1 Subject

Partitioning the dataset in a learning set and a test set. Learning a cost sensitive decision tree.

Error rate evaluation is a key point of the induction process. A usual approach is to partition the dataset in a learning set, which is used for the induction of the classification model, and in a test set, which is used for the performance evaluation. The first subject of this tutorial is to show how to make a partition of the dataset with SIPINA. Then, we build the tree on the first part of the dataset. Later, we classify the examples of the second part of the dataset. We compare the predicted value and the true value. We obtain honest error rate estimation.

The second main subject of this document is to show how to take into account the misclassification costs during the learning process and the evaluation process. We use a slightly modified version of C4.5[1].

2 Dataset

We handle a spam detection problem. The learning sample contains 3601 examples; there are 1000 examples for the test set. The goal is to determine if a message is a spam or not, using their characteristics (frequencies of word, length of message, etc.) [2]. An additional column enables to know if the example belongs to the learning or the test sample.

3 Tree induction with the C4.5 algorithm

3.1 Data importation

The easiest way to import a dataset into SIPINA is to open the file into EXCEL spreadsheet. Using the SIPINA.XLA add-in, we can send directly the selected dataset to SIPINA.

Note: For the add-in installation and utilization, see the tutorial "SIPINA ADD-IN FOR EXCEL SPREADSHEET" (http://eric.univ-lyon2.fr/~ricco/sipina_download.html).

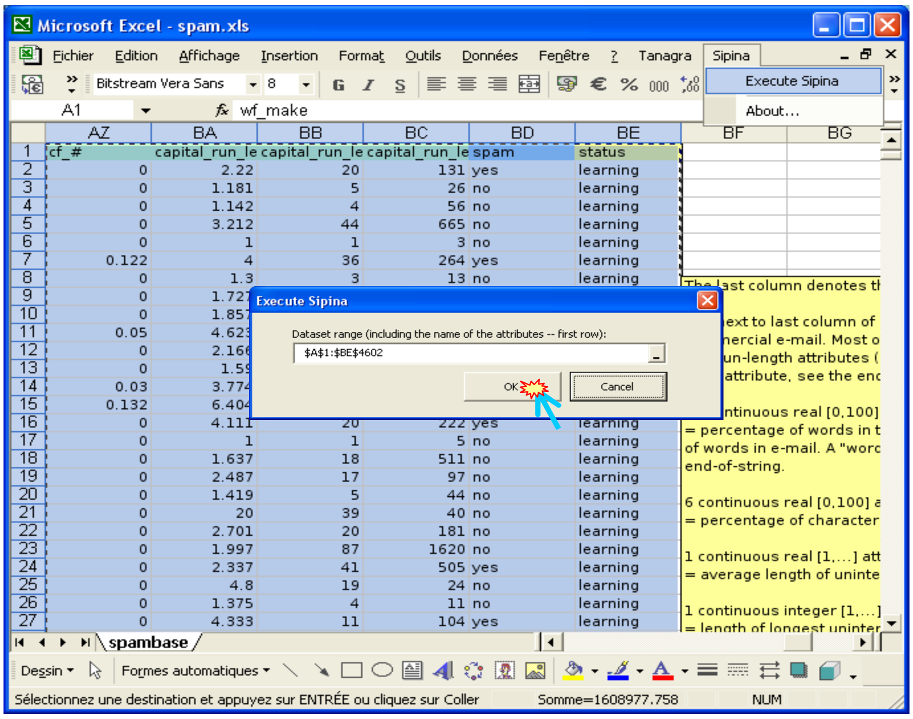
We select the cells corresponding to the dataset into EXCEL. We click on the SIPINA/EXECUTE SIPINA menu. A dialog box appears. We check the cells selection and we validate. The first row of the selection must correspond to the name of the variables (Figure 1).

SIPINA is automatically launched. The dataset is downloaded. We check in the status bar the number of columns (57) and the number of rows (4601).

Note: We can edit the dataset into the editor of SIPINA. But this is not really interesting. EXCEL offers more powerful functionalities about data manipulation.

[1] See http://www.informatik.uni-freiburg.de/~ml/ecmlpkdd/WS-Proceedings/w10/chauchat_workshop.pdf

[2] http://eric.univ-lyon2.fr/~ricco/dataset/spam.xls ; the original database is available on the following website http://archive.ics.uci.edu/beta/datasets.html (spambase)

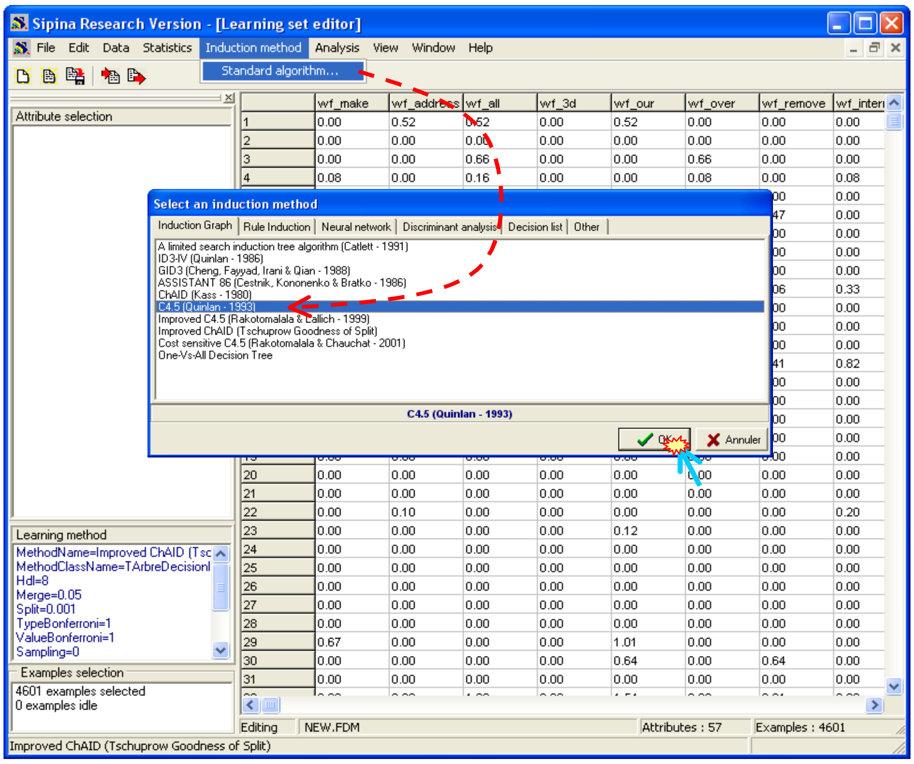


Dataset range (including the name of the attributes -- first row): $A$1:$BE$4602

Figure 1 - Transferring the dataset from EXCEL to SIPINA

3.2 Selecting the learning method – C4.5

Now we have to select the learning method. We click on the INDUCTION METHOD / STANDARD ALGORITHM menu. A dialog box which describes the available methods appears. We select the C4.5 approach (QUINLAN, 1993).


Sipina Research Version - [Learning set editor]
File Edit Data Statistics Induction method Analysis View Window Help
Standard algorithm...
Select an induction method
Induction Graph | Rule Induction | Neural network | Discriminant analysis | Decision list | Other
OK
Annuler

A limited search induction tree algorithm (Catlett - 1991)
ID3-IV (Quinlan - 1986)
GID3 (Cheng, Fayyad, Irani & Qian - 1988)
ASSISTANT 86 (Cestnik, Kononenko & Bratko - 1986)
ChAID (Kass - 1980)
C4.5 (Quinlan - 1993)
Improved C4.5 (Rakotomalala & Lallich - 1999)
Improved ChAID (Tschuprow Goodness of Split)
Cost sensitive C4.5 (Rakotomalala & Chauchat - 2001)
One-Vs-All Decision Tree

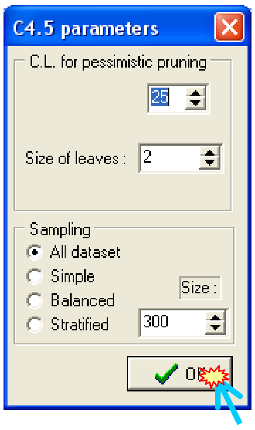
C4.5 (Quinlan - 1993)

Learning method
MethodName=Improved ChAID (Tsc
MethodClassName=TArbreDecision
Hdl=8
Merge=0.05
Split=0.001
TypeBonferroni=1
ValueBonferroni=1
Sampling=0

Examples selection
4601 examples selected
0 examples idle

Editing | NEW.FDM | Attributes : 57 | Examples : 4601
Improved ChAID (Tschuprow Goodness of Split)

When we have selected the learning method, another dialog box enables us to specify the algorithm parameters. We use the default parameters. We validate by clicking OK.



C4.5 parameters

C.L. for pessimistic pruning: 25

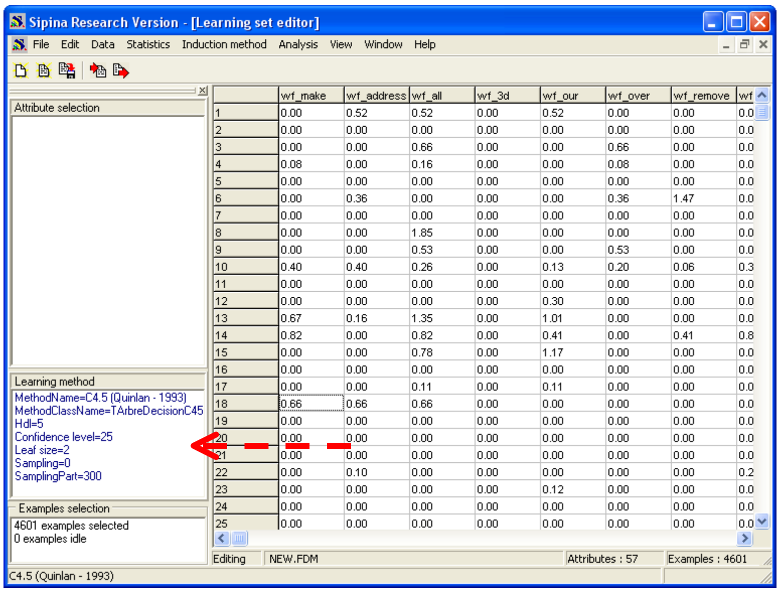
Size of leaves : 2

Sampling
- [✓ All dataset]
- Simple
- Balanced
- Stratified

Size : 300

We choose the C4.5 algorithm because this is a state-of-the-art method. An improved version which takes into account misclassification costs is also available into SIPINA

The parameters appear in the middle part of the project explorer (Figure 2).



Learning method
MethodName=C4.5 (Quinlan - 1993)
MethodClassName=TArbreDecisionC45
Hdl=5
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300

Examples selection
4601 examples selected
0 examples idle

	wf_make	wf_address	wf_all	wf_3d	wf_our	wf_over	wf_remove	wf
1	0.00	0.52	0.52	0.00	0.52	0.00	0.00	0.0
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
3	0.00	0.00	0.66	0.00	0.00	0.66	0.00	0.0
4	0.08	0.00	0.16	0.00	0.00	0.08	0.00	0.0
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
6	0.00	0.36	0.00	0.00	0.00	0.36	1.47	0.0
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
8	0.00	0.00	1.85	0.00	0.00	0.00	0.00	0.0
9	0.00	0.00	0.53	0.00	0.00	0.53	0.00	0.0
10	0.40	0.40	0.26	0.00	0.13	0.20	0.06	0.3
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
12	0.00	0.00	0.00	0.00	0.30	0.00	0.00	0.0
13	0.67	0.16	1.35	0.00	1.01	0.00	0.00	0.0
14	0.82	0.00	0.82	0.00	0.41	0.00	0.41	0.8
15	0.00	0.00	0.78	0.00	1.17	0.00	0.00	0.0
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
17	0.00	0.00	0.11	0.00	0.11	0.00	0.00	0.0
18	0.66	0.66	0.66	0.00	0.00	0.00	0.00	0.0
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
22	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.2
23	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.0
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0

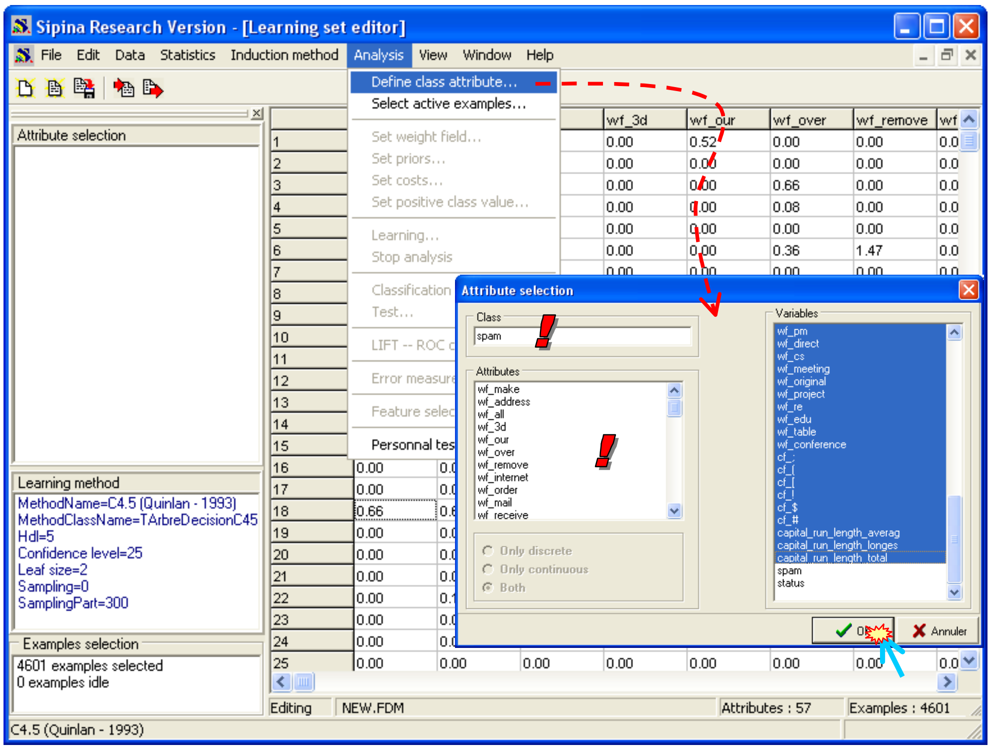
Editing NEW.FDM Attributes : 57 Examples : 4601

C4.5 (Quinlan - 1993)

Figure 2 – C4.5 is the selected method

3.3 Defining the class attribute

We click on the ANALYSIS / DEFINE CLASS ATTRIBUTE menu in order to define the class attribute (SPAM) and the predictive attributes (the other ones, excepting STATUS). In the dialog box which appears, using drag and drop, we do the selection.



Attribute selection

Class: spam

Attributes: wf_make, wf_address, wf_all, wf_3d, wf_our, wf_over, wf_remove, wf_internet, wf_order, wf_mail, wf_receive

Variables: wf_pm, wf_direct, wf_cs, wf_meeting, wf_original, wf_project, wf_re, wf_edu, wf_table, wf_conference, cf_;, cf_(, cf_[, cf_!, cf_$, cf_#, capital_run_length_averag, capital_run_length_longes, capital_run_length_total, spam, status

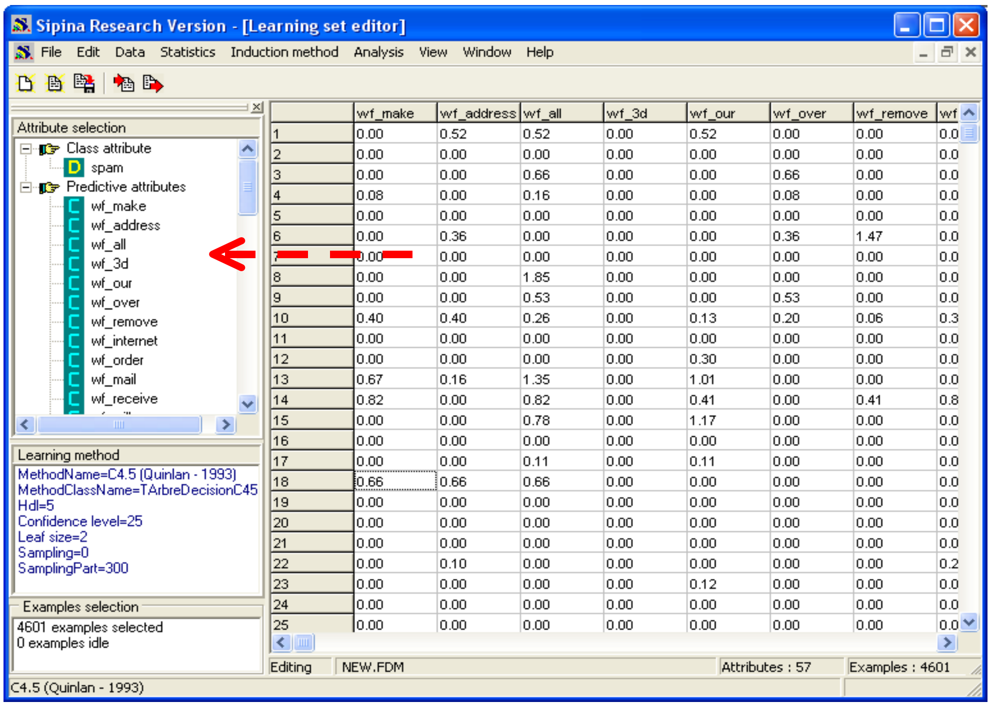
( ) Only discrete
( ) Only continuous
(•) Both

Learning method
MethodName=C4.5 (Quinlan - 1993)
MethodClassName=TArbreDecisionC45
Hdl=5
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300

Examples selection
4601 examples selected
0 examples idle

	wf_3d	wf_our	wf_over	wf_remove	wf
1	0.00	0.52	0.00	0.00	0.0
2	0.00	0.00	0.00	0.00	0.0
3	0.00	0.00	0.66	0.00	0.0
4	0.00	0.00	0.08	0.00	0.0
5	0.00	0.00	0.00	0.00	0.0
6	0.00	0.00	0.36	1.47	0.0

Variable selection appears in the top part of the project explorer.



Attribute selection
- Class attribute
  - D spam
- Predictive attributes
  - C wf_make
  - C wf_address
  - C wf_all
  - C wf_3d
  - C wf_our
  - C wf_over
  - C wf_remove
  - C wf_internet
  - C wf_order
  - C wf_mail
  - C wf_receive

Learning method
MethodName=C4.5 (Quinlan - 1993)
MethodClassName=TArbreDecisionC45
Hdl=5
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300

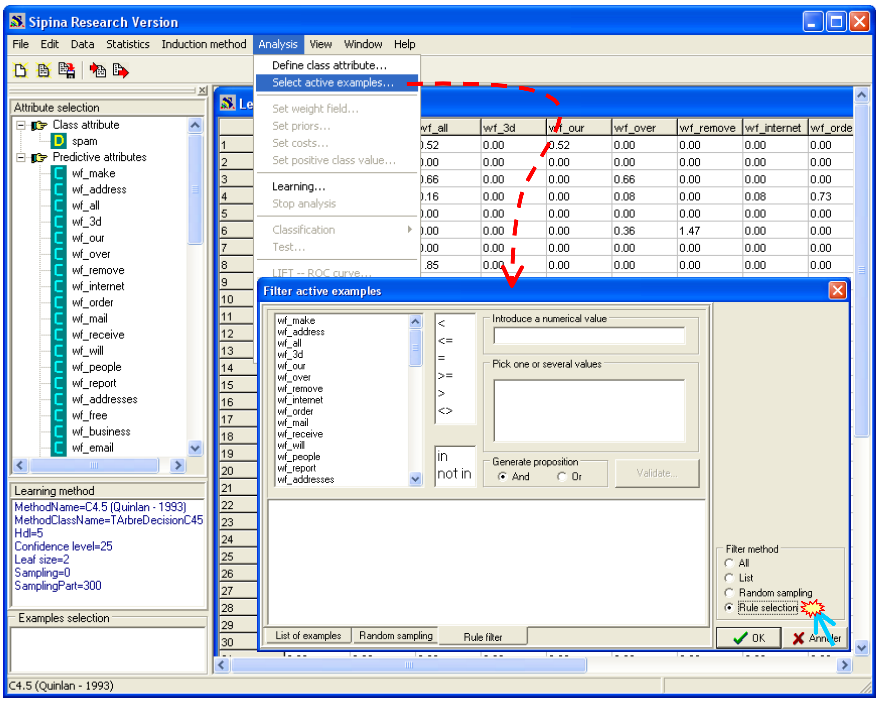
Examples selection
4601 examples selected
0 examples idle

	wf_make	wf_address	wf_all	wf_3d	wf_our	wf_over	wf_remove	wf
1	0.00	0.52	0.52	0.00	0.52	0.00	0.00	0.0
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
3	0.00	0.00	0.66	0.00	0.00	0.66	0.00	0.0
4	0.08	0.00	0.16	0.00	0.00	0.08	0.00	0.0
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
6	0.00	0.36	0.00	0.00	0.00	0.36	1.47	0.0
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
8	0.00	0.00	1.85	0.00	0.00	0.00	0.00	0.0
9	0.00	0.00	0.53	0.00	0.00	0.53	0.00	0.0
10	0.40	0.40	0.26	0.00	0.13	0.20	0.06	0.3
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
12	0.00	0.00	0.00	0.00	0.30	0.00	0.00	0.0
13	0.67	0.16	1.35	0.00	1.01	0.00	0.00	0.0
14	0.82	0.00	0.82	0.00	0.41	0.00	0.41	0.8
15	0.00	0.00	0.78	0.00	1.17	0.00	0.00	0.0
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
17	0.00	0.00	0.11	0.00	0.11	0.00	0.00	0.0
18	0.66	0.66	0.66	0.00	0.00	0.00	0.00	0.0
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
22	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.2
23	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.0
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0

The acronym "D" is used for discrete variable ("C" for continuous attribute).

3.4 Partitioning the dataset into learning and test sets

We use the additional column (STATUS) for this operation. We click on the ANALYSIS / SELECT ACTIVE EXAMPLES menu. A dialog box appears. We select the RULE SELECTION option.

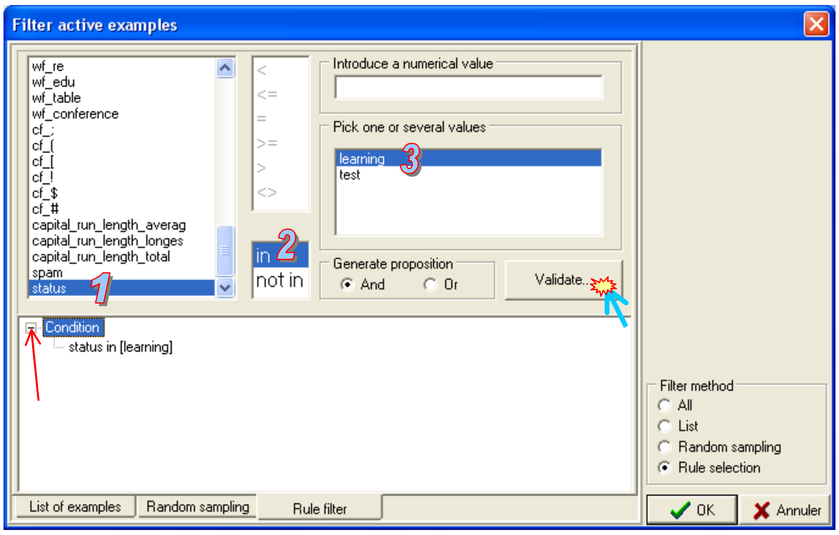


	wf_all	wf_3d	wf_our	wf_over	wf_remove	wf_internet	wf_orde
1	.52	0.00	0.52	0.00	0.00	0.00	0.00
2	.00	0.00	0.00	0.00	0.00	0.00	0.00
3	.66	0.00	0.00	0.66	0.00	0.00	0.00
4	.16	0.00	0.00	0.08	0.00	0.08	0.73
5	.00	0.00	0.00	0.00	0.00	0.00	0.00
6	.00	0.00	0.00	0.36	1.47	0.00	0.00
7	.00	0.00	0.00	0.00	0.00	0.00	0.00
8	.85	0.00	0.00	0.00	0.00	0.00	0.00

```
MethodName=C4.5 (Quinlan - 1993)
MethodClassName=TArbreDecisionC45
Hdl=5
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300
```

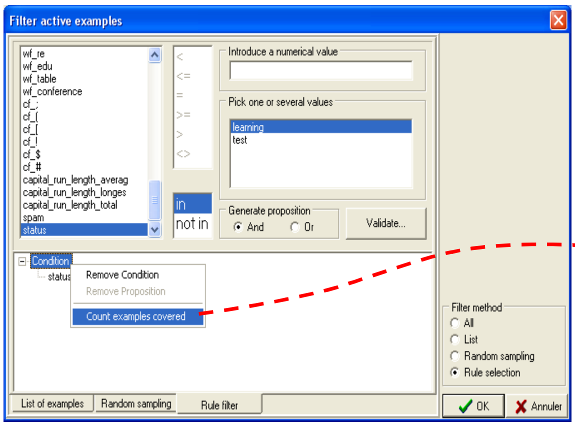
We intend to specify that the examples which are STATUS = LEARNING correspond to the learning sample. We work first in the top part of the window. We have to do the following steps: (1) We select the STATUS attribute in the list box. (2) We select the IN condition. (3) Then, we select the LEARNING value in the left part of the window.

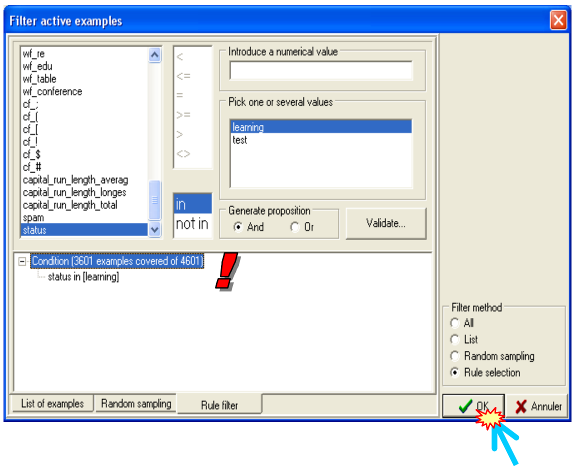
We validate this rule selection by clicking the VALIDATE button. The rule is available in the bottom part of the window.



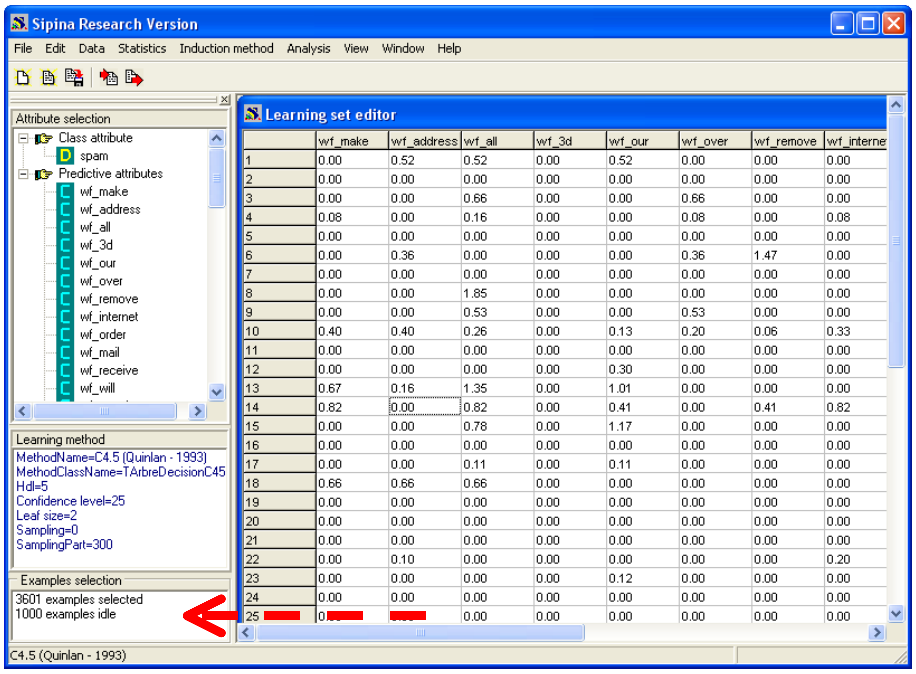
```
Condition
   status in [learning]
```

We can count the covered examples by using the contextual menu. We observe that 3601 examples are available for the learning process.





We click the OK button. The selection is displayed in the bottom part of the project explorer.



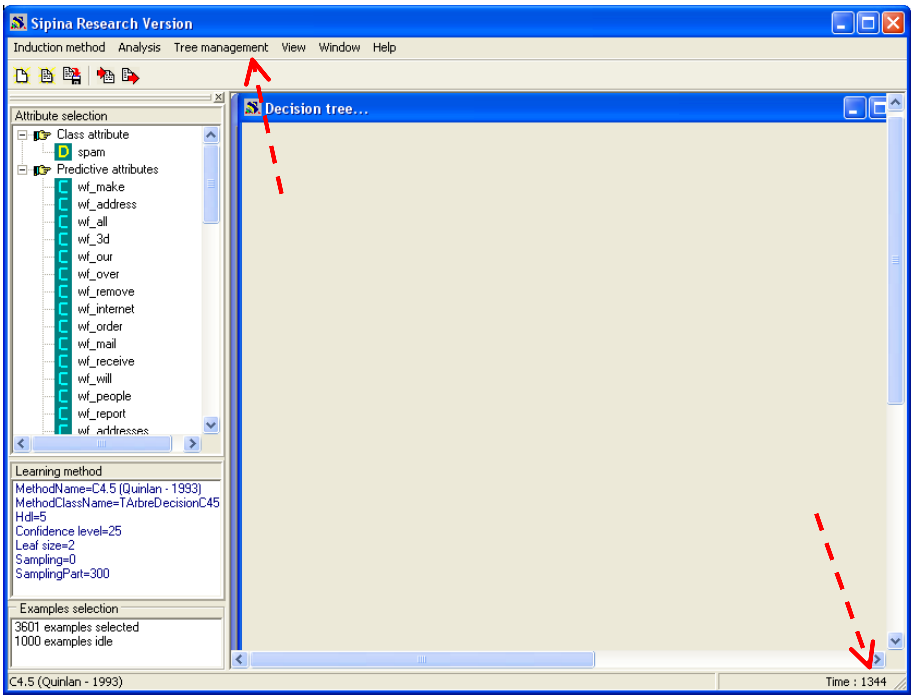
MethodName=C4.5 (Quinlan - 1993)
MethodClassName=TArbreDecisionC45
Hdl=5
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300

3601 examples selected
1000 examples idle

	wf_make	wf_address	wf_all	wf_3d	wf_our	wf_over	wf_remove	wf_interne
1	0.00	0.52	0.52	0.00	0.52	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.66	0.00	0.00	0.66	0.00	0.00
4	0.08	0.00	0.16	0.00	0.00	0.08	0.00	0.08
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.36	0.00	0.00	0.00	0.36	1.47	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	1.85	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.53	0.00	0.00	0.53	0.00	0.00
10	0.40	0.40	0.26	0.00	0.13	0.20	0.06	0.33
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.30	0.00	0.00	0.00
13	0.67	0.16	1.35	0.00	1.01	0.00	0.00	0.00
14	0.82	0.00	0.82	0.00	0.41	0.00	0.41	0.82
15	0.00	0.00	0.78	0.00	1.17	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.11	0.00	0.11	0.00	0.00	0.00
18	0.66	0.66	0.66	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.20
23	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	[illegible]	[illegible]	0.00	0.00	0.00	0.00	0.00	0.00

3.5 Decision tree learning

We start the learning phase by clicking the ANALYSIS / LEARNING menu. The window seems strangely empty. This is because the tree is very large. We must use the TREE MANAGEMENT / ZOOM OUT menu in order to show the whole tree.



Learning method

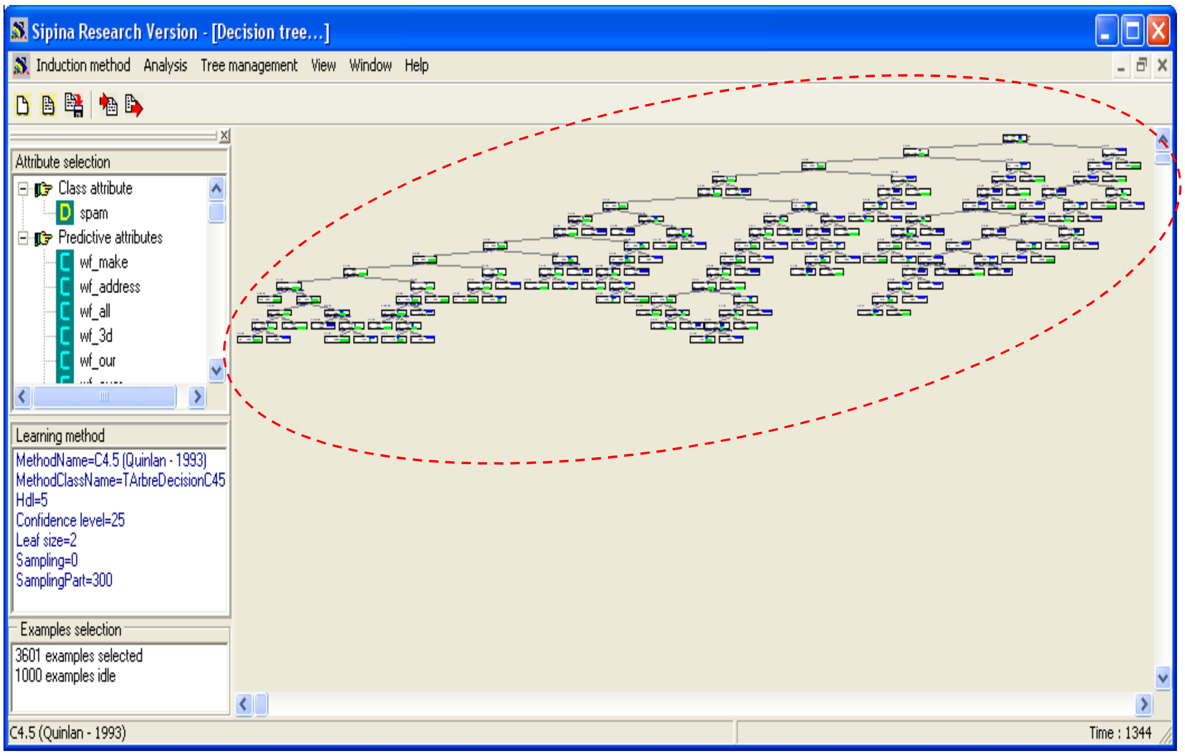
MethodName=C4.5 (Quinlan - 1993)
MethodClassName=TArbreDecisionC45
Hdl=5
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300

Examples selection

3601 examples selected
1000 examples idle

C4.5 (Quinlan - 1993)

We can use also the shortcut CTRL + Q.



Learning method

MethodName=C4.5 (Quinlan - 1993)
MethodClassName=TArbreDecisionC45
Hdl=5
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300

Examples selection

3601 examples selected
1000 examples idle

C4.5 (Quinlan - 1993)

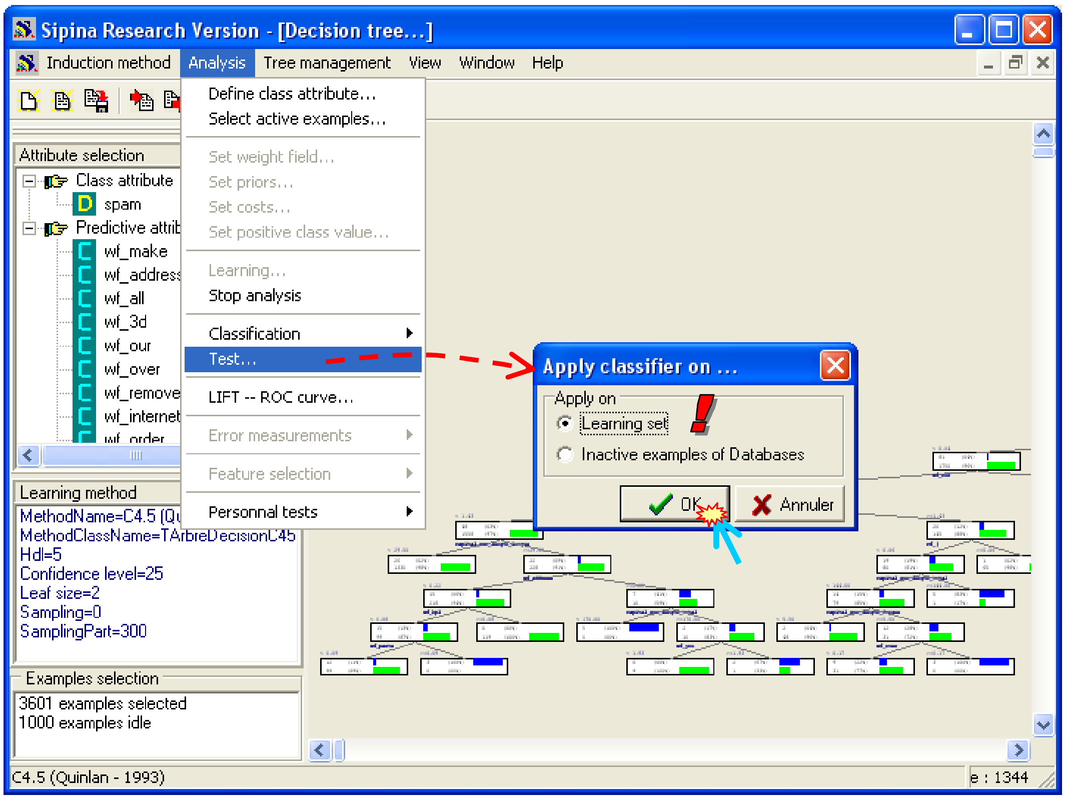
The tree is very large. We cannot really extract useful knowledge from these rules. But it is not the goal in this tutorial. We want only to elaborate an accurate classifier.

3.6 Error rate measurement

3.6.1 Confusion matrix and resubstitution error rate

First, we want to compare the true and the predicted values of the class attribute on the learning set. We obtain a confusion matrix and the resubstitution error rate.

In order to obtain this matrix, we activate the ANALYSIS /TEST menu. In the dialog box, we select first the learning set.



MethodName=C4.5 (Qu
MethodClassName=TArbreDecisionC45
Hdl=5
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300

3601 examples selected
1000 examples idle

C4.5 (Quinlan - 1993)

Apply on: [✓ Learning set] Inactive examples of Databases

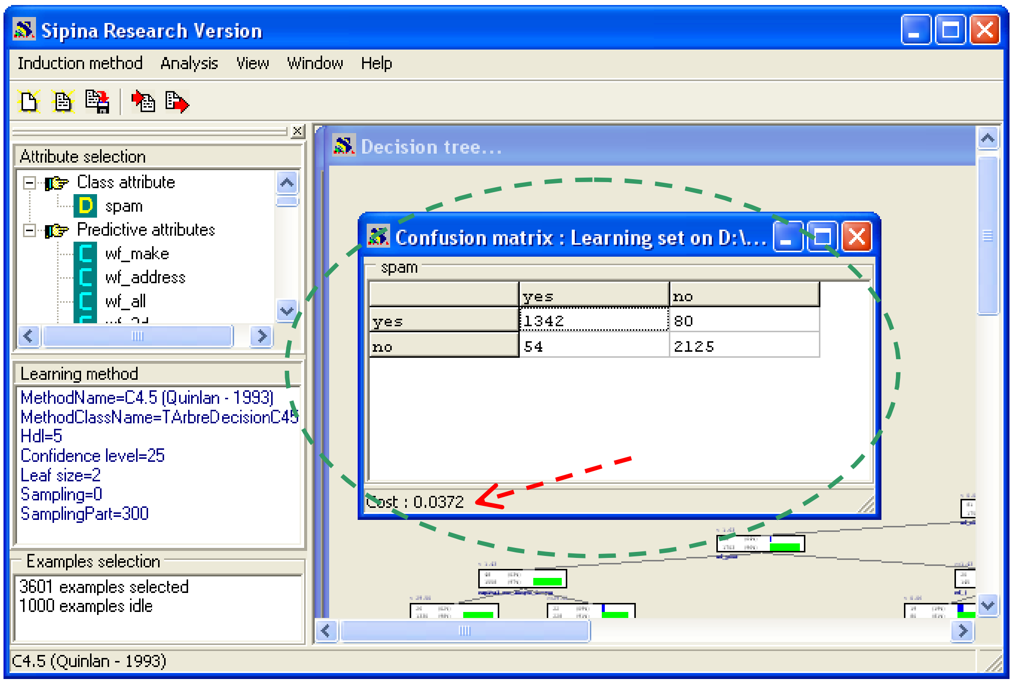
The confusion matrix and the error rate are displayed in a new window (Figure 3). We read:

- 80 "spam" are classified as regular messages;
- 54 regular messages are classified as "spam";
- The other messages are properly classified. (1342 "spam", 2125 regular messages).

In the following confusion matrix:

Row : true values Column : predicted values	Positive prediction	Negative prediction
+ (« positive » is "spam")	a	b
-	c	d

We compute the error rate as follows: $e = \dfrac{b+c}{a+b+c+d}$



spam

	yes	no
yes	1342	80
no	54	2125

Cost : 0.0372

MethodName=C4.5 (Quinlan - 1993)
MethodClassName=TArbreDecisionC45
Hdl=5
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300

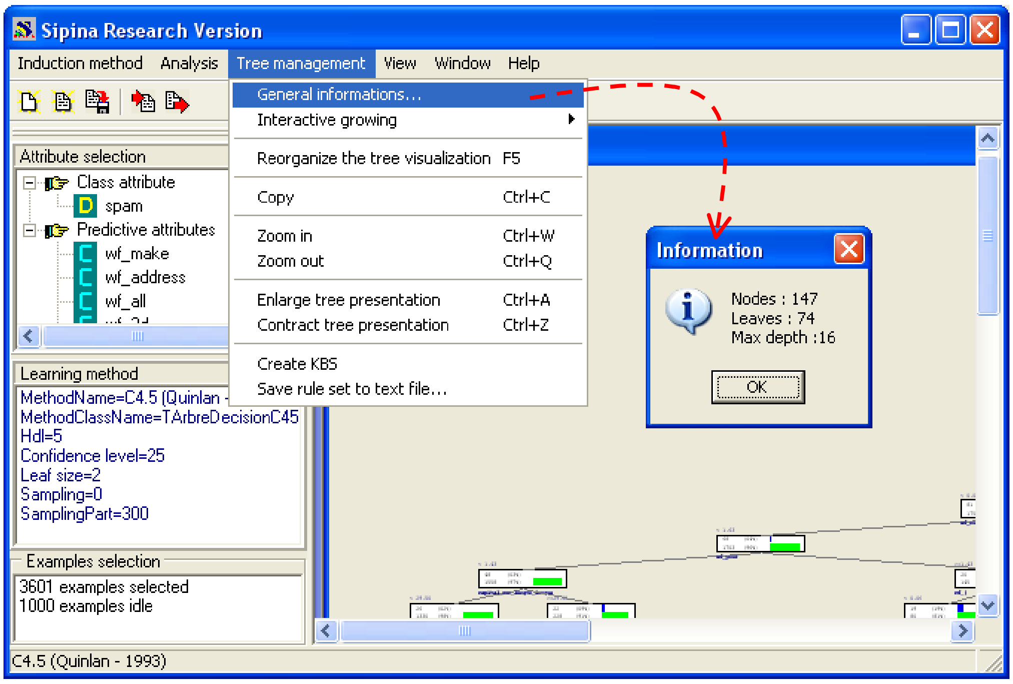
3601 examples selected
1000 examples idle

Figure 3 – Confusion matrix and resubstitution error rate

The resubstitution error rate is

$$e_r = \frac{80+54}{3601} = 3.7\%$$

The classification tree seems accurate. But, this result is not really credible. We use the same dataset in order to learn and evaluate the model. The resulting error rate is often optimist, especially about the decision tree with many rules (Figure 4).



Nodes : 147
Leaves : 74
Max depth :16

MethodName=C4.5 (Quinlan -
MethodClassName=TArbreDecisionC45
Hdl=5
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300

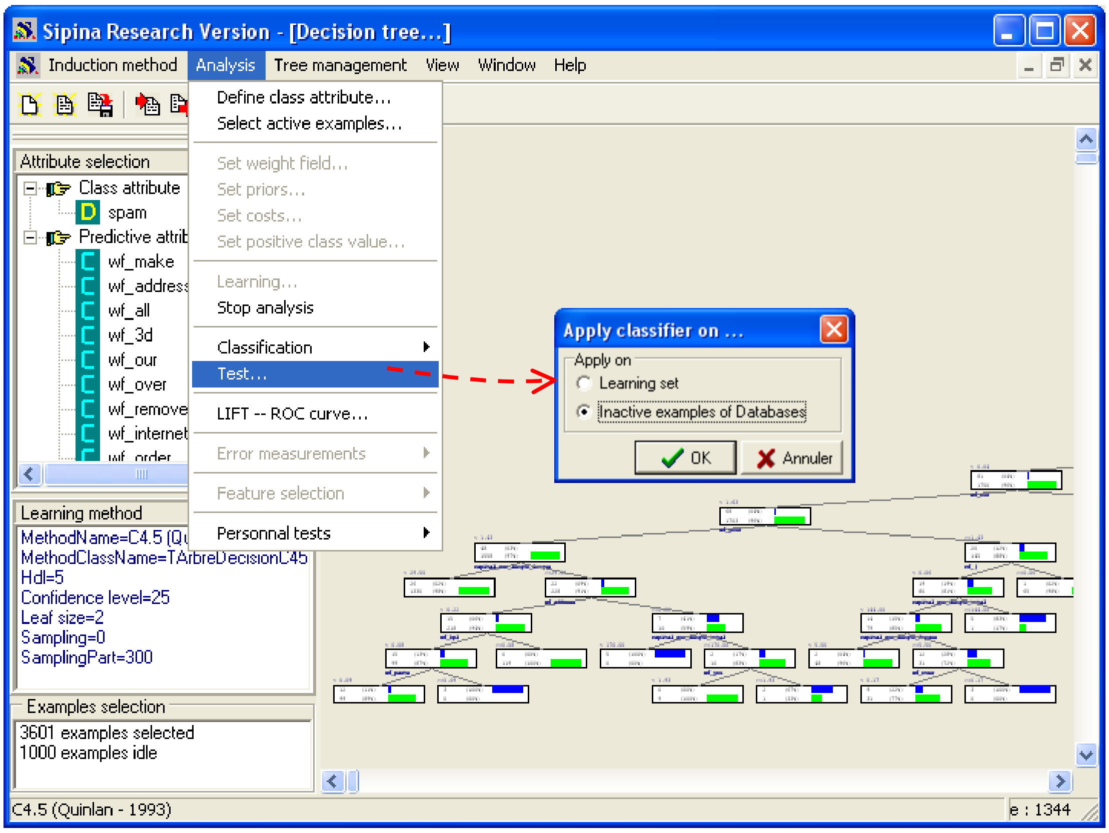
3601 examples selected
1000 examples idle

Figure 4 – Tree characteristics

3.6.2 Confusion matrix and test error rate

In order to obtain an unbiased error rate evaluation, we must use the test set.

Again, we click on the ANALYSIS / TEST menu. But we choose now the INACTIVE EXAMPLES OF DATABASES option.

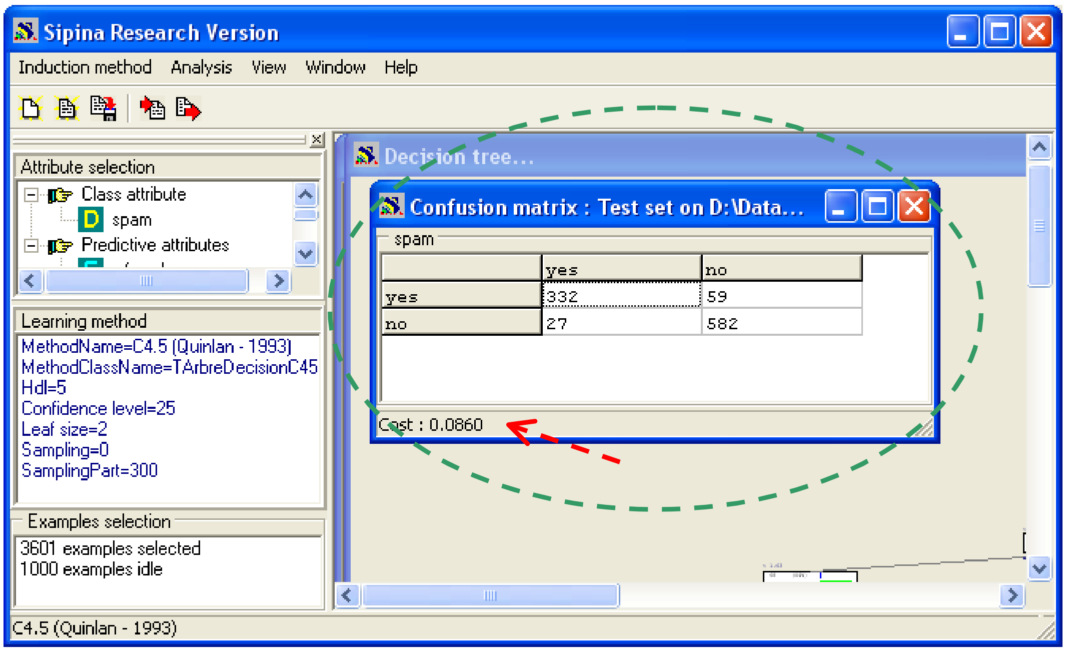


Apply on: Learning set [✓ Inactive examples of Databases]

MethodName=C4.5 (Qu
MethodClassName=TArbreDecisionC45
Hdl=5
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300

3601 examples selected
1000 examples idle

If we compute the sum of the values of the matrix, we have 1000, the test sample size.



MethodName=C4.5 (Quinlan - 1993)
MethodClassName=TArbreDecisionC45
Hdl=5
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300

3601 examples selected
1000 examples idle

spam

	yes	no
yes	332	59
no	27	582

Cost : 0.0860

The test error rate is

$$e_t = \frac{59+27}{1000} = 8.6\%$$

The true probability of misclassification of the tree on unseen cases is 8.6%.

3.6.3 Other indicators: recall, precision and F-Measure

In our study and it is the case very often, the two values of the class attribute has not the same importance, nor the same significance. Predicting a spam as a normal message when the message is automatically filtered on the server is not really problematic. The user must delete them manually. On the other hand, predicting a normal message as a spam is very harmful. In fact, the message is automatically deleted, the user is not informed about this deletion, even if it is a very important message. The error rate does not take into account this difference. This measure gives the same importance to the two kinds of misclassification.

For this reason, in the text categorization context, the users prefer two more relevant measures to the domain:

- The recall $r = \frac{a}{a+b}$ is the fraction of "spam" which is recovered. In our test set, we have r = 332/(332 + 59) = 84.9%.
- The precision $p = \frac{a}{a+c}$ is the fraction of true positive among the message predicted as "spam". In our study, p = 332/(332 + 27) = 92.5.

A good classifier must have a high recall AND a high precision.

To handle two values simultaneously is never quite easy, especially when we want to compare the performances of several models. One proposes in the literature a synthetic criterion: the F-MEASURE. The formula is the following; it is a weighted harmonic mean of recall and precision.

$$F_\alpha = \frac{(1+\alpha^2) \times r \times p}{\alpha^2 \times p + r}$$

In the best case: r = 1, p = 1, then F = 1.

α is the weight. If we set the same weight to recall and precision, we will choose $\alpha = 1$, then the F-Measure will be $F_1 = \frac{2 \times 0.925 \times 0.849}{1 \times 0.925 + 0.849} = 0.885$.

If we set $\alpha = 0.5$, we give twice more importance to precision compared with recall; if we set $\alpha = 2$, we give twice more importance to recall compared with precision (see http://en.wikipedia.org/wiki/Information_retrieval).

The good value of α depends on the constraints of the domain and the goal of the study. We can also define a range of values to study the behavior of the models on various scenarios.

4 Using non-symmetrical misclassification costs

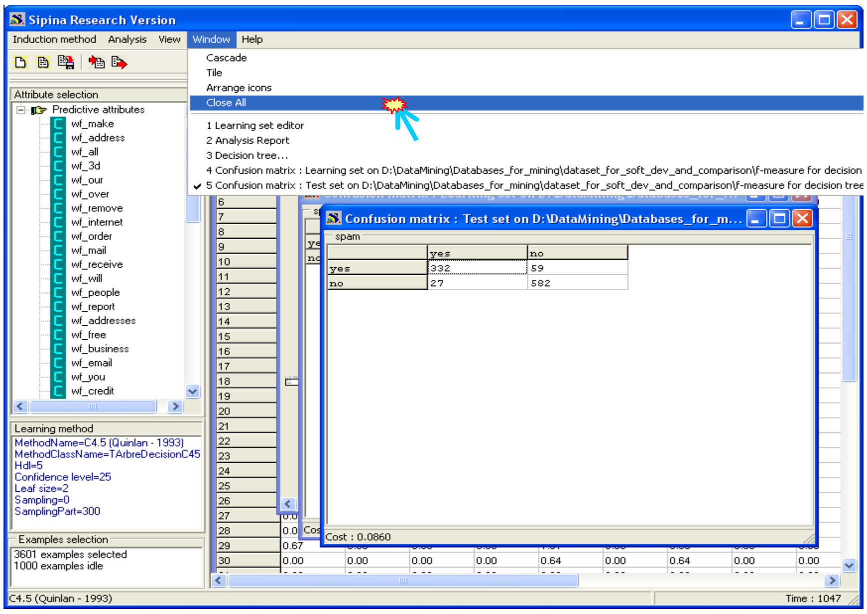
In our spam detection problem, we would like to give ten times more importance to precision compared with recall. How can we insert this information in the decision tree induction process?

Concerning the evaluation, we modify the parameter of F-Measure by adopting the weight $\alpha = 0.1$. But how can we incorporate this information in the induction process?

The simplest approach is to integrate this new constraint in the form of misclassification costs matrix, and to use a method which takes into account this matrix explicitly. SIPINA implements a cost-sensitive decision tree induction approach, derived from C4.5 algorithm (Chauchat et al., 2001). In the following, we select this approach; then, we insert the misclassification cost matrix.

4.1 Stopping the preceding analysis

First, we must interrupt the preceding analysis. We click on the WINDOW / CLOSE ALL menu. All the windows of analysis are closed.



Learning method
MethodName=C4.5 (Quinlan - 1993)
MethodClassName=TArbreDecisionC45
Hdl=5
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300

Examples selection
3601 examples selected
1000 examples idle

spam

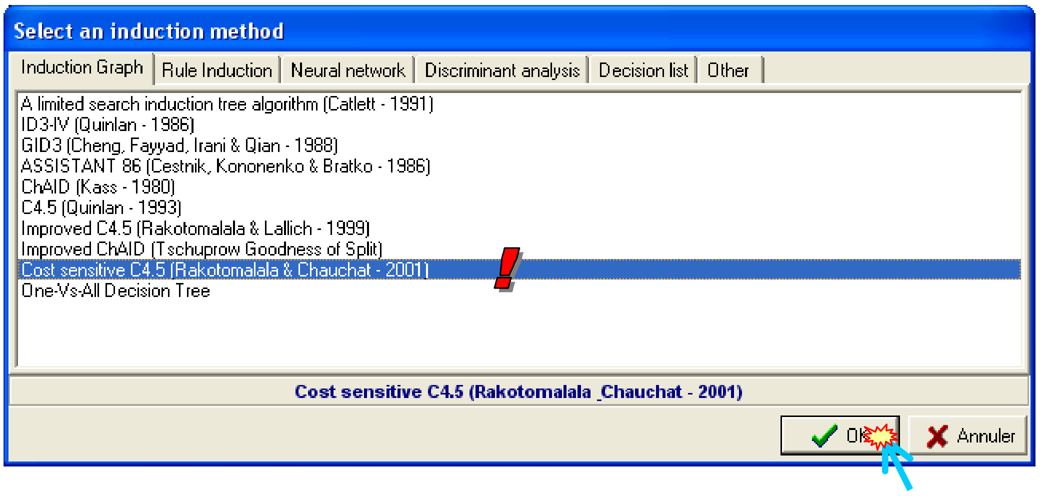
	yes	no
yes	332	59
no	27	582

Cost : 0.0860

C4.5 (Quinlan - 1993) Time : 1047

4.2 Selecting the cost sensitive approach

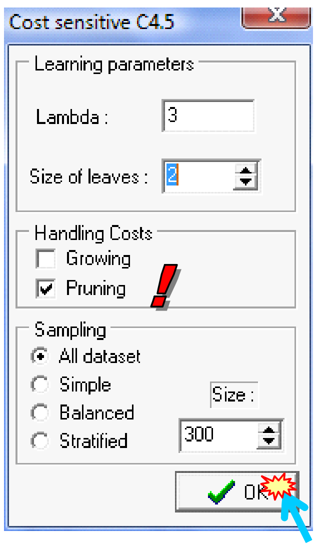
We click again on the INDUCTION METHOD / STANDARD ALGORITHM menu in order to choose the appropriate method. In the dialog box, we choose COST SENSITIVE C4.5 (CHAUCHAT & RAKOTOMALALA, 2001).


Select an induction method

A limited search induction tree algorithm (Catlett - 1991)
ID3-IV (Quinlan - 1986)
GID3 (Cheng, Fayyad, Irani & Qian - 1988)
ASSISTANT 86 (Cestnik, Kononenko & Bratko - 1986)
ChAID (Kass - 1980)
C4.5 (Quinlan - 1993)
Improved C4.5 (Rakotomalala & Lallich - 1999)
Improved ChAID (Tschuprow Goodness of Split)
Cost sensitive C4.5 (Rakotomalala & Chauchat - 2001)
One-Vs-All Decision Tree

Cost sensitive C4.5 (Rakotomalala _Chauchat - 2001)

In the parameter settings, we can insert the costs matrix during the growing phase (not really efficient) or/and the pruning phase (efficient).


Cost sensitive C4.5

Learning parameters
Lambda : 3
Size of leaves : 2

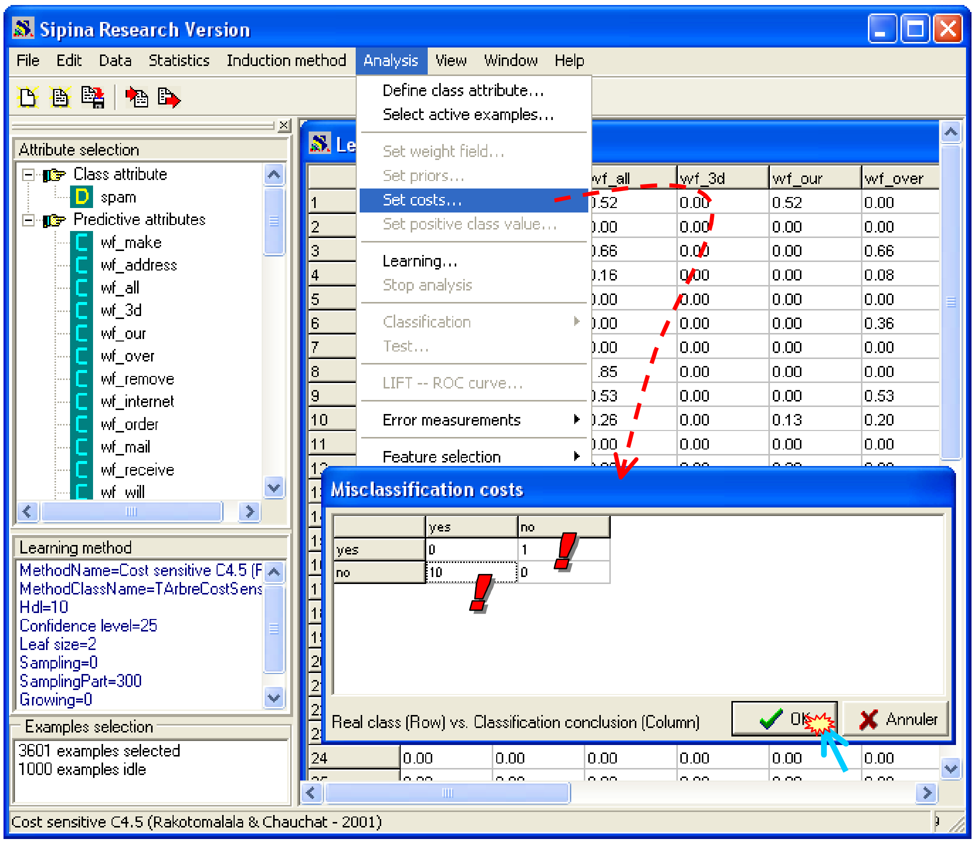
Handling Costs
Growing [ ]
Pruning [✓]

Sampling
[✓ All dataset]
Simple
Balanced
Stratified
Size : 300

The LAMBDA parameter is used for the laplacian estimation of the class distribution within nodes. It can be understand as a smoothing parameter for the probability estimation.

4.3 Defining the misclassification cost matrix

We must define the misclassification costs. We click on the ANALYSIS / SET COSTS menu. In the dialog box, we insert the values. We validate by clicking on the OK button.



Misclassification costs

	yes	no
yes	0	1
no	10	0

Real class (Row) vs. Classification conclusion (Column)

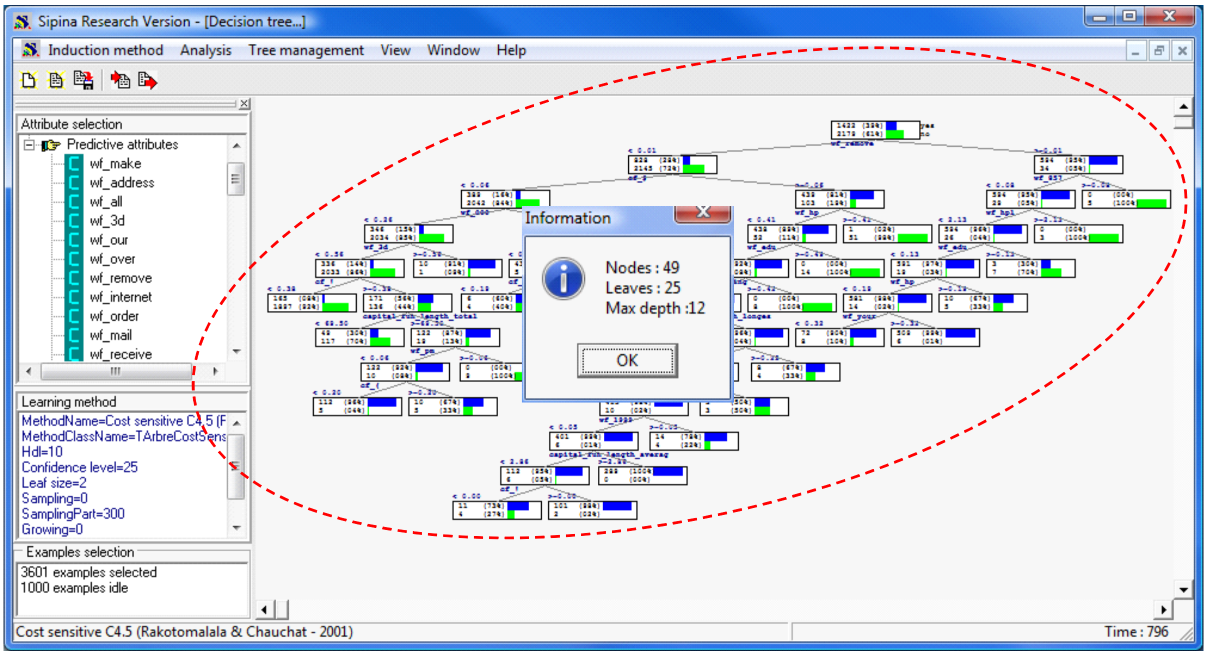
Learning method
MethodName=Cost sensitive C4.5 (F
MethodClassName=TArbreCostSens
Hdl=10
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300
Growing=0

Examples selection
3601 examples selected
1000 examples idle

Cost sensitive C4.5 (Rakotomalala & Chauchat - 2001)

4.4 Training and testing the tree

We start the learning phase with the ANALYSIS / LEARNING menu. The tree is shorter than the preceding, with 36 leaves however.



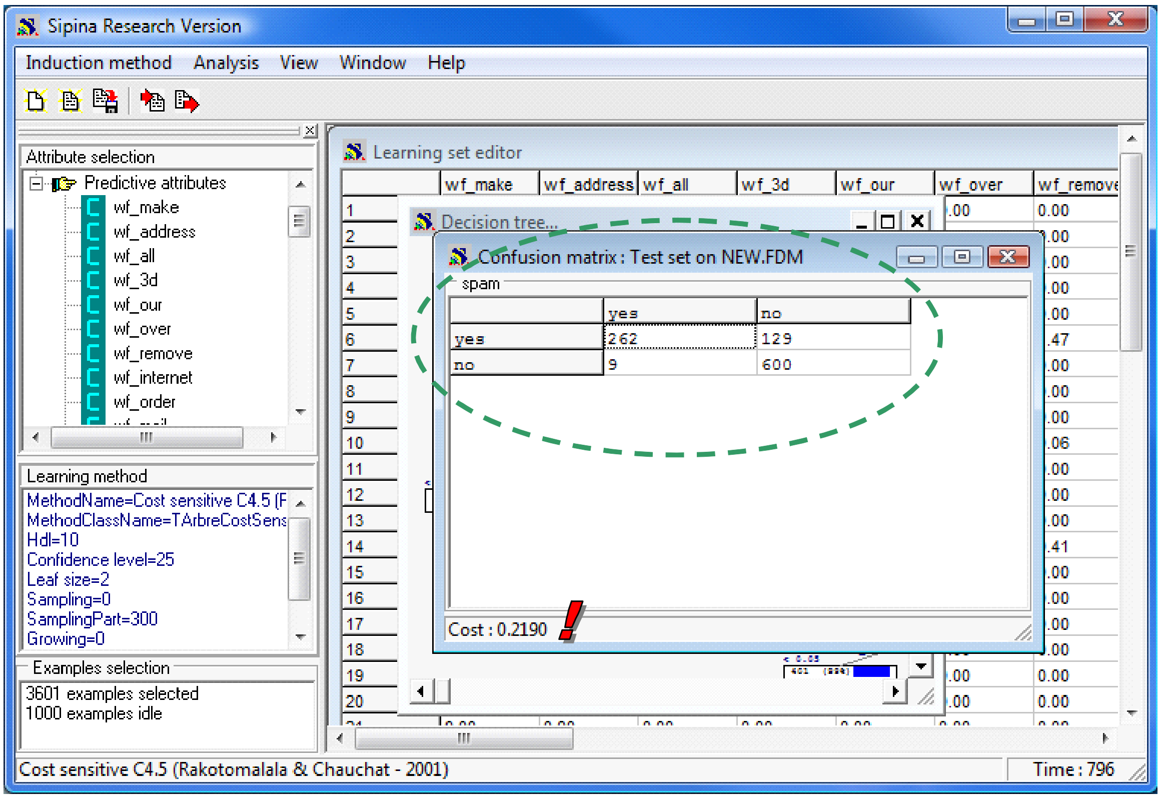
Nodes : 49
Leaves : 25
Max depth :12

Learning method
MethodName=Cost sensitive C4.5 (F
MethodClassName=TArbreCostSens
Hdl=10
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300
Growing=0

Examples selection
3601 examples selected
1000 examples idle

Cost sensitive C4.5 (Rakotomalala & Chauchat - 2001) Time : 796

We apply now the tree on the test set (ANALYSIS / TEST / INACTIVE EXAMPLES OF DATABASES). We obtain the following confusion matrix.



MethodName=Cost sensitive C4.5 (F
MethodClassName=TArbreCostSens
Hdl=10
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300
Growing=0

3601 examples selected
1000 examples idle

spam

	yes	no
yes	262	129
no	9	600

Cost : 0.2190

Cost sensitive C4.5 (Rakotomalala & Chauchat - 2001)   Time : 796

We compare this result with the "standard" C4.5 method (Tableau 1).

Indicator	**C4.5 « standard » (M1)**	**C4.5 « cost sensitive » (M2)**
True positive	332	262
False positive	27	9
Error rate	8.6%	13.8%
Recall	84.9%	67.0%
Precision	92.5%	96.7%
F-Measure (0.1)	0.924	0.963

Tableau 1 – Comparison of the two approaches

According the error rate, M2 seems worst. But our main preoccupation is the precision. We observe that M2 is significantly better; it is also better according the F-Measure.

5 Conclusion

This tutorial had a double goal. The first was to show that it is possible to carry out the "train - test" sequence on a predefined subdivision of the dataset with SIPINA. That makes it possible to compare the performances of several algorithms implemented in SIPINA, but also implemented in other software which can handle this kind of subdivision.

The second important subject treated is the integration of the misclassification costs during the construction of the models. This functionality is unfortunately unusual in the Data Mining Software. It is implemented in SIPINA. It appeared interesting to show how to implement it, what to think about it, and how to read the results in this configuration.