Data Mining III – Lesson 7

Tamara B. Sipes, Ph.D.

Last Time: Lesson 6

- Real life datasets are unclean and often difficult to model, therefore we need help by combining these weak classifiers
- Ensemble Modeling or meta learning combines weak classifiers. We learned:
  - Why and How
  - Common methods
- Hands on experience with ensemble modeling in weka:
  - Bagging
  - Boosting
  - Stacking

Lesson 6: The Lesson Learned

- Real life datasets are unclean, complex and often difficult to model
- Sometimes you just need to wait for more data, or ask for a different protocol for collecting the data, more input variables, etc.
- Sometimes using a meta learner can help solve the problem
- We built up your experience and intuition a little bit more

Lesson 7 Overview

- Lots of paths to traverse in the search for “the best” model – Experimenter is designed to help with this effort
- How to use the Experimenter + hands-on example
- Summary of what we learned in this class
- Data Mining Tips and Useful Guidance
- Conclusions
- What’s Next?

Instructions

- Hands-on lesson #7
- First part: Using the Experimenter (please be ready to have both the Lesson 7 and the weka Experimenter open)
- Follow the step by step instructions, and perform the modeling of the dataset yourself as well
- Second part: Tips and Useful Guidance
- Third part: Conclusion and Summary

- Let's get started!

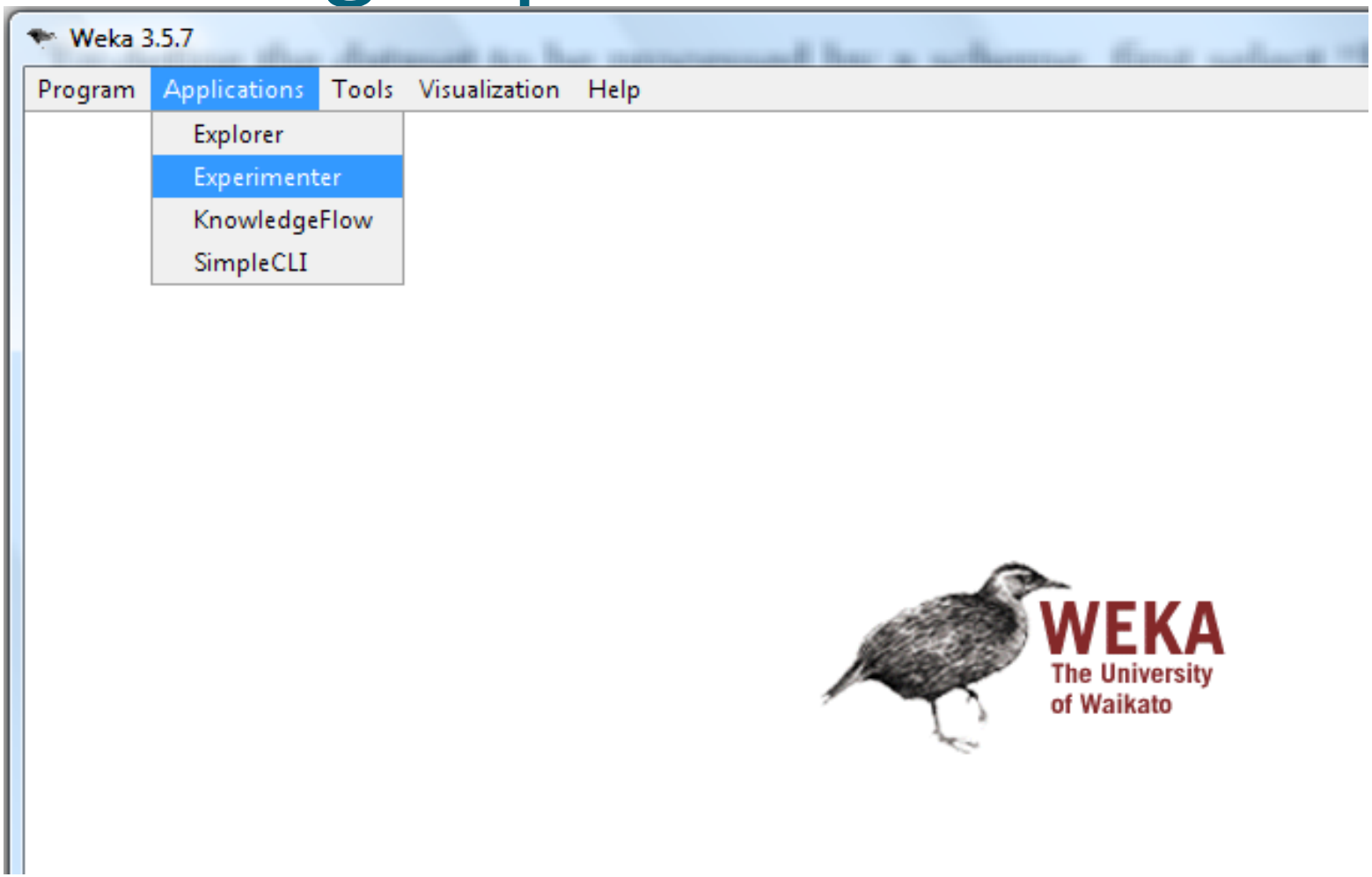
Weka's Experimenter

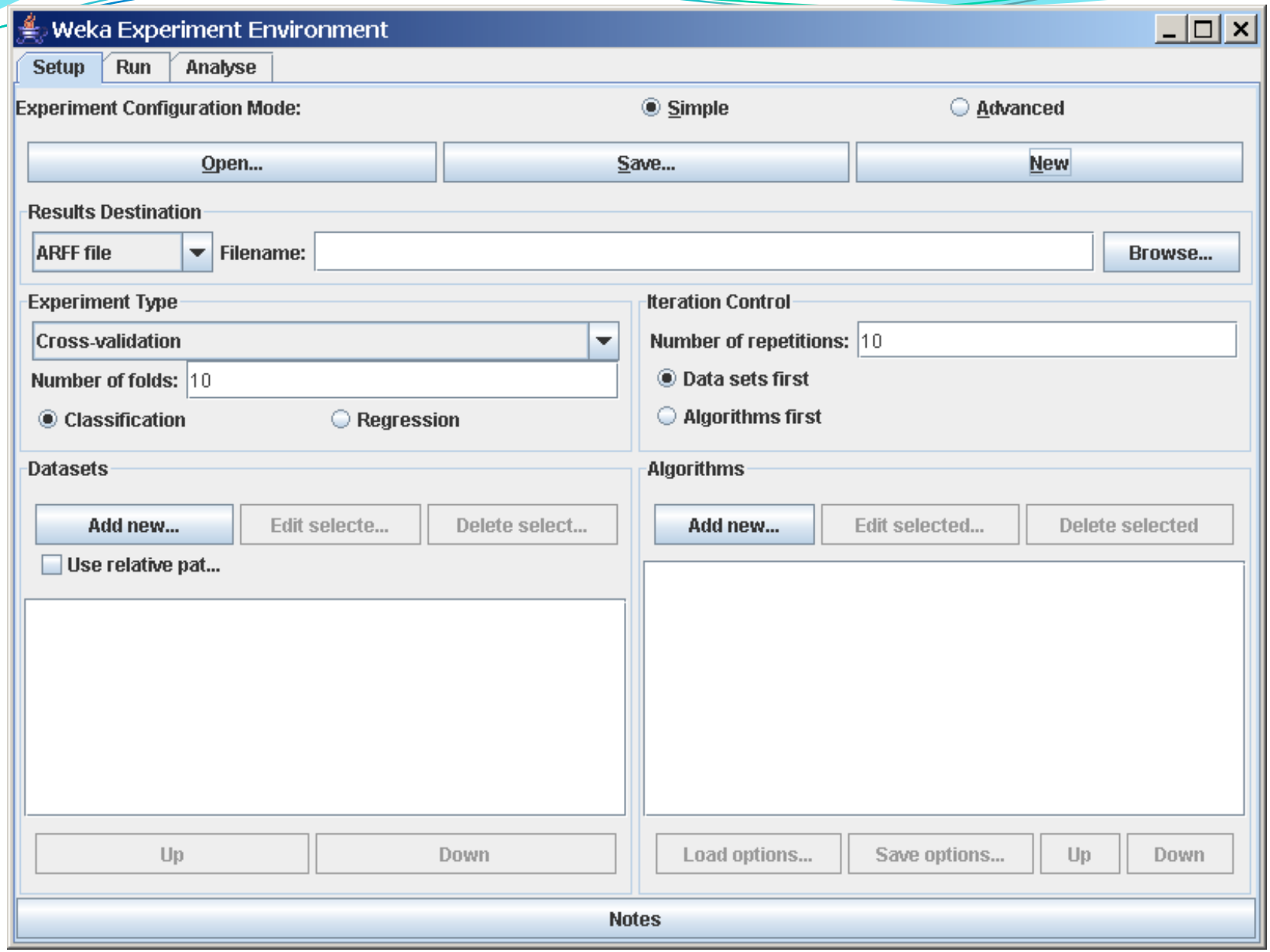
- The Weka Experiment Environment enables the user to create, run, modify, and analyze experiments in a more convenient manner than is possible when processing the schemes individually
- For example, the user can create an experiment that runs several schemes against a series of datasets and then analyze the results to determine if one of the schemes is (statistically) better than the other schemes

Two Flavors

- The Experimenter comes in two flavors:
  - **Simple** - provides most of the functionality one needs for experiments
  - **Advanced** - full access to the Experimenter's capabilities
- Both setups allow you to setup standard experiments, that are run locally on a single machine, or remote experiments, which are distributed between several hosts (which cuts down the time the experiments will take until completion, but the setup takes more time)

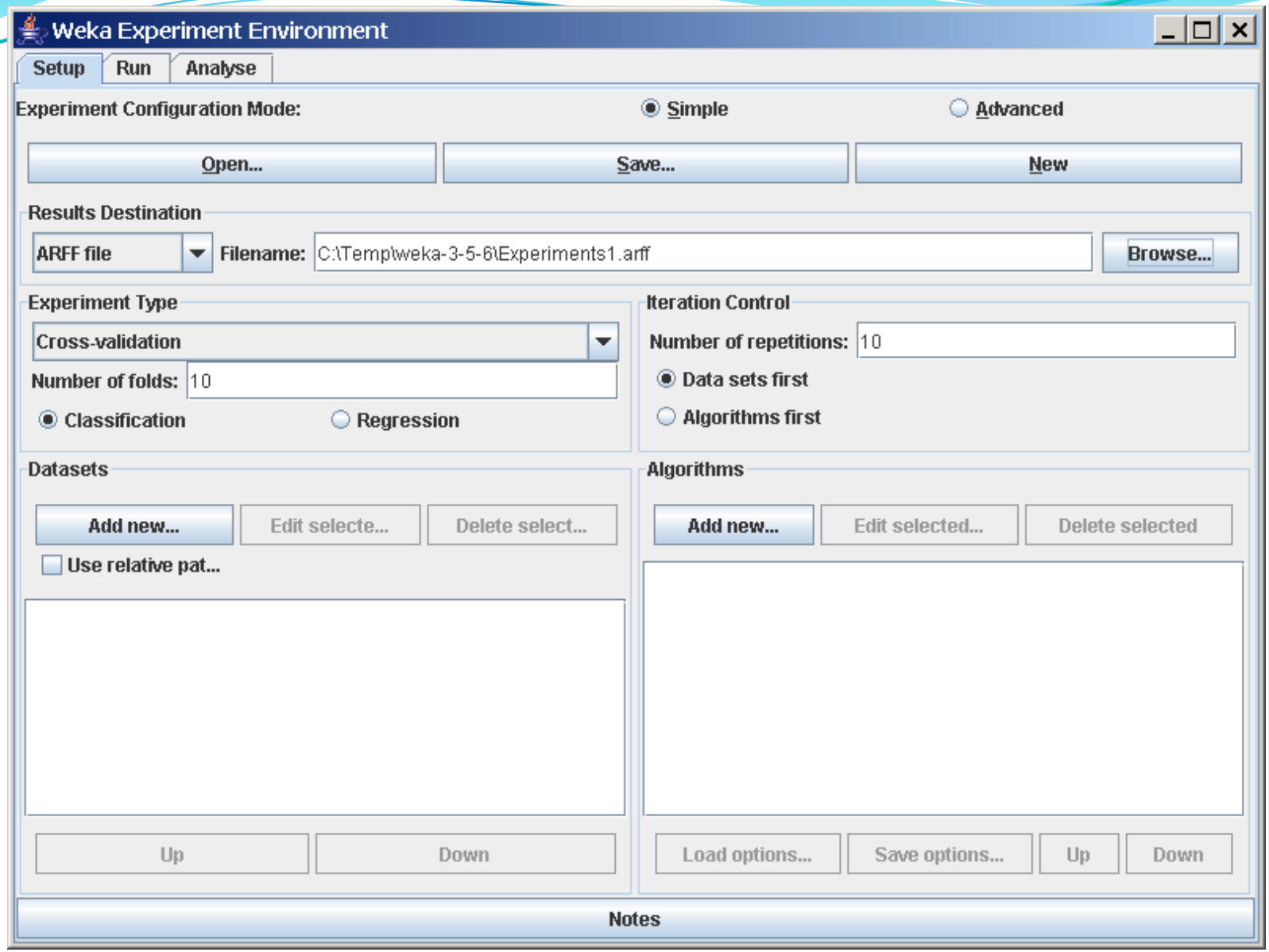
Running Experimenter




Weka Experiment Environment
Setup
Run
Analyse
Experiment Configuration Mode:
Simple
Advanced
Open...
Save...
New
Results Destination
ARFF file
Filename:
Browse...
Experiment Type
Cross-validation
Number of folds:
10
Classification
Regression
Iteration Control
Number of repetitions:
10
Data sets first
Algorithms first
Datasets
Add new...
Edit selecte...
Delete select...
Use relative pat...
Up
Down
Algorithms
Add new...
Edit selected...
Delete selected
Load options...
Save options...
Up
Down
Notes

New Experiment

- New experiment
- After clicking “New” default parameters for an Experiment are defined
- Results destination
- By default, an ARFF file is the destination for the results output. But you can choose between:
  - ARFF file
  - CSV file
  - JDBC database


Weka Experiment Environment
Setup
Run
Analyse
Experiment Configuration Mode:
Simple
Advanced
Open...
Save...
New
Results Destination
ARFF file
Filename:
C:\Temp\weka-3-5-6\Experiments1.arff
Browse...
Experiment Type
Cross-validation
Number of folds:
10
Classification
Regression
Iteration Control
Number of repetitions:
10
Data sets first
Algorithms first
Datasets
Add new...
Edit selecte...
Delete select...
Use relative pat...
Up
Down
Algorithms
Add new...
Edit selected...
Delete selected
Load options...
Save options...
Up
Down
Notes

Experiment Type

- Cross-validation (default)

  performs stratified cross-validation with the given number of folds
- Train/Test Percentage Split (data randomized)

  splits a dataset according to the given percentage into a train and a test file (one cannot specify explicit training and test files in the Experimenter), after the order of the data has been randomized and stratified
- Train/Test Percentage Split (order preserved)

Classification/Regression

- Additionally, one can choose between Classification and Regression, depending on the datasets and classifiers one uses
- For decision trees like J48 (Weka's implementation of Quinlan's C4.5) and the iris dataset, Classification is necessary, for a numeric classifier like M5P, on the other hand, Regression is used
- Classification is selected by default

Open Dataset

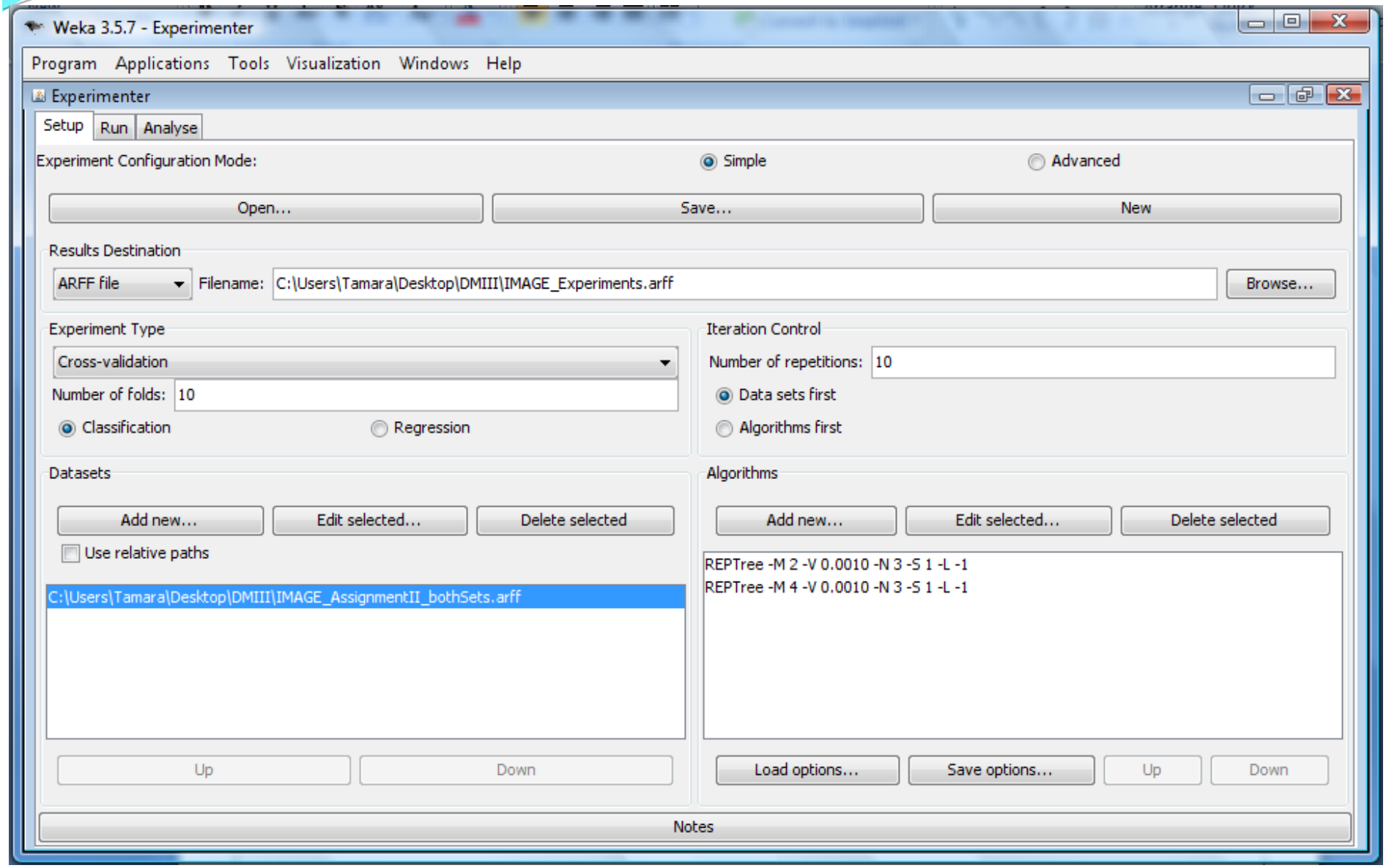
- One can add dataset files either with an absolute path or with a relative one
- Absolute path makes it often easier to run experiments on different machines
- Use relative paths, before clicking on Add new….
- Click on Add new: IMAGE_AssignmentII_BothSets.arff

Iteration Control

- **Number of repetitions**
- In order to get statistically meaningful results, the default number of iterations is 10. In case of 10-fold cross-validation this means 100 calls of one classifier with training data and tested against test data.
- **Data sets first/Algorithms first**
- As soon as one has more than one dataset and algorithm, it can be useful to switch from datasets being iterated over first then to algorithms. This is the case if one stores the results in a database and wants to complete the results for all the datasets for one algorithm as early as possible.

Algorithms

- New algorithms can be added via the Add new… button
- Opening this dialog for the first time, ZeroR is presented, otherwise the one that was selected last.
- With the Choose button one can open the GenericObjectEditor and choose another classifier
- Multiple versions of one method are able to be used!


Weka 3.5.7 - Experimenter
Program Applications Tools Visualization Windows Help
Experimenter
Setup Run Analyse
Experiment Configuration Mode:
Simple
Advanced
Open...
Save...
New
Results Destination
ARFF file
Filename: C:\Users\Tamara\Desktop\DMIII\IMAGE_Experiments.arff
Browse...
Experiment Type
Cross-validation
Number of folds: 10
Classification
Regression
Iteration Control
Number of repetitions: 10
Data sets first
Algorithms first
Datasets
Add new...
Edit selected...
Delete selected
Use relative paths
C:\Users\Tamara\Desktop\DMIII\IMAGE_AssignmentII_bothSets.arff
Up
Down
Algorithms
Add new...
Edit selected...
Delete selected
REPTree -M 2 -V 0.0010 -N 3 -S 1 -L -1
REPTree -M 4 -V 0.0010 -N 3 -S 1 -L -1
Load options...
Save options...
Up
Down
Notes

Loading/Saving the Options

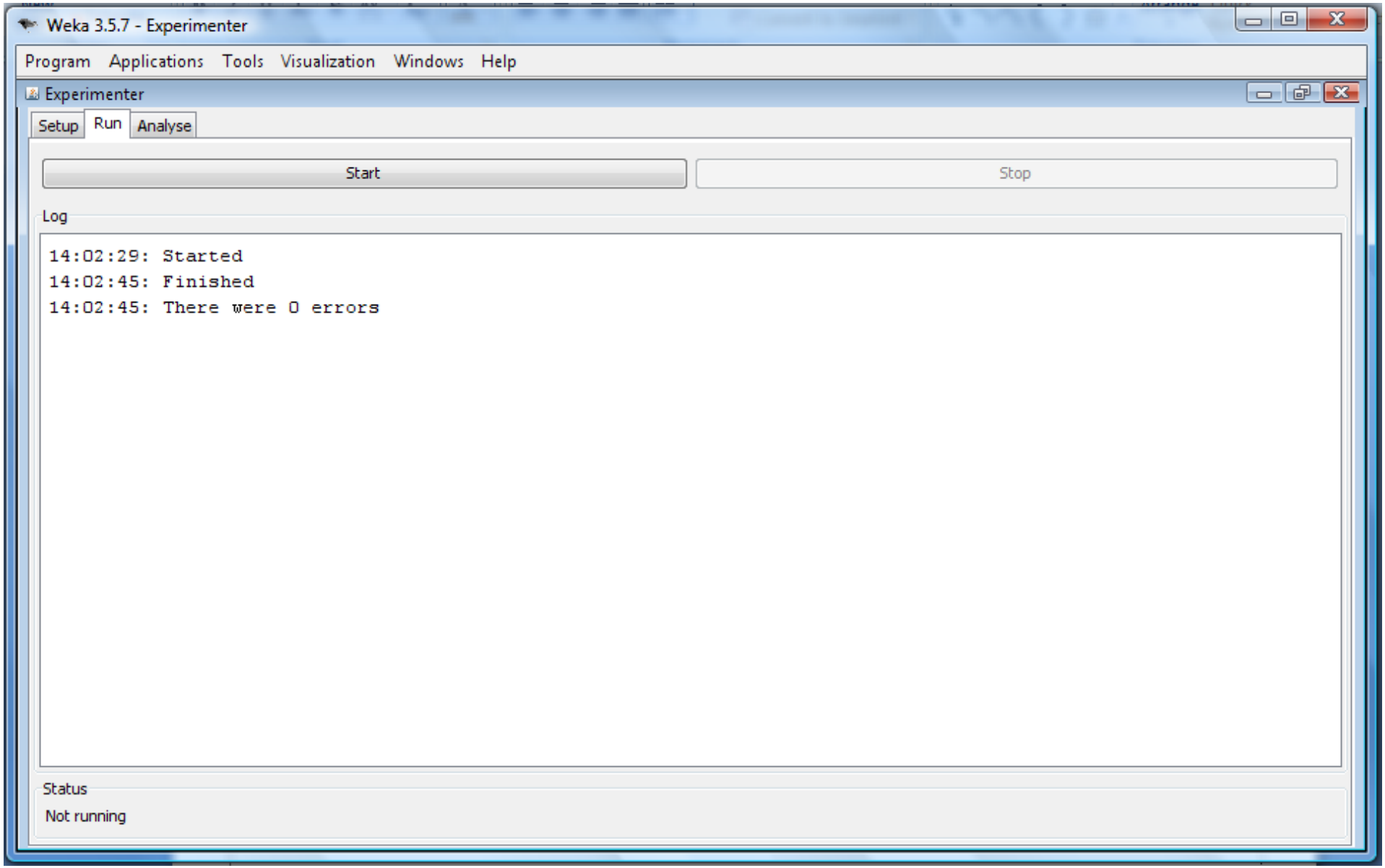
- With the 'Load options...' and 'Save options...' buttons one can load and save the setup of a selected classifier from and to XML.
- This is especially useful for highly configured classifiers (e.g., nested meta-classifiers), where the manual setup takes quite some time, and which are used often.
- One can also paste classifier settings here by right-clicking (or Alt-Shift-left-clicking) and selecting the appropriate menu point from the popup menu, to either add a new classifier or replace the selected one with a new setup.
- This is very useful for transferring a classifier setup from the Weka Explorer over to the Experimenter without having to setup the classifier from scratch.

Saving the Setup

- For future re-use, one can save the current setup of the experiment to a file by clicking on Save… at the top of the window.
- By default, the format of the experiment files is the binary format that Java serialization offers.
- The drawback of this format is the possible incompatibility between different versions of Weka.
- A more robust alternative to the binary format is the XML format.
- Previously saved experiments can be loaded again via the Open… button.

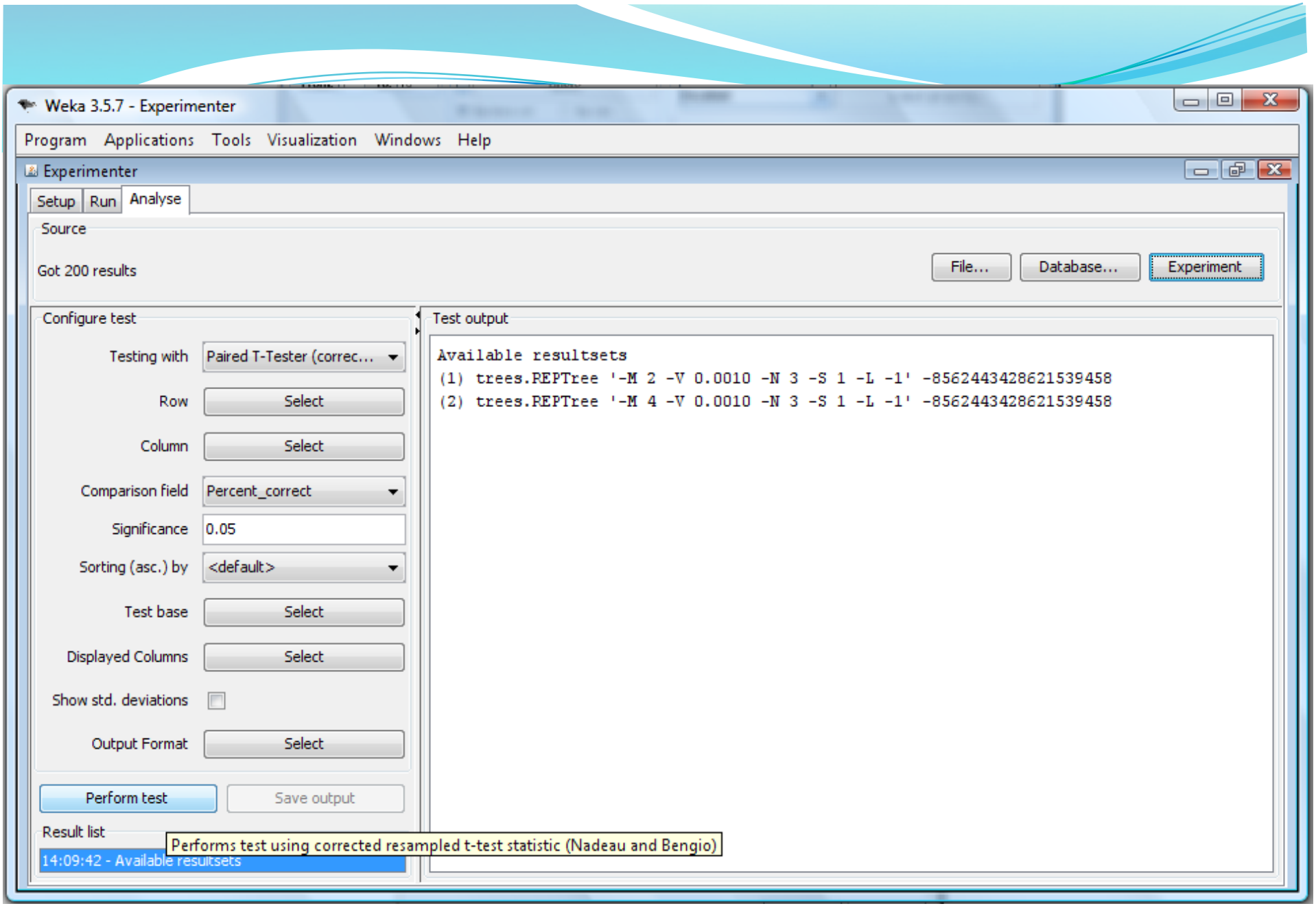
Running the Experiment

- Save as Exp1.arff
- To run the current experiment, click the Run tab at the top of the Experiment Environment window.
- The current experiment performs 10 runs of 10-fold stratified cross-validation on the IMAGE dataset using the two RepTree shemas.


Weka 3.5.7 - Experimenter
Program Applications Tools Visualization Windows Help
Experimenter
Setup Run Analyse
Start
Stop
Log
14:02:29: Started
14:02:45: Finished
14:02:45: There were 0 errors
Status
Not running

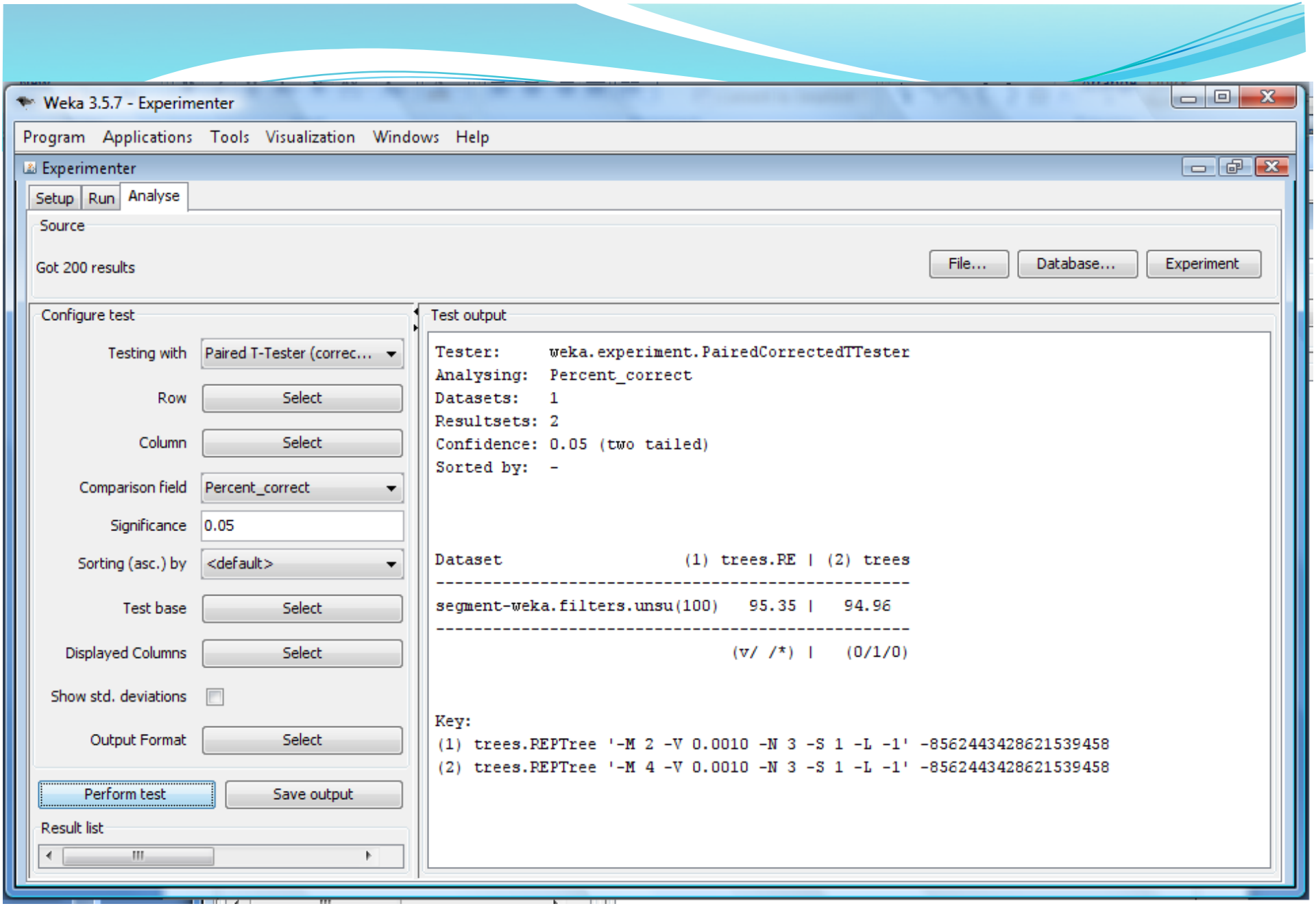
Analyze Results

- Compare different schema results using significance tests
- Click on Analyze Tab
- Click on Source: Experiment
- You get available results tests


Weka 3.5.7 - Experimenter
Program Applications Tools Visualization Windows Help
Experimenter
Setup Run Analyse
Source
Got 200 results
File... Database... Experiment
Configure test
Testing with Paired T-Tester (correc...
Row Select
Column Select
Comparison field Percent_correct
Significance 0.05
Sorting (asc.) by <default>
Test base Select
Displayed Columns Select
Show std. deviations
Output Format Select
Perform test Save output
Result list
14:09:42 - Available resultsets
Performs test using corrected resampled t-test statistic (Nadeau and Bengio)
Test output
Available resultsets
(1) trees.REPTree '-M 2 -V 0.0010 -N 3 -S 1 -L -1' -8562443428621539458
(2) trees.REPTree '-M 4 -V 0.0010 -N 3 -S 1 -L -1' -8562443428621539458

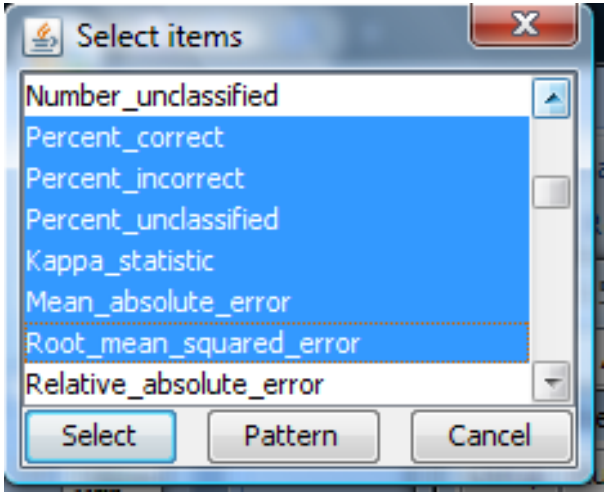
Configure Test

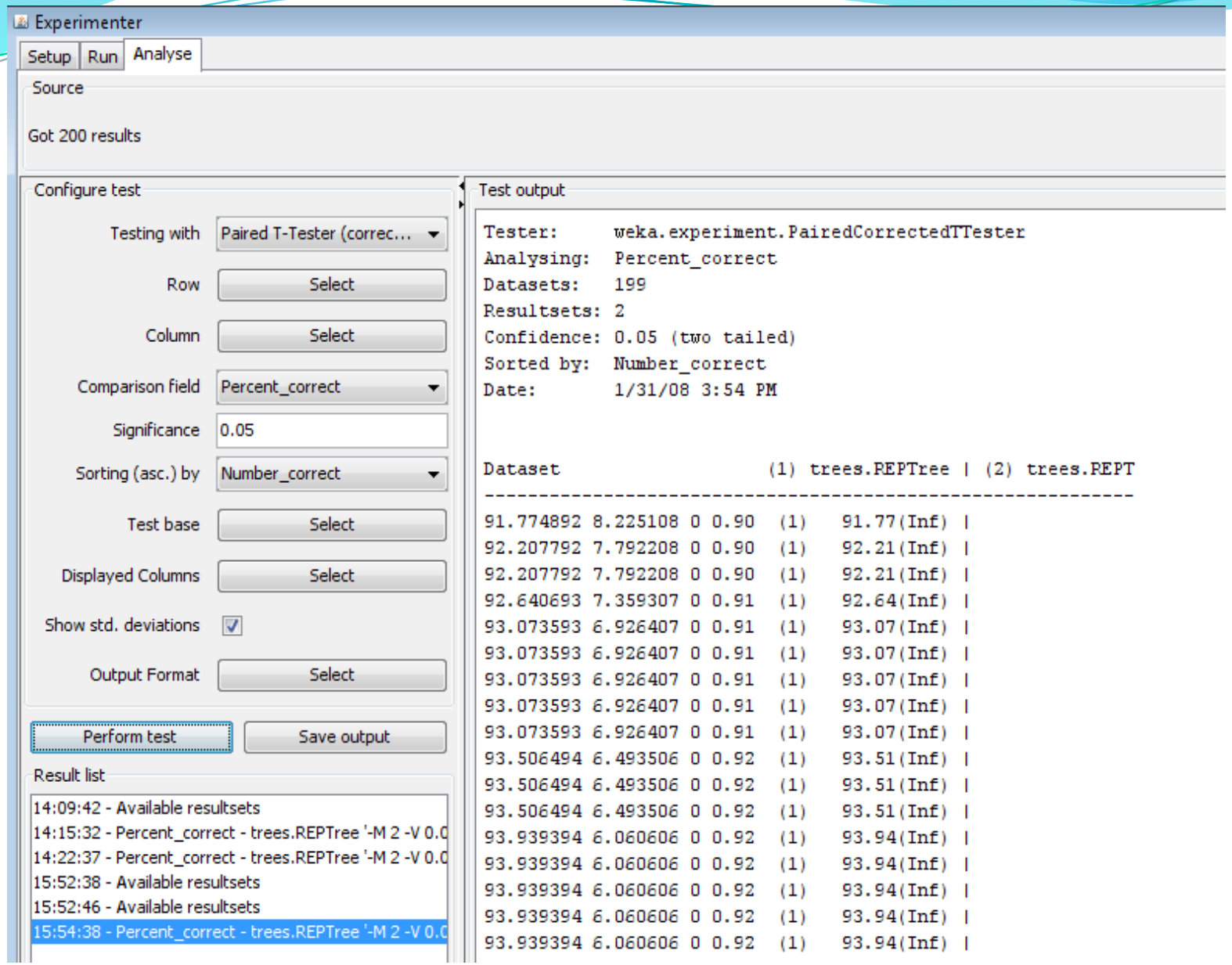
- On the left, configure the test
- The default values give the following results


Weka 3.5.7 - Experimenter
Program Applications Tools Visualization Windows Help
Experimenter
Setup Run Analyse
Source
Got 200 results
File... Database... Experiment
Configure test
Testing with Paired T-Tester (correc...
Row Select
Column Select
Comparison field Percent_correct
Significance 0.05
Sorting (asc.) by <default>
Test base Select
Displayed Columns Select
Show std. deviations
Output Format Select
Perform test Save output
Result list
Test output
Tester: weka.experiment.PairedCorrectedTTester
Analysing: Percent_correct
Datasets: 1
Resultsets: 2
Confidence: 0.05 (two tailed)
Sorted by: -
Dataset (1) trees.RE | (2) trees
segment-weka.filters.unsu(100) 95.35 | 94.96
(v/ /*) | (0/1/0)
Key:
(1) trees.REPTree '-M 2 -V 0.0010 -N 3 -S 1 -L -1' -8562443428621539458
(2) trees.REPTree '-M 4 -V 0.0010 -N 3 -S 1 -L -1' -8562443428621539458

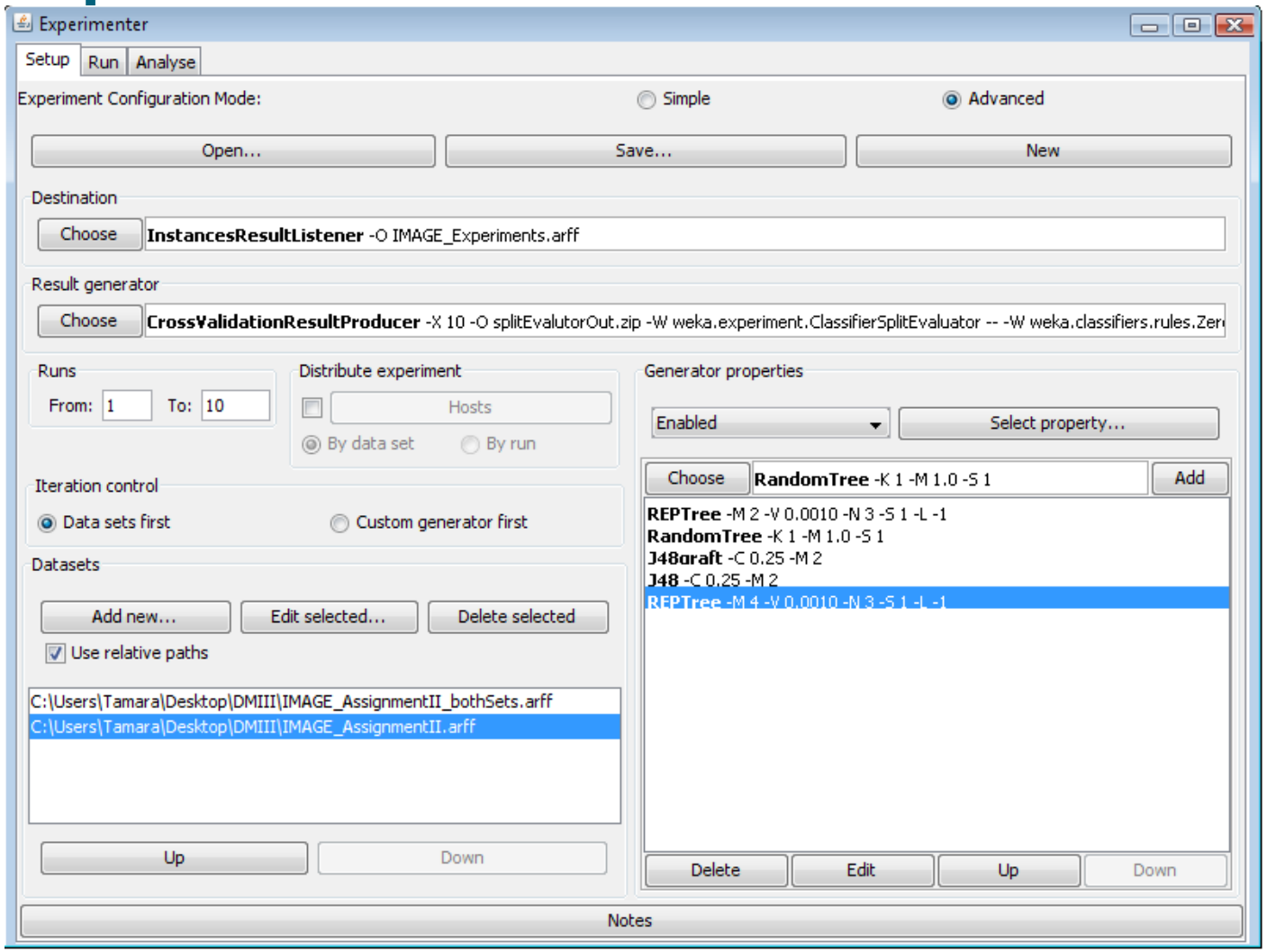
Another Analyze Run

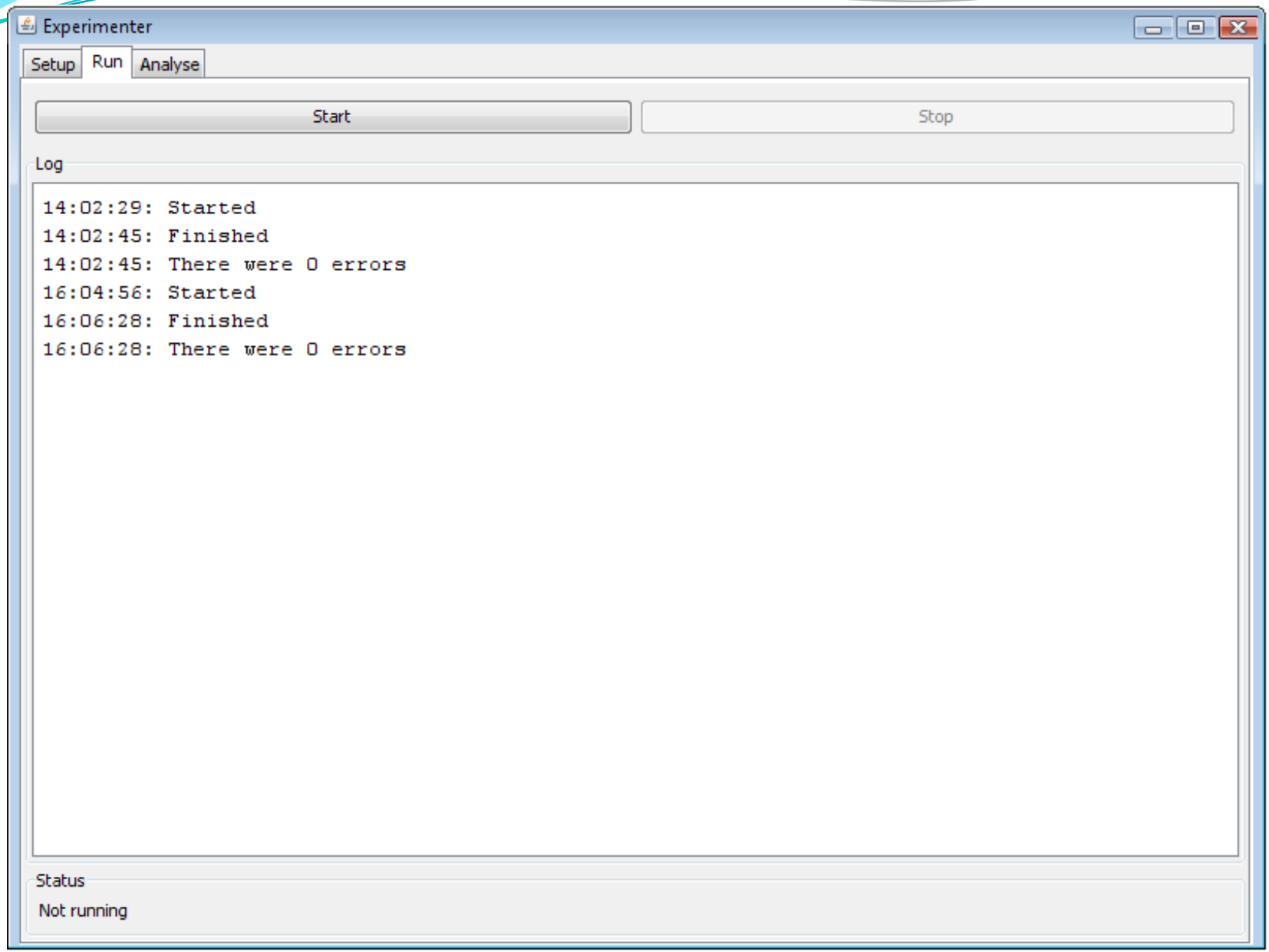
- Row select




Experimenter
Setup
Run
Analyse
Source
Got 200 results
Configure test
Testing with
Paired T-Tester (correc...
Row
Select
Column
Select
Comparison field
Percent_correct
Significance
0.05
Sorting (asc.) by
Number_correct
Test base
Select
Displayed Columns
Select
Show std. deviations
Output Format
Select
Perform test
Save output
Result list
14:09:42 - Available resultsets
14:15:32 - Percent_correct - trees.REPTree '-M 2 -V 0.0
14:22:37 - Percent_correct - trees.REPTree '-M 2 -V 0.0
15:52:38 - Available resultsets
15:52:46 - Available resultsets
15:54:38 - Percent_correct - trees.REPTree '-M 2 -V 0.0
Test output
Tester: weka.experiment.PairedCorrectedTTester
Analysing: Percent_correct
Datasets: 199
Resultsets: 2
Confidence: 0.05 (two tailed)
Sorted by: Number_correct
Date: 1/31/08 3:54 PM
Dataset (1) trees.REPTree | (2) trees.REPT
91.774892 8.225108 0 0.90 (1) 91.77(Inf) |
92.207792 7.792208 0 0.90 (1) 92.21(Inf) |
92.207792 7.792208 0 0.90 (1) 92.21(Inf) |
92.640693 7.359307 0 0.91 (1) 92.64(Inf) |
93.073593 6.926407 0 0.91 (1) 93.07(Inf) |
93.073593 6.926407 0 0.91 (1) 93.07(Inf) |
93.073593 6.926407 0 0.91 (1) 93.07(Inf) |
93.073593 6.926407 0 0.91 (1) 93.07(Inf) |
93.073593 6.926407 0 0.91 (1) 93.07(Inf) |
93.506494 6.493506 0 0.92 (1) 93.51(Inf) |
93.506494 6.493506 0 0.92 (1) 93.51(Inf) |
93.506494 6.493506 0 0.92 (1) 93.51(Inf) |
93.939394 6.060606 0 0.92 (1) 93.94(Inf) |
93.939394 6.060606 0 0.92 (1) 93.94(Inf) |
93.939394 6.060606 0 0.92 (1) 93.94(Inf) |
93.939394 6.060606 0 0.92 (1) 93.94(Inf) |
93.939394 6.060606 0 0.92 (1) 93.94(Inf) |

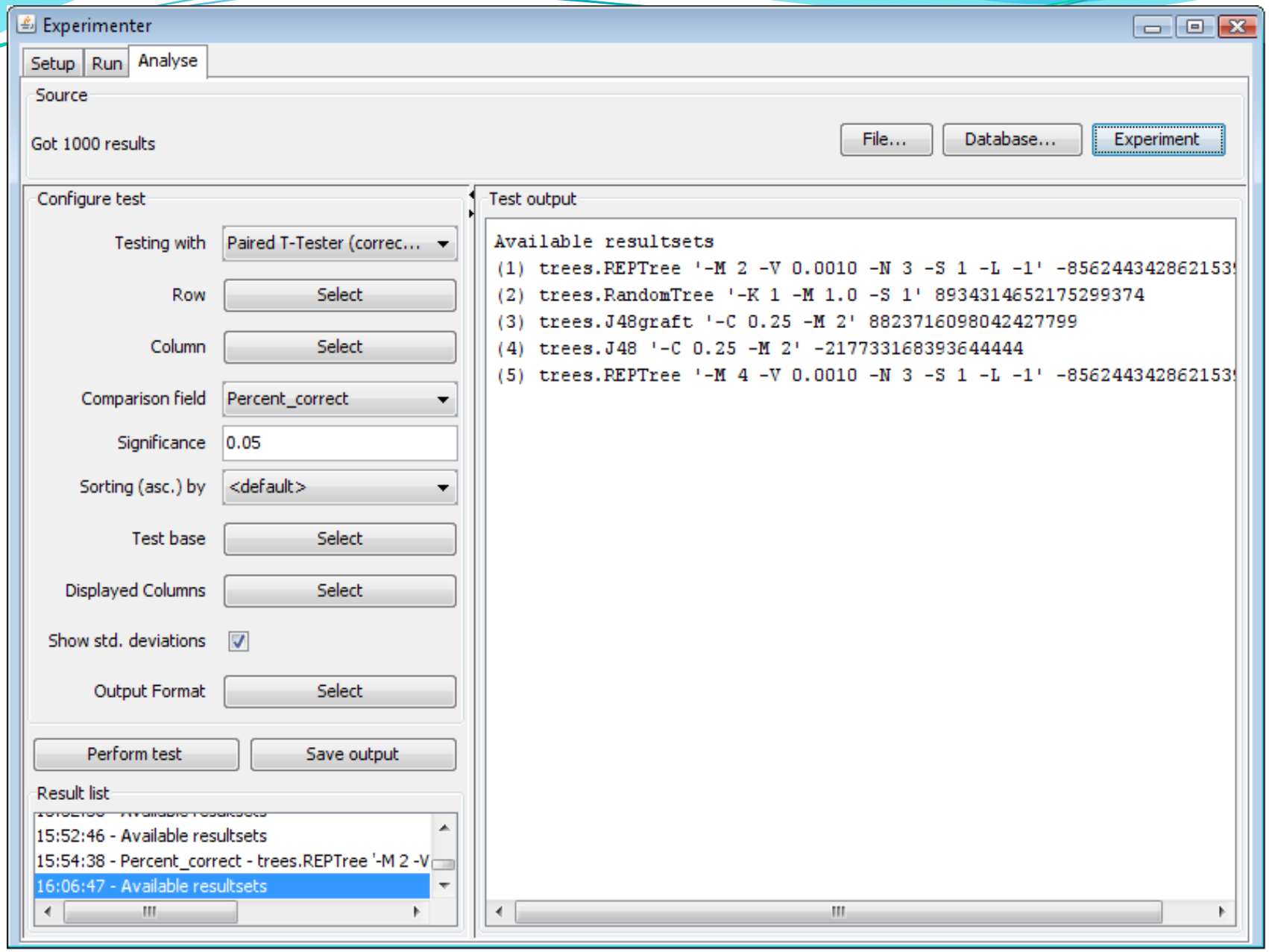
Let’s Experiment More

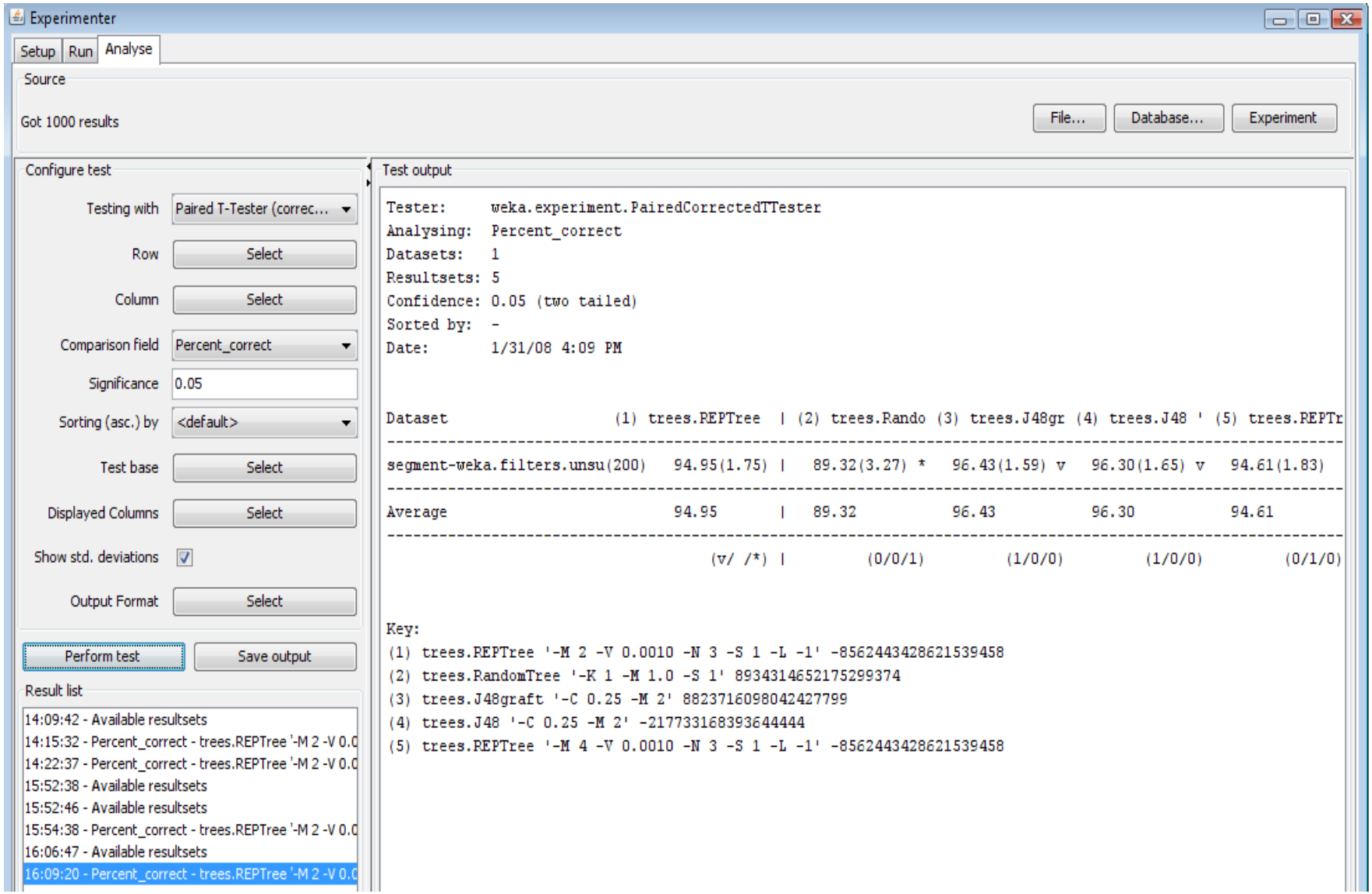



Experimenter
Setup
Run
Analyse
Start
Stop
Log
14:02:29: Started
14:02:45: Finished
14:02:45: There were 0 errors
16:04:56: Started
16:06:28: Finished
16:06:28: There were 0 errors
Status
Not running

Now, the Analysis

- Analysis tab
- Make sure you click on Experiment to upload all the methods we ran
- Set up the parameters (let's leave them as they were)
- Click on Perform test


Experimenter
Setup
Run
Analyse
Source
Got 1000 results
File...
Database...
Experiment
Configure test
Testing with
Paired T-Tester (correc...
Row
Select
Column
Select
Comparison field
Percent_correct
Significance
0.05
Sorting (asc.) by
<default>
Test base
Select
Displayed Columns
Select
Show std. deviations
Output Format
Select
Perform test
Save output
Result list
15:52:46 - Available resultsets
15:54:38 - Percent_correct - trees.REPTree '-M 2 -V
16:06:47 - Available resultsets
Test output
Available resultsets
(1) trees.REPTree '-M 2 -V 0.0010 -N 3 -S 1 -L -1' -856244342862153
(2) trees.RandomTree '-K 1 -M 1.0 -S 1' 8934314652175299374
(3) trees.J48graft '-C 0.25 -M 2' 8823716098042427799
(4) trees.J48 '-C 0.25 -M 2' -217733168393644444
(5) trees.REPTree '-M 4 -V 0.0010 -N 3 -S 1 -L -1' -856244342862153


Experimenter
Setup
Run
Analyse
Source
Got 1000 results
File...
Database...
Experiment
Configure test
Testing with
Paired T-Tester (correc...
Row
Select
Column
Select
Comparison field
Percent_correct
Significance
0.05
Sorting (asc.) by
<default>
Test base
Select
Displayed Columns
Select
Show std. deviations
Output Format
Select
Perform test
Save output
Result list
14:09:42 - Available resultsets
14:15:32 - Percent_correct - trees.REPTree '-M 2 -V 0.0
14:22:37 - Percent_correct - trees.REPTree '-M 2 -V 0.0
15:52:38 - Available resultsets
15:52:46 - Available resultsets
15:54:38 - Percent_correct - trees.REPTree '-M 2 -V 0.0
16:06:47 - Available resultsets
16:09:20 - Percent_correct - trees.REPTree '-M 2 -V 0.0
Test output
Tester: weka.experiment.PairedCorrectedTTester
Analysing: Percent_correct
Datasets: 1
Resultsets: 5
Confidence: 0.05 (two tailed)
Sorted by: -
Date: 1/31/08 4:09 PM
Dataset (1) trees.REPTree | (2) trees.Rando (3) trees.J48gr (4) trees.J48 ' (5) trees.REPTr
segment-weka.filters.unsu(200) 94.95(1.75) | 89.32(3.27) * 96.43(1.59) v 96.30(1.65) v 94.61(1.83)
Average 94.95 | 89.32 96.43 96.30 94.61
(v/ /*) | (0/0/1) (1/0/0) (1/0/0) (0/1/0)
Key:
(1) trees.REPTree '-M 2 -V 0.0010 -N 3 -S 1 -L -1' -8562443428621539458
(2) trees.RandomTree '-K 1 -M 1.0 -S 1' 8934314652175299374
(3) trees.J48graft '-C 0.25 -M 2' 8823716098042427799
(4) trees.J48 '-C 0.25 -M 2' -217733168393644444
(5) trees.REPTree '-M 4 -V 0.0010 -N 3 -S 1 -L -1' -8562443428621539458

Conclusions

- Experimenter is a great tool!
- Lets you run various methods, with different parameters and statistically compare the results
- To explore the problem, filter the data, prepare the data and so on, first use Explorer
- Once you are merely searching the model space and know which parameters you want to vary, which methods to use, then use the Experimenter

Summary of What We Learned

- Hands-on class with 7 real data mining projects, performed in a step-by-step manner
- Lots of screen shots and the specifics
- Particular attention to details
- Lots of your “how to” and “why” questions answered
- Excellent data mining practice and experience!
- We built up your data mining “intuition”
- Learned the in’s and out’s of weka, and the Explorer and Experimenter tools it provides

Summary of What We Learned

- Learned about the details of practical data preparation and data mining or real life datasets
- Acquired theoretical and practical knowledge of ensemble/meta/hybrid modeling
- Experienced data mining one of those “nightmare” datasets and explored several solution paths
- Realized that each of the data mining projects is sort of a detective work in discovering that perfect model and searching through a haystack of solutions, guided by some vague clues

Data Mining Tips and Useful Guidance

- Always start with a problem definition
- Learn about your data first, before you ask the data mining method to learn about it
- Thorough cleaning and data preparation is essential
- More than one iteration of data preparation may be needed
- Know what type of a model you need to produce before you attempt to build one

Data Mining Tips and Useful Guidance - continued

- Know what various methods are capable of doing, what their parameters control
- More than one iteration of the modeling is most likely needed
- You may need to utilize a meta learner to combine weaker models
- You might need to wait for more data or add new information to the existing dataset
- The key – creating a model that will do well on unseen data (try to avoid overfitting)

Overfit vs. Underfit

- Overfitting → learning the peculiarities and the details of the training data too well; unable to generalize well, which means that the performance on the unseen/future data will be weak
- Underfitting → too much generalization; not learning or overlooking the important details; the model performs poorly on both the training data and the unseen data
- Solid Data Mining practice is a constant battle of balancing in between overfitting and underfitting and getting just the right model, that will test well on unseen data

Conclusion

- Data mining is an exciting field with wonderful capabilities for extracting knowledge, information, patterns hidden deeply in the data, the gist of the data
- An amazing automated way of getting the most information and knowledge out of your data
- No strict steps to follow
- Knowledge of the tools is essential
- Data Mining experience is fundamental as well
- Hands-on practice is vital

Conclusion - continued

- Real life datasets are unclean and often difficult to model
- Thorough cleaning and preparation is needed
- You need to know a lot about the dataset at hand: the history, any revisions, the meaning of the attributes, how the data was collected, etc.
- Sometimes you just need to wait for more data, or ask for a different protocol for collecting the data, more input variables, added background information, etc.
- Sometimes using a meta learner can help solve the problem

What's Next?

- What's Next?
- Hands on experience – get as much as possible practical data mining
- Helpful resources:
  - Weka's data folder and other available .arff files
  - UCI – machine learning repository with datasets and references to published research:

    http://archive.ics.uci.edu/ml/

Next

- Assignment IV (Final Assignment)
- End of the Class!!
- Thank You