Data Matrix Modeling

Version: BINARY CLASSIFIER

Updated 12/18/2017 by E. Gates. See readme on github

WIP: semi-supervised feature and genetic variable selection

WIP: custom model selection

WIP: Compare to null model for imbalanced datasets

BUG: preprocess crashes when trying too many image features

BUG: statement "No univariate significant values" not printing.

Compiled: 2017-Dec-19 09:47:04

Target Variable: MutationalStatus

Input File: DF_mutation_Dec182017/pyradiomicsout.csv

Target and inputs are column headings in csv file, everything else is ignored

Pre-processing data: By default removes columns with zero variance and discards variables correlated >0.8

Pre-Processing results:

Started with 112 non-NA variables.

```
## Created from 32 samples and 110 variables
##
## Pre-processing:
##   - centered (36)
##   - ignored (0)
##   - removed (74)
##   - scaled (36)
```

36 remained after pre-processing

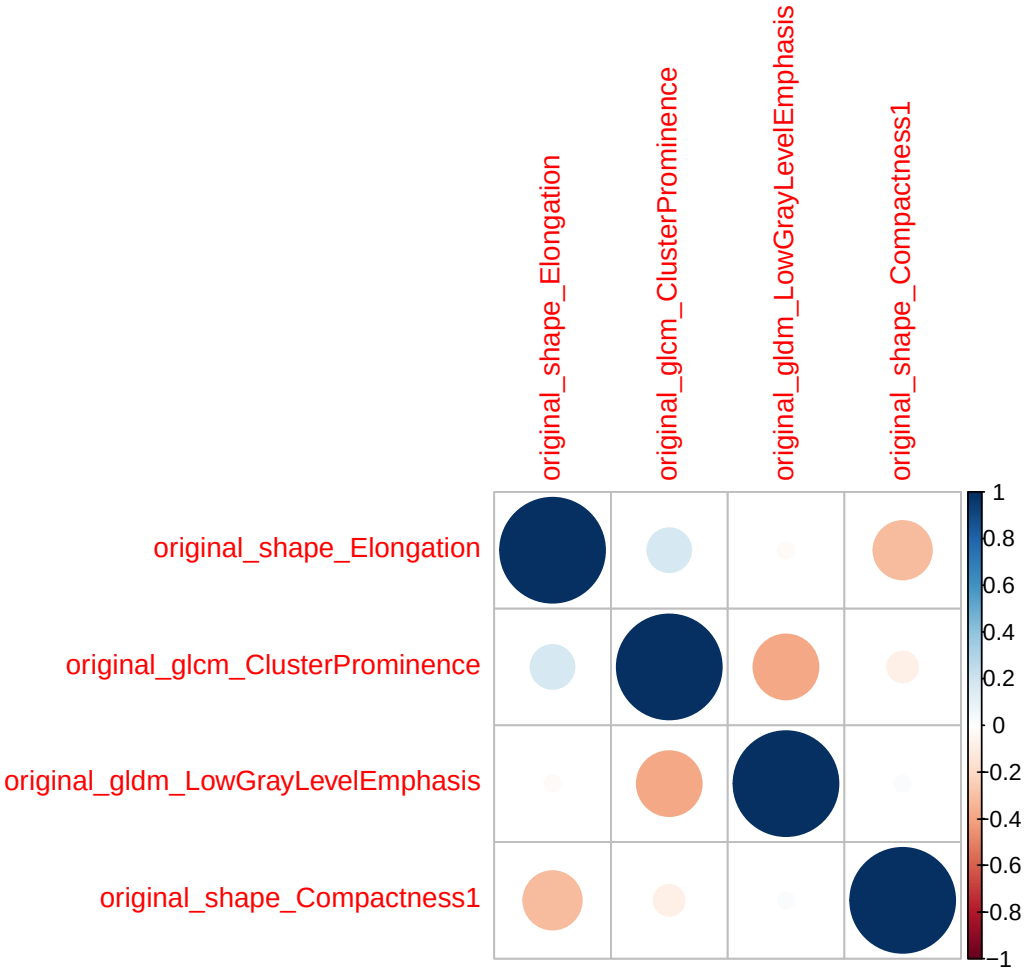
Variable Selection:

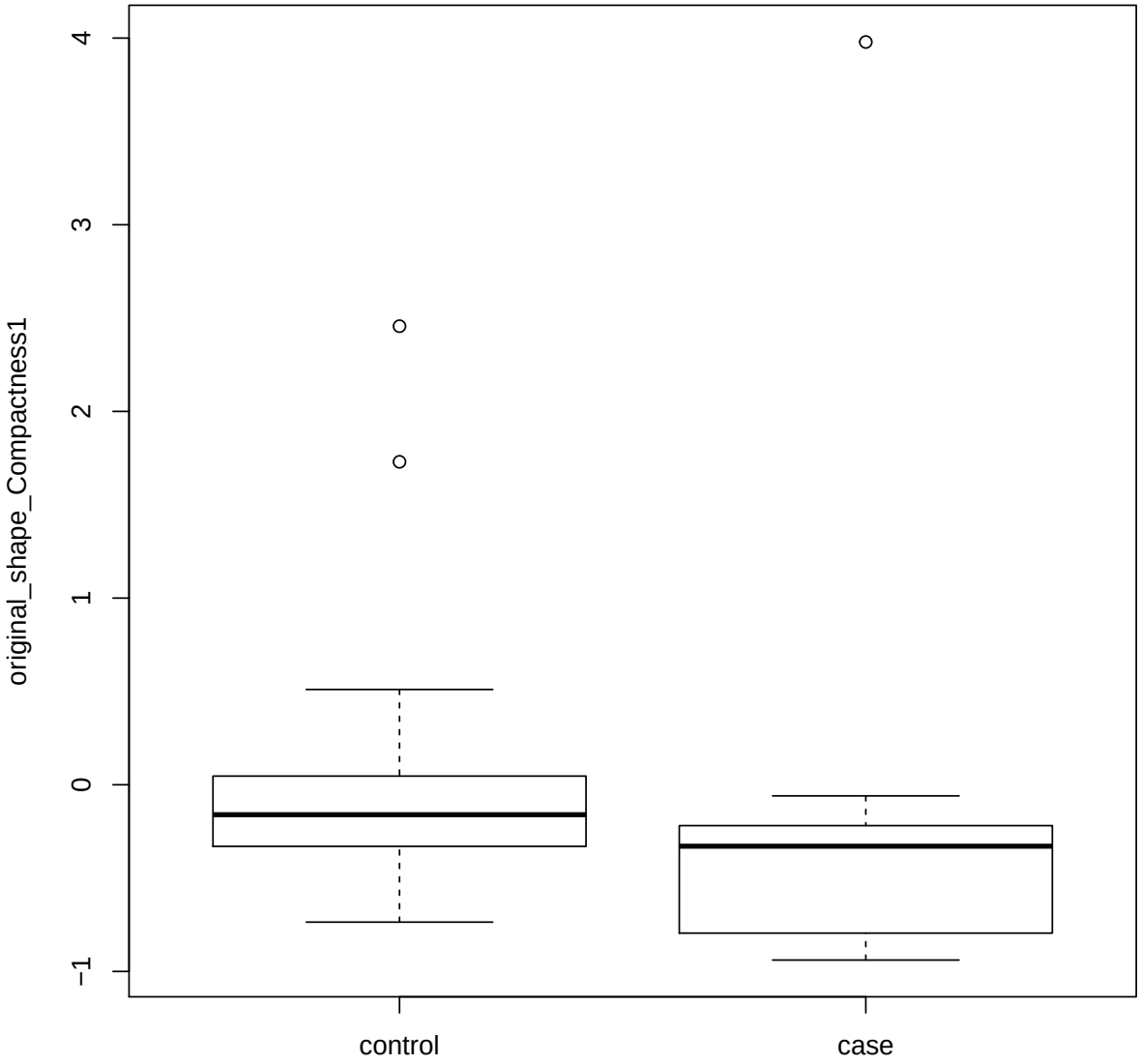
Default is Boruta and Wilcoxon test (P value cutoff 0.20/ 36). Wilcoxon currently only tests numeric input variables

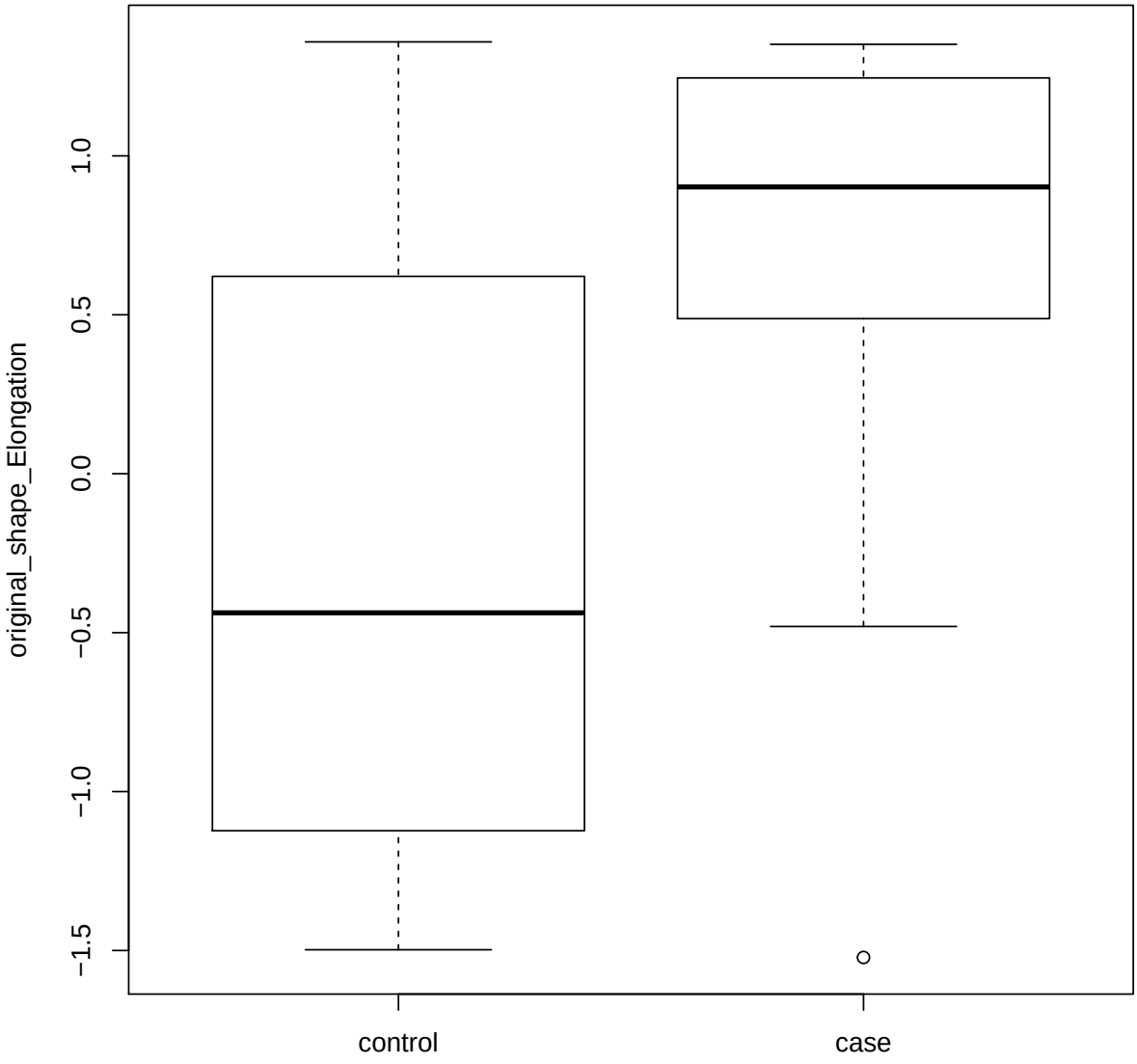
```
## [1] "Finished Boruta variable selection"
## Boruta performed 499 iterations in 5.9462 secs.
##  4 attributes confirmed important:
## original_glcm_ClusterProminence,
## original_gldm_LowGrayLevelEmphasis, original_shape_Compactness1,
## original_shape_Elongation;
##  28 attributes confirmed unimportant:
## original_firstorder_10Percentile,
## original_firstorder_InterquartileRange,
## original_firstorder_Kurtosis,
```

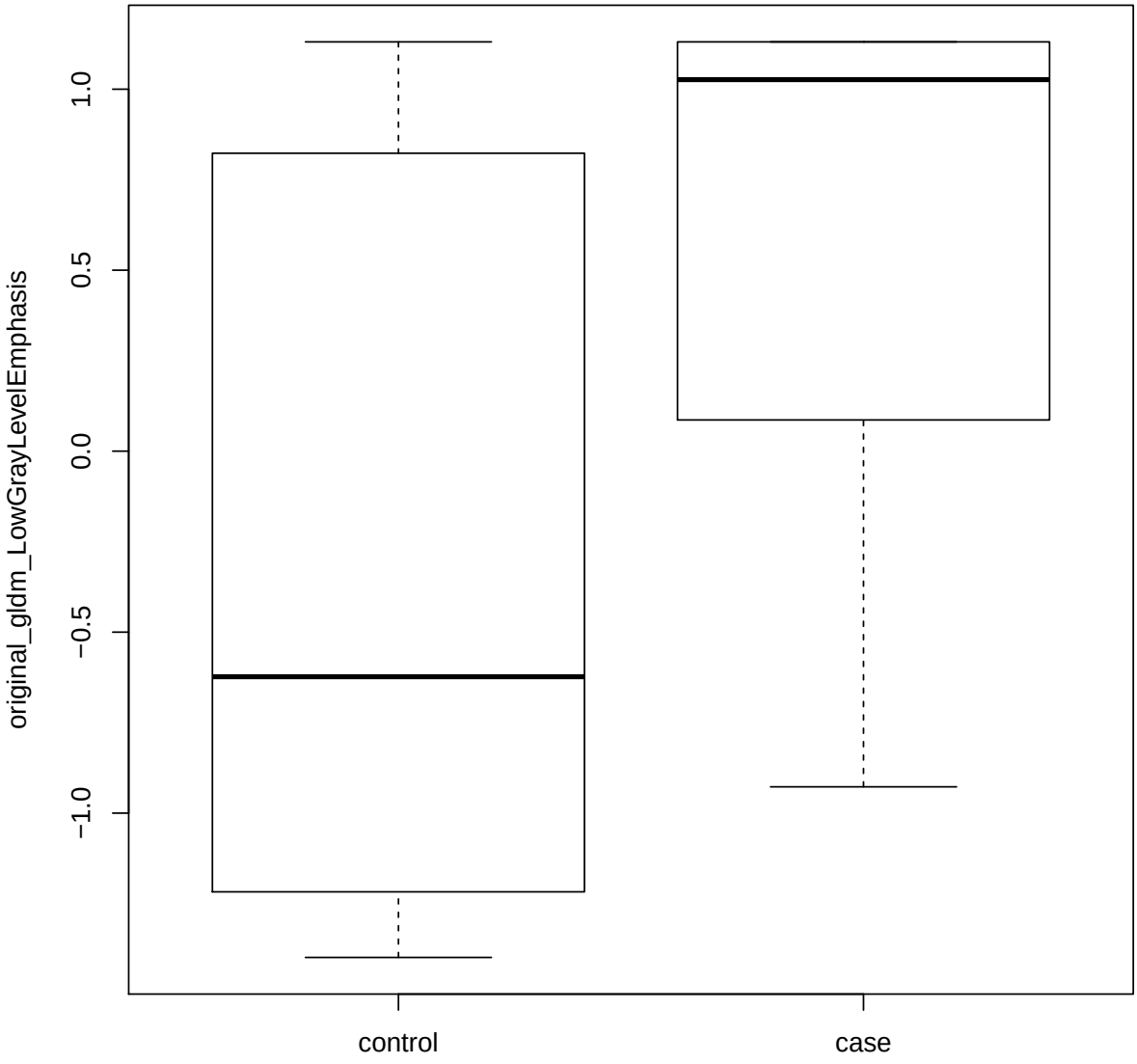
```
## original_firstorder_RobustMeanAbsoluteDeviation,
## original_firstorder_Uniformity and 23 more;
##  4 tentative attributes left:
## original_firstorder_MeanAbsoluteDeviation,
## original_firstorder_Skewness,
## original_shape_Maximum2DDiameterColumn, original_shape_MinorAxis;
```

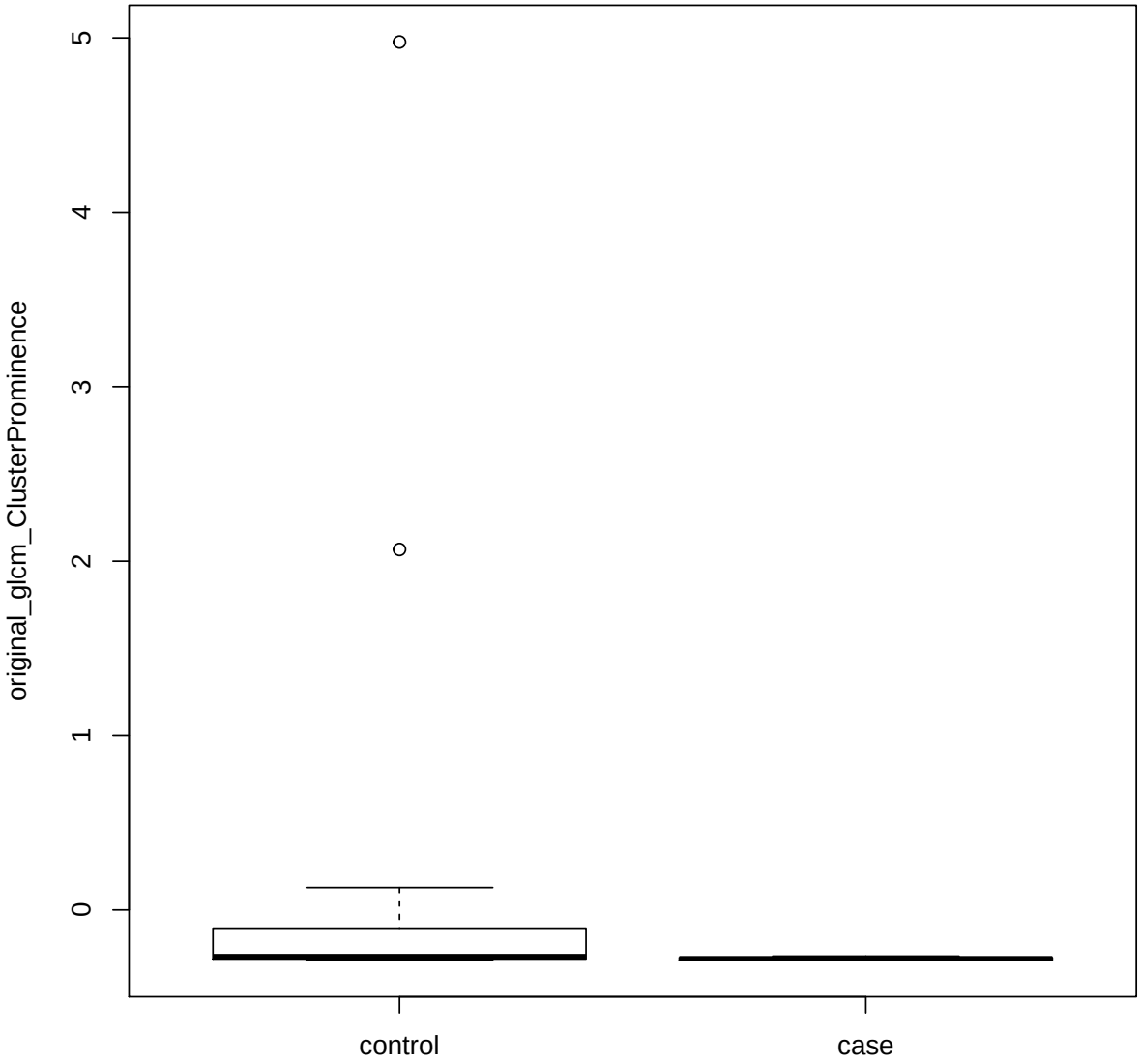
Correlations for Boruta method




original_shape_Compactness1
4
3
2
1
0
−1
control
case


original_shape_Elongation
1.0
0.5
0.0
−0.5
−1.0
−1.5
control
case


original_gldm_LowGrayLevelEmphasis
1.0
0.5
0.0
−0.5
−1.0
control
case



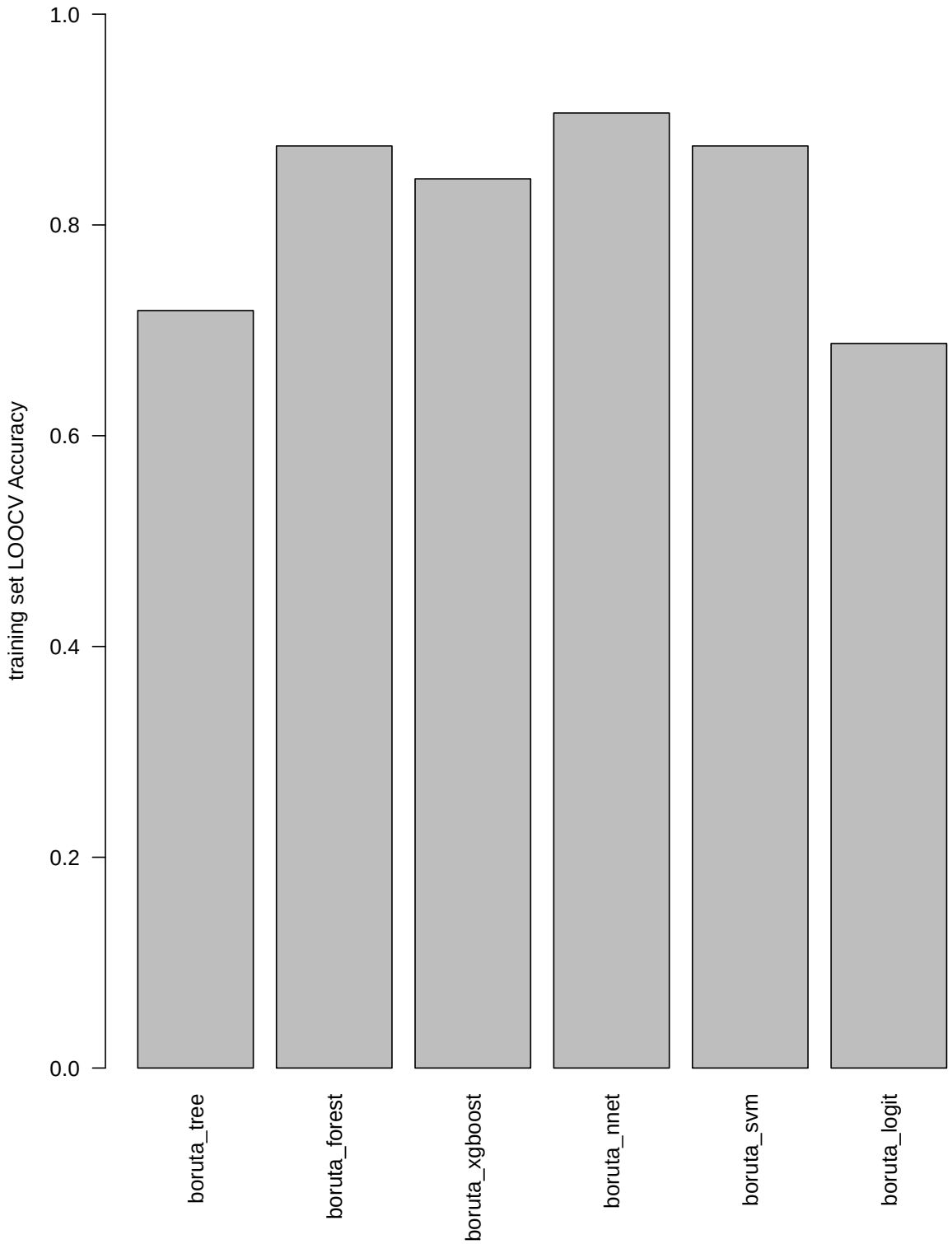

```
## Warning in wilcox.test.default(x = c(-0.393660316090896,
## 1.78677583435781, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(0.744456817818816,
## -0.912363001344349, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(0.00697820632996377,
## -0.237231217538845, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(1.3586085329901, -0.562539082437358, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.694331318549416,
```

```
## -0.881557807404732, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.777035765922205,
## -0.644256884115954, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(1.01202541063194,
## -0.490657655778258, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.912058145411943,
## -1.28814252485697, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-1.13145998882554,
## -1.35885526906859, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(0.749557528981004,
## 0.610859469734699, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.240436974046883,
## 0.128216354385191, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.249360513798991,
## 0.357165311356997, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-1.2227353462982,
## -0.677434592599839, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.560613683855736,
## -0.836608112830868, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-1.42081560623198, -1.9118727578487, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(0.656295769363097,
## 0.479926564341781, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.355512368681753,
## 0.0400684550436253, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.791744781792387,
## -0.839846904767932, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-1.56045170587466,
## -1.59842467514309, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(1.01347566751555, 2.37763123276732, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(0.963624124953188,
## 0.349808419566771, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(0.100125733844822, 2.12189793287319, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.626754934348351,
## -0.435092324395663, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.872065831780065,
## 1.04830341581084, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.00742345373215554,
## 2.37961198869889, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-1.10225296659161,
## 0.187204278687348, : cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = c(-0.534289079919695,
## -1.05950610449881, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.560224013396778,
## -0.262613703101298, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.549581639408882,
## 0.128940260332245, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.1980725732724, 0.828532126936452, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-1.17965997337117,
## -1.68006598369419, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.397520837014066,
## -0.744346408121432, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.71234554654753,
## -0.712349151376613, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.443428773452489,
## -0.265067732188947, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(0.0443210639965818,
## -0.0281116762050019, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(-0.389005866492521,
## -0.38265704203561, : cannot compute exact p-value with ties
```

Modeling using tree, forest, xgboost, nnet, svm, logit.

Use Leave-one-out cross validation: TRUE



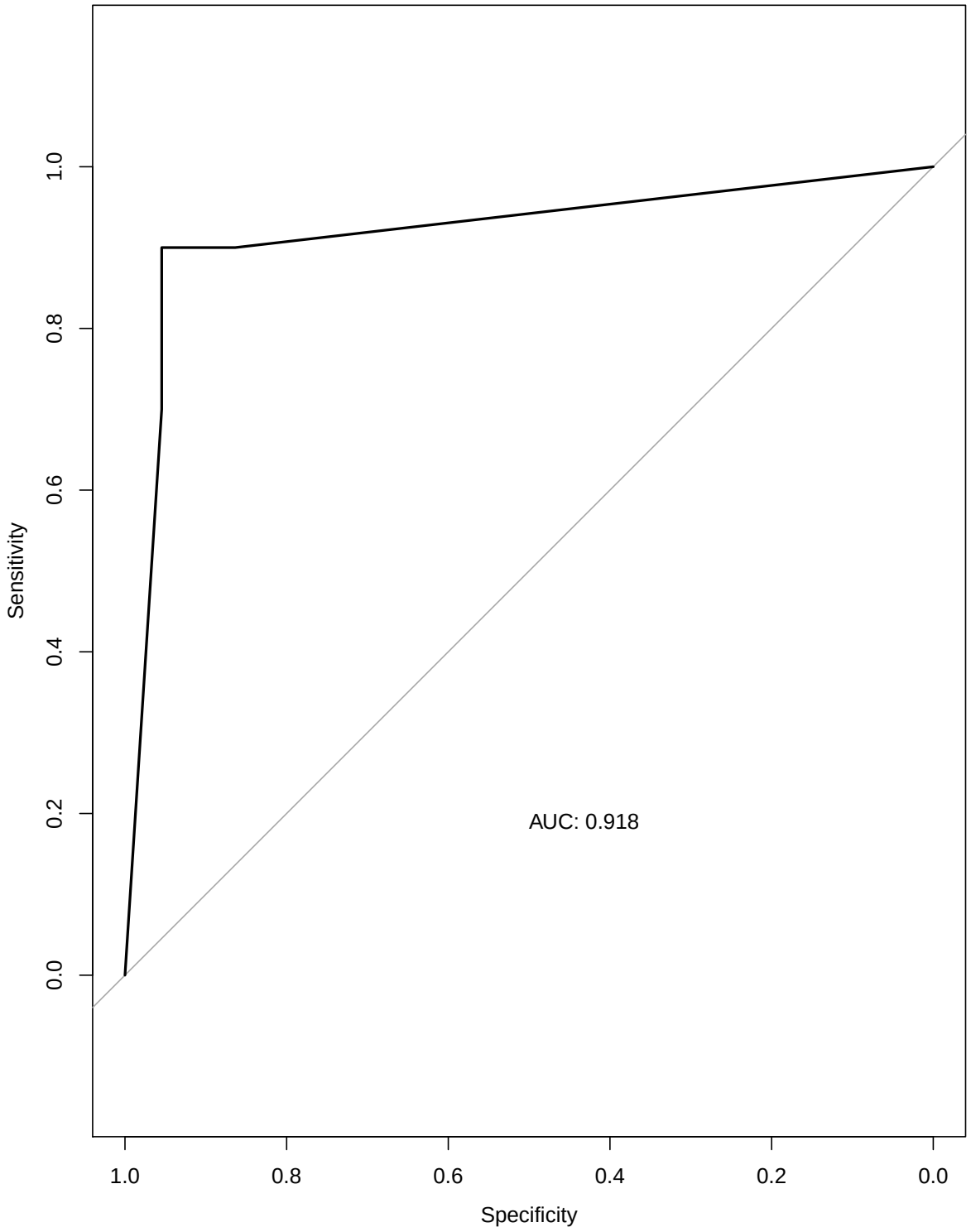
Best model(s): boruta_nnet

Accuracy: 0.9062

```
## [[1]]
```

```
## Neural Network
##
## 32 samples
##  4 predictor
##  2 classes: 'control', 'case'
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 31, 31, 31, 31, 31, 31, ...
## Resampling results across tuning parameters:
##
##   size  decay  Accuracy  Kappa
##   1     0e+00  0.87500   0.7090909
##   1     1e-04  0.87500   0.7090909
##   1     1e-01  0.84375   0.6261682
##   3     0e+00  0.84375   0.6638655
##   3     1e-04  0.87500   0.6923077
##   3     1e-01  0.84375   0.6261682
##   5     0e+00  0.90625   0.7876106
##   5     1e-04  0.87500   0.7090909
##   5     1e-01  0.84375   0.6261682
##
## Accuracy 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] "Building ROC Curve for model boruta_nnet"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction control case
##    control      20    1
##    case          2    9
##
##                Accuracy : 0.9062
##                  95% CI : (0.7498, 0.9802)
##     No Information Rate : 0.6875
##     P-Value [Acc > NIR] : 0.003623
##
##                   Kappa : 0.7876
##  Mcnemar's Test P-Value : 1.000000
##
##             Sensitivity : 0.9000
##             Specificity : 0.9091
##          Pos Pred Value : 0.8182
##          Neg Pred Value : 0.9524
##              Prevalence : 0.3125
##          Detection Rate : 0.2812
##    Detection Prevalence : 0.3438
##       Balanced Accuracy : 0.9045
##
##        'Positive' Class : case
##
##
## Call:
```

```
## roc.default(response = results$obs, predictor = results$case)
##
## Data: results$case in 22 controls (results$obs control) < 10 cases (results$obs case).
## Area under the curve: 0.9182
```



threshold	0.9999884
sensitivity	0.9000000
specificity	0.9545455