Subject

TANAGRA, ORANGE and WEKA: Comparison of learning algorithms using a predefined learning and test set.

Very often, we use the accuracy to compare the performances of the algorithms. We then select the method that is the most accurate. So that the comparison is rigorous, it is necessary that we use the same dataset in training and test phase.

We show in this tutorial, how to implement this process in three data mining software: TANAGRA, ORANGE and WEKA. We chose to compare the performances of a SVM (linear kernel), a logistic regression and a decision tree.

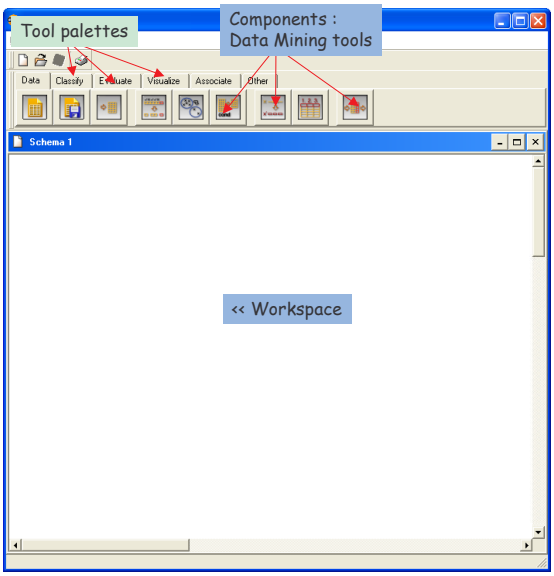
Dataset

We use the BREAST dataset (UCI IRVINE). We have a binary class attribute (benign or malignant tumor), 9 continuous descriptors, and 699 examples.

We have selected 499 examples for the training phase, 200 examples for the test. **We use the same subdivision for our three packages.**

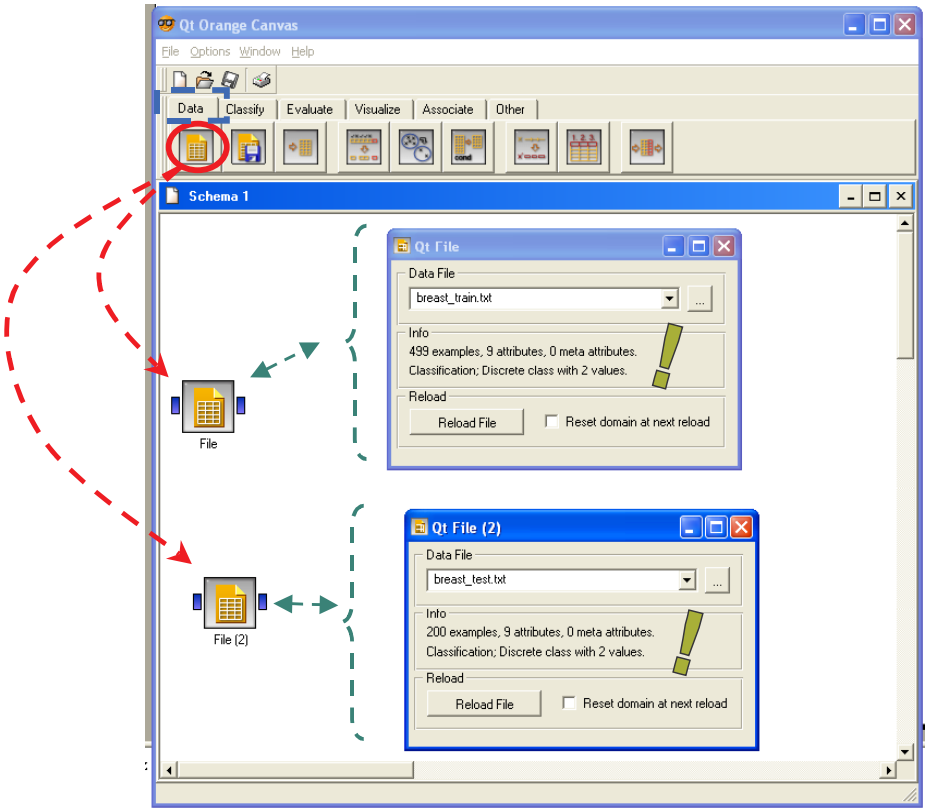
Algorithms comparison with ORANGE

When we execute ORANGE, we have the following interface.



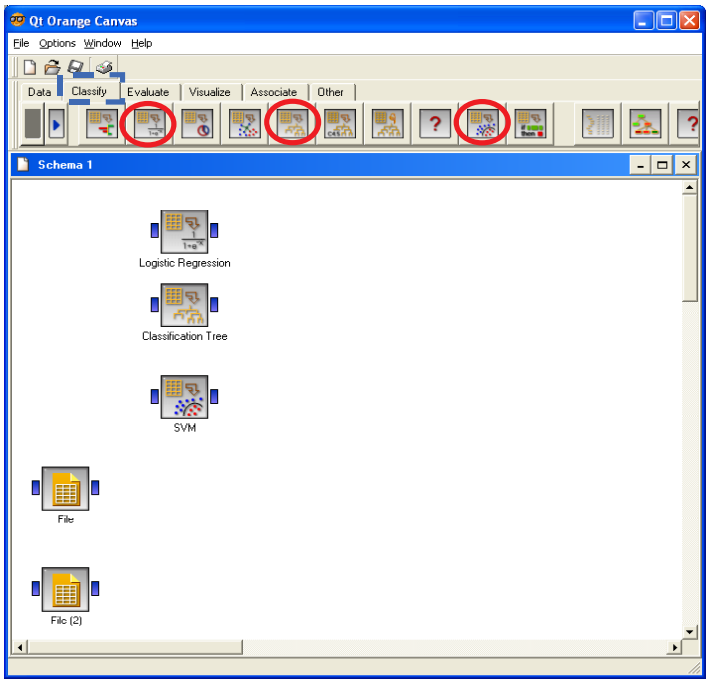
Data preparation

We divide the whole dataset into two files: BREAST_TRAIN.TXT for training, BREAST_TEST.TXT for testing. We set two data access components in the diagram; we parameterize them by activating the OPEN menu.



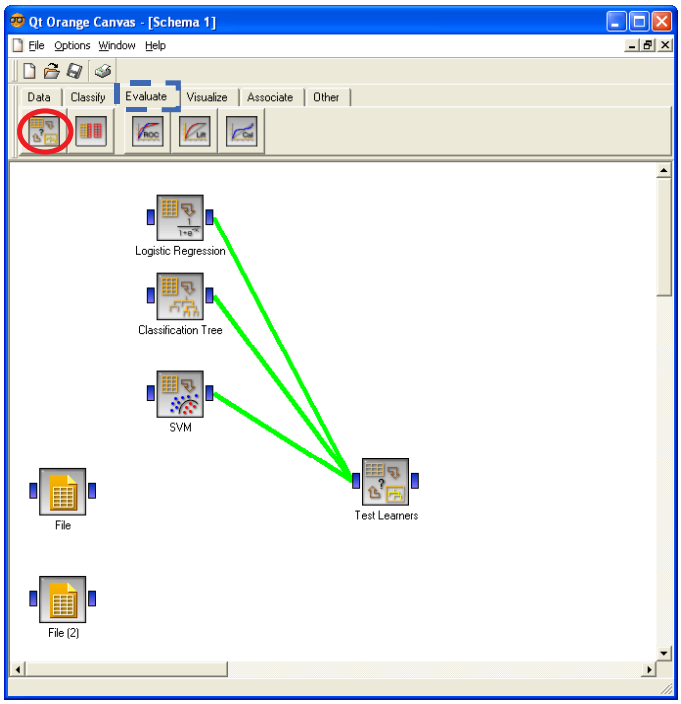
Learning components

We want to compare three learning methods from the CLASSIFY tab, we set them in the diagram.



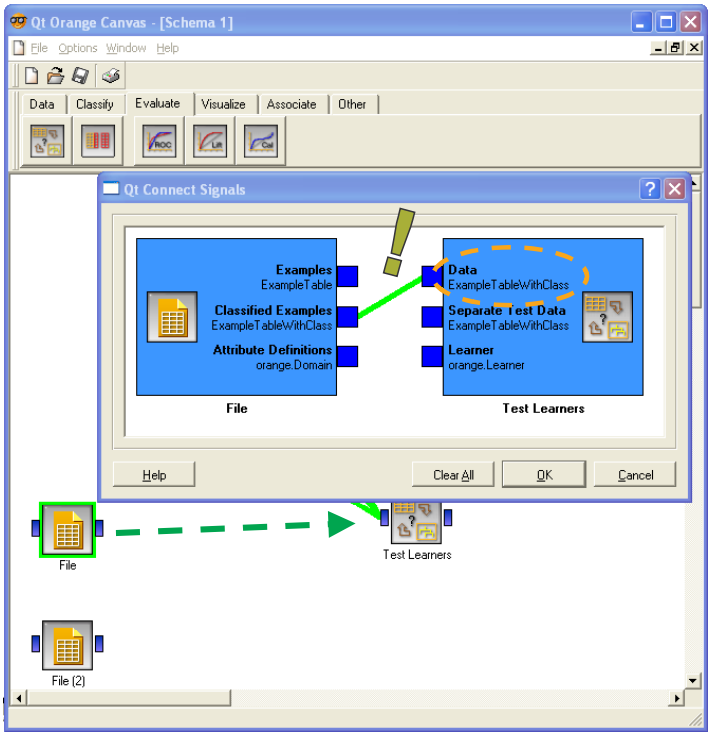
Evaluation component

The comparison can be gathered in only one component, the TEST LEARNERS component from the EVALUATE tab. We connect the three learning method to this new component.

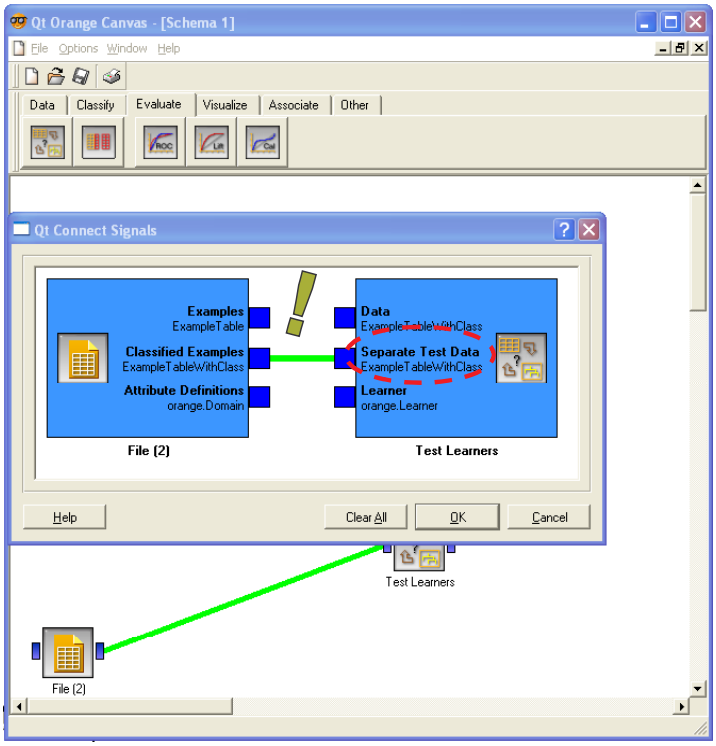


We must now specify which are the data to be used for the training. We connect the first data source [FILE] to the TEST LEARNERS. A dialog box appears, it is of primary importance

because it enables us to check that we transmit the training set (DATA). The training phase is automatically started.

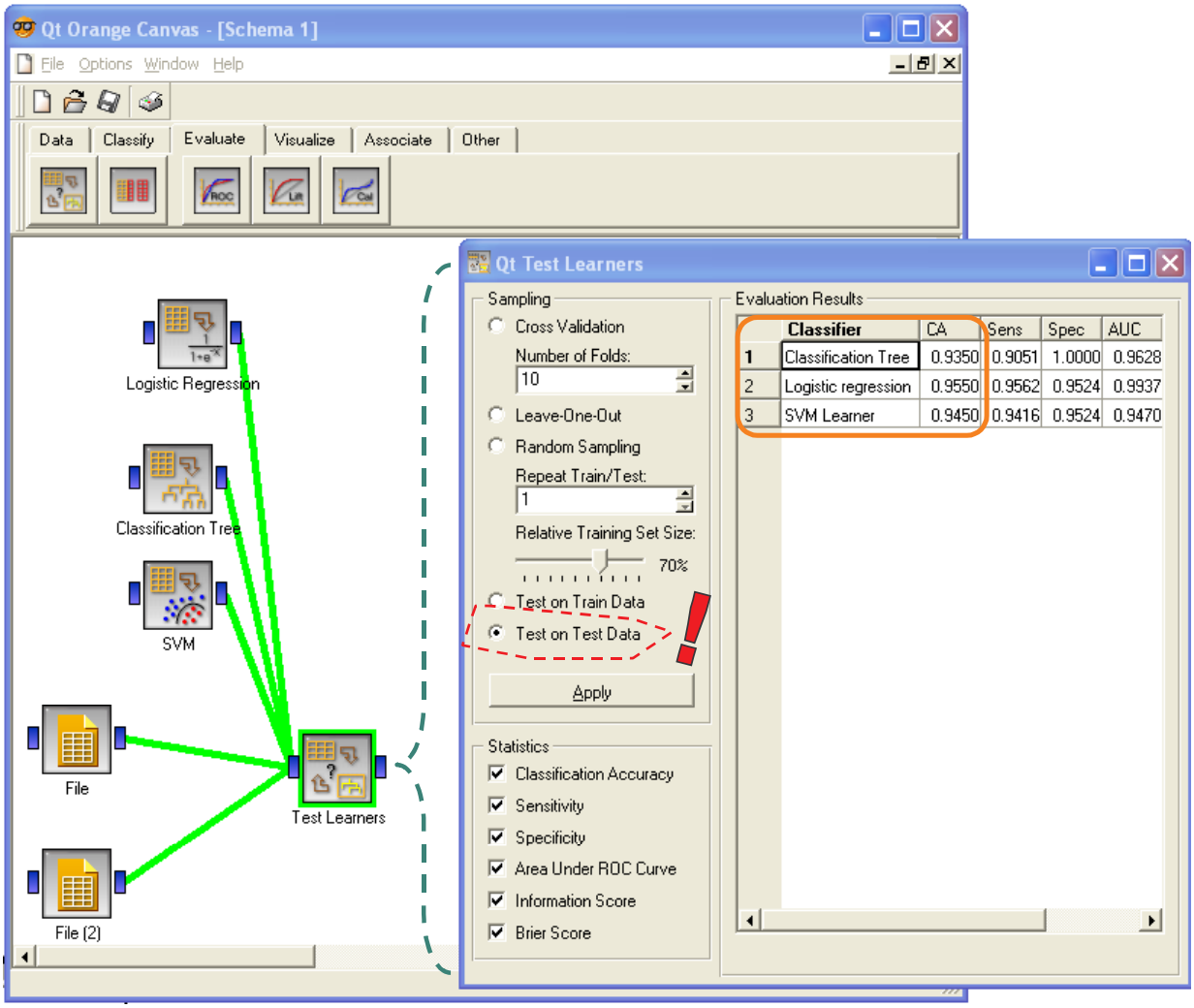


In the next step, we connect the second data source [FILE (2)] to the TEST LEARNERS component. ORANGE considers that this second data source is the test set (SEPARATE TEST DATA). We can modify this type of the connection when we double-click on the link; it is not necessary here.



Seeing the results

To display the results, we select the OPEN menu of the TEST LEARNERS component.



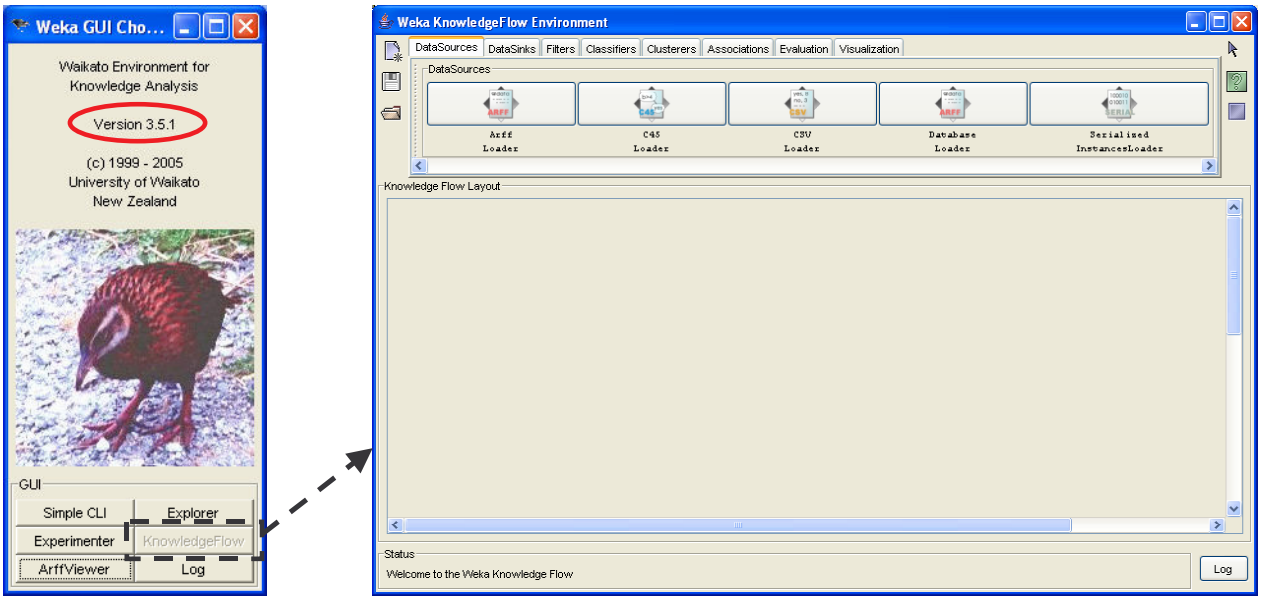
We check the "Test on test data" option. Various statistics are available; we are interested primarily in the accuracy in our tutorial:

- Classification tree: 93.5% (error rate 6.5%);
- Logistic regression: 95.5%
- Linear SVM[1]: 94.5%.

[1] Check that you use really a linear kernel in your diagram (KERNEL – LINEAR).

Algorithms comparison with WEKA

A dialog box appears when we execute WEKA; we choose the **KNOWLEDGE FLOW** paradigm. We have used the **3.5.1** version.



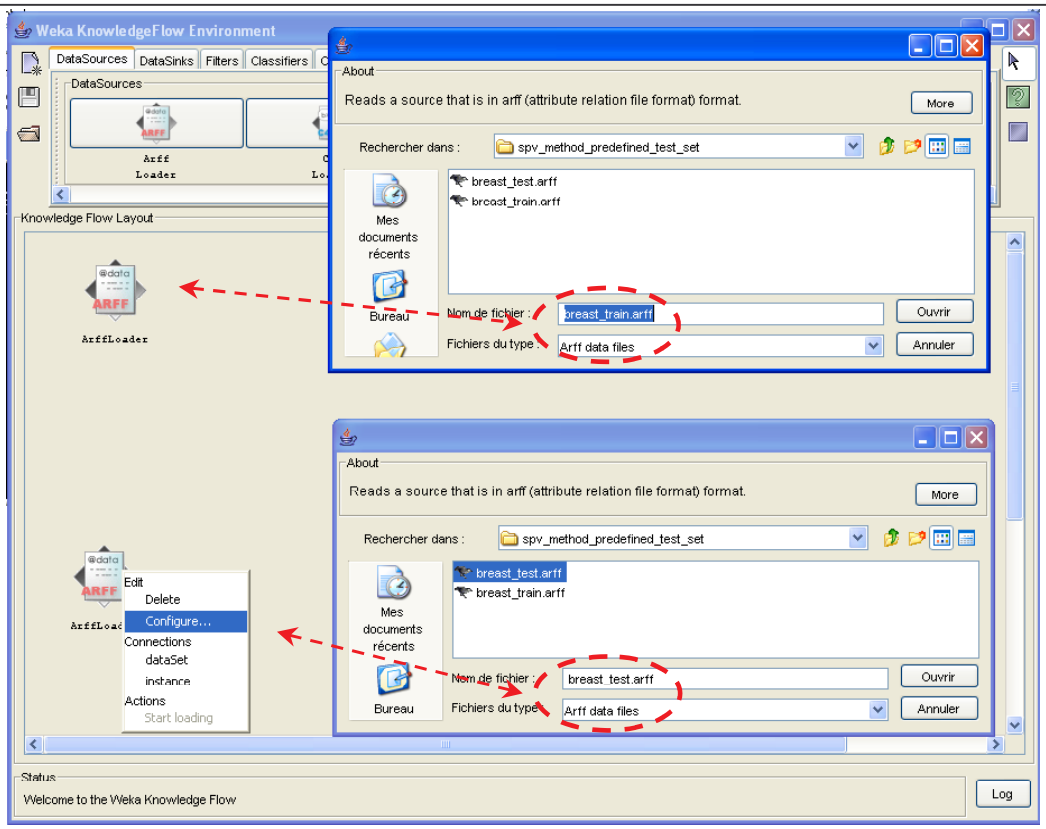
Data preparation

We have to use two separate dataset with WEKA. We use the ARFF file format; we check carefully that the description of the attributes is the same one.

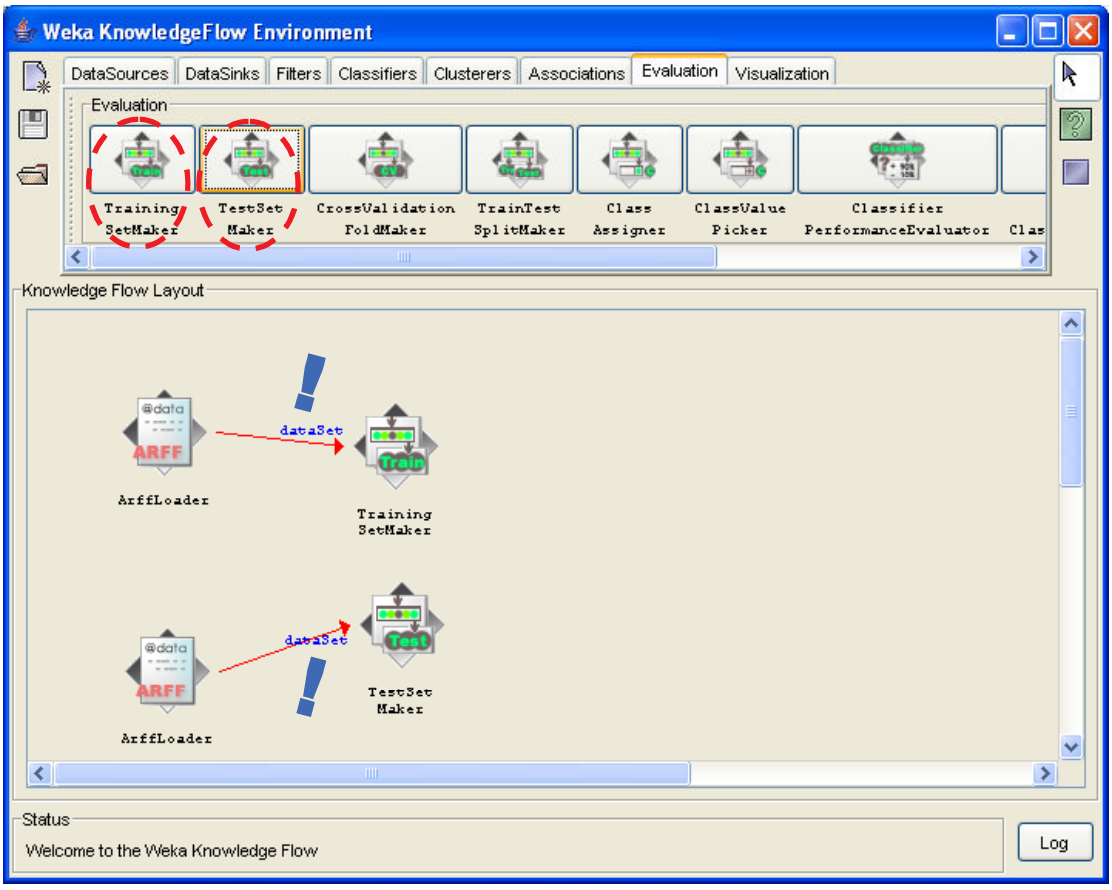
```
@relation breast_train.arff
@attribute clump REAL
@attribute ucellsize REAL
@attribute ucellshape REAL
@attribute mgadhesion REAL
@attribute sepics REAL
@attribute bnuclei REAL
@attribute bchromatin REAL
@attribute normnucl REAL
@attribute mitoses REAL
@attribute class {begnin,malignant}
```

```
@relation breast_test.arff
@attribute clump REAL
@attribute ucellsize REAL
@attribute ucellshape REAL
@attribute mgadhesion REAL
@attribute sepics REAL
@attribute bnuclei REAL
@attribute bchromatin REAL
@attribute normnucl REAL
@attribute mitoses REAL
@attribute class {begnin,malignant}
```

We set two ARFF LOADER components in the diagram; we select the datasets.

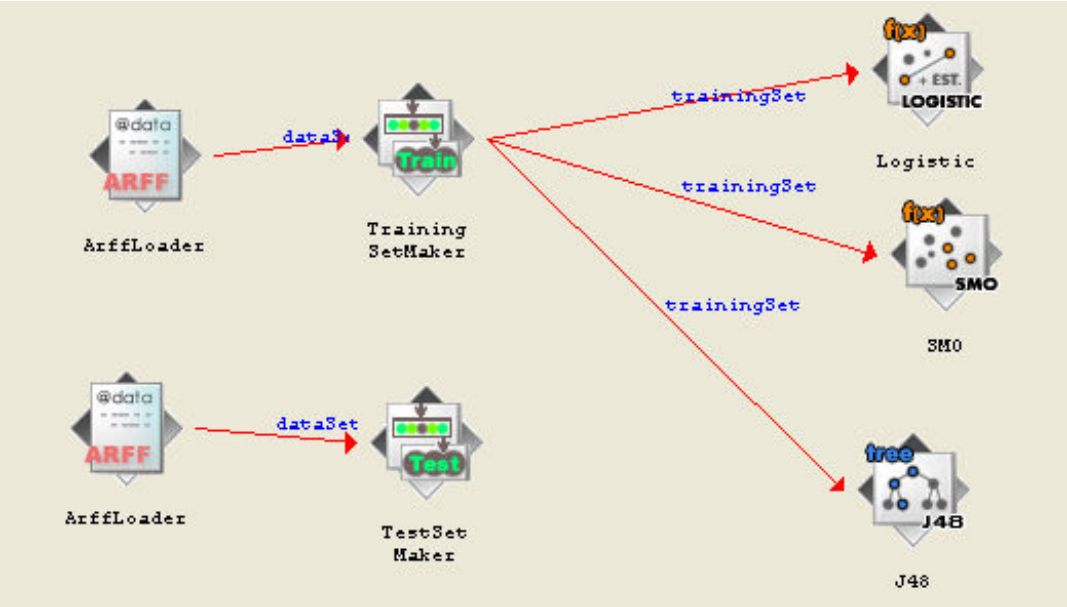


We must now specify the role of these data in the diagram. We use the TRAINING SET MAKER and TEST SET MAKER components (EVALUATE tab). We set the adequate connections.

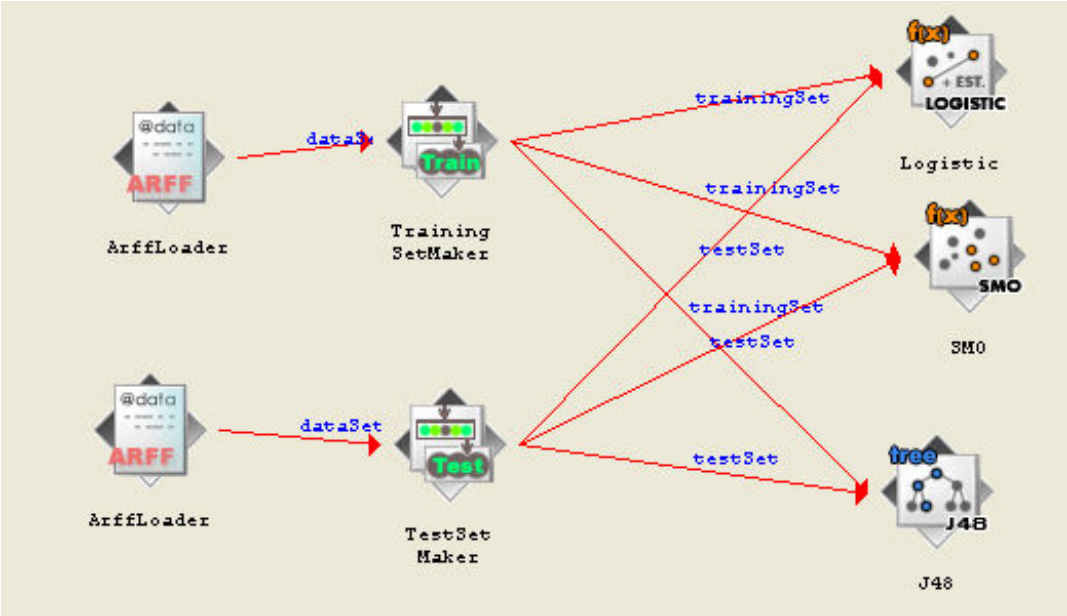


Learning methods

We set three learning algorithms components (CLASSIFIERS tab) in the diagram. About SMO, we check that we really use a linear kernel (exponent = 1, not RBF kernel). We connect the three TRAINING SET MAKER, …

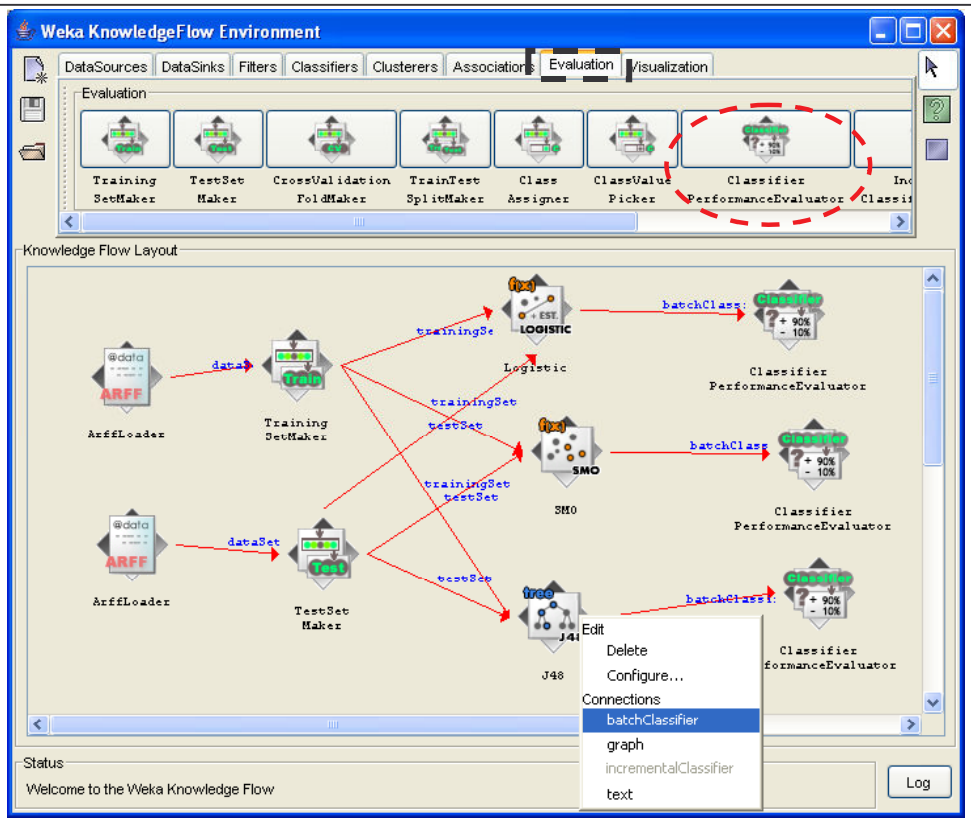


… and the three TEST SET MAKER.



Evaluation components

To compute the accuracy of the classifiers, we set CLASSIFIER PERFORMANCE EVALUATOR component (EVALUATION), one for each learning method. The type of the connection must be BATCH CLASSIFIER.



Visualization component

We set the TEXT VIEWER (VISUALIZATION) component in order to display the results. Only one component is necessary, it makes it possible to join together the results in the same window. We use the TEXT connection.

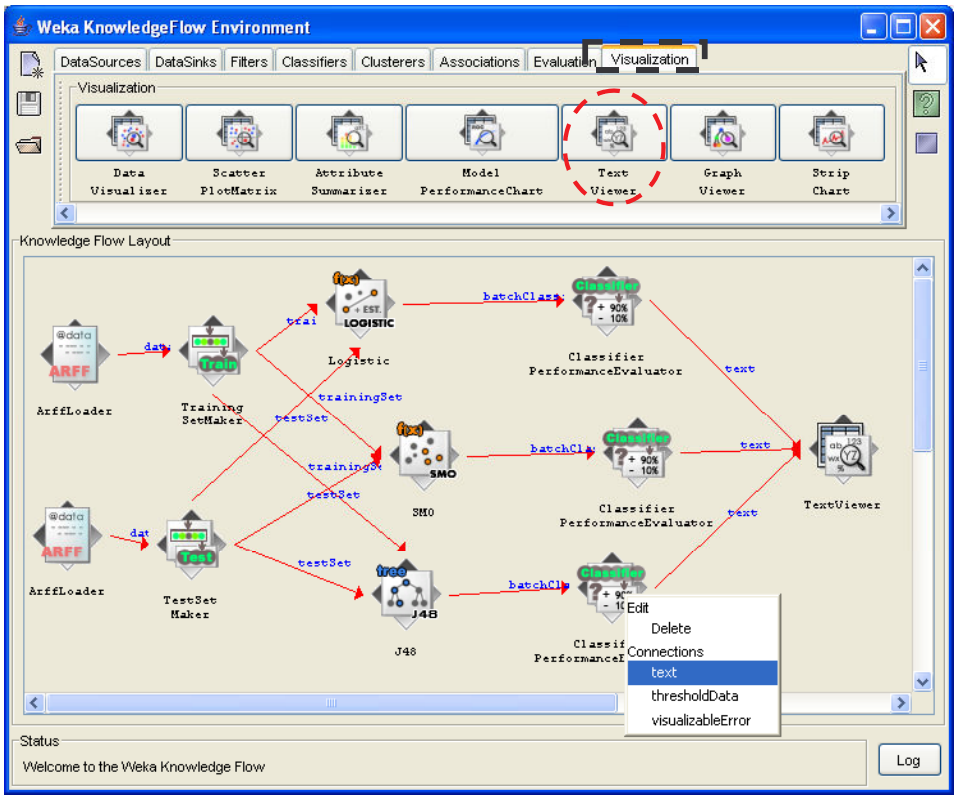
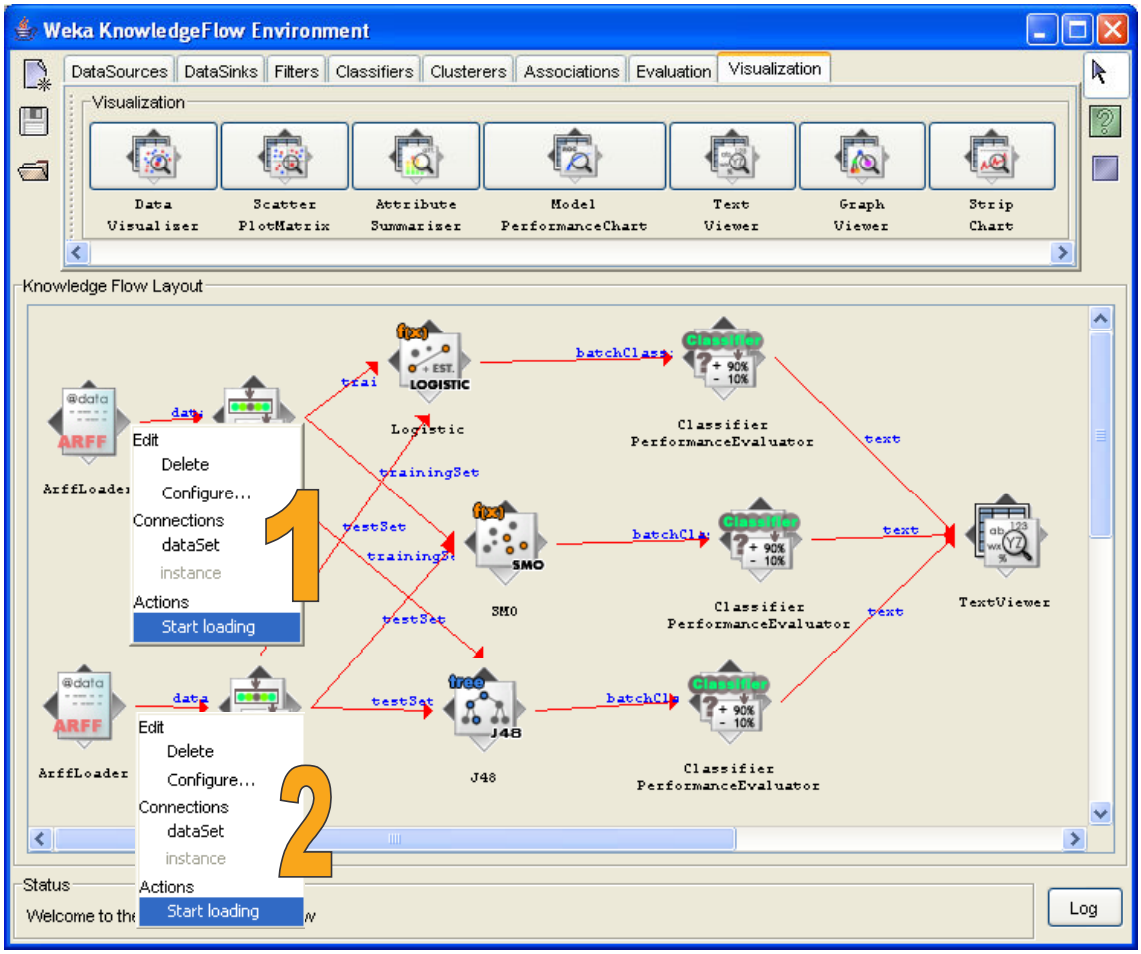
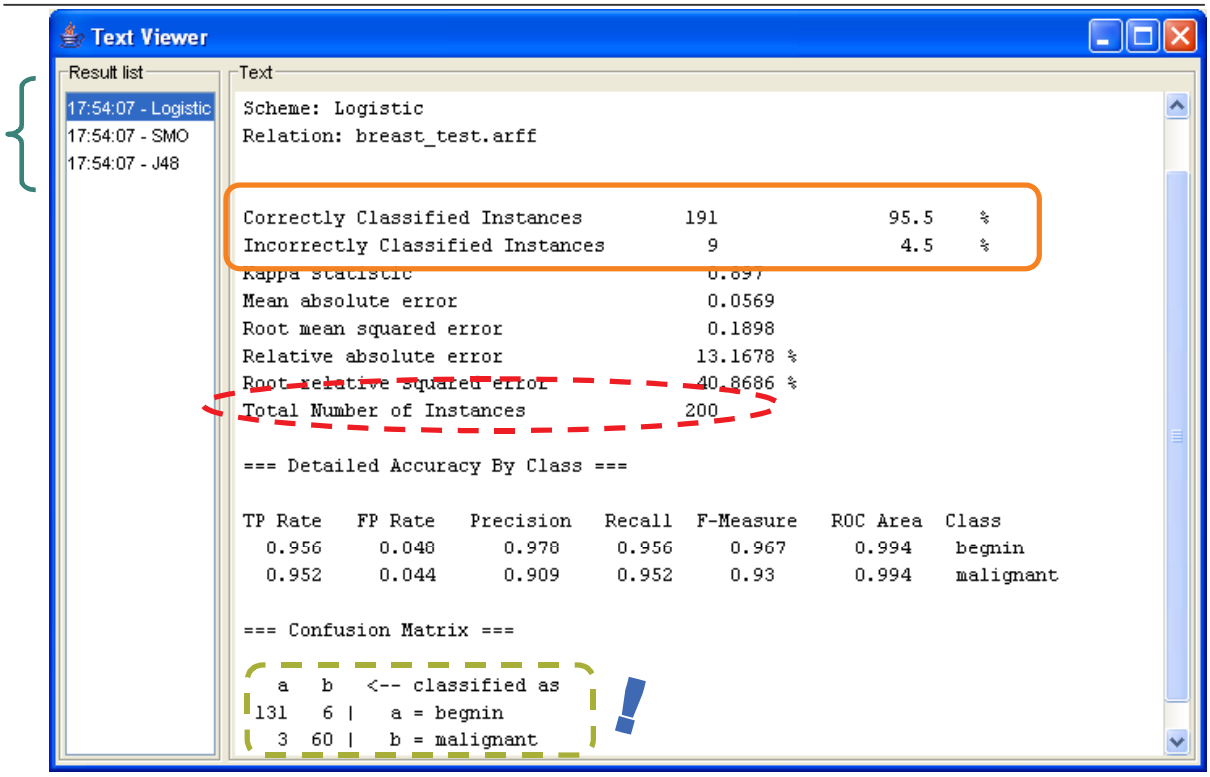


Diagram execution

The execution of the diagram is done into two steps: [1] we select the START LOADING menu of the first data source (learning set), the prediction models are computed; [2] we select the START LOADING menu of the second data source, the test set, the accuracy of the models is computed.



When we select the SHOW RESULTS menu of the TEXT VIEWER component, we can see the detailed results for each learning algorithm.



We obtain the following accuracy rate:

- Decision tree: 93.5% (error rate 6.5%);
- Logistic regression: 95.5%
- Linear SVM: 95.5%.

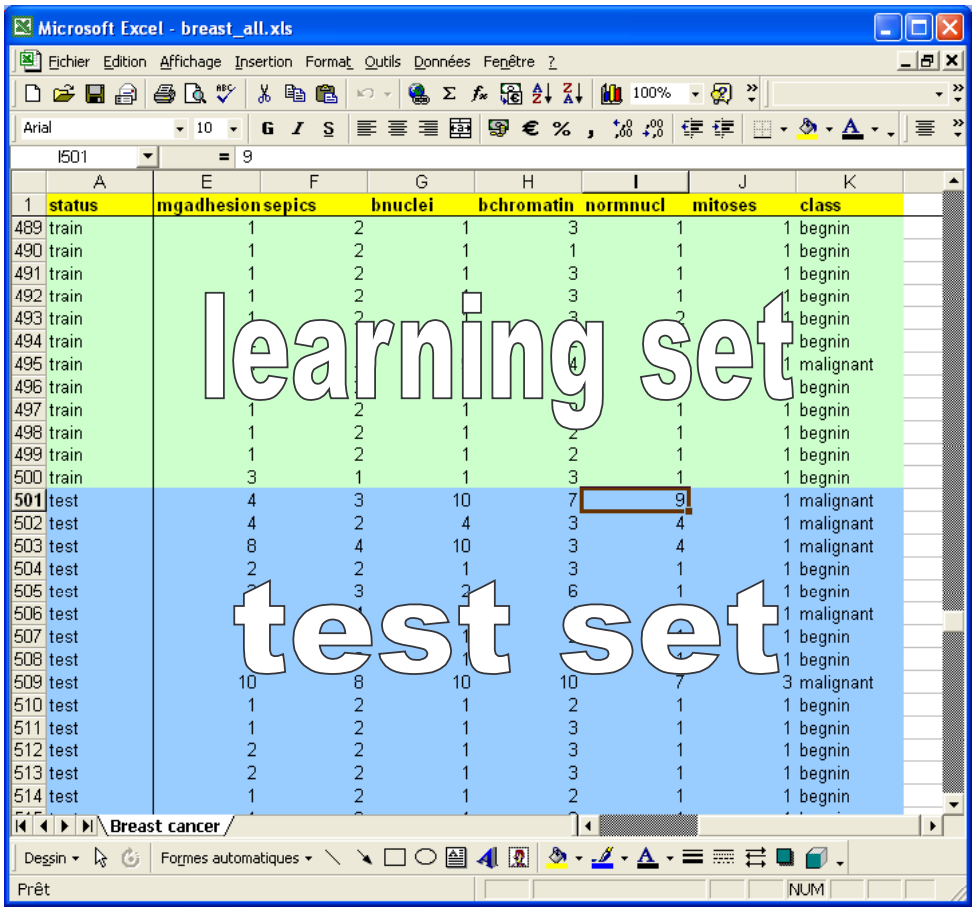
We note that SVM and Logistic regression have the same accuracy rate but not the same confusion matrix; the structure of the error is not the same one.

Algorithms comparison with TANAGRA

Compared to the two other packages, TANAGRA uses a tree to represent the treatments. That simplifies its structure, but induced a strong constraint; it is not possible to specify two data sources. It is consequently necessary to prepare the data differently.

Data preparation

We use BREAST_ALL.XLS[2]. All the examples are gathered in the same sheet; we add a new column, which enables us to distinguish training set and test set (STATUS).



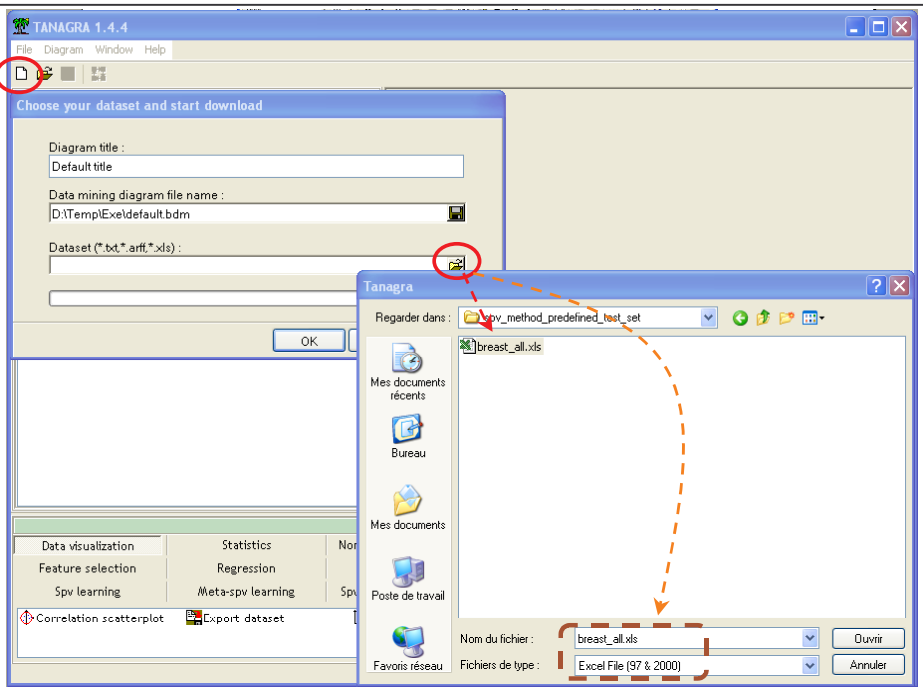
	status	mgadhesion	sepics	bnuclei	bchromatin	normnucl	mitoses	class
489	train	1	2	1	3	1	1	begnin
490	train	1	2	1	1	1	1	begnin
491	train	1	2	1	3	1	1	begnin
492	train	1	2	1	3	1	1	begnin
493	train	1	2	1	3	2	1	begnin
494	train	1	2	1	2	1	1	begnin
495	train	5	1	1	4	1	1	malignant
496	train	1	2	1	2	1	1	begnin
497	train	1	2	1	2	1	1	begnin
498	train	1	2	1	2	1	1	begnin
499	train	1	2	1	2	1	1	begnin
500	train	3	1	1	3	1	1	begnin
501	test	4	3	10	7	9	1	malignant
502	test	4	2	4	3	4	1	malignant
503	test	8	4	10	3	4	1	malignant
504	test	2	2	1	3	1	1	begnin
505	test	3	3	2	6	1	1	begnin
506	test	4	1	10	4	1	1	malignant
507	test	1	1	1	2	1	1	begnin
508	test	1	1	1	2	1	1	begnin
509	test	10	8	10	10	7	3	malignant
510	test	1	2	1	2	1	1	begnin
511	test	1	2	1	3	1	1	begnin
512	test	2	2	1	3	1	1	begnin
513	test	2	2	1	3	1	1	begnin
514	test	1	2	1	2	1	1	begnin

Data importation

We close EXCEL [3] and execute TANAGRA. We create a new diagram and import BREAT_ALL.XLS.

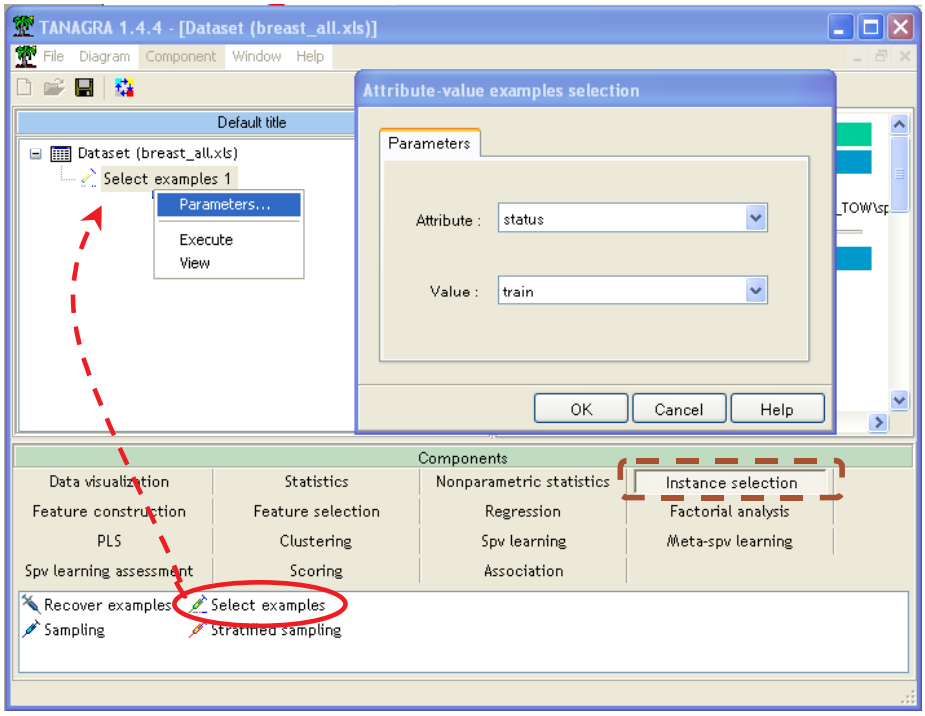
[2] TANAGRA can read XLS format. The dataset must be in the first sheet of the workbook.

[3] EXCEL locks the file that it is handling; we must close the file before executing TANAGRA.



Training and test subdivision

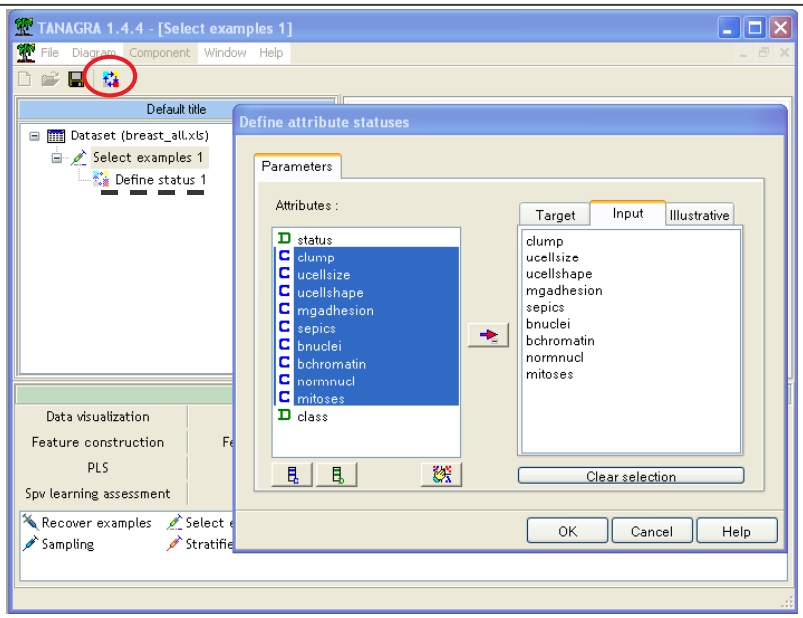
We use the SELECT EXAMPLES component (INSTANCE SELECTION) in order to select the active (training) examples.



When we execute the component (VIEW menu), we see that we have 499 selected examples for the following computations.

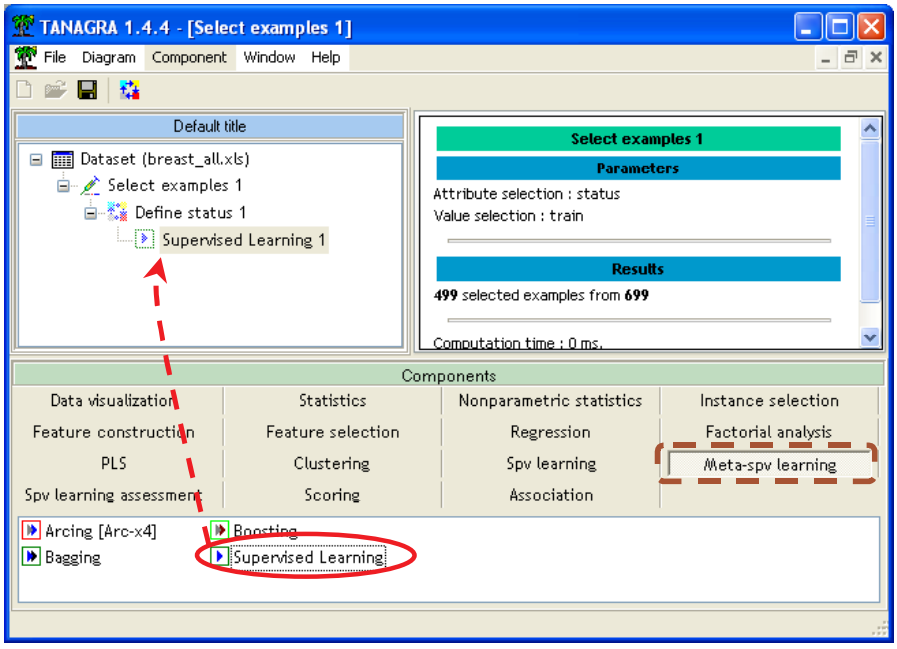
Select attributes

We add a DEFINE STATUS component in order to select the TARGET attribute (CLASS); the continuous attributes are INPUT. We do not need use STATUS attribute here.

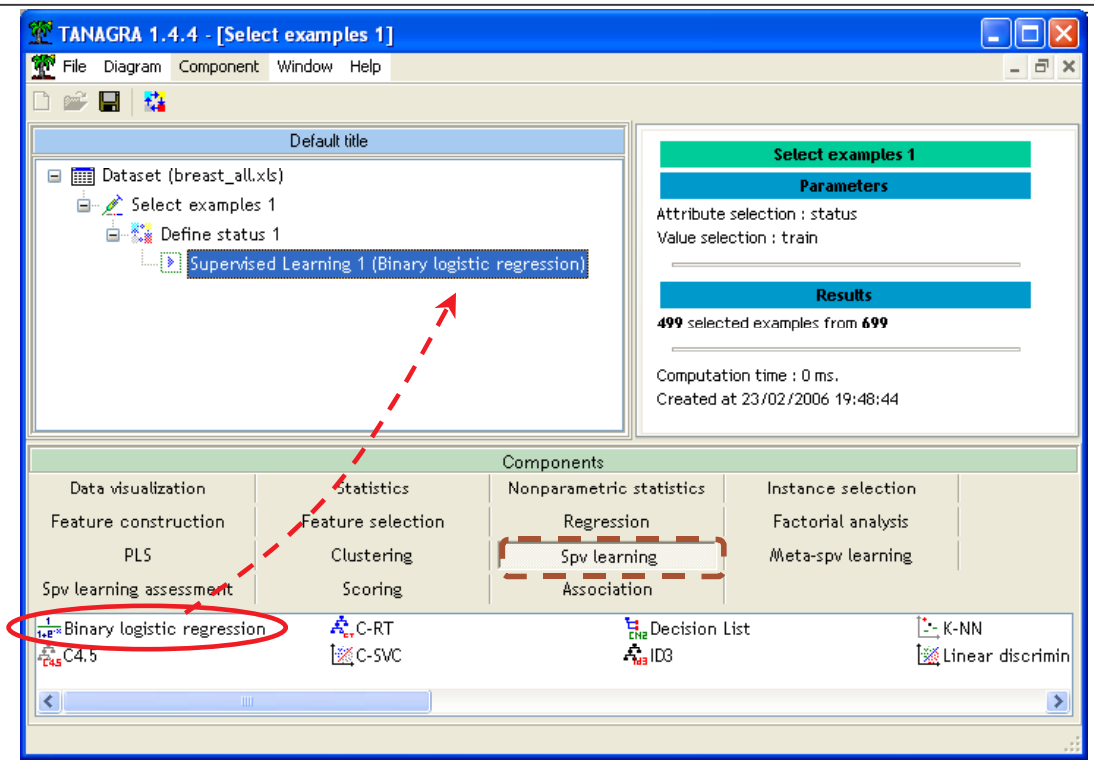


Learning method

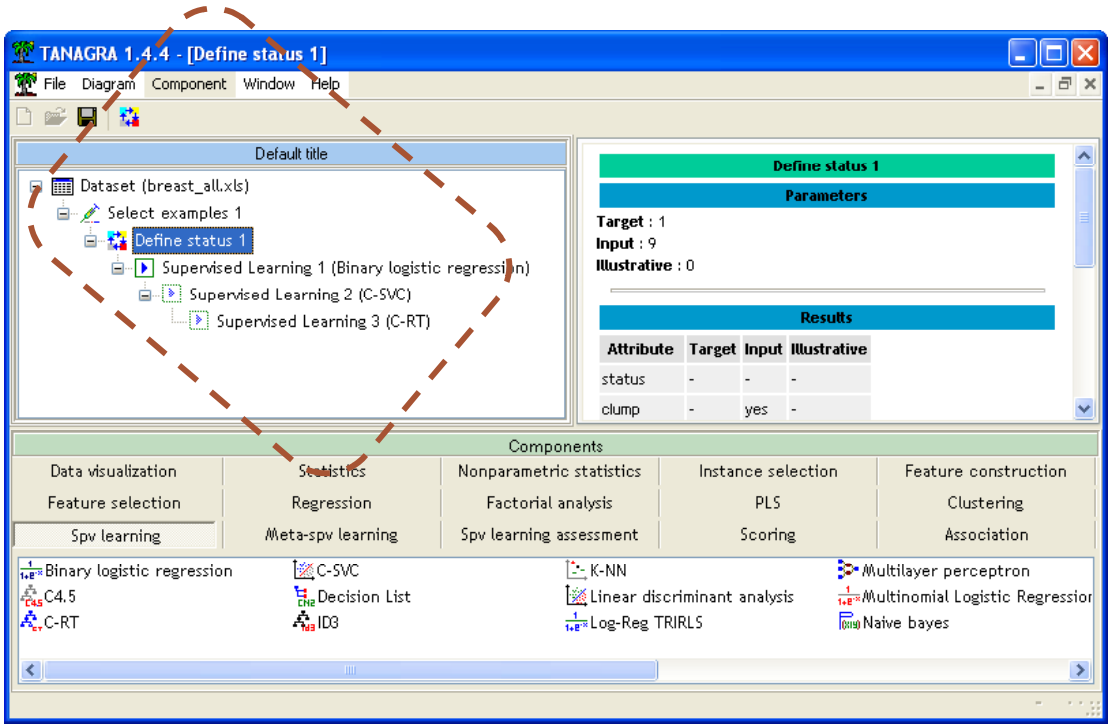
We must insert the three learning methods in the diagram. We present the detailed operation for the logistic regression. There are two steps when we want to add a supervised algorithm in the diagram: first, we insert a meta-supervised component that defines the aggregation strategy (SUPERVISED LEARNING – META SPV LEARNING tab)



Second, we embed in this component the learning strategy BINARY LOGISTIC REGRESSION (SPV LEARNING tab). This implementation of logistic regression is slightly slower than the others, but it has the advantage of providing a series of additional statistics.



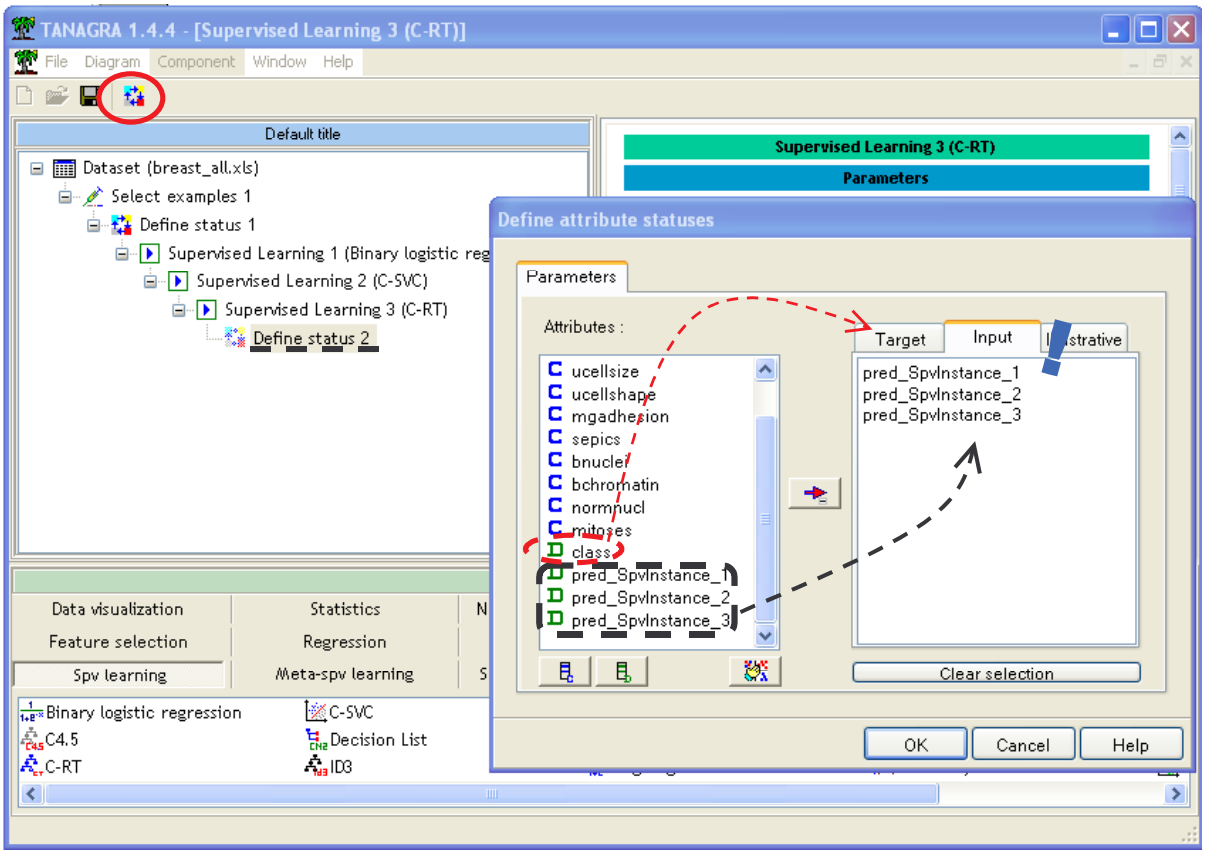
We insert the other learning methods in the diagram: SVM (C-SVC) and the decision tree (C-RT). We obtain the following diagram.



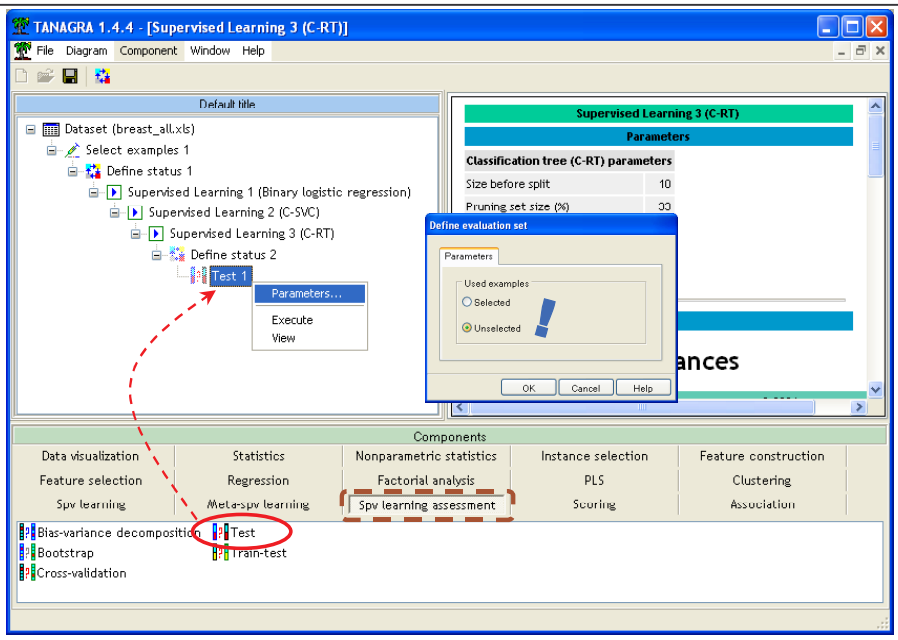
We start the execution of the whole diagram by selecting the VIEW menu of the last component. **The models are built using only the selected examples**.

Comparison of performances

To compare the performances, we must insert again the DEFINE STATUS component in the diagram by clicking on the short cut into the toolbar. We set the CLASS attribute as TARGET; the predictions of each method are the INPUT attributes. We note that **these predictions are computed for the whole dataset, including the unselected examples**.



We add the TEST (SPV LEARNING ASSESMENT tab) in the diagram. We do not forget to specify that the confusion matrix computation must be done on the unselected examples, which represents the test set.



The view menu displays the following results:

Test 1

Parameters

Evaluation set : **unselected** examples

Results

pred_SpvInstance_1

Error rate	0.0450

Values prediction

Value	Recall	1-Precision
begnin	0.9562	0.0224
malignant	0.9524	0.0909

Confusion matrix

	begnin	malignant	Sum
begnin	131	6	137
malignant	3	60	63
Sum	134	66	200

pred_SpvInstance_2

Error rate	0.0550

Values prediction

Value	Recall	1-Precision
begnin	0.9416	0.0227
malignant	0.9524	0.1176

Confusion matrix

	begnin	malignant	Sum
begnin	129	8	137
malignant	3	60	63
Sum	132	68	200

pred_SpvInstance_3

Error rate	0.0750

Values prediction

Value	Recall	1-Precision
begnin	0.9051	0.0159
malignant	0.9683	0.1757

Confusion matrix

	begnin	malignant	Sum
begnin	124	13	137
malignant	2	61	63
Sum	126	74	200

The classification accuracy rates are:

- Decision tree: 92.5% (error rate 7.5%);
- Logistic regression: 95.5%
- Linear SVM: 94.5%.

Conclusion

We see in this tutorial that it is easy to perform a comparison of algorithms using a predefined test set with ORANGE, WEKA and TANAGRA.

The results can be slightly different between the packages. This is not surprising because of the heuristic nature of learning algorithms. The effect of the implementation choices also is not negligible. Nevertheless, very large differences would have been alarming.