	0.3942972	0.3480131
8	1.8020623	0.5537039	0.3379326	0.4635714	0.018628177	0.3967499	0.3392337
9	1.6800351	0.5581678	0.3433392	0.4814286	0.043088900	0.3853038	0.3532020
10	1.5334626	0.5654217	0.3492420	0.4857143	0.040582373	0.3899712	0.3534975
11	1.4054843	0.5751680	0.3511997	0.4957143	0.059547125	0.4200033	0.3634483
12	1.3313716	0.5793099	0.3610371	0.5014286	0.065836519	0.4213799	0.3666010
13	1.3028985	0.5827957	0.3636950	0.5121429	0.081692382	0.4244858	0.3708484
14	1.2359870	0.5801250	0.3543548	0.5121429	0.078877813	0.4513379	0.3682102
15	1.1662873	0.5810417	0.3587676	0.5178571	0.086657887	0.4423274	0.3733005
16	1.1013638	0.5756614	0.3580422	0.5114286	0.069761292	NaN	0.3615982
17	1.1017772	0.5732400	0.3525260	0.4964286	0.040371924	NaN	0.3498413
18	1.0587828	0.5703087	0.3582241	0.5042857	0.053261487	NaN	0.3554242
19	1.0358610	0.5698573	0.3577822	0.5050000	0.051870680	NaN	0.3541325
20	1.0351075	0.5711431	0.3609683	0.5064286	0.055014249	NaN	0.3554023
21	1.0327581	0.5731885	0.3567892	0.5000000	0.041034067	NaN	0.3496880
22	1.0329074	0.5719367	0.3576886	0.4992857	0.040034498	NaN	0.3499288
23	1.0333360	0.5704217	0.3610005	0.5028571	0.044409162	NaN	0.3511768
24	1.0129892	0.5675493	0.3602013	0.4964286	0.032469647	NaN	0.3468637
25	1.0138692	0.5679919	0.3622698	0.4957143	0.029842873	NaN	0.3457033
26	1.0128046	0.5689153	0.3668276	0.4957143	0.028126724	NaN	0.3451779
27	0.9898370	0.5732480	0.3665719	0.4985714	0.029765880	NaN	0.3457909
28	0.9680365	0.5730823	0.3636153	0.5007143	0.035717724	NaN	0.3482211
29	0.9421735	0.5779769	0.3717445	0.5071429	0.046236034	NaN	0.3523591
30	0.9424923	0.5774535	0.3749702	0.5021429	0.036001862	NaN	0.3486152
31	0.9406746	0.5810367	0.3779631	0.4992857	0.028389580	NaN	0.3455501
32	0.9403977	0.5820836	0.3724056	0.5014286	0.032846997	NaN	0.3471045
33	0.9396675	0.5840786	0.3772028	0.5042857	0.036340334	NaN	0.3489436

0.1680952	0.5192782
0.1683333	0.5185971
0.1688095	0.5196694
0.1666667	0.5146465
0.1664286	0.5145200
0.1676190	0.5159270
0.1654762	0.5117176
0.1652381	0.5107212
0.1652381	0.5101762
0.1661905	0.5106803
0.1669048	0.5128619
0.1690476	0.5164758
0.1673810	0.5129319
0.1664286	0.5103553
0.1671429	0.5118733
0.1680952	0.5131596
0.1669048	0.5094114
0.1669048	0.5083158
0.1666667	0.5068620
0.1673810	0.5079791
0.1652381	0.5019792
0.1657143	0.5027070
0.1626190	0.4952826
0.1650000	0.4994298
0.1669048	0.5031371
0.1664286	0.5012066
0.1645238	0.4972677
0.1647619	0.4972560
0.1654762	0.4985171
0.1664286	0.4997976
0.1680952	0.5020512
0.1664286	0.4995407
0.1678571	0.5014304

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 15.

```
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     22   13      4
    None      7    8      1
    Severe    0    0      1

Overall Statistics

               Accuracy : 0.5536
                 95% CI : (0.4147, 0.6866)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.34484

                  Kappa : 0.1581

 Mcnemar's Test P-Value : 0.07855

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.7586      0.3810       0.16667
Specificity               0.3704      0.7714       1.00000
Pos Pred Value            0.5641      0.5000       1.00000
Neg Pred Value            0.5882      0.6750       0.90909
Prevalence                0.5179      0.3750       0.10714
Detection Rate            0.3929      0.1429       0.01786
Detection Prevalence      0.6964      0.2857       0.01786
Balanced Accuracy         0.5645      0.5762       0.58333
```

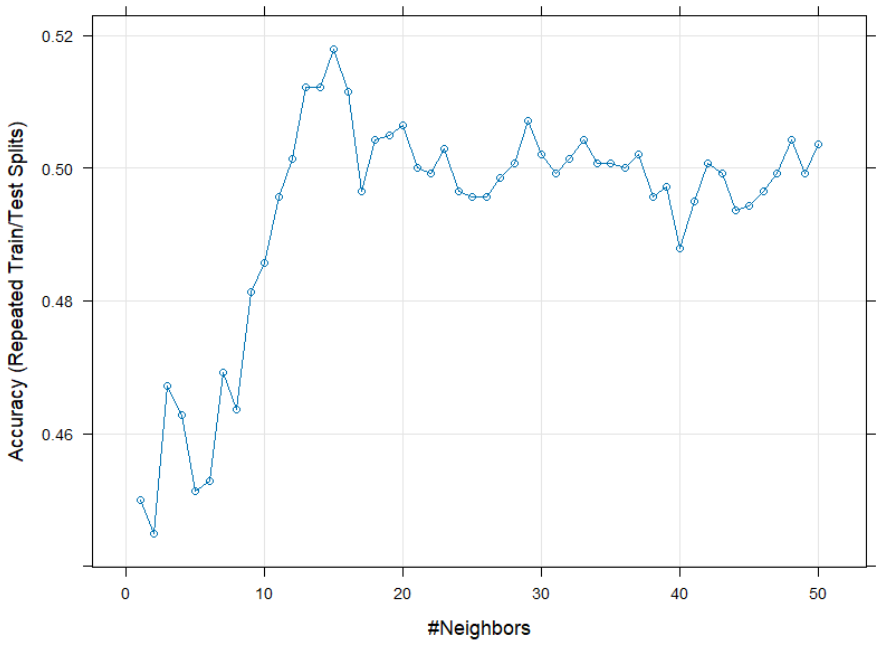


Fig 7: Accuracy vs Number of Neighbors for KNN model

Naïve Bayes

```
Naive Bayes

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

No pre-processing
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 169, 169, 169, 169, 169, 169, ...
Resampling results across tuning parameters:

  fL  logLoss   AUC        prAUC      Accuracy   Kappa        Mean_F1    Mean_Sensitivity  Mean_Specificit
  2   15.67154  0.4958025  0.3056379  0.3314286  -0.03444965  0.3017131  0.3225616         0.650781
  3   15.67154  0.4958025  0.3056379  0.3314286  -0.03444965  0.3017131  0.3225616         0.650781
  4   15.67154  0.4958025  0.3056379  0.3314286  -0.03444965  0.3017131  0.3225616         0.650781
  Mean_Pos_Pred_Value  Mean_Neg_Pred_Value  Mean_Precision  Mean_Recall  Mean_Detection_Rate
  0.3154702            0.6535535            0.3154702       0.3225616    0.1104762
  0.3154702            0.6535535            0.3154702       0.3225616    0.1104762
  0.3154702            0.6535535            0.3154702       0.3225616    0.1104762
  Mean_Balanced_Accuracy
  0.4866713
  0.4866713
  0.4866713

Tuning parameter 'usekernel' was held constant at a value of TRUE
Tuning parameter 'adjust' was held
 constant at a value of TRUE
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were fL = 2, usekernel = TRUE and adjust = TRUE.
```

```
Confusion Matrix and Statistics

          Reference
Prediction Mild None Severe
    Mild     15    6      2
    None      8    8      1
    Severe    6    7      3

Overall Statistics

               Accuracy : 0.4643
                 95% CI : (0.3299, 0.6026)
    No Information Rate : 0.5179
    P-Value [Acc > NIR] : 0.82537

                  Kappa : 0.1667

 Mcnemar's Test P-Value : 0.07905

Statistics by Class:

                     Class: Mild Class: None Class: Severe
Sensitivity               0.5172      0.3810       0.50000
Specificity               0.7037      0.7429       0.74000
Pos Pred Value            0.6522      0.4706       0.18750
Neg Pred Value            0.5758      0.6667       0.92500
Prevalence                0.5179      0.3750       0.10714
Detection Rate            0.2679      0.1429       0.05357
Detection Prevalence      0.4107      0.3036       0.28571
Balanced Accuracy         0.6105      0.5619       0.62000
```

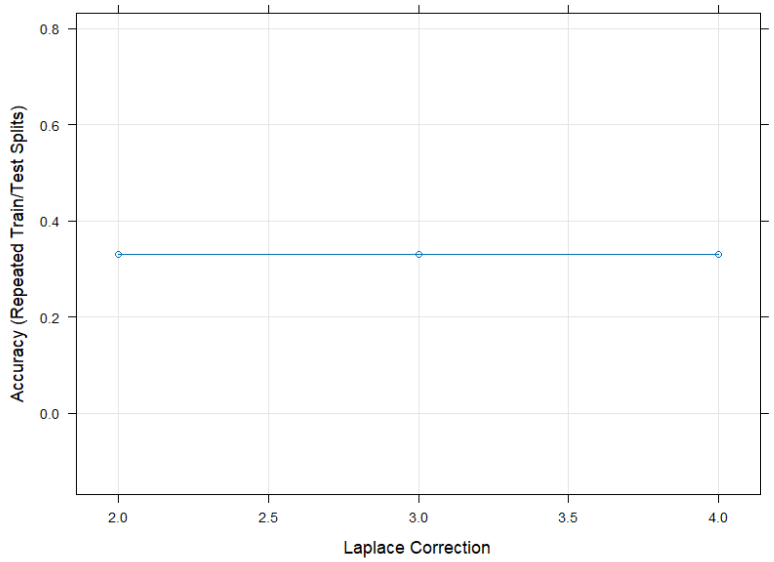


Fig 8: Accuracy for Naïve Bayes Model

Model	**Best Tuning Parameter**	**Training**		**Testing**	
		Kappa	**Accuracy**	**Kappa**	**Accuracy**
MDA	Subclasses=9	0.05298365	0.458571	0.0451	0.4464
RDA	Gamma=0.1, lambda=0.03	0.05904163	0.4884157	0.1194	0.5179
Neural Networks	Size=5, decay=1	0.006727395	0.5200	-0.0208	0.5000
FDA	Degree =1, nprune=4	0.044165306	0.5162173	0.0558	0.4821
SVM	Sigma=0.003923164, C=4	0.0123684544	0.5164286	0.0597	0.5179
KNN	K=15	0.086657887	0.5178571	0.1581	0.5536
Naïve Bayes	2	-0.03444965	0.3314286	0.1667	0.4643

Table 1: Model Summary

Table 1 gives a summary of all the models and their performance on both the training and testing datasets. From the table, KNN is the best performing model on the testing set with an accuracy of 55.36% and Kappa of 0.0579. On the training set, Neural Networks is the best performing model, with an accuracy of 52% and Kappa of 0.006727395, followed closely by KNN with an accuracy of 51.79% on the training set. RDA and SVM are the closest models to KNN on testing with an accuracy of 51.79% for both . Looking at the performance of the models, the best model would be KNN as it has a higher accuracy rate on testing and second-best accuracy on training which makes it the ideal model.

b) Improve classification

From the results of the model performance, it is evident that nonlinear structure of these models helps improve the classification performance. Nonlinear models have an average accuracy of 0.49746 while the linear models had an average accuracy of 0.4776. Average accuracy of the nonlinear models is much higher compared to the linear models.

c) Top five important predictors

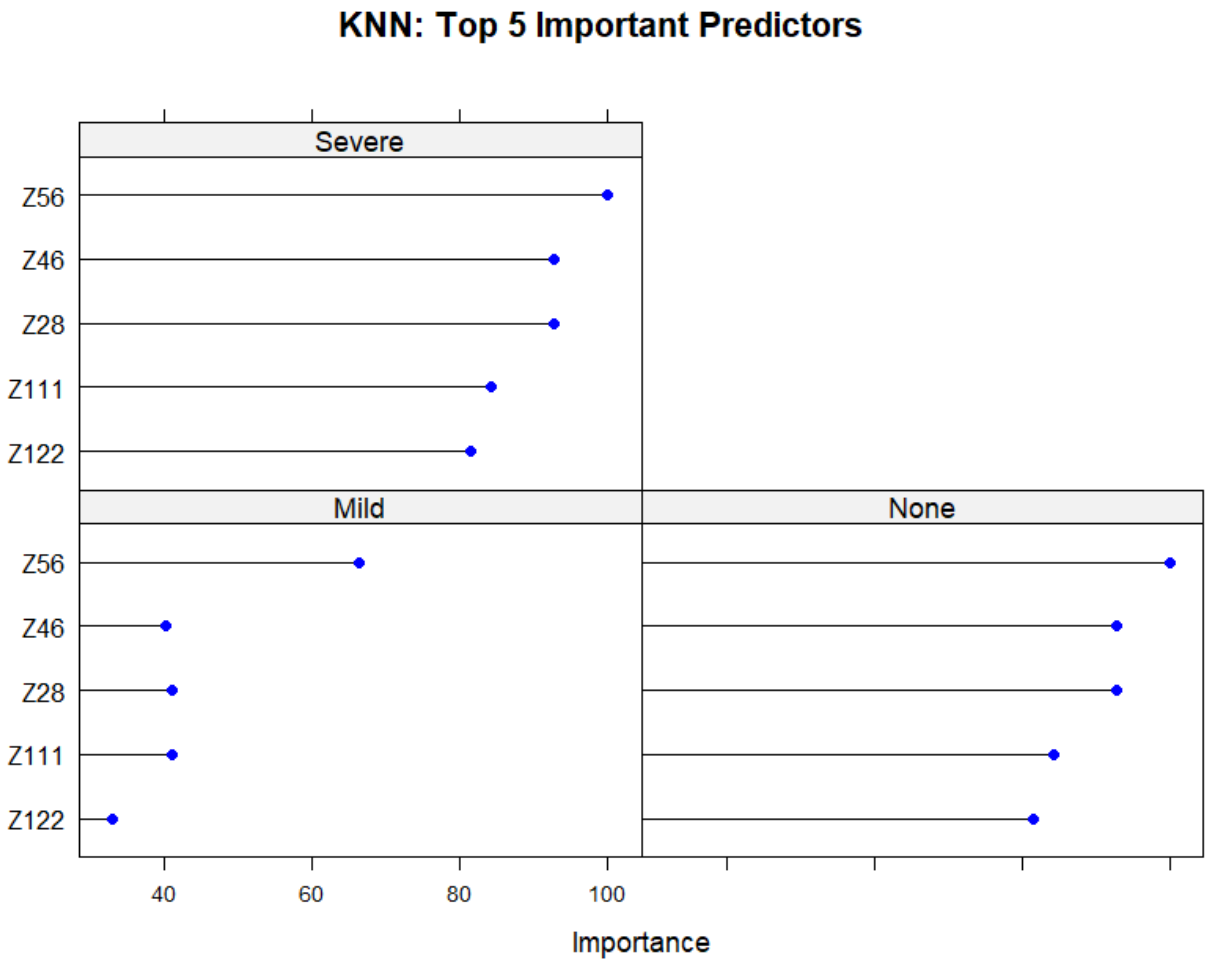


Fig 9: Top 5 important predictors for the best model(KNN)

The table shows the top five most important predictors for the KNN model for the three different classes.

Appendix

```
#####Question 13.1####
library(mda)
library(caret)
library(AppliedPredictiveModeling)
data(hepatic)
injury
table(injury)
barplot(table(injury), col=c('yellow','green','blue'),main='Class Distribution')
#part c: Nearzero & Corr
bio <- bio[, -nearZeroVar(bio)]
highCorrbio <- findCorrelation(cor(bio), cutoff=0.90)
bio<- bio[, -highCorrbio]
#checking for negative values
negatives_values <- sapply(bio, function(col) any(col < 0))
print(negatives_values)
library(e1071)
skewValues <- apply(bio, 2, skewness)
head(skewValues)
#####Transformation######
bio_trans <- preProcess(bio, method = c("BoxCox", "center", "scale", "spatialSign"))
bio_trans
bio_trsfmd <- predict(bio_trans, bio)
head(bio_trsfmd)
set.seed(100)
trainR <- createDataPartition(injury, p=0.8, list=FALSE)
X.train <- bio_trsfmd[trainR, ]
y.train <- injury[trainR]
```

```
X.test <- bio_trsfmd[-trainR, ]
y.test <- injury[-trainR]
ctrl <- trainControl(summaryFunction = multiClassSummary,classProbs = TRUE,
method='LGOCV',savePredictions = TRUE)

#####Nonlinear Classification Models######

####MDA####
library(caret)
set.seed(100)
mdaFit <- caret::train(x = X.train,
                       y = y.train,
                       method = "mda",
                       metric = "Accuracy",
                       tuneGrid = expand.grid(.subclasses = 1:14),
                       trControl = ctrl)
mdaFit
plot(mdaFit)
pred_bio<-predict(mdaFit,X.test)
confusionMatrix(data=pred_bio,
                reference=y.test)
##RDA##
install.packages("rda")
install.packages("rrcov")
library(rda)
library(caret)
```

```
set.seed(100)
ctrl <- trainControl(summaryFunction = defaultSummary,
                     classProbs = TRUE)
library(rrcov)
tunegrid <- expand.grid(
  .gamma = seq(0.1, 1, by = 0.1),    # Example values for gamma
  .lambda = seq(0.01, 0.1, by = 0.01) # Example values for lambda
)
rdaFit <- caret::train(X.train,
                y.train,
                method = "rda",
                metric = "Accuracy",
                tuneGrid = tunegrid,
                trControl = ctrl)
rdaFit
plot(rdaFit)
# prediction
predrda <- predict(rdaFit,newdata=X.test)



# confusion Matrix
confusionMatrix(predrda,y.test)

####NeuralNetworks#####
nnetGrid <- expand.grid(.size = 1:10, .decay = c(0, .1, 1, 2))
maxSize <- max(nnetGrid$.size)
numWts <- (maxSize * (96+ + 1) + (maxSize+1)*3)
nnetFit <- caret::train(x = X.train,
```

```
                    y = y.train,
                    method = "nnet",
                    metric = "Accuracy",
                    preProc = c("center", "scale", "spatialSign"),
                    tuneGrid = nnetGrid,
                    trace = FALSE,
                    maxit = 2000,
                    MaxNWts = numWts,
                    trControl = ctrl)
nnetFit
plot(nnetFit)
nnetpred <- predict(nnetFit,X.test)
nnetpred
confusionMatrix(nnetpred,y.test)

####FDA####
library(mda)
library(earth)
marsGrid <- expand.grid(.degree = 1:2, .nprune = 2:38)
fdaTuned <- caret::train(x = X.train,
                     y = y.train,
                     method = "fda",
                     metric="Accuracy",
                     tuneGrid = marsGrid,
                     trControl = trainControl(method = "cv"))
fdaTuned
plot(fdaTuned)
fdaPred <- predict(fdaTuned, newdata = X.test)
```

```
confusionMatrix(data = fdaPred,reference =y.test)



#####SVM#####
set.seed(100)
library(kernlab)
library(caret)
sigmaRangeReduced <- sigest(as.matrix(X.train[,1:96]))
svmRGridReduced <- expand.grid(.sigma = sigmaRangeReduced[1],
                               .C = 2^seq(-4, 14))
svmRModel <- caret::train(x = X.train,
                          y =y.train,
                          method = "svmRadial",
                          metric = "Accuracy",
                          preProc = c("center", "scale"),
                          tuneGrid = svmRGridReduced,
                          fit = FALSE,
                          trControl = ctrl)
svmRModel
plot(svmRModel)
svmPred <- predict(svmRModel, newdata = X.test)
confusionMatrix(data = svmPred,reference =y.test)



####KNN####
library(caret)
set.seed(100)
knnFit <- caret::train(x = X.train,
```

```
                    y = y.train,
                    method = "knn",
                    metric = "Accuracy",
                    preProc = c("center", "scale"),
                    ##tuneGrid = data.frame(.k = c(4*(0:5)+1, 20*(1:5)+1, 50*(2:9)+1)), ## 21 is the
best
                    tuneGrid = data.frame(.k = 1:50),
                    trControl = ctrl)
knnFit
plot(knnFit)
knnpred <- predict(knnFit,X.test)
knnpred
confusionMatrix(knnpred,
                y.test)
######Naive Bayes#####
#install.packages("klaR")
# Create a tuning grid for Naive Bayes
nbGrid <- expand.grid(
  .fL =c(2,3,4),
  .usekernel = TRUE,
  .adjust = TRUE
)
library(klaR)
set.seed(100)
nbFit <- caret::train( x = X.train,
                    y = y.train,
                    method = "nb",
                    metric = "Accuracy",
                    ## preProc = c("center", "scale"),
```

```
                ##tuneGrid = data.frame(.k = c(4*(0:5)+1, 20*(1:5)+1, 50*(2:9)+1)), ## 21 is the
best
                tuneGrid = nbGrid,
                trControl = ctrl)
nbFit
plot(nbFit)
nbPred <- predict(nbFit, newdata = X.test)
confusionMatrix(data = nbPred,reference =y.test)

###Important Predictors###



knn_ImpVals=varImp(knnFit)
knn_ImpVals
# top 5.

plot(knn_ImpVals,
    top = 5,
    scales = list(y = list(cex = .95)),
    col = "blue",
    main="KNN: Top 5 Important Predictors "
)
```