|  |  |
| --- | --- |
| **Tester** | Tah Wen Zhong |
| **Test type** | Integration |
| **Component** | Main program |
| **Number of test suites** | 5 |
| **Number of test cases** | 32 |
| **Status** | Complete |
| **Test file** | main\_program\_test.py |
| **Date of completion** | 19/9/2021 |

Contents

[Description 3](#_Toc83045800)

[Test suite 1: Read and process data (10/9/2021) 5](#_Toc83045801)

[First run: Fail 5](#_Toc83045802)

[Second run: Pass 6](#_Toc83045803)

[Screenshots 7](#_Toc83045804)

[Test suite 2: Feature selection + Pre-processing (10/9/2021) 8](#_Toc83045805)

[First run: Fail 8](#_Toc83045806)

[Second run: Pass 10](#_Toc83045807)

[Screenshots 12](#_Toc83045808)

[Test suite 3: Model creation (16/9/2021) 13](#_Toc83045809)

[First run: Pass 13](#_Toc83045810)

[Screenshots 15](#_Toc83045811)

[Test suite 4: Evaluation (17/9/2021) 16](#_Toc83045812)

[First run: Pass 16](#_Toc83045813)

[Screenshots 17](#_Toc83045814)

[Test suite 5: CSV writer + Main function (18/9/2021) 18](#_Toc83045815)

[First run: Fail 18](#_Toc83045816)

[Second run: Pass 21](#_Toc83045817)

[Screenshots 23](#_Toc83045818)

Testing the functionality of the Main program

# Description

The table below describes the steps within the main program:

|  |  |  |
| --- | --- | --- |
| **Step** | **Description** | **External algorithms used** |
| 1 | Read and process data | - |
| 2 | Feature selection + Pre-processing | Pre-process algorithm, Feature selection algorithm |
| 3 | Model creation | All base and ensemble predictor algorithms |
| 4 | Evaluation | All evaluation algorithms |
| 5 | Output CSV | - |

For testing:

* Test suite (1-4) are dedicated towards testing the individual steps as described in the table above
* Test suite 5 will test the csv output along with the overall process of the whole program

This document keeps track of the state of the algorithm throughout testing. If a test suite fails, the information of the bug fixes will be documented, and the test will be performed once more with the same test suite.

* Yellow highlight indicates failed test cases
* Purple encoded text indicates test cases that can run but results were affected by a failed test case
* Red encoded text indicates test cases that cannot run due to another failed test case

Additional note

* The main program was built through a modular approach, so several parts of the main function were made into sub functions. Hence, testing these sub functions essentially verifies the functionality its involved in the main function as well.
* The user interface will ensure at least one model (either base or ensemble) and feature will be selected. So minimum bound for tests involving the model and feature selections will have at least 1 selection for each.
* All test suites were included within the same test file, that is main\_test.py
* The 3 dataset test files involved can be found in the Appendix A, B and C
* The main function will handle errors which can only be identified during execution. These include issues such as an invalid dataset, k-fold or train size being passed to the algorithm.

# Test suite 1: Read and process data (10/9/2021)

## First run: Fail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test case description | Test data/setup | Expected Result | Actual Result | Pass/Fail |
| MP.1.1 | Check if the data is extracted correctly from read function | test1.arff.txt | Data is extracted correctly  (An array containing values for each row) | Data is extracted correctly  (An array containing values for each row) | Pass |
| MP.1.2 | Test process which separates the labels and metrics | test1.arff.txt | Correctly identify the metrics and labels | Failed to extract the labels | Fail |
| MP.1.3 | Check correctness of label conversion function | test1.arff.txt | Labels converted from Boolean/string to integer (0,1) | - | - |
| MP.1.4 | Test run with actual dataset | test2.arff.txt  (KC dataset) | Extracted data with labels processed correctly | - | - |
| Bug description | | | * The data extraction function extracts data column with “Defective” as the column name. | | |
| Bug fix | | | Rework data extraction function to take the last column as label | | |

Test suite 1: Read and process data (10/9/2021)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test case description | Test data/setup | Expected Result | Actual Result | Pass/Fail |
| MP.1.2 | Test process which separates the labels and metrics | test1.arff.txt | Correctly identify the metrics and labels | Correctly identify the metrics and labels | Pass |
| MP.1.3 | Check correctness of label conversion function | test1.arff.txt | Labels converted from Boolean/string to integer (0,1) | Labels converted from Boolean/string to integer (0,1) | Pass |
| MP.1.4 | Test run with actual dataset | test2.arff.txt  (KC dataset) | Extracted data with labels processed correctly | Extracted data with labels processed correctly | Pass |

## Second run: Pass

## Screenshots

|  |
| --- |
| **Screenshot 1: Test code** |
|  |
| **Screenshot 2: Test suite 1 output** |
|  |

# Test suite 2: Feature selection + Pre-processing (10/9/2021)

## First run: Fail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test case description | Test data/setup | Expected Result | Actual Result | Pass/Fail |
| MP.2.1 | Test the base functionality of the function | test3.arff.txt  (After processing) | Output successful, with no errors | Failed to output, error raised | Fail |
| MP.2.2 | Check if a single feature selection is correctly identified | Selection set to only CFS | Output contains result using CFS algorithm | - | - |
| MP.2.3 | Check if multiple feature selections are correctly identified | Selection set to CFS and RFE | Output contains result using CFS and RFE algorithm | - | - |
| MP.2.4 | Check k-fold argument correctly passed to pre-processing algorithm | test3.arff.txt  (k\_fold = 4) | Returns 4 train-test splits for all output (All, CFS, RFE) | - | - |
| MP.2.5 | Check train size argument correctly passed to feature selection algorithm | test3.arff.txt  (train\_size = 3) | Feature selection (CFS, RFE) outputs contain 3 columns (3 Software metrics) | - | - |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MP.2.6 | Check if function handles errors appropriately  (k\_fold) | test3.arff.txt  (k\_fold = 10000) | No error occurs and return a tuple containing (False, 1) | - | Fail |
| MP.2.7 | Check if function handles errors appropriately  (train\_size) | test3.arff.txt  (train\_size = 10000) | No error occurs and return a tuple containing (False, 2) | - | Fail |
| MP.2.8 | Test run, including previous steps, with actual dataset | test2.arff.txt  (KC dataset) | Output successful, with no errors | Output successful, with no errors | Pass |
| Bug description | | | * The bug was found within the argument passed to the pre-processing algorithm * The failed runs are because the fold written in the argument did not overwrite the default folds * Test case 2.6 was successful because it uses actual dataset, so the data size is larger than the default folds | | |
| Bug fix | | | The function was fixed so the folds argument was correctly passed to the pre-processing algorithm | | |

Test suite 2: Feature selection + Pre-processing (10/9/2021)

First run (continued): Fail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test case description | Test data/setup | Expected Result | Actual Result | Pass/Fail |
| MP.2.1 | Test the base functionality of the function | test3.arff.txt  (After processing) | Output successful, with no errors | Output successful, with no errors | Pass |
| MP.2.2 | Check if a single feature selection is correctly identified | Selection set to only CFS | Output contains result using CFS algorithm | Output contains result using CFS algorithm | Pass |
| MP.2.3 | Check if multiple feature selections are correctly identified | Selection set to CFS and RFE | Output contains result using CFS and RFE algorithm | Output contains result using CFS and RFE algorithm | Pass |
| MP.2.4 | Check k-fold argument correctly passed to pre-processing algorithm | test3.arff.txt  (k\_fold = 4) | Returns 4 train-test splits for all output (All, CFS, RFE) | Returns 4 train-test splits for all output (All, CFS, RFE) | Pass |
| MP.2.5 | Check train size argument correctly passed to feature selection algorithm | test3.arff.txt  (train\_size = 3) | Feature selection (CFS, RFE) outputs contain 3 columns (3 Software metrics) | Feature selection (CFS, RFE) outputs contain 3 columns (3 Software metrics) | Pass |

Test suite 2: Feature selection + Pre-processing (10/9/2021)

## Second run: Pass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MP.2.6 | Check if function handles errors appropriately  (k\_fold) | test3.arff.txt  (k\_fold = 10000) | No error occurs and return a tuple containing (False, 1) | No error occurs and return a tuple containing (False, 1) | Pass |
| MP.2.7 | Check if function handles errors appropriately  (train\_size) | test3.arff.txt  (train\_size = 10000) | No error occurs and return a tuple containing (False, 2) | No error occurs and return a tuple containing (False, 2) | Pass |
| MP.2.8 | Test run, including previous steps, with actual dataset | test2.arff.txt  (KC dataset) | Output successful, with no errors | Output successful, with no errors | Pass |

Test suite 2: Feature selection + Pre-processing (16/9/2021)

Second run (continued): Pass

## Screenshots

|  |
| --- |
| **Screenshot 1: Test code** |
|  |
| **Screenshot 2: Test suite 2 output** |
|  |

# Test suite 3: Model creation (16/9/2021)

## First run: Pass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test case description | Test data/setup | Expected Result | Actual Result | Pass/Fail |
| 3.1 | Test whether all models can fit processed data (All) | test3.arff.txt  (After processing) | Output successful, with no errors | Output successful, with no errors | Pass |
| 3.2 | Test whether all models can fit processed data after CFS selection | test3.arff.txt  (After processing and CFS) | Output successful, with no errors | Output successful, with no errors | Pass |
| 3.3 | Test whether all models can fit processed data after RFE selection | test3.arff.txt  (After processing and RFE) | Output successful, with no errors | Output successful, with no errors | Pass |
| 3.4 | Test base models selection argument | test3.arff.txt  (base\_pred = [0,3]) | Result contains complement naïve bayes and MLP models | Result contains complement naïve bayes and MLP models | Pass |
| 3.5 | Test ensemble models selection argument | test3.arff.txt  (ensemble\_preds = [0]) | Result contains random forest model | Result contains random forest model | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3.6 | Test whether all models contain the predict() functionality | test3.arff.txt  (All models selected) | All models contain the predict() function and can perform predictions | All models contain the predict() function and can perform predictions | Pass |

Test suite 3: Model creation (16/9/2021)

First run (continued): Pass

## Screenshots

|  |
| --- |
| **Screenshot 1: Test code** |
|  |
| **Screenshot 2: Test suite 3 output** |
|  |

# Test suite 4: Evaluation (17/9/2021)

## First run: Pass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test case description | Test data/setup | Expected Result | Actual Result | Pass/Fail |
| 4.1 | Test whether model built can be evaluated | test3.arff.txt  (Model used: Decision tree) | Output successful, with no errors | Output successful, with no errors | Pass |
| 4.2 | Test whether evaluation result contains a list of 4 items | test3.arff.txt | Output array contains 4 items | Output array contains 4 items | Pass |
| 4.3 | Check if the results are all appropriate values | test3.arff.txt | Output array contains values within the range 0 to 1 | Output array contains values within the range 0 to 1 | Pass |
| 4.4 | Check whether the evaluation result follows the correct order | test3.arff.txt | Output array follows this order:  AUC, F1-score, FPR, FNR | Output array follows this order:  AUC, F1-score, FPR, FNR | Pass |

## Screenshots

|  |
| --- |
| **Screenshot 1: Test file** |
|  |
| **Screenshot 2: Test suite 4 output** |
|  |

# Test suite 5: CSV writer + Main function (18/9/2021)

## First run: Fail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test case description | Test data/setup | Expected Result | Actual Result | Pass/Fail |
| 5.1 | Test the main function with all arguments set to max | test3.arff.txt  (All models selected) | Output successful, with no errors | Output successful, with no errors | Pass |
| 5.2 | Test the main function with all arguments set to minimum | test3.arff.txt  k-fold = 2  train = 1  (1 model selected) | Output successful, with no errors | - | Fail |
| 5.3 | Check if the base model selection argument functioning correctly | test3.arff.txt  (Selected CNB model) | Output only contains evaluation results for the Complement Naïve Bayes model | - | - |
| 5.4 | Check if the ensemble model selection argument functioning correctly | test3.arff.txt  (Selected random forest model) | Output only contains evaluation results for the Random Forest model | - | - |
| 5.5 | Check if the main function can handle string input instead of integers | test3.arff.txt  (k-fold and train argument values are String type) | Output successful, with no errors | - | - |

Test suite 5: CSV writer + Main function (18/9/2021)

First run (continued): Fail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5.6 | Check if the main program passes the feature selection argument correctly | test3.arff.txt  (All and RFE selected) | Evaluation result contains both data that contains all metrics and metrics after reduction from RFE | - | - |
| 5.7 | Check if the main program handles invalid train size input appropriately | test3.arff.txt  (train\_size = 1000) | The function halts with no error and returns a tuple containing (False, 2) | - | - |
| 5.8 | Check if the main program handles invalid k-fold input appropriately | test3.arff.txt  (k\_fold = 1000) | The function halts with no error and returns a tuple containing (False, 1) | - | - |
| 5.9 | Check if the main program handles invalid datasets appropriately | fail1.arff.txt  (An invalid dataset) | The function halts with no error and returns a tuple containing (False, 0) | - | - |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5.10 | Check if the main program correctly produces the CSV file | test3.arff.txt  k-fold = 2  train = 1  (3 model selected) | CSV file containing the correct values and format | - | - |
| Bug description | | | * The code contains two variables that represent the k-fold value, one of which is not updated based on the argument | | |
| Bug fix | | | Reduce the number of variables representing the k-fold value to one | | |

Test suite 5: CSV writer + Main function (18/9/2021)

First run (continued): Fail

Test suite 5: CSV writer + Main function (18/9/2021)

## Second run: Pass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test case description | Test data/setup | Expected Result | Actual Result | Pass/Fail |
| 5.1 | Test the main function with all arguments set to max | test3.arff.txt  (All models selected) | Output successful, with no errors | Output successful, with no errors | Pass |
| 5.2 | Test the main function with all arguments set to minimum | test3.arff.txt  k-fold = 2  train = 1  (1 model selected) | Output successful, with no errors | Output successful, with no errors | Pass |
| 5.3 | Check if the base model selection argument functioning correctly | test3.arff.txt  (Selected CNB model) | Output only contains evaluation results for the Complement Naïve Bayes model | Output only contains evaluation results for the Complement Naïve Bayes model | Pass |
| 5.4 | Check if the ensemble model selection argument functioning correctly | test3.arff.txt  (Selected random forest model) | Output only contains evaluation results for the Random Forest model | Output only contains evaluation results for the Random Forest model | Pass |
| 5.5 | Check if the main function can handle string input instead of integers | test3.arff.txt  (k-fold and train argument values are String type) | Output successful, with no errors | Output successful, with no errors | Pass |

Test suite 5: CSV writer + Main function (18/9/2021)

Second run (continued): Pass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5.6 | Check if the main program passes the feature selection argument correctly | test3.arff.txt  (All and RFE selected) | Evaluation result contains both data that contains all metrics and metrics after reduction from RFE | Evaluation result contains both data that contains all metrics and metrics after reduction from RFE | Pass |
| 5.7 | Check if the main program handles invalid train size input appropriately | test3.arff.txt  (train\_size = 1000) | The function halts with no error and returns a tuple containing (False, 2) | The function halts with no error and returns a tuple containing (False, 2) | Pass |
| 5.8 | Check if the main program handles invalid k-fold input appropriately | test3.arff.txt  (k\_fold = 1000) | The function halts with no error and returns a tuple containing (False, 1) | The function halts with no error and returns a tuple containing (False, 1) | Pass |
| 5.9 | Check if the main program handles invalid datasets appropriately | fail1.arff.txt  (An invalid dataset) | The function halts with no error and returns a tuple containing (False, 0) | The function halts with no error and returns a tuple containing (False, 0) | Pass |
| 5.10 | Check if the main program correctly produces the CSV file | test3.arff.txt  k-fold = 2  train = 1  (3 model selected) | CSV file containing the correct values and format | CSV file containing the correct values and format  (Screenshot 3) | Pass |

## Screenshots

|  |
| --- |
| **Screenshot 1: Test file** |
|  |

|  |
| --- |
| **Screenshot 1 (continued): Test file** |
|  |
| **Screenshot 2: Test suite 4 output** |
|  |

|  |
| --- |
| **Screenshot 3: CSV output** |
|  |