Modélisation Informatique de Clients Douteux, En utilisant les Techniques de DATAMINING

Mostafa Hanoune, Fouzia Benabbou

MODELISATION INFORMATIQUE DE CLIENTS DOUTEUX, EN UTILISANT LES TECHNIQUES DE DATAMINING

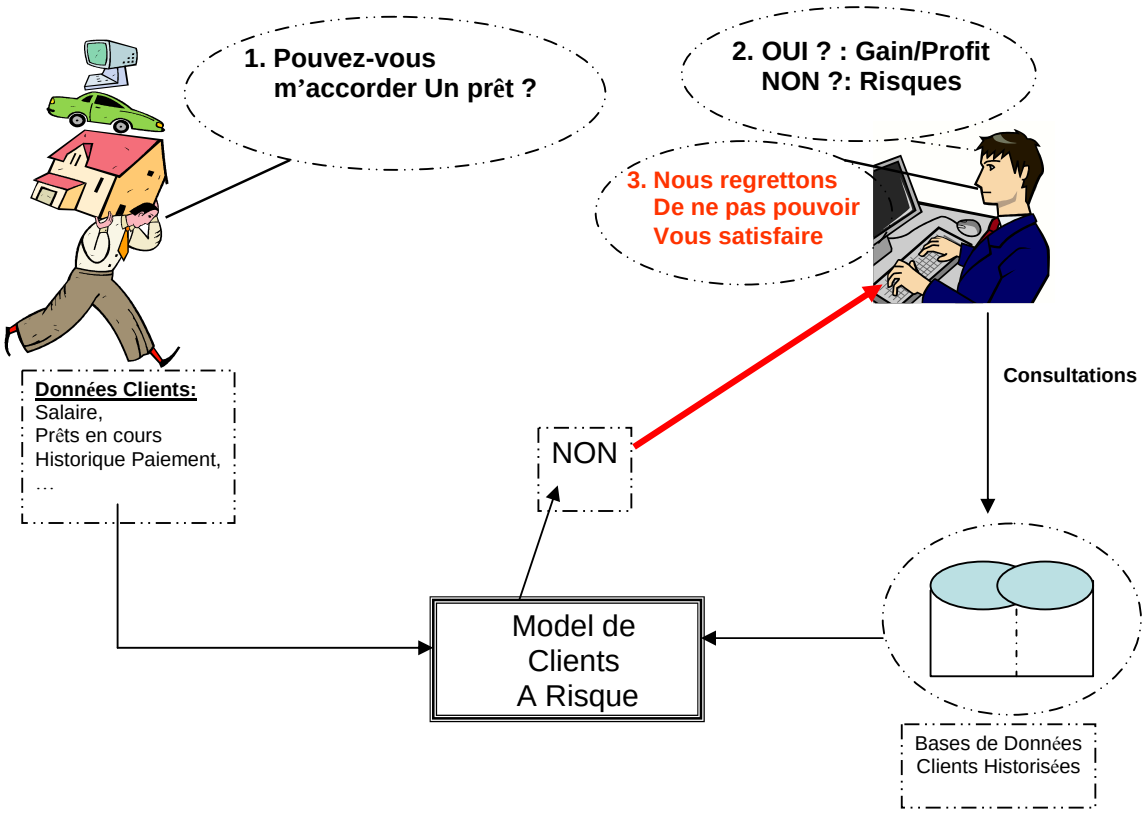
Mostafa. Hanoune Faculté des sciences Ben M'Sik, m_hanonune@yahoo.fr
Fouzia. Benabbou Faculté des sciences Ben M'Sik, hgfbenabbou@menara.ma

MOTS CLES : *Knowedge Discovery in Databases (KDD),*Datamining, Extraction de la connaissance, Fouille de données, Intelligence Artificielle, Algorithmes, Arbres de decision, Théorie de décision, …

RESUME

Le but de ce travail, est la conception et réalisation d'un logiciel permettant la modélisation de clients douteux, par l'extraction de connaissances à partir de bases de donnes. Une telle connaissance pourrait être utilisée pour permettre aux décideurs et responsables stratégiques de prendre des décisions dans des situations bien précises ;

A-PROBLEMATIQUE



La problématique est la suivante :
Lorsqu'un client se présente, à une société de crédit, pour avoir un prêt, la société de crédit est devant un embarras, surtout pour les clients qu'elle ne connaît pas encore.
Va-t-elle accepter cette demande de prêt, ce qui est légitime pour toute banque, en vue d'accroître le profit ?
Va-t-elle refuser cette demande pour ne pas risquer de tomber sur de mauvais payeur, dans telle cas, se sera une perte sèche pour la banque ?
Pour répondre à ces questions, on propose l'utilisation des techniques de DATAMINING, pour essayer de trouver les informations cachées dans la base de données à l'aide d'algorithmes avancés.

Pour extraire de la connaissance à partir de la base de données historisées de la banque.
Quelle sera alors la méthode utilisée ?

B-METHODOLOGIE

Par ce travail nous avons créé un programme permettant d'aider le banquier ou le décideur de prêt, a prendre une décision en confrontons les données personnelles du demandeur a l'arbre déjà crée a partir de la base de données des clients.
La méthode théorique utilisée est: **LA CLASSIFICATION** : qui consiste à examiner des caractéristiques d'un élément nouvellement présenté (Les informations relatives au demandeur de prêt : Age, Salaire, Situation Familiale, …) afin de l'affecter à une classe d'un ensemble prédéfini. A partir de la base de données des clients.

1. Les arbres de décision :

Un arbre de décision est une structure qui permet de déduire un résultat à partir de décisions Successives, Pour parcourir un arbre de décision et trouver une solution il faut partir de la racine.

Chaque nœud est ou bien une feuille dénotant une décision ou bien une branche spécifiant un test sur une valeur d'un attribut. Le nombre de descendants de chaque nœud dépend des résultats du test effectué à ce niveau.

Généralement un nœud pose une question sur un attribut de la base de données, la valeur de cet attribut permet de savoir sur quel fils descendre. Pour les attributs énumérées il est parfois possible d'avoir un fils par valeur, on peut aussi décider que plusieurs valeurs différentes mènent au même sous arbre. Pour les attributs continus il n'est pas imaginable de créer un nœud qui aurai potentiellement un nombre de fils infini, on doit discrétiser le domaine continu (arrondis, approximation), donc décider de segmenter le domaine en sous ensembles.

Plus l'arbre est simple, et plus il semble techniquement rapide à utiliser. En fait, il est plus intéressant d'obtenir un arbre qui est adapté aux probabilités des variables à tester. La plupart du temps un arbre équilibre sera un bon résultat. Si un sous arbre ne peut mener qu'à une solution unique, alors tout ce sous arbre peut être réduit à sa simple conclusion, cela simplifie le traitement et ne change rien au résultat final. Ross Quinlan a travaillé sur ce genre d'arbres de décision dans le but de créer un joueur d'échecs virtuel efficace. Il a mis en évidence le fait qu'un arbre peut être construit à la volée peu à peu, c'est le principe "Top-Down Inducton of Decision Tree". Un problème ainsi insolvable par la trop grande quantité d'informations à traiter, est localement simplifie. On se sert des probabilités pour découvrir le prochain test à réaliser. Comme dans le cas d'un algorithme glouton on cherche simplement à effectuer le test le plus efficace sans regarder trop en avance (ce qui dans le cas du jeu d'échecs deviendra trop long).

2. ID3

J. Ross Quinlan de l'université de Sydney, est le créateur de l'algorithme **ID3**, ce fut en 1975 .

Principe du ID3 :

Construire un arbre de décision d'un ensemble fixe d'exemples. L'arbre résultant est employé pour classifier de futurs échantillons.

L'exemple a plusieurs attributs et appartient à une classe (comme oui ou non). Les nœuds de feuille de l'arbre de décision contiennent le nom de classe contrairement aux nœuds de décision. Le nœud de décision est un essai d'attribut avec chaque branchement (à un autre arbre de décision) étant une valeur possible de l'attribut. Gain de l'information des utilisations ID3 pour l'aider à décider quel attribut entre dans un nœud de décision. L'avantage d'apprendre un arbre de décision est qu'un programme, plutôt qu'un ingénieur de la connaissance, obtient la connaissance d'un expert.

Contraintes du ID3 :

Les exemples utilisé par ID3 doivent respecter certaines contraintes, tel que :

- Etre fixes.
- Les mêmes attributs doivent décrire chaque variable, et avoir un nombre fixe de valeurs.
- Un exemple d'attributs doit être déjà définit.

• Des classes bien délimitées.
• Un nombre suffisant d'exemples.

ID3 met en jeux deux concepts majeurs :

• L'entropie : concept permettant de trouver les paramètres les plus significatifs.
• Les arbres de décision : efficace et intuitive permettent d'organiser les descripteurs qui peuvent être utilisés comme fonction prédictive.

Entropie

Le concept d'entropie est utilisé pour ordonnée la liste des descripteurs avec le respect des ensembles de données, et des classes. L'entropie fournie une définition de descripteurs les plus significatifs, et c'est l'un des concepts majeurs de la méthode ID3.

Les objets, dans l'ensemble de données avec lesquels la méthode ID3 travail, sont caractérisés par le classificateur. L'ensemble le plus structuré (c'est-à-dire non aléatoire) est celui dans lequel tous les objets ont la même valeur pour le classificateur : Cet ensemble a une entropie de zéro. La situation la plus aléatoire consiste en ce quand il y a une distribution égale de toutes les valeurs de classificateur différentes : Cet ensemble a une entropie d'un. Si l'ensemble a un nombre improportionnel d'une valeur par opposition aux autres, donc il a une entropie entre le zéro et un.

Le calcul de l'entropie est donné par la formule mathématique suivante :

$$I(p) = -\sum_{i=1}^{N} p(c_i)\log_2 p(c_i)$$

Où **p(c*i*)** est la probabilité que la classe c***i*** soit correcte. La quantité **log2p(c*i*)** est la quantité d'information que l'on donne quand la classe est identifiée comme **c*i***, la somme pondérée sur toutes les classes est la valeur attendue de ce contenu de l'information. Cette valeur attendue est la mesure d'entropie de l'ensemble.

Ainsi, si l'on calcule l'entropie de P= (0.5, 0.5) alors I (P)=1, si P= (0,1) alors l'entropie I (P) = 0. Donc si S est une collection de 14 exemples avec 9 OUI et 5 NON alors :

$$I(S) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

On note |X| le cardinal de l'ensemble X. Si un ensemble T d'enregistrements partitionné par la valeur de l'attribut catégorique en classes disjointes {C1,C2,…..,Ck}, alors l'information nécessaire pour reconnaître la classe d'un élément de T est associée à l'entropie I(P), où P est la distribution de probabilité de la partition (C1,C2,…..,Ck) :

$$P = \left(\frac{|C_1|}{|T|}, \frac{|C_2|}{|T|}, ..., \frac{|C_k|}{|T|}\right)$$

Si l'on partition notre ensemble T en valeurs disjointes, alors l'information nécessaire pour quantifier la classe d'un élément de T, alors l'information nécessaire pour identifier la classe d'un élément de T devient :

$$Info(X,T) = \sum_{i=0}^{m} \frac{|T_i|}{|T|} \times Info(T_i)$$

Où T*i* est un sous ensemble de A pour la valeur ***i***, et ∑ est pour chaque valeur de ***i*** toute les valeurs possibles de l'attribut A.

De là nous pouvons définir ce qu'est le gain, qui représente le gain obtenu dû au partitionnement par l'attribut X :

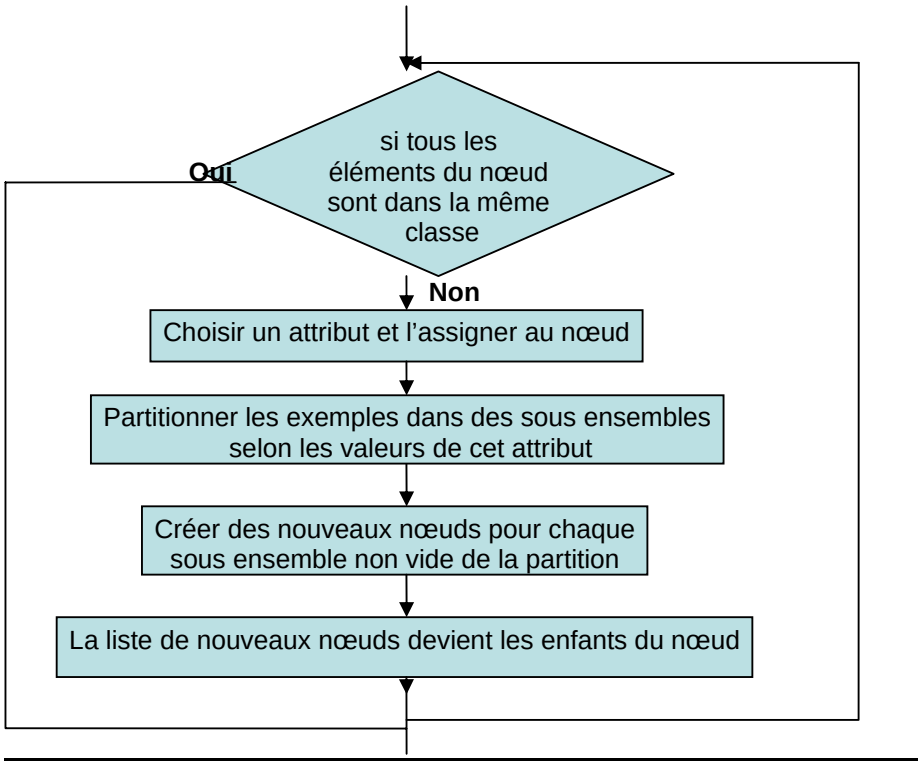
gain (X, T) = Info (X) – Info (X, T)

Grâce au gain, ID3 va pouvoir trier les attributs et construire un arbre de décision.

ID3 pose quelques problèmes. D'abord nous avons vu qu'il vaut mieux avoir des tables totalement remplies, ensuite, la quantité de calcul est assez importante. Le principe de diviser la table à chaque itération permet néanmoins d'obtenir de bons résultats. Cet algorithme n'applique pas une recherche exhaustive, mais permet de se fier aux probabilités des attributs qui ont le plus de chance d'aiguiller le résultat.

La présence du ou va provoquer la redondance des mêmes tests dans différents sous arbres. Le ou exclusif est encore plus problématique.

Notez, que l'algorithme ID3 n'est pas applicable dans des réalisations industrielles.



3. C4.5

Si on recense les problèmes que pose ID3 on se rend compte qu'il n'est pas possible de traiter correctement les champs NULL. Les attributs sont discrétisés, ce qui n'est pas toujours une solution acceptable. L'arbre produit peut comporter des sous arbres dans lesquels on ne va presque jamais.

L'algorithme C4.5 a été élaboré par Quinlan en 1993, cet algorithme n'est en fait qu'une amélioration de ID3, c'est pourquoi nous ne déroulerons pas d'exemple avec ce dernier, cela n'ayant pas d'intérêt.

Cette méthode utilise un critère plus élaboré : « le gain ratio » dont le but est de limiter la prolifération de l'arbre en pénalisant les variables qui ont beaucoup de modalités.

Parmi les améliorations qu'apporte C4.5, il y a la possibilité d'utiliser :

- Des valeurs continues pour les attributs.
- Des valeurs (discrètes) nominales d'un attribut simple peuvent être groupées ensemble, pour supporter des essais plus complexes.
- La valeur NULL
- Des attributs dont la valeur est manquante.

L'utilisation du ration du gain, plutôt que le gain lui même permet donc de ce passer de certaines valeurs. Pour l'utilisation de l'arbre avec des valeurs inconnues, on calcul la probabilité dans toutes les sous branches de l'arbre pour chaque valeur du champ « cible ».

Le calcul du ration de gain s'effectue comme suit :

$$SplitInfo(D,T) = I\left(\frac{|T_1|}{|T|}, \frac{|T_2|}{|T|}, \ldots, \frac{|T_n|}{|T|}\right)$$

$$GainRatio(D,T) = \frac{Gain(D,T)}{SplitInfo(D,T)}$$

Ainsi, C4.5 va pouvoir supprimer certains sous arbres, et va par conséquent élaguer l'arbre de décision produit, le rendant beaucoup plus simple, et plus rapide à utiliser, les résultats obtenue peuvent être très impressionnant.

C4.5 permet également d'extraire des règles depuis un arbre de décisions, ces règles s'écrivent de la forme « if LHS then RHS », où LHS est une conjonction des différentes valeurs d'attribut testés, et RHS est l'assignement d'une classe.

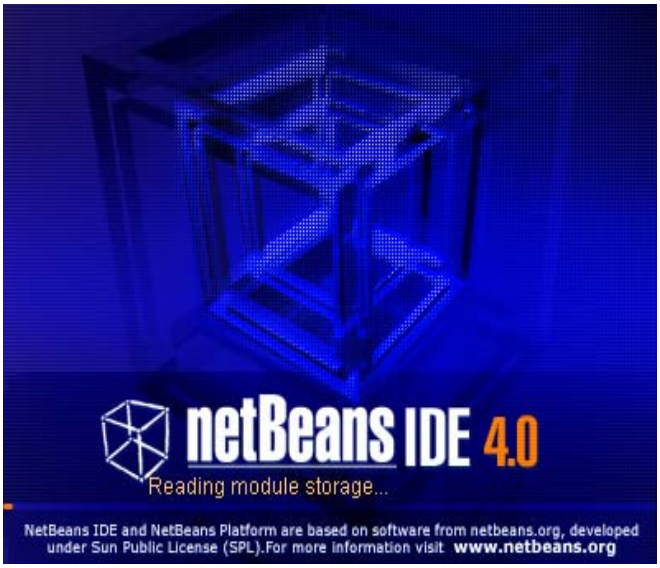
Voici les modifications de C4.5 par rapport à ID3 :

- Si la valeur est nulle, ne pas prendre en compte l'attribut pour les calculs sur le champ.
- Si champ continue, le discrétiser.
- Lorsqu'on crée un sous arbre on vérifie qu'il n'y a pas besoin de l'élaguer.
- On remplace le calcule du gain, par le calcul du ratio du gain.

Contrairement à ID3, C4.5 est parfaitement réalisable dans des applications industrielles.

I- Environnement de développement

1. NetBeans :



NetBeans c'est un outil open source ayant un succès et une base d'utilisateur très large, une communauté en croissance constante, et près de 100 partenaires mondiaux. Sun Micro System a fondé le outil open source NetBeans en Juin 2000.

Aujourd'hui, deux projets existent: L'EDI NetBeans et la Plateforme NetBeans.

L'EDI NetBeans est un environnement de développement - un outil pour les programmeurs pour écrire, compiler, déboguer et déployer des programmes. Il est écrit en Java - mais peut supporter n'importe quel langage de programmation. Il y a également un grand nombre de modules pour étendre l'EDI NetBeans. L'EDI NetBeans est un produit gratuit, sans aucune restriction quant à son usage.

Également disponible, La Plateforme NetBeans; une fondation modulable et extensible utilisée comme brique logicielle pour la création d'applications bureautiques. Les partenaires privilégiés fournissent des modules à valeurs rajoutées qui s'intègrent facilement à la Plateforme et peuvent être utilisés pour développer ses propres outils et solutions.

Les deux produits sont open source et gratuits pour un usage commercial et non-commercial. Le code source est disponible pour réutilisation sous la Licence Public Sun (SPL).

2. Oracle

Utilisé Entant que système de gestion de bases de données, pour créer la base de données, choix expliqué par le fait de pouvoir faire une étude comparative, entre notre logiciel est Oracle Datamining:

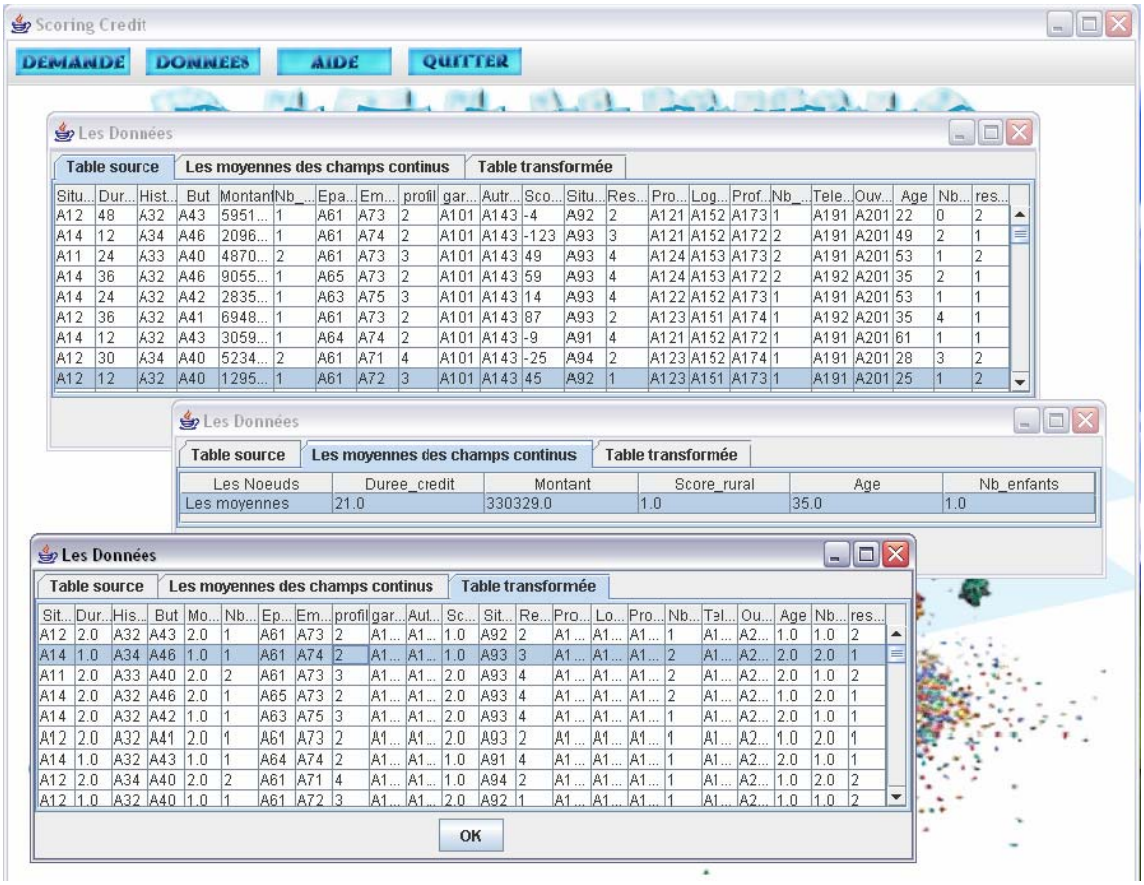
II- Réalisation :

Cette fenêtre représente le menu principal à partir duquel on peut visualiser toutes les autres fenêtres. Elle permet l'accès aux autres fenêtres pour simplifier et optimiser les tâches.

Le menu général est constitué de quatre éléments :

- ❖ Le menu « Données » : se compose de deux sous-menu :
  - Les tables : permet l'accès à la fenêtre « Données ».

- ***Fenêtre « Données » :***

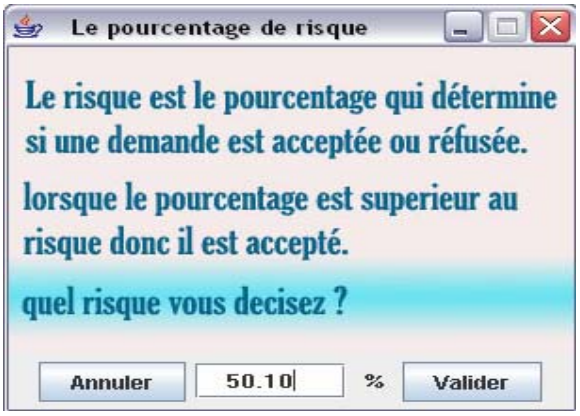


La fenêtre est formée de trois onglets :

1. Table source : affiche la base de connaissance obtenue à partir de la base de données nommée « Banque ».
2. Les moyennes de champs continus : l'algorithme de data mining implémenté C4.5 (supervisé), nécessite le calcul d'une moyenne pour les champs discrets, cet onglet affiche les cinq classes numériques avec leurs moyennes.
3. Table transformée : affiche la base de connaissance transformée, les champs numériques discrets deviennent continus par l'application de la règle « si l'attribut est supérieur à la moyenne, il sera remplacé par 2 sinon par 1 ».

- Le risque : permet l'accès à la fenêtre « Le pourcentage de risque ».

- ***Fenêtre « Le pourcentage de risque » :***

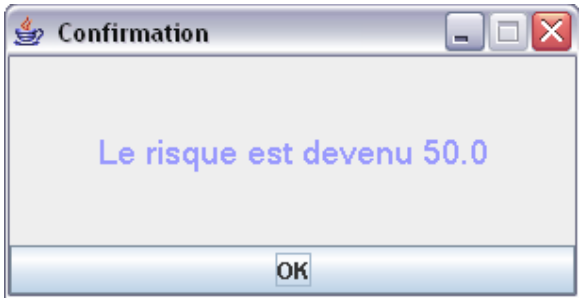


La fenêtre permet la mise à jour de risque.

Le bouton « Annuler » annule la mise à jour.

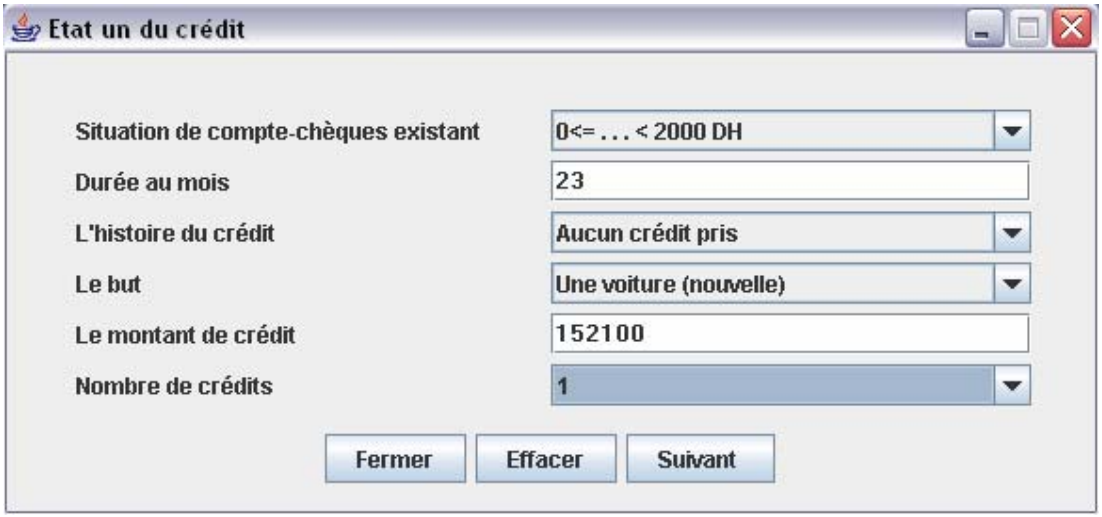
Le bouton « Valider » effectue la mise à jour, une fois valider vous recevez une fenêtre confirmant la mise à jour.

Fenêtre de confirmation :



- Menu « Demande » : donne la naissance d'une nouvelle demande de crédit.

1. Fenêtre « L'état un du crédit »



« Etat un du crédit » : c'est la première fenêtre concernant l'état client, contient un formulaire à remplir, composé des champs situation de compte de chèque existant, la durée au mois, l'histoire du crédit, le but, le montant de crédit et le nombre de crédits.

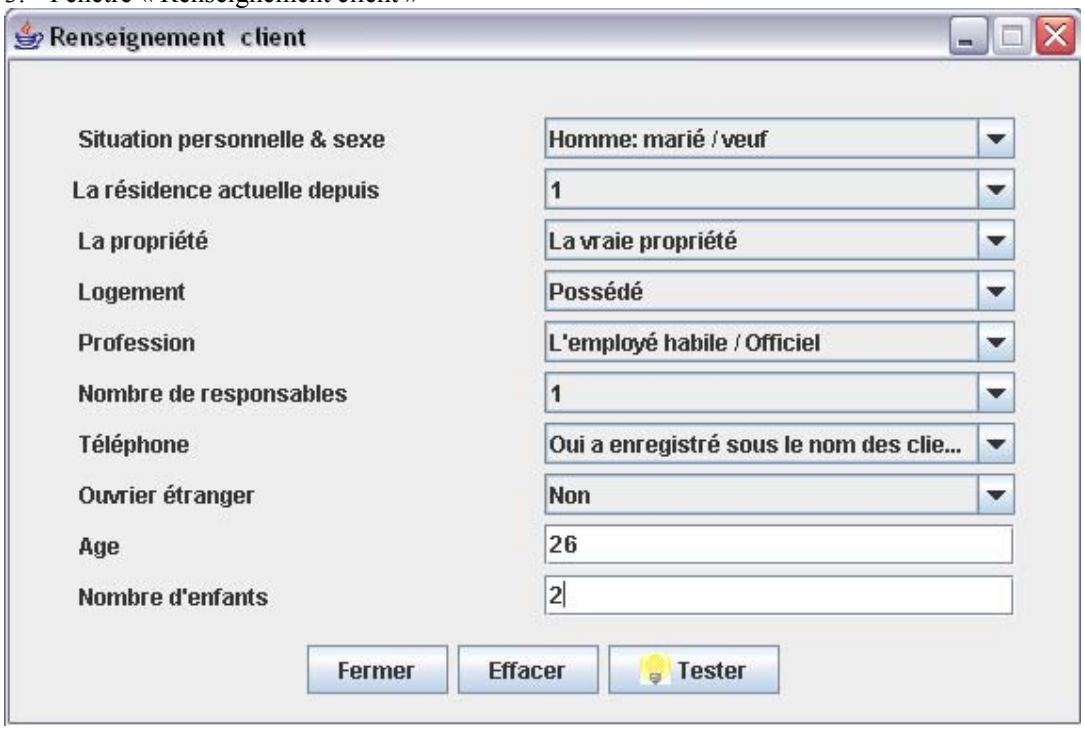
Le bouton « Suivant » donne l'accès à la deuxième fenêtre « L'état deux du crédit ».

2. Fenêtre « L'état deux du crédit »

Cette fenêtre ressemble à la première « Etat un du crédit » contient la suite des informations concernant l'état crédit.

Le bouton « Suivant » donne la main à l'utilisateur d'accéder à la troisième fenêtre « Renseignement client ».

3. Fenêtre « Renseignement client »



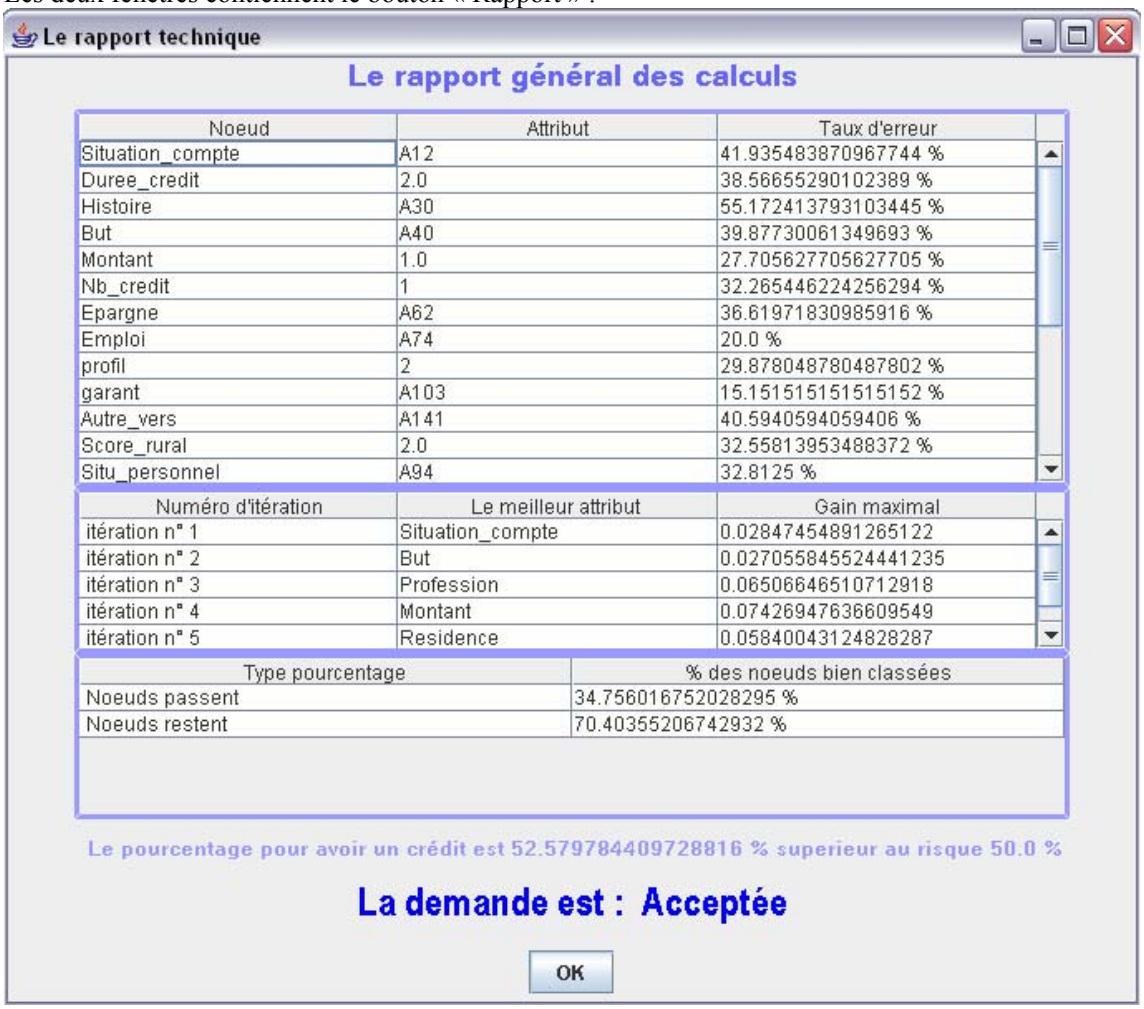
S'intéresse aux informations sociodémographiques (Situation personnelle, la résidence, la propriété, logement, profession, nombre de responsables, téléphone, age et nombre d'enfants).

Le bouton « Tester » donne le résultat de classification (Accepter ou refuser une demande de crédit) à partir des données saisies.

Si la demande de client est acceptée, la fenêtre « Résultat positif » s'affiche :

Sinon la fenêtre « Résultat négatif » s'affiche :

Les deux fenêtres contiennent le bouton « Rapport » :

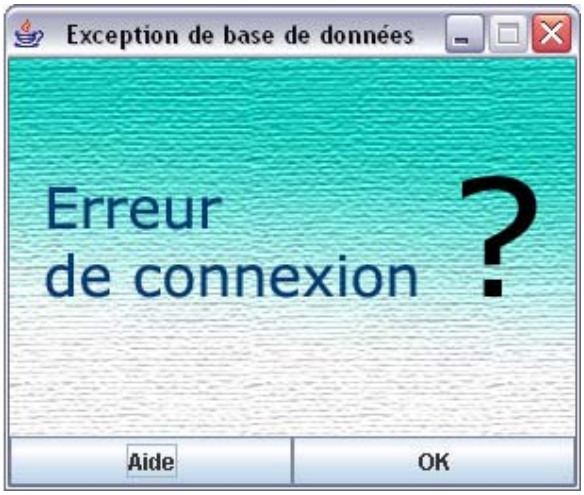


Cette fenêtre résume les différentes tâches suivies pour décider si une demande est acceptée. Le premier tableau donne le taux d'erreur pour chaque attribut saisi avec leur racine, le deuxième tableau cite les différentes itérations réalisées accompagnées par leur meilleure racine extraite avec leur gain maximum calculé, si les données saisies ne sont pas conformes avec les données de la base de connaissance, le troisième tableau sera rempli par le pourcentage des nœuds passants et le

pourcentage des nœuds restants, et un message sera affiché au dessous des tables comportant le pourcentage pour avoir un crédit comparé par le risque. Le dernier libellé affiche le résultat final de la demande de crédit.

NB : l'application intègre aussi la gestion des erreurs, pour cela nous avons créé deux fenêtres.

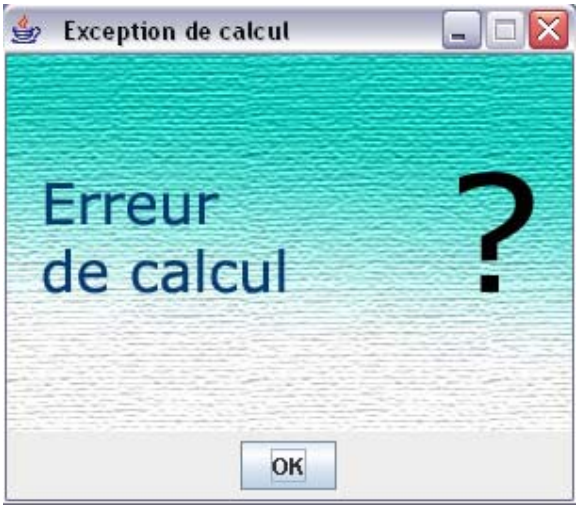
- Fenêtre « Exception de base de données »



Cette fenêtre est déclenchée lorsqu'il y a une erreur de connexion avec la base de données dans les cas suivants :

1) Au moment de création d'une nouvelle demande.
2) Au moment la mise à jour de risque.
3) Au moment de teste.

- Fenêtre « Exception de base de calcul »



Elle est déclenchée lorsqu'il y a une erreur dans les calculs résultants :

- Au moment de passer à un formulaire à l'autre sans remplir l'un ou plusieurs champs.

Ou

- Au lieu de saisir une valeur numérique, l'utilisateur saisi des autres caractères.

REFERENCES :

- http://www.stat.washington.edu/raftery/Research/Mclust/mclust papers.html.
- www-iasc.enst-bretagne.fr/˜picouet/french/projets/1a-pap/past/reglesasso/algorithmes/algo apriorihyb.html
- www.ir.iit.edu/~dagr/DataMiningCourse/Spring2001/Presentations/
- http://www.web-datamining.net/
- Johan Baltié
- . Data-Mining ID3 et C4.5
- . Epita SCIA, 2002.

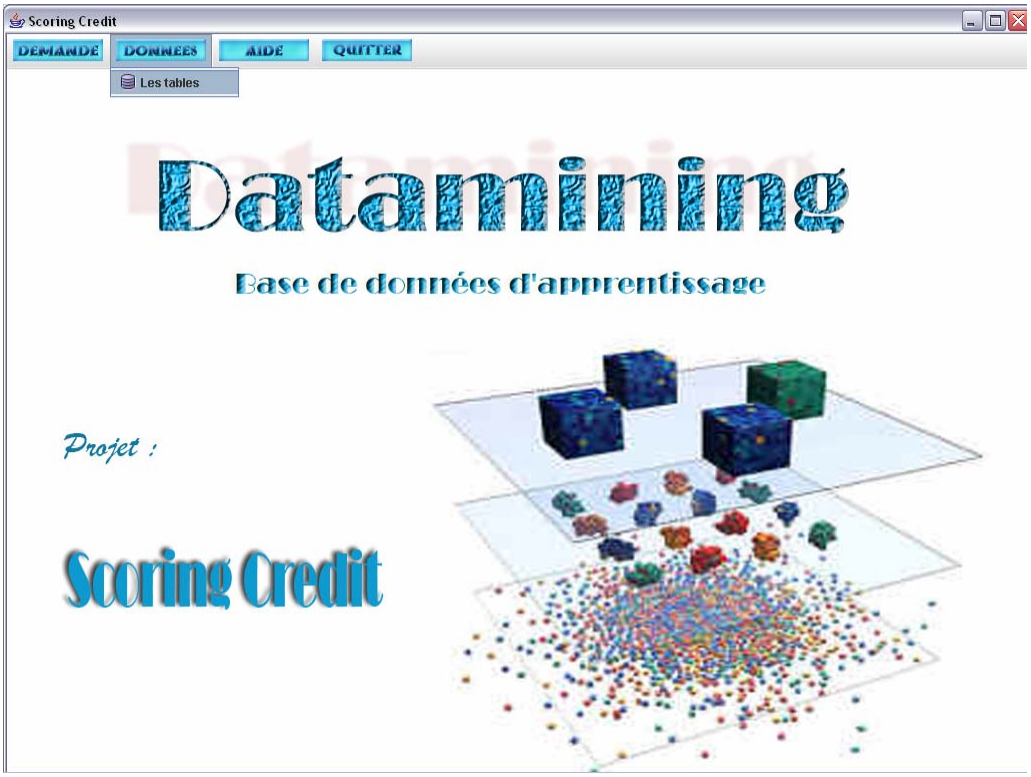
ID3

- http://cbl.leeds.ac.uk/nikos/pail/intml/subsection3.11.1.html
- http://www.hiraeth.com/books/ai96/QBB/id3.html
- http://www.risc.unilinz.ac.at/people/blurock/ANALYSIS/manual/document/node26.html
- http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm

C4.5

- http://www.aaai.org

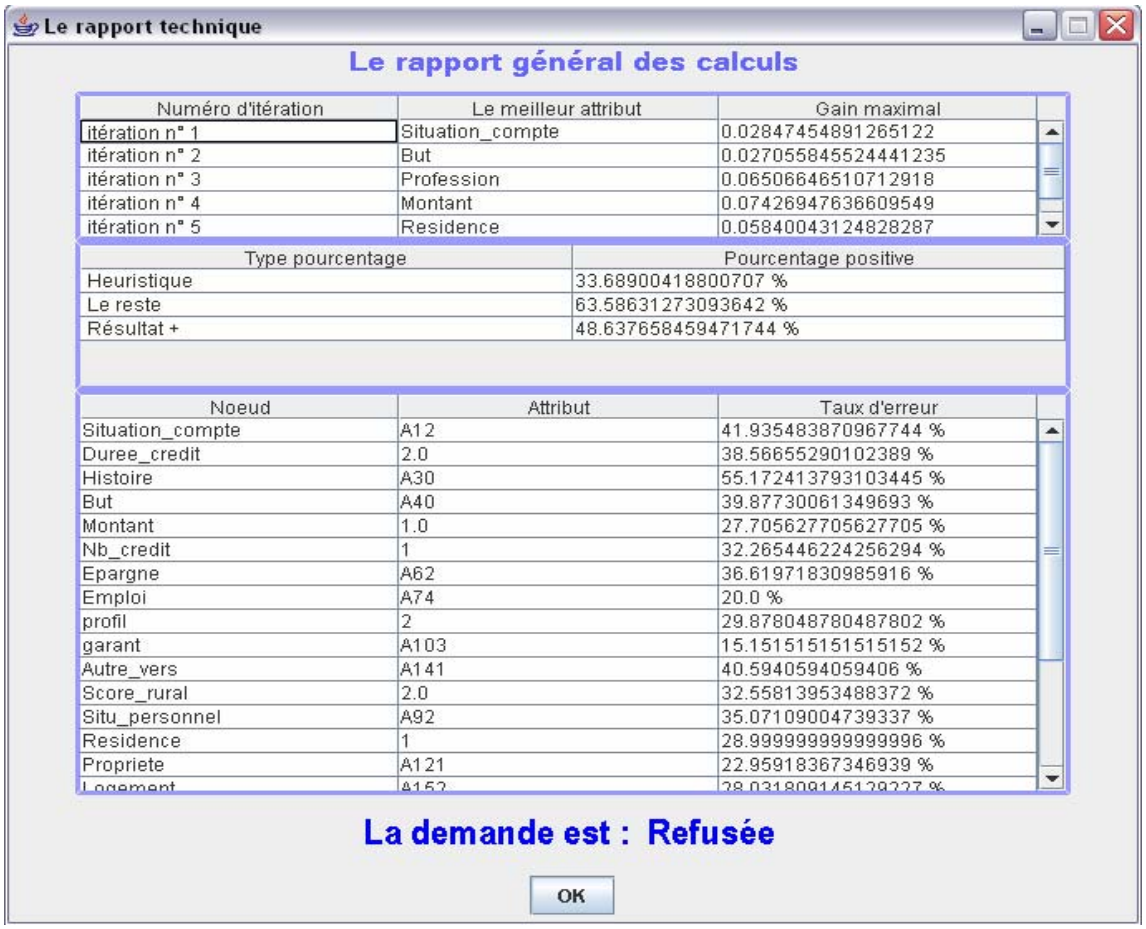
- ***Menu principal :***



Cette fenêtre représente le menu principal à partir duquel on peut visualiser toutes les autres fenêtres. Il est affiché dans chaque fenêtre pour simplifier et optimiser les tâches.

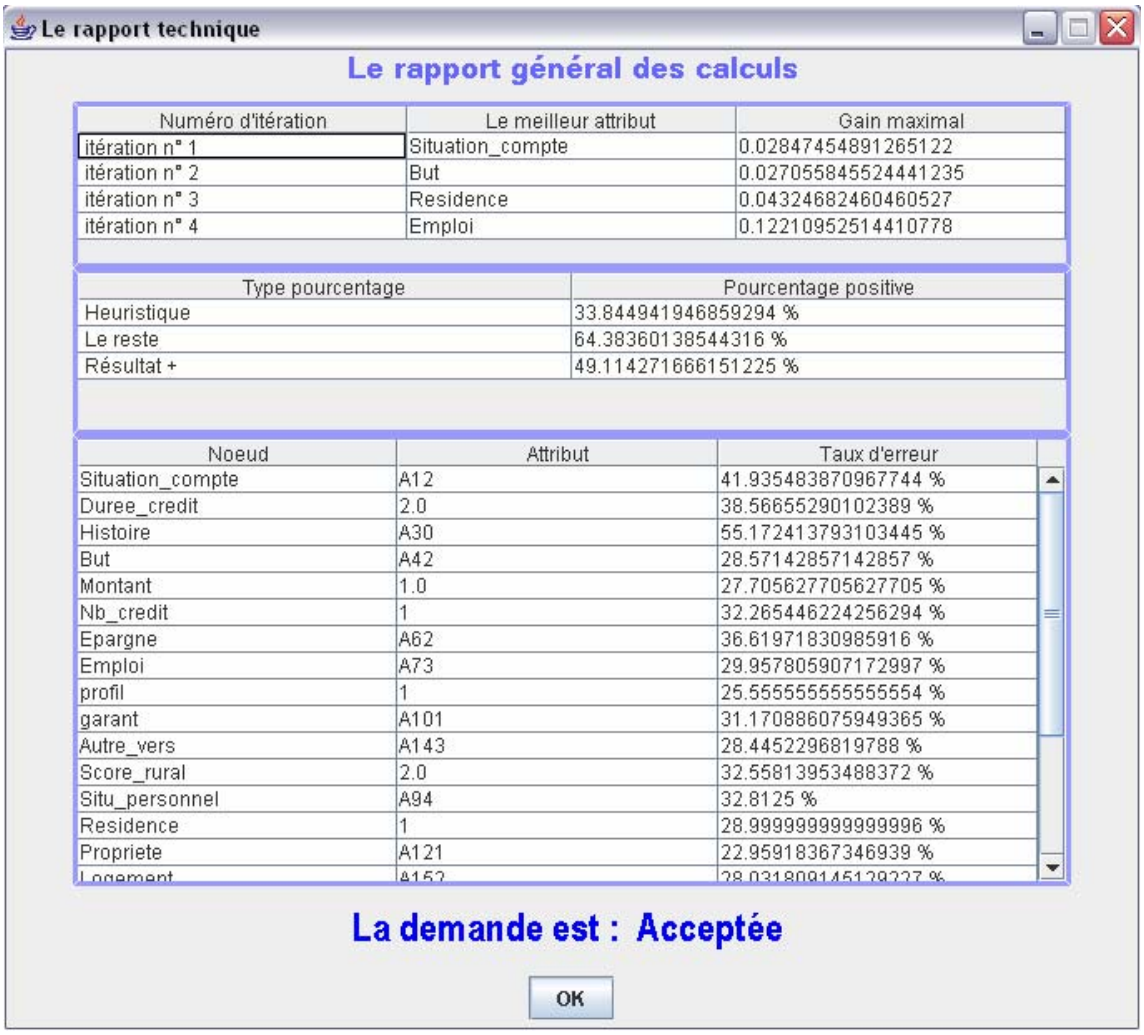
- ***Fenêtre de résultat négatif :***

- ***Le rapport général des calculs « résultat négatif »***



- ***Fenêtre de résultat positif :***

- ***Le rapport général des calculs « résultat positif »***



Mostafa. Hanoune Faculté des sciences Ben M'Sik, m_hanonune@yahoo.fr
Fouzia. Benabbou Faculté des sciences Ben M'Sik, hgfbenabbou@menara.ma