

# **FlexDM - Flexible Data Mining with WEKA**

## **User Manual**

Main developer: Madison Flannery<sup>1</sup>

Collaborators: David Budden<sup>2</sup> and Alexandre Mendes<sup>3</sup>

<sup>1</sup> Melbourne Graduate School of Science, University of Melbourne, Melbourne, VIC, Australia

[mflannery@student.unimelb.edu.au](mailto:mflannery@student.unimelb.edu.au)

<sup>2</sup>Systems Biology Laboratory, University of Melbourne, Melbourne, VIC, Australia

[dbudden@student.unimelb.edu.au](mailto:dbudden@student.unimelb.edu.au)

<sup>3</sup> School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW, Australia.

[alexandre.mendes@newcastle.edu.au](mailto:alexandre.mendes@newcastle.edu.au)

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Dependencies . . . . .	2
1.2	Running FlexDM . . . . .	2
1.3	System Input: Creating an XML File . . . . .	2
1.3.1	XML File Format . . . . .	3
1.3.2	XML Tags & Attributes . . . . .	3
1.3.3	Setting a Range Parameter . . . . .	4
1.4	System Output: Finding the Results . . . . .	5
1.5	Error Reporting . . . . .	5
<b>2</b>	<b>System Design</b>	<b>5</b>
<b>3</b>	<b>Examples</b>	<b>6</b>
3.1	Example 1 . . . . .	6
3.2	Example 2 . . . . .	9

# 1 Introduction

## Getting Started

### 1.1 Dependencies

FlexDM requires the Java Runtime Environment (JRE) version 7 or above, which can be downloaded from <http://www.java.com/en/download/index.jsp>

### 1.2 Running FlexDM

- Download and unzip FlexDM.zip from <http://sourceforge.net/projects/flexdm/>.
- Create an XML file using the specifications described in Section 1.3.1. A template is available at <http://sourceforge.net/projects/flexdm/files/>
- Using command prompt (Windows) or terminal (Ubuntu/Mac), cd to the FlexDM directory.
- Run FlexDM by using the following command:  
`java -jar FlexDM.jar <Name of XML file>`
- The results will be stored in a Results subdirectory, within the FlexDM directory.

### 1.3 System Input: Creating an XML File

The input to the system consists of a single XML file.

### 1.3.1 XML File Format

```
<!DOCTYPE flexdm SYSTEM "flexdm.dtd">
<flexdm>
  <dataset name="" test="" results="">
    <classifier name="">
      <parameter name="" value="" />
    </classifier>
  </dataset>
</flexdm>
```

### 1.3.2 XML Tags & Attributes

- *<flexdm>*: is the container for the entire experiment.
- *<dataset>*: stores information about a dataset. Contains one or more classifier elements to be trained on this dataset. This element has three required attributes:
  - *name*: the absolute file path to your file, inclusive of the file name itself.
  - *test*: the evaluation method to be used for the classifiers. Options are as follows:
    - \* *"training"*: Use the training set to evaluate the classifier.
    - \* *"test <dataset>"*: Use a test set to evaluate the classifier. *<dataset>* is the name of the dataset containing the test data.
    - \* *"xval <numfolds>"*: Use cross validation to evaluate the classifier. *<numfolds>* is the number of folds for the cross validation.
    - \* *"percent <percentage>"*: Use a percentage split to create a train and test set from a single dataset. *<percentage>* is the percentage of the dataset which should be used for training, expressed as a decimal in the range [0, 1].
    - \* *"leavexval"*: Use leave-one-out cross validation to evaluate the classifier.
  - *results*: the results to print to each file. Options are as follows:

- \* *model*: Stores the classification model on the full training set.
  - \* *stats*: Stores the precision/recall and true/false statistics for each class.
  - \* *entropy*: Stores the entropy evaluation measures.
  - \* *predictions*: Stores the predictions on the testing data. *If used with cross-validation, the sample numbers do not correspond to the location in the data.*
  - \* *matrix*: Stores the confusion matrix of the classifier's prediction.
- *<classifier>*: contains information about a classifier. Contains zero or more parameter elements. This element has a single attribute, *name*, which should include the full WEKA classifier path for the classifier.  
For example, for a J48 classifier: `<classifier name="weka.classifiers.trees.j48">`.  
For information about the list of classifiers available, their valid options and default values, we refer the reader to the online documentation:  
<http://weka.sourceforge.net/doc.dev/index.html?weka/classifiers/Classifier.html>
  - *<parameter>* is an optional element, containing information about a hyperparameter for a classifier. If no hyperparameters are specified the default values are used. A parameter element has two attributes:
    - *name*: The name of the parameter, identical to the name you would provide in the WEKA command line interface (CLI). I.e. for a confidence factor, the name would be `"-C"`
    - *value*: This attribute is optional. You can also set a range of values here; see Section 1.3.3

### 1.3.3 Setting a Range Parameter

A range may be specified inside the value attribute of a single parameter. A number of separate classifiers will be created using each parameter value (or combinations of values). These values can be declared in 3 ways:

- *Comma separated values*: Two or more comma separated values, i.e.  $[a, b, c, d]$
- *Range, step size not specified*: Specified as  $[a, c]$ , where  $a$  is the starting value,  $c$  is the final value, and the step size is 1.

- *Range, step value specified*: Specified as  $[a, b, c]$ , where  $a$  is the starting value,  $c$  is the final value, and  $b$  is the step size.

The following three entries are equivalent:  $[1, 2, 3, 4, 5]$ ;  $[1 : 5]$ ;  $[1 : 1 : 5]$ .

## 1.4 System Output: Finding the Results

A *summary file* is created, containing details about each classifier and the accuracy that was obtained. For each classifier, the specified results are stored in .txt files, saved in the “*Results*” subdirectory (within the FlexDM directory). Each subdirectory is named using the dataset & test method, and each classifier has its own directory within that.

## 1.5 Error Reporting

If an error occurs, it will only affect a single classifier. These errors will be shown in the program output and will list the classifier name, dataset name, parameters used, and a specific error message detailing what occurred. No results will be output for that classifier.

# 2 System Design

The system was designed using a modular, pipelined approach (shown in Figure 1) and consists of three main stages:

- *The XML parser*: Takes a single XML file and extracts the data using a parser based on the Java SAX API. The output from this module is a collection of objects representing data sets, classifiers and parameters.
- *The parameter range processor*: Takes the objects given by the XML parser and detects any parameter values containing a range. Once these ranges have been identified, they are processed into individual individual values, placed into jobs, and each job is placed into a queue.
- *The batch processor*: Takes the job queue and processes the execution of the jobs by scheduling threads to utilise the highest amount of computational resources available.



Figure 1: Flowchart of the WEKA batch processor solution.

The execution of each job involves creating the classifier, processing the test options, training the classifier, and then evaluating the classifier. The results from the evaluation are stored in an individual .txt file.

## 3 Examples

In this section, we will present some examples of FlexDM usage.

### 3.1 Example 1

Firstly, the XML input file must be created. Consider the following XML file, available from <http://sourceforge.net/projects/flexdm/files/Example%20Files/>:

```

<!DOCTYPE flexdm SYSTEM "flexdm.dtd">
<flexdm>
  <dataset name="weather.arff" test="training"
    results="predictions,model,matrix,stats,entropy">
    <classifier name="weka.classifiers.trees.J48">
    </classifier>
  </dataset>
</flexdm>

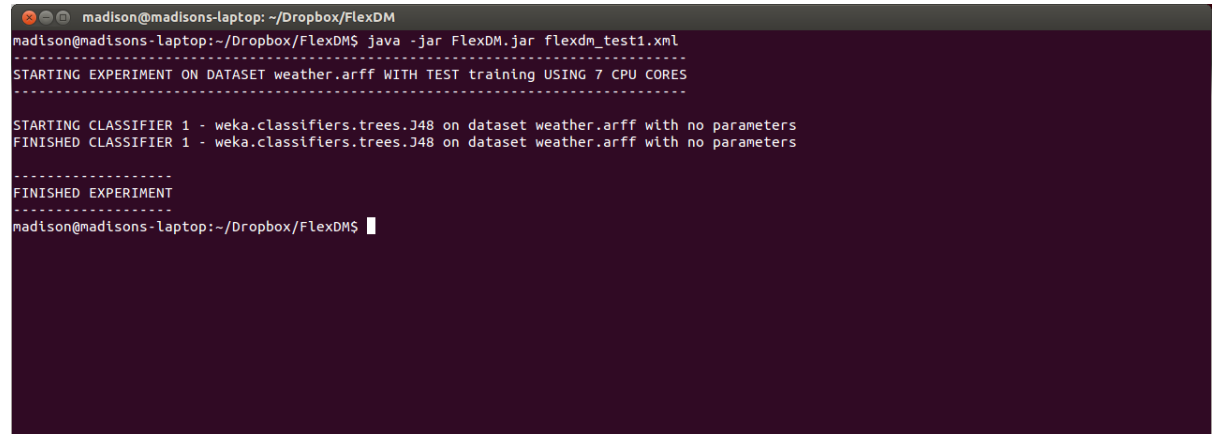
```

In the XML file above, the dataset `weather.arff` [1] (available from <http://sourceforge.net/projects/flexdm/files/Example%20Files/>) and the classifier `weka.classifiers.trees.J48` with default parameters are used together with the training test option and all available results options.

Run FlexDM by firstly using `cd` to change to the FlexDM directory, and then using the following command:

```
java -jar FlexDM.jar flexdm_test1.xml
```

The output from FlexDM following this is shown in Figure 2.

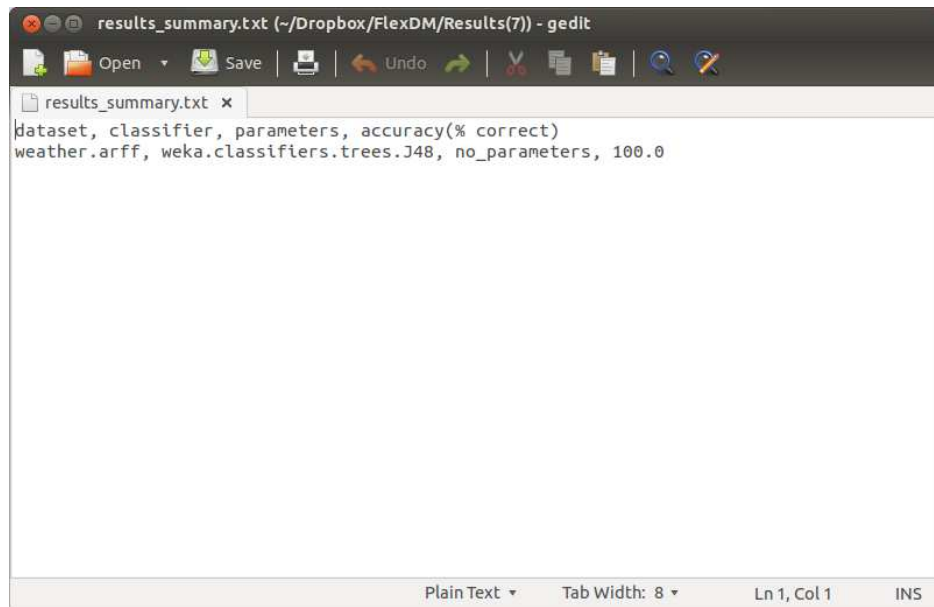
A terminal window with a dark purple background and white text. The window title is 'madison@madisons-laptop: ~/Dropbox/FlexDM'. The command 'java -jar FlexDM.jar flexdm\_test1.xml' has been executed. The output shows the start of an experiment on the 'weather.arff' dataset using 7 CPU cores. It lists classifier 1 as 'weka.classifiers.trees.J48' on the dataset with no parameters. The experiment is finished.

```
madison@madisons-laptop: ~/Dropbox/FlexDM
madison@madisons-laptop:~/Dropbox/FlexDM$ java -jar FlexDM.jar flexdm_test1.xml
-----
STARTING EXPERIMENT ON DATASET weather.arff WITH TEST training USING 7 CPU CORES
-----
STARTING CLASSIFIER 1 - weka.classifiers.trees.J48 on dataset weather.arff with no parameters
FINISHED CLASSIFIER 1 - weka.classifiers.trees.J48 on dataset weather.arff with no parameters
-----
FINISHED EXPERIMENT
-----
madison@madisons-laptop:~/Dropbox/FlexDM$
```

Figure 2: The output from FlexDM when the program is run with the XML file above. Generated using FlexDM on Ubuntu 13.04

After processing, the summary file can be found in the Results directory, and looks as shown in Figure 3. The results that would normally go to WEKA's output window are stored in a text file, shown in Figure 4. This file can be found by firstly entering the Results-weather.arff-training folder, and then the weka.classifiers.trees.J48 folder.





The image shows a screenshot of a gedit text editor window. The title bar reads "results\_summary.txt (~/.Dropbox/FlexDM/Results(7)) - gedit". The menu bar includes "Open", "Save", "Undo", and other standard editing functions. The text area contains the following content:

```
dataset, classifier, parameters, accuracy(% correct)
weather.arff, weka.classifiers.trees.J48, no_parameters, 100.0
```

The status bar at the bottom indicates "Plain Text", "Tab Width: 8", "Ln 1, Col 1", and "INS".

Figure 3: The summary file obtained from the J48 classifier run on the weather.arff training set, using the training test method.

```

results_no_parameters.txt x
CLASSIFIER: weka.classifiers.trees.J48
DATASET: weather.arff
PARAMETERS: no_parameters

Correctly Classified Instances      14      100 %
Incorrectly Classified Instances    0       0 %
Kappa statistic                    1
Mean absolute error                 0
Root mean squared error            0
Relative absolute error             0 %
Root relative squared error        0 %
Total Number of Instances         14

=== Detailed Accuracy By Class ===

```

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	1	yes
	1	0	1	1	1	1	no
Weighted Avg.	1	0	1	1	1	1	

```

J48 pruned tree
-----
outlook = sunny
|  humidity <= 75: yes (2.0)
|  humidity > 75: no (3.0)
outlook = overcast: yes (4.0)
outlook = rainy
|  windy = TRUE: no (2.0)
|  windy = FALSE: yes (3.0)

Number of Leaves :      5

```

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 INS

Figure 4: The results file obtained from the J48 classifier run on the weather.arff training set, using the training test method.

## 3.2 Example 2

Firstly, the XML input file must be created. Consider the following XML file, available from <http://sourceforge.net/projects/flexdm/files/Example%20Files/>:

```

<!DOCTYPE flexdm SYSTEM "flexdm.dtd">
<flexdm>
  <dataset name="weather.arff" test="leavexval" results="matrix,stats">
    <classifier name="weka.classifiers.trees.J48">
      <parameter name="-M" value="[1,2]" />
      <parameter name="-C" value="[0.1:0.1:0.5]" />
    </classifier>
    <classifier name="weka.classifiers.rules.PART">
      <parameter name="-C" value="0.5" />
    </classifier>
  </dataset>
</flexdm>

```

In the XML file above, the dataset `weather.arff` [1] (available from <http://sourceforge.net/projects/flexdm/files/Example%20Files/>) and the classifiers `weka.classifiers.trees.J48` and `weka.classifiers.rules.PART` are used, together with the leave-one-out cross-validation test option. The result file for each classifier will contain the confusion matrix and classifier statistics. The J48 classifier has the `-M` (minimum number of instances per leaf) hyperparameter set to 1 and 2, and the `-C` hyperparameter (confidence value) set to the values 0.1 to 0.5 with increments of 0.1. A single test will be performed for each possible hyperparameter combination. The PART classifier just has its `-C` (confidence value) hyperparameter set to 0.5.

Run FlexDM by firstly using `cd` to change to the FlexDM directory, and then using the following command:

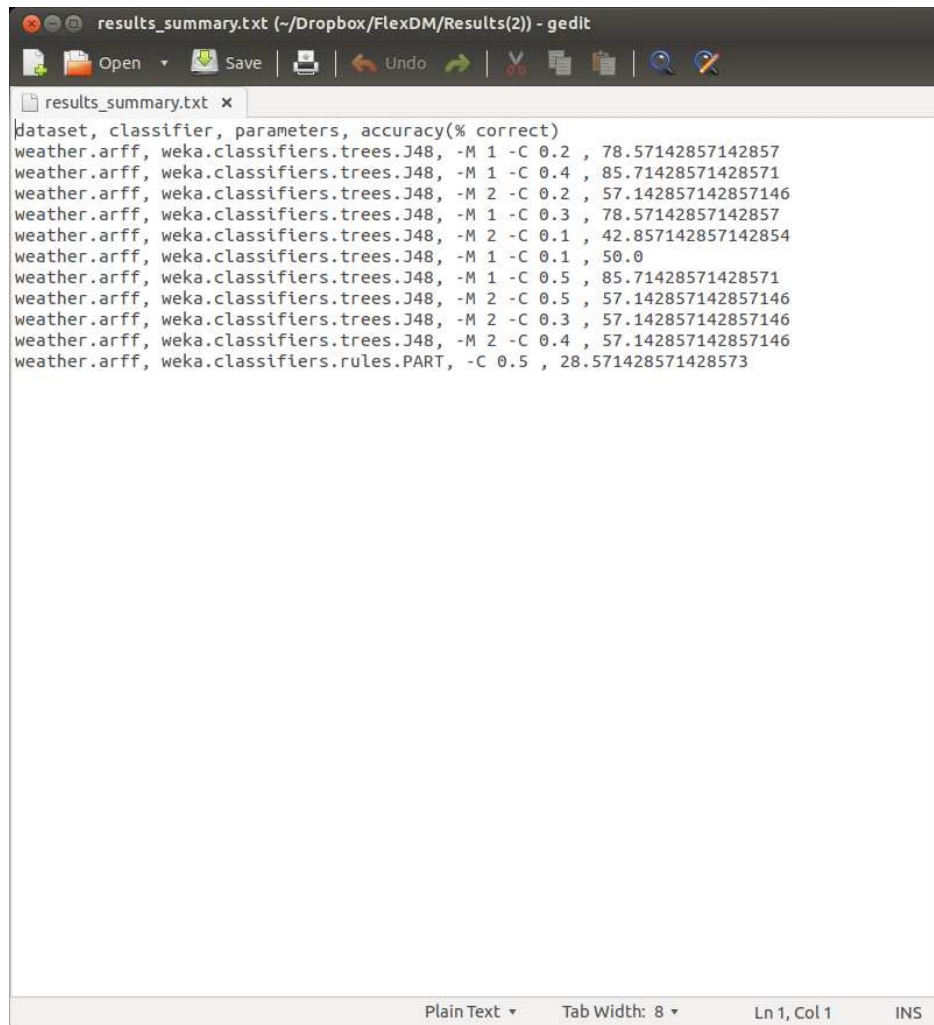
```
java -jar FlexDM.jar flexdm_test2.xml
```

The output from FlexDM following this is shown in Figure 5.

```
madison@madisons-laptop: ~/Dropbox/FlexDM
madison@madisons-laptop:~$ cd Dropbox/FlexDM
madison@madisons-laptop:~/Dropbox/FlexDM$ java -jar FlexDM.jar flexdm_test2.xml
-----
STARTING EXPERIMENT ON DATASET weather.arff WITH TEST leavexval USING 7 CPU CORES
-----
STARTING CLASSIFIER 1 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 1 -C 0.1
STARTING CLASSIFIER 2 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 1 -C 0.2
STARTING CLASSIFIER 3 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 1 -C 0.3
STARTING CLASSIFIER 4 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 1 -C 0.4
STARTING CLASSIFIER 5 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 1 -C 0.5
STARTING CLASSIFIER 6 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 2 -C 0.1
STARTING CLASSIFIER 7 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 2 -C 0.2
FINISHED CLASSIFIER 1 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 1 -C 0.1
STARTING CLASSIFIER 8 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 2 -C 0.3
FINISHED CLASSIFIER 5 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 1 -C 0.5
FINISHED CLASSIFIER 3 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 1 -C 0.3
FINISHED CLASSIFIER 2 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 1 -C 0.2
STARTING CLASSIFIER 9 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 2 -C 0.4
STARTING CLASSIFIER 10 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 2 -C 0.5
FINISHED CLASSIFIER 7 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 2 -C 0.2
FINISHED CLASSIFIER 6 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 2 -C 0.1
FINISHED CLASSIFIER 4 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 1 -C 0.4
STARTING CLASSIFIER 11 - weka.classifiers.rules.PART on dataset weather.arff with parameters -C 0.5
FINISHED CLASSIFIER 8 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 2 -C 0.3
FINISHED CLASSIFIER 9 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 2 -C 0.4
FINISHED CLASSIFIER 10 - weka.classifiers.trees.J48 on dataset weather.arff with parameters -M 2 -C 0.5
FINISHED CLASSIFIER 11 - weka.classifiers.rules.PART on dataset weather.arff with parameters -C 0.5
-----
FINISHED EXPERIMENT
-----
madison@madisons-laptop:~/Dropbox/FlexDM$
```

Figure 5: The output from FlexDM when the program is run with the XML file above.

After processing, the summary file can be found in the Results directory, and looks as shown in Figure 6. The results that would normally go to WEKA's output window are stored in a text file for each classifier. An example of a results file is shown in Figure 7. These files can be found by firstly entering the Results-weather.arff-leavexval folder, and then the weka.classifiers.trees.J48 or weka.classifiers.rules.PART folder.



The screenshot shows a gedit text editor window titled "results\_summary.txt (~/.Dropbox/FlexDM/Results(2)) - gedit". The window contains a table of results for the weather.arff dataset, comparing the J48 and PART classifiers across various parameters and cross-validation folds. The table has four columns: dataset, classifier, parameters, and accuracy(% correct). The data is as follows:

dataset	classifier	parameters	accuracy(% correct)
weather.arff	weka.classifiers.trees.J48	-M 1 -C 0.2	78.57142857142857
weather.arff	weka.classifiers.trees.J48	-M 1 -C 0.4	85.71428571428571
weather.arff	weka.classifiers.trees.J48	-M 2 -C 0.2	57.142857142857146
weather.arff	weka.classifiers.trees.J48	-M 1 -C 0.3	78.57142857142857
weather.arff	weka.classifiers.trees.J48	-M 2 -C 0.1	42.857142857142854
weather.arff	weka.classifiers.trees.J48	-M 1 -C 0.1	50.0
weather.arff	weka.classifiers.trees.J48	-M 1 -C 0.5	85.71428571428571
weather.arff	weka.classifiers.trees.J48	-M 2 -C 0.5	57.142857142857146
weather.arff	weka.classifiers.trees.J48	-M 2 -C 0.3	57.142857142857146
weather.arff	weka.classifiers.trees.J48	-M 2 -C 0.4	57.142857142857146
weather.arff	weka.classifiers.rules.PART	-C 0.5	28.571428571428573

The status bar at the bottom indicates "Plain Text", "Tab Width: 8", "Ln 1, Col 1", and "INS".

Figure 6: The summary file obtained from the J48 and PART classifiers run on the weather.arff training set, using the leave-one-out cross-validation test method.

```
-M 2 -C 0.1.txt x
CLASSIFIER: weka.classifiers.trees.J48
DATASET: weather.arff
PARAMETERS: -M 2 -C 0.1

Correctly Classified Instances      6      42.8571 %
Incorrectly Classified Instances    8      57.1429 %
Kappa statistic                    -0.3659
Mean absolute error                 0.424
Root mean squared error             0.5661
Relative absolute error             86.4451 %
Root relative squared error        111.6587 %
Total Number of Instances          14

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.667    1.000    0.545    0.667    0.6        0.6      yes
      0.000    0.333    0.000    0.000    0.0        0.6      no
Weighted Avg.   0.429    0.762    0.351    0.429    0.386     0.6

=== Confusion Matrix ===

a b  <-- classified as
6 3 | a = yes
5 0 | b = no
```

Figure 7: The results file obtained from the J48 classifier with hyperparameters -C 0.1 and -M 2, run on the weather.arff training set, using the leave-one-out cross-validation test method.

## References

- [1] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

## Appendix A: Test dataset

### weather.arff

@relation weather

@attribute outlook sunny, overcast, rainy

@attribute temperature real

@attribute humidity real

@attribute windy TRUE, FALSE

@attribute play yes, no

@data

sunny,85,85,FALSE,no

sunny,80,90,TRUE,no

overcast,83,86,FALSE,yes

rainy,70,96,FALSE,yes

rainy,68,80,FALSE,yes

rainy,65,70,TRUE,no

overcast,64,65,TRUE,yes

sunny,72,95,FALSE,no

sunny,69,70,FALSE,yes

rainy,75,80,FALSE,yes

sunny,75,70,TRUE,yes

overcast,72,90,TRUE,yes

overcast,81,75,FALSE,yes

rainy,71,91,TRUE,no