This document spells out when and how to change the reproduction test files.

As of 2019\_12\_17, there are **two** sets of tests performed, both under the BasicReproductions.java test class.

**First**, reproductions of results made on a test dataset (ItalyPowerDemand) when the classifier was (at least to the best of knowledge accepted to be) in a working and correct state. The classifier with default parameters (but seeded if it is Randomisable) is trained and tested on the same, baked-in, version of the IPD dataset. Exact matches between predictions and distributions (within reasonable precision) and overall performance is confirmed. Timings are (by default) ignored, however can be turned on (in code, requires a push rather than parameterised) to make sure timings are within some acceptable threshold on average.

A selection of ‘important’ classifiers and/or ones that provide coverage of many systems are included in the tests. There’s a trade off in this regard between speed of tests and coverage.

If any of the recreations are not sufficiently correct, the tests fail and e.g. Travis/github will not allow merging with dev or master.

A classifier failing a recreation could mean, among other things:

* Default parameters of one of the classifiers has changed
* The way an algorithm works has been changed
* The output of classifier evaluation code may have changed, e.g. ClassifierResults, evaluators

These tests are mainly designed to capture *unintentional* changes in classifiers.

If an *intentional* change has been made, i.e. such that the correct output of the classifier has changed, then the stored results needs to be updated.

This process involves deleting the old classifier results file(s) from the repo, and regenerating them.

The results files are stored under

src\main\java\experiments\reproductions

Delete the files to be changed, and then you can run

BasicReproductionTests.generateMissingExpectedResults()

To reproduce them.

The classifiers to be tested are stored in the classifierPaths array. If there are refactors and a classifier changes location, or new classifiers need to be added to the list of those tested, this array needs to be updated. Results generation and comparison looks at this array.

**Second**, a severe subset of the results from the CAWPE paper are reproduced from scratch and the summaries compared for exact matches to expected values.

* Small selection of small datasets, a few base classifiers, a few folds
* Base classifiers trained, and performance estimated via CV on each dset/fold
* Ensembles trained on base classifier results from file, tested and estimated via CV
* Ensemble’s performances compared with MCE, summary compared to expected string.

This test gives a reasonable coverage of a lot of different aspects of the code base, though obviously only particular settings of each part.

The expected results for these shouldn’t ever really need changing. Only cases I can imagine would be if e.g. prediction tie resolution behaviour changes code-base wide, or if the summary outputs of MCE are overhauled by design. Otherwise, parameters of base Weka classifiers, parameters of competing ensembles, the fundamentals of cross validation etc shouldn’t ever really change.