

Modélisation prédictive et apprentissage statistique de la carte visa premier



2
0
2
3

Préparé par

Pety Ialimita RAKOTONIAINA
Numéro : 42005638

SOMMAIRE

RESUME	3
<i>Liste des figures</i>	4
<i>Liste des tableaux</i>	4
INTRODUCTION	5
<i>Chapitre 1 : Identification des règles d'associations</i>	7
<i>Chapitre 2 : Classifieur bayésien naïf</i>	18
<i>Chapitre 3 : Les Supports Vecteurs Machines</i>	24
<i>Chapitre 4 : Les Supports Vecteurs Machines</i>	37
CONCLUSION	41
BIBLIOGRAPHIE	42
ANNEXE	43
<i>Annexe 01 : Liste des variables explicatives</i>	43
<i>Annexe 02 : Description des variables explicatives</i>	44
<i>Annexe 03 : Code R sur le chapitre 1</i>	45
<i>Annexe 04 : Code R sur le chapitre 2</i>	47
<i>Annexe 05 : Code R sur le chapitre 3</i>	49
<i>Annexe 06 : Extrait Code R sur le chapitre 4</i>	54

RESUME

Dans cette étude portant sur les données de promotion de la carte Visa Premier dans la banque BRED, trois méthodes de modélisation prédictive ont été utilisées : la recherche des règles d'association, le classifieur naïf de Bayes et le support vecteur machine. La recherche des règles d'association a permis d'identifier des associations significatives entre les caractéristiques des clients et leur propension à souscrire à la carte Visa Premier, tandis que le classifieur naïf de Bayes a offert une performance de classification élevée avec une précision de 88%. Le support vecteur machine a quant à lui donné le meilleur pouvoir discriminant avec une précision pouvant atteindre 98%. La combinaison de la première et de la troisième méthode via l'analyse de scoring et la prévision du SVM est recommandée pour améliorer la stratégie de marketing de la banque et augmenter le taux de retour des campagnes de marketing.



Liste des figures

<i>Figure 1 : Représentation des règles d'associations détectées</i>	12
<i>Figure 2 : Illustration du mécanisme du SVM linéaire</i>	24
<i>Figure 3 : Performance du SVM linéaire</i>	29
<i>Figure 4 : Performance du SVM radial gaussien</i>	31
<i>Figure 5 : SVM linéaire appliqué aux 02 axes factoriels</i>	34

Liste des tableaux

<i>Tableau 1 : Top 10 des règles d'associations de la possession de la carte visa premier</i>	15
<i>Tableau 2 : Top 10 des règles d'associations de la non-possession de la carte visa premier</i>	17
<i>Tableau 3 : Liste des modalités à fortes proportion</i>	21
<i>Tableau 4 : Performance du SVM linéaire</i>	30
<i>Tableau 5 : Grille de score</i>	37
<i>Tableau 6 : Score des top règles d'associations</i>	39



INTRODUCTION

Le Groupe Banque populaire et à la Caisse d'épargne française (BPCE) est actuellement le deuxième acteur bancaire en France. Avec 100 000 collaborateurs, il est au service de 36 millions de clients dans le monde, particuliers, professionnels, entreprises, investisseurs et collectivités locales. Il est présent dans la banque de proximité et l'assurance en France avec ses deux grands réseaux Banque Populaire et Caisse d'Epargne ainsi que la Banque Palatine. Il déploie également au niveau mondial, avec Natixis, les métiers de gestion d'actifs, de paiements et de banque de grande clientèle. Soucieux de rayonner à l'international, elle se rue vers la promotion de la carte visa premier via sa banque populaire en particulier à La Réunion.

La nécessité de développer ce produit de gamme à l'échelle internationale est indéniable pour faire face à une rude concurrence exercée sur les banques à réseau par les sociétés de crédit et les nouveaux entrants notamment les banques étrangères à l'image de JP Morgan. Ce produit est de plus en plus important pour satisfaire aussi les attentes accrues en termes de qualité de service de nos clients. Actuellement, la part de client a remplacé la part de marché. Ainsi, la compréhension de leur attente et l'anticipation de leur besoin deviennent un objectif majeur pour notre entreprise qui souhaite augmenter la rentabilité et la fidélité de la clientèle. Pour y parvenir, la maîtrise de l'information fournie par les clients est un élément primordial de performance. Mener une gestion de la relation Client (GRC) analytique¹ efficace, en vue d'un marketing one-to-one², est alors au cœur de la problématique. Pour cette raison, notre banque désire automatiser sa règle de décision d'octroi de la Carte VISA premier via notamment le scoring.

Comme son nom l'indique, le scoring consiste à affecter un score à chaque client prospect. Le score obtenu traduit généralement la probabilité qu'un individu réponde à une sollicitation marketing ou appartienne à la cible recherchée, le fait d'être intéressé ou non par la carte en question. Il mesure donc particulièrement l'appétence pour l'offre potentielle. C'est évident, l'utilisation de la technique de score d'appétence permet d'améliorer le taux de retour des campagnes de marketing de la banque. Avec elle, la banque peut atteindre plus de clients réceptifs sans augmenter le nombre de clients contactés voire parfois en les diminuant. Ce qui permettra à notre banque d'améliorer probablement ses chiffres d'affaires en moins de rendez-vous. Ce projet de Data mining

¹ La GRC analytique concerne, essentiellement, la restitution des informations pertinentes qui permettront de mieux appréhender les clients de l'entreprise : leur profil ou leurs besoins.

² Cette approche individualisée assure une connaissance des clients afin de leur proposer ce qui leur convient le mieux surtout dans le secteur tertiaire, puisque les produits y afférents se ressemblent beaucoup d'un agent à un autre, où les prix ne sont souvent déterminant pour constituer et développer sa clientèle, et où le service puis la relation clientèle qui donnent le pouvoir de démarcation.

prédictif³ est-il réalisable ? Telle est la question principale qui nous anime tout au long de notre travail.

Après avoir fixé les objectifs du travail, la question se pose : quelles sont les données à analyser pour y atteindre ? Grâce à notre système d'information, nous avons pu recueillir une base de données initiale légalement et techniquement exploitables, fiables et suffisamment à jour sur les caractéristiques d'un échantillon de 1063 clients étudiés à La Réunion. Ces caractéristiques couplées avec une base des données retraçant l'historique mensuel bancaire nous permettent d'établir la liste des variables à manipuler. Cette dernière est composée de la variable à expliquer « cible » dénommée CARVP, avec sa modalité Coui (Possesseurs de la carte visa premier : 357 observations soit 33,6%) et sa modalité Cnon (Non-possesseurs de la carte visa premier : 706 observations soit 66,4%), et de variables 34 variables explicatives retenues après avoir fait l'exploration et la transformation des données. Elles sont décrites en détails dans le tableau 1 de l'annexe.

Disposant les données bien préparées, nous passerons directement à la phase d'apprentissage et modélisation en utilisant le logiciel R principalement. Cette phase consiste à lancer divers algorithmes⁴ en fonction des objectifs poursuivis puis de la nature et caractéristiques des données explorées et transformées. Il s'agit aussi à interpréter leurs résultats et surtout comparer leur pertinence. Dans cette optique, la règle d'association, une des méthodes les plus populaires dans les supermarchés, fera l'objet d'un essai dans notre banque. Après cela, nous verrons le naïf de Bayes qui est souvent préférée dans les projets de Data mining grâce à sa simplicité théorique et surtout son efficacité empirique. Ensuite, nous exposerons comment les Supports Vector Machines (SVM) s'appliquent à nos données (« machines à vecteurs de support » ou « séparateurs à vaste marge »). Cette méthode est très utilisée dans de nombreux domaines surtout dans le monde financier comme le credit scoring (Belloti et Crook, 2009). Non seulement elle est souvent plus précise⁵ dans les résultats des études antérieurs mais, contrairement aux réseaux des neurones, elle est aussi plus robuste⁶ et convient plus à notre base de données de petite taille (Tuffery,2017).

³ La prédiction ne doit pas être confondue avec la prévision, laquelle consiste à évaluer la valeur d'une variable à l'instant t sachant ses valeurs aux instant précédents.

⁴ Par algorithme ou méthode, nous entendons la procédure qui permet, à partir des données en entrée, d'obtenir en sortie un résultat sous la forme d'un modèle prédictif notamment

⁵ Le taux d'erreur et la proportion d'individus mal classés sont faibles. Un indicateur de qualité comme l'aire sous la courbe de ROC est relativement plus proche de 1.

⁶ Le modèle dépend aussi peu de l'échantillon particulier d'apprentissage utilisé et bien se généraliser à d'autres échantillons. Il sera moins sensible possible aux fluctuations aléatoires de certaines variables. Même si les données évoluent dans le temps, le modèle s'adapte correctement à de nouveaux échantillons, durant un laps de temps raisonnable qui dépend particulièrement de la vitesse d'évolution des offres et des publics.

Chapitre 1 : Identification des règles d'associations

Dans ce chapitre, nous allons décrire brièvement le cadre théorique de la règle d'association (1.1) avant de passer aux étapes de la modélisation prédictive (1.2 à 1.5).

1.1 La recherche des règles d'associations : une méthode supervisée exceptionnelle

D'une manière générale, rechercher des règles d'associations signifie précisément rechercher les valeurs conjointes les plus fréquentes d'un espace de variables $X = (X_1, X_2, \dots, X_p)$ d'un jeu de données en particulier les variables retenues dans notre base des données vispremv. De ce fait, la détection des règles d'associations n'est donc pas une méthode véritablement prédictive ; la variable à expliquer n'est pas fixée. Cela ne veut pas dire qu'il dire qu'il serait impossible de prédire la possession de la carte visa premier par la règle d'association. Au contraire, notre objectif sera atteint. Parce qu'elle n'est pas non plus purement descriptive. L'intérêt est de prédire mais dont l'objet n'est pas fixé ce qui permet de trouver des règles inattendues. La fréquence d'une règle reflète le degré de réapparition simultanée des différentes observations. De ce fait, une règle d'association traduit seulement une co-occurrence -Si Condition alors Résultat- et non une relation de causalité. Au vu de ce qui précède, les relations découvertes peuvent être représentées sous forme d'ensemble d'items fréquents. En effet, soit x_i des valeurs, on dit que $(X_i = x_i)$ est un item et que $T \cap_{i \in I_j} (X_i = x_i)$ est une collection d'un ou pratiquement d'un ensemble d'items appelé itemset. La règle d'association peut être donc dans cet angle comme l'identification des itemsets qui se reproduisent ensemble le plus souvent. Ce nombre d'apparition commune est mesurée par ce qu'on appelle le compte de support k et on dit k -itemset. Le support, tout court ou indice de support, indique la fréquence de support de compte dans le nombre de total de transaction. C'est la probabilité $P(T_j)$. Il est possible de se limiter à la recherche de ces ensembles d'items. Cependant, on s'intéresse généralement à ce qu'on appelle les règles d'associations. Ce sont des applications de la forme $X \rightarrow Y$ ⁷ où X et Y sont des ensembles d'items disjoints : $T_j = \{\text{Condition } X \cup \text{Résultat } Y\} = \{\text{Antécédent } X \cup Y\}$ ⁸

On parle aussi d'antécédent (ou lhs : left_hand-side of the rule) et de conséquent (ou rhs : right_hand-side of the rule) de la règle. Et la formule du support est exactement ci-dessous. Par exemple fictif, les clients, qui sont caractérisés par un taux d'endettement élevé et plusieurs comptes à vue, pourraient détenir fréquemment la carte visa premier. Si cette situation se répète cinq cent fois dans la base des données de 1063 observations, le support vaut 47% (500/1063). Cela veut dire que 47% des clients possèdent simultanément, à chacun, les caractéristiques précédentes. Malgré le fait que ce support se

⁷ $Y \in X$, Y pourrait être CARVP et bien d'autres variables.

⁸ C'est un abus de langage. En réalité, T_j est l'intersection de l'Antécédent et du Conséquent.

rapproche du $\frac{1}{2}$, il nous est impossible de dire que la reproduction simultanée des itemsets de X et Y sont fréquents et suffisamment grande.

Il se peut que ce phénomène soit le fruit du hasard. Il faut donc établir un seuil au-dessus duquel, ou au moins égal, le support peut être significatif. Cette condition s'applique aussi sur la confiance. A la différence du support, la confiance mesure la répétition dans la transaction des items de Y contenus dans X. Pour détecter les règles d'associations respectant ces conditions, le principal algorithme le plus utilisé, dont nous allons déployer, est l'algorithme Apriori (Agrawal R. et al, 1993). Il cherche consécutivement les supports et les confiances supérieures à un certain seuil donné. C'est à partir de là que nous pouvons mesurer la force de chaque règle d'association : le support mesure sa fiabilité et la confiance sa précision.

Les éléments de mesure de la règle d'association

$$\text{Support}, s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}$$

$$\text{Confiance}, c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

En remodelant la formule de la confiance, nous arrivons à l'égalité suivante :

$$\text{Confiance} = \frac{P(X \cap Y)}{P(X)} = P(Y|X)$$

La confiance n'est autre chose que la probabilité conditionnelle de Y sachant X. Pour se situer dans le cadre de Data mining, la recherche menée ici relève d'une méthode non supervisée, contrairement aux méthodes supervisées, où l'on s'intéresse à la probabilité conditionnelle $P(Y|X)$ -avec Y la variable cible CARVP- et non à la probabilité $P(X)$.

Si le seuil de confiance est trop faible, la règle n'est pas fiable pour prédire le conséquent. Si le seuil est trop faible, la règle sera trop peut-être fiable mais pourra être trop rarement appliquée pour être véritablement utiles. Cependant, la règle pourrait satisfaire ces deux critères mais elle n'apporte pas un gain suffisant par rapport à une prédiction au hasard du conséquent.

Par prolongement, l'amélioration apportée par une règle d'association, par rapport à une réponse au hasard est appelée « lift » :

$$\text{Lift} = \frac{P(Y|X)}{P(Y)}$$

Lift est significatif s'il est > 1 . En effet, dans ce cas l'antécédent favorise le conséquent dans la règle d'association puisque $P(Y|X) > P(Y)$. Le lift ne l'est s'il est inférieur ou égale à 1. Dans le cas où $\text{lift} = 1$, $P(Y|X) = P(Y)$. Cela veut dire simplement que le résultat Y est indépendant de l'antécédent X. Quand lift est inférieur à 1, la règle n'apporte rien puisque $P(Y|X) < P(Y)$. Dans ce cas, il faut penser à manipuler le lift de la règle inverse, qui est supérieur à 1, en espérant que cette dernière soit intéressante en termes de marketing.

1.2 Présentation de la base des données appropriée à la recherche des règles d'association

Tout d'abord, l'algorithme Apriori de recherche des règles d'association fonctionne avec des variables correspondantes aux items soient des facteurs (valeurs discrètes). C'est une contrainte qu'il fallait respecter.

Ainsi, nous allons utiliser donc la base des données vispremv. A titre de rappel, visprem détient l'ensemble des variables brutes de la banque au départ. Après une préparation et exploration des données, nous avons essentiellement sélectionné 35 variables sur 45, elles forment la base des données vispremt. Nous avons transformé, par la suite, en facteurs l'ensemble des variables après avoir les discrétiser en principe sauf pour les quatre variables déjà en forme qualitative. La variable Carvp à expliquer principalement est aussi transformée, car elle sera analysée comme les autres. Pour alléger la base des données obtenues, il faut sélectionner les variables que nous utiliserons dans la règle d'association.

Finalement, la base des données contiennent 1063 observations et autant de transactions. En colonne, il y a 25 variables qualitatives au total dont 24 sont explicatives et 1 (Carvp) la variable à expliquer. Les différentes modalités des variables sont les items. Concrètement, la variable cible est représentée par deux items : Coui et Cnon. La variable explicative AGEQ est composée de trois modalités : A0 lorsque l'âge est compris entre 18 et 36 ans (les clients jeunes), A1 lorsque l'âge est compris entre 37 et 49 (âge moyen des clients), A3 lorsque l'âge est plus de 50 ans mais inférieur à 65 (les clients vieux). La variable âge, une variable quantitatif discret au départ, nous permet de dégager donc trois items après discrétisation. Toujours à titre illustratif, le taux d'endettement (endetq) vaut 0 si sa modalité est E0 et supérieur à 0 - compris entre 0,69 et 4,6 exactement- si elle est E1. Cette variable quantitative continue initiale nous donne alors deux items après transformation. A part ces variables, force est de constater qu'il y a déjà des variables sous formes qualitatives non recodées d'une part. Tel est le cas de la catégorie socio-professionnelle (pcsq) avec 05 modalités : cadre, employé, interim, ouvrier, sans emploi. D'autre part, il y a les variables qualitatives recodées et regroupées. Il s'agit par exemple de la situation familiale (famiq). La modalité « seul » synthétise les clients célibataire, divorcée, veuf. Quant à la modalité « Fcoup », on fait référence surtout aux clients mariés. Ces exemples nous ont permis d'avoir un contour significatif des différents types de variables dans notre nouvelle base de données. Chaque variable peut faire l'objet de cette description. Pour laisser de la marge à d'autres points plus importants, nous avons décidé de les mettre dans l'annexe 02.

1.3 Les caractéristiques globales des clients de la banque à forte potentielle de règles d'associations

L'algorithme Apriori est implémenté dans la fonction « apriori () » du package « arules » dans le logiciel R. Il faut bien comprendre les arguments de cette fonction pour bien savoir comment initier l'algorithme et interpréter les résultats. La fonction « apriori () » est appelée avec le nom de la structure de données (comme d'habitude), et trois listes de paramètres : parameter, appearance, control. Parameter nous permet principalement de fixer le seuil du support, celui de la confiance, le nombre maximal d'items contenus dans chaque règle et le nombre minimal. Appearance, quant à lui, il nous sert à manipuler le rhs et lhs. Enfin, l'argument control nous assure de manipuler la vitesse de calcul.

Nous lançons pour la première fois l'algorithme, avec les paramètres par défaut. En d'autres termes :

- les valeurs par défaut de l'indice de confiance est égal à 0,8 ou 80% ;
- le support vaut 0,1 ou 10% ;
- le nombre minimum d'éléments à chaque règle est fixé à 1 de sorte qu'au moins on a une règle à un seul item conséquent mais non pas antécédent ;
- le nombre maximum d'items est limité à 10 automatiquement (avec un conséquent et un antécédent) ;
- ce sont des règles d'associations qui sont attendues mais non pas uniquement d'items les plus fréquents.

Apriori

Parameter specification:

confidence	minval	smax	arem	aval	originalSupport	maxtime	support	minlen	maxlen	target	ext
0.8	0.1	1	none	FALSE	TRUE	5	0.1	1	10	rules	TRUE

Algorithmic control:

filter	tree	heap	memopt	load	sort	verbose
0.1	TRUE	TRUE	FALSE	TRUE	2	TRUE

Absolute minimum support count: 106

```
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[65 item(s), 1063 transaction(s)] done [0.00s].
sorting and recoding items ... [63 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10 done [0.49s].
writing ... [651379 rule(s)] done [0.98s].
creating S4 object ... done [0.33s].
```

On retrouve les 1063 dossiers et les 65 modalités des 25 variables du data frame analysé. Dans cette population, on remarque qu'il existe 651 379 de règles d'associations générées par l'algorithme apriori lorsqu'il est lancé par défaut. Afin d'avoir un aperçu de ces règles d'associations c'est-à-dire leur statistique de bases, nous avons lancé la fonction appelée « summary ».

set of 651379 rules

rule length distribution (lhs + rhs):sizes

1	2	3	4	5	6	7	8	9	10
4	254	4834	26320	72233	126701	155249	136389	87379	42016

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.000	6.000	7.000	7.102	8.000	10.000

summary of quality measures:

support	confidence	coverage	lift	count
Min. :0.1007	Min. :0.8000	Min. :0.1007	Min. :0.8989	Min. :107.0
1st Qu.:0.1082	1st Qu.:0.8800	1st Qu.:0.1167	1st Qu.:1.1284	1st Qu.:115.0
Median :0.1195	Median :0.9350	Median :0.1298	Median :1.2930	Median :127.0
Mean :0.1296	Mean :0.9280	Mean :0.1404	Mean :1.4677	Mean :137.8
3rd Qu.:0.1402	3rd Qu.:0.9855	3rd Qu.:0.1524	3rd Qu.:1.5308	3rd Qu.:149.0
Max. :0.8899	Max. :1.0000	Max. :1.0000	Max. :3.8100	Max. :946.0

mining info:

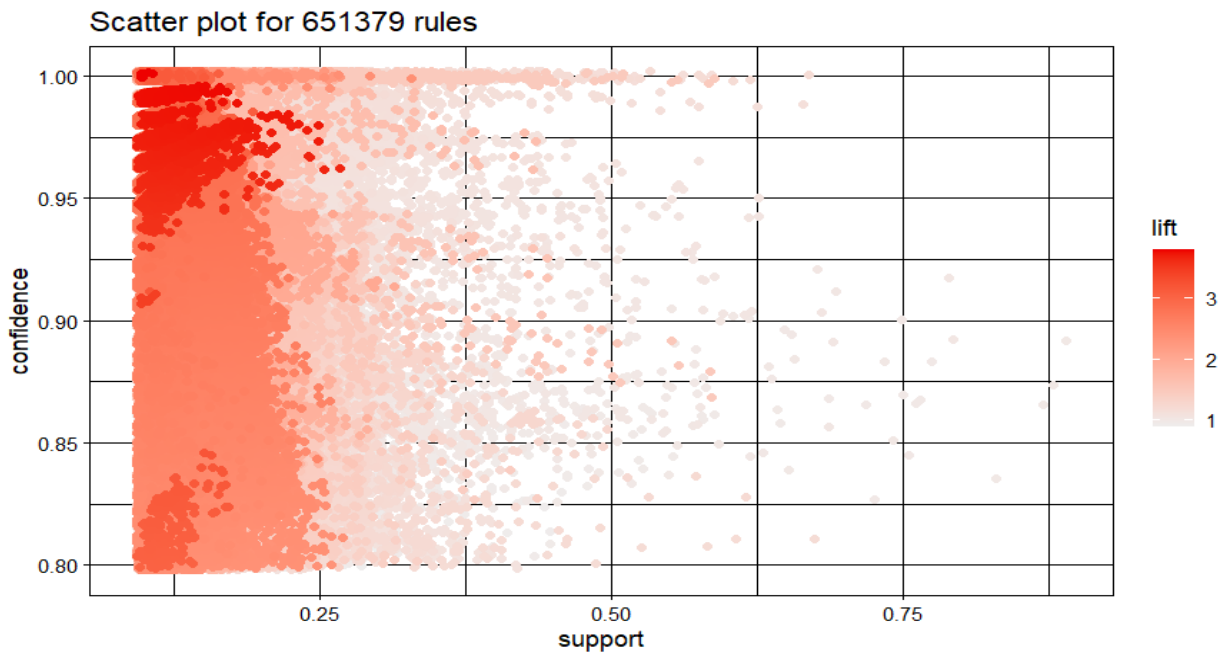
data	ntransactions	support	confidence	call
vispremh	1063	0.1	0.8	apriori(data = vispremh)

A titre interprétatif, parlons de la distribution des règles en général. Quatre (04) règles d'associations peuvent être définies comme ayant 1-itemset. En effet, ce sont les rares règles d'association qui détiennent 1 item uniquement. Par analogie, Il y en a 254 règles avec 2-itemsets. Ce nombre de règle ne cessent d'augmenter et devient 155249 jusqu'aux 7-itemsets. C'est à peu près le nombre moyen des règles d'associations d'ailleurs et également le median. Presque les 50% des règles de transactions ont donc un itemset inférieur à 7.

Ensuite par rapport aux indicateurs de performances des règles, les statistiques de la boîte de distribution rend compte de la tendance centrale, de la dispersion, des valeurs éloignées ou extrêmes et de la forme de la distribution. Concrètement, la moyenne du support est égale à 13% environ. Il n'est pas vraiment dispersé puisque son étendue interquartil vaut 0.0395 uniquement (=Q3 – Q1). Sa distribution n'est pas symétrique puisque la médiane diffère significativement de sa moyenne (0,11 contre 0,12). Ainsi, le support se concentre plus à gauche, dans les valeurs faibles. Mais il y a des valeurs extrêmement élevées qui est égale 0,89. Comparé avec la valeur minimale, on obtient une étendue assez conséquente de l'ordre de presque 0,8. Cette répartition est visible via le graphique ci-dessous. Quant à la confiance, on observe l'opposé exact de l'évolution du support. Sauf que la distribution reste asymétrie vers la droite. Il y a donc une grande chance de trouver un indice de confiance très élevée pour les règles d'associations de la carte visa premier que d'indice de support.

Pour le lift, plus de 75% des valeurs sont supérieures à 1. Les relations entre support, indice de confiance et lift est représenté par le graphique ci-dessous.

Figure 1 : Représentation des règles d'associations détectées



Dans ce graphique, chaque point correspond à une règle, avec son support en abscisse, son indice de confiance en ordonnée, et rouge d'autant plus foncé que son lift est élevé. Comme dans l'application de l'algorithme Apriori, on sera plus sévère sur le seuil de confiance que de support. En effet, les possesseurs de carte, comme nous l'avons eu, sont relativement moins dans nos données avec 33% environ des observations et donc des transactions. Trouver des règles d'explication de CARVP peut être vu donc comme trouver des règles rares. Dans ce cadre, un exemple courant de filtre, selon Tuffery(2012), sera de 75% pour la confiance et 5% pour le support (et bien sûr 1 pour le lift). Mais nous n'allons pas exactement adopter ce choix. Et voici la raison.

D'abord, peu de règles ont un support relativement très élevé, et c'est souvent au prix d'un moindre indice de confiance. On détecte néanmoins des règles intéressantes à l'indice de confiance très élevé (jusqu'à 1) couplé avec un support largement supérieur à 0.5. D'autres règles aussi peuvent être intéressantes car elles conjuguent un indice de confiance de l'ordre de 0.95 et un support supérieur à 0.4. Mais il sera difficile de trouver des règles pertinentes dans cette zone. En effet, pour le lift, ses valeurs les plus élevées se coïncident avec les bonnes valeurs de la confiance mais avec un support plus faible, ce sont les règles les plus complexes. Rien ne nous empêche quand même de percer des règles très intéressantes dans cette partie. Puisque même si les supports sont restés relativement faibles, ils peuvent être fixés supérieurs à 10 tout en gardant la chance de maximiser les autres indicateurs. L'indicateur de confiance sera aussi gardé.

1.4 Top des règles d'associations performants : au carrefour du fonctionnement des comptes courants et des produits bancaires

La fonction « inspect() » permet d'afficher les règles et fournit aussi le support, l'indice de confiance et le lift de chaque règle. En essayant de faire apparaître les 10 règles censées les plus forts en supports, nous obtenons :

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{}	=> {kvunbq=K0}	0.8899341	0.8899341	1.0000000	1.0000000	946
[2]	{}	=> {nptagq=N0}	0.8739417	0.8739417	1.0000000	1.0000000	929
[3]	{}	=> {lgagtq=L0}	0.8673565	0.8673565	1.0000000	1.0000000	922
[4]	{}	=> {vienbq=V0}	0.8353716	0.8353716	1.0000000	1.0000000	888
[5]	{lgagtq=L0}	=> {kvunbq=K0}	0.7939793	0.9154013	0.8673565	1.0286169	844
[6]	{kvunbq=K0}	=> {lgagtq=L0}	0.7939793	0.8921776	0.8899341	1.0286169	844
[7]	{nptagq=N0}	=> {kvunbq=K0}	0.7723424	0.8837460	0.8739417	0.9930465	821
[8]	{kvunbq=K0}	=> {nptagq=N0}	0.7723424	0.8678647	0.8899341	0.9930465	821
[9]	{lgagtq=L0}	=> {nptagq=N0}	0.7554092	0.8709328	0.8673565	0.9965571	803
[10]	{nptagq=N0}	=> {lgagtq=L0}	0.7554092	0.8643703	0.8739417	0.9965571	803

La liste commence par les quatre (04) règles de longueur 1, qui sont des règles sans antécédent. La première règle exprime simplement le fait que la base contient 88% de clients détenant un seul compte à vue ou pas du tout. D'une manière générale, les quatre (04) premières règles correspondent aux modalités qui dépassent 80% de la population. L'indice de confiance d'une règle sans antécédent est égal à son indice de support. Ce
 Passons à la cinquième règle : {lgagtq=L0} => {kvunbq=K0}. Elle signifie qu'un engagement à long terme égale à zéro(0) implique le fait d'avoir un nombre de compte à vue très faible. Si on se base sur le classement par ordre de support, c'est la règle d'association la mieux placée non seulement en termes de support mais aussi en termes de lift (=1.0286) et d'indice de confiance (=0.91). Son support 0.79 se lit indirectement dans le tableau croisé suivant :

	K0	K1
L0	844	78
L1	102	39

Le support vaut $844/1063 = 79,39\%$. Quant à l'indice de la règle, il vaut $844/922 = 0.91$ avec $922 = 844 + 78$. Il est très élevé car un peu rare sont les clients ayant à la fois un engagement à long terme positif tout en possédant un nombreux compte à vue. La sixième règle est aussi retenue et elle est l'inverse de la cinquième ; l'antécédent devient le conséquent avec les mêmes performances. Le lift commence à être inférieur à 1 à partir de la septième règle, donc on s'arrête là. En termes de maximisation de support donc, nous n'avons pas pu trouver les règles avec des lifts vraiment élevés. Voici comment nous affichons les cinq(05) règles au plus fort lift.

```
> inspect(sort(rules.all, by="lift", decreasing = TRUE) [1:5])
```

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{relatq=r0, xlgnbq=X0, ylvnbq=Y0, itavcq=I0}	=> {uemnbq=U0}	0.1034807	1	0.1034807	3.810036	110
[2]	{relatq=r0, xlgnbq=X0, ylvnbq=Y0, havefq=H0, itavcq=I0}	=> {uemnbq=U0}	0.1015992	1	0.1015992	3.810036	108
[3]	{relatq=r0, vienbq=V0, xlgnbq=X0, ylvnbq=Y0, itavcq=I0}	=> {uemnbq=U0}	0.1025400	1	0.1025400	3.810036	109
[4]	{relatq=r0, xlgnbq=X0, ylvnbq=Y0, lgagtq=L0, itavcq=I0}	=> {uemnbq=U0}	0.1015992	1	0.1015992	3.810036	108
[5]	{relatq=r0, vienbq=V0, xlgnbq=X0, ylvnbq=Y0, havefq=H0, itavcq=I0}	=> {uemnbq=U0}	0.1015992	1	0.1015992	3.810036	108

Il n'y a pas des règles reliées avec les règles précédentes. De même, le lift reste inchangé à 3.81 dans le top 05. Même si ces règles sont très fortes en termes d'indices de confiances et de lift, ces règles ne nous intéressent pas forcément au même titre que toute règle sur l'achat d'un produit, par exemple, n'intéresse pas nécessairement un service marketing. Puisque nous n'allons pas chercher des règles les plus informatives, nous nous limiterons sur des règles bien spécifique à CARVP.

Nous regardons maintenant les règles dont l'indice de confiance est le plus élevé, et nous affichons les dix règles au plus fort indice de confiance.

```
> inspect(sort(rules.all, by="confidence", decreasing = TRUE) [1:10])
```

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{lgagtq=L1}	=> {gagetq=G1}	0.1326435	1	0.1326435	2.410431	141
[2]	{vienbq=V1}	=> {havefq=H1}	0.1646284	1	0.1646284	3.099125	175
[3]	{xlgnbq=X2}	=> {uemnbq=U2}	0.1768579	1	0.1768579	2.372768	188
[4]	{ylvnbq=Y2}	=> {uemnbq=U2}	0.1787394	1	0.1787394	2.372768	190
[5]	{uemnbq=U0}	=> {ylvnbq=Y0}	0.2624647	1	0.2624647	2.295896	279
[6]	{uemnbq=U0}	=> {xlgnbq=X0}	0.2624647	1	0.2624647	1.855148	279
[7]	{gagetq=G0}	=> {lgagtq=L0}	0.5851364	1	0.5851364	1.152928	622
[8]	{havefq=H0}	=> {vienbq=V0}	0.6773283	1	0.6773283	1.197072	720
[9]	{endetq=E1, lgagtq=L1}	=> {gagetq=G1}	0.1241769	1	0.1241769	2.410431	132
[10]	{rocnbq=R1, lgagtq=L1}	=> {gagetq=G1}	0.1006585	1	0.1006585	2.410431	107

Les premières règles ont un indice de confiance de 100%. A titre d'exemple, la première règle affichée (141/141) est vraie pour les 141 dossiers d'après notamment le support et le tableau suivant :

	G0	G1
L0	622	300
L1	0	141

Si « lgagtq = L1 » alors « gagetq = G1 ». Autrement dit, compte tenu des codes préétablis dans l'annexe, un engagement à long terme positif implique un total des engagements en

francs positifs. Cette règle n'apporte rien de nouveau surtout dans une perspective de scoring puisque par définition un tel phénomène se reproduira toujours. Elle sert juste à vérifier la cohérence des données. Cela est confirmée par le fait qu'en septième règle, on rencontre son opposé exact. Il en est de même des règles qui s'ensuivent.

- La deuxième règle nous parle qu'un nombre de produits contrat à vie élevé implique un total des avoirs épargnes financières élevé, l'opposé de cette règle se situe dans le huitième règle ;
- Dans le troisième, un nombre de produit d'épargne logement élevé va de pair avec un nombre de produit d'épargne monétaire aussi élevé, son opposé se trouve dans la sixième règle ;
- Dans le quatrième, un montant des comptes sur le livret en francs élevé entraine un nombre de produit d'épargne monétaire aussi élevé, son opposé se situe dans la cinquième règle ;
- La neuvième et dixième ne sont que des cas particuliers de la première règle.

On commence se rendre compte que la détection des règles d'association doit comporter un travail de sélection des règles obtenues (qui pourrait être effectué automatiquement par le logiciel R). En même temps, on est satisfait de la cohérence des données. Entrons maintenant dans le cœur du travail qui consiste à prédire un item spécifique, il s'agit de « Coui » voir « Cnon ».

1.5 La possession de la carte visa premier : la situation socio-professionnelle des clients s'ajoutent aux facteurs généraux

C'est à ce stade que les paramètres « appearance » de la fonction « apriori() » vont nous intéresser. Nous indiquons que le conséquent (rhs) doit être « Cible=1 » et que tous les items peuvent figurer dans l'antécédent d'une règle (default='lhs').

Etant donné que l'ajout d'une forte contrainte sur le conséquent entraine une diminution du nombre de règles, on a fixé des seuils plus larges que précédemment sur l'indice de confiance (supérieur ou égal à 60%), le nombre de règles par item (inférieur ou égal à 5, y compris le conséquent). Le support reste inchangé puisque nous l'avons déjà fixé à la hausse. Les règles seront hiérarchisées par l'indice de confiance. Puisque nous aimerions promouvoir la vente de la carte visa premier.

Tableau 1 : Top 10 des règles d'associations de la possession de la carte visa premier

	lhs <chr>	rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>	count <int>
[1]	{sexeq=Shom, moyrvq=M2, dmvtpq=D2}	=> {carvp=Coui}	0.1025400	0.8449612	0.1213547	2.515949	109
[2]	{nptagq=N0, moyrvq=M2, dmvtpq=D2}	=> {carvp=Coui}	0.1081844	0.8333333	0.1298213	2.481326	115
[3]	{moyrvq=M2, dmvtpq=D2}	=> {carvp=Coui}	0.1185325	0.8289474	0.1429915	2.468266	126
[4]	{sexeq=Shom, rocnbq=R1, qsmoyq=Q2, moyrvq=M2}	=> {carvp=Coui}	0.1072437	0.7916667	0.1354657	2.357260	114
[5]	{sexeq=Shom, nptagq=N0, gagetq=G1, moyrvq=M2}	=> {carvp=Coui}	0.1232361	0.7844311	0.1571025	2.335715	131
[6]	{sexeq=Shom, moyrvq=M2, itavcq=I2}	=> {carvp=Coui}	0.1091251	0.7837838	0.1392286	2.333788	116
[7]	{sexeq=Shom, nptagq=N0, moyrvq=M2, itavcq=I2}	=> {carvp=Coui}	0.1015992	0.7826087	0.1298213	2.330289	108
[8]	{sexeq=Shom, qsmoyq=Q2, moyrvq=M2}	=> {carvp=Coui}	0.1232361	0.7797619	0.1580433	2.321812	131
[9]	{sexeq=Shom, moyrvq=M2, jnbjdc=J2}	=> {carvp=Coui}	0.1025400	0.7785714	0.1317027	2.318267	109
[10]	{sexeq=Shom, nptagq=N0, qsmoyq=Q2, moyrvq=M2}	=> {carvp=Coui}	0.1138288	0.7756410	0.1467545	2.309542	121

On note que les premières règles sont caractérisées par un indice de confiance très élevé (supérieur à 75%) et un lift relativement très élevé (supérieur à 2,40) qui signale l'apport heuristique de ces règles pour caractériser le conséquent, c'est-à-dire pour prédire la possession de la carte visa premier.

Les trois premières règles ont un indice de confiance de plus de 80%, ces règles sont :

- Le fait d'être homme, une moyenne des mouvements nets créditeurs des 3 mois en Kf très élevé, et l'âge du dernier mouvement pas assez récent puisqu'il est supérieur à 2,70 jours et pouvant s'étendre jusqu'à 7 jours environ, la combinaison de ces trois items explique la possession de la carte visa premier ;
- La deuxième règle est à peu près la même chose mais le fait d'être un homme est changé en nombre de carte point argent égale à zéro ;
- La troisième règle est la généralisation de ces deux premières règles.

Cette dernière est plus intéressante en un sens car son indice de confiance est presque aussi élevé de 83% environ, mais son support est bien plus grand égal à 11,85%. A peu près comme elle mais avec des règles de confiance un moins élevé relativement, les règles cinq(5), huit(08) et dix(10) se démarquent. Il s'agit de :

- L'antécédent comme la règle 01 mais avec du nombre de paiements par carte bancaire à M-1 nettement positif et pouvant aller de 1 à 69 fois, puis un total des engagements en francs aussi supérieur à zéro(0) ;
- En supprimant ces deux items et leur substituant par des moyennes des soldes moyens sur 03 mois élevé compris entre 7932 et 241827, nous obtenons la huitième règle ;
- La dixième n'est qu'un cas particulier de cette dernière avec le nombre de carte point argent nul en plus.

En se focalisant sur ces 10 premières règles, on voit que l'item « Coui » va de plus en plus avec le fait d'être un homme et une moyenne de mouvement nets créditeurs de trois mois élevés c'est-à-dire la somme des entrées d'argent est supérieure aux dépenses.

On peut multiplier les exemples, mais on voit déjà comment cette technique de détection des règles d'associations peut permettre de découvrir des règles correspondant à des profils très précis de nos clients. Dans le problème qui nous intéresse, certaines règles détectées pourraient être utilisées pour une campagne de marketing facilitée pour des profils plus assurés comme nous venons de le voir.

Les règles franchissant les filtres précédents pour le conséquent « carvp = Cnon » (non possesseurs) sont :

Tableau 2 : Top 10 des règles d'associations de la non-possession de la carte visa premier

	lhs <chr>	=>	rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>	count <int>
[1]	{sexeq=Sfem, facanq=f0, dmvtpq=D1}	=>	{carvp=Cnon}	0.1138288	1.0000000	0.1138288	1.505666	121
[2]	{sexeq=Sfem, kvunbq=K0, facanq=f0, dmvtpq=D1}	=>	{carvp=Cnon}	0.1025400	1.0000000	0.1025400	1.505666	109
[3]	{facanq=f0, dmvtpq=D1, boppnq=B1}	=>	{carvp=Cnon}	0.1119473	0.9916667	0.1128881	1.493119	119
[4]	{sexeq=Sfem, havefq=H0, opgnbq=O0, moyrvq=M0}	=>	{carvp=Cnon}	0.1081844	0.9913793	0.1091251	1.492686	115
[5]	{pcspq=Pemp, kvunbq=K0, lgagtq=L0, dmvtpq=D1}	=>	{carvp=Cnon}	0.1063029	0.9912281	0.1072437	1.492458	113
[6]	{famiq=Fseu, facanq=f0, lgagtq=L0, dmvtpq=D1}	=>	{carvp=Cnon}	0.1063029	0.9912281	0.1072437	1.492458	113
[7]	{kvunbq=K0, facanq=f0, dmvtpq=D1, boppnq=B1}	=>	{carvp=Cnon}	0.1034807	0.9909910	0.1044214	1.492101	110
[8]	{sexeq=Sfem, endetq=E0, havefq=H0, opgnbq=O0}	=>	{carvp=Cnon}	0.1373471	0.9864865	0.1392286	1.485319	146
[9]	{sexeq=Sfem, gagetq=G0, havefq=H0, opgnbq=O0}	=>	{carvp=Cnon}	0.1317027	0.9859155	0.1335842	1.484459	140
[10]	{sexeq=Sfem, gagetq=G0, havefq=H0, moyrvq=M0}	=>	{carvp=Cnon}	0.1279398	0.9855072	0.1298213	1.483844	136

D'une manière générale, même si les règles de prédiction de « Coui » sont largement satisfaisant, celles de « Cnon » sont bien meilleures. Rapidement, il n'est pas difficile de voir dans les deux premières règles les deux indices de confiances égales à 1, des événements sûrs à se produire. A part cela, les indices de confiances sont supérieurs à 98%, ce qui est vraiment excellents. La proportion des règles ayant un support nettement supérieur à 10 reste quand même à 4/10. Les interprétations se passent exactement de la même manière que les règles en Coui, il suffit de se référer à l'annexe.

Chapitre 2 : Classifieur bayésien naïf

Dans ce chapitre, nous décrire les points théoriques centraux de la méthode classifieur bayésien naïf (2.1). Par la suite, nous allons successivement lancer et interpréter le modèle (2.2), puis apprendre le classifieur à prédire les possesseurs et non possesseurs(2.3), enfin vérifier la capacité du modèle à généraliser(2.4)

2.1 Le théorème de Bayes et la classification probabiliste comme points théoriques centraux de la méthode

Le classifieur naïf de Bayes est un algorithme de classification probabiliste basé sur le théorème de Bayes. Il est utilisé pour prédire la probabilité qu'une donnée d'entrée appartienne à une classe particulière. Le classifieur se base sur l'hypothèse naïve selon laquelle les caractéristiques d'entrée sont conditionnellement indépendantes les unes des autres, sachant la classe.

Plus précisément, le classifieur naïf de Bayes utilise la probabilité conditionnelle pour estimer la probabilité qu'une donnée d'entrée appartienne à une classe spécifique, sachant les caractéristiques d'entrée. La formule de base utilisée par le classifieur est le théorème simple de Bayes suivant :

$$P(C|x) = \frac{P(x|C) * P(C)}{P(x)}$$

Où :

- $P(C|x)$ est la probabilité de la classe C, sachant les caractéristiques d'entrée x.
- $P(x|C)$ est la probabilité des caractéristiques d'entrée x, sachant la classe C.
- $P(C)$ est la probabilité de la classe C, indépendamment des caractéristiques d'entrée.
- $P(x)$ est la probabilité des caractéristiques d'entrée x, indépendamment de la classe.

Le classifieur naïf de Bayes suppose que toutes les caractéristiques d'entrée x (= x1, x2, x3, x4, ... xn) sont indépendantes les unes des autres, sachant la classe. En pratique donc, seul le numérateur nous intéresse, puisque le dénominateur ne dépend pas de C et les valeurs des caractéristiques x sont données ou en d'autres termes constants. Cela permet de simplifier la formule de base en supposant que $P(x|C)$ est le produit de la probabilité conditionnelle de chaque caractéristique, sachant la classe. Cette simplification conduit à l'équation suivante :

$$P(C|x) = \frac{P(C) * P(x_1|C) * P(x_2|C) * ... * P(x_n|C)}{P(x)}$$

Où xi est la ième caractéristique d'entrée.

Pour classer une donnée d'entrée, le classifieur calcule la probabilité $P(C|x)$ pour chaque classe C possible, et sélectionne la classe avec la probabilité la plus élevée.

D'une manière un peu plus particulière, il est tout à fait possible de reformuler cette propriété dans le langage admis des problèmes de classement. La variable binaire à expliquer est Y , on cherche à expliquer $Y = 1$ (comme dans la régression logistique), et les variables explicatives sont X_1, \dots, X_n . On a alors :

$$P(Y = 1|X_1, \dots, X_p) = \frac{P(X_1, \dots, X_p|Y = 1) P(Y = 1)}{P(X_1, \dots, X_p)}$$

En admettant naïvement l'indépendance des variables (conditionnellement à Y), on peut écrire :

$$P(X_1, \dots, X_p|Y = k) = p(X_1, \dots, X_p|Y = k, (X_1, \dots, X_p))$$

Par conséquent, la probabilité conditionnelle peut s'écrire de la forme suivante :

$$P(X_1, \dots, X_p|Y = k) = \prod_{i=1}^p P(X_i|Y = k), k = 0,1$$

D'où la formule appliquée par le classifieur bayésien naïf :

$$P(Y = 1|X_1, \dots, X_p) = \frac{P(Y = 1) \prod_{i=1}^p P(X_i|Y = 1)}{P(Y = 1) \prod_{i=1}^p P(X_i|Y = 1) + P(Y = 0) \prod_{i=1}^p P(X_i|Y = 0)}$$

En résumé, le classifieur naïf de Bayes est un algorithme simple mais efficace de classification probabiliste basé sur le théorème de Bayes et l'hypothèse naïve selon laquelle les caractéristiques d'entrée sont conditionnellement indépendantes les unes des autres, sachant la classe.

2.2 Lancement et analyse intuitive du modèle Classifieur naïves de bayes

Le premier code qu'on a utilisé est la fonction `set.seed` pour initialiser le générateur de nombres aléatoires, ce qui permet de reproduire les mêmes résultats chaque fois que nous exécutons les codes. Ensuite, on utilise la fonction « `sample()` » pour sélectionner aléatoirement 2/3 des observations de la base de données `vispremh` pour créer un ensemble d'apprentissage, puis il utilise l'ensemble des observations restantes pour créer un ensemble de test. Cela ressemble à une bonne première étape pour diviser votre ensemble de données en un ensemble d'apprentissage et un ensemble de test. Le paramètre `replace` a été défini à « `F` » pour éviter le remplacement des observations échantillonnées.

Nous avons créé aussi un modèle Naive Bayes avec la fonction « naiveBayes() » du package e1071 en utilisant la variable cible carvp et toutes les autres variables explicatives de la base de données apprent. Pour afficher les détails du modèle, nous pouvons simplement taper le nom de l'objet Naive_Bayes_Model dans la console R :

```
Naive Bayes Classifier for Discrete Predictors
```

```
Call:
```

```
naiveBayes.default(x = X, y = Y, laplace = laplace)
```

```
A-priori probabilities:
```

```
Y
```

	Cnon	Coui
Y	0.6624294	0.3375706

```
Conditional probabilities:
```

```
famiq
```

Y	Fcou	Fseu
Cnon	0.5031983	0.4968017
Coui	0.6359833	0.3640167

```
sexeq
```

Y	Sfem	Shom
Cnon	0.4904051	0.5095949
Coui	0.1757322	0.8242678

Comme nous pouvons le voir dans cet extrait, en particulier, la probabilité pour qu'un client détienne la carte visa premier est de l'ordre de 33,75%. Ce nombre est égal à 357/1063, c'est-à-dire le nombre des possesseurs divisé par le nombre total des clients. De même, les non-possesseurs s'élèvent à 66,24%. L'affichage aussi nous montre les statistiques descriptives pour chaque variable explicative, ainsi que les probabilités a priori et les probabilités conditionnelles pour chaque modalité de la variable cible carvp.

Pour les tableaux de contingence, ils montrent les proportions de chaque niveau des variables catégorielles (famiq et sexeq par exemple) pour chaque classe de la variable cible (carvp). Les cellules d'en haut correspondent à la classe "non" de la variable cible et les cellules d'en bas correspondent à la classe "oui". Par exemple, la première table montre que pour la classe "non" de la variable cible, environ 50,3% des clients sont mariés (Fcou pour "famille mariée") et environ 49,7% sont séparés de corps (Fseu pour "famille séparée de corps") ce qui fait 100% au total. Pour la classe "oui", la majorité des clients (environ 82,4%) sont des hommes (Shom pour "sexe homme").

D'une manière intuitive, ces résultats peuvent être utilisés pour interpréter la façon dont les variables explicatives sont associées à la variable cible. Par exemple, si la variable sexeq avait montré une proportion plus élevée de femmes à hauteur de 75% dans la classe "oui" de la variable cible, on pourrait conclure que le genre est associé à une plus forte probabilité d'être possesseur. Les modalités qui respectent ces conditions de grande proportion sont :

Tableau 3 : Liste des modalités à fortes proportion

variables	Coui	Cnon
sex	shom 82%	
	sfem 17%	
pcspq	Pcad 46%	
Kvnumb	KO 77%	KO 94,45%
		K1 5%
rocnbq	R1 70%	

2.3 Très bonne performance de l'apprentissage prédictive du modèle Classifieur naïves de bayes

Cependant, ces conclusions doivent être interprétées avec prudence et doivent être confirmées par des analyses plus approfondies ultérieurement.

Nous allons prédire les valeurs de la variable réponse pour un ensemble de données de test ou d'apprentissage. Le paramètre « type » permet de spécifier le format de la sortie de la fonction de prédiction « predict() ». Dans notre base des données, la variable réponse est carvp, qui est une variable binaire. Le modèle entraîné est un modèle de classification naïve bayésienne bien évidemment. Lorsque type="raw" comme dans notre cas ici, la fonction predict renvoie un tableau avec les probabilités de chaque observation appartenant à chaque classe (Cnon et Coui dans ce cas). Ainsi, Pred1 correspond aux probabilités de chaque observation appartenant à chaque classe, tandis que Pred2 correspond aux classes prédites pour chaque observation. Normalement, il devrait y avoir 1063 observations comme le nombre des clients, mais nous allons utiliser un extrait :

```

      Cnon      Coui
[1,] 0.616997257 0.3830027
[2,] 0.803660373 0.1963396
[3,] 0.003622139 0.9963779
[4,] 0.006711946 0.9932881
[5,] 0.132659369 0.8673406
[6,] 0.008814230 0.9911858

```

Pour le client 1, il a 61% de chance d'être un non-possesseur et bien sûr 38% environ de détenir la carte visa premier. Dans cet extrait en général, nous voyons bien qu'il y a beaucoup de chance pour les clients de détenir la carte. Pour les clients 03, 04 et 06, leur chance s'élèvent à 99% voir plus. C'est presque un évènement sûr que ces clients détiennent la carte.

Dans l'objet Pred2 maintenant, les quelques résultats sont les suivants :

```

[1] Cnon Cnon Coui Coui Coui Coui
Levels: Cnon Coui

```

On voit que notre modèle a prédit la classe **Coui** pour les individus possédant une probabilité élevée dans d'appartenir à cette classe. Il s'agit surtout de la classe 03, 04 et 06. Avec une probabilité égale à 86% par rapport à ce contexte, le client 05 figure aussi parmi les prédits de détenir la carte. Afin de voir à quel point le modèle prédit correctement les

valeurs de carvp d'une manière globale, nous allons afficher une table de contingence entre les vraies valeurs de la variable cible carvp et les prédictions faites par le modèle Naive_Bayes_Model sur l'ensemble d'apprentissage apprend. C'est la matrice de confusion suivante :

	Pred2	
	Cnon	Coui
Cnon	398	71
Coui	66	173

La matrice de confusion montre que sur l'échantillon d'apprentissage, le modèle a correctement classé 398 clients dans la catégorie "Cnon" et 173 clients dans la catégorie "Coui". En revanche, il y a eu 71 clients classés à tort dans la catégorie "Coui" et 66 clients classés à tort dans la catégorie "Cnon". Pour obtenir le taux de classification globale, il faut sommer les éléments diagonaux de la table de confusion et diviser par le nombre total d'observations. Dans ce cas, le taux de classification globale sur l'échantillon d'apprentissage est : 80,6% $(=(398+173)/708)$ ⁹. Quant au taux de mauvais classement, il est légèrement inférieur à 20% donc, elle vaut 19,3% plus précisément.

Il nous faut aussi calculer la courbe de ROC pour analyser encore une fois la performance du modèle :

```
> auc <- performance(Pred3,"auc")
> performance(Pred3,"auc")@y.values[[1]]
[1] 0.8809985
```

La performance « (Pred3, "auc")@y.values[[1]] » renvoie une valeur de 0.88, alors cela signifie que l'aire sous la courbe ROC (AUC) pour la prédiction dans Pred3 est de 0.88. Cela indique que le modèle a une bonne capacité à distinguer entre la classe positive et celle négative. En effet, une AUC de 1 représente une performance parfaite ou une distinction sans faille tandis qu'une AUC de 0.5 représente une performance aléatoire.

Par conséquent, une AUC de 0.88 plus proche de 1 indique que le modèle a une performance relativement élevée pour cette tâche de classification binaire. Mais a-t-il une capacité de généralisation élevée ? C'est ce qui nous anime à faire une analyse de l'échantillon test.

2.4 Une généralisation du modèle d'apprentissage acceptable

Sans vouloir entrer dans les détails, on retrouve analogiquement la matrice de confusion liée à l'échantillon test suivante :

	Predi2	
	Cnon	Coui
Cnon	194	43
Coui	30	88

⁹ 708 le nombre d'observations dans l'échantillon apprentissage.

Comparativement, le modèle établi sur l'échantillon test a correctement classé 194 clients (au lieu de 398) dans la catégorie « Cnon » et 88 dans la classe positive (au lieu de 173). En revanche, il y a eu 43 clients (au lieu de 71) classés à tort dans la catégorie "Coui" et 30 clients (au lieu de 66) classés à tort dans la catégorie "Cnon". Cette comparaison en termes absolu n'apporte en rien finalement. Puisque l'effectif de l'échantillon test représente à peu 1/3 de la population seulement. Il faut raisonner donc en termes relatifs.

En conséquence, la classification globale de l'échantillon test est estimé à 79,4%. Ce qui est légèrement inférieur à celle de l'échantillon apprentissage. Logiquement, son taux de mal classé dépasse sensiblement les 20%, il est égal à 20,5% exactement.

Pour obtenir le taux de classification globale, il faut sommer les éléments diagonaux de la table de confusion et diviser par le nombre total d'observations. Dans ce cas, le taux de classification globale sur l'échantillon d'apprentissage est : 80,6% $(=(398+173)/708)$ ¹⁰. Quant au taux de mauvais classement, il est légèrement inférieur à 20% donc, elle vaut 19,3% plus précisément.

¹⁰ 708 le nombre d'observations dans l'échantillon apprentissage.

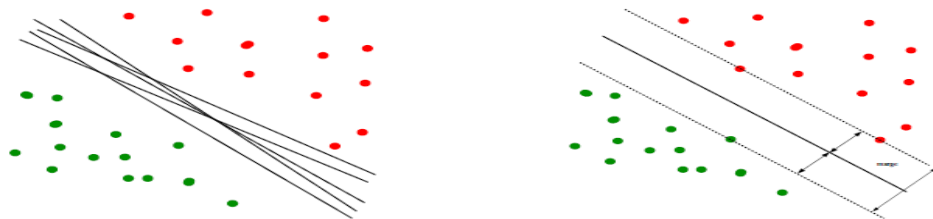
Chapitre 3 : Les Supports Vecteurs Machines

Quelles sont les bases théoriques de la méthode SVM ? Comment procède-t-on pour implémenter cet algorithme ? Quels sont les résultats d'analyses ? Telle est la problématique abordée successivement dans la section 3.1, 3.2 puis le reste.

3.1 SVM : une méthode fondée sur l'hyperplan et par nature souple

Les SVM sont une famille d'algorithmes dédiés aux problèmes de régression et de classification supervisée. Nous présenterons la démarche dans le cas de la classification supervisée avec une variable à expliquer Y binaire. On désigne par $X = (X_1, \dots, X_P)$ le vecteur des variables explicatives. Dans ce cadre, l'approche SVM consiste à trouver un hyperplan de l'espace des variables explicatives qui sépare « au mieux » les observations. Une illustration est proposée sur un exemple jouet sur les graphiques suivants (2 variables explicatives, le label Y est représenté par la couleur).

Figure 2 : Illustration du mécanisme du SVM linéaire



Les données sont ici linéairement séparables, on voit en effet sur le graphique de gauche qu'il existe une infinité d'hyperplans séparateurs. Les SVM consistent à trouver l'hyperplan qui maximise la distance de l'observation la plus proche de l'hyperplan, c'est-à-dire la largeur du couloir entre les observations des deux classes (figure de droite). Cette maximisation de la marge diminue la complexité du modèle obtenu et lui assure une meilleure généralisation. Cette distance maximale est appelée marge et les points qui se trouvent sur la marge sont appelés vecteurs supports.

D'une manière générale donc, rechercher la séparation optimale revient à chercher un hyperplan d'équation $a \cdot x + b = 0$ ($a \cdot x$ est le produit scalaire de a et x) remplissant simultanément deux conditions :

- Il sépare bien les classes à discriminer A et B (bon ajustement du modèle), au sens où la fonction définie par $f(x) = a \cdot x + b > 0$ si et seulement si $x \in A$, et $f(x) \leq 0$ si $x \in B$.
- Il est le plus loin possible de toutes les observations (robustesse du modèle), sachant que la distance d'une observation x à l'hyperplan est $|a \cdot x + b| / \|a\|$

La marge, qui est par définition la largeur du couloir, vaut $2 / \|a\|$ et doit être maximisée. Cette contrainte explique l'expression de « séparateur à vaste marge ».

Bien entendu, ce dessin représente un cas très particulier où les données sont séparables, ce qui n'est pas la situation générale. Dans le cas non séparable et c'est souvent le cas dans la pratique, on autorise des points à être dans la marge ou à être mal classés, c'est-à-dire du mauvais côté de l'hyperplan. Afin de quantifier l'erreur de classement, pour chaque individu ou client, on mesure la distance le séparant de la frontière de la marge du côté de sa classe. On normalise ensuite cette distance en la divisant par la demi-marge et on obtient un terme δ_i , appelé « variable ressort ». Un individu est à l'intérieur de la marge si $\delta_i \in]0,1]$ et une erreur du modèle est un individu pour lequel $\delta_i > 1$. La somme de tous les δ_i représentent l'ensemble des erreurs de classement. A la contrainte de la maximisation de la marge s'ajoute alors simultanément une contrainte de minimisation de la somme des erreurs et l'hyperplan séparant les classes doit satisfaire deux conditions¹¹ :

- Pour tout i , $y_i(a \cdot x_i + b) \geq 1 - \delta_i$
- $\frac{1}{2} \|a\| + C \sum \delta_i$ est minimum (marge maximale)

Si $\delta_i > 0$, la condition $y_i(a \cdot x_i + b) \geq 1 - \delta_i$ est un relâchement par rapport à la condition initiale $y_i(a \cdot x_i + b) \geq 1$. Cela se produit non seulement pour les observations mal classées ($\delta_i > 1$), mais aussi pour celles qui sont à l'intérieur de la marge ($\delta_i \in]0,1]$).

Dans la seconde condition, on affecte aux points mal classés le coût C lié à la distance à l'hyperplan séparateur multiplié par un paramètre d'échelle à calibrer. C pénalise les erreurs et permet de contrôler l'ajustement du modèle. Plus C est grand et plus grande est la sensibilité des erreurs et des individus à l'intérieur de la marge, un coût plus grand diminuera sur l'échantillon d'apprentissage, avec moins d'erreurs de classement. Il convient donc de bien choisir C pour garantir un bon compromis entre l'ajustement et la robustesse.

Les SVM ne se limitent pas à vrai dire à la séparation purement linéaire et dans un espace à deux (02) dimensions. En effet, les SVM sont une sorte de généralisation de l'analyse discriminante linéaire, dans laquelle la séparation linéaire des classes, quand elle n'est pas possible dans l'espace initial, s'opère par prolongement dans un espace de dimension plus grande¹² sans perte d'efficacité. Ce point très crucial repose sur le fait que l'algorithme permettant d'obtenir l'hyperplan de marge maximale n'a besoin que des produits scalaires entre données pour fonctionner. Plus précisément, il est possible que les données ne soient pas linéairement séparables mais qu'elles le deviennent après leur avoir appliqué une transformation pertinente. Une telle transformation revient à plonger les données dans un nouvel espace, souvent de plus grande dimension que l'espace original. Il s'avère qu'appliquer la SVM linéaire sur les données transformées revient à appliquer une SVM sur les données initiales en utilisant un noyau à la place du produit scalaire usuel. Cette astuce,

¹¹ Etant donné les points (x_i, y_i) , avec $y_i = 1$ si x_i est dans A et $y_i = -1$ si x_i est dans B, trouver l'hyperplan optimal $a \cdot x + b = 0$ équivaut un couple (a, b) satisfait les deux conditions susmentionnés sans δ_i et $C \sum \delta_i$.

¹² Théoriquement, cette dimension peut aller jusqu'à l'infinie.

dite astuce du noyau, incite à définir directement le noyau (le produit scalaire après transformation) et non la transformation explicite.

3.2 Directive d'implémentation de l'algorithme SVM efficacement

On a bien compris qu'à part le SVM linéaire, plusieurs types de SVM à noyaux existent. L'approche classique consiste à établir le meilleur SVM linéaire, d'essayer plusieurs noyaux et à se donner une grille de valeurs pour les paramètres quantitatifs ou des comparaisons successives. Pour chaque jeu de paramètres et pour chaque noyau, on estime ensuite un critère (par exemple par validation croisée) et on choisit le jeu de paramètres qui optimise ce critère estimé (Husson, 2018). Pour notre cas, après avoir lancé le SVM à noyau linéaire, nous proposons les trois noyaux les plus utilisés dans le data mining (Tuffery, 2017) :

- Le noyau radial gaussien ;
- Le noyau sigmoïde ;
- Le noyau polynomial ;
- Le noyau polynomial et le noyau radial (le noyau linéaire étant un cas particulier du noyau polynomial).

Tuffery affirme aussi que le SVM à noyau linéaire a une efficacité inégalable quand il est lancé après avoir fait une Analyse des Correspondances Multiples (ACM). Nous essayerons cette technique en dernier ressort.

3.3 SVM à noyau linéaire : un modèle suffisamment performant

On sépare l'échantillon en un échantillon d'apprentissage de taille 693 (environ 65% de l'ensemble des clients dans la base des données) et un échantillon de validation de taille 370. L'échantillon d'apprentissage sera utilisé pour construire les algorithmes de SVM et sélectionner leurs paramètres, celui de validation pour estimer les performances de ces algorithmes et vérifier qu'ils ne font pas de sur-ajustement.

Il existe plusieurs fonctions sur R permettant de faire des SVM. Les deux plus utilisées sont certainement la fonction `svm` du package `e1071` et la fonction `ksvm` du package `kernlab`. Nous présentons ici la première option. La syntaxe est similaire aux autres fonctions de régression ou de discrimination : on utilise une formule pour indiquer la variable à expliquer et les variables explicatives. Il faut faire attention aux paramètres : par défaut, le paramètre de coût C vaut 1.


```
Call:
svm(formula = carvp ~ ., data = vispremh[id, vars], kernel = "linear", probability = TRUE, cost = 1)
```

```
Parameters:
  SVM-Type:  C-classification
  SVM-Kernel: linear
  cost: 1
```

```
Number of Support Vectors: 201
```

```
( 100 101 )
```

```
Number of Classes: 2
```

```
Levels:
Cnon Coui
```

La sortie renseigne sur :

- le type de SVM ajusté : C-classification, puisque la variable à expliquer est Carvp et est qualitative, SVM se comporte donc comme une analyse discriminatoire en particulier ;
- le paramètre cost, qui correspond au paramètre de coût C, vaut 1 ;
- la transformation des données effectuées : ici le noyau est linéaire, les variables sont conservées en l'état ;
- le nombre de vecteurs supports (Support Vectors), qui est le nombre de données à conserver pour calculer la fonction de classification : ici il vaut 201, ce qui signifie qu'il suffit de mémoriser 201 exemples pour calculer la solution et non pas les 693 clients de la base d'apprentissage ;
- la valeur de la fonction optimisée pour obtenir les vecteurs supports à l'optimum (cette valeur est rarement utile) ;
- Le nombre des modalités de la variable expliquée égale à 02 : Coui et Cnon.
- L'erreur de classification n'est pas calculée sur les données d'apprentissage.

Les indices des vecteurs supports comptés précédemment peuvent être affichés. Mais ce n'est pas vraiment important. Ce qui nous intéresse ce sont les coefficients de ces vecteurs :

```
> head(svm1in$coefs)
      [,1]
[1,]    1
[2,]    1
[3,]    1
[4,]    1
[5,]    1
[6,]    1
```

Il s'agit en fait des quantités $a_i y_i$ qui intègrent le signe de y_i , et il y a donc une répartition équitable de 100 coefficients positifs et 100 coefficients négatifs, qui correspondent aux deux frontières de la marge. On vérifie que ces quantités sont toutes inférieures ou égales en valeur absolue au paramètre de cout, ici égale à 1. Pour avoir le modèle le plus

parcimonieux possible, on souhaite avoir le plus petit nombre de vecteurs de supports, ce qui est d'autant plus facile que les classes à discriminer sont bien séparées. On constate que la plupart des coefficients non nuls ai valent 1 et on rappelle que ces coefficients égaux au cout (=1) correspondent aux vecteurs supports mal classés ou dans la marge. On peut rapporter les 216 vecteurs supports aux 693 individus de l'échantillon d'apprentissage. Ce qui nous donne un rapport de 31% environ. Cette valeur n'est pas vraiment habituelle. Puisque normalement, le ratio est compris entre 33% et 76%, mais le plus souvent entre 50% et 70% selon Xindong Wu et Vipin Kumar (2009). Essayons de voir aussi les erreurs.

Les prédictions du modèle SVM sur l'échantillon d'apprentissage sont contenues dans « svm\$fit » et on peut les croiser avec les valeurs réelles de la variable à expliquer :

```
> table(svm$fit, vispremh[id,"carvp"])
```

	Cnon	Coui
Cnon	438	44
Coui	26	184

On voit que sur les 464 mauvais dossiers, 438 sont prédits comme tels, et que sur les 228 bons dossiers, 184 sont prédits comme tels. Le taux d'erreur est donc de 10% $(=(26+42)/659)$.

Cette valeur peut être comparée au taux d'erreur par validation croisée x-fold dont le calcul est déclenché si l'on ajoute cross = x à « svm() ». Ici, pour x=10, on trouve un taux d'erreur d'environ 13%. Largement inférieur à cette dernière, le taux d'erreur du modèle est acceptable.

Pour appliquer le modèle SVM à un autre jeu de données notamment celui du test, on utilise la fonction « predict() », qui renvoie ici, dans notre problème de classement à deux classes, un vecteur prédit. Elle renvoie aussi des valeurs de la fonction de score si l'on a spécifié l'option « decision.value = TRUE », ou les probabilités correspondantes à chaque classe si l'on a spécifié l'option « probability = TRUE » comme ici. On extrait les probabilités d'appartenance à chaque classe puis on calcule le taux d'erreur et l'aire sous la courbe de ROC.

```
> test$svm = predict(svm, test, probability = T)
> table(test$svm,test$carvp)

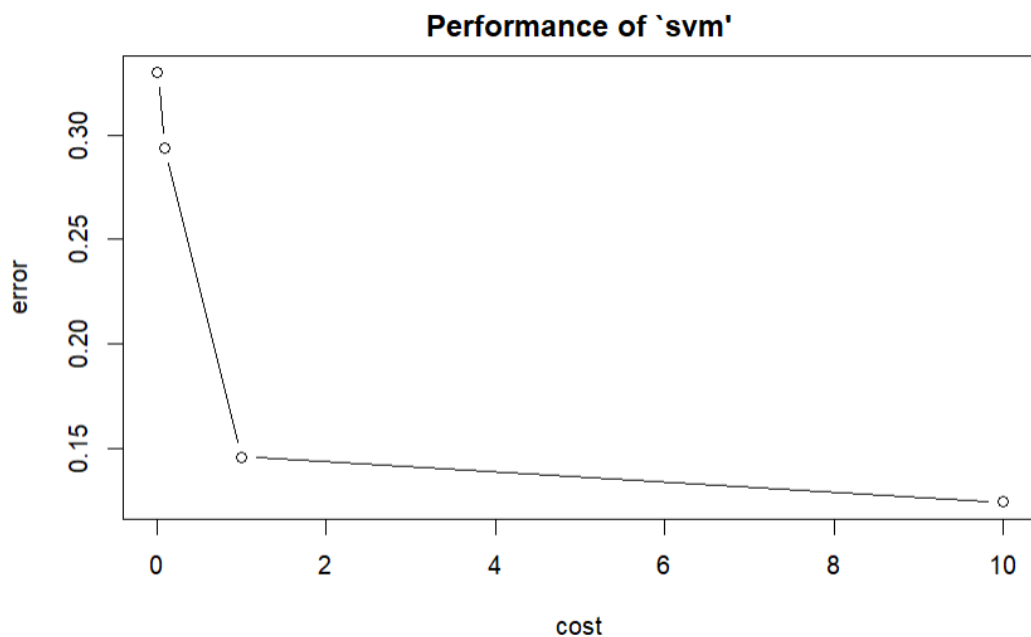
      Cnon Coui
Cnon  233   34
Coui    9   95
> mean(test$svm != test$carvp)
[1] 0.115903
> pred <- prediction(attr(test$svm,"probabilities"),[1], test$carvp,label.ordering=c("Cnon", "Coui"))
> performance(pred,"auc")@y.values
[[1]]
[1] 0.9525915
```

Avec un taux d'erreur de 10,78% et une performance globale de 95,25%, le SVM sur l'échantillon test s'avère très bon. On peut de la même façon appliquer la démarche à l'échantillon d'apprentissage et extraire les probabilités pour mesurer l'aire sous la courbe

ROC en apprentissage : elle vaut non pas exactement la même chose, mais avec une valeur plus ou moins proche de 0,9465. Son léger écart avec la valeur en test nous signale peut-être la présence de sous-apprentissage, et suggère d'augmenter la valeur du paramètre de cout pour augmenter la capacité de généralisation du modèle.

La fonction « `tune.svm()` » cherche la valeur de paramètre qui minimise le taux d'erreur de classement. Cette recherche se fait essentiellement par validation croisée de la façon suivante :

Figure 3 : Performance du SVM linéaire



Nous avons testé les valeurs du paramètre de cout comprises entre 10^{-3} et 10, et la valeur minimisant l'erreur est 10 mais non pas 1 (voir graphique ci-dessus). A part cela, nous pouvons aussi chercher à optimiser le paramètre sur la base de test. Pour cela, nous spécifions l'échantillon de test par « `validation.x` » pour les variables explicatives et « `validation.y` » pour la variable à expliquer, et nous précisons dans « `tune.control` » les paramètres d'échantillonnage. Le choix ici `tune.control` (`sampling` = « `fix` », `fix` = 1) signifie qu'on travaille avec un échantillon de test dont on prend 100% des observations. Le résultat nous montre un cout égal à 1 cette fois-ci. Quelle valeur doit-on prendre alors : 1 ou 10 finalement ?

Parameter tuning of 'svm':

- sampling method: fixed training/validation set

- best parameters:

```
cost
  1
```

- best performance: 0.1078167

- Detailed performance results:

```
cost      error dispersion
1 1e-03 0.3477089      NA
2 1e-02 0.1698113      NA
3 1e-01 0.1266846      NA
4 1e+00 0.1078167      NA
5 1e+01 0.1078167      NA
```

Pour trancher finalement, nous allons recomparer les aires sous la courbe ROC sur les échantillons d'apprentissage et de test, obtenues avec des paramètres de cout égaux aux deux valeurs.

Tableau 4 : Performance du SVM linéaire

Cout	AUC_apprentissage	taux_erreur_apprentissage	AUC_test	taux_erreur_test
1	0.946565	0.1026012	0.9525915	0.1024259
10	0.947652	0.1040462	0.9509257	0.1105121

Logiquement, c'est le cout le plus fort qui assure une meilleure généralisation puisqu'il y a un sous-ajustement du modèle. C'est aussi le cout le plus fort qui, pénalisant plus les erreurs de classement lors de l'apprentissage, procure un surplus d'ajustement. La robustesse est largement acceptable puisque ses taux d'erreurs sont inférieurs au seuil du taux d'erreur par validation croisée x-fold (13%).

Même lorsque les populations à discriminer sont relativement bien séparées, la séparation peut ne pas être linéaire. On cherche alors, au moyen d'une transformation non linéaire, à passer de l'espace d'origine dans un espace de dimension plus grande mais doté d'un produit scalaire et où existe une séparation linéaire. En conséquence, après avoir testé un SVM à noyau linéaire, nous allons passer au noyau radial.

3.4 Application du SVM à noyau radiale gaussien : nettement inférieur à celle du SVM linéaire

Dans un noyau radial, au paramètre de cout à régler pour le SVM à noyau linéaire s'ajoute un second paramètre, le paramètre « γ ». Nous pouvons le choix de ces paramètres. Nous le faisons là encore au moyen de la fonction « `tune.svm()` » en faisant varier gamma dans le vecteur (0.001,0.01,0.1,1) et le coût entre 1 et 10 avec un pas de 0.1. Nous cherchons une optimisation sur l'échantillon de test plutôt que par validation croisée.

Parameter tuning of 'svm':

- sampling method: fixed training/validation set

- best parameters:

```
gamma cost
0.1 1.3
```

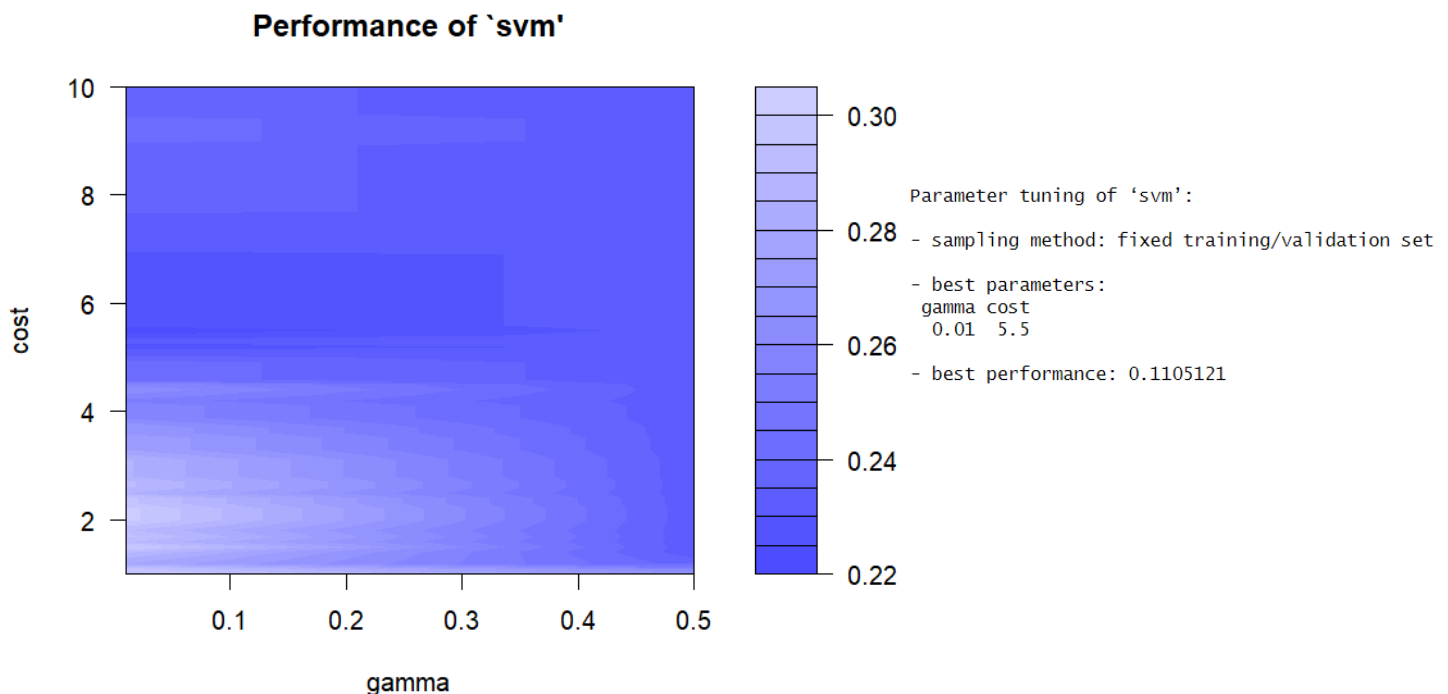
- best performance: 0.09433962

- Detailed performance results:

	gamma	cost	error	dispersion
1	0.001	1.0	0.34770889	NA
2	0.010	1.0	0.15094340	NA
3	0.100	1.0	0.10242588	NA
4	1.000	1.0	0.35040431	NA
5	0.001	1.1	0.34770889	NA

Gamma = 0.1 et cost = 1.3 sont les meilleurs paramètres du modèle selon les résultats. Nous avons affiné la recherche de gamma en le faisant varier entre 0.01 et 0.5 cette-fois avec un pas de 0.01(et le cout entre 0.1 et 5), et nous avons trouvé un optimum légèrement différent. La figure ci-dessous représente l'erreur en fonction des deux paramètres, et nous voyons qu'elle est minimale pour des valeurs plutôt petites de gamma et une valeur du cout visuellement entre 5 et 6.

Figure 4 : Performance du SVM radial gaussien



Si l'on retient les paramètres gamma = 0.01 et cost = 5.5, voici le pouvoir discriminant obtenu sur l'échantillon de test :

```
> performance(pred, "auc")@y.values[[1]]
[1] 0.9218568
```

L'aire sous la courbe ROC atteint une valeur nettement supérieure à 0.90, elle est égale à 0.92. Malgré cela, elle est moins bien qu'avec SVM linéaire atteignant les 0.9525.

3.5 SVM sigmoïde pratiquement à égalité de performance que SVM radial gaussien

Après avoir lancé les SVM à noyau radiale Gaussien, il nous faut aussi tester les SVM à noyau sigmoïde

Parameter tuning of 'svm':

- sampling method: fixed training/validation set

- best parameters:

```
gamma cost  
0.001 69.5
```

- best performance: 0.1428571

- Detailed performance results:

	gamma	cost	error	dispersion
1	1e-04	1.0	0.3423181	NA
2	5e-04	1.0	0.3423181	NA
3	1e-03	1.0	0.3423181	NA

Avec les paramètres $\gamma = 0.001$ et $\text{cost} = 69.5$, voici le pouvoir discriminant obtenu sur l'échantillon de test.

```
> performance(pred, "auc")@y.values[[1]]  
[1] 0.9171453
```

L'aire sous la courbe de ROC vaut à peu près la même chose que celle obtenue dans la forme gaussienne précédemment. Mais la mise en œuvre du modèle est plus simple et nécessite donc moins de temps de traitement.

3.6 SVM à noyau polynomial moins performant que le SVM à noyau linéaire

Le dernier noyau SVM testé sera le noyau polynomial. Nous ne testerons que les noyaux de degré 2 ; au-delà, la complexité du SVM augmente et il ne classe pas nécessairement mieux les individus (Tuffery, 2017).

Comme précédemment, nous rechercherons la combinaison du cost et du paramètre γ minimisant l'erreur sur l'échantillon de test. Nous avons décidé d'afficher directement le meilleur modèle, et non la liste procurée de toutes les combinaisons des valeurs du cost et de γ avec les erreurs correspondants. Ce modèle est le suivant :


```
Call:
best.svm(x = carvp ~ ., data = vispremh[id, vars], degree = 2,
  gamma = seq(0.01, 1, by = 0.01), cost = seq(0.1,
    10, by = 0.1), kernel = "polynomial", validation.x = test[,
    -13], validation.y = test[, "carvp"], tunecontrol = tune.control(sampling = "fix",
    fix = 1))
```

```
Parameters:
  SVM-Type:  C-classification
  SVM-Kernel: polynomial
    cost:    0.3
  degree:    2
  coef.0:    0
```

```
Number of Support Vectors: 268
```

Le meilleur modèle SVM polynomial a pour coût 0.3 et pour valeur de gamma 0.29. Pour l'instant, nous avons laissé à 0 la valeur de la constante « coef0 » intervenant dans la formule du noyau.

$$k(x, x') = (x * x' + coef0)^2$$

Le cout très faible peut assurer une très bonne généralisation du modèle et on peut le vérifier en calculant l'aire sous la courbe de ROC sur l'échantillon de test comme d'habitude.

```
> performance(pred, "auc")@y.values[[1]]
[1] 0.9139183
```

On retrouve donc une valeur 91,39%. A la surprise générale, la valeur du meilleur modèle SVM polynomial n'est pas le plus élevé. Or, apriori, ce modèle prédomine normalement les autres si l'on se base des différents résultats antérieurs y compris dans des situations différentes que la reconnaissance d'image ou de la voix (Tuffery, 2017). Comment pouvons-nous expliquer ce constat ?

En réalité, sur les prédicteurs de base, c'est le noyau polynomial qui assure généralement le pouvoir discriminant. Mais avec une l'utilisation des données vispremh contenant déjà exhaustivement des variables qualitatives y compris des variables quantitatives discrétisées et le regroupement de modalités des variables qualitatives, la non-linéarité de la réponse par rapport aux variables explicatives continues sont prises en compte. Ce qui explique le ROC du modèle SVM linéaire élevé et dans une certaine mesure le sigmoïde. En conséquence, les résultats nous paraissent pertinents. Il faut maintenant établir la grille de score avec le modèle linéaire.

3.7 SVM linéaire après réduction de dimension

Un moyen classique et efficace de réduire la dimension est d'effectuer une analyse factorielle puis d'élaborer le modèle SVM sur les axes factoriels. Comme dans le rapport déjà fait, nous réalisons une analyse des correspondances multiples sur les facteurs du data frame portant sur les variables qualitatives dont on a regroupé les modalités au vu de leurs

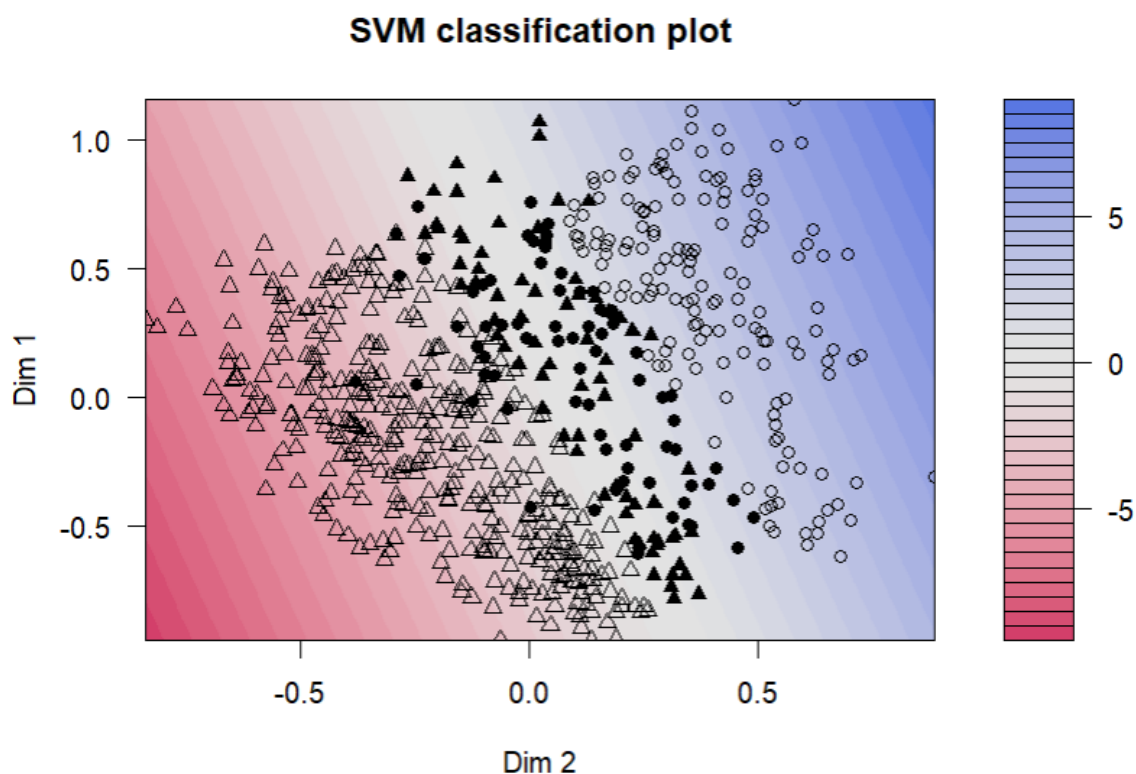
effectifs surtout, les variables quantitatives ont été discrétisées. L'analyse des correspondances multiples porte sur les 24 variables, la 13^è étant la variable à expliquer « carvp ». Nous ne revenons plus sur les explications de ces axes.

Nous avons choisi 39 axes ($ncp = 39$) pour la raison suivante. On sait que le nombre de valeurs propres non triviales est égale au nombre d'axes factoriels, égal à la différence entre le nombre de modalités et le nombre de variables. On calcule ainsi le nombre de modalités de chaque variable. On en déduit la différence entre le nombre de modalités et le nombre de variables qui est égale à 39.

```
> sum(modal) - ncol(vispremh[, -13])  
[1] 39
```

En deux dimensions, on peut représenter les observations des deux classes (distinguées par leur forme) et les points supports (formes pleines).

Figure 5 : SVM linéaire appliqué aux 02 axes factoriels



Nous voyons très bien la distinction entre les deux classes en triangle et en rondes. Au milieu, il y a les points supports.

On applique ce modèle à l'échantillon de test et on mesure l'aire sous la courbe ROC.

```
> performance(pred, "auc")@y.values[[1]]  
[1] 0.9516563
```

Cette aire élevée obtenue avec les deux axes et un coût = 10 nous incite à rechercher le nombre d'axes factoriels et le coût optimaux. Une fonction $f(i)$ est créée pour retourner l'aire sous la courbe ROC en test du modèle SVM à noyau linéaire construit sur les vecteurs x et y avec un coût = i .

Nous faisons ensuite varier le nombre n d'axes entre 2 et 13, le coût entre 1 et 20 (avec un pas de 0,1), et nous recherchons pour chaque n l'aire maximale et le coût correspondant. Bien sûr, la solution optimale ne se trouve pas dans la séquence axe1, axe1 + axe2, axe1+axe2+axe3,etc... et nous aurions pu tester toutes les combinaisons. Mais les calculs auraient été beaucoup très longs.

```
nb facteurs = 2  AUC test max = 0.9573926 coût = 1.5
nb facteurs = 3  AUC test max = 0.9594694 coût = 1
nb facteurs = 4  AUC test max = 0.9596369 coût = 5.4
nb facteurs = 5  AUC test max = 0.9595029 coût = 2.1
nb facteurs = 6  AUC test max = 0.9715951 coût = 1.2
nb facteurs = 7  AUC test max = 0.9754137 coût = 1.1
nb facteurs = 8  AUC test max = 0.9751792 coût = 2
nb facteurs = 9  AUC test max = 0.9768875 coût = 1
nb facteurs = 10 AUC test max = 0.9757486 coût = 17.6
nb facteurs = 11 AUC test max = 0.9752462 coût = 13.6
nb facteurs = 12 AUC test max = 0.976921 coût = 2.4
nb facteurs = 13 AUC test max = 0.9827159 coût = 1
```

Avec les 13 premiers axes, nous obtenons une aire sous la courbe ROC plus élevée qu'avec n'importe quel modèle SVM jusqu'à présent : plus de 98%. De même que le coût est nettement très modéré, il est égal à 1. C'est l'effet de l'Analyse des Correspondances Multiples.

3.8 Expression explicite du modèle SVM sur composantes factorielles

Nous avons vu que le modèle SVM est plus discriminant quand il est calculé sur les douze premiers axes factoriels d'une analyse multiples des prédicteurs, plutôt que sur les prédicteurs eux-mêmes. Ces axes factoriels permettent de capter l'essentiel de l'information. Comme ils peuvent s'exprimer comme combinaison des prédicteurs initiaux, il est possible de passer des coefficients du modèle SVM par rapport aux axes factoriels aux coefficients par rapport aux prédicteurs initiaux. Et c'est ce qui nous intéresse le plus pour avoir les scores

Pour simplifier, nous calculons SVM sur les axes non centrés-réduits. Voici les coefficients du score SVM par rapport aux axes factoriels :

```
> w <- t(svmLin$coefs) %*% x[svmLin$index,]
> w
      Dim 1      Dim 2      Dim 3      Dim 4      Dim 5      Dim 6      Dim 7      Dim 8      Dim 9      Dim 10      Dim 11      Dim 12
[1,] -7.760168 -16.50829 -1.532291 -1.862581 0.1621552 6.124401 -4.470993 -0.351543 2.048459 2.580958 -0.4202769 -5.890929
```

Et ci-dessous un extrait des coordonnées des modalités sur les axes factoriels :

```
> ACM$var$coord
```

	Dim 1	Dim 2	Dim 3	Dim 4
Fcou	0.20418405	0.199194940	-0.03635056	-0.503875149
Fseu	-0.25372237	-0.247522827	0.04516978	0.626123339
Sfem	-0.35513098	-0.434821678	0.22168621	0.158050468
Shom	0.21597981	0.264445256	-0.13482278	-0.096121465
Pcad	0.30010375	0.726590291	-0.08283426	-0.191345076
Pemp	-0.05205145	-0.372085267	-0.20054650	0.152882931
Pint	0.51366005	0.094432990	0.13128849	-0.410903529
Pouv	0.01825220	-0.768581944	-0.57602418	-0.627154688
Psan	-0.78629176	-0.214947904	0.47234573	0.660428112
A0	-0.41741618	-0.163500174	-0.08216666	0.935607365
A1	0.20510571	0.237887040	-0.15915967	-0.300186676
A2	0.22834570	-0.075476600	0.25381771	-0.678826443
r0	-0.53054995	0.409320014	-0.22460706	0.194329649
r1	0.14245795	-0.300462695	0.07431004	0.474844390
R2	0.38768957	-0.108008554	0.15008710	-0.670515408
K0	-0.12904477	-0.095933084	0.03712652	-0.090830739
K1	1.04338759	0.775664082	-0.30018535	0.734409220

Multiplions la matrice ligne w des coefficients SVM par la matrice 12 * p suivante :

	Fcou	Fseu	Sfem	Shom
Dim 1	0.204184049	-0.253722374	-0.35513098	0.21597981
Dim 2	0.199194940	-0.247522827	-0.43482168	0.26444526
Dim 3	-0.036350555	0.045169783	0.22168621	-0.13482278
Dim 4	-0.503875149	0.626123339	0.15805047	-0.09612146
Dim 5	-0.007519214	0.009343496	-0.10793239	0.06564118
Dim 6	0.082896100	-0.103008023	0.19099662	-0.11615831
Dim 7	-0.184954932	0.229827964	-0.43626346	0.26532211
Dim 8	-0.024215073	0.030090038	0.14327928	-0.08713808
Dim 9	0.170926313	-0.212395777	-0.26730461	0.16256650
Dim 10	-0.117029091	0.145422225	0.32232860	-0.19603040
Dim 11	0.052050259	-0.064678486	-0.11117571	0.06761367
Dim 12	-0.081325006	0.101055757	0.07698796	-0.04682173

Nous obtenons un extrait des coefficients du score SVM par rapport aux prédicteurs initiaux suivants :

```
> coef <- w %*% t(ACM$var$coord[,1:naxes])
> coef
```

	Fcou	Fseu	Sfem	Shom	Pcad	
[1,]	-2.031455	2.524318	12.22992	-7.437861	-14.90576	
	Pemp	Pint	Pouv	Psan	A0	A1
[1,]	9.077799	-4.85087	12.546	6.430028	1.451453	-3.707004

Pour appliquer ces coefficients aux indicatrices des modalités prédicteurs dans l'échantillon test, nous allons créer ces indicatrices en leur donnant le nom des modalités non préfixé par le nom des variables. Nous procéderons ainsi puisque l'expression coef montre que seuls apparaissent le nom des modalités dans la matrice « ACM\$var\$coord ». Nous pouvons ensuite appliquer les coefficients précédents aux indicatrices des modalités, au lieu des coefficients des axes factoriels, et obtenir la prédiction du modèle SVM comparable à celle obtenue précédemment.

Chapitre 4 : Les Supports Vecteurs Machines

Ce chapitre, très court, est l'aboutissement de nos travaux. Pour ce faire, nous allons parler des règles de scoring (4.1), l'interprétation du scoring proprement dit (4.2) et un exemple de prédiction (4.3)

4.1 . Règles de scoring

Le meilleur modèle (SVM linéaire sur les axes factoriels) peut se transformer en une grille de score, attribuant à chaque modalité un nombre de points supérieur ou égal à 0, d'autant moins élevé que la modalité correspond à un profil ayant plus de chance à détenir la carte visa premier, et normalisé en sorte que chaque dossier ait un nombre total de points compris entre 0 (possesseur de la carte) et 100 (non-possesseur de la carte). Quand nous parlons de nombre de points, nous nous référons principalement au tableau suivant :

Tableau 5 : Grille de score

	VARIABLE	MODALITE	POIDS				
1	ageq	A0	1	30	moyrvq	M2	0
2	ageq	A1	0	31	nptag	N0	0
3	ageq	A2	3	32	nptag	N1	6
4	boppnq	B0	3	33	opgnbq	O0	2
5	boppnq	B1	2	34	opgnbq	O1	0
6	boppnq	B2	0	35	opgnbq	O2	3
7	dmvtpq	D0	0	36	pcspq	Pcad	0
8	dmvtpq	D1	12	37	pcspq	Pemp	9
9	dmvtpq	D2	3	38	pcspq	Pint	2
10	facan	f0	3	39	pcspq	Pouv	9
11	facan	f1	0	40	pcspq	Psan	8
12	famiq	Fcou	0	41	qsmoyq	Q0	6
13	famiq	Fseu	1	42	qsmoyq	Q1	10
14	gagetq	G0	3	43	qsmoyq	Q2	0
15	gagetq	G1	0	44	relatq	r0	0
16	havefq	H0	2	45	relatq	r1	5
17	havefq	H1	0	46	relatq	R2	3
18	itavcq	I0	5	47	rocnbq	R0	4
19	itavcq	I1	3	48	rocnbq	R1	0
20	itavcq	I2	0	49	Sexeq	Sfem	8
21	jnbjddq	J0	3	50	Sexeq	Shom	0
22	jnbjddq	J1	1	51	uenmnbq	U0	0
23	jnbjddq	J2	0	52	uenmnbq	U1	1
24	kvnunbq	K0	5	53	uenmnbq	U2	1
25	kvnunbq	K1	0	54	vienbq	V0	2
26	lgagtq	L0	2	55	vienbq	V1	0
27	lgagtq	L1	0	56	xlgnbq	X0	1
28	moyrvq	M0	9	57	xlgnbq	X1	0
29	moyrvq	M1	12	58	xlgnbq	X2	1
				59	ylvnbq	Y0	0
				60	ylvnbq	Y1	1
				61	ylvnbq	Y2	1

Ce tableau nous permette d'obtenir les principales informations d'un scoring. En réalité, d'une part le tableau nous donne une certaine idée des modalités qui contribuent le plus à la possession de la carte visa premier ou non. D'autre part, les associations effectuées dans le chapitre 1 nous a permis de faire sortir les différents segments des clients. Ce qui est vraiment complémentaire. Le tableau nous assure d'attribuer donc des scores aux top règles d'associations et de prédire un individu quelconque appartenant à une classe ou une autre qu'il soit potentiellement un client de la carte visa premier et permet d'assurer la publicité.

Avant d'entrer dans les interprétations, il faut d'abord déterminer normalement les seuils de score. Cela consiste à découper en tranches le nombre de points. Dans ce genre de problématique, on constitue généralement deux ou trois tranches de scores. Dans notre cas, nous établissons deux tranches uniquement correspondent à l'acceptation ou refus :

Une tranche la moins plausible dans laquelle l'offre est, sinon omis, du moins transmise à l'échelon hiérarchique supérieur (directeur d'agence par exemple) pour un examen approfondi du dossier qu'il fixer le seuil ultérieurement.

Une tranche la plus plausible où la réponse à la publicité sera assurée, il faut juste bien observer les différents profils et façonner la publicité afin de satisfaire au plus les besoins. C'est le cas d'un client avec un score à fixer ultérieurement.

4.2. Interprétations

Dans la tranche la plus plausible, nous avons le tableau des mieux classés dans les règles d'associations prédisant Coui:

	lhs <chr>	rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>	count <int>
[1]	{sexeq=Shom, moyrvq=M2, dmvtpq=D2}	=> {carvp=Coui}	0.1025400	0.8449612	0.1213547	2.515949	109
[2]	{nptagq=N0, moyrvq=M2, dmvtpq=D2}	=> {carvp=Coui}	0.1081844	0.8333333	0.1298213	2.481326	115
[3]	{moyrvq=M2, dmvtpq=D2}	=> {carvp=Coui}	0.1185325	0.8289474	0.1429915	2.468266	126
[4]	{sexeq=Shom, rocnbq=R1, qsmoyq=Q2, moyrvq=M2}	=> {carvp=Coui}	0.1072437	0.7916667	0.1354657	2.357260	114
[5]	{sexeq=Shom, nptagq=N0, gagetq=G1, moyrvq=M2}	=> {carvp=Coui}	0.1232361	0.7844311	0.1571025	2.335715	131
[6]	{sexeq=Shom, moyrvq=M2, itavcq=I2}	=> {carvp=Coui}	0.1091251	0.7837838	0.1392286	2.333788	116
[7]	{sexeq=Shom, nptagq=N0, moyrvq=M2, itavcq=I2}	=> {carvp=Coui}	0.1015992	0.7826087	0.1298213	2.330289	108
[8]	{sexeq=Shom, qsmoyq=Q2, moyrvq=M2}	=> {carvp=Coui}	0.1232361	0.7797619	0.1580433	2.321812	131
[9]	{sexeq=Shom, moyrvq=M2, jnbjldq=J2}	=> {carvp=Coui}	0.1025400	0.7785714	0.1317027	2.318267	109
[10]	{sexeq=Shom, nptagq=N0, qsmoyq=Q2, moyrvq=M2}	=> {carvp=Coui}	0.1138288	0.7756410	0.1467545	2.309542	121

Dans la tranche la moins plausible :

	lhs <chr>	rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>	count <int>
[1]	{sexeq=Sfem, facanq=f0, dmvtpq=D1}	=> {carvp=Cnon}	0.1138288	1.0000000	0.1138288	1.505666	121
[2]	{sexeq=Sfem, kvunbq=K0, facanq=f0, dmvtpq=D1}	=> {carvp=Cnon}	0.1025400	1.0000000	0.1025400	1.505666	109
[3]	{facanq=f0, dmvtpq=D1, boppnq=B1}	=> {carvp=Cnon}	0.1119473	0.9916667	0.1128881	1.493119	119
[4]	{sexeq=Sfem, havefq=H0, opgnbq=O0, moyrvq=M0}	=> {carvp=Cnon}	0.1081844	0.9913793	0.1091251	1.492686	115
[5]	{pcspq=Pemp, kvunbq=K0, lgagtq=L0, dmvtpq=D1}	=> {carvp=Cnon}	0.1063029	0.9912281	0.1072437	1.492458	113
[6]	{famiq=Fseu, facanq=f0, lgagtq=L0, dmvtpq=D1}	=> {carvp=Cnon}	0.1063029	0.9912281	0.1072437	1.492458	113
[7]	{kvunbq=K0, facanq=f0, dmvtpq=D1, boppnq=B1}	=> {carvp=Cnon}	0.1034807	0.9909910	0.1044214	1.492101	110
[8]	{sexeq=Sfem, endetq=E0, havefq=H0, opgnbq=O0}	=> {carvp=Cnon}	0.1373471	0.9864865	0.1392286	1.485319	146
[9]	{sexeq=Sfem, gagetq=G0, havefq=H0, opgnbq=O0}	=> {carvp=Cnon}	0.1317027	0.9859155	0.1335842	1.484459	140
[10]	{sexeq=Sfem, gagetq=G0, havefq=H0, moyrvq=M0}	=> {carvp=Cnon}	0.1279398	0.9855072	0.1298213	1.483844	136

Nous allons transformer respectivement ces tableaux en scoring suivant :

Tableau 6 : Score des top règles d'associations

	Antécédent 1	Antécédent 2	Antécédent 3	Antécédent 4
Classe 1	Shom	M2	D2	
Scores	0	0	3	
Classe 2	N0	M2	D2	
Scores	0	0	3	
Classe 3	M2	D2		
Scores	0	3		
Classe 4	Shom	R1	Q2	M2
Scores	0	0	0	
Classe 5	shom	N0	G1	M2
Scores	0	0	0	
Classe 6	Shom	M2	I2	
Scores	0	0	0	
Classe 7	Shom	N0	M2	I2
Scores	0	0	0	
Classe 8	Shom	Q2	M2	
Scores	0	0	0	
Classe 9	Shom	M2	J2	
Scores	0	0	0	
Classe 10	Shom	N0	Q2	M2
Scores	0	0	0	0

	Antécédent 1	Antécédent 2	Antécédent 3	Antécédent 4	Total
Classe 1	Sfem	fo	D1		
Scores	8	3	12		23
Classe 2	Sfem	K0	fo	D1	
Scores	8	5	3	12	28
Classe 3	F0	D1	B1		
Scores	3	12	2		17
Classe 4	Sfem	H0	O0	M0	
Scores	8	2	2	9	21
Classe 5	Pemp	K0	L0	D1	
Scores	9	5	2	12	28
Classe 6	Fseu	f0	L0	D1	
Scores	1	3	2	12	18
Classe 7	K0	f0	D1	B1	
Scores	5	3	12	2	22
Classe 8	Sfem	E0	H0	O0	
Scores	8	5	2	2	17
Classe 9	Sfem	G0	H0	O0	
Scores	8	3	2	2	15
Classe 10	SFem	GO	H0	M0	
Scores	8	3	2	9	22

Pour interpréter ces tableaux, il ne faut pas vraiment faire dans l'absolue comme dans la règle de scoring d'habitude. Il faut plutôt raisonner en des termes relatifs. En effet, plus les points obtenus par les clients sont élevés, plus l'est aussi leur note et vice-versa. Cependant, cette affirmation est limitée. En réalité, il peut y avoir des règles obtenant des notes très élevées mais ayant une confiance ou lift très faible. Ces règles sont presque inutile dans ce cadre de même pour les cas contraire. Le scoring dépend donc directement des performances des classes dans la règle d'associations.

En conséquence, lorsque le nombre de points obtenus est relativement élevé c'est-à-dire compris entre 15 à 28, le client a de très forte chance de ne pas intéresser la carte. Plus précisément, une note de 23 à 28 signifie que le client ne s'intéresse sûrement à la carte puisque la confiance est égale à 1. C'est l'exemple des clients féminins ayant un montant nul de montant facturé dans l'année en francs et dont l'âge du dernier mouvement est moins récent. Dans le cas contraire, le nombre de points obtenu doit être inférieur ou égal à 3 (les relations sont déjà détaillées dans le chapitre 1).

Qu'est-ce qu'il faut faire alors si les règles d'associations sont significatives mais n'appartenant pas dans ces intervalles ?

4.3. Prédiction

On essaie d'appliquer le modèle sur l'ensemble d'une population nouvelle:

```
> predict(svm1in)
 400 461 193 671  29 227 264 991  47  11 874 983 753 258 638 518 100 575 719 112 679
Cnon Cnon Coui Coui Coui Coui Coui Cnon Coui Coui Cnon Cnon Cnon Coui Cnon Cnon Coui Cnon Cnon Coui Cnon
 285 587 222 744 627 897 1023 845 828 232 131 893 648 107 633 950 311 659 938 834 836
```

Le modèle prédit par exemple que le 400^e nouveau client n'est pas un client prospect contrairement au client 193^e. Nous pouvons aussi faire une assimilation une par une. Par exemple, si un client a le profil suivant, le code de sortie sera :

```
new_data <- data.frame(famiq="Fcou",sexeq="Shom",pcspq="Pcad",
  ageq = "A1", relatq = "r1",kvunbq = "K1", vienbq = "V1", uemnbq = "U1",xlgnbq = "X2",
  ylvn bq = "Y1", rocn bq = "R0",
  nptagq = "N0",endetq = "E1",gagetq = "G0", facanq = "F1",
  lgagtq = "L1",havefq = "H1",
  qsmoyq = "Q1", opgn bq = "O1",
  moyrvq = "M2",dmvtpq = "D1",
  boppnq = "B2",itavcq = "I2",
  jnbj dq = "J1")
new_data
prediction <- predict(svm1in, new_data)
prediction
```

Nous aurions pu terminer cette illustration, malheureusement, nous n'avons plus du temps. Ce n'est pas grave puisque le modèle est déjà prêt à se déployer.

CONCLUSION

Notre étude avait pour objectif d'explorer les données de promotion de la carte Visa Premier dans une banque, d'identifier les comportements des clients qui sont susceptibles d'être intéressés par cette carte afin d'étudier la faisabilité d'un scoring pour mieux prédire. Pour cela, nous avons utilisé trois méthodes de modélisation prédictive : la recherche des règles d'association, le classifieur naïf de Bayes et le support vecteur machine.

Nous avons commencé notre analyse par la méthode mixte de recherche des règles d'association, qui nous a permis particulièrement de mettre en évidence des associations significatives entre les caractéristiques des clients et leur propension à souscrire à la carte Visa Premier. Nous avons observé que les clients non seulement les plus actifs en détention des comptes courants mais utilisent beaucoup les produits bancaires sont plus susceptibles de souscrire à la carte Visa Premier. Nous avons également identifié que les clients qui possèdent une très bonne qualité socio-professionnelle sont plus susceptibles de souscrire à la carte Visa Premier. Il s'agit surtout des personnes cadres et de sexe masculin.

Ensuite, nous avons utilisé le classifieur naïf de Bayes pour construire un modèle de classification pour prédire si un client souscrira ou non à la carte Visa Premier. Nous avons constaté que cette méthode offre une performance de classification élevée, avec une précision de 88% et un rappel acceptable. Cela signifie que notre modèle est capable de détecter avec une grande précision les clients qui sont susceptibles de souscrire à la carte Visa Premier, tout en minimisant les faux positifs.

Enfin, nous avons utilisé le support vecteur machine pour construire un modèle de classification binaire dans le même but. Nous constatons que sur les prédicteurs de base, le noyau linéaire assure déjà le meilleur pouvoir discriminant, devant le noyau radial, sigmoïde et polynomial. Mais il est extrêmement performant, quand on l'applique aux premiers facteurs obtenus après une analyse des correspondances multiples. Avec une précision nettement le plus élevée allant jusqu'à 98% voir plus, c'est la méthode la mieux classée dans notre étude D'où un scoring fondé sur cette méthode. Ces constats sur les différents noyaux sont plus proches de l'étude de Belloti et Crook (2009) où le noyau polynomial de degré 2 apparaît nettement en retrait par rapport au noyau linéaire et radial.

En conclusion, pour bien mener la GRC opérationnelle de la carte visa premier, nous recommandons l'utilisation de la première et la troisième méthode pour améliorer la stratégie de marketing de la banque. Leur combinaison se fait via l'analyse de scoring et la prévision du SVM. Grâce à ces outils modernes, notre département pourrait améliorer nettement le taux de retour des campagnes de marketing. Il suffit juste de continuer à retranscrire dans notre base de données les caractéristiques de nouveaux clients.

BIBLIOGRAPHIE

Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB* (Vol. 1215, pp. 487-499).

Bellotti, T., & Crook, J. (2009). Support vector machines for credit scoring and discovery of significant features. *Expert systems with applications*, 36(2), 3302-3308.

Husson, F., Cornillon, P. A., Guyader, A., Jégou, N., Josse, J., Klutchnikoff, N., ... & Thieurmél, B. (2018). *R pour la statistique et la science des données*. Presses universitaires de Rennes.

Stéphane, T. (2012). *Data mining et statistique décisionnelle : l'intelligence des données*. Editions Technip.

Stéphane, T. (2015). *Modélisation prédictive et apprentissage statistique avec R*. Éditions Technip.

Wu, X., & Kumar, V. (Eds.). (2009). *The top ten algorithms in data mining*. CRC press

ANNEXE

Annexe 01 : Liste des variables explicatives

Ident.	Libellé		Ident.	Libellé
ager	Age en années		nptag	Nombre de cartes point argent
boppn	Nombre d'opérations à M-1		opgnb	Nombre d'opérations par guichet dans le mois
dmvtp	Age du dernier mouvement (en jours		pcspq	Catégorie socio-professionnelle (code num)
endet	Taux d'endettement		pvs	Point de vente
facan	Montant facturé dans l'année en francs		qcred	Moyenne des mouvements créditeurs en kf
famiq	Situation familiale (Fmar : marié, Fcel : célibataire, Fdiv : divorcé, Fuli : union libre, Fsep : séparé de corps, Fveu : veuf)		qsmoy	Moyennes des soldes moyens sur 3 mois
gagetc	Total des engagements court terme en francs		relat	Ancienneté de relation en mois
gagem	Total des engagements moyen terme en francs		rocnb	Nombre de paiements par carte bancaire à M-1
gaget	Total des engagements en francs		segv2s	Segmentation version 2
havef	Total des avoirs épargne nanciè`re en francs		sexeq	Sexe (qualitatif)
impnbs	Nombres d'impayés en cours		tavep	Total des avoirs épargne monétaire en francs
itavc	Total des avoirs sur tous les comptes		uemnb	Nombre de produits épargne monétaire
jnbjd1s	Nombre de jours à débit à M		viemt	Montant des produits contrats vie en francs
kvunb	Nombre de compte à vue		vienb	Nombre de produits contrats vie
lgagt	Engagement long terme		xlgmt	Montant des produits d'épargne logement en francs
matric	Matricule (identifiant client)		xlgnb	Nombre de produits d'épargne logement
moyrv	Moyenne des mouvements nets créditeurs des 3 mois en kf		ylvmt	Montant des comptes sur livret en francs
			ylvnb	Nombre de comptes sur livret

Annexe 02 : Description des variables explicatives

Ident.	Libellé	Modalités	Signification(s)
ageq	Age en années	A0	[18 ; 36]
		A1	[37 ; 49]
		A2	[50 ; 65]
boppnq	Nombre d'opérations à M-1	B0	[0 ; 2,56]
		B1	[2,64 ; 3,61]
		B2	[3,64 ; 5,57]
dmvtpq	Age du dernier mouvement	D0	[0 ; 2,56]
		D1	2,64
		D2	[2,71 ; 6,85]
endetq	Taux d'endettement	E0	0
		E1	>0 et [0,69 ; 4,6]
facan	Montant facturé	F0	0
		F1	>0 et [6,084 ; 14,1]
famiq	Situation familiale	Fseu	seul
		Fcou	En couple
gagetq	Total des engagements	G0	0
		G1	>0 et [1,01 ; 15,06]
havefq	Total des avoirs	H0	0
		H1	>0 et [3,9 ; 15,77]
itavcq	Total des avoirs sur tous les	I0	[0 ; 8,96]
		I1	[8,97 ; 11,17]
		I2	[11,18 ; 16,05]
jnbjdaq	Nombre de jours à débit à M	J0	0
		J1	[0,69 ; 1,61]
		J2	[1,79 ; 4,91]
kvunbq	Nombre de compte à	K0	[0 ; 1]
		K1	[2 ; 4]
lgagtq	Engagement long	L0	0
		L1	[6,78 ; 15,06]

Ident.	Libellé	Modalités	Signification(s)
moyrvq	Moyenne des mouvements nets créditeurs des 3 mois	M0	[0 ; 1,95]
		M1	[2,08 ; 3,05]
		M2	[3,1 ; 7,45]
nptag	Nombre de cartes point argent	N0	0
		N1	[1 ; 4]
opgnbq	Nombre d'opérations par guichet dans le mois	O0	0
		O1	0,69
		O2	[1,1 ; 3,37]
pcspq	Catégorie socio-professionnelle	Pcad	Cadre
		Pemp	employé
		Pint	interim
		Pouv	ouvrier
		Psan	sans emploi
qsmoyq	Moyennes des soldes moyens sur 3 mois	Q0	[-70050 ; 1570]
		Q1	[1605 ; 7923]
		Q2	[7932 ; 241827]
relatq	Ancienneté de relation en mois	r0	[-2 ; 74]
		r1	[75 ; 175]
		R2	[176 ; 393]
rocnbq	Nombre de paiements par carte bancaire à M-	R0	0
		R1	[1 ; 69]
sexeq	Sexe (qualitatif)	Shom	Sexe homme
		Sfem	Sexe femme
uemnbq	Nombre de produits épargne monétaire	U0	0
		U1	1
		U2	[2 ; 9]
vienbq	Nombre de produits contrats vie	V0	0
		V1	[1 ; 13]
xlgnbq	Nombre de produits d'épargne logement	X0	0
		X1	1
		X2	[2 ; 4]
ylvnbq	Nombre de comptes sur livret	Y0	0
		Y1	1
		Y2	[2 ; 4]

Annexe 03 : Code R sur le chapitre 1

```
---
title: "Règle d'associations"
author: "RAKOTONIAINA Pety Ialimita"
date: "2023-02-05"
output: word_document
---

# 1. Téléchargement des packages

```{r}
library(arules)
library(arulesViz)
library(tidyverse)
```

# 2. Définition du répertoire de travail
```{r}
setwd("D:/Université de la Réunion/MQME Université de La Réunion/Data mining 2/Devoir à rendre/DM_2")
```

# 3. Importation et légère modification des bases des données
```{r}
visprem <- read.table(file.choose(),header=TRUE, sep = ",")
vispremt <- read.table(file.choose(),header=TRUE, sep = ",")
vispremv <- read.table(file.choose(),header=TRUE, sep = ",")
summarise(vispremv)

vispremh <-
 select(vispremv,
 famiq=FAMIQ,sexeq=SEXEQ,pcspq=PCSPQ,ageq,relatq,
 kvunbq,vienbq,uemnbq,xlgnbq,
 ylvnbq,rocnbq,nptagq,carvp=CARVP,endetq,
 gagetq,facandq,lgagtq,havefq,qsmoyq,
 opgnbq,moyrvq,dmvtppq,boppnq,itavcq,jnbjdg)

for (k in 1:ncol(vispremh)) vispremh[,k] <- as.factor(vispremh[,k])
```

# 4. Nombre des possesseurs et non possesseurs de la carte

```{r}
vispremt %>%
 group_by (CARVP) %>%
 summarise(effectif = n()) %>%
 mutate(freq = effectif /sum(effectif))
```

# 5. Identification des transactions et mitems

```{r}
head(vispremh, 8)
edit(vispremh)
```

# 6. Extraction des règles d'associations avec les paramètres par défaut

```{r}
rules.all <- apriori(vispremh)
summary(rules.all)
plot(rules.all)
```
```

```

# 7. Classement par indicateurs de performances
```{r}
rule.sorted <- sort(rules.all, by="support")

rules.sorted <- sort(rules.all)

inspect(rules.sorted [1:10])

Tableau de la règle d'association le mieux placé en support
tab1 <- table(vispremh$lgagtq, vispremh$kvunbq)
tab1

Règles les mieux classés en termes de lift
inspect(sort(rules.all, by="lift", decreasing = TRUE) [1:5])

Règles les mieux classés en termes d'indice de confiance
inspect(sort(rules.all, by="confidence", decreasing = TRUE) [1:10])
tab2 <- table(vispremh$lgagtq, vispremh$gagetq)
tab2

...

8. Prévission Coui
```{r}
rules_Coui <- apriori(vispremh, control = list (verbose = F),
  parameter = list(supp = 0.1, conf=0.6, maxlen = 5),
  appearance = list(rhs = "carvp=Coui", default="lhs"))
inspect(sort(rules_Coui, by = "confidence", decreasing = TRUE)[1:10])
plot(rules_Coui)
plot(rules_Coui, method="grouped")

...

# 9. Prévission Cnon
```{r}
rules_Cnon <- apriori(vispremh, control = list (verbose = F),
 parameter = list(supp = 0.1, conf=0.6, maxlen = 5),
 appearance = list(rhs = "carvp=Cnon", default="lhs"))
inspect(sort(rules, by = "confidence", decreasing = TRUE) [1:10])

rules_Cnon <- apriori(vispremh, control = list (verbose = F),
 parameter = list(supp = 0.1, conf=0.6, maxlen = 5),
 appearance = list(rhs = c("carvp=Coui", "carvp=Cnon"), default="lhs"))

inspect(sort(rules_Cnon, by = "confidence", decreasing = TRUE) [1:10])
```

```

Annexe 04 : Code R sur le chapitre 2

```
---
title: "Classifieur de Bayes"
author: "RAKOTONIAINA Pety Ialimita"
date: "2023-02-17"
output: html_document
---

##1. Création échantillon apprentissage et test
```{r}

échantillon/test
set.seed(2345)
id<-sort(sample(nrow(vispremh), nrow(vispremh)*2/3, replace=F))
apprent<-vispremh[id,]
test<-vispremh[-id,]
validation croisée
```

## 2. Classifieur naïves de Bayes
```{r}
Naive_Bayes_Model<-naiveBayes(carvp ~., data=apprent)

Naive_Bayes_Model
```

# Prédiction
```{r}
Pred1<-predict(Naive_Bayes_Model, apprent, type="raw")
head(Pred1)

Pred2<-predict(Naive_Bayes_Model, apprent)
head(Pred2)
```

##4. Deuxième analyse de la performance de la prédiction
```{r}
library(ROCR)

Pred3 <- prediction(Pred1[,2],apprent$carvp,label.ordering=c("Cnon","Coui"))
auc <- performance(Pred3,"auc")
performance(Pred3,"auc")@y.values[[1]]
```

##5. Troisième analyse : l'analyse de l'échantillon test
```{r}
Pred1<-predict(Naive_Bayes_Model, test, type="raw")
head(Pred1)

Pred2<-predict(Naive_Bayes_Model, test)
head(Pred2)

mc2<-table(test$carvp,Pred2)
mc2
```

```
tx_mauvais_classement2<-sum(Predi2!=test$carvp)/nrow(test)
tx_mauvais_classement2
```

```
Predi3 <- prediction(Predi1[,2],test$Cible,label.ordering=c(1,2))
auc <- performance(Predi3,"auc")
performance(Predi3,"auc")@y.values[[1]]
```



## Annexe 05 : Code R sur le chapitre 3

```

title: "SVM"
author: "RAKOTONIAINA Pety Ialimita"
date: "2023-02-13"
output: html_document

1. Téléchargement des packages
```{r setup, include=FALSE}
library(e1071)
library(ROCR)
library(Matrix)
library("FactoMineR")
library("kernlab")
```

2. Constitution d'un échantillon d'apprentissage et de test
```{r setup, include=FALSE}
index <- 1:nrow(vispremh)
index

id <- sample(index, trunc(length(index)*65.19285/100))

apprent <- vispremh[id,]

summary(apprent)

test <- vispremh[-id,]
summary(test)
```

3. SVM à noyau linéaire
```{r setup, include=FALSE}
vars <- names(vispremh)

svmlin = svm(carvp ~ ., data=vispremh[id,vars], kernel="linear", probability=TRUE,
cost=1)

summary(svmlin)
```

4. Analyse des résultats
```{r setup, include=FALSE}
# Indices des vecteurs supports
svmlin$index

# Coefficients des vecteurs supports
# Il s'agit des quantités  $\alpha_i y_i$ 
head(svmlin$coefs)

# Matrice de confusion
table(svmlin$fitted, vispremh[id,"carvp"])
mean(svmlin$fitted != vispremh [id,"carvp"])

# Taux d'erreur par validation croisée
set.seed(235)
svmlin = svm(carvp ~ ., data=vispremh[id,vars], kernel="linear", probability=TRUE,
cost=1, cross=10)
summary(svmlin)
```

5. Comparaison des SVM sur l'échantillon test et apprentissage
```{r setup, include=FALSE}
#Extrait des probabilités d'appartenance à chaque classe et ROC : Test
test$svmlin = predict(svmlin, test, probability = T)
test$svmlin
table(test$svmlin,test$carvp)
mean(test$svmlin != test$carvp)
```

```

taux_erreur_test_1 <- mean(test$svmlin != test$carvp)
pred <- prediction(attr(test$svmlin,"probabilities")[,1],
test$carvp,label.ordering=c("Cnon", "Coui"))
performance(pred,"auc")@y.values
AUC_test_1 <- performance(pred,"auc")@y.values

#Extrait des probabilités d'appartenance à chaque classe et ROC : Apprentissage
apprent$svmlin <- predict(svmlin, apprent, probability = T)
apprent$svmlin
table(apprent$svmlin,apprent$carvp)
mean(apprent$svmlin != apprent$carvp)
taux_erreur_apprent_1 <- mean(test$svmlin != test$carvp)
pred <- prediction(attr(apprent$svmlin,"probabilities")[,1],
apprent$carvp,label.ordering=c("Cnon", "Coui"))
performance(pred,"auc")@y.values[[1]]
AUC_apprent_1 <- performance(pred,"auc")@y.values[[1]]
```

6. Optimisation des SVM linéaires
```{r}
#Validation croisée
tuned <- tune.svm(carvp ~ ., data = vispremh[id,vars], cost = 10^(-3:1))
summary(tuned)
plot(tuned)

#Optimisation par la base d'un test
svmlin = tune.svm(carvp ~ ., data=vispremh[id,vars], kernel="linear", cost=10^(-3:1),
validation.x = test[,13],
validation.y = test[, "carvp"],
tunecontrol = tune.control(sampling = "fix", fix=1))
summary(svmlin)

# Prédiction
svmlin <- svm(carvp ~ .,
data=vispremh[id,vars],kernel="linear",probability=TRUE,cost=10)

predsvmlin <- predict(svmlin,type="prob",vispremh[id,vars],probability=TRUE)
pred <- prediction(attr(predsvmlin,"probabilities")
[,2],vispremh[id,"carvp"],label.ordering=c("Cnon", "Coui"))
performance(pred,"auc")@y.values[[1]]

# Sur l'échantillon test
test$svmlin = predict(svmlin,type="prob",test,probability=TRUE)
table(test$svmlin,test$carvp)
mean(test$svmlin != test$carvp)
taux_erreur_test_2 <- mean(test$svmlin != test$carvp)
pred=prediction(attr(test$svmlin,"probabilities")
[,2],test$carvp,label.ordering=c("Cnon", "Coui"))
performance(pred,"auc")@y.values[[1]]
AUC_test_2 <- performance(pred,"auc")@y.values[[1]]

# Sur l'échantillon apprentissage
apprent$svmlin <- predict(svmlin, apprent, probability = T)
apprent$svmlin
table(apprent$svmlin,apprent$carvp)
mean(apprent$svmlin != apprent$carvp)
taux_erreur_apprent_2 <- mean(test$svmlin != test$carvp)
pred <- prediction(attr(apprent$svmlin,"probabilities")[,1],
apprent$carvp,label.ordering=c("Cnon", "Coui"))
performance(pred,"auc")@y.values[[1]]
AUC_apprent_2 <- performance(pred,"auc")@y.values[[1]]

## Grille de performance
Cout <- c(1,10)
AUC_apprentissage <- c(AUC_apprent_1, AUC_apprent_2)

```

```

AUC_test <- c(AUC_test_1, AUC_test_2)
taux_erreur_apprentissage <- c(taux_erreur_test_1, taux_erreur_test_2)
taux_erreur_test <- c(taux_erreur_apprent_1, taux_erreur_apprent_2)

Grille_SVM_lineaire <- data.frame(Cout, AUC_apprentissage, taux_erreur_apprentissage,
AUC_test, taux_erreur_test)
...

# 7. SVM à noyau radial gaussien
```{r}
Extractions des probabilités d'appartenance à chaque classe
svmrاد = tune.svm(carvp ~ . , data=vispremh[id,vars], kernel="radial",
cost=seq(0.1,10,by=0.1),
 gamma = c(0.001,0.01,0.1,1),
 validation.x = test [,-13],
 validation.y = test [,"carvp"],
 tunecontrol = tune.control(sampling = "fix", fix=1))
summary(svmrad)

svmrاد = tune.svm(carvp ~ . , data=vispremh[id,vars], kernel="radial",
cost=seq(0.01,5,by=0.01),
 gamma = seq(0.1,0.5,by = 0.01),
 validation.x = test [,-13],
 validation.y = test [,"carvp"],
 tunecontrol = tune.control(sampling = "fix", fix=1))
summary(svmrad)
plot(svmrad)

#Pouvoir discriminant
svmrاد <- svm(carvp ~ . , data=vispremh[id,vars], kernel="radial",
probability=TRUE,cost=5.5, gamma=0.01)
test$svmrاد <- predict(svmrad, test, probability = TRUE)
pred <- prediction(attr(test$svmrاد, "probabilities")[,1],
 test$carvp, label.ordering = c("Coui", "Cnon"))
performance(pred,"auc")@y.values[[1]]
summary(svmrad)
...

8.SVM à noyau sigmoïde
```{r}
## Extraction des probabilités d'appartenance à chaque classe
svmsig = tune.svm(carvp ~ . , data=vispremh[id,vars], kernel="sigmoid",
cost=seq(1,100,by=0.5),
      gamma = c(0.0001,0.0005,0.001,0.01,0.1,1),
      validation.x = test [,-13],
      validation.y = test [,"carvp"],
      tunecontrol = tune.control(sampling = "fix", fix=1))
summary(svmsig)

#Pouvoir discriminant
svmsig <- svm(carvp ~ . , data=vispremh[id,vars], kernel="sigmoid",
probability=TRUE,cost=69.5, gamma=0.001)
test$svmrاد <- predict(svmrad, test, probability = TRUE)
pred <- prediction(attr(test$svmrاد, "probabilities")[,1],
      test$carvp, label.ordering = c("Coui", "Cnon"))
performance(pred,"auc")@y.values[[1]]
summary(svmsig)

##
...

# 9. SVM à noyau polynomial
```{r}
Extraction des probabilités d'appartenance à chaque classe

```

```

svmpol = tune.svm(carvp ~ . , data=vispremh[id,vars], kernel="polynomial",
 degree=2,
 gamma = seq(0.01,1,by = 0.01),
 cost=seq(0.1,10,by=0.1),
 validation.x = test[, -13],
 validation.y = test[, "carvp"],
 tunecontrol = tune.control(sampling = "fix", fix=1))

meilleur modèle
svmpol$best.model

Extraction de gamma
svmpol$best.model$gamma

#Scénario 1 et ROC
svmpol <- svm(carvp ~ . , data=vispremh[id,vars],
 kernel="polynomial", probability=TRUE,
 degree =2,
 cost=0.3,
 gamma=0.29)

test$svmpol <- predict(svmpol, test, probability = TRUE)
pred <- prediction(attr(test$svmpol, "probabilities")[,1],
 test$carvp, label.ordering = c("Coui", "Cnon"))
performance(pred,"auc")@y.values[[1]]
summary(svmpol)

test$svmpol <- predict(svmpol, test, probability = TRUE)
pred <- prediction(attr(test$svmpol, "probabilities")[,1],
 test$carvp, label.ordering = c("Coui", "Cnon"))
performance(pred,"auc")@y.values[[1]]
summary(svmpol)

#Scénario 2 et ROC
svmpol <- svm(carvp ~ . , data=vispremh[id,vars],
 kernel="polynomial", probability=TRUE,
 degree =2,
 cost=seq(0.1,10,by=0.1),
 coeff0 = seq(0,1,by=0.01),
 gamma=seq(0.01,0.1,by=0.01),
 validation.x = test[, -20],
 validation.y = test[, "carvp"],
 tunecontrol = tune.control(sampling = "fix", fix = 1))

test$svmpol <- predict(svmpol, test, probability = TRUE)
pred <- prediction(attr(test$svmpol, "probabilities")[,1],
 test$carvp, label.ordering = c("Coui", "Cnon"))
performance(pred,"auc")@y.values[[1]]
summary(svmpol)
...

10. SVM après réduction de dimension

```{r}
## SVM après réduction de dimension

# Etape 1 : ACM
ACM <- MCA(vispremh, ncp = 39, axes = c(1,2),
  graph = TRUE, quali.sup = 14)

# Etape 2 : nombre modalité des variables
modal = apply(vispremh[, -13], 2, function(x)
  nlevels(as.factor(x)))
modal
sum(modal) - ncol(vispremh[, -13])

## Etape 3 : SVM sur les 02 axes factoriels
naxes <- 2
x <- ACM$ind$coord[id, 1:naxes]
y <- vispremh[id, "carvp"]
ksmvlin <- ksvm(x, y, type="C-svc",
  kernel = "vanilladot",
  C=10,

```

```

        prob.model=T)

## Etape 4 : Représentation graphique dy SVM sur les 02 axes
plot(ksmvlin, data = cbind(x,y))

## Etape 5 : Application échantillon test
xt <- ACM$ind$coord[-id,1:naxes]
yt <- vispremh[-id,"carvp"]
predksvm <- predict(ksmvlin, type = "prob", xt)
pred <- prediction(predksvm[,2], yt, label.ordering = c("Cnon", "Coui"))
performance(pred,"auc")@y.values[[1]]

## Etape 6 : Etablissement de la Fonction de la courbe de ROC
f <- function(i)
{
  ksvm <- svm(x,y, kernel = "linear", cost = i, probability = TRUE)
  predksvm <- predict(ksvm, xt, probability = TRUE)
  pred <- prediction(attr(predksvm, "probabilities")[,1], yt,
                    label.ordering = c("Coui", "Cnon"))
  performance(pred,"auc")@y.values[[1]]
}
f(1)

## Etape 7 : Recherche axe optimale
for (n in seq(2,13,by=1))
{
  i <- seq(1, 20, by = 0.1)
  x <- ACM$ind$coord[id,1:n]
  xt <- ACM$ind$coord[-id,1:n]
  k <- Vectorize(f)(i)
  cout <- i[which(k==max(k), arr.ind = TRUE)[1]]
  cat("\n", "nb facteurs = ", n, " AUC test max = ", max(k),
      "coût = ", cout)
}

# Etape 8 : SVM non centrée reduite
naxes <- 13
x <- ACM$ind$coord[id,1:naxes]
y <- vispremh[id,"carvp"]
xt <- ACM$ind$coord[-id,1:naxes]
yt <- vispremh[-id,"carvp"]
smvlin <- svm(x,y, kernel = "linear",
             probability = TRUE,
             cost = 1,
             scale = F)

# Etape 9 : Les coefficients du score SVM par rapport aux axes factoriels
w <- t(svm$coef) %*% x[svm$index,]
w

# Etape 10 : Coordonnées des modalités à chaque axe factoriel :
a <- ACM$var$coord
a

# Etape 11 : Multiplication de la matrice w des coefficients SVM par la matrice 13 * p
b <- t(a[,1:naxes])
b

## Etape 10 : Coeff prédicteurs initiaux
coef <- w %*% b
coef
...

```

Annexe 06 : Extrait Code R sur le chapitre 4

```
## 1. Choix du modèle et Scoring
```{r}
write.table(coef, "coef.csv", sep = ";", row.names = FALSE)

setwd("D:/Université de la Réunion/MQME Université de La Réunion/Data mining 2/Devoir
rendre/DM_2")

var_mod_coef <- read.table(file.choose(),header=TRUE)

mini = aggregate(data.frame(min = var_mod_coef$COEF),
 by = list(VARIABLE = var_mod_coef$VARIABLE), min)
mini

maxi = aggregate(data.frame(max = var_mod_coef$COEF),
 by = list(VARIABLE = var_mod_coef$VARIABLE), max)
maxi

total = merge(mini, maxi)

total$diff = as.numeric(total$max) - as.numeric(total$min)

total$diff

poids_total <- sum(total$diff)

grille <- merge(var_mod_coef, mini, all.x = TRUE)

grille$delta <- as.numeric(grille$COEF) - as.numeric(grille$min)

grille$POIDS <- abs(round((100*grille$delta) / poids_total))

grille$POIDS

grille[order(grille$VARIABLE, grille$MODALITE), c("VARIABLE","MODALITE", "POIDS")]
```

<b>SOMMAIRE</b>	<b>2</b>
<b>RESUME</b>	<b>3</b>
<b>Liste des figures</b>	<b>4</b>
<b>Liste des tableaux</b>	<b>4</b>
<b>INTRODUCTION</b>	<b>5</b>
<b>Chapitre 1 : Identification des règles d'associations</b>	<b>7</b>
1.1 La recherche des règles d'associations : une méthode supervisée exceptionnelle	7
1.2 Présentation de la base des données appropriée à la recherche des règles d'association	9
1.3 Les caractéristiques globales des clients de la banque à forte potentielle de règles d'associations	10
1.4 Top des règles d'associations performants : au carrefour du fonctionnement des comptes courants et des produits bancaires	13
1.5 La possession de la carte visa premier : la situation socio-professionnelle des clients s'ajoutent aux facteurs généraux	15
<b>Chapitre 2 : Classifieur bayésien naïf</b>	<b>18</b>
2.1 Le théorème de Bayes et la classification probabiliste comme points théoriques centraux de la méthode	18
2.2 Lancement et analyse intuitive du modèle Classifieur naïves de bayes	19
2.3 Très bonne performance de l'apprentissage prédictive du modèle Classifieur naïves de bayes	21
2.4 Une généralisation du modèle d'apprentissage acceptable	22
<b>Chapitre 3 : Les Supports Vecteurs Machines</b>	<b>24</b>
3.1 SVM : une méthode fondée sur l'hyperplan et par nature souple	24
3.2 Directive d'implémentation de l'algorithme SVM efficacement	26
3.3 SVM à noyau linéaire : un modèle suffisamment performant	26
3.4 Application du SVM à noyau radiale gaussien : nettement inférieur à celle du SVM linéaire	30
3.5 SVM sigmoïde pratiquement à égalité de performance que SVM radial gaussien	32
3.6 SVM à noyau polynomial moins performant que le SVM à noyau linéaire	32
3.7 SVM linéaire après réduction de dimension	33
3.8 Expression explicite du modèle SVM sur composantes factorielles	35
<b>Chapitre 4 : Les Supports Vecteurs Machines</b>	<b>37</b>
<b>CONCLUSION</b>	<b>41</b>
<b>BIBLIOGRAPHIE</b>	<b>42</b>
<b>ANNEXE</b>	<b>43</b>
Annexe 01 : Liste des variables explicatives	43

<b><i>Annexe 02 : Description des variables explicatives</i></b>	<b>44</b>
<b><i>Annexe 03 : Code R sur le chapitre 1</i></b>	<b>45</b>
<b><i>Annexe 04 : Code R sur le chapitre 2</i></b>	<b>47</b>
<b><i>Annexe 05 : Code R sur le chapitre 3</i></b>	<b>49</b>
<b><i>Annexe 06 : Extrait Code R sur le chapitre 4</i></b>	<b>54</b>