Classifieur basé sur un ensemble de profils lorsque les données sont indépendantes et identiquement distribuées

1 Introduction

Dans cette thèse, nous proposons une méthode de classement basée sur les règles d'association binaire dans le but d'améliorer les performances d'une règle de classement lorsque la classe cible de la variable réponse binaire est faiblement représentée. Généralement dans une telle situation, la règle de classement a une forte spécificité. Donc pour améliorer les performances de la règle de classement, nous nous intéressons plus aux profils de classement dont les classifieurs associés ont des sensibilités fortes.

A travers les indices de performances présentés au chapitre II, on peut affirmer que l'apprentissage du classifieur associé à un profil est fortement dépendant de la valeur prédictive positive (VPP). Généralement on estime ce dernier par le maximum de vraisemblance. Mais dans une situation où le support (la couverture) du profil est trop faible, il est recommandé d'estimer la VPP par une forme corrigée de Laplace [11] définie par

$$VPP(U,Y) = \frac{\Pr\{\phi(X,U)=1, Y=1\} + 1}{\Pr\{\phi(X,U)=1, Y=1\} + \Pr\{\phi(X,U)=1, Y=0\} + |\mathrm{Dom}(Y)|}$$

Dans la suite, nous verrons qu'il est possible d'avoir une interprétation Bayésienne de la formule de Laplace.

Soit $\mathcal{D}_n = (y_i, x_i)$ un ensemble fini d'éléments générés de façon aléatoire par la loi du couple (Y, X), où Y est une variable binaire et $X = (X_j)_{j=1:p}$ est un vecteur de variables aléatoires, où la variable X_j peut être numérique ou catégorielle. A l'aide des outils statistiques présentés dans le chapitre II, nous présentons un algorithme d'apprentissage dont les performances sont comparables avec d'autres méthodes très connues pour un classement binaire.

2 Algorithme d'apprentissage d'un classifieur basé sur un ensemble de profils

Dès qu'un phénomène, qu'il soit physique, biologique ou autre, est trop complexe ou encore trop bruité pour accéder à une description analytique débouchant sur une modélisation déterministe, un ensemble d'approches est élaboré afin d'en décrire au mieux le comportement à partir d'une série d'observations. On appelle apprentissage statistique l'ensemble d'approches élaboré [5]. C'est une combinaison à la fois de l'apprentissage automatique et de la statistique [26]. L'apprentissage automatique consiste à utiliser des ordinateurs pour optimiser un modèle de traitement de l'information selon certains critères de performance à partir d'observations. Tandis que la statistique permet de formaliser le processus, de garantir sa qualité et éventuellement de suggérer de nouvelles techniques. Cependant le principe de l'apprentissage reste le même, mais la démarche est différente selon que la taille du jeu de données est grande ou petite.

2.1 Présentation de l'algorithme de construction du classifieur

Lorsque la taille des données est suffisamment grande, on adoptera l'approche Apprentissage/Validation/Test pour la sélection d'un ensemble optimal de profils. Cette approche consiste à subdiviser les données de manière aléatoire en trois ensembles : un ensemble d'apprentissage, un ensemble de validation et un ensemble test. L'apprentissage statistique que nous proposons peut être résumée par les différentes étapes suivantes :

1. Discrétiser toutes les variables numériques par une méthode de discrétisation (au choix)
2. A partir d'un ensemble d'apprentissage :
   (a) Spécifier le paramètre d'apprentissage $\lambda = (s_0, c_0, l_0)$
   (b) Générer un ensemble $\mathcal{U}_\lambda$ de profils
   (c) Elaguer les profils redondants dans $\mathcal{U}_\lambda$ pour constituer un petit ensemble

$$\mathcal{U}_\lambda^1 = \{[\phi(X, U) = 1] \rightarrow [Y = 1]; U \in \mathcal{U}_\lambda\}$$

3. A partir d'un ensemble de validation :
   (a) Réévaluer l' indicateur de performance VPP (ou RVP ou RVN) de toutes les règles dans $\mathcal{U}_\lambda^1$
   (b) Supprimer les profils dont le RVP est inférieur à un (1)
   (c) Parmi les profils dans $\mathcal{U}_\lambda^1$ qui sont emboîtés, ne retenir que le profil dont le VPP (ou le RVP ou le RVN) est le plus significatif.

4. Au sortir de l'étape 3, on dispose alors d'un ensemble de profils $\mathcal{U}_\lambda^2$ tel que $|\mathcal{U}_\lambda^2| \leq |\mathcal{U}_\lambda^1|$.
5. Définir la règle de classement (classifieur) ϕ d'une observation X par

$$\phi(X,\lambda) = \begin{cases} 1 & \text{si} \quad \sum_{m=1}^{|\mathcal{U}_\lambda^2|} \phi(X, U_m) > 0 \\ 0 & \text{sinon} \end{cases}$$

Le classifieur $\phi(X,\lambda)$ est un cas particulier du classifieur défini au chapitre II à la section 3.2 où on a choisi k égal à zéro. On choisit alors de classer positive une observation X lorsqu'elle vérifie au moins un profil parmi ceux qui sont dans l'ensemble $\mathcal{U}_\lambda^2$.

Dans tout ce qui suit, on fixe à un le nombre minimum de profils à vérifier pour qu'une observation soit classée positive.

3 Prétraitement des données : discrétisation des covariables numériques

Un ensemble de données pour un classement est normalement sous la forme d'un tableau de données qui est décrit par un ensemble de variables distinctes. La plupart des applications réelles (données réelles) pour une classification supervisée comportent à la fois des variables numériques (continues) et des variables nominales (catégorielles). Certaines méthodes de classement, particulièrement l'algorithme d'apprentissage des règles d'association, exigent que toutes les covariables soient nominales. Ainsi il est nécessaire de convertir les variables continues en des variables discrètes. L'idée consiste à transformer chaque variable numérique X_j en une variable catégorielle X_j^*. La variable X_j^* est obtenue en subdivisant le domaine des valeurs de X_j en q_j intervalles $m_h^{X_j}, h = 1 : q_j$. La variable X_j^* sera utilisée à la place de X_j pour construire le classifieur.

En général une variable continue est une variable dont le domaine de définition est totalement ordonné. La discrétisation doit être choisie de manière à apporter des informations de classification utiles sans modifier les classes auxquelles les observations du domaine de la variable appartiennent. En général, une discrétisation est simplement une condition logique, en termes d'une ou plusieurs valeurs évaluées, qui sert à partitionner les données en au moins deux sous-ensembles. Supposons que X_j soit une variable numérique et l'intervalle $[a,b]$ soit son domaine. Une partition π_{X_j} sur $[a,b]$ est définie comme le sous-ensemble des k intervalles suivants

$$\pi_{X_j} = \{[x_{j0}, x_{j1}), [x_{j1}, x_{j2}), \ldots, [x_{j(k-1)}, x_{jk}]\}$$

où $x_{j0} = a$, $x_{j(i-1)} < x_{ji}$ pour $i = 1 : k$ et $x_{jk} = b$. Ainsi la discrétisation est le processus qui produit

une partition π_{X_j} sur $[a, b]$.

Plusieurs méthodes de discrétisation des variables numériques ont été étudiées dans la littérature. On peut, par exemple, considérer des combinaisons linéaires de plusieurs variables et comparer le résultat avec un seuil (Breiman et al., 1984)[7]. Il est aussi possible d'éviter le seuillage en formant une condition qui compare les valeurs de deux ou plusieurs variables directement. Cependant le nombre de telles expressions possibles rend l'espace de recherche très vaste.
La méthode de discrétisation d'une variable numérique la plus simple reste la méthode de largeur d'intervalle égale (Equal Interval Width Method). Elle consiste à partitionner son domaine en intervalles de largeur égales.

Une méthode de discrétisation de variable numérique par la discrétisation adaptative a été proposée dans [8]. La méthode consiste à diviser d'abord le domaine de la variable en deux intervalles de largeur égale et un processus d'apprentissage est lancé pour générer les règles. Ensuite, la qualité des règles est testée en évaluant les performances des règles. Si la mesure de performance est inférieure à un seuil fixe, l'un des intervalles est subdivisé en outre, et le processus est répété. Le principal inconvénient de cette méthode, cependant, est la répétition du processus d'apprentissage jusqu'à ce que le niveau de performance finale soit atteint.

Une discrétisation basée sur l'entropie marginale maximale a été introduite dans [30]. Ce procédé consiste à diviser le domaine de la variable numérique de telle sorte que la fréquence d'échantillonnage dans chaque intervalle soit approximativement égale. Ce procédé est généralement appelé la méthode par intervalle de fréquence égale (Equal Frequency per Interval Method). Le seul paramètre fourni par l'utilisateur est le nombre d'intervalles à induire sur le domaine d'origine. La discrétisation par la mesure de l'entropie utilise les bornes du domaine de la variable pour induire les intervalles souhaités. Cette méthode de sélection d'un point de coupure est utilisée dans l'algorithme ID3 [23], dans l'algorithme CART [6], et d'autres [15].

Lorsque nous traitons un problème de classification supervisée, il est naturel de penser à discrétiser les variables numériques en fonction de la variable réponse. Ceci constitue l'un des points faibles des différentes méthodes de discrétisation citées précédemment. Ce concept est pris en compte par la méthode de discrétisation avec la classe-entropie comme critère pour sélectionner le meilleur point de coupure [13]. Dans tout ce qui suit, nous avons utilisé la méthode de discrétisation dont le critère d'arrêt est basé sur le principe de la longueur de description minimum plus connu sous le nom de MDLP (Minimum Description Length Principle). Cette méthode est initiée par Fayyad et Irani [13, 14]. La méthode est présentée comme une méthode efficace pour la discrétisation pour l'apprentissage des arbres de décision et du classifieur de Bayes Naïf [2] (voir l'annexe B pour plus de détails).

4 Extraction d'un ensemble initial de profils

L'ensemble des profils $\mathcal{U}_\lambda$, généré au départ pour l'apprentissage du classifieur, est caractérisé par c_0, une estimation de la VPP, et s_0, une estimation du support. L'un des plus connus algorithmes d'exploration des règles d'association, utilisant c_0 et s_0 pour l'extraction des règles les plus fréquentes, reste l'algorithme *"apriori"*. Il est l'un des algorithmes d'extraction de règles d'association qui a utilisé en premier l'élagage basé sur le support pour contrôler systématiquement la croissance exponentielle des règles candidates. C'est la raison pour laquelle, nous avons choisi de l'utiliser pour la suite. On pouvait utiliser d'autres algorithmes d'extraction de règles fréquentes existant dans la littérature par exemple l'algorithme *"FP-Growth"*(*FPtree structure*) [17]. Un choix de l'algorithme d'extraction est laissé à l'utilisateur. Ci-après (Tableau III.1), nous présentons un pseudo code de la partie de génération des profils fréquents par l'algorithme *"apriori"*. Soit C_k l'ensemble des profils de longueur k candidats, $\mathcal{D}$ l'ensemble de toutes les observations et F_k l'ensemble des profils fréquents et de longueur k.

Algorithme : Génération de règles fréquentes par l'algorithme *"apriori"*

– Entrées : $\mathcal{D}$ un ensemble d'observations, s_0 un support minimum et c_0 une confiance minimum
– Sorties : $\mathcal{U}_\lambda$ un ensemble de profils fréquents

1 : k=1
2 : $F_k =$ {Trouver tous les 1-itemsets fréquents}
3 : **répéter**
4 : k=k+1
5 : C_k = apriori-gen(F_{k-1}). {Générer les profils candidats}
6 : **pour** chaque observation $t \in D$ **faire**
7 : C_t = subset(C_k , t). {Identifier tous les candidats contenus dans t}
8 : **pour** chaque profil candidat $c \in C_t$ **faire**
9 : $supp(c) = supp(c) + 1$. {Incrémenter le compte du support}
10 : **si** $t.class = c.class$ **faire** { $t.class$: la classe associée à l'observation t}
11 : $conf(c) = conf(c) + 1$. {Incrémenter le compte de la confiance}
12 : **fin si**
13 : **fin pour**
14 : **fin pour**
15 : F_k ={c $\in C_k$ | supp(c) $\geq s_0$; conf(c)/supp(c) $\geq c_0$ }
{Extraire les profils fréquents de taille k}
16 : **jusqu'à** $F_k = \emptyset$
17 : Retourner : $\mathcal{U}_\lambda = \bigcup_k F_k$

Tableau III.1 – Algorithme de génération des règles fréquentes (*"apriori"*)

Pour la suite, nous nous intéresserons aux profils générés à partir de l'algorithme *"apriori"* qui sont corrélés avec la variable réponse et qui vérifient les conditions d'apprentissages suivantes : support $\geq s_0$,

confiance $\geq c_0$, risque relatif $\geq r_0$, taille $\leq l_0$. Cette étape de l'algorithme est élaborée sur l'échantillon d'apprentissage. Au sortir de cette phase, un vaste ensemble $\mathcal{U}_\lambda$, $\lambda = (s_0, c_0, r_0, l_0)$ contenant à la fois des profils redondants et des profils de faibles performances, est généré. Il est donc nécessaire d'élaborer une procédure d'élagage des profils redondants pour réduire le vaste ensemble $\mathcal{U}_\lambda$ à un ensemble $\mathcal{U}_\lambda^1$ ne contenant que des profils fréquents et non redondants.

5 Elagage des profils redondants

Dans cette section, nous nous intéressons aux profils qui sont liés à la variable réponse. La suppression des profils qui ne sont pas corrélés à la variable réponse et des profils redondants permettra de sélectionner un ensemble réduit de profils dont on pourra se servir pour construire un classifieur performant.

Soient $U_1 = \left(m_h^{X_j}\right)_{j\in J}$ et $U_2 = \left(m_h^{X_l}\right)_{l\in L}$ deux profils tels que U_2 soit emboîté dans U_1. L'application des résultats théoriques précédents nécessite de faire un test d'hypothèse sur l'égalité des couvertures, sur l'égalité des supports ou sur l'égalité des spécificités de deux profils emboîtés. Pour cela, il est possible de faire un test stochastique

5.1 Test stochastique (randomisé) pour la sélection entre deux profils emboîtés

En principe, si l'égalité n'est pas vérifiée sur un échantillon donné, on peut affirmer qu'elle n'est pas vérifiée sur la population dont est issu l'échantillon. Par contre on ne peut pas en dire autant lorsqu'elle est vraie sur un échantillon. C'est la raison pour laquelle un test stochastique (ou test randomisé) est nécessaire.

On note par $\phi(X, U_1) = \prod_{j\in J} \mathbb{1}\left(X_j = m_h^{X_j}\right)$ et $\phi(X, U_2) = \prod_{l\in L} \mathbb{1}\left(X_l = m_h^{X_l}\right)$ les fonctions de classement générées respectivement par U_1 et U_2. Puisque U_2 est emboîté dans U_1, on a $[\phi(X, U_2) = 1] \subset [\phi(X, U_1) = 1]$.

(a) Soit le paramètre θ_1 défini par $\theta_1 = \Pr(\phi(X, U_1) = 1) - \Pr(\phi(X, U_2) = 1)$. Nous voulons tester si oui ou non θ_1 est nulle i.e décider entre les deux hypothèses

$$H_0^1 : \theta_1 = 0 \quad vs \quad H_1^1 : \theta_1 \neq 0$$

Nous allons considérer la variable aléatoire définie par

$$Z_1(X) = \phi(X, U_1) - \phi(X, U_2)$$

Puisque $[\phi(X,U_2)=1] \subset [\phi(X,U_1)=1]$, on peut écrire

$$Z_1(X) = \begin{cases} 1 & \text{si } \phi(X,U_1)=1 \text{ et } \phi(X,U_2)=0 \\ \\ 0 & \text{si } \phi(X,U_1)=\phi(X,U_2) \end{cases}$$

(b) Pour tester l'égalité des sensibilités de U_1 et U_2, on considère le paramètre θ_2 défini par $\theta_2 = \Pr([\phi(X,U_1)=1, Y=1]) - \Pr([\phi(X,U_2)=1, Y=1])$. Les hypothèses à tester sont :

$$H_0^2 : \theta_2 = 0 \quad vs \quad H_1^2 : \theta_2 \neq 0$$

On peut associer au test la variable aléatoire $Z_2(X)$ définie par

$$Z_2(X) = \mathbb{1}\left([\phi(X,U_1)=1, Y=1]\right) - \mathbb{1}\left([\phi(X,U_2)=1, Y=1]\right)$$

Puisque $[\phi(X,U_2)=1, Y=1] \subset [\phi(X,U_1)=1, Y=1]$, on peut écrire

$$Z_2(X) = \begin{cases} 1 & \text{si } \mathbb{1}\left(\phi(X,U_1)=1, Y=1]\right)=1 \text{ et } \mathbb{1}\left(\phi(X,U_2)=1, Y=1]\right)=0 \\ \\ 0 & \text{si } \mathbb{1}\left(\phi(X,U_1)=1, Y=1]\right)=\mathbb{1}\left([\phi(X,U_2)=1, Y=1]\right) \end{cases}$$

(c) Pour tester l'égalité des spécificités de U_1 et U_2, on considère le paramètre θ_3 défini par $\theta_3 = \Pr([\phi(X,U_2)=0, Y=0]) - \Pr([\phi(X,U_1)=0, Y=0]$. L' hypothèse nulle et son alternative sont données par :

$$H_0^3 : \theta_3 = 0 \quad vs \quad H_1^3 : \theta_3 \neq 0$$

La variable aléatoire $Z_3(X)$ associée au test est définie par

$$Z_3(X) = \mathbb{1}\left([\phi(X,U_2)=0, Y=0]\right) - \mathbb{1}\left([\phi(X,U_1)=0, Y=0]\right)$$

Puisque $[\phi(X,U_2)=0, Y=0] \supset [\phi(X,U_1)=0, Y=0]$, on peut écrire

$$Z_3(X) = \begin{cases} 1 & \text{si } \mathbb{1}\left([\phi(X,U_1)=0, Y=0]\right)=0 \text{ et } \mathbb{1}\left([\phi(X,U_2)=0, Y=0]\right)=1 \\ \\ 0 & \text{si } \mathbb{1}\left([\phi(X,U_1)=0, Y=0]\right)=\mathbb{1}\left([\phi(X,U_2)=0, Y=0]\right) \end{cases}$$

Les variables $(Z_k(X))_{k=1:3}$ sont donc des variables aléatoires de Bernoulli de paramètre $(\theta_k)_{k=1:3}$.

On considère une suite d'éléments aléatoires $\mathcal{D}_n = (X_i, Y_i)_{i\in 1:n}$ indépendants et identiquement distribués, où Y_i est une réalisation d'une variable de Bernoulli Y et X_i est une suite finie de p réalisations d'un vecteur de variables aléatoires non numériques $(X_j)_{j=1:p}$ à q_j modalités $m_h^{X_j}; h = 1 : q_j, j = 1 : p$. Puisque les observations $(X_i)_{i=1:n}$ sont indépendantes alors les $Z_k(X_i)_{i=1:n}$ constituent des réalisations indépendantes. Donc la somme $\sum_{i=1}^n Z_k(X_i)$ est une réalisation d'une variable aléatoire suivant la loi binomiale $\mathcal{BN}(n, \theta_k)$. Nous considérons le test stochastique défini comme suit : Pour tout $k = 1 : 3$

$$\varphi_k(\mathcal{D}_n) = \begin{cases} 1 & \text{si } \sum_{i=1}^n Z_k(X_i) > 0 \\ \\ 1-\gamma_k & \text{si } \sum_{i=1}^n Z_k(X_i) = 0 \quad et \quad 0 < \gamma_k \leq 1 \end{cases}$$

On tire un nombre μ uniformément réparti entre 0 et 1. Si $\mu \geq 1-\gamma_k$ on rejette H_0^k et si $\mu < 1-\gamma_k$ on accepte H_0^k avec $0 < \gamma_k \leq 1$. L'application du test stochastique s'effectue comme suit :

- Si $\varphi_k(\mathcal{D}_n) = 1$: rejeter H_0^k
- Si $\varphi_k(\mathcal{D}_n) = 1-\gamma_k$: rejeter H_0^k avec une probabilité γ_k i.e. on génère une valeur μ uniforme sur 0 et 1. Si $\mu \geq 1-\gamma_k$, on rejette H_0^k, sinon on accepte.

Le niveau du test est obtenu en calculant

$$\begin{aligned}
\Pr\left(\text{rejeter } H_0^k | H_0^k\right) &= \Pr\left(\varphi_k(\mathcal{D}_n) = 1 \,|\, H_0^k\right) + \Pr\left(\varphi_k(\mathcal{D}_n) = 1-\gamma_k, \mu \geq 1-\gamma_k \,|\, H_0^k\right) \\
&= \Pr\left(\varphi_k(\mathcal{D}_n) = 1 \,|\, H_0^k\right) + \Pr\left(\varphi_k(\mathcal{D}_n) = 1-\gamma_k | \, H_0^k\right) \Pr\left(\mu \geq 1-\gamma_k \,|\, H_0^k\right) \\
&= \Pr\left(\sum_{i=1}^n Z_k(X_i) > 0 \,|\, H_0^k\right) + \Pr\left(\sum_{i=1}^n Z_k(X_i) = 0 \,|\, H_0^k\right) \Pr\left(\mu \geq 1-\gamma_k\right) \\
&= 1 - \Pr\left(\sum_{i=1}^n Z_k(X_i) = 0 \,|\, H_0^k\right) + \Pr\left(\sum_{i=1}^n Z_k(X_i) = 0 \,|\, H_0^k\right) \left(1 - \Pr\left(\mu < 1-\gamma_k\right)\right) \\
&= 1 - \Pr\left(\sum_{i=1}^n Z_k(X_i) = 0 \,|\, H_0^k\right) + \Pr\left(\sum_{i=1}^n Z_k(X_i) = 0 \,|\, H_0^k\right) \gamma_k \\
&= \gamma_k \qquad \text{puisque} \quad \Pr\left(\sum_{i=1}^n Z_k(X_i) = 0 \,|\, H_0^k\right) = 1
\end{aligned}$$

Et on obtient la puissance du test en calculant

$$\begin{aligned}
\Pr\left(\text{rejeter } H_0^k | H_1^k\right) &= \Pr\left(\varphi_k(\mathcal{D}_n) = 1 \,|\, H_1^k\right) + \Pr\left(\varphi_k(\mathcal{D}_n) = 1-\gamma_k, \mu \geq 1-\gamma_k \,|\, H_1^k\right) \\
&= \Pr\left(\varphi_k(\mathcal{D}_n) = 1 \,|\, H_1^k\right) + \Pr\left(\varphi_k(\mathcal{D}_n) = 1-\gamma_k | \, H_1^k\right) \Pr\left(\mu \geq 1-\gamma_k \,|\, H_1^k\right) \\
&= \Pr\left(\sum_{i=1}^n Z_k(X_i) > 0 \,|\, H_1^k\right) + \Pr\left(\sum_{i=1}^n Z_k(X_i) = 0 \,|\, H_1^k\right) \Pr\left(\mu \geq 1-\gamma_k\right) \\
&= 1 - \Pr\left(\sum_{i=1}^n Z_k(X_i) = 0 \,|\, H_1^k\right) + \Pr\left(\sum_{i=1}^n Z_k(X_i) = 0 \,|\, H_1^k\right) \left(1 - \Pr\left(\mu < 1-\gamma_k\right)\right) \\
&= 1 - \Pr\left(\sum_{i=1}^n Z_k(X_i) = 0 \,|\, H_1^k\right) \Pr\left(\mu < 1-\gamma_k\right) \\
&= 1 - (1-\theta_k)^n (1-\gamma_k)
\end{aligned}$$

5.2 Algorithme de la procédure d'élagage

En se basant sur les résultats présentés dans la section précédente, on peut proposer une procédure d'élagage des profils redondants comme suit.

Algorithme : Procédure d'élagage des profils redondants

– Entrées : $\mathcal{R}$ un ensemble de profils
– Sorties : $\mathcal{R}'$ un ensemble de profils non redondants

1 : On se donne $\mathcal{R}$ un ensemble de profils
2 : **pour** tout profil $U \in \mathcal{R}$ **faire**

3 : $\mathcal{S}_U = \text{subset}(U, \mathcal{R})$ {le sous-ensemble de profils de $\mathcal{R}$ emboîtés dans U}

4 : **pour** tout profil $U' \in \mathcal{S}_U$ **faire**

5 : Tester $H_0^1 : \Pr\{\phi(X,U) = 1\} = \Pr\{\phi(X,U') = 1\}$ vs $H_1^1 : \Pr\{\phi(X,U) = 1\} \neq \Pr\{\phi(X,U') = 1\}$

6 : **Si** H_0^1 est vraie, $\mathcal{S}'_U = \text{delete}(U', \mathcal{S}_U)$ {supprimer U' de $\mathcal{S}_U$ en vertu de la proposition 3.}

7 : **Sinon**
8 : Tester $H_0^2 : \Pr\{\phi(X,U) = 1, Y = 1\} = \Pr\{\phi(X,U') = 1, Y = 1\}$ contre son opposée H_1^2

9 : Si H_0^2 est vraie, $\mathcal{S}'_U = \text{delete}(U, \mathcal{S}_U)$ {supprimer U de $\mathcal{S}_U$ en vertu de la proposition 4.}

10 : Tester $H_0^3 : \Pr\{\phi(X,U) = 0, Y = 0\} = \Pr\{\phi(X,U') = 0, Y = 0\}$ contre son opposée H_1^3

11 : Si H_0^3 est vraie, $\mathcal{S}'_U = \text{delete}(U', \mathcal{S}_U)$ {supprimer U' de $\mathcal{S}_U$ selon la proposition 5.}

12 : **fin si**
13 : **fin pour** U'
14 : **fin pour** U
15 : Retourner $\mathcal{R}' = \bigcup_{U \in \mathcal{R}} \mathcal{S}'_U$

Tableau III.2 – Algorithme d'élagage des profils redondants

Le test stochastique présenté ci-dessus est applicable quelle que soit la taille des données d'analyse. Habituellement, l'ensemble $\mathcal{U}_\lambda^1$ contient un grand nombre de profils, certainement plus qu'il en faut pour construire une fonction de classification qui est efficace et facile à mettre en œuvre.

6 Détermination d'un ensemble optimal de profils

6.1 Lorsque les données sont de grande taille

D'une manière générale, on peut utiliser un test comparant les valeurs prédictives positives de deux profils emboîtés pour sélectionner le profil le plus adéquat. Ce test est basé sur la normalité asymptotique du logarithme de rapport des valeurs prédictives positives des deux profils emboîtés.

6.1.1 Test d'hypothèse asymptotique pour la sélection d'un ensemble optimal de profils

Proposition 6. Soient $U_1 = \left(m_h^{X_j}\right)_{j\in J}$ et $U_2 = \left(m_h^{X_j}\right)_{j\in L}$ deux profils emboîtés tels que $J \subset L$. Soient $\widehat{VPP}(U_1, Y)$ et $\widehat{VPP}(U_2, Y)$ les estimateurs empiriques de $VPP(U_1, Y)$ et $VPP(U_2, Y)$ respectivement. La variable aléatoire $log\left(\frac{\widehat{VPP}(U_1,Y)}{\widehat{VPP}(U_2,Y)}\right)$ est asymptotiquement distribuée suivant une loi normale centrée de variance

$$\Sigma = \sum_{i=1}^{6} p_i \nabla_i^2 - \left(\sum_{i=1}^{6} p_i \nabla_i\right)^2$$

où

$$\begin{pmatrix} \nabla_1 \\ \nabla_2 \\ \nabla_3 \\ \nabla_4 \\ \nabla_5 \\ \nabla_6 \end{pmatrix} = \begin{pmatrix} \frac{1}{p_1+p_4} + \frac{1}{p_1+p_2} - \frac{1}{p_1} - \frac{1}{p_1+p_2+p_4+p_5} \\ \frac{1}{p_1+p_2} - \frac{1}{p_1+p_2+p_4+p_5} \\ 0 \\ \frac{1}{p_1+p_4} - \frac{1}{p_1+p_2+p_4+p_5} \\ -\frac{1}{p_1+p_2+p_4+p_5} \\ 0 \end{pmatrix}$$

Preuve. Soit le vecteur aléatoire $(Y, \phi(X, U_1), \phi(X, U_2))$. On considère les événements suivants :

$$E_1 = \{Y = 1, \phi(X, U_1) = 1, \phi(X, U_2) = 1\} \qquad E_2 = \{Y = 1, \phi(X, U_1) = 1, \phi(X, U_2) = 0\}$$
$$E_3 = \{Y = 1, \phi(X, U_1) = 0, \phi(X, U_2) = 0\} \qquad E_4 = \{Y = 0, \phi(X, U_1) = 1, \phi(X, U_2) = 1\}$$
$$E_5 = \{Y = 0, \phi(X, U_1) = 1, \phi(X, U_2) = 0\} \qquad E_6 = \{Y = 0, \phi(X, U_1) = 0, \phi(X, U_2) = 0\}$$

dont les probabilités de réalisation sont p_1, p_2, p_3, p_4, p_5 et p_6 respectivement avec

$$\sum_{i=1}^{6} p_i = 1$$

Compte tenu du fait que U_2 soit emboîté dans U_1, on a

$$p_1 = \Pr(Y = 1, \phi(X, U_2) = 1) \qquad p_4 = \Pr(Y = 0, \phi(X, U_2) = 1)$$
$$p_3 = \Pr(Y = 1, \phi(X, U_1) = 0) \qquad p_6 = \Pr(Y = 0, \phi(X, U_1) = 0)$$

On note par $I_{E_k}, k = 1 : 6$ la fonction indicatrice de l'événement E_k. La distribution de Bernoulli généralisée de paramètres $\theta = (p_1, \ldots, p_6)$ de la variable aléatoire $Z = (I_{E_1}, \ldots, I_{E_6})$ admet comme matrice de variance covariance la matrice

$$\Lambda(\theta) = diag(\theta) - \theta^T\theta$$

Soit $(Z_i)_{i=1:n}$ une suite indépendante de distribution la Bernoulli généralisée. Si on considère

$$\widehat{\theta}_n = \frac{1}{n}\sum_{i=1}^{n} Z_i$$

l'estimateur empirique de θ, le théorème central limite permet de dire que

$$\sqrt{n}\left(\widehat{\theta}_n - \theta\right) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \Lambda(\theta))$$

Par ailleurs nous avons

$$VPP(U_1, Y) = \frac{\Pr\{Y = 1, \phi(X, U_1) = 1\}}{\Pr\{\phi(X, U_1) = 1\}} = \frac{p_1 + p_2}{p_1 + p_2 + p_4 + p_5}$$

$$VPP(U_2, Y) = \frac{\Pr\{Y = 1, \phi(X, U_2) = 1\}}{\Pr\{\phi(X, U_2) = 1\}} = \frac{p_1}{p_1 + p_4}$$

d'où

$$\frac{VPP(U_1, Y)}{VPP(U_2, Y)} = \frac{(p_1 + p_4)(p_1 + p_2)}{p_1(p_1 + p_2 + p_4 + p_5)}$$

Soit la fonction

$$g(\theta) = log\left(\frac{VPP(U_1, Y)}{VPP(U_2, Y)}\right)$$

On a

$$\nabla g(\theta) = \begin{pmatrix} \frac{1}{p_1+p_4} + \frac{1}{p_1+p_2} - \frac{1}{p_1} - \frac{1}{p_1+p_2+p_4+p_5} \\ \frac{1}{p_1+p_2} - \frac{1}{p_1+p_2+p_4+p_5} \\ 0 \\ \frac{1}{p_1+p_4} - \frac{1}{p_1+p_2+p_4+p_5} \\ -\frac{1}{p_1+p_2+p_4+p_5} \\ 0 \end{pmatrix}$$

En utilisant la Méthode Delta Multivariée, on démontre que

$$\sqrt{n}\left(g(\widehat{\theta}_n) - g(\theta)\right) \xrightarrow{\mathcal{L}} \mathcal{N}\left(0, {}^T\nabla g(\theta)\Lambda(\theta)\nabla g(\theta)\right)$$

où

$$\begin{aligned} {}^T\nabla g(\theta)\Lambda(\theta)\nabla g(\theta) &= {}^T\nabla g(\theta) diag(\theta)\nabla g(\theta) - (\theta\nabla g(\theta))^T (\theta\nabla g(\theta)) \\ &= \sum_{i=1}^{6} p_i \nabla_i^2 - \left(\sum_{i=1}^{6} p_i \nabla_i\right)^2 \end{aligned}$$

avec

$$\nabla g(\theta) = \begin{pmatrix} \nabla_1 \\ \vdots \\ \nabla_6 \end{pmatrix}$$

Etant donné que $\sum_{i=1}^{6} p_i = 1$, alors ${}^T\nabla g(\theta)\Lambda(\theta)\nabla g(\theta) > 0$ puisque c'est une variance du vecteur $(\nabla_1, \ldots, \nabla_6)$ qui n'est pas colinéaire avec le vecteur $\mathbb{1} = (1, \ldots, 1)$. □

L'application : $\theta \longmapsto \nabla g(\theta)$ est continue de même que l'application : $\theta \longmapsto \Lambda(\theta)$. Et puisque $\widehat{\theta}_n$ converge en presque sûrement vers θ, on obtient alors

$${}^T\nabla g(\widehat{\theta}_n)\Lambda(\widehat{\theta}_n)\nabla g(\widehat{\theta}_n) \xrightarrow{p.s} {}^T\nabla g(\theta)\Lambda(\theta)\nabla g(\theta)$$

Grâce au théorème de Slutsky, on peut conclure que

$$\frac{\sqrt{n}\left(g(\widehat{\theta}_n) - g(\theta)\right)}{\sqrt{{}^T\nabla g(\widehat{\theta}_n)\Lambda(\widehat{\theta}_n)\nabla g(\widehat{\theta}_n)}} \xrightarrow{\mathcal{L}} \mathcal{N}(0,1)$$

Sous l'hypothèse que $\dfrac{VPP(U_1, Y)}{VPP(U_2, Y)} = 1$, si la taille de l'échantillon est suffisamment grande alors

$$\frac{\sqrt{n}\left(g(\widehat{\theta}_n)\right)}{\sqrt{{}^T\nabla g(\widehat{\theta}_n)\Lambda(\widehat{\theta}_n)\nabla g(\widehat{\theta}_n)}} \xrightarrow{\mathcal{L}} \mathcal{N}(0,1)$$

Ce qui nous permet de construire une stratégie de sélection du profil le plus adéquat. Si on note par $q_{1-\alpha/2}$ le quantile d'ordre $1-\alpha/2$ de la loi normale centrée réduite, on peut effectuer les tests suivants.

1. **Test 1** :

(a) Sélectionner le profil U_1 si

$$g(\widehat{\theta}_n) \geq q_{1-\alpha/2}\sqrt{\frac{{}^T\nabla g(\widehat{\theta}_n)\Lambda(\widehat{\theta}_n)\nabla g(\widehat{\theta}_n)}{n}}$$

(b) Sélectionner le profil U_2 si

$$g(\widehat{\theta}_n) \leq -q_{1-\alpha/2}\sqrt{\frac{{}^T\nabla g(\widehat{\theta}_n)\Lambda(\widehat{\theta}_n)\nabla g(\widehat{\theta}_n)}{n}}$$

(c) Choisir au hasard entre U_1 et U_2 si

$$g(\widehat{\theta}_n) \in \left]\ -q_{1-\alpha/2}\sqrt{\frac{{}^T\nabla g(\widehat{\theta}_n)\Lambda(\widehat{\theta}_n)\nabla g(\widehat{\theta}_n)}{n}}, q_{1-\alpha/2}\sqrt{\frac{{}^T\nabla g(\widehat{\theta}_n)\Lambda(\widehat{\theta}_n)\nabla g(\widehat{\theta}_n)}{n}}\ \right[$$

Cette troisième étape du test utilise le principe du test stochastique (test randomisé) où on génère une réalisation b d'une variable de Bernoulli de paramètre 1/2. On sélectionne U_1 si $b = 1$ sinon on sélectionne U_2.

2. **Test 2** :

(a) Sélectionner le profil U_2 si

$$g(\widehat{\theta}_n) < -q_{1-\alpha/2}\sqrt{\frac{{}^T\nabla g(\widehat{\theta}_n)\Lambda(\widehat{\theta}_n)\nabla g(\widehat{\theta}_n)}{n}}$$

(b) sinon Sélectionner le profil U_1

Le Test 2 permet de favoriser les profils les plus courts. Les résultats présentés dans cette analyse sont obtenus en utilisant le Test 2.

6.1.2 Algorithme

A partir d'un ensemble de validation, nous cherchons à réduire l'ensemble $\mathcal{U}_\lambda^1$ en utilisant la valeur prédictive positive comme paramètre de comparaison. Les indicateurs de performance tels que les rapports de vraisemblance positifs (RVP) ou les rapports de vraisemblance négatifs (RVN) peuvent également être utilisés.

Algorithme : Réduction de l'ensemble $\mathcal{U}_\lambda^1$

– Entrées : $\mathcal{D}$ un ensemble d'observation ; $\mathcal{U}_\lambda^1$ un ensemble de règles non redondantes
– Sorties : $\mathcal{U}_\lambda^2$ un ensemble optimal de profils

1 : **pour** tout profil $C \in \mathcal{U}_\lambda^1$ **faire**
2 : $S = is.subset(C, \mathcal{U}_\lambda^1)$ {le sous-ensemble des profils emboîtés dans C}
3 : **pour** tout profil $C' \in S$ **faire**
4 : Evaluer les indicateurs suivants
5 : $\widehat{\theta}_n = (p_1, \ldots, p_6|\mathcal{D})$
6 : $g(\widehat{\theta}_n) = log(VPP(C, Y|\widehat{\theta}_n)) - log(VPP(C', Y|\widehat{\theta}_n))$
7 : $\Lambda(\widehat{\theta}_n) = diag(\widehat{\theta}_n) - \widehat{\theta}_n^t \widehat{\theta}_n$
8 : $\nabla_n = \nabla g(\widehat{\theta}_n)$
9 : **fin pour**
10 : **Si** il existe $C' \in S$ tel que $g(\widehat{\theta}_n) < -q_{1-\alpha/2}\sqrt{\frac{\nabla_n^t \Lambda(\widehat{\theta}_n)\nabla_n}{n}}$ **faire**
11 : $\mathcal{U}_\lambda^2 = delete(C, \mathcal{U}_\lambda^1)$ {Supprimer le profil C}
12 : **Sinon**
$\mathcal{U}_\lambda^2 = delete(S, \mathcal{U}_\lambda^1)$ {Supprimer le sous-ensemble S}
13 : **fin si**
14 : **fin pour**
15 : **Résultat** $\mathcal{U}_\lambda^2$

Tableau III.3 – Algorithme de réduction de l'ensemble non redondant

Le processus d'apprentissage, tel qu'il a été décrit jusqu'ici requiert une grande base de données qu'il faudra échantillonner en trois sous-ensembles (apprentissage, validation et test) de tailles suffisamment grandes. Habituellement dans la tâche de l'apprentissage automatique, il est courant que le nombre d'observations disponibles ne permettent pas une subdivision des données en trois échantillons, un pour l'apprentissage, un pour la validation et un pour le test. Le recours à l'échantillon de validation permet d'évaluer les paramètres de performance sur un échantillon différent mais issu de la même distribution que l'échantillon d'apprentissage. On peut envisager alors une procédure bootstrap.

6.2 Lorsque les données sont de taille petite

Lorsqu'on ne dispose pas de données suffisantes pour une subdivision en trois sous-ensembles : apprentissage, validation et test, on peut recourir à une procédure de bootstrap pour la validation du

classifieur. En effet lorsque n,la taille de l'échantillon, est petite, la condition

$$S = \frac{\sqrt{n}\left(g(\widehat{\theta}_n) - g(\theta)\right)}{\sqrt{{}^T\nabla g(\widehat{\theta}_n)\Lambda(\widehat{\theta}_n)\nabla g(\widehat{\theta}_n)}} \xrightarrow{\mathcal{L}} \mathcal{N}(0,1)$$

n'est plus assurée. D'où la nécessité de recourir à un test d'hypothèse bootstrap.

6.2.1 Test d'hypothèse boostrap pour la sélection d'un ensemble optimal de profils

Le bootstrap est une technique de ré-échantillonnage bien connue dans la littérature [9, 10]. Le principe fondamental du bootstrap est de substituer à la distribution inconnue F, dont est issu l'échantillon d'apprentissage, la distribution empirique F_n qui donne un poids $1/n$ à chaque réalisation. Ainsi on obtient un échantillon de taille n dit échantillon bootstrap selon la distribution empirique F_n par n tirages aléatoires avec remise parmi les n observations initiales.

La statistique d'intérêt S a une distribution d'échantillonnage notée F_S. Cette distribution dépend de la distribution G_Z de la variable aléatoire Z dont les valeurs observées sont $z_1, \ldots, z_n$. On écrit $F_S(s, G_Z)$, où G_Z est la distribution de Bernoulli généralisée de la variable Z. La distribution G_Z, quant à elle, dépend de la distribution F_X de la variable aléatoire X dont les observations sont $x_1, \ldots, x_n$. On note $G_Z(z, F_X)$. En résumé, la distribution F_S dépend de la réalisation z de la variable Z et de la distribution F_X de la variable X. On écrit $F_S(s, z, F_X)$.
Puisque F_X est inconnue, on travaille avec une estimation de F_X que l'on note $\widehat{F}_X$ et qui est la distribution empirique F_n des données $\{x_1, \ldots, x_n\}$. Le fait de remplacer F_X par F_n va donner une distribution d'échantillonnage F_S également modifiée. On écrit $F_S(s, z, F_n)$ au lieu de $F_S(s, z, F_x)$. Remplacer F_X par F_n et générer un échantillon de taille n selon la distribution F_n revient de même que de tirer avec remise n éléments de l'ensemble de données originales $\{x_1, \ldots, x_n\}$.
On a $g(\widehat{\theta}_n)$ un estimateur de la quantité $g(\theta)$ et $\widehat{\sigma}_n = \sqrt{\frac{1}{n}\left({}^T\nabla g(\widehat{\theta}_n)\Lambda(\widehat{\theta}_n)\nabla g(\widehat{\theta}_n)\right)}$ un estimateur de l'écart type de $g(\widehat{\theta}_n) - g(\theta)$. On note par $g(\widehat{\theta}^*_n)$ une estimation de $g(\theta)$ et $\widehat{\sigma}^*_n$ une estimation de l'écart type de $g(\widehat{\theta}^*_n) - g(\widehat{\theta}_n)$ toutes deux calculées à partir d'un échantillon bootstrap. En particulier $\widehat{\sigma}^*_n$ est l'estimation empirique bootstrap de l'écart type de $g(\widehat{\theta}^*_n) - g(\widehat{\theta}_n)$. Alors la distribution bootstrap de $\left(g(\widehat{\theta}^*_n) - g(\widehat{\theta}_n)\right)/\widehat{\sigma}^*_n$ estime la distribution bootstrap de $\left(g(\widehat{\theta}_n) - g(\theta)\right)/\widehat{\sigma}_n$ sous l'hypothèse nulle [16]. Baser le test d'hypothèse sur la distribution bootstrap de $\left(g(\widehat{\theta}^*_n) - g(\widehat{\theta}_n)\right)/\widehat{\sigma}^*_n$ permet d'améliorer la précision du niveau du test sans modifier la puissance du test [4, 16].
Pour appliquer le test bilatéral bootstrap de $H_0 : g(\theta) = 0$ au niveau α, on effectue les instructions suivantes : commence par

1. Calculer la valeur de la statistique S pour l'échantillon de départ : soit s_0 la valeur observée.
2. Simuler B échantillons de taille n observations tirées de façon aléatoire avec remise à partir de

l'ensemble de données originales, et obtenir ainsi B valeurs simulées de s_b^* de S :

$$s_b^* = \frac{g(\widehat{\theta}_n^b) - g(\widehat{\theta}_n)}{\widehat{\sigma}_n^b}, \qquad b = 1, \ldots, B$$

3. Calculer la $p - value\ bootstrap$

$$p^* = \frac{1}{B} \sum_{b=1}^{B} I(s_b^* > s_0)$$

On peut formuler alors la règle de décision suivante :

1. **Test 1.**

   (a) Sélectionner le profil U_2 si $p^* < \alpha/2$

   (b) Sélectionner le profil U_1 si $p^* > 1 - \alpha/2$

   (c) Choisir au hasard entre U_1 et U_2 si $p^* \in [\,\alpha/2,\, 1 - \alpha/2\,]$
   Cette troisième étape du test utilise le principe du test stochastique (test randomisé) où on génère une réalisation b d'une variable de Bernoulli de paramètre 1/2. On sélectionne U_1 si $b = 1$ sinon on sélectionne U_2.

2. **Test 2** :

   (a) Sélectionner le profil U_2 si $p^* < \alpha/2$

   (b) sinon Sélectionner le profil U_1

Le Test 2 permet de favoriser les profils les plus courts. Les résultats présentés dans cette analyse sont obtenus en utilisant le Test 2.

6.2.2 Algorithme

L'algorithme d'apprentissage statistique, adapté au bootstrap, est le suivant :

Algorithme : Réduction de l'ensemble $\mathcal{U}_\lambda^1$

– Entrées : $\mathcal{D}$ un ensemble d'observation ; $\mathcal{U}_\lambda^1$ un ensemble de règles non redondantes, $\alpha = 0.05$ le niveau du test et B le nombre d'échantillon bootstrap (20 par défaut).
– Sorties : $\mathcal{U}_\lambda^2$ un ensemble optimal de profils

pour tout profil $C \in \mathcal{U}_\lambda^1$ **faire**
  $S = is.supset(C, \mathcal{U}_\lambda^1)$ {le sous-ensemble des profils emboîtés dans C}
  pour tout profil $C' \in S$ **faire**
    Evaluer les indicateurs suivants
    $\widehat{\theta}_n = (p_1, \ldots, p_6 | \mathcal{D})$
    $g(\widehat{\theta}_n) = log(VPP(C, Y|\widehat{\theta}_n)) - log(VPP(C', Y|\widehat{\theta}_n))$
    $\Lambda(\widehat{\theta}_n) = diag(\widehat{\theta}_n) - \widehat{\theta}_n^t \widehat{\theta}_n$
    $\nabla_n = \nabla g(\widehat{\theta}_n)$
    $\widehat{\sigma}_n = \sqrt{\frac{1}{n}\left(\nabla_n^t \Lambda(\widehat{\theta}_n) \nabla_n\right)}$
    $s_0 = g(\widehat{\theta}_n)/\widehat{\sigma}_n$
    pour tout échantillon boostrap $\mathcal{D}^b$ **faire**
      $\widehat{\theta}_n^b = (p_1, \ldots, p_6 | \mathcal{D}^b)$
      $g(\widehat{\theta}_n^b) = log(VPP(C, Y|\widehat{\theta}_n^b)) - log(VPP(C', Y|\widehat{\theta}_n^b))$
      $\Lambda(\widehat{\theta}_n^b) = diag(\widehat{\theta}_n^b) - (\widehat{\theta}_n^b)^t \widehat{\theta}_n^b$
      $\nabla_n = \nabla\left(g(\widehat{\theta}_n^b) - g(\widehat{\theta}_n)\right)$
      $\widehat{\sigma}_n^b = \sqrt{\frac{1}{n}\left(\nabla_n^t \Lambda(\widehat{\theta}_n^b) \nabla_n\right)}$
      $s_b^* = \left(g(\widehat{\theta}_n^b) - g(\widehat{\theta}_n)\right)/\widehat{\sigma}_n^b$
    fin pour
    Calculer la $p - value$

$$p^* = \frac{1}{B}\sum_{b=1}^{B} I(s_b^* > s_0)$$

    si $p^* < \alpha/2$ **faire**
      $\mathcal{U}_\lambda^2 = delete(C, \mathcal{U}_\lambda^1)$ {Supprimer le profil C}
    sinon
      $\mathcal{U}_\lambda^2 = delete(C', \mathcal{U}_\lambda^1)$ {Supprimer le profil C'}
    fin si
  fin pour
fin pour
Résultat $\mathcal{U}_\lambda^2$

Tableau III.4 – Algorithme de réduction de l'ensemble non redondant lorsque l'échantillon d'apprentissage est de petite taille

7 Application à des données de la littérature

Toutes les données que nous avons utilisé pour l'application de l'algorithme d'apprentissage sont issues du répertoire d'apprentissage automatique UCI (UCI Machine Leraning Repository) [3]. Toutes les analyses relatives à la méthode de classement proposée ont été réalisées dans l'environnement de programmation R [25]. L'exploration des règles d'association a été faite en utilisant le package arules [1]. Nous avons également utilisé le package rpart [28], la package partykit [18], le package $e1071$ [22] et le package DMwR [29] pour comparer notre approche avec celles existantes dans la littérature.

7.1 Données Adult Data Set

Les données d'application sont extraites de la base de données du bureau de recensement de 1994 [19]. Elles contiennent essentiellement des sujets âgés de plus de 16 ans et ayant à la fois un revenu brut ajusté supérieur à 1 et un volume horaire de travail positif. Au total, elles contiennent 45222 sujets hormis les données manquantes. Les sujets sont échantillonnés sur deux ensembles : un ensemble d'apprentissage de 30162 sujets (2/3 de données totales) et un ensemble test de 15060 sujets. Les données contiennent 14 covariables dont 5 sont continues et 8 sont nominales dont une variable réponse binaire indexant le revenu annuel d'un sujet à plus de $ 50K ou moins. L'objectif visé dans cette analyse est de trouver un profil prédictif du niveau de revenu d'un sujet donné.
Pour évaluer la procédure d'apprentissage des règles d'association binaire, nous allons effectuer plusieurs expériences en sur-échantillonnant ou en sous-échantillonnant le jeu de données census. Pour obtenir un ensemble de données déséquilibrées, on commence par sélectionner toutes les observations de la classe prévalente ; ensuite on se fixe une proportion α de la classe rare. Soit n le nombre d'observations de la classe prévalente. On sélectionne $n' = n\alpha/(1-\alpha)$ observations de la classe rare. On obtient ainsi, un échantillon de $n + n'$ observations avec une proportion α de la classe rare.
Dans tout ce qui suit, nous avons fixé le paramètre de la taille maximale des règles à 4, le paramètre du risque relatif minimal égal à 1 et le paramètre de la p-value minimale associée au test exact de Fisher égale à 0.05. Après avoir construit notre échantillon déséquilibré, on se fixe un seuil de support minimale (minsup) et un seuil de valeur prédictive positive minimale (minconf). Ces derniers nous permettront de générer l'ensemble de règles d'association fréquentes $\mathcal{R}$. Pour chaque expérience, on subdivise aléatoirement l'échantillon en deux parties : apprentissage et validation. Un ensemble test est utilisé pour évaluer les performances du classifieur. Cependant, on peut évaluer deux types d'erreurs de classement : l'erreur de classement lorsque la distribution de l'ensemble d'apprentissage est différente de la distribution de l'ensemble test et l'erreur de classement lorsque la distribution de l'ensemble d'apprentissage est identique à la distribution de l'ensemble test.

7.1.1 Performances du classifieur lorsque la distribution de l'échantillon test est identique à celui de l'échantillon d'apprentissage

Proportions	Nb profils dans $\mathcal{U}_\lambda$	Erreur.cl $\mathcal{U}_\lambda$	Nb profils dans $\mathcal{U}_\lambda^2$	Sensibilité	Spécificité	Erreur.clt	Minsup Minconf
	76	0.22	12	0.68	0.81	0.19	
	129	0.28	10	0.69	0.78	0.22	
	110	0.25	15	0.70	0.78	0.23	
	69	0.19	14	0.60	0.83	0.17	
	92	0.24	12	0.72	0.80	0.20	
<=50K >50K	101	0.27	16	0.71	0.81	0.19	0.001 0.028
0.993 0.007	130	0.32	12	0.80	0.77	0.23	
	145	0.30	11	0.74	0.81	0.19	
	126	0.35	17	0.74	0.74	0.26	
	101	0.24	06	0.62	0.83	0.17	
	110	0.22	13	0.74	0.81	0.19	
	104	0.23	11	0.60	0.81	0.19	
	61	0.19	10	0.67	0.83	0.17	
	62	0.19	10	0.67	0.85	0.16	
	69	0.21	11	0.72	0.82	0.18	
	34	0.08	04	0.49	0.93	0.08	
	91	0.23	09	0.71	0.83	0.17	
<=50K >50K	81	0.21	09	0.61	0.85	0.15	0.002 0.06
0.985 0.015	70	0.19	10	0.71	0.83	0.17	
	59	0.22	15	0.80	0.78	0.22	
	67	0.21	08	0.72	0.84	0.16	
	91	0.24	11	0.70	0.80	0.20	
	69	0.21	09	0.72	0.83	0.18	
	92	0.23	07	0.60	0.89	0.12	

Tableau III.5 – Performance prédictive sur 12 expériences : (0.7% & 1.5%)

Proportions	Nb profils dans $\mathcal{U}_\lambda$	Erreur.cl $\mathcal{U}_\lambda$	Nb profils dans $\mathcal{U}_\lambda^2$	Sensibilité	Spécificité	Erreur.clt	Minsup Minconf
	56	0.23	22	0.79	0.77	0.23	
	64	0.25	19	0.77	0.79	0.21	
	43	0.19	15	0.68	0.84	0.17	
	55	0.26	09	0.68	0.83	0.17	
	35	0.19	06	0.48	0.92	0.10	
<=50K >50K	44	0.20	10	0.67	0.86	0.14	0.005 0.10
0.97 0.03	35	0.22	09	0.70	0.83	0.17	
	66	0.25	16	0.71	0.81	0.20	
	51	0.20	11	0.75	0.83	0.18	
	59	0.24	11	0.68	0.81	0.19	
	58	0.24	16	0.80	0.77	0.23	
	50	0.26	13	0.81	0.77	0.23	
	67	0.20	21	0.76	0.80	0.20	
	83	0.22	24	0.77	0.78	0.22	
	69	0.20	15	0.73	0.83	0.18	
	74	0.16	20	0.66	0.86	0.16	
	73	0.20	14	0.70	0.83	0.18	
<=50K >50K	50	0.20	14	0.71	0.83	0.17	0.01 0.23
0.93 0.07	50	0.16	16	0.63	0.88	0.14	
	63	0.18	20	0.67	0.83	0.18	
	50	0.16	19	0.64	0.86	0.16	
	55	0.20	16	0.72	0.83	0.17	
	67	0.20	17	0.73	0.83	0.18	
	75	0.18	18	0.67	0.83	0.18	

Tableau III.6 – Performance prédictive sur 12 expériences : (3% & 7%)

Proportions	Nb profils dans $\mathcal{U}_\lambda$	Erreur.cl $\mathcal{U}_\lambda$	Nb profils dans $\mathcal{U}_\lambda^2$	Sensibilité	Spécificité	Erreur.clt	Minsup Minconf
	62	0.23	19	0.67	0.86	0.17	
	56	0.22	20	0.78	0.78	0.22	
	62	0.23	17	0.63	0.83	0.20	
	60	0.22	18	0.68	0.83	0.19	
	49	0.23	22	0.75	0.78	0.23	
<=50K >50K	40	0.20	10	0.58	0.88	0.16	0.025 0.4
0.85 0.15	54	0.23	21	0.70	0.83	0.19	
	59	0.23	18	0.61	0.86	0.18	
	33	0.19	13	0.64	0.86	0.18	
	44	0.21	18	0.73	0.80	0.21	
	65	0.23	20	0.67	0.86	0.17	
	46	0.23	11	0.64	0.86	0.18	
	58	0.20	17	0.65	0.88	0.16	
	66	0.20	22	0.68	0.83	0.20	
	62	0.20	21	0.67	0.86	0.18	
	66	0.20	21	0.68	0.83	0.20	
	46	0.18	18	0.61	0.88	0.18	
<=50K >50K	64	0.20	23	0.65	0.88	0.16	0.03 0.5
0.80 0.20	53	0.18	18	0.65	0.88	0.17	
	75	0.22	19	0.68	0.83	0.20	
	57	0.18	19	0.65	0.88	0.16	
	49	0.19	19	0.68	0.84	0.19	
	58	0.18	20	0.67	0.86	0.18	
	67	0.22	20	0.74	0.81	0.20	

Tableau III.7 – Performance prédictive sur 12 expériences : (15% & 20%)

Avec un diagramme-boîtes en parallèle, nous avons représenté, pour chaque série de 100 valeurs des différentes mesures de performances (sensibilité, spécificité et erreur de classement), la distribution de celles-ci de manière très simplifiée avec la médiane (trait épais), une boîte qui s'étend du quartile 0.25 au quartile 0.75, et des moustaches qui s'étendent par défaut jusqu'à la valeur distante d'au maximum 1.5 fois la distance inter-quartile.

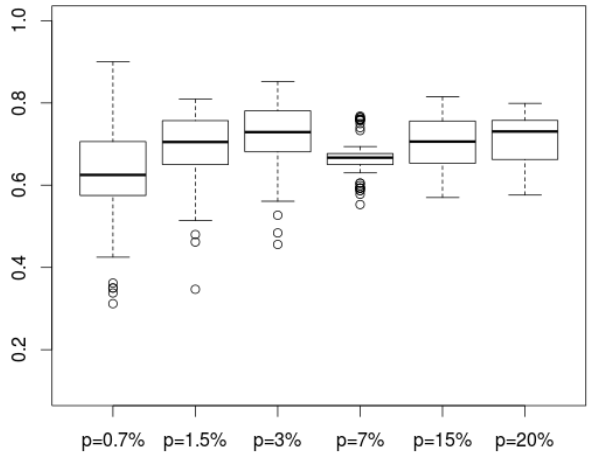


Figure III.1 – Distribution de la sensibilité estimée sur 100 échantillons

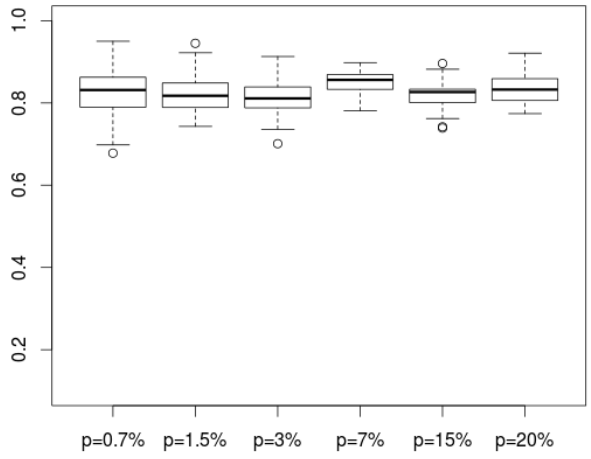


Figure III.2 – Distribution de la spécificité estimée sur 100 échantillons

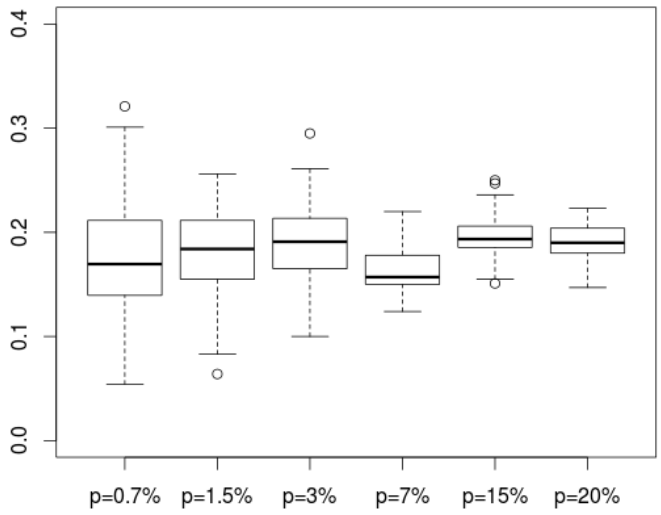


Figure III.3 – Distribution de l'erreur de classement estimée sur 100 échantillons

7.1.2 Performances du classifieur lorsque la distribution de l'échantillon test est différente de celui de l'échantillon d'apprentissage

Proportions	Nb profils dans $\mathcal{U}_\lambda$	Erreur.cl $\mathcal{U}_\lambda$	Nb profils dans $\mathcal{U}_\lambda^2$	Sensibilité	Spécificité	Erreur.clt	Minsup	Minconf
	80	0.24	12	0.71	0.77	0.24		
	101	0.26	14	0.78	0.73	0.25		
	80	0.23	11	0.74	0.78	0.23		
	47	0.20	13	0.70	0.82	0.20		
	93	0.24	07	0.57	0.83	0.23		
<=50K >50K	46	0.21	14	0.73	0.83	0.20	0.001	0.028
0.993 0.007	113	0.27	14	0.72	0.73	0.27		
	71	0.21	12	0.55	0.83	0.23		
	94	0.23	13	0.74	0.79	0.22		
	51	0.20	08	0.54	0.85	0.22		
	102	0.23	04	0.46	0.91	0.19		
	53	0.19	06	0.49	0.90	0.20		
	53	0.19	08	0.58	0.89	0.18		
	67	0.20	16	0.66	0.83	0.21		
	40	0.19	10	0.64	0.84	0.21		
	59	0.18	16	0.66	0.86	0.18		
	100	0.23	16	0.74	0.77	0.23		
<=50K >50K	53	0.19	13	0.69	0.84	0.19	0.002	0.06
0.985 0.015	46	0.17	08	0.57	0.93	0.16		
	64	0.18	14	0.65	0.86	0.18		
	73	0.19	12	0.59	0.85	0.21		
	60	0.18	11	0.61	0.90	0.17		
	74	0.18	14	0.65	0.89	0.17		
	97	0.21	13	0.74	0.83	0.19		

Tableau III.8 – Performance prédictive sur 12 expériences : (0.7% & 1.5%)

Proportions	Nb profils dans $\mathcal{U}_\lambda$	Erreur.cl $\mathcal{U}_\lambda$	Nb profils dans $\mathcal{U}_\lambda^2$	Sensibilité	Spécificité	Erreur.clt	Minsup Minconf
	74	0.26	20	0.66	0.81	0.23	
	75	0.24	18	0.82	0.75	0.24	
	63	0.22	16	0.76	0.81	0.20	
	68	0.24	12	0.67	0.79	0.24	
	61	0.21	11	0.58	0.88	0.19	
<=50K >50K	61	0.22	16	0.77	0.80	0.21	0.005 0.1
0.97 0.03	51	0.21	10	0.70	0.82	0.21	
	74	0.27	20	0.75	0.76	0.25	
	51	0.21	11	0.70	0.82	0.21	
	85	0.25	14	0.72	0.80	0.22	
	62	0.24	15	0.77	0.77	0.23	
	71	0.24	21	0.80	0.75	0.24	
	73	0.20	25	0.75	0.81	0.20	
	71	0.22	23	0.74	0.83	0.19	
	76	0.20	24	0.75	0.81	0.20	
	85	0.22	24	0.75	0.81	0.20	
	75	0.20	20	0.66	0.86	0.18	
<=50K >50K	73	0.20	20	0.74	0.83	0.19	0.01 0.23
0.93 0.07	71	0.18	19	0.66	0.86	0.18	
	71	0.21	23	0.76	0.80	0.21	
	71	0.18	22	0.66	0.86	0.18	
	67	0.20	20	0.73	0.84	0.18	
	76	0.20	20	0.66	0.86	0.18	
	79	0.22	18	0.68	0.83	0.21	

Tableau III.9 – Performance prédictive sur 12 expériences : (3% & 7%)

Proportions	Nb profils dans $\mathcal{U}_\lambda$	Erreur.cl $\mathcal{U}_\lambda$	Nb profils dans $\mathcal{U}_\lambda^2$	Sensibilité	Spécificité	Erreur.clt	Minsup Minconf
	56	0.20	16	0.66	0.86	0.18	
	65	0.22	19	0.76	0.81	0.20	
	62	0.22	17	0.66	0.86	0.18	
	59	0.22	14	0.65	0.89	0.17	
	55	0.22	17	0.66	0.86	0.18	
<=50K >50K	52	0.21	21	0.70	0.81	0.21	0.025 0.4
0.85 0.15	60	0.22	21	0.74	0.81	0.21	
	56	0.22	22	0.75	0.81	0.20	
	52	0.23	15	0.60	0.86	0.20	
	58	0.22	20	0.77	0.79	0.22	
	64	0.22	18	0.66	0.86	0.18	
	56	0.22	17	0.75	0.81	0.20	
	62	0.21	20	0.72	0.84	0.18	
	65	0.21	21	0.76	0.81	0.20	
	59	0.21	21	0.74	0.82	0.20	
	75	0.22	22	0.74	0.83	0.19	
	54	0.18	17	0.65	0.89	0.17	
<=50K >50K	62	0.20	22	0.66	0.87	0.18	0.03 0.5
0.80 0.20	54	0.20	20	0.68	0.84	0.20	
	58	0.20	22	0.75	0.81	0.20	
	54	0.18	19	0.65	0.89	0.17	
	46	0.18	18	0.64	0.89	0.17	
	56	0.20	19	0.74	0.82	0.20	
	70	0.22	25	0.74	0.83	0.19	

Tableau III.10 – Performance prédictive sur 12 expériences : (15% & 20%)

Avec un diagramme-boîtes en parallèle, nous avons représenté, pour chaque série de 100 valeurs des différentes mesures de performances (sensibilité, spécificité et erreur de classement), la distribution de celles-ci de manière très simplifiée avec la médiane (trait épais), une boîte qui s'étend du quartile 0.25 au quartile 0.75, et des moustaches qui s'étendent par défaut jusqu'à la valeur distante d'au maximum 1.5 fois la distance inter-quartile.

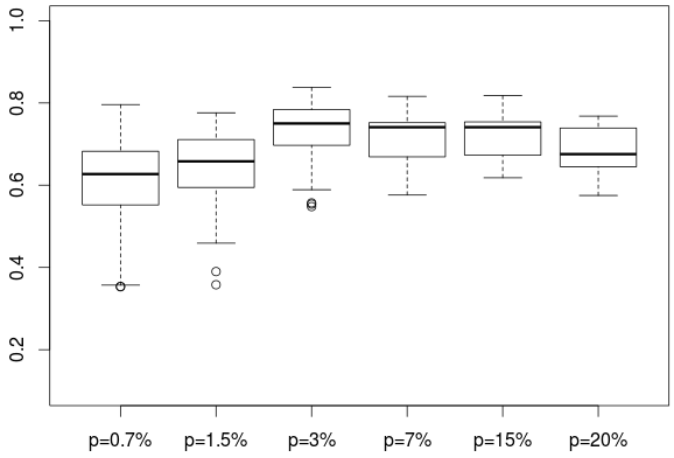


Figure III.4 – Distribution de la sensibilité estimée sur 100 échantillons

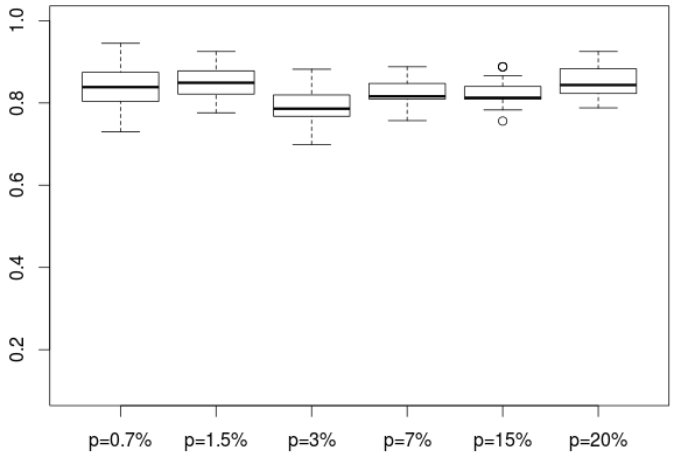


Figure III.5 – Distribution de la spécificité estimée sur 100 échantillons

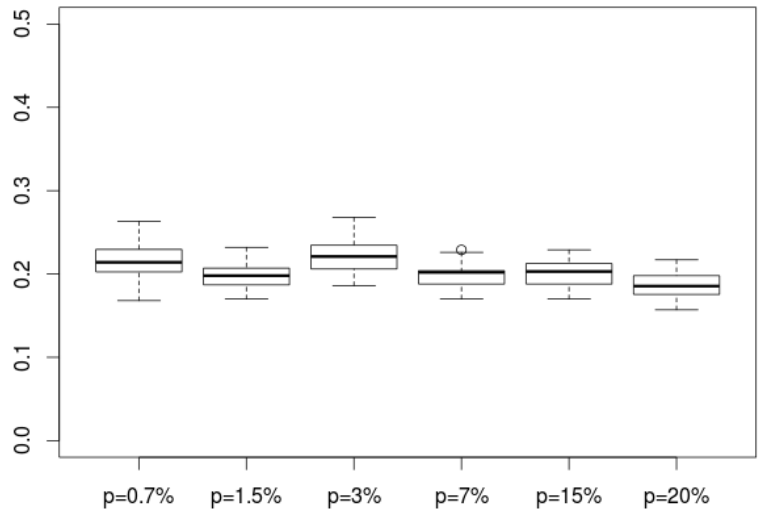


Figure III.6 – Distribution de l'erreur de classement estimée sur 100 échantillons

7.2 Comparaison de la méthode d'apprentissage avec des méthodes alternatives

Le classement binaire basé sur la régression logistique ou les arbres binaires de régression implique l'ajustement d'un modèle paramétrique ou non paramétrique aux probabilités conditionnelles $\Pr(Y = y|X = x)$ où $y \in \mathcal{D}om(Y)$ et $x \in \mathcal{D}om(X)$. Notons par $\Pr(Y = y|X = x, \mathcal{D})$ la probabilité ajustée aux données $\mathcal{D}$ et considérée comme un score. Dans ces cas, le classifieur ϕ est alors défini par la donnée d'un seuil $\lambda \in]0, 1[$ par

$$\phi(x|\lambda) = \begin{cases} 1 & \text{si } \Pr(Y = y|X = x, \mathcal{D}) > \lambda \\ 0 & \text{sinon} \end{cases}$$

Dans le cas de l'analyse discriminante ou des réseaux bayésiens comme le réseau bayésien naïf on considère une loi a priori π pour la distribution de probabilité des classes et on ajuste un modèle paramétrique ou non paramétrique aux lois conditionnelles de X sachant que $Y = y$. Notons par $\Pr(X = x|Y = y)$ la densité conditionnelle de X sachant $Y = y$ selon que X est discrète ou non. Le classifieur est obtenu à partir de la loi a posteriori de Y sachant que $X = x$ qui est définie par $\dfrac{\Pr(x|Y = y, \mathcal{D})\pi(y)}{\Pr(x|\mathcal{D})}$ considérée comme un score où $\Pr(x|Y = y, \mathcal{D})$ est la loi ajustée en utilisant les données $\mathcal{D}$ et $\Pr(x|\mathcal{D})$ est la loi marginale de X correspondant au couple $(\Pr(x|y, \mathcal{D}), \pi(y))$. Ce classifieur est alors défini, pour $\lambda > 0$ fixé, par

$$\phi(x|\lambda) = \begin{cases} 1 & \text{si } \Pr(Y = y|X = x, \mathcal{D})\pi(y) > \lambda \\ 0 & \text{sinon} \end{cases}$$

Il se pose alors la question de sélectionner un classifieur optimal sur la base d'un compromis sur des mesures de performance comme la sensibilité , la spécificité, le taux d'erreur, etc. La courbe ROC et la mesure AUC sont généralement utilisées pour réaliser cet objectif. Cette démarche peut être étendue aux méthodes d'agrégation de classifieur comme le boosting d'arbre binaire de classement ou le random forest. Généralement ces méthodes utilisent un seuil $\lambda = 0.5$ par défaut. Très souvent le classifieur $\phi(x|\lambda)$ associé au seuil $\lambda = 0.5$ ne fournit pas de meilleurs performances. Ainsi pour comparer notre méthode de classement à ces différentes méthodes, nous considérons la stratégie suivante :

1. Nous identifions le seuil optimal pour chaque méthode associant un score à une observation. C'est à dire le seuil qui produit le classifieur dont les mesures de performance fournit le meilleur compromis.
2. Nous comparons alors les classifieurs ainsi obtenus à notre classifieur. Les résultats obtenus sont présentés dans les tableaux ci-dessous.

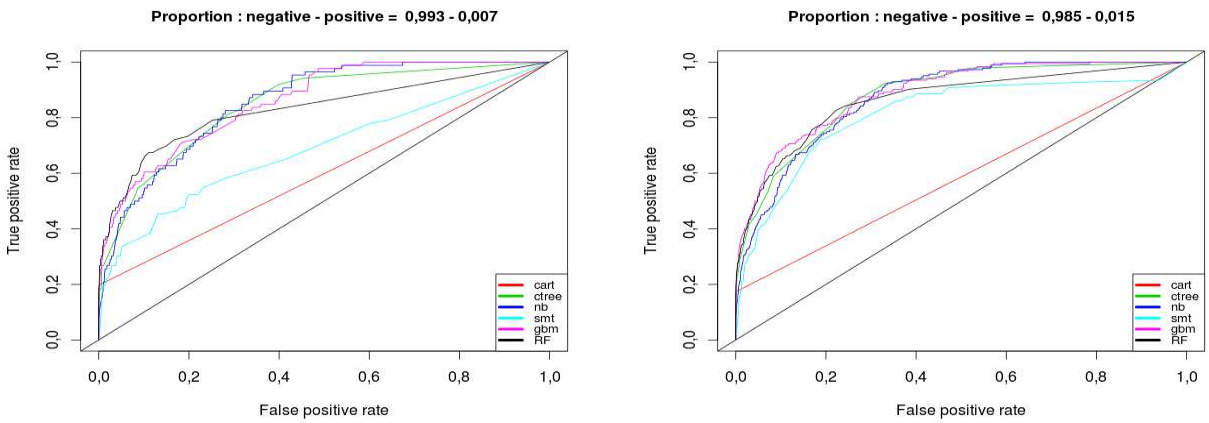
Les résultats présentés ci-dessous sont obtenus en utilisant le package **caret**[20] (**c**lassification and **re**gression **t**raining) dans l'environnement de programmation **R**. Ce dernier contient un riche ensemble de fonctions de modélisation à la fois pour la classification et la régression. Le package **caret** permet d'éliminer la différence syntaxique située entre un grand nombre d'algorithme pour la construction et la prédiction de modèles. Il contient un ensemble d'approches raisonnables semi-automatisées pour l'optimisation des valeurs des paramètres d'apprentissage. A l'aide du package **caret**, on peut donc trouver, pour la plus part des méthodes (classification ou régression), le classifieur optimal qui ajuste le mieux les données d'apprentissage grâce à sa fonction *train*. La fonction *train* est utilisée pour sélectionner les valeurs du(des) paramètre(s) d'apprentissage du modèle et/ou d'estimer les performances du modèle en utilisant une méthode d'échantillonnage. En utilisant une méthode d'échantillonnage telle que le bootstrap ou la validation croisée, un ensemble d'observations est simulé conditionnellement aux données d'apprentissage. A chaque ensemble échantillonné correspond un classifieur. Pour chaque combinaison de paramètres d'apprentissage candidats, un modèle est ajusté aux données échantillonnées et ensuite est utilisé pour la prédiction. La performance du modèle est estimée en agrégeant les prédictions du modèle sur les données échantillonnées. Ces performances estimées sont utilisées pour évaluer laquelle des combinaisons des paramètres d'apprentissage est appropriée. Pour des données de grande taille telles que les données "Adult Dataset" nous avons choisi la validation croisée comme méthode d'échantillonnage et pour les données de petite taille, par exemple les données "Credit Approval Dataset", nous avons utilisé le bootstrap comme méthode de ré-échantillonnage.

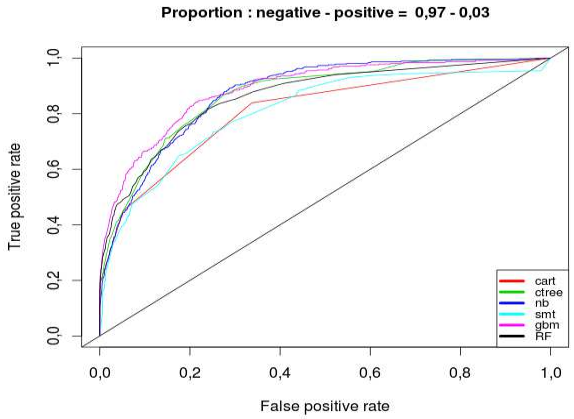
Le taux d'erreur de classement est la mesure de performance généralement associée aux algorithmes d'apprentissage automatique. Dans le contexte des ensembles de données symétriques et des ensembles de données avec des coûts de mauvais classement égaux, il est raisonnable d'utiliser le taux d'erreur comme mesure de performance. Par contre lorsque les données sont déséquilibrées ou lorsqu'elles sont associées à des coûts d'erreur inégaux, il est plus approprié d'utiliser la courbe ROC ou d'autres

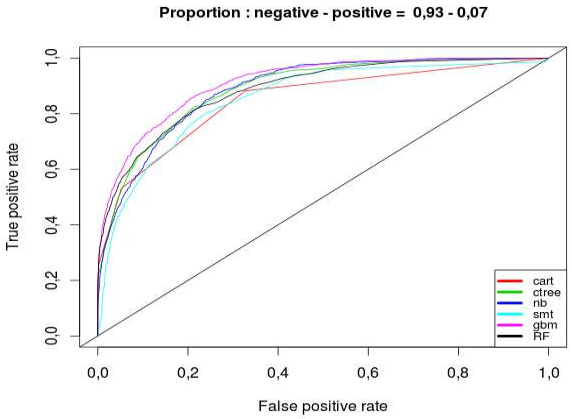
techniques similaires (Ling et Li, 1998 ; Drummond & Holte, 2000 ; Provost & Fawcett, 2001 ; Bradley, 1997 ; Turney 1996). L'aire sous la courbe ROC (AUC) est une mesure utile de la performance du classificateur car elle est indépendante du critère de décision choisi et aux changements de la distribution des classes [12]. La comparaison des AUC peut établir une relation de domination entre les classifieurs.
Le score de Pierce constitue aussi une mesure de performance conçue pour la prévision d'événements climatiques rares afin de pénaliser les modèles ne prévoyant jamais ces événements ou encore générant trop de fausses alertes. Le modèle idéal prévoit tous les événements rares sans fausse alerte. Le score de Pierce : $Sensibilité + Spécificité - 1$, compris entre -1 et 1, évalue la qualité d'un modèle de prévision. Si ce score est supérieur à 0, le taux de bonnes prévisions est supérieur à celui des fausses alertes et plus il est proche de 1, meilleur est le modèle.
Dans la suite, nous avons choisi de comparer notre méthode à des méthodes alternatives qui associent un score à chaque observation. Pour ces méthodes il est donc possible de construire leurs courbes ROC. Pour chaque méthode alternative, on peut produire un ensemble de classifieurs et puis sélectionner le classifieur le plus pertinent suivant un critère de sélection à l'aide de la fonction *train* du package **caret**. Dans cette analyse nous avons choisi la précision (taux de bien classés) comme critère de sélection. Par la suite, nous allons comparer les performances des meilleurs classifieurs sélectionnés avec les performances de notre classifeur. Les résultats sont présentés sous forme de tableaux.

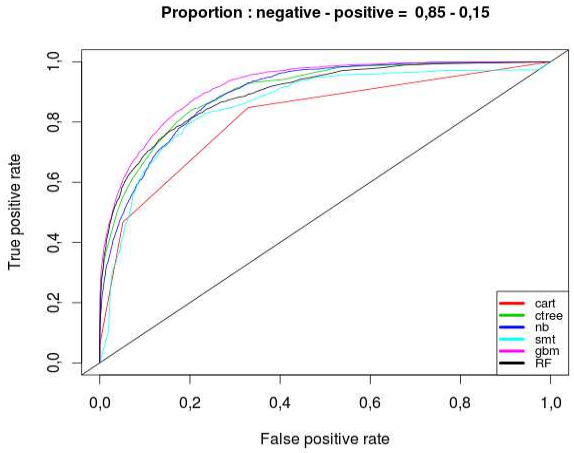
7.2.1 Données Adult Data Set

1. **Lorsque la distribution de l'échantillon test est identique à celui de l'échantillon d'apprentissage**









On peut constater à partir des graphes ci-dessus que lorsque la proportion d'observations positives devient de plus en plus grande, les courbes ROC se rapprochent de plus en plus.

Distributions "0" - "1"	ARM					CART					CTREE				
	sensib	spécf	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss
0.993 - 0.007	0,700	0,779	0,222	0,740	0,479	0,105	1,000	0,007	0,552	0,105	0,593	0,878	0,124	0,736	0,471
0.985 - 0.015	0,671	0,821	0,181	0,746	0,492	0,152	1,000	0,014	0,576	0,152	0,793	0,777	0,222	0,785	0,570
0.970 - 0.030	0,644	0,807	0,198	0,726	0,451	0,450	0,948	0,067	0,699	0,398	0,812	0,766	0,232	0,789	0,578
0.930 - 0.070	0,754	0,799	0,204	0,776	0,553	0,530	0,948	0,083	0,739	0,478	0,797	0,805	0,196	0,801	0,602
0.850 - 0.150	0,791	0,774	0,223	0,782	0,565	0,467	0,949	0,127	0,708	0,416	0,813	0,818	0,183	0,815	0,631

Distributions "0" - "1"	ARM					Naive Bayes					SMOTE				
	sensib	spécf	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss
0.993 - 0.007	0,700	0,779	0,222	0,74	0,479	0,814	0,719	0,279	0,766	0,533	0,547	0,770	0,231	0,659	0,317
0.985 - 0.015	0,671	0,821	0,181	0,746	0,492	0,799	0,761	0,238	0,780	0,560	0,696	0,821	0,181	0,758	0,517
0.970 - 0.030	0,644	0,807	0,198	0,726	0,451	0,842	0,750	0,247	0,796	0,592	0,716	0,755	0,247	0,736	0,471
0.930 - 0.070	0,754	0,799	0,204	0,776	0,553	0,850	0,760	0,233	0,805	0,610	0,783	0,772	0,227	0,778	0,555
0.850 - 0.150	0,791	0,774	0,223	0,782	0,565	0,832	0,785	0,207	0,808	0,617	0,805	0,792	0,207	0,798	0,597

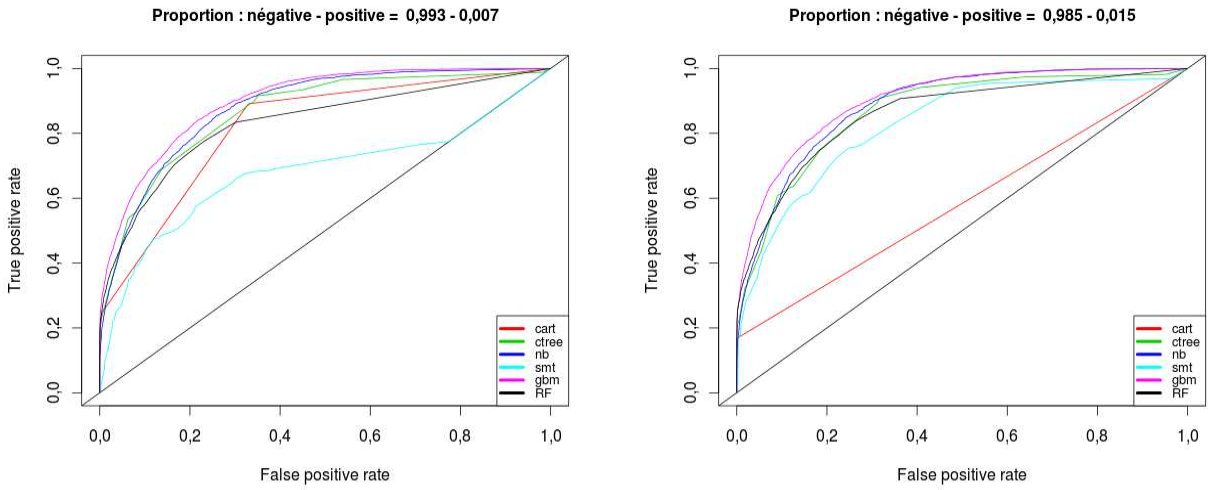
Distributionss "0" - "1"	ARM					Boosting					Random forests				
	sensib	spécf	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss
0.993 - 0.007	0,700	0,779	0,222	0,740	0,479	0,756	0,793	0,208	0,774	0,549	0,698	0,851	0,150	0,774	0,549
0.985 - 0.015	0,671	0,821	0,181	0,746	0,492	0,766	0,814	0,187	0,790	0,580	0,799	0,794	0,205	0,796	0,593
0.970 - 0.030	0,644	0,807	0,198	0,726	0,451	0,823	0,801	0,199	0,812	0,624	0,791	0,780	0,220	0,786	0,571
0.930 - 0.070	0,754	0,799	0,204	0,776	0,553	0,842	0,804	0,193	0,823	0,646	0,804	0,794	0,204	0,799	0,598
0.850 - 0.150	0,791	0,774	0,223	0,782	0,565	0,836	0,828	0,171	0,832	0,664	0,780	0,833	0,176	0,806	0,613

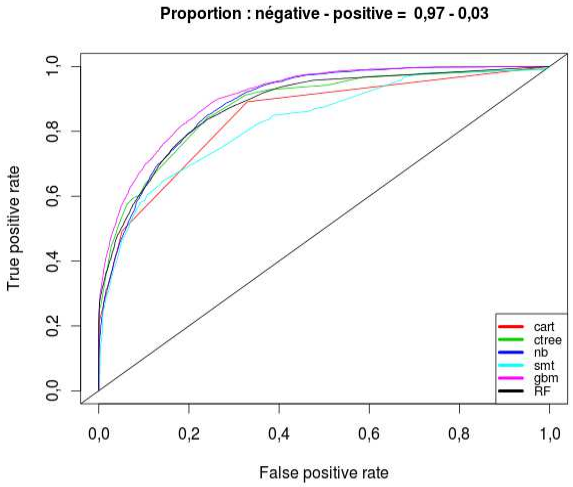
Tableau III.11 – Performances prédictives des méthodes alternatives

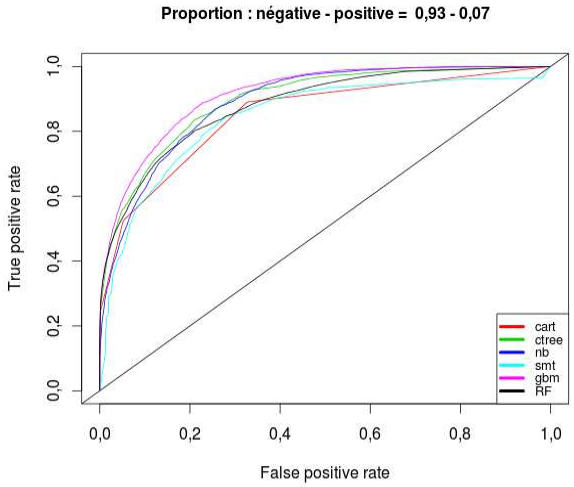
On peut constater que notre méthode d'apprentissage (ARM) est plus performante que la méthode CART. Du point de vue de l'aire en dessous de la courbe ROC (AUC) et du score de Pierce (PSS), la méthode ARM enregistre des valeurs largement au dessus des valeurs de la méthode CART. Elle produit également des sensibilités plus élevées variant entre 62% et 80% tandis que la méthode CART enregistre des sensibilités entre 10% et 50%. Par contre la méthode CART est plus spécifique (95%-100%) et admet des erreurs de classement plus faibles (7%-12%) contre (77%-81%) et (18%-22%) respectivement pour la méthode ARM.
Le classifieur naïf de Bayes, malgré qu'il produit des sensibilités, des AUC et des PSS plus élevés que ceux produits par la méthodes ARM, enregistre de forts taux d'erreurs de classement entre 21% et 28% avec des spécificités plus petites que celles de la méthodes ARM.
Les résultats présentés dans le tableau III.11 ci-dessus montrent une forte équivalence entre la méthode ARM et les méthodes SMOTE, Boosting et forêts aléatoires. Réputées d'être les meilleurs méthodes de classement en terme de performance, la méthode boosting et la méthode des forêts aléatoires présentent des performances sensiblement égales aux performances de la méthode ARM.

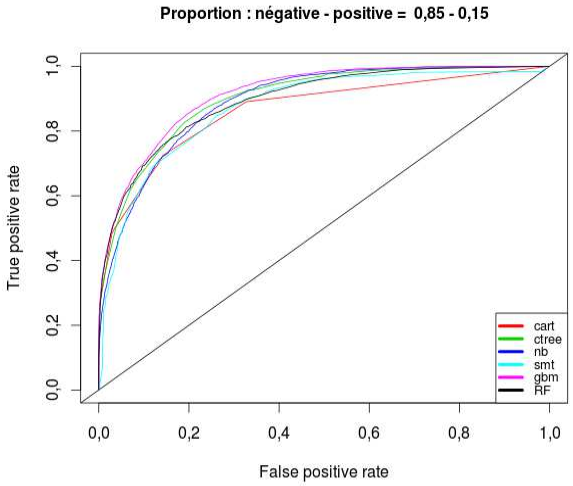
2. **Lorsque la distribution de l'échantillon test est différente de celui de l'échantillon d'apprentissage**

A ma connaissance, les performances d'un classifeur binaire sont généralement évaluées à partir d'un ensemble test dont la distribution est identique à celle de l'ensemble d'apprentissage qui a servis à construire le classifieur. Nous voulons évaluer les performances de la méthode d'apprentissage statistique et de les comparer avec les performances des méthodes alternatives lorsque la distribution de l'échantillon d'apprentissage est différente de la distribution de l'ensemble test.









De même on peut constater aussi, à partir des graphes ci-dessus, que lorsque la proportion d'observations positives devient de plus en plus grande, les courbes ROC se rapprochent de plus en plus.

Distributions "0" - "1"	ARM					CART					CTREE				
	sensib	spécf	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss
0.993 - 0.007	0,729	0,763	0,245	0,746	0,492	0,248	0,995	0,189	0,621	0,243	0,555	0,922	0,168	0,738	0,477
0.985 - 0.015	0,594	0,866	0,201	0,730	0,46	0,168	0,999	0,205	0,584	0,167	0,637	0,874	0,184	0,756	0,511
0.970 - 0.030	0,697	0,750	0,263	0,724	0,447	0,493	0,948	0,164	0,720	0,441	0,840	0,761	0,220	0,800	0,601
0.930 - 0.070	0,752	0,800	0,212	0,776	0,552	0,525	0,948	0,156	0,736	0,473	0,811	0,804	0,194	0,808	0,615
0.850 - 0.150	0,754	0,799	0,212	0,776	0,553	0,724	0,858	0,175	0,791	0,582	0,819	0,816	0,183	0,817	0,635

Distributions "0" - "1"	ARM					Naive Bayes					SMOTE				
	sensib	spécf	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss
0.993 - 0.007	0,729	0,763	0,245	0,746	0,492	0,814	0,775	0,216	0,794	0,589	0,649	0,705	0,309	0,677	0,354
0.985 - 0.015	0,594	0,866	0,201	0,730	0,460	0,829	0,773	0,213	0,801	0,602	0,728	0,776	0,236	0,752	0,504
0.970 - 0.030	0,697	0,750	0,263	0,724	0,447	0,831	0,776	0,211	0,804	0,607	0,649	0,855	0,196	0,752	0,504
0.930 - 0.070	0,752	0,800	0,212	0,776	0,552	0,835	0,770	0,214	0,802	0,605	0,793	0,768	0,226	0,780	0,561
0.850 - 0.150	0,754	0,799	0,212	0,776	0,553	0,825	0,784	0,206	0,804	0,609	0,825	0,754	0,229	0,790	0,579

Distributions "0" - "1"	ARM					Boosting					Random forests				
	sensib	spécf	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss
0.993 - 0.007	0,729	0,763	0,245	0,746	0,492	0,806	0,807	0,193	0,806	0,613	0,733	0,809	0,210	0,771	0,542
0.985 - 0.015	0,594	0,866	0,201	0,730	0,460	0,820	0,807	0,190	0,814	0,627	0,793	0,775	0,220	0,784	0,568
0.970 - 0.030	0,697	0,750	0,263	0,724	0,447	0,823	0,812	0,185	0,818	0,635	0,800	0,794	0,205	0,797	0,594
0.930 - 0.070	0,752	0,800	0,212	0,776	0,552	0,839	0,817	0,177	0,828	0,656	0,788	0,800	0,203	0,794	0,588
0.850 - 0.150	0,754	0,799	0,212	0,776	0,553	0,831	0,832	0,169	0,831	0,663	0,808	0,808	0,191	0,808	0,616

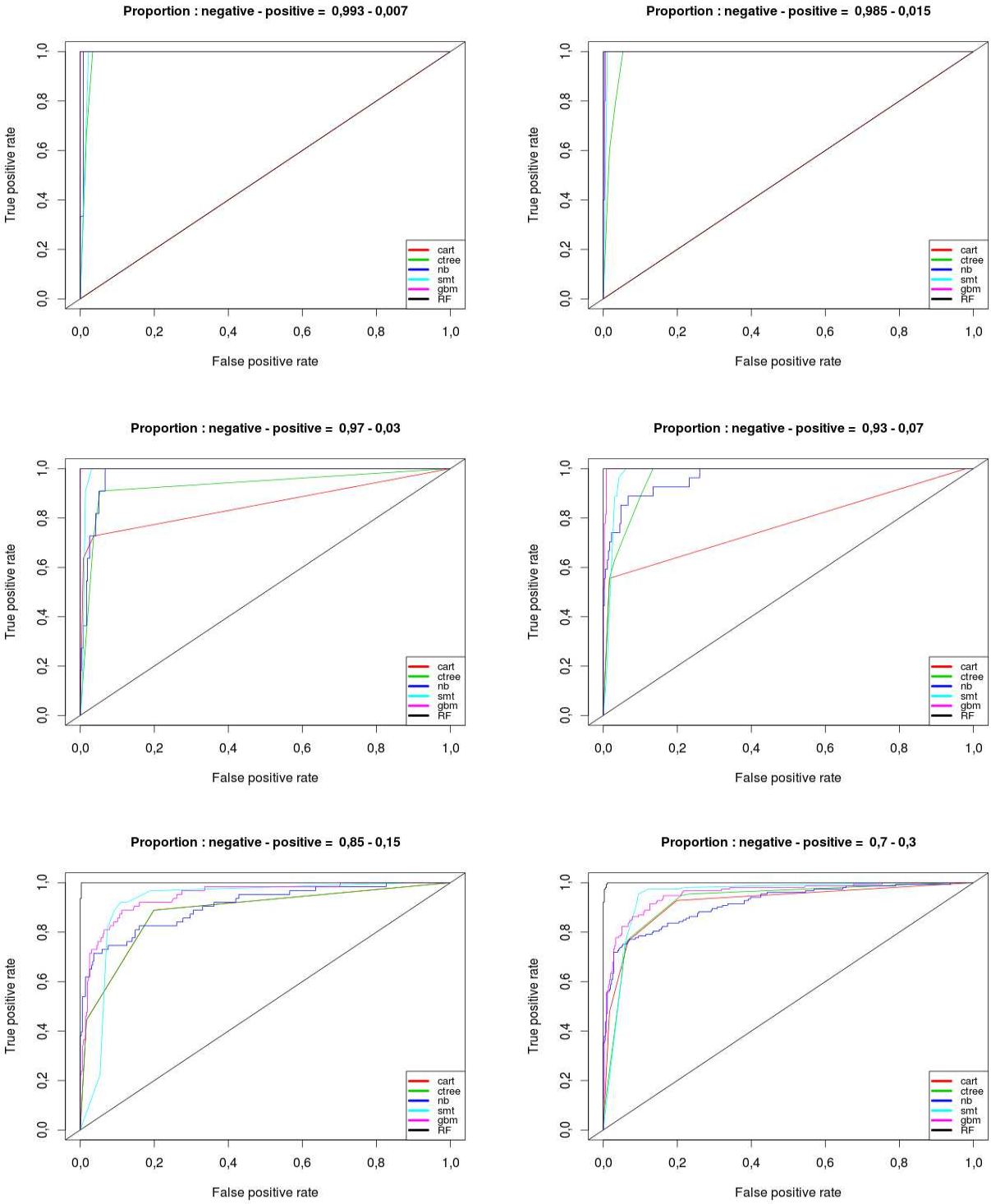
Tableau III.12 – Performances prédictives des méthodes alternatives

Ici aussi on obtient des résultats analogiques aux résultats obtenus lorsque la distribution de l'ensemble d'apprentissage est identique à la distribution de l'ensemble test. On observe que la méthode d'apprentissage ARM est plus performante que la méthode CART. Du point de vue de l'aire en dessous de la courbe ROC (AUC) et du score de Pierce (PSS), la méthode ARM enregistre des valeurs largement au dessus des valeurs de la méthode CART. Elle produit également des sensibilités plus élevées variant entre 59% et 75% tandis que la méthode CART enregistre des sensibilités entre 16% et 72%. Par contre la méthode CART est plus spécifique et admet des erreurs de classement plus faibles sur tous les échantillons simulés.
Dans le cas où la distribution d'apprentissage est différente de la distribution test, les indices de performances du classifieur naïf de Bayes sont meilleurs que les indices de performance de la méthode d'apprentissage ARM sur tous les échantillons simulés sauf au niveau de la spécificité où on a enregistré des taux sensiblement égaux. On peut constater aussi que la méthode Boosting domine largement la méthode ARM sur tous les échantillons en plus elle enregistre des taux d'erreur inférieurs à 20% des scores de Pierce supérieurs à 61% . Tandis que la méthode des forêts aléatoires enregistre des taux d'erreurs inférieurs à 22% et des scores de Pierce compris entre $54 - 61\%$. Là où la méthode ARM enregistre des taux d'erreurs supérieurs à 20% et des scores de Pierce inférieurs à 55%.

– ARM : Association Rules Mining ; CART : Classification And Regression Tree ; CTREE : Conditional tree ; Naive Bayes : Naive Bayes Classifier ; SMOTE : Synthetic Minority Oversampling Technique,

7.2.2 Données Credit Approval Data Set

Le jeu de données "credit approval" concerne des demandes de carte de crédit [24]. Tous les noms et valeurs des variables ont été modifiés pour protéger la confidentialité des données. Les données contiennent au total 690 observations incluant les données manquantes. Elles sont constituées d'un mélange de 6 variables numériques, de 9 variables non-numériques et d'une variable réponse binaire ("+","-"). L'objectif visé dans cette analyse est de trouver un profil prédictif d'approbation d'une carte crédit à un sujet donné.



On constate également que lorsque la proportion d'observations positives devient de plus en plus grande, les courbes ROC se rapprochent de plus en plus.

Distributions	ARM					CART					CTREE				
"-" "+"	sensib	spécf	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss
0.993 - 0.007	1.000	0.852	0.147	0.926	0.852	-	-	-	-	-	1.000	0,966	0,033	0,983	0,966
0.985 - 0.015	1.000	0.832	0.166	0.916	0.832	-	-	-	-	-	1.000	0,947	0,052	0,974	0,947
0.970 - 0.030	0.909	0.714	0.280	0.811	0.632	0,727	0,964	0,043	0,845	0,691	0,909	0,947	0,055	0,928	0,856
0.930 - 0.070	0.889	0.818	0.177	0.853	0.707	0,556	0,983	0,047	0,770	0,539	1,000	0,866	0,125	0,933	0,866
0.850 - 0.150	0.857	0.765	0.221	0.811	0.622	0,889	0,801	0,186	0,845	0,690	0,889	0,801	0,186	0,845	0,690
0.700 - 0.300	0.935	0.625	0.283	0.780	0.560	0,928	0,801	0,161	0,864	0,729	0,948	0,790	0,163	0,869	0,738

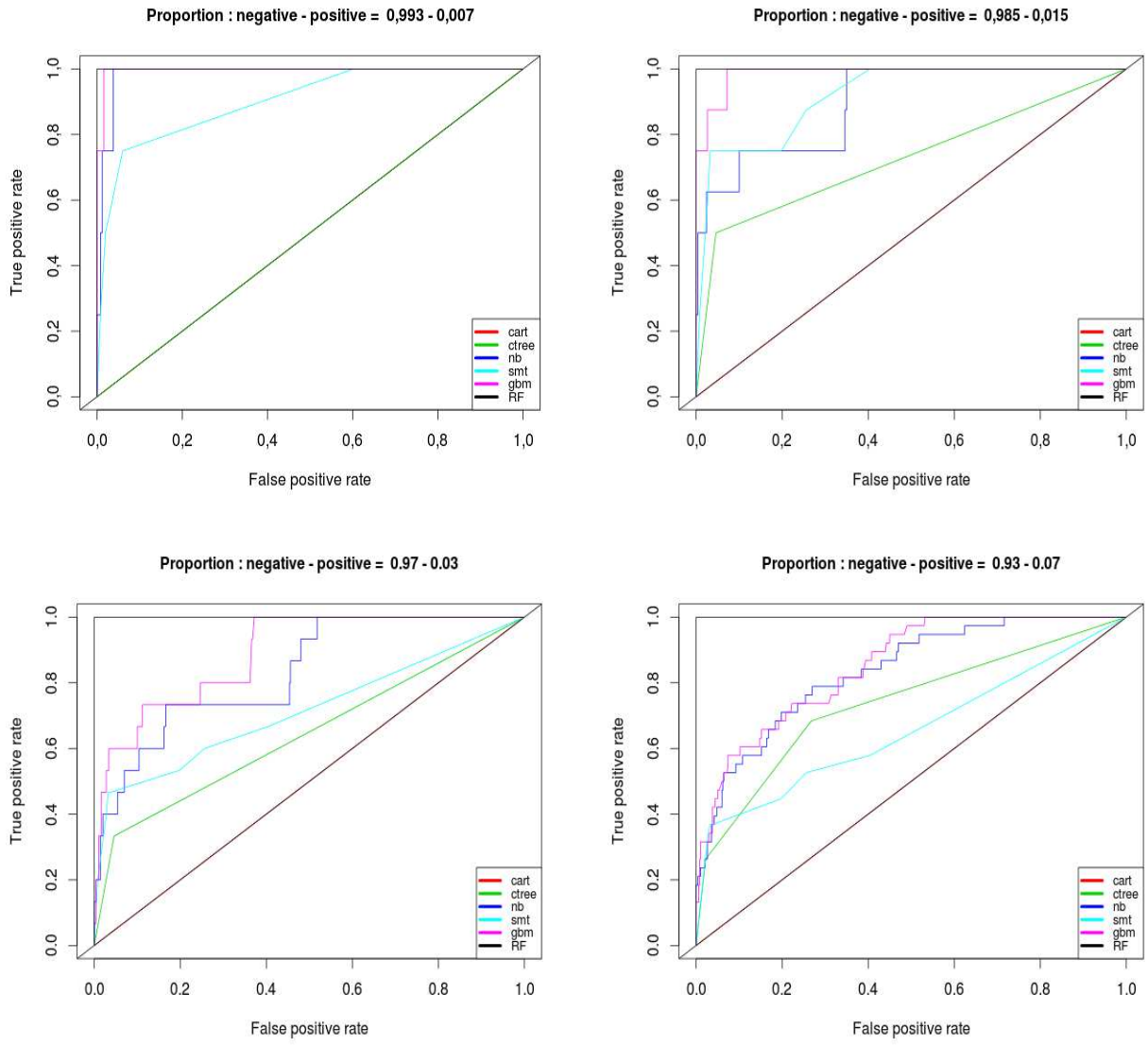
Distributions	ARM					Naive Bayes					SMOTE				
"-" "+"	sensib	spécf	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss
0.993 - 0.007	1.000	0.852	0.147	0.926	0.852	1.000	0,992	0,008	0,996	0,992	1.000	0,992	0,008	0,996	0,992
0.985 - 0.015	1.000	0.832	0.166	0.916	0.832	1.000	0,997	0,003	0,998	0,997	1.000	0,997	0,003	0,998	0,997
0.970 - 0.030	0.909	0.714	0.280	0.811	0.632	1,000	0,933	0,065	0,966	0,933	1,000	0,933	0,065	0,966	0,933
0.930 - 0.070	0.889	0.818	0.177	0.853	0.707	0,889	0,933	0,070	0,911	0,822	0,889	0,933	0,070	0,911	0,822
0.850 - 0.150	0.857	0.765	0.221	0.811	0.622	0,825	0,840	0,162	0,832	0,665	0,825	0,840	0,162	0,832	0,665
0.700 - 0.300	0.935	0.625	0.283	0.780	0.560	0,784	0,905	0,132	0,844	0,689	0,784	0,905	0,132	0,844	0,689

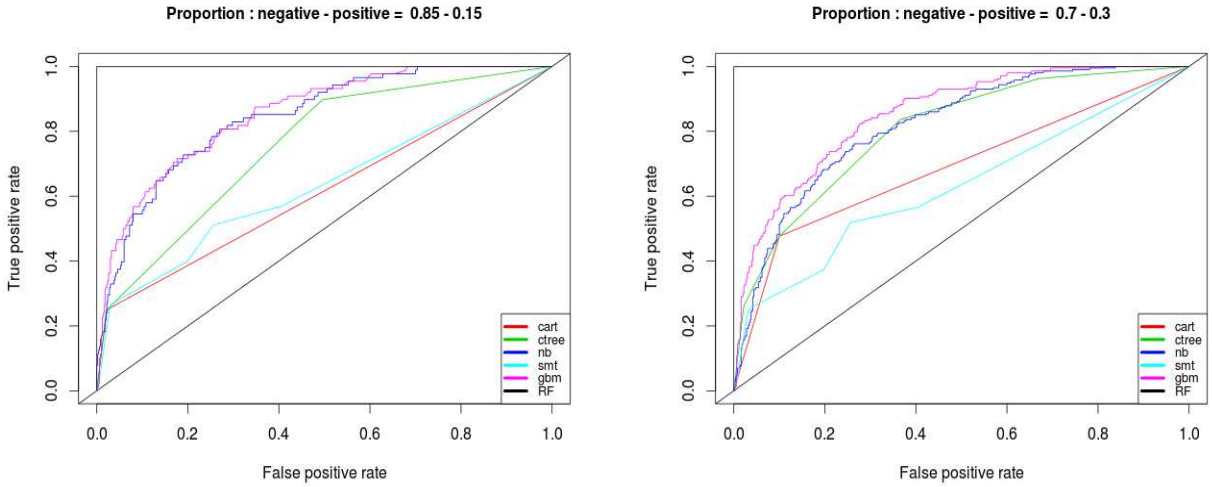
Distrbutions	ARM					Boosting					Random Forests				
"-" "+"	sensib	spécf	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss
0.993 - 0.007	1.000	0.852	0.147	0.926	0.852	1.000	1,000	0,000	1,000	1,000	1.000	1,000	0,000	1,000	1,000
0.985 - 0.015	1.000	0.832	0.166	0.916	0.832	1.000	0,994	0,006	0,997	0,994	1.000	1,000	0,000	1,000	1,000
0.970 - 0.030	0.909	0.714	0.280	0.811	0.632	1,000	1,000	0,000	1,000	1,000	1,000	1,000	0,000	1,000	1,000
0.930 - 0.070	0.889	0.818	0.177	0.853	0.707	1,000	0,992	0,008	0,996	0,992	1,000	1,000	0,000	1,000	1,000
0.850 - 0.150	0.857	0.765	0.221	0.811	0.622	0,889	0,888	0,112	0,889	0,777	1,000	0,997	0,002	0,998	0,997
0.700 - 0.300	0.935	0.625	0.283	0.780	0.560	0,915	0,874	0,113	0,895	0,789	0,993	0,992	0,008	0,992	0,985

Tableau III.13 – Performances prédictives des méthodes alternatives par bootstrap

7.2.3 Données Pima Indians Diabetes Data Set

Le jeu de données "pima-indians-diabetes" est constitué par des femmes d'au moins 21 ans d'origine indienne Pima auxquelles on a administré un test pour le diabète [27]. L'échantillon est constitué de 8 variables numériques et d'une variable réponse binaire qui prend la valeur 1 si le test est positif. Il contient au total 768 observations. L'objectif de l'analyse est de déterminer si oui ou non la patiente présente des signes de diabète selon les normes de l'organisation mondiale de la santé.





On constate de même que lorsque la proportion d'observations positives devient de plus en plus grande, les courbes ROC se rapprochent de plus en plus.

Distributions		ARM					CART					CTREE				
"-"	"+"	sensib	spécf	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss
0.993	0.007	1.000	0.824	0.175	0.912	0.824	-	-	-	-	-	-	-	-	-	-
0.985	- 0.015	0.875	0.846	0.154	0.860	0.721	-	-	-	-	-	0,50	0,954	0,053	0,727	0,454
0.970	- 0.030	0.933	0.786	0.210	0.859	0.719	-	-	-	-	-	0,333	0,954	0,064	0,644	0,287
0.930	- 0.070	0.711	0.782	0.223	0.746	0.492	-	-	-	-	-	0,684	0,732	0,271	0,708	0,416
0.850	- 0.150	0.784	0.602	0.370	0.693	0.482	0,250	0,978	0,131	0,614	0,228	0,807	0,574	0,391	0,690	0,381
0.700	- 0.300	0.785	0.682	0.287	0.733	0.466	0,477	0,900	0,227	0,688	0,377	0,836	0,634	0,305	0,735	0,470

Distributions		ARM					Naive Bayes					SMOTE				
"-"	"+"	sensib	spécf	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss
0.993	- 0.007	1.000	0.824	0.175	0.912	0.824	1.000	0,962	0,038	0,981	0,962	1.000	0,962	0,038	0,981	0,962
0.985	- 0.015	0.875	0.846	0.154	0.860	0.721	0,750	0,900	0,102	0,825	0,650	0,750	0,900	0,102	0,825	0,650
0.970	- 0.030	0.933	0.786	0.210	0.859	0.719	0,733	0,834	0,169	0,784	0,567	0,733	0,834	0,169	0,784	0,567
0.930	- 0.070	0.711	0.782	0.223	0.746	0.492	0,789	0,730	0,266	0,760	0,519	0,789	0,730	0,266	0,760	0,519
0.850	- 0.150	0.784	0.602	0.370	0.693	0.482	0,784	0,748	0,246	0,766	0,532	0,784	0,748	0,246	0,766	0,532
0.700	- 0.300	0.785	0.682	0.287	0.733	0.466	0,762	0,736	0,256	0,749	0,498	0,762	0,736	0,256	0,749	0,498

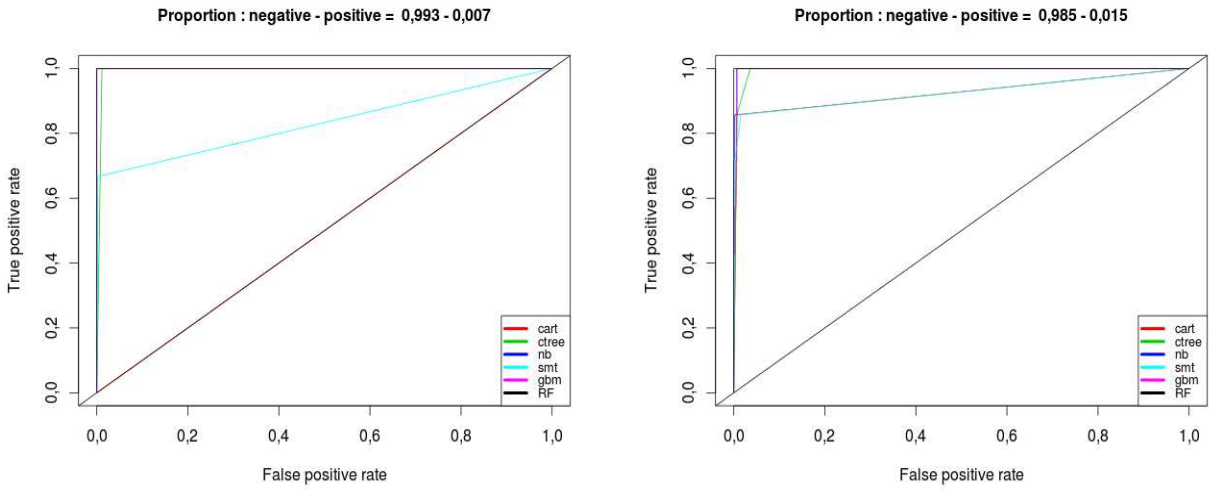
Distributions		ARM					Boosting					Random Forests				
"-"	"+"	sensib	spécf	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss
0.993	- 0.007	1.000	0.824	0.175	0.912	0.824	1.000	0,984	0,016	0,992	0,984	1.000	1,000	0,000	1,000	1,000
0.985	- 0.015	0.875	0.846	0.154	0.860	0.721	1,000	0,928	0,071	0,964	0,928	1,000	1,000	0,000	1,000	1,000
0.970	- 0.030	0.933	0.786	0.210	0.859	0.719	0,800	0,758	0,241	0,779	0,558	1,000	1,000	0,000	1,000	1,000
0.930	- 0.070	0.711	0.782	0.223	0.746	0.492	0,737	0,778	0,225	0,758	0,515	1,000	1,000	0,000	1,000	1,000
0.850	- 0.150	0.784	0.602	0.370	0.693	0.482	0,716	0,824	0,193	0,770	0,540	1,000	1,000	0,000	1,000	1,000
0.700	- 0.300	0.785	0.682	0.287	0.733	0.466	0,822	0,724	0,246	0,773	0,546	1,000	1,000	0,000	1,000	1,000

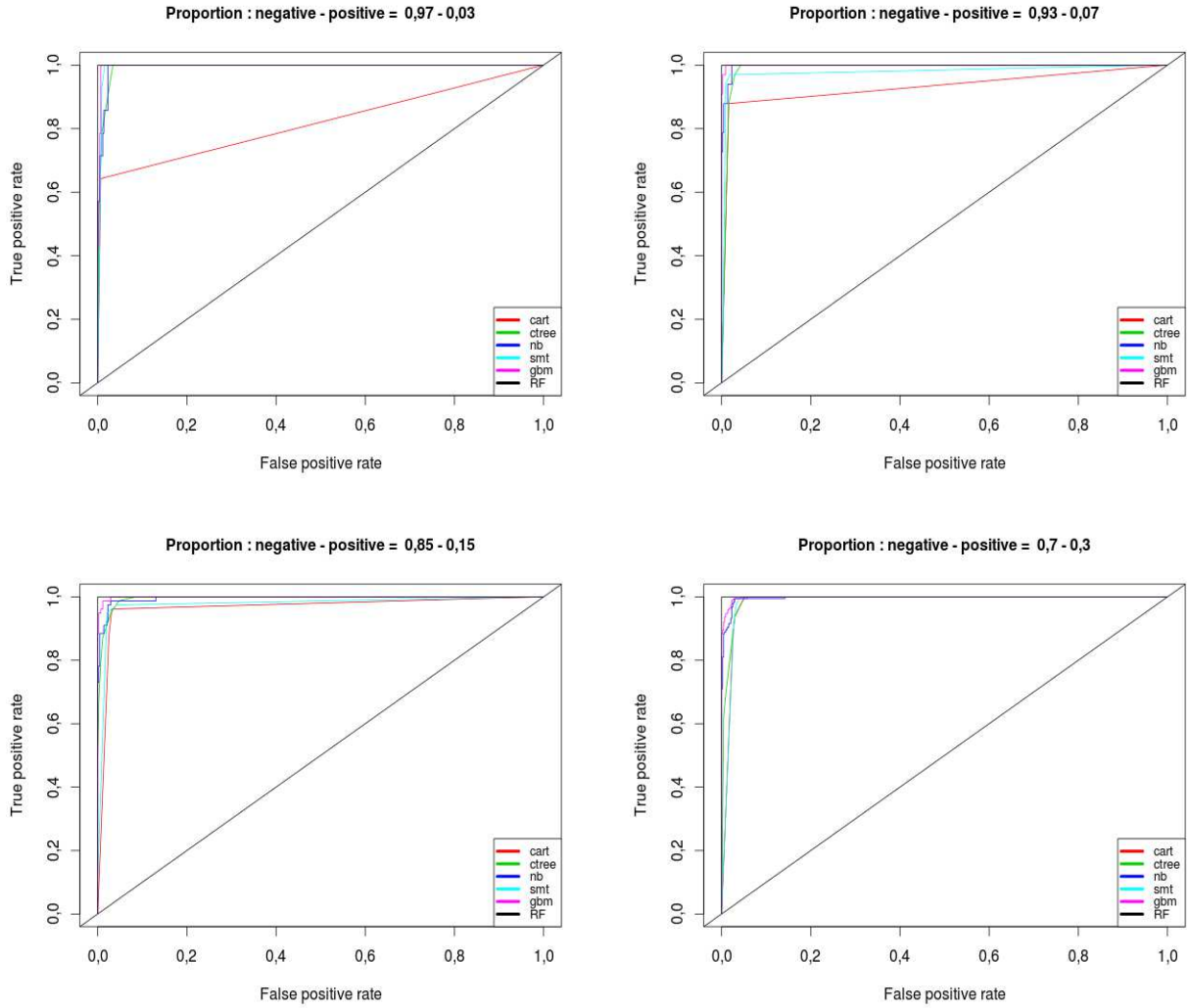
Tableau III.14 – Performances prédictives des méthodes alternatives par bootstrap

Les résultats obtenus à partir des données "Pima Indians Diabetes Dataset" et "Credit Approval Dataset" montrent que, même en présence d'un jeux de données de petite taille, la méthode ARM reste toujours meilleur que la méthode CART de même que la méthode CTREE dont, pour les données "Credit Approval Dataset", les taux d'erreur peuvent aller jusqu'à 39% et les scores de Pierce inférieurs à 47% tandis que la méthode ARM enregistre des scores supérieurs à 47%. Ce pendant elles enregistrent des scores de même ordre de grandeur pour les données "Pima Indians Diabetes Dataset" mais avec des taux d'erreur plus élevés pour la méthode ARM. Il faut noter aussi que pour les deux jeux de données les indicateurs de performance (sensibilité, spécificité, AUC et PSS) décroissent et le taux d'erreur croît au fure et à mesure que la proportion d'observations positives augmente.

7.2.4 Données Breast Cancer Data Set

Les données obtenues à partir du diagnostique de Wisconsin du cancer du sein (WDBC), fourni par le Centre Hospitalier Universitaire de Wisconsin, a été dérivé d'un groupe d'images par aspiration à l'aiguille fine (FNA) de la poitrine [21]. Une programmation génétique avec différentes tailles de la population a été utilisée pour cette étude. L'objectif est d'identifier la classe "benign" ou "malignant" de chaque numéro. Les échantillons arrivent périodiquement comme le Dr Wolberg rapporte ses cas cliniques. La base de données reflète donc ce regroupement chronologique des données. Chaque variable à l'exception de la première a été convertie en 11 attributs numériques primitifs avec des valeurs allant de 0 à 10. Il y a 16 valeurs manquantes. Les données contiennent 699 observations sur 11 variables, l'une étant une variable de caractère, 9 étant ordonnées ou nominales, et une classe cible.





On peut remarquer également que lorsque la proportion d'observations positives devient de plus en plus grande, les courbes ROC se rapprochent de plus en plus.

Distributions	ARM					CART					CTREE				
"-" "+"	sensib	spécf	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss
0.993 - 0.007	1,000	0,964	0,036	0,982	0,964	-	-	-	-	-	1.000	0,989	0,011	0,994	0,989
0.985 - 0.015	1,000	0,908	0,091	0,954	0,908	0,857	0,993	0,009	0,925	0,850	1.000	0,964	0,035	0,982	0,964
0.970 - 0.030	1,000	0,883	0,114	0,942	0,883	0,643	0,993	0,018	0,818	0,636	1.000	0,966	0,033	0,983	0,966
0.930 - 0.070	1,000	0,858	0,132	0,929	0,858	0,879	0,984	0,023	0,932	0,863	0,97	0,971	0,029	0,970	0,941
0.850 - 0.150	0,987	0,858	0,123	0,922	0,845	0,962	0,968	0,033	0,965	0,930	0,987	0,953	0,042	0,970	0,940
0.700 - 0.300	0,995	0,858	0,101	0,926	0,853	1,000	0,948	0,036	0,974	0,948	1,000	0,948	0,036	0,974	0,948

Distributions	ARM					Naive Bayes					SMOTE				
"-" "+"	sensib	spécf	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss
0,993 - 0,007	1,000	0,964	0,036	0,982	0,964	1,000	1,000	0,000	1,000	1,000	1,000	1,000	0,000	1,000	1,000
0,985 - 0,015	1,000	0,908	0,091	0,954	0,908	1,000	0,993	0,007	0,996	0,993	1,000	0,993	0,007	0,996	0,993
0,970 - 0,030	1,000	0,883	0,114	0,942	0,883	1,000	0,977	0,022	0,988	0,977	1,000	0,977	0,022	0,988	0,977
0,930 - 0,070	1,000	0,858	0,132	0,929	0,858	1,000	0,977	0,021	0,988	0,977	1,000	0,977	0,021	0,988	0,977
0,850 - 0,150	0,987	0,858	0,123	0,922	0,845	0,987	0,971	0,027	0,979	0,958	0,987	0,971	0,027	0,979	0,958
0,700 - 0,300	0,995	0,858	0,101	0,926	0,853	0,995	0,971	0,023	0,983	0,966	0,995	0,971	0,023	0,983	0,966

Distributions	ARM					Boosting					Random Forests				
"-" "+"	sensib	spécf	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss	sensib	spécif	err.cl	auc	pss
0.993 - 0.007	1,000	0,964	0,036	0,982	0,964	1,000	1,000	0,000	1,000	1,000	1,000	0,000	1,000	1,000	1,000
0.985 - 0.015	1,000	0,908	0,091	0,954	0,908	1,000	0,993	0,007	0,996	0,993	1,000	1,000	0,000	1,000	1,000
0.970 - 0.030	1,000	0,883	0,114	0,942	0,883	1,000	0,993	0,007	0,996	0,993	1,000	1,000	0,000	1,000	1,000
0.930 - 0.070	1,000	0,858	0,132	0,929	0,858	1,000	0,991	0,008	0,996	0,991	1,000	1,000	0,000	1,000	1,000
0.850 - 0.150	0,987	0,858	0,123	0,922	0,845	0,987	0,989	0,012	0,988	0,976	1,000	1,000	0,000	1,000	1,000
0.700 - 0.300	0,995	0,858	0,101	0,926	0,853	0,989	0,977	0,019	0,983	0,966	1,000	1,000	0,000	1,000	1,000

Tableau III.15 – Performances prédictives des méthodes alternatives à partir de 20 échantillons bootstrap

Les résultats obtenus à partir des données "Breast Cancer Dataset" confirment donc que en présence de données de petite taille et déséquilibrées, la méthode ARM domine la méthode CART et enregistre des performances sensiblement équivalentes aux performances obtenues à partir des méthodes de classement telles que la méthode Boosting et la méthode des forêts aléatoires.

Il ressort de cette analyse que notre méthode d'apprentissage est largement plus performante que la méthode CART. Ce pendant elle est comparable à la méthode CTREE, le classifieur naif de Bayes, la méthode SMOTE, le boosting d'arbres de classement et la méthode random forest. Du point de vue de la sensibilité, de la spécificité, de l'aire en dessous de la courbe ROC et du score de Pierce, notre méthode d'apprentissage à les même ordres de valeur que les méthodes citées précédemment. Par contre elle enregistre une erreur de classement supérieur à celles des autres méthodes de l'ordre de 10^{-1} à 10^{-2}.
Par ailleurs, on peut remarquer que si CART et CTREE permettent de fournir un outil d'aide à la décision (arbre de décision) permettant de visualiser des profils pertinents cela n'est pas le cas des méthodes comme le boosting et les forêts aléatoires qui parfois ont des performances supérieures à ceux obtenues par la méthode d'apprentissage étudiée dans la thèse. D'où l'avantage de cette dernière sur les autres car elle permet d'avoir des performances sensiblement égales aux méthodes comme le boosting et les forêts aléatoires mais aussi elle permet de visualiser les profils les plus pertinents pour construire une règle de classement.

8 Conclusion

La procédure permet de surmonter l'impuissance des méthodes de régression qui sous-estiment les probabilités conditionnelles de l'apparition de la classe cible lorsque la fréquence des instances qui appartiennent à cette classe est très faible. De plus les interactions d'attributs qui sont fortement corrélées avec la classe cible sont spécifiées, ainsi la fonction de classification n'apparaît pas comme une boîte noire. Néanmoins il faut remarquer qu'une étape de prétraitement des données est nécessaire avant d'effectuer la procédure car il est supposé que les variables soient évaluées sur une échelle non numérique.