Objectif

Montrer le fonctionnement des Listes de Décision (Decision Lists – DL) dans TANAGRA.

Les listes de décision font partie des méthodes d'apprentissage supervisé très en vogue au début des années 90. La méthode est très proche de l'induction par arbres, elle produit des règles de production triées et mutuellement exclusives du type « **Si** condition_1 **Alors** conclusion_1 **Sinon Si** condition_2 **Alors** conclusion_2 **Sinon Si**… ».

Si le biais de représentation est très proche des arbres de décision, les DL construisent des hyper-rectangles les plus homogènes possibles du point de vue de la variable à prédire, le biais de préférence est différent, elles procèdent par « Separate-and-conquer » et non par « Divide-and-conquer », elles disposent donc un degré de liberté supplémentaire pour mettre en avant les solutions les plus spécialisées. Cette liberté a un prix, cette méthode est souvent sujette au sur-apprentissage à force de vouloir détecter à tout prix les sous-populations les plus intéressantes, son paramétrage est alors primordial pour éviter qu'elle ne produise des règles couvrant trop peu d'individus et de ce fait non significatives.

La méthode que nous avons implémentée dans TANAGRA est très largement inspirée de CN2 (Clark & Niblett, ML-1989). Nous avons pris quelques libertés : (1) en simplifiant l'exploration de l'espace des hypothèses, nous utilisons un « hill-climbing » simple plutôt qu'un « best-first search », nettement plus coûteux en CPU mais pas vraiment plus efficace ; (2) en introduisant un paramètre supplémentaire, le support de la règle, pour éviter que des règles sur-spécialisées et non-significatives apparaissent.

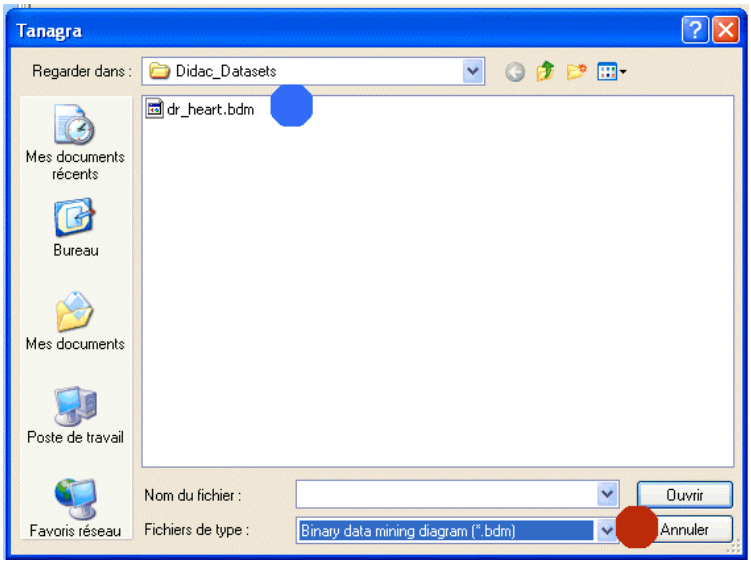
Fichier

Nous utilisons le fichier HEART déjà présenté dans d'autres didacticiels, il s'agit de prédire l'occurrence d'une maladie cardio-vasculaire à partir des caractéristiques des patients.

DECISION LIST

Charger le fichier de données

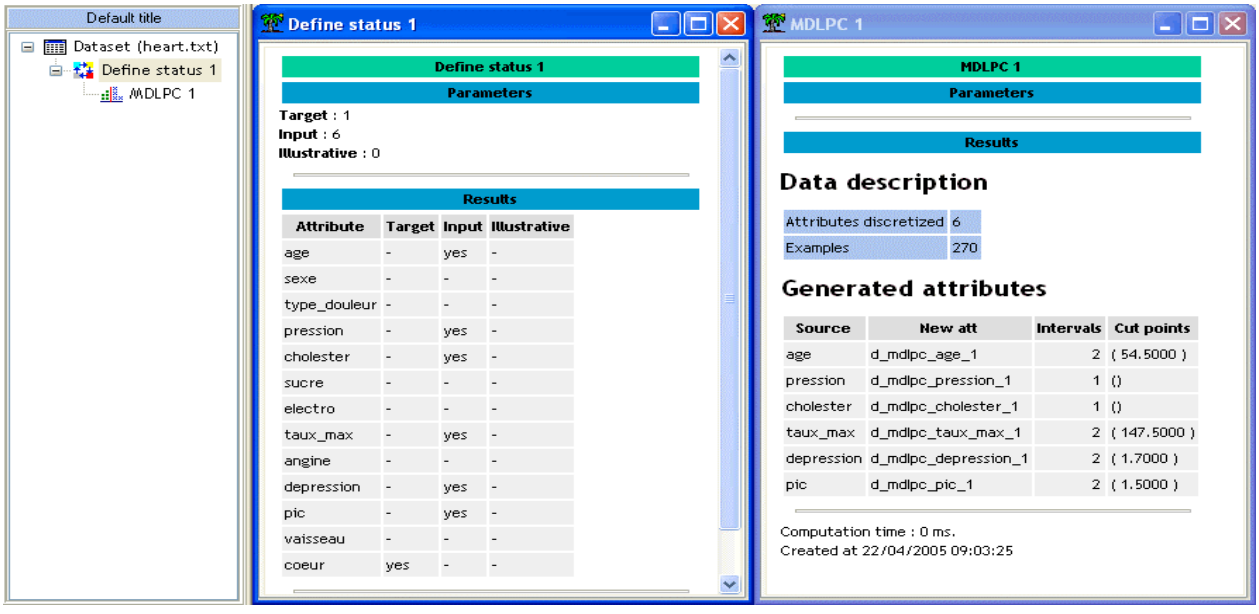
Ouvrir le fichier de données DR_HEART.BDM à partir du menu « Fichier / Ouvrir ».



Préparation des données

DL ne peut traiter que les descripteurs discrets, il est donc nécessaire de discrétiser au préalable les attributs continus, en utilisant par exemple la méthode MDLPC.

Sélectionnez en tant qu'INPUT tous les descripteurs continus et en tant que TARGET l'attribut CŒUR, placez alors le composant MDLPC (Feature Construction). Vous devez obtenir le résultat suivant.

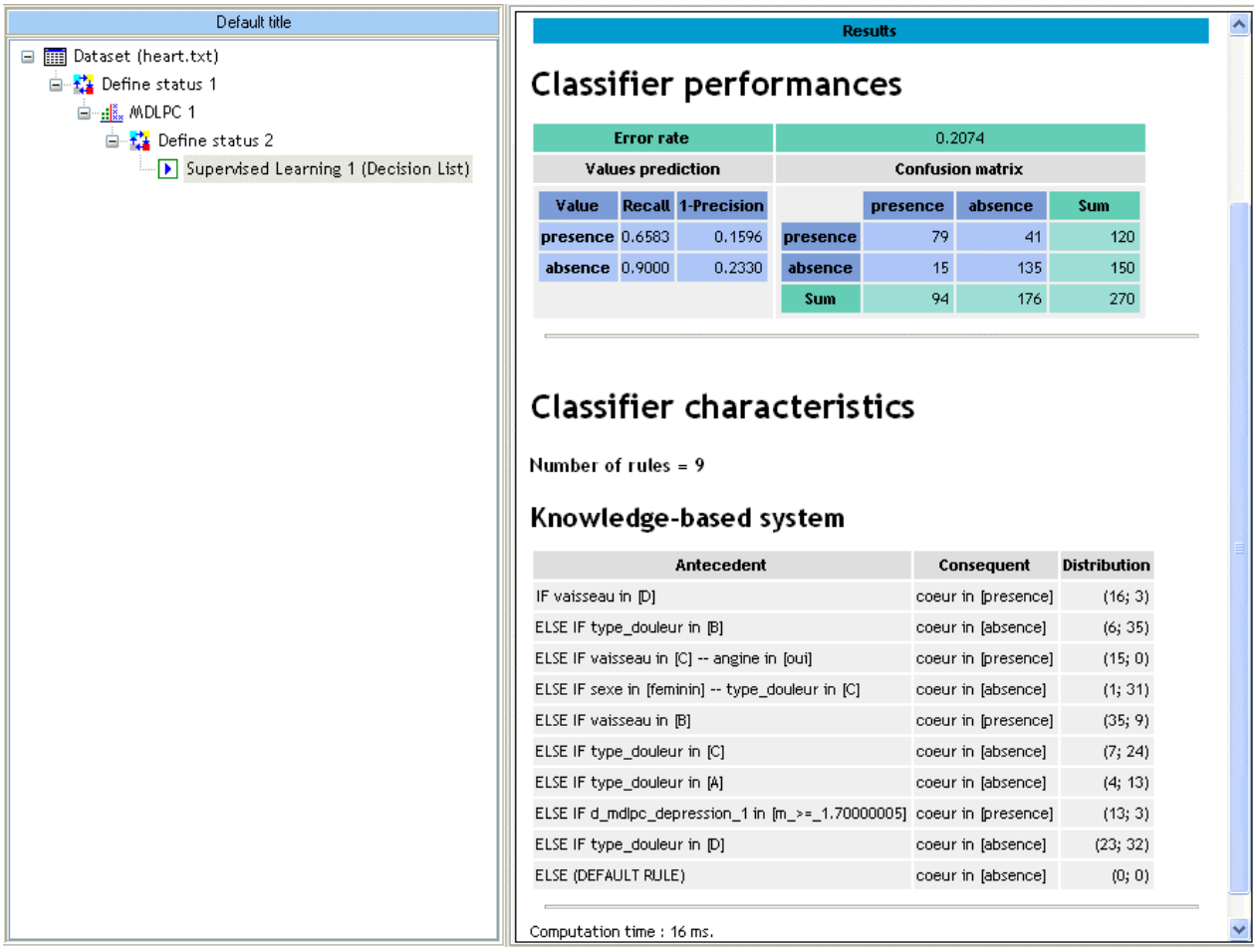


Apprentissage des DL

L'étape suivante consiste à sélectionner les variables adéquates puis placer les composants d'apprentissage.

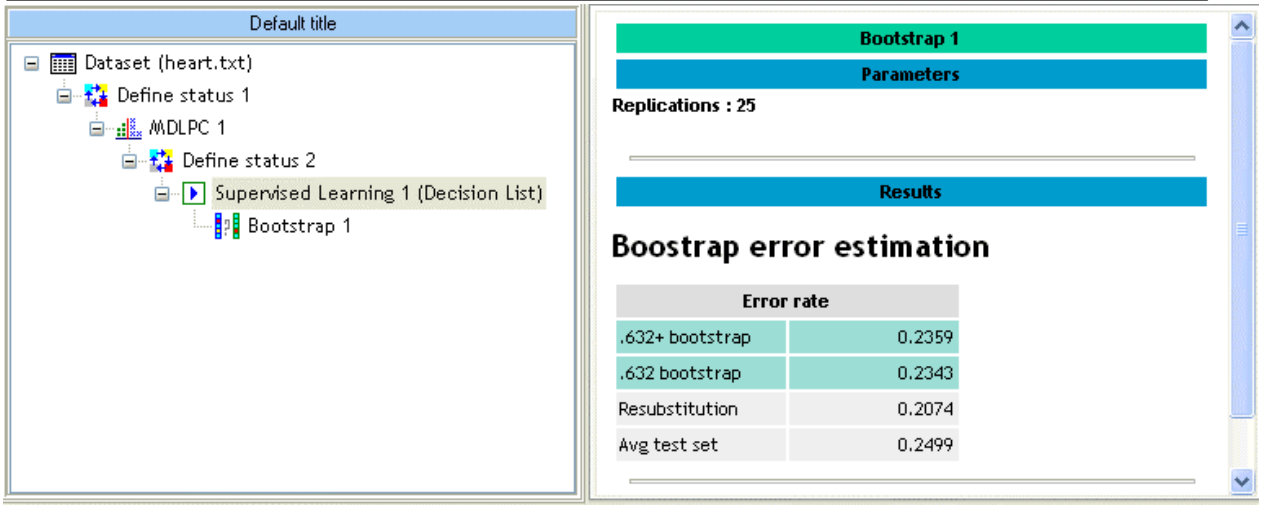
Mettez en INPUT tous les attributs discrets (y compris les variables discrétisées) à l'exception de la variable CŒUR, placez cette dernière en TARGET.

Placez alors les composants d'apprentissage, le diagramme de traitements correspondant est le suivant.



Evaluation de l'erreur

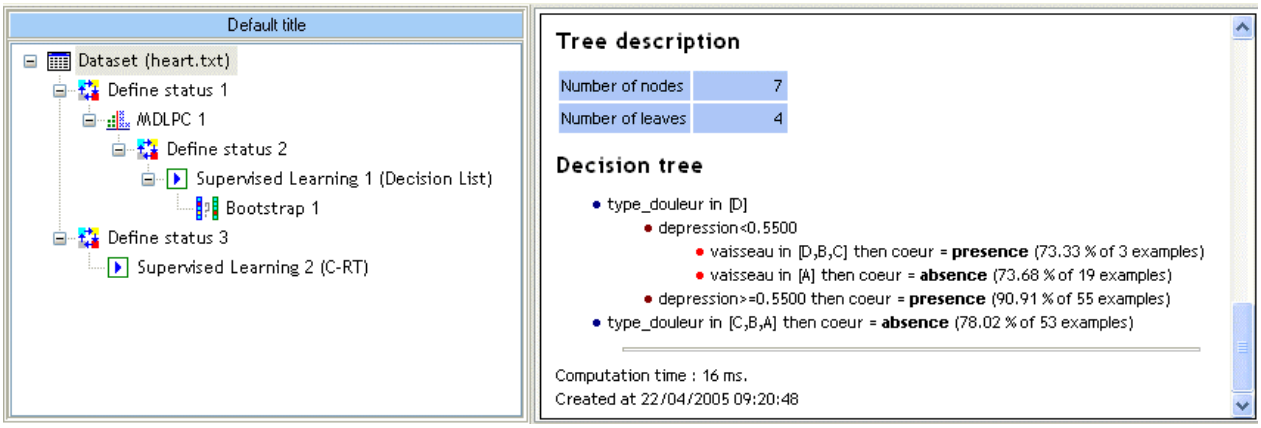
Pour avoir une idée des performances réelles de la méthode sur ce fichier, nous évaluons l'erreur à l'aide de la procédure BOOTSTRAP, les résultats montrent que le taux d'erreur en généralisation est proche de 24%.



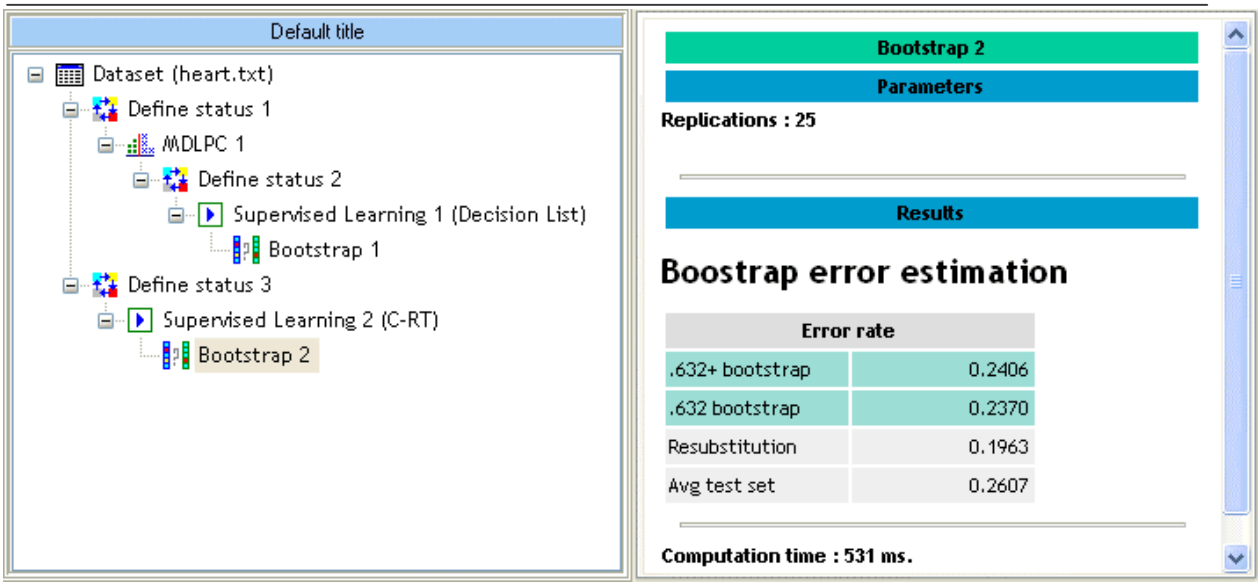
Comparaison avec C-RT

En comparaison, la méthode d'induction d'arbres de Breiman et al. (C-RT -- 1984) produit un classifieur avec un taux d'erreur estimé de 24%, notons qu'il n'est pas nécessaire dans ce cas de procéder à une discrétisation préalable des attributs continus, nous avons donc directement placé CŒUR en TARGET et tous les autres attributs en INPUT.

L'arbre de décision est le suivant.



L'estimation de l'erreur par BOOTSTRAP correspond au diagramme de traitements suivant.



La plupart des tests à grande échelle menés sur des données artificielles et réelles ont montré que les listes de décision et les arbres de décision présentaient des performances similaires.

Cette méthode est très peu connue en dehors du cercle des chercheurs en apprentissage automatique, essentiellement parce que, à ma connaissance, elle n'a jamais été implémentée dans des logiciels de grande diffusion.