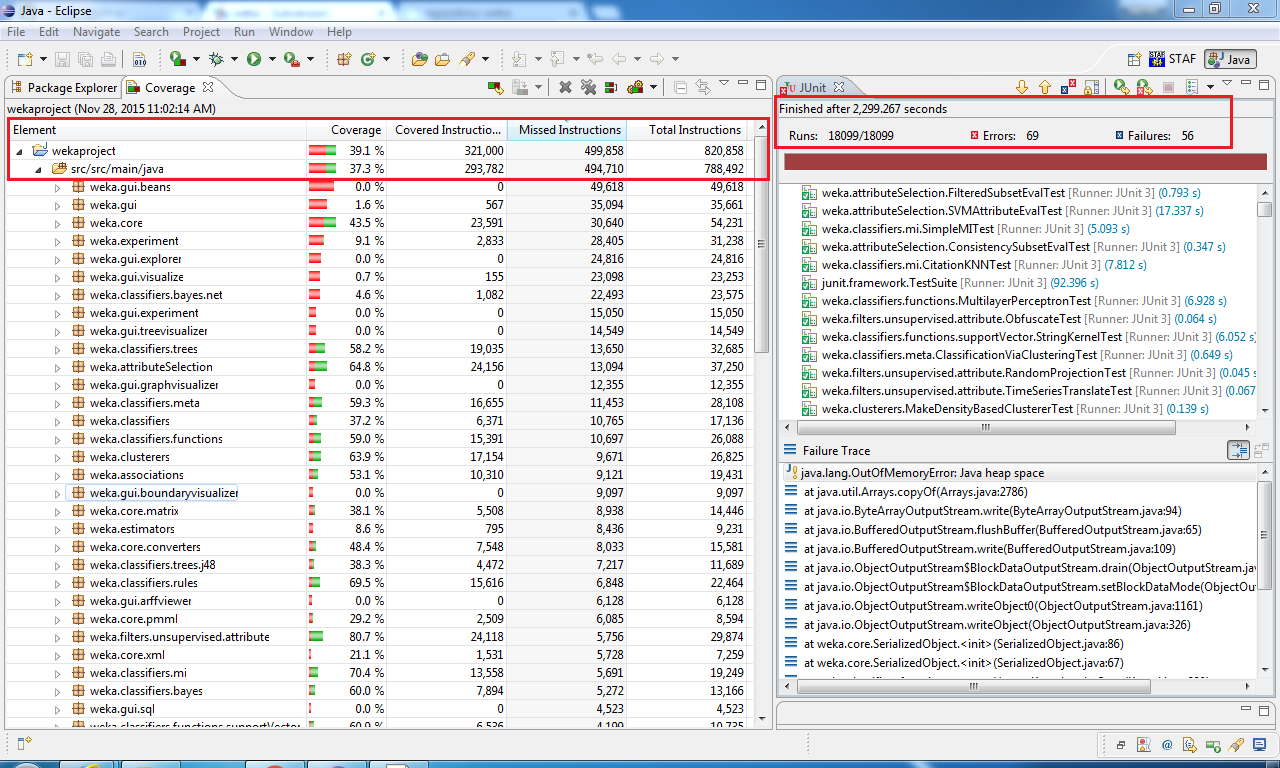
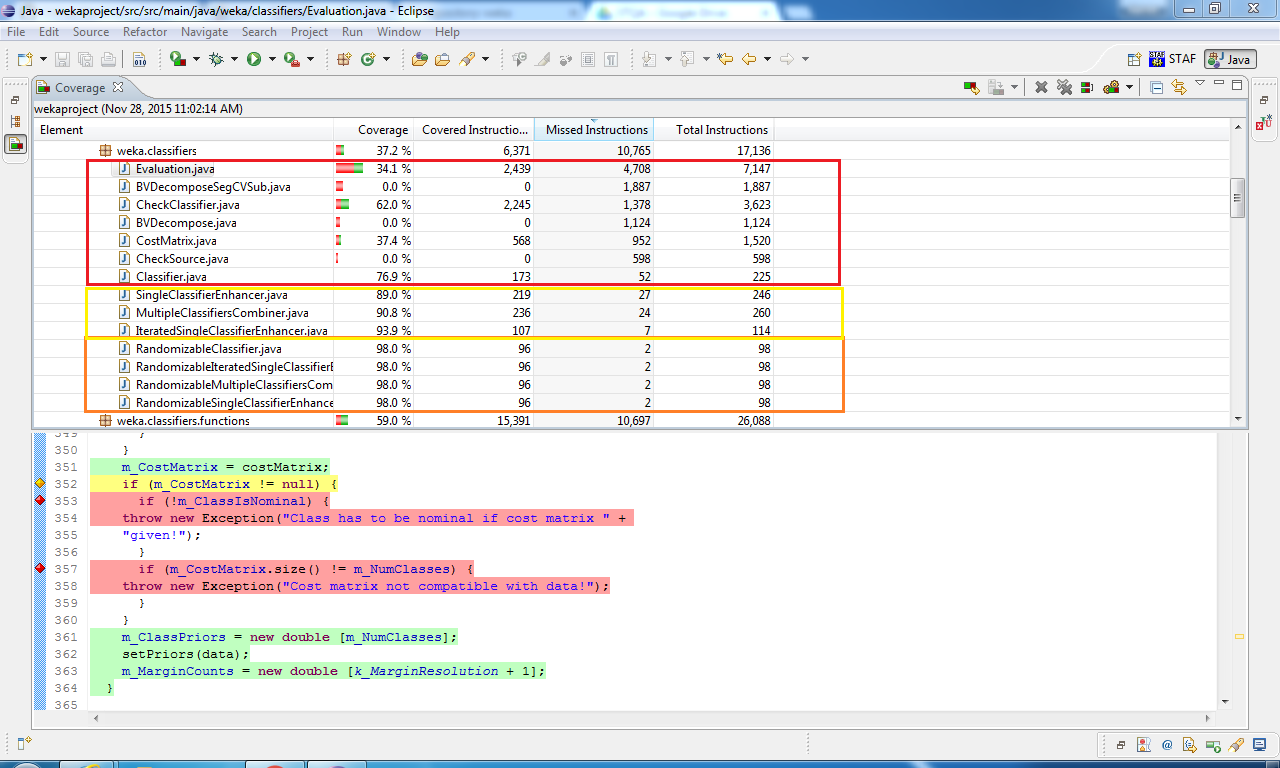
**Emma Coverage Report:**

1. Below is the result of overall code coverage for Classifier module:



1. Abstract classes are classes containing one or more abstract methods. An abstract method is a method that is declared, but contains no implementation. And for interfaces, the body of an interface includes only abstract methods and final fields (constants). Hence these classes contain a few numbers of lines of code and hence it is showing high percentage of code coverage. Please check the analysis in below report



Below are those abstract classes of Classifiers module:

Java classes are all abstract utility classes for handling settings common to randomizable classifiers

*RandomizableClassifier*

*RandomizableIteratedSingleClassifierEnhancer*

*RandomizableMultipleClassifiersCombiner*

*RandomizableSingleClassifierEnhancer*

Interfaces: Interface for classifiers that can induce models of growing complexity one step at a time.

Below are interfaces of Classifiers module:

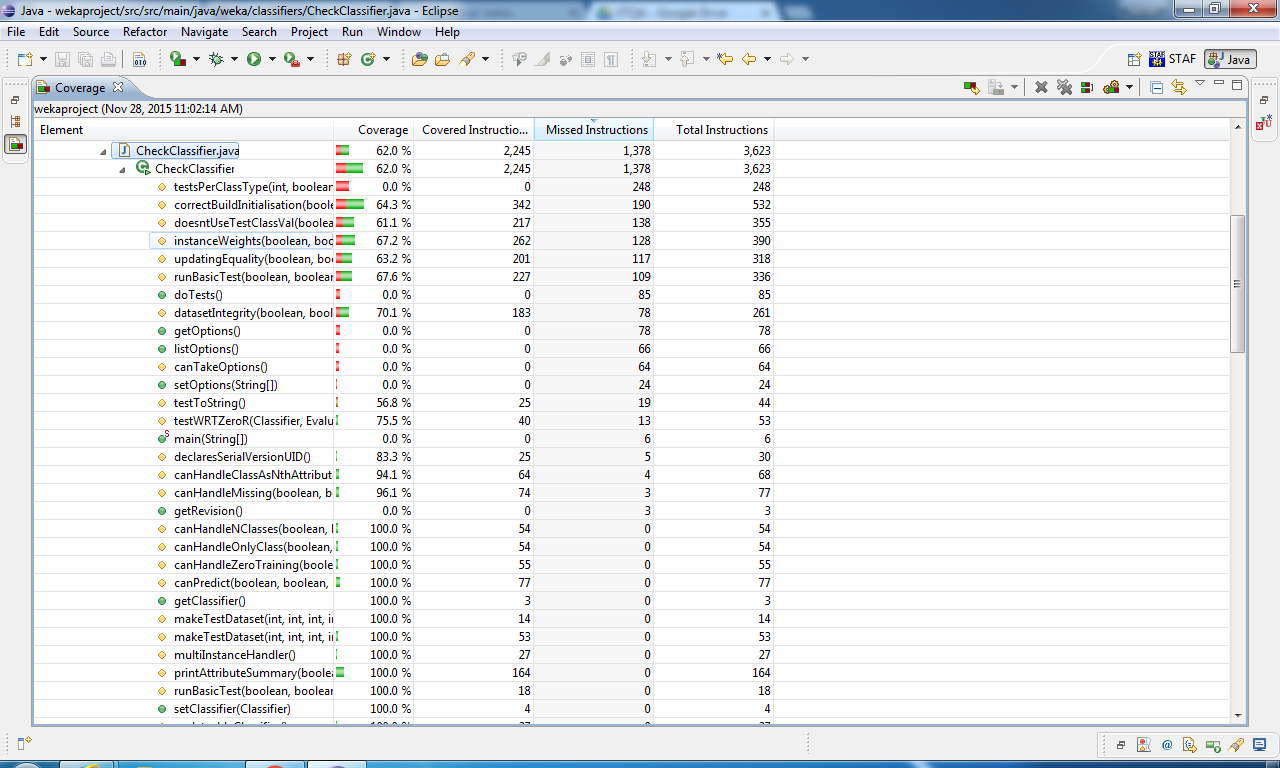
*IntervalEstimator*

*IterativeClassifier*

*Sourcable*

*UpdateableClassifier*

Code coverage for CheckClassifiers is 62%. CheckClassifier is a test class created for examining the capabilities and finding problems with classifiers. It is created to ensure robustness and correct operation. This class contains main logic of the program and has methods and function which implements classification verifying data type of the attributes of input data. While analyzing code coverage Emma, excludes lines of code written for exceptions i.e., for try and catch which is approximately more than 400 lines of code. Hence considering this, code coverage for this class if substantially good that out of 3623 instructions 2245 instructions are covered and validated. Also there is usage of overridden methods which are not calling the super class methods and as such it’s not being covered in the tests.



We have generated EclEmma report as JArchitect didn’t provide coverage for our project.