Université de Montréal

Prédiction de l'attrition en date de renouvellement en assurance automobile avec l'aide de processus gaussiens

par
Sylvain Pannetier Lebeuf

Département de mathématiques et statistiques
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en mathématique

Août, 2011

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé:

Prédiction de l'attrition en date de renouvellement en assurance automobile avec l'aide de processus gaussiens

présenté par:

Sylvain Pannetier Lebeuf

a été évalué par un jury composé des personnes suivantes:

Jean-François Angers,	président-rapporteur
Manuel Morales,	directeur de recherche
Yoshua Bengio,	codirecteur
Louis Doray,	membre du jury

Mémoire accepté le: . . . . . .

RÉSUMÉ

Le domaine de l'assurance automobile fonctionne par cycles présentant des phases de profitabilité et d'autres de non-profitabilité. Dans les phases de non-profitabilité, les compagnies d'assurance ont généralement le réflexe d'augmenter le coût des primes afin de tenter de réduire les pertes. Par contre, de très grandes augmentations peuvent avoir pour effet de massivement faire fuir la clientèle vers les compétiteurs. Un trop haut taux d'attrition pourrait avoir un effet négatif sur la profitabilité à long terme de la compagnie. Une bonne gestion des augmentations de taux se révèle donc primordiale pour une compagnie d'assurance.

Ce mémoire a pour but de construire un outil de simulation de l'allure du portefeuille d'assurance détenu par un assureur en fonction du changement de taux proposé à chacun des assurés. Une procédure utilisant des régressions à l'aide de processus gaussiens univariés est développée. Cette procédure offre une performance supérieure à la régression logistique, le modèle généralement utilisé pour effectuer ce genre de tâche.

Mots clés: forage de données, processus gaussien, attrition, assurance automobile

ABSTRACT

The field of auto insurance is working by cycles with phases of profitability and other of non-profitability. In the phases of non-profitability, insurance companies generally have the reflex to increase the cost of premiums in an attempt to reduce losses. For cons, very large increases may have the effect of massive attrition of the customers. A too high attrition rate could have a negative effect on long-term profitability of the company. Proper management of rate increases thus appears crucial to an insurance company.

This thesis aims to build a simulation tool to predict the content of the insurance portfolio held by an insurer based on the rate change proposed to each insured. A procedure using univariate Gaussian Processes regression is developed. This procedure offers a superior performance than the logistic regression model typically used to perform such tasks.

Keywords: Data Mining, Gaussian Process, churn, automobile insurance

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES

GP	Processus gaussien
GLM	Modèle linéaire généralisé
LOESS	Lissage local de nuage de point
ROC	Receiver operating characteristic

NOTATION

I_n	La matrice identité de dimension n
$N_m(\vec{\mu}, \sigma^2)$	La distribution multi-normale de dimension m avec moyenne $\vec{\mu}$ et variance σ^2.

(Sir Michael Atiyah) Mathematicians are generally thought of as some kind of intellectual machine, a great brain that crunches numbers and spits out theorems. In fact, we are, as Hermann Weyl said, more like creative artists. Although strongly constrained by the rules of logic and by physical experience, we use our imagination to make great leaps into the unknown.

REMERCIEMENTS

Pour commencer, j'aimerais remercier Yoshua Bengio qui a accepté à pieds levés de me diriger suite au départ de Charles Dugas. Malgré son horaire très chargé, il a accepté de m'aider à mener à bien ce projet. Sa vision claire, son jugement ainsi que sa créativité ont rendu très agréable le projet. Je voudrais aussi remercier Clément Brunet de m'avoir accordé une grande marge de manoeuvre dans la réalisation du projet. La confiance qu'il m'a témoignée ainsi que sa créativité contagieuse ont rendu les séances de "philosophie" de fin d'après-midi très agréables. Merci à Charles Dugas pour m'avoir initialement donné ma chance et pour m'avoir fait bénéficier de sa vision d'apprentissage par la pratique. Je remercie aussi Manuel Morales d'avoir accepté de prendre la relève de Charles au niveau administratif, malgré la surcharge de travail qui en a découlé. Finalement, j'aimerais remercier mes parents de m'avoir encouragé dans ce projet. J'aimerais spécialement remercier mon amoureuse d'avoir toujours cru en moi.

Ce mémoire a été réalisé grâce au support de MITACS via le programme de stage en recherche industrielle.

CHAPITRE 1

INTRODUCTION

1.1 Motivation du forage de données

Depuis le début des années 1980, les ordinateurs ont pris une place de plus en plus importante dans la société. Les tâches sont de plus en plus administrées et même rendues possibles grâce à l'apport de l'informatique. Qu'on pense à la téléphonie sans-fil, à l'assurance automobile, aux cartes de crédit ou même à la sécurité nationale, les systèmes informatiques jouent une part essentielle. De par cette omniprésence de l'informatique, de très grandes quantités de données sont emmagasinées. En fait, il est estimé que la quantité de données emmagasinées dans le monde double tous les vingt ans [26].

Malheureusement, toutes ces données disponibles ne sont d'aucune utilité si aucune information n'en est tirée. À ce sujet, le forage de données est une discipline qui consiste à transformer les données en informations pertinentes. En fait, le but premier du forage de données est de découvrir et d'extraire des liens entre des variables à l'intérieur d'un jeu de données afin de donner du sens aux données brutes [23]. La très grande quantité de données à analyser fait en sorte qu'une très grande puissance de calcul est nécessaire pour mener à bien la tâche. Le forage de données est un domaine de recherche de plus en plus populaire depuis vingt ans de par la disponibilité de très grande quantité de données de haute qualité, de la puissance de calcul qui est de plus en plus grande ainsi que par la qualité des informations qui peuvent être découvertes par ce genre de méthode[55].

Sans ordinateur, de grandes limitations quant à la taille du jeu de données ainsi que le nombre de variables prédictives étaient présentes. Lorsque l'ordinateur est apparu, la puissance de calcul disponible a très rapidement augmenté, tout comme la quantité de données disponibles. Il a donc fallu développer de nouvelles techniques afin de pouvoir manipuler de très grandes bases de données dans un temps raisonnable tout en y extrayant des informations pertinentes.

Ainsi, le forage de données est basé sur la statistique en y intégrant des éléments de

technologies de bases de données et d'apprentissage machine [30]. La principale différence entre la statistique et le forage de données est que de ce dernier hérite de l'approche moins conservatrice de l'apprentissage machine [30]. Cette discipline est née vers 1960 dans quelques laboratoires de recherche industrielle. C'est vers le début des années 1980 que ces techniques sont devenues beaucoup plus présentes hors de la communauté de la recherche grâce à l'avancement de l'informatique [54].

À ce sujet, l'entreprise Walmart est connue comme étant une pionnière dans l'implantation de solutions provenant du forage de données afin de maximiser les profits tout en gérant les inventaires de façon efficace [27]. Depuis, le forage de données s'est taillé une place très importante dans plusieurs industries, dont les télécommunications ([1, 58, 60]), la détection de fraudes ([11, 22, 36, 40, 46]) et même l'assurance([2, 3, 9, 10, 19, 20, 37, 56, 62]).

1.2 Le domaine de l'assurance

Le domaine de l'assurance a longtemps été exclusivement étudié par les actuaires. Ces derniers étant spécialisés dans la tarification, peu de travail a été fait pour résoudre scientifiquement des problèmes autres que ceux reliés à la tarification. Lors de la popularisation du forage de données dans les années 1980 et 1990, les assureurs y ont vu de belles opportunités d'acquérir de nouvelles informations à partir de la très grande quantité de données à leur disposition. En effet, les assureurs ont de très grande quantités de données sur leur clients et elles sont de grande qualité. Lors d'une réclamation, si l'assureur se rend compte qu'un client a menti au moment de la souscription, l'assurance peut devenir caduque. De par cette règle, l'hypothèse de véracité des données peut être effectuée sans trop de difficulté.

La grande qualité ainsi que la grande quantité de données ne sont pas des caractéristiques présentes dans tous les domaines où le forage de données est très populaire. Par exemple, dans le milieu de la télécommunication cellulaire, les compagnies ne disposent pas de beaucoup d'informations sur leur clientèle autre que le forfait choisi ainsi que l'endroit où est envoyé la facture (dans le cas d'un service post-payé). Dans ce do-

maine, il est très souvent nécessaire d'acheter des données démographiques peu précises agrégées au niveau des codes postaux afin d'obtenir plus d'informations sur la clientèle. Malgré ces données manquant de précision, l'industrie de la télécommunication cellulaire bénéficie grandement de l'apport du forage de données. Cette situation illustre bien tous les bénéfices que peut apporter le forage de données dans le milieu de l'assurance où les données disponibles sont de grande qualité.

1.3 Rétention de la clientèle

Des gains à plusieurs niveaux peuvent être effectués par le forage de données. Par exemple, la détection de fraude en assurance peut aider à réduire les pertes reliées à des réclamations frauduleuses. À cet effet, la coalition contre les fraudes en assurance (Coalition Against Insurance Fraud) estime qu'en 2006 aux États-Unis, la fraude a coûté 80 milliards de dollar [25]. Ce coût est malheureusement assumé par tous les assurés. Réduire la fraude aiderait grandement à diminuer le coût des assurances.

Des gains très importants peuvent aussi être effectués en rétention de la clientèle. En effet, à chaque année, les clients d'une compagnie d'assurance peuvent décider d'arrêter d'assurer leurs risques avec leur compagnie actuelle pour se tourner vers une autre compagnie. L'attrition dans le domaine de l'assurance est un problème majeur, non seulement par la perte de clients, mais aussi par le coût d'acquisition de nouvelle clientèle de remplacement qui est très élevé. Il a été estimé qu'en moyenne, il coûte 100$ pour acquérir un nouveau client tandis qu'il en coûte 30$ pour conserver un client [42]. De plus, une augmentation de la rétention de l'ordre de 5% amène une augmentation de 80% de la valeur à vie d'un client [45]. Une analyse semblable stipule que l'industrie de l'assurance a un des coût d'acquisition de la clientèle les plus élevé et que réduire le taux d'attrition de deux points serait équivalent à réduire les dépenses de la compagnie de 10% [57]. De plus, les anciens clients sont plus profitables que les nouveaux [12], ce qui montre bien l'importance de la rétention de la clientèle.

Une question très importante dans toute relation commerciale est l'élasticité des prix. En effet, chaque client est prêt à accepter une certaine augmentation dans le prix d'un

article ou d'un service sans chercher à changer ses habitudes de consommation. Par contre, au-delà de ce seuil, le client va magasiner afin de trouver une meilleure offre dans la compétition. Ce seuil est évidemment différent pour chacun des clients. Déterminer ce seuil est un travail très complexe étant donné que beaucoup de variables sont impliquées dans la relation. À cet effet, il a été démontré que le prix et la qualité du produit n'ont aucun effet sur l'élasticité du prix lorsque la satisfaction de la clientèle est prise en compte [28]. Le niveau de satisfaction de la clientèle est donc influencé en partie par le prix ainsi que par la qualité du produit offert. Ainsi, la relation de confiance entre le client et la compagnie est un facteur très important dans la satisfaction de la clientèle, surtout dans le domaine de l'assurance.

De plus, des études ont démontré que les clients qui ont une plus longue expérience avec une compagnie ont tendance à se fier plus fortement à leur expérience passée et à donner moins d'importance aux événements plus récents [5]. Ainsi, plus le client a une longue expérience avec sa compagnie d'assurance, moins le changement de prix affectera son choix de rester client au moment du renouvellement de sa police. De plus, la satisfaction de la clientèle explique 26% de la variance dans la durée de la relation client-assureur [5]. Finalement, une étude a démontré que la relation entre la satisfaction de la clientèle et sa tolérance au changement de prix suit une courbe en S inversé [33]. Ainsi, les plus grands impacts de la satisfaction de la clientèle sur la tolérance aux changements de prix se fait pour les valeurs extrêmes de satisfaction. De façon pratique, ceci démontre que la satisfaction de la clientèle devrait influencer la façon dont les compagnies font leur tarification, particulièrement les compagnies d'assurance de par la confiance nécessaire dans la transaction d'assurance.

Étant donné l'importance de la tarification pour les compagnies d'assurance ainsi que le très grand nombre de variables impliquées dans l'élasticité des prix pour la clientèle, le forage de données peut apporter une aide précieuse dans ce domaine. Beaucoup de bénéfices peuvent être apportés aux compagnies en apportant de la science dans l'art de bien comprendre l'élasticité des prix. Ainsi, une politique de tarification plus adaptée aiderait la compagnie à mieux prospérer tout en améliorant la satisfaction de la clientèle.

1.4 Contribution et structure de ce mémoire

Ce mémoire a pour but de développer un outil de prédiction de l'allure du portefeuille d'assurance détenu par un assureur en fonction du changement de taux proposé à chacun des assurés. Présentement, pour la majorité des assureurs canadiens, de simples modèles intuitifs sont utilisés à cette fin. Ces modèles sont généralement efficaces dans le cas de petits changements de taux, mais manquent de précision dans le cas de changements de taux moins habituels. Ainsi, ce mémoire propose un modèle capable de bien modéliser l'attrition suivant des changements de taux plus extrêmes. Une telle capacité apporte un réel avantage à ce modèle par rapport au modèle traditionnellement utilisé.

Dans le chapitre 2, certains concepts généraux de forage de données sont présentés. Le chapitre 3 présente le contexte de recherche, les différents modèles envisagés, l'approche qui sera utilisée ainsi que la mesure de performance qui a été développée spécifiquement pour ce problème. Le chapitre 4 présente les résultats obtenus par les différents modèles essayés. Une analyse des résultats obtenus sera présentée dans le chapitre 5.

CHAPITRE 2

APPROCHE GÉNÉRALE

Le forage de données est né d'un besoin de trouver des liens pertinents dans des bases de données contenant un très grand nombre d'observations ainsi que beaucoup de variables prédictives. La pertinence des liens découverts ainsi que la validité de la méthode utilisée pour trouver ces liens est d'une importance capitale pour ce genre de recherche.

2.1 Différentes tâches pour différent types d'algorithmes

Pour commencer, un algorithme d'apprentissage est une fonctionnelle qui prend en entrée un ensemble de données (appelé ensemble d'apprentissage) et qui produit une fonction prédictive. L'algorithme d'apprentissage construit et optimise un modèle mathématique qui a pour but d'effectuer une tâche prédictive. Il existe deux types d'algorithmes d'apprentissage effectuant des tâches complètement différentes.

2.1.1 Apprentissage supervisé

L'apprentissage supervisé consiste à demander à un algorithme d'apprendre une fonction dont on connait la valeur cible.

Plus formellement, un algorithme d'apprentissage supervisé prend en entrée un ensemble d'observations indépendantes et identiquement distribuées (*i.i.d)*) avec cibles connues (appelé ensemble d'entraînement) sur lequel le modèle se base pour effectuer des prédictions futures sur un ensemble différent de même distribution (appelé ensemble de test).

Une fonction de perte représentant le tort associé à une erreur doit être définie. Elle sert à quantifier l'erreur. Par exemple, pour une régression linéaire, la fonction de perte associée à chacune des observations est généralement la norme L_2 entre l'observation et la prédiction $L(\vec{x}) = (y - \hat{y}(\vec{x}))^2$ où y représente la "vraie" valeur cible associée au

prédicteur $\vec{x}$ et $\hat{y}(\vec{x})$ représente la prédiction de la valeur cible associée au prédicteur $\vec{x}$.

Par la suite, un critère d'apprentissage est défini à l'aide de la fonction de perte. C'est à l'aide du critère d'apprentissage que l'algorithme optimise le modèle. Le critère d'apprentissage fait généralement intervenir la fonction de perte ainsi qu'un terme de régularisation. Ce dernier sert à pénaliser l'utilisation de modèles trop complexes ayant tendance au sur-apprentissage (voir section 2.2). Par exemple, dans le cas de la régression logistique, le critère d'apprentissage est généralement de la forme $\hat{R}(L, D_n) = \frac{1}{n}\sum_{i=1}^{n} L(\vec{x}_i) + \lambda(D_n)$ où L est la fonction de perte, D_n représente l'ensemble d'entraînement, $\vec{x}_i$ représente les variables prédictrices de la i^e observation de l'ensemble d'entraînement et λ représente un terme de régularisation à définir par l'utilisateur (généralement un multiple de la somme du carré de la valeur des paramètres du modèle dans le cas de la régression linéaire). Lors de la phase d'entraînement, l'algorithme tente de minimiser ce critère d'apprentissage en faisant varier les paramètres du modèle.

C'est ce type d'algorithme qui sera utilisé dans ce travail afin de tenter de mieux prédire l'annulation en assurance automobile. Plusieurs tâches peuvent être effectuées par les représentants de ce type d'algorithmes :

- Régression : Consiste à prédire une valeur continue à partir des observations. Par exemple : prédire la hauteur en centimètres d'une personne en fonction de la hauteur de ses parents.
- Classification : Consiste à classer chacune des observations dans différentes catégories données. Par exemple : prédire la variété de fleur en fonction de la mesure de ses pétales et sépales.

2.1.2 Apprentissage non supervisé

L'apprentissage non supervisé consiste à demander à un algorithme d'apprentissage de faire une tâche sans pour autant en avoir soi-même la solution sur l'ensemble d'entraînement. Tout comme dans le cas de l'apprentissage supervisé, l'algorithme tente de minimiser le critère d'entraînement. Par contre, la fonction de perte pour un algorithme d'apprentissage non supervisé diffère par rapport à un algorithme d'apprentissage supervisé. En fait, la définition de la fonction de perte est très dépendante de la tâche effectuée

par le modèle.

Ce type d'algorithmes sert à effectuer des tâches complètement différentes des algorithmes supervisés.

- Regroupement (*clustering*) : Consiste à classer les observations dans des catégories non définies au départ. Seulement le nombre de catégories voulues (N) est parfois spécifié à l'algorithme. Les observations sont donc classées en N catégories les plus homogènes possibles.
- Estimation de densité : Consiste à prédire la probabilité qu'une certaine observation provienne de la distribution d'entraînement.

2.2 Le juste apprentissage

Étant donné l'ampleur des bases de données actuelles, il n'est pas possible de spécifier directement à l'ordinateur la nature des liens voulus. Si cela était possible, le forage de données ne serait d'aucun intérêt. Il est donc nécessaire de faire découvrir les liens par l'ordinateur sans que l'utilisateur soit au courant à l'avance de ce qui va ressortir de l'algorithme. On demande donc à l'algorithme d'apprendre les liens. La phase de découverte des liens est appelée la phase d'entraînement ou phase d'apprentissage.

Il est essentiel de pouvoir estimer la validité des liens découverts durant l'entraînement afin de pouvoir se rassurer que l'algorithme n'est pas complètement hors champ. Pour répondre à ces interrogations, la question de sur-entraînement (appelé *overfitting* en anglais) a été étudiée. Des approches différentes pour chaque modèle sont envisagées pour se prémunir de ce problème en contrôlant la capacité de modélisation du modèle. Par exemple, le principe de parcimonie est utilisé dans le cas de régressions [32], l'arrêt prématuré de l'entraînement est envisagé dans le cas des réseaux de neurones [53], l'élagage est utilisé dans le cas d'arbres de décision [6], tandis que l'utilisation de courbes ROC [21] est préconisée dans le cas de tâche de classification [47] afin de voir si la capacité du modèle est bien adaptée. Le terme de régularisation dans le critère d'apprentissage est aussi utile pour bien contrôler la capacité du modèle. En fait, la capacité du modèle influence la complexité des fonctions pouvant être représentées par ce modèle.

Un modèle possédant une grande capacité peut apprendre une fonction beaucoup plus complexe qu'un modèle possédant une faible capacité.

Plus formellement, il importe de mesurer l'erreur de généralisation du modèle afin d'avoir une idée de son pouvoir prédictif. L'erreur de généralisation est définie comme l'espérance de la perte sur de nouveaux exemples tirés de la même distribution que l'ensemble d'entraînement pour une fonction obtenue selon un algorithme d'apprentissage. De façon générale, le sur-entraînement est le fait d'apprendre des détails du jeu de données d'entraînement qui ne sont pas pertinents pour généraliser à un jeu de données semblable. Cela revient à apprendre par coeur la distribution d'entraînement. Ainsi, le sur-entraînement se produit lorsque la capacité du modèle est trop grande par rapport à la complexité de la tâche demandée.

Une façon simple de détecter le sur-entraînement consiste à diviser l'ensemble total des données en deux parties : l'ensemble d'entraînement et l'ensemble de test. Le premier servant, comme son nom l'indique, à entraîner le modèle et le second ne servant qu'à tester la capacité de généralisation du modèle. Lorsque le taux d'erreur augmente sur l'ensemble de test quand la capacité du modèle est augmentée (principalement à l'aide d'hyper-paramètres (voir section 2.3)), cela indique que le modèle commence à avoir une trop grande capacité et a tendance à apprendre des détails du jeu d'entraînement qui ne sont pas pertinents sur un ensemble de données différent provenant de la même distribution que le jeu d'entraînement. On dit alors que le modèle est sur-entraîné et il importe donc de diminuer la capacité du modèle car la capacité optimale a été dépassée. Cela se fait généralement par l'utilisation d'hyper-paramètres.

2.3 Les paramètres, les hyper-paramètres et l'utilité d'un ensemble de validation

Il existe une distinction importante entre un paramètre et un hyper-paramètre dans un modèle.

Un paramètre est une variable prenant une certaine valeur permettant au modèle de mieux modéliser les données. Par exemple, dans le cas d'une distribution normale $N(\mu, \sigma)$, μ et σ sont deux paramètres permettant à la distribution normale de mieux

représenter les données. En apprentissage machine, les paramètres doivent être optimisés sur la distribution d'entraînement car ils servent à mieux représenter les données étant donné une certaine forme de fonction choisie pour les représenter.

Un hyper-paramètre sert plutôt à faire un choix éclairé sur la forme de la fonction à utiliser pour modéliser les données. On peut voir les hyper-paramètres comme les paramètres d'un *a priori*. Par exemple, l'algorithme des k-plus-proches-voisins, un algorithme de classification (et régression) qui consiste à classifier (donner une valeur à) une observation selon la valeur prise par ses k plus proches voisins fait intervenir un hyper-paramètre. En effet, k est un hyper-paramètre dans le modèle. Il ne sert pas directement à modéliser la fonction choisie sur les données, mais plutôt à choisir la forme de la fonction à utiliser pour la modélisation. Un k trop petit aura pour effet d'être très centré sur la distribution d'entraînement et ne permettra pas à l'algorithme d'avoir une bonne capacité de généralisation, tandis qu'un k trop grand aura pour effet de produire un modèle n'ayant pas de capacité de discrimination. Les hyper-paramètres ne doivent pas être choisis à l'aide de la distribution d'entraînement, car cela aurait pour effet de favoriser le sur-apprentissage. Par exemple, dans l'algorithme des k-plus-proches voisins, si k est optimisé sur l'ensemble d'entraînement, une valeur de 1 sera choisie étant donné qu'elle produira un modèle parfait sur l'ensemble d'entraînement. Par contre, avec une telle valeur, le modèle n'aura pas une grande capacité de généralisation. Un autre exemple simple d'hyper-paramètre est l'allure et la sévérité du terme de régularisation du critère d'entraînement.

Pour optimiser les hyper-paramètres, il faut donc utiliser un jeu de données indépendant du jeu d'entraînement. L'utilisation de l'ensemble de test n'est pas très appropriée étant donné qu'elle favorisera une sous-estimation de l'erreur de test par la suite. Il faut donc construire un troisième ensemble de données indépendant des autres (bien entendu, les trois ensembles doivent être *i.i.d*). Ainsi, lorsque le modèle envisagé comporte un hyper-paramètre, il faut séparer le jeu de données en trois parties : l'ensemble d'entraînement, l'ensemble de validation et l'ensemble de test. Les hyper-paramètres du modèle pourront être optimisés à l'aide de l'ensemble de validation sans nuire à la capacité de généralisation du modèle ou sous-estimer l'erreur de test.

2.4 Approche paramétrique ou approche non paramétrique

Il existe deux grandes familles de modèles, la famille paramétrique et la famille non paramétrique. Ces deux méthodes sont basées sur des philosophies très différentes, mais peuvent aussi s'allier afin de développer des algorithmes hybrides.

2.4.1 Le côté paramétrique de la force

La famille paramétrique est basée sur le fait de poser un *a priori* sur la distribution d'entraînement. À ce sujet, la plus connue des méthodes paramétriques est sans contredit la régression linéaire. En effet, ce modèle est basé sur l'hypothèse que les données peuvent être représentées par un modèle de type :

$$\vec{y} = \mathbf{X}\vec{\beta} + \varepsilon, \quad \varepsilon \sim N(\vec{0}, \Sigma^2) \tag{2.1}$$

Le modèle repose sur le fait que la relation entre la cible ($\vec{y}$) et les prédicteurs ($\mathbf{X}$) est seulement linéaire et que les résidus de la régression suivent une loi normale. Si ces hypothèses ne sont pas validées *a posteriori*, le modèle ne peut pas être utilisé légitimement [59]. La connaissance des hypothèses effectuées est donc primordiale à l'utilisation de telles méthodes.

2.4.2 Le côté non paramétrique de la force

La famille non paramétrique quant à elle ne repose sur aucune hypothèse aussi forte sur la distribution d'entraînement. En fait, c'est la philosophie contraire qui est appliquée. À la place de faire une hypothèse sur la forme de la distribution d'entraînement, une hypothèse sur la façon dont est exprimée la variabilité de la fonction est effectuée. À ce sujet, une méthode non paramétrique très connue est le classificateur des k-plus-proches-voisins (voir section 2.3 pour plus de détails). Dans cette méthode, aucune hypothèse d'une forme stricte pour la distribution d'entraînement n'est effectuée. Par contre, l'hypothèse selon laquelle les points d'entraînement agissent comme leurs voisins est effectuée.

Les méthodes non paramétriques sont intéressantes du fait qu'il n'y a pas d'hypothèse importante à satisfaire, mais il y a un très grand risque de sur-apprentissage. Par exemple, comme mentionné précédemment, un classificateur du 1-proche-voisin aura généralement une performance parfaite sur l'ensemble d'entraînement, mais une très piètre sur tout autre ensemble. Avec les méthodes non paramétriques, il faut donc porter une très grande attention à la capacité de généralisation du modèle en contrôlant la capacité des modèles. En fait, les méthodes non paramétriques standards sont la plupart du temps locales. C'est à dire que la prédiction effectuée se base sur le voisinage du nouveau point. C'est de par cette caractéristique que le risque de sur-apprentissage se pointe.

2.4.3 Le côté non paramétrique non local de la force

Comme mentionné précédemment, les méthodes non paramétriques sont locales la plupart du temps. Beaucoup d'avantages sont reliés à ce type de modèles, mais ils apportent aussi beaucoup d'irritants. C'est à cet égard que les modèles non paramétriques non locaux peuvent être d'une très grande utilité. Ainsi, le réseau de neurones (ou *Artificial Neural Network* en anglais) est un digne représentant des modèles non paramétriques non locaux. En effet, le réseau de neurones artificielles ne demande pas de faire des hypothèses sur la distribution d'entraînement, mais ne produit pas non plus des prédictions basées uniquement sur le voisinage du nouveau point. Ce genre de modèle présente beaucoup d'avantages quant à la versatilité de l'approche et l'universalité des résultats obtenus. Par contre, les réseaux de neurones artificielles demandent tout de même une attention particulière pour éviter le sur-apprentissage de par leur très grande capacité de représentation (contrairement aux modèles non paramétriques locaux qui demandent une attention particulière pour éviter le sur-apprentissage de par leur prédiction locale).

2.5 Utilisation typique dans le domaine de l'assurance

Cette section sur ce qui se fait en forage de données dans le domaine de l'assurance est simplement un aperçu de ce qui se fait dans le domaine. Il ne se veut pas un exercice

exhaustif du fait que beaucoup de publications sont effectuées dans le domaine et qu'une exhaustivité n'apporterait pas beaucoup à cette recherche à cause de la très grande variété de tâches effectuées à l'aide du forage de données. Par contre, les articles précis quant aux méthodes utilisées sont rares de par le fait que les compagnies d'assurance désirent garder le secret des méthodes utilisées afin de s'assurer d'un avantage compétitif [19]. Les articles disponibles ne sont pas très précis pour la plupart. Tout de même, le forage de données est un sujet qui est voué à prendre de plus en plus de place en assurance [62].

2.5.1 Tarification

La tarification est un des principaux problèmes mathématiques dans le domaine de l'assurance. En effet, lorsqu'on vend un produit, il est primordial de savoir à quel prix le vendre. Par contre, l'assurance, contrairement aux produits conventionnels, est un produit pour lequel on effectue la tarification avant même de connaître combien le service offert va vraiment coûter. Ainsi, les actuaires sont des spécialistes de la tarification en assurance. Les techniques utilisées sont principalement issues de la statistique[1]. De plus, il est difficile d'innover dans le domaine de la tarification étant donné les législations sévères présentes dans plusieurs provinces canadiennes et états américains. En effet, afin de pouvoir utiliser une nouvelle façon de tarifer les assurances, il faut souvent démontrer sans l'ombre d'un doute la validité du modèle proposé. Les autorités sont réticentes à adopter de nouveaux modèles, particulièrement ceux qui sont moins facilement explicables (*e.g.* les réseaux de neurones). Il y a tout de même certaines recherches qui sont faites dans le domaine. Ainsi, il a été démontré que l'utilisation de mélange de réseaux de neurones avec fonction d'activation "softplus" pour la tarification apporte un réel avantage par rapport aux méthodes traditionnellement utilisées pour la tarification [10, 20].

$$softplus(s) = \log(1 + e^s) \tag{2.2}$$

Le mélange proposée comporte trois réseaux de neurones entraînés sur des distribu-

[1] Les modèles linéaires généralisés font toujours la loi en ce domaine

tions d'entraînement différentes. La distribution d'entraînement est séparée en ces trois groupes pour ensuite y entraîner un réseau de neurones différents sur chacun des segments :

- Pas de réclamation ;
- Des réclamations pour moins de 10000$;
- Des réclamations pour plus de 10000$.

Dans ces précédents articles, l'hypothèse de la présence de trois populations différentes à l'intérieur du portefeuille d'assurances est faite. Allant dans le même sens, un autre modèle de tarification a été développé se basant sur une idée semblable. Ainsi, un spline multi-adaptatif de régression[2] est construit dans les feuilles finales venant d'un arbre de classification et régression[3] (section 3.4.1). Ce type de modèles donne de bons résultats en assurance maladie [37].

2.5.2 Détection de fraude et analyse des réclamations

L'assurance est une industrie où la fraude est présente. Le coût de ces fraudes étant assumé par tous les autres assurés honnêtes, l'atténuation du nombre de fraudes peut apporter un grand avantage compétitif à une compagnie d'assurance en permettant de réduire le coût des assurances.

À ce sujet, un arbre de décision et une régression pas-à-pas (ou *stepwise* en anglais) peuvent être utilisées afin de voir pour quels types de réclamation d'assurance invalidité le recours à un évaluateur externe amène à une baisse des prestations payées par l'assureur [19]. Cela revient à déterminer les fraudes qui consistent à gonfler les demandes d'indemnisation.

L'analyse des réclamations est aussi un sujet en assurance qui peut être traité avec l'aide du forage de données. En effet, le règlement des réclamations est la principale responsabilité des assureurs, mais aussi la principale dépense des assureurs. À ce sujet, l'utilisation d'un modèle hybride d'arbre de décision couplé avec une régression linéaire peut être utilisé afin de classifier les réclamations selon leur propension à devenir coû-

[2] multi-adaptative regression spline (MARS)
[3] classification and regression tree (C&R tree)

teuses [37]. Aussi, une comparaison entre la régression logistique, les arbres CHAID (*CHi-squared Automatic Interaction Detector*) et les arbres C5.0 (une version améliorée de l'algorithme *Iterative Dichotomiser 3* (ID3)) a été faite afin de tenter de prédire la présence d'hyper-tension chez une personne afin de mieux connaître les coûts futurs de l'assurance santé d'une personne. Il a été trouvé que l'arbre CHAID est le meilleur des trois algorithmes pour ce problème [9].

2.5.3 Analyse intégrée d'annulation et de profitabilité

Un schéma intégré d'analyse d'annulation et de profitabilité peut être effectué à l'aide du forage de données.

À ce sujet, l'utilisation de réseaux de neurones apporte de bon résultats dans ce domaine. Les résultats sont meilleurs qu'avec un arbre de décision. Aussi, l'utilisation de regroupements effectués à l'aide des k-moyennes est souvent faite. Ainsi, les segments sont automatiquement créés à partir des données et on peut voir la profitabilité de chacun des segments et tenter de corriger le tir si un segment n'est pas profitable. De plus, on peut voir l'évolution de ces segments à travers les années si des données sur plusieurs années sont disponibles. Il est alors possible de voir la transformation du portefeuille à travers le temps. Finalement, un schéma contenant les deux précédents modèles est présenté afin de mieux tarifer les assurances [56].

2.6 L'analyse du non renouvellement

L'analyse du non renouvellement [4] en assurance est un sujet complexe et important. C'est un sujet d'étude important de par la nécessité d'avoir une bonne clientèle pour un assureur. Par contre, une bonne analyse du non renouvellement est difficile à produire. Au Canada, les contrats d'assurance dommage sont généralement d'une durée d'un an et renouvelables tous les ans. Il est donc possible pour les assurés de cesser un contrat en cours de terme avec pénalités et à la fin du terme sans aucune pénalité. Les deux problématiques se ressemblent dans le but, mais diffèrent largement dans les méthodes

[4] appelé "*churn*" en anglais

nécessaires pour arriver à de bon résultats.

Les articles traitant de cette problématique étant rares dans le domaine de l'assurance, des articles provenant du milieu de la télécommunication seront aussi considérés étant donné la similarité de la problématique. Par contre, la majeure différence réside dans l'analyse du non renouvellement de service en téléphonie cellulaire avec forfaits pré-payés où l'arrêt de service peut se faire tous les mois à la place de tous les ans. Ainsi, l'analyse temporelle est plus importante en télécommunication qu'elle l'est en assurance. Par contre, ces méthodes seront tout de même brièvement discutées dans les prochaines sections, car elles peuvent être source d'inspiration en assurance.

2.6.1 Non renouvellement à la fin du terme

L'analyse du non renouvellement en fin de terme est le plus simple exercise d'analyse de rétention de la clientèle en assurance au Canada. En effet, des pénalités sont infligées aux clients qui décident d'annuler leur assurance en cours de terme. De plus, le changement de taux effectué par une compagnie d'assurance devient effectif pour le client seulement en date du renouvellement. L'analyse en est facilitée de par le fait qu'il faut simplement prédire la probabilité de non renouvellement à une date précise dans le temps. La composante temporelle n'intervient donc pas dans le modèle.

À ce sujet, une étude comparative de prédiction de non renouvellement dans le domaine de la télécommunication montre la supériorité des arbres de décision et de la régression logistique sur les techniques plus traditionnelles [44]. Cette même étude a démontré que ces types de modèles gardent une valeur prédictive même si l'entraînement n'est pas effectué sur les plus récentes données.

Un plus récent article sur l'analyse du non renouvellement dans le domaine de la communication sans-fil estime la supériorité des arbres de décision sur la régression logistique [63]. Une bonne amélioration de la performance des arbres de décision repose dans l'agrégation des réponses des arbres de décisions [5] [38]. Ainsi, pour avoir de meilleures performances avec des arbres de décision, une option intéressante repose sur le fait d'effectuer une forêt aléatoire avec un rebalancement de classe tout en définissant

[5] appelée forêt aléatoire ou plus généralement *bagging*

des pénalité pour une mauvaise classification. La performance de ce genre de modèles est supérieure à un simple arbre de décision [64].

Une approche basée sur une première segmentation pour ensuite appliquer une régression logistique sur chacun des segments a aussi été développée. Ainsi, une régression logistique appliquée sur chacun des segments construits selon la rentabilité des clients (4 segments ont été construits pour cet article) est utilisée avec succès sans par contre la comparer avec d'autres méthodes [50].

Une approche hybride a aussi été développée afin d'améliorer la performance d'un simple modèle. Ainsi, un simple modèle est construit pour ensuite utiliser un classificateur qui prédit si le modèle de base se trompe dans sa prédiction. Enfin, le classificateur est appliqué sur les données d'entraînement afin de classifier chacune des observations. Si le modèle principal est prédit comme étant performant, celui-çi est utilisé sur cette observation. Un second modèle est entraîné pour les observations étant présenties pour ne pas être bien classifiées par le modèle principal. Cette technique hybride est significativement plus performante que de simplement utiliser le modèle principal [39].

2.6.2 Non renouvellement en cours de terme

L'analyse du non renouvellement en cours de terme présente une difficulté supplémentaire. En effet, le moment de l'arrêt du contrat est important dans cette situation et présente le réel défi. À cet effet, il a été démontré que l'historique de relation avec la compagnie d'assurance est importante pour déterminer la longévité de la relation entre le client et la compagnie d'assurance. Cette même étude propose d'étudier en profondeur les changements dans les caractéristiques de la police afin de mieux comprendre l'annulation de polices.[43].

La régression de Cox a aussi été utilisée afin de prédire le moment de l'annulation de la police. Un modèle différent pour chacun des mois de renouvellement a été bâti [29].

Aussi, un modèle des k-plus-proches-séquences (une sorte de k-plus-proche-voisins, mais pour séquences d'événements) a été développé pour prédire l'annulation de service en prenant en compte la dimension temporelle. Les résultats ont montré la supériorité de cette méthode sur les méthodes traditionnelles d'analyse de non renouvellement ne

prenant pas en compte la dimension temporelle [51].

CHAPITRE 3

MÉTHODOLOGIE

Pour mener à bien cette recherche, la première étape consiste à déterminer les besoins de la compagnie. Ensuite, l'inventaire des outils disponibles ainsi que des données pouvant être utilisées est effectuée. Puis, vient une présentation des modèles envisagés. Finalement, l'approche utilisée pour mettre ensemble les différents modèles envisagés ainsi que la mesure de performance servant à évaluer les différents modèles seront présentées.

3.1 Les besoins de la compagnie

Comme mentionné précédemment, la tarification en assurance est un sujet complexe de par le fait que la tarification est effectuée avant même que le service ne soit offert. C'est un des rares domaines où le coût du service offert n'est connu qu'une fois le contrat terminé. Ainsi, des ajustements de tarification sont très souvent nécessaires afin de demander le juste prix pour le service offert. Au Canada, les contrats d'assurance dommage sont généralement d'une durée d'un an et doivent donc être renouvelés à chaque année. C'est au moment du renouvellement annuel de leur contrat que les clients sont affectés par une nouvelle tarification qui aurait été mise en vigueur durant la durée du contrat venant d'expirer. Suite à l'ajustement, la prime demandée peut baisser, mais elle peut aussi monter. Le client a donc l'opportunité de magasiner une nouvelle assurance chez un concurrent s'il juge que la prime demandée n'est plus raisonnable. L'attrition cause des pertes monétaires importantes dans le domaine de l'assurance et se doit d'être contrôlée afin d'assurer une meilleure rentabilité à la compagnie.

Il est certain que tout changement apporté à la tarification a un impact sur la rétention de la clientèle. Intuitivement, une baisse de tarif est un gage de haut taux de rétention tandis qu'une grande hausse amènera probablement beaucoup de clients à se tourner vers des compétiteurs. Une telle analyse qualitative du comportement général de la clientèle

est simple à effectuer, mais il n'est pas aisé de quantifier le taux d'attrition produit par une nouvelle tarification. Une estimation précise est très importante afin de mieux comprendre comment bien tarifer les produits d'assurance. En effet, il peut être nécessaire d'effectuer une hausse de taux de 20% selon les actuaires. Par contre, une telle hausse aurait un effet désastreux sur la rétention de la clientèle. Il vaut peut-être mieux rajuster les prix graduellement afin de conserver la clientèle, même si la compagnie sera déficitaire quelques années dans certains segments de son porte-feuille. Ainsi, une quantification juste de l'attrition advenant un changement de taux est nécessaire afin de pouvoir bien doser les changements tarifaires demandés par l'équipe d'actuariat.

Normalement, une telle quantification est effectuée par des experts de longue date se basant sur leur flair et leurs connaissances du domaine. Une telle analyse donne généralement des résultats corrects, mais manque de rigueur et de précision. Un outil mathématique rigoureux étant capable de prédire avec une certaine précision l'attrition étant donné une certaine tarification serait donc le bienvenu. Un tel outil aiderait grandement la mise en place d'une tarification mieux planifiée en permettant d'obtenir un meilleur contrôle sur le niveau d'attrition.

L'outil désiré aurait par contre certaines spécifications essentielles :

- le modèle doit présenter une dérivée non nulle par rapport au changement de prix (*i.e.* le taux d'attrition doit varier de façon continue par rapport au changement de prix de la couverture) ;
- le modèle doit être le plus précis possible tout en fournissant une performance constante ;
- le modèle doit être efficace sur différents segments afin de pouvoir bien cerner l'évolution du portefeuille d'assurance suivant un changement de tarification ;
- le modèle doit être capable de produire une analyse juste sur des scénarios extrêmes (très grands changements de taux) ;
- le modèle doit faire du sens au point de vue des affaires et idéalement être facilement explicable aux patrons afin qu'ils puissent avoir confiance au modèle.

3.2 Les outils disponibles

Un autre but important du projet, outre un modèle performant, est de produire un modèle qui sera facilement exportable à d'autres activités de la compagnie, que ce soit un changement de territoire ou un changement de type d'assurance. De plus, une telle exportation devra être facile à effectuer, même pour quelqu'un qui n'est pas familier avec le forage de données. Il faut produire un outil simple d'utilisation, versatile et facilement intégrable dans le cadre de la compagnie.

Pour tenter de rendre les modèles produits plus facilement intégrables, l'outil de forage de données PASW-SPSS a été acheté et devrait être utilisé en priorité. De plus, l'utilisation de SAS, un outil statistique très répandu, est fortement encouragée par la compagnie, car les utilisateurs potentiels de l'outil sont très familiers avec SAS. Finalement, si aucun de ces deux outils n'est utilisable pour les besoins du modèle, l'utilisation d'un langage de programmation comme Excel ou R pourrait être envisagée. Évidemment, l'utilisation de PASW-SPSS et SAS amène des contraintes supplémentaires à cause de leur rigidité.

3.3 Les données utilisées

Un point important en faveur de l'utilisation du forage de données dans le domaine de l'assurance est la très grande qualité des données. Par contre, comme dans tous les domaines, un contrôle de qualité rigoureux doit être effectué afin de s'assurer de la cohérence de l'analyse.

Pour l'élaboration de cet outil, beaucoup de données de différentes sources ont été utilisées. De plus, seulement deux années de données historiques sont utilisées à cause du coût élevé d'acquisition de données de position concurrentielle. Si le modèle s'avère efficace, plus d'efforts pourront être mis dans le futur afin d'avoir un modèle entraîné avec plus de deux ans de données passées.

3.3.1 Les données internes

Une importante source de données est certainement les informations dont dispose l'assureur. Ces données proviennent en partie des questions posées à la souscription. On y retrouve aussi les informations sur les articles assurés ainsi que les protections choisies. Ce sont ces données qu'on suppose véridiques. Ces champs contiennent des informations très intéressantes pour bien caractériser chacune des polices.

3.3.2 Position concurrentielle

Il est certain que le changement de tarification de l'assurance influence la volonté des clients de magasiner. Par contre, magasiner ne signifie pas pour autant annuler sa police d'assurance pour passer chez un compétiteur. En effet, si tous les compétiteurs chargent plus cher, il n'est pas intéressant de changer d'assureur. Ainsi, un indice de la compétitivité de la compagnie est essentiel au bon fonctionnement du modèle.

À cet effet, le prix chargé par cinq autres assureurs pour douze profils types dans chacun des FSA[1] de la région étudiée est disponible. De plus, nous disposons de ces mesures pour les deux dernières années à des intervalles de six mois. Nous avons donc quatre points dans le temps où nous connaissons le prix demandé par certains compétiteurs pour les douze profils types choisis.

3.3.3 Les données externes

En plus de ces deux précédentes sources de données, des données achetées sont aussi disponibles. Ces données externes contiennent des informations sociologiques et démographiques au niveau des codes postaux.

Statistique Canada récolte énormément de données au moment des recensements. Par la suite, l'agence gouvernementale publie au niveau des FSA les données qu'elle a ramassées. Ce niveau d'agrégation est choisi afin de préserver la vie privée des citoyens. Par contre, certaines compagnies se spécialisent dans le changement d'échelle de ces données. Par des moyens mathématiques secrets, ils tentent de transformer ces données

[1] *Forward Sortation Area*, ou plus simplement les trois premières lettres du code postal

pour les mettre au niveau des codes postaux. À ce niveau, les données sont maintenant utilisables au point de vue du marketing. Par contre, ces moyens de changer l'échelle des données étant secrets, il n'est pas possible de savoir à quel point cela est précis et rigoureux. De plus, même en faisant l'hypothèse que le changement d'échelle est fait parfaitement, les informations fournies ne sont qu'au niveau du voisinage des clients visés. Ces données sont donc prises en compte parce qu'elles sont disponibles, mais aucune demande n'aurait été faite afin d'acheter ces données pour le bien de ce projet. Les données disponibles à l'interne ainsi que les indices de positions concurrentielles sont d'une beaucoup plus grande utilité ainsi que d'une précision beaucoup supérieure. Ces données externes doivent être vues comme donnant de l'information sur le voisinage des clients plutôt que sur les clients eux-mêmes.

3.3.4 Nettoyage des données

Une partie très importante du travail de forage de données réside dans le fait de s'assurer de la bonne qualité des données utilisées par les modèles. En effet, l'adage "déchets en entrée produit des déchets à la sortie" [2] s'applique parfaitement au domaine du forage de données [48]. Plus clairement, un modèle ne pourra jamais donner de bons résultats si les données en entrée ne sont pas de bonne qualité. Il est généralement convenu que le choix du modèle a une incidence importante dans la précision de celui-ci, mais il ne faut surtout pas oublier de s'assurer de la qualité des données. Généralement, au moins la moitié du temps réservé au projet passe en préparation de données.

Parce que les données proviennent de plusieurs bases de données différentes qui ne sont pas toujours cohérentes entre elles et aussi parce que ces différentes bases de données peuvent contenir des erreurs, beaucoup de travail est nécessaire pour s'assurer de la "propreté" des données.

Ainsi, la première étape consiste généralement à s'assurer que chacun des champs contient des informations appropriées. À cette étape, il n'est pas rare de voir plusieurs façons d'encoder dans la base la même information. Par exemple, le champ "sexe du client" peut contenir "M", "F" ainsi que "H". Les valeurs "M" et "H" référant toutes deux

[2] "*garbage-in, garbage-out*" en anglais

au sexe masculin, une harmonisation est nécessaire afin de ne pas mélanger le modèle inutilement. À cette étape de vérification de toutes les valeurs possibles dans chacun des champs, une attention particulière aux informations non nécessaires doit aussi être portée. Ainsi, certaines valeurs dans des champs ne présentent que des différences mineures au niveau administratif qui ne sont pas nécessaires à la bonne modélisation. Des regroupements peuvent donc être effectués à l'intérieur de certains champs afin de réduire la cardinalité de ces derniers. Cette mesure simple améliorera grandement la robustesse du modèle tout en n'enlevant pas de précision. Enfin, dans la vérification de la validité des valeurs présentes dans chacun des champs, une grande attention doit aussi être portée aux champs présentant des valeurs manquantes. Les modèles qui utiliseront ces données demandent généralement d'avoir tous les champs complets. Une valeur manquante peut simplement vouloir dire un manque d'information, mais peut aussi être le fruit d'une dépendance avec une autre variable. Il ne faut donc pas simplement régler le problème en y rajoutant une valeur signifiant le manque d'information. Par exemple, une valeur vide dans le champ "propriétaire ou locataire" ne signifie pas un manque d'information sur le logis du client, mais plutôt que ce client ne possède pas d'assurance habitation avec notre compagnie. Il est donc primordial de toujours comprendre pourquoi une valeur vide est présente afin de savoir comment s'en occuper. Si aucun traitement raisonnable n'est possible pour certaines entrées ou pour un champ présentant beaucoup de valeurs vides, la suppression de certaines entrées ou même du champ au complet peut être envisagée.

Une fois les champs nettoyés, il faut ensuite s'assurer que les données provenant de bases de données différentes soient cohérentes et que le lien s'est bien fait. Par exemple, lier des données de différentes bases de données par le nom du client n'est pas une bonne idée parce que certains noms sont courants. Dans ce cas, le numéro de client qui est unique devrait plutôt être privilégié. En liant des données provenant de différentes sources, il faut toujours s'assurer que le nombre de lignes de données est constant et que tout se passe comme prévu. De plus, avec la liaison de données provenant de différentes sources, il est possible de valider la qualité de certains champs qui sont présents dans différentes bases de données. Par exemple, il est possible de vérifier la qualité du champ "présence d'assurance habitation en plus de l'assurance automobile" en liant les bases de

données d'assurance automobile et d'assurance habitation. Des vérifications de ce type sont toujours intéressantes à effectuer afin de s'assurer le plus possible de la qualité des champs. De plus, des champs présentant des valeurs manquantes peuvent être complétés à l'aide de données provenant d'une autre base de donnée.

De telles vérifications sont longues à effectuer, mais ce temps est très bien investi. Une très grande minutie est nécessaire lors de telles opérations.

3.3.5 Utilisation des données de position concurrentielle

Comme mentionné précédemment, les données de position concurrentielle sont disponibles pour 12 profils types et seulement pour 4 dates passées dans chacun des FSA. Cette situation est due à la longueur du travail nécessaire à l'extraction de ces données.

Étant donné que les clients réels de la compagnie sont différents des douze profils types et que des renouvellements se produisent chaque jours, pas seulement à 4 dates précises, il faut trouver un moyen d'approximer la position concurrentielle de la compagnie pour chacun des clients. Ce travail sera fait en deux étapes qui consistent à trouver le profil dans les douze qui est le plus proche de chacun des clients et par la suite déterminer laquelle des quatre dates présentant une mesure doit être utilisée pour chacun des clients.

3.3.5.1 Détermination du profil le plus proche

La première étape consiste simplement à trouver pour chacun des clients dans la base de données lequel des douze profils types est le plus semblable à lui. Pour ce faire, une "association floue" [3] sera effectuée. C'est une sorte de fusion qui tire son fonctionnement de la logique floue [4], une sorte de logique fonctionnant avec un certain degré de vérité. En logique conventionnelle, un énoncé peut être classé vrai ou faux, mais en logique floue, un événement peut être classé vrai à 70% par exemple. La procédure d'association se base sur cette idée de pourcentage de vérité ou plutôt pourcentage de similitude dans notre cas.

[3] "*fuzzy merge*" en anglais
[4] "*fuzzy logic*"

En effet, il faut trouver une mesure de position concurrentielle pour chacun des clients de la base de données, mais une association parfaite n'est pas possible. Il faut donc trouver lequel des douze profils types est le plus près de chaque client. Pour ce faire, un pointage de similitude sera créé pour chacun des profils types pour chaque client de la base de donnée. Il sera donc aisé de voir lequel des douze profils types est le plus similaire à chacun des clients.

Les douze profils types présentent chacun seize caractéristiques qui les distinguent les uns des autres. Ainsi, pour chacune de ces caractéristiques, une valeur entre 0 et 1 sera attribuée selon la similitude entre le client et chacun des profils types. Cette valeur sera rajoutée au pointage de chaque profil type. Une valeur de 0 signifie que le client et le profil type sont totalement différents, tandis qu'une valeur de 1 signifie une similarité totale. De cette façon, chacun des clients de la base de données aura un pointage entre 0 et 16 pour chacun des douze profils types. Finalement, pour déterminer le profil type le plus proche de chacun des clients, il faut simplement regarder lequel de ces profils obtient le plus grand pointage.

3.3.5.2 Détermination de la bonne date de prise de mesure

Une fois le profil type le plus similaire trouvé pour chacun des clients, il faut maintenant savoir à quelle date prendre la mesure de l'indice de compétitivité. En effet, nous avons deux ans de données, mais seulement quatre mesures temporelles de compétitivité. Nous avons une mesure à chaque six mois. Par contre, les renouvellements arrivent chaque jour , pas seulement 4 jours en deux ans.

Plusieurs méthodes d'association sont possibles :

- prendre le point de mesure le plus proche du moment du renouvellement ;
- prendre le point de mesure le plus proche du renouvellement en autant que la mesure soit effectuée avant le renouvellement lorsque cela est possible ;
- prendre le point de mesure le plus proche du renouvellement en autant que la mesure soit dans la même période tarifaire pour notre compagnie.

Ces trois méthodes sont valables, mais reposent sur des hypothèses différentes. La première repose sur l'hypothèse selon laquelle la position concurrentielle change de fa-

çon continue. La seconde repose sur l'hypothèse selon laquelle la position concurrentielle change doucement, mais peut seulement être mesurée dans le passé. La troisième méthode repose sur le fait que les changements majeurs à la position compétitive sont attribuables à notre compagnie. Ces trois hypothèses sont plausibles, mais la troisième méthode a été retenue. Elle semble l'hypothèse la plus plausible dans la situation. En effet, durant ces deux dernières années, de gros changements ont été apportés à la tarification de la compagnie afin de charger un prix plus juste aux clients. Ainsi, la compagnie a été en grande partie responsable de son changement de position compétitive. Il est donc raisonnable d'associer à chaque renouvellement une mesure de compétitivité prise durant la même période tarifaire pour notre compagnie.

3.3.6 Changement de prime naturel

Finalement, une mesure de changement naturel de prime est calculée pour chacun des clients. En effet, chaque année, les clients doivent payer un montant différent pour leur assurance. Ce changement dans la prime provient de deux sources distinctes : l'évolution du dossier et une décision de la compagnie de rajuster sa tarification. Par exemple, à chaque année, le véhicule assuré devient plus vieux et donc diminue de valeur, ce qui aurait pour effet de réduire le montant de la prime d'assurance. Les clients s'attendent normalement à un changement à leur prime d'assurance à cause de l'évolution normale du dossier, mais aussi pour tout changement apporté par eux à leur couverture. Par contre, les changements de primes décidés par la compagnie pour rajuster sa tarification ne sont pas attendus par les clients. Il serait donc intéressant de rajouter dans le modèle les changements naturels de la prime, car ces changements sont "attendus" par les clients, mais nous ne savons pas comment les clients réagissent à de tels changements.

Pour ce faire, l'augmentation naturelle de la prime a été mesurée en calculant la prime qui serait demandée au client s'il avait cette année les caractéristiques de l'an passé et en la soustrayant du montant de la prime demandée au renouvellement. Plus clairement :

Soit $\Pi(x_i|k)$ la prime demandée au temps i en prenant en compte les caractéristiques

du client au temps k

Soit $\Pi_i = \Pi(x_i|i)$ la prime normalement demandée au temps i

Alors,

$$AUG = \Pi_i - \Pi(x_i|i-1) \tag{3.1}$$

est calculée, où AUG représente l'augmentation de la prime seulement due à l'évolution du dossier. Par contre, l'hypothèse de la constance de la grille tarifaire doit être faite. Cette hypothèse n'est pas totalement satisfaite, mais la mesure AUG reste tout de même assez précise dans ce cas étant donné que seulement une année sépare la grille tarifaire et les caractéristiques du client. De plus, la précision de la mesure AUG est très correcte vue l'utilisation qui sera faite de la variable.

De plus, évidemment, le changement de prime est aussi utilisé dans le modèle sous forme absolue (équation 3.2), mais aussi sous forme relative (équation 3.3).

$$\Delta\Pi_i = \Pi_i - \Pi_{i-1} \ , \tag{3.2}$$

$$\Delta\Pi_{iRel} = \frac{\Pi_i - \Pi_{i-1}}{\Pi i} \ . \tag{3.3}$$

3.4 Les modèles utilisés

Pour mener à bien la tâche, plusieurs modèles bien connus ont été utilisés.

3.4.1 Arbre de classification et régression

L'arbre de classification et régression [5] [7] est un modèle de classification locale non paramétrique largement répandu dans le domaine extra-universitaire. En effet, ce type de modèle connu de longue date et implanté dans la plupart des logiciels statistiques commerciaux présente plusieurs avantages intéressants. Il est simple d'utilisation, simple à expliquer à un patron (ce qui n'est pas négligeable en industrie) et capable de modéliser des interactions non linéaires entre les variables. Par contre, ce type de modèle ne pré-

[5] *Classification and Regression Tree* ou *C&R Tree*

sente pas que des avantages. Plusieurs irritants sont aussi associés à ce type de modèle. Par exemple, il ne peut modéliser facilement une relation linéaire, ne présente pas de solution continue dans le cas de tâche de régression et ne peut modéliser facilement des relations très complexes.

3.4.1.1 La mécanique sous-jacente

L'arbre C&R [7, 34] est un type d'arbre construit dans le but de réduire l'impureté des noeuds créés. L'impureté est définie par l'hétérogénéité d'un noeud. Dans le cas d'une tâche de classification où la cible est binaire (Comme par exemple, renouvellement de la police ou non renouvellement), l'arbre tente de produire des noeuds où tous les individus présents ont le même comportement. Chaque noeud est séparé en deux sous-noeuds jusqu'à l'obtention de noeuds finaux satisfaisants. L'algorithme est théoriquement plutôt simple. Pour une tâche de classification, l'arbre essaie chaque séparation possible sur chacun des prédicteurs et en calcule l'impureté. Par la suite, la séparation choisie est celle dont l'impureté est minimale. Plusieurs mesures d'impureté sont généralement utilisées, mais dans notre cas c'est la mesure de Gini qui a été sélectionnée de par sa versatilité. L'impureté, selon le critère de Gini se calcule de la sorte :

$$g(t) = \sum_{j \neq i} p(j|t)p(i|t) = 1 - \sum_{j} p(j|t)^2 \tag{3.4}$$

où

t = le noeud en question

j = chacune des catégories (renouvellement ou non renouvellement)

i = catégorie dominante dans le noeud

$p(j|t) = \frac{p(j,t)}{p(t)}$

$p(j,t) = \frac{\Pi(j)N_j(t)}{N_j}$

$\Pi(j)$ est la proportion d'observations dans la catégorie j

$N_t(t)$ est le nombre d'observations de la catégorie j dans le noeud t

N_j est le nombre total d'observations dans la catégorie j

$p(t) = \sum_j p(j,t)$.

De plus, un coût de mauvaise classification peut être précisé afin de compenser pour un débalancement de classe. Lorsqu'un coût de mauvaise classification est défini, la mesure d'impureté de Gini devient donc

$$g(t) = \sum_{j \neq i} C(i|j)p(j|t)p(i|t) \ . \tag{3.5}$$

Avec *C(i|j)* signifiant le coût défini pour classer dans la classe i une observation provenant de la classe j.

Une fois la mesure d'impureté définie, il est maintenant possible de définir comment effectuer chacune des séparations. En effet, une séparation peut être effectuée à un très grand nombre de places pour chacune des variables. L'endroit de la séparation (sep) est décidé dans le but de maximiser cette mesure :

$$\phi(sep,t) = g(t) - P_L \ g(t_l) - P_R \ g(t_R) \tag{3.6}$$

où

sep = endroit de la séparation, cela définit comment sont créés les noeuds enfants

P_L signifie la proportion du noeud parent qui va dans le noeud enfant de gauche

t_R signifie les observations présentes dans le noeud enfant de droite.

On définit $P_L = \frac{P(t_L)}{P(t)}$.

Comme mentionné plus haut dans la section 2.4.2, les modèles non paramétriques locaux nécessitent généralement une attention particulière pour éviter le sur-apprentissage. L'arbre de décision n'y fait pas exception. En effet, les dernières branches d'un arbre bâti jusqu'à ce que chaque noeud final soit homogène sont généralement basées sur des spécificités du jeu d'entraînement. Pour avoir un modèle capable de généraliser sur un ensemble de test, il faut éliminer les branches construites uniquement avec des spécificités de l'ensemble d'entraînement. Pour ce faire, un élagage de l'arbre doit être effectué. Cet élagage peut être effectué manuellement lorsque le nombre d'arbres à effectuer est petit ou lorsqu'un contrôle de l'aspect de l'arbre est désiré. Sinon, l'élagage peut être

effectué en tentant de minimiser un critère se basant sur le coût de mauvaise classification, la complexité de l'arbre et un certain hyper-paramètre pénalisant le grand nombre de noeuds finaux,

$$R_\alpha(T) = R(T) + \alpha|\ddot{T}| \tag{3.7}$$

où

$|\ddot{T}|$ = nombre de noeuds finaux (sans enfants)

$R(T) = \sum_{t\in\ddot{T}} r(t)$ est le risque de mauvaise classification de l'arbre T

$r(t) = \frac{1}{N}\sum_j N_j(t)C(j^*(t)|j)$ est l'estimation du risque de mauvaise classification pour le noeud t

N est le nombre d'observations dans le jeu d'entraînement

j représente les classifications possible

$N_j(t)$ représente le nombre d'observations de la classe j dans le noeud t

$j^*(t) = \min_i \sum_j C(i|j)p(j|t)$ représente la prédiction de l'arbre pour le noeud t

α est l'hyper-paramètre de pénalisation du nombre de noeuds finaux.

3.4.2 Régression logistique

La régression logistique [24] est une technique statistique bien connue pour prédire une cible binaire. C'est en fait un cas particulier de modèle linéaire généralisé [6] construit pour prédire une cible provenant d'une distribution Bernouilli. C'est la fonction logistique (équation 3.8) qui est appliquée dans le modèle linéaire généralisé, c'est à dire

$$l(z) = \frac{e^z}{e^z+1} = \frac{1}{1+e^{-z}} \tag{3.8}$$

Ce type de modèle est très utilisé en industrie pour sa simplicité d'exécution, son court temps d'entraînement, son explicabilité (par opposition aux réseaux de neurones qui ne sont pas facilement explicables) ainsi que la justesse des résultats obtenus. Par contre, comme tous les modèles, il ne possède pas que des qualités. Ce type de modèle possède une faible capacité, il ne peut modéliser des fonctions trop complexes de par le

[6] *Generalised Linear Model* ou GLM

fait qu'il ne prend en compte que les relations linéaires entre les prédicteurs. De plus, il est très lourdement affecté par des valeurs aberrantes dans les données. Enfin, une attention particulière doit aussi être portée à l'élimination de variables colinéaires qui rendent impossible l'optimisation du modèle sans la présence d'un terme de régularisation de la capacité du modèle. Ce modèle est tout de même un des plus utilisés dans le cas de modélisation de distribution Bernouilli.

3.4.2.1 La mécanique sous-jacente

La régression logistique [24] peut s'écrire

$$f(\vec{X}_i) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_{i,1}+\beta_2 X_{i,2}+...+\beta_k X_{i,k})}} \ , \tag{3.9}$$

où $\vec{X}_i$ représente les variables explicatives de l'observation i et y_i représente la prédiction.

Ainsi, $y_i = \delta_{f(\vec{X}_i)>0.5}$ représente la prédiction du modèle pour l'observation i. Idéalement, y_i aurait toujours la même valeur que Y_i, la valeur cible. En fait, nous pouvons écrire

$$p(y_i) \equiv Pr(Y_i = y_i) = f(\vec{X}_i)^{Y_i}(1-f(\vec{X}_i))^{1-Y_i} \ . \tag{3.10}$$

De façon équivalente, on peut écrire l'équation(3.9) sous la forme

$$\log_e\left(\frac{f(\vec{X}_i)}{1-f(\vec{X}_i)}\right) = \beta_0+\beta_1 X_{i,1}+\beta_2 X_{i,2}+...+\beta_k X_{i,k} \ . \tag{3.11}$$

Aussi,

$$\frac{f(\vec{X}_i)}{1-f(\vec{X}_i)} = e^{\beta_0+\beta_1 X_{i,1}+\beta_2 X_{i,2}+...+\beta_k X_{i,k}} \tag{3.12}$$

$$= e^{\beta_0} e^{\beta_1 X_{i,1}} e^{\beta_2 X_{i,2}} ... e^{\beta_k X_{i,k}} \ . \tag{3.13}$$

La forme vectorielle de l'équation (3.11) est simplement

$$\log_e \left(\frac{f(\vec{X}_i)}{1-f(\vec{X}_i)} \right) = \vec{X_i^T} \vec{\beta} \ , \tag{3.14}$$

où $\vec{X_i^T} = (1, X_{i,1}, X_{i,2}, ..., X_{i,k})$ la rangée i de la matrice $\mathbf{X}$ et $\vec{\beta} = (\beta_0, \beta_1, \beta_2, ..., \beta_k)$ le vecteur de paramètres.

Pour pouvoir optimiser $\vec{\beta}$, il faut trouver la vraisemblance du modèle et ensuite tenter de la maximiser.

Ainsi, dans les hypothèses de départ, nous prenons pour acquis que le jeu de données est i.i.d. [7] et provient d'une distribution Bernouilli. De ce fait, en se servant de l'équation (3.10) , nous pouvons écrire la probabilité jointe comme ceci

$$\begin{aligned} p(y_1, y_2, ..., y_n) &= p(y_1)p(y_2)...p(y_n) \\ &= \Pi_{i=1}^n p(y_i) \\ &= \Pi_{i=1}^n f(\vec{X}_i)^{y_i}(1-f(\vec{X}_i))^{1-y_i} \\ &= \Pi_{i=1}^n \left(\frac{f(\vec{X}_i)}{1-f(\vec{X}_i)} \right)^{y_i} (1-f(\vec{X}_i)) \ . \end{aligned} \tag{3.15}$$

En reprenant le résultat de l'équation (3.14) et en effectuant certaines manipulations algébriques simples, on peut obtenir une expression plus utile pour l'équation (3.15).

$$p(y_1, y_2, ..., y_n) = \Pi_{i=1}^n \left(\exp(\vec{X_i^T} \vec{\beta})^{y_i} \left(\frac{1}{1+\exp(\vec{X_i^T} \vec{\beta})} \right) \right) \ . \tag{3.16}$$

Alors, la vraisemblance du modèle peut s'écrire comme :

$$L(\vec{\beta}) = P(y_1, y_2, ..., y_n | \mathbf{X}) = \Pi_{i=1}^n \left(\exp(\vec{X_i^T} \vec{\beta})^{y_i} \left(\frac{1}{1+\exp(\vec{X_i^T} \vec{\beta})} \right) \right) \ . \tag{3.17}$$

De par certaines propriétés intéressantes des fonctions exponentielles, l'utilisation de

[7] indépendant et identiquement distribué

la log-vraisemblance devient beaucoup plus simple. Il est possible de l'écrire ainsi

$$LL(\vec{\beta}) \equiv log(L(\vec{\beta})) = \sum_{i=1}^{n} Y_i \vec{X_i^T} \vec{\beta} - \sum_{i=1}^{n} log(1 + e^{\vec{X_i^T}\vec{\beta}}) \ . \tag{3.18}$$

La dérivée partielle de la log-vraisemblance par rapport à $\vec{\beta}$ se calcule

$$\begin{aligned} \frac{\partial LL(\vec{\beta})}{\partial \vec{\beta}} &= \sum_{i=1}^{n} Y_i \vec{X_i} - \sum_{i=1}^{n} \left(\frac{exp(\vec{X_i^T}\vec{\beta})}{1 + exp(\vec{X_i^T}\vec{\beta})} \right) \vec{x_i} \\ &= \sum_{i=1}^{n} Y_i \vec{X_i} - \sum_{i=1}^{n} \left(\frac{1}{1 + exp(-\vec{X_i^T}\vec{\beta})} \right) \vec{x_i} \ . \end{aligned} \tag{3.19}$$

Il est donc possible de trouver la valeur optimale du vecteur $\vec{\beta}$ en forçant (3.19) à égaler zéro. Par contre, l'optimisation de $\vec{\beta}$ n'est pas triviale et ne possède pas de solution explicite. Il faut donc passer par des méthodes numériques. La méthode généralement utilisée pour effectuer cette optimisation est celle de Newton-Raphson. Cette méthode nécessite par contre l'inversion d'une matrice faisant intervenir **X**. Pour s'assurer que cette matrice soit inversible, il faut que **X** soit non singulière, ce qui demande donc d'éviter la colinéarité entre les variables présentes dans le modèle. Pour plus de détails à propos de cette technique d'optimisation, voir [8].

Cette technique est très probablement la plus utilisée en assurance pour la modélisation du non renouvellement.

3.4.3 Ajustement d'un polynôme

Une technique paramétrique d'interpolation en deux dimensions très connue en mathématique est l'ajustement polynomial [8]. Cette technique consiste à tenter de faire passer un polynôme de degré donné par les points d'entraînement. Cette technique est seulement possible en présence d'un seul prédicteur qui doit absolument être continu. Cette méthode bien connue est intéressante, car elle produit une solution analytique relativement simple à évaluer et ne demande pas trop de temps de calcul pour arriver à une bonne solution. Par contre, ce type de méthode possède un très grand désavantage

qui fait que son utilisation pour l'extrapolation est généralement proscrite. En effet, cette méthode produit une solution qui diverge rapidement en dehors du domaine des points d'entraînement. Cette situation est due au comportement des polynômes. Par contre, dans notre cas, nous pouvons dire que lorsque le changement de prime est très négatif, le taux d'attrition sera nul, tandis que lorsque l'augmentation tarifaire est très positive, le taux d'attrition sera énorme. Par contre, il faut tout de même faire attention, car le taux d'attrition doit toujours se situer entre 0 et 100%. Cette méthode peut être intéressante, mais elle doit être étudiée avec de grandes précautions. On y fait l'hypothèse que la fonction à modéliser se comporte comme un polynôme de degré k, k étant une valeur à déterminer manuellement.

3.4.3.1 La mécanique sous-jacente

Pour s'assurer d'avoir un polynôme qui passe par tous les points d'entraînement (disons n points), il faut bâtir un polynôme de degré au plus $n-1$ (un degré moindre est possible dans certains cas particuliers). Par contre, dans notre cas, beaucoup de points sont présents et un polynôme d'une très grande dimension aura un comportement erratique entre les points. Comme nous désirons plutôt avoir une approximation lisse de la courbe, nous devons essayer d'y ajuster un polynôme d'ordre plus petit. Il est évident que dans ce cas, le résultat ne sera pas parfait sur les points d'entraînement, mais la capacité de généralisation du modèle en sera grandement augmentée. Pour facilement effectuer une telle opération, une régression linéaire [59] a été mise à contribution. En effet, il est possible de simplement créer des nouvelles variables comme étant des puissances de la variable prédictrice voulue. Ainsi, pour ajuster un polynôme de degré k sur les données, une régression linéaire de cette forme a été ajustée :

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_k x^k \tag{3.20}$$

Une telle procédure permet d'ajuster efficacement un polynôme de degré k sur les données. Il faut tout de même s'assurer que $k < n$ afin d'avoir un modèle cohérent.

3.4.4 Lissage local d'un nuage de points

Le lissage local d'un nuage de points [8] est une technique non paramétrique locale pour tenter d'approximer une courbe ayant des prédicteurs continu. Cette méthode initialement découverte en 1979 [13] et améliorée en 1988 [14] souffrait d'une très grande sensibilité aux valeurs aberrantes. Ainsi, en 1991 [15] , une version robuste de cette méthode fut développée. Cette version robuste de la méthode est implantée en SAS [16] sous le nom *PROC LOESS*. C'est cette implantation qui sera utilisée lors de la modélisation. Cette technique est intéressante du fait qu'elle peut approximer n'importe quelle courbe parce qu'elle est non paramétrique. Elle ne demande pas d'hypothèse de départ sur la distribution à modéliser, ce qui la rend très intéressante. Par contre, cette technique est gourmande côté calculs, il faut donc l'utiliser avec des petits jeux de données. De plus, cette technique ne produit pas de solution explicite au problème d'interpolation. Le jeu d'entraînement doit toujours être présent pour pouvoir prédire un nouveau point, ce qui cause quelques inconvénients par rapport à la régression logistique qui ne nécessite pas une grande quantité de mémoire afin de contenir le modèle. De plus, une extrapolation avec ce type de modèle n'est pas toute indiquée.

3.4.4.1 La mécanique sous-jacente

La méthode est conceptuellement simple, mais nécessite une bonne puissance de calcul. Elle consiste à bâtir une courbe par morceaux en effectuant des interpolations polynomiales impliquant $100\lambda\,\%$ des données d'entraînement les plus proches du point à estimer. L'hyper-paramètre λ est appelé l'hyper-paramètre de lissage. Habituellement, sa valeur se situe entre 0,2 et 0,5. Évidemment, plus sa valeur est petite, plus la courbe est saccadée. Un autre hyper-paramètre important à déterminer est l'ordre du polynôme utilisé pour l'interpolation. SAS propose uniquement des polynômes de degré 1 ou 2, mais une implantation personnelle peut utiliser des polynômes de tout ordre. Le choix de la valeur de ces deux hyper-paramètres peut être effectué à l'aide de graphiques illustrant la qualité de l'ajustement.

[8] *Locally weighted scatterplot smoothing*

3.4.5 Mélange de gaussiennes

Le mélange de gaussiennes est généralement un modèle d'estimation de densité non paramétrique local pour des données continues. Ce modèle consiste à appliquer une distribution normale sur chaque point d'entraînement. Par la suite, lorsqu'un nouveau point se présente, il est possible de déterminer la propension de ce point à provenir de la distribution d'entraînement à cause de sa proximité avec les points de la distribution d'entraînement. Par contre, cet algorithme peut être adapté afin d'effectuer une régression plutôt qu'une estimation de densité. Pour transformer cette méthode d'estimation de densité en méthode de régression avec seulement un prédicteur, la procédure est plutôt simple. Pour commencer, il faut appliquer une distribution normale sur chacune des données d'entraînement. Par la suite, pour utiliser le modèle avec un nouveau point, disons r, il faut seulement calculer la contribution de chacun des points d'entraînement à r. Cette contribution est basée sur la distance [9] entre chacun des points d'entraînement et r. Plus le point est proche, plus sa contribution sera importante. En fonction de cette contribution et de la valeur cible de chacun de ces points d'entraînement, il faut utiliser une moyenne pondérée afin de calculer la valeur cible prédite pour ce nouveau point r. Cette méthode présente une très grande capacité. Par contre, avec un gros jeu de données, l'estimation est gourmande au point de vue numérique, la paramétrisation est difficile à effectuer et surtout, cette méthode est très dépendante du jeu d'entraînement. De plus, comme toute méthode non paramétrique locale, une attention particulière doit être portée pour éviter le sur-apprentissage.

3.4.5.1 La mécanique sous-jacente

En pratique, on fait l'hypothèse que chacun des points d'entraînement a une influence sur son environnement qui décroit selon une distribution normale. Par la suite, on fait une moyenne des valeur cibles des points environnants pondérées en fonction de leur distance gaussienne par rapport au nouveau point.

La densité d'une distribution normale a la forme

[9] Distance calculée à partir d'une métrique définie par l'utilisateur

$$f(x) = \frac{1}{2\pi\sqrt{\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad . \tag{3.21}$$

Ainsi, pour trouver la distance gaussienne entre un point d'entraînement et un nouveau point, il faut simplement remplacer $(x-\mu)$ par la distance euclidienne[10] entre les deux points en question. De plus, il faut déterminer la valeur de l'hyper-paramètre σ qui détermine l'étendue de l'influence de chacun des points de la distribution d'entraînement. Plus σ est grand, plus les points d'entraînement ont une influence étendue. Une règle du pouce stipule qu'un bon choix de valeur pour σ dans le cas de points d'entraînement équidistants est la distance entre deux points. Cette valeur représente une bonne valeur de départ pour cet hyper-paramètre qui pourra être optimisé à l'aide d'un ensemble de validation.

En notant par y_i la valeur cible de x_i, le i^e point d'entraînement on obtient donc

$$y^* = \sum_{i=1}^{n} \frac{c_{(i,x)} y_i}{\sum_{j=1}^{n} c_{(j,x)}} \tag{3.22}$$

avec $c_{(k,x)}$, la distance gaussienne entre le point d'entraînement $k(x_k)$ et le point x

$$c_{(k,x)} = \frac{1}{2\pi\sqrt{\sigma^2}} e^{-\frac{||x_k-x||^2}{2\sigma^2}} \quad . \tag{3.23}$$

Cette méthode est très flexible, mais nécessite une grande quantité de mémoire et de temps de calcul pour effectuer une prédiction, spécialement lorsque le jeu d'entraînement est très grand. La méthode sera testée, mais *a priori*, elle présente de grosses lacunes qui lui font perdre beaucoup d'intérêt.

3.4.6 Processus Gaussien ou *G.P.*

La régression à l'aide d'un processus gaussien est un modèle très intéressant pour tenter de modéliser des phénomènes qui ne sont pas nécessairement linéaires. De plus, son grand avantage réside dans sa résistance aux jeux de données bruitées et sa capacité

[10] c'est la norme L_2 qui est utilisée dans notre cas, mais toute autre mesure de similarité pourrait aussi être utilisée

à obtenir des résultats fiables avec peu de données d'entraînement. Cette méthode est intuitivement simple à comprendre. Elle vise à faire passer un processus gaussien par les points d'entraînement. En fait, la méthode suppose que les valeurs cibles suivent une loi multi-normale par rapport aux prédicteurs. Par contre, une telle méthode présente aussi des problèmes, surtout par rapport à son optimisation. En effet, plusieurs hyper-paramètres sont présents dans ce modèle, ce qui rend une optimisation plus difficile à effectuer que dans le cas d'un mélange de gaussiennes, d'un lissage local de nuage de points ou bien d'un ajustement de polynôme. L'optimisation de cette méthode demande beaucoup de ressource de calcul, particulièrement lorsque les exemples d'entraînement sont nombreux et en haute dimension.

3.4.6.1 La mécanique sous-jacente

Étant une méthode non paramétrique locale, la régression à l'aide de processus gaussiens [17, 49, 52, 61] se sert des données d'entraînement afin de pouvoir effectuer une prédiction. Une régression utilisant un processus gaussien avec deux prédicteurs est appelée *krigeage* [11] [41] et est très répandue dans le domaine de la géologie.
Soit :

n le nombre d'observations dans l'ensemble d'entraînement ;

d le nombre de prédicteurs ;

$T = [t_1, ..., t_n]$ les valeurs observées du processus. *i.e.* les valeurs cibles pratiques du processus étant donné que le processus est bruité ;

$Y = [y(\vec{x_1}), ..., y(\vec{x_n})]$ les valeurs cibles théoriques si le processus ne présentait pas de bruit ;

$X = [\vec{x_1}, ..., \vec{x_n}]$ les variables dépendantes des observations du jeu de données ;

$\vec{x_i} = \left[x_{i,1}, ..., x_{i,d}\right]$ les d prédicteurs pour chacune des n observations.

De façon théorique, avant même de connaître l'allure de la distribution d'entraînement (*i.e. a priori*), le modèle se formule de la sorte :

[11] ou *kriging* en anglais

$$T \sim N_n(M,V) \ , \tag{3.24}$$

où

M est le vecteur "moyenne *a priori*" de la variable dépendante des observations et est défini en fonction des variables indépendantes, $M = [\mu(\vec{x_1}),...,\mu(\vec{x_n})]'$ avec $\mu(\vec{x_i})$ une fonction moyenne à définir.

$V = \sigma^2_{GP}K + N$ la matrice de variance-covariance du processus.

La matrice de corrélation entre les valeurs cibles théoriques K est représentée à l'aide d'un noyau gaussien et est formée par $K_{i,j} = K(\vec{x_i},\vec{x_j})$, avec

$corr(y(\vec{x_p}),y(\vec{x_q})) = K(\vec{x_p},\vec{x_q}) = exp\left\{\sum_{l=1}^{d} -\beta_l(x_{p,l} - x_{q,l})^2\right\}$,

$\beta = [\beta_1,...,\beta_d]$ sont les paramètres de corrélation entre les prédicteurs,

N est appelée la matrice "*pépite*"[12] et représente la variance dans T due à l'aspect stochastique de cette variable. C'est littéralement la matrice de bruit. Lorsque la réponse n'est pas stochastique, cette matrice est nulle. Cette matrice est très utile lors de la modélisation sur des données répétées ou agrégées (dans le cas d'un seul prédicteur). Elle sert à tenir compte du bruit contenu dans les données et est représentée comme $N = \sigma^2_{GP}aN_s$ où $N_s = diag(variance(y(\hat{x_i}))), 1 \leq i \leq n$.

σ^2_{GP} est la variance de Y et par maximum de vraisemblance, $\hat{\sigma}^2_{GP} = \frac{1}{n}(T - (\hat{M}|T))'(K + aN_s)^{-1}(T - (\hat{M}|T))$,

a est le paramètre d'échelle du bruit dans le processus.

En prenant connaissance de l'allure de la distribution d'entraînement, le modèle se raffine afin de mieux représenter la distribution d'entraînement.

En effet, en premier lieu, l'allure de la fonction $\mu(x)$ doit être déterminée. Selon le type de problème, elle peut être constante aussi bien que dépendante de l'observation x donnée en entrée de cette fonction et est construite à l'aide de la distribution d'entraînement. Dans le cas constant, $\hat{\mu}(x|T) = \frac{1}{n}\sum_{i=1}^{n} T_i \ \forall \ x$ et signifie que la moyenne *a priori* du processus à modéliser est constante peu importe l'observation. Ce choix est

[12] ou *nugget matrix* en anglais

seulement utile dans des circonstances particulières. Dans notre cas, nous nous attendons à ce que la moyenne *a priori* du processus ne soit pas constante étant donné que les caractéristiques d'un client ont un grand risque d'influencer sa propension à l'annulation de sa police. Dans un tel cas, il est possible d'utiliser un modèle avec moyenne *a priori* dépendante de l'entrée et la fonction $\hat{\mu}(x|T)$ sera plutôt de la forme : $\hat{\mu}(\vec{x_*}|T) = \vec{x_*}' \left(X'V^{-1}X\right)^{-1} X'V^{-1}T + \vec{b}$. Ainsi, $\hat{\mu}(x|T)$ peut être formé à l'aide d'une régression linéaire généralisée sur les variables indépendantes pour prédire la variable dépendante. Ainsi, nous obtenons un processus qui a une moyenne *a priori* qui n'est pas constante et qui reflète plus les données.

En second lieu, les variations inexpliquées par $\hat{\mu}(x|T)$ (les résidus résultant de $\hat{\mu}(x|T)$), sont modélisés par un processus gaussien. De cette façon, la structure dans les résidus est captée et sert à obtenir une prédiction plus précise du comportement de la variable cible. On peut voir cette seconde partie de l'équation (3.25) comme une correction apportée à l'utilisation de $\hat{\mu}(x|T)$.

Ainsi, *a posteriori*, pour effectuer une prédiction sur une nouvelle observation, disons $\vec{x_*}$, nous utiliserons

$$\hat{y}_* = E[y(\vec{x_*})|T] = \hat{\mu}(\vec{x_*}|T) + \hat{\sigma}^2_{GP} k(\vec{x_*}) V^{-1}(T - (\hat{M}|T)) \ , \tag{3.25}$$

avec

$k(\vec{x_*}) = [K(\vec{x_*}, \vec{x_1}), ..., K(\vec{x_*}, \vec{x_n})]$.

L'optimisation d'un tel modèle se fait par maximum de vraisemblance et nécessite l'utilisation de techniques numériques. Dans le package "mlegp" [17] en R, l'optimisation se fait avec une méthode du simplexe [18] et une méthode BFGS[13] [4]. La méthode maximisant le mieux la vraisemblance est retenue pour le modèle final.

Une telle méthode demande beaucoup de temps de calcul pour un jeu de données d'une grande envergure. Des approximations peuvent être utilisées dans le modèle afin d'en faciliter l'optimisation. Par exemple, la librairie "*kernlab*" [35] en R offre un modèle

[13] Broyden-Fletcher-Goldfarb-Shanno

beaucoup plus rapide, mais chacun des prédicteurs est pris avec une importance égale dans le modèle. Cette approximation rend le modèle beaucoup plus rapide, mais enlève beaucoup de précision en contrepartie. Un tel modèle simplifié n'est pas utilisable dans le contexte de l'assurance, car l'approximation effectuée est loin d'être valide.

3.5 L'approche utilisée

Les modèles précédemment discutés ne répondent pas exactement aux besoins de la compagnie. Afin d'obtenir un modèle présentant de meilleures caractéristiques, il est possible d'utiliser une architecture formée de plus d'un modèle.

Comme discuté précédemment, le modèle choisi doit être en mesure d'obtenir une bonne précision lors de l'estimation du taux de non renouvellement sur différent segments du porte-feuille de polices d'assurance afin de pouvoir obtenir une idée de la composition de ce porte-feuille après un changement de taux donné. Un modèle uniquement entraîné au niveau global risque de manquer de précision sur certains segments ne présentant pas beaucoup d'observations. Cette situation est due à l'optimisation du modèle qui favorise une bonne performance globale plutôt qu'une bonne performance sur certain segments précis afin de tenter d'éviter le sur-apprentissage. Afin d'avoir un modèle plus précis, une approche plus locale est proposée. Ainsi, le modèle envisagé propose de commencer par effectuer une segmentation pour ensuite entraîner un modèle dans chacun des regroupements effectués par la segmentation initiale. Ainsi, si la segmentation initiale sépare le jeu de données d'entraînement en k segments, il faudra par la suite entraîner k modèles distincts. De cette façon, lors de la prédiction de la valeur cible d'une nouvelle observation, le modèle utilisé sera plus précis. Par contre, avec une telle architecture, il faut porter une attention très spéciale au sur-apprentissage, particulièrement lorsque le modèle est doté d'une grande capacité. En effet, une grande capacité couplée à un jeu de données plus restreint pourrait avoir tendance au sur-entraînement. Plus k augmente, plus le risque de sur-apprentissage est grand. Par contre, un k trop petit produira un modèle manquant de précision. Des tests devront être effectués à ce sujet afin de déterminer une valeur acceptable de segments.

Le but premier étant de prédire le non renouvellement en fonction d'une nouvelle tarification, il faut s'assurer que le changement de prix ait une place de prédilection dans le nouveau modèle proposé. De plus, le modèle doit impérativement présenter une dérivée non nulle par rapport à cette même variable. Dans cette optique, les architectures proposées se présentent en deux étapes. La première étape consiste à produire une segmentation en utilisant toutes les variables, sauf celles ayant rapport au changement de prix. Par la suite, dans chacun des segments créé, un modèle prédisant le non renouvellement est entraîné. Deux différentes voies pour les modèles prédisant le non renouvellement sont proposées. La première consiste à prédire le non renouvellement dans chacun des segments en utilisant uniquement l'augmentation de prime. La seconde consiste à utiliser toutes les variables pertinentes pour prédire le non renouvellement dans chacun des segments créés.

Utiliser seulement l'augmentation de prime dans le modèle a pour effet de simplifier grandement le modèle tout en gardant la variable la plus importante au niveau de l'utilisation future. De plus, le sur-entraînement ne peut pratiquement pas survenir dans un tel contexte. En contre-partie, un modèle prenant en compte toutes les variables pertinentes a tendance à être plus précis, mais nécessite une grande attention afin de ne pas sur-apprendre la distribution d'entraînement.

3.5.1 Segmentation

Plusieurs méthodes sont possibles pour la segmentation. Une segmentation [14] non supervisée peut être effectuée, mais un arbre de classification et régression pourrait aussi être utilisé.

La segmentation non supervisée serait intéressante pour essayer d'avoir des observations "proches les unes des autres" selon une métrique définie par l'utilisateur. Par contre, une telle approche demande de faire une hypothèse importante sur le comportement de la distribution par rapport au non renouvellement. En effet, nous souhaitons avoir une segmentation qui produirait des groupements homogènes qui auraient de très grandes différences entre eux quant au taux de non renouvellement. De plus, la cohérence des

[14] *clustering*

segments au point de vue des affaires est nécessaire parce que le modèle final sera différent pour chacun des segments. Si les segments sont construit sans trop de cohérence, les différents modèles se basant sur cette segmentation seront porté à sur-apprendre plutôt qu'à se spécialiser sur des segments pertinents. Il faut donc s'assurer d'avoir une segmentation la meilleure possible. Pour obtenir une segmentation cohérente et efficace à l'aide d'une segmentation non supervisée, il faudrait faire des hypothèses sur ce qui produit un changement de comportement chez le client. C'est justement pour éviter de devoir faire de telles hypothèses que ce travail de modélisation est nécessaire. L'intérêt du modèle deviendrait nul si une telle hypothèse devait être posée.

Pour pallier à ce problème, un arbre de classification et régression pourrait être utilisé afin de produire les segments voulus. En effet, un tel arbre aurait comme critère principal de regrouper ensemble les gens prompt à ne pas renouveler. Pour produire les embranchements, l'arbre choisirait les variables les plus significatives (les variables par rapport au changement de la prime ne seraient pas données à l'arbre). Nous aurions alors des segments ayant de fortes différences dans leur taux de non renouvellement de par la nature de l'arbre. De plus, les segments devraient normalement être cohérents. Aussi, dans le cas d'un problème de cohérence dans l'arbre, un ajustement manuel pourrait toujours être effectué pour corriger la situation. L'arbre C&R sera utilisé pour cette tâche. Une implantation efficace de cet algorithme est disponible dans PASW-SPSS. Un hyper-paramètre important reste tout de même à déterminer. En effet, le nombre de segments produits par l'arbre doit être choisi par l'utilisateur. Plusieurs essais devront être fait afin de mieux comprendre l'effet de cet hyper-paramètre sur la performance du modèle final.

De plus, nos données présentent un certain débalancement de classes. En effet, il est loin d'y avoir 50% de non renouvellement. À cet égard, un coût de mauvaise classification différent pour les deux types d'erreurs doit être défini afin que l'arbre ne soit pas tenté de simplement faire des sous-groupes sans signification et étiqueter le tout comme étant des renouvellements. Afin de rendre l'arbre de décision sensible à la classe minoritaire, un coût de mauvaise classification élevé sera donné lorsqu'un non renouvellement est étiqueté comme étant un renouvellement par le modèle. Par contre, une différence trop importante dans le coût de mauvaise classification aurait pour effet de rendre le

modèle trop sensible aux non renouvellements. Le coût de mauvaise classification a été défini ainsi :

$$\begin{aligned} C(non_r|r) &= 1 \\ C(r|non_r) &= \frac{x}{2} \\ x &= \frac{nb_{ren}}{nb_{non_ren}} \end{aligned} \tag{3.26}$$

avec

$C(non_r|r)$ le coût de classer comme étant un non renouvellement une observation qui est en fait un renouvellement

$C(r|non_r)$ le coût de classer comme étant un renouvellement une observation qui est en fait un non renouvellement

nb_{ren} le nombre d'observations dans le jeu d'entraînement qui sont des renouvellements

nb_{non_ren} le nombre d'observations dans le jeu d'entraînement qui sont des non renouvellements

De brefs essais à l'aide d'un ensemble de validation ont montré la supériorité des arbres de décision utilisant un coût de mauvaise classification différent plutôt qu'uniformes. La performance des arbres utilisant la structure de coût présentée est un peu supérieure à celle obtenue avec un arbre utilisant $C(r|non_r) = \frac{nb_{ren}}{nb_{non_ren}}$, mais la différence est très minime. Pour obtenir des résultats plus convaincants, des essais supplémentaires pourraient être fait. Étant donné que cet hyper-paramètre ne représente pas un point très sensible dans cette recherche de par le fait que les arbres produits sont toujours vérifiés visuellement avec la possibilité de changer certaines branches si nécessaire, l'optimisation de cet hyper-paramètre est laissée pour de futur travaux.

En utilisant un coût de mauvaise classification dépendant de la classe de l'observation, le modèle va considérer les observations de non renouvellement comme ayant une

importance plus élevée que celles menant à un renouvellement. Une importance égale pour les observations provenant des deux groupes aurait été discutable de par le fait que l'arbre aurait créé des noeuds finaux moins discriminants. Il nous importe d'être capable de discerner les observations à risque de ne pas renouveler, mais un arbre tout de même conservateur reste important car la distribution d'entraînement originale présente un débalancement que l'arbre doit connaître pour effectuer une bonne classification. Pour ces raisons, le coût de mauvaise classification est dépendant de la classe du point en question et est défini par les équations (3.26) .

3.5.2 Modélisation locale

Pour la modélisation locale sur chacun des segments créés par l'arbre de décision, deux avenues sont envisagées. La première étape est d'essayer les différents types de modèles retenus avec seulement l'augmentation de prime comme prédicteur. Une fois que la performance des différents modèles est évaluée lors de l'utilisation d'un seul prédicteur, seul les meilleurs modèles seront retenus pour la tentative avec tous les prédicteurs pertinents.

3.5.2.1 Un seul prédicteur

Une fois les segments connus, un modèle semblable doit être bâti sur chacun d'eux pour tenter de prédire le non renouvellement en fonction du changement de prix de la police d'assurance. Une première sélection de modèles a été faite : régression logistique (section 3.4.2), ajustement polynomial (section 3.4.3), lissage local d'un nuage de points (section 3.4.4), mélange de gaussiennes (section 3.4.5) ainsi que régression avec processus gaussien (section 3.4.6). Nous disposons par contre de données sous forme de triplets : (# segment, non renouvellement, changement de prime). Étant donné que certains modèles sont très gourmands côté calculs (en particulier le processus gaussien) une certaine agrégation des données est effectuée lorsque nécessaire. En effet, les données sont placées en ordre croissant de changement de prime. Par la suite, trente (nombre purement arbitraire qui pourrait éventuellement être changé) regroupements de largeurs

égales (par opposition à regroupements contenant tous le même nombre d'observations) sont produits. Ainsi, les modèles autres que la régression logistique (qui prend uniquement des données à cible binaire) sont maintenant entraînés avec seulement 30 observations, ce qui rend possible l'utilisation d'un processus gaussien. De plus, une telle agrégation des données aide à réduire le bruit présent, ce qui aide aussi les modèles à offrir une meilleure performance.

Enfin, les modèles utilisés produisent des prédictions du taux de non renouvellement pour chaque nouvelle observation. Normalement, la régression logistique produit des réponses binaires, mais il est aussi possible d'obtenir la probabilité de non renouvellement pour chacune des nouvelles observations. C'est avec cette probabilité qu'est calculée la prédiction binaire généralement donnée par ce modèle. Ce sont seulement les probabilités de non renouvellement qui seront utilisées pour définir la performance du modèle. En effet, ces mesures sont sur la même échelle pour tous les modèles (autrement, la régression logistique est à part) et sont plus utiles pour calculer la probabilité moyenne de non renouvellement pour n'importe quel groupe.

3.5.2.2 Plusieurs prédicteurs

La dimensionnalité du jeu de données étant très grande, ce ne sont pas toutes les méthodes énumérées dans la sous-section précédente qui sont capables de prendre en compte un tel jeu de données. Par exemple, le GP a une limitation sur le nombre de données présentes dans le jeu d'entraînement étant donné que la méthode est de l'ordre de n^3. Ainsi, passé 10,000 observations, la méthode ne s'applique plus sur les meilleurs ordinateurs personnels présentement sur le marché. Le jeu de données à analyser contient plus de 50,000 exemples d'entraînement. De plus, il n'existe pas vraiment de version "classification" de l'algorithme d'ajustement polynomial ou du lissage local de nuage de points n'ayant pas de grave problème si les classes sont débalancées. Sachant ces importantes limitations, un cadre innovateur est proposé afin de tirer le maximum des méthodes de régression que nous avons tout en obtenant un résultat fiable et utile pour l'entreprise. L'idée derrière le modèle est de traiter indépendamment tous les prédicteurs et par la suite unifier tous les petits modèles à un seul prédicteur pour en former un plus

gros qui sera beaucoup plus précis. Ainsi, dans un premier temps, les prédicteurs discrets et continus seront traités différemment.

Pour chacun des prédicteurs discrets, un simple calcul du taux de non renouvellement par catégorie sera effectué sur l'ensemble d'entraînement. De cette façon, on obtient de façon univariée l'effet de chacun des prédicteurs discrets.

Pour chacun des prédicteurs continus, un modèle semblable à celui décrit dans la section 3.5.2.1 sera utilisé. Par contre, le nombre de regroupements effectués dépend de la variable qui est traitée. Il faut toujours garder en tête que chacun des regroupements doit tout de même être crédible et ne pas contenir uniquement du bruit.

Une fois que tous les modèles univariés sont construits, simplement les appliquer sur les ensembles d'entraînement et de test. Il en résulte un jeu de données contenant un pointage pour chacune des variables présentes.

Enfin, reste à construire une régression logistique sur l'ensemble d'entraînement en utilisant comme cible la variable de non renouvellement et comme prédicteurs les pointages obtenus par les précédents modèles univariés (dans le cas de prédicteurs continus) et par les tables de probabilités (dans le cas de prédicteurs discrets). La régression logistique a la faculté de produire une probabilité plutôt qu'une simple classification binaire en utilisant la valeur donnée à la fonction sigmoïde. De cette façon, une distinction plus grande des différentes observations peut se faire en fonction de la propension à l'attrition.

Finalement, il reste à utiliser ce modèle final afin de calculer la probabilité d'attrition de chacune des observations de l'ensemble de test.

3.6 Métrique de performance

Afin de déterminer lequel des modèles proposés est le meilleur pour effectuer la tâche demandée, il faut développer une méthode objective pour qualifier quantitativement chacune des méthodes essayées. Le modèle n'ayant pas une structure habituelle et les besoins étant très clairement définis, une mesure de performance personnalisée serait souhaitable et facile à développer.

Comme le modèle final contient des sous-modèles entraînés sur différent groupes, la mesure de performance doit en tenir compte. De plus, il faut que le modèle soit bon sur chacun des sous-groupes, mais en évitant de se spécialiser sur un seul des sous-groupes, spécialement s'il est petit. Ainsi, pour répondre à ce besoin, une métrique de performance nouvelle a été développée. C'est une sorte de méthode des moindres carrés pondérés adaptée à la situation définie par

$$metrique = \left(\sum_{i=1}^{s} \frac{w_i}{w} (\hat{c}_i - c_i)^2 \right) \tag{3.27}$$

où

s le nombre de sous-groupes sur lesquels le modèle est testé

w_i le nombre d'observations dans le sous-groupe i

w le nombre total d'observations dans l'ensemble d'entraînement

$\hat{c}_i$ le taux prédit de non renouvellement dans le sous-groupe i

c_i le taux avéré de non renouvellement dans le sous-groupe i.

Cette mesure pénalise beaucoup les grands écarts entre le taux prédit et le taux effectif de non renouvellement pour chacun des groupes de par l'utilisation du carré de la différence. Il est préférable pour le modèle de faire des erreurs moyennes pour chacun des sous-groupes plutôt que certain sous-groupes parfaits et certains autres avec de très grandes erreurs. De plus la pondération des carrés des erreurs est utile pour spécifier au modèle que les plus grands sous-groupes ont une plus grande importance. Cette mesure de performance est bien adaptée au modèle dans le sens qu'elle quantifie bien les objectifs auxquels le modèle doit répondre. Les sous-groupes peuvent être ceux sur lesquels le modèle est entraîné aussi bien que des sous-groupes pour lesquels nous voulons tester le modèle.

De par sa construction, cette mesure de performance amalgame toutes les observations à l'intérieur d'un sous-groupe. Elle calcule ainsi la performance du modèle sur différents sous-groupes. Le but premier du modèle étant de prédire l'allure du portefeuille par rapport à certaines segmentations en fonction de l'incrément de prime au

renouvellement, cette mesure de performance est toute adaptée aux besoins.

3.7 Différentes distributions de test

En gardant en tête que l'outil à développer doit servir à pouvoir analyser des scénarios habituels autant que des scénarios présentant des grands changements de taux rarement vus de par le passé, une méthode d'évaluation de cette capacité doit être effectuée. En effet, un modèle très efficace sur la distribution originale, mais qui est incapable de bien prédire sur une distribution présentant des grand changements de taux ne serait d'aucun intérêt pour les futurs utilisateurs de cet outil. Pour évaluer cette capacité, la distribution de test peut être modifiée (par contre, la distribution d'entraînement doit toujours rester originale) dans le but d'y changer la variation moyenne de taux. À cette fin, une méthode approximative peut être utilisée afin de choisir le changement moyen de taux à l'intérieur de la distribution de test. La méthode développée est approximative, mais la précision n'est pas obligatoire pour ce genre de test. Il faut simplement avoir des résultats reproductibles afin de faire une comparaison éclairée des différents modèles. Le sous-échantillonnage sera la clef du problème. La méthode est simple et consiste à choisir toutes les observations au-dessus/en-dessous (dépendamment de si le changement de taux moyen voulu est au-dessus ou en-dessous de celui de la distribution originale) du changement voulu. Par la suite, simplement choisir aléatoirement le même nombre d'observations à partir des observations restantes de façon à obtenir un nouvel ensemble de test qui possède autant d'observations au-dessus qu'en-dessous du changement de taux voulu. Cet ensemble possèdera approximativement le bon changement de taux moyen. Une grande précision n'est pas nécessaire pour cet exercice.

CHAPITRE 4

RESULTATS

Un hyper-paramètre important des modèles expliqués plus haut est le nombre de feuilles finales dans les arbres de décision initiaux. Cet hyper-paramètre définit combien de sous-groupes seront utilisés par le second modèle. Un trop petit nombre n'apportera pas de précision supplémentaire au modèle par rapport à l'utilisation d'un modèle global, tandis qu'un trop grand nombre poussera les modèles locaux à apprendre des informations non pertinentes qui nuiront à la capacité de généralisation du modèle.

4.1 Un seul prédicteur utilisé : l'incrément de prime

La première étape de l'expérimentation consiste à construire des modèles n'utilisant qu'un seul prédicteur. Ces modèles sont faciles à créer et ont une capacité moindre, ce qui fait qu'ils sont bien adaptés à une utilisation sur des sous-ensembles de la distribution d'entraînement.

4.1.1 Une feuille finale ou 4 feuilles finales ?

La première expérience consiste à comparer la performance des modèles avec quatre feuilles finales et des modèles à une seule feuille finale utilisant seulement l'incrément de prime comme prédicteur. En effet, une seule feuille finale dans l'arbre de décision revient à modéliser sur tout l'ensemble d'entraînement, mais seulement avec un seul prédicteur dans notre cas. Cette comparaison va être utile pour évaluer la pertinence d'effectuer des modèles locaux plutôt qu'un seul modèle global et surtout la pertinence d'utiliser seulement un seul prédicteur.

4.1.1.1 Exploration de l'allure des distributions à modéliser

Une fois que l'arbre de classification et régression a partitionné l'ensemble d'entraînement, il est possible de regarder les données afin de voir l'allure de la distribution dans

chacune des feuilles. Une telle opération est très utile pour mieux comprendre les données afin de pouvoir effectuer la modélisation plus facilement et mieux comprendre le comportement des différents modèles. Sur les graphiques, la courbe rouge représente le taux de non renouvellement de chacune des 30 bandes dans lesquelles l'ensemble d'entraînement a été divisé par rapport à l'incrément absolu de la prime. La courbe bleue représente quant à elle le nombre d'observations dans chacune des bandes. Il est à noter que les quantités ont été retirées des axes par souci de confidentialité.

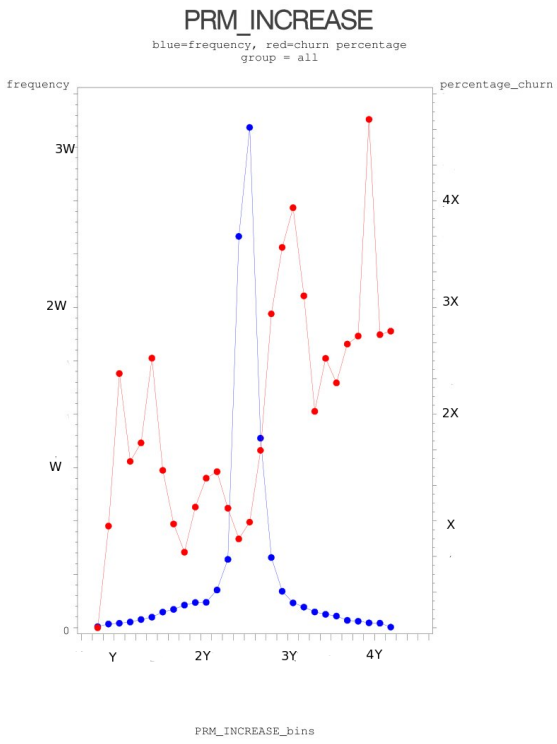

Figure 4.1 – Distribution d'entraînement originale

Sur la figure 4.1, il est possible de voir que l'occurrence des changement de prime (bleu) ressemble à une distribution normale, en un peu plus concentrée. Par contre, le taux de non renouvellement (rouge) ne se comporte pas de façon linéaire par rapport au changement de prime. Il faut par contre tenir en compte que la distribution est plutôt bruitée lorsque les changements de taux sont plus extrêmes. De la même manière, il est aussi possible de faire une inspection visuelle des distributions des sous-groupes formés par l'arbre de segmentation à quatre feuilles finales à l'aide de la figure 4.2.

Sur cette seconde figure, il est possible de voir que les distributions du non renouvellement dans les différents sous-groupes sont assez différentes de la distribution contenant toutes les observations. La normalité des changements de prime est préservée dans cha-

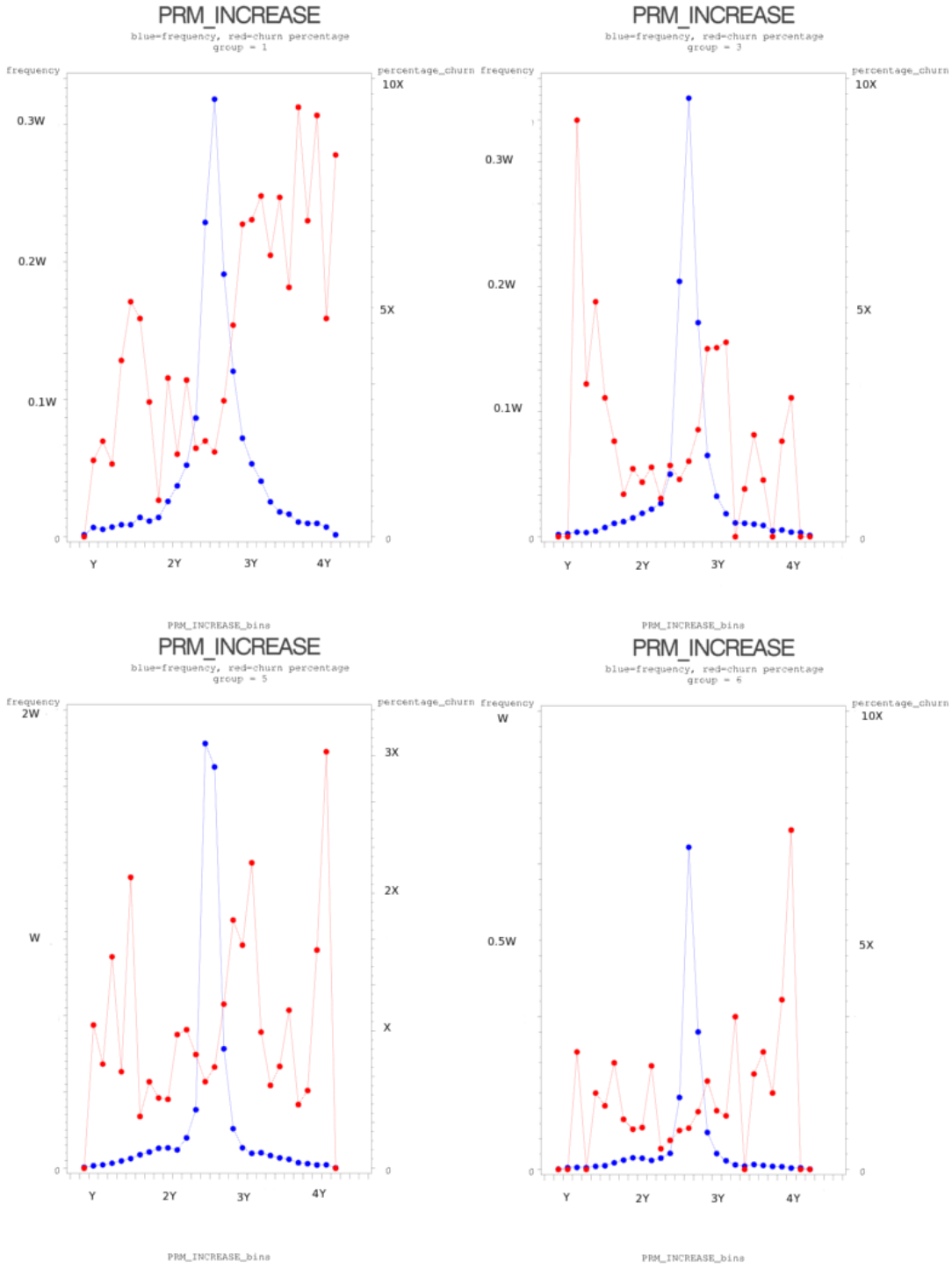

Figure 4.2 – Distribution d'entraînement des sous-groupes

cun des sous-groupes. Par contre, le comportement des gens par rapport au changement de prime est très différent dans chacun des sous-groupes. De plus, les sous-groupes sont de grandeurs assez différentes.

4.1.1.2 Ajustement polynomial

Lors de l'ajustement polynomial, il faut définir le degré du polynôme qui passera par les points. Cet hyper-paramètre est très important, il définit comment se comportera la courbe prédictrice. En regardant la distribution d'entraînement originale, on voit que la courbe présente 4 changements de convexité, ce qui indiquerait qu'un polynôme de degré 5 au maximum pourrait être intéressant. Par contre, ajuster un polynôme de ce degré implique de retrouver ces 4 changements de convexité dans la solution, ce qui n'est pas nécessairement souhaitable. En effet, il pourrait sûrement être intéressant de tenter de lisser un peu plus la courbe étant donné que certain changements de convexité sont peut-être simplement dus à du bruit. De plus, lors de l'utilisation de l'arbre pour effectuer la segmentation, les distributions obtenues ont toujours moins de 4 changements de convexité. À cet effet, des polynômes de degré 2, 3, 4 ainsi que 5 seront testés sur les données.

4.1.1.3 Lissage local d'un nuage de points

Comme mentionné plus haut, deux hyper-paramètres importants sont impliqués dans la mise en marche d'un lissage local d'un nuage de points. Ces hyper-paramètres doivent être optimisés à l'aide d'un ensemble de validation. De plus, les hyper-paramètres sont forcés d'avoir la même valeur dans chacun des segments afin de se prémunir contre un certain sur-apprentissage caractéristique des méthodes non paramétriques. La variation dans les solutions produites par chacun des segments réside simplement dans la composition de la distribution d'entraînement. Différentes valeurs pour les hyper-paramètres sont essayées afin de tenter d'utiliser le meilleur modèle (degré du polynôme ajusté : 1 et 2, λ=0,2 , 0,3 , 0,4 , 0,5). De plus, il est possible de mettre des poids différents pour chacun des points afin qu'ils aient des importances différentes lors du lissage. À cette fin,

degré poly.	λ	nb. feuilles	dist. orig.	inc=500	inc=750	inc=1000	inc=-500
1	0.2	1	0.10	0.003	0.07	0.07	0.20
		4	0.08	0.46	1.60	1.20	1.72
	0.3	1	0.46	0.049	0.25	0.29	0.073
		4	0.24	0.49	1.78	1.21	1.59
	0.4	1	0.76	0.096	0.41	0.56	0.029
		4	0.38	0.53	1.93	1.24	1.59
	0.5	1	1.27	0.15	0.61	0.93	0.0006
		4	0.66	0.58	2.09	1.34	1.66
2	0.2	1	0.019	0.0002	0.027	0.032	0.27
		4	0.04	0.46	1.54	1.07	1.82
	0.3	1	0.07	0.001	0.046	0.033	0.22
		4	0.061	0.46	1.57	1.19	1.73
	0.4	1	0.20	0.02	0.15	0.16	0.03
		4	0.12	0.44	1.60	1.11	1.43
	0.5	1	0.40	0.048	2.78	0.26	0.09
		4	0.19	0.49	12.01	1.17	1.55

Tableau 4.I – Comparaison de la performance sur l'ensemble de validation des modèles de lissage local de nuage de points

le nombre d'observations présentes dans chacun des points regroupés sera le paramètre de poids. De cette manière, le lissage sera bien effectué globalement, ce qui apporte une certaine protection contre le sur-apprentissage.

En regardant le tableau 4.I de la mesure de performance pour les différentes configurations d'hyper-paramètres sur différentes façons de partitionner l'ensemble de validation, il est possible de voir que tous les choix d'hyper-paramètres ne procurent pas la même qualité d'ajustement. Dans le tableau, les cases vertes représentent les meilleures performances pour le modèle à quatre feuilles finales, tandis que les cases en jaunes représentent les meilleures performances pour le modèle à une seule feuille finale. On peut y voir la supériorité générale des polynômes de second degré. En regard des performances obtenues sur l'ensemble de validation, le choix retenu est donc un lissage à l'aide de polynômes de degré 2 avec $\lambda = 0.2$. Pour une meilleure compréhension de l'effet des hyper-paramètres, les différentes configurations d'hyper-paramètres sur l'ensemble d'entraînement sont illustrées dans le graphique 4.3.

4.1.1.4 Régression logistique

Dans le cas de la régression logistique, un modèle différent pour chacun des segments a été ajusté au lieu d'un même modèle pour tous les segments où les seules différences proviennent des distributions d'entraînement distinctes. Ce choix est dû au fait que ce type de modèle a une faible capacité. Le sur-apprentissage serait donc très improbable lorsqu'un seul prédicteur est utilisé.

4.1.1.5 Mélange de gaussiennes

Dans le cas du mélange de gaussiennes, le choix de l'hyper-paramètre σ est très important. Cet hyper-paramètre détermine le rayon d'influence de chacun des points d'entraînement. Pour ce modèle, il a été choisi d'utiliser la même valeur de σ pour les modèles spécifiques à chacun des segments (le modèle global possède par contre sa valeur distincte de σ) . Ce choix de forcer l'hyper-paramètre σ à avoir la même valeur pour les modèles dans chacun des segments est naturel. En effet, σ représente le rayon

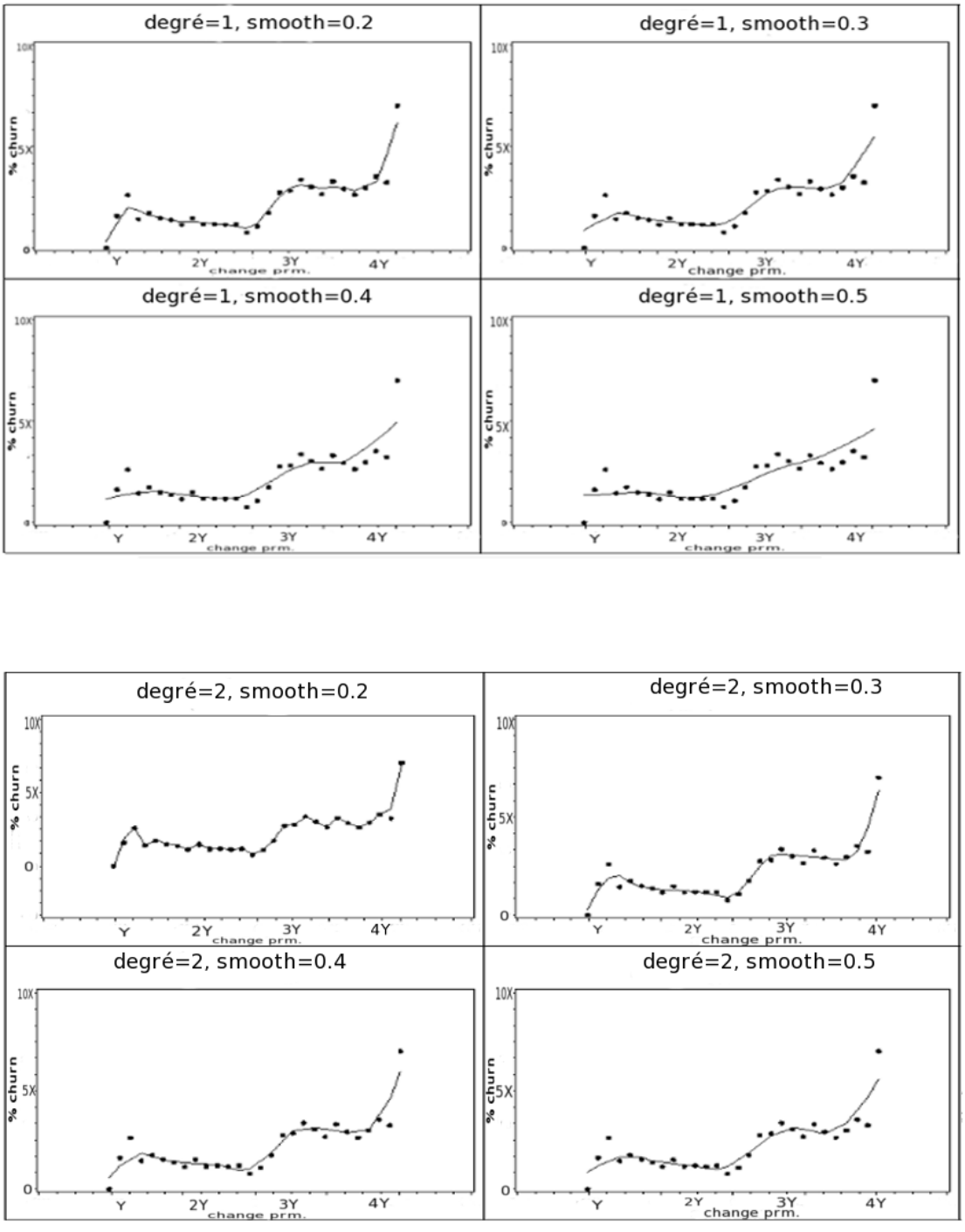

Figure 4.3 – Effet des hyper-paramètres de LOESS sur l'ensemble d'entraînement

d'influence de chacun des points. Il serait normal que ce rayon d'influence soit le même dans chacun des segments. De plus, le fait de forcer ce choix aide à diminuer les effets négatifs dûs au fait que ce modèle est de type non paramétrique local. La capacité de généralisation est augmentée par ce choix.

Pour effectuer l'optimisation de cet hyper-paramètre, l'utilisation d'un ensemble de validation est nécessaire. En effet, il faut simplement optimiser la métrique de performance (section 3.6) sur cet ensemble de validation en fonction de σ. De cette façon, l'hyper-paramètre prend une valeur qui fait en sorte que la performance de chacun des modèles est acceptable et qu'au plus il y a d'observations dans le segment, au plus σ est choisi de manière à bien prédire les observations présentes dans ce segment.

4.1.1.6 Processus gaussien

Tout comme dans le cas du mélange de gaussiennes, les hyper-paramètres seront forcés de prendre la même valeur pour chacun des modèles sur des segments différents. La capacité de généralisation du modèle en sera grandie en éloignant la possibilité de surapprentissage. Ce modèle simple sera construit à l'aide du logiciel Excel, car il fait partie des outils couramment utilisés dans une compagnie d'assurance. De plus, les modèles construits sur les différents segments ne sont pas d'une grande complexité de par le peu d'observations utilisés (30 observations créées par agrégation) et de la présence d'un seul prédicteur, ce qui fait qu'ils peuvent être construit à l'aide d'un logiciel déployant une faible capacité de calcul.

Lors de l'optimisation des hyper-paramètres, il faut s'assurer que l'ajustement soit correct tout en ayant du sens. L'utilisation d'un ensemble de validation est nécessaire pour s'assurer que l'ajustement se fasse tout en gardant une bonne capacité de généralisation. De plus, comme seulement un prédicteur est présent, il est possible de vérifier graphiquement l'ajustement. Une telle vérification nous assure un contrôle efficace du processus d'optimisation.

4.1.1.7 Tableaux des résultats

Le tableau 4.II compare la performance de chacun des modèles pour des distributions de test présentant différentes augmentations moyennes de prime. Les différentes distribution testées sont la distribution de test originale (dist. orig), un sous-ensemble de la distribution de test originale présentant un incrément de prime moyen de 500$ (inc=500), de 750$ (inc=750), de 1000$ (inc=1000) ainsi qu'un sous-ensemble de la distribution de test originale présentant une baisse moyenne de prime de 500$ (inc=-500). De plus, la comparaison entre l'utilisation d'un modèle global à un seul prédicteur et d'un modèle bâti sur différent segments peut ici être faite. Les mesures présentes dans le tableau sont effectuées grâce à la métrique développée spécifiquement pour ce type de modèles (section 3.6). Le tableau présente la performance obtenue par chaque modèle en entraînant sur l'ensemble d'entraînement complet ainsi que sur les quatre feuilles finales. Dans le cas des modèles entraînés sur une seule feuille finale, la performance sur un ensemble de test complet et la performance sur un ensemble de test divisé selon les quatre groupes formés par l'arbre de segmentation initial ont été calculées. Dans le cas des modèles entraînés sur quatre segments, la mesure de performance a aussi été calculée sur quatre groupes. Ce choix a été fait, car les modèles sont développés dans le but d'offrir une bonne performance sur différentes segmentations de l'ensemble de test (qui pourront être définies par l'utilisateur). De plus, si la performance obtenue est bonne sur quatre segments distincts de l'ensemble de test, elle l'est nécessairement aussi pour le jeu de données dans son ensemble, mais l'inverse n'est pas vrai.

Dans le tableau 4.II le premier constat est que l'utilisation d'un modèle local pour chacun des quatre segments est de loin préférable à l'utilisation d'un modèle global lorsqu'un seul prédicteur est utilisé. En effet, le modèle global offre une très bonne performance lorsque testé sur l'ensemble de test globalement, mais une piètre performance lorsqu'on lui demande d'être bon sur chacun des quatre segments construits à l'aide de l'arbre de segmentation. Cette situation démontre bien l'inutilité d'un tel modèle étant donné que le but premier est d'être capable de prédire sur des séparations du portefeuille qui sont inconnues lors de l'entraînement. Le second constat est que le processus

Modèle	nb. seg.	dist. orig.	inc=500	inc=750	inc=1000	inc=-500
Polynôme 3	1	1.37	0.01	1.32	2.96	0.31
	1 (test sur 4)	8.01	28.67	27.82	29.96	6.85
	4	1.31	0.35	1.42	3.07	1.09
Polynôme 4	1	1.39	0.02	1.32	2.96	0.31
	1 (test sur 4)	8.07	28.71	27.86	30.05	6.85
	4	0.68	0.26	1.46	3.25	0.86
Polynôme 5	1	1.85	0.29	1.59	2.02	0.14
	1 (test sur 4)	8.22	27.65	26.41	29.45	6.58
	4	1.13	0.50	2.24	2.32	0.82
LOWESS	1	0.04	0.02	0.58	0.90	0.0003
	1 (test sur 4)	5.53	25.00	24.03	26.63	5.89
	4	0.23	0.32	1.79	2.28	0.67
Reg. logistique	1	0.00	0.81	0.01	0.41	1.10
	1 (test sur 4)	6.59	29.09	25.46	27.53	7.59
	4	0.16	0.92	0.79	1.10	1.40
Mélange gauss.	1	1.92	0.56	1.97	2.57	0.33
	1 (test sur 4)	7.77	26.93	26.27	29.47	6.35
	4	1.59	1.28	4.03	3.50	0.80
G.P.	1	0.35	0.64	0.04	0.01	0.54
	1 (test sur 4)	5.80	25.77	23.60	25.97	6.30
	4	0.05	0.65	0.27	0.42	0.57

Tableau 4.II – Comparaison de la performance sur l'ensemble de test des modèles pour un seul et quatre segments distincts

gaussien est le modèle local ayant généralement la meilleure performance. Seule la régression logistique obtient une performance tout de même acceptable en regard au temps nécessaire à la construction du modèle (ce qui est un avantage non négligeable) et mérite d'être considéré dans de futur tests.

Pour tester la robustesse des très bon résultats obtenus par le processus gaussien et la régression logistique sur quatre segments sur les ensembles de test présentant des changements de prime différents, cinq différentes façons de séparer l'ensemble total en partition d'entraînement et de test seront testées. Ils seront simplement générés à l'aide d'une graine [1] différente dans le générateur de nombre aléatoire. De cette façon, la bonne performance sera validée de façon plus robuste.

Modèle	dist. originale	inc=500	inc=750	inc=1000	inc=-500
G.P. #1	0.05	0.65	0.27	0.42	0.57
G.P. #2		0.92	0.29	0.50	0.48
G.P. #3		0.64	0.05	0.63	0.33
G.P. #4		0.81	0.89	0.49	0.24
G.P. #5		0.88	0.35	0.63	0.27
G.P. moyenne	0.05	0.78	0.37	0.53	0.38
Reg. logistique. #1	0.16	0.92	0.79	1.08	1.40
Reg. logistique. #2		1.38	1.24	1.19	1.04
Reg. logistique. #3		0.75	0.33	1.68	1.16
Reg. logistique. #4		1.03	1.46	1.17	0.31
Reg. logistique. #5		1.14	0.91	1.73	0.94
Reg. logistique. moyenne	0.16	1.04	0.95	1.37	0.97

Tableau 4.III – Performance des meilleurs modèles 4 feuilles finales sur différent ensemble de test

Dans le tableau 4.III, on peut voir qu'encore une fois, le processus gaussien est gran-

[1] *seed*

dement supérieur à la régression logistique avec quatre feuilles finales et un seul prédicteur. Cette supériorité des processus gaussiens pourrait être attribuables à la non linéarité de la prédiction prédite à l'aide de l'incrément de prime. En effet, en regardant l'allure de la distribution d'entraînement sur les différents segments (voir la figure 4.2), nous pouvons voir la non linéarité de la relation entre l'incrément de prime demandée à l'assuré et le taux d'annulation. Les processus gaussiens étant capables de modéliser des relations non linéaires, ils bénéficient d'un avantage marqué par rapport aux régressions logistiques.

4.1.2 Avec plus de quatre feuilles finales

Dans la sous-section précédente, les meilleurs modèles étaient le processus gaussien suivi de la régression logistique. De plus, la supériorité des modèles spécifiques à des segments sur les modèles globaux a été montré de façon claire. Par contre, le choix du nombre de sous-groupes à produire lors de la segmentation initiale reste à faire. Il a été montré qu'il est préférable d'avoir quatre segments plutôt qu'un seul, mais peut-être serait-il préférable d'en utiliser plus de quatre. Ainsi des segmentations à 9 et 15 feuilles finales ont été utilisées en partenariat avec le processus gaussien et la régression logistique. Les autres types de modèles ayant antérieurement démontré une piètre performance, seul les deux meilleurs modèles (régression logistique et processus gaussiens) ont été retenus pour les tests futurs.

4.1.2.1 Tableau des résultats

Pour bien comparer la performance des modèles lorsque le raffinement de la segmentation change, certain résultats présentés plus haut seront reportés dans le tableau 4.IV.

Dans ce tableau, il est possible de voir la supériorité des modèles à un seul et quatre feuilles finales lorsque le modèle est testé sur un ensemble de test ayant la même structure que l'ensemble d'entraînement, surtout lorsque l'incrément de prime moyen est changé par rapport à la distribution originale. Cette situation démontre un certain sur-

# segments	Régression	dist. orig.	inc=500	inc=750	inc=1000	inc=-500
1	G.P.	0.35	0.64	0.04	0.01	0.54
	logistique	0.00	0.81	0.01	0.41	1.10
4	G.P.	0.05	0.65	0.27	0.42	0.57
	logistique	0.16	0.92	0.79	1.10	1.40
9	G.P.	0.22	1.47	3.50	11.04	0.50
	logistique	0.14	1.60	0.62	2.80	1.74
15	G.P.	0.19	2.49	3.20	6.95	1.70
	logistique	0.16	4.67	7.32	3.53	2.68

Tableau 4.IV – Performance des meilleurs modèles sur différentes segmentations originales

entraînement lorsque la segmentation initiale comporte 9 ou 15 feuilles finales. Un test plus révélateur de la performance réelle pouvant être obtenue de ces modèle serait de tester chacun sur des distributions de test stratifiées selon différentes variables reliées au client ou à sa couverture.

4.1.3 Tests avec différent partitionnements de la distribution de test

Afin de mieux valider l'utilité des modèles les plus prometteurs, ces derniers seront testés sur des ensembles de test séparés selon certaines variables prédictrices. Cette procédure tente de reproduire l'utilisation qui sera faite du modèle afin de déterminer lequel est le plus apte à effectuer la tâche demandée (voir section 3.7). Le tableau 4.V présente un résumé des mesures de performance réalisées sur 15 séparations différentes de l'ensemble de test.

Suivant ce test, le modèle possédant la plus grande robustesse (surligné en jaune) serait le processus gaussien sur quatre segments, tandis que le modèle présentant la plus grande précision en général (surligné en vert) serait la régression logistique sur quatre segments. Il est aussi possible de voir qu'en général, la régression logistique produit un

# seg.	Rég.	Mesure	orig.	inc=500	inc=750	inc=1000	inc=-500	Moyenne
1	G.P.	$\bar{x}$	0.62	2.79	3.44	2.88	0.84	2.11
		σ	0.82	4.72	4.68	3.01	0.75	2.80
	log.	$\bar{x}$	1.18	6.27	5.43	5.39	2.18	4.09
		σ	1.69	9.23	8.84	8.00	1.56	5.86
4	G.P.	$\bar{x}$	0.19	1.11	2.10	2.45	0.50	1.27
		σ	0.17	0.83	0.62	1.36	0.41	0.68
	log.	$\bar{x}$	0.21	1.45	1.10	1.27	0.97	1.00
		σ	0.24	1.44	1.15	1.22	0.37	0.88
9	G.P.	$\bar{x}$	0.71	3.70	4.84	4.89	1.06	3.04
		σ	0.88	5.10	5.29	4.33	0.75	3.27
	log.	$\bar{x}$	0.82	5.12	3.79	3.16	1.56	2.89
		σ	1.41	8.22	7.05	5.01	1.19	4.58
15	G.P.	$\bar{x}$	0.57	2.73	3.28	2.75	0.85	2.04
		σ	0.84	4.59	4.57	2.99	0.74	2.75
	log.	$\bar{x}$	0.82	5.12	3.53	3.12	1.98	2.91
		σ	1.38	7.50	6.48	4.90	1.63	4.38

Tableau 4.V – Performance des meilleurs modèles sur différentes façons de séparer l'ensemble de test

modèle ayant une variance de performance plus grande que la régression avec processus gaussien tout en fournissant une performance presque semblable.

4.1.4 Discussion sur les résultats obtenus

Les performances du processus gaussien et de la régression logistique sont assez différentes malgré qu'ils soient entraînés avec exactement les même données. Cette différence est principalement due à la structure très différente des deux modèles.

La régression logistique propose un cadre beaucoup plus rigide quant à la distribution d'entraînement. Elle repose sur l'hypothèse que le modèle peut être représenté comme $ln(\frac{P(y=1)}{P(y=0)}) = a_0 + a_1x_1 + ... + a_jx_j$. Cela implique que le modèle présente une faible capacité lorsque cette hypothèse n'est pas respectée. De cette façon, le modèle peut être très bon pour prédire certaines régions où l'hypothèse est plus respectée et présenter une faible performance lorsque les hypothèses sont moins respectées. Pour cette raison, la régression logistique offre une très bonne performance sur certaines divisions alors que la bonne performance n'est pas au rendez-vous sous certaines autres divisions. Les performances ne sont pas toujours égales.

De son côté, le processus gaussien présente une performance générale un peu moins bonne que la régression logistique, mais les résultats sont plus constants. Ceci est principalement explicable par le fait que le processus gaussien est un modèle prenant en compte le bruit dans les données. Le modèle présente une très grande capacité, mais elle est contrôlée par des paramètres judicieusement choisis. Ainsi, le modèle est construit de façon à prendre en compte le plus possible les variations de la fonction d'entraînement tout en gardant une bonne capacité de généralisation grâce au paramètre de bruit. Ce paramètre fait en sorte que le modèle se plie à la distribution d'entraînement le plus possible afin d'avoir la meilleure performance possible, mais reste tout de même générale en choisissant judicieusement un paramètre caractérisant le bruit dans les données. De plus, le fait de forcer le modèle à prendre les même paramètres sur chacun des segments aide grandement à accroitre la capacité de généralisation. Ce gain en capacité de généralisation se traduit par une plus grande stabilité des résultats obtenus tout en affectant un peu la performance générale du modèle. Ce type de problème s'appelle le dilemme

"biais-variance". Dans de futurs tests, il pourrait être intéressant de ne plus forcer les paramètres du processus gaussien à être les même sur les quatre segments. La précision du modèle en serait ainsi augmentée et la matrice *pépite* pourrait s'assurer d'éviter le sur-entraînement.

Finalement, trop de feuilles dans la segmentation initiale produit un certain sur-entraînement probablement dû au fait que le modèle s'entraîne sur de trop petites distributions sujettes au bruit. Ceci suggère qu'une belle façon d'améliorer ces modèles serait de trouver une meilleure façon d'effectuer la segmentation initiale en essayant d'avoir des segments plus homogènes contenant moins de bruit.

4.2 Plus d'un prédicteur utilisé

Enfin, afin de valider la valeur du modèle, il serait judicieux de comparer les modèles utilisant un seul prédicteur à ceux utilisant tous les prédicteurs disponibles. Pour ces tests, seulement les modèles présentant les meilleures performances avec un seul prédicteur seront envisagés. Ainsi, la régression logistique et la régression à l'aide d'un processus gaussien seront utilisées.

4.2.1 Régression logistique

Le modèle généralement utilisé pour ce genre de tâche est la régression logistique sur tout le jeu d'entraînement avec toutes les variables jugées pertinentes. Ce dernier modèle est celui qui est généralement utilisé de par sa simplicité d'application et de par sa grande pénétration dans le domaine de l'assurance via les actuaires qui utilisent beaucoup de modèles linéaires généralisés. Il est aussi possible d'utiliser un tel modèle complet sur chacune des quatre feuilles finales crées par l'arbre de classification initial.

Le tableau 4.VI présente les résultats des deux meilleurs modèles à un seul prédicteur ainsi que ceux de la régression logistique utilisant les 25 variables disponibles qui ne présentent pas de colinéarité. Les mesures de performance sont effectuées sur 15 différentes séparations du jeu de test (voir section 4.1.3) . Dans le tableau, les cellules en vert représentent la meilleure moyenne de mesure de performance sur chacun des ensembles

de test présentant des changements d'incrément de prime moyenne et les cellules en jaune représentent le meilleur écart type de mesure de performance.

Modèle	Mesure	orig.	inc=500	inc=750	inc=1000	inc=-500	Moyenne
G.P. 4 seg.	$\bar{x}$	0.19	1.11	2.10	2.45	0.50	1.27
1 pred.	σ	0.17	0.83	0.62	1.36	0.41	0.68
log. 4 seg.	$\bar{x}$	0.21	1.45	1.10	1.27	0.97	1.00
1 pred.	σ	0.24	1.44	1.15	1.22	0.37	0.88
logistique	$\bar{x}$	0.12	1.60	1.12	1.00	0.71	0.91
25 pred.	σ	0.17	1.37	1.31	1.03	0.42	0.86
log. 4 seg.	$\bar{x}$	0.20	1.36	0.71	1.20	1.26	0.95
25 pred.	σ	0.25	0.76	0.84	1.43	0.48	0.75

Tableau 4.VI – Comparaison de la performance des meilleurs modèles à un seul prédicteur et de régressions logistiques utilisant plus d'un seul prédicteur sur différentes façons de séparer l'ensemble de test.

Il est possible de voir que la performance du modèle logistique conventionnel est meilleure que celle du modèle logistique construit sur quatre segments et utilisant seulement l'incrément de prime pour la régression. Par contre, le constat n'est pas aussi clair en comparant le modèle logistique conventionnel et le modèle de processus gaussien sur quatre segments. Il est possible de voir que la régression logistique apporte une précision plus grande que la régression avec processus gaussien sur quatre segments, mais en contre-partie, la variance de performance du résultat obtenu est plus grande.

4.2.2 Processus gaussien sur tout l'ensemble d'entraînement

L'utilisation de plus d'un prédicteur apporte de la précision supplémentaire à la régression logistique. Il serait donc intéressant d'effectuer un test similaire avec la régression à l'aide d'un processus gaussien.

Le jeu de données contenant plus de 50000 observations, il n'est pas possible d'utili-

ser directement un processus gaussien. Afin de contourner le problème le mieux possible, une décomposition suivant les différents prédicteurs comme présentée dans la section 3.5.2.2 est nécessaire.

Le tableau 4.VII présente les résultats obtenus avec différents processus gaussiens entraînés à l'aide de tous les prédicteurs disponibles. Le modèle présentant la performance la plus précise et la plus constante sur les différentes façons de séparer l'ensemble de test est la régression à l'aide de processus gaussiens utilisant une matrice pépite [2] ainsi qu'une moyenne variable [3]. De plus, en comparant les mesures de performance obtenues par le processus gaussien utilisant 25 prédicteurs sur 4 segments, une matrice pépite et une moyenne variable avec les mesures de performance obtenues par la régression logistique avec 25 prédicteurs, il est possible de voir que ce dernier est un peu plus précis que le premier lorsque l'augmentation de prime moyenne n'est pas de grande envergure. Par contre, la régression logistique avec 25 prédicteurs est moins précise et constante lorsque l'augmentation de prime moyenne est changée.

[2] noté pépite dans le tableau
[3] noté MV dans le tableau

Modèle	Mesure	orig.	inc=500	inc=750	inc=1000	inc=-500	Moyenne
G.P. 4 seg. 1 pred.	$\bar{x}$	0.19	1.11	2.10	2.45	0.50	1.27
	σ	0.17	0.83	0.62	1.36	0.41	0.68
log. 4 seg. 1 pred	$\bar{x}$	0.21	1.45	1.10	1.27	0.97	1.00
	σ	0.24	1.44	1.15	1.22	0.37	0.88
G.P. 1 seg. 25 pred.	$\bar{x}$	0.26	0.88	0.69	1.21	0.54	0.72
	σ	0.21	0.68	0.91	1.17	0.47	0.69
G.P. 1 seg. 25 pred. pépite	$\bar{x}$	0.25	0.85	0.69	1.22	0.50	0.70
	σ	0.20	0.68	0.87	1.25	0.41	0.68
G.P. 1 seg. 25 pred. MV	$\bar{x}$	0.24	0.83	0.59	1.02	0.61	0.66
	σ	0.15	0.62	0.77	0.86	0.39	0.56
G.P. 1 seg. 25 pred. pépite MV	$\bar{x}$	0.23	0.82	0.62	0.95	0.55	0.63
	σ	0.14	0.60	0.80	0.81	0.36	0.54
G.P. 4 seg. 25 pred.	$\bar{x}$	0.23	0.82	0.65	1.10	0.51	0.66
	σ	0.23	0.66	0.97	1.12	0.46	0.69
G.P. 4 seg. 25 pred. pépite	$\bar{x}$	0.21	0.71	0.58	1.03	0.46	0.60
	σ	0.20	0.58	0.82	1.03	0.42	0.61
G.P. 4 seg. 25 pred. MV	$\bar{x}$	0.21	0.72	0.58	1.05	0.54	0.62
	σ	0.18	0.54	0.82	0.87	0.39	0.56
G.P. 4 seg. 25 pred. pépite VM	$\bar{x}$	0.18	0.68	0.53	0.84	0.51	0.55
	σ	0.16	0.50	0.75	0.76	0.37	0.51

Tableau 4.VII – Comparaison de la performance des modèles à un seul prédicteur et de la régression à l'aide de processus gaussien avec tous les prédicteurs pertinants.

CHAPITRE 5

DISCUSSION

La régression logistique et la régression à l'aide de processus gaussiens sont les modèles essayés qui ont démontré la meilleure performance lors des tests sur différentes segmentations de l'ensemble de test. Ces deux types de modèles ont aussi obtenu une meilleure mesure de performance lorsque plus d'un prédicteur a été utilisé. L'utilisation de seulement l'augmentation de prime comme prédicteur est intéressante dans le sens que cela réduit la complexité du modèle et, par le fait même, le travail nécessaire à la mise en place d'un tel modèle. Malgré que le but du modèle développé est de prévoir l'allure du portefeuille suivant l'augmentation de prime projetée, l'utilisation de cette variable seule par un modèle entraîné sur un seul ou plusieurs segments de la distribution n'apporte pas une solution intéressante par rapport à l'utilisation d'une simple régression logistique sur tout l'ensemble d'entraînement. Ce constat démontre que le changement de prime seul ne peut expliquer la propension des clients à changer de compagnie d'assurance au moment de renouveler leur police d'assurance.

Il est alors possible de pousser plus loin le modèle développé pour un seul prédicteur afin d'obtenir un modèle capable de mieux prendre en compte les caractéristiques des clients permettant de prédire leur comportement en rapport au changement de prime projetée. Ainsi, une procédure plus complexe utilisant plus d'une variable a été développée. Étant donné la grandeur du jeu de données, il n'est pas possible d'utiliser directement une classification à l'aide de processus gaussien afin d'effectuer la tâche. Le modèle développé modélise de façon univariée la propension à l'attrition de chacun des clients avant d'utiliser une régression logistique afin de prendre en compte les interactions entre les prédicteurs et ainsi effectuer une prédiction plus précise du comportement de chacun. La performance obtenue à l'aide de ce type de modèle est supérieure aux autres qui ont été testés. Cela montre clairement que beaucoup de facteurs entrent en compte quand vient le moment de déterminer si un client va annuler sa police d'assurance.

De plus, le modèle utilisant des régressions à l'aide de processus gaussien pour en-

suite utiliser une régression logistique est supérieur à celui utilisant seulement une régression logistique sur les variables originales. Cette situation démontre la non linéarité dans l'action des différent prédicteurs sur le phénomène. En effet, la régression logistique ne peut modéliser qu'un effet linéaire de chacun des prédicteurs dans la variabilité de la variable cible. L'utilisation de régression à l'aide de processus gaussien amène donc une flexibilité supplémentaire à la régression logistique.

Aussi, il est naturel d'utiliser une moyenne variable pour les processus gaussiens modélisant la propension à l'annulation de chacun des assurés par rapport à chacune des variables. En effet, l'hypothèse stipulant que la propension moyenne à l'attrition de chaque individu fluctue autour d'une fonction linéaire par rapport au prédicteur est très naturelle. Ne pas faire cette hypothèse impliquerait une fluctuation autour d'une fonction constante par rapport au prédicteur. Cette seconde option est moins en harmonie avec la philosophie derrière l'utilisation du processus gaussien pour la modélisation d'un phénomène comme celui qui nous intéresse. La meilleure performance des modèles utilisant une moyenne variable par rapport à ceux ne l'utilisant pas conforte cette position.

Enfin, l'ajout d'une matrice pépite dans les processus gaussiens apporte plus de précision ainsi que plus de stabilité dans les résultats obtenus. Cette matrice sert à modéliser le bruit dans les données. Il appert donc que les données sont bruitées et que l'utilisation d'un modèle prenant en compte cette réalité est une bonne chose. Étant donné que le comportement humain n'est pas toujours rationnel et que chaque individu possède une personnalité propre, la modélisation de phénomènes impliquant des humains devrait être résistante à un certain niveau de bruit.

Finalement, la régression à l'aide de processus gaussien produit un modèle plus robuste que la régression logistique, même lorsque les paramètres des GP ne sont pas forcés de prendre la même valeur dans les modèles entraînés sur chacune des feuilles finales. Cela montre l'effet positif de la matrice pépite dans le contrôle de la capacité du modèle.

CHAPITRE 6

CONCLUSION

L'objectif à atteindre dans ce projet était de développer un modèle capable d'effectuer des simulations de scénarios de changements de prime sur un portefeuille d'assurance automobile. Nous voulons prédire le contenu de celui-ci en rapport à l'attrition qui découle du changement de prime. La disponibilité d'un tel outil apporte un avantage concurrentiel certain à une compagnie d'assurance. Cela permet de mieux gérer l'allure du porte-feuille de risques couverts en ajustant les changements de taux projetés dans le but de ne pas mettre en application un changement qui aurait pour effet de changer le porte-feuille de manière négative pour l'entreprise. Cela permet de déterminer s'il est préférable de mettre en application un changement de taux moindre que celui voulu initialement afin de garder une clientèle profitable dans le futur, quitte à accuser des pertes durant la période qui s'en vient.

Dans le milieu de l'assurance, de par la forte présence d'actuaires utilisant principalement des modèle de la famille des GLM, la régression logistique est généralement utilisée pour effectuer ce genre de tâche. Une approche alternative a été développée dans le cadre de ce mémoire. Cette approche consiste à modéliser le non renouvellement des polices d'assurance en date d'échéance de la protection sur différents segments de la distribution d'entraînement à l'aide de régressions avec processus gaussiens entraînés de façon unidimensionnelles.

Les différents essais effectués sur les données d'un gros assureur canadien montrent que le modèle alternatif utilisant une segmentation initiale à quatre segments ainsi que toutes les variables disponibles avec des processus gaussiens unidimensionnels suivis d'une régression logistique offre une solution innovatrice et plus performante que la régression logistique seule. Cette nouvelle solution offre aussi une performance supérieure à la régression logistique sur un jeu de données présentant des scénarios de changement de primes très élevés ainsi que sur différentes façons de segmenter l'ensemble de test.

L'apport de ce mémoire aura été d'apporter un modèle différent pour la prédiction

du non renouvellement au moment de l'échéance de la police en assurance automobile qui présente une performance plus constante et plus précise que la régression logistique seule.

BIBLIOGRAPHIE

[1] A.Baritchi et D. J. Cook. Discovering structural patterns in telecommunications data. Dans *Proceding of the Florida Artificial Intelligence Research Symposium*, pages 82–85, 2000.

[2] C. Apte, E. Grossman, E. P. D. Pednault, B. Rosen, F. Tipu et B. White. Probabilistic estimation based data mining for discovering insurance risks. *Intelligent Systems and their Applications, IEEE*, 14(6):49–58, 1999.

[3] C. Apte, B. Liu, E. P. D. Pednault et P. Smyth. Business applications of data mining. *Communications of the ACM*, 45(8):49–53, 2002.

[4] M. Avriel. *Nonlinear Programming : Analysis and Methods*. Dover Publications, 2003.

[5] R. N. Bolton. A dynamic model of duration of the customer's relationship with a continuous service provider : The role of satisfaction. *Marketing Science*, 17(1): 45–65, 1998.

[6] J. P. Bradford, C. Kunz, R. Kohavi, C. Brunk et C. E. Brodley. Pruning decision trees with misclassification costs. Dans *Proceedings of ECML-98*, pages 131–136, 1998.

[7] L. Breiman, J. Friedman, R. Olshen et C. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[8] R. L. Burden et D. J. Faires. *Numerical Analysis, ninth edition*. Brooks/Cole, Cengage Learning, 2010.

[9] Y. M. Chae, S. H. Ho, K. W. Cho, D. H. Lee et S. H. Ji. Data mining approach to policy analysis in a health insurance domain. *International Journal of Medical Informatics*, 62(2-3):103–111, 2001.

[10] N. Chapados, Y. Bengio, P. Vincent, J. Ghosn, C. Dugas, I. Takeushi et L. Meng. Estimating car insurance premia : a case study in high-dimensional data inference. *DIRO Technical Report*, 1199, 2001.

[11] W. S. Chen et Y. K. Du. Using neural networks and data mining techniques for the financial distress prediction model. *Expert Systems with Applications*, 36(2): 4075–4086, 2009.

[12] P. W. Chen-Seng et L. Hua. Large scale analysis of renewals discounts for p&c insurance. *2010 CAS Ratemaking and Product Management Seminar UND-1*, 2010.

[13] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836, 1979.

[14] W. S. Cleveland et W. S. Devlin. Locally-weighted regression : An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610, 1988.

[15] W. S. Cleveland et E. Grosse. Computational methods for local regression. *Statistics and Computing*, 1:47–62, 1991.

[16] R. A. Cohen. An introduction to proc loess for local regression. Dans M. Lajiness, éditeur, *Proceedings of the Twenty-fourth SAS Users Group International Conference*, pages 273–281. SAS Institute, 1999.

[17] G. Dancik. mlegp : an r package for gaussian process modeling and sensitivity analysis. *http ://cran.r-project.org/web/packages/mlegp/vignettes/mlegp.pdf*, 2007.

[18] G. B. Dantzig et M. N. Thapa. *Linear Programming : 1 : Introduction*. Springer Series in Operations Research, 1997.

[19] S. P. D'Arcy. Predictive modeling in automobile insurance : A preliminary analysis. Dans *World Risk and Insurance Economics Congress*, 2005.

[20] C. Dugas, Y. Bengio, N. Chapados, P. Vincent, G. Denoncourt et C. Fournier. Statistical learning algorithms applied to automobile insurance ratemaking. *CAS Forum*, 1(1):179–214, 2003.

[21] T. Fawcett. Roc graphs : Notes and practical considerationsfor researchers. Dans *Tech Report HPL-2003-4*. HP Laboratories, 2004.

[22] T. Fawcett et F. Provost. Adaptative fraud detection. *Data Mining and Knowledge Discovery Journal*, 1(3):291–316, 1997.

[23] U. Fayyad, G. Piatetsky-Shapiro et P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37–54, 1996.

[24] J. Fox. *Applied Regression Analysis and Generalized Linear Models*. SAGE, 2008.

[25] Coalition Against Insurance Fraud. Annual report. *Coalition Against Insurance Fraud*, 2006.

[26] W. J. Frawley, G. Piatetsky-Shapiro et C. J. Matheus. Knowledge discovery in databases : An overview. Dans G. Piatetsky-Shapiro et W. J. Frawley, éditeurs, *Knowledge Discovery in Databases*, pages 1–27. AAAI/MIT Press, 1991.

[27] R. M. Grant. *Contemporary Strategy Analysis*. WileyPLUS, 2004.

[28] A. Gustafsson, M. D. Johnson et I. Roos. The effect of customer satisfaction, relationship commitment dimension, and triggers on customers retention. *Journal of Marketing*, 69:210–218, 2005.

[29] E. Gustafsson. Customer duration in non-life insurance industry. *Master thesis, Stockholm University*, 2009.

[30] D. J. Hand. Statistics and data mining : Intersecting disciplines. *ACM SIGKDD Explorations*, 1(1):16–19, 1999.

[31] T. Hastie, R. Tibshirani et J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics, 2009.

[32] D. M. Hawkins. The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, 44:1–12, 2004.

[33] C. Homburg, N. Koschate et W. D. Hoyer. Do satisfied costumers really par more ? a study of the relationship between customer satisfaction and willingness to pay. *Journal of Marketing*, 69:84–96, 2005.

[34] SPSS Inc. *PASW Modeler 14 Algorithms Guide*. SPSS, 2010.

[35] Alexandros Karatzoglou, Alex Smola, Kurt Hornik et Achim Zeileis. kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20, 2004. URL `http://www.jstatsoft.org/v11/i09/`.

[36] E. Kirkos, C. Spathis et Y. Manolopoulos. Data mining techniques for the detection of fraudulent financial statement. *Expert Systems with Applications*, 32(4):995–1003, 2007.

[37] I. Kolyshkina et R. Brookes. Data mining approaches to modelling insurance risk. *Report, PricewaterhouseCoopers*, 2002.

[38] B. Larivière et D. Van der Poel. Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications*, 29:472–484, 2005.

[39] J. S. Lee et J. C. Lee. Customer churn prediction by hybrid model. *Lecture Notes in Computer Science*, 4093:959–966, 2006.

[40] W. Lee, S. Stolfo, P. Chan, E. Eskin, W. Fan, M. Miller, S.Hershkop et J. Zhang. Real time data mining-based intrusion detection. Dans *DARPA Information Survivability Conference and Exposition II*, 2001.

[41] G. Matheron. Krigeage d'un panneau rectangulaire par sa périphérie. *Note géostatistique no.28, CG, École des Mines de Paris*, 1960.

[42] C. Modlin. Modeling policy holder retention. *2004 CAS seminar on Ratemaking*, 2004.

[43] K. Morik et H. Köpcke. Analysing customer churn in insurance data-a case study. *Lecture Notes in Computer Science*, 3202/2004:325–336, 2004.

[44] S. A. Neslin, S. Gupta, W. Kamakura, J. Lu et C. Mason. Defection detection : Improving predictive accuracy of customer churn models. *Journal of Marketing Research*, 43(2):204–211, 2006.

[45] N. Palmer, S. Tanner, C. Detrick et I. Wagner. The top line is the bottom line in insurance. *Bain & Company*, 2006.

[46] P.Chan, W. Fan, A. Prodromidis et S. stolfo. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems*, 14(6):67–74, 1999.

[47] F. Provost, T. Fawcett et R. Kohavi. The case against accuracy estimation for comparing induction algorithms. Dans J. Shavlik, éditeur, *ICML-98*, pages 445–453. Morgan Kaufmann, 1998.

[48] E. Rahm et H. H. Do. Data cleaning : Problems and current approaches. *IEEE Bulletin on Data Engineering*, 23(4):3–13, 2000.

[49] C. E. Rasmussen et C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[50] M. Richeldi et A. Perrucci. Churn analysis case study. *Deliverable D17.2, IST Project MiningMart*, IST-11993, 2002.

[51] D. Ruta, D. Nauck et B. Azvine. K nearest sequence method and its application to churn prediction. *Lecture Notes in Computer Science*, 4224:207–215, 2006.

[52] T. J. Santner, B. J. Williams et W. I. Notz. *The Design and Analysis of Computer Experiments*. Springer Series in Statistics, 2003.

[53] W. S. Sarle. Stopped training and other remedies for overfitting. Dans *Proceedings of the 27th Symposium on the Interface of Computing Science and Statistics*, pages 352–360, 1995.

[54] A. Silberschatz, M. Stonebraker et J. D. Ullman. Database systems : Achivements and opportunities. *Communications of the ACM*, 34(10):110–120, 1991.

[55] A. Silberschatz, M. Stonebraker et J. D. Ullman. Database research : Achivements and opportunities into the 21st century. Dans *Report of an NSF Workshop on the Future of Database Systems Research*, May 1995.

[56] K. A. Smith, R. T. Willis et M. Brooks. An analysis of customer retention and insurance claim patterns using data mining : A case study. *The Journal of the Operational Research Society*, 51(5):532–541, 2000.

[57] L. Thomas. Customer loyalty and retention primer. *http://findarticles.com/p/articles/mi_qa3615/is_199802/ai_n8789950 viewed on 2010/07/14*, 2006.

[58] C. P. Wei et I. T. Chiu. Turning telecommunications call details to churn prediction : A data mining approach. *Expert Systems with Applications*, 23:103–112, 2002.

[59] S. Weisberg. *Applied Linear Regression*. Wiley Series in Probability and Statistics, 2005.

[60] G. M. Weiss. Data mining in telecommunications. Dans O. Maimon et L. Rokach, éditeurs, *Data Mining and Knowledge Discovery Handbook*. Springer, 2005.

[61] C. K. I. Williams et D. Barber. Bayesian classification with gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

[62] Y. Yan et H. Xie. Research on the application of data mining technology in insurance informatization. *2009 Ninth International Conference on Hybrid Intelligent Systems*, 3:202–205, 2009.

[63] L. S. Yang et C. Chiu. Knowledge discovery on customer churn prediction. Dans *Proceedings of the 10th WSEAS Interbational Conference on APPLIED MATHEMATICS*, pages 523–528, 2006.

[64] W. Ying, X. Li, Y. Xie et E. Johnson. Preventing customer churn by using random forests modeling. *Information Reuse and Integration, 2008. IRI 2008. IEEE International Conference on*, pages 429–434, 2008.