**MASTER TEST PLAN**

**1.** **Introduction:**

This document details the features of Master test planning of Weka - Data Mining and Machine Learning.

**1.1. Document identifier**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Status** |
| 1.0.1 | 11/28/2015 | Mohammed Aamer  #M08802120 | Draft |
| 1.0.1 | 11/28/2015 | Kshitija Joshi  #M08766983 | Draft |

**Approvers List**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Approver/Reviewer** | **Approval/Review Date** |
| Vignesh Subbian | Collaborator | Reviewer |  |

**1.2. Scope**

The testing is done for weka stable source code 3.6.10. Static analysis, Code coverage, Unit tests validation, Data flow testing and Test automation are performed for the above features. Assumptions & Limitations: Testing is limited to three features which are associations, attribute selection and classifiers.

**1.3. References**

**1.3.1 External References**

|  |  |
| --- | --- |
| **Version** | **Name** |
| IEEE Std 829TM-2008 | IEEE Standard for Software and System Test Documentation |
| IEEE Std 730TM-2014 | IEEE Standard for Software Quality Assurance Processes |
|  | http://weka.wikispaces.com/Primer |

**1.3.2 Internal References**

|  |  |
| --- | --- |
| **Version** | **Name** |
| 1.0 | hopper.g\_projectproposal |
| 3.6.0 | Weka |

**1.4. System overview and key features**

Weka is a platform for Data Mining and Machine Learning, written in Java. It contain collection of visualization tools and algorithms for data analysis and predictive modeling. It has graphical and command-line interface; graphical interface can be used for small to medium sized data analysis, for large amount of data command-line interface is used. The main features of the tool are preprocess, classify, associate, clustering and visualization of the data.

**1.5. Test overview**

Below is the table which include test schedule, test resources, techniques, and methods necessary to perform the testing.

**1.5.1** **Organization**

Enlisted the requirements in Tarantula. Created test cases and mapped respective test cases to the test objects for generation of reports to analyse the execution of test cases.

**1.5.2** **Master test schedule**

Describe the test activities within the project life cycle and milestones. Summarize the overall schedule of the testing tasks, identifying where task results feed back to the development, organizational, and supporting processes (e.g., quality assurance and configuration management). Describe the task iteration policy for the re-execution of test tasks and any dependencies.

**1.5.3** **Integrity level schema**

At the beginning of each process, the assignment of integrity levels is reassessed with respect to changes that may need to be made in the integrity levels as a result of architecture selection, design choices, code construction, or other development activities.

**1.5.4** **Resources summary**

· Eclipse - IDE, To built the project

· SVN - Repository for the project

· Word - Creating of level/master test plans

· Tarantula - Test planning

·

**1.5.5** **Responsibilities**

Performed roles such as inspector, reader and moderator by Kshitija Joshi and Mohammed Aamer.

**1.5.6 Testing Tools, techniques, methods, and metrics**

· JUnit - Unit testing

· Eclemma- Code Coverage tool

· FindBugs - Static Analysis tool

· STAF - Test Automation

· JArchitect - Static Analysis and Code metrics tool

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Schedule ID:** | TSW001 | | |
| **Product ID / Name:** | Weka – Data mining tool | | |
| **Product Version or Build:** | 3.6.1.0 stable Standard Ed. Version | | |
| **Present Owner :** | Kshitija Joshi | | |
| **Created On:** | 11-04-2015 | | |
| **Review On:** | 11-11-2015 || 11-25-2015 || 11-28-2015 | | |
| **Review By:** | Mohammed Aamer FNU - Student | | |
| **Review Comments:** | Updated testing results | | |
| **Current Version:** | 1.0.1 | | |
| **Change Details:** | Implemented and generated reports using Emma and Jarchitect. | | |
| **Current Status:** | Completed | | |
| **Signing Off Authority:** | **Name** | **Position** | **Signature, Date** |
| Vignesh Subbian | Professor |  |

|  |  |  |
| --- | --- | --- |
| **Test Step** | **Duration** | **Status** |
| **First Spiral** |  |  |
| **(A) Information gathering** |  |  |
| a) Summarize Findings | **6 Days** | **Completed** |
| **(B) Test Planning** |  |  |
| b1) Build Test Plan | **8 Hours** | **Completed** |
| b2) Define the Metric Objectives | **12 hours** | **Completed** |
| b3) Review / Approve Plan | **4 Hours** | **Completed** |
| **(C) Test Case Design** |  |  |
| C1) Analyzing Function Tests | **10 Hours** | **Completed** |
| c2) Analyzing designed Tests For Classifier Class | **20 Hours** | **Completed** |
| C3) Review Design |  | **Completed** |
| **(D) Test Development** |  |  |
| d1) Review Developed Test cases |  | **In process** |
| **(E) Test Execution/Evaluation** |  |  |
| e1) Setup and Testing | **20 Hours** | **Completed** |
| e2) Evaluation | **2 Hours** | **Completed** |
| **(F) Prepare for the Next Spiral** |  |  |
| f1) Refine the tests | **-** | **Not started** |
| f3) Publish Report | **1 Hour** | **Completed** |
| **Last Spiral …** |  |  |
| **(G) Test Execution/Evaluation** |  |  |
| g1) Setup and Testing | **20 Hours** | **Completed** |
| g2) Evaluation | **5 hours** | **Completed** |
| **(H) Complete Testing** |  |  |
| h1) Complete Level Test Plan | **10 Hours** | **Completed** |
| h2) Complete Master Test Plan | **8 Hours** | **Completed** |
| H3) Complete Tests Execution | **-** | **Completed** |
| **(J) Summarize/Report Test Results** |  |  |
| J1) Prepare Final Test Report | **-** | **In progress** |
| J2) Review / Approve the Final Test Report | **-** | **In progress** |

**2. Details of the Master Test Plan**

**Test processes including definition of test levels**

2.1.1 Process: Management

2.1.1.1 Management of test effort: Created test plans for five weeks and five releases for analyzing and evaluating test case execution of test cases written for Weka

2.1.2 Process: Supply

2.1.3.1 Planning test: Created test case plan in Tarantula for getting results in the form of reports. And developed MTP and LTP as per IEEE standards

2.1.3 Process: Development

2.1.4.1 Concept: Software Application to implement data mining and machine learning algorithm concepts.

2.1.4.2 Requirements: To implement data mining, visualization and machine learning concepts

2.1.4.3 Test: Unit test cases

2.1.4 Process: Maintenance

2.1.6.1 Maintenance tests: Scripts can be created to automate the test plans as execution of the test cases daily/monthly.

**3. General**

**3.1. Glossary:**

References for MTP:

· The Art of Software Testing - Myers, Glenford J.

· Software Testing & Continuous Quality Improvement – W E Lewis

**3.2. Document change procedures and history**

Draft 1 : 11-18-2016

Draft 2 : 11-25-2016

Draft 3 : 11-28-2016

Test and Quality Assurance Plan

**1. Introduction**

This document details the features which will be tested, analysed in the Weka - Data Mining and Machine Learning Platform, for the stable version 3.6.10. The source code will be analysed with the FindBugs tools, STAF test automation framework.

**1.1. Document identifier**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Status** |
| 1 | 11/04/2015 | Mohammed Aamer  #M08802120 | Draft |
| 1 | 11/04/2015 | Kshitija Joshi  #M08766983 | Draft |

**Approvers List**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Approver/Reviewer** | **Approval/Review Date** |
| Vignesh Subbian | Collaborator | Reviewer | November 6, 2015 |
|  |  |  |  |

**1.2. Scope**

The testing is done for weka stable source code 3.6.10. Testing is limited to three features which are associations, attribute selection and classifiers.

# **1.3. References**

|  |  |
| --- | --- |
| **Version** | **Name** |
| IEEE Std 829TM-2008 | IEEE Standard for Software and System Test Documentation |
| IEEE Std 730TM-2014 | IEEE Standard for Software Quality Assurance Processes |
|  | https://weka.wikispaces.com/Primer |

# **2. Details for this level of test plan**

Features of the weka such as classifiers will be mostly concentrated on. Static analysis and the regression testing of the same is to be performed.

# **2.1 Test items and their identifiers**

Testing will be exercised on mainly functionality related to association, attribute selection and classifications.

**2.2 Test Traceability Matrix**

As per the list of the requirements mentioned in Tarantula will be exercised by test cases mapped. The requirements are software-based system functions and there are no nonfunctional requirements are mentioned.

**2.3 Features to be tested**

The following features in the packages will be tested -

Associations

weka.associations

Attribute Selections

weka.attributeSelection

Classifiers

weka.classifiers

weka.classifiers.bayes

weka.classifiers.data

weka.classifiers.functions

weka.classifiers.functions.supportVector

weka.classifiers.lazy

weka.classifiers.meta

weka.classifiers.meta.nestedDichotomies

weka.classifiers.mi

weka.classifiers.mi.supportVector

weka.classifiers.misc

weka.classifiers.pmml.consumer

weka.classifiers.pmml.data

weka.classifiers.rules

weka.classifiers.trees

Test automation

Static Analysis

**2.4 Features not to be tested**

These features are not going to be tested in this test plan -

Clusterers,

Core,

Date generators,

Filter.

# **2.5 Approach**

Whitebox Testing, Analysis, Inspection

# **2.6 Test deliverables**

Generate report of unit test cases executed,

Report on FindBugs static analysis report and Emma report,

Report on automation of test cases using STAF

# **3. Test management**

# **3.1 Planned activities and tasks**

Unit test cases and regression test cases are to be executed and analysed. Use FindBugs for static analysis and compare the results with the formal inspection. Automate test cases using STAF.

# **3.2 Environment/infrastructure**

Tools/Platforms needed to test weka source code

Java 1.6 SDK

Eclipse Helios

>= Ant 1.6.5

>= Junit 3.8.2

# **3.3 Training**

Skills/Knowledge of different associations and classifiers is required for understanding the source code and test cases.

# **3.4 Risk(s) and contingency(s)**

As the tool handles data for datasets which can be very large or very small, there is chance of improper allocation of heap space in the Java Virtual Machine. Proper heap space should be allocated for the JVM to handle large datasets.

# **4. General**

# **4.1 Quality assurance procedures**

Peer Reviews, Verification and Validation of the results of tests.

# **4.2 Metrics**

Size & comment percentage

• Defect density

• Cyclomatic complexity

• Weighted methods per class

**4.3 Test coverage**

Planning to use Emma to analyze the code coverage such as statement coverage, branch coverage and analyze and interpret the results

**Results:**

Analyzed results executed by Emma. Included results and analysis of line coverage, branch coverage in folder ‘Test Support’

**4.4 Glossary**

# **Acronyms**

|  |  |
| --- | --- |
| STAF | Software Testing and Automation Framework |
| FindBugs | Static Code Analysis Tool |
| Emma | Java Code Coverage Tool |