**Problem 1**

TP: 628

TN: 394

FP: 59

FN: 137

P = TP + FN = 628 + 137 = 765

N = FP + TN = 59 + 394 = 453

All = 765 + 453 = 1218

**Sensitivity** = TP/P

= 628/765

= **0.821**

**Specificity** = TN/N

= 394/453

= **0.870**

**Precision** = TP/(TP + FP)

= 628/(628+59)

= **0.914**

**Accuracy** = (TP + TN)/All

= (628+394)/1218

= **0.839**

Recall = TP/(TP + FN)

= 628/(628+137)

= 0.821

**F-measure** = (2\*precision\*recall)/(precision + recall)

= (2\*0.914\*0.821)/(0.914+0.821)

= **0.865**

**F2** = ((1+22) \* precision \* recall) / (22 \* precision + recall)

= ((1+22) \* 0.914\* 0.821) / (22 \* 0.914+ 0.821)

= **0.838**

**Problem 2: Omit per instruction**

Since we did not discuss hypothesis testing in the class yesterday, you  
don't need to do Problem 2.

**Problem 3**

For this problem, you are required to run, on Weka, **Naïve Bayes, J48, SimpleLogistic, RandomForest, neural network** (Multilayer Perceptron),and **OneR** classification algorithms on ***german-bank.arff***dataset and compare the performance of the models built by these six algorithms. Make sure that you select “Cross-validation” for “Test options.” If you have to choose one model, which one would you choose and why? Note that the neural network algorithm will take a longer time than other algorithms.

After you run each classification algorithm, you need to capture screenshot of Weka's result window, which shows all performance measures including a confusion matrix, and include it in your submission.

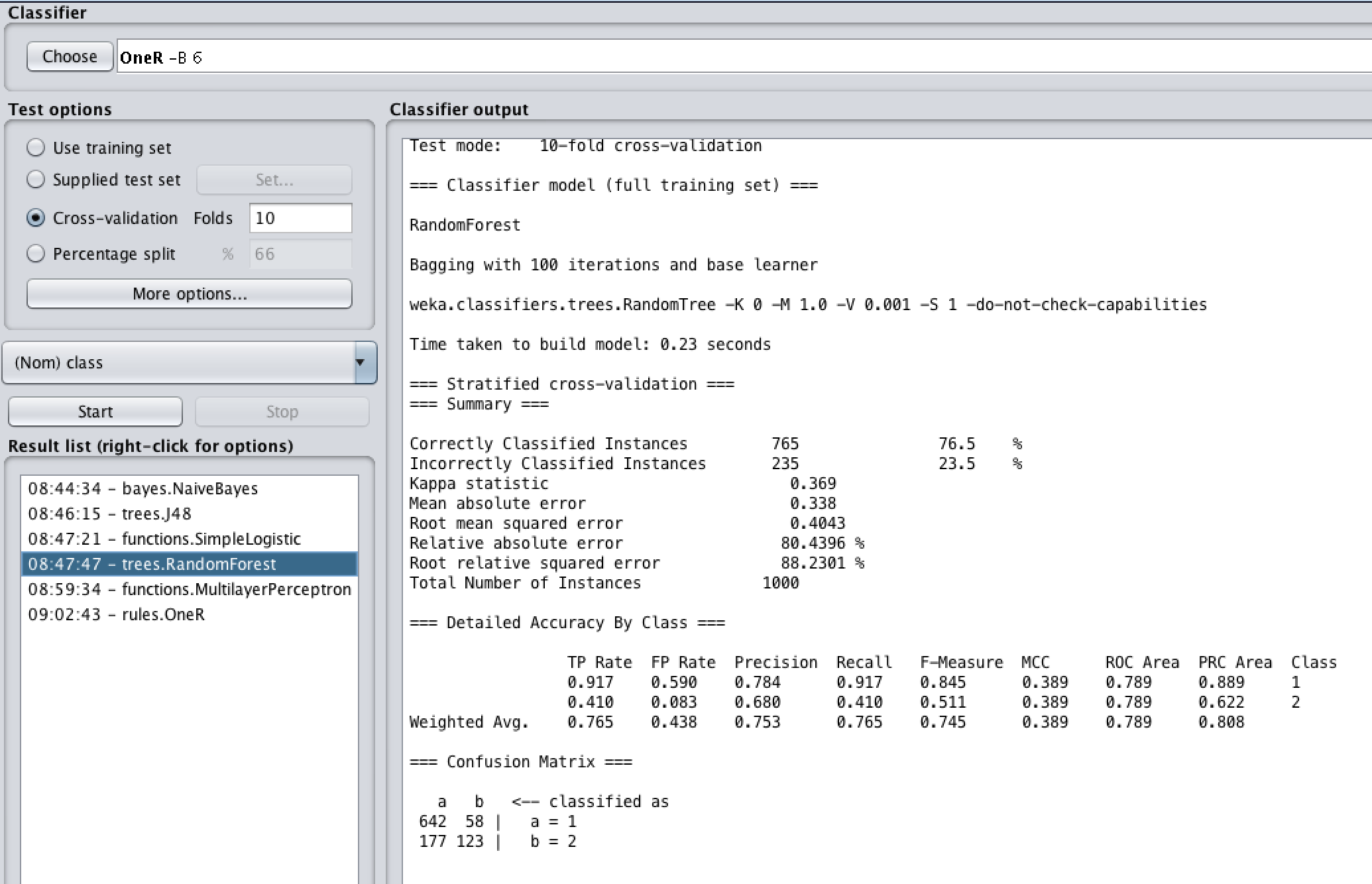
The correctly classified instance percentage pertaining to each algorithm are:

|  |  |  |
| --- | --- | --- |
| **algorithm** | **Correct (%)** | **params** |
| weka.classifiers.trees.RandomForest | **76.5** | -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1 |
| weka.classifiers.functions.SimpleLogistic | **75.9** | -I 0 -M 500 -H 50 -W 0.0 |
| weka.classifiers.bayes.NaiveBayes | **75.4** |  |
| weka.classifiers.functions.MultilayerPerceptron | **72.3** | -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a |
| weka.classifiers.trees.J48 | **70.7** | -C 0.25 -M 2 |
| weka.classifiers.rules.OneR | **66.1** | -B 6 |

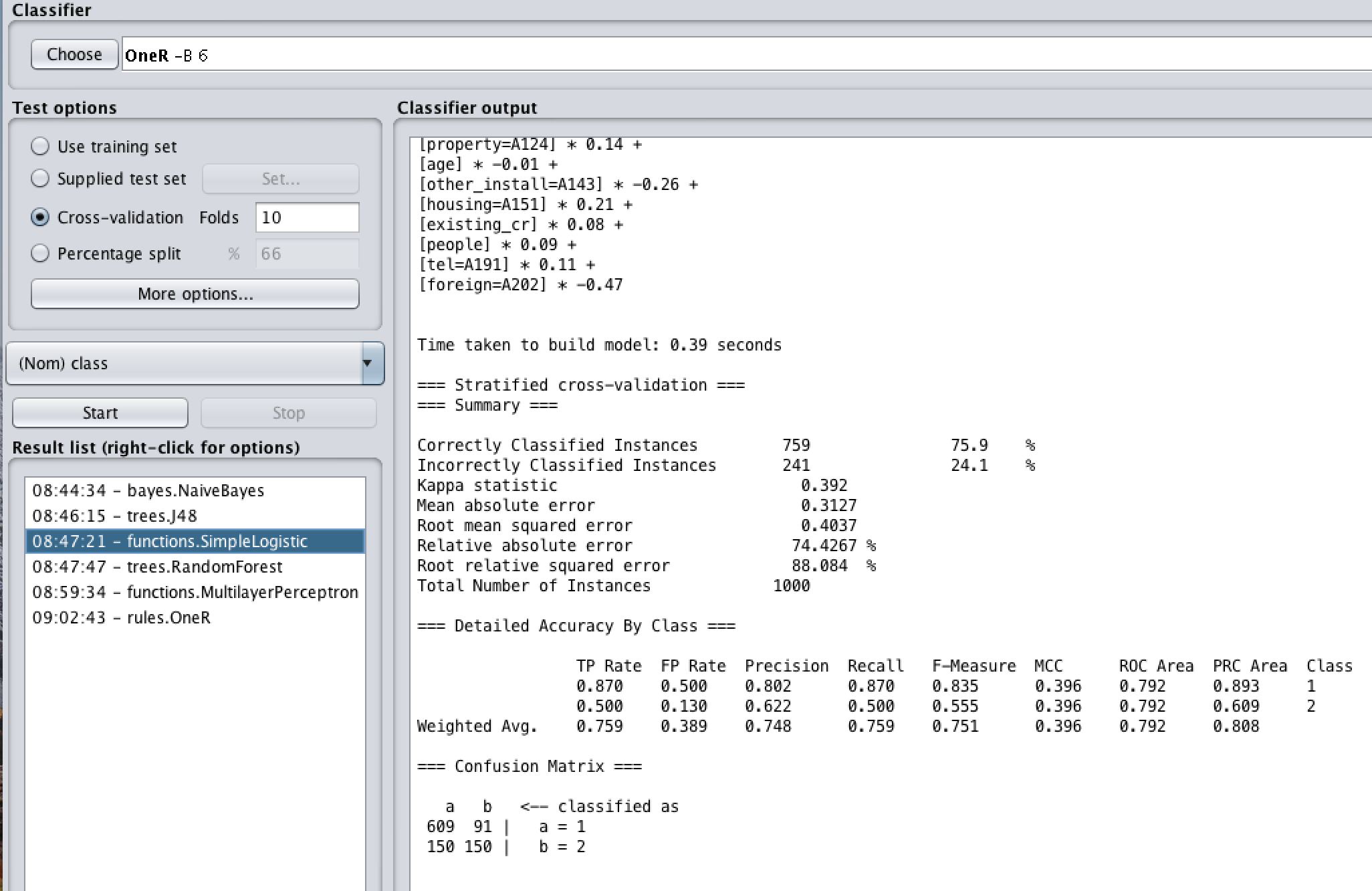
**RandomForest** has the most correctly classified instances. However, the description of the german dataset specifically notes “*It is worse to class a customer as good when they are bad (5), than it is to class a customer as bad when they are good (1).*” Therefore, the algorithm that produces the lowest rate of false positives on class 1 is 0.477 for the **MultilayerPerceptron**.

**Algorithm result window screenshots**

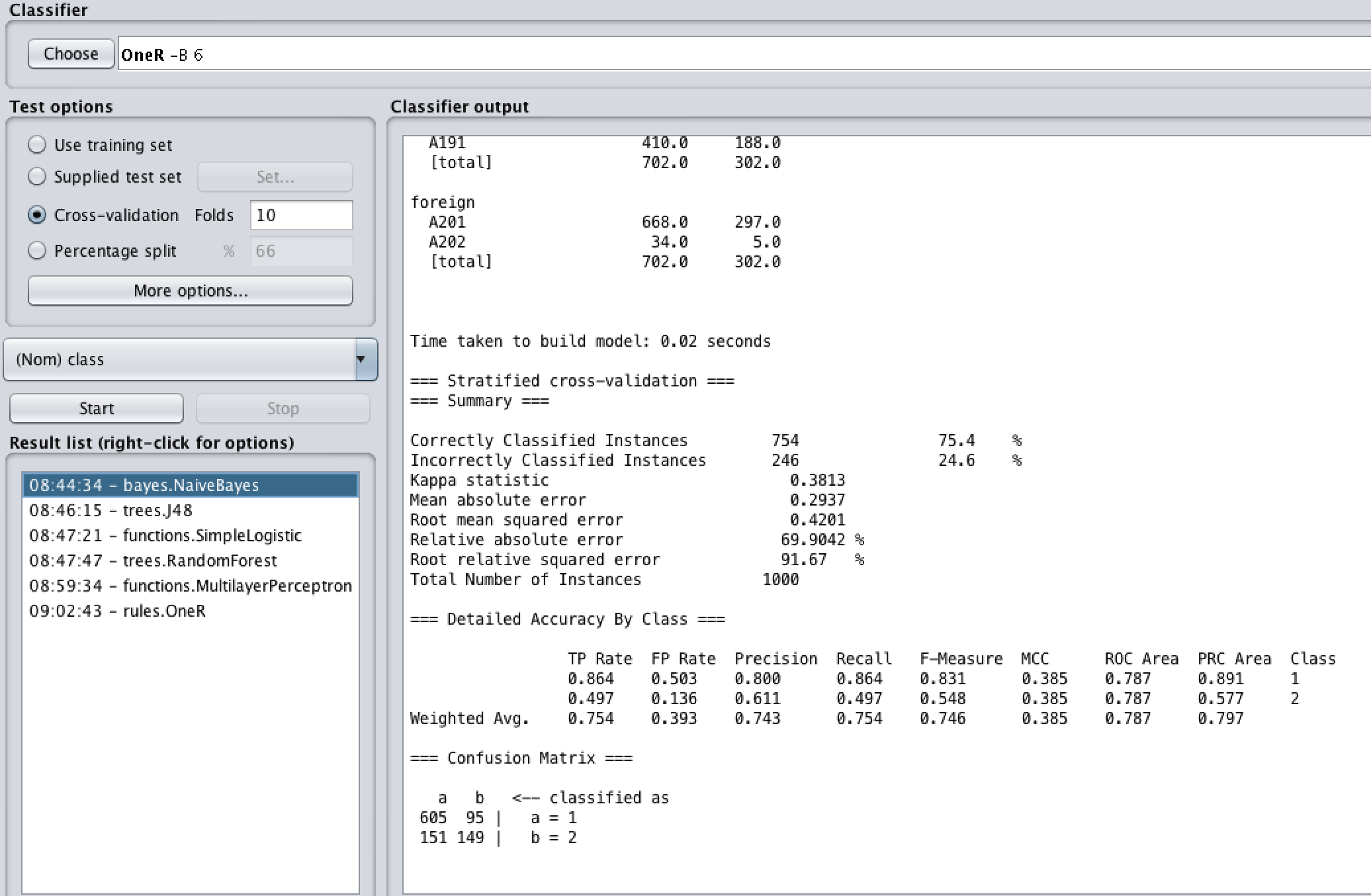
***weka.classifiers.trees.RandomForest***



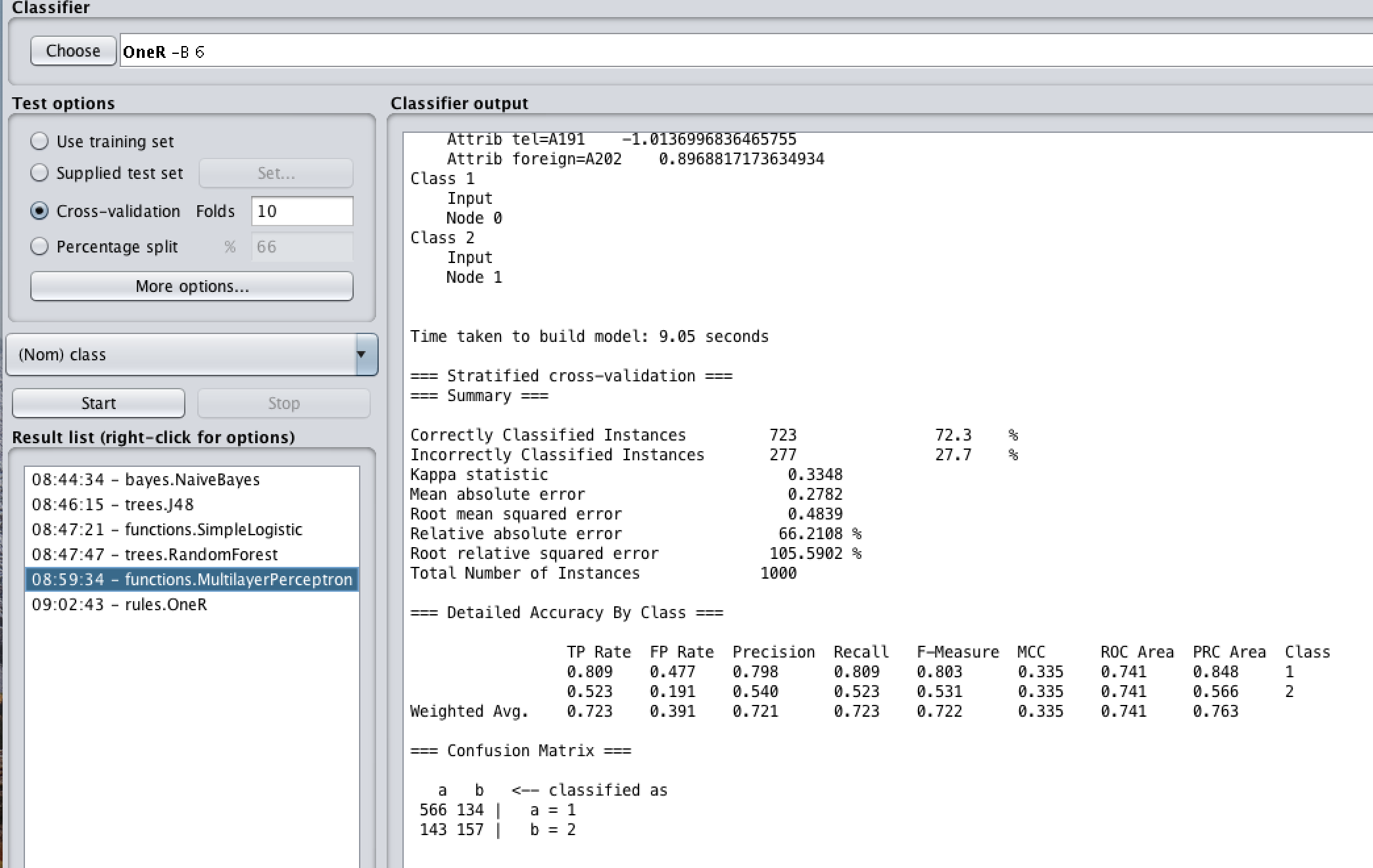
***weka.classifiers.functions.SimpleLogistic***



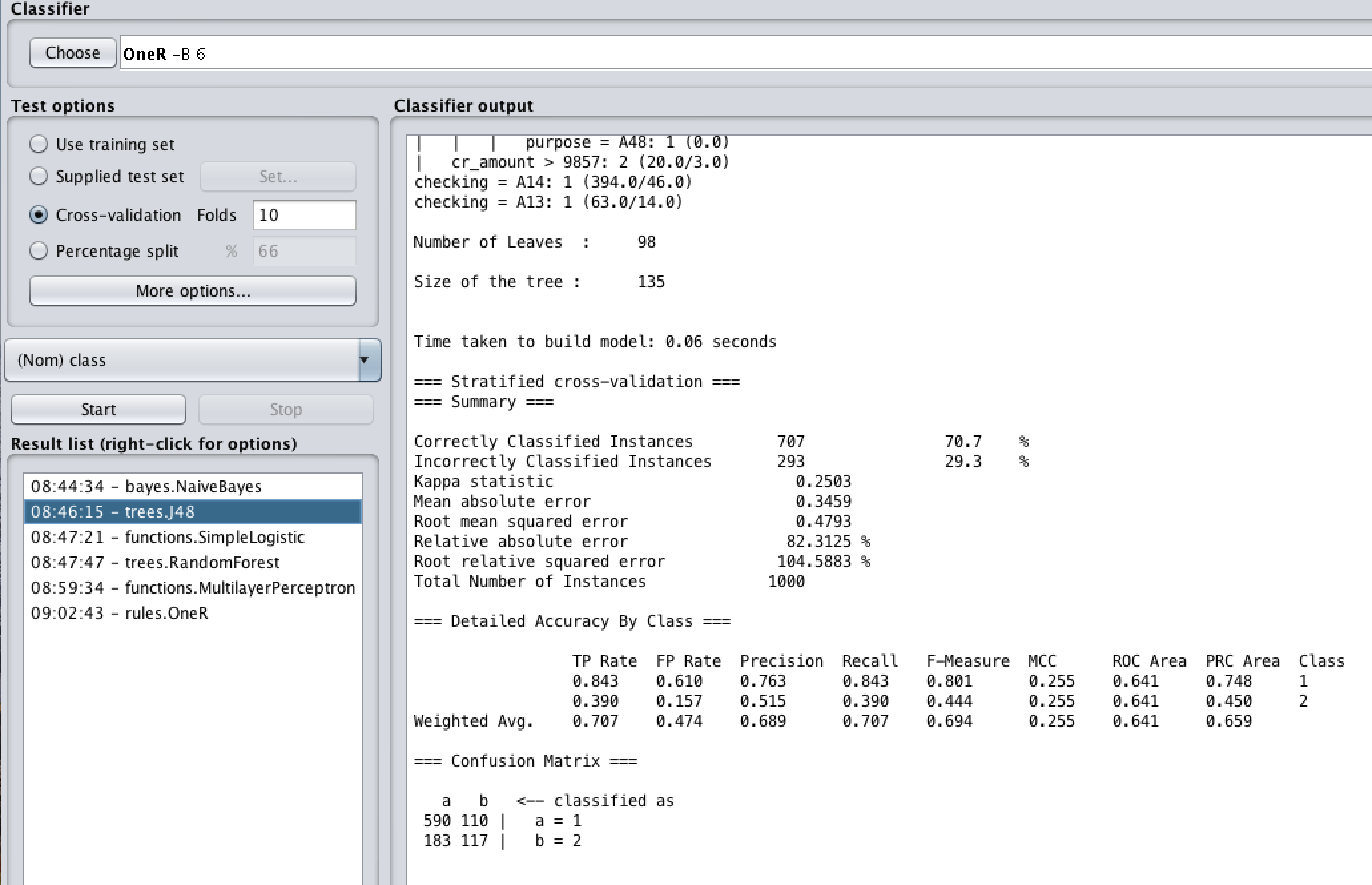
***weka.classifiers.bayes.NaiveBayes***



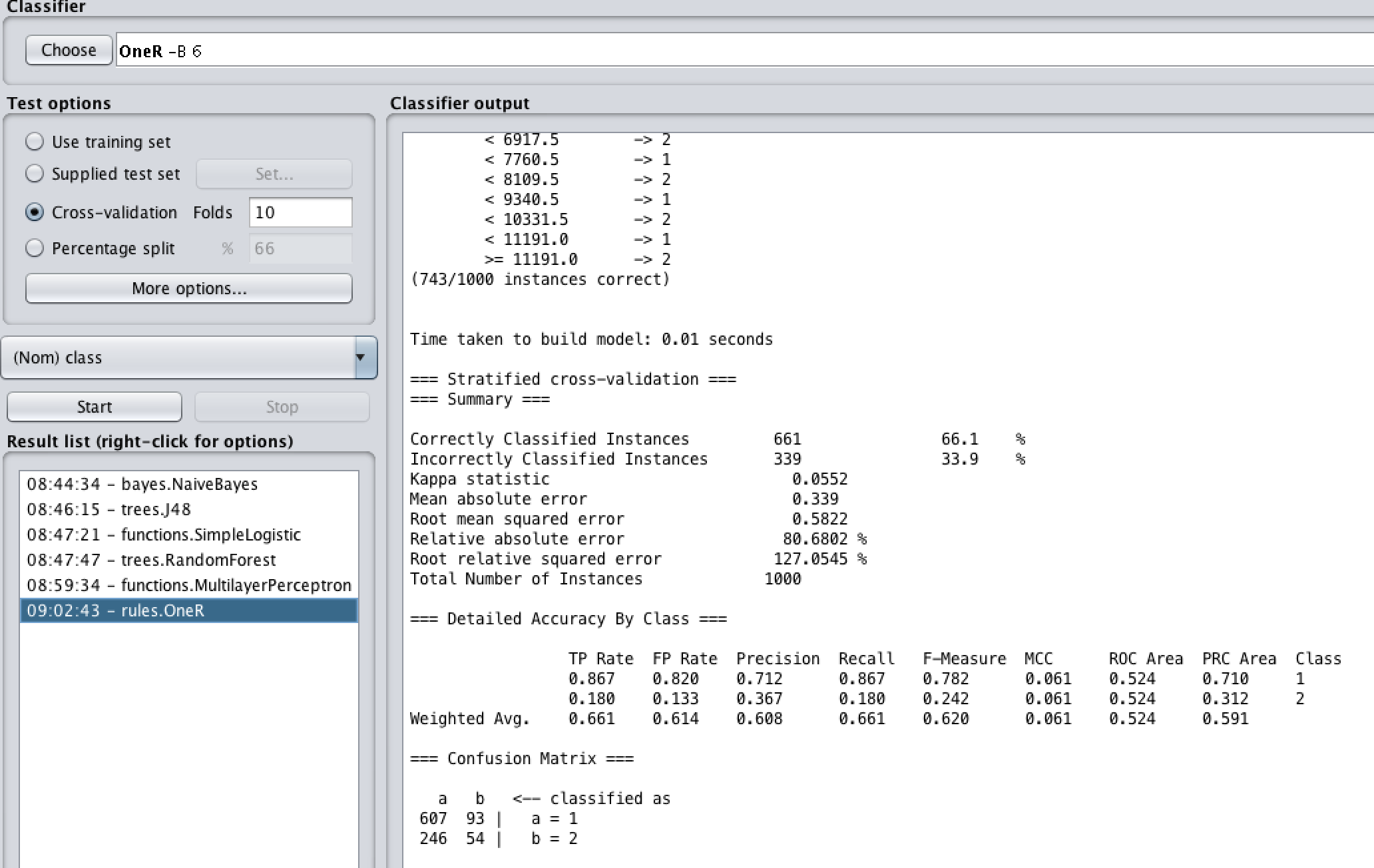
***weka.classifiers.functions.MultilayerPerceptron***



***weka.classifiers.trees.J48***



***weka.classifiers.rules.OneR***



**Note**: Classifier output for each classification algorithm are attached in **logs** subdirectory