UNIVERSITE DE VERSAILLES
SAINT-QUENTIN-EN-YVELINES

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

D.E.S.S. d'Ingénierie de la Statistique

Année 2001 / 2002

Option Prévision et Etudes de Marchés

Etude du logiciel SIPINA

(Version Expérimentale)

PROJET

Cindy LEVIEUX
Matthieu FIHUE

Enseignant : P.L. GONZALEZ

SOMMAIRE

INTRODUCTION

Le logiciel SIPINA a été développé par le laboratoire E.R.I.C (Equipe de Recherche en Ingénierie des Connaissances) de l'université de Lyon.

"L'Ingénierie des Connaissances" couvre des problématiques relatives à l'extraction, la modélisation et le transfert des connaissances sur ordinateurs en vue d'aboutir à des machines dites "intelligentes", c'est-à-dire des machines capables d'approcher ou de dépasser les performances humaines dans l'accomplissement de certaines tâches cognitives complexes comme reconnaître, apprendre, décider...

Le projet scientifique d'E.R.I.C. est le développement des méthodes et des outils informatiques pour l'extraction automatique des connaissances à partir des données quelles que soient leurs types : bases de données structurées, images, texte libre...

Ce logiciel permet de traiter les problèmes d'explication. Ce type de problèmes se retrouve surtout dans le domaine des sciences de l'homme (sociologie, biologie, psychologie, médecine,…). SIPINA a été utilisé, par exemple, pour l'aide au diagnostic, l'évaluation d'un risque, l'établissement de profils caractéristiques de certaines populations.

SIPINA signifie Système Interactif pour les Processus d'Interrogation Non Arborescent. C'est à la base une méthode d'analyses des données. Elle fait partie des méthodes dites explicatives comme l'analyse discriminante, la segmentation, la régression… Cette méthode permet d'expliquer une variable qualitative par des variables explicatives elles-mêmes toutes qualitatives.

Il existe en réalité 2 logiciels SIPINA :

- la Version 2.5
- la Version Expérimentale.

Nous nous intéresserons à la Version Expérimentale. Cette dernière n'est pas définitive, elle sert d'outil de recherche, elle a plus une vocation d'outil d'expérimentation que de logiciel dédié au Data Mining en entreprise. C'est la raison pour laquelle elle est distribuée gratuitement pour la recherche et l'enseignement.

1. Présentation du menu du logiciel SIPINA Version Expérimentale

Ce logiciel est téléchargeable à l'adresse suivante : ftp://rikotoma.univ-lyon2.fr/.
Le nom du fichier à charger est "Setup_DataMiningSoftaware.zip".

A l'ouverture de ce logiciel, l'interface se présente sous la forme de différentes fenêtre ayant chacune une utilité particulière :

- l'éditeur de données appelé Learning set editor contenant les données organisées sous la forme d'un tableau
- une fenêtre Analyse Report permettant de rédiger son analyse
- l'explorateur de projet contenant un module Attributes selection permettant d'afficher la variable à expliquer et ses variables explicatives, un module Learning method affichant la méthode choisie et ses paramètres et un module Examples selection affichant les individus sélectionnés pour l'apprentissage
- une barre de status indiquant la méthode en cours d'utilisation et son temps d'exécution
- une barre d'outils reprenant la plupart des options de l'item File
- un menu composé de 9 items : File, Edit, Data, Statistics, Induction Method, Analysis, View, Window, Help.

1.1. L'item File

• New

New permet de créer une nouvelle feuille de données. L'utilisateur saisit alors les données de son étude.

De plus, en se positionnant sur la feuille de données et avec le bouton droit de la souris, il est possible d'ajouter ou de supprimer une variable, un individu et de définir ses spécificités. Il doit alors donner un nom à la variable, puis choisir son type (continue ou discrète). Lorsque la variable est de type discrète, l'utilisateur doit alors entrer le nombre de modalités et leurs descriptifs (noms et codes associés).

• Open

Pour effectuer des traitements, il faut que l'utilisateur possède un jeu de données enregistré sous le format *.FDM, celui est ouvert grâce à cette commande.

• Save et Save As

Ces fonctionnalités permettent de sauvegarder les fichiers ouverts.

• Access foreign databases

- Import

Cette fonction permet d'importer des fichiers de données au format txt, Lotus, Sipina et C4.5.

- Export

cette fonction permet d'exporter des fichiers de données du Learning set editor. Il est possible d'exporter toutes les données ou simplement une partie.

• Subsample management

- Create samples

Cette fonction permet de créer de différentes manières des sous-échantillons à partir des individus dont l'utilisateur dispose.

- Save active examples

Cette fonction permet de sauver le sous-ensemble de données rendu actif par l'utilisateur.

- Save inactive examples

Cette fonction permet de sauver le sous-ensemble de données rendu inactif par l'utilisateur.

• Exit

Cette fonction permet de sortir du logiciel. Attention aux mauvaises manipulations, il n'y a pas de demande de confirmation avant la fermeture du programme !

1.2. L'item Edit

Cut, Copy, Paste et **Replace** sont les fonctionnalités classiques d'un logiciel. Elles sont utilisables sur l'éditeur de données.

1.3. L'item Data

• Add examples

Cette fonction permet d'ajouter un nombre choisi par l'utilisateur d'individus à l'emplacement désiré.

• Define labels

Cette fonction permet de donner un nom d'observation à un individu du tableau de données, ce label est affiché à gauche, dans la partie grisée du tableau.
L'utilisateur peut ainsi suivre le cheminement d'un individu dans un arbre de décision

• Add variables

Cette fonction permet d'ajouter un nombre choisi par l'utilisateur de variables à l'emplacement désiré.

• Define specifications

Une fois la variable ajoutée, l'utilisateur doit alors lui donner un nom, puis choisir son type (continue ou discrète). Lorsque la variable est de type discrète, l'utilisateur doit alors entrer le nombre de modalités et leurs descriptifs (noms et codes associés).

1.4. L'item Statistics

• Describe statistics

- univariate

Cette fonction permet de faire de la statistique descriptive univariée sur une variable préalablement choisie par l'utilisateur.

- bivariate

Cette fonction permet de faire de la statistique descriptive bivariée sur deux ou trois (à l'aide d'une variable illustrative) variables préalablement choisies par l'utilisateur. Les résultats sont présentés sous forme de graphiques

- conditionnal

Cette fonction permet d'obtenir des éléments de statistique descriptive pour une variable continue en fonction d'une variable discrète.

- contingency tables

Cette fonction permet de réaliser un tableau de contingence.

• Transform

Cette fonction permet de transformer une variable discrète en une variable continue par l'intermédiaire d'indicatrices et de transformer une variable continue en une autre variable continue (normalisation…) ou en une variable discrète par partitionnement.

• Missing data

Cette fonction permet le traitement des données manquantes : soit elles sont ignorées (suppression des individus), soit elles sont remplacées par une valeur choisie par l'utilisateur (moyenne, médiane…)

1.5. L'item Induction method

• Standard algorithm

Cette fonction permet de choisir différentes méthodes selon divers traitements statistiques comme les graphes d'induction, les règles d'induction, les réseaux de neurones, l'analyse discriminante…

• Boosting strategies

Cette fonction permet de choisir différentes méthodes parmi les méthodes d'agrégation de classifieurs.

Ces méthodes seront développées dans le paragraphe suivant.

1.6. L'item Analysis

• Define class attributes

Cette fonction permet de préciser la variable à expliquer et les variables explicatives.

• Select active examples

Cette fonction permet de sélectionner les individus pris en compte dans l'apprentissage. Il existe 4 modes de filtrage : sans filtrage, filtrage manuel, échantillonnage aléatoire et filtrage suivant des règles.

Par extension, cette option permet donc de définir les individus actifs et inactifs, repris dans certaines fonctionnalités du logiciel (ex. sauvegarde des individus, apprentissage et test, etc.)

• Set weigth field

Cette fonction est destinée à sélectionner la variable qui définirait le poids de chaque individu lors de l'apprentissage. Ce menu n'est pas actif pour le moment

• Set priors

Cette fonction est destinée à définir la distribution a priori de l'attribut à prédire lorsque l'échantillon est issu d'un tirage rétrospectif : elle permet d'opérer un redressement des probabilités d'affectation. Cette option n'est active que pour quelques méthodes (ex. Improved CHAID pour les arbres de décision, etc.)

• Set costs

Cette fonction est destinée à introduire une matrice de coût de mauvaise affectation pour évaluer le classifieur construit, pour certaines méthodes, le coût est introduit directement lors de l'apprentissage.

Par exemple, prédire le cancer alors qu'il n'y en a pas coûte en soins supplémentaires et inquiétudes inutiles, alors que prédire l'absence d'un cancer alors qu'en réalité le patient en souffre coûte le plus souvent la mort du patient qui n'est pas soigné de manière adéquate

• Learning

Cette fonction permet le lancement de l'apprentissage.

• Stop analysis

Cette fonction permet l'arrêt de l'analyse en cours.

• Classification

Cette fonction permet de reclasser les individus grâce au classifieur obtenu (arbre, réseau de neurones, etc.).

• Test

Cette fonction permet d'obtenir la matrice de confusion des individus soit en prenant l'échantillon d'apprentissage (resubstitution), soit en prenant un échantillon test.

• Error measurement

Cette fonction permet de lancer les méthodes de ré-échantillonnage d'évaluation des classifieurs. Elle permet de répondre à la question : quel va être le taux d'erreur de la méthode que l'utilisateur a appliqué sur ses données lorsqu'il inférera sur la population ? Il s'agit d'une sorte de mesure de l'erreur en test sans avoir à faire une subdivision explicite de l'ensemble des données.

• Feature selection

Cette fonction contient les méthodes de sélection de variables.
Il y en de deux types :

- les méthodes de filtres qui évacuent les attributs non pertinents avant l'apprentissage même
- les méthodes « wrapper » qui utilisent de manière explicite le classifieur pour choisir le sous-ensemble d'attributs réellement discriminants

1.7. L'item View

Cet item permet à l'utilisateur de choisir s'il souhaite ou non afficher les fenêtres **Toolbars**, **Status bar** et **Project explorer**.

1.8. L'item Window

Cascade, **Title**, **Arrange Icon** et **Close All** sont les fonctionnalités classiques d'un logiciel permettant d'avoir les données présentées selon le souhait de l'utilisateur et d'accéder aux données plus aisément. Dans cet item, l'utilisateur trouve aussi la liste des fenêtres ouvertes.

1.9. L'item Help

Cet item contient uniquement les informations concernant ce logiciel.

2. Présentation des principales méthodes proposées dans le logiciel SIPINA Version Expérimentale

L'item Induction Method offre à l'utilisateur le choix entre deux catégories de méthodes :

- les méthodes "Standard"
- les méthodes "Boosting"

Nous allons résumer le principe de ces différentes méthodes. Pour plus de renseignements, l'utilisateur pourra consulter les ouvrages cités en annexes.

2.1. Les méthodes "Standard"

2.1.1. Les graphes d'induction (Induction Graph)

Ces méthodes concernent les variables explicatives continues et/ou discrètes.

Les méthodes à base de graphes d'induction construisent une succession de partitions sur l'ensemble d'apprentissage de plus en plus fine. Le processus de construction des graphes est élémentaire. Nous partons de la partition grossière située à la racine de l'arbre. Ensuite, nous cherchons parmi les variables explicatives celle qui donne la "meilleure partition" nouvelle au sens d'un critère variant d'une méthode à une autre. Sur chaque élément de cette partition, nous répétons le processus de segmentation comme s'il s'agissait de la racine, sans se préoccuper de ce qui se passe sur les autres sommets de l'arbre.
Un sommet est dit saturé s'il n'existe aucune variable qui permet d'engendrer localement une sous partition qui puisse améliorer la valeur du critère utilisé. Le processus s'arrête si tous les sommets sont saturés. La condition de saturation peut être enrichie par d'autres paramètres comme l'introduction d'une contrainte d'admissibilité sur les sommets qui permet d'éviter l'apparition de sommets dont les effectifs seraient trop faible pour être significatifs sur le plan statistique.
Chercher la meilleure partition équivaut à rechercher la variables la plus discriminante des classes. Le pouvoir discriminatif (ou pouvoir informatif) d'une variable est généralement exprimé par une variation d'entropie lors du passage d'une partition à une autre.

• A limited search induction tree algorithm

Cette méthode consiste à construire un arbre, en spécifiant au préalable, le nombre de segmentation qui seront réalisées. A l'instar d'ID3, le choix de la variable de segmentation sur une feuille repose sur le calcul d'un gain informationnel.

• ID3 : Induction Decision Tree

Cette méthode résulte d'un algorithme ancien de construction de graphes d'induction arborescent dans lequel le critère de sélection des variables prédictives est le gain informationnel.

• GID3 : ID3 Généralisé

Cette méthode est une généralisation de la méthode précédente.

• ASSISTANT 86

Cette méthode est quasi identique à ID3, elle produit un arbre binaire. Cependant, elle se distingue par sa capacité à regrouper les feuilles produites par une segmentation et par des règles d'arrêt assez complexes, difficiles à fixer pour un problème réel.

• ChAID : Chi-square Automatic Interaction Detector

Cette méthode propose une stratégie de segmentation unique, quelle que soit la nature, quantitative ou qualitative, des prédicteurs. La particularité de ChAID réside dans le fait qu'elle effectue des regroupements entre certaines modalités du prédicteur en vue de construire des sous arbres ayant un nombre de branches optimal. Ainsi, ChAID réalise des fusions entre sommets mais contrairement à SIPINA, qui généralise ce principe à tous les sommets terminaux de la partition, ChAID ne fusionne que des sommets terminaux issus d'un même père.

• Improved ChAID

Cette méthode est une version améliorée de la méthode ChAID.

• C4.5

C4.5 est le descendant de ID3 et permet d'une part la construction d'arbre d'induction en utilisant un critère plus élaboré (le "ratio du gain") et d'autre part, d'élaguer les arbres, simplifier les règles qui en découlent, gérer les données manquantes, etc…
Pour produire des arbres de taille plus petites, la méthode C4.5 développe l'arbre au maximum et applique ensuite une procédure d'élagage qui vise à supprimer les sous arbres ne vérifiant pas une condition reposant sur le taux d'erreur.

• Improved C4.5

Cette méthode est une version améliorée de la méthode C4.5.

• SIPINA

La méthode SIPINA fournit une suite de partitions non nécessairement hiérarchisées de l'ensemble d'apprentissage : c'est une méthode conduisant à un graphe d'induction non arborescent.
L'algorithme de construction est une heuristique qui consiste à produire une succession de partitions par fusion et/ou éclatement de nœuds du graphe. L'objectif est d'optimiser la variation d'incertitude entre deux partitions (comme mesures d'incertitude, il est possible d'utiliser les mesures d'entropie), pour obtenir une partition "meilleure" que celle de l'étape précédente.

La méthode SIPINA réduit certains inconvénients des méthodes arborescentes qui considèrent certaines distributions sur les classes comme équivalentes alors qu'elles ne le sont pas (ceci étant dû à un processus exclusivement divisif et à des critères insensibles à la taille de l'échantillon).
SIPINA est une méthode qui autorise les fusions et utilise une mesure sensible aux effectifs. Cette méthode demande donc à l'utilisateur deux paramètres :

- le paramètre lambda qui contrôle le développement du graphe et pénalise les nœuds de faible effectif et de ce fait favorise les fusions entre sommets semblables.
- l'utilisateur doit également fixer le nombre minimum d'individu que doit comporter chaque sommet.

Dans tous les cas, l'arbre peut être traduit en un ensemble de règles d'induction, sans perte d'informations.

• Cost Sentive Decision Tree

Cette méthode prend en compte un matrice de coût lors de la construction de l'arbre de décision, elle l'intègre lors de la segmentation. Cette méthode est cependant moins performante que la suivante.

• Cost Sensitive C4.5

Cette méthode est une adaptation de C4.5 dans laquelle le coût serait introduit à la fois lors de la segmentation et lors de l'élagage de l'arbre. Cependant, seul l'élagage se révèle réellement efficace pour tenir compte du coût.

2.1.2. Les règles d'induction (Rule Induction)

• CN2

Cette méthode concerne uniquement les variables explicatives discrètes.

Cette méthode permet d'obtenir des règles de la forme "SI prémisse ALORS conclusion" où prémisse est une conjonction de propositions de type attribut-valeur et conclusion désigne la classe que l'on veut prédire à partir de l'ensemble d'apprentissage. Pour cela, l'algorithme recherche parmi toutes les règles la "meilleure" puis supprime les individus concernés par cette règle. Il répète ensuite œ processus jusqu'à ce qu'il n'y ait plus de "bonne" règle. L'algorithme définit la "meilleure" règle en utilisant des mesures d'entropie et des tests de signification.

• Improved CN2

Cette méthode est une version améliorée de la méthode CN2.

Cette méthode concerne les variables explicatives continues et/ou discrètes.

• C4.5 Rules

Cette méthode concerne les variables explicatives continues et/ou discrètes.

C4.5 Rules est la méthode la plus connue. Cette méthode comporte deux étapes distinctes :

- une première étape consiste à tronquer les règles obtenues en enlevant les propositions les moins intéressantes. Cette première étape permet d'écourter les règles.
- la deuxième phase vise à exprimer la base de règles de la manière la plus concise possible.

Remarquons qu'avec cette méthode, il est possible que certains individus ne soient concernés par aucune règle (ils sont alors affectés à la classe de plus gros effectif).

2.1.3. Les réseaux de neurones (Neural Network)

Ces méthodes concernent uniquement les variables explicatives continues. Il faut cependant centrer et réduire ces variables avant l'apprentissage.

• Singlelayer Perceptron

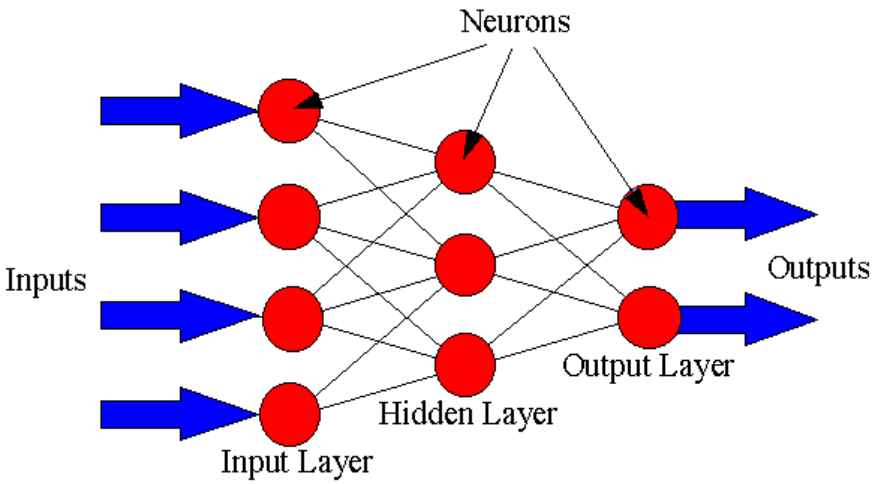
Le Perceptron Monocouche se compose d'un ou plusieurs neurones mis en parallèle. Chaque neurone, dans le Perceptron Monocouche, fournit un réseau de sortie et est habituellement relié à toutes les entrées.
Le Perceptron Monocouche possède de nombreuses limitations et ne permet pas de bonnes estimations.

• Multilayer Perceptron

Seulement, certaines variations du Perceptron Multi-couches sont disponibles.

Le Perceptron Multi-couches contient des neurones organisés en couches. La couche C_0 couche d'entrée, contient n cellules imposées de l'extérieur. La couche C_{C+1} , couche de sortie, contient p cellules. Les couches $(C_k)_{1=k=C}$ sont les couches cachées. Les seules connexions autorisées sont d'une cellule de C_k à une cellule de C_l avec k<l.

Exemple d'un Perceptron Multi-couches composé de 3 couches



• Multilayer Perceptron (with error validation control)

C'est la même méthode que le Perceptron Multi-couches, il se distingue tout simplement par un découpage apprentissage / test de l'ensemble de données. Ainsi, lors de l'apprentissage, l'utilisateur peut surveiller l'évolution des erreurs en apprentissage et en test, si le premier continue à diminuer alors que le second stagne, cela indique qu'il est en train de faire du sur-apprentissage.

2.1.4. L'analyse discriminante (Discriminant Analysis)

Ces méthodes concernent uniquement les variables explicatives continues.

L'objectif de l'analyse discriminante est de prédire une variable qualitative à k classes à l'aide de p prédicteurs quantitatifs. Nous pouvons distinguer deux aspects :

- descriptif : Quelles sont les combinaisons linéaires de variables qui permettent de séparer le mieux possible les k catégories ?
- décisionnel : Connaissant les valeurs des prédicteurs pour un nouvel individu, il s'agit de décider dans quelle catégories nous pouvons l'affecter.

Cette méthode consiste à rechercher les meilleures fonctions discriminantes.

• Linear Discriminant Analysis

Dans le cas de l'analyse discriminante linéaire, la fonction discriminante est linéaire, cela revient à réaliser une Analyse en Composantes Principales sur le nuage des centres de gravité des groupes muni de différentes métriques.

• Quadratique Discriminant Analysis

Dans le cas de l'analyse discriminante quadratique, la fonction discriminante est quadratique.

2.1.5. Liste de décision (Decision List)

Ces méthodes concernent les variables explicatives continues et/ou discrètes.

• CN2

Cette méthode consiste en la recherche en étoile de la conjonction de proposition la plus précise. Elle peut se rapprocher de la construction d'un graphe d'induction où l'utilisateur essaierait non pas de chercher la segmentation la plus efficace (divide and conquer) mais la spécialisation la plus performante (separate and conquer).

• IREP

C'est le même genre de méthode que CN2. Cependant, il y est introduit un mécanisme d'élagage sur l'échantillon test qui permet de réduire la complexité des règles.

2.1.6. Autres (Others)

Ces méthodes concernent les variables explicatives continues et/ou discrètes.

• Naive Bayes

Cette méthode permet la lecture d'un jeu de données et utilise le théorème de Bayes pour estimer les probabilités a posteriori de toutes les classes. Pour chaque donnée, la classe avec la probabilité a posteriori la plus élevée est choisie comme prédiction.

• k-NN

Dans cette méthode, il faut centrer et réduire les variables car cette méthode s’appuie sur une distance euclidienne. Il n’y a pas réellement d’apprentissage pour cette méthode, les individus qui serviront de référence pour le classement sont simplement stockés à part.

2.2. Les méthodes "Boosting"

Les méthodes **Single Learning**, **Bagging**, **Arcing** et **Boosting** permettent de construire un ensemble de classifieurs relatifs à la méthode "Standard" sélectionnée et de les agréger en classement (lorsque l'utilisateur veut les appliquer) sur un nouvel individu. Le plus souvent, l'agrégation consiste en un vote pondéré.

Ces méthodes sont utiles pour améliorer la précision des résultats obtenus par une méthode d'apprentissage. Ces méthodes montrent en général de bons résultats en améliorant les performances des classifieurs.

Ces méthodes sont de deux types : celles (**Boosting** et **Arcing**) changeant la distribution de l'ensemble d'apprentissage en se basant sur la performance des classifieurs précédents (précédent dans la technique de boostrap) et les autres

Si l'erreur est décomposée en termes de biais et de variance, **Bagging** réduit la variance des méthodes instables alors que les méthodes **Boosting** et **Arcing** réduisent à la fois le biais et la variance des méthodes instables cependant elles augmentent la variance de méthodes stables (Naïve Bayes par exemple).

Ces méthode prennent en entrée un classifieur et un échantillon d'apprentissage.

• Single learning

Un apprentissage simple signifie que l'utilisateur se contente uniquement de l'apprentissage effectuer sur l'échantillon d'apprentissage.

• Bagging : Boostrap Aggregating

Il "valide" la méthode en générant différents échantillons à partir de l'échantillon d'apprentissage. Ces échantillons sont utilisés pour construire des classifieurs. Un classifieur final est ensuite construit à partir de tous ces classifieurs. La classe prédite par ce classifieur final est celle qui apparaît le plus dans les classifieurs intermédiaires

• Boosting : Adaptive Boosting

Comme le Bagging, l'algorithme Boosting génère un ensemble de classifieurs. Mais Boosting les génère séquentiellement (alors que Bagging peut le fait en parallèle). Boosting change donc les poids de l'exemple d'apprentissage. Le but est de minimiser l'erreur attendue selon différentes distributions en entrée.
Il faut spécifier un nombre d'essais, c'est-à-dire un nombre de classifieurs à générer. Le poids de chaque classifieur pour la construction du classifieur final dépend de ses performances (le poids est inversement proportionnel à l'erreur commise).

• Arcing : Adaptively Resample and Combine

Comme Boosting, des classifieurs sont générés séquentiellement, le poids des classifieurs est proportionnel au nombre d'erreur des classifieurs précédents, à la puissance 4, plus un.

Les méthodes **Robust**, **Consus Filtering** et **Stratified K-Means Sampling Filters** permettent de filtrer les individus qui apportent peu d'informations pour l'apprentissage.

3. Présentation de quelques méthodes proposées dans le logiciel SIPINA Version Expérimentale au travers d'exemples

Nous allons travailler sur trois types d'exemples (fournis avec le logiciel) afin de pouvoir illustrer nos différentes méthodes :

- le premier contenant uniquement des variables continues,
- le deuxième contenant uniquement des variables discrètes,
- le troisième contenant les deux types de variables.

Description des données du premier exemple : (fichier iris.txt)

Cet exemple est le célèbre exemple des Iris de Fisher. Il contient des données concernant trois familles d'iris différentes :

- la famille des Setosa,
- la famille des Verisolor,
- la famille des Virginica.

Il est constitué de quatre variables continues :

- sep_length : longueur des sépales,
- sep_width : largeur des sépales,
- pet_length : longueur des pétales,
- pet_width : largeur des pétales,

et d'une variable discrète comportant trois modalités :

- type : iris-setosa, iris-versicolor et iris-virginia.

Il comporte 150 individus.
Nous chercherons à expliquer la variable "type" en fonction des variables explicatives "sep_length", "sep_width", "pet_length" et "pet_width".

Description des données du deuxième exemple : (fichier titanic.txt)

Cet exemple contient des données concernant les passagers du Titanic : survivants ou non.
Il est constitué de quatre variables discrètes :

- CLASSE : classe de voyage du passager, cette variable comporte quatre modalités : 1^{ST}, 2^{ND}, 3^{RD} et CREW.
- AGE : âge du passager, cette variable comporte deux modalités : ADULT et CHILD.
- SEXE : sexe du passager, cette variable comporte deux modalités : MALE et FEMALE.
- SURVIVANT : cette variable comporte deux modalités : YES et NO.

Il comporte 2201 individus.
Nous chercherons à expliquer la variable "SURVIVANT" en fonction des variables explicatives "CLASSE", "AGE" et "SEXE".

Description des données du troisième exemple : (fichier loan.txt)

Cet exemple contient des données concernant les clients d'un organisme de crédits.
Il est constitué de onze variables discrètes :

- Reason : moyen de découverte de l'organisme, cette variable comporte trois modalités : Other, TV et Mail.
- Marital status : statut marital, cette variable comporte six modalités : Married, Single, Partner, Separated, Divorced et Widowed.
- Title : titre, cette variable comporte trois modalités : Mrs, Mr et Miss.
- Spouse Title : titre du conjoint, cette variable comporte quatre modalités : Mr, Mrs, None et Miss.
- Guarantee : garantie, cette variable comporte deux modalités : Yes et No.
- Insurance : assurance, cette variable comporte deux modalités : Yes et No.
- Housing : logement, cette variable comporte trois modalités : Rent, Owner et Family.
- Housing type : type de logement, cette variable comporte deux modalités : Flat et House.
- Job : emploi, cette variable comporte dix modalités : Office employee, Retired, Worker, Employee, Skilled Worker, Salesman, Teacher, Executive, Unemployed et Non-working.
- Refunding : remboursement, cette variable comporte trois modalités : Slow, Fast et Normal.
- Success : succès, cette variable comporte deux modalités : N et Y.

et de dix variables continues :

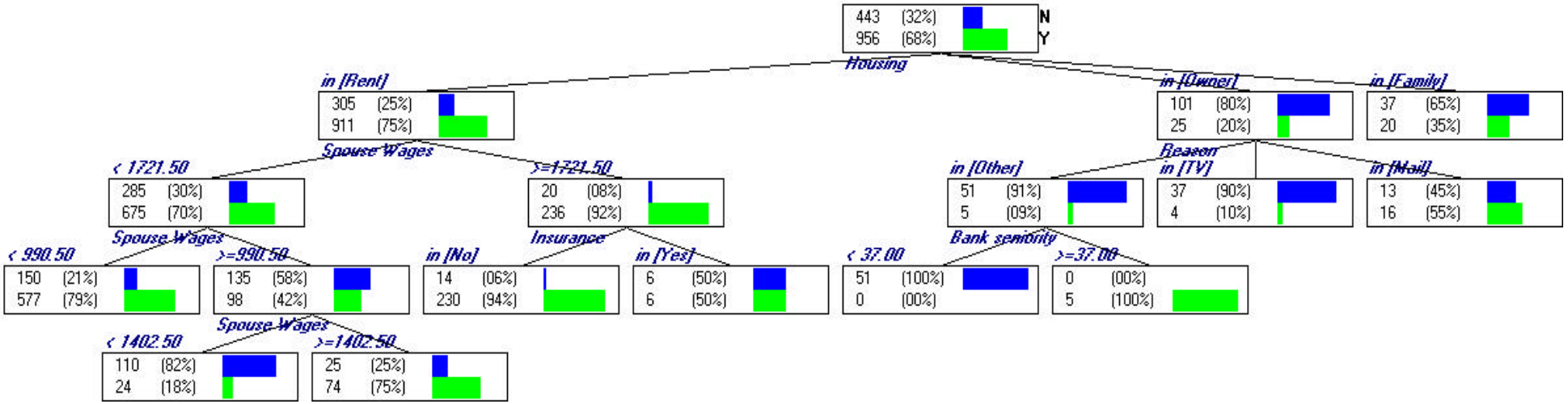
- Children : nombre d'enfants.
- Bank Seniority : ancienneté.
- Wages : salaire.
- Spouse Wages : salaire du conjoint.
- Family count : nombre de personnes dans le foyer.
- Income/head : revenu par personne.
- Rent : loyer.
- Age : âge.
- Current loans : montant des remboursements des emprunts en cours.
- Loan length : durée de l'emprunt.

Il comporte 2000 individus.
Nous chercherons à expliquer la variable "Success" en fonction des variables explicatives "Reason", "Marital status", "Title", "Spouse Title", "Guarantee", "Insurance", "Housing", "Housing Type", "Job", "Refunding", "Children", "Bank Seniority", "Wages", "Spouse Wages", "Family count", "Income/head", "Rent", "Age", "Current loans" et "Loan length".

3.1. Les graphes d'induction (Induction Graph)

Ces méthodes fonctionnant pour n'importe quel type de variables explicatives, nous utiliserons le troisième exemple.

Classifieur obtenu à l'aide de la méthode ID3



Dans notre échantillon, constitué de 70% des individus, nous avons 68% de clients qui ont obtenu un crédit. Ce graphe d'induction nous permet de remarquer que la variable la plus "discriminante" est le logement (Housing) : 75% des clients locataires de leurs logements (Housing in Rent) ont obtenu un crédit tandis que le pourcentage des clients ayant obtenu un crédit est seulement de 20% pour les propriétaires (Owner) et de 35% pour les clients logés par leur famille (Family). Nous pouvons noter l'importance du revenu du conjoint (Spouse Wages) : plus le salaire du conjoint est élevé, plus le pourcentage d'acceptation du crédit augmente. Nous pouvons également constater l'influence de la variable d'ancienneté (Bank seniority).

La matrice de confusion obtenue à l'aide de l'échantillon test, les 30% des individus restant, est la suivante :

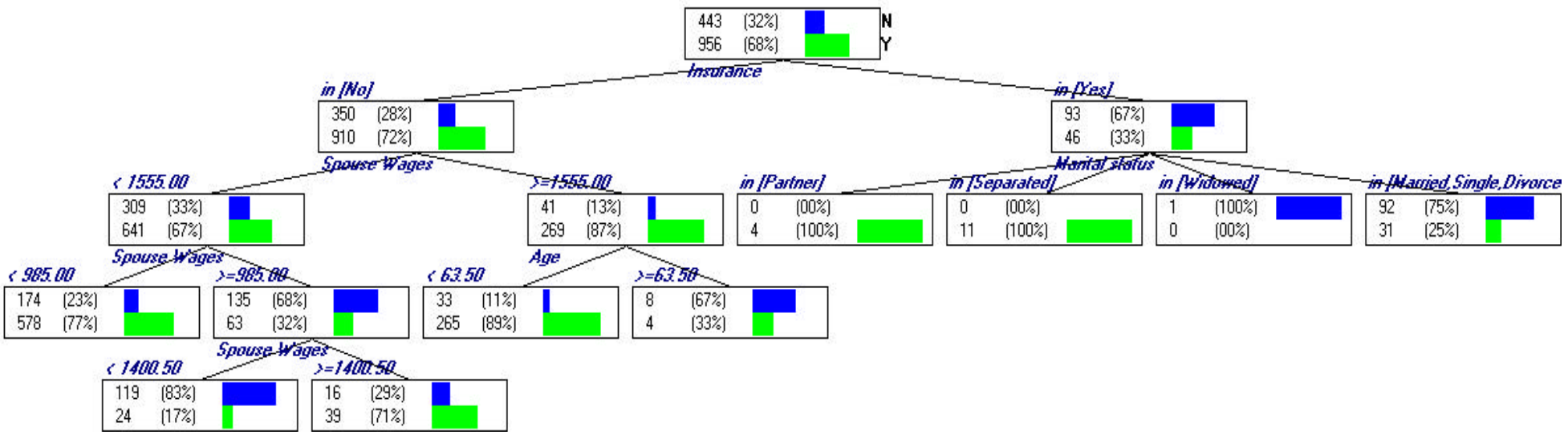
	N	Y
N	105	83
Y	23	390

Nous en déduisons donc :

- bien classés = 82.4%
- sensibilité = 55.8%
- spécificité = 94.4%.

Nous sommes donc en présence d'un bon modèle prédictif.

Classifieur obtenu à l'aide de la méthode GID3



Dans notre échantillon, constitué de 70% des individus, nous avons 68% de clients qui ont obtenu un crédit. Ce graphe d'induction nous permet de remarquer que la variable la plus "discriminante" est l'assurance (Insurance) : 72% des clients n'ayant pas souscrit d'assurance (Insurance in No) ont obtenu un crédit tandis que le pourcentage des clients ayant obtenu un crédit est seulement de 33% pour ceux en ayant souscrit une. Nous pouvons noter l'importance du revenu du conjoint (Spouse Wages) : plus le salaire du conjoint est élevé, plus le pourcentage d'acceptation du crédit augmente.

La matrice de confusion obtenue à l'aide de l'échantillon test, les 30% des individus restant, est la suivante :

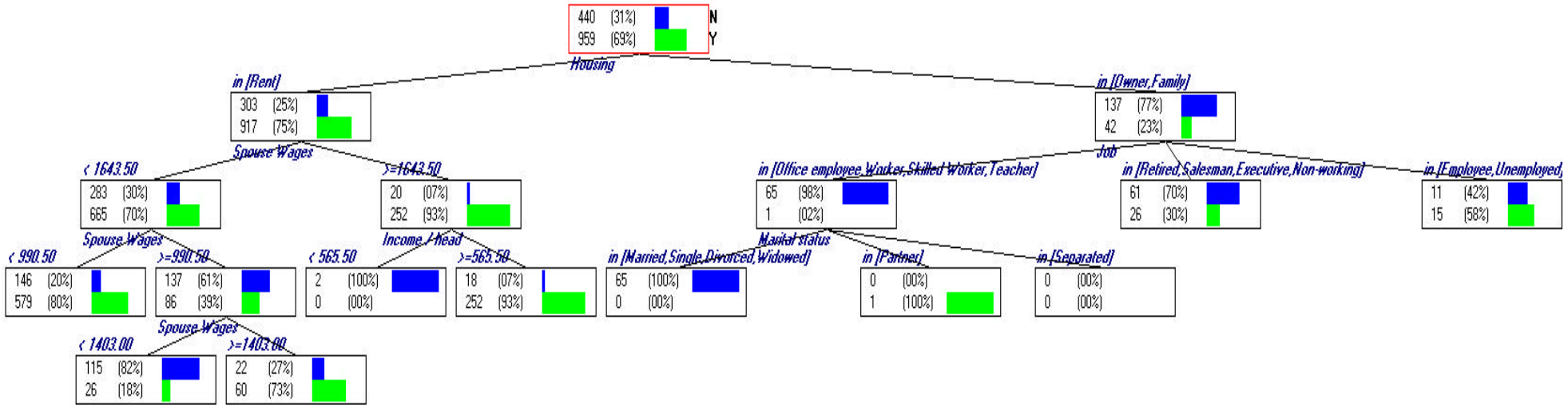
	N	Y
N	88	100
Y	34	379

Nous en déduisons donc :

- bien classés = 77.7%
- sensibilité = 46.8%
- spécificité = 91.8%.

Nous sommes donc en présence d'un assez bon modèle prédictif.

Classifieur obtenu à l'aide de la méthode ChAID



Dans notre échantillon, constitué de 70% des individus, nous avons 68% de clients qui ont obtenu un crédit. Ce graphe d'induction nous permet de remarquer que la variable la plus "discriminante" est le logement (Housing) : 75% des clients locataires de leurs logements (Housing in Rent) ont obtenu un crédit tandis que le pourcentage des clients ayant obtenu un crédit est seulement de 23% pour les propriétaires (Owner) et les clients logés par leur famille (Family). Nous pouvons noter l'importance du revenu du conjoint (Spouse Wages) : plus le salaire du conjoint est élevé, plus le pourcentage d'acceptation du crédit augmente.

La matrice de confusion obtenue à l'aide de l'échantillon test, les 30% des individus restant, est la suivante :

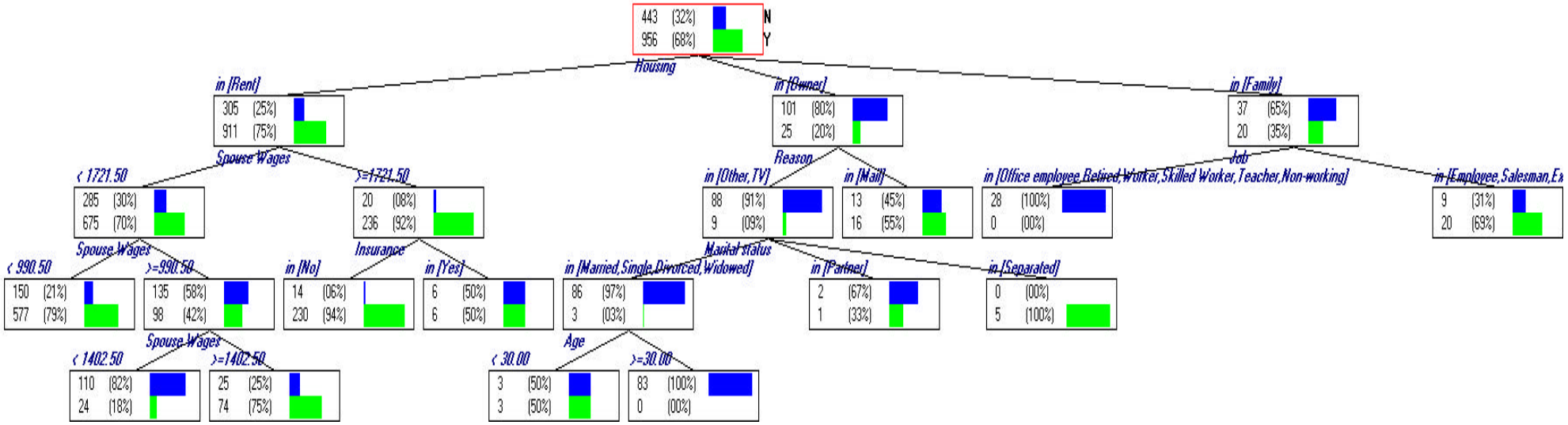
	N	Y
N	97	94
Y	25	385

Nous en déduisons donc :

- bien classés = 80.2%
- sensibilité = 50.8%
- spécificité = 93.9%.

Nous sommes donc en présence d'un bon modèle prédictif.

Classifieur obtenu à l'aide de la méthode C4.5



Dans notre échantillon, constitué de 70% des individus, nous avons 68% de clients qui ont obtenu un crédit. Ce graphe d'induction nous permet de remarquer que la variable la plus "discriminante" est le logement (Housing) : 75% des clients locataires de leurs logements (Housing in Rent) ont obtenu un crédit tandis que le pourcentage des clients ayant obtenu un crédit est seulement de 20% pour les propriétaires (Owner) et de 35% les clients logés par leur famille (Family). Nous pouvons noter l'importance du revenu du conjoint (Spouse Wages) : plus le salaire du conjoint est élevé, plus le pourcentage d'acceptation du crédit augmente.

La matrice de confusion obtenue à l'aide de l'échantillon test, les 30% des individus restant, est la suivante :

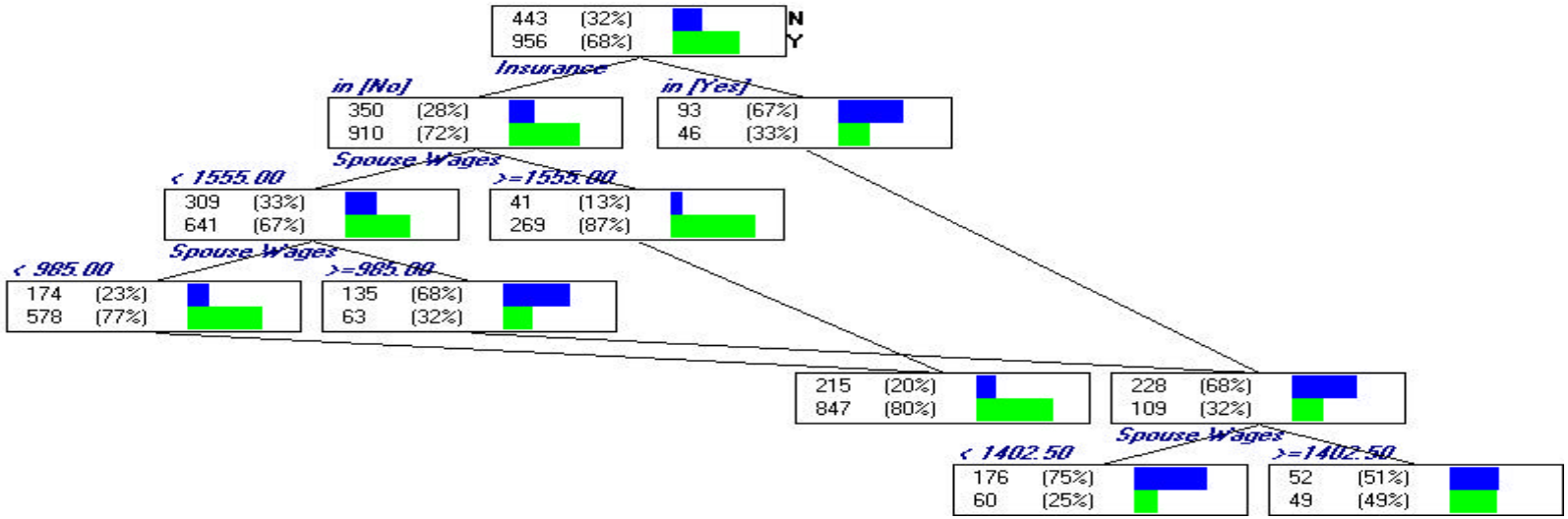
	N	Y
N	120	68
Y	67	346

Nous en déduisons donc :

- bien classés = 77.5%
- sensibilité = 63.8%
- spécificité = 83.8%.

Nous sommes donc en présence d'un assez bon modèle prédictif

Classifieur obtenu à l'aide de la méthode SIPINA



Dans notre échantillon, constitué de 70% des individus, nous avons 68% de clients qui ont obtenu un crédit. Ce graphe d'induction nous permet de remarquer que la variable la plus "discriminante" est l'assurance (Insurance) : 72% des clients n'ayant pas souscrit d'assurance (Insurance in No) ont obtenu un crédit tandis que le pourcentage des clients ayant obtenu un crédit est seulement de 33% pour ceux en ayant souscrit une. Nous pouvons noter l'importance du revenu du conjoint (Spouse Wages) : plus le salaire du conjoint est élevé, plus le pourcentage d'acceptation du crédit augmente.

La matrice de confusion obtenue à l'aide de l'échantillon test, les 30% des individus restant, est la suivante :

	N	Y
N	94	94
Y	41	372

Nous en déduisons donc :

- bien classés = 77.5%
- sensibilité = 50%
- spécificité = 90.1%.

Nous sommes donc en présence d'un assez bon modèle prédictif.

3.2. Les règles d'induction (Rule Induction)

Ces méthodes fonctionnant pour CN2 sur des variables discrètes et pour C4.5 Rules sur n'importe quel type de variables explicatives, nous utiliserons le deuxième exemple.

• CN2

Les règles obtenues sont :

N°		Prémisse		Conclusion	Support	Confiance	J-Mesure	Implication
1	if	SEXE in [FEMALE]	then	SURVIVANT in [YES]	470	0.732	0.108	1.000
2	if	SEXE in [MALE]	then	SURVIVANT in [NO]	1731	0.788	0.034	1.000

Default rules : YES = 367 et NO =126, c'est-à-dire que 367 hommes et 126 femmes sont mal classés.

Cette méthode donne deux règles assez simples : les femmes survivent et les hommes meurent (cette règle est vérifiée pour 73% des femmes et 78% des hommes).

• C4.5 Rules

Les règles obtenues sont :

N°		Prémisse		Conclusion	Support	Confiance	J-Mesure	Implication
1	if	SEXE in [FEMALE] and CLASSE in [1ST]	then	SURVIVANT in [YES]	145	0.972	0.093	1.000
2	if	SEXE in [FEMALE] and CLASSE in [2ND]	then	SURVIVANT in [YES]	106	0.877	0.046	1.000
3	if	SEXE in [FEMALE] and CLASSE in [CREW]	then	SURVIVANT in [YES]	23	0.889	0.012	1.000
4	if	SEXE in [MALE] and AGE in [CHILD] and CLASSE in [1ST]	then	SURVIVANT in [YES]	5	1.000	0.004	0.966
5	if	SEXE in [MALE] and AGE in [CHILD] and CLASSE in [2ND]	then	SURVIVANT in [YES]	11	1.000	0.008	0.999
6	if	CLASSE in [3RD]	then	SURVIVANT in [NO]	706	0.541	0.005	0.000
7	if	SEXE in [MALE] and AGE in [ADULT]	then	SURVIVANT in [NO]	1667	0.797	0.039	1.000

Default rules : YES = 0 et NO =0.

La variable SEXE est une variable très discriminante : les femmes ont beaucoup plus de chance de survie que les hommes. Notons également, l'importance de la variable CLASSE : les première et deuxième classes sont moins "dangereuses" que la troisième classe. Enfin, la variable AGE nous montre que les enfants ont plus de chance de survie que les adultes.

3.3. Les réseaux de neurones (Neural Network)

Ces méthodes fonctionnant uniquement pour des variables explicatives continues, nous utiliserons le premier exemple.

• Single-layer Perceptron

Cette fonction nous présente l'architecture du réseau de neurones utilisé, un graphique sur l'évolution du taux d'erreur et la matrice de confusion obtenue à l'aide d'un échantillon, constitué de 70% de l'échantillon initial, et pour 5000 itérations.

	Iris-setosa	Iris-versicolor	Iris-virginica
Iris-setosa	34	1	0
Iris-versicolor	8	12	10
Iris-virginica	0	0	39

L'échantillon test, constitué des 30% restant, nous permet d'obtenir la matrice de confusion suivante :

	Iris-setosa	Iris-versicolor	Iris-virginica
Iris-setosa	15	0	0
Iris-versicolor	5	12	3
Iris-virginica	0	0	11

Nous en déduisons donc que le pourcentage de bien classés vaut : 82.6%.

• Multi-layer Perceptron

Nous avons choisi d'utiliser un réseau de neurones à une couche cachée.
Cette fonction nous présente l'architecture du réseau de neurones utilisé, un graphique sur l'évolution du taux d'erreur et la matrice de confusion obtenue à l'aide d'un échantillon, constitué de 70% de l'échantillon initial, et pour 218 itérations.

	Iris-setosa	Iris-versicolor	Iris-virginica
Iris-setosa	35	0	0
Iris-versicolor	0	27	3
Iris-virginica	0	0	39

L'échantillon test, constitué des 30% restant, nous permet d'obtenir la matrice de confusion suivante :

	Iris-setosa	Iris-versicolor	Iris-virginica
Iris-setosa	15	0	0
Iris-versicolor	0	18	2
Iris-virginica	0	0	11

Nous en déduisons donc que le pourcentage de bien classés vaut : 95.6%.

3.4. L'analyse discriminante (Discriminant Analysis)

Ces méthodes fonctionnant uniquement pour des variables explicatives continues, nous utiliserons le premier exemple.

• Linear Discriminant Analysis

Cette fonction nous présente la matrice de corrélation et la fonction de score obtenue à l'aide d'un échantillon, constitué de 70% de l'échantillon initial.

	Iris-setosa	Iris-versicolor	Iris-virginica
sep_length	23.0533	15.4510	12.5476
sep_width	21.8068	6.3383	3.1091
pet_length	-14.9924	5.1909	11.6893
pet_width	-19.3914	4.9677	19.7327
Constant	-81.4541	-69.3497	-98.7441

L'échantillon test, constitué des 30% restant, nous permet d'obtenir la matrice de confusion suivante :

	Iris-setosa	Iris-versicolor	Iris-virginica
Iris-setosa	15	0	0
Iris-versicolor	0	19	1
Iris-virginica	0	0	11

Nous en déduisons donc que le pourcentage de bien classés vaut : 97.8%.

CONCLUSION

Nous avons choisi de travailler sur la version recherche de SIPINA car, même si elle est encore en cours de développement, elle permet d'appliquer un grand nombre de méthodes de segmentation et/ou d'apprentissage et cela, sous des formes très variées (arbres de décision, règles de décision, réseaux de neurones, analyse discriminante...).

Après avoir présenté les différentes commandes afin de découvrir les larges possibilités du logiciel, puis expliqué quelques-unes une des méthodes d'apprentissages programmées sous SIPINA Version Expérimentale, nous avons pu appliquer certaines de ces méthodes sur des jeux de données fournies avec le logiciel.

Nous n'avons cependant pas comparé les méthodes entre elles car l'expérience montre qu'il n'en existe pas de meilleure dans l'absolu : les performances d'un algorithme dépendent pour beaucoup du jeu de données en présence.

L'utilisateur, en possession d'un jeu de donnée, devra donc adapter la méthode à ses données. Il devra en essayer plusieurs. Il devra également choisir une méthode adapter à ses besoins et à l'utilisation qu'il souhaite faire des résultats obtenus : par exemple une méthode basée sur un réseau de neurones donne d'excellent résultats en termes d'apprentissage mais ne donnent que peu d'explications sur la manière d'obtenir les résultats. A l'inverse, un graphe d'induction jouit d'une excellente lisibilité (si l'arbre ne possède pas un nombre trop important de nœuds !) et d'un très bonne compréhension même pour un utilisateur peu averti.

Dans tous les cas, quelque soit la méthode utilisée, il est nécessaire de valider les résultats obtenus (même si la taille de l'échantillon d'apprentissage est grande) à laide d'une validation croisée ou d'un échantillon test.

ANNEXES

Les graphes d'induction (Induction Graph)

La plupart de ces méthodes sont décrites dans le livre suivant :
"Graphes d'induction - Apprentissage et Data Mining", D.A. Zighed and R. Rakotomalala, Hermes, 2000

A limited search induction tree algorithm	J. CATLETT, "Megainduction : machine learning on very large databases", PhD thesis, University of Sidney, 1991.
ID3-IV	J.R. QUINLAN, "Induction of decision trees", Machine Learning, 1:81-106, 1986.
GID3	J. CHENG, U. FAYYAD, K. IRANI, Z. QIAN, "Improved decision trees : a generalized version of ID3", in Proceedings of the 5th ICML, pp.100-108, 1988.
ASSISTANT 86	B. CESTNIK, I. KONONENKO, I. BRATKO, "Assitant-96 : a knowledge elicitation tool for sophisticated users", in I. Bratko and N. Lavrac editors, Progress in Machine Learning, 1987.
CHAID	G.V. KASS, "An exploratory technique for investigating large quantities of categorical data", Applied Statistics, 29(2):119-127, 1980.
C4.5	J.R. QUINLAN, "C4.5 : Programs for Machine Learning", Morgan Kaufmann, 1993.
Improved C4.5	R. RAKOTOMALALA, S. LALLICH, "Handling noise with generalized entropy of type beta in induction graphs algorithm", in Proceedings of International Conference on Computer Science and Informatics, pp. 25-27, 1998.
SIPINA	D. ZIGHED, "Sipina : Méthode et logiciel", Lacassagne, 1992.
Improved CHAID (Tschuprow goodness of split)	R. RAKOTOMALALA, D. ZIGHED, "Mesures d'association dans les graphes d'induction : une approche statistique de l'arbitrage généralité-précision", in Proceedings of AIDRI'97, pp.131-134, 1997.
Cost Sensitive Decision Tree	J-H. CHAUCHAT, R. RAKOTOMALALA, M. CARLOZ, C. PELLETIER, "Targeting Customer Groups using Gain and Cost Matrix : a Marketing Application", in Data Mining for Marketing Applications (Working Notes), PKDD'2001, pp. 1-13, September 2001.
Cost Sensitive C4.5	J-H. CHAUCHAT, R. RAKOTOMALALA, M. CARLOZ, C. PELLETIER, "Targeting Customer Groups using Gain and Cost Matrix : a Marketing Application", in Data Mining for Marketing Applications (Working Notes), PKDD'2001, pp. 1-13, September 2001.

Les règles d'induction (Rule Induction)

CN2	P. CLARK, R. BOSWELL, "Rule induction with CN2 : some recents improvements", in Proceedings of ECML'91, pp.151-163, 1991.
Improved CN2	R. RAKOTOMALALA, D. ZIGHED, F. FESCHET, "Empirical evaluation of rule characterization in rule induction process", in Proceedings of the Fourteenth European Meeting on Cybernetics and Systems Research, pp.779-804, 1998.
C4.5 RULES	J.R. QUINLAN, "C4.5 : Programs for Machine Learning", Morgan Kaufmann, 1993.

Les réseaux de neurones (Neural Network)

Singlelayer Perceptron	K. MEHROTRA, C.K. MOHAN, S. RANKA, "Elements of Artificial Neural Networks", The MIT Press, 1997.
Multilayer Perceptron	K. MEHROTRA, C.K. MOHAN, S. RANKA, "Elements of Artificial Neural Networks", The MIT Press, 1997.
Multilayer Perceptron (with error validation control)	K. MEHROTRA, C.K. MOHAN, S. RANKA, "Elements of Artificial Neural Networks", The MIT Press, 1997.

L'analyse discriminante (Discriminant Analysis)

Linear Discriminant Analysis	G. SAPORTA, "Probabilités, Analyse des Données et Statistique", Technip, 1990.
Quadratic Discriminant Analysis	G. SAPORTA, "Probabilités, Analyse des Données et Statistique", Technip, 1990.

Liste de décision (Decision List)

CN2	P. CLARK, T. NIBLETT, "The CN2 induction algorithm", Machine Learning, 3(4):261-283, 1989.
IREP	J. FURNKRANTZ, G. WIDMER, "Incremental reduced error pruning", in Proceedings of ICML'95, pp.244-251, 1994.

Autres (Others)

Naive Bayes	T.M. MITCHELL, "Machine Learning", McGraw-Hill International Editions, 1997.
k-NN	T.M. MITCHELL, "Machine Learning", McGraw-Hill International Editions, 1997.