Pôle Validation Département Pilotage Consolidé et Modèles
Direction Risques Groupe

Prévision du défaut sur le périmètre de TPE

Présentation

XU Kai

2 novembre 2017

Table des matières

Introduction
Sujet

TPE : un sous-ensemble dont la chiffre d'affaires compris entre 1,5 et 5M€ ou montant total des engagements accordés plus que 1 M€.

L'approche traditionnelle consiste à différencier au mieux les clients défaillants et les clients sains à partir de l'analyse financière. Cependant, elle est moins efficace face à une grande population comme TPE dont les défauts sont rares annuellement.

Par contre, nous essayons de les différencier du point de vue de l'apprentissage statistique.

- *Objective 1 : prévoir des clients défaillants de TPE*
  Des clients défaillants n'occupent que une petite proportion et ses caractéristiques communs sont difficile à capter au sens général. Pour améliorer l'allocation des crédits, une meilleure identification des risques est à besoins.

Introduction
Sujet

- *Objective 1 : prévoir des clients défaillants de TPE*
  Des clients défaillants n'occupent que une petite proportion et ses caractéristiques communs sont difficile à capter au sens général. Pour améliorer l'allocation des crédits, une meilleure identification des risques est à besoins.
- *Objective 2 : construire un cadre efficace pour la prévision*
  Basé sur la connaissance obtenue au cours de l'apprentissage des données, nous essayons et modifions des modèles existants à notre besoins. Un nouveau cadre sera proposée à partir de nos travaux en vue de la prévision performante et robuste.

Sujet : Construire un cadre prédictif des défauts de TPE à partir des modèles existants et proposer un nouveau modèle raffiné profitant des avantages des méthodes différentes.

Introduction
Variables disponibles

La base totale est un échantillon occupant environ 1/6 des sociétés qualifiés dans 4 cohortes de Juin 2012 à Décembre 2013 à pas semestriel. Elle compte 18902 contrats de TPE :

- Variable d'intérêt : indicatrice de défaut. Au total, environ 3% de clients tombent en défaut chaque année.

Introduction
Variables disponibles

La base totale est un échantillon occupant environ 1/6 des sociétés qualifiés dans 4 cohortes de Juin 2012 à Décembre 2013 à pas semestriel. Elle compte 18902 contrats de TPE :

- Variable d'intérêt : indicatrice de défaut. Au total, environ 3% de clients tombent en défaut chaque année.
- Variables explicatives : 202 variables parmi lesquelles 190 variables quantitatives et 12 variables qualitatives de 4 catégories :
  - compte
  - données externes
  - ratios financiers
  - signalétique.
- Les variables qualitatives se transforment en nouvelles variables binaires, ce qui produit environ 56 nouvelle variables.

En gros, 246 variables sont utilisée dans la prévision.

Données quantitatives :

Pour éviter les problèmes d'échelle et l'influence d'une hétérogénéité des variances, les variables quantitatives sont centrées et réduites.

La transformation du z-score :

$$\frac{V - \mu}{\sigma}$$

Soient μ l'espérance et σ l'écart-type des valeurs d'une variable explicative V

Données qualitatives :

Une variable qualitative à *K* modalités est remplacée par *K* variables binaires, et chacune correspondant à une des modalités.

Exemple : un tableau regroupant les informations des chiens

Tableau original :

Individu	Taille	Poids
bass	Petite	Léger
beau	Grande	Moyen
boxe	Moyenne	Moyen

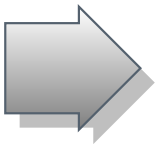
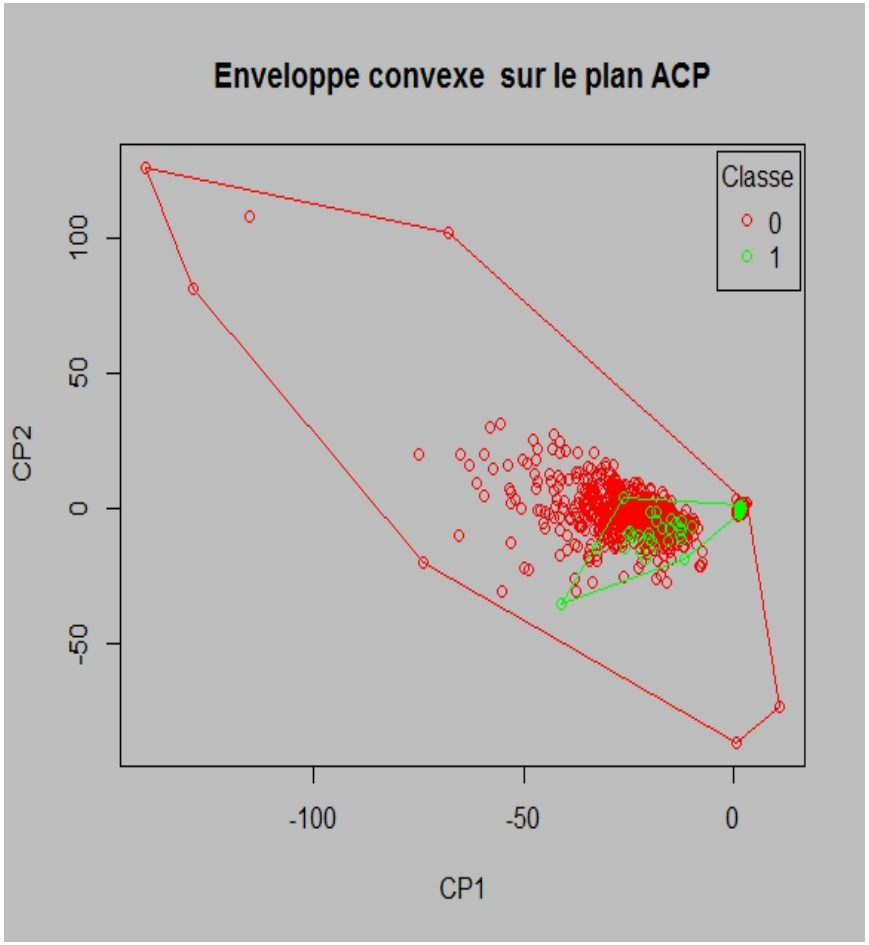


Tableau disjonctif complet :

Individu	Taille P	Taille M	Taille G	Poids L	Poids M
bass	1	0	0	1	0
beau	0	0	1	0	1
boxe	0	1	0	0	1

Enveloppe convexe :



- Enveloppe convexe d'une classe est l'ensemble convexe le plus petit parmi ceux qui la contiennent
- Examiner la séparation linéaire
- Nous la visualiser sur le plan ACP
- Classe 0 : Clients sains
  Classe 1 : Clients défaillants

t-SNE (t-distributed stochastic neighbor embedding)

- trouver une projection optimale des points $\mathbf{x}_1, \ldots, \mathbf{x}_N$ à grande dimension dans un espace de 2 dimension
- La fonction à optimiser :

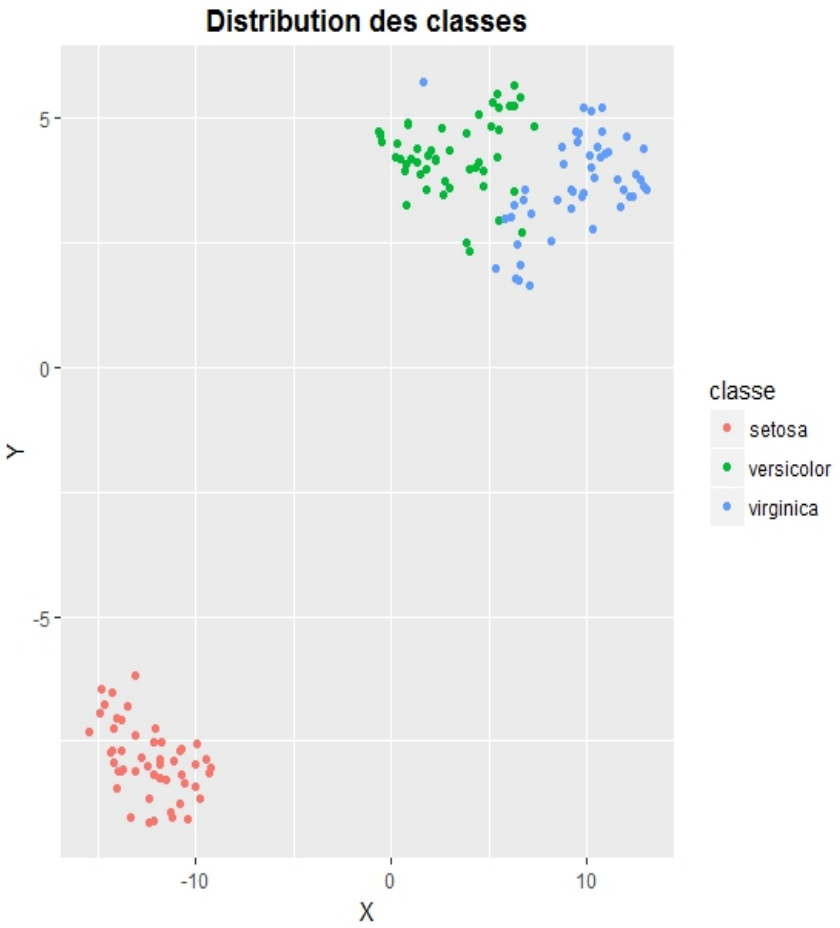
$$\inf_{y_1,\ldots,y_N} KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Les points original $\mathbf{x}_1, \ldots, \mathbf{x}_N$ et ses projections $\mathbf{y}_1, \ldots, \mathbf{y}_N$:

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

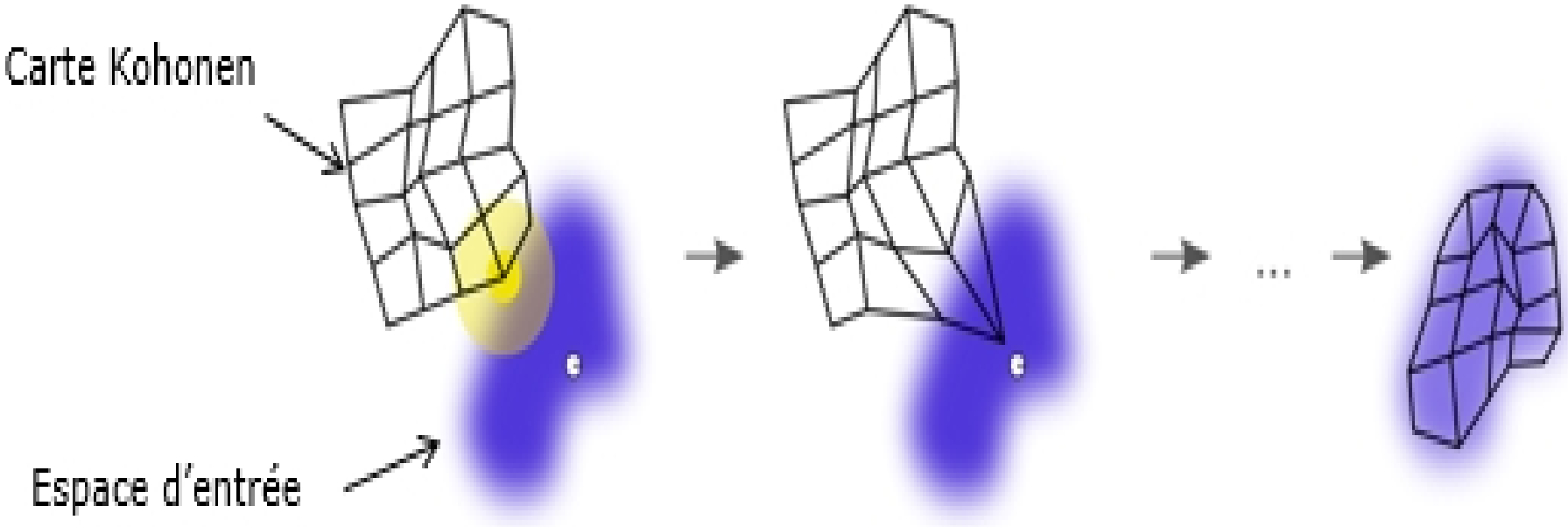
Apprentissage semi-supervisé
Structure topologique

Base de Iris :



Base de TPE :

Apprentissage de la carte Kohonen :



L'apprentissage se déroule d'une manière de la compétition en utilisant des vecteurs poids $\vec{w}$ appartenant aux neurones.

Initialisation de la carte Kohonen

- Initialisation aléatoire des vecteurs poids $\vec{w}$ selon la loi uniforme sur l'intervalle [0,1] de haute dimension

L'apprentissage se déroule d'une manière de la compétition en utilisant des vecteurs poids $\vec{w}$ appartenant aux neurones.

Initialisation de la carte Kohonen

- Initialisation aléatoire des vecteurs poids $\vec{w}$ selon la loi uniforme sur l'intervalle [0,1] de haute dimension

Compétiontion des neurones

- A instant t, une observation $x(t+1) \subset R^d$ est choisie aléatoirement et présentée au réseau
- Le neurone gagnant est donc déterminé par :

$$i_0 = \underset{i}{\text{arginf}} \|x(t+1) - w_i(t)\|^2$$

Compétiontion des neurones

- A instant t, une observation $x(t+1) \subset R^d$ est choisie aléatoirement et présentée au réseau
- Le neurone gagnant est donc déterminé par :

$$i_0 = \underset{i}{\text{arginf}} \|x(t+1) - w_i(t)\|^2$$

Évolution des neurones

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

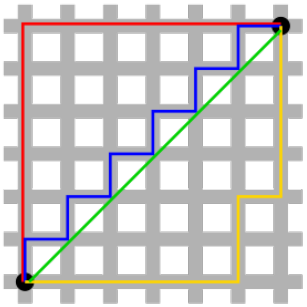
$$\begin{cases} w_i^{t+1} = w_i^t + \epsilon(t)(x(t+1) - w_i^t) & \forall i \in V_{r_t}(i_0) \\ w_i^{t+1} = w_i^t & \forall i \notin V_{r_t}(i_0) \end{cases}$$

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

On explique ici les notions concernant l'apprentissage de la carte Kohonen :

- Distance de Manhattan : Entre deux points A et B, de coordonnées respectives (X_A, Y_A) et (X_B, Y_B).

$$d(A, B) = |X_B - X_A| + |Y_B - Y_A|$$

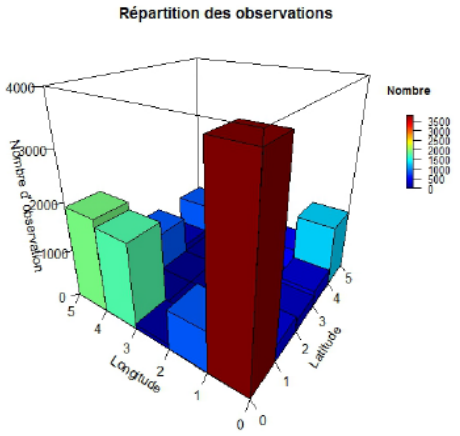


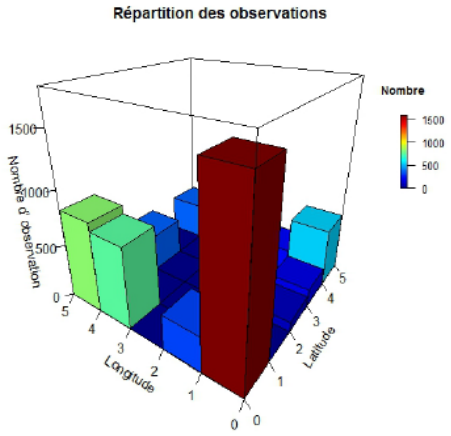
Distance de Manhattan : chemins rouge, jaune et bleu ; ils sont équivalents au sens de la distance de Manhattan
Distance euclidienne : chemin vert

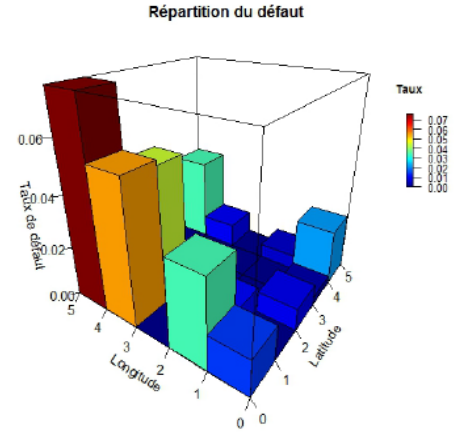
- Taux d'apprentissage $\epsilon(t)$: la vitesse à laquelle les vecteurs poids $\vec{w}$ sont ajustées
- Rayon $r(t)$: la distance de Manhattan de neurone gagnant à ses voisins les plus éloignés.

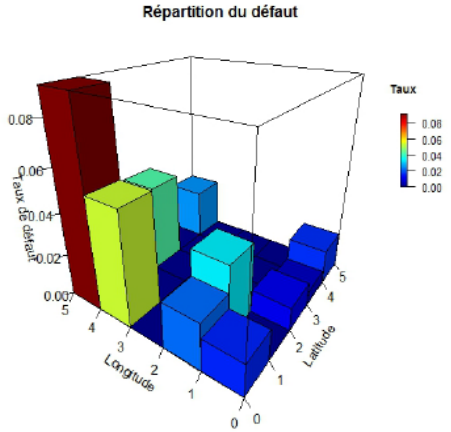
Apprentissage semi-supervisé
Carte Kohonen

Carte Kohonen :



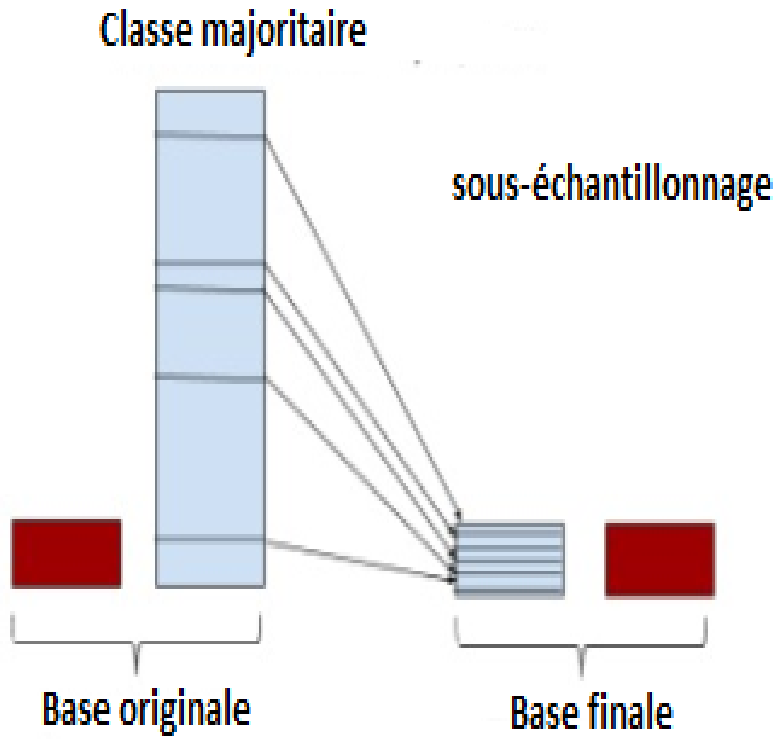






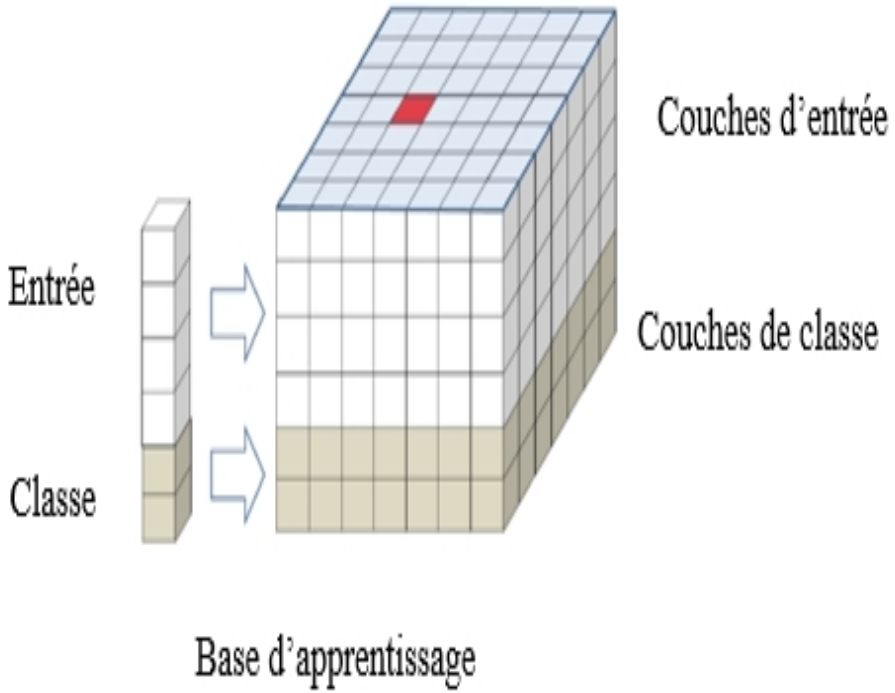
- Le neurone avec des observations le plus nombreuses et ses voisins possèdent les taux de défaut plus bas
- La différence n'est pas évident pour que un neurone soit dominé par les individus défaillants

Sous-échantillonnage :



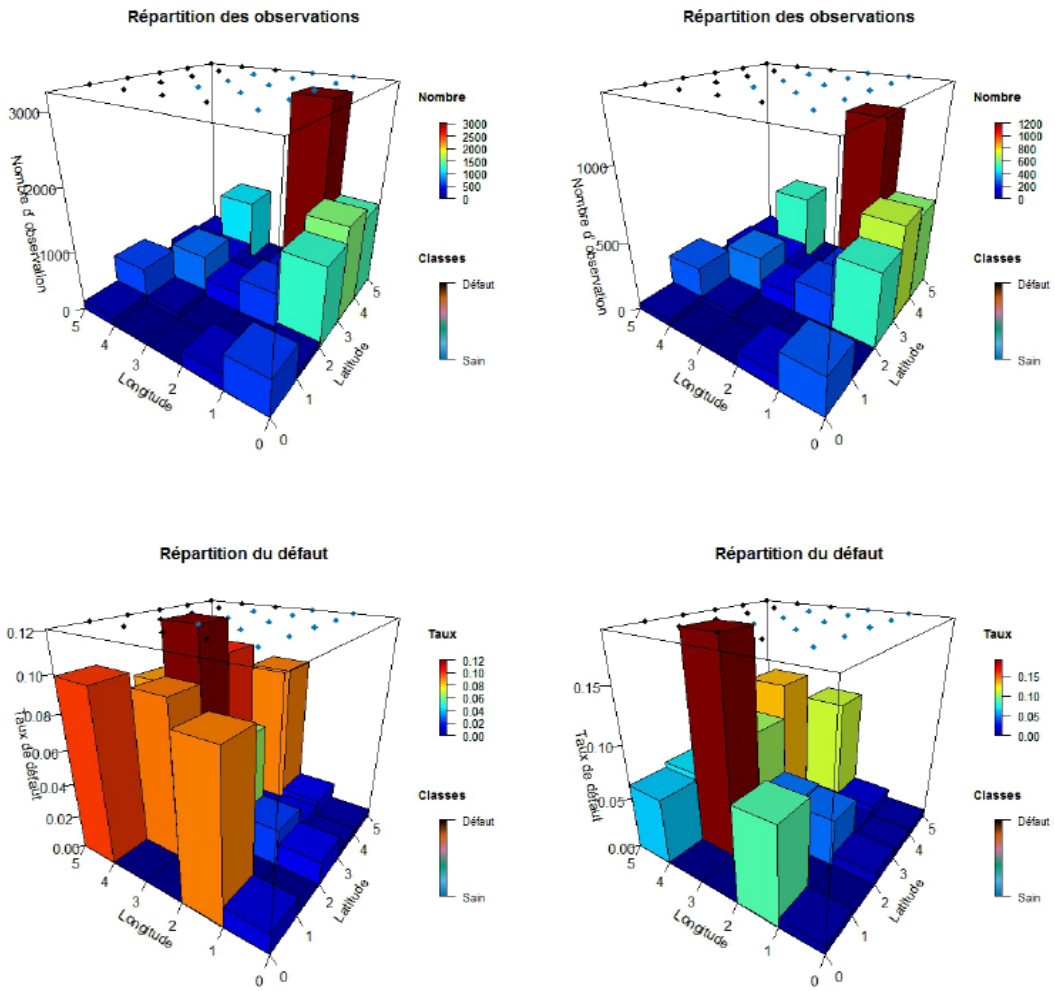
- Contre le déséquilibre des classes
- Diminuer des observations de classe majoritaire
- Réalisé par tirage aléatoire selon une loi uniforme

Carte Kohonen supervisée :



- Profiter à la fois des entrées et des classes à l'étape d'apprentissage
- Possible de faire une pondération entre les couches de deux types
- La classification ne dépend que de l'entrée à l'étape de prévision

Carte Kohonen supervisée :

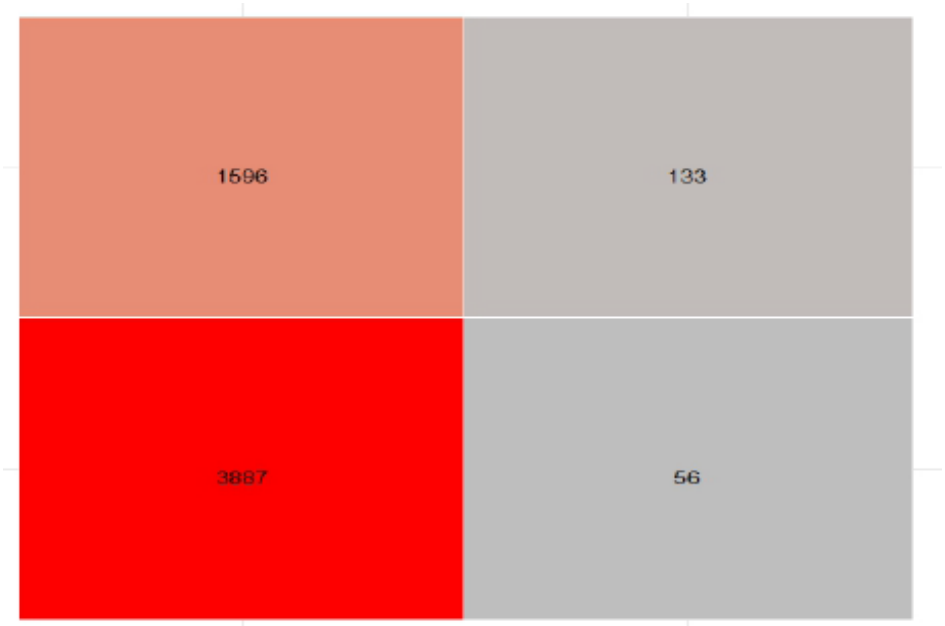


- La carte supervisée peut identifier les neurones dominés par les classes différentes dans la base sous-échantillonnée
- En remettant les observations exclues de l'échantillon, la performance reste meilleur que celle de la carte non supervisée

Apprentissage semi-supervisé
Carte Kohonen supervisée

Matrice de confusion :

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



- Dans la base de test, sur les 5480 clients sains, 3887 seront estimés comme sains, soit 71% des clients sains sont correctement prédits
- Sur les 189 clients défaillants, 133 seront estimés comme défaillants, soit 70.4% des clients défaillants sont correctement prédits.

Apprentissage supervisé

L'apprentissage supervisé consiste à reproduire des classes y tandis que l'étude précédente se concentre sur la structure topologique.

But de l'apprentissage supervisé

Estimer le lien entre des entrées x et des classes y grâce à la fonction de prédiction (le classifieur) $f(\cdot)$ apprise des données connus.
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 $l(f(x), y)$.

Arbre de décision

La fonction de perte de l'arbre de décision est

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

Génération de l'arbre

Le découpage à chaque nœud intérieur choisit une variable d'entrée qui réalise la réduction la plus grande de l'entropie de Shannon pondérée :

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

Régression logistique pénalisée

Le but est de minimiser le risque moyen

$$\inf_{\beta_0,\beta} E[l(y_i, \beta_0 + \beta^T x_i)]$$

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

- Par rapport à la fonction de perte 0-1, la fonction de perte logistique est différentiable

Pour effectuer une régularisation, une terme de pénalisation de type Lasso est ajoutée dans l'espérance :

$$\inf_{\beta_0,\beta} \quad E[l(y_i, \beta_0 + \beta^T x_i)] + \lambda||\beta||_1$$

Forêts aléatoires

Des forêts aléatoires se construisent en appliquant une multitude d'arbres de décision aux sous ensembles des variables explicatives dans un cadre de l'agrégation de modèles.

Le but est donc de minimiser le risque moyen de l'agrégation des arbres concernés

$$E[I(f(x), y)] = E[\sum_{n=1}^{N} I_n(f(x), y)]$$

où

$$I_n(f(x), y) = \begin{cases} 1 & f_n(x) = y \\ 0 & f_n(x) \neq y \end{cases}$$

Gradient Boosting

Le Boosting effectuera un vote majoritaire pondéré :

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

Il s'agit d'une série d'optimisation sur w et f :

$$F_0(x) = \arg\inf_w \sum_{n=1}^{N} l(y_n, w),$$

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

Gradient Boosting

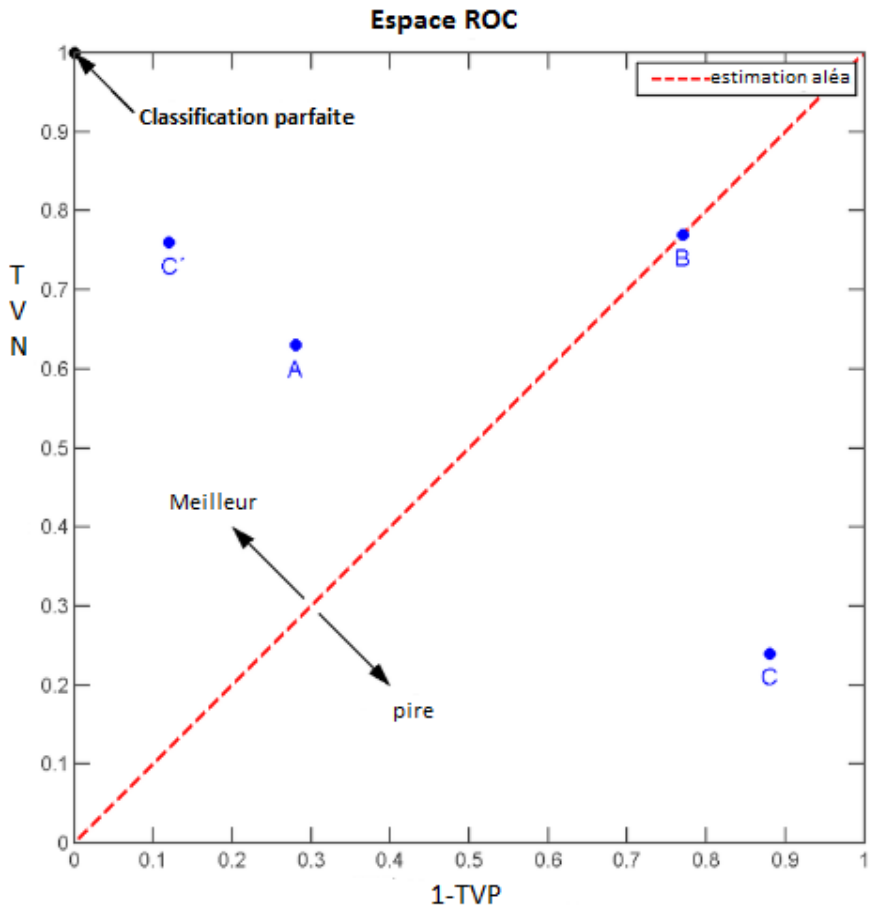
Au delà du boosting général, le boosting dont la fonction de perte est différentiable s'appelle gradient boosting. Elle profite de l'existence du gradient de la fonction de perte.

$$F_m(x) = F_{m-1}(x) - w_m \sum_{n=1}^{N} \nabla_{F_{m-1}} l(y_n, F_{m-1}(x_n)),$$
$$w_m = \arg\min_{w} \sum_{n=1}^{N} L\left(y_n, F_{m-1}(x_n) - w \nabla_{F_{m-1}} l(y_n, F_{m-1}(x_n))\right),$$

Ici, nous utilisons la fonction de perte logistique mentionnée dans la régression logistique pénalisée. En ce cas, le gradient $\nabla_{F_{m-1}}$ se définit par :

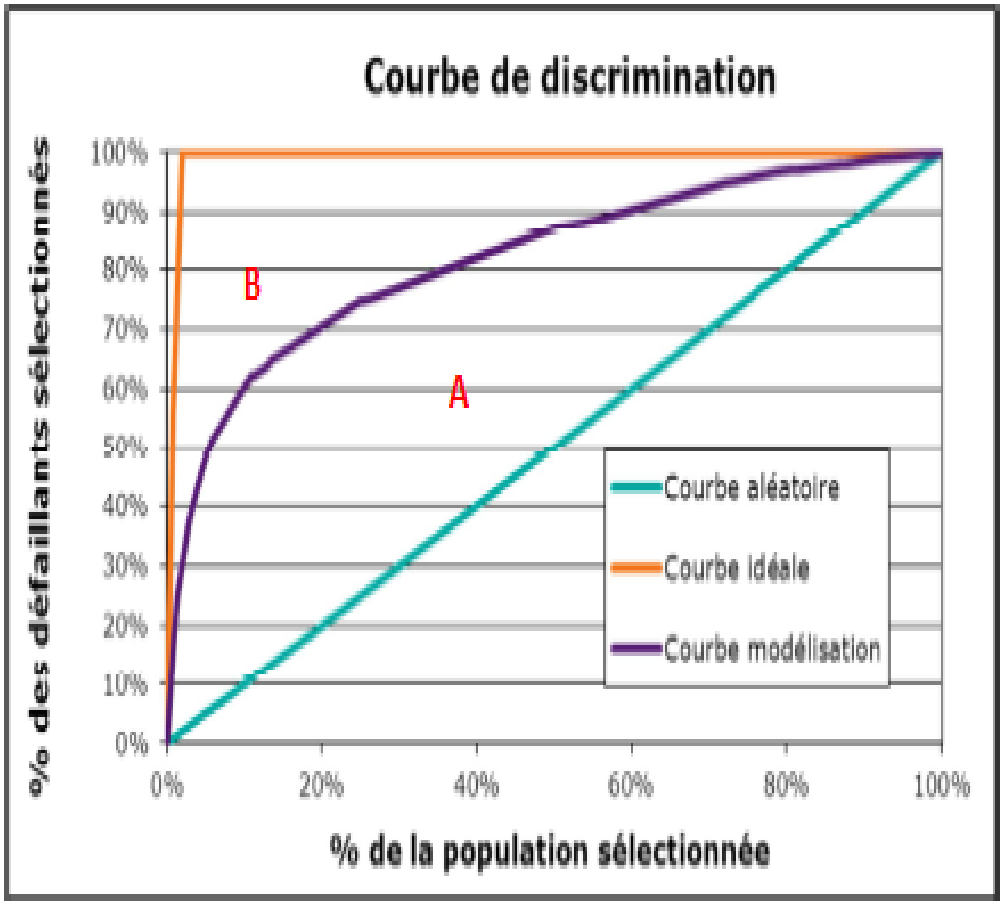
$$\nabla_{F_{m-1}} l(y_n, F_{m-1}(x_n)) = \frac{\partial l(y_n, F_{m-1}(x_n))}{\partial F_{m-1}(x_n)} = \frac{y - (1-y)e^{F_{m-1}(x_n)}}{1 + e^{F_{m-1}(x_n)}}$$

Courbe ROC :



- TVP = Sensibilité (sensibility) = VP/Positifs
- TVN = Spécificité (specificity) = VN/Négatifs
- en abscisse le TVN, en ordonnée le TVP ou 1-TVP

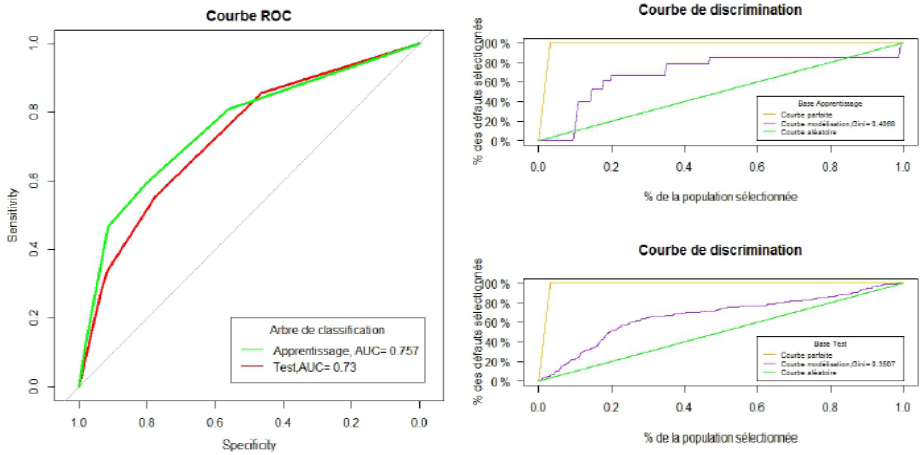
Indice de Gini :



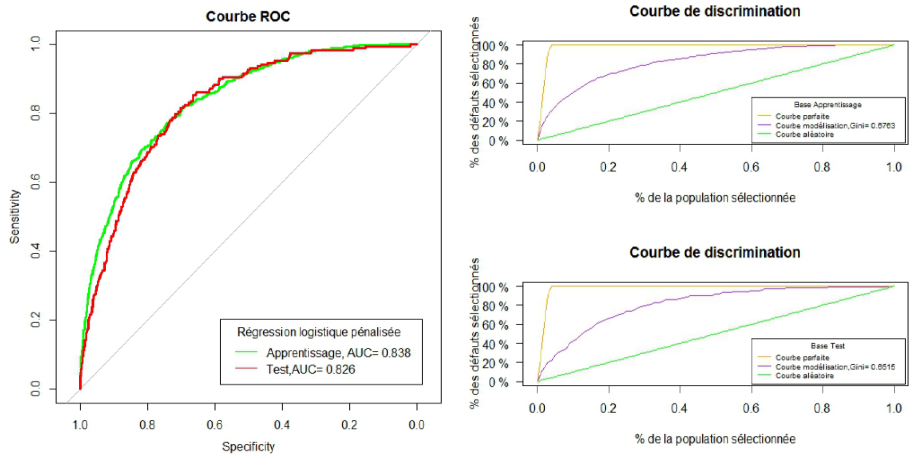
- A : aire entre la courbe du modèle construite et la droite de l'aléatoire
- B : aire entre la courbe du modèle idéale et la droite de l'aléatoire
- $0 \leq Gini = \frac{A}{B} \leq 1$

Apprentissage supervisé
La comparaison des résultats

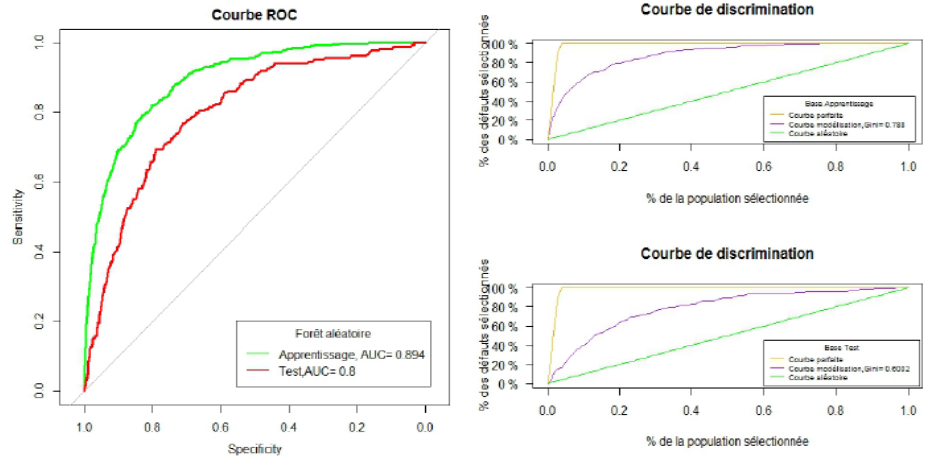
Arbre de décision :



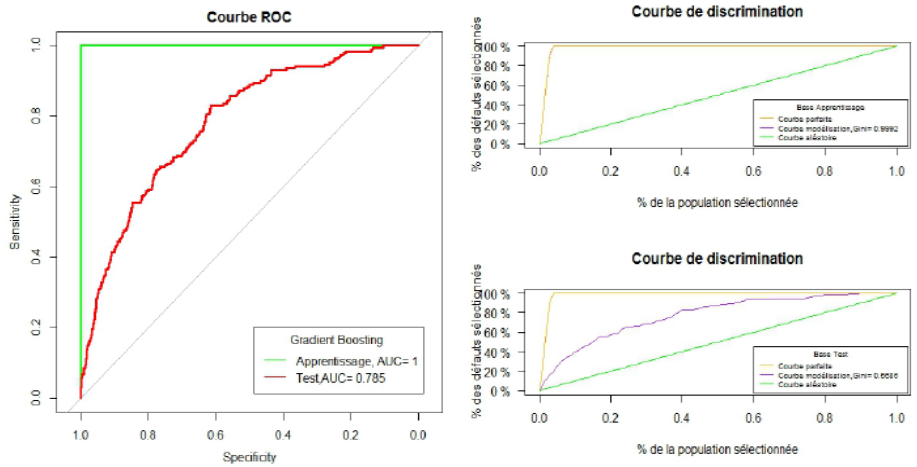
Régression logistique pénalisée :



Forêts aléatoires :



Gradient boosting :



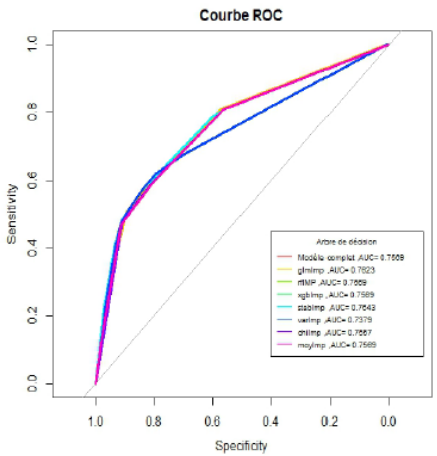
Sélection de variables I

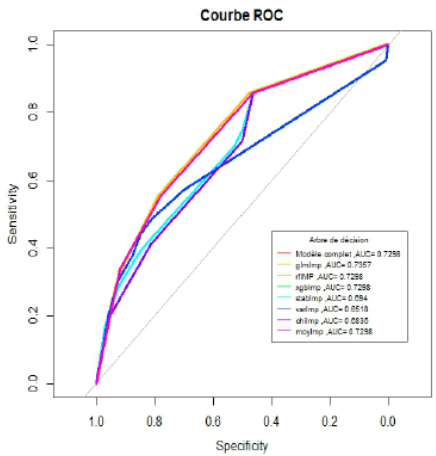
- Filter de variance :
  Éliminer des variables de basse variance
- Critère de χ^2 :
  Le test de χ^2 sert à déterminer l'existence d'une relation entre deux variables catégorielles.
- Sélection de type lasso :
  Éliminer des variables par la régression logistique pénalisée
- Sélection de type forêts aléatoires :
  Éliminer des variables par l'importance des variables obtenue des forêts aléatoires

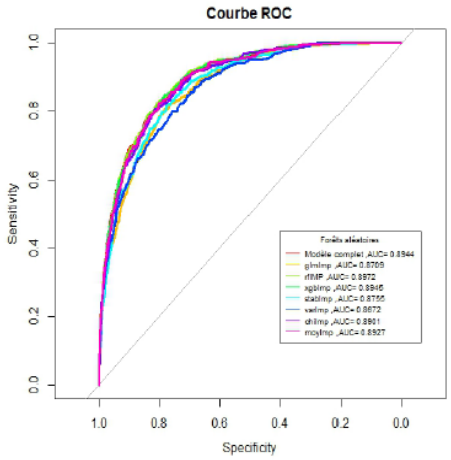
Sélection de variables II

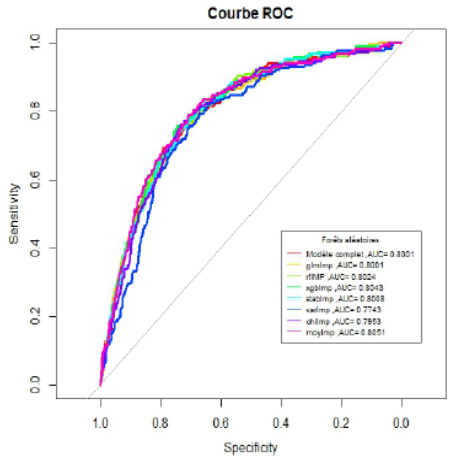
- Sélection de type gradient boosting :
  Éliminer des variables par l'importance des variables obtenue du gradient boosting
- Sélection stabilisante :
  Appliquer le modèle de base aux sous-ensembles des observations et variables,ce qui est différent que les forêts aléatoire effectué sur seulement des sous-ensembles des variables.
- Sélection moyenne :
  Comparer les moyennes des notes précédentes qui évaluent l'importance des variables.

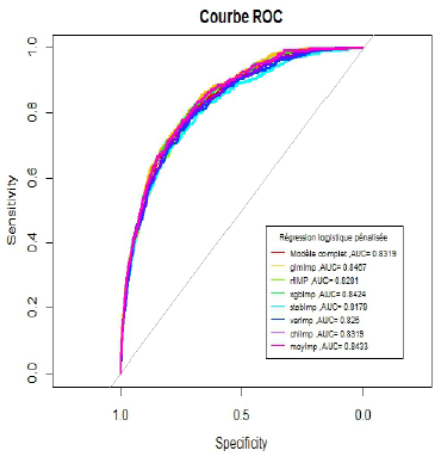
Apprentissage supervisé
Sélection de variables

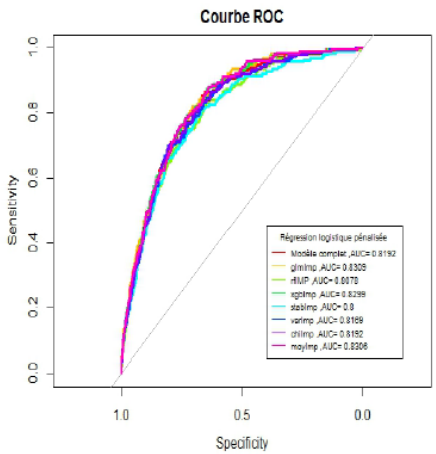


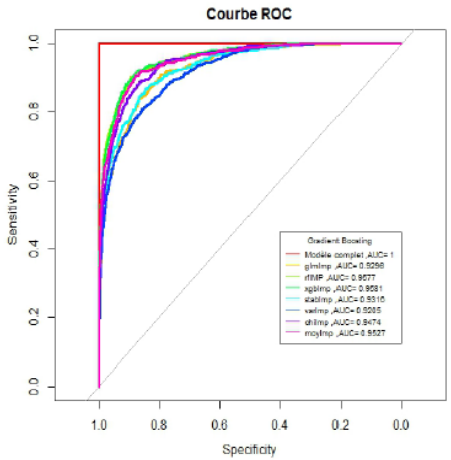


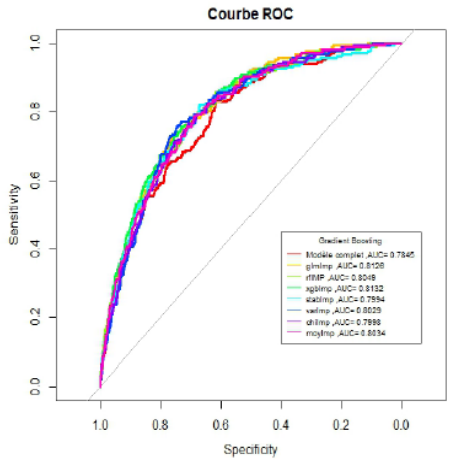




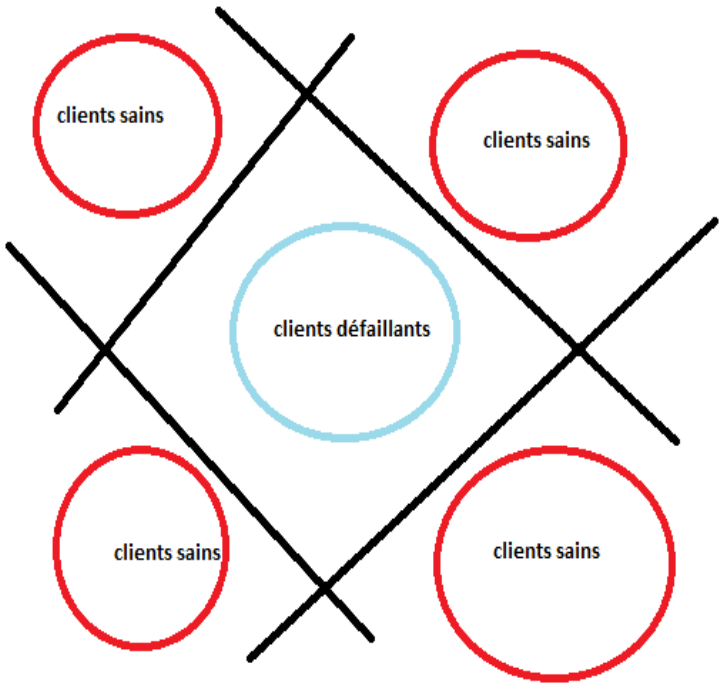








Hétérogénéité :



- Les propriétés d'une partie des données sont différents que les autres.
- Fréquent dans la population dont les classes sont déséquilibrées

Regroupement des clients sains

- Entraîner une carte Kohonen classique sur tous les clients sains de la base d'apprentissage.
- Regrouper des clients sains selon les neurones pour former des sous-populations similaires à la structure topologique

Classifieur local

- Remettre des clients défaillant dans toutes les sous-populations générées dans la procédure précédente
- Construire des classifieurs locaux sur les sous-populations
- Le classifieur local choisi est un SVM (Machine à vecteur de support) muni de noyau de RBF : $h(x) = w^T \phi(x) + w_0$
  où $< \varphi(x), \varphi(x') >= K(x, x') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$

Distance signée

Étant donné un hyperplan de séparation S, la fonction de la distance signée $f(x)$ est définie par les ensembles A et B qui sont caractérisés selon S

- $$f(x) = \begin{cases} d(x,S) & \text{si } x \in A \\ 0 & \text{si } x \in S \\ -d(x,S) & \text{si } x \in B \end{cases}$$
- Calculer les distances signées en utilisant des classifieurs locaux sur la base d' apprentissage
- Les distances signées servirent de variables synthétiques propres au modèle

Prévision de défaut

Effectuer une régression logistique pénalisée pour produire la probabilité d'être défaillant

Partie d'apprentissage non-supervisé

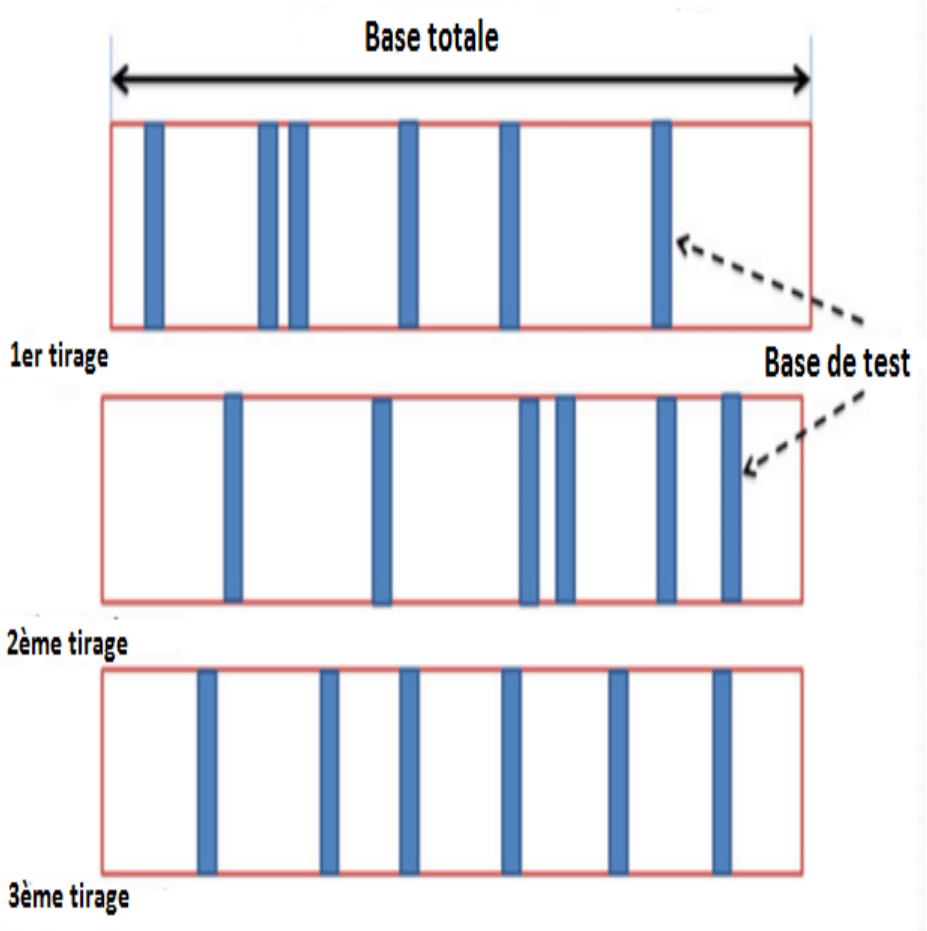
- Regroupement des clients défaillants en utilisant la carte Kohonen

Partie d'apprentissage supervisé

- Les classifieurs locaux s'interviennent dans la génération des variables synthétiques x
- Des classes y prédites sont obtenues par le biais d'estimation du lien entre des variables synthétiques x et des classes y

Validation croisée
Évaluation de la robustesse

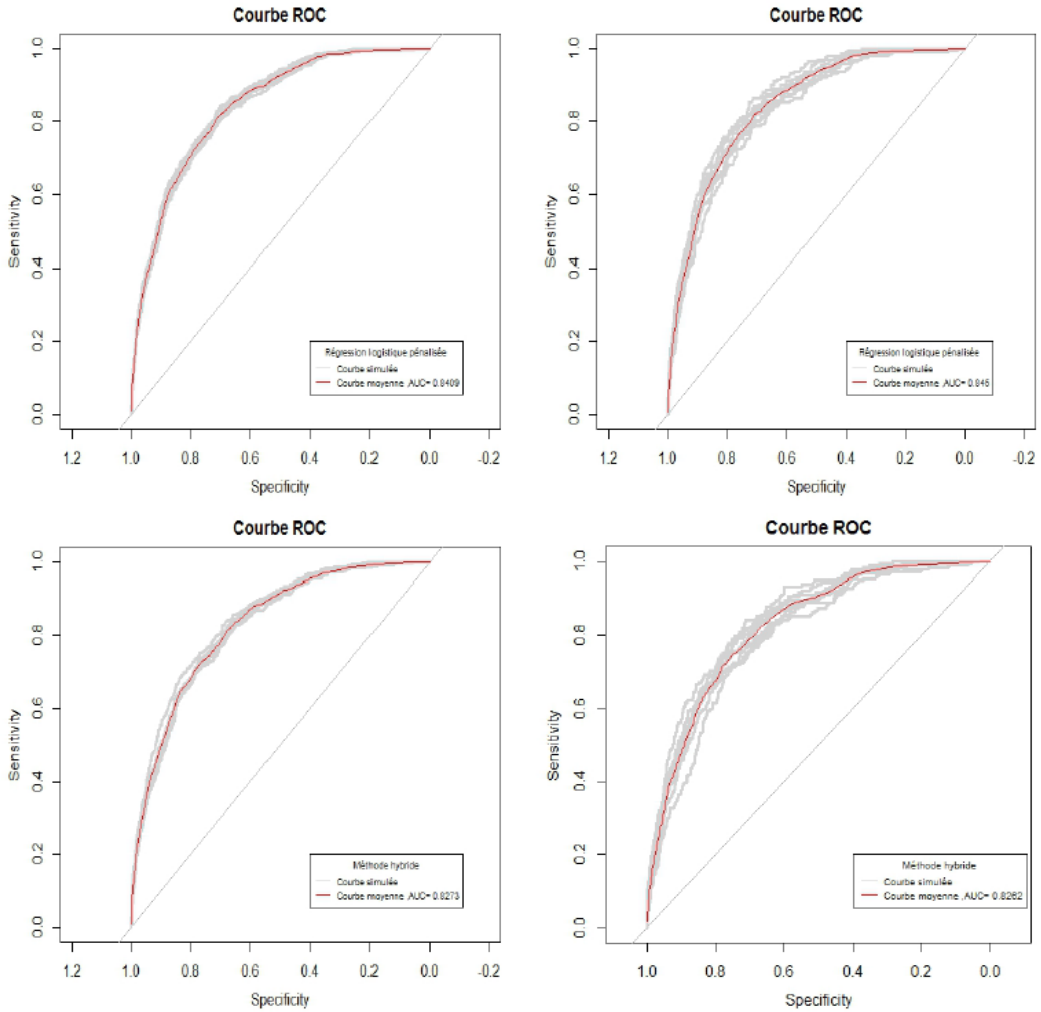
Validation croisée Monte Carlo stratifiée :



- Tirage aléatoire sans remise pour la construction de la base de test
- Sélectionner un échantillon au sein de chaque classe pour éviter le déséquilibre
- Pas de souci de choisir une partition raisonnable comme validation croisée k-fold

Validation croisée
Résultat

Performance des modèles :



- La comparaison est entre la régression pénalisée et la méthode hybride sur les variables sélectionnées
- La méthode hybride dont les variables explicatives propres au modèle est aussi performant que le modèle d'apprentissage supervisé dépendant des entrées originaux
- Le fléau de dimension est possible à éviter en utilisant les variables synthétiques

Perspective

- Selon Vapnik, la limite supérieure de l'erreur de test sera diminuée sous la taille de la base d'apprentissage N plus grande :

$$\Pr\left(\text{Erreur de test} \leqslant \text{Erreur d'apprentissage} + \sqrt{\frac{1}{N}\left[D\left(\log\left(\frac{2N}{D}\right)+1\right)-\log\left(\frac{\eta}{4}\right)\right]}\right) = 1-\eta$$

- La performance des classifieurs qui ont l'air de surapprentissage est possible d'être améliorée avec la base de donnée plus grande
- 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

Merci pour votre attention !