1 Objectif

Importer un fichier de format WEKA dans SIPINA. Partitionner les individus en apprentissage/test. Construire et évaluer un arbre de décision.

WEKA [1] est un logiciel de Data Mining libre très populaire dans la communauté « Machine Learning ». Il intègre un grand nombre de méthodes, articulées essentiellement autour des approches supervisées et non supervisées[2].

WEKA peut importer différents formats de fichier. Mais il propose surtout un format propriétaire (*.ARFF) qui est un format texte, avec des spécifications ad hoc sur documenter les variables. Importer un fichier ARFF ne pose donc pas de problèmes particuliers, dès lors que l'on sait appréhender un fichier texte.

De manière générale, le format texte présente l'avantage de la souplesse. Nous pouvons manipuler le fichier dans n'importe quel éditeur de texte, en dehors du logiciel, pour le visualiser, voire le corriger. En revanche, par rapport au format binaire, il est présente l'inconvénient de la lenteur. A chaque chargement dans le logiciel, il doit être décomposé, interprété. Il est difficile d'organiser une lecture par blocs efficace. Lorsque la base est d'une taille raisonnable, quelques milliers d'observations et des dizaines de variables, la différence de performances est peu perceptible.

Le format ARFF est composé de 3 parties. La partie haute correspond aux éventuels commentaires destinés à décrire la base. Chaque ligne de commentaire doit commencer par le caractère « % ».

```
%1. Title: Johns Hopkins University Ionosphere database
%
%2. Source Information:
%   -- Donor: Vince Sigillito (vgs@aplcen.apl.jhu.edu)
%   -- Date: 1989
%   -- Source: Space Physics Group
%               Applied Physics Laboratory
%               Johns Hopkins University
%               Johns Hopkins Road
%               Laurel, MD 20723
%
%3. Past Usage:
%   -- Sigillito, V. G., Wing, S. P., Hutton, L. V., \& Baker, K. B. (1989).
%      Classification of radar returns from the ionosphere using neural
%      networks. Johns Hopkins APL Technical Digest, 10, 262-266.
%...
```

La partie intermédiaire, initiée par le mot clé « @relation », correspond au dictionnaire des variables. Si la variable est discrète, les modalités sont énumérées. Par défaut, pour une tâche supervisée, la variable à prédire doit être placée en dernière position.

[1] http://www.cs.waikato.ac.nz/ml/weka/

[2] Voir http://eric.univ-lyon2.fr/~ricco/data-mining/logiciels/ pour un comparatif de quelques logiciels gratuits et open source de Data Mining.

```
@relation ionosphere

@attribute a01 real
@attribute a02 real
@attribute a03 real
@attribute a04 real
@attribute a05 real
@attribute a06 real
@attribute a07 real
@attribute a08 real
@attribute a09 real
@attribute a10 real
```

Enfin, la troisième partie du fichier, commencée par le mot clé « @data » correspond à la description des observations. Les données ne sont pas encodées. Dans la grande majorité des cas, on pourrait l'exploiter sans tenir compte du dictionnaire. Notons que le point décimal est forcément le « . », quelle que soit la configuration du système d'exploitation.

```
@data

1,0,0.99539,-0.05889,0.85243,0.02306,0.83398,-0.37708,1,0.03760,0.85243,-
0.17755,0.59755,-0.44945,0.60536,-0.38223,0.84356,-0.38542,0.58212,-
0.32192,0.56971,-0.29674,0.36946,-0.47357,0.56811,-0.51171,0.41078,-
0.46168,0.21266,-0.34090,0.42267,-0.54487,0.18641,-0.45300,g
1,0,1,-0.18829,0.93035,-0.36156,-0.10868,-0.93597,1,-0.04549,0.50874,-
0.67743,0.34432,-0.69707,-0.51685,-0.97515,0.05499,-0.62237,0.33109,-1,-0.13151,-
0.45300,-0.18056,-0.35734,-0.20332,-0.26569,-0.20468,-0.18401,-0.19040,-0.11593,-
0.16626,-0.06288,-0.13738,-0.02447,b
…
```

Nous notons, avec un certain amusement, que l'on retrouve là l'ancien format de fichier de SIPINA (version 2.5 et antérieures). A la différence que nous utilisions 2 fichiers : « *.PAR » pour le dictionnaire des données, « *.DAT » pour les observations. Le fichier « .DAT » exploitait le dictionnaire pour obtenir un format plus compact et cohérent, c'est un point positif. Mais l'obligation de manipuler 2 fichiers perturbait les personnes qui essayaient de prendre en main le logiciel. Ce format a été abandonné sur la version suivante de SIPINA (l'actuelle).

Le format ARFF comporte des qualités et des défauts, comme tout format. En revanche, un des facteurs de succès du logiciel est d'avoir converti une très grande majorité des fichiers du serveur UCI (http://archive.ics.uci.edu/ml/datasets.html) qui sert de référence pour l'étalonnage des méthodes (les fameux « benchmarks ») dans les publications en apprentissage automatique. Les chercheurs se sont engouffrés dans la brèche, ils ont tôt fait de s'approprier un outil qui permet de traiter des données déjà préparées. Cela permet de produire ces fameux grands tableaux comparatifs, essentiellement basé sur le taux d'erreur, où l'on tente avec plus ou moins de bonheur de montrer l'avantage décisif de sa méthode par rapport aux standards du domaine.

Dans ce didacticiel, nous montrons comment charger un fichier au format ARFF dans SIPINA. Nous initions par la suite une analyse très classique où, après avoir subdivisé le fichier en données « apprentissage » et données « test », nous construisons puis évaluons un arbre de décision élaborée à l'aide de la méthode C4.5 (Quinlan, 1993).

2 Données

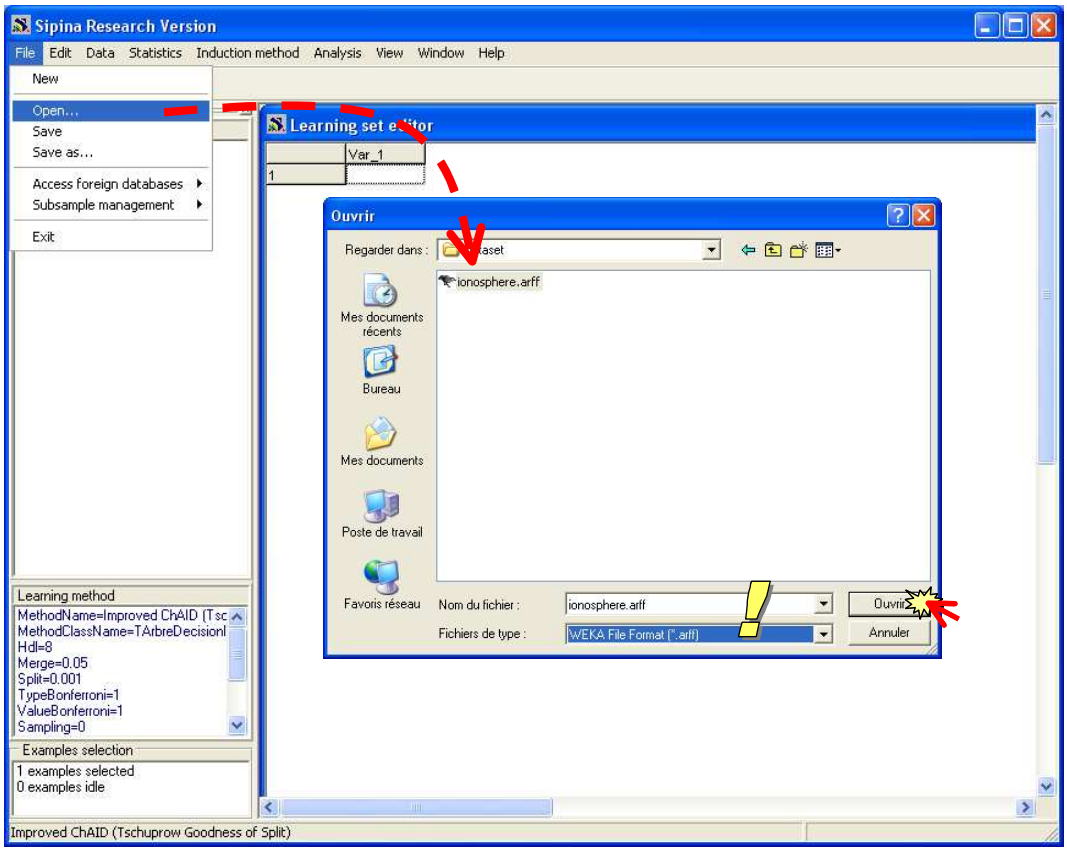
Pour illustrer ce didacticiel, nous utilisons le fichier IONOSPHERE.ARFF, accessible sur le serveur UCI[3]. La variable à prédire « CLASS » est binaire « good » et « bad » ; 34 variables prédictives, toutes continues, sont disponibles.

3 Traitement dans SIPINA

3.1 Importation d'un fichier ARFF

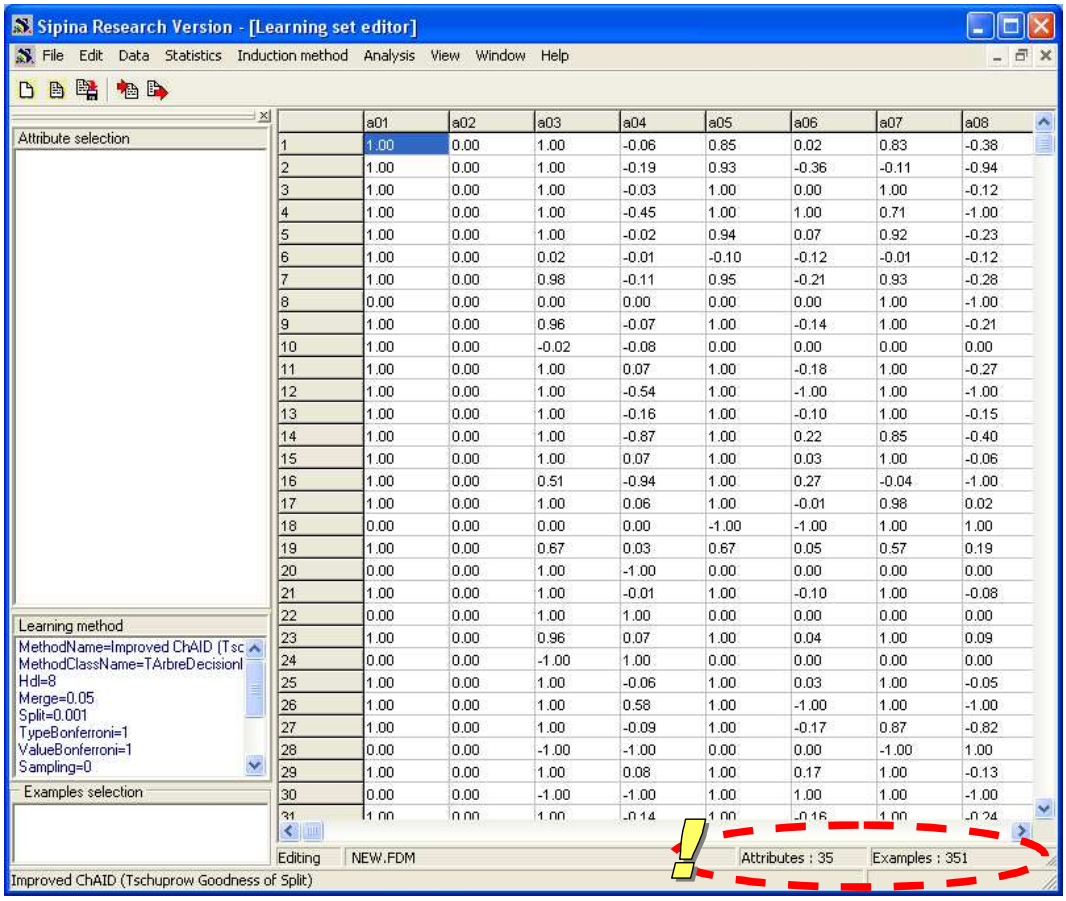
Nous démarrons SIPINA à partir de l'item dans le menu Windows DEMARRER / PROGRAMMES / STATISTICAL PACKAGE / SUPERVISED LEARNING. L'interface est subdivisée en différentes parties. Pour l'instant la grille des données est vide.

Pour charger les données, nous activons le menu FILE / OPEN. Une boîte de dialogue apparaît, nous spécifions le répertoire adéquat. Parmi les types de fichiers gérés, nous choisissons le format WEKA FILE FORMAT (*.ARFF). Nous sélectionnons le fichier IONOSPHERE.ARFF, puis nous validons.



Les données sont automatiquement chargées dans la grille, ce qui permet de vérifier d'emblée le succès de l'importation. Nous double cliquons sur la barre de titre de fenêtre contenant les données, nous voyons mieux la barre d'état : 35 attributs et 351 observations sont disponibles.

[3] http://archive.ics.uci.edu/ml/datasets/Ionosphere. Le fichier est accessible directement sur notre serveur à l'URL suivant : http://eric.univ-lyon2.fr/~ricco/dataset/ionosphere.arff



	a01	a02	a03	a04	a05	a06	a07	a08
1	1.00	0.00	1.00	-0.06	0.85	0.02	0.83	-0.38
2	1.00	0.00	1.00	-0.19	0.93	-0.36	-0.11	-0.94
3	1.00	0.00	1.00	-0.03	1.00	0.00	1.00	-0.12
4	1.00	0.00	1.00	-0.45	1.00	1.00	0.71	-1.00
5	1.00	0.00	1.00	-0.02	0.94	0.07	0.92	-0.23
6	1.00	0.00	0.02	-0.01	-0.10	-0.12	-0.01	-0.12
7	1.00	0.00	0.98	-0.11	0.95	-0.21	0.93	-0.28
8	0.00	0.00	0.00	0.00	0.00	0.00	1.00	-1.00
9	1.00	0.00	0.96	-0.07	1.00	-0.14	1.00	-0.21
10	1.00	0.00	-0.02	-0.08	0.00	0.00	0.00	0.00
11	1.00	0.00	1.00	0.07	1.00	-0.18	1.00	-0.27
12	1.00	0.00	1.00	-0.54	1.00	-1.00	1.00	-1.00
13	1.00	0.00	1.00	-0.16	1.00	-0.10	1.00	-0.15
14	1.00	0.00	1.00	-0.87	1.00	0.22	0.85	-0.40
15	1.00	0.00	1.00	0.07	1.00	0.03	1.00	-0.06
16	1.00	0.00	0.51	-0.94	1.00	0.27	-0.04	-1.00
17	1.00	0.00	1.00	0.06	1.00	-0.01	0.98	0.02
18	0.00	0.00	0.00	0.00	-1.00	-1.00	1.00	1.00
19	1.00	0.00	0.67	0.03	0.67	0.05	0.57	0.19
20	0.00	0.00	1.00	-1.00	0.00	0.00	0.00	0.00
21	1.00	0.00	1.00	-0.01	1.00	-0.10	1.00	-0.08
22	0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00
23	1.00	0.00	0.96	0.07	1.00	0.04	1.00	0.09
24	0.00	0.00	-1.00	1.00	0.00	0.00	0.00	0.00
25	1.00	0.00	1.00	-0.06	1.00	0.03	1.00	-0.05
26	1.00	0.00	1.00	0.58	1.00	-1.00	1.00	-1.00
27	1.00	0.00	1.00	-0.09	1.00	-0.17	0.87	-0.82
28	0.00	0.00	-1.00	-1.00	0.00	0.00	-1.00	1.00
29	1.00	0.00	1.00	0.08	1.00	0.17	1.00	-0.13
30	0.00	0.00	-1.00	-1.00	1.00	1.00	1.00	-1.00
31	1.00	0.00	1.00	-0.14	1.00	-0.16	1.00	-0.24

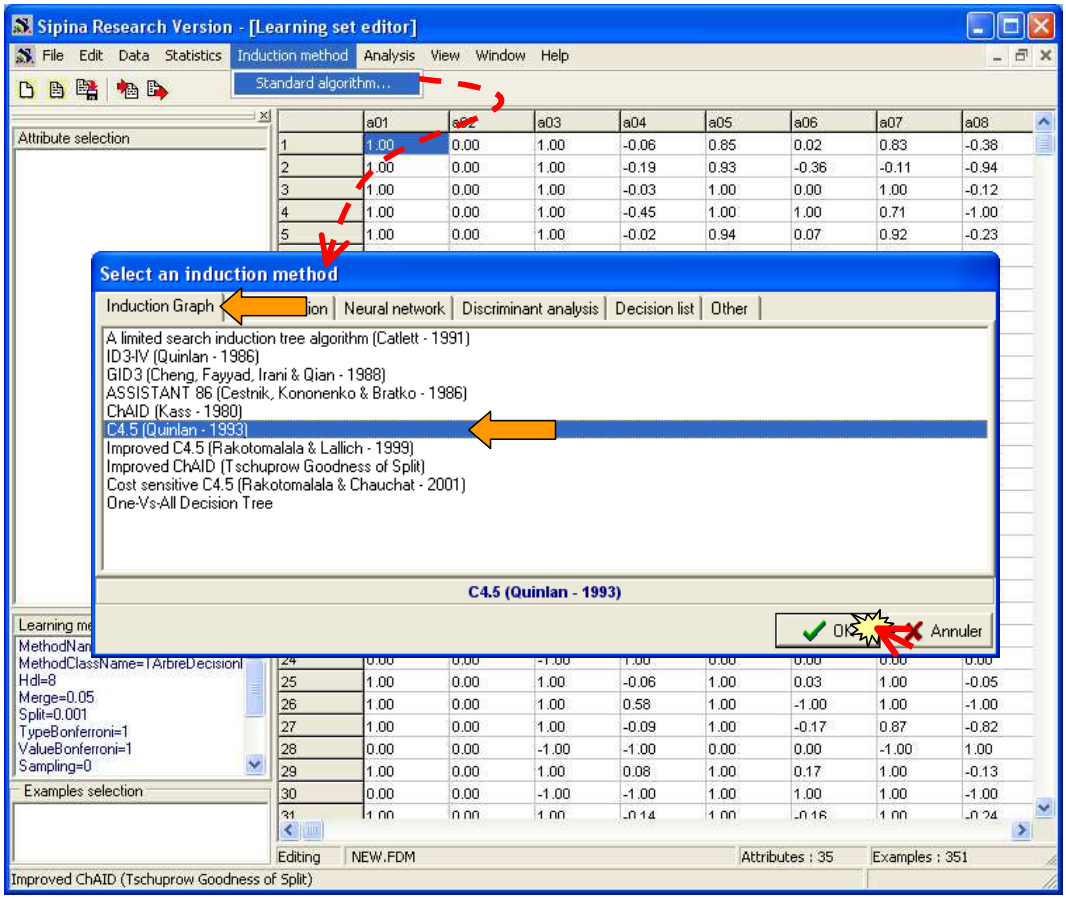
A partir de maintenant, nous pouvons initier une analyse classique. Dans ce tutoriel, nous choisissons d'instancier et de valider un arbre de décision.

3.2 Choix de la méthode d'apprentissage

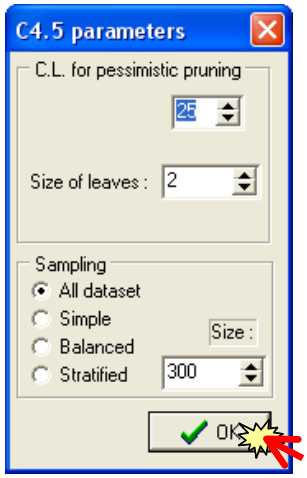
Nous choisissons la méthode C4.5, très connue dans la communauté de l'apprentissage automatique. Notre implémentation reprend fidèlement la description proposée dans l'ouvrage « C4.5 – Programs for Machine Learning » de Quinlan (1993). Il intègre surtout les deux grandes particularités de cette approche : le choix des variables de segmentation à l'aide du « gain ratio », le post élagage à l'aide de l'erreur pessimiste.

L'auteur par ailleurs distribue un code source qui comporte un nombre assez important d'options destinées à bonifier la recherche des solutions. Mais à vrai dire, ils sont très peu connus.

Pour activer une méthode d'apprentissage dans SIPINA, nous cliquons sur le menu INDUCTION METHOD / STANDARD ALGORITHM. Une boîte de dialogue apparaît, nous sélectionnons l'item C4.5 (QUINLAN – 1993), puis nous validons.

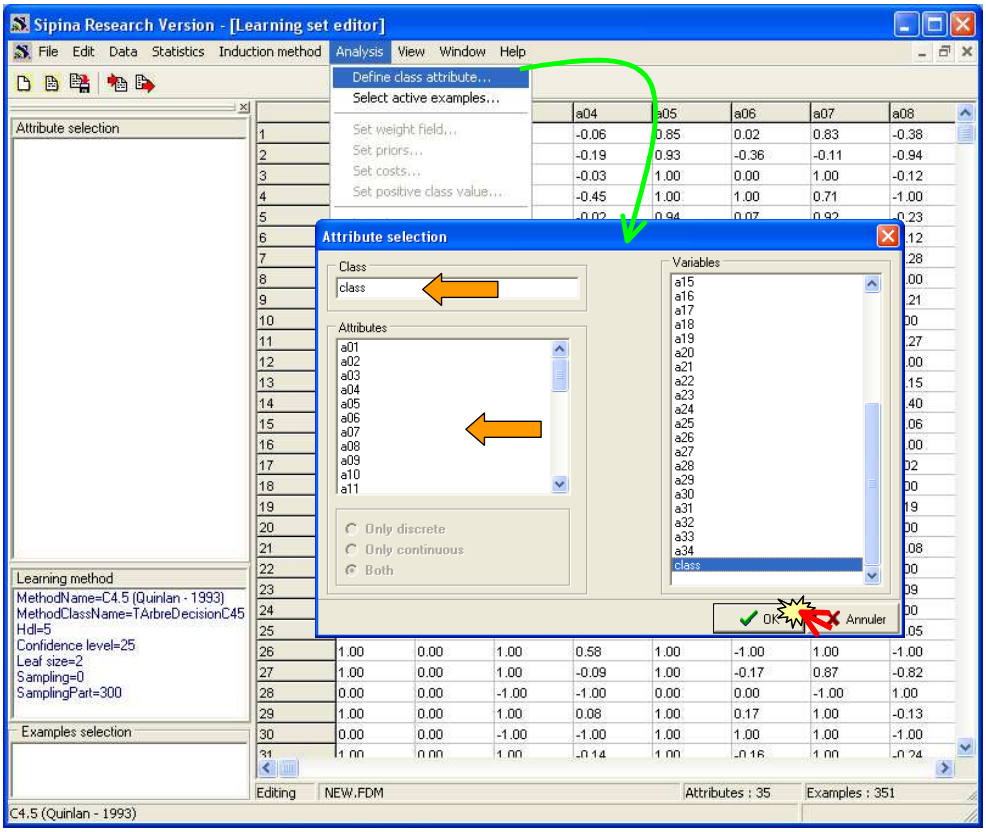


Une seconde fenêtre apparaît, elle permet de spécifier les paramètres de l'algorithme. Nous nous en tenons aux valeurs par défaut dans notre cas, nous validons directement.

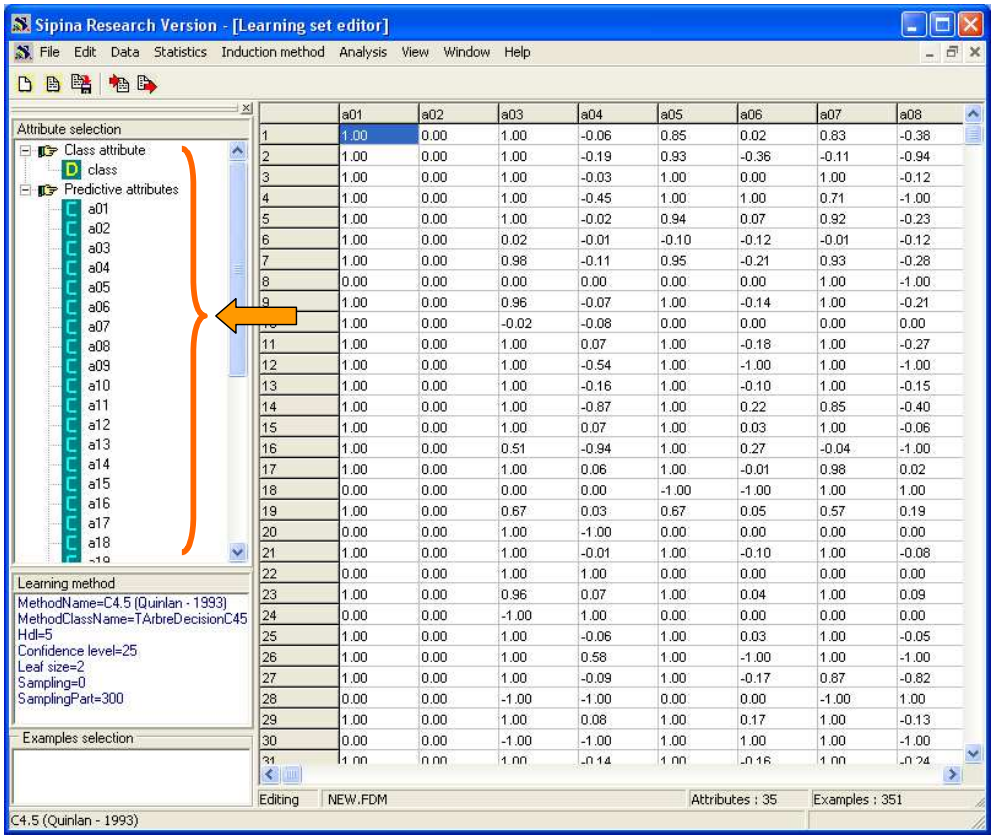


3.3 Statut des variables dans l'analyse

Nous voulons prédire la variable CLASS à partir des descripteurs continus. Pour spécifier cette information, nous activons le menu ANALYSIS / DEFINE CLASS ATTRIBUTE. Dans la boîte de dialogue qui vient, nous plaçons, par glisser déposer, CLASS en « CLASS », et les autres variables, de a01 à a34, en « ATTRIBUTES ».



Après validation, le statut des variables est résumé dans la partie haute de l'explorateur de projet SIPINA.

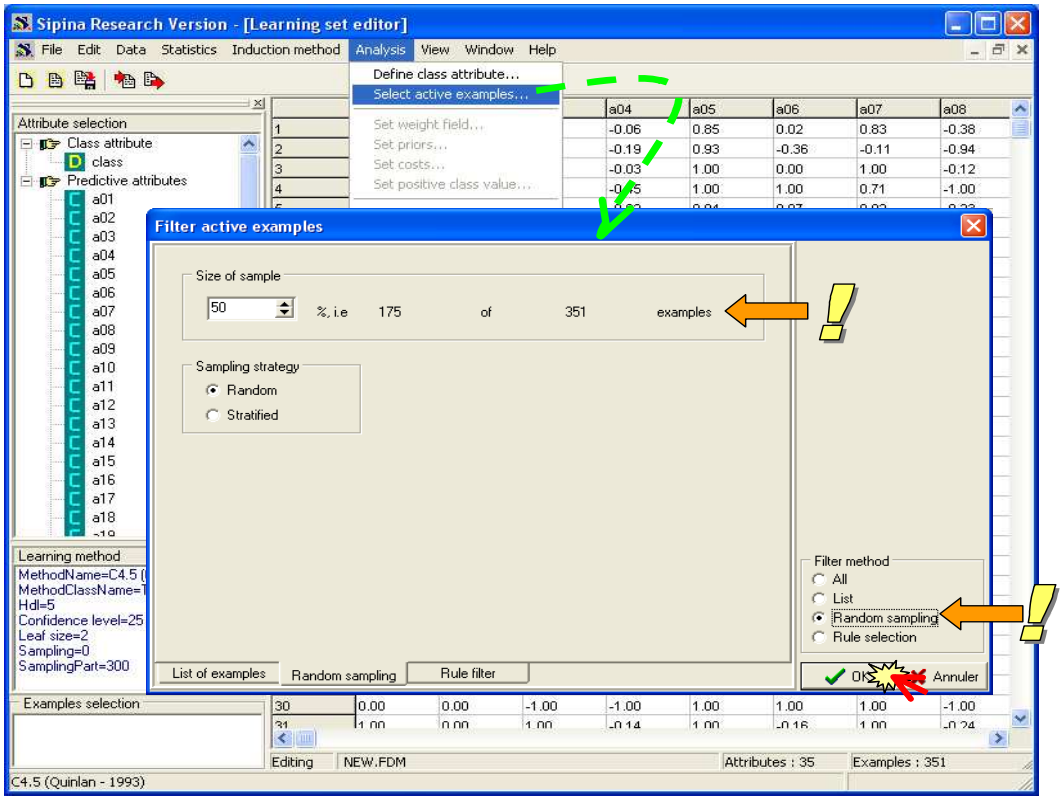


3.4 Subdivision apprentissage et test des données

Dernière étape avant la construction de l'arbre, nous devons subdiviser aléatoirement les données en 2 parties distinctes. La première, dite échantillon d'apprentissage, sert à construire l'arbre selon l'algorithme demandé. Nous obtenons un modèle prédictif. Il est important par la suite de le valider c.-à-d. en connaître la précision en classement lorsqu'il sera déployé dans la population[4]. Pour cela nous utiliserons la seconde fraction des données, dite échantillon test, qui n'a pas pris part à la construction du modèle.

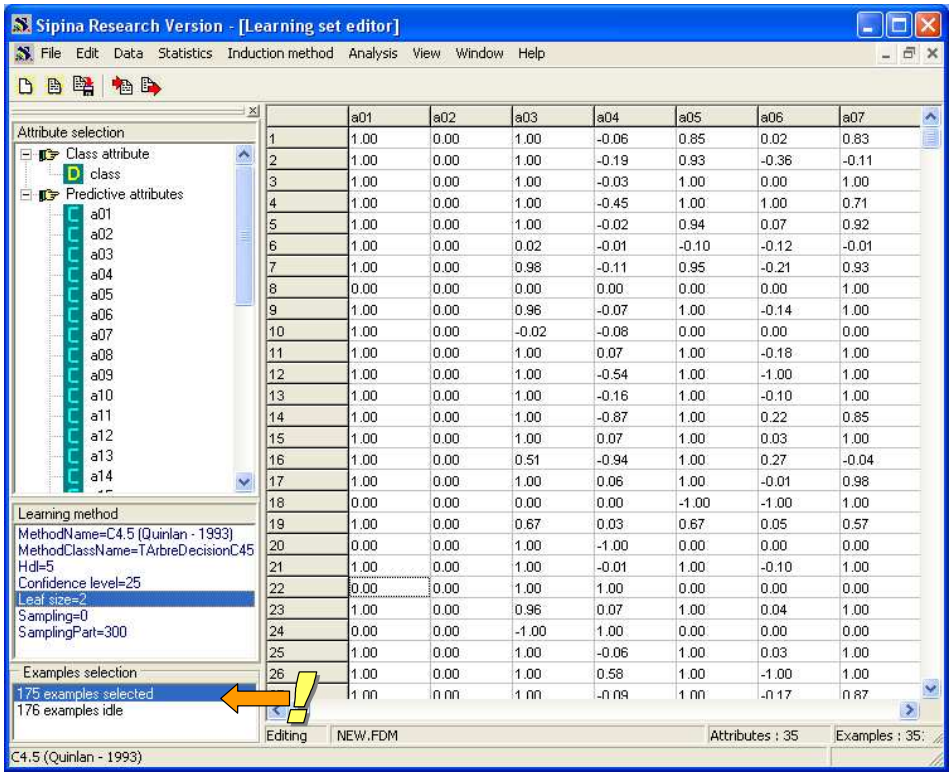
En effet, pour évaluer un classifieur, il suffit de l'appliquer sur un ensemble de données et de mesurer la proportion de mauvaises prédictions. Mais il faut faire attention. La tentation est grande de ré appliquer le modèle sur les données qui ont servi à construire l'arbre, elles ont le mérite d'être déjà disponibles. Hélas, dans la majorité des cas, les performances mesurées dans ces circonstances sont très souvent optimistes, laissant croire à tort à un apprentissage parfait. La démarche est d'autant plus biaisée que l'optimisme dépend des données et des caractéristiques de la méthode. S'agissant des arbres de décision justement, plus ils seront profonds, plus il faudra se méfier du taux d'erreur mesuré sur l'échantillon d'apprentissage. C'est pour cela qu'il est important de réserver une partie des données pour l'évaluation du modèle. La proportion de l'erreur mesurée à partir d'un ensemble test, de taille suffisamment importante, est non biaisée.

Pour subdiviser aléatoirement les données, nous activons le menu ANALYSIS / SELECT ACTIVE EXAMPLES. Dans la boîte de dialogue qui apparaît, nous choisissons le partitionnement aléatoire RANDOM SAMPLING : 50% des individus (175) seront réservés à l'apprentissage, le reste (351 - 175 = 176) pour la validation.



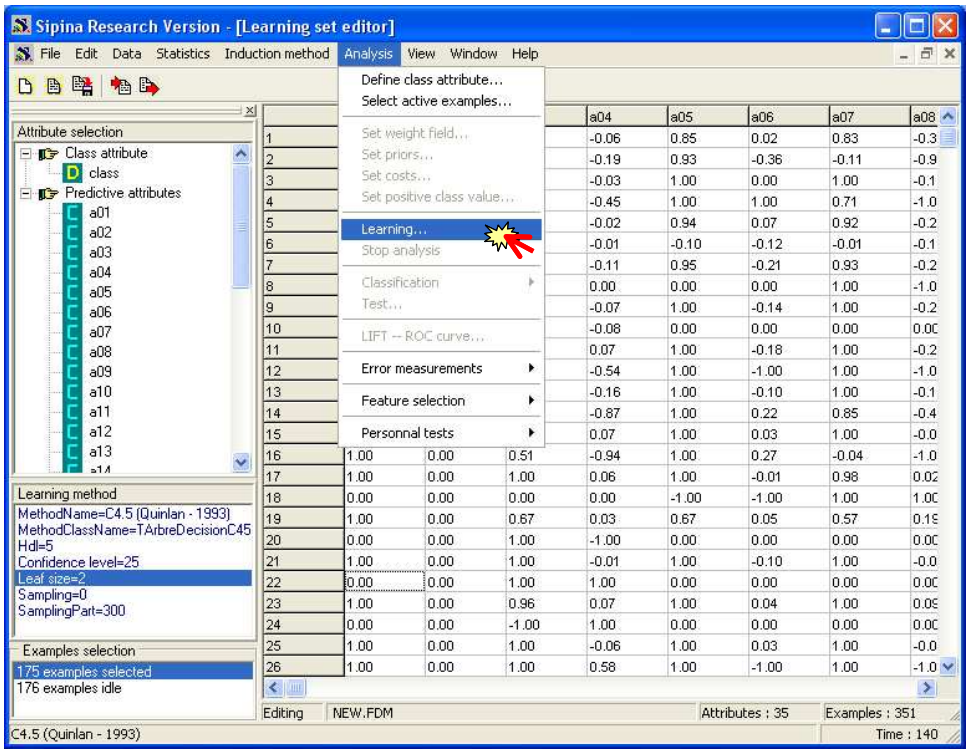
[4] La démarche est purement mécanique dans ce cas. D'autres types de validation existent, la validation experte notamment, elle intègre les connaissances du domaine pour apprécier pleinement les résultats proposés par l'arbre.

Les résultats sont repris dans la partie basse de l'explorateur de projet SIPINA.

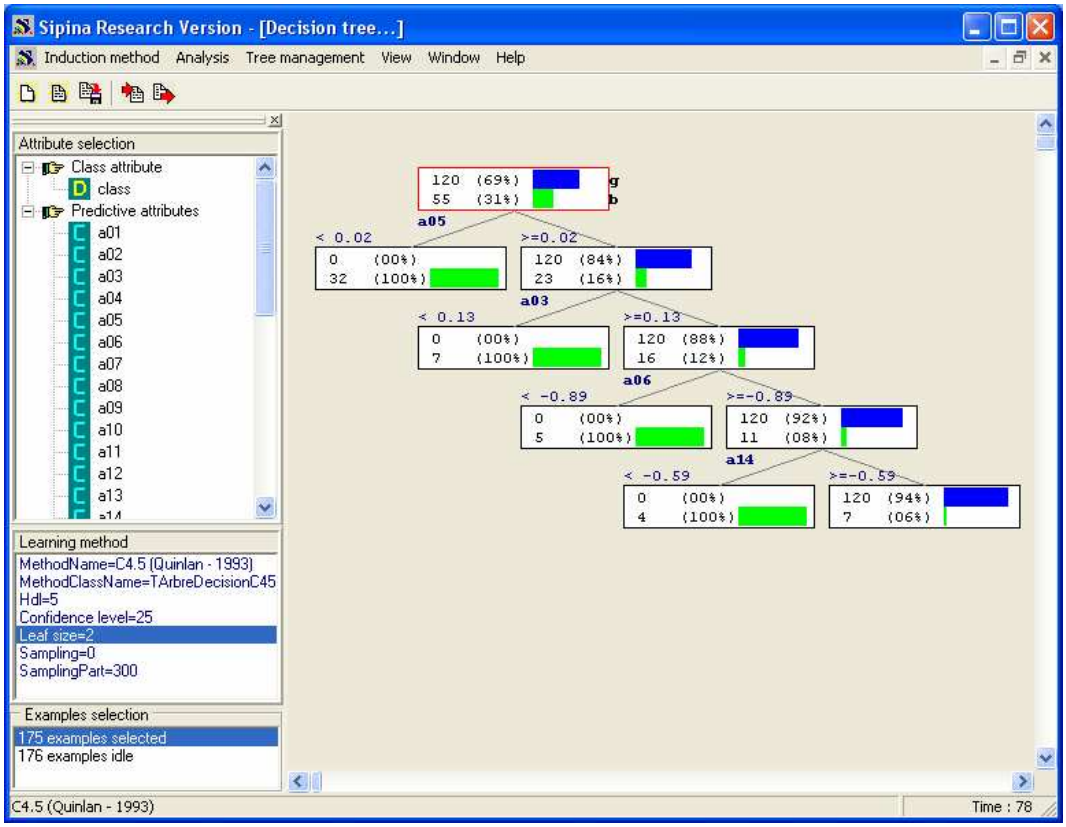


3.5 Construction de l'arbre (échantillon d'apprentissage)

Il ne nous reste plus qu'à construire l'arbre de décision. Nous activons le menu ANALYSIS / LEARNING.



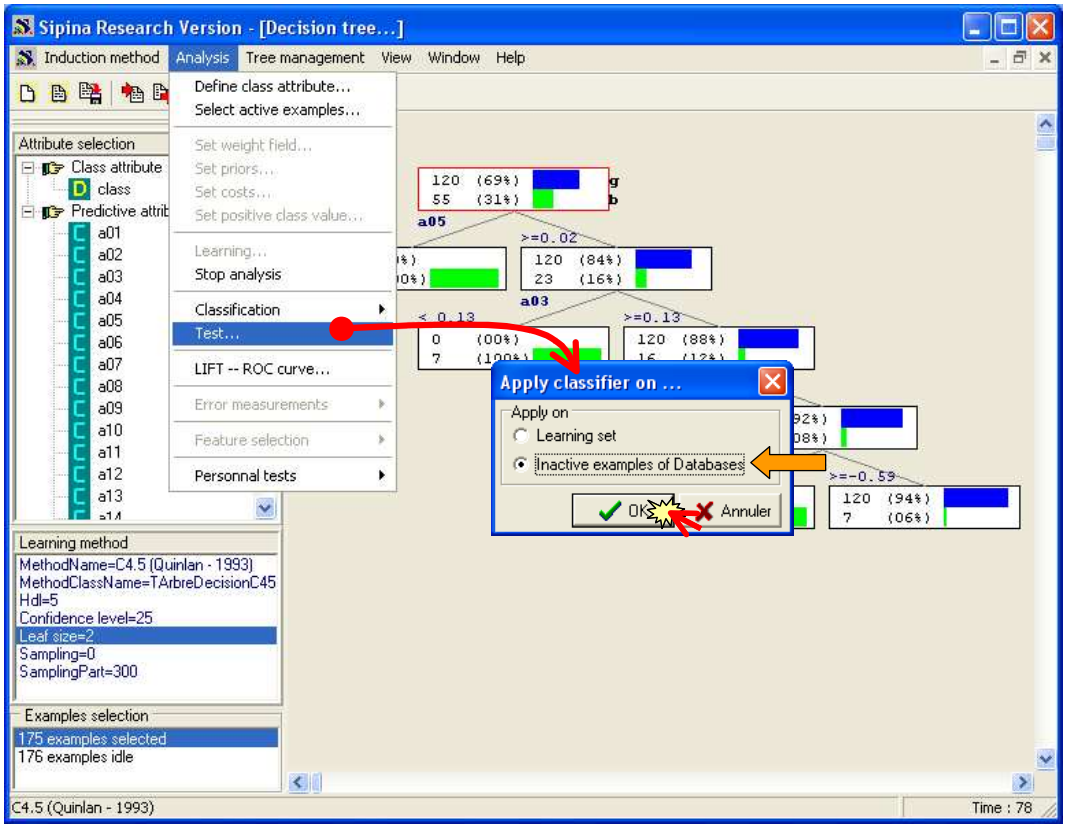
L'arbre s'affiche automatiquement. Sur chaque nœud, nous observons les distributions conditionnelles de la variable à prédire. Ayant peu de connaissances sur le phénomène étudié, nous ne nous attarderons pas sur l'interprétation des résultats.



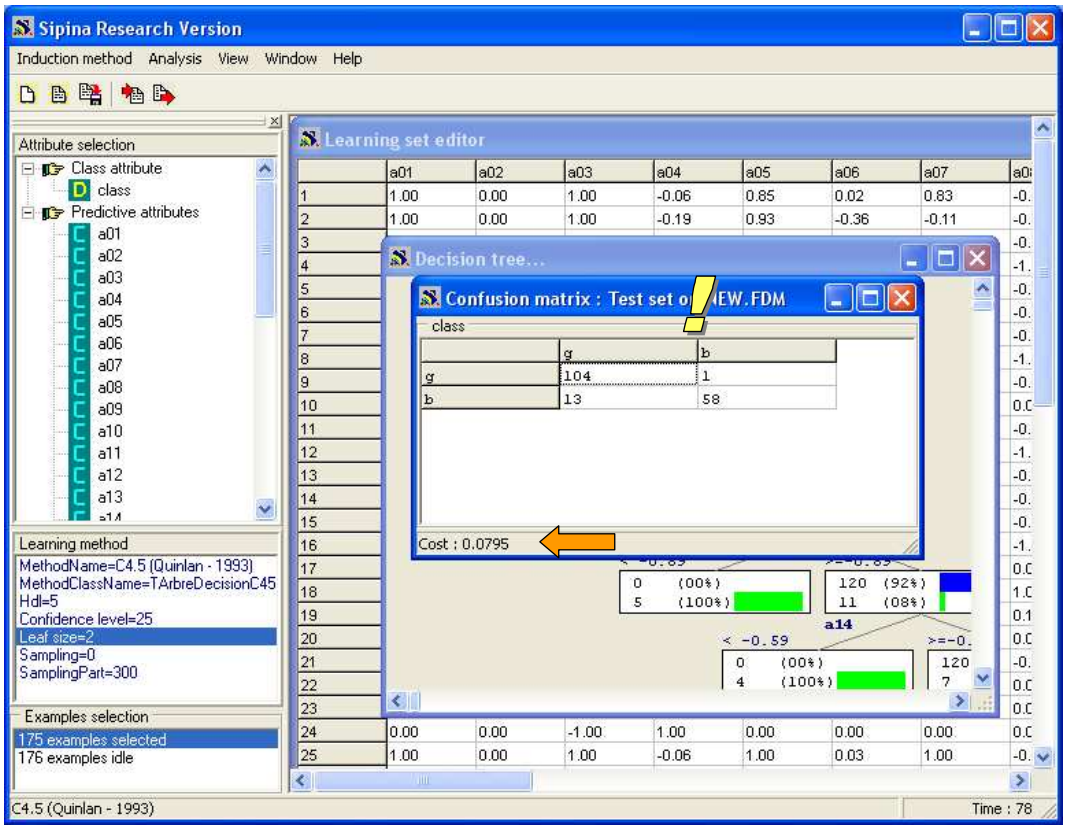
3.6 Validation de l'arbre (échantillon test)

Vient maintenant une question cruciale : quelle est la probabilité de prédire de manière erronée si nous appliquons cet arbre sur un individu pris au hasard dans la population ? Pour obtenir une estimation crédible, nous allons appliquer l'arbre sur les données préalablement mises de côté, l'échantillon test.

Dans SIPINA, il faut activer le menu ANALYSIS / TEST. Dans la boîte de dialogue qui apparaît, nous choisissons d'appliquer l'arbre sur les données inactives, les 176 individus qui n'ont pas participé à la construction de l'arbre.



Une nouvelle fenêtre est créée. Elle comporte la matrice de confusion, confrontant la vraie valeur de la variable à prédire et la prédiction de l'arbre, calculée sur l'échantillon test. Nous observons dans la barre d'état de la fenêtre le taux d'erreur (7.95%), il estime la probabilité de se tromper pour un nouvel individu à classer.



4 Conclusion

Ce didacticiel avait pour principale ambition de montrer comment charger et traiter un fichier au format ARFF (WEKA) dans SIPINA.

La facilité à importer les données est un facteur clé de succès d'un logiciel de Data Mining, il aurait été dommage de ne pas proposer une procédure simple pour gérer des fichiers en provenance de WEKA, un standard du domaine, au moins dans la communauté des chercheurs.

Pour les praticiens du Data Mining, non informaticiens, et ils sont nombreux, force est de constater que le format EXCEL reste une référence incontournable. Ne pas pouvoir importer directement un fichier XLS est un handicap majeur, c'est ce qui nous a poussé à développer une macro complémentaire qui permet de faire le lien entre le tableur et SIPINA (http://sipina.over-blog.fr/article-17592277.html).