



ENSAE 3ème année  
Stage de fin d'études  
Année scolaire 2016-2017

---

## Prévision du défaut sur le périmètre de TPE

---

XU KAI

Maître de stage : Smaïl IBBOU  
du 15 mai au 14 novembre

# Prévision du défaut sur le périmètre de TPE

Stage de fin d'études Ensaïe

Note de synthèse

Maître de stage : Smaïl IBBOU

Septembre 2017

Ce rapport se concentre sur une série des méthodes d'apprentissage statistiques qui est adaptée pour la prévision du défaut des TPEs. Il se décompose en trois étapes progressives :

La première partie aborde un sujet de classification sous l'angle d'apprentissage semi-supervisé. L'enjeu est d'appliquer un réseau de neurones-carte Kohonen à la classification des clients défaillants en profitant de la structure topologique des entrées. Face à la faiblesse découverte, deux ajustements sont prises en compte : soit utiliser la variable de classe comme une nouvelle entrée pour entraîner une carte Kohonen supervisée dans le cadre d'apprentissage semi-supervisé, soit employer des algorithmes supervisés en traitant la variable de classe comme la sortie dans le cadre d'apprentissage supervisé. Le premier ajustement est présenté dans la première partie. En pratique, le sous-échantillonnage et l'information supplémentaire de variable de classe sont trouvés utiles pour la discrimination.

La deuxième partie est consacrée au deuxième ajustement-l'apprentissage supervisé. Quatre classifieurs non-linéaires sont proposés : arbre de décision, régression logistique pénalisée, forêt aléatoire et gradient boosting. Une sélection de variables est effectuée pour trouver les variables les plus pertinentes. Les modèles sont puis appliqués aux variables sélectionnées par les critères différents. Deux modèles basés sur la régression logistique pénalisée sont jugés à la fois performants et robustes. Ils sont construits sur deux ensembles des variables choisis par deux critères différents. Nous les comparons avec la nouvelle méthode que nous proposons dans la validation croisée.

Dans la troisième partie, nous présentons une méthode d'apprentissage hybride qui profite de la combinaison de l'apprentissage non supervisé et l'apprentissage supervisé. Cette méthode est construite sur les progrès obtenus dans les parties précédentes. En utilisant la carte Kohonen pour le regroupement, un classifieur local est effectué sur chaque groupe dans le but de construire des variables synthétiques. La construction des variables se fait dans l'espace des variables les plus discriminantes choisis par des critère différents. Une régression logistique pénalisée sera effectuée sur des variables synthétiques pour la classification finale.

Hors des trois parties, une validation croisée est effectuée. Les meilleurs modèles de l'apprentissage supervisé et ceux de l'apprentissage hybride. Leur performance et robustesse sont tous satisfaisant, ce qui montre que les variables propres au modèle sont assez informative pour la prévision.

En bref, malgré des problèmes manifestés dans notre base de données de TPE, les modèles d'apprentissage statistique ont réussi à obtenir des prévisions raisonnables sous les traitement appropriés.

De plus, une méthode innovante est construit dans un cadre d'apprentissage hybride.

**Mots-clés** : apprentissage semi-supervisé, sous-échantillonnage, apprentissage supervisé, sélection des variables, apprentissage hybride, variable synthétique, validation croisée

## Référence

- [1] IBBOU Smail, *Classification, analyse des correspondances et methodes neuronales*. Thèse, Université Paris 1, 1992.
- [2] Melssen, W., Wehrens, R., Buydens, L. *Supervised Kohonen networks for classification problems*. Chemometrics and Intelligent Laboratory Systems, 83(2), 99-113 (2006).
- [3] Boczko, E. M., Xie, M., Wu, D., and Young, T. *Comparison of binary classification based on signed distance functions with support vector machines*. OCCBIO'09. Ohio Collaborative Conference on (pp. 139-143). IEEE.2009

# Prediction of the default of TPE

Ensaïe internship

## Abstract

Supervisor : Smaïl IBBOU (Groupe BPCE)

September 2017

This article focuses on a series of statistical learning methods elaborated for the prediction of the default of VSB (very small business). It consists of three progressive stages :

The first part deals with the problem of classification from the view of semi-supervised learning. The challenge is to apply a Kohonen map to classify the clients in default by taking advantage of the topological structure of the input space. Facing the misclassification, two adjustments are taken into account : either using the class variable as a new input to train a supervised Kohonen map in the semi-supervised learning framework, or using supervised algorithms by processing the class variable such as output in the setting of supervised learning. The first adjustment is presented in the first part. In practice, under-sampling and additional class variable information are found useful for discrimination.

The second part is devoted to the second adjustment - supervised learning. Four nonlinear classifiers are proposed : decision tree, penalized logistic regression, random forest and gradient boosting. The feature selection is carried out to find the most relevant variables. The models are then applied on the variables selected by the different criteria. Two models based on penalized logistic regression are judged to be both efficient and robust. They are constructed on two sets of variables chosen by two different criteria. We compare them with the new method we propose in cross-validation.

In the third part, we present a hybrid learning method that takes advantage of the combination of unsupervised learning and supervised learning. This method is built on the progress obtained in the previous parts. Using the Kohonen map for clustering, a local classifier is performed on each group in order to construct synthetic variables. The variables are constructed in the space of the most discriminating variables chosen by different criteria. A penalized logistic regression will be performed on synthetic variables.

Besides the three parts, a cross-validation is carried out on the best models of supervised learning and those of hybrid learning. Their performance and robustness are all satisfactory, which shows that the model dependent variables are fairly informative for the forecasting.

In conclusion, despite problems in our VSB database, statistical learning models have been able to obtain reasonable predictions under appropriate treatment. In addition, an innovative method is built in a hybrid learning framework.

**Keywords :** semi-supervised learning, undersampling, supervised learning, feature selection, hybrid

learning, constructed feature, cross validation

## Reference

- [1] IBBOU Smail, *Classification, analyse des correspondances et methodes neuronales*. Thèse, Université Paris 1, 1992.
- [2] Melssen, W., Wehrens, R., Buydens, L. *Supervised Kohonen networks for classification problems*. Chemometrics and Intelligent Laboratory Systems, 83(2), 99-113 (2006).
- [3] Boczko, E. M., Xie, M., Wu, D., and Young, T. *Comparison of binary classification based on signed distance functions with support vector machines*. OCCBIO'09. Ohio Collaborative Conference on (pp. 139-143). IEEE.2009 Hors des trois parties, En bref, ce rapport présentera une technique réalisable pour l'optimisation de l'EQM et comparera sa performance sur les données fictives et réelles.

# Remerciements

En premier lieu, j'adresse particulièrement mon remerciement à M.Samil IBBOU, mon maître de stage au sein de BPCE, pour m'avoir permis d'effectuer mon stage de fin d'étude dans de bonnes conditions, et pour le sujet passionnant sur lequel j'ai pu travailler.

Je tiens également à remercier M.Quentin JAMMES pour le temps qu'il m'a consacré et son suivi tout au long de mon stage. Merci pour les connaissances qu'il m'a transmises ainsi que ses remarques précieuses qui m'ont permis d'approfondir ma compréhension du sujet.

Enfin je remercie tous les collègues de mon équipe pour leur patience et leur soutien tout au long de mon stage.

# Sommaire

Dans ce rapport, nous présentons une série des méthodes d'apprentissage statistiques destinée à faire la prévision du défaut des TPEs. Les modèles s'organisent dans le cadre d'une organisation évolutive.

D'abord, les cartes Kohonen différentes sont appliquées à notre base de données au fur et à mesure de notre exploration de la structure topologique des entrées. Ensuite, les modèles d'apprentissage supervisé sont proposés de faire la même tâche du point de vue différent. Enfin, une modèle d'apprentissage hybride est mis en œuvre afin de unifier l'apprentissage non-supervisé et l'apprentissage supervisé. Sa caractéristique est de faire la prévision basée sur les variables synthétiques propres au modèle.

La validation croisée montre que les variables propres aux entrées peut se remplacer par celles propres au modèle en ne pas dégradant la qualité de prévision. À travers l'apprentissage statistique, notre étude de la prévision du défaut des TPEs se déroule d'une manière progressive et finit par une méthode innovantes et performante.

# Summary

In this report, we present a series of statistical learning methods to predict the failure of VSBs. The models are organized in a progressive way.

First, the different Kohonen maps are applied to our database of VSBs so as to explore the topological structure of the input space. Then, the supervised learning models are proposed to do the same task from the different point of view. Finally, a hybrid learning model is implemented to unify unsupervised learning and supervised learning. Its characteristic is to make the prediction based on the synthetic variables which are model dependent.

The cross-validation shows that the models of input dependent variables can be replaced by those models of model dependent variables by not degrading the quality of prediction. Through statistical learning, our study of the prediction of the default of the VSBs takes place in a progressive way and ends up with an innovative and efficient method.

# Présentation du groupe BPCE

Le Groupe BPCE s'appuie sur ses deux principaux réseaux, Banque Populaire et Caisse d'épargne, ainsi que sur ses filiales pour mener à bien toutes les métiers de la banque et de l'assurance. Il est au service de 32 millions de clients avec ses 108 000 collaborateurs. Grâce à ses filiales, le groupe offre à ses clients un service complet : solutions d'épargne et d'investissement, service de placement, de trésorerie, de financement, d'assurance et gestion d'actifs.

Il est le deuxième groupe bancaire en France issu de la fusion en 2009 qui est mise en œuvre en réponse à la crise des subprimes. Les deux réseaux coopératifs gardent leur enseignes et leur indépendance mais coordonne et mettre en commun leur politique commune. En ce cas, des services de back-office s'intègrent tels que la direction des risques qui comprend l'unité Validation du pôle Analyses consolidées et Modèles dans laquelle j'ai effectué mon stage de fin d'études.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Contexte . . . . .	1
1.2	Présentation du sujet . . . . .	1
<b>2</b>	<b>Traitemen t des données</b>	<b>3</b>
2.1	Alimentation de la base des données . . . . .	3
2.2	Traitemen t des variables . . . . .	3
<b>3</b>	<b>Apprentissage semi-supervisé</b>	<b>5</b>
3.1	Carte Kohonen (SOM) . . . . .	5
3.2	Détection de structure topologique . . . . .	8
3.2.1	Enveloppe convexe . . . . .	8
3.2.2	Algorithm e t-SNE . . . . .	9
3.3	Carte Kohonen supervisée (SSOM) . . . . .	10
3.3.1	Sous-échantillonnage . . . . .	10
3.3.2	Modèle de carte Kohonen supervisée . . . . .	11
3.4	Évaluation des performances . . . . .	13
3.4.1	Indice de performance . . . . .	13
3.4.2	Comparaison des performances . . . . .	15
<b>4</b>	<b>Apprentissage supervisé</b>	<b>16</b>
4.1	Évaluation des performances . . . . .	16
4.2	Calibrage des paramètres . . . . .	17
4.3	Erreur de généralisation . . . . .	18
4.4	Modèle d'apprentissage . . . . .	18
4.4.1	Modèle de référence : Arbre de décision . . . . .	18
4.4.2	Régression logistique pénalisée . . . . .	19
4.4.3	Forêts aléatoires . . . . .	21
4.4.4	Gradient Boosting . . . . .	22
4.5	Sélection de variables . . . . .	23
4.5.1	Filter . . . . .	23
4.5.2	Embedded . . . . .	24
4.5.3	Variables plus importantes . . . . .	25
4.6	Résultat des modèles modifiés . . . . .	26

<b>5</b>	<b>Apprentissage hybride</b>	<b>29</b>
5.1	Construction des variables synthétiques . . . . .	29
5.1.1	Hétérogénéité . . . . .	29
5.1.2	Classifieur local et distance signée . . . . .	30
5.2	Modèle de prévision . . . . .	31
5.2.1	Procédure de modélisation . . . . .	31
5.2.2	Résultat empirique . . . . .	31
<b>6</b>	<b>Validation croisée</b>	<b>33</b>
6.1	Validation croisée Monte Carlo . . . . .	33
6.2	Résultat empirique . . . . .	33
<b>7</b>	<b>Conclusion et perspectives</b>	<b>36</b>

# Liste des tableaux

3.1	Définition : Matrice de confusion	13
3.2	Base d'apprentissage	15
3.3	Base de test	15
4.1	Sélection des variables	26

# Table des figures

3.1	Carte Kohonen . . . . .	5
3.2	Base d'apprentissage . . . . .	7
3.3	Base de test . . . . .	7
3.4	Test de séparabilité linéaire . . . . .	9
3.5	Base de Iris . . . . .	10
3.6	Base d'apprentissage TPE . . . . .	10
3.7	Carte Kohonen supervisé . . . . .	11
3.8	Base d'apprentissage . . . . .	12
3.9	Base de test . . . . .	12
3.10	Matrice de confusion : base de test . . . . .	13
4.1	Courbe ROC . . . . .	19
4.2	Courbe Gini . . . . .	19
4.3	Courbe ROC . . . . .	20
4.4	Courbe Gini . . . . .	20
4.5	Courbe ROC . . . . .	21
4.6	Courbe Gini . . . . .	21
4.7	Courbe ROC . . . . .	23
4.8	Courbe Gini . . . . .	23
4.9	Base d'apprentissage . . . . .	27
4.10	Base de test . . . . .	27
4.11	Base d'apprentissage . . . . .	28
4.12	Base de test . . . . .	28
5.1	Hétérogénéité géométrique . . . . .	30
5.2	Courbe ROC . . . . .	32
5.3	Courbe Gini . . . . .	32
5.4	Base d'apprentissage . . . . .	32
5.5	Base de test . . . . .	32
6.1	Base d'apprentissage . . . . .	34
6.2	Base de test . . . . .	34
6.3	Base d'apprentissage . . . . .	35
6.4	Base de test . . . . .	35

# Chapitre 1

## Introduction

### 1.1 Contexte

Les TPEs (très petites entreprises) constituent un pilier important de l'économie française en occupant environ 20% de l'emploi dans le secteur concurrentiel, et 20% de la valeur ajoutée créée par l'ensemble des entreprises. Actuellement, elles représentent un contingent important du Corporate. Cependant, les modèles actuels de prévision de défaut sur la classe d'actifs Corporate ne peuvent pas bien capter ses comportements spécifiques.

L'apprentissage statistique a été élaboré pour décrire au mieux un phénomène à partir des données volumineuses. Pourvu que la population modélisée soit suffisamment grande et représentative et stable, l'apprentissage statistique sera un moyen idéal à mettre en place afin de décrire au mieux les caractéristiques du défaut de TPE. La recherche des caractéristiques du défaut de TPE est une élément important pour le calcul de PD et LGD. Les deux dernières participent au calcul des RWA (Risk-Weighted Assets) et fonds propres ( $EL=EAD \times PD \times LGD$ )

En effectuant une fusion des données des réseaux Banques populaires, Caisse d'épargne et Natixis, le groupe BPCE a accumulé des données volumineuses de TPE à la fois comportementales et financières. Ainsi, les données de TPE du groupe possèdent les caractéristiques de quantité volumineuse et de grande dimension, ce qui permet d'en faire l'objet potentiel de l'apprentissage statistique.

### 1.2 Présentation du sujet

La prévision du défaut des clients est l'un des problèmes fondamentaux au cœur de l'attention des banques. L'approche traditionnel consiste à chercher les variables, principalement comptables, qui diffèrent au mieux les clients défaillants et les clients sains à partir de l'analyse financière. Cependant, un écart de performances est observé sur la population de TPE dans les modèles existants.

L'objet de notre travail est donc d'appliquer la technique d'apprentissage statistique sur les TPEs afin d'améliorer la prévision du défaut associée à leur comportement spécifique. En ce cas, une

meilleure allocation des crédits sera réalisée par une meilleure identification des risques.

Il est possible que les TPEs susceptibles de passer en défaut se distinguent de leurs homologues sains par la structure topologique de leurs caractéristiques. Nous souhaitons pouvoir représenter nos contrats en conservant la topologie d'entrée, et sans être handicapés par la présence de nombreuses données manquantes. c'est pourquoi des méthodes neuronales de carte Kohonen qui permet de faire la classification en préservant la topologie des individus sont implémentées. Basé sur les résultats obtenus de la carte Kohonen, les algorithmes d'apprentissages supervisés sont proposés dans le chapitre 4 pour ce type de tâches. Ensuite, une méthode hybride est mise en place dans le chapitre 5 en fusionnant nos cartes de Kohonen et les méthodes supervisés afin de construire un cadre assez performant et raffiné. Enfin, les validations croisées sont effectuées dans le chapitre 6 sur les modèles modifiés en vue de tester la robustesse de nos modèles. Il s'agira d'un côté d'explorer l'espace des entrées et de l'autre d'en développer un cadre prédictif sur les clients TPEs.

Pour la réalisation du stage, les éléments listés ci dessous sont mis à ma disposition :

- Un ordinateur opérant sous Windows 7 avec SAS et R, relié au réseau BPCE et à un serveur SAS
- L'entrepôt TPE de BPCE contenant les bases de données des TPEs, et les fichiers de SAS macros concernés
- Logiciels utilisés : SAS, R, WinSCP, Putty, pack Microsoft Oce

# Chapitre 2

## Traitement des données

### 2.1 Alimentation de la base des données

Les TPE correspondent à un sous-ensemble des portefeuilles actuels de la Banque de Détail (Retail) Professionnel et Entreprise (Corporate). La critère d'être TPE est que la chiffre d'affaires compris entre 1,5 et 5M€ ou montant total des engagements accordés plus que 1 M€. L'objet sur lequel nous travaillons est constitué des comptes à vue avec bilan dans la population TPE.

Dans un premier temps, il a fallu utiliser la base fournie par la pôle Modélisation. Il sera intéressant de réutiliser le modèle avec l'ensemble des données. La base totale est un échantillon occupant environ 1/6 des sociétés qualifiées dans 4 cohortes de Juin 2012 à Décembre 2013 à pas semestriel. La population atypique (associations, entreprises étrangères, SCI, ...) est supprimée de la base.

Cette base compte 18902 contrats dont environ 3% de clients sont en défaut, ce qui pose un problème de données déséquilibrées dans les classes différentes. Elle est divisée en 2 parties pour l'apprentissage : une base d'apprentissage comportant 70% des effectifs qui sert à construire et estimer les modèles différents ; une base de test comportant 30% des effectifs utilisée pour tester la stabilité des modèles.

### 2.2 Traitement des variables

Pour tenter de faire la prévision, nous disposons de variables de 4 catégories : compte, données externes, ratios financiers et signalétique. Au total, nous disposons de 202 variables explicatives parmi lesquelles 290 variables quantitatives et 12 variables qualitatives. Les variables comme Montant d'autorisation Escompte et Engagement court terme sont incluses dans la catégorie à cause de leur lien avec l'information de compte et de contrat. Les ratios calculés à partir du bilan et du état sont attribués à la catégorie ratios financiers. Les catégories de données externes et de signalétique sont plutôt les informations supplémentaires qui peuvent aider la prévision, par exemple, Effectif de l'entreprise, etc.

Les variables qualitatives sont mises sous forme de tableau disjonctif pour se transformer en nou-

velles variables binaires, ce qui produit environ 56 nouvelle variables. Et les variables quantitatives se normalisent pour être centrées réduites. Du coup, nous travaillons sur 246 variables explicatives générées. En raison de la roubustesse de la carte de Kohonen face aux données manquantes, nous n'avons pas besoin de remplacement de données manquantes. Pour les modèles de l'apprentissage statistique plus délicat, le remplacement peut se faire ensuite si besoin, grâce à la valeur moyenne/-médiane au sein de chaque classe.

# Chapitre 3

## Apprentissage semi-supervisé

### 3.1 Carte Kohonen (SOM)

La carte Kohonen est un modèle de réseau neurone artificiel qui nous permet d'apprendre la structure topologique des entrées de manière non supervisée. Elle est formée d'une couche de neurones disposé en grille ou ficelle pour regrouper les entrées. Chaque neurone est caractérisée par un vecteur poids de même dimension que les entrées qui détermine sa position dans l'espace d'entrée. L'apprentissage se déroule d'une manière de la compétition parmi les neurones pour la représentation des entrées qui lui sont présentées. Une partition de Voronoi sera implémenté en vue de découper l'espace d'entrée à partir des vecteurs poids appartenant aux neurones.

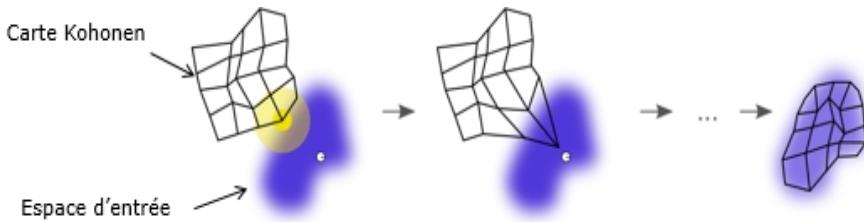


FIGURE 3.1 – Carte Kohonen

Contrairement aux méthodes factorielles telles que l'analyse en composantes principales ou l'analyse de correspondances multiples qui réalisent des projections sur des sous-espaces linéaires de l'espace des entrées, la carte Kohonen capture plutôt la structures topologique des sous-groupes au lieu de l'emplacement précis des individus. D'ailleurs, elle est plus robustes face aux données manquantes [1].

L'algorithme de carte Kohonen [1] :

- 1) Initialisation aléatoire des vecteurs poids  $\vec{w}$ ; l'initialisation de  $w$  se fait en utilisant la loi uniforme sur l'intervalle  $[0,1]$  de haute dimension
- 2) A instant  $t$ , une observation  $x(t+1) \subset R^d$  est choisie aléatoirement et présentée au réseau
- 3) Le neurone gagnant est définie par :

$$i_0 = \operatorname{arginf}_i \|x(t+1) - w_i(t)\|^2$$

pour la distance euclidienne

4) Les vecteurs poids de neurone gagnant  $w_{i_0}^t$  et ses voisins sont mis à jour par :

$$\begin{cases} w_i^{t+1} = w_i^t + \epsilon(t)(x(t+1) - w_i^t) & \forall i \in V_{r_t}(i_0) \\ w_i^{t+1} = w_i^t & \forall i \notin V_{r_t}(i_0) \end{cases}$$

où  $V_{r_t}(i_0)$  est le voisinage de rayon  $r(t)$  autour du neurone gagnant  $i_0$  mesuré par la distance de Manhattan  $d(\cdot, \cdot)$

$$1_{i \in V_{r_t}(i_0)} = \begin{cases} 1 & \text{if } d(i, i_0) \leq r_t \\ 0 & \text{if } d(i, i_0) > r_t \end{cases}$$

et où  $\epsilon(t)$  est le taux d'apprentissage qui satisfait les conditions de Robbins–Monro :

$$\sum_{t=0}^{\infty} \epsilon_t = \infty \quad \text{and} \quad \sum_{t=0}^{\infty} \epsilon_t^2 < \infty$$

Cependant, cet algorithme souffre de deux faiblesses [6] : 1) la convergence prématuée ; 2) la mauvaise initialisation. La convergence prématuée est un phénomène dans lequel le réseau de neurones converge si tôt que le résultat se trouve dans un optimum local. Et une mauvaise initialisation pourrait conduire à un réseau moins performant. Pour éviter les deux faiblesses, la quatrième étape du algorithme est modifiée :

$$w_i^{t+1} = w_i^t + \epsilon'(t) e^{\frac{-\|(w_{(i_0)} - w_i^t)\|^2}{r(t)^2}} (x(t+1) - w_i^t)$$

De plus, une décroissance exponentielle aux hyperparamètres est effectuée en fonction du temps  $t$  et du nombre d'itérations  $T$  borné :

$$\epsilon'(t) = \epsilon(0) \left( \frac{\epsilon(T)}{\epsilon(0)} \right)^{\frac{T}{t}} \quad r(t) = r(0) \left( \frac{r(T)}{r(0)} \right)^{\frac{T}{t}}$$

Il est évident que la nouvelle mise à jour satisfait aussi la condition de Robbins–Monro :

$$\begin{aligned} \sum_{t=0}^{\infty} \epsilon'(t) e^{\frac{-\|(w_{(i_0)} - w_i^t)\|^2}{r(t)^2}} &= \sum_{t=0}^T \epsilon'(t) e^{\frac{-\|(w_{(i_0)} - w_i^t)\|^2}{r(t)^2}} + \sum_{t=T+1}^{\infty} \epsilon(t) = \infty \\ \sum_{t=0}^{\infty} \epsilon'(t)^2 e^{\frac{-2\|(w_{(i_0)} - w_i^t)\|^2}{r(t)^2}} &< T \epsilon(0)^2 + \sum_{t=T+1}^{\infty} \epsilon(t)^2 < \infty \end{aligned}$$

Le quadrillage des paramètres se développe par deux étapes. Pour la première étape, un quadrillage plus gros est utilisé  $\epsilon(0) = \{0.1, \dots, 0.6\} \times \epsilon(T) = \{0.01, 0.02, \dots, 0.1\} \times r(0) = \{1, 1.5, \dots, 3\} \times r(T) = \{0.2, 0.4, \dots, 1\}$  sous les cartes tailles  $\{3, 4, \dots, 9\} \times \{3, 4, \dots, 9\}$ . Pour la deuxième étape, un quadrillage plus fin est utilisé  $\epsilon(0) = \{0.01, \dots, 0.1\} \times \epsilon(T) = \{0.01, 0.02, \dots, 0.05\} \times r(0) = \{2.9, 2.91, \dots, 3\} \times r(T) = \{0.6, 0.61, \dots, 0.7\}$  sous la carte de la taille  $5 \times 5$ .

Après le calibrage de paramètre, nous choisissons une  $5 \times 5$  carte munie des hyperparamètres :  $\epsilon(0) = 0.04, \epsilon(T) = 0.01, r(0) = 2.99, r(T) = 0.65$  pour un nombre d'itérations  $T=13231$ . Nous entendons faire une discrimination en utilisant le regroupement généré par notre carte de Kohonen entraînée. Chaque neurone représente d'une classe caractérisant la plupart des observations représentées par cet neurone.

Le déséquilibre de la répartition du nombre d'observations et de la répartition du défaut est observé dans les résultat de deux bases de données. Conformément à notre attente, le neurone avec des observations le plus nombreuses et ses voisins possèdent les taux de défaut plus bas que ceux des neurone avec des observations moins nombreuses. Cependant, la différence n'est pas évident, si bien que toutes les neurones sont marqués de la classe saine, c'est-à-dire qu'une prévision individuelle à chaque contrat reviendrait à tous les classer comme sain, ce qui est contre notre attente.

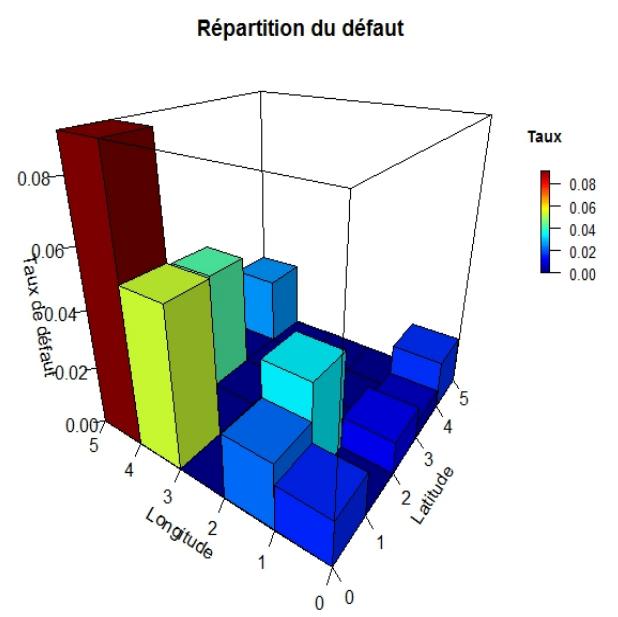
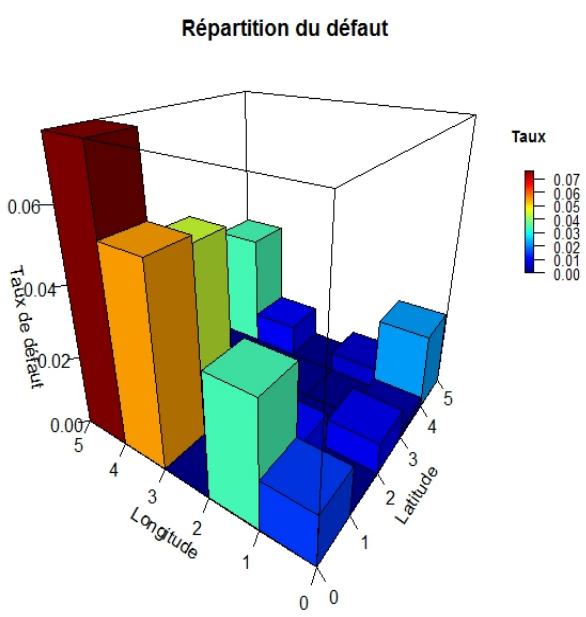
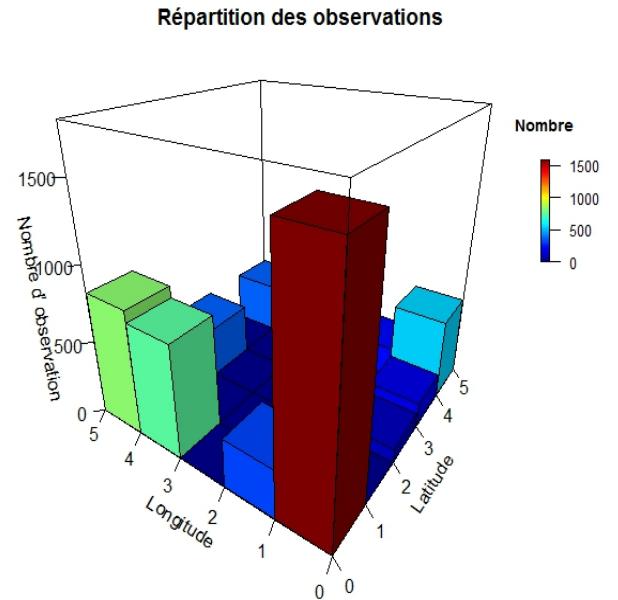
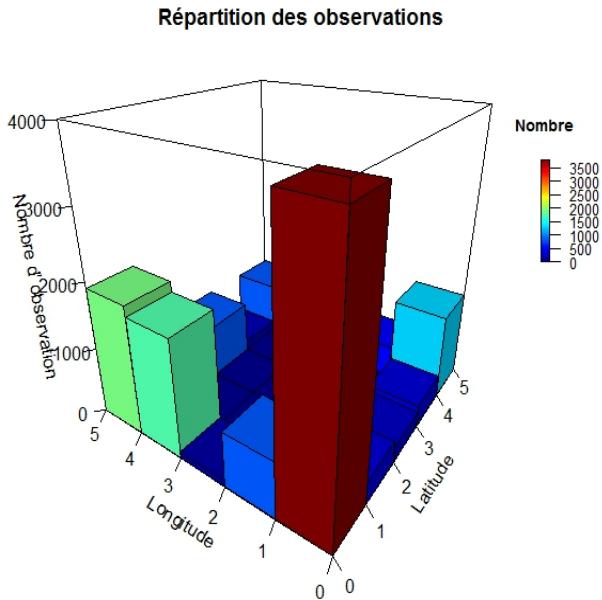


FIGURE 3.2 – Base d'apprentissage

FIGURE 3.3 – Base de test

La carte Kohonen est responsable de faire le regroupement. Basée sur son résultat, l'opération de donner les prévisions aux individus est en effet une technique de la propagation d'étiquette. Les

classes des nouvelles entrées sont déterminées par la classe attribué au neurone gagnant. Du coup, notre apprentissage est un variant de l'apprentissage semi-supervisé. Selon Chapelle et al [3], la réussite de l'apprentissage semi-supervisé réside dans trois hypothèse : 1) les entrées regroupées dans le même groupe tendent à être de même classe ; 2) l'hyperplan qui sépare les entrées de classes différentes devrait traverser une zone des entrées à faible densité ; 3) les entrées de haute dimension se tombent approximativement sur un objet géométrique de basse dimension.

En ce cas, notre découverte surprenante peut expliquée par les écarts entre les hypothèses et la structure topologique de nos entrées réelles. Nous explorons donc la structure topologique de nos entrées.

## 3.2 Détection de structure topologique

### 3.2.1 Enveloppe convexe

Pour un problème de classification à deux classes, il faut examiner tout d'abord la séparation linéaire pour vérifier la validité des discriminants linéaires. La séparation linéaire est une caractéristique selon laquelle les données de deux classes peuvent être classifiées par un hyperplan. C'est à dire, pour une population  $\vec{x}$  de deux classes, il existe un vecteur  $\vec{w}$  et un seuil  $c$  qui permettent que le hyperplan  $\vec{w} \cdot \vec{x} = c$  sépare les deux classes.

L'enveloppe convexe d'une classe est l'ensemble convexe le plus petit parmi ceux qui le contiennent. Si les enveloppes convexes se croisent, les deux classes se voient comme inséparables linéairement. Ici, l'analyse en composantes principales est effectuée pour visualiser les enveloppes convexes en 2 dimensions. L'opération de trouver l'enveloppe convexe se fait par le Rpackage grDevices.

La figure 3.1 révèle que les classes ne sont pas séparables linéairement parce que les enveloppes se chevauchent. Cette découverte nous montre que l'analyse discriminante linéaire ne peut pas être appliquée à nos données. De plus, les méthodes plus avancées permettant la séparation non-linéaire seront implémentées.

Classe 0 : Clients sains      Classe 1 : Clients défaillants

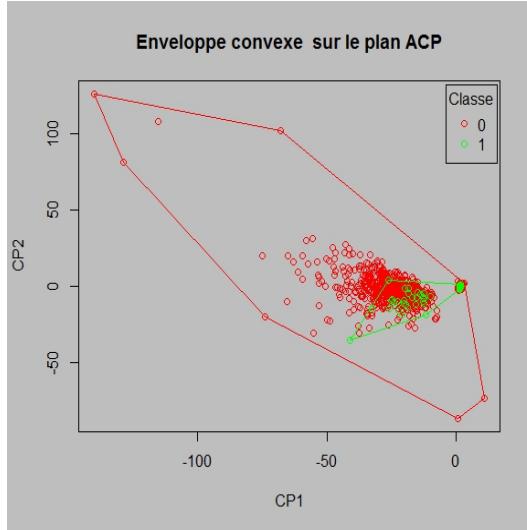


FIGURE 3.4 – Test de séparabilité linéaire

### 3.2.2 Algorithme t-SNE

Pour démontrer la faiblesse de la carte Kohonen non-supervisée qui ne profite que la structure topologique, une visualisation des données est effectuée par l'algorithme t-SNE (t-distributed stochastic neighbor embedding). Pour la carte Kohonen, la préservation de la structure topologique se concentre sur les sous groupes de la populations et ignore l'emplacement des individus. Par contre, le t-SNE fait une visualisation au niveau des individus en préservant la structure topologique, ce qui est mieux pour explorer l'espace des entrées.

Pour les points à grande dimension  $\mathbf{x}_1, \dots, \mathbf{x}_N$ , une distribution de probabilité  $p_{ij}$  est définie à partir de la probabilité conditionnelle  $p_{j|i}$  s'agissant de la similarité des paires des individus :

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

Une distribution de probabilité  $q_{ij}$  est également définie de la même manière pour les points projetés  $\mathbf{y}_1, \dots, \mathbf{y}_N$  :

$$q_{j|i} = \frac{\exp(-\|\mathbf{y}_i - \mathbf{y}_j\|^2)}{\sum_{k \neq i} \exp(-\|\mathbf{y}_i - \mathbf{y}_k\|^2)}$$

$$q_{ij} = \frac{q_{j|i} + q_{i|j}}{2N}$$

L'algorithme t-SNE [4] est destiné à trouver une projection optimale des points à grande dimension dans un espace de 2 dimension en minimisant la divergence de Kullback-Leibler entre les deux distributions. De cette manière, la proximité des individus est bien gardée. Les points projetés  $\mathbf{y}_1, \dots, \mathbf{y}_N$  sont déterminés de manière suivante :

$$\inf_{y_1, \dots, y_N} KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (3.1)$$

Le succès de la classification en utilisant la structure topologique réside dans la séparation topologique des individus de classes différentes. Beaucoup de bases de données de l'apprentissage machine profitent de cette propriété, par exemple, la base de Iris. Cependant la population de notre base ne manifeste la séparation provenant de la structure topologique, ce qui devient une faible de la carte Kohonen non-supervisée. Les figures suivantes illustre cette faiblesse.

Classe 0 : Clients sains      Classe 1 : Clients défaillants

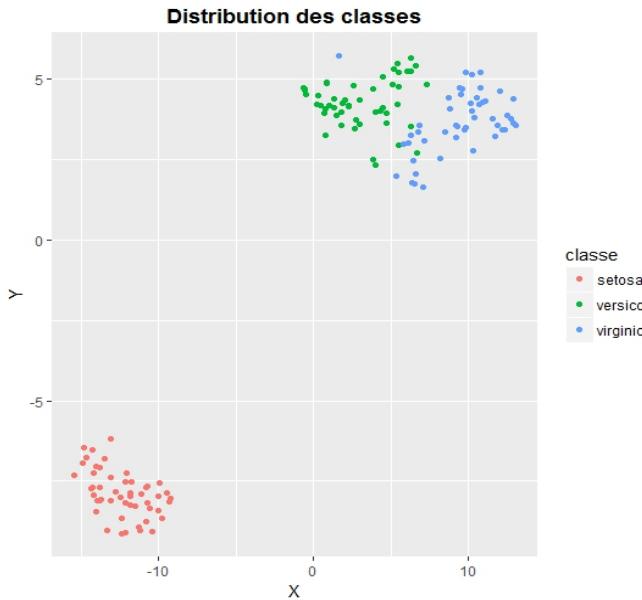


FIGURE 3.5 – Base de Iris

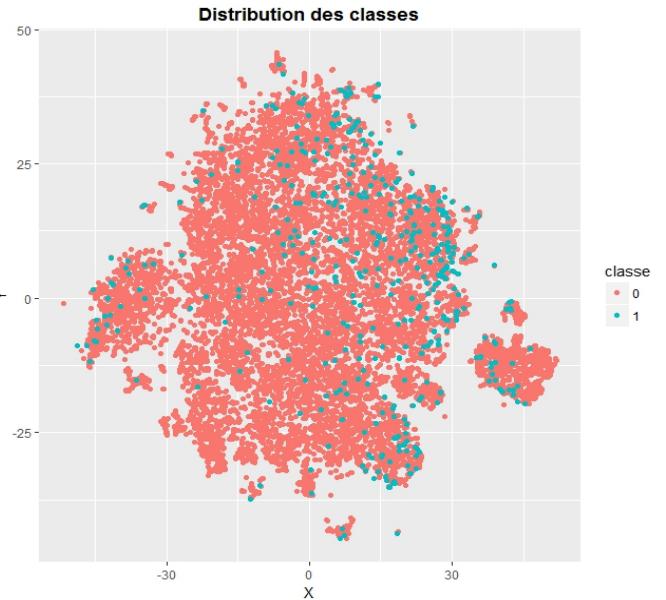


FIGURE 3.6 – Base d'apprentissage TPE

On voit sur la carte Kohonen que certaines zones où un taux de défaut plus fort, mais qu'une séparation claire n'est pas possible. La raison réside dans la dispersion des emplacements des clients défaillants et des clients sains de haute densité à proximité de ceux défaillants dans l'espace des entrées.

### 3.3 Carte Kohonen supervisée (SSOM)

La section précédente nous montre que seulement la structure topologique n'est pas suffisant pour faire la classification. Le déséquilibre des classes nous pose aussi un problème. Notre regroupement nécessite des informations supplémentaires et des traitements particuliers. C'est pourquoi le sous échantillonnage et la carte Kohonen supervisée sont effectués.

#### 3.3.1 Sous-échantillonnage

Le sous échantillonnage est une technique qui lutte contre le déséquilibre des classes en diminuant des observations de classe majoritaire dans la base d'apprentissage.[18][5] Il est préférable

d'utiliser cette méthode lorsque la base de données est énorme et la réduction du nombre des entrées aide à améliorer le temps d'exécution et les problèmes de stockage. Ici, un sous-échantillonnage des observations de classe majoritaire est réalisé par tirage aléatoire selon une loi uniforme.

### 3.3.2 Modèle de carte Kohonen supervisée

D'après Melssen et al [6], la carte KOonen supervisé est un réseau fusionné qui profite à la fois des entrées et des classes à l'étape d'apprentissage. Ce réseau se décompose en deux types de couches : couches d'entrée et couches de sortie. Ici, nous pouvons décomposer de plus les couches d'entrée en deux nouveaux types : couches d'entrée quantitative et couches d'entrée qualitative. De cette manière, on obtiendra trois types de couches au lieu de deux types de couches en modèle traditionnel. Le réseau s'entraîne de même manière que celui non supervisé sauf que une pondération des couches est possible.

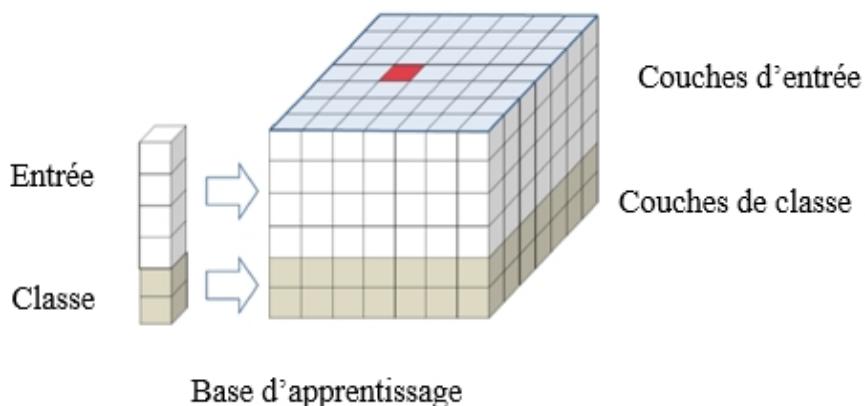
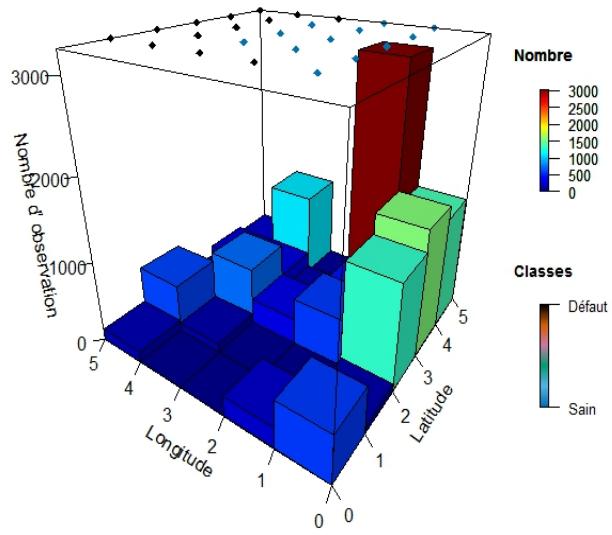


FIGURE 3.7 – Carte Kohonen supervisé

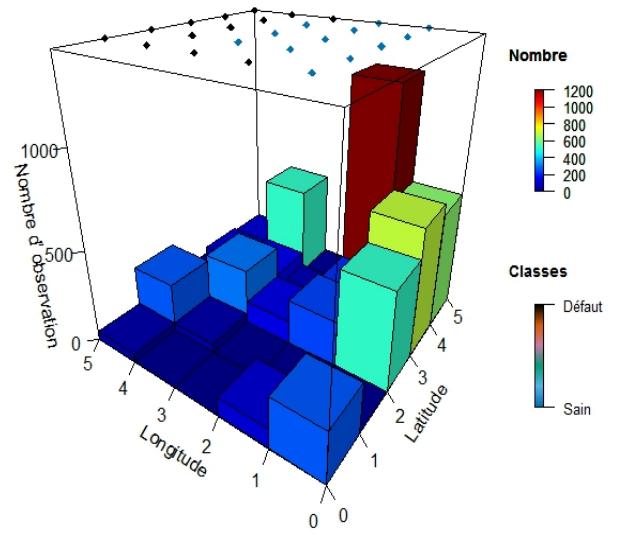
Nous effectuons tout d'abord une sous échantillonnage en diminuant le nombre des clients sains dans la base d' apprentissage à environ 1.1 fois celui de clients défaillants. Ensuite, un calibrage de paramètre est réalisé. Pour la simplicité, nous supposons les poids parmi les couches de même type sont pareils. De ce fait, on ne fait que le calibrage de paramètre sur les poids de différent type. Enfin, un réseau dont un poids de  $1/2$  est affecté aux couches d'entrée et un poids de  $1/2$  est affecté aux couches de sortie se trouve le plus raisonnable. En plus, le poids affecté aux couches d'entrée se décompose en deux parties :  $3/8$  pour les couches d'entrée quantitative et  $1/8$  pour celles d'entrée qualitative.

La carte Kohonen non supervisée ne peut pas produire les neurone marqué de classe défaillante. Par contre, la carte Kohonen supervisée combinée avec le sous échantillonnage est capable de produire des neurones défaillantes, ce qui est démontré dans les plafonds des figures de boîte suivantes. Le reste des observations sont classifiées selon la classe de neurone qui les représente. Le résultat de la classification sur des deux bases complètes se démontre dans les figures suivantes. Cette fois, la relation inverse entre le nombre des observations de neurone et le taux de défaut local est plus évident. Le taux de défaut augmente sur les neurones défaillantes malgré un manque de la dominance du côté de nombre des clients défaillants.

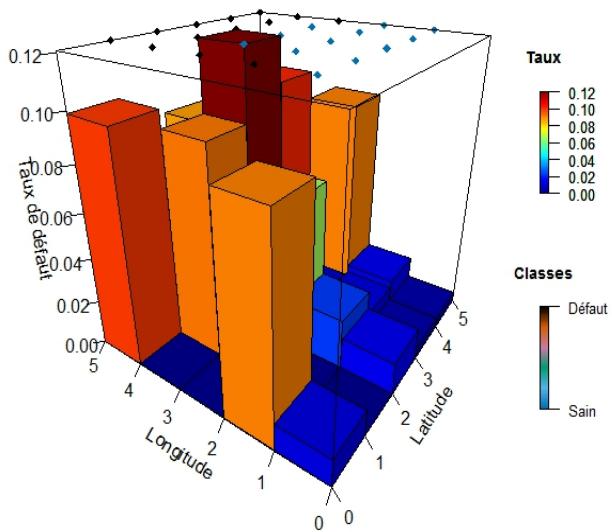
Répartition des observations



Répartition des observations



Répartition du défaut



Répartition du défaut

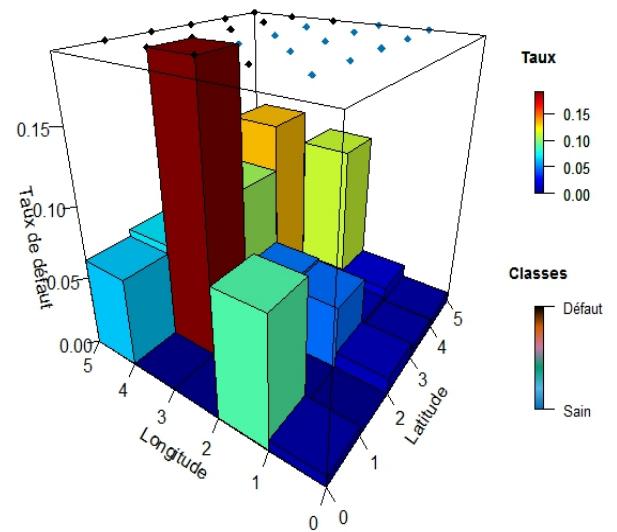


FIGURE 3.8 – Base d'apprentissage

D'ailleurs, une matrice de confusion est construit sur la base de test pour manifester la qualité de classification de la carte Kohonen supervisée. Chaque colonne de la matrice représente le nombre d'occurrences d'une classe réelle lorsque chaque ligne représente le nombre d'occurrences d'une classe estimée.

FIGURE 3.9 – Base de test

	Classe réelle 0	Classe réelle 1
Classe estimée 1	FP (faux positifs)	VP (vrais positifs)
Classe estimée 0	VN (vrais négatifs)	FN (faux négatifs)

Tableau 3.1 – Définition : Matrice de confusion

La matrice nous donne les informations suivantes : sur les 5480 clients sains, 3887 seront estimés comme sains, soit 71% des clients sains sont correctement prédits ; et sur les 189 clients défaillants, 133 seront estimés comme défaillants, soit 70.4% des clients sains sont correctement prédits. En ce cas, plus de 70% défauts exacts sont posés dans les premiers 20% des clients prédits les plus susceptibles d'être défaillants. Ce résultat est de bonne performance, mais il nous reste à voir s'il existe des modèles traiter mieux des informations fournies.

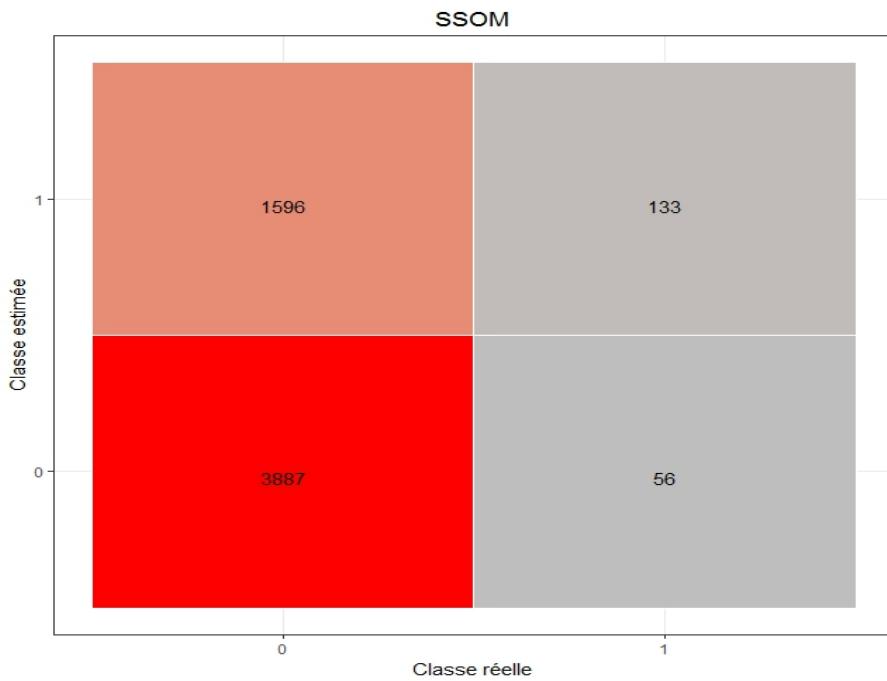


FIGURE 3.10 – Matrice de confusion : base de test

## 3.4 Évaluation des performances

Selon Schütze [10], la performance du regroupement peut être mesuré des deux côtés : la proportion des individus dominant en chaque groupe dans le total et point de vue de l'information mutuelle entre le regroupement et les classes exactes.

### 3.4.1 Indice de performance

Au cours du regroupement, il nous faut évaluer la qualité de regroupement pour la comparaison et le calibrage de paramètre. Les méthodes d'évaluation doivent être compétent pour mesurer la performance des réseaux générés et tolérer le cas où les neurones marquées de classe défaillante manquent comme on l'a vu avec l'échec de la carte Kohonen non supervisé. Du coup, trois critères

sont choisis pour faire l'évaluation : pureté, information mutuelle normalisée et indice de Rand. Sauf explication spécifique, les partitions ci-dessous sont deux à deux disjoints par défaut.

## Pureté

La pureté est un critère qui évalue la mesure dans laquelle les groupes contiennent une classe unique. Compte tenu d'un ensemble de groupes  $M$  et d'un ensemble de classes  $D$ , tous les deux divisent  $N$  individus. La pureté se calcule de manière suivante :

$$\frac{1}{N} \sum_{m \in M} \max_{d \in D} |m \cap d|$$

C'est la proportion des individus dominant en chaque groupe dans le total qui égale 1 lorsque le regroupement et les classes sont bien corrélés.

## Information mutuelle normalisée (NMI)

Contrairement à la pureté qui préfère le grand nombre de groupes, l'information mutuelle normalisée pénalise le grand nombre de groupes. En ce cas, la taille de la carte Kohonen se contrôle. Elle mesure la similarité entre deux partitions d'un ensemble. Compte tenu d'un ensemble de groupes  $U$  de  $R$  groupes et d'un ensemble de classes  $V$  de  $C$  classes, les entropies des deux partition se définissent par :

$$H(U) = - \sum_{i=1}^R P(i) \log P(i); \quad H(V) = - \sum_{j=1}^C P'(j) \log P'(j)$$

Et l'information mutuelle de deux partitions se définit par :

$$MI(U, V) = \sum_{i=1}^R \sum_{j=1}^C P(i, j) \log \frac{P(i, j)}{P(i)P'(j)}$$

$$NMI(U, V) = \frac{MI(U; V)}{[H(X) + H(Y)]/2}$$

Le regroupement qui possède le NMI grand sera bien corrélé avec les classes au sens de la dépendance statistique, ce qui indique une bonne qualité de la classification.

## Indice de Rand (RI)

L'indice de Rand mesure aussi la similarité entre deux partitions d'un ensemble. L'enjeu de ce critère est de mesurer la consistance (le taux d'accord) entre deux partitions. Nous notons deux partitions  $U$  et  $V$  :

$U \setminus V$	Co-groupée	Non co-groupée
Co-groupée	a	b
Non co-groupée	c	d

L'indice de Rand se calcule par :

$$RI(\pi_1, \pi_2) = (a + d) / \binom{n}{2}$$

C'est la proportion des paires co-groupées dans tous les paires de U et V. Un grand indice indiquera une bonne qualité de classification.

### 3.4.2 Comparaison des performances

Indice \ Méthode	SOM	SSOM	<i>SSOM*</i>
Pureté	0.967	0.967	1
NMI	0.00179	0.0126	0.421
RI	0	0.00647	0.185

Tableau 3.2 – Base d'apprentissage

Indice \ Méthode	SOM	SSOM
Pureté	0.967	0.967
NMI	0.00283	0.0119
RI	0.00145	0.00647

Tableau 3.3 – Base de test

*SSOM\** : Carte Kohonen supervisée sur la base d'apprentissage sous échantillonnée

Selon les tableau ci-dessus, l'effet de la combinaison du sous-échantillonnage et l'apprentissage supervisé de SSOM est vraiment évident. La carte *SSOM\** possède les meilleures performances sur toutes les mesures. Les trois mesure de performance de *SSOM\** sont toutes supérieures que celles de carte Kohonen. Néanmoins, il nous faut remettre les individus exclus par le sous échantillonnage pour faire la prévision. En ce cas, l'effet de sous-échantillonnage s'affaiblit et une baisse de la performance est observée. Cependant, l'effet du apprentissage supervisé de SSOM reste dans toutes les deux bases de données, ce qui permet une meilleure performance de SSOM sur la base complète que celle de SOM classique.

Dans le chapitre suivant, nous explorerons l'apprentissage supervisé pour étudier comment profiter mieux de l'interaction entre des entrées et les classes. Dans le chapitre 5, nous introduirons une méthode hybride qui essaie de garder l'effet de sous échantillonnage en utilisant la carte Kohonen lorsque l'apprentissage se déroule de manière supervisée.

# Chapitre 4

## Apprentissage supervisé

Ce chapitre présente les algorithmes d'apprentissage supervisé qui cherche à produire automatiquement des règles à partir d'une base de données d'apprentissage. Pour la classification, le but des algorithmes d'apprentissage supervisé est d'estimer le lien entre des entrées  $x$  et des classes  $y$  grâce à la fonction de prédiction (le classifieur) apprise des données connus  $f(\cdot)$ . En général, cette fonction de prédiction se détermine dans un espace de fonctions, soit espace d'hypothèses, en minimisant un risque moyen  $E[l(f(x), y)]$  défini par la fonction de perte  $l(f(x), y)$ .

D'après le chapitre précédent, il existe deux enjeux pour la construction de classifieur : l'information de classe et la séparation non linéaire. En effectuant l'apprentissage supervisé, nous profitons bien de l'information de classe des entrées. Quant à la séparation non linéaire, quatre classifieurs non linéaires sont choisis : arbre de décision, régression logistique pénalisée, forêts aléatoires et gradient boosting. Nous rappelons dans les sections suivantes les principaux des algorithmes utilisés et définitions usuelles.

### 4.1 Évaluation des performances

#### Courbe ROC

La courbe ROC est un outil d'évaluation et de comparaison des modèles qui est associée à la matrice de confusion introduite dans le chapitre précédent. La définition de la courbe de taux nécessite de calculer le taux de vrais positif (TVP) et le taux de vrais négatifs (TVN).

$$\text{TVP} = \text{Sensibilité (sensitivity)} = \text{VP/Positifs}$$

$$\text{TVN} = \text{Spécificité (specificity)} = \text{VN/Négatifs}$$

La construction de la courbe ROC consiste à faire varier le « seuil » de classification de 1 à 0 et, pour chaque cas, calculer le TVP et le TVN que l'on reporte dans un graphique : en abscisse le TVN, en ordonnée le TVP.

L'aire sous la courbe (AUC) est la surface de l'aire située sous le tracé de la courbe ROC dessinée dans un repère. Elle indique la probabilité pour que le classifieur  $f(\cdot)$  place un positif devant un négatif (AUC = 1 pour le meilleur classifieur). Cette mesure sera aussi inclue dans notre évaluation.

## Indice de Gini

L'indice de Gini développé par le statisticien Corrado Gini mesure la qualité de la classification binaire. Il est le rapport des deux aires suivantes : A : aire entre la courbe du modèle construit et la droite de l'aléatoire B : aire entre la courbe du modèle idéal et la droite de l'aléatoire :

$$0 \leq Gini = \frac{A}{B} \leq 1$$

Ce indice s'approche vers 1 quand la performance du modèle est meilleur.

## 4.2 Calibrage des paramètres

Le calibrage des paramètres est le processus de sélection des valeurs pour les paramètres d'un modèle qui maximisent la précision du modèle. Deux méthodes sont implémentées pour que les paramètres soient déterminés au coût acceptable. Elles s'appliquent aux modèles différents : l'AICc pour les modèles de régression et la recherche par quadrillage pour le reste.

### AICc

Le critère d'information d'Akaike (AIC) mesure la qualité d'un modèle statistique en prenant compte à la fois de la vraisemblance et du nombre de paramètres. Il se définit par :

$$AIC = 2k - 2 \ln(L)$$

où k est le nombre de paramètres à estimer du modèle et L est le maximum de la fonction de vraisemblance du modèle. Il est efficace pour traiter les modèles imbriqués.

L'AICc est une correction de l'AIC prenant également compte du nombre des observations  $n$  :

$$AICc = AIC + \frac{2k(k+1)}{n-k-1}$$

### Recherche par quadrillage

Puisque la plupart des classificateurs non linéaires ne sont pas des modèles imbriqués, l'AICc n'est pas adapté aux tels modèles. Prenant compte du coût d'apprentissage, nous décidons de faire la recherche par quadrillage combiné avec «holdout method » pour le calibrage des modèles autre que la régression. C'est à dire, une recherche exhaustive est effectuée à travers un sous-ensemble spécifique de l'espace de paramètre. La mesure de performance est mise en place sur la base de test. Les paramètres qui possèdent la meilleure performance sur la base de test seront choisis.

Par exemple, un modèle de deux paramètres  $C$  et  $\gamma$  s'applique à la classification. Les deux paramètres sont continus, afin d'effectuer une recherche par quadrillage, on sélectionne un ensemble fini de

valeurs "raisonnables" pour chacun d'eux :  $C \in \{10, 100, 1000\}$ ;  $\gamma \in \{0.1, 0.2, 0.5, 1.0\}$ . La recherche aura lieu dans l'ensemble des paires de  $\{10, 100, 1000\}$  et  $\{0.1, 0.2, 0.5, 1.0\}$ , soit  $\{(10, 0.1), (10, 0.2), (10, 0.5), (10, 1.0), (100, 0.1), (100, 0.2), (100, 0.5), (100, 1.0), (1000, 0.1), (1000, 0.2), (1000, 0.5), (1000, 1.0)\}$ . La paire de paramètres ayant la plus meilleure sera choisie.

## 4.3 Erreur de généralisation

L'objectif d'apprentissage est de trouver une fonction  $f(x)$  qui permet de prédire les valeurs de sortie  $y$  basées sur des entrées  $x$ . Le risque moyen  $E[l(f_n)]$  d'une telle fonction  $f_n$  se définit comme suit :

$$E[l(f_n)] = \int_{X \times Y} l(f_n(x, y)) \rho(x, y) dx dy$$

où  $\rho(x, y)$  est la loi conjointe de probabilité inconnue pour  $x$  et  $y$ .

Puisque la loi conjointe de probabilité  $\rho(x, y)$  est inconnue, nous ne pouvons que calculer le risque moyen empirique sur la base d'apprentissage  $S$  :

$$E[l_S(f_n)] = \frac{1}{n} \sum_{i=1}^n l(f_n(x_i), y_i)$$

L'erreur de généralisation  $G$  est donc la différence entre le risque moyen sur la loi de probabilité conjointe sous-jacente de  $x$  et  $y$  le risque moyen empirique sur la base d'apprentissage. Elle est définie par :

$$G = E[l(f_n)] - E[l_S(f_n)]$$

Les algorithmes d'apprentissage sont toujours utilisés pour minimiser  $E[l_S(f_n)]$ . Cependant, il faut faire l'attention à l'erreur de généralisation qui nous permet d'éviter le sur-apprentissage sensible au bruit de la base de l'apprentissage. Cependant, elle ne peut pas être mesurée directement. C'est pourquoi les validations croisées et l'évaluation de performance sur les deux bases de données sont effectuées sur l'échantillon de validation pour expliciter l'erreur de généralisation. D'un autre côté, il est important d'éviter les problèmes de surapprentissage et de sousapprentissage entraînés par l'erreur de généralisation. Le sousapprentissage doit être éviter par le biais du calibrage de paramètres. Le surapprentissage se résoudra dans le cadre de la régularisation de modèles.

## 4.4 Modèle d'apprentissage

### 4.4.1 Modèle de référence : Arbre de décision

L'arbre de décision [7] est un classifieur classique pour la séparation non-linéaire des entrées, ce qui lui fait un bon candidat pour le modèle de référence. Elles se basent sur un découpage de l'espace engendrée par les variables en utilisant les hyperplans parallèles aux axe. Chaque nœud intérieur de l'arbre représente un découpage, et les nœuds terminaux représentent une valeur des classes pour la classification. Le découpage s'effectue en divisant l'arbre du sommet vers les les nœud terminaux en choisissant à chaque étape une variable d'entrée qui réalise le meilleur partage de l'ensemble des

observations selon certain critère.

La fonction de perte de l'arbre de décision est /

$$l(f(x), y) = \begin{cases} 1 & \text{if } f(x) = y \\ 0 & \text{if } f(x) \neq y \end{cases}$$

Le critère de l'entropie de Shannon est choisi pour le découpage :

$$MI(X, Y) = \sum_{x \in X} \sum_{y \in Y} P(x, y) \log \frac{P(x, y)}{P(x)P'(y)}$$

Pour lutte contre les classes déséquilibre, un critère modifié sont construit à partir du critère original en imposant les poids de classes :

$$MI(X, Y) = \sum_{x \in X} \sum_{y \in Y} w(y)^{-1} P(x, y) \log \frac{P(x, y)}{P(x)P'(y)}$$

Après le calibrage de paramètres, les paramètres  $w(y) = \{0.7, 0.3\}$  se trouve le plus raisonnable. La paire de paramètres se choisit de dix paires  $\{0.1, 0.9\}, \{0.2, 0.8\} \cdots \{0.9, 0.1\}$ . Le résultat de l'arbre généré est comme suit :

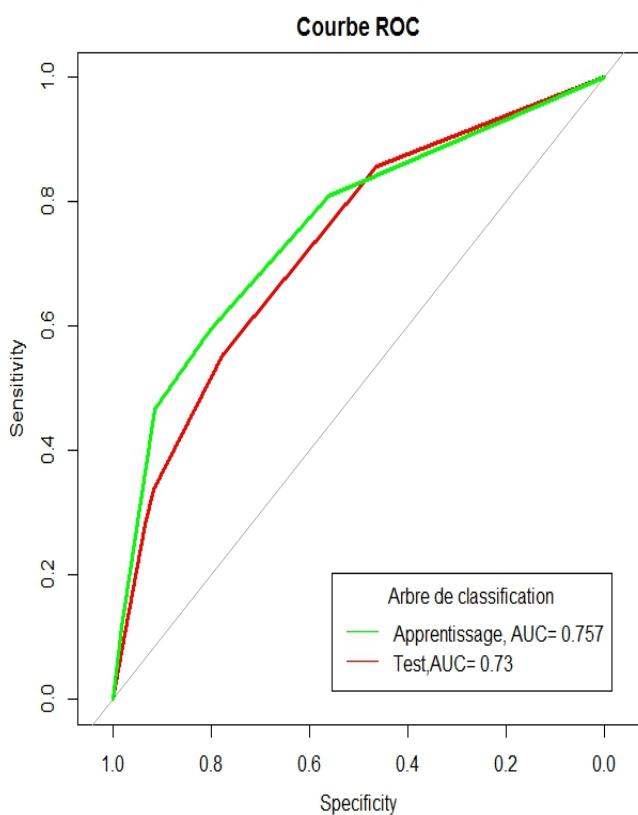


FIGURE 4.1 – Courbe ROC

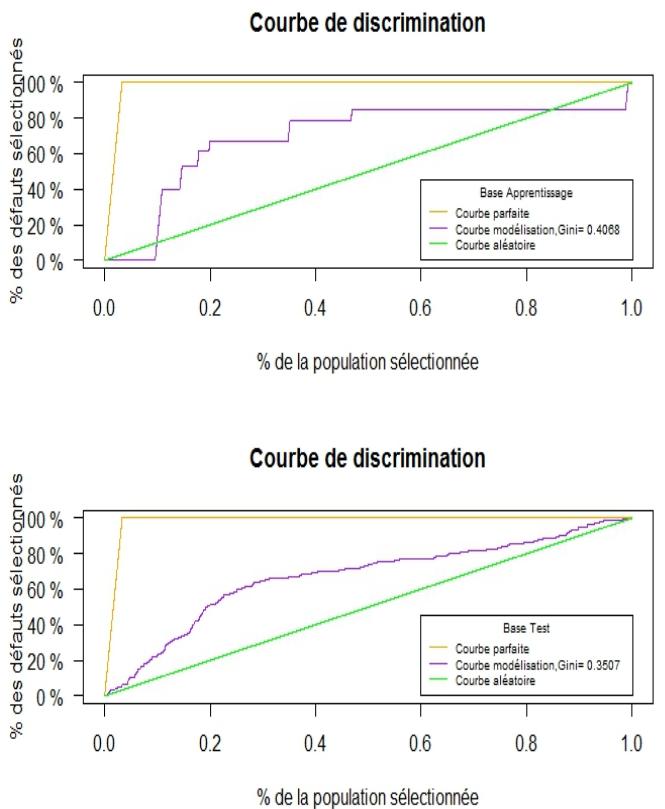


FIGURE 4.2 – Courbe Gini

#### 4.4.2 Régression logistique pénalisée

La régression logistique [9] est un modèle largement utilisé lorsque la réponse est catégorique. Cependant, les modèles de régression se trouve inappropriés pour des entrées de haute dimensions

à cause des problèmes inverses. C'est pourquoi une régression logistique pénalisée est introduite. Le but de la régression logistique est de minimiser le risque moyen empirique :

$$\inf_{\beta_0, \beta} \frac{1}{N} \sum_{i=1}^N l(y_i, \beta_0 + \beta^T x_i)$$

où  $l(y, \beta_0 + \beta^T x)$  est la perte logarithmique de fonction logistique :

$$l(y, \beta_0 + \beta^T x) = -y\beta^T x + \ln(1 + e^{y\beta^T x})$$

En ajoutant une terme de pénalisation de type Lasso dans l'équation, un modèle de la régression logistique pénalisée est élaboré pour minimiser le risque moyen empirique pénalisé :

$$\inf_{\beta_0, \beta} \frac{1}{N} \sum_{i=1}^N l(y_i, \beta_0 + \beta^T x_i) + \lambda \|\beta\|_1$$

Un algorithme particulier est appliqué à la minimisation du risque moyen empirique.[9]. Le modèle avec le meilleur AICc est choisi pour la classification et le résultat est comme suit :

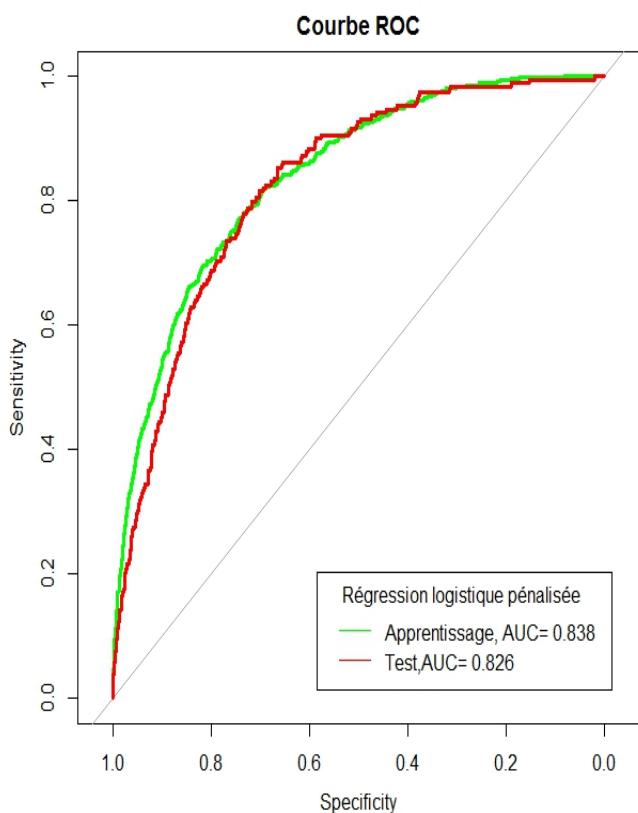


FIGURE 4.3 – Courbe ROC

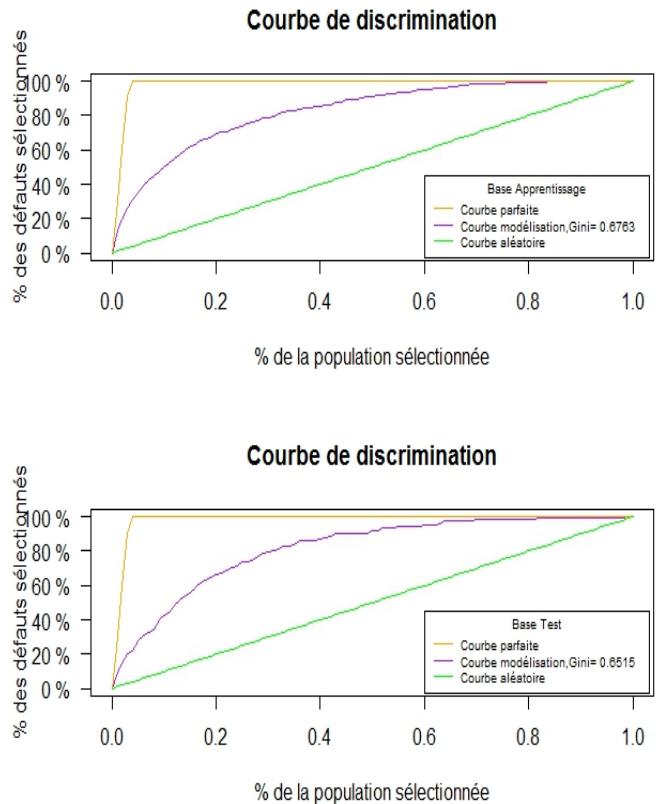


FIGURE 4.4 – Courbe Gini

Conformément à notre attente, la régression pénalisée améliore la performance de la classification. La performance d'arbre de décision est plutôt modeste avec un AUC de 0.73 sur la base de test. Par contre, la performance de la régression pénalisée atteint un AUC de 0.826 sur la base de test qui est

proche de celui de 0.838 sur la base d'apprentissage, ce qui montre une petite erreur de généralisation.

#### 4.4.3 Forêts aléatoires

La section précédente a introduit un modèle de référence qui utilise l'arbre de décision. À partir de ce modèle, le modèle de des forêts aléatoires [8] construit en appliquant une multitude d'arbres de décision aux sous ensembles des variables explicatives dans un cadre de l'agrégation de modèles. La fonction de perte de l'arbre de décision est donc :

$$l(f(x), y) = \sum_{n=1}^N l_n(f(x), y)$$

où  $l_i(f(x), y)$  est la perte de  $n$ ième arbre dans les forêts.

Trois paramètres sont choisis pour situer le modèle : nombre d'arbre  $N$ , nombre minimal de nœuds terminaux d'arbre  $na$  et nombre maximal de nœuds terminaux des forêts  $nf$ . Après la recherche par quadrillage sur  $N = \{100, 200, \dots, 800\} \times na = \{5, 6, \dots, 15\} \times nf = \{10, 15, \dots, 30\}$ , des forêts aléatoires du triplet (800, 10, 25) sont choisi pour la classification.

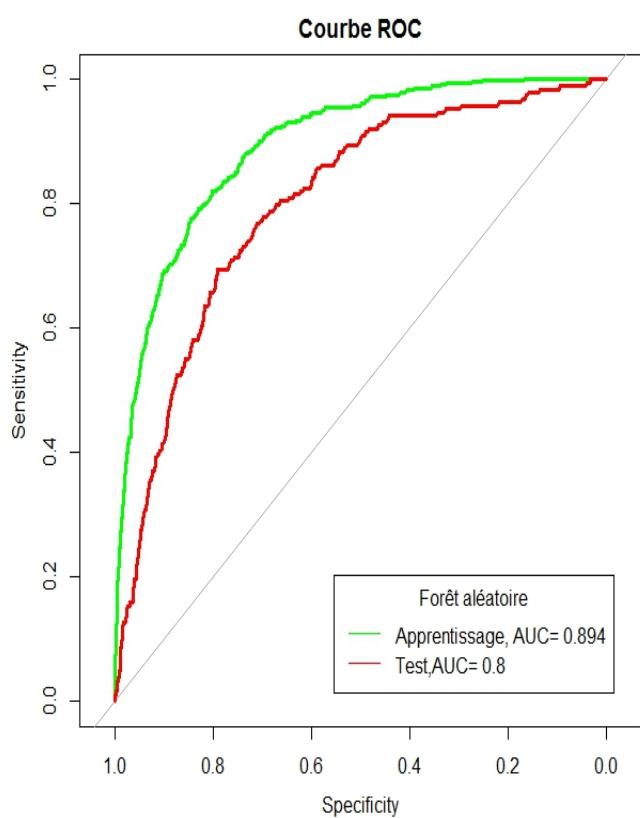


FIGURE 4.5 – Courbe ROC

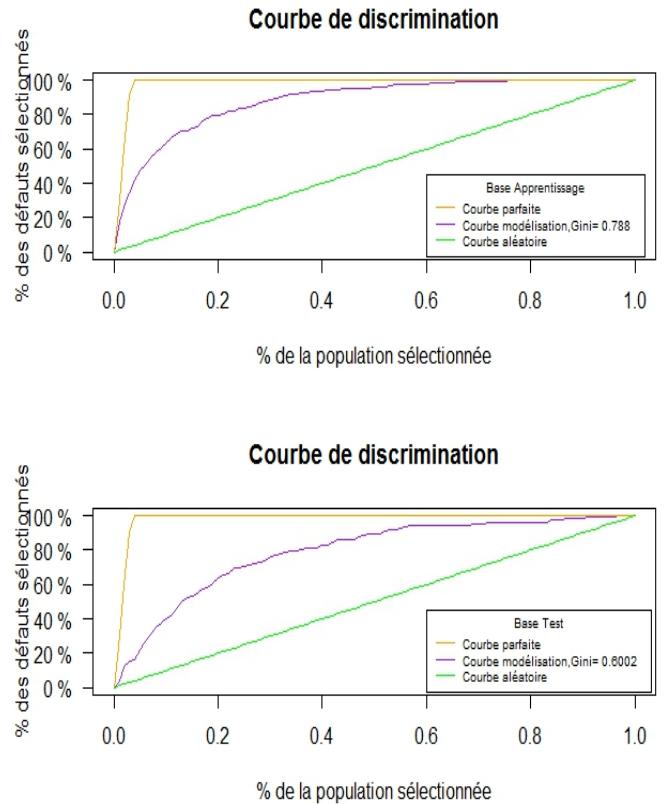


FIGURE 4.6 – Courbe Gini

L'amélioration de la performance des forêts aléatoires est évident par rapport au modèle de référence. Cependant, par rapport à la régression logistique pénalisée, l'erreur de généralisation augmente

au fur et à mesure que une augmentation de la performance dans la base d'apprentissage, ce qui nous fait un souci d'apprentissage basé sur des arbres de décision.

#### 4.4.4 Gradient Boosting

Alors que l'algorithme de forêts aléatoires utilise la classe majoritaire, le Boosting effectuera un vote majoritaire pondéré. La fonction de perte du Boosting est donc :

$$l(y, w) = \sum_{n=1}^N w_n l_n(f(x_n), y_n)$$

Il s'agit d'une itération pour déterminer  $w$  et  $f$  :

$$\begin{aligned} F_0(x) &= \arg \inf_w \sum_{n=1}^N l(y_n, w), \\ F_m(x) &= F_{m-1}(x) + \arg \inf_{f \in \mathcal{H}} \sum_{n=1}^N l(y_n, F_{m-1}(x_n) + f(x_n)) \end{aligned}$$

Cependant, La recherche de la solution sera coûteux. En ce cas, nous recourons à une simplification dont la fonction de perte se limite dans les fonctions différentiables au lieu de l'espace d'hypothèses entier. Ici, la perte logistique sera utilisée pour remplacer la perte de 0-1. Une étape de descente plus abrupte est appliquée à ce problème de minimisation comme ce que fait dans la recherche linéaire. En ce cas, l'entraînement se déroulera de la manière suivante :

$$\begin{aligned} F_m(x) &= F_{m-1}(x) - w_m \sum_{n=1}^N \nabla_{F_{m-1}} l(y_n, F_{m-1}(x_n)), \\ w_m &= \arg \min_w \sum_{n=1}^N L(y_n, F_{m-1}(x_n) - w \nabla_{F_{m-1}} l(y_n, F_{m-1}(x_n))) , \end{aligned}$$

Puisque nous prenons la pertes logistique ici, le gradient  $\nabla_{F_{m-1}}$  se définit par :

$$\nabla_{F_{m-1}} l(y_n, F_{m-1}(x_n)) = \frac{\partial l(y_n, F_{m-1}(x_n))}{\partial F_{m-1}(x_n)} = \frac{y - (1-y)e^{F_{m-1}(x_n)}}{1 + e^{F_{m-1}(x_n)}}$$

Pour bien situer le modèle, trois paramètres sont choisis : taux d'apprentissage  $\eta$ , réduction minimale de perte pour la partition  $\gamma$  et la taille maximal d'arbre individu  $md$ . Après la recherche par quadrillage sur  $\eta = \{0, 0.05, \dots, 0.3\} \times \gamma = \{0, 0.04, \dots, 0.2\} \times md = \{4, 6, \dots, 10\}$ , le modèle des paramètre du triplet  $(0.05, 0.08, 6)$  est choisi pour la classification.

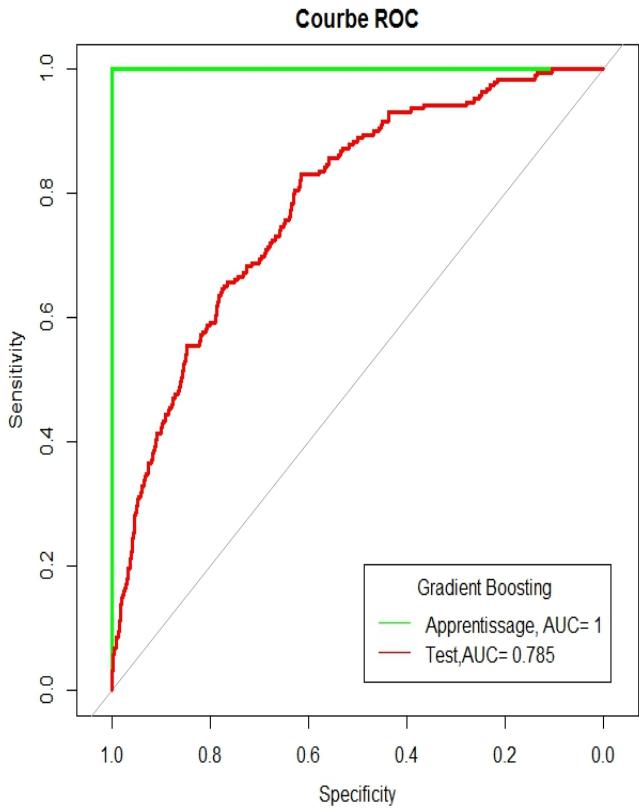


FIGURE 4.7 – Courbe ROC

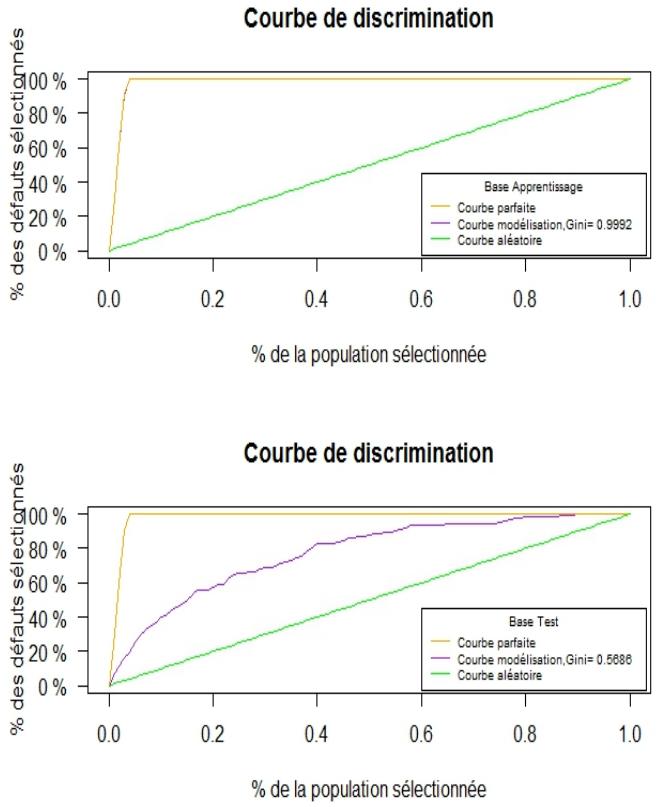


FIGURE 4.8 – Courbe Gini

Le surapprentissage est observé selon la courbe ROC et la courbe Gini, ce qui nous montre que une sélection de variables est nécessaire pour éliminer du bruit dans la base d'apprentissage, soit les variables non discriminantes.

## 4.5 Sélection de variables

La sélection de variables consiste à trouver un sous-ensemble "pertinent" de variables parmi celles de l'ensemble de départ. Il existe principalement trois catégories des algorithmes de sélection : "wrapper", "filter", et "embedded". À cause du temps coûteux pour effectuer les méthodes "wrapper", nous ne prenons compte que six méthodes de deux dernières parmi lesquelles deux sont de catégorie "filter" et les autres appartiennent à la catégorie "embedded".

### 4.5.1 Filter

#### Filter de variance

L'élimination des variables de basse variance est justifiée par Kuhn et al [12]. Elle est une méthode naïve mais robuste pour les modèles variés. Au lieu de l'élimination, nous donnons une note à chaque variable qui égale la proportion des variances dans ses sommes.

## Critère de $\chi^2$

Pour utiliser le critère de  $\chi^2$ , nous devons transformer des variables quantitatives en variables catégorielles. Ici, chaque variable quantitative est regroupée en 5 catégories selon sa grandeur. Le test de  $\chi^2$  [10] sert à déterminer l'existence d'une relation entre deux variables catégorielles : variable explicative  $x$  et variable des classes  $y$ .

Un tableau de contingence est construit pour les deux variables. Notons  $O_{ij}$  l'effectif observé de données pour lesquelles  $x$  prend la valeur  $i$  et  $y$  la valeur  $j$ . Une valeur d'espérance d'occurrence  $E_{ij}$  sous l'hypothèse d'indépendance est définie par :

$$E_{ij} = \frac{O_{i+} \times O_{+j}}{N}$$

où  $O_{ij}$  est nombre d'observations pour lesquelles  $X = i$  et  $O_{+j}$  est nombre d'observations pour lesquelles  $Y = j$ . En utilisant les valeurs observées  $O_{ij}$  et  $E_{ij}$ , la statistique de  $\chi$  sera construite :

$$T = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

L'inverse de p-valeur du test servira la mesure de l'importance des variables. Comme les traitement précédents, la sommes des inverse valeurs-p sera normalisée à 100.

### 4.5.2 Embedded

#### Sélection de type lasso

Dans la section précédente, on a déjà effectué une régression logistique pénalisée du terme de Lasso.[13] L'avantage de cet régression réside dans le fait que la pénalisation impose une loi priori de Laplace aux coefficients de Laplace. De cette manière, les coefficients de variables non significatives tendent à 0 à cause d'une loi de priori plus pointue. Nous ferons une normalisation selon les coefficients en transformant ses somme en 100. Les coefficients normalisés serviront à la mesure de l'importance de variables.

#### Sélection de type forêts aléatoires

Les forêts aléatoires peuvent aussi faire une sélection des variables.[11] La réduction de Gini moyenne est la moyenne de la contribution d'une variable dans la réduction de Gini-gain. La réduction de Gini-gain aura lieu quand un variable utile est enlevé du modèle. Donc les variables avec les grande réduction de Gini moyenne seront les variables les plus importantes. De même manière que la méthode précédente, la somme des réductions de Gini moyennes se transforme en 100.

#### Sélection de type gradient boosting

L'importance des variables est évaluée en calculant la contribution des variables à l'amélioration de précision.[14] En ajoutant un nœud à une variable discriminant, la précision du classifieur s'augmente.

La proportion de la contribution des variables à la précision augmentée nous indique l'importance des variables. Cette proposition est donc proposée pour distinguer les variables discriminantes.

## Sélection stabilisante

La sélection stabilisante est une nouvelle approche[15] basée sur le bootstrap et le modèle de base. Le modèle de base peut être un classifieur arbitraire. Elle consiste à appliquer le modèle de base aux sous-ensembles des observations et variables, ce qui est différent que les forêts aléatoire effectué sur seulement des sous-ensembles des variables. Le gradient boosting de perte logistique est choisi comme le modèle de base ici. La fréquence que une variable est considérée importante dans des modèles de base sert à la mesure de l'importance. Pour la comparaison, la somme de la fréquence se normalise à 100. Idéalement, la note des variables inutiles sera proche de zéro.

## Sélection moyenne

Nous entendons faire une nouvelle évaluation robuste depuis les six méthodes précédentes. Donc, la sélection de moyenne est de calculer une moyenne des notes précédentes pour évaluer l'importance des variables. En ce cas, l'importance des variables évaluée sera un vote des méthodes différentes.

### 4.5.3 Variables plus importantes

Nous construisons sept ensembles de variables pour vérifier l'effet de la sélection des variables. Le nombre maximale des éléments des ensemble est 172 qui provient du "filter" de gradient boosting. Dans le classifieur de gradient boosting, seulement 172 qui participent à la prévision. Nous profitons donc de cette troncation au sens de variables en le jugeant objective malgré propre au modèle.

Après le calcul de sept mesures, nous choisissons les 20 variables les plus importantes dans l'ensemble des variables sélectionné par le critère de la sélection moyenne. Nous pouvons voir que les sept méthodes ne sont pas toujours d'un commun accord, ce qui fait varier la performance de classification sur des ensembles différentes. Ici, les variables TOP\_INCD\_SIREN, TOP\_INCD\_12M\_SIREN\_C, etc représentent les donnée internes de la catégorie compte qui occupe une majorité. D'ailleurs, les variables externes comme COT\_BDF\_ENTREP nous aident aussi à la classification mais en jouant un rôle supplémentaire. L'exploration des données externes sera un développement potentiel.

Variable \ Méthode	glmImp	rfIMP	xgbImp	stabImp	varImp	chiImp	moyImp
COT_BDF_ENTREP.Sup	15.77	0.08	0.01	10.31	0.00	4.60	5.13
TOP_INCID_SIREN.0	17.54	0.08	0.02	9.79	0.00	3.02	5.08
NOTE_MOTEUR_T	3.55	2.66	4.38	10.50	0.51	6.92	4.75
TOP_INCD_12M_SIREN_C.0	6.53	1.87	0.01	10.61	0.04	5.99	4.17
NBJ_DEPASS_MOIS_SIREN_12M_2	1.97	4.79	0.10	10.39	0.51	5.06	3.80
NB_JOUR_DEBT_SIREN_12M	1.14	3.51	1.68	7.67	0.51	4.62	3.19
ALLUR_CAV	0.97	1.24	0.08	7.53	0.51	4.00	2.39
NOTCOF	2.52	0.37	0.10	6.92	0.51	2.32	2.12
COT_BDF_ENTREP.5	4.01	0.06	0.01	3.31	0.03	1.36	1.46
COT_BDF_ENTREP.3	7.81	0.10	0.02	0.00	0.04	0.75	1.45
TOP_INCD_12M_SIREN_C.0.1	0.00	2.53	0.00	0.00	0.04	5.99	1.43
TOP_INCD_12M_SIREN_C.1.1	0.74	1.66	0.00	0.00	0.04	5.99	1.40
SLDE_MOY_SIREN_3M	0.17	5.57	1.47	0.16	0.51	0.08	1.33
TOP_INCID_SIREN.1	0.00	0.07	0.00	4.34	0.00	3.02	1.24
MNT_EE_CLI_12M_SIREN	0.91	2.16	0.75	2.29	0.51	0.33	1.16
SLDE_MOY_SIREN_1M	0.00	4.37	1.50	0.00	0.51	0.17	1.09
NOTMOT	0.00	1.14	0.09	0.09	0.51	4.45	1.05
COD_ACT_REF.6	5.85	0.00	0.05	0.08	0.01	0.14	1.02
PD_COFACE	0.00	0.41	0.09	2.80	0.51	1.98	0.96
RSS1	0	1.29	0.64	3.03	0	0	0.83

Tableau 4.1 – Sélection des variables

Les notations des méthodes s'écrivent comme suit :

varImp : Filter de variance, chiImp : Critère de  $\chi^2$ , glmImp : Sélection de type lasso, rfIMP : Sélection de type forêts aléatoires, xgbImp : Sélection de type gradient boosting, stabImp : Sélection stabilisante, moyImp : Sélection de moyenne.

## 4.6 Résultat des modèles modifiés

Les modèles originaux sont modifiés en se restreignant aux variables des ensembles construits par une des 7 méthodes. L'erreur de généralisation diminue et la performance de la base de test s'améliore dans la plupart des cas. Parmi les modèles modifiés, la régression logistique pénalisée est la plus performante. Nous la choisissons pour la validation croisée dans le but de la vérification de robustesse et de la comparaison avec notre nouveau modèle dans le chapitre 5. Puisque la classification est refaite sur les modèles de même paramètres, l'effet de la sélection des variables est validé. Cependant, la performance des méthodes de sélection face aux modèles différents varie.

Les méthodes dont la performance reste robuste et supérieure à celle des modèles originaux sont le "filter" de type lasso et le "filter" de la sélection moyenne. Donc nous utiliserons des modèles combinant des deux méthodes pour la validation croisée.

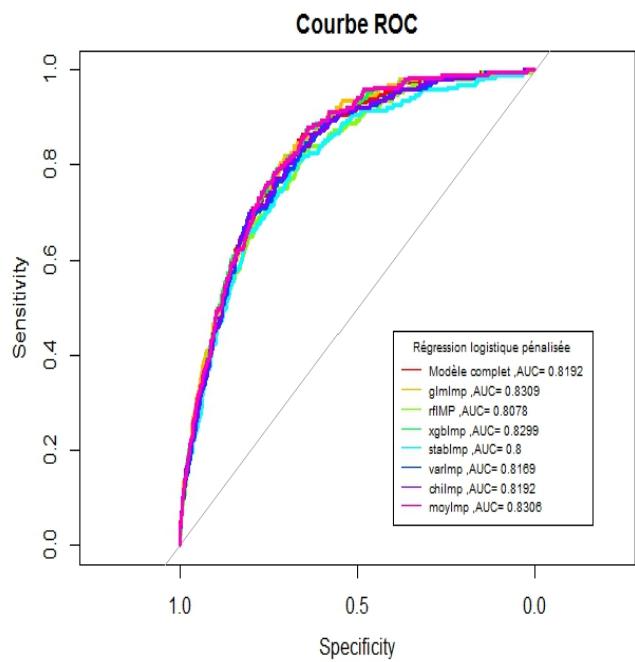
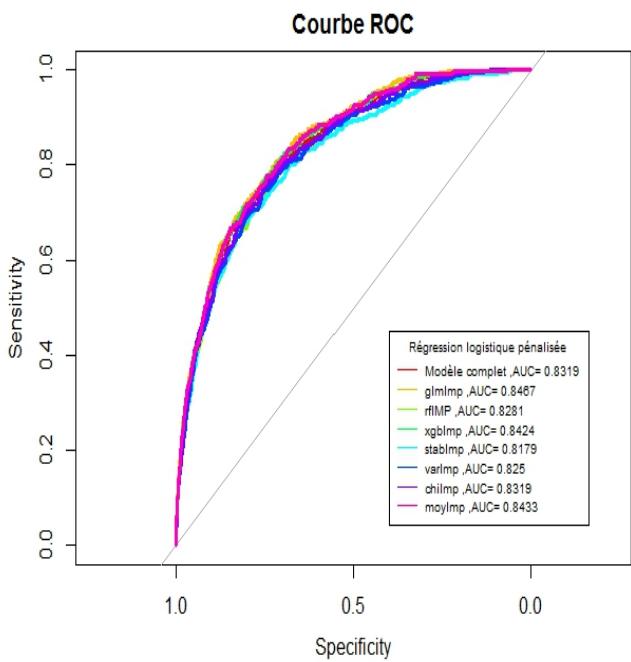
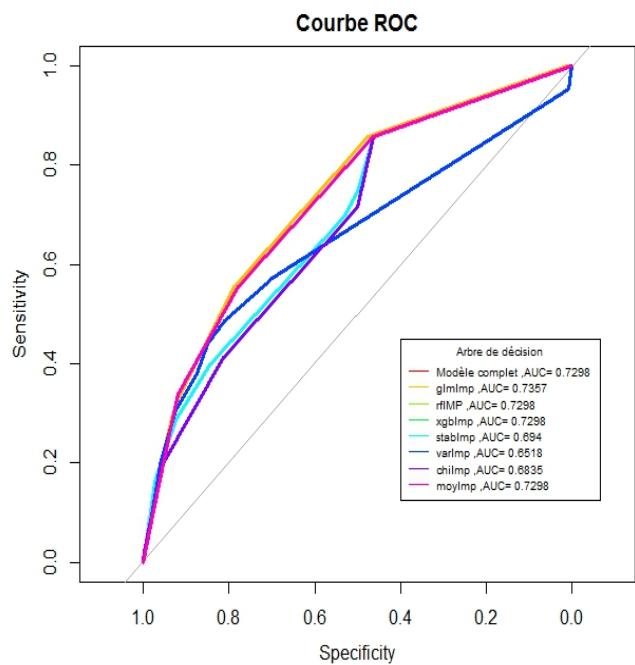
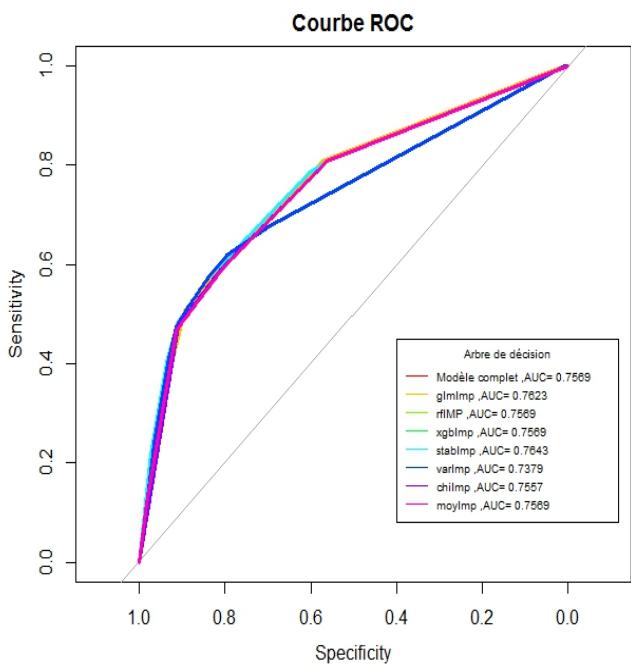


FIGURE 4.9 – Base d'apprentissage

FIGURE 4.10 – Base de test

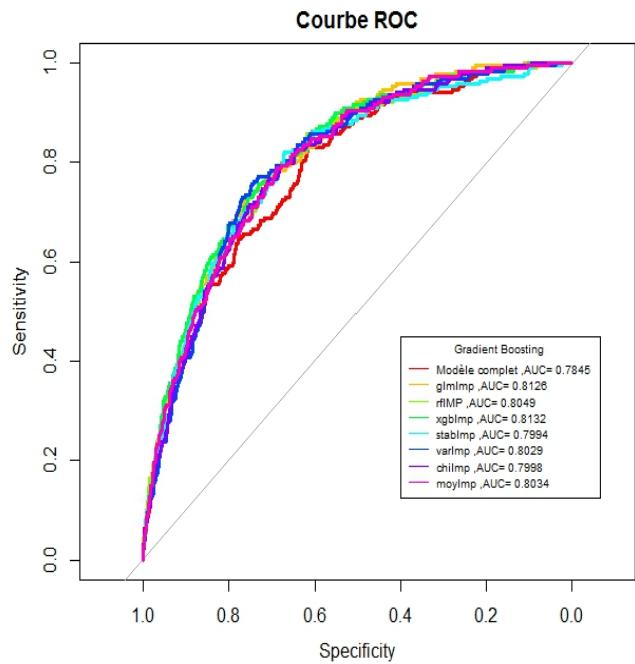
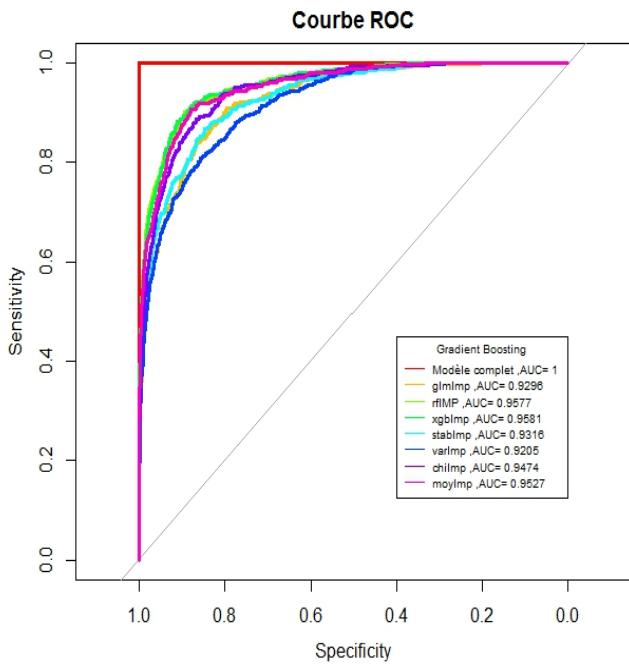
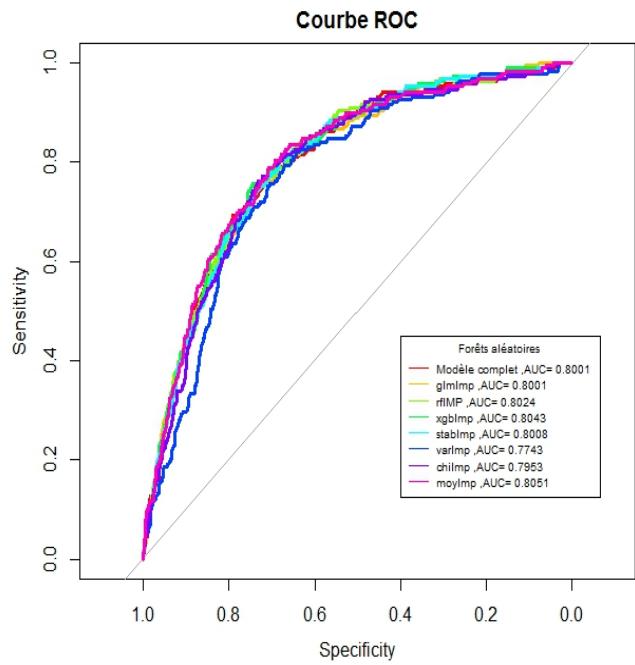
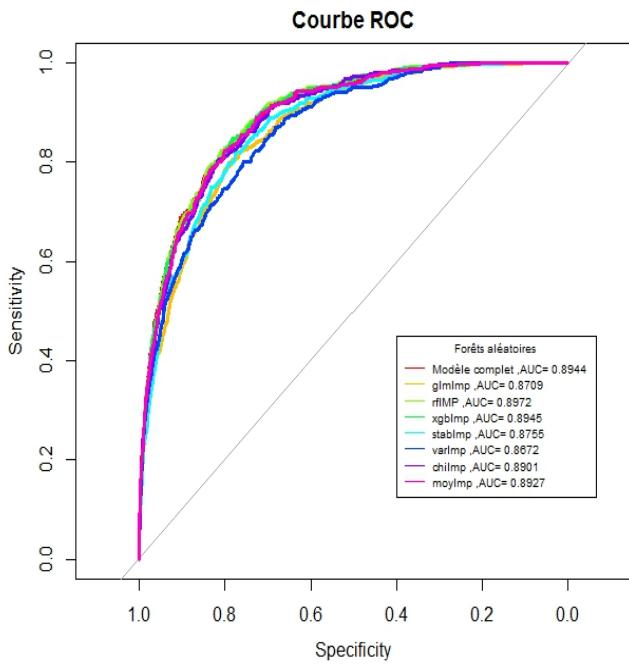


FIGURE 4.11 – Base d'apprentissage

FIGURE 4.12 – Base de test

# Chapitre 5

## Apprentissage hybride

Dans ce chapitre, nous présentons un cadre d'apprentissage hybride qui profite à la fois de l'apprentissage non supervisé et de l'apprentissage supervisé. Selon le chapitre 3, le sous-échantillonnage améliore la performance d'apprentissage. Cependant, cet avantage se réduit lorsque nous remettons les observations de la base originale exclues par le sous-échantillonnage. Nous voulons donc garder l'avantage de l'échantillonnage d'une manière moins directe pour éviter la baisse entraînée par la remise des observations exclues.

D'après l'avis de Chang[17], nous trouvons que un sous-échantillonnage peut être remplacé par une division des observations de classe majoritaire. La carte Kohonen sera un outil idéal pour la division grâce à sa compétence pour la préservation de la structure topologique. Basé sur la division, les ensembles locaux d'apprentissage qui comportent chacun un sous-ensemble des observations de classe majoritaire et le total des observations de classe minoritaire seront construits et ils seront plus équilibrés. Les variables synthétiques s'apprennent à partir des ensembles en utilisant des classificateurs locaux. Un classificateur final sera proposé pour apprendre le lien entre les classes et des variables synthétiques.

Un avantage de ce cadre est que le nombre des variables mises dans le classifieur final sont modélisés dépendant au lieu de entrée dépendant, ce qui nous aide à contrôler la dimension des entrées. D'un autre côté, l'apprentissage local coûtera moins de piles au meure d'apprentissage.

### 5.1 Construction des variables synthétiques

#### 5.1.1 Hétérogénéité

L'hétérogénéité géométrique est le cauchemar pour la classification des classes. Il est compliqué d'avoir une bonne précision de prévision s'il existe une certaine hétérogénéité dans la base d'apprentissage. Il est idéal de limiter les clients dans une sous-population la plus susceptible d'être défaillants sur laquelle nous faisons l'apprentissage. Cependant, nous disposons peu d'informations à priori sur les clients défaillants. Par conséquent, les clients sains des autres sous-populations sont pris dans la base d'apprentissage. En ce cas, un seul classifieur n'est pas suffisant pour la classification comme

montré dans la figure suivante. C'est pourquoi la division des clients sains est effectuée. La carte de Kohonen servira de la méthode de regroupement en profitant de son avantage sur l'étude de structure topologique.

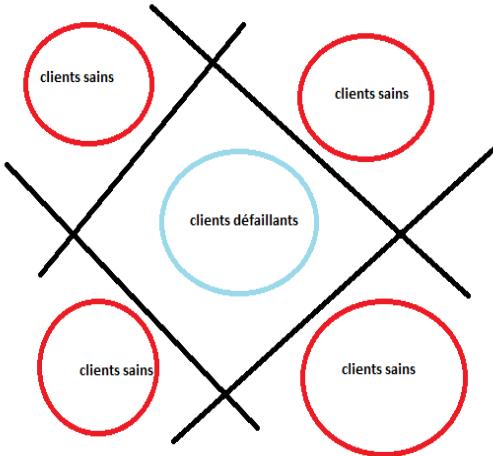


FIGURE 5.1 – Hétérogénéité géométrique

### 5.1.2 Classifieur local et distance signée

Un ensemble local comporte un groupe individuel des clients sains découpé par la carte Kohonen et le total des clients défaillant. Un classifieur local est imposé sur chaque ensemble local. D'après la détection de la structure topologique, la séparation d'un ensemble local est susceptible d'être non-linéaire. Selon Boczko et al[16], un classifieur local de SVM (Machine à vecteur supports) muni de noyau de RBF sera utilisé. De plus, la distance signée  $d(x)$  générée du classifieur local sert de variables synthétiques à entrer dans le classifieur final.

Le noyau de RBF de paramètre  $\sigma$  entre deux entrées  $x$  et  $x'$  s'écrit comme suuit :

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

Le SVM muni de noyau de RBF recherche à ramener un problème de classification à celui de trouver un hyperplan optimal  $\langle w, \varphi(x) \rangle + b = 0$  où  $\varphi(x)$  est une transformation de  $x$  dont le produit scalaire est le noyau de RBF  $\langle \varphi(x), \varphi(x') \rangle = K(x, x')$ . Le problème d'optimisation du SVM se définit comme suit :

$$\begin{cases} \inf_w & \frac{1}{2} \|w\|^2 \\ sc & \forall i, y_i (\langle w, x_i \rangle + b) \geq 1 \end{cases}$$

La distance signée  $d(x)$  est donc :

$$d(x) = \frac{\langle w, x_i \rangle + b}{\|w\|}$$

Cette distance est calculée sur le total des clients au lieu d'une partie que nous utilisons dans la construction des classificateurs locaux. En ce cas, les nouvelles caractéristiques par rapport aux hyperplans de séparation s'apprennent et donc sont propres au modèle.

## 5.2 Modèle de prévision

### 5.2.1 Procédure de modélisation

La régression logistique pénalisée s'avère être à la fois robuste et performante dans la classification. Nous la prenons pour le classifieur final. Les trois éléments principaux de modélisation sont donc déterminés : la division des observations de classe majoritaire, les classificateurs locaux et le classifieur final. L'algorithme d'apprentissage hybride est comme suit :

- 1) Faire un regroupement basé sur la carte Kohonen de  $9 \times 9$  sur les entrées de classe majoritaire  $\vec{w}$ . La taille de la carte est calibrée sur un quadrillage  $\{5, 6, \dots, 10\} \times \{5, 6, \dots, 10\}$ . La construction de la carte du chapitre 2 se fait une référence.
- 2) Construire les ensembles locaux qui se composent d'un sous ensemble des entrées de classe majoritaire et le total des entrées de classe minoritaire.
- 3) Dans chaque groupe, le SVM muni de noyau de RBF est entraînée pour construire un hyperplan de séparation locale
- 4) À partir des hyperplans locaux, calculer des distances signées pour tous les entrées
- 5) En utilisant des distances signées comme variables explicatives synthétiques, effectuer une régression logistique pénalisée pour la prévision de probabilité d'être défaillant.

### 5.2.2 Résultat empirique

Une petite erreur de généralisation est observée dans le modèle hybride. Comparé aux modèles d'apprentissage supervisé, la performance est plutôt satisfaisant avec les variables explicatives maximale de 81 au lieu de l'entier de variables explicatives.

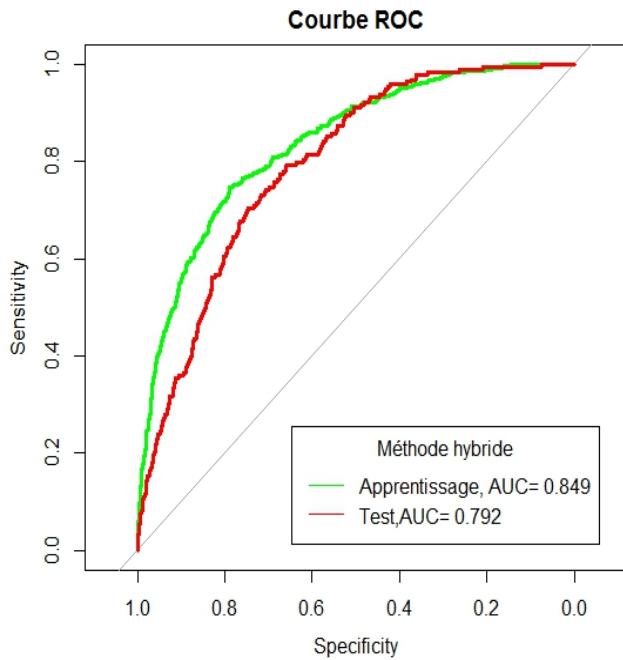


FIGURE 5.2 – Courbe ROC

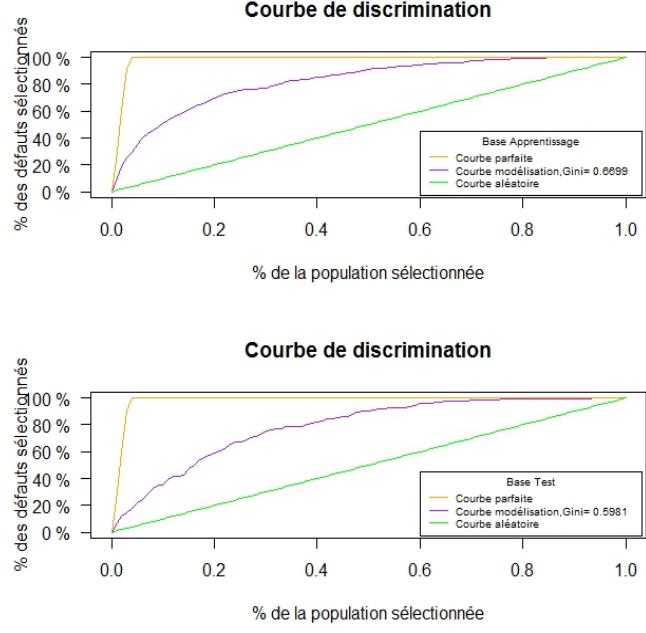


FIGURE 5.3 – Courbe Gini

Pour voir l'effet de la sélection des variables, le modèle est refait sur les variables les plus importantes. Les classifiants locaux se construisent sur l'espace des variables les plus discriminantes déterminées par des critères différents. Parmi lesquels, les performances des modèles des "filter" de lasso et "filter" de moyenne sont meilleur à la fois sur la base d'apprentissage et sur la base de test par rapport au modèle original. Donc nous choisissons les deux modèles modifiés pour la validation croisée suivante.

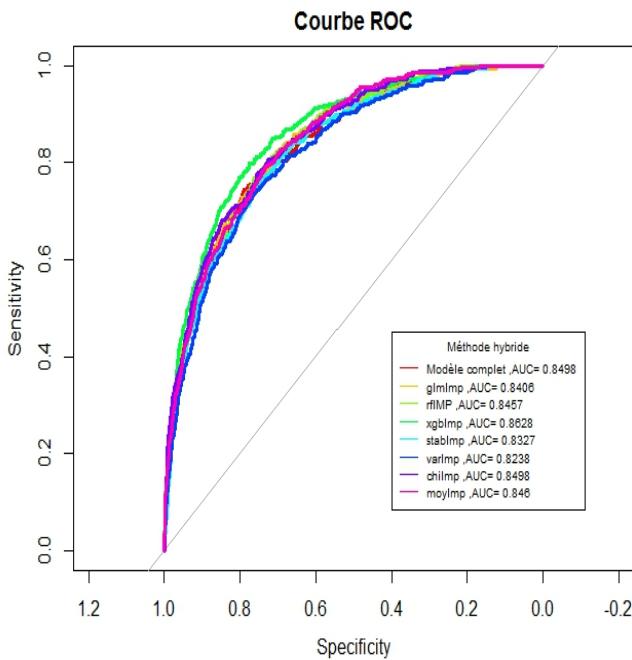


FIGURE 5.4 – Base d'apprentissage

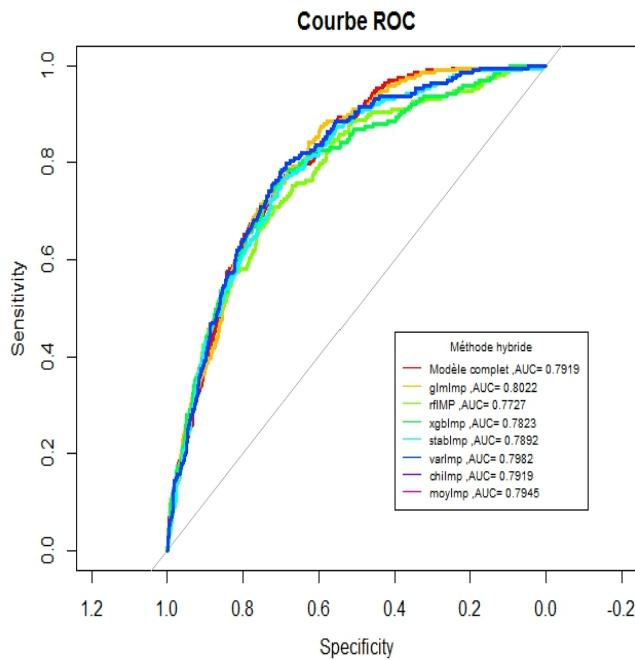


FIGURE 5.5 – Base de test

# Chapitre 6

## Validation croisée

### 6.1 Validation croisée Monte Carlo

La validation croisée consiste à vérifier la robustesse des modèle d'apprentissage statistique. La méthode la plus usuelle est validation croisée de k-fold. On divise la base d'apprentissage en k parties, et puis on sélectionne une partie comme base de validation et les (k-1) autres parties comme nouvelle base d'apprentissage. En répétant k fois cette opération, on obtiendra une suite de la mesure de performance comme AUC ou erreur quadratique pour la base de validation. La moyenne de la suite sera utilisée pour évaluer la performance. Normalement, il nous faut un grand k pour diminuer le biais. Cependant, les clients défaillants n'occupent que 3% du total et un grand k comme 10 rend la base de validation non représentatif à cause de la dispersion de répartition topologique des clients défaillants.

Par contre, la validation croisée Monte Carlo [18] stratifiée n'a pas de souci de sous-représentation. Du point de vue de Chen et al[19], la validation croisée Monte Carlo rend le modèle plus raisonnable. En effectuant un échantillonnage stratifié selon les classes, nous pouvons toujours construire une base d'apprentissage de 70% et une base de validation de 30% du total. De cette manière, le représentatif de la base de validation est mieux assuré. C'est pourquoi la validation croisée Monte Carlo est effectuée au lieu de celle de k fold.

### 6.2 Résultat empirique

Les validations sont mise en place sur deux modèles modifiés : régression logistique pénalisé s'appuyant sur les variables les plus importantes et modèle hybride s'appuyant sur les variables les plus importantes. Les variables les plus importantes sont choisies selon la critère de "filter" de type lasso et "filter" de moyenne. Les performances de la base de test est plus variant que celle de la base d'apprentissage. Les petites écarts de AUC moyen sont observée sur tous les deux modèles, ce qui montre la robustesse des modèles. D'ailleurs, la performance de notre modèle hybride est proche de celle du modèle supervisé. C'est une preuve qui montre les variables synthétiques sont informatives en classification. Les variables modélisées dépendantes à l'air d'être une cure potentielle du fléau de dimension.

"Filter" de type lasso :

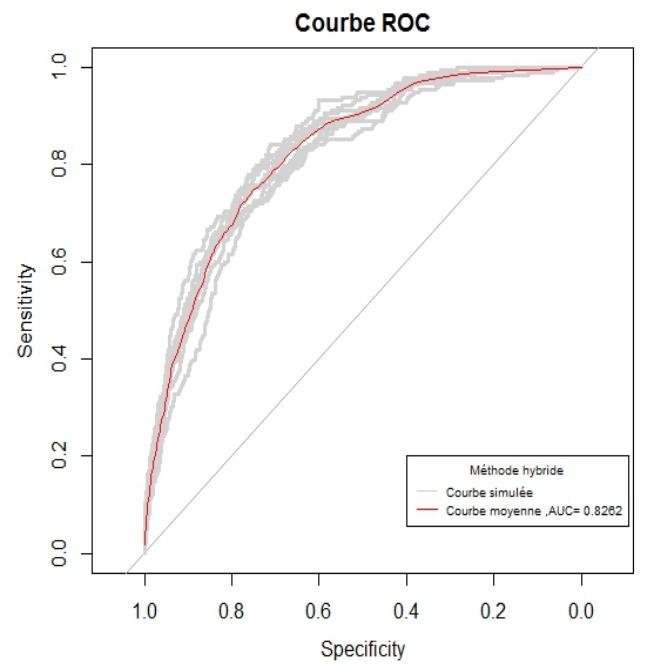
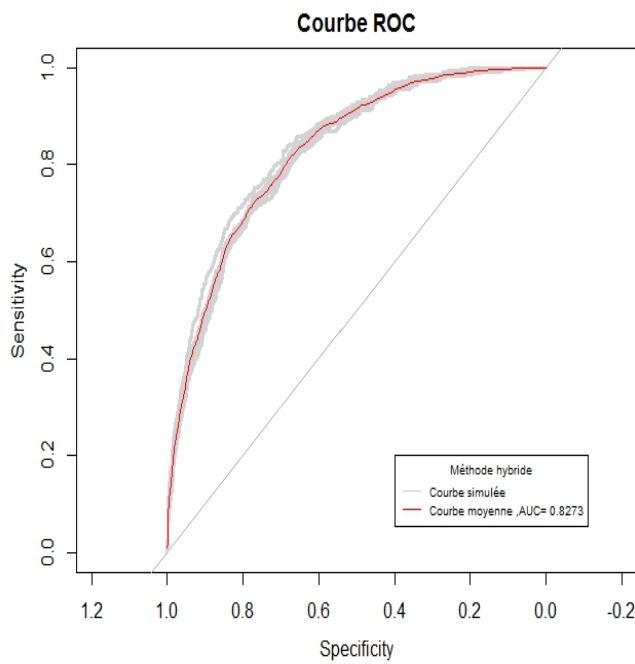
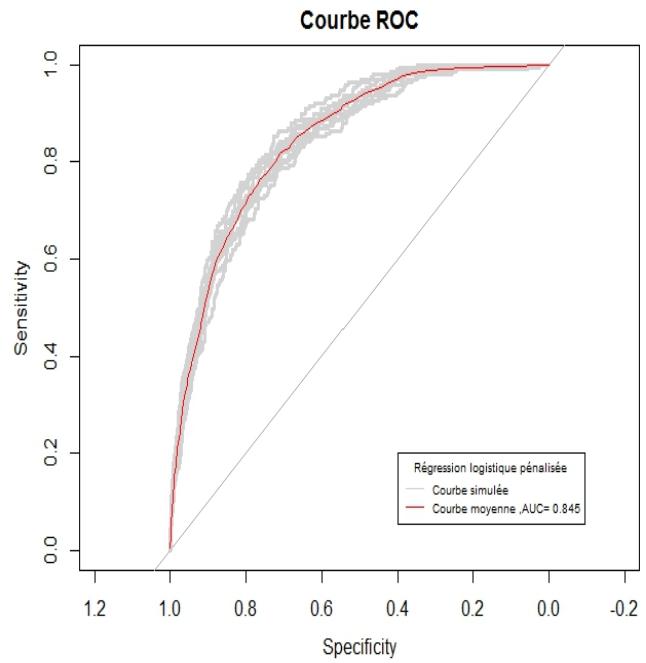
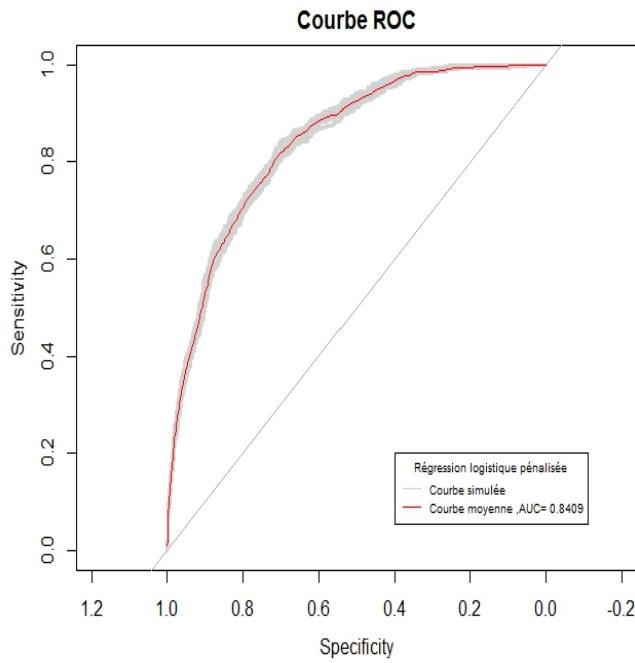


FIGURE 6.1 – Base d'apprentissage

"Filter" de moyenne :

FIGURE 6.2 – Base de test

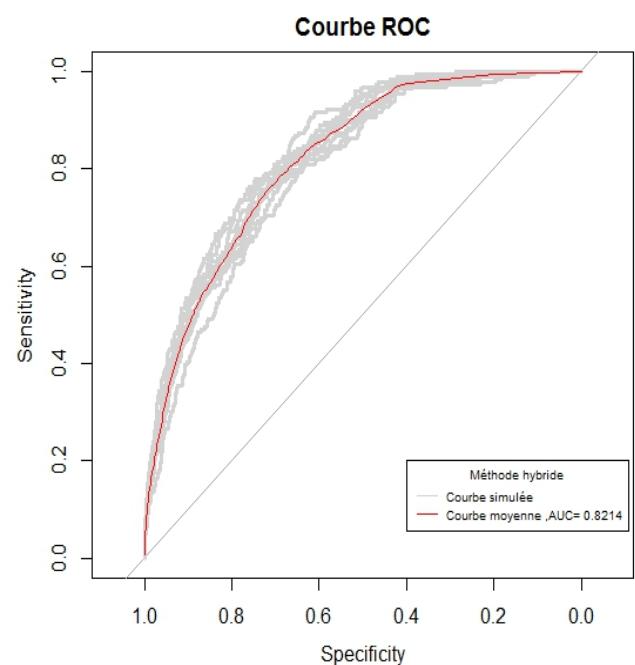
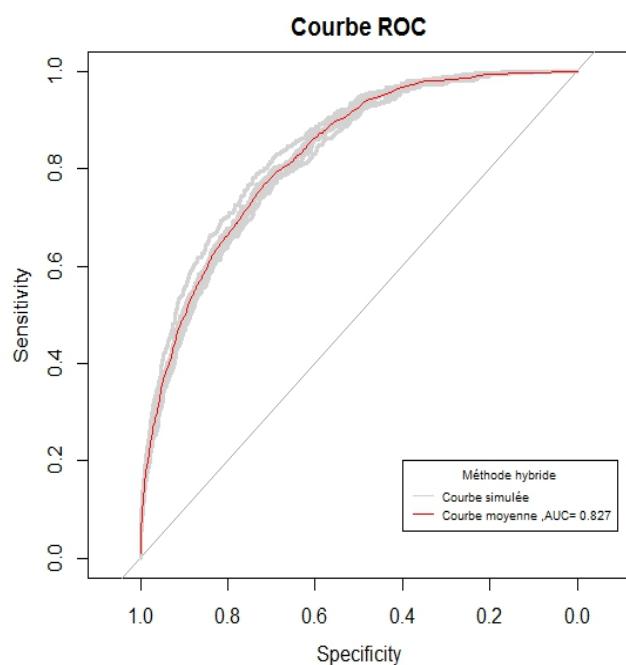
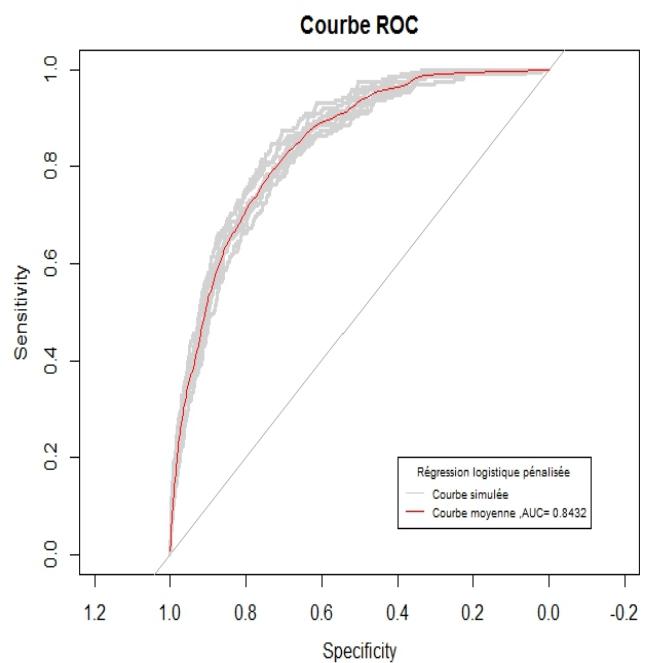
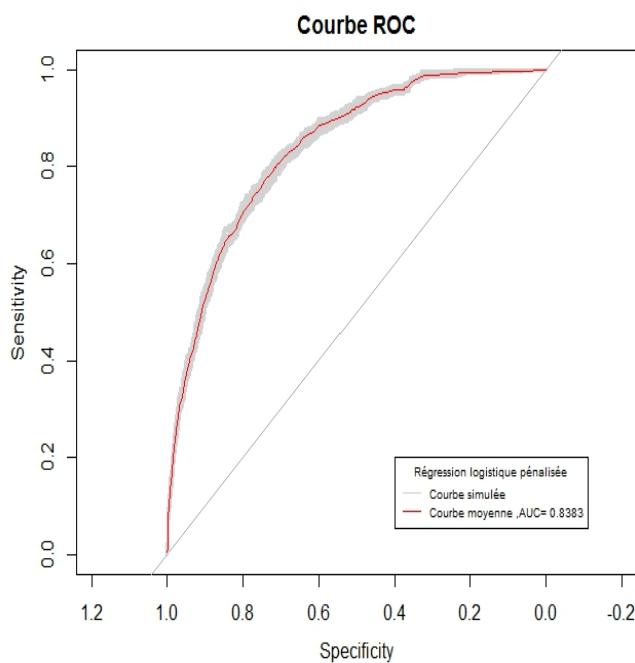


FIGURE 6.3 – Base d'apprentissage

FIGURE 6.4 – Base de test

# Chapitre 7

## Conclusion et perspectives

Durant l'étude du chapitre 3, on a vu que la classification s'appuyant sur l'apprentissage de la structure topologique dépend de la séparation naturelle. Au cas où la situation est plus complexe, il nous faut l'information supplémentaire de la classe et la reconstruction de base d'apprentissage par sous-échantillonnage. Deux essais sont effectués : soit utiliser l'information de classe comme nouvelle entrée combinée avec le sous-échantillonnage pour entraîner une carte Kohonen supervisé, soit prendre l'information de classe comme sortie et employer les modèles d'apprentissage supervisé pour étudier le lien entre les entrées et les sorties. De plus, une sélection des variables est effectuée pour trouver un sous espace des entrées le plus discriminant.

Basée sur les deux essais, une méthode d'apprentissage hybride est proposé pour profiter de l'avantage de deux essais. La carte Kohonen joue le rôle de sous-échantillonnage d'une manière plus robuste. Les classificateurs locaux sont mises en place pour construire les variables synthétiques. La performance de cette méthode est satisfaisant comparée au modèle de référence. Selon le résultat de validation croisée, elle est compétitive comme une méthode des variables modèles dépendantes face aux modèles d'apprentissage supervisés des variables entrée dépendantes.

Il existe encore des points d'attention. D'abord, du côté de la qualité de donnée, il nous faut bien traiter le fusionnement à partir des bases de données de cohortes. La base d'apprentissage se construit des bases de cohortes différentes pour en faire une échantillonnage stratifié. Cependant, un simple fusionnement des bases risque de sous-estimer le taux des clients défaillants. Les entrées des clients tiré des cohortes avant de la dernière sont préférable d'être renouvelées dans le fusionnement des bases de données.

Ensuite, une pondération des entrées peut être implémentée en ajustant la fonction de perte selon les cohortes. En ce cas, nous prendrons compte de l'effet du temps.

Enfin, les modèles plus avancés et les nouvelles méthodes de modifier l'espace des entrées peuvent être proposés pour améliorer la prévision et approfondir notre compréhension des données.

# Bibliographie

- [1] IBBOU Smail *Classification, analyse des correspondances et methodes neuronales*, . Thèse, Université Paris 1, 1992.
- [2] Melssen, Willem, Ron Wehrens, and Lutgarde Buydens. *Supervised Kohonen networks for classification problems*. Chemometrics and Intelligent Laboratory Systems 83.2 (2006) : 99-113.
- [3] Chapelle, Olivier, Bernhard Scholkopf, and Alexander Zien. *Semi-Supervised Learning*. MIT Press, 2006
- [4] Maaten, Laurens van der, and Geoffrey Hinton. *Visualizing data using t-SNE*. Journal of Machine Learning Research 9.Nov (2008) : 2579-2605.
- [5] Chawla, Nitesh V. *Data mining for imbalanced datasets : An overview*. Data mining and knowledge discovery handbook. Springer US, 2009. 875-886.
- [6] Melssen, W., Wehrens, R., Buydens, L. *Supervised Kohonen networks for classification problems*. Chemometrics and Intelligent Laboratory Systems, 83(2), 99-113 (2006).
- [7] Breiman L., Friedman J. H., Olshen R. A., and Stone, C. J. *Classification and Regression Trees*. (1984) Wadsworth.
- [8] Leo Breiman *Random Forests*. Mach. Learn. 45, 1 (October 2001), 5-32.
- [9] Jerome Friedman, Trevor Hastie, Robert Tibshirani *Regularization Paths for Generalized Linear Models via Coordinate Descent*. Journal of Statistical Software, 33(1), 1-22. (2010).
- [10] Schütze, H. *Introduction to information retrieval* . Proceedings of the international communication of association for computing machinery conference.(2008, June).
- [11] Louppe, G., Wehenkel, L., Sutera, A., Geurts, P. *Understanding variable importances in forests of randomized trees* . Advances in neural information processing systems (pp. 431-439), (2013).
- [12] Kuhn, M., Johnson, K. . *Applied predictive modeling* New York, NY : Springer (2013)
- [13] Richard G. Baraniuk *Compressive Sensing*. IEEE Signal Processing Magazine
- [14] Xu, Z., Huang, G., Weinberger, K. Q., Zheng, A. X. *Gradient boosted feature selection*. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 522-531). ACM. (2014, August).
- [15] Meinshausen, N., Bühlmann, P. *Stability selection*. Journal of the Royal Statistical Society : Series B (Statistical Methodology), 72(4), 417-473.(2010).
- [16] Boczkó, E. M., Xie, M., Wu, D., and Young, T. *Comparison of binary classification based on signed distance functions with support vector machines*. OCCBIO'09. Ohio Collaborative Conference on (pp. 139-143). IEEE.2009

- [17] Yuan-chin Ivan Chang *Boosting SVM classifiers with logistic regression.*  
[www.stat.sinica.edu.tw/library/c tec rep/2003 - 03.pdf](http://www.stat.sinica.edu.tw/library/c tec rep/2003 - 03.pdf), 2003
- [18] Dubitzky,, Werner ; Granzow, Martin ; Berrar, Daniel *Fundamentals of data mining in genomics and proteomics*. Springer Science & Business Media. p. 178. (2007)
- [19] W Chen, Y Du, F Zhang, R Zhang, B Ding *Sampling error profile analysis (SEPA) for model optimization and model evaluation in multivariate calibration*. Journal of Chemometrics(2017)