# Rf\_relief

RandomForestClassifier(bootstrap=True, class\_weight='balanced',

criterion='entropy', max\_depth=15, max\_features='auto',

max\_leaf\_nodes=None, min\_impurity\_decrease=0.0,

min\_impurity\_split=None, min\_samples\_leaf=1,

min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0,

n\_estimators=20, n\_jobs=1, oob\_score=False, random\_state=2018,

verbose=0, warm\_start=False)

{'class\_weight': 'balanced', 'criterion': 'entropy', 'max\_depth': 15, 'min\_samples\_leaf': 1, 'n\_estimators': 20}

## train

RF model准确度分数: 0.7471022128556375

RF model混淆矩阵

[[ 582 516 128 8]

[ 19 306 3 0]

[ 161 8 416 133]

[ 21 0 203 2241]]

RF model召回率 0.7232696894857129

RF model分类报告

precision recall f1-score support

MCI 0.74 0.47 0.58 1234

Normal 0.37 0.93 0.53 328

VMD 0.55 0.58 0.57 718

dementia 0.94 0.91 0.92 2465

avg / total 0.79 0.75 0.75 4745

## test

RF model准确度分数: 0.7817836812144212

RF model混淆矩阵

[[ 57 44 12 0]

[ 5 46 0 0]

[ 20 0 63 15]

[ 2 0 17 246]]

RF model召回率 0.7443861481810958

RF model分类报告

precision recall f1-score support

Normal 0.68 0.50 0.58 113

MCI 0.51 0.90 0.65 51

VMD 0.68 0.64 0.66 98

dementia 0.94 0.93 0.94 265

avg / total 0.80 0.78 0.78 527

# rf\_infor

RandomForestClassifier(bootstrap=True, class\_weight='balanced',

criterion='gini', max\_depth=20, max\_features='auto',

max\_leaf\_nodes=None, min\_impurity\_decrease=0.0,

min\_impurity\_split=None, min\_samples\_leaf=1,

min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0,

n\_estimators=20, n\_jobs=1, oob\_score=False, random\_state=2018,

verbose=0, warm\_start=False)

{'class\_weight': 'balanced', 'criterion': 'gini', 'max\_depth': 20, 'min\_samples\_leaf': 1, 'n\_estimators': 20}

## train

RF model准确度分数: 0.7844046364594309

RF model混淆矩阵

[[ 736 371 117 10]

[ 27 299 2 0]

[ 132 6 420 160]

[ 24 0 174 2267]]

RF model召回率 0.7531633498297041

RF model分类报告

precision recall f1-score support

MCI 0.80 0.60 0.68 1234

Normal 0.44 0.91 0.60 328

VMD 0.59 0.58 0.59 718

dementia 0.93 0.92 0.92 2465

avg / total 0.81 0.78 0.79 4745

## test

RF model准确度分数: 0.7836812144212524

RF model混淆矩阵

[[ 64 35 14 0] M

[ 6 45 0 0] N

[ 19 0 53 26] V

[ 3 0 11 251]] D

RF model召回率 0.7341776901109417

RF model分类报告

precision recall f1-score support

MCI 0.70 0.57 0.62 113

Normal 0.56 0.88 0.69 51

VMD 0.68 0.54 0.60 98

dementia 0.91 0.95 0.93 265

avg / total 0.79 0.78 0.78 527

# rf\_rf

RandomForestClassifier(bootstrap=True, class\_weight='balanced\_subsample',

criterion='gini', max\_depth=15, max\_features='auto',

max\_leaf\_nodes=None, min\_impurity\_decrease=0.0,

min\_impurity\_split=None, min\_samples\_leaf=2,

min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0,

n\_estimators=15, n\_jobs=1, oob\_score=False, random\_state=2018,

verbose=0, warm\_start=False)

{'class\_weight': 'balanced\_subsample', 'criterion': 'gini', 'max\_depth': 15, 'min\_samples\_leaf': 2, 'n\_estimators': 15}

## Train

RF model准确度分数: 0.7650158061116965

RF model混淆矩阵

[[ 806 143 270 15]

[ 71 252 5 0]

[ 198 9 417 94]

[ 39 0 271 2155]]

RF model召回率 0.7191181079845135

RF model分类报告

precision recall f1-score support

MCI 0.72 0.65 0.69 1234

Normal 0.62 0.77 0.69 328

VMD 0.43 0.58 0.50 718

dementia 0.95 0.87 0.91 2465

avg / total 0.79 0.77 0.77 4745

## test

RF model准确度分数: 0.7571157495256167

RF model混淆矩阵

[[ 71 12 29 1]

[ 6 43 2 0]

[ 28 1 51 18]

[ 3 0 28 234]]

RF model召回率 0.7187207175406479

RF model分类报告

precision recall f1-score support

MCI 0.66 0.63 0.64 113

Normal 0.77 0.84 0.80 51

VMD 0.46 0.52 0.49 98

dementia 0.92 0.88 0.90 265

avg / total 0.77 0.76 0.76 527

# naïve bayes\_relief

BernoulliNB(alpha=2.0, binarize=0.0, class\_prior=None, fit\_prior=True)

{'alpha': 2.0, 'binarize': 0.0, 'fit\_prior': True}

## Train

naive bayes model准确度分数: 0.7785036880927292

naive bayes model混淆矩阵

[[1110 0 120 4]

[ 327 0 1 0]

[ 169 0 489 60]

[ 17 0 353 2095]]

naive bayes model召回率 0.6076177130701366

naive bayes model分类报告

precision recall f1-score support

MCI 0.68 0.90 0.78 1234

Normal 0.00 0.00 0.00 328

VMD 0.51 0.68 0.58 718

dementia 0.97 0.85 0.91 2465

avg / total 0.76 0.78 0.76 4745

## test

naive bayes model准确度分数: 0.7855787476280834

naive bayes model混淆矩阵

[[105 0 8 0]

[ 51 0 0 0]

[ 19 0 69 10]

[ 1 0 24 240]]

naive bayes model召回率 0.6347363874586402

naive bayes model分类报告

precision recall f1-score support

MCI 0.60 0.93 0.73 113

Normal 0.00 0.00 0.00 51

VMD 0.68 0.70 0.69 98

dementia 0.96 0.91 0.93 265

avg / total 0.74 0.79 0.75 527

# naïve bayes \_infor

BernoulliNB(alpha=3.0, binarize=0.0, class\_prior=None, fit\_prior=True)

{'alpha': 3.0, 'binarize': 0.0, 'fit\_prior': True}

## Train

naive bayes model准确度分数: 0.7766069546891464

naive bayes model混淆矩阵

[[ 771 343 117 3]

[ 40 287 1 0]

[ 163 2 496 57]

[ 14 0 320 2131]]

naive bayes model召回率 0.7637770622115942

naive bayes model分类报告

precision recall f1-score support

MCI 0.78 0.62 0.69 1234

Normal 0.45 0.88 0.60 328

VMD 0.53 0.69 0.60 718

dementia 0.97 0.86 0.92 2465

avg / total 0.82 0.78 0.79 4745

## Test

naive bayes model准确度分数: 0.8102466793168881

naive bayes model混淆矩阵

[[ 70 34 9 0]

[ 8 43 0 0]

[ 14 0 71 13]

[ 1 0 21 243]]

naive bayes model召回率 0.776019302361118

naive bayes model分类报告

precision recall f1-score support

MCI 0.75 0.62 0.68 113

Normal 0.56 0.84 0.67 51

VMD 0.70 0.72 0.71 98

dementia 0.95 0.92 0.93 265

avg / total 0.82 0.81 0.81 527

# naïve bayes \_rf

BernoulliNB(alpha=1.0, binarize=0.0, class\_prior=None, fit\_prior=True)

{'alpha': 1.0, 'binarize': 0.0, 'fit\_prior': True}

## Train

naive bayes model准确度分数: 0.7471022128556375

naive bayes model混淆矩阵

[[ 920 147 155 12]

[ 72 254 2 0]

[ 285 11 384 38]

[ 46 0 432 1987]]

naive bayes model召回率 0.7152093319653186

naive bayes model分类报告

precision recall f1-score support

MCI 0.70 0.75 0.72 1234

Normal 0.62 0.77 0.69 328

VMD 0.39 0.53 0.45 718

dementia 0.98 0.81 0.88 2465

avg / total 0.79 0.75 0.76 4745

## test

naive bayes model准确度分数: 0.8102466793168881

naive bayes model混淆矩阵

[[ 70 34 9 0]

[ 8 43 0 0]

[ 14 0 71 13]

[ 1 0 21 243]]

naive bayes model召回率 0.776019302361118

naive bayes model分类报告

precision recall f1-score support

MCI 0.75 0.62 0.68 113

Normal 0.56 0.84 0.67 51

VMD 0.70 0.72 0.71 98

dementia 0.95 0.92 0.93 265

avg / total 0.82 0.81 0.81 527

# svm\_relief

SVC(C=100, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape='ovr', degree=3, gamma=0.001, kernel='rbf',

max\_iter=-1, probability=True, random\_state=None, shrinking=True,

tol=0.001, verbose=False)

{'C': 100, 'gamma': 0.001}

## Train

SVM model准确度分数: 0.7968387776606954

SVM model混淆矩阵

[[1126 0 99 9]

[ 327 0 1 0]

[ 184 0 356 178]

[ 20 0 146 2299]]

SVM model召回率 0.5852396671278927

SVm model分类报告

precision recall f1-score support

MCI 0.68 0.91 0.78 1234

Normal 0.00 0.00 0.00 328

VMD 0.59 0.50 0.54 718

dementia 0.92 0.93 0.93 2465

avg / total 0.75 0.80 0.77 4745

## test

SVM model准确度分数: 0.7969639468690702

SVM model混淆矩阵

[[108 0 4 1]

[ 51 0 0 0]

[ 19 0 55 24]

[ 2 0 6 257]]

SVM model召回率 0.621697005735004

SVM model分类报告

precision recall f1-score support

MCI 0.60 0.96 0.74 113

Normal 0.00 0.00 0.00 51

VMD 0.85 0.56 0.67 98

dementia 0.91 0.97 0.94 265

avg / total 0.74 0.80 0.76 527

# svm\_infor

SVC(C=1000, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape='ovr', degree=3, gamma=0.001, kernel='rbf',

max\_iter=-1, probability=True, random\_state=None, shrinking=True,

tol=0.001, verbose=False)

{'C': 1000, 'gamma': 0.001}

## Train

SVM model准确度分数: 0.8120126448893572

SVM model混淆矩阵

[[1134 1 93 6]

[ 327 0 1 0]

[ 153 0 391 174]

[ 19 0 118 2328]]

SVM model召回率 0.6019882186678931

SVm model分类报告

precision recall f1-score support

MCI 0.69 0.92 0.79 1234

Normal 0.00 0.00 0.00 328

VMD 0.65 0.54 0.59 718

dementia 0.93 0.94 0.94 2465

avg / total 0.76 0.81 0.78 4745

## test

SVM model准确度分数: 0.7912713472485768

SVM model混淆矩阵

[[104 0 9 0]

[ 51 0 0 0]

[ 15 0 55 28]

[ 2 0 5 258]]

SVM model召回率 0.6137908444392952

SVM model分类报告

precision recall f1-score support

MCI 0.60 0.92 0.73 113

Normal 0.00 0.00 0.00 51

VMD 0.80 0.56 0.66 98

dementia 0.90 0.97 0.94 265

avg / total 0.73 0.79 0.75 527

# svm\_rf

SVC(C=100, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape='ovr', degree=3, gamma=0.001, kernel='rbf',

max\_iter=-1, probability=True, random\_state=None, shrinking=True,

tol=0.001, verbose=False)

{'C': 100, 'gamma': 0.001}

## Train

SVM model准确度分数: 0.7888303477344574

SVM model混淆矩阵

[[ 935 141 120 38]

[ 73 252 3 0]

[ 274 12 249 183]

[ 68 0 90 2307]]

SVM model召回率 0.702172629638573

SVm model分类报告

precision recall f1-score support

MCI 0.69 0.76 0.72 1234

Normal 0.62 0.77 0.69 328

VMD 0.54 0.35 0.42 718

dementia 0.91 0.94 0.92 2465

avg / total 0.78 0.79 0.78 4745

## Test

SVM model准确度分数: 0.7628083491461101

SVM model混淆矩阵

[[ 79 13 15 6]

[ 7 43 1 0]

[ 34 1 31 32]

[ 5 0 11 249]]

SVM model召回率 0.6995503678178568

SVM model分类报告

precision recall f1-score support

MCI 0.63 0.70 0.66 113

Normal 0.75 0.84 0.80 51

VMD 0.53 0.32 0.40 98

dementia 0.87 0.94 0.90 265

avg / total 0.74 0.76 0.75 527

# nn\_relief

MLPClassifier(activation='tanh', alpha=0.0001, batch\_size='auto', beta\_1=0.9,

beta\_2=0.999, early\_stopping=False, epsilon=1e-08,

hidden\_layer\_sizes=3, learning\_rate='constant',

learning\_rate\_init=0.001, max\_iter=200, momentum=0.9,

nesterovs\_momentum=True, power\_t=0.5, random\_state=2018,

shuffle=True, solver='lbfgs', tol=0.0001, validation\_fraction=0.1,

verbose=False, warm\_start=False)

{'activation': 'tanh', 'hidden\_layer\_sizes': 3, 'learning\_rate': 'constant', 'solver': 'lbfgs'}

## Train

mlp model准确度分数: 0.7993677555321391

mlp model混淆矩阵

[[1139 2 87 6]

[ 326 0 2 0]

[ 177 1 371 169]

[ 24 0 158 2283]]

mlp model召回率 0.5914735018080745

mlp model分类报告

precision recall f1-score support

MCI 0.68 0.92 0.79 1234

Normal 0.00 0.00 0.00 328

VMD 0.60 0.52 0.56 718

dementia 0.93 0.93 0.93 2465

avg / total 0.75 0.80 0.77 4745

## test

mlp model准确度分数: 0.8064516129032258

mlp model混淆矩阵

[[108 0 5 0]

[ 51 0 0 0]

[ 18 0 59 21]

[ 1 0 6 258]]

mlp model召回率 0.6328444835940721

mlp model分类报告

precision recall f1-score support

MCI 0.61 0.96 0.74 113

Normal 0.00 0.00 0.00 51

VMD 0.84 0.60 0.70 98

dementia 0.92 0.97 0.95 265

avg / total 0.75 0.81 0.77 527

# nn\_infor

MLPClassifier(activation='logistic', alpha=0.0001, batch\_size='auto',

beta\_1=0.9, beta\_2=0.999, early\_stopping=False, epsilon=1e-08,

hidden\_layer\_sizes=3, learning\_rate='constant',

learning\_rate\_init=0.001, max\_iter=200, momentum=0.9,

nesterovs\_momentum=True, power\_t=0.5, random\_state=2018,

shuffle=True, solver='lbfgs', tol=0.0001, validation\_fraction=0.1,

verbose=False, warm\_start=False)

{'activation': 'logistic', 'hidden\_layer\_sizes': 3, 'learning\_rate': 'constant', 'solver': 'lbfgs'}

## Train

mlp model准确度分数: 0.8061116965226555

mlp model混淆矩阵

[[1146 2 81 5]

[ 327 0 1 0]

[ 149 1 392 176]

[ 22 0 156 2287]]

mlp model召回率 0.6006093113872175

mlp model分类报告

precision recall f1-score support

MCI 0.70 0.93 0.80 1234

Normal 0.00 0.00 0.00 328

VMD 0.62 0.55 0.58 718

dementia 0.93 0.93 0.93 2465

avg / total 0.76 0.81 0.78 4745

## test

mlp model准确度分数: 0.7893738140417458

mlp model混淆矩阵

[[105 0 7 1]

[ 51 0 0 0]

[ 16 0 55 27]

[ 2 0 7 256]]

mlp model召回率 0.614116441366996

mlp model分类报告

precision recall f1-score support

MCI 0.60 0.93 0.73 113

Normal 0.00 0.00 0.00 51

VMD 0.80 0.56 0.66 98

dementia 0.90 0.97 0.93 265

avg / total 0.73 0.79 0.75 527

# nn\_rf

MLPClassifier(activation='identity', alpha=0.0001, batch\_size='auto',

beta\_1=0.9, beta\_2=0.999, early\_stopping=False, epsilon=1e-08,

hidden\_layer\_sizes=3, learning\_rate='constant',

learning\_rate\_init=0.001, max\_iter=200, momentum=0.9,

nesterovs\_momentum=True, power\_t=0.5, random\_state=2018,

shuffle=True, solver='lbfgs', tol=0.0001, validation\_fraction=0.1,

verbose=False, warm\_start=False)

{'activation': 'identity', 'hidden\_layer\_sizes': 3, 'learning\_rate': 'constant', 'solver': 'lbfgs'}

## Train

mlp model准确度分数: 0.7905163329820865

mlp model混淆矩阵

[[ 949 134 112 39]

[ 75 252 1 0]

[ 286 8 259 165]

[ 57 0 117 2291]]

mlp model召回率 0.7068681104364145

mlp model分类报告

precision recall f1-score support

MCI 0.69 0.77 0.73 1234

Normal 0.64 0.77 0.70 328

VMD 0.53 0.36 0.43 718

dementia 0.92 0.93 0.92 2465

avg / total 0.78 0.79 0.78 4745

## test

mlp model准确度分数: 0.7760910815939279

mlp model混淆矩阵

[[ 84 12 11 6]

[ 7 43 1 0]

[ 34 1 36 27]

[ 4 0 15 246]]

mlp model召回率 0.7205372280820828

mlp model分类报告

precision recall f1-score support

MCI 0.65 0.74 0.69 113

Normal 0.77 0.84 0.80 51

VMD 0.57 0.37 0.45 98

dementia 0.88 0.93 0.90 265

avg / total 0.76 0.78 0.76 527

# adaboost\_relief

AdaBoostClassifier(algorithm='SAMME',

base\_estimator=LogisticRegression(C=10, class\_weight='balanced', dual=False,

fit\_intercept=True, intercept\_scaling=1, max\_iter=100,

multi\_class='ovr', n\_jobs=1, penalty='l2', random\_state=2018,

solver='liblinear', tol=0.0001, verbose=0, warm\_start=False),

learning\_rate=1.0, n\_estimators=10, random\_state=2018)

{'algorithm': 'SAMME', 'n\_estimators': 10}

## Train

AdaBoost model准确度分数: 0.7877766069546891

AdaBoost model混淆矩阵

[[1105 5 116 8]

[ 326 1 1 0]

[ 179 5 410 124]

[ 21 1 221 2222]]

AdaBoost model召回率 0.7877766069546891

AdaBoost model分类报告

precision recall f1-score support

MCI 0.68 0.90 0.77 1234

Normal 0.08 0.00 0.01 328

VMD 0.55 0.57 0.56 718

dementia 0.94 0.90 0.92 2465

avg / total 0.76 0.79 0.76 4745

## Test

AdaBoost model准确度分数: 0.7741935483870968

AdaBoost model混淆矩阵

[[ 63 40 8 2]

[ 7 44 0 0]

[ 17 0 57 24]

[ 1 0 20 244]]

AdaBoost model召回率 0.7741935483870968

AdaBoost model分类报告

precision recall f1-score support

MCI 0.72 0.56 0.63 113

Normal 0.52 0.86 0.65 51

VMD 0.67 0.58 0.62 98

dementia 0.90 0.92 0.91 265

avg / total 0.78 0.77 0.77 527

# adaboost\_infor

AdaBoostClassifier(algorithm='SAMME',

base\_estimator=LogisticRegression(C=10, class\_weight='balanced', dual=False,

fit\_intercept=True, intercept\_scaling=1, max\_iter=100,

multi\_class='ovr', n\_jobs=1, penalty='l2', random\_state=2018,

solver='liblinear', tol=0.0001, verbose=0, warm\_start=False),

learning\_rate=1.0, n\_estimators=10, random\_state=2018)

{'algorithm': 'SAMME', 'n\_estimators': 10}

## train

AdaBoost model准确度分数: 0.7943097997892519

AdaBoost model混淆矩阵

[[1116 5 109 4]

[ 327 1 0 0]

[ 159 1 417 141]

[ 25 1 204 2235]]

AdaBoost model召回率 0.7943097997892519

AdaBoost model分类报告

precision recall f1-score support

MCI 0.69 0.90 0.78 1234

Normal 0.12 0.00 0.01 328

VMD 0.57 0.58 0.58 718

dementia 0.94 0.91 0.92 2465

avg / total 0.76 0.79 0.77 4745

## test

AdaBoost model准确度分数: 0.7741935483870968

AdaBoost model混淆矩阵

[[ 61 35 17 0] M

[ 8 43 0 0] N

[ 13 0 54 31] V

[ 0 0 15 250]] D

AdaBoost model召回率 0.7741935483870968

AdaBoost model分类报告

precision recall f1-score support

M 0.74 0.54 0.63 113

N 0.55 0.84 0.67 51

VMD 0.63 0.55 0.59 98

dementia 0.89 0.94 0.92 265

avg / total 0.78 0.77 0.77 527

# adaboost\_rf

AdaBoostClassifier(algorithm='SAMME',

base\_estimator=LogisticRegression(C=10, class\_weight='balanced', dual=False,

fit\_intercept=True, intercept\_scaling=1, max\_iter=100,

multi\_class='ovr', n\_jobs=1, penalty='l2', random\_state=2018,

solver='liblinear', tol=0.0001, verbose=0, warm\_start=False),

learning\_rate=1.0, n\_estimators=10, random\_state=2018)

{'algorithm': 'SAMME', 'n\_estimators': 10}

## Train

AdaBoost model准确度分数: 0.7869336143308746

AdaBoost model混淆矩阵

[[ 926 143 133 32]

[ 73 252 3 0]

[ 277 12 303 126]

[ 67 1 144 2253]]

AdaBoost model召回率 0.7869336143308746

AdaBoost model分类报告

precision recall f1-score support

MCI 0.69 0.75 0.72 1234

Normal 0.62 0.77 0.68 328

VMD 0.52 0.42 0.47 718

dementia 0.93 0.91 0.92 2465

avg / total 0.79 0.79 0.78 4745

## test

AdaBoost model准确度分数: 0.7609108159392789

AdaBoost model混淆矩阵

[[ 66 14 29 4]

[ 7 43 1 0]

[ 26 3 50 19]

[ 4 0 19 242]]

AdaBoost model召回率 0.7609108159392789

AdaBoost model分类报告

precision recall f1-score support

MCI 0.64 0.58 0.61 113

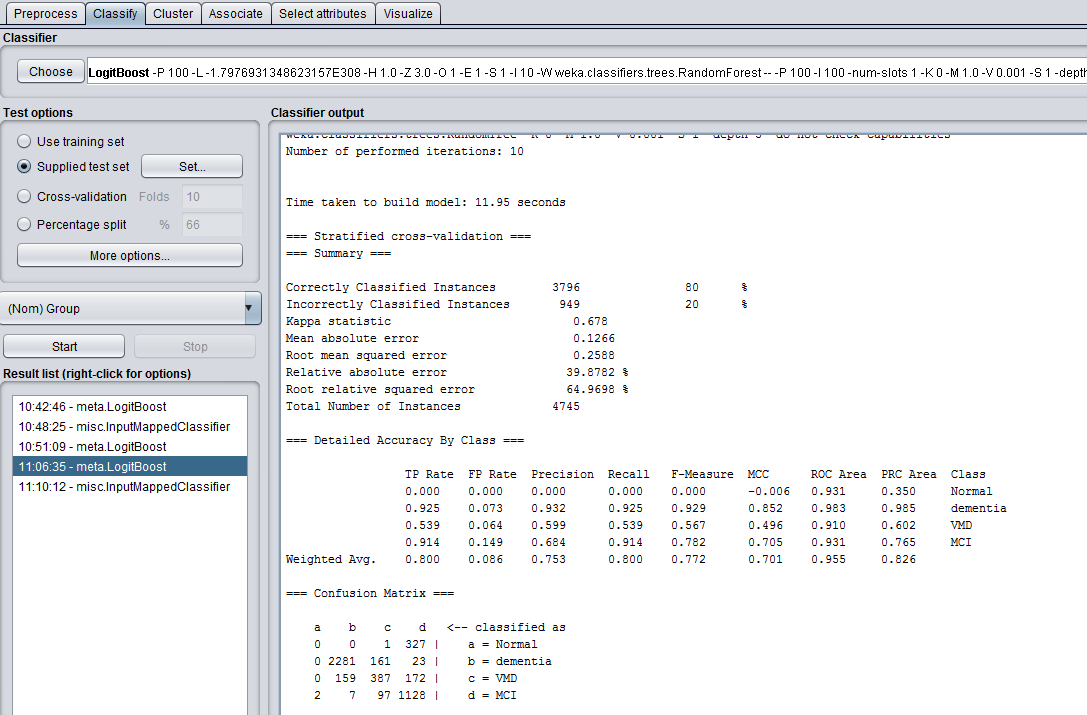
Normal 0.72 0.84 0.77 51

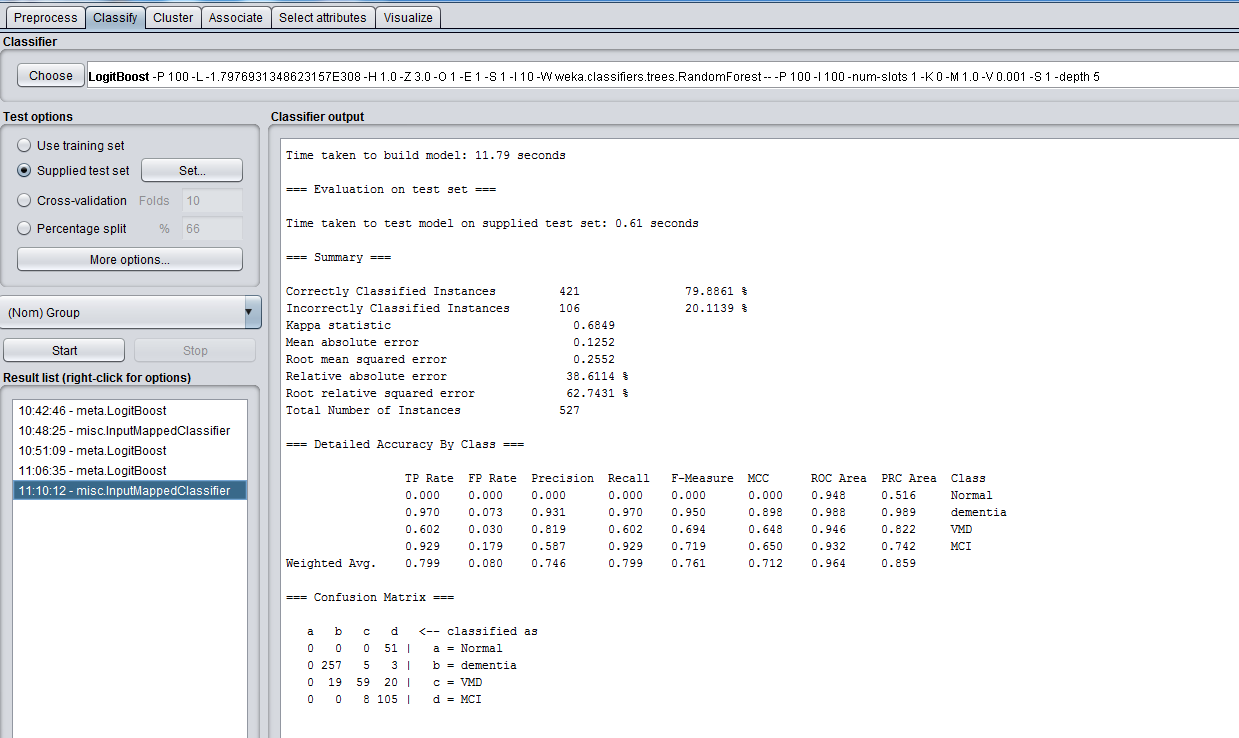
VMD 0.51 0.51 0.51 98

dementia 0.91 0.91 0.91 265

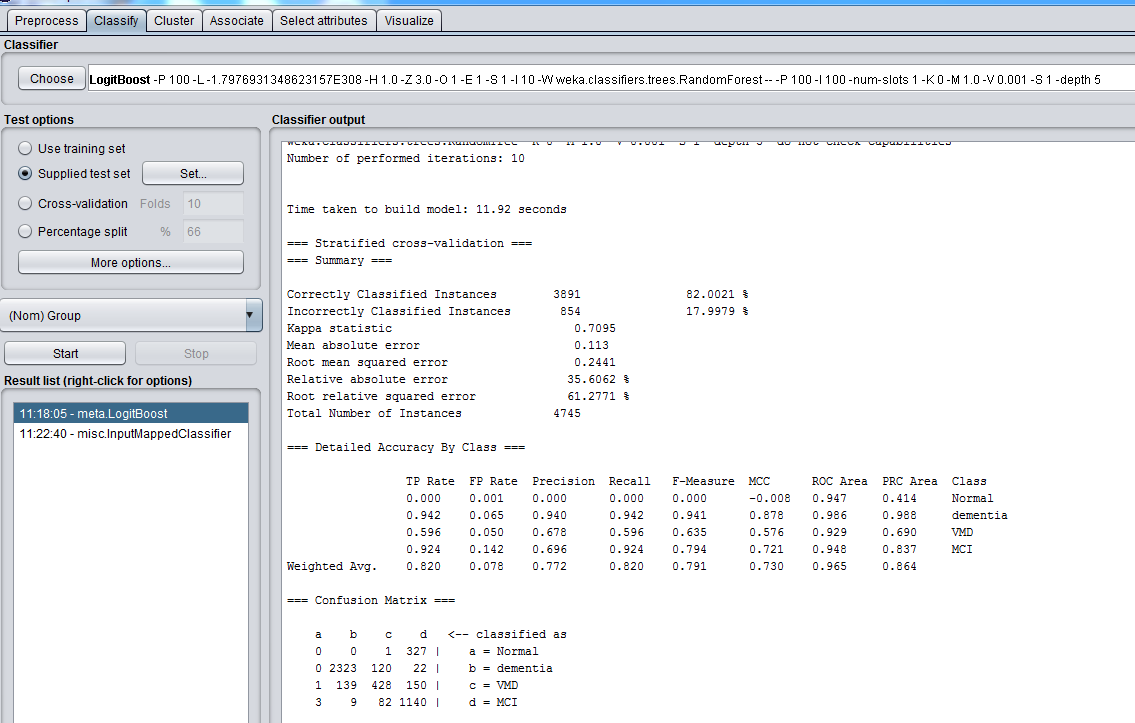
avg / total 0.76 0.76 0.76 527

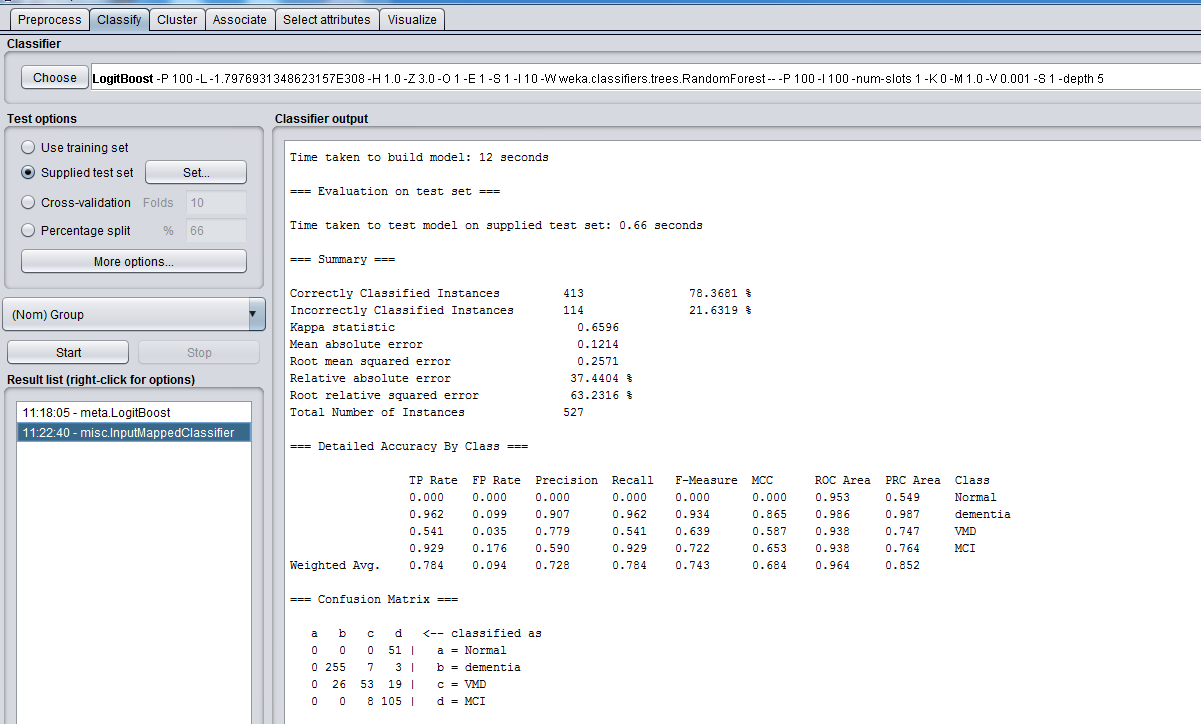
# logitboost\_relief



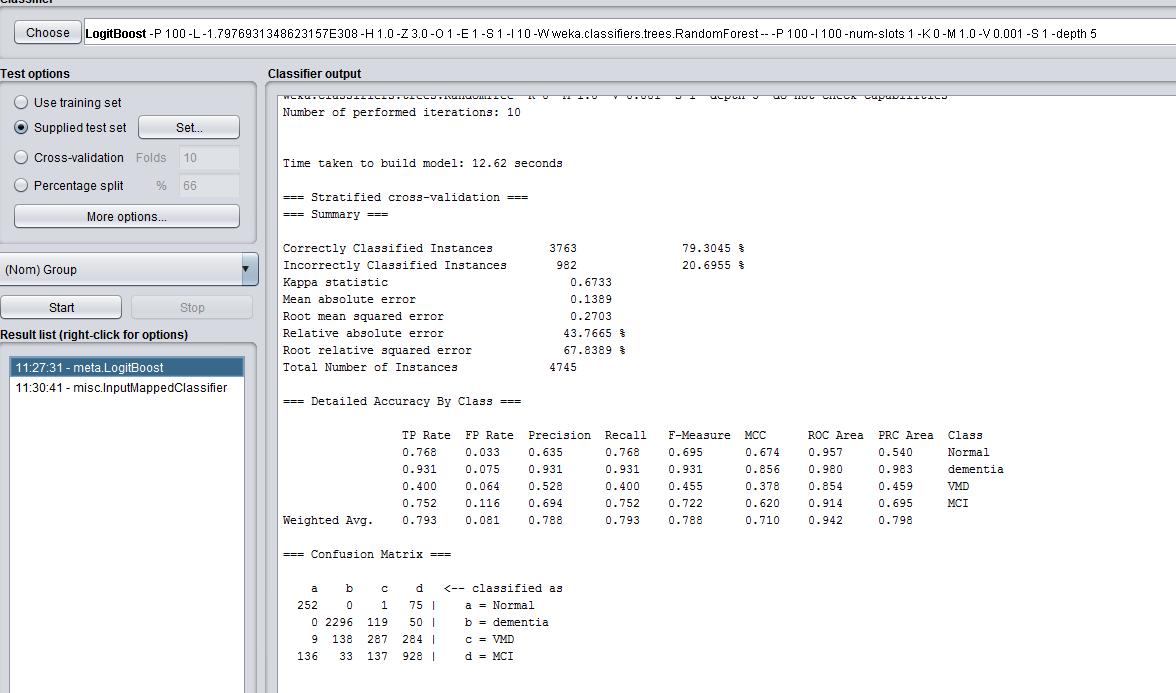


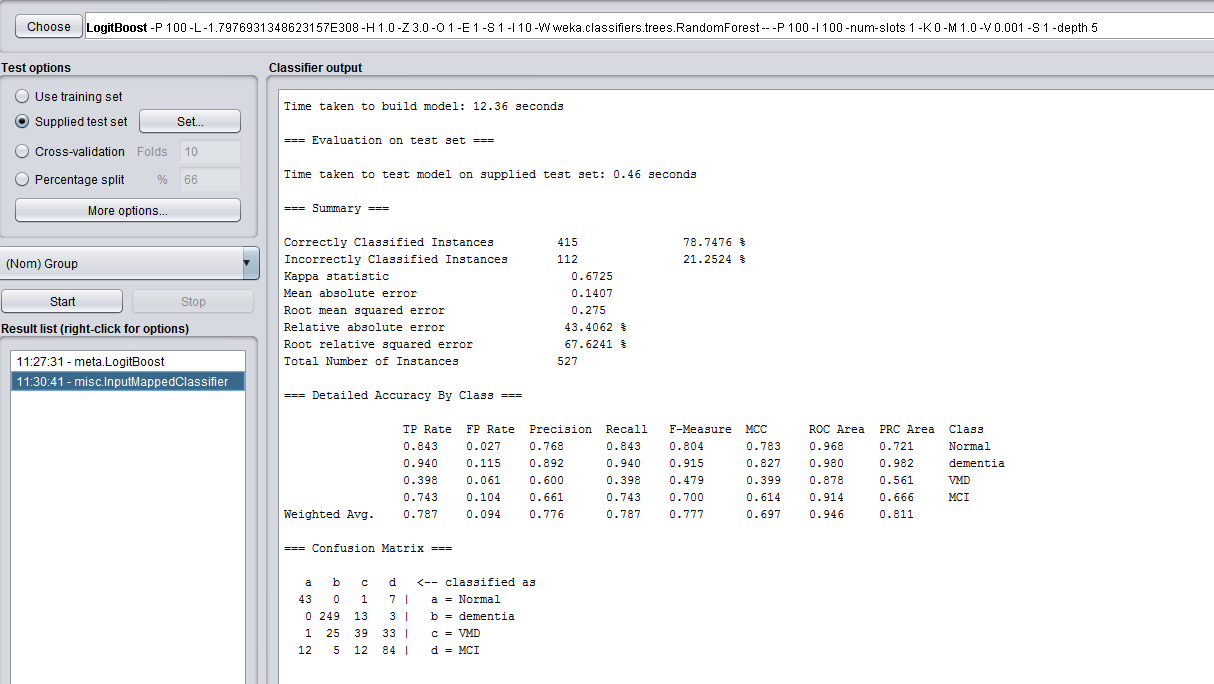
# Logit\_infor





# Logit\_rf





Performance of Naive Bayes (Accuracy, sensitivity, specificity, AUC). Results were obtained using the features selection.