sonarmodels

Mike Simko

8/14/2018

##Introduction

An exercise to evaluate various classificartion models in the CARET package using the sonar dataset.

The sonar dataset (used by Gorman and Sejnowski) has 208 rows and 61 columns of numeric values between 0 an 1 with a final nominal column called “Class” which is recorded as either “R” for rock or “M” for mine/metal. • http://www.ics.uci.edu/~mlearn/MLRepository.html

#Import dataset

#Split data into training and testing sets

```
## There are 146 rows in the training data set
```

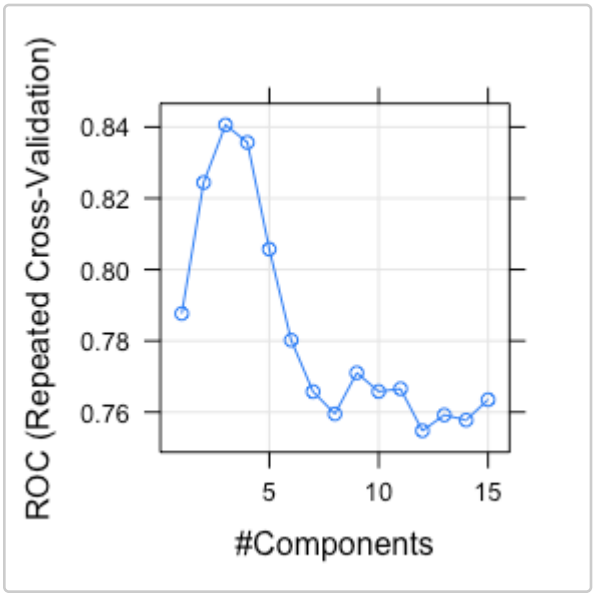
```
## There are 62 rows in the testing data set
```

#Compare four common classification models to one another

```
## Partial Least Squares
##
## 146 samples
##  60 predictor
##   2 classes: 'M', 'R'
##
## Pre-processing: centered (60), scaled (60)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 131, 131, 131, 131, 132, 131, ...
## Resampling results across tuning parameters:
##
##   ncomp  ROC        Sens       Spec
##    1     0.7876559  0.7089286  0.6396825
##    2     0.8244331  0.7755952  0.7492063
##    3     0.8405896  0.8035714  0.7476190
##    4     0.8356718  0.8148810  0.7428571
##    5     0.8057398  0.7607143  0.7071429
##    6     0.7802012  0.7392857  0.6960317
##    7     0.7657596  0.7511905  0.6650794
##    8     0.7595096  0.7434524  0.6746032
##    9     0.7710176  0.7440476  0.6904762
##   10     0.7658022  0.7434524  0.6603175
##   11     0.7665533  0.7517857  0.6658730
##   12     0.7547761  0.7303571  0.6650794
```

```
##   13     0.7591978  0.7261905  0.6452381
##   14     0.7577664  0.7297619  0.6404762
##   15     0.7634921  0.7202381  0.6404762
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was ncomp = 3.
```

```
##  Factor w/ 2 levels "M","R": 2 1 2 2 1 2 2 2 2 2 ...
```



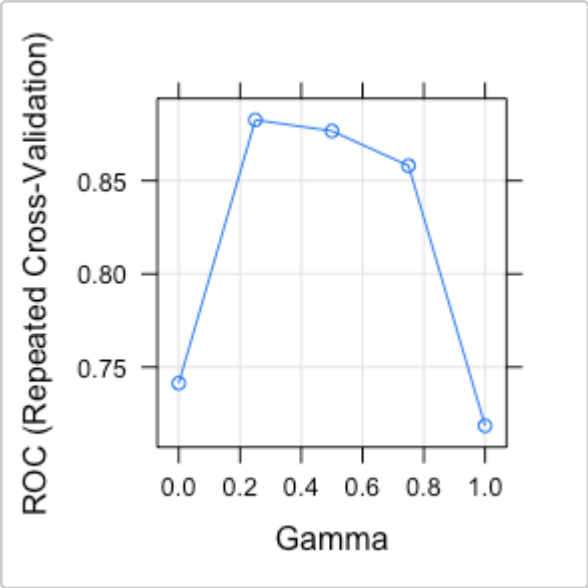

```
##            M         R
## 2  0.4398947 0.5601053
## 3  0.7254289 0.2745711
## 13 0.4543292 0.5456708
## 15 0.4919948 0.5080052
## 21 0.5210875 0.4789125
## 38 0.2936443 0.7063557
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  M  R
##          M 26  5
##          R  7 24
##
##                Accuracy : 0.8065
##                  95% CI : (0.6863, 0.8958)
##     No Information Rate : 0.5323
##     P-Value [Acc > NIR] : 6.468e-06
##
##                   Kappa : 0.6129
##  Mcnemar's Test P-Value : 0.7728
```

```
##
##             Sensitivity : 0.7879
##             Specificity : 0.8276
##          Pos Pred Value : 0.8387
##          Neg Pred Value : 0.7742
##              Prevalence : 0.5323
##          Detection Rate : 0.4194
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.8077
##
##        'Positive' Class : M
##
```

```
## Regularized Discriminant Analysis
##
## 146 samples
##  60 predictor
##   2 classes: 'M', 'R'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 131, 131, 131, 131, 131, 132, ...
## Resampling results across tuning parameters:
##
##   gamma  ROC        Sens       Spec
##   0.00   0.7413832  0.7755952  0.6142857
##   0.25   0.8825964  0.8791667  0.6976190
##   0.50   0.8766582  0.8875000  0.7134921
##   0.75   0.8580499  0.8660714  0.6896825
##   1.00   0.7185658  0.6470238  0.6055556
##
## Tuning parameter 'lambda' was held constant at a value of 0.75
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were gamma = 0.25 and lambda = 0.75.
```

```
##  Factor w/ 2 levels "M","R": 1 1 1 2 1 2 2 2 2 1 ...
```



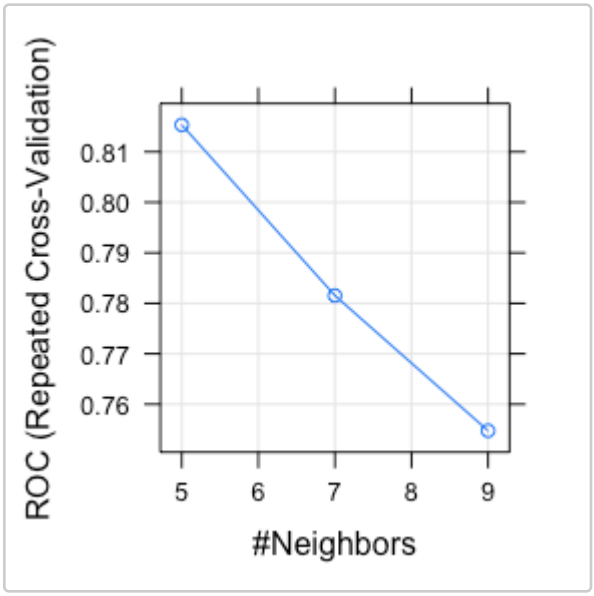

```
##            M         R
## 1 0.50670934 0.49329066
## 2 0.98870923 0.01129077
## 3 0.61748651 0.38251349
## 4 0.12600958 0.87399042
## 5 0.57768951 0.42231049
## 6 0.01037806 0.98962194
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  M  R
##          M 30  7
##          R  3 22
##
##                Accuracy : 0.8387
##                  95% CI : (0.7233, 0.9198)
##     No Information Rate : 0.5323
##     P-Value [Acc > NIR] : 3.903e-07
##
##                   Kappa : 0.6733
##  Mcnemar's Test P-Value : 0.3428
##
##             Sensitivity : 0.9091
##             Specificity : 0.7586
##          Pos Pred Value : 0.8108
##          Neg Pred Value : 0.8800
##              Prevalence : 0.5323
##          Detection Rate : 0.4839
##    Detection Prevalence : 0.5968
##       Balanced Accuracy : 0.8339
```

```
##
##        'Positive' Class : M
##
```

```
## k-Nearest Neighbors
##
## 146 samples
##  60 predictor
##   2 classes: 'M', 'R'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 131, 132, 131, 131, 131, 132, ...
## Resampling results across tuning parameters:
##
##   k  ROC        Sens       Spec
##   5  0.8153274  0.8940476  0.5841270
##   7  0.7815334  0.8130952  0.5238095
##   9  0.7547548  0.7821429  0.5230159
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

```
##  Factor w/ 2 levels "M","R": 1 2 1 2 1 2 2 2 1 1 ...
```




```
##     M   R
## 1 0.8 0.2
## 2 0.4 0.6
## 3 0.8 0.2
## 4 0.2 0.8
## 5 0.8 0.2
```

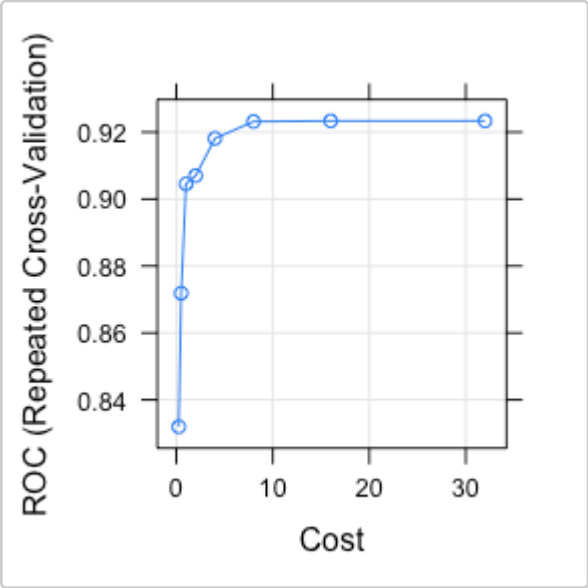
```
## 6 0.0 1.0
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  M  R
##          M 27 13
##          R  6 16
##
##                Accuracy : 0.6935
##                  95% CI : (0.5635, 0.8044)
##     No Information Rate : 0.5323
##     P-Value [Acc > NIR] : 0.007171
##
##                   Kappa : 0.3754
##  Mcnemar's Test P-Value : 0.168669
##
##             Sensitivity : 0.8182
##             Specificity : 0.5517
##          Pos Pred Value : 0.6750
##          Neg Pred Value : 0.7273
##              Prevalence : 0.5323
##          Detection Rate : 0.4355
##    Detection Prevalence : 0.6452
##       Balanced Accuracy : 0.6850
##
##        'Positive' Class : M
##
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 146 samples
##  60 predictor
##   2 classes: 'M', 'R'
##
## Pre-processing: centered (60), scaled (60)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 132, 131, 131, 131, 131, 131, ...
## Resampling results across tuning parameters:
##
##   C      ROC        Sens       Spec
##    0.25  0.8319586  0.7136905  0.7706349
##    0.50  0.8718679  0.8238095  0.7420635
##    1.00  0.9045351  0.8630952  0.7706349
##    2.00  0.9070011  0.8892857  0.7809524
##    4.00  0.9181264  0.8726190  0.7658730
##    8.00  0.9231718  0.8726190  0.8055556
```

```
##   16.00  0.9233560  0.8684524  0.7809524
##   32.00  0.9233560  0.8773810  0.7753968
##
## Tuning parameter 'sigma' was held constant at a value of 0.01186619
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.01186619 and C = 16.
```

```
##  Factor w/ 2 levels "M","R": 2 1 2 2 1 2 2 2 2 1 ...
```




```
##            M         R
## 1 0.38762751 0.6123725
## 2 0.70661864 0.2933814
## 3 0.42303865 0.5769613
## 4 0.13292557 0.8670744
## 5 0.55207248 0.4479275
## 6 0.06471954 0.9352805
```

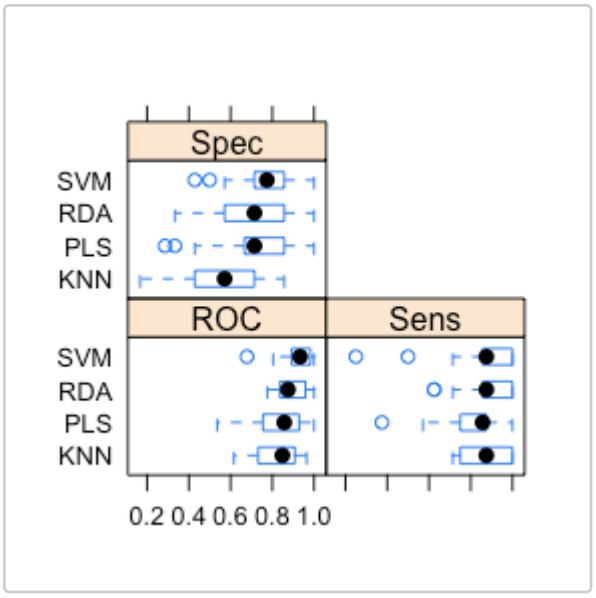
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  M  R
##          M 29  3
##          R  4 26
##
##                Accuracy : 0.8871
##                  95% CI : (0.7811, 0.9534)
##     No Information Rate : 0.5323
##     P-Value [Acc > NIR] : 2.418e-09
##
##                   Kappa : 0.7737
##  Mcnemar's Test P-Value : 1
```

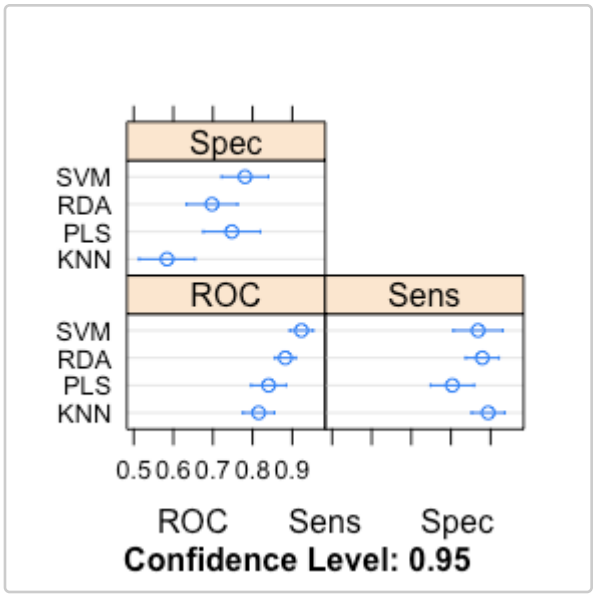
```
##
##             Sensitivity : 0.8788
##             Specificity : 0.8966
##          Pos Pred Value : 0.9062
##          Neg Pred Value : 0.8667
##              Prevalence : 0.5323
##          Detection Rate : 0.4677
##    Detection Prevalence : 0.5161
##       Balanced Accuracy : 0.8877
##
##        'Positive' Class : M
##
```

```
##
## Call:
## resamples.default(x = list(PLS = plsFit, RDA = rdaFit, KNN = knnFit, SVM
##  = svmFit))
##
## Models: PLS, RDA, KNN, SVM
## Number of resamples: 30
## Performance metrics: ROC, Sens, Spec
## Time estimates for: everything, final model fit
```

```
##
## Call:
## summary.resamples(object = resampall)
##
## Models: PLS, RDA, KNN, SVM
## Number of resamples: 30
##
## ROC
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## PLS 0.5357143 0.7590349 0.8571429 0.8405896 0.9285714 1.0000000    0
## RDA 0.7755102 0.8373724 0.8750000 0.8825964 0.9553571 1.0000000    0
## KNN 0.6160714 0.7388393 0.8482143 0.8153274 0.9043367 0.9642857    0
## SVM 0.6785714 0.8973214 0.9330357 0.9233560 0.9821429 1.0000000    0
##
## Sens
##          Min.   1st Qu.    Median      Mean 3rd Qu. Max. NA's
## PLS 0.3750000 0.7500000 0.8571429 0.8035714   0.875    1    0
## RDA 0.6250000 0.8571429 0.8750000 0.8791667   1.000    1    0
## KNN 0.7142857 0.7500000 0.8750000 0.8940476   1.000    1    0
## SVM 0.2500000 0.8571429 0.8750000 0.8684524   1.000    1    0
##
## Spec
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## PLS 0.2857143 0.6666667 0.7142857 0.7476190 0.8571429 1.0000000    0
```

```
## RDA 0.3333333 0.5714286 0.7142857 0.6976190 0.8571429 1.0000000    0
## KNN 0.1666667 0.4285714 0.5714286 0.5841270 0.7142857 0.8571429    0
## SVM 0.4285714 0.7142857 0.7738095 0.7809524 0.8571429 1.0000000    0
```

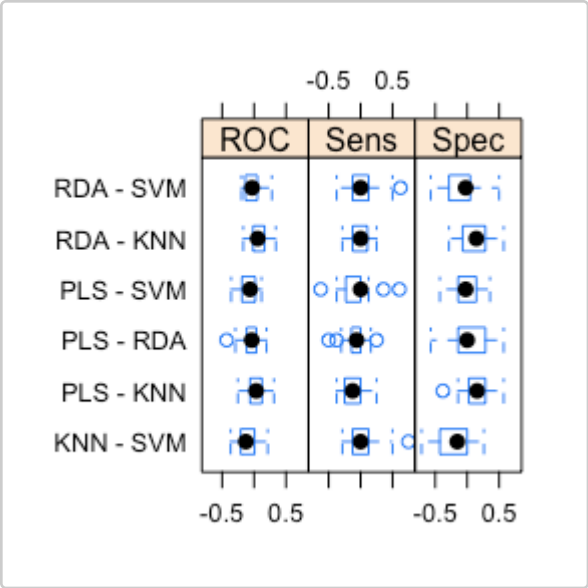


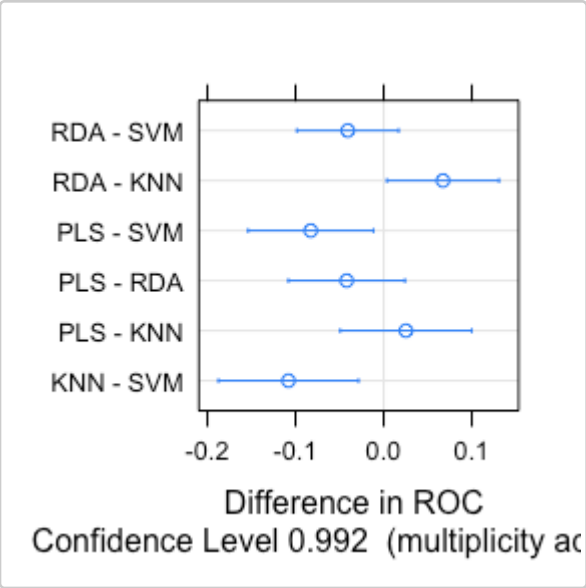



```
##
## Call:
## diff.resamples(x = resampall)
##
## Models: PLS, RDA, KNN, SVM
## Metrics: ROC, Sens, Spec
## Number of differences: 6
## p-value adjustment: bonferroni
```

```
##
## Call:
## summary.diff.resamples(object = diffVals)
##
## p-value adjustment: bonferroni
## Upper diagonal: estimates of the difference
## Lower diagonal: p-value for H0: difference = 0
##
## ROC
##     PLS      RDA      KNN      SVM
## PLS          -0.04201  0.02526 -0.08277
## RDA 0.494604           0.06727 -0.04076
## KNN 1.000000 0.033288          -0.10803
## SVM 0.015652 0.329352 0.003595
##
## Sens
##     PLS     RDA      KNN      SVM
## PLS         -0.07560 -0.09048 -0.06488
```

```
## RDA 0.10676          -0.01488  0.01071
## KNN 0.07705 1.00000            0.02560
## SVM 0.82821 1.00000  1.00000
##
## Spec
##     PLS       RDA       KNN       SVM
## PLS            0.05000   0.16349  -0.03333
## RDA 1.0000000             0.11349  -0.08333
## KNN 0.0025121 0.0615114           -0.19683
## SVM 1.0000000 0.5509128 0.0007307
```





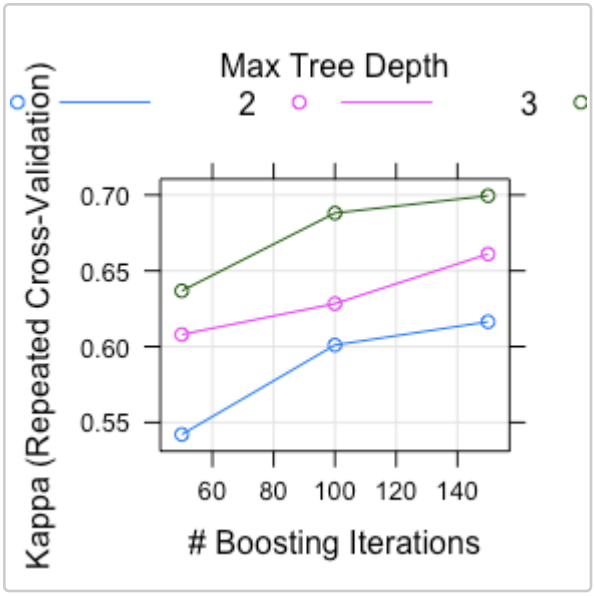
The results of these four models show that the support vector machine results in the best accuracy and kappa values after running the test set. k-nearest neighbor had the poorest performance.

#Create a set of gradient boosted models with various parameters

```
## Stochastic Gradient Boosting
##
## 146 samples
##  60 predictor
##   2 classes: 'M', 'R'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 131, 131, 133, 131, 131, 131, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa
##   1                   50      0.7748791  0.5420898
##   1                  100      0.8035678  0.6010533
##   1                  150      0.8109011  0.6163990
##   2                   50      0.8067253  0.6079958
##   2                  100      0.8164872  0.6283452
```

```
##   2                  150      0.8323443  0.6610801
##   3                   50      0.8206777  0.6368086
##   3                  100      0.8462564  0.6880813
##   3                  150      0.8516923  0.6995081
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Kappa was used to select the optimal model using the largest value.
## The final values used for the model were n.trees = 150,
##  interaction.depth = 3, shrinkage = 0.1 and n.minobsinnode = 10.
```

```
##  Factor w/ 2 levels "M","R": 1 1 2 2 1 2 2 2 2 2 ...
```




```
##             M         R
## 1 0.575209804 0.4247902
## 2 0.614288582 0.3857114
## 3 0.347564677 0.6524353
## 4 0.209458982 0.7905410
## 5 0.845470390 0.1545296
## 6 0.001098182 0.9989018
```

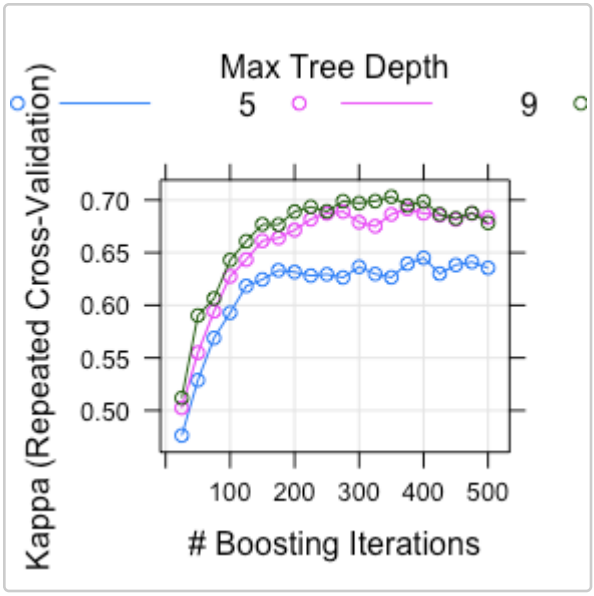
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  M  R
##          M 31  5
##          R  2 24
##
##                Accuracy : 0.8871
##                  95% CI : (0.7811, 0.9534)
```

```
##     No Information Rate : 0.5323
##     P-Value [Acc > NIR] : 2.418e-09
##
##                   Kappa : 0.7718
##  Mcnemar's Test P-Value : 0.4497
##
##             Sensitivity : 0.9394
##             Specificity : 0.8276
##          Pos Pred Value : 0.8611
##          Neg Pred Value : 0.9231
##              Prevalence : 0.5323
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5806
##       Balanced Accuracy : 0.8835
##
##        'Positive' Class : M
##
```

```
## Stochastic Gradient Boosting
##
## 146 samples
##  60 predictor
##   2 classes: 'M', 'R'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 132, 131, 132, 131, 131, 132, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa
##   1                   25      0.7420696  0.4762769
##   1                   50      0.7673700  0.5288172
##   1                   75      0.7871941  0.5688793
##   1                  100      0.7989487  0.5928594
##   1                  125      0.8112821  0.6182919
##   1                  150      0.8145678  0.6244830
##   1                  175      0.8188205  0.6331991
##   1                  200      0.8180037  0.6315565
##   1                  225      0.8162088  0.6281495
##   1                  250      0.8168755  0.6293846
##   1                  275      0.8152015  0.6263611
##   1                  300      0.8202564  0.6364524
##   1                  325      0.8167802  0.6296847
##   1                  350      0.8152015  0.6265295
##   1                  375      0.8216850  0.6394619
##   1                  400      0.8243516  0.6450199
##   1                  425      0.8168755  0.6300219
##   1                  450      0.8208755  0.6382732
```

```
##   1                  475      0.8223040  0.6410559
##   1                  500      0.8195897  0.6354781
##   5                   25      0.7544176  0.5024759
##   5                   50      0.7799414  0.5548912
##   5                   75      0.7993626  0.5942509
##   5                  100      0.8159963  0.6274506
##   5                  125      0.8237656  0.6433487
##   5                  150      0.8325275  0.6612628
##   5                  175      0.8339487  0.6638989
##   5                  200      0.8373370  0.6714253
##   5                  225      0.8426703  0.6816750
##   5                  250      0.8455348  0.6872935
##   5                  275      0.8462015  0.6889717
##   5                  300      0.8413993  0.6788824
##   5                  325      0.8393443  0.6751036
##   5                  350      0.8449634  0.6863050
##   5                  375      0.8475824  0.6914837
##   5                  400      0.8455348  0.6873903
##   5                  425      0.8447729  0.6855531
##   5                  450      0.8428278  0.6815579
##   5                  475      0.8455897  0.6872580
##   5                  500      0.8436850  0.6834715
##   9                   25      0.7593297  0.5119011
##   9                   50      0.7983773  0.5902493
##   9                   75      0.8060037  0.6064336
##   9                  100      0.8236777  0.6429928
##   9                  125      0.8324872  0.6608050
##   9                  150      0.8402088  0.6768933
##   9                  175      0.8401136  0.6765445
##   9                  200      0.8461136  0.6889427
##   9                  225      0.8483516  0.6934479
##   9                  250      0.8463040  0.6892145
##   9                  275      0.8511136  0.6988321
##   9                  300      0.8503516  0.6970564
##   9                  325      0.8511612  0.6989358
##   9                  350      0.8533040  0.7030782
##   9                  375      0.8490659  0.6947208
##   9                  400      0.8511136  0.6986002
##   9                  425      0.8450183  0.6865250
##   9                  450      0.8429231  0.6825311
##   9                  475      0.8455897  0.6875599
##   9                  500      0.8409231  0.6781975
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 20
## Kappa was used to select the optimal model using the largest value.
## The final values used for the model were n.trees = 350,
##  interaction.depth = 9, shrinkage = 0.1 and n.minobsinnode = 20.
```

```
##  Factor w/ 2 levels "M","R": 1 1 2 2 1 2 2 2 2 2 ...
```




```
##              M         R
## 1 0.7243295263 0.27567047
## 2 0.9722652381 0.02773476
## 3 0.0233780339 0.97662197
## 4 0.0125598349 0.98744017
## 5 0.9312177190 0.06878228
## 6 0.0008655557 0.99913444
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  M  R
##          M 30  4
##          R  3 25
##
##                Accuracy : 0.8871
##                  95% CI : (0.7811, 0.9534)
##     No Information Rate : 0.5323
##     P-Value [Acc > NIR] : 2.418e-09
##
##                   Kappa : 0.7728
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9091
##             Specificity : 0.8621
##          Pos Pred Value : 0.8824
##          Neg Pred Value : 0.8929
##              Prevalence : 0.5323
##          Detection Rate : 0.4839
```

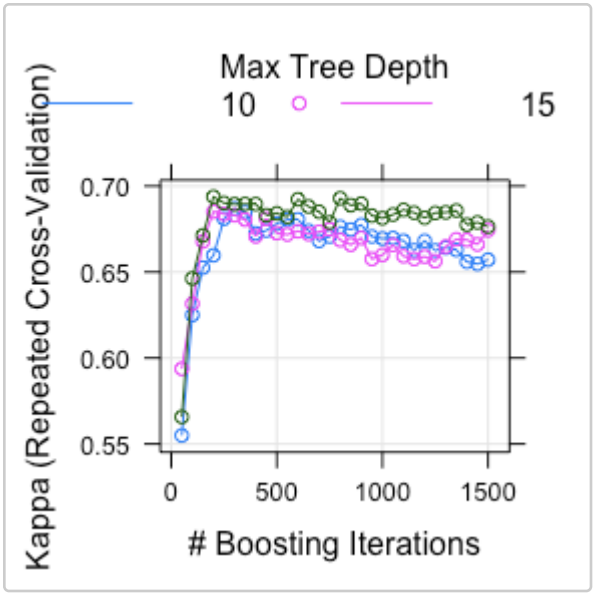
```
##    Detection Prevalence : 0.5484
##       Balanced Accuracy : 0.8856
##
##        'Positive' Class : M
##
```

```
## Stochastic Gradient Boosting
##
## 146 samples
##  60 predictor
##   2 classes: 'M', 'R'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 132, 131, 131, 131, 131, 131, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa
##    5                   50     0.7802161  0.5549457
##    5                  100     0.8140806  0.6249849
##    5                  150     0.8280183  0.6525231
##    5                  200     0.8314615  0.6597235
##    5                  250     0.8422637  0.6810088
##    5                  300     0.8451758  0.6868993
##    5                  350     0.8439927  0.6848116
##    5                  400     0.8378974  0.6724609
##    5                  450     0.8389121  0.6741270
##    5                  500     0.8409670  0.6781820
##    5                  550     0.8419927  0.6805240
##    5                  600     0.8420952  0.6806476
##    5                  650     0.8386117  0.6740022
##    5                  700     0.8358022  0.6680589
##    5                  750     0.8372308  0.6706635
##    5                  800     0.8398974  0.6763508
##    5                  850     0.8392308  0.6746958
##    5                  900     0.8405092  0.6773772
##    5                  950     0.8370733  0.6706652
##    5                 1000     0.8364689  0.6694752
##    5                 1050     0.8365165  0.6697961
##    5                 1100     0.8358498  0.6681261
##    5                 1150     0.8331355  0.6626657
##    5                 1200     0.8358974  0.6681201
##    5                 1250     0.8330806  0.6625897
##    5                 1300     0.8338425  0.6641354
##    5                 1350     0.8331355  0.6628509
##    5                 1400     0.8297473  0.6559335
##    5                 1450     0.8291355  0.6547095
##    5                 1500     0.8304689  0.6571511
```

```
##   10                  50     0.7988571  0.5936946
##   10                 100     0.8176044  0.6315717
##   10                 150     0.8352161  0.6677956
##   10                 200     0.8439377  0.6851415
##   10                 250     0.8434542  0.6837119
##   10                 300     0.8433663  0.6830303
##   10                 350     0.8419377  0.6804832
##   10                 400     0.8371758  0.6707069
##   10                 450     0.8420330  0.6809337
##   10                 500     0.8378352  0.6726507
##   10                 550     0.8377949  0.6718438
##   10                 600     0.8386593  0.6737140
##   10                 650     0.8379377  0.6720537
##   10                 700     0.8386044  0.6734244
##   10                 750     0.8392711  0.6749124
##   10                 800     0.8359927  0.6686068
##   10                 850     0.8347070  0.6658813
##   10                 900     0.8367070  0.6698107
##   10                 950     0.8304615  0.6575623
##   10                1000     0.8317949  0.6599303
##   10                1050     0.8347546  0.6657778
##   10                1100     0.8313736  0.6592262
##   10                1150     0.8306044  0.6575811
##   10                1200     0.8313663  0.6589590
##   10                1250     0.8298901  0.6563082
##   10                1300     0.8341355  0.6648332
##   10                1350     0.8361832  0.6690304
##   10                1400     0.8361355  0.6687052
##   10                1450     0.8347546  0.6660417
##   10                1500     0.8388498  0.6743009
##   15                  50     0.7853993  0.5657030
##   15                 100     0.8248828  0.6462755
##   15                 150     0.8373407  0.6713168
##   15                 200     0.8483883  0.6937748
##   15                 250     0.8465458  0.6901588
##   15                 300     0.8463956  0.6894896
##   15                 350     0.8466410  0.6895832
##   15                 400     0.8465861  0.6895306
##   15                 450     0.8431502  0.6830201
##   15                 500     0.8437692  0.6841484
##   15                 550     0.8431099  0.6820891
##   15                 600     0.8479670  0.6922826
##   15                 650     0.8459194  0.6881234
##   15                 700     0.8444908  0.6850629
##   15                 750     0.8414286  0.6788907
##   15                 800     0.8483736  0.6928941
##   15                 850     0.8462857  0.6888573
##   15                 900     0.8469048  0.6897773
##   15                 950     0.8435165  0.6828164
```

```
##   15                 1000     0.8426593  0.6813360
##   15                 1050     0.8436740  0.6834072
##   15                 1100     0.8450073  0.6861596
##   15                 1150     0.8442857  0.6844866
##   15                 1200     0.8428095  0.6815369
##   15                 1250     0.8442454  0.6843716
##   15                 1300     0.8442930  0.6846793
##   15                 1350     0.8449597  0.6857384
##   15                 1400     0.8407070  0.6775272
##   15                 1450     0.8414212  0.6788332
##   15                 1500     0.8400879  0.6764314
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 20
## Kappa was used to select the optimal model using the largest value.
## The final values used for the model were n.trees = 200,
##  interaction.depth = 15, shrinkage = 0.1 and n.minobsinnode = 20.
```

```
##  Factor w/ 2 levels "M","R": 1 1 2 2 1 2 2 2 2 2 ...
```




```
##           M         R
## 1 0.64744094 0.35255906
## 2 0.89930855 0.10069145
## 3 0.03316382 0.96683618
## 4 0.12167888 0.87832112
## 5 0.94926539 0.05073461
## 6 0.00140377 0.99859623
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction  M  R
##          M 31  5
##          R  2 24
##
##                Accuracy : 0.8871
##                  95% CI : (0.7811, 0.9534)
##     No Information Rate : 0.5323
##     P-Value [Acc > NIR] : 2.418e-09
##
##                   Kappa : 0.7718
##  Mcnemar's Test P-Value : 0.4497
##
##             Sensitivity : 0.9394
##             Specificity : 0.8276
##          Pos Pred Value : 0.8611
##          Neg Pred Value : 0.9231
##              Prevalence : 0.5323
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5806
##       Balanced Accuracy : 0.8835
##
##        'Positive' Class : M
##
```

The gradient boosted models lost performance with more interaction depth and tree size. The most simple model produced results closest to the support vector machine output. Both models misclassify 7 items out of the total test set of 62 items. The svm model is “better” at finding rocks while the gbm model predicts mine/metal more accurately.