# Yelp dataset (1000 documents)

p1: taste

p2: service

p3: cost

testing :110 documents

**<<DocumentTermMatrix (documents: 1110, terms: 1878)>>**

**Non-/sparse entries: 6089/2078491**

**Sparsity: 100%**

**Maximal term length: 17**

**Weighting: term frequency (tf)**

**$SVM**

Call:

svm.default(x = container@training\_matrix, y = container@training\_codes, kernel = kernel,

cost = cost, cross = cross, probability = TRUE, method = method)

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 100

gamma: 0.0005324814

Number of Support Vectors: 829

**$TREE**

node), split, n, deviance, yval, (yprob)

\* denotes terminal node

1) root 1000 1386.000 2 ( 0.49900 0.50100 )

2) X712 < 0.5 936 1293.000 1 ( 0.53312 0.46688 )

4) X414 < 0.5 914 1259.000 1 ( 0.54595 0.45405 )

8) X698 < 0.5 828 1129.000 1 ( 0.57609 0.42391 )

16) X29 < 0.5 807 1093.000 1 ( 0.58984 0.41016 )

32) X588 < 0.5 795 1071.000 1 ( 0.59874 0.40126 )

64) X1090 < 0.5 772 1031.000 1 ( 0.61269 0.38731 )

128) X86 < 0.5 763 1013.000 1 ( 0.61992 0.38008 )

256) X656 < 0.5 744 979.200 1 ( 0.63172 0.36828 )

512) X556 < 0.5 736 963.000 1 ( 0.63859 0.36141 )

1024) X958 < 0.5 728 946.600 1 ( 0.64560 0.35440 )

2048) X1195 < 0.5 720 929.800 1 ( 0.65278 0.34722 )

4096) X957 < 0.5 704 899.400 1 ( 0.66335 0.33665 ) \*

4097) X957 > 0.5 16 15.440 2 ( 0.18750 0.81250 ) \*

2049) X1195 > 0.5 8 0.000 2 ( 0.00000 1.00000 ) \*

1025) X958 > 0.5 8 0.000 2 ( 0.00000 1.00000 ) \*

513) X556 > 0.5 8 0.000 2 ( 0.00000 1.00000 ) \*

257) X656 > 0.5 19 16.570 2 ( 0.15789 0.84211 ) \*

129) X86 > 0.5 9 0.000 2 ( 0.00000 1.00000 ) \*

65) X1090 > 0.5 23 17.810 2 ( 0.13043 0.86957 ) \*

33) X588 > 0.5 12 0.000 2 ( 0.00000 1.00000 ) \*

17) X29 > 0.5 21 8.041 2 ( 0.04762 0.95238 ) \*

9) X698 > 0.5 86 97.800 2 ( 0.25581 0.74419 ) \*

5) X414 > 0.5 22 0.000 2 ( 0.00000 1.00000 ) \*

3) X712 > 0.5 64 0.000 2 ( 0.00000 1.00000 ) \*

**RF**

Call:

randomForest(x = as.matrix(container@training\_matrix), y = container@training\_codes, ntree = ntree)

Type of random forest: classification

Number of trees: 200

No. of variables tried at each split: 43

OOB estimate of error rate: 24.1%

Confusion matrix:

1 2 class.error

1 416 83 0.1663327

2 158 343 0.3153693

**ACCURACY TABLE**

**Tree**

1 2

1 48 2

2 17 43

**SVM**

1 2

1 48 2

2 5 55

**MaxEnt**

1 2

1 50 0

2 0 60

**RF**

1 2

1 49 1

2 2 58

**Recal accuracty:**

Tree: 82.72727%

SVM: 93.63636%

Max: 100%

RF: 97.27273%

**ENSEMBLE SUMMARY**

n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL

n >= 1 1.00 0.95

n >= 2 1.00 0.95

n >= 3 0.97 0.97

n >= 4 0.82 1.00

**ALGORITHM PERFORMANCE**

SVM\_PRECISION SVM\_RECALL

0.935 0.940

SVM\_FSCORE FORESTS\_PRECISION

0.935 0.970

FORESTS\_RECALL FORESTS\_FSCORE

0.975 0.970

TREE\_PRECISION TREE\_RECALL

0.850 0.840

TREE\_FSCORE MAXENTROPY\_PRECISION

0.830 1.000

MAXENTROPY\_RECALL MAXENTROPY\_FSCORE

1. 1.000

**CROSS VALIDATION :**

**TREE:**

Fold 1 Out of Sample Accuracy = 0.6788991

Fold 2 Out of Sample Accuracy = 0.6956522

Fold 3 Out of Sample Accuracy = 0.7090909

Fold 4 Out of Sample Accuracy = 0.728

Fold 5 Out of Sample Accuracy = 0.773913

Fold 6 Out of Sample Accuracy = 0.7407407

Fold 7 Out of Sample Accuracy = 0.729927

Fold 8 Out of Sample Accuracy = 0.7173913

Fold 9 Out of Sample Accuracy = 0.7244898

Fold 10 Out of Sample Accuracy = 0.7096774

$meanAccuracy

[1] 0.7207781

**SVM:**

Fold 1 Out of Sample Accuracy = 0.8229167

Fold 2 Out of Sample Accuracy = 0.7833333

Fold 3 Out of Sample Accuracy = 0.8494624

Fold 4 Out of Sample Accuracy = 0.8018018

Fold 5 Out of Sample Accuracy = 0.7923077

Fold 6 Out of Sample Accuracy = 0.7636364

Fold 7 Out of Sample Accuracy = 0.7788462

Fold 8 Out of Sample Accuracy = 0.7868852

Fold 9 Out of Sample Accuracy = 0.733871

Fold 10 Out of Sample Accuracy = 0.83

$meanAccuracy

[1] 0.7943061

**MAXIMUM ENTROPY:**

Fold 1 Out of Sample Accuracy = 1

Fold 2 Out of Sample Accuracy = 1

Fold 3 Out of Sample Accuracy = 1

Fold 4 Out of Sample Accuracy = 1

Fold 5 Out of Sample Accuracy = 0.9896907

Fold 6 Out of Sample Accuracy = 1

Fold 7 Out of Sample Accuracy = 0.9809524

Fold 8 Out of Sample Accuracy = 0.9922481

Fold 9 Out of Sample Accuracy = 0.9897959

Fold 10 Out of Sample Accuracy = 1

$meanAccuracy

[1] 0.9952687

**RF:**

Fold 1 Out of Sample Accuracy = 0.76

Fold 2 Out of Sample Accuracy = 0.8165138

Fold 3 Out of Sample Accuracy = 0.826087

Fold 4 Out of Sample Accuracy = 0.7745098

Fold 5 Out of Sample Accuracy = 0.8359375

Fold 6 Out of Sample Accuracy = 0.7757009

Fold 7 Out of Sample Accuracy = 0.8095238

Fold 8 Out of Sample Accuracy = 0.776

Fold 9 Out of Sample Accuracy = 0.8113208

Fold 10 Out of Sample Accuracy = 0.7964602

meanAccuracy

[1] 0.7982054

**Applying WeightSMART:**

**Spec = NNN**

<<DocumentTermMatrix (documents: 1110, terms: 1878)>>

Non-/sparse entries : 6089/2078491

Sparsity : 100%

Maximal term length : 17

Weighting : SMART nnn (SMART)

**$RF**

Call:

randomForest(x = as.matrix(container@training\_matrix), y = container@training\_codes, ntree = ntree)

Type of random forest: classification

Number of trees: 200

No. of variables tried at each split: 43

OOB estimate of error rate: 24.6%

Confusion matrix:

1 2 class.error

1 419 80 0.1603206

2 166 335 0.3313373

Recall accuracy:

Tree: 82.72727%

SVM: 93.63636%

Max: 100%

RF: 99.09091%

ENSEMBLE SUMMARY

n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL

n >= 1 1.00 0.95

n >= 2 1.00 0.95

n >= 3 0.95 0.99

n >= 4 0.82 1.00

ALGORITHM PERFORMANCE

SVM\_PRECISION SVM\_RECALL SVM\_FSCORE

0.935 0.940 0.935

FORESTS\_PRECISION FORESTS\_RECALL FORESTS\_FSCORE

0.990 0.990 0.990

TREE\_PRECISION TREE\_RECALL TREE\_FSCORE

0.850 0.840 0.830

MAXENTROPY\_PRECISION MAXENTROPY\_RECALL MAXENTROPY\_FSCORE

1.000 1.000 1.000

Cross Validation:

Tree:

Fold 1 Out of Sample Accuracy = 0.776699

Fold 2 Out of Sample Accuracy = 0.7553191

Fold 3 Out of Sample Accuracy = 0.7222222

Fold 4 Out of Sample Accuracy = 0.675

Fold 5 Out of Sample Accuracy = 0.7876106

Fold 6 Out of Sample Accuracy = 0.7317073

Fold 7 Out of Sample Accuracy = 0.6190476

Fold 8 Out of Sample Accuracy = 0.68

Fold 9 Out of Sample Accuracy = 0.6916667

Fold 10 Out of Sample Accuracy = 0.7903226

meanAccuracy

[1] 0.7229595

SVM:

Fold 1 Out of Sample Accuracy = 0.7837838

Fold 2 Out of Sample Accuracy = 0.754386

Fold 3 Out of Sample Accuracy = 0.8018018

Fold 4 Out of Sample Accuracy = 0.8440367

Fold 5 Out of Sample Accuracy = 0.8

Fold 6 Out of Sample Accuracy = 0.8521739

Fold 7 Out of Sample Accuracy = 0.79

Fold 8 Out of Sample Accuracy = 0.7868852

Fold 9 Out of Sample Accuracy = 0.7931034

Fold 10 Out of Sample Accuracy = 0.8303571

meanAccuracy

[1] 0.8036528

MAxent:

Fold 1 Out of Sample Accuracy = 1

Fold 2 Out of Sample Accuracy = 0.9915966

Fold 3 Out of Sample Accuracy = 0.9907407

Fold 4 Out of Sample Accuracy = 1

Fold 5 Out of Sample Accuracy = 1

Fold 6 Out of Sample Accuracy = 0.9903846

Fold 7 Out of Sample Accuracy = 0.990566

Fold 8 Out of Sample Accuracy = 0.9894737

Fold 9 Out of Sample Accuracy = 1

Fold 10 Out of Sample Accuracy = 1

meanAccuracy

[1] 0.9952762

RF:

Fold 1 Out of Sample Accuracy = 0.8380952

Fold 2 Out of Sample Accuracy = 0.8269231

Fold 3 Out of Sample Accuracy = 0.8425197

Fold 4 Out of Sample Accuracy = 0.7087379

Fold 5 Out of Sample Accuracy = 0.8347826

Fold 6 Out of Sample Accuracy = 0.7333333

Fold 7 Out of Sample Accuracy = 0.7886179

Fold 8 Out of Sample Accuracy = 0.8034188

Fold 9 Out of Sample Accuracy = 0.7319588

Fold 10 Out of Sample Accuracy = 0.7719298

meanAccuracy

[1] 0.7880317

SPEC = lnn

<<DocumentTermMatrix (documents: 1110, terms: 1878)>>

Non-/sparse entries: 6089/2078491

Sparsity : 100%

Maximal term length: 17

Weighting : SMART lnn (SMART)

$RF

Call:

randomForest(x = as.matrix(container@training\_matrix), y = container@training\_codes, ntree = ntree)

Type of random forest: classification

Number of trees: 200

No. of variables tried at each split: 43

OOB estimate of error rate: 24%

Confusion matrix:

1 2 class.error

1 425 74 0.1482966

2 166 335 0.3313373

same accuracy table and recall accuracy

ENSEMBLE SUMMARY

n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL

n >= 1 1.00 0.95

n >= 2 1.00 0.95

n >= 3 0.95 0.98

n >= 4 0.82 1.00

ALGORITHM PERFORMANCE

SVM\_PRECISION SVM\_RECALL SVM\_FSCORE

0.935 0.940 0.935

FORESTS\_PRECISION FORESTS\_RECALL FORESTS\_FSCORE

0.975 0.970 0.970

TREE\_PRECISION TREE\_RECALL TREE\_FSCORE

0.850 0.840 0.830

MAXENTROPY\_PRECISION MAXENTROPY\_RECALL MAXENTROPY\_FSCORE

1.000 1.000 1.000

Cross validation:

Tree:

Fold 1 Out of Sample Accuracy = 0.7636364

Fold 2 Out of Sample Accuracy = 0.7154472

Fold 3 Out of Sample Accuracy = 0.73

Fold 4 Out of Sample Accuracy = 0.7007874

Fold 5 Out of Sample Accuracy = 0.7017544

Fold 6 Out of Sample Accuracy = 0.7345133

Fold 7 Out of Sample Accuracy = 0.7179487

Fold 8 Out of Sample Accuracy = 0.7142857

Fold 9 Out of Sample Accuracy = 0.7047619

Fold 10 Out of Sample Accuracy = 0.7727273

meanAccuracy

[1] 0.7255862

SVM:

Fold 1 Out of Sample Accuracy = 0.7372881

Fold 2 Out of Sample Accuracy = 0.8823529

Fold 3 Out of Sample Accuracy = 0.7358491

Fold 4 Out of Sample Accuracy = 0.8076923

Fold 5 Out of Sample Accuracy = 0.7864078

Fold 6 Out of Sample Accuracy = 0.7438017

Fold 7 Out of Sample Accuracy = 0.8317757

Fold 8 Out of Sample Accuracy = 0.7983193

Fold 9 Out of Sample Accuracy = 0.8256881

Fold 10 Out of Sample Accuracy = 0.8173077

meanAccuracy

[1] 0.7966483

max:

Fold 1 Out of Sample Accuracy = 1

Fold 2 Out of Sample Accuracy = 1

Fold 3 Out of Sample Accuracy = 0.9910714

Fold 4 Out of Sample Accuracy = 1

Fold 5 Out of Sample Accuracy = 0.9904762

Fold 6 Out of Sample Accuracy = 0.9907407

Fold 7 Out of Sample Accuracy = 0.9896907

Fold 8 Out of Sample Accuracy = 1

Fold 9 Out of Sample Accuracy = 1

Fold 10 Out of Sample Accuracy = 0.9923077

[[1]]

[1] 1.0000000 1.0000000 0.9910714 1.0000000 0.9904762 0.9907407 0.9896907

[8] 1.0000000 1.0000000 0.9923077

$meanAccuracy

[1] 0.9954287

Rf

Fold 1 Out of Sample Accuracy = 0.8141593

Fold 2 Out of Sample Accuracy = 0.7864078

Fold 3 Out of Sample Accuracy = 0.7890625

Fold 4 Out of Sample Accuracy = 0.7641509

Fold 5 Out of Sample Accuracy = 0.7521368

Fold 6 Out of Sample Accuracy = 0.7777778

Fold 7 Out of Sample Accuracy = 0.81

Fold 8 Out of Sample Accuracy = 0.7981651

Fold 9 Out of Sample Accuracy = 0.7941176

Fold 10 Out of Sample Accuracy = 0.8086957

[[1]]

[1] 0.8141593 0.7864078 0.7890625 0.7641509 0.7521368 0.7777778 0.8100000

[8] 0.7981651 0.7941176 0.8086957

$meanAccuracy

[1] 0.7894673

**spec = ltc**

svm

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 100

gamma: 0.0005324814

Number of Support Vectors: 723

$TREE

node), split, n, deviance, yval, (yprob)

\* denotes terminal node

1) root 1000 1386.000 2 ( 0.49900 0.50100 )

2) X712 < 0.437331 936 1293.000 1 ( 0.53312 0.46688 )

4) X698 < 1.02047 859 1178.000 1 ( 0.56112 0.43888 )

8) X414 < 0.714098 838 1143.000 1 ( 0.57518 0.42482 )

16) X29 < 0.816478 817 1107.000 1 ( 0.58874 0.41126 )

32) X1090 < 1.43491 796 1070.000 1 ( 0.60176 0.39824 )

64) X588 < 0.776182 784 1048.000 1 ( 0.61097 0.38903 )

128) X656 < 2.13312 772 1025.000 1 ( 0.62047 0.37953 )

256) X86 < 1.19229 763 1007.000 1 ( 0.62779 0.37221 )

512) X1362 < 2.31655 754 989.400 1 ( 0.63528 0.36472 )

1024) X957 < 2.30691 746 973.100 1 ( 0.64209 0.35791 )

2048) X958 < 1.1324 738 956.500 1 ( 0.64905 0.35095 )

4096) X1195 < 0.984643 730 939.600 1 ( 0.65616 0.34384 )

8192) X138 < 1.55327 707 897.700 1 ( 0.66902 0.33098 )

16384) X556 < 0.753007 700 882.100 1 ( 0.67571 0.32429 )

32768) X752 < 1.258 693 866.200 1 ( 0.68254 0.31746 ) \*

32769) X752 > 1.258 7 0.000 2 ( 0.00000 1.00000 ) \*

16385) X556 > 0.753007 7 0.000 2 ( 0.00000 1.00000 ) \*

8193) X138 > 1.55327 23 26.400 2 ( 0.26087 0.73913 ) \*

4097) X1195 > 0.984643 8 0.000 2 ( 0.00000 1.00000 ) \*

2049) X958 > 1.1324 8 0.000 2 ( 0.00000 1.00000 ) \*

1025) X957 > 2.30691 8 0.000 2 ( 0.00000 1.00000 ) \*

513) X1362 > 2.31655 9 0.000 2 ( 0.00000 1.00000 ) \*

257) X86 > 1.19229 9 0.000 2 ( 0.00000 1.00000 ) \*

129) X656 > 2.13312 12 0.000 2 ( 0.00000 1.00000 ) \*

65) X588 > 0.776182 12 0.000 2 ( 0.00000 1.00000 ) \*

33) X1090 > 1.43491 21 13.210 2 ( 0.09524 0.90476 ) \*

17) X29 > 0.816478 21 8.041 2 ( 0.04762 0.95238 ) \*

9) X414 > 0.714098 21 0.000 2 ( 0.00000 1.00000 ) \*

5) X698 > 1.02047 77 81.300 2 ( 0.22078 0.77922 ) \*

3) X712 > 0.437331 64 0.000 2 ( 0.00000 1.00000 ) \*

$RF

Call:

randomForest(x = as.matrix(container@training\_matrix), y = container@training\_codes, ntree = ntree)

Type of random forest: classification

Number of trees: 200

No. of variables tried at each split: 43

OOB estimate of error rate: 23.7%

Confusion matrix:

1 2 class.error

1 423 76 0.1523046

2 161 340 0.3213573

> table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results1[,"TREE\_LABEL"])

1 2

1 47 3

2 18 42

> table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results1[,"SVM\_LABEL"])

1 2

1 50 0

2 0 60

> table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results1[,"MAXENTROPY\_LABEL"])

1 2

1 50 0

2 0 60

> table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results1[,"FORESTS\_LABEL"])

1 2

1 48 2

2 1 59

recall\_acc :

tree:80.90909

svm:100

max:100

rf:97.27273

ENSEMBLE SUMMARY

n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL

n >= 1 1.00 0.99

n >= 2 1.00 0.99

n >= 3 0.97 1.00

n >= 4 0.81 1.00

ALGORITHM PERFORMANCE

SVM\_PRECISION SVM\_RECALL SVM\_FSCORE

1.000 1.000 1.000

FORESTS\_PRECISION FORESTS\_RECALL FORESTS\_FSCORE

0.975 0.970 0.970

TREE\_PRECISION TREE\_RECALL TREE\_FSCORE

0.825 0.820 0.810

MAXENTROPY\_PRECISION MAXENTROPY\_RECALL MAXENTROPY\_FSCORE

1.000 1.000 1.000

cross validation:

tree:

Fold 1 Out of Sample Accuracy = 0.7288136

Fold 2 Out of Sample Accuracy = 0.736

Fold 3 Out of Sample Accuracy = 0.65

Fold 4 Out of Sample Accuracy = 0.7653061

Fold 5 Out of Sample Accuracy = 0.7241379

Fold 6 Out of Sample Accuracy = 0.6333333

Fold 7 Out of Sample Accuracy = 0.7410714

Fold 8 Out of Sample Accuracy = 0.752

Fold 9 Out of Sample Accuracy = 0.752381

Fold 10 Out of Sample Accuracy = 0.7722772

[[1]]

[1] 0.7288136 0.7360000 0.6500000 0.7653061 0.7241379 0.6333333 0.7410714

[8] 0.7520000 0.7523810 0.7722772

$meanAccuracy

0.7255321

svm:

Fold 1 Out of Sample Accuracy = 0.8095238

Fold 2 Out of Sample Accuracy = 0.826087

Fold 3 Out of Sample Accuracy = 0.8181818

Fold 4 Out of Sample Accuracy = 0.8076923

Fold 5 Out of Sample Accuracy = 0.7716535

Fold 6 Out of Sample Accuracy = 0.7692308

Fold 7 Out of Sample Accuracy = 0.8557692

Fold 8 Out of Sample Accuracy = 0.8083333

Fold 9 Out of Sample Accuracy = 0.8508772

Fold 10 Out of Sample Accuracy = 0.847619

[[1]]

[1] 0.8095238 0.8260870 0.8181818 0.8076923 0.7716535 0.7692308 0.8557692

[8] 0.8083333 0.8508772 0.8476190

$meanAccuracy

[1] 0.8164968

maxent:

Fold 1 Out of Sample Accuracy = 1

Fold 2 Out of Sample Accuracy = 0.984127

Fold 3 Out of Sample Accuracy = 1

Fold 4 Out of Sample Accuracy = 0.9897959

Fold 5 Out of Sample Accuracy = 0.991453

Fold 6 Out of Sample Accuracy = 1

Fold 7 Out of Sample Accuracy = 0.9901961

Fold 8 Out of Sample Accuracy = 1

Fold 9 Out of Sample Accuracy = 1

Fold 10 Out of Sample Accuracy = 1

[[1]]

[1] 1.0000000 0.9841270 1.0000000 0.9897959 0.9914530 1.0000000 0.9901961

[8] 1.0000000 1.0000000 1.0000000

$meanAccuracy

[1] 0.9955572

rf:

Fold 1 Out of Sample Accuracy = 0.7837838

Fold 2 Out of Sample Accuracy = 0.7227723

Fold 3 Out of Sample Accuracy = 0.75

Fold 4 Out of Sample Accuracy = 0.7876106

Fold 5 Out of Sample Accuracy = 0.8461538

Fold 6 Out of Sample Accuracy = 0.8429752

Fold 7 Out of Sample Accuracy = 0.8058252

Fold 8 Out of Sample Accuracy = 0.8099174

Fold 9 Out of Sample Accuracy = 0.7121212

Fold 10 Out of Sample Accuracy = 0.8055556

[[1]]

[1] 0.7837838 0.7227723 0.7500000 0.7876106 0.8461538 0.8429752 0.8058252

[8] 0.8099174 0.7121212 0.8055556

$meanAccuracy

[1] 0.7866715

spec = lnu

<<DocumentTermMatrix (documents: 1110, terms: 1878)>>

Non-/sparse entries: 6089/2078491

Sparsity : 100%

Maximal term length: 17

Weighting : SMART lnc (SMART)

$SVM

Call:

svm.default(x = container@training\_matrix, y = container@training\_codes,

kernel = kernel, cost = cost, cross = cross, probability = TRUE,

method = method)

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 100

gamma: 0.0005324814

Number of Support Vectors: 972

$TREE

node), split, n, deviance, yval, (yprob)

\* denotes terminal node

1) root 1000 1386.000 2 ( 0.49900 0.50100 )

2) X712 < 0.114708 936 1293.000 1 ( 0.53312 0.46688 )

4) X698 < 0.295093 859 1178.000 1 ( 0.56112 0.43888 )

8) X414 < 0.129099 838 1143.000 1 ( 0.57518 0.42482 )

16) X29 < 0.150756 817 1107.000 1 ( 0.58874 0.41126 )

32) X1090 < 0.277968 796 1070.000 1 ( 0.60176 0.39824 )

64) X588 < 0.125 784 1048.000 1 ( 0.61097 0.38903 )

128) X656 < 0.427731 772 1025.000 1 ( 0.62047 0.37953 )

256) X86 < 0.188982 763 1007.000 1 ( 0.62779 0.37221 )

512) X1362 < 0.427731 754 989.400 1 ( 0.63528 0.36472 )

1024) X957 < 0.393106 746 973.100 1 ( 0.64209 0.35791 )

2048) X958 < 0.166667 738 956.500 1 ( 0.64905 0.35095 )

4096) X1195 < 0.150756 730 939.600 1 ( 0.65616 0.34384 )

8192) X138 < 0.30887 707 897.700 1 ( 0.66902 0.33098 )

16384) X556 < 0.121268 700 882.100 1 ( 0.67571 0.32429 )

32768) X752 < 0.176777 693 866.200 1 ( 0.68254 0.31746 ) \*

32769) X752 > 0.176777 7 0.000 2 ( 0.00000 1.00000 ) \*

16385) X556 > 0.121268 7 0.000 2 ( 0.00000 1.00000 ) \*

8193) X138 > 0.30887 23 26.400 2 ( 0.26087 0.73913 ) \*

4097) X1195 > 0.150756 8 0.000 2 ( 0.00000 1.00000 ) \*

2049) X958 > 0.166667 8 0.000 2 ( 0.00000 1.00000 ) \*

1025) X957 > 0.393106 8 0.000 2 ( 0.00000 1.00000 ) \*

513) X1362 > 0.427731 9 0.000 2 ( 0.00000 1.00000 ) \*

257) X86 > 0.188982 9 0.000 2 ( 0.00000 1.00000 ) \*

129) X656 > 0.427731 12 0.000 2 ( 0.00000 1.00000 ) \*

65) X588 > 0.125 12 0.000 2 ( 0.00000 1.00000 ) \*

33) X1090 > 0.277968 21 13.210 2 ( 0.09524 0.90476 ) \*

17) X29 > 0.150756 21 8.041 2 ( 0.04762 0.95238 ) \*

9) X414 > 0.129099 21 0.000 2 ( 0.00000 1.00000 ) \*

5) X698 > 0.295093 77 81.300 2 ( 0.22078 0.77922 ) \*

3) X712 > 0.114708 64 0.000 2 ( 0.00000 1.00000 ) \*

$RF

Call:

randomForest(x = as.matrix(container@training\_matrix), y = container@training\_codes, ntree = ntree)

Type of random forest: classification

Number of trees: 200

No. of variables tried at each split: 43

OOB estimate of error rate: 23.7%

Confusion matrix:

1 2 class.error

1 421 78 0.1563126

2 159 342 0.3173653

table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results2[,"TREE\_LABEL"])

1 2

1 47 3

2 18 42

> table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results2[,"SVM\_LABEL"])

1 2

1 47 3

2 6 54

> table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results2[,"MAXENTROPY\_LABEL"])

1 2

1 50 0

2 0 60

> table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results2[,"FORESTS\_LABEL"])

1 2

1 48 2

2 1 59

recall accuracy

tree: 80.9090

svm:91.81818

max:100

rf:97.27273

ENSEMBLE SUMMARY

n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL

n >= 1 1.00 0.92

n >= 2 1.00 0.92

n >= 3 0.94 0.98

n >= 4 0.80 1.00

ALGORITHM PERFORMANCE

SVM\_PRECISION SVM\_RECALL SVM\_FSCORE

0.920 0.920 0.915

FORESTS\_PRECISION FORESTS\_RECALL FORESTS\_FSCORE

0.975 0.970 0.970

TREE\_PRECISION TREE\_RECALL TREE\_FSCORE

0.825 0.820 0.810

MAXENTROPY\_PRECISION MAXENTROPY\_RECALL MAXENTROPY\_FSCORE

1.000 1.000 1.000

cross validation:

tree

Fold 1 Out of Sample Accuracy = 0.6810345

Fold 2 Out of Sample Accuracy = 0.6759259

Fold 3 Out of Sample Accuracy = 0.7304348

Fold 4 Out of Sample Accuracy = 0.7070707

Fold 5 Out of Sample Accuracy = 0.7171717

Fold 6 Out of Sample Accuracy = 0.76

Fold 7 Out of Sample Accuracy = 0.6808511

Fold 8 Out of Sample Accuracy = 0.778626

Fold 9 Out of Sample Accuracy = 0.6979167

Fold 10 Out of Sample Accuracy = 0.7716535

[[1]]

[1] 0.6810345 0.6759259 0.7304348 0.7070707 0.7171717 0.7600000

[7] 0.6808511 0.7786260 0.6979167 0.7716535

$meanAccuracy

[1] 0.7200685

svm

Fold 1 Out of Sample Accuracy = 0.6509434

Fold 2 Out of Sample Accuracy = 0.745614

Fold 3 Out of Sample Accuracy = 0.606383

Fold 4 Out of Sample Accuracy = 0.7207207

Fold 5 Out of Sample Accuracy = 0.8105263

Fold 6 Out of Sample Accuracy = 0.5982143

Fold 7 Out of Sample Accuracy = 0.7413793

Fold 8 Out of Sample Accuracy = 0.7692308

Fold 9 Out of Sample Accuracy = 0.7192982

Fold 10 Out of Sample Accuracy = 0.6610169

[[1]]

[1] 0.6509434 0.7456140 0.6063830 0.7207207 0.8105263 0.5982143

[7] 0.7413793 0.7692308 0.7192982 0.6610169

$meanAccuracy

[1] 0.7023327

max:

Fold 1 Out of Sample Accuracy = 0.9854015

Fold 2 Out of Sample Accuracy = 1

Fold 3 Out of Sample Accuracy = 1

Fold 4 Out of Sample Accuracy = 1

Fold 5 Out of Sample Accuracy = 0.98

Fold 6 Out of Sample Accuracy = 0.9919355

Fold 7 Out of Sample Accuracy = 1

Fold 8 Out of Sample Accuracy = 1

Fold 9 Out of Sample Accuracy = 1

Fold 10 Out of Sample Accuracy = 1

[[1]]

[1] 0.9854015 1.0000000 1.0000000 1.0000000 0.9800000 0.9919355

[7] 1.0000000 1.0000000 1.0000000 1.0000000

$meanAccuracy

[1] 0.9957337

rf:

Fold 1 Out of Sample Accuracy = 0.8113208

Fold 2 Out of Sample Accuracy = 0.784

Fold 3 Out of Sample Accuracy = 0.8173077

Fold 4 Out of Sample Accuracy = 0.7256637

Fold 5 Out of Sample Accuracy = 0.8061224

Fold 6 Out of Sample Accuracy = 0.8053097

Fold 7 Out of Sample Accuracy = 0.8050847

Fold 8 Out of Sample Accuracy = 0.7863248

Fold 9 Out of Sample Accuracy = 0.7425743

Fold 10 Out of Sample Accuracy = 0.8

[[1]]

[1] 0.8113208 0.7840000 0.8173077 0.7256637 0.8061224 0.8053097

[7] 0.8050847 0.7863248 0.7425743 0.8000000

$meanAccuracy

[1] 0.7883708

spec = npc

$SVM

Call:

svm.default(x = container@training\_matrix, y = container@training\_codes,

kernel = kernel, cost = cost, cross = cross, probability = TRUE,

method = method)

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 100

gamma: 0.0005324814

Number of Support Vectors: 669

$TREE

node), split, n, deviance, yval, (yprob)

\* denotes terminal node

1) root 1000 1386.000 2 ( 0.49900 0.50100 )

2) X712 < 1.16028 936 1293.000 1 ( 0.53312 0.46688 )

4) X698 < 2.98488 859 1178.000 1 ( 0.56112 0.43888 )

8) X414 < 1.30585 838 1143.000 1 ( 0.57518 0.42482 )

16) X29 < 1.5249 817 1107.000 1 ( 0.58874 0.41126 )

32) X1090 < 2.81166 796 1070.000 1 ( 0.60176 0.39824 )

64) X588 < 1.26438 784 1048.000 1 ( 0.61097 0.38903 )

128) X656 < 4.32652 772 1025.000 1 ( 0.62047 0.37953 )

256) X86 < 1.91156 763 1007.000 1 ( 0.62779 0.37221 )

512) X1362 < 4.32652 754 989.400 1 ( 0.63528 0.36472 )

1024) X957 < 3.97629 746 973.100 1 ( 0.64209 0.35791 )

2048) X958 < 1.68584 738 956.500 1 ( 0.64905 0.35095 )

4096) X1195 < 1.5249 730 939.600 1 ( 0.65616 0.34384 )

8192) X138 < 3.12423 707 897.700 1 ( 0.66902 0.33098 )

16384) X556 < 1.22663 700 882.100 1 ( 0.67571 0.32429 )

32768) X752 < 1.7881 693 866.200 1 ( 0.68254 0.31746 ) \*

32769) X752 > 1.7881 7 0.000 2 ( 0.00000 1.00000 ) \*

16385) X556 > 1.22663 7 0.000 2 ( 0.00000 1.00000 ) \*

8193) X138 > 3.12423 23 26.400 2 ( 0.26087 0.73913 ) \*

4097) X1195 > 1.5249 8 0.000 2 ( 0.00000 1.00000 ) \*

2049) X958 > 1.68584 8 0.000 2 ( 0.00000 1.00000 ) \*

1025) X957 > 3.97629 8 0.000 2 ( 0.00000 1.00000 ) \*

513) X1362 > 4.32652 9 0.000 2 ( 0.00000 1.00000 ) \*

257) X86 > 1.91156 9 0.000 2 ( 0.00000 1.00000 ) \*

129) X656 > 4.32652 12 0.000 2 ( 0.00000 1.00000 ) \*

65) X588 > 1.26438 12 0.000 2 ( 0.00000 1.00000 ) \*

33) X1090 > 2.81166 21 13.210 2 ( 0.09524 0.90476 ) \*

17) X29 > 1.5249 21 8.041 2 ( 0.04762 0.95238 ) \*

9) X414 > 1.30585 21 0.000 2 ( 0.00000 1.00000 ) \*

5) X698 > 2.98488 77 81.300 2 ( 0.22078 0.77922 ) \*

3) X712 > 1.16028 64 0.000 2 ( 0.00000 1.00000 ) \*

$RF

Call:

randomForest(x = as.matrix(container@training\_matrix), y = container@training\_codes, ntree = ntree)

Type of random forest: classification

Number of trees: 200

No. of variables tried at each split: 43

OOB estimate of error rate: 24.6%

Confusion matrix:

1 2 class.error

1 420 79 0.1583166

2 167 334 0.3333333

> table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results2[,"TREE\_LABEL"])

1 2

1 47 3

2 18 42

> table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results2[,"SVM\_LABEL"])

1 2

1 50 0

2 0 60

> table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results2[,"MAXENTROPY\_LABEL"])

1 2

1 50 0

2 0 60

> table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results2[,"FORESTS\_LABEL"])

1 2

1 48 2

2 2 58

recall accuracy :

svm, tree, max = same previous

rf = 96.3636

ENSEMBLE SUMMARY

n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL

n >= 1 1.00 0.98

n >= 2 1.00 0.98

n >= 3 0.96 1.00

n >= 4 0.81 1.00

ALGORITHM PERFORMANCE

SVM\_PRECISION SVM\_RECALL SVM\_FSCORE

1.000 1.000 1.000

FORESTS\_PRECISION FORESTS\_RECALL FORESTS\_FSCORE

0.965 0.965 0.965

TREE\_PRECISION TREE\_RECALL TREE\_FSCORE

0.825 0.820 0.810

MAXENTROPY\_PRECISION MAXENTROPY\_RECALL MAXENTROPY\_FSCORE

1.000 1.000 1.000

> cross\_validate(container2,N,"TREE")

Fold 1 Out of Sample Accuracy = 0.7358491

Fold 2 Out of Sample Accuracy = 0.6909091

Fold 3 Out of Sample Accuracy = 0.7657658

Fold 4 Out of Sample Accuracy = 0.7297297

Fold 5 Out of Sample Accuracy = 0.6788991

Fold 6 Out of Sample Accuracy = 0.7398374

Fold 7 Out of Sample Accuracy = 0.7196262

Fold 8 Out of Sample Accuracy = 0.7364341

Fold 9 Out of Sample Accuracy = 0.7906977

Fold 10 Out of Sample Accuracy = 0.7033898

[[1]]

[1] 0.7358491 0.6909091 0.7657658 0.7297297 0.6788991 0.7398374

[7] 0.7196262 0.7364341 0.7906977 0.7033898

$meanAccuracy

[1] 0.7291138

> cross\_validate(container2,N,"SVM")

Fold 1 Out of Sample Accuracy = 0.8105263

Fold 2 Out of Sample Accuracy = 0.8559322

Fold 3 Out of Sample Accuracy = 0.8640777

Fold 4 Out of Sample Accuracy = 0.7583333

Fold 5 Out of Sample Accuracy = 0.8278689

Fold 6 Out of Sample Accuracy = 0.8389831

Fold 7 Out of Sample Accuracy = 0.7916667

Fold 8 Out of Sample Accuracy = 0.8256881

Fold 9 Out of Sample Accuracy = 0.8415842

Fold 10 Out of Sample Accuracy = 0.7211538

[[1]]

[1] 0.8105263 0.8559322 0.8640777 0.7583333 0.8278689 0.8389831

[7] 0.7916667 0.8256881 0.8415842 0.7211538

$meanAccuracy

[1] 0.8135814

> cross\_validate(container2,N,"MAXENT")

Fold 1 Out of Sample Accuracy = 1

Fold 2 Out of Sample Accuracy = 1

Fold 3 Out of Sample Accuracy = 0.9903846

Fold 4 Out of Sample Accuracy = 0.9909091

Fold 5 Out of Sample Accuracy = 1

Fold 6 Out of Sample Accuracy = 1

Fold 7 Out of Sample Accuracy = 1

Fold 8 Out of Sample Accuracy = 1

Fold 9 Out of Sample Accuracy = 0.980198

Fold 10 Out of Sample Accuracy = 0.9912281

[[1]]

[1] 1.0000000 1.0000000 0.9903846 0.9909091 1.0000000 1.0000000

[7] 1.0000000 1.0000000 0.9801980 0.9912281

$meanAccuracy

[1] 0.995272

> cross\_validate(container2,N,"RF")

Fold 1 Out of Sample Accuracy = 0.7636364

Fold 2 Out of Sample Accuracy = 0.8165138

Fold 3 Out of Sample Accuracy = 0.7652174

Fold 4 Out of Sample Accuracy = 0.7948718

Fold 5 Out of Sample Accuracy = 0.7614679

Fold 6 Out of Sample Accuracy = 0.7818182

Fold 7 Out of Sample Accuracy = 0.7657658

Fold 8 Out of Sample Accuracy = 0.8782609

Fold 9 Out of Sample Accuracy = 0.8272727

Fold 10 Out of Sample Accuracy = 0.8365385

[[1]]

[1] 0.7636364 0.8165138 0.7652174 0.7948718 0.7614679 0.7818182

[7] 0.7657658 0.8782609 0.8272727 0.8365385

$meanAccuracy

[1] 0.7991363

spec = anc

$SVM

Call:

svm.default(x = container@training\_matrix, y = container@training\_codes,

kernel = kernel, cost = cost, cross = cross, probability = TRUE,

method = method)

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 100

gamma: 0.0005324814

Number of Support Vectors: 976

$TREE

node), split, n, deviance, yval, (yprob)

\* denotes terminal node

1) root 1000 1386.000 2 ( 0.49900 0.50100 )

2) X712 < 0.122068 936 1293.000 1 ( 0.53312 0.46688 )

4) X698 < 0.29003 858 1176.000 1 ( 0.56177 0.43823 )

8) X414 < 0.129099 837 1141.000 1 ( 0.57587 0.42413 )

16) X29 < 0.150756 816 1105.000 1 ( 0.58946 0.41054 )

32) X1090 < 0.277968 794 1067.000 1 ( 0.60327 0.39673 )

64) X588 < 0.134704 782 1044.000 1 ( 0.61253 0.38747 )

128) X656 < 0.427731 770 1021.000 1 ( 0.62208 0.37792 )

256) X86 < 0.188982 761 1003.000 1 ( 0.62943 0.37057 )

512) X556 < 0.130066 753 987.400 1 ( 0.63612 0.36388 )

1024) X957 < 0.393106 745 971.000 1 ( 0.64295 0.35705 )

2048) X958 < 0.166667 737 954.400 1 ( 0.64993 0.35007 )

4096) X1195 < 0.166667 729 937.400 1 ( 0.65706 0.34294 )

8192) X138 < 0.30887 706 895.500 1 ( 0.66997 0.33003 )

16384) X752 < 0.176777 699 879.800 1 ( 0.67668 0.32332 ) \*

16385) X752 > 0.176777 7 0.000 2 ( 0.00000 1.00000 ) \*

8193) X138 > 0.30887 23 26.400 2 ( 0.26087 0.73913 ) \*

4097) X1195 > 0.166667 8 0.000 2 ( 0.00000 1.00000 ) \*

2049) X958 > 0.166667 8 0.000 2 ( 0.00000 1.00000 ) \*

1025) X957 > 0.393106 8 0.000 2 ( 0.00000 1.00000 ) \*

513) X556 > 0.130066 8 0.000 2 ( 0.00000 1.00000 ) \*

257) X86 > 0.188982 9 0.000 2 ( 0.00000 1.00000 ) \*

129) X656 > 0.427731 12 0.000 2 ( 0.00000 1.00000 ) \*

65) X588 > 0.134704 12 0.000 2 ( 0.00000 1.00000 ) \*

33) X1090 > 0.277968 22 13.400 2 ( 0.09091 0.90909 ) \*

17) X29 > 0.150756 21 8.041 2 ( 0.04762 0.95238 ) \*

9) X414 > 0.129099 21 0.000 2 ( 0.00000 1.00000 ) \*

5) X698 > 0.29003 78 81.790 2 ( 0.21795 0.78205 ) \*

3) X712 > 0.122068 64 0.000 2 ( 0.00000 1.00000 ) \*

$RF

Call:

randomForest(x = as.matrix(container@training\_matrix), y = container@training\_codes, ntree = ntree)

Type of random forest: classification

Number of trees: 200

No. of variables tried at each split: 43

OOB estimate of error rate: 23.4%

Confusion matrix:

1 2 class.error

1 421 78 0.1563126

2 156 345 0.3113772

recall accuracy

svm: 91.81818

tree:80.90

max:100

rf:97.27273

ENSEMBLE SUMMARY

n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL

n >= 1 1.00 0.92

n >= 2 1.00 0.92

n >= 3 0.94 0.98

n >= 4 0.80 1.00

ALGORITHM PERFORMANCE

SVM\_PRECISION SVM\_RECALL SVM\_FSCORE

0.920 0.920 0.915

FORESTS\_PRECISION FORESTS\_RECALL FORESTS\_FSCORE

0.975 0.970 0.970

TREE\_PRECISION TREE\_RECALL TREE\_FSCORE

0.825 0.820 0.810

MAXENTROPY\_PRECISION MAXENTROPY\_RECALL MAXENTROPY\_FSCORE

1.000 1.000 1.000

> cross\_validate(container2,N,"TREE")

Fold 1 Out of Sample Accuracy = 0.6633663

Fold 2 Out of Sample Accuracy = 0.6717557

Fold 3 Out of Sample Accuracy = 0.7391304

Fold 4 Out of Sample Accuracy = 0.7555556

Fold 5 Out of Sample Accuracy = 0.6904762

Fold 6 Out of Sample Accuracy = 0.6989247

Fold 7 Out of Sample Accuracy = 0.795082

Fold 8 Out of Sample Accuracy = 0.7391304

Fold 9 Out of Sample Accuracy = 0.7857143

Fold 10 Out of Sample Accuracy = 0.6571429

[[1]]

[1] 0.6633663 0.6717557 0.7391304 0.7555556 0.6904762 0.6989247

[7] 0.7950820 0.7391304 0.7857143 0.6571429

$meanAccuracy

[1] 0.7196279

> cross\_validate(container2,N,"SVM")

Fold 1 Out of Sample Accuracy = 0.7222222

Fold 2 Out of Sample Accuracy = 0.6285714

Fold 3 Out of Sample Accuracy = 0.7222222

Fold 4 Out of Sample Accuracy = 0.6666667

Fold 5 Out of Sample Accuracy = 0.7346939

Fold 6 Out of Sample Accuracy = 0.6434109

Fold 7 Out of Sample Accuracy = 0.7592593

Fold 8 Out of Sample Accuracy = 0.6229508

Fold 9 Out of Sample Accuracy = 0.7857143

Fold 10 Out of Sample Accuracy = 0.6893204

[[1]]

[1] 0.7222222 0.6285714 0.7222222 0.6666667 0.7346939 0.6434109

[7] 0.7592593 0.6229508 0.7857143 0.6893204

$meanAccuracy

[1] 0.6975032

> cross\_validate(container2,N,"MAXENT")

Fold 1 Out of Sample Accuracy = 1

Fold 2 Out of Sample Accuracy = 0.9908257

Fold 3 Out of Sample Accuracy = 1

Fold 4 Out of Sample Accuracy = 0.990566

Fold 5 Out of Sample Accuracy = 0.9909091

Fold 6 Out of Sample Accuracy = 1

Fold 7 Out of Sample Accuracy = 1

Fold 8 Out of Sample Accuracy = 0.9833333

Fold 9 Out of Sample Accuracy = 1

Fold 10 Out of Sample Accuracy = 1

[[1]]

[1] 1.0000000 0.9908257 1.0000000 0.9905660 0.9909091 1.0000000

[7] 1.0000000 0.9833333 1.0000000 1.0000000

$meanAccuracy

[1] 0.9955634

> cross\_validate(container2,N,"RF")

Fold 1 Out of Sample Accuracy = 0.7864078

Fold 2 Out of Sample Accuracy = 0.8148148

Fold 3 Out of Sample Accuracy = 0.8350515

Fold 4 Out of Sample Accuracy = 0.7938931

Fold 5 Out of Sample Accuracy = 0.7745098

Fold 6 Out of Sample Accuracy = 0.8534483

Fold 7 Out of Sample Accuracy = 0.7593985

Fold 8 Out of Sample Accuracy = 0.8350515

Fold 9 Out of Sample Accuracy = 0.75

Fold 10 Out of Sample Accuracy = 0.7478261

[[1]]

[1] 0.7864078 0.8148148 0.8350515 0.7938931 0.7745098 0.8534483

[7] 0.7593985 0.8350515 0.7500000 0.7478261

$meanAccuracy

[1] 0.7950401

specs = bnc

SVM

Call:

svm.default(x = container@training\_matrix, y = container@training\_codes,

kernel = kernel, cost = cost, cross = cross, probability = TRUE,

method = method)

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 100

gamma: 0.0005324814

Number of Support Vectors: 977

$TREE

node), split, n, deviance, yval, (yprob)

\* denotes terminal node

1) root 1000 1386.000 2 ( 0.49900 0.50100 )

2) X712 < 0.125 936 1293.000 1 ( 0.53312 0.46688 )

4) X698 < 0.295093 856 1174.000 1 ( 0.56192 0.43808 )

8) X414 < 0.129099 835 1138.000 1 ( 0.57605 0.42395 )

16) X29 < 0.150756 814 1102.000 1 ( 0.58968 0.41032 )

32) X1090 < 0.277968 792 1064.000 1 ( 0.60354 0.39646 )

64) X588 < 0.138675 780 1041.000 1 ( 0.61282 0.38718 )

128) X656 < 0.427731 768 1018.000 1 ( 0.62240 0.37760 )

256) X86 < 0.188982 759 1000.000 1 ( 0.62978 0.37022 )

512) X556 < 0.133631 751 984.400 1 ( 0.63648 0.36352 )

1024) X957 < 0.393106 743 968.100 1 ( 0.64334 0.35666 )

2048) X958 < 0.166667 735 951.400 1 ( 0.65034 0.34966 )

4096) X1195 < 0.166667 727 934.500 1 ( 0.65750 0.34250 )

8192) X752 < 0.176777 720 919.300 1 ( 0.66389 0.33611 )

16384) X138 < 0.30887 698 879.000 1 ( 0.67622 0.32378 ) \*

16385) X138 > 0.30887 22 25.780 2 ( 0.27273 0.72727 ) \*

8193) X752 > 0.176777 7 0.000 2 ( 0.00000 1.00000 ) \*

4097) X1195 > 0.166667 8 0.000 2 ( 0.00000 1.00000 ) \*

2049) X958 > 0.166667 8 0.000 2 ( 0.00000 1.00000 ) \*

1025) X957 > 0.393106 8 0.000 2 ( 0.00000 1.00000 ) \*

513) X556 > 0.133631 8 0.000 2 ( 0.00000 1.00000 ) \*

257) X86 > 0.188982 9 0.000 2 ( 0.00000 1.00000 ) \*

129) X656 > 0.427731 12 0.000 2 ( 0.00000 1.00000 ) \*

65) X588 > 0.138675 12 0.000 2 ( 0.00000 1.00000 ) \*

33) X1090 > 0.277968 22 13.400 2 ( 0.09091 0.90909 ) \*

17) X29 > 0.150756 21 8.041 2 ( 0.04762 0.95238 ) \*

9) X414 > 0.129099 21 0.000 2 ( 0.00000 1.00000 ) \*

5) X698 > 0.295093 80 85.310 2 ( 0.22500 0.77500 ) \*

3) X712 > 0.125 64 0.000 2 ( 0.00000 1.00000 ) \*

$RF

Call:

randomForest(x = as.matrix(container@training\_matrix), y = container@training\_codes, ntree = ntree)

Type of random forest: classification

Number of trees: 200

No. of variables tried at each split: 43

OOB estimate of error rate: 23.8%

Confusion matrix:

1 2 class.error

1 427 72 0.1442886

2 166 335 0.3313373

recall accuracy

rf = 98.18182

tree = 80.90

svm =91.81

max = 100

ENSEMBLE SUMMARY

n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL

n >= 1 1.00 0.93

n >= 2 1.00 0.93

n >= 3 0.93 0.99

n >= 4 0.80 1.00

ALGORITHM PERFORMANCE

SVM\_PRECISION SVM\_RECALL SVM\_FSCORE

0.920 0.920 0.915

FORESTS\_PRECISION FORESTS\_RECALL FORESTS\_FSCORE

0.980 0.980 0.980

TREE\_PRECISION TREE\_RECALL TREE\_FSCORE

0.825 0.820 0.810

MAXENTROPY\_PRECISION MAXENTROPY\_RECALL MAXENTROPY\_FSCORE

1.000 1.000 1.000

> N=10

> cross\_validate(container2,N,"TREE")

Fold 1 Out of Sample Accuracy = 0.7089552

Fold 2 Out of Sample Accuracy = 0.6633663

Fold 3 Out of Sample Accuracy = 0.6969697

Fold 4 Out of Sample Accuracy = 0.6929825

Fold 5 Out of Sample Accuracy = 0.7338129

Fold 6 Out of Sample Accuracy = 0.7128713

Fold 7 Out of Sample Accuracy = 0.7473684

Fold 8 Out of Sample Accuracy = 0.7572816

Fold 9 Out of Sample Accuracy = 0.7169811

Fold 10 Out of Sample Accuracy = 0.7288136

[[1]]

[1] 0.7089552 0.6633663 0.6969697 0.6929825 0.7338129 0.7128713

[7] 0.7473684 0.7572816 0.7169811 0.7288136

$meanAccuracy

[1] 0.7159403

> cross\_validate(container2,N,"SVM")

Fold 1 Out of Sample Accuracy = 0.6530612

Fold 2 Out of Sample Accuracy = 0.792

Fold 3 Out of Sample Accuracy = 0.7179487

Fold 4 Out of Sample Accuracy = 0.745098

Fold 5 Out of Sample Accuracy = 0.7

Fold 6 Out of Sample Accuracy = 0.6492537

Fold 7 Out of Sample Accuracy = 0.6851852

Fold 8 Out of Sample Accuracy = 0.7079646

Fold 9 Out of Sample Accuracy = 0.6862745

Fold 10 Out of Sample Accuracy = 0.6534653

[[1]]

[1] 0.6530612 0.7920000 0.7179487 0.7450980 0.7000000 0.6492537

[7] 0.6851852 0.7079646 0.6862745 0.6534653

$meanAccuracy

[1] 0.6990251

> cross\_validate(container2,N,"MAXENT")

Fold 1 Out of Sample Accuracy = 1

Fold 2 Out of Sample Accuracy = 0.9826087

Fold 3 Out of Sample Accuracy = 1

Fold 4 Out of Sample Accuracy = 1

Fold 5 Out of Sample Accuracy = 0.9918699

Fold 6 Out of Sample Accuracy = 0.990566

Fold 7 Out of Sample Accuracy = 0.990991

Fold 8 Out of Sample Accuracy = 1

Fold 9 Out of Sample Accuracy = 1

Fold 10 Out of Sample Accuracy = 1

[[1]]

[1] 1.0000000 0.9826087 1.0000000 1.0000000 0.9918699 0.9905660

[7] 0.9909910 1.0000000 1.0000000 1.0000000

$meanAccuracy

[1] 0.9956036

> cross\_validate(container2,N,"RF")

Fold 1 Out of Sample Accuracy = 0.8148148

Fold 2 Out of Sample Accuracy = 0.7663551

Fold 3 Out of Sample Accuracy = 0.7363636

Fold 4 Out of Sample Accuracy = 0.8173913

Fold 5 Out of Sample Accuracy = 0.8389831

Fold 6 Out of Sample Accuracy = 0.8585859

Fold 7 Out of Sample Accuracy = 0.75

Fold 8 Out of Sample Accuracy = 0.8113208

Fold 9 Out of Sample Accuracy = 0.7723577

Fold 10 Out of Sample Accuracy = 0.8017241

[[1]]

[1] 0.8148148 0.7663551 0.7363636 0.8173913 0.8389831 0.8585859

[7] 0.7500000 0.8113208 0.7723577 0.8017241

$meanAccuracy

[1] 0.7967896

spec = Lnc

$SVM

Call:

svm.default(x = container@training\_matrix, y = container@training\_codes,

kernel = kernel, cost = cost, cross = cross, probability = TRUE,

method = method)

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 100

gamma: 0.0005324814

Number of Support Vectors: 972

$RF

Call:

randomForest(x = as.matrix(container@training\_matrix), y = container@training\_codes, ntree = ntree)

Type of random forest: classification

Number of trees: 200

No. of variables tried at each split: 43

OOB estimate of error rate: 23.2%

Confusion matrix:

1 2 class.error

1 419 80 0.1603206

2 152 349 0.3033932

recall accuracy

rf = 97.27273

tree = 80.90

svm =91.81

max = 100

ENSEMBLE SUMMARY

n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL

n >= 1 1.00 0.92

n >= 2 1.00 0.92

n >= 3 0.94 0.98

n >= 4 0.80 1.00

ALGORITHM PERFORMANCE

SVM\_PRECISION SVM\_RECALL SVM\_FSCORE

0.920 0.920 0.915

FORESTS\_PRECISION FORESTS\_RECALL FORESTS\_FSCORE

0.975 0.970 0.970

TREE\_PRECISION TREE\_RECALL TREE\_FSCORE

0.825 0.820 0.810

MAXENTROPY\_PRECISION MAXENTROPY\_RECALL MAXENTROPY\_FSCORE

1.000 1.000 1.000

> N=10

> cross\_validate(container2,N,"TREE")

Fold 1 Out of Sample Accuracy = 0.7070707

Fold 2 Out of Sample Accuracy = 0.7368421

Fold 3 Out of Sample Accuracy = 0.7380952

Fold 4 Out of Sample Accuracy = 0.8198198

Fold 5 Out of Sample Accuracy = 0.7407407

Fold 6 Out of Sample Accuracy = 0.7232143

Fold 7 Out of Sample Accuracy = 0.6754386

Fold 8 Out of Sample Accuracy = 0.7333333

Fold 9 Out of Sample Accuracy = 0.7075472

Fold 10 Out of Sample Accuracy = 0.6806723

[[1]]

[1] 0.7070707 0.7368421 0.7380952 0.8198198 0.7407407 0.7232143

[7] 0.6754386 0.7333333 0.7075472 0.6806723

$meanAccuracy

[1] 0.7262774

> cross\_validate(container2,N,"SVM")

Fold 1 Out of Sample Accuracy = 0.6283186

Fold 2 Out of Sample Accuracy = 0.7767857

Fold 3 Out of Sample Accuracy = 0.6339286

Fold 4 Out of Sample Accuracy = 0.6504854

Fold 5 Out of Sample Accuracy = 0.625

Fold 6 Out of Sample Accuracy = 0.7214286

Fold 7 Out of Sample Accuracy = 0.7920792

Fold 8 Out of Sample Accuracy = 0.775

Fold 9 Out of Sample Accuracy = 0.7321429

Fold 10 Out of Sample Accuracy = 0.6534653

[[1]]

[1] 0.6283186 0.7767857 0.6339286 0.6504854 0.6250000 0.7214286

[7] 0.7920792 0.7750000 0.7321429 0.6534653

$meanAccuracy

[1] 0.6988634

> cross\_validate(container2,N,"MAXENT")

Fold 1 Out of Sample Accuracy = 1

Fold 2 Out of Sample Accuracy = 1

Fold 3 Out of Sample Accuracy = 0.990566

Fold 4 Out of Sample Accuracy = 1

Fold 5 Out of Sample Accuracy = 1

Fold 6 Out of Sample Accuracy = 1

Fold 7 Out of Sample Accuracy = 1

Fold 8 Out of Sample Accuracy = 0.9918033

Fold 9 Out of Sample Accuracy = 0.991453

Fold 10 Out of Sample Accuracy = 0.9824561

[[1]]

[1] 1.0000000 1.0000000 0.9905660 1.0000000 1.0000000 1.0000000

[7] 1.0000000 0.9918033 0.9914530 0.9824561

$meanAccuracy

[1] 0.9956278

> cross\_validate(container2,N,"RF")

Fold 1 Out of Sample Accuracy = 0.7863248

Fold 2 Out of Sample Accuracy = 0.7837838

Fold 3 Out of Sample Accuracy = 0.789916

Fold 4 Out of Sample Accuracy = 0.8365385

Fold 5 Out of Sample Accuracy = 0.754386

Fold 6 Out of Sample Accuracy = 0.7610619

Fold 7 Out of Sample Accuracy = 0.7821782

Fold 8 Out of Sample Accuracy = 0.8504673

Fold 9 Out of Sample Accuracy = 0.7923077

Fold 10 Out of Sample Accuracy = 0.787234

[[1]]

[1] 0.7863248 0.7837838 0.7899160 0.8365385 0.7543860 0.7610619

[7] 0.7821782 0.8504673 0.7923077 0.7872340

$meanAccuracy

[1] 0.7924198

Spec = lpc

SVM

Call:

svm.default(x = container@training\_matrix, y = container@training\_codes,

kernel = kernel, cost = cost, cross = cross, probability = TRUE,

method = method)

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 100

gamma: 0.0005324814

Number of Support Vectors: 670

$TREE

node), split, n, deviance, yval, (yprob)

\* denotes terminal node

1) root 1000 1386.000 2 ( 0.49900 0.50100 )

2) X712 < 1.16028 936 1293.000 1 ( 0.53312 0.46688 )

4) X698 < 2.98488 859 1178.000 1 ( 0.56112 0.43888 )

8) X414 < 1.30585 838 1143.000 1 ( 0.57518 0.42482 )

16) X29 < 1.5249 817 1107.000 1 ( 0.58874 0.41126 )

32) X1090 < 2.81166 796 1070.000 1 ( 0.60176 0.39824 )

64) X588 < 1.26438 784 1048.000 1 ( 0.61097 0.38903 )

128) X656 < 4.32652 772 1025.000 1 ( 0.62047 0.37953 )

256) X86 < 1.91156 763 1007.000 1 ( 0.62779 0.37221 )

512) X1362 < 4.32652 754 989.400 1 ( 0.63528 0.36472 )

1024) X957 < 3.97629 746 973.100 1 ( 0.64209 0.35791 )

2048) X958 < 1.68584 738 956.500 1 ( 0.64905 0.35095 )

4096) X1195 < 1.5249 730 939.600 1 ( 0.65616 0.34384 )

8192) X138 < 3.12423 707 897.700 1 ( 0.66902 0.33098 )

16384) X556 < 1.22663 700 882.100 1 ( 0.67571 0.32429 )

32768) X752 < 1.7881 693 866.200 1 ( 0.68254 0.31746 ) \*

32769) X752 > 1.7881 7 0.000 2 ( 0.00000 1.00000 ) \*

16385) X556 > 1.22663 7 0.000 2 ( 0.00000 1.00000 ) \*

8193) X138 > 3.12423 23 26.400 2 ( 0.26087 0.73913 ) \*

4097) X1195 > 1.5249 8 0.000 2 ( 0.00000 1.00000 ) \*

2049) X958 > 1.68584 8 0.000 2 ( 0.00000 1.00000 ) \*

1025) X957 > 3.97629 8 0.000 2 ( 0.00000 1.00000 ) \*

513) X1362 > 4.32652 9 0.000 2 ( 0.00000 1.00000 ) \*

257) X86 > 1.91156 9 0.000 2 ( 0.00000 1.00000 ) \*

129) X656 > 4.32652 12 0.000 2 ( 0.00000 1.00000 ) \*

65) X588 > 1.26438 12 0.000 2 ( 0.00000 1.00000 ) \*

33) X1090 > 2.81166 21 13.210 2 ( 0.09524 0.90476 ) \*

17) X29 > 1.5249 21 8.041 2 ( 0.04762 0.95238 ) \*

9) X414 > 1.30585 21 0.000 2 ( 0.00000 1.00000 ) \*

5) X698 > 2.98488 77 81.300 2 ( 0.22078 0.77922 ) \*

3) X712 > 1.16028 64 0.000 2 ( 0.00000 1.00000 ) \*

$RF

Call:

randomForest(x = as.matrix(container@training\_matrix), y = container@training\_codes, ntree = ntree)

Type of random forest: classification

Number of trees: 200

No. of variables tried at each split: 43

OOB estimate of error rate: 24%

Confusion matrix:

1 2 class.error

1 421 78 0.1563126

2 162 339 0.3233533

Recall accuracy

Tree =80.91

Svm = 100

Rf = =97.27273

Max = 100

ENSEMBLE SUMMARY

n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL

n >= 1 1.00 0.99

n >= 2 1.00 0.99

n >= 3 0.97 1.00

n >= 4 0.81 1.00

ALGORITHM PERFORMANCE

SVM\_PRECISION SVM\_RECALL SVM\_FSCORE

1.000 1.000 1.000

FORESTS\_PRECISION FORESTS\_RECALL FORESTS\_FSCORE

0.975 0.970 0.970

TREE\_PRECISION TREE\_RECALL TREE\_FSCORE

0.825 0.820 0.810

MAXENTROPY\_PRECISION MAXENTROPY\_RECALL MAXENTROPY\_FSCORE

1.000 1.000 1.000

> N=10

> cross\_validate(container2,N,"TREE")

Fold 1 Out of Sample Accuracy = 0.7328244

Fold 2 Out of Sample Accuracy = 0.6666667

Fold 3 Out of Sample Accuracy = 0.6495726

Fold 4 Out of Sample Accuracy = 0.7666667

Fold 5 Out of Sample Accuracy = 0.7043478

Fold 6 Out of Sample Accuracy = 0.7325581

Fold 7 Out of Sample Accuracy = 0.7454545

Fold 8 Out of Sample Accuracy = 0.7154472

Fold 9 Out of Sample Accuracy = 0.7236842

Fold 10 Out of Sample Accuracy = 0.7244094

[[1]]

[1] 0.7328244 0.6666667 0.6495726 0.7666667 0.7043478 0.7325581

[7] 0.7454545 0.7154472 0.7236842 0.7244094

$meanAccuracy

[1] 0.7161632

> cross\_validate(container2,N,"SVM")

Fold 1 Out of Sample Accuracy = 0.7948718

Fold 2 Out of Sample Accuracy = 0.7542373

Fold 3 Out of Sample Accuracy = 0.8557692

Fold 4 Out of Sample Accuracy = 0.7941176

Fold 5 Out of Sample Accuracy = 0.8220339

Fold 6 Out of Sample Accuracy = 0.847619

Fold 7 Out of Sample Accuracy = 0.8203125

Fold 8 Out of Sample Accuracy = 0.7666667

Fold 9 Out of Sample Accuracy = 0.7802198

Fold 10 Out of Sample Accuracy = 0.8224299

[[1]]

[1] 0.7948718 0.7542373 0.8557692 0.7941176 0.8220339 0.8476190

[7] 0.8203125 0.7666667 0.7802198 0.8224299

$meanAccuracy

[1] 0.8058278

> cross\_validate(container2,N,"MAXENT")

Fold 1 Out of Sample Accuracy = 1

Fold 2 Out of Sample Accuracy = 0.9805825

Fold 3 Out of Sample Accuracy = 1

Fold 4 Out of Sample Accuracy = 0.9911504

Fold 5 Out of Sample Accuracy = 1

Fold 6 Out of Sample Accuracy = 1

Fold 7 Out of Sample Accuracy = 0.9912281

Fold 8 Out of Sample Accuracy = 1

Fold 9 Out of Sample Accuracy = 0.9906542

Fold 10 Out of Sample Accuracy = 1

[[1]]

[1] 1.0000000 0.9805825 1.0000000 0.9911504 1.0000000 1.0000000

[7] 0.9912281 1.0000000 0.9906542 1.0000000

$meanAccuracy

[1] 0.9953615

> cross\_validate(container2,N,"RF")

Fold 1 Out of Sample Accuracy = 0.8018018

Fold 2 Out of Sample Accuracy = 0.7857143

Fold 3 Out of Sample Accuracy = 0.8018868

Fold 4 Out of Sample Accuracy = 0.8205128

Fold 5 Out of Sample Accuracy = 0.7933884

Fold 6 Out of Sample Accuracy = 0.7657658

Fold 7 Out of Sample Accuracy = 0.7582418

Fold 8 Out of Sample Accuracy = 0.8

Fold 9 Out of Sample Accuracy = 0.7946429

Fold 10 Out of Sample Accuracy = 0.7982456

[[1]]

[1] 0.8018018 0.7857143 0.8018868 0.8205128 0.7933884 0.7657658

[7] 0.7582418 0.8000000 0.7946429 0.7982456

$meanAccuracy

[1] 0.79202

Spec = ntc

<<DocumentTermMatrix (documents: 1110, terms: 1879)>>

Non-/sparse entries: 6090/2079600

Sparsity : 100%

Maximal term length: 17

Weighting : SMART ntc (SMART)

$SVM

Call:

svm.default(x = container@training\_matrix, y = container@training\_codes, kernel = kernel, cost = cost,

cross = cross, probability = TRUE, method = method)

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 100

gamma: 0.000532198

Number of Support Vectors: 723

$TREE

node), split, n, deviance, yval, (yprob)

\* denotes terminal node

1) root 1000 1386.000 2 ( 0.49900 0.50100 )

2) X713 < 0.437331 936 1293.000 1 ( 0.53312 0.46688 )

4) X699 < 1.02047 859 1178.000 1 ( 0.56112 0.43888 )

8) X415 < 0.714098 838 1143.000 1 ( 0.57518 0.42482 )

16) X29 < 0.816478 817 1107.000 1 ( 0.58874 0.41126 )

32) X1091 < 1.43491 796 1070.000 1 ( 0.60176 0.39824 )

64) X589 < 0.776182 784 1048.000 1 ( 0.61097 0.38903 )

128) X657 < 2.13312 772 1025.000 1 ( 0.62047 0.37953 )

256) X86 < 1.19229 763 1007.000 1 ( 0.62779 0.37221 )

512) X1363 < 2.31655 754 989.400 1 ( 0.63528 0.36472 )

1024) X958 < 2.30691 746 973.100 1 ( 0.64209 0.35791 )

2048) X959 < 1.1324 738 956.500 1 ( 0.64905 0.35095 )

4096) X1196 < 0.984643 730 939.600 1 ( 0.65616 0.34384 )

8192) X138 < 1.55327 707 897.700 1 ( 0.66902 0.33098 )

16384) X557 < 0.753007 700 882.100 1 ( 0.67571 0.32429 )

32768) X753 < 1.258 693 866.200 1 ( 0.68254 0.31746 ) \*

32769) X753 > 1.258 7 0.000 2 ( 0.00000 1.00000 ) \*

16385) X557 > 0.753007 7 0.000 2 ( 0.00000 1.00000 ) \*

8193) X138 > 1.55327 23 26.400 2 ( 0.26087 0.73913 ) \*

4097) X1196 > 0.984643 8 0.000 2 ( 0.00000 1.00000 ) \*

2049) X959 > 1.1324 8 0.000 2 ( 0.00000 1.00000 ) \*

1025) X958 > 2.30691 8 0.000 2 ( 0.00000 1.00000 ) \*

513) X1363 > 2.31655 9 0.000 2 ( 0.00000 1.00000 ) \*

257) X86 > 1.19229 9 0.000 2 ( 0.00000 1.00000 ) \*

129) X657 > 2.13312 12 0.000 2 ( 0.00000 1.00000 ) \*

65) X589 > 0.776182 12 0.000 2 ( 0.00000 1.00000 ) \*

33) X1091 > 1.43491 21 13.210 2 ( 0.09524 0.90476 ) \*

17) X29 > 0.816478 21 8.041 2 ( 0.04762 0.95238 ) \*

9) X415 > 0.714098 21 0.000 2 ( 0.00000 1.00000 ) \*

5) X699 > 1.02047 77 81.300 2 ( 0.22078 0.77922 ) \*

3) X713 > 0.437331 64 0.000 2 ( 0.00000 1.00000 ) \*

$RF

Call:

randomForest(x = as.matrix(container@training\_matrix), y = container@training\_codes, ntree = ntree)

Type of random forest: classification

Number of trees: 200

No. of variables tried at each split: 43

OOB estimate of error rate: 24.1%

Confusion matrix:

1 2 class.error

1 417 82 0.1643287

2 159 342 0.3173653

> table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results1[,"TREE\_LABEL"])

1 2

1 47 3

2 18 42

> table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results1[,"SVM\_LABEL"])

1 2

1 50 0

2 0 60

> table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results1[,"MAXENTROPY\_LABEL"])

1 2

1 50 0

2 0 60

> table(as.numeric(as.factor(df1[1001:(1000+(length(p1))), 3])), results1[,"FORESTS\_LABEL"])

1 2

1 48 2

2 1 59

Recall accuracy:

Tree:80.90909

Svm:100

Max:100

Rf:97.27273

|  |
| --- |
| ENSEMBLE SUMMARY  n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL  n >= 1 1.00 0.99  n >= 2 1.00 0.99  n >= 3 0.97 1.00  n >= 4 0.81 1.00  ALGORITHM PERFORMANCE  SVM\_PRECISION SVM\_RECALL SVM\_FSCORE FORESTS\_PRECISION FORESTS\_RECALL  1.000 1.000 1.000 0.975 0.970 FORESTS\_FSCORE TREE\_PRECISION TREE\_RECALL TREE\_FSCORE MAXENTROPY\_PRECISION  0.970 0.825 0.820 0.810 1.000  MAXENTROPY\_RECALL MAXENTROPY\_FSCORE  1.000 1.000 |
|  |
| |  | | --- | |  | |

Cross validate;

cross\_validate(container1,N,"TREE")

Fold 1 Out of Sample Accuracy = 0.7232143

Fold 2 Out of Sample Accuracy = 0.6752137

Fold 3 Out of Sample Accuracy = 0.7603306

Fold 4 Out of Sample Accuracy = 0.75

Fold 5 Out of Sample Accuracy = 0.7475728

Fold 6 Out of Sample Accuracy = 0.7878788

Fold 7 Out of Sample Accuracy = 0.7578947

Fold 8 Out of Sample Accuracy = 0.6981132

Fold 9 Out of Sample Accuracy = 0.5614035

Fold 10 Out of Sample Accuracy = 0.7086614

[[1]]

[1] 0.7232143 0.6752137 0.7603306 0.7500000 0.7475728 0.7878788 0.7578947 0.6981132 0.5614035 0.7086614

$meanAccuracy

[1] 0.7170283

> cross\_validate(container1,N,"SVM")

Fold 1 Out of Sample Accuracy = 0.7943925

Fold 2 Out of Sample Accuracy = 0.8536585

Fold 3 Out of Sample Accuracy = 0.8067227

Fold 4 Out of Sample Accuracy = 0.8425926

Fold 5 Out of Sample Accuracy = 0.8269231

Fold 6 Out of Sample Accuracy = 0.8064516

Fold 7 Out of Sample Accuracy = 0.862069

Fold 8 Out of Sample Accuracy = 0.7755102

Fold 9 Out of Sample Accuracy = 0.7943925

Fold 10 Out of Sample Accuracy = 0.7980769

[[1]]

[1] 0.7943925 0.8536585 0.8067227 0.8425926 0.8269231 0.8064516 0.8620690 0.7755102 0.7943925 0.7980769

$meanAccuracy

[1] 0.816079

> cross\_validate(container1,N,"MAXENT")

Fold 1 Out of Sample Accuracy = 1

Fold 2 Out of Sample Accuracy = 0.9888889

Fold 3 Out of Sample Accuracy = 1

Fold 4 Out of Sample Accuracy = 1

Fold 5 Out of Sample Accuracy = 1

Fold 6 Out of Sample Accuracy = 0.9891304

Fold 7 Out of Sample Accuracy = 1

Fold 8 Out of Sample Accuracy = 0.9918033

Fold 9 Out of Sample Accuracy = 1

Fold 10 Out of Sample Accuracy = 0.9793814

[[1]]

[1] 1.0000000 0.9888889 1.0000000 1.0000000 1.0000000 0.9891304 1.0000000 0.9918033 1.0000000 0.9793814

$meanAccuracy

[1] 0.9949204

> cross\_validate(container1,N,"RF")

Fold 1 Out of Sample Accuracy = 0.8

Fold 2 Out of Sample Accuracy = 0.8106061

Fold 3 Out of Sample Accuracy = 0.8

Fold 4 Out of Sample Accuracy = 0.8495575

Fold 5 Out of Sample Accuracy = 0.77

Fold 6 Out of Sample Accuracy = 0.7758621

Fold 7 Out of Sample Accuracy = 0.7478261

Fold 8 Out of Sample Accuracy = 0.7387387

Fold 9 Out of Sample Accuracy = 0.7798165

Fold 10 Out of Sample Accuracy = 0.8269231

[[1]]

[1] 0.8000000 0.8106061 0.8000000 0.8495575 0.7700000 0.7758621 0.7478261 0.7387387 0.7798165 0.8269231

$meanAccuracy

[1] 0.789933

Spec = atc

$SVM

Call:

svm.default(x = container@training\_matrix, y = container@training\_codes, kernel = kernel, cost = cost,

cross = cross, probability = TRUE, method = method)

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 100

gamma: 0.000532198

Number of Support Vectors: 722

$TREE

node), split, n, deviance, yval, (yprob)

\* denotes terminal node

1) root 1000 1386.000 2 ( 0.49900 0.50100 )

2) X713 < 0.465393 936 1293.000 1 ( 0.53312 0.46688 )

4) X699 < 1.00296 858 1176.000 1 ( 0.56177 0.43823 )

8) X415 < 0.714098 837 1141.000 1 ( 0.57587 0.42413 )

16) X29 < 0.816478 816 1105.000 1 ( 0.58946 0.41054 )

32) X1091 < 1.43491 794 1067.000 1 ( 0.60327 0.39673 )

64) X589 < 0.836438 782 1044.000 1 ( 0.61253 0.38747 )

128) X657 < 2.13312 770 1021.000 1 ( 0.62208 0.37792 )

256) X86 < 1.19229 761 1003.000 1 ( 0.62943 0.37057 )

512) X557 < 0.807642 753 987.400 1 ( 0.63612 0.36388 )

1024) X958 < 2.30691 745 971.000 1 ( 0.64295 0.35705 )

2048) X959 < 1.1324 737 954.400 1 ( 0.64993 0.35007 )

4096) X1196 < 1.08856 729 937.400 1 ( 0.65706 0.34294 )

8192) X138 < 1.55327 706 895.500 1 ( 0.66997 0.33003 )

16384) X753 < 1.258 699 879.800 1 ( 0.67668 0.32332 ) \*

16385) X753 > 1.258 7 0.000 2 ( 0.00000 1.00000 ) \*

8193) X138 > 1.55327 23 26.400 2 ( 0.26087 0.73913 ) \*

4097) X1196 > 1.08856 8 0.000 2 ( 0.00000 1.00000 ) \*

2049) X959 > 1.1324 8 0.000 2 ( 0.00000 1.00000 ) \*

1025) X958 > 2.30691 8 0.000 2 ( 0.00000 1.00000 ) \*

513) X557 > 0.807642 8 0.000 2 ( 0.00000 1.00000 ) \*

257) X86 > 1.19229 9 0.000 2 ( 0.00000 1.00000 ) \*

129) X657 > 2.13312 12 0.000 2 ( 0.00000 1.00000 ) \*

65) X589 > 0.836438 12 0.000 2 ( 0.00000 1.00000 ) \*

33) X1091 > 1.43491 22 13.400 2 ( 0.09091 0.90909 ) \*

17) X29 > 0.816478 21 8.041 2 ( 0.04762 0.95238 ) \*

9) X415 > 0.714098 21 0.000 2 ( 0.00000 1.00000 ) \*

5) X699 > 1.00296 78 81.790 2 ( 0.21795 0.78205 ) \*

3) X713 > 0.465393 64 0.000 2 ( 0.00000 1.00000 ) \*

$RF

Call:

randomForest(x = as.matrix(container@training\_matrix), y = container@training\_codes, ntree = ntree)

Type of random forest: classification

Number of trees: 200

No. of variables tried at each split: 43

OOB estimate of error rate: 24.2%

Confusion matrix:

1 2 class.error

1 422 77 0.1543086

2 165 336 0.3293413

Recall accuracy:

Same above

ENSEMBLE SUMMARY

n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL

n >= 1 1.00 0.99

n >= 2 1.00 0.99

n >= 3 0.97 1.00

n >= 4 0.81 1.00

ALGORITHM PERFORMANCE

SVM\_PRECISION SVM\_RECALL SVM\_FSCORE FORESTS\_PRECISION FORESTS\_RECALL

1.000 1.000 1.000 0.975 0.970

FORESTS\_FSCORE TREE\_PRECISION TREE\_RECALL TREE\_FSCORE MAXENTROPY\_PRECISION

0.970 0.825 0.820 0.810 1.000

MAXENTROPY\_RECALL MAXENTROPY\_FSCORE

1.000 1.000

> cross\_validate(container1,N,"TREE")

Fold 1 Out of Sample Accuracy = 0.65625

Fold 2 Out of Sample Accuracy = 0.7207207

Fold 3 Out of Sample Accuracy = 0.77

Fold 4 Out of Sample Accuracy = 0.789916

Fold 5 Out of Sample Accuracy = 0.7247706

Fold 6 Out of Sample Accuracy = 0.6747967

Fold 7 Out of Sample Accuracy = 0.6942149

Fold 8 Out of Sample Accuracy = 0.7272727

Fold 9 Out of Sample Accuracy = 0.6929825

Fold 10 Out of Sample Accuracy = 0.7118644

[[1]]

[1] 0.6562500 0.7207207 0.7700000 0.7899160 0.7247706 0.6747967 0.6942149 0.7272727 0.6929825 0.7118644

$meanAccuracy

[1] 0.7162789

> cross\_validate(container1,N,"SVM")

Fold 1 Out of Sample Accuracy = 0.7475728

Fold 2 Out of Sample Accuracy = 0.848

Fold 3 Out of Sample Accuracy = 0.8035714

Fold 4 Out of Sample Accuracy = 0.8230088

Fold 5 Out of Sample Accuracy = 0.8240741

Fold 6 Out of Sample Accuracy = 0.7981651

Fold 7 Out of Sample Accuracy = 0.8468468

Fold 8 Out of Sample Accuracy = 0.8534483

Fold 9 Out of Sample Accuracy = 0.7913043

Fold 10 Out of Sample Accuracy = 0.7346939

[[1]]

[1] 0.7475728 0.8480000 0.8035714 0.8230088 0.8240741 0.7981651 0.8468468 0.8534483 0.7913043 0.7346939

$meanAccuracy

[1] 0.8070686

> cross\_validate(container1,N,"MAXENT")

Fold 1 Out of Sample Accuracy = 1

Fold 2 Out of Sample Accuracy = 0.990566

Fold 3 Out of Sample Accuracy = 0.9915254

Fold 4 Out of Sample Accuracy = 1

Fold 5 Out of Sample Accuracy = 0.9813084

Fold 6 Out of Sample Accuracy = 1

Fold 7 Out of Sample Accuracy = 1

Fold 8 Out of Sample Accuracy = 0.9917355

Fold 9 Out of Sample Accuracy = 1

Fold 10 Out of Sample Accuracy = 1

[[1]]

[1] 1.0000000 0.9905660 0.9915254 1.0000000 0.9813084 1.0000000 1.0000000 0.9917355 1.0000000 1.0000000

$meanAccuracy

[1] 0.9955135

> cross\_validate(container1,N,"RF")

Fold 1 Out of Sample Accuracy = 0.7719298

Fold 2 Out of Sample Accuracy = 0.745098

Fold 3 Out of Sample Accuracy = 0.8053097

Fold 4 Out of Sample Accuracy = 0.8080808

Fold 5 Out of Sample Accuracy = 0.7777778

Fold 6 Out of Sample Accuracy = 0.7981651

Fold 7 Out of Sample Accuracy = 0.8495575

Fold 8 Out of Sample Accuracy = 0.7826087

Fold 9 Out of Sample Accuracy = 0.8360656

Fold 10 Out of Sample Accuracy = 0.7264151

[[1]]

[1] 0.7719298 0.7450980 0.8053097 0.8080808 0.7777778 0.7981651 0.8495575 0.7826087 0.8360656 0.7264151

$meanAccuracy

[1] 0.7901008

Spec = btc

SVM

Call:

svm.default(x = container@training\_matrix, y = container@training\_codes, kernel = kernel, cost = cost,

cross = cross, probability = TRUE, method = method)

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 100

gamma: 0.000532198

Number of Support Vectors: 722

$TREE

node), split, n, deviance, yval, (yprob)

\* denotes terminal node

1) root 1000 1386.000 2 ( 0.49900 0.50100 )

2) X713 < 0.47657 936 1293.000 1 ( 0.53312 0.46688 )

4) X699 < 1.02047 856 1174.000 1 ( 0.56192 0.43808 )

8) X415 < 0.714098 835 1138.000 1 ( 0.57605 0.42395 )

16) X29 < 0.816478 814 1102.000 1 ( 0.58968 0.41032 )

32) X1091 < 1.43491 792 1064.000 1 ( 0.60354 0.39646 )

64) X589 < 0.861096 780 1041.000 1 ( 0.61282 0.38718 )

128) X657 < 2.13312 768 1018.000 1 ( 0.62240 0.37760 )

256) X86 < 1.19229 759 1000.000 1 ( 0.62978 0.37022 )

512) X557 < 0.829773 751 984.400 1 ( 0.63648 0.36352 )

1024) X958 < 2.30691 743 968.100 1 ( 0.64334 0.35666 )

2048) X959 < 1.1324 735 951.400 1 ( 0.65034 0.34966 )

4096) X1196 < 1.08856 727 934.500 1 ( 0.65750 0.34250 )

8192) X753 < 1.258 720 919.300 1 ( 0.66389 0.33611 )

16384) X138 < 1.55327 698 879.000 1 ( 0.67622 0.32378 ) \*

16385) X138 > 1.55327 22 25.780 2 ( 0.27273 0.72727 ) \*

8193) X753 > 1.258 7 0.000 2 ( 0.00000 1.00000 ) \*

4097) X1196 > 1.08856 8 0.000 2 ( 0.00000 1.00000 ) \*

2049) X959 > 1.1324 8 0.000 2 ( 0.00000 1.00000 ) \*

1025) X958 > 2.30691 8 0.000 2 ( 0.00000 1.00000 ) \*

513) X557 > 0.829773 8 0.000 2 ( 0.00000 1.00000 ) \*

257) X86 > 1.19229 9 0.000 2 ( 0.00000 1.00000 ) \*

129) X657 > 2.13312 12 0.000 2 ( 0.00000 1.00000 ) \*

65) X589 > 0.861096 12 0.000 2 ( 0.00000 1.00000 ) \*

33) X1091 > 1.43491 22 13.400 2 ( 0.09091 0.90909 ) \*

17) X29 > 0.816478 21 8.041 2 ( 0.04762 0.95238 ) \*

9) X415 > 0.714098 21 0.000 2 ( 0.00000 1.00000 ) \*

5) X699 > 1.02047 80 85.310 2 ( 0.22500 0.77500 ) \*

3) X713 > 0.47657 64 0.000 2 ( 0.00000 1.00000 ) \*

$RF

Call:

randomForest(x = as.matrix(container@training\_matrix), y = container@training\_codes, ntree = ntree)

Type of random forest: classification

Number of trees: 200

No. of variables tried at each split: 43

OOB estimate of error rate: 23.1%

Confusion matrix:

1 2 class.error

1 430 69 0.1382766

2 162 339 0.3233533

Accuracy = same

ENSEMBLE SUMMARY

n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL

n >= 1 1.00 0.99

n >= 2 1.00 0.99

n >= 3 0.97 1.00

n >= 4 0.81 1.00

ALGORITHM PERFORMANCE

SVM\_PRECISION SVM\_RECALL SVM\_FSCORE FORESTS\_PRECISION FORESTS\_RECALL

1.000 1.000 1.000 0.975 0.970

FORESTS\_FSCORE TREE\_PRECISION TREE\_RECALL TREE\_FSCORE MAXENTROPY\_PRECISION

0.970 0.825 0.820 0.810 1.000

MAXENTROPY\_RECALL MAXENTROPY\_FSCORE

1.000 1.000

> N=10

> cross\_validate(container1,N,"TREE")

Fold 1 Out of Sample Accuracy = 0.6730769

Fold 2 Out of Sample Accuracy = 0.75

Fold 3 Out of Sample Accuracy = 0.7631579

Fold 4 Out of Sample Accuracy = 0.7735849

Fold 5 Out of Sample Accuracy = 0.7

Fold 6 Out of Sample Accuracy = 0.6521739

Fold 7 Out of Sample Accuracy = 0.7719298

Fold 8 Out of Sample Accuracy = 0.6486486

Fold 9 Out of Sample Accuracy = 0.6803279

Fold 10 Out of Sample Accuracy = 0.77

[[1]]

[1] 0.6730769 0.7500000 0.7631579 0.7735849 0.7000000 0.6521739 0.7719298 0.6486486 0.6803279 0.7700000

$meanAccuracy

[1] 0.71829

> cross\_validate(container1,N,"SVM")

Fold 1 Out of Sample Accuracy = 0.8229167

Fold 2 Out of Sample Accuracy = 0.8039216

Fold 3 Out of Sample Accuracy = 0.745098

Fold 4 Out of Sample Accuracy = 0.8032787

Fold 5 Out of Sample Accuracy = 0.8152174

Fold 6 Out of Sample Accuracy = 0.7830189

Fold 7 Out of Sample Accuracy = 0.8050847

Fold 8 Out of Sample Accuracy = 0.8106061

Fold 9 Out of Sample Accuracy = 0.8306452

Fold 10 Out of Sample Accuracy = 0.8448276

[[1]]

[1] 0.8229167 0.8039216 0.7450980 0.8032787 0.8152174 0.7830189 0.8050847 0.8106061 0.8306452 0.8448276

$meanAccuracy

[1] 0.8064615

> cross\_validate(container1,N,"MAXENT")

Fold 1 Out of Sample Accuracy = 0.9902913

Fold 2 Out of Sample Accuracy = 1

Fold 3 Out of Sample Accuracy = 1

Fold 4 Out of Sample Accuracy = 0.978022

Fold 5 Out of Sample Accuracy = 1

Fold 6 Out of Sample Accuracy = 1

Fold 7 Out of Sample Accuracy = 1

Fold 8 Out of Sample Accuracy = 0.9902913

Fold 9 Out of Sample Accuracy = 0.9915254

Fold 10 Out of Sample Accuracy = 1

[[1]]

[1] 0.9902913 1.0000000 1.0000000 0.9780220 1.0000000 1.0000000 1.0000000 0.9902913 0.9915254 1.0000000

$meanAccuracy

[1] 0.995013

> cross\_validate(container1,N,"RF")

Fold 1 Out of Sample Accuracy = 0.8037383

Fold 2 Out of Sample Accuracy = 0.862069

Fold 3 Out of Sample Accuracy = 0.847619

Fold 4 Out of Sample Accuracy = 0.779661

Fold 5 Out of Sample Accuracy = 0.6525424

Fold 6 Out of Sample Accuracy = 0.7634409

Fold 7 Out of Sample Accuracy = 0.796875

Fold 8 Out of Sample Accuracy = 0.8347826

Fold 9 Out of Sample Accuracy = 0.7821782

Fold 10 Out of Sample Accuracy = 0.8073394

[[1]]

[1] 0.8037383 0.8620690 0.8476190 0.7796610 0.6525424 0.7634409 0.7968750 0.8347826 0.7821782 0.8073394

$meanAccuracy

[1] 0.7930246

Spec = Ltc

SVM

Call:

svm.default(x = container@training\_matrix, y = container@training\_codes, kernel = kernel, cost = cost,

cross = cross, probability = TRUE, method = method)

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 100

gamma: 0.000532198

Number of Support Vectors: 723

$TREE

node), split, n, deviance, yval, (yprob)

\* denotes terminal node

1) root 1000 1386.000 2 ( 0.49900 0.50100 )

2) X713 < 0.437331 936 1293.000 1 ( 0.53312 0.46688 )

4) X699 < 1.02047 859 1178.000 1 ( 0.56112 0.43888 )

8) X415 < 0.714098 838 1143.000 1 ( 0.57518 0.42482 )

16) X29 < 0.816478 817 1107.000 1 ( 0.58874 0.41126 )

32) X1091 < 1.43491 796 1070.000 1 ( 0.60176 0.39824 )

64) X589 < 0.776182 784 1048.000 1 ( 0.61097 0.38903 )

128) X657 < 2.13312 772 1025.000 1 ( 0.62047 0.37953 )

256) X86 < 1.19229 763 1007.000 1 ( 0.62779 0.37221 )

512) X1363 < 2.31655 754 989.400 1 ( 0.63528 0.36472 )

1024) X958 < 2.30691 746 973.100 1 ( 0.64209 0.35791 )

2048) X959 < 1.1324 738 956.500 1 ( 0.64905 0.35095 )

4096) X1196 < 0.984643 730 939.600 1 ( 0.65616 0.34384 )

8192) X138 < 1.55327 707 897.700 1 ( 0.66902 0.33098 16384) X557 < 0.753007 700 882.100 1 ( 0.67571 0.32429 )

32768) X753 < 1.258 693 866.200 1 ( 0.68254 0.31746 ) \*

32769) X753 > 1.258 7 0.000 2 ( 0.00000 1.00000 ) \*

16385) X557 > 0.753007 7 0.000 2 ( 0.00000 1.00000 ) \*

8193) X138 > 1.55327 23 26.400 2 ( 0.26087 0.73913 ) \*

4097) X1196 > 0.984643 8 0.000 2 ( 0.00000 1.00000 ) \*

2049) X959 > 1.1324 8 0.000 2 ( 0.00000 1.00000 ) \*

1025) X958 > 2.30691 8 0.000 2 ( 0.00000 1.00000 ) \*

513) X1363 > 2.31655 9 0.000 2 ( 0.00000 1.00000 ) \*

257) X86 > 1.19229 9 0.000 2 ( 0.00000 1.00000 ) \*

129) X657 > 2.13312 12 0.000 2 ( 0.00000 1.00000 ) \*

65) X589 > 0.776182 12 0.000 2 ( 0.00000 1.00000 ) \*

33) X1091 > 1.43491 21 13.210 2 ( 0.09524 0.90476 ) \*

17) X29 > 0.816478 21 8.041 2 ( 0.04762 0.95238 ) \*

9) X415 > 0.714098 21 0.000 2 ( 0.00000 1.00000 ) \*

5) X699 > 1.02047 77 81.300 2 ( 0.22078 0.77922 ) \*

3) X713 > 0.437331 64 0.000 2 ( 0.00000 1.00000 ) \*

$RF

Call:

randomForest(x = as.matrix(container@training\_matrix), y = container@training\_codes, ntree = ntree)

Type of random forest: classification

Number of trees: 200

No. of variables tried at each split: 43

OOB estimate of error rate: 24%

Confusion matrix:

1 2 class.error

1 424 75 0.1503006

2 165 336 0.3293413

Recall accuracy = same as above

ENSEMBLE SUMMARY

n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL

n >= 1 1.00 0.99

n >= 2 1.00 0.99

n >= 3 0.97 1.00

n >= 4 0.81 1.00

ALGORITHM PERFORMANCE

SVM\_PRECISION SVM\_RECALL SVM\_FSCORE FORESTS\_PRECISION FORESTS\_RECALL

1.000 1.000 1.000 0.975 0.970

FORESTS\_FSCORE TREE\_PRECISION TREE\_RECALL TREE\_FSCORE MAXENTROPY\_PRECISION

0.970 0.825 0.820 0.810 1.000

MAXENTROPY\_RECALL MAXENTROPY\_FSCORE

1.000 1.000

> N=10

> cross\_validate(container1,N,"TREE")

Fold 1 Out of Sample Accuracy = 0.7096774

Fold 2 Out of Sample Accuracy = 0.7254902

Fold 3 Out of Sample Accuracy = 0.682243

Fold 4 Out of Sample Accuracy = 0.7142857

Fold 5 Out of Sample Accuracy = 0.6949153

Fold 6 Out of Sample Accuracy = 0.7169811

Fold 7 Out of Sample Accuracy = 0.7589286

Fold 8 Out of Sample Accuracy = 0.7863248

Fold 9 Out of Sample Accuracy = 0.6811594

Fold 10 Out of Sample Accuracy = 0.7410714

[[1]]

[1] 0.7096774 0.7254902 0.6822430 0.7142857 0.6949153 0.7169811 0.7589286 0.7863248 0.6811594 0.7410714

$meanAccuracy

[1] 0.7211077

> cross\_validate(container1,N,"SVM")

Fold 1 Out of Sample Accuracy = 0.7863248

Fold 2 Out of Sample Accuracy = 0.8053097

Fold 3 Out of Sample Accuracy = 0.8165138

Fold 4 Out of Sample Accuracy = 0.7272727

Fold 5 Out of Sample Accuracy = 0.8659794

Fold 6 Out of Sample Accuracy = 0.7788462

Fold 7 Out of Sample Accuracy = 0.8198198

Fold 8 Out of Sample Accuracy = 0.7619048

Fold 9 Out of Sample Accuracy = 0.7627119

Fold 10 Out of Sample Accuracy = 0.8

[[1]]

[1] 0.7863248 0.8053097 0.8165138 0.7272727 0.8659794 0.7788462 0.8198198 0.7619048 0.7627119 0.8000000

$meanAccuracy

[1] 0.7924683

> cross\_validate(container1,N,"MAXENT")

Fold 1 Out of Sample Accuracy = 1

Fold 2 Out of Sample Accuracy = 0.9896907

Fold 3 Out of Sample Accuracy = 0.9904762

Fold 4 Out of Sample Accuracy = 0.992126

Fold 5 Out of Sample Accuracy = 1

Fold 6 Out of Sample Accuracy = 1

Fold 7 Out of Sample Accuracy = 1

Fold 8 Out of Sample Accuracy = 0.9913043

Fold 9 Out of Sample Accuracy = 1

Fold 10 Out of Sample Accuracy = 0.9908257

[[1]]

[1] 1.0000000 0.9896907 0.9904762 0.9921260 1.0000000 1.0000000 1.0000000 0.9913043 1.0000000 0.9908257

$meanAccuracy

[1] 0.9954423

> cross\_validate(container1,N,"RF")

Fold 1 Out of Sample Accuracy = 0.8125

Fold 2 Out of Sample Accuracy = 0.75

Fold 3 Out of Sample Accuracy = 0.7474747

Fold 4 Out of Sample Accuracy = 0.8058252

Fold 5 Out of Sample Accuracy = 0.7520661

Fold 6 Out of Sample Accuracy = 0.8037383

Fold 7 Out of Sample Accuracy = 0.8030303

Fold 8 Out of Sample Accuracy = 0.8256881

Fold 9 Out of Sample Accuracy = 0.7433628

Fold 10 Out of Sample Accuracy = 0.8813559

[[1]]

[1] 0.8125000 0.7500000 0.7474747 0.8058252 0.7520661 0.8037383 0.8030303 0.8256881 0.7433628 0.8813559

$meanAccuracy

[1] 0.7925042

spec = ltn