Mémoire présenté le : 08/07/2021

pour l'obtention du Diplôme Universitaire d'actuariat de l'ISFA et l'admission à l'Institut des Actuaires

Par : David DELRIO

Titre : Méthodes d'apprentissage statistique pour la segmentation des sinistres et l'évaluation des provisions non-vie

Confidentialité : ☐ NON ☒ OUI (Durée : ☐ 1 an ☒ 2 ans)
Les signataires s'engagent à respecter la confidentialité indiquée ci-dessus

Membres présents du jury de l'Institut des Actuaires — *Signature*

Frédéric SCHWACH

Charlotte HUTHER

Camille CHAPUIS

Membres présents du jury de l'ISFA

Denis CLOT

Entreprise :
Nom : AXA France
Signature :

Directeur de mémoire en entreprise :
Nom : Renaud MOUYRIN
Signature :

Invité :
Nom :
Signature :

Autorisation de publication et de mise en ligne sur un site de diffusion de documents actuariels *(après expiration de l'éventuel délai de confidentialité)*

Signature du responsable entreprise

Signature du candidat

Table des matières

Résumé

Mots-clefs : Provisionnement, Segmentation, Apprentissage Statistique, K-Means, K-Modes, K-Prototypes, Gradient Boosting Machine, CART, Forêts aléatoires, XGBoost, Assurance IARD, Chain-Ladder

Dans le cadre du provisionnement non-vie, la segmentation des sinistres automobile repose en général sur un regroupement selon plusieurs garanties (Incendie, Vol, Bris de Glace, ...) mais également selon le type de sinistralité (attritionnels, graves et climatiques).

Cette segmentation en vigueur dans le contexte réglementaire actuel a pour but d'homogénéiser les classes de sinistres afin d'augmenter la précision des méthodes de provisionnement principalement basées sur des méthodes agrégées, telles que la méthode *Chain-Ladder*. Cette méthode reposant sur une hypothèse forte de stabilité de facteurs de développement individuels, il est important de regrouper les sinistres selon des critères de durée ou de coût par exemple.

Ce mémoire aura ainsi pour but de proposer une solution de segmentation alternative basée sur des méthodes de *Clustering* en vue d'identifier des groupes de sinistralité pour effectuer des estimations de provisions par classe de sinistres. L'estimation des provisions sera effectuée selon la même méthode afin de pouvoir assurer une comparabilité des résultats.

Les méthodes de *Clustering*, constitueront une approche non supervisée en vue de créer des classes de sinistralité homogènes. La minimisation de la variance intra-groupes ainsi que la maximisation de variance inter-groupes pourront ainsi permettre de minimiser les erreurs d'estimation finales lorsqu'elles seront effectuées par une méthode de provisionnement agrégée. Une approche supervisée sera ensuite utilisée afin de classifier chaque sinistre selon ses caractéristiques dans une de ces nouvelles classes de sinistralité. Cette seconde approche utilisera des modèles de substitution, principalement des algorithmes de classification basés sur des arbres de décision puis sur des méthodes d'apprentissage statistique plus complexes telles que les forets aléatoires ou l'utilisation du *gradient boosting machine*.

La validation des modèles sera effectuée dans un premier temps selon une comparaison entre les montants retenus par projection *Chain-Ladder* sur les sinistres selon la segmentation actuelle et celle issue des modèles d'apprentissage statistique. Enfin, un test de performance sera effectué afin de proposer une segmentation optimale parmi celles proposées, tout en respectant des contraintes opérationnelles liées aux méthodes agrégées de provisionnement.

Abstract

Keywords : Reserving, Segmentation, Machine Learning, K-Means, K-Modes, K-Prototypes, Gradient Boosting Machine, CART, Random Forest, XGBoost, Non-Life Insurance, Chain-Ladder

In property and casualty claims reserving context, motor insurance claims are usually grouped by type of claim (Third Party Liability, or motor damage insurance) and size of the incurred loss. This is made according to the main lines of business (Fire, Theft, Windscreen, ...) but also regarding the type of claim (Attritional, Large or Climatic).

Claim reserving is usually performed using techniques like Chain-Ladder based on the sum of many claims. The group claim homogeneity then determines the reserve evaluation accuracy.

This work investigates the use of clustering method in order to identify relevant groups of claim to improve the reserves estimation.

Clustering methods will constitute an unsupervised approach in order to create homogeneous claim classes. The minimization of the intra-group variance as well as the maximization of the inter-group variance will thus allow to minimize the final estimation errors when they are carried out by an aggregate provisioning method. A supervised approach will then be used in order to classify each claim according to its characteristics within one of these claims classes priorly created by the clustering algorithms. This second approach will be based on surrogate models, mainly classification algorithms like decision trees and then on more complex machine learning methods such as random forest and gradient boosting machine.

The models will be first validated according to a trade off between the amounts retained by Chain-Ladder projection on claims according to the current segmentation and that resulting from Machine Learning models. Finally, a backtesting study will be carried out in order to recommend an optimal segmentation among those proposed, while respecting operational constraints of aggregated reserving methods.

Remerciements

Tout d'abord, je souhaiterais remercier Renaud Mouyrin pour son accueil au sein de son équipe et ses conseils de qualité tout au long de l'année. Je remercie également Marie Vogt pour sa disponibilité lors de mes divers questionnements.

Je tiens également à remercier Matthias Servel, qui, de part son expérience sur le sujet ainsi que sur l'environnement des systèmes d'information d'AXA m'a grandement aidé lors de l'extraction de données.

Enfin, je remercie mon tuteur auprès de l'Institut des Sciences Financières et d'Assurances Pierre-Olivier Goffard pour ses conseils et pistes de recherche, mais également pour le suivi apporté au cours de cette année dans ce contexte particulier.

Introduction

L'activité d'assurance se caractérise par l'incertitude liée aux engagements de l'assureur vis-à-vis de l'assuré. L'accord contractuel entre les deux parties, assureur et assuré, implique le respect de droits et d'obligations de tous ses acteurs. Un contrat d'assurance engage ainsi les responsabilités respectives des deux parties telles que le versement d'une prime pour l'assuré, appelée prime émise, ou le versement d'une prestation pour l'assureur. La prime que s'engage à verser l'assuré est liée à la couverture d'un évènement incertain et aléatoire, et calculée à partir d'informations exogènes sur le risque associé à la police d'assurance. La prestation que s'engage à verser l'assureur en cas de survenance de l'évènement incertain couvert par l'accord contractuel correspond au montant du préjudice assuré dans les conditions établies au préalable par le contrat d'assurance. Si le montant de la prime est préalablement connu, le montant de la prestation demeure quant à lui incertain et s'estime généralement par le biais de modélisations statistiques. Il s'agît d'une spécificité propre au secteur assurantiel, appelée inversion du cycle de production, signifiant le fait qu'un produit d'assurance soit vendu avant d'en connaître le coût final.

Dans le cadre de l'assurance IARD **(Incendie, Accidents et Risques Divers)**, la couverture des risques porte sur les dommages survenus aux biens de l'assuré s'opposant ainsi à l'assurance VIE qui protège les personnes à travers des produits spécifiques à la santé, la vie ou le décès de l'assuré. Dans ce contexte d'incertitude, les primes émises collectées ne peuvent refléter totalement la richesse d'un organisme assureur tant que l'intégralité des sinistres d'un portefeuille d'assuré n'est pas connue. Il est donc nécessaire de constituer des provisions techniques afin de pouvoir honorer les engagements futurs de l'assureur.

Les provisions spécifiques à l'assurance IARD correspondent ainsi aux montants de charges finales non réglées des sinistres déclarés par l'assuré à l'assureur mais également aux montants de charges finales des sinistres non encore déclarés mais prévisibles. L'évaluation de ces montants de provisions repose ainsi sur l'historique de sinistralité observée, et sur l'utilisation de méthodes de provisionnement généralement agrégées. De plus, l'évaluation de ces montants de provisions techniques s'effectuant dans un contexte réglementaire prudentiel, celle-ci se doit d'être la meilleure estimation possible. La provision technique est ainsi égale à la somme du Best Estimate et de la Marge de Risque.

Les provisions évaluées lors de l'inventaire sont alors inscrites au passif du bilan de l'assureur avec une marge de risque calculée indépendamment, venant s'ajouter aux fonds propres de l'organisme assureur. La figure 1 schématise la structure du bilan d'une compagnie d'assurance dans le cadre de la directive solvabilité.

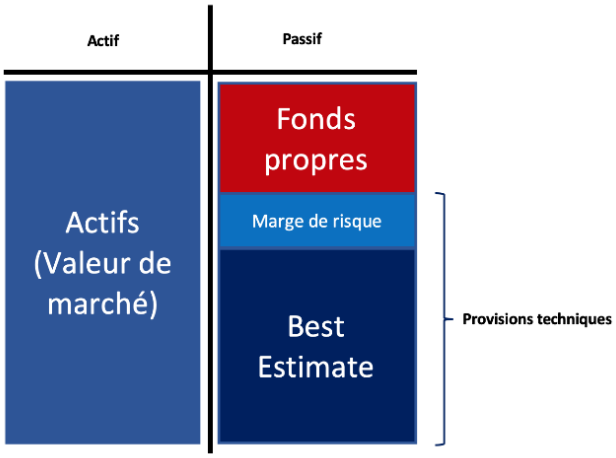

Figure 1 – Bilan Solvabilité II

La bonne estimation des provisions constitue un enjeu majeur pour l'assureur étant donné leur poids dans le passif du bilan. La sous-évaluation des provisions implique un risque de ruine accru ainsi qu'un manque de liquidité tandis que sa surévaluation constitue un manque de rentabilité et peut inévitablement amener à une immobilisation de capitaux.
L'estimation de ces provisions repose généralement sur des méthodes de provisionnement agrégées qui consistent à créer des triangles de développement à partir de l'information que l'on souhaite étudier. La méthode *Chain-Ladder* est la méthode déterministe la plus utilisée par les assureurs de part sa simplicité d'implémentation et son interprétabilité. Sa version stochastique, la méthode de Mack, permet également d'obtenir des précisions concernant l'incertitude liée à l'estimation de montants finaux.
Afin de garantir une estimation qualitative des provisions à travers ces méthodes, une bonne segmentation des sinistres doit être appliquée. Un premier levier de segmentation consiste à créer des classes homogènes de sinistralité selon leur durée et leur coût. Ces classes sont ainsi implicitement crées lors d'une segmentation par branches d'activités. Il est cependant possible de créer une nouvelle segmentation en améliorant l'homogénéité des classes à l'aide de méthodes d'apprentissage statistique non-supervisées principalement basées sur la création de *clusters*.
Les méthodes non supervisées de classification *K-Means*, *K-Modes* ou encore *K-Prototypes*, nous permettront à travers les différentes variables de notre base de données d'effectuer des regroupements par similarité des observations. L'objectif conjoint de ces méthodes sera de minimiser la variance d'indicateurs au sein de ces classes tout en maximisant cette variance entre les classes.

Les méthodes basées sur des arbres tels que *CART* ou les forêts aléatoires et les modèles agrégés de type *gradient boosting machine* (ou *XGBoost*) permettront à partir des caractéris-

tiques des sinistres récents d'associer une classe de sinistralité appropriée à chacun.

Appliqué à la base de sinistralité automobile 4 roues Particuliers d'AXA France, il sera possible de comparer les classes modélisées et les classes issues de la segmentation actuarielle en vigueur chez AXA France. Les méthodes *Chain-Ladder* sans jugement d'expert seront ainsi appliquées à ces nouvelles classes en vue de comparer les estimations sur les classes existantes. Un test de validation permettra également de mesurer l'erreur réelle d'estimation des méthodes mais également l'impact d'une segmentation basée sur des méthodes statistiques sur les erreurs d'estimation (boni/mali de liquidation).

1 Le provisionnement non-vie

1.1 La charge finale prévisible

La provision technique spécifique à l'assurance IARD se compose de plusieurs éléments. Nous introduisons ici les différents éléments techniques et de vocabulaire propres à la sinistralité IARD ainsi qu'au provisionnement.

Dans le cas de l'assurance non-vie, on distingue deux principales provisions techniques : les provisions pour sinistres et les provisions pour primes. Dans le cadre de ce mémoire, nous nous intéresserons principalement aux provisions pour sinistres. La Charge Finale Prévisible se décompose ainsi :

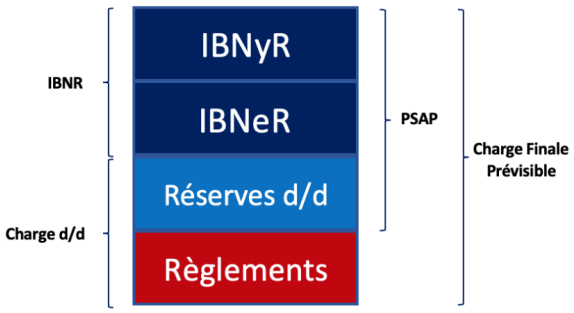

Figure 2 – Décomposition de la Charge Finale Prévisible

1.1.1 Charge dossier/dossier

La charge dossier/dossier est un montant de provisions pour les sinistres déclarés et par conséquent connus par l'assureur. Le montant de provisions dossier/dossier est généralement évalué individuellement par les gestionnaires de sinistres, il s'agît donc de la somme de chaque montant probable de décaissement pour chaque sinistre. Il est également possible de calculer le montant total de la charge dossier/dossier par estimation globale du montant total des décaissements à prévoir sur les sinistres en cours de gestion.

1.1.2 Provision *Incurred But Not Reported* (IBNR)

La provision IBNR est elle-même constituée de deux provisions distinctes :

- ***La provision IBNeR***, *Incurred But Not enough Reserved*, correspondant à la couverture d'une potentielle insuffisance de provisionnement de sinistres survenus et déclarés à la date de clôture des états financiers.

- ***La provision IBNyR***, ***I****ncurred* ***B****ut* ***N****ot* ***y****et* ***R****eported*, correspondant à une estimation du coût ultime des sinistres survenus mais non encore déclarés à la date de clôture.

On peut ainsi noter :

$$IBNR = IBNeR + IBNyR$$

1.1.3 Provision pour Sinistres à Payer (PSAP)

Cette provision représente une part importante des provisions inscrites au passif du bilan de l'assureur. La bonne estimation de son montant représente donc un intérêt majeur. Cette provision est définie par l'article R343-7 du Code des Assurances comme la *"valeur estimative des dépenses en principal et en frais, tant internes qu'externes, nécessaires au règlement de tous les sinistres survenus et non payés, y compris les capitaux constitutifs des rentes non encore mises à la charge de l'entreprise."*

La PSAP représente donc le montant estimatif à régler pour des sinistres déclarés ou non à la date d'inventaire. On peut donc la noter ainsi :

$$PSAP = IBNR + \text{Provision d/d}$$

1.2 Contexte

1.2.1 Exemple de déroulement d'un sinistre

Les différentes étapes de la vie d'un sinistre peuvent être représentées ainsi :

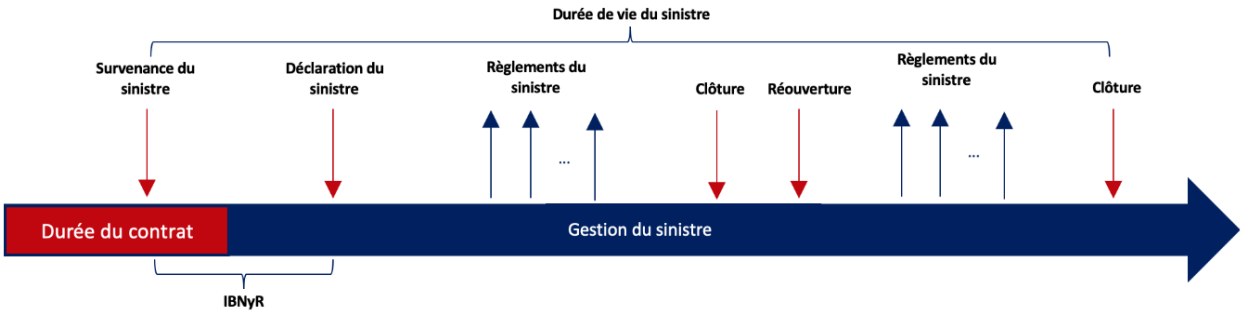

Figure 3 – Déroulement d'un sinistre

Le temps écoulé entre la survenance du sinistre et sa déclaration peut aller jusqu'à plusieurs années. Il est également possible que la charge totale d'un sinistre ne soit écoulée qu'au bout de

plusieurs années, ces sinistres sont communément appelés sinistres à branches longues, il s'agît généralement de sinistres corporels. Un sinistre peut également être rouvert à la suite d'une clôture lorsque des informations supplémentaires peuvent être ajoutées au dossier du sinistre par exemple. Toutes ces informations doivent alors être prises en compte dans le calcul des provisions.

Afin d'estimer au mieux le montant de provisions à allouer à un sinistre, il est nécessaire de connaître son développement. L'utilisation de méthodes agrégées comme la méthode *Chain-Ladder* permettra ainsi de calculer des facteurs de développement afin de déduire des estimations de charges finales prévisibles.

1.2.2 Risque sous-jacent de provisionnement

Une grande partie de la qualité de l'estimation des provisions techniques en assurance non-vie réside dans la segmentation des sinistres. En effet, l'estimation de la charge finale d'un sinistre à développement court tel qu'un sinistre bris de glace dans un triangle de sinistre corporels sera vue à la hausse et inversement. Plus généralement, effectuer un calcul de provisions techniques à l'aide d'une méthode agrégée sur l'ensemble du portefeuille sans segmentation préalable contribuera à la projection de sinistres à développements différents suivant le même cadencement. La charge finale de certains sinistres pourrait alors être sur ou sous-estimée.

1.2.3 Le provisionnement sous Solvabilité II

Dans le contexte prudentiel imposé par Solvabilité II, il est important pour les assureurs de constituer suffisamment de provisions techniques afin de couvrir l'intégralité de leurs engagements futurs. En assurance non-vie, il s'agît d'estimer le montant probable de sinistres à venir en prenant compte également des frais de gestion associés.

Cette provision technique est la somme de la meilleure estimation des sinistres et de la marge de risque. La directive Solvabilité II définit la meilleure estimation (Best Estimate) comme suit :

« La moyenne pondérée par leur probabilité des flux de trésorerie futurs, compte tenu de la valeur temporelle de l'argent (valeur actuelle attendue des flux de trésorerie futurs), estimée sur la base de la courbe des taux sans risque pertinents. Le calcul de la meilleure estimation est fondé sur des informations actualisées et crédibles et des hypothèses réalistes et il fait appel à des méthodes actuarielles et statistiques adéquates, applicables et pertinentes ».

Le Best Estimate étant calculé brut de réassurance.

La marge de risque est calculée de manière à garantir que la valeur des provisions techniques est équivalente au montant que les entreprises d'assurance et de réassurance demanderaient pour reprendre et honorer les engagements d'assurance et de réassurance. La marge de risque ainsi que le Best Estimate sont calculés de manière indépendante.

Dans le but de rendre l'estimation de provisions plus justes, la directive Solvabilité II impose une segmentation effectuée au minimum par ligne d'activité, comme le stipule l'article 80 de la directive 2009/138/CE du parlement européen et du conseil du 25 novembre 2009 sur l'accès aux activités de l'assurance et de la réassurance et leur exercice (solvabilité II) :

Segmentation

« Lorsqu'elles calculent leurs provisions techniques, les entreprises d'assurance et de réassurance segmentent leurs engagements d'assurance et de réassurance en groupes de risques homogènes et, au minimum, par ligne d'activité. »

1.2.4 Périmètre d'étude

Ce mémoire s'intéressera principalement au périmètre automobile 4 roues Particulier d'AXA France IARD pour des sinistres survenus à partir de 2010 vus à la date d'inventaire de fin décembre 2019. L'estimation des charges finales prévisibles s'effectue par branches d'activités en distinguant la responsabilité civile corporelle, les garanties dommages et la responsabilité civile matérielle (figure 4). La segmentation actuellement en vigueur chez AXA France consiste à estimer ces trois agrégats de garanties de manière indépendante. D'autres regroupements en tranches seront ensuite crées au sein de chaque agrégat.

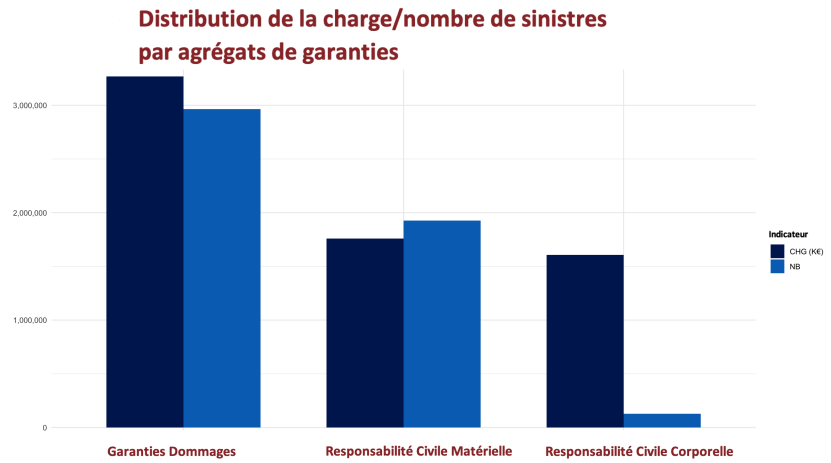

Figure 4 – Distribution du nombre et de la charge de sinistres par agrégats de garanties

La garantie Dommages

La garantie dommages regroupe les garanties suivantes :

- **Bris de Glace**
- **Incendie / Vol**
- **Garantie Principale Conducteur** (couvrant les dommages corporels du conducteur)
- **Dommages tous accidents**
- **Autres dommages**

La garantie Dommages représente 59% de notre portefeuille global et 3 267M€ de charges totales à la dernière date d'inventaire. Après de nombreuses études, l'estimation des provisions sur cette branche d'activité est faite selon deux segments distincts, regroupant pour le premier segment l'ensemble des dommages matériels (garanties bris de glace, incendie/Vol ainsi que les dommages tous accidents et autres) puis un second segment regroupant les dommages corporels (garantie principale conducteur et une part des autres dommages). La sinistralité de chacun de ces segments sera ensuite estimée à l'aide de la méthode du *Chain-Ladder* de charges ou de règlements.

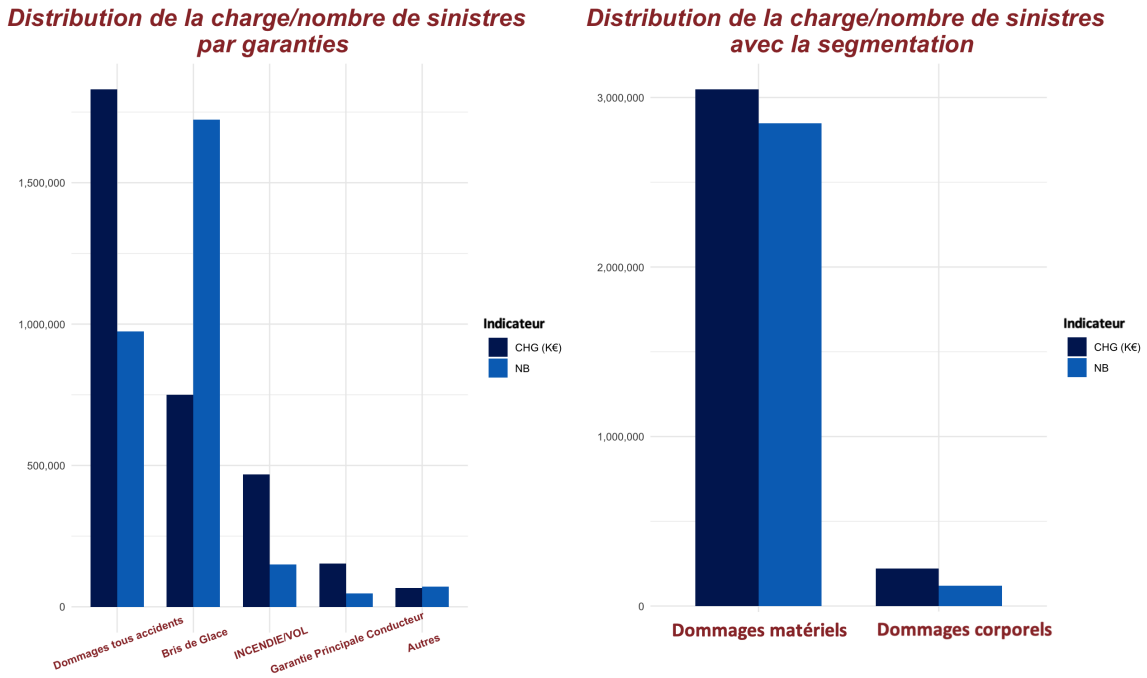

Figure 5 – Distribution du nombre et de la charge de sinistres par garanties et après regroupements

La Responsabilité Civile automobile Corporelle

La Responsabilité Civile corporelle automobile représente dans notre portefeuille 2,5% de la sinistralité, et 1 608M€ de charges totales à la dernière date d'inventaire.

La branche Responsabilité Civile corporelle est regroupée en 3 segments afin de pouvoir estimer de manière plus précise les provisions. Ces regroupements sont, pour la segmentation en vigueur chez AXA, effectués selon le montant de charge du sinistre lors de la date d'inventaire :

- **T1** : Sinistres dont la charge est inférieure à 150K à date d'inventaire
- **T2** : Sinistres dont la charge est entre 150K et 750K à date d'inventaire
- **T3** : Sinistres supérieurs à 750K à date d'inventaire

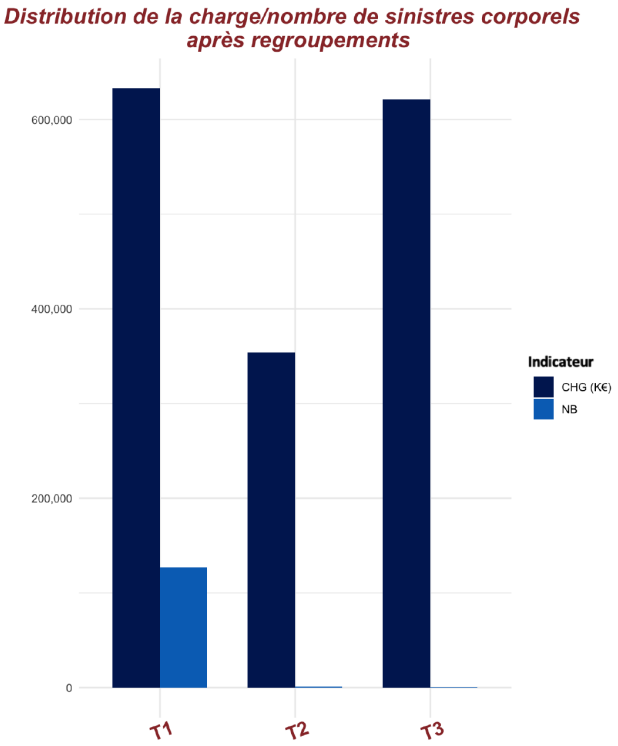

Figure 6 – Distribution du nombre et de la charge de sinistres corporels par tranches de coûts et après regroupements

La méthode classique du *Chain-Ladder* de charges est appliquée aux deux premières tranches de coût. Pour la tranche 3, une méthode de Mack sur un triangle de charges est appliquée en déterminant le quantile à 75% d'une distribution des provisions supposée suivre une loi Log-Normale.

La branche Responsabilité Civile Matérielle

La Responsabilité Civile automobile matérielle couvre l'ensemble des dommages causés à un tiers sur son bien. Dans le cadre des estimations des provisions, cette garantie est considérée indépendemment et est estimée par la méthode *Chain-Ladder* sur règlements. Cette garantie représente environ 38,5% de notre portefeuille et 1 758M€ de charges.

1.2.5 Objectifs

A partir d'une base de données contenant des sinistres survenus à partir du 1^{er} janvier 2010 jusqu'au 31 décembre 2019, observés à chaque fin d'année, contenant les informations de l'assuré sinistré, les caractéristiques de son véhicule ainsi que les informations propres au sinistre, nous souhaitons recréer une segmentation afin de regrouper les sinistres en classes homogènes en termes de sévérité et de développement pour ainsi diminuer le risque sous-jacent de sur ou sous-provisionnement.

L'utilisation d'une méthode de provisionnement agrégée sur des sinistres à développement différents augmente l'erreur de prédiction. Ces développements différents peuvent être expliqués par une hétérogénéité des sévérités ou des fréquences de règlements au sein des sinistres d'un même groupe. Nous déterminerons donc à l'aide de méthodes de *clustering* de nouvelles classes davantage homogènes afin d'en estimer leurs charges finales prévisibles.

Des indicateurs propres à la sinistralité de chacune des garanties seront utilisés de manière à homogénéiser les différentes classes. Les sinistres concernant des dommages corporels utiliseront des informations propres aux victimes telles que le nombre de blessés, ainsi que la gravité des blessures. Ce qui présente un impact direct sur le préjudice généré. Les sinistres concernant des dommages matériels verront leurs groupes créés à partir d'informations propres aux biens endommagés ou détruits, tels que le type de véhicule, son style ou encore son ancienneté.

1.3 Méthodes de provisionnement

L'utilisation de méthodes de provisionnement doit permettre à l'assureur d'estimer au plus juste ses engagements envers ses assurés. Il est donc important d'utiliser des méthodes d'estimation adaptées. Il existe ainsi plusieurs méthodes de provisionnement afin d'estimer les PSAP. Deux ensembles de méthodes se distinguent : les méthodes déterministes et stochastiques.

Les méthodes déterministes telles que la méthode du coût moyen ou *Chain-Ladder* sont des méthodes classiques et très fréquemment utilisées. Elles permettent une estimation correcte

des PSAP mais ne fournissent cependant pas d'estimations de la variance ou d'intervalles de confiance des estimations de PSAP. Les méthodes stochastiques s'appuient sur des modèles d'estimation paramétriques comme proposé par le modèle de Mack (1993)[16]. Le modèle de Mack permet de mesurer l'erreur quadratique moyenne des prévisions (MSEP). Nous utiliserons la méthode du *Chain-Ladder* afin d'estimer les charges ultimes ainsi que le modèle de Mack pour mesurer l'erreur de prédiction des prédictions issues des différentes segmentations de sinistres.

1.3.1 Construction d'un triangle de liquidation

Les méthodes classiques d'estimation de provisions sont des méthodes agrégées et reposent donc sur l'utilisation de triangles de liquidation. Ces triangles de liquidation permettent d'étudier la dynamique d'un ensemble de sinistres. Ils peuvent ainsi être utilisés afin d'analyser le développement des charges, des règlements des réserves ou encore des recours ou des nombres de sinistres. Si l'on considère un inventaire réalisé en fin d'année J, (le $31/12/J$), on peut alors représenter le triangle cumulé de charges comme ceci :

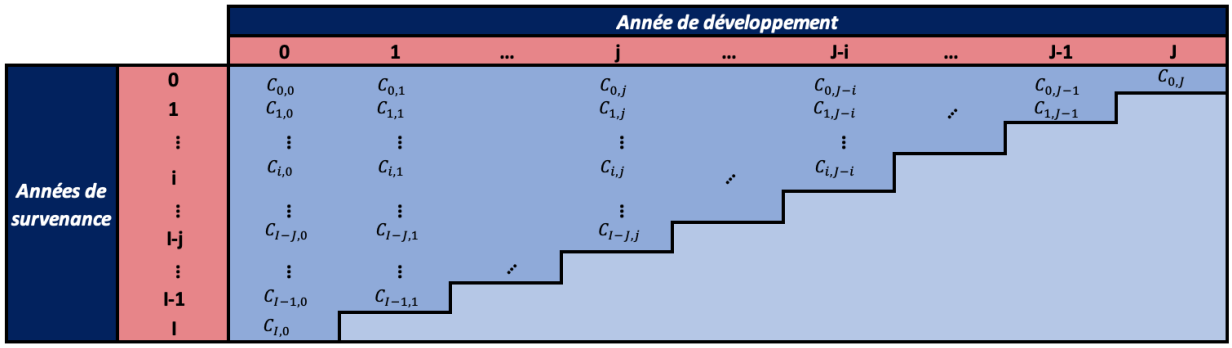

Figure 7 – Triangle de charges cumulé

On peut ainsi introduire les notations suivantes :

- i représente la date de survenance du sinistre

- j le développement du sinistre pouvant être semestriel, mensuel ou encore annuel dans le cadre de notre étude.

- Les $C_{i,j}$, montants cumulés de charges pour l'année d'origine i jusqu'à l'année de vision j, avec $C_{i,j} = \Sigma_{k=0}^{j} X_{i,k}$ qui revient à écrire $X_{i,j} = C_{i,j+1} - C_{i,j}$ où $X_{i,j}$ est le montant de charge non cumulé. Il s'agît du montant correspondant à la survenance i vue à la fin d'année j.

Les facteurs de développement peuvent ainsi être calculés comme suit :

$$f_{i,j} = \frac{C_{i,j+1}}{C_{i,j}}$$

où $f_{i,j}$ est le $j^{ième}$ facteur de développement de l'année de survenance i, avec $j \in \{0, \dots, J-1\}$.

Le triangle de charges est donc utile aux méthodes de provisionnement agrégées, qui consistent à estimer sa partie inférieure, permettant l'estimation de la charge ultime et des réserves.

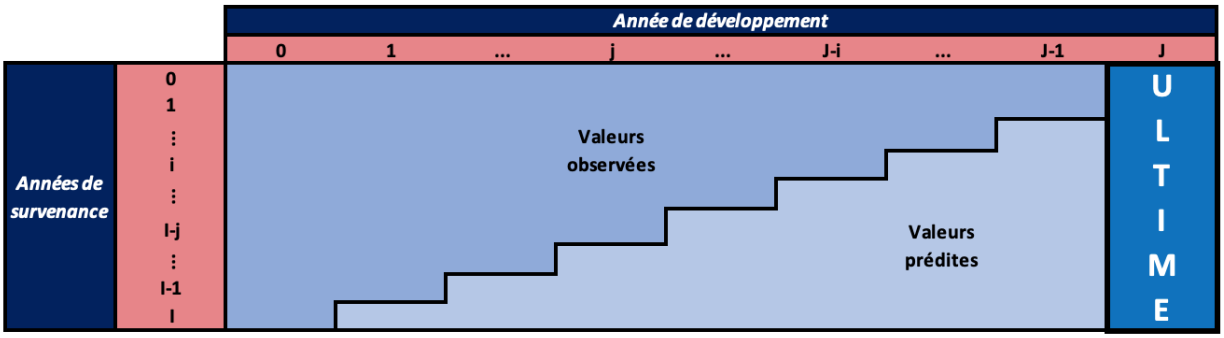

Figure 8 – Triangle inférieur à estimer et charge Ultime

Pour la $i^{ème}$ année de survenance, on note le montant estimé de charge ultime

$$\hat{S}_i = C_{i,J}$$

le montant nécessaire de provisions par année de survenance i se note :

$$\hat{R}_i = S_i - C_{i,J-i}$$

pour $i \in \{0, \dots, I\}$. Enfin, le montant total de provisions s'écrit :

$$\hat{R} = \sum_{i=0}^{I} \hat{R}_i$$

1.3.2 *Chain-Ladder*

La méthode *Chain-Ladder* est la méthode déterministe de référence pour le provisionnement non-vie. Il s'agît d'une méthode relativement ancienne et très fréquemment utilisée en actuariat de part sa simplicité d'utilisation, d'implémentation et d'interprétation. Elle permet l'estimation de charges ultimes mais peut également s'utiliser sur différents types de données tels que les règlements cumulés ou encore les nombres de sinistres. En terme de provisionnement, cette

méthode est généralement privilégiée afin d'obtenir un premier aperçu de la sinistralité d'un portefeuille. Dans certains cas, il est nécessaire d'effectuer des retraitements afin d'améliorer la précision des estimations.

Le principe du *Chain-Ladder* repose sur l'évolution des montants cumulés qui doit rester identique d'une année de développement à une autre pour toute année de survenance. Cette évolution est caractérisée par un coefficient de développement.

Un pré-traitement des données est essentiel lorsque l'on utilise la méthode *Chain-Ladder*. En effet, basé sur une hypothèse de constance des cadences de règlements, le portefeuille considéré doit être homogène et suffisamment grand.

Le modèle repose également sur une hypothèse d'indépendance des facteurs de développement individuels $f_{i,j}$ de l'année de survenance i.

Pour $j \in \{0, \dots, J-1\}$,

$$\frac{C_{1,j+1}}{C_{1,j}} = \frac{C_{2,j+1}}{C_{2,j}} = \dots = \frac{C_{i,j+1}}{C_{i,j}} = \frac{C_{I-j,j+1}}{C_{I-j,j}}$$

Ainsi, le facteur de développement individuel est identique quelque soit l'année de survenance i. En d'autres termes, il ne dépend que du nombre d'années écoulées depuis la survenance du sinistre.

On suppose N années de survenance et N années de développement. Afin de compléter la partie inférieure du triangle de charges, nous devons calculer un facteur de développement entre les années k et $k+1$ comme suit :

$$\hat{\lambda}_k = \frac{\sum_{i=1}^{N-k} C_{i,j+1}}{\sum_{i=1}^{N-k} C_{i,j}}$$

On peut donc compléter le triangle de charges inférieur en calculant les $\hat{C}_{i,j}$, charges du triangle inférieur à estimer :

$$\forall j \geq I-i, \quad \hat{C}_{i,j} = C_{i,N-i} \prod_{k=N-i}^{j-1} \hat{\lambda}_k$$

La charge ultime de la $i^{ème}$ année de survenance peut donc s'exprimer :

$$\hat{S}_i = \hat{C_{i,J}} = C_{i,N-i} \prod_{k=N-i}^{N-1} \hat{\lambda}_k$$

Enfin, pour chaque année de survenance, on peut estimer les réserves ainsi que les réserves totales :

$\forall i \in \{1, \dots, I\}$,

$$\hat{R}_i = \hat{C}_{i,J} - \hat{C}_{i,N-i}$$

$$\hat{R} = \sum_{i=1}^{N} \hat{R}_i$$

La méthode *Chain-Ladder* présente de nombreux avantages pour l'estimation de provisions grâce notamment au fait que les charges puissent avoir des valeurs incrémentales négatives, impliquant un coefficient de passage $\hat{\lambda}_k$ inférieur à 1, ce qui n'est pas rare pour un assureur car dans un contexte prudentiel, il est possible de constater des bonis de liquidation. Il est également possible d'ajuster un modèle *Chain-Ladder* avec des facteurs de queue lorsque l'historique d'un triangle ne permet pas d'évaluer l'ensemble des cadences. On projette ainsi des facteurs de développement au-delà de la taille du triangle. Ces nouveaux facteurs sont généralement obtenus après lissage d'une courbe de référence qui dépend de la période j sur les facteurs connus pour calculer les coefficients futurs.

Les principales limites d'un tel modèle reposent sur l'homogénéité des triangles considérés mais il faut également vérifier que le développement des charges est identique quelque soit l'année de survenance i. Il est possible de vérifier graphiquement ceci en vérifiant que pour chaque période j, les points $(C_{i,j}, C_{i,j+1})$ sont sensiblement alignés.

1.3.3 Méthode de Mack

Par son extension du modèle de *Chain-Ladder*, la méthode de Mack [1993][16] est une méthode non paramétrique permettant l'évaluation de l'erreur commise lors de l'estimation d'un montant de provisions. Cette extension du modèle *Chain-Ladder* est la méthode stochastique la plus utilisée par les assureurs car elle permet de mesurer le risque lié à l'estimation des réserves. Les principales hypothèses du modèle de Mack sont les suivantes :

- (**H1**) Les règlements cumulés $C_{i,j}$ ne dépendent pas de l'année de survenance i :

$$\forall i, j, \{C_{i,1}, \dots, C_{i,J}\} \perp\!\!\!\perp \{C_{j,1}, \dots, C_{j,J}\}$$

- (**H2**) Il existe $\hat{\lambda}_1, \dots, \hat{\lambda}_j$ tel que $\forall i \in \{0, \dots, I\}, j \in \{0, \dots, J\}$,

$$E\left[C_{i,j+1} \mid C_{i,1}, \dots, C_{i,j}\right] = \lambda_j C_{i,j}$$

Le passage d'une année de développement à l'autre est ainsi décrit en termes d'espérance.

Les $\hat{\lambda}_j$ sont des estimateurs sans biais de λ_j.

Preuve :

Soit $D_j = \{C_{i,k} : k \leq j, i+j \leq N+1\}$, la partie du triangle supérieur limitée à l'année de règlement j. Pour $j = 1, ..., N$ on a :

$$E\left[C_{i,j+1} \mid D_j\right] = \lambda_j C_{i,j}$$

Or,

$$E\left[\hat{\lambda}_j \mid D_j\right] = E\left[\frac{\sum_{i=1}^{N-j} C_{i,j+1}}{\sum_{i=1}^{N-j} C_{i,j}} \mid D_j\right] = \frac{\sum_{i=1}^{N-j} E\left[C_{i,j+1} \mid D_j\right]}{\sum_{i=1}^{N-j} C_{i,j}} = \lambda_j$$

On a donc par conditionnement :

$$E\left[\hat{\lambda}_j\right] = E\left[E\left[\hat{\lambda}_j \mid D_j\right]\right] = E\left[\lambda_j\right] = \lambda_j$$

$\hat{\lambda}_j$ est donc un estimateur sans biais de λ_j

- (**H3**) Il existe $\sigma_1^2, \ldots, \sigma_j^2$ tel que $\forall i \in \{0, \ldots, I\}, j \in \{0, \ldots, J\}$,

$$V\left[C_{i,j+1} \mid C_{i,1}, \ldots, C_{i,j}\right] = \sigma_j^2 C_{i,j}$$

On obtient λ_j directement à l'aide de la méthode *Chain-Ladder*.

Pour $j < N-1$, l'estimateur ${\hat{\sigma}_j}^2$ de σ_j^2 est :

$$\hat{\sigma}_j^2 = \frac{1}{I-j-1} \sum_{i=0}^{I-j-1} C_{i,j} \left(\frac{C_{i,j+1}}{C_{i,j}} - \hat{\lambda}_j\right)^2$$

Pour $j = J-1$,

$$\hat{\sigma}_j^2 = \min\left\{\frac{\left(\hat{\sigma}_{j-2}^2\right)^2}{\hat{\sigma}_{j-3}^2}, \min\left\{\hat{\sigma}_{j-2}^2, \hat{\sigma}_{j-3}\right\}\right\}$$

L'estimateur $\hat{\sigma}_j^2$ proposé par Mack est un estimateur sans biais de σ_j^2.

Preuve :

$\forall j \leq N-1$,

$$
\begin{aligned}
E\left[\hat{\sigma}_j^2 \mid D_j\right] &= E\left[\frac{1}{N-j-1}\sum_{i=0}^{N-j} C_{i,j}\left(\frac{C_{i,j+1}}{C_{i,j}} - \lambda_j\right)^2 \mid D_j\right] \\
&= \frac{1}{N-j-1}\left(\sum_{i=0}^{N-j}\frac{E\left[C_{i,j+1}^2 \mid D_j\right]}{C_{i,j}} - 2E\left[\sum_{i=0}^{N-j} C_{i,j+1}\hat{\lambda}_j \mid D_j\right] + E\left[\sum_{i=0}^{N-j} C_{i,j}\hat{\lambda}_j^2 \mid D_j\right]\right) \\
&= \frac{1}{N-j-1}\left(\sum_{i=0}^{N-j}\frac{E\left[C_{i,j+1}^2 \mid D_j\right]}{C_{i,j}} - 2E\left[\sum_{i=0}^{N-j} C_{i,j}\hat{\lambda}_j^2 \mid D_j\right] + E\left[\sum_{i=0}^{N-j} C_{i,j}\hat{\lambda}_j^2 \mid D_j\right]\right) \\
&= \frac{1}{N-j-1}\left(\sum_{i=0}^{N-j}\frac{E\left[C_{i,j+1}^2 \mid D_j\right]}{C_{i,j}} - E\left[\sum_{i=0}^{N-j} C_{i,j}\hat{\lambda}_j^2 \mid D_j\right]\right)
\end{aligned}
$$

Or,

$$
\begin{aligned}
V\left[\hat{\lambda}_j \mid D_j\right] &= V\left[\frac{\sum_{i=1}^{N-j} C_{i,j+1}}{\sum_{i=1}^{N-j} C_{i,j}} \mid D_j\right] \\
&= \frac{\sum_{i=1}^{N-j} V\left[C_{i,j+1} \mid D_j\right]}{\left(\sum_{i=1}^{N-j} C_{i,j}\right)^2} \\
&= \frac{\sum_{i=1}^{N-j} C_{i,j}\sigma_j^2}{\left(\sum_{i=1}^{N-j} C_{i,j}\right)^2} \\
&= \frac{\sigma_j^2}{\sum_{i=1}^{N-j} C_{i,j}}
\end{aligned}
$$

Puis,

$$
\begin{aligned}
E\left[\hat{\sigma}_j^2 \mid D_j\right] &= \frac{1}{N-j-1}\left(\sum_{i=1}^{N-j}\frac{C_{i,j}\sigma_j^2 + C_{i,j}^2\lambda_j^2}{C_{i,j}} - \sum_{i=1}^{N-j} C_{i,j}\left(\frac{\sigma_j^2}{\sum_{i=1}^{N-j} C_{i,j}} + \lambda_j^2\right)\right) \\
&= \sigma_j^2
\end{aligned}
$$

Enfin,

$$
E\left[\hat{\sigma}_j^2\right] = E\left[E\left[\hat{\sigma}_j^2 \mid D_j\right] = \hat{\sigma}_j^2\right.
$$

Les règlements cumulés sont supposés par le modèle de Mack suivre une distribution normale avec une moyenne et une variance qui sont décrites dans les hypothèses (H2) et (H3). Il est important de noter que la première hypothèse de ce modèle n'est plus vérifiée lorsque l'on constate un changement de gestion dans le portefeuille de sinistralité étudié.

Le modèle de Mack sous ses deux hypothèses nous permet d'obtenir les mêmes estimations

que la méthode déterministe du *Chain-Ladder*. Il permet également d'obtenir une formule fermée de la variance de la charge ultime. Il est ainsi possible de déduire l'erreur quadratique moyenne des prédictions plus généralement appelée MSEP (*Mean Square Error of Predictions*) en conditionnant par rapport aux données passées :
Soit $T = \{C_{i,j}, i+j \leq N+1\}$

$$MSEP\left(\hat{C}_{i,N}\right) = E\left[\left(\hat{C}_{i,N} - C_{i,N}\right)^2 \mid T\right]$$

On en déduit :

$$MSEP\left(\hat{C}_{i,N}\right) = \text{Var}\left[C_{i,N} \mid C_{i,j} : i+j \leq N+1\right] + \left(E\left[C_{i,N} \mid C_{i,j} : i+j \leq N+1\right] - \hat{C}_{i,N}\right)^2$$

En posant $\hat{R}_i = \hat{C}_{i,N} - C_{i,N-i+1}$, on remarque que $R_i - \hat{R}_i = C_{i,N} - \hat{C}_{i,N}$, on obtient alors que $MSEP\left(\hat{R}_i\right) = MSEP\left(\hat{C}_{i,N}\right)$. On exprime alors l'erreur standard comme la racine carrée de l'erreur quadratique des prédictions :

$$SEP\left(\hat{R}_i\right) = \sqrt{MSEP\left(\hat{R}_i\right)}$$

Pour obtenir l'estimateur $\hat{MSEP}(\hat{R}_i)$ de $MSEP(\hat{R}_i)$, pour $i = 1, ..., N$, on pose $\hat{C}_{i,I-1} = C_{i,I-1}$ et on note :

$$\hat{MSEP}\left(\hat{R}_i\right) = \hat{C}_{i,J}^2 \sum_{j=I-1}^{J-1} \frac{\hat{\sigma}_j^2}{\hat{\lambda}_j^2}\left[\frac{1}{\hat{C}_{i,j}} + \frac{1}{\sum_{k=0}^{I-j-1} C_{k,j}}\right]$$

Enfin, $MSEP(\hat{R}_i)$ est estimé par :

$$\hat{MSEP}(\hat{R}) = \sum_{i=1}^{I} \hat{MSEP}\left(\hat{R}_i\right) + \hat{C}_{i,J}\left(\sum_{k=i+1,k<I}^{I} \hat{C}_{k,J}\right) \sum_{j=I-i}^{J-1} \frac{2\hat{\sigma}_j^2}{\hat{\lambda}_j^2 \sum_{h=0}^{I-j-1} C_{h,j}}$$

En plus de fournir des erreurs d'estimation, il est possible grâce aux méthodes stochastiques d'obtenir des intervalles de confiance de nos prédictions. Pour calculer ces intervalles ainsi que les quantiles de ces prédictions, il est nécessaire d'émettre une hypothèse paramétrique sur la distribution des provisions R. Les paramètres de la loi retenue seront déterminés par la méthode des moments grâce aux estimateurs de la moyenne et de l'écart-type conditionnels à R. Supposons que $ln(R)$ suive une loi normale $\mathcal{N}(\alpha, \sigma^2)$, telle que le couple (α, σ^2) soit solution de l'équation :

$$\begin{cases} e^{\alpha + \frac{\sigma^2}{2}} = \hat{R} \\ e^{2\alpha+\sigma^2}\left(e^{\sigma^2} - 1\right) = MSEP\left(\hat{R}^2\right) \end{cases}$$

On obtient alors :

$$\begin{cases} \sigma^2 = \ln\left[1 + \frac{MSEP\left(\hat{R}^2\right)}{\hat{R}^2}\right] \\ \alpha = \ln(R) - \frac{\sigma^2}{2} \end{cases}$$

Sous l'hypothèse d'une distribution normale, l'intervalle de prédiction à 95% pour R est :

$$\left[e^{\alpha-1,96\sigma}, e^{\alpha+1,96\sigma}\right]$$

2 Présentation des données d'étude

2.1 Création de la base de données

Afin de pouvoir réaliser une nouvelle segmentation, il a été nécessaire de constituer une base de données de sinistres adaptée et avec suffisamment de profondeur en vue de pouvoir par la suite reconstituer des triangles de charges entre les années de survenance 2010 et 2019. Il était également nécessaire de pouvoir observer l'état d'un sinistre au cours de son développement, en considérant plusieurs dates d'inventaires qui seront par défaut les derniers jours des années 2010 à 2019. Un programme SAS a donc été développé afin de récupérer l'ensemble des sinistres survenus entre le 1^{er} janvier 2010 et le 31 décembre 2019 pour 10 dates d'inventaires différentes afin d'obtenir un cadencement annuel de ces sinistres. Cette nouvelle base de données contient ainsi des informations financières telles que les règlements, les réserves ainsi que la charge à la date d'inventaire concernée mais également leurs dates de survenance, d'ouverture et de clôture ou de réouverture ainsi que l'état du sinistre. Les sinistres considérés dans notre base n'étant pas nécessairement clos à la dernière date d'inventaire. A ce stade de l'extraction de la base de sinistres, il est possible de constituer des triangles de liquidation en vue d'utiliser une des méthodes agrégées vues précédemment. Dans le cadre de ce mémoire, d'autres informations propres au véhicule sinistré, au sinistre, ou encore au profil des victimes ont été récupérées et associées à nos sinistres. A l'issue de ces étapes d'extraction, et après avoir retiré certaines observations incohérentes, nous obtenons une importante base de données composée de 5 020 546 sinistres , toutes garanties confondues, observés à 10 dates d'inventaire différentes.

Nous disposons également d'une base de données constituée des montants de charge cumulées par année de survenance pour les 10 années d'observation pour chaque branche d'activité. Il est donc possible à l'aide de la librairie *ChainLadder* de R de recréer les triangles de charges associés.

2.2 Présentation de la base

La base de données issue de l'extraction des sinistres contient 4 catégories de variables :

Variables d'informations financières

- REG_J : Les règlements bruts de recours effectués pour un sinistre au 31/12/J

- RES_J : Le montant de réserves pour un sinistre au 31/12/J
- CHG_J : Le montant cumulé de charges D/D pour un sinistre au 31/12/J, il s'agît de la somme des règlements et des réserves
- CHG : La charge observée à la dernière date d'inventaire connue
- $EVRGUP_{OUV}$ **ou** RES_{OUV} : La réserve d'ouverture, il s'agît d'une première évaluation du coût du sinistre

Informations temporelles

- $DTSURV$: Date de survenance du sinistre
- $DTOUV$: Date d'ouverture du sinistre
- $DTCLOT$: Date de clôture d'un sinistre, si le sinistre est clos
- $DTREOUV$: Date de réouverture d'un sinistre, que l'on peut généralement constater lors de l'aggravation d'un préjudice par exemple.
- $DUREE$: Durée du sinistre, elle est exprimée en jours comme ceci :

$$Durée = \frac{DTCLOT - DTOUV}{365,25}$$

Informations sur le sinistre

- Dep : Département de survenance du sinistre
- $AlcoholDrugTestFlag$: Présence ou non de test d'alcoolémie suite au sinistre
- $TYPSIN$: Type de sinistre
- $FireOfNewVehicleFlag$: Incendie d'un nouveau véhicule ou non
- $HightheftLevelFlag$: Préjudice important ou non lors du vol du véhicule
- $VehicleSituation$: Situation du véhicule au moment du sinistre
- $NBRQUART$: Taux de responsabilité de l'assuré lors du sinistre, codifié en 0,2,4 correspondant respectivement aux sinistres non responsables, à responsabilité partagée, et responsables.

Informations sur le véhicule

- $Model$: Modèle du véhicule

- $Make$: Marque du véhicule
- $VehicleEngine$: Moteur du véhicule
- $IRSALicensePlateIP$: Plaque d'immatriculation
- $IRSALicensePlateTP$: Plaque d'immatriculation du véhicule tiers
- $Style$: Style du véhicule
- $Usage$: Type d'utilisation
- $VehicleType$: Type de véhicule
- $YearId$: Année du véhicule
- $RSAClass$: Classe SRA du véhicule
- $RSAGroup$: Groupe SRA du véhicule

Les variables SRA [1] propres aux véhicules permettront de résumer les informations sur ceux-ci. SRA (Sécurité et Réparation Automobile) est un organisme professionnel fondé en 1977. Toutes les sociétés d'assurance automobiles françaises y sont adhérentes. Un fichier recensant toutes les caractéristiques techniques et commerciales des véhicules afin de permettre leur identification a été créé par cet organisme. Il concerne «l'ensemble des véhicules terrestres à moteur de moins de 3,5 tonnes destinés au marché français et commercialisés par un constructeur ou un importateur officiel».

Le groupe SRA d'un véhicule représente sa puissance, plus cet indice est grand, plus le véhicule est puissant. Cet indicateur est important pour un assureur car il entre en compte dans l'estimation des dommages que peut infliger un véhicule selon sa puissance ainsi que sa dangerosité.

La classe SRA utilisée ici est la classe de réparation qui, notée de A à Z5, représente le coût des réparations d'un véhicule suite à un accident. Un véhicule est attribué à une classe en fonction de la valeur de ses pièces lors de la réparation. Il existe également une classe SRA en fonction de la valeur à neuf TTC du véhicule, utilisée notamment en cas de vol.

Informations sur la gestion du sinistre

- *Litige* : Présence de contestation ou non lors de la proposition d'indemnisation à l'assuré

[1] www.index-assurance.fr/pratique/devis-souscription/code-sra-dune-voiture-avec-groupe-classe-cnit-ou-type-mines

- CAT : Catégorie du sinistre, selon qu'il soit considéré par AXA France comme attritionnel ou grave
- $REOUV$: Si le sinistre a été rouvert. Variable créée à partir de la variable $DTREOUV$

Informations sur les victimes

A partir d'un registre de victimes, il a été possible de créer de nouvelles variables propres à la sinistralité corporelle, qui nous donne des informations sur les victimes impliquées dans un accident de la circulation.

- $NBVICTCORP$: Nombre de sinistrés blessés lors de l'accident
- $NBVICTGRAVECORP$: Nombre de sinistrés grièvement blessés lors de l'accident
- $NBVICTDECE$: Nombre de victimes décédées lors de l'accident
- $NBPIETONS$: Nombre de piétons impliqués dans l'accident
- $NBpassa$: Nombre de passagers dans le véhicule lors de l'accident
- $NBcycl$: Nombre de cyclistes impliqués dans l'accident
- $NBmineurs$: Nombre de mineurs au moment de l'accident
- $NBseniors$: Nombre de seniors (plus de 70 ans) au moment de l'accident

Informations sur le développement du sinistre

Nous observons chaque sinistre tous les ans depuis son ouverture. A partir de ces informations, il est possible de constater des évolutions de charge et de règlements. Cependant, plus un sinistre est récent, moins nous disposons d'informations sur son développement sur le long terme. Nous calculons pour chaque observation les nouvelles variables de développement suivantes :

- λ_1^{charge} : Premier coefficient de développement individuel de charge, il s'agît du rapport entre la charge d'un sinistre observée lors de sa seconde date d'inventaire et la charge de ce même sinistre observée lors de sa première date d'inventaire.
- $\lambda_1^{règlement}$: Premier coefficient de développement individuel de règlement, il s'agît du rapport entre le règlement d'un sinistre observé lors de sa seconde date d'inventaire et le règlement de ce même sinistre observé lors de sa première date d'inventaire.

- DIP_{Charge} **ou** $DIP_{Règlement}$: Développements Individuels Pondérés créés à partir des développements de chaque sinistre tels que :

  Pour $i \in (1, ..., n)$, avec i le $i^{ème}$ sinistre et n le nombre de sinistres :

$$DIP_i = \lambda_1 + \sum_{j=2}^{J} j \times (\lambda_j - \lambda_{j-1}) \mathbf{1}_{\{y+j \leq 2019\}}$$

  Avec J la dernière période de développement et y, l'année de survenance du $i^{ème}$ sinistre.

Ces derniers coefficients de développement (DIP) ont pour but de caractériser les développements des sinistres. Ils permettent de prendre en compte les développements dans le temps. Ces variables de développement ne sont donc pas disponibles pour les nouveaux sinistres, survenus à partir de l'année de dernière date d'inventaire.

2.3 Base d'étude

Nous nous concentrerons dans un premier temps sur la sinistralité corporelle afin de comparer la segmentation effectuée par tranches de coûts de la segmentation issue de modèles de classification en K classes d'observations. L'étude sera par la suite élargie et appliquée à des garanties matérielles. Dans le cadre de la sinistralité corporelle nous ne conserverons pas les variables suivantes :

Les variables véhicule : Les informations sur le véhicule contiennent près de 79% de valeurs manquantes dans notre sous-échantillon de la base de sinistres, de plus, les informations disponibles sont généralement concentrées sur les années de survenance récentes, un retraitement de la base de données sera alors nécessaire lorsque le processus de segmentation sera appliqué à une garantie matérielle.

FireOfNewVehicleFlag, HightheftLevelFlag : Variables non concernées par la sinistralité corporelle, mais utiles lorsque l'on s'intéresse a la sinistralité incendie/vol de véhicules.

Informations temporelles : Nous conserverons seulement la durée, qui est calculée grâce aux dates d'ouverture et de clôture du sinistre, ainsi que l'indicateur de ré-ouverture ou non d'un sinistre.

2.4 Statistiques descriptives

Nous nous intéressons à la distribution de la charge de sinistres pour la branche responsabilité civile corporelle en regard des autres variables de notre base d'étude. Il sera également

intéressant d'observer les différentes durées de gestion du sinistre que l'on observe dans notre portefeuille et d'étudier les corrélations entre ces deux informations.

2.4.1 Distribution de la variable Charge

Nous observons tout d'abord la distribution de la charge des sinistres à la dernière date d'inventaire à travers l'histogramme suivant :

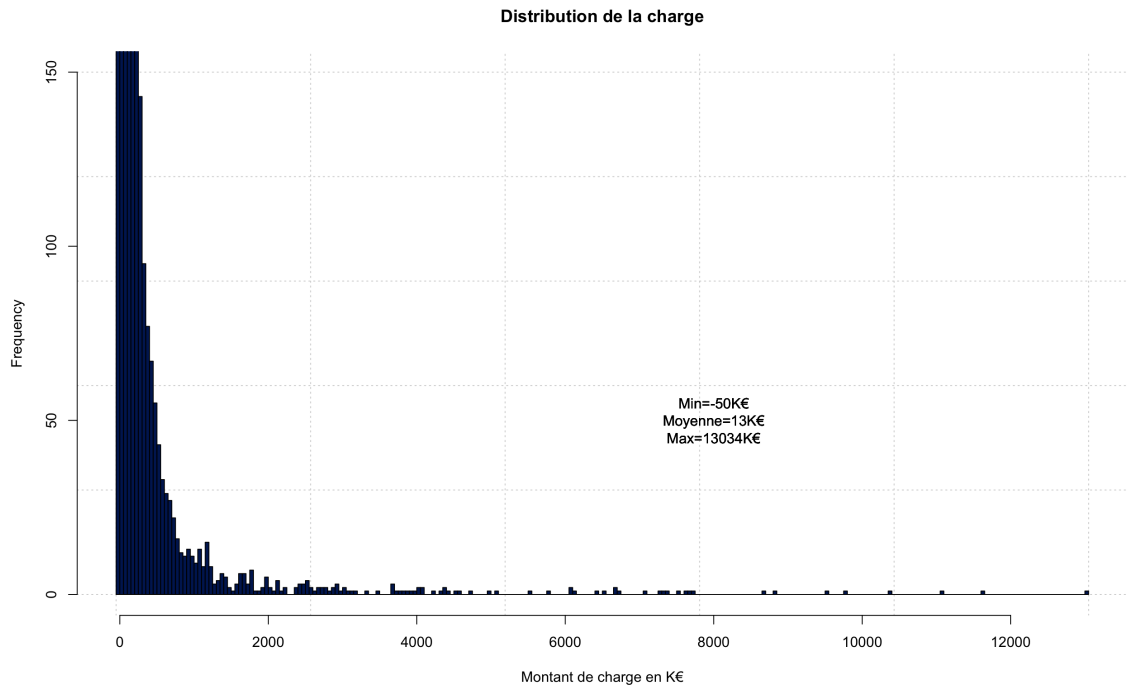

Figure 9 – Distribution de la charge de sinistres corporels

Une majeure partie de la charge est concentrée autour de 0, ce qui peut être en partie expliqué par la part des sinistres non responsables de notre portefeuille. Les montants de charges négatifs sont engendrés par des montants de recours supérieurs aux charges totales. La charge moyenne de sinistres est de 12 526€.

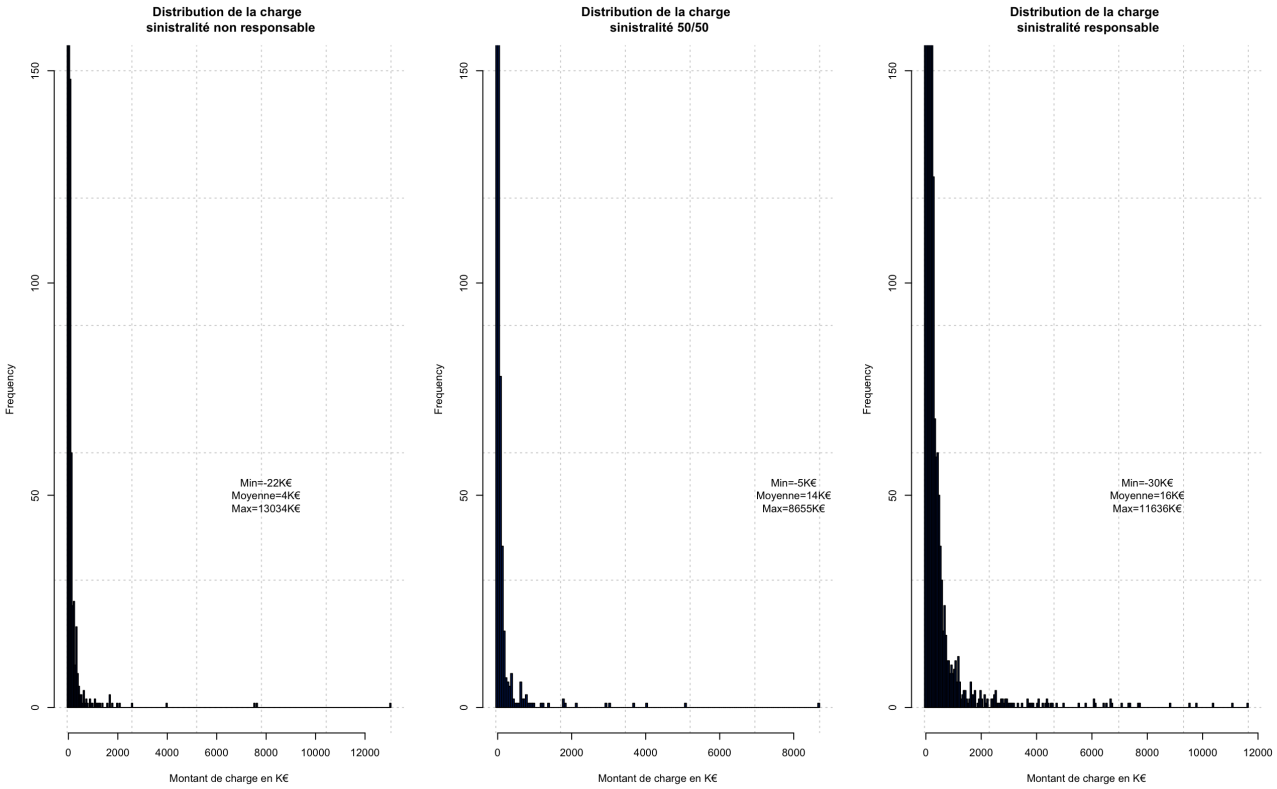

Figure 10 – Distribution de la charge de sinistres corporels selon la responsabilité

En observant les distributions de charges selon la responsabilité de l'assuré, on observe logiquement une hausse globale de la charge. Nous continuerons l'analyse descriptive de la charge en fonction d'autres indicateurs. Afin de mieux visualiser l'impact de la présence ou non de litige, de l'état du sinistre ou encore du nombre de victimes blessées lors de l'accident sur la charge, nous choisissons de les représenter à travers des *boxplots* (figures 11,12 et 14).

Charge de sinistres selon la variable Litige :

Lorsque la décision d'indemnisation est refusée par la victime d'un sinistre, un litige est généralement ouvert, pouvant entraîner une gestion plus longue des sinistres et plus généralement une hausse globale de la charge. Les sinistres avec la présence d'un litige sont classés comme sinistres contentieux et représentent 1,9% de la sinistralité soit 2393 sinistres.

Bien que peu nombreux, les sinistres non réglés à l'amiable constituent à eux seuls 323M€. Pour rappel, la charge de sinistre totale de notre portefeuille à la dernière date d'inventaire était de 1 608M€. Ces sinistres représentent ainsi près de 20% de la charge globale.

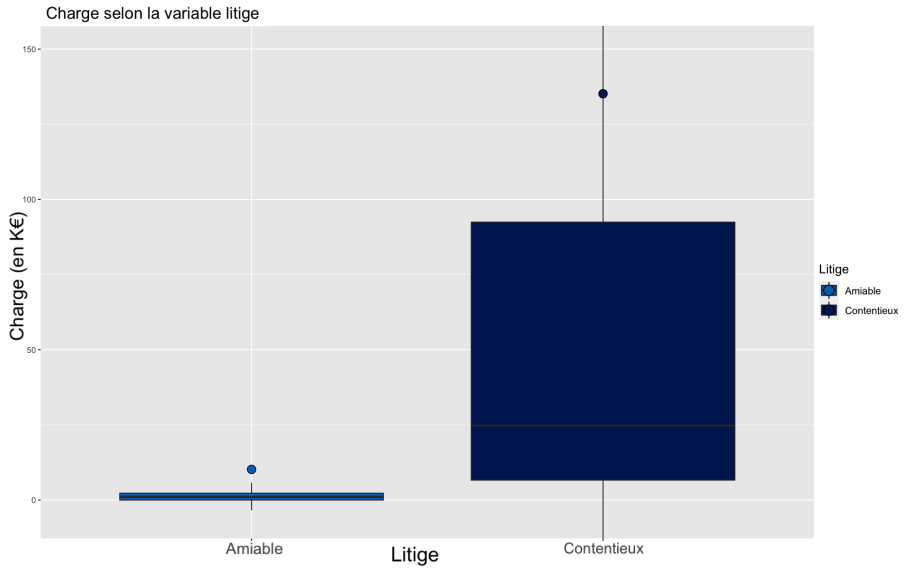

Figure 11 – Charge de sinistres selon la variable Litige

La charge moyenne des sinistres est plus importante lorsque le sinistre est en traitement contentieux. Nous pouvons donc nous attendre à une forte significativité de cette variable dans nos modèles.

Charge de sinistres selon l'état :

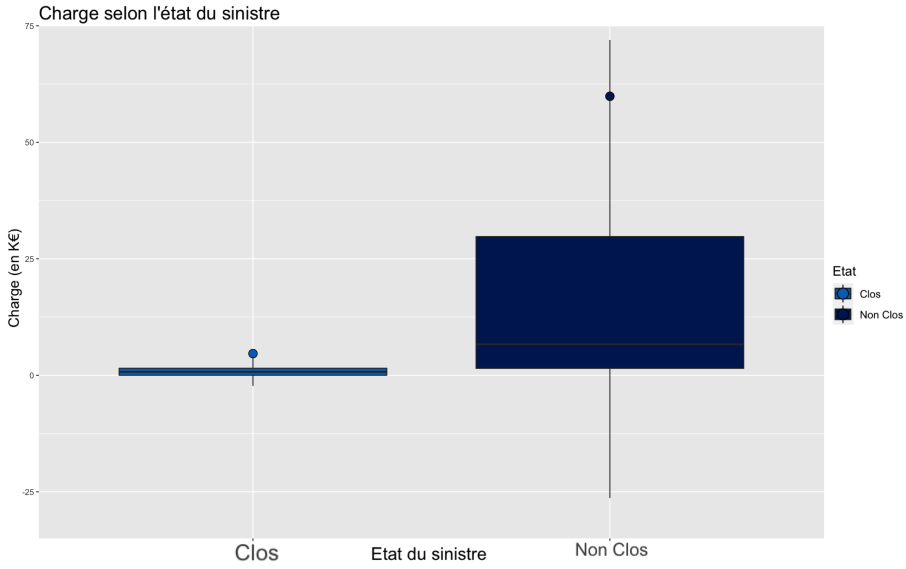

Figure 12 – Charge de sinistres selon son état

Certains sinistres de notre base d'étude ne sont pas encore clos, entraînant ainsi un phénomène de censure. 18 299 sinistres soit 14% des sinistres de notre portefeuille ne sont pas clos, la charge

moyenne de ces sinistres est de près de 60K€ tandis que la charge moyenne des sinistres clos est de près de 5K€. Ne pas considérer ces sinistres entraînerait donc un biais évident lors de la modélisation des classes de sinistralité.

2.4.2 Distribution de la durée de gestion des sinistres

Nous observons ensuite la distribution de la durée en jours des sinistres entre son ouverture et sa clôture si elle est connue à la dernière date d'inventaire à travers l'histogramme suivant :

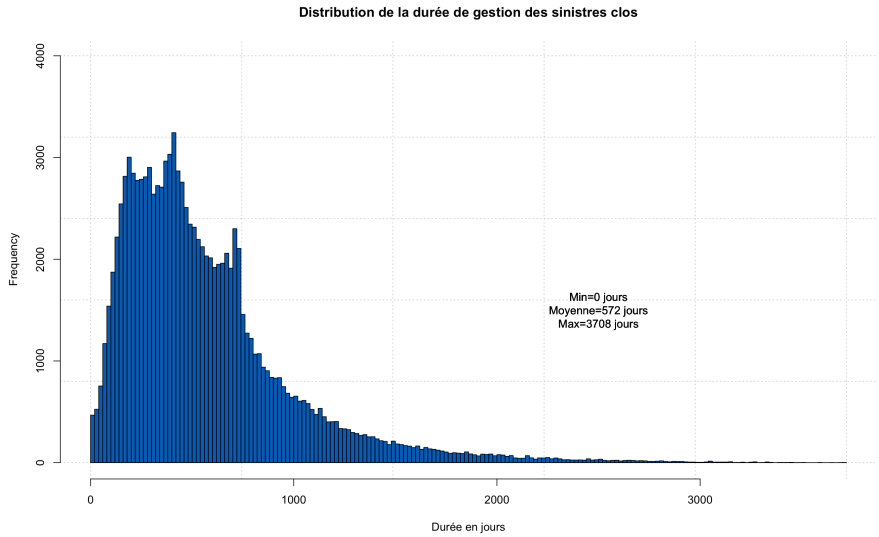

Figure 13 – Distribution de la durée de gestion des sinistres

La durée moyenne des sinistres clos est de 572 jours. La grande majorité des sinistres sont réglés dés la première année de développement.

Durée de sinistres selon la variable Litige :

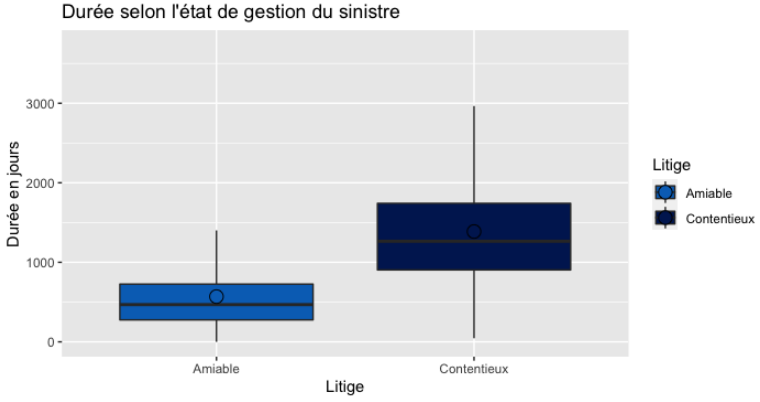

Figure 14 – Durée de gestion des sinistres selon la présence de litige ou non

La durée des sinistres est sans surprise impactée par la présence d'un contentieux au cours de la gestion. La durée moyenne passe ainsi de 567 jours à 1383 jours.

Durée de sinistres selon le nombre de victimes :

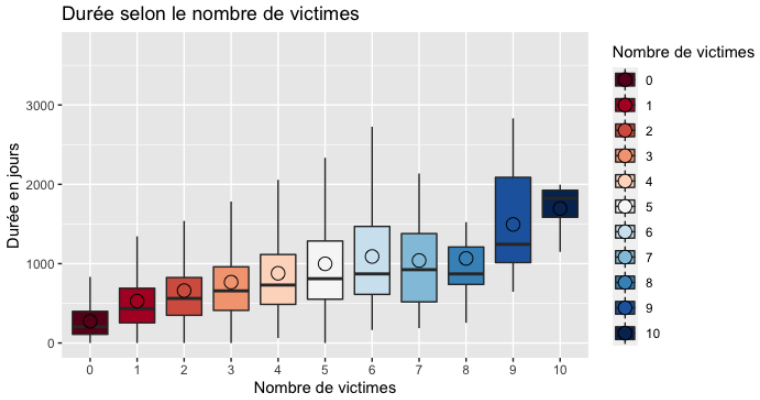

Figure 15 – Durée selon le nombre de victimes blessées

Le nombre de victimes blessées lors de l'accident a également un effet sur la durée. En moyenne, plus le nombre de victimes est grand, plus la durée de gestion d'un sinistre est longue. On n'observe cependant pas ce phénomène pour les sinistres ayant causé 7 ou 8 victimes, ce qui est lié à leur faible nombre dans notre portefeuille (0,08% et 0,03%), entraînant une forte volatilité de la moyenne des durées.

2.5 Analyse des corrélations des variables sélectionnées

Afin d'identifier les relations qu'il y a entre nos différentes variables, on s'intéresse aux corrélations entre celles-ci. Certains modèles peuvent être biaisés lorsqu'ils sont appliqués à des variables interdépendantes. Cette section permettra de déterminer l'existence d'interdépendances entre les variables en étudiant leurs corrélations.

2.5.1 Étude graphique des corrélations des variables quantitatives

Coefficient de corrélation de Pearson

On rappelle la définition du coefficient de corrélation entre deux variables X et Y prenant respectivement les valeurs $(x_i)_{i=1...n}$ et $(y_i)_{i=1...n}$:

$$\rho_{XY} = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y}$$

On calcule l'estimateur sans biais $\hat{\rho}_{XY}$ de ρ_{XY} :

$$\hat{\rho}_{XY} = \frac{\widehat{\sigma}_{XY}}{\hat{\sigma}_X \hat{\sigma}_Y}$$

Avec σ_X et σ_Y les écarts-types respectifs des variables X et Y :

$$\hat{\sigma}_X = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} \text{ et } \hat{\sigma}_Y = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

et $\widehat{\sigma}_{XY}$ la covariance empirique de X et Y :

$$\hat{\sigma}_{XY} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}$$

ρ_{XY} est compris entre -1 et 1 et représente une forte corrélation positive lorsqu'il est compris entre 0,5 et 1 et une forte corrélation négative lorsqu'il est compris entre -1 et 0,5.

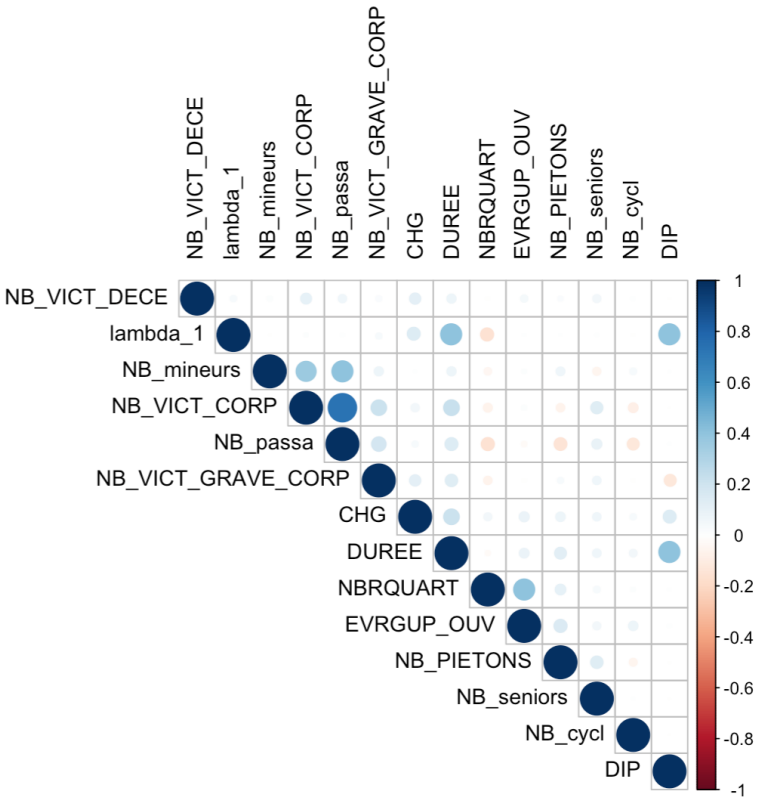

Figure 16 – Corrélogramme des variables quantitatives

On peut constater sur le corrélogramme une interdépendance logique entre le nombre de passagers ainsi que le nombre de blessés. Le coefficient de corrélation ρ est de 0,74.

2.5.2 Étude graphique des corrélations des variables quantitatives

Afin d'étudier la corrélation entre nos différentes variables quantitatives, nous nous baserons sur la mesure du V de Cramer qui se base sur le test d'indépendance du χ^2. Le test du χ^2 permet de déterminer l'existence d'une relation entre deux variables qualitatives. Lorsque les variables sont quantitatives, celles-ci sont considérées comme des variables qualitatives avec un nombre important de modalités. Cette valeur ne quantifie pas la dépendance entre deux variables. La mesure du χ^2 est comprise entre 0 et $+\infty$. La mesure du V de Cramer est elle comprise entre 0 et 1.

Le test d'indépendance du χ^2

Le test du χ^2 a pour but de vérifier l'hypothèse suivante : «H_0 : X et Y sont indépendantes». Afin de valider cette hypothèse, il faut vérifier que la statistique du χ^2 suit une distribution du χ^2 avec une probabilité α. Cette statistique quantifie l'écart entre les effectifs observés et théoriques. On la calcule ainsi :

On observe pour les variables qualitatives X et Y respectivement p et q modalités avec $(p, q \geq 2)$. Soit le tableau des effectifs suivant :

	Y_1	Y_2	...	Y_q	Total
X_1	$n_{1,1}$	$n_{1,2}$	...	$n_{1,q}$	$n_{1,.}$
X_2	$n_{2,1}$	$n_{2,2}$	...	$n_{2,q}$	$n_{2,.}$
...	...	...	...	...	...
X_p	$n_{p,1}$	$n_{p,2}$	...	$n_{p,q}$	$n_{p,.}$
Total	$n_{.,1}$	$n_{.,2}$	...	$n_{.,q}$	n

Table 1 – Effectif des modalités de X et Y

Alors,

$$\chi^2 = \sum_{p,q} \frac{\left(n_{p,q} - \frac{n_{p,.}, n_{.,q}}{n}\right)^2}{\frac{n_{p,.}, n_{.,q}}{n}}$$

Si $\chi^2 > \chi^2_{1-\alpha}[(p-1)(q-1)]$, le quantile d'ordre $1-\alpha$ de la loi du χ^2 à $(p-1)(q-1)$ degrés de liberté, alors on rejette H_0. En général, on choisira $\alpha = 5\%$, valeur délimitant la région critique du test.

Le V de Cramer

Le V de Cramer permet de normaliser la valeur de la statistique de test du χ^2 et de quantifier la dépendance entre deux variables qualitatives. Cette mesure s'exprime ainsi :

$$V = \sqrt{\frac{\chi^2}{n(\min(p-1, q-1))}}$$

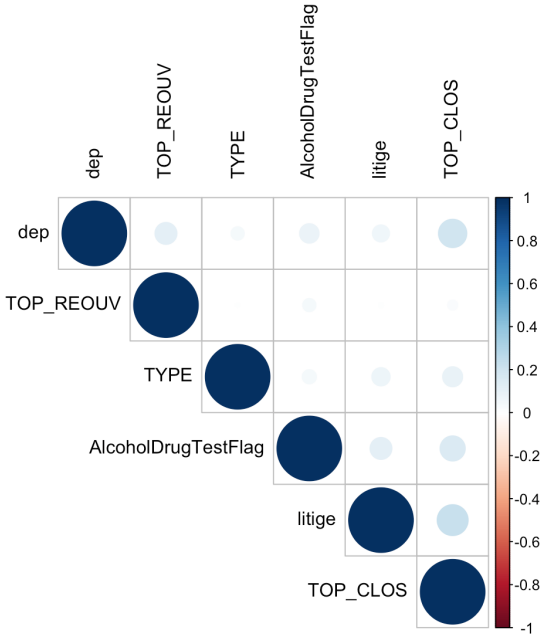

Figure 17 – Corrélogramme des variables qualitatives

3 Classification des sinistres

A partir de l'ensemble de la sinistralité corporelle de notre étude, nous décidons de créer des classes homogènes à partir des caractéristiques individuelles des sinistres. L'objectif sera d'identifier des groupes d'observations ayant des caractéristiques similaires. Nous souhaitons ainsi regrouper les sinistres ayant les mêmes comportements (d'aggravation, ou de cadence de règlements) dans les mêmes groupes tout en créant des groupes les plus différents possibles entre eux. Le but de cette démarche étant d'identifier les structures sous-jacentes dans nos données, résumer des comportements et pouvoir par la suite affecter des nouveaux sinistres à ces nouvelles catégories.

3.1 Classification non supervisée

Afin de réaliser le partitionnement de nos observations en classes, nous utiliserons les algorithmes de création de groupes homogènes *K-Means*, *K-Modes* et *K-Prorotypes*. Il s'agît donc de classification non supervisée dans la mesure où les classes ne sont pas encore déterminées. L'intérêt du *K-Prototype* consiste à effectuer une classification sur un jeu de données comportant à la fois des variables quantitatives et qualitatives. Nous détaillerons dans les sections suivantes les algorithmes utilisés en vue de créer un nombre de classes pouvant être du même ordre que le nombre de tranches utilisées par AXA France.

3.1.1 Algorithme *K-Means*

Le *clustering K-Means* est une méthode permettant de trouver des clusters ainsi que des observations représentant les centres de ces clusters dans les données. En choisissant un nombre C de centres appelés centroïdes, l'algorithme *K-Means* assigne chaque observation aux centroïdes en minimisant l'intravariance totale. A partir d'un échantillon initial de centroïdes, cet algorithme effectue deux étapes alternées :

- Identification à partir de chaque centroïde des observations plus proches que de n'importe quel autre centroïde.

- Pour chaque nouvelle observation associée à un cluster, calcul de la moyenne des observations du cluster afin d'obtenir son nouveau centre.

Ces deux étapes sont répétées jusqu'à convergence. Les C centroïdes initiaux sont généralement choisis aléatoirement.

L'algorithme *K-Means* est réputé pour pouvoir être performant dans le cadre du *clustering*

sur des jeux de données à variables quantitatives. Cet algorithme incrémental associe à chacune des observations un groupe selon la notion de similarité. Nous détaillons l'algorithme de *K-Means* comme suit :

Algorithme 1 : *K-Means*

Initialisation :

-Choisir les variables quantitatives avec lesquelles effectuer le *clustering*

-Sélectionner un nombre C d'observations où C=Nb de clusters

tant que *Individus réaffectés à des nouveaux groupes* **faire**

> **Étape 1 :**
>
> -Calcul de la distance euclidienne de chaque observation avec les centroïdes
>
> **Étape 2 :**
>
> -Affectation de chaque observation au centroïde le plus proche
>
> **Étape 3 :**
>
> -Calcul du centre de gravité des groupes créés, devenant les nouveaux centroïdes

fin

Résultat : C classes d'observations

3.1.2 Algorithme *K-Modes*

L'algorithme *K-Modes* se base sur le principe du *K-Means* mais utilise des variables qualitatives pour affecter les observations aux classes.

Cet algorithme utilisera la notion de dissimilarité entre une observation et un centroïde, ici appelé mode.

La mesure de dissimilarité s'exprime ainsi :

$$D\left(x_{j,f}, y_{i,f}\right) = \frac{\left(m_{x_{j,f}} + m_{y_{i,f}}\right)}{\left(m_{x_{j,f}} \times m_{y_{i,f}}\right)} \times \delta\left(x_{j,f}, y_{i,f}\right)$$

Avec :

f la $f^{ième}$ variable et $f \in (1, ..., F)$

$x_{j,f}$, la valeur de la $j^{ième}$ observation pour la $f^{ième}$ variable

$y_{i,f}$ la valeur de la $i^{ième}$ observation pour la $f^{ième}$ variable

$m_{x_{j,f}}$ le nombre de fois qu'apparaît la valeur de $x_{j,f}$ dans l'ensemble des modes

$m_{y_{i,f}}$ le nombre de fois qu'apparaît la valeur de $y_{i,f}$ dans les modes

Enfin, on a

$$\delta\left(x_{j,f}, y_{i,f}\right) = \begin{cases} 0 \text{ si } x_{j,f} = y_{i,f} \\ 1 \text{ si } x_{j,f} \neq y_{i,f} \end{cases}$$

Le processus des *K-Modes* peut être présenté comme ceci :

Algorithme 2 : *K-Modes*

Initialisation :

-Choisir les variables qualitatives avec lesquelles effectuer le *clustering*

-Trier les modalités de chaque variable en fonction de leur fréquence d'apparition (du plus grand nombre au plus petit).

-Sélectionner un nombre C d'observations où C=Nb de clusters

pour *i=1 à n* **faire**

  pour *f=1 à F* **faire**

    pour *j=1 à k* **faire**

      Étape 1 :

      -Calculer la dissimilarité entre une observation et le mode j pour la variable f

    fin

    Étape 2 :

    -Calcul de $\bar{D_j} = \frac{1}{F} \sum_{f=1}^{F} D_{j,f}$

  fin

  Étape 3 :

  -Associer l'observation i au mode dont la mesure de dissimilarité est la plus faible.

fin

Étape 4 :

-Calculer la fonction de coût associée à ce processus :

$$C(Q) = \sum_{j=1}^{k} \sum_{i=1}^{n} \sum_{f=1}^{F} \delta\left(x_{j,f}, y_{i,f}\right)$$

Résultat : C classes d'observations, Fonction de coût associée au nombre C de clusters.

Pour chaque observation $y_i (i \in n)$ avec n le nombre d'observations, on a les modes $x_j (j \in k)$ avec k le nombre de clusters.

3.1.3 Algorithme *K-Prototype*

L'algorithme *K-Prototype* est un algorithme de classification non supervisée pour les données mixtes, nous l'utiliserons afin de créer les classes de sinistralité de notre portefeuille. Les étapes de cet algorithme sont les suivantes :

Algorithme 3 : *K-Prototype*

Initialisation :

-Choisir les variables qualitatives et quantitatives avec lesquelles effectuer le *clustering*

-Trier les modalités de chaque variable en fonction de leur fréquence d'apparition (du plus grand nombre au plus petit) pour les variables qualitatives

-Trier les variables quantitatives dans l'ordre croissant

-Sélectionner un nombre C d'observations où C=Nb de clusters, ces observations sont les modes ou centroïdes.

pour *i=1 à n* **faire**

  pour *f=1 à F* **faire**

    pour *j=1 à k* **faire**

      Étape 1 :

      -Calculer la dissimilarité entre une observation et le mode j pour la variable f si elle est qualitative

      -Calculer la distance euclidienne entre une observation et le centroïde j pour la variable f si elle est quantitative

    fin

    Étape 2 :

    -Calcul de $\bar{D_j} = \frac{1}{F}\sum_{f=1}^{F} D_{j,f}$

    -Calcul de $\bar{d_j} = \frac{1}{F}\sum_{f=1}^{F} d_{j,f}$

  fin

  Étape 3 :

  -Calculer $\bar{d_j} + \bar{D_j}$ pour chaque mode/centroïde et associer l'observation i au mode/centroïde avec la valeur la plus faible.

fin

Étape 4 :

-Calculer la fonction de coût associée à ce processus

-Calculer l'erreur quadratique pour les variables quantitatives

Résultat : C classes d'observations, Fonction de coût associée au nombre C de clusters, erreur quadratique E

3.2 Partitionnement de notre portefeuille en classes homogènes

3.2.1 Analyse du nombre de classes optimal

Nous décidons d'utiliser ces trois algorithmes de *clustering* sur notre base de données, dans le but de déduire différentes classes pour les sinistres corporels. Les méthodes seront appliquées aux variables qualitatives et quantitatives de notre base. Nous étudierons dans cette sous-section les différentes possibilités en terme de nombre de classes qui peuvent se présenter.

Nous mesurons les intravariances au sein de chaque classe de sinistralité obtenues par ces trois algorithmes. L'objectif principal commun à ces trois algorithmes est la maximisation de l'homogénéité au sein des classes et de l'hétérogénéité entre-elles. Afin d'optimiser le nombre de classes nécessaires à la nouvelle segmentation, nous faisons varier K pour les 3 méthodes :

- ***K-Means*** : Sur les variables quantitatives avec les variables temporelles de développement
- ***K-Modes*** : Sur les variables qualitatives
- ***K-Prototype*** : Sur l'ensemble des variables avec les variables temporelles de développement

Nous obtenons les mesures d'intravariances suivantes pour $K \in 1, ..., 20$:

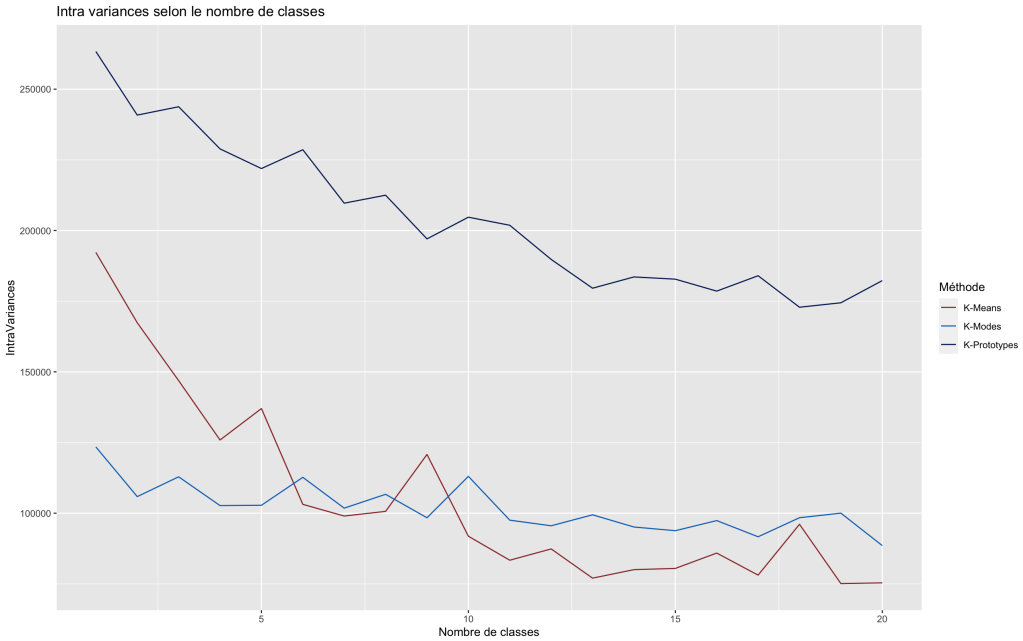

Figure 18 – Évolution des intravariances en fonction du nombre de classes

Les intravariances diminuent fortement pour les deux algorithmes utilisant des variables qualitatives lorsque K est compris entre 1 et 10. On observe une stabilisation au-delà. Afin d'obtenir

des classes homogènes et avec suffisamment d'observations, nous retiendrons $K = 9$ classes pour notre nouvelle segmentation. Nous décrirons dans la partie 3.2.3 le processus de validation sur l'erreur des flux de charges obtenue avec ce nombre de classes.

3.2.2 Analyse de la nouvelle segmentation

Les $K = 9$ classes de sinistralité créées, nous pouvons visualiser leur répartition au sein de notre portefeuille global :

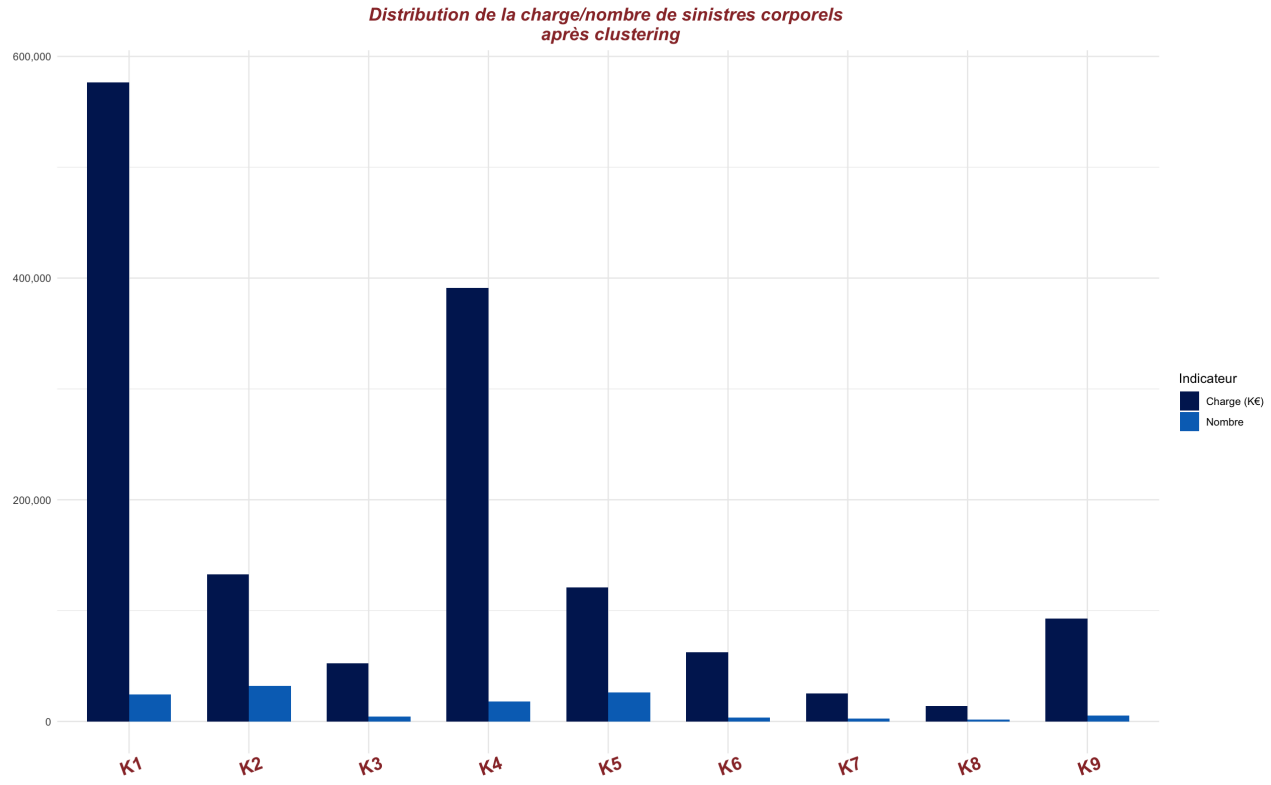

Figure 19 – Répartition de la charge de sinistre et du nombre d'observations par classes

Nous pouvons désormais nous intéresser aux estimations de charges ultimes sur les tranches de coûts utilisées par AXA France représentant 3 triangles de charges et notre nouvelle base de segmentation représentant 9 triangles de charges. Afin de pouvoir comparer les différentes performances et challenger la segmentation actuelle, nous effectuerons divers tests de validation sur les années de développement les plus récentes.

3.2.3 Estimation des flux de charge pour les dernières années de développement

Afin de mesurer les erreurs d'estimation des deux segmentations, nous appliquons la méthode *Chain-Ladder* de charge sur les tranches T1 à T3 ainsi que sur les nouvelles classes obtenues K1 à K9, en retirant les trois dernières années de développement dans le but de pouvoir effectuer

un test de validation sur ces dernières années. Nous représentons ci-après le triangle de charges toutes tranches confondues de la branche responsabilité civile corporelle :

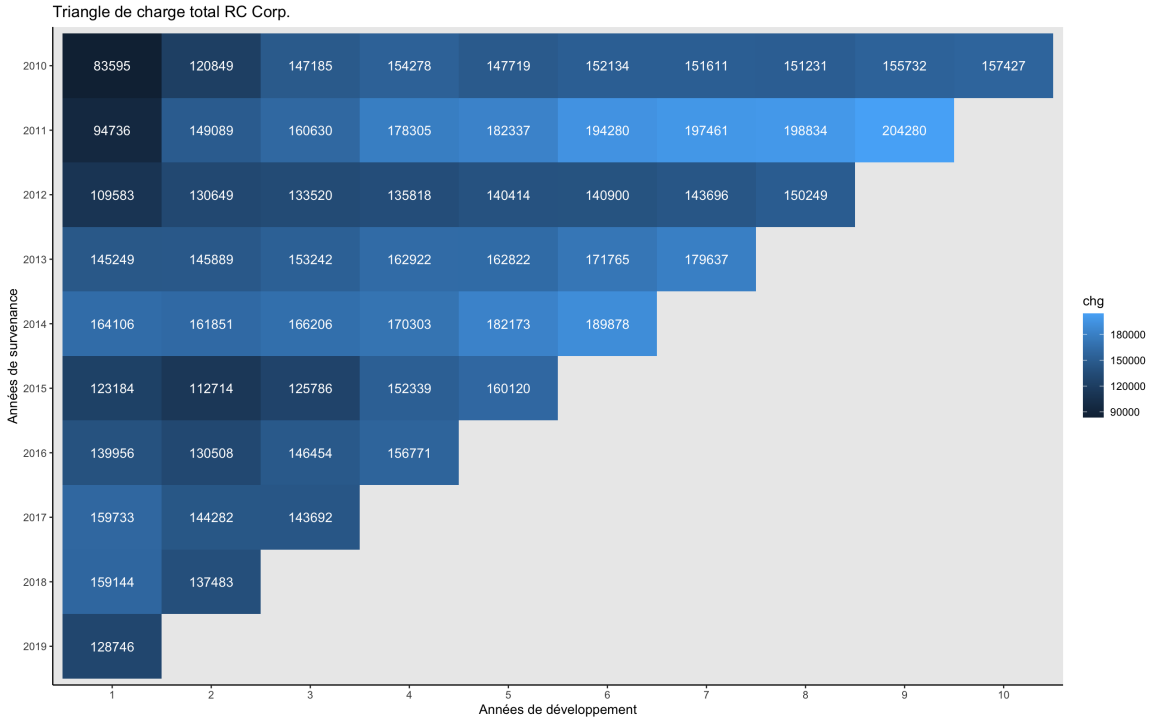

Figure 20 – Triangle de charges Responsabilité Civile corporelle

On s'intéresse ensuite à la somme des flux pour les charges des trois dernières de développement connues par année de survenance, il s'agît des trois dernières diagonales.

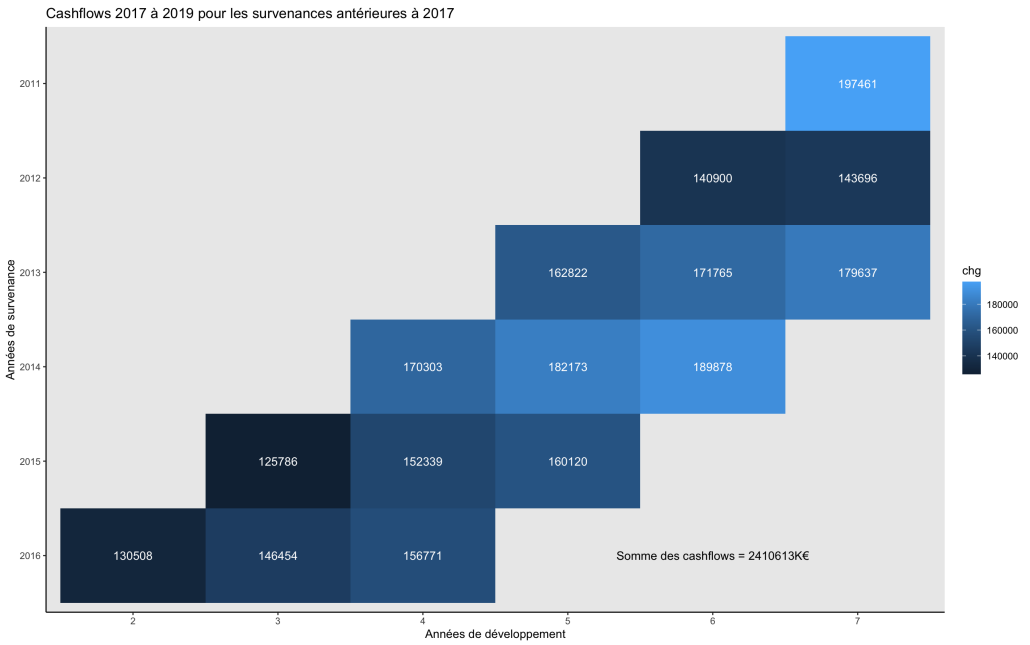

Figure 21 – Flux de charge pour les trois dernières années de survenance

Nous estimons par *Chain-Ladder* les trois triangles de tranches de coût T1 à T3 en retirant les années de survenance supérieures à 2017. En sommant les triangles des trois tranches de coût, nous obtenons des estimations de charges que nous pouvons comparer aux charges réellement observées en 2017 et 2018, en écartant 2019. Nous estimons également les charges des années retirées sur les triangles de charges de la segmentation issue du *clustering*.

Ci-dessous, les différences d'estimation de charges pour les dernières années :

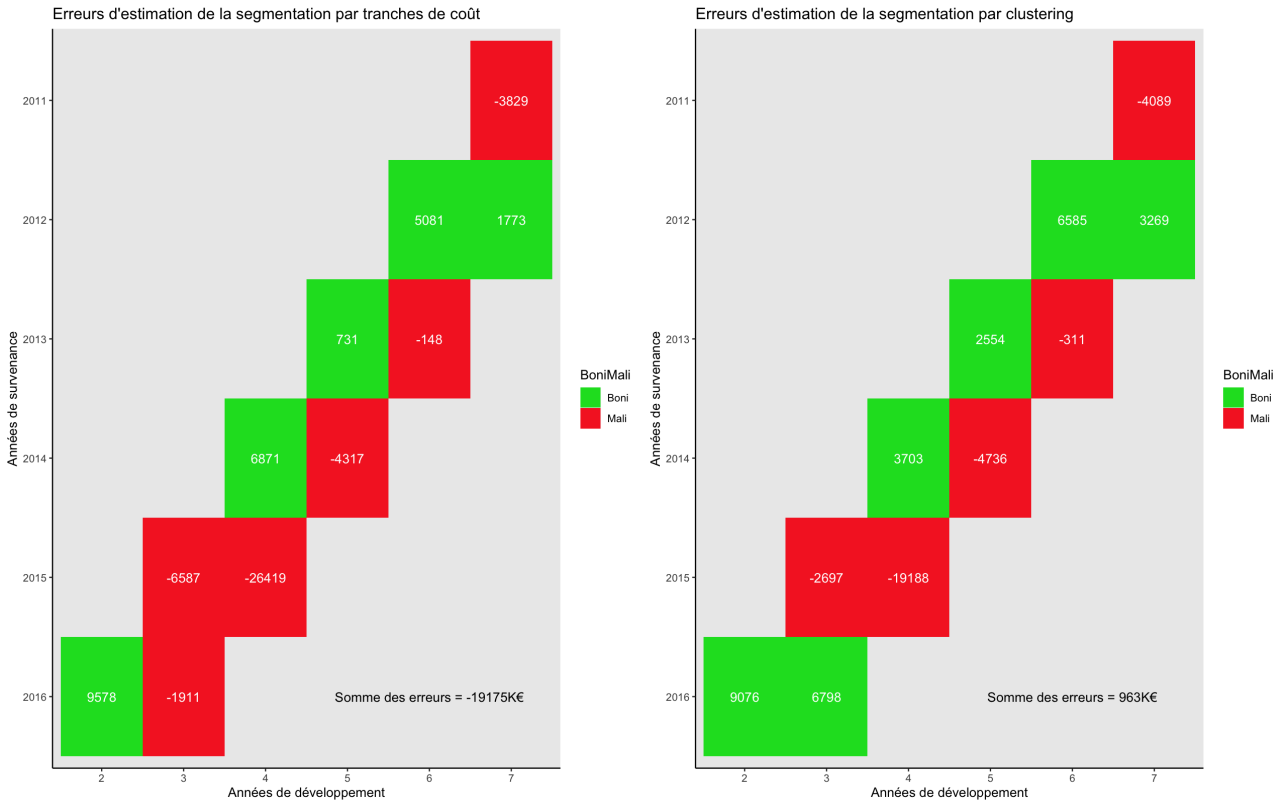

Figure 22 – Erreurs d'estimation de la segmentation par tranches de coût et par *clustering*

Nous obtenons une somme globale des boni/mali de liquidation sur l'ensemble des années de survenance inférieures avec la segmentation issue de l'algorithme *K-Prototype*. On peut noter les RMSE (*Root Mean Square Error*) pour chacune des deux segmentations de la manière suivante pour $i \in (2011, 2016)$, années de survenance et $j \in (2, ..., 7)$ correspondant aux périodes de développement :

$$RMSE_{cout} = \sqrt{\frac{\sum_i \sum_j (C_{i,j}^{Total} - \hat{C}_{i,j}^{cout})^2 \mathbf{1}_{2018 \leq i+j \leq 2019}}{\sum_i \sum_j \mathbf{1}_{2018 \leq i+j \leq 2019}}}$$
$$\approx 4395,63$$

$$RMSE_{clusters} = \sqrt{\frac{\sum_i \sum_j (C_{i,j}^{Total} - \hat{C}_{i,j}^{clusters})^2 \mathbf{1}_{2018 \leq i+j \leq 2019}}{\sum_i \sum_j \mathbf{1}_{2018 \leq i+j \leq 2019}}}$$
$$\approx 3550,15$$

On résume les premiers résultats obtenus par test de validation dans le tableau ci-dessous. Le choix du nombre de classe peut également être orienté par l'erreur effectuée sur ces deux

Méthode	Flux de charge (K€)	Écart	Écart (%)	RMSE
Observé	1 607 757	-	-	-
Tranches de coût	1 588 582	-19 175	-1,19%	4 395,63
K-Prototype	1 608 720	963	0,06%	3 550,15

Table 2 – Écarts d'estimation de provisions selon la méthode utilisée

diagonales. En effectuant un *K-Prototype* en K classes, pour $K \in 1, ..., 10$, on remarque les écarts de flux de charge suivants :

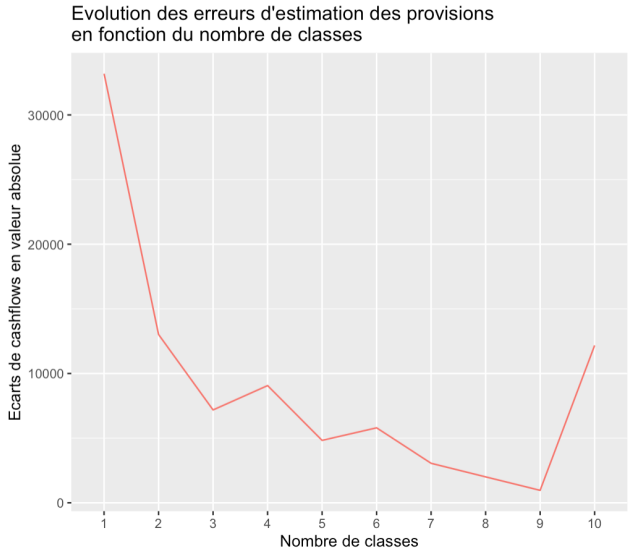

Figure 23 – Erreurs d'estimation selon le nombre de clusters

3.2.4 Analyse des Charges Finales Prévisibles

A partir des segmentations disponibles, il est possible de comparer les charges ultimes par années de survenance obtenues à partir de trois segmentations. Sur les triangles : Total, par tranches de coûts de T1 à T3 ainsi que par *clustering* $K1$ à $K9$, on observe les charges ultimes suivantes :

	2010	2011	2012	2013	2014	2015	2016	2017	2018
Charge Dossier/Dossier	155 732	198 834	143 696	171 765	182 173	152 339	146 454	144 282	159 144
Charge ultime triangle total	155 732	204 752	148 393	179 366	197 980	168 416	174 218	186 330	220 772
Charge ultime Tranches de coût	155 732	206 648	147 567	180 318	200 665	168 891	176 857	188 208	211 308
Charge ultime Clustering	155 732	200 908	151 847	182 508	204 472	167 288	179 098	193 731	201 171

Figure 24 – Charges ultimes par années de survenance à partir des trois types de segmentation

La figure 24 montre les charges ultimes des différentes méthodes comparées à la charge Dossier/Dossier.

La figure 25 permet de visualiser le développement de la charge Dossier/Dossier pour chaque année de survenance. De manière générale, estimer les provisions sur un triangle agrégé total sur-estimerait la charge ultime.

Les écarts de flux de charges apportaient un premier indicateur de performance de la méthode des *K-Prototypes*. Les estimations issues des triangles T1 à T3 sous-estimaient ces flux. Plus l'année de survenance est récente, plus l'estimation de la charge ultime est incertaine. Les estimations par tranche de coût ainsi que sur le triangle total semblent cependant plus conservatrices lors des années récentes. En effet, ne pas effectuer de segmentation au sein de la garantie Responsabilité Civile corporelle contribue au développement de certains sinistres vers des niveaux de charge beaucoup plus élevés que prévu. Les tranches de coût montrent également sur la figure 25 un développement plus important des charges pour la survenance 2018.

La principale limite de la méthode *Chain-Ladder* est la nécessité d'observer des développements similaires entre années de survenance. La sinistralité corporelle présentant la particularité d'avoir des sinistres se développant très différemment les uns des autres, un regroupement prenant compte ces caractéristiques semble important. Considérer uniquement les montants de charge de sinistre ne contribue alors plus suffisamment à l'obtention de triangles de charge ayant des développements comparables entre années de survenances.

Outre la meilleure performance obtenue sur les diagonales censurées de la figure 22, le choix de la méthode des *K-Prototypes* est clairement plus approprié dans la mesure où les développements des sinistres sont pris en compte dans le regroupement en $K = 9$ classes.

De plus, cette dernière méthode permet d'exploiter de l'information qui n'est généralement pas utilisée lors des estimations classiques de charges ultimes. Cependant, le choix de ces informations peut jouer un rôle important sur l'homogénéité des classes crées.

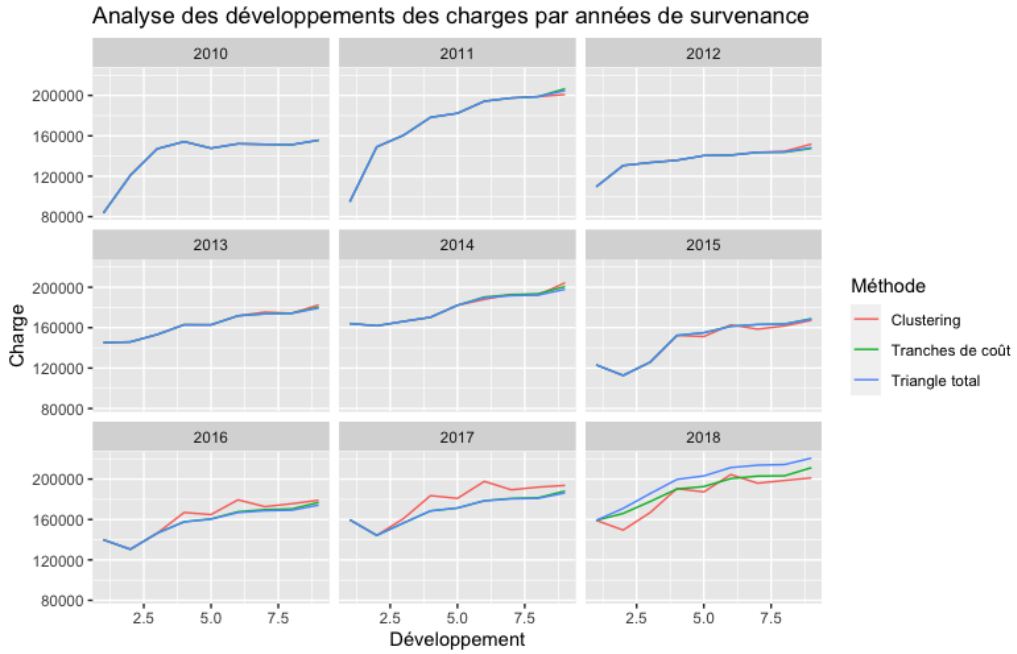

Figure 25 – Charges ultimes par années de survenance à partir des trois types de segmentation

3.2.5 Choix des variables pour la création des classes

Le modèle final de regroupement d'observations utilisé précédemment a été obtenu en utilisant l'ensemble des variables disponibles, y compris les variables créées à partir de l'évolution des informations financières des sinistres. Ce sont ces dernières variables qui ont joué un rôle important sur l'homogénéité des groupes obtenus. Il aurait été possible de créer des groupes de sinistralité sans ces variables afin de pouvoir disposer du même niveau d'information pour les sinistres anciens et récents. Cependant, la qualité d'estimation des provisions aurait fortement été impactée.

Pour $K \in (1, ..., 10)$, nous pouvons représenter en valeur absolue les montants d'erreur de prédiction qu'auraient obtenu les modèles *Chain-Ladder* classiques avec et sans les variables de développement.

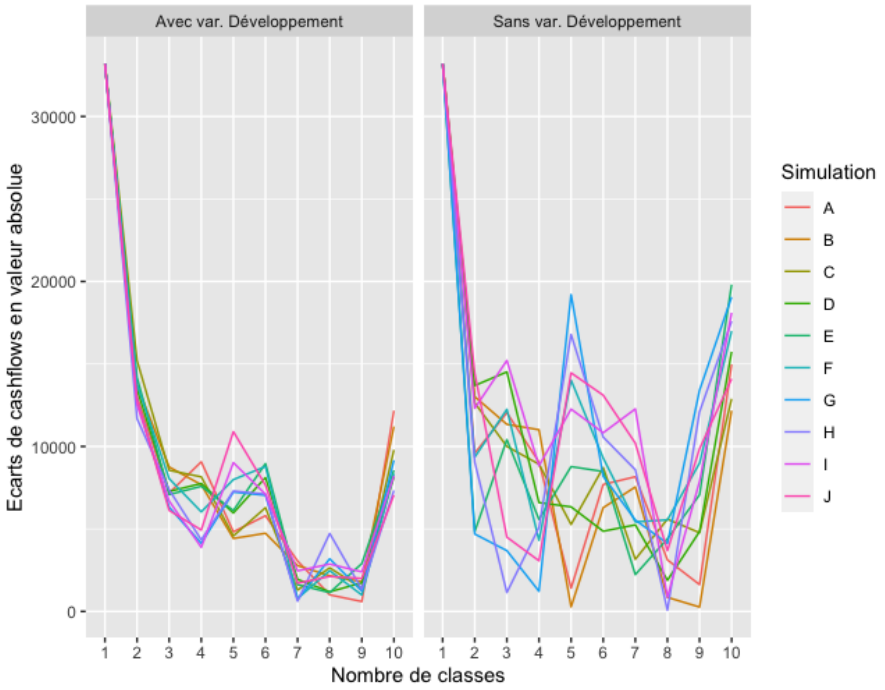

Figure 26 – Erreurs d'estimation en valeur absolue selon le nombre de classes

Comme le montre la figure 26, sans les variables de développement, l'erreur de prédiction est très instable. Certaines configurations permettent d'obtenir de bons niveaux de performance en terme d'erreur de flux de charge. Cependant, selon la simulation ces flux de charge peuvent avoir des niveaux très différents pour le même nombre de classes. Le choix initial des centroïdes lors de l'exécution des *K-Prototypes* impacte alors fortement la qualité de prédiction du *Chain-Ladder* avec ces nouvelles classes. Les variables de développement permettent d'obtenir des niveaux de performance plus constants entre les différentes simulations.

Ne pas utiliser ces variables de développement pourrait faciliter l'association des futurs sinistres aux classes créées. En effet, nous pourrions calculer les centroïdes de chaque nouvelle observation et l'associer à sa nouvelle classe de sinistralité. Cependant, nous venons de voir que l'instabilité des performances des estimations de provisions sans ces variables nous poussait à les utiliser.

Ne disposant pas des informations de développement des nouveaux sinistres, nous utiliserons des modèles de substitution afin d'estimer leur classe d'appartenance. Ces modèles utiliseront l'ensemble des caractéristiques propres aux sinistres comme variables explicatives.

4 Les modèles de substitution

4.1 Principe

Les *Surrogate models* ou modèles de substitution sont des modèles utilisés pour des problématiques issues des secteurs assurantiels, bancaires ou encore industriels (Henckaerts et al.,2020)[14]. En assurance, ces modèles ont déjà prouvé lors de l'estimation de pertes liées à des dégâts consécutifs à des inondations. Ces modèles ont permis d'assimiler des modèles complexes hydrologiques et hydrauliques (Andreas Paul Zischg et al.[20]).

Les modèles de substitution sont initialement utilisés pour imiter le comportement des modèles jugés non interprétables afin de les rendre plus compréhensibles.

Si un résultat d'intérêt est coûteux, long ou difficile à mesurer (par exemple, parce qu'il provient d'une simulation informatique complexe), un modèle de substitution plus rapide et proposant un niveau de performance acceptable peut être utilisé à la place.

La différence entre les modèles de substitution utilisés en ingénierie et en apprentissage automatique est que le modèle sous-jacent est un modèle d'apprentissage automatique (et non une simulation) et que le modèle de substitution doit être interprétable.

L'objectif des modèles de substitution est d'approcher les prédictions du modèle sous-jacent aussi précisément que possible et d'être interprétables en même temps.

En plus d'être capables de reproduire un modèle donné, les modèles de substitution peuvent parfois présenter l'avantage d'être plus rapides lors de l'exécution et du paramétrage. Ces modèles s'enrichissent ainsi des modèles initiaux en assimilant leurs caractéristiques de manière plus compacte, garantissant une performance optimale.

4.2 Utilisation

Dans le cadre de l'estimation des classes de sinistralité des nouveaux sinistres, nous souhaitons créer des modèles pouvant reconstituer le comportement du modèle ayant minimisé l'erreur d'estimation des provisions : ici, le modèle des *K-Prototypes* avec $K = 9$ classes.

La figure 27 montre le processus utilisé pour associer les classes aux nouveaux sinistres sans disposer des caractéristiques de développement individuelles.

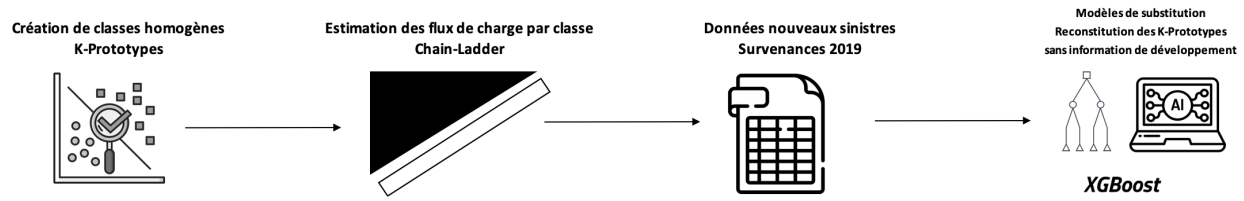

Figure 27 – Processus d'optimisation et d'affectation des classes de sinistralité

Les coefficients de développement $\lambda_i^{Règlements}$ ou λ_i^{Charge} ainsi que les développements individuels pondérés $DIP_i^{Règlements}$ ou DIP_i^{Charge} ayant contribué à l'homogénéité des classes, les modèles de substitution devront, seulement à partir des caractéristiques observées des sinistres antérieurs à 2019, déterminer des structures sous-jacentes de ces classes.

4.3 Modèles utilisés

Des algorithmes d'apprentissage statistique tels que les forêt aléatoires ou l'algorithme *XGBoost* serviront de modèles de substitution afin de reconstituer ces classes sur les sinistres antérieurs à 2019. Ce qui permettra d'évaluer la qualité de prédiction par validation croisée.

Enfin, il est important de préciser que le flux de données des nouveaux sinistres devra respecter le format des données du modèle initial. Une fois le modèle de substitution entraîné, les nouvelles données connues à chaque future date d'inventaire pourront, à travers ces nouveaux modèles de substitution se voir attribuée une nouvelle classe à partir d'un de ces modèles d'apprentissage statistique.

4.4 Qualité d'estimation

Alors qu'une classification des sinistres par association des observations à leurs centroïdes entraînerait une classification parfaite, le fait de ne pas disposer de suffisamment de recul sur les développements engendrera un biais de classification. En effet, ces informations ayant participé à la création des classes de sinistralité, les modèles de substitution que nous utiliserons pourront commettre des erreurs de classification et voir leur performance réduite.

Les paramètres des modèles de substitution devront être judicieusement choisis afin d'éviter l'association d'une partie des sinistres à une mauvaise classe. Ce qui pourrait engendrer des erreurs d'estimation des provisions lors de l'utilisation des méthodes agrégées. Ces choix seront établis par validation croisée sur un échantillon de test (section 5.1.1).

5 Apprentissage statistique

Les méthodes d'apprentissage statistique, (*Machine Learning*) seront dans notre cas utilisées afin de prédire les classes d'appartenance de nouveaux sinistres survenus, différentes méthodes seront ainsi utilisées pour ce problème de classification. Ces méthodes seront ensuite comparées afin d'en déduire le modèle le plus performant.

Les modèles d'apprentissage statistique sont réputés pour leur fort pouvoir prédictif sur un échantillon de données indépendant, généralement appelé échantillon de test. Pour un problème de régression ou de classification, l'utilisation de ces modèles est toujours associée à la recherche de performances optimales, sur des indicateurs forcément différents selon l'approche. La comparaison ainsi que l'évaluation de la qualité des différents modèles s'effectuera après optimisation par validation croisée sur un échantillon de test. Dans ce chapitre, nous évoquerons les bonnes pratiques en matière de préparation de données et de recherche de performance lors de l'utilisation de modèles d'apprentissage statistique.

5.1 Sélection d'un modèle optimal

Dans cette section, les différents modèles d'apprentissage statistique qui seront utilisés pour associer chaque nouveau sinistre à sa classe de sinistralité seront présentés. Ces modèles seront comparés selon des critères de performance propres à la classification sur un échantillon de test. Le processus de sélection sera fait selon deux principaux objectifs :

- Sélection d'un modèle parmi plusieurs après avoir estimé leurs performances afin de retenir le meilleur sur un échantillon de test.
- Estimation des classes des sinistres de 2019.

5.1.1 Partitionnement des données

En présence de suffisamment d'observations, la meilleure approche est de partitionner de manière aléatoire nos données en trois sous ensembles : un échantillon d'apprentissage, sur lequel seront entraînés les différents modèles, un échantillon de validation, permettant de mesurer les erreurs de prédiction des modèles et un échantillon de test indépendant permettant la généralisation des erreurs de prédiction, servant idéalement uniquement en toute fin d'analyse. Nous choisirons le modèle selon la meilleure performance obtenue sur ce dernier échantillon. Nous décomposerons les données de la manière suivante :

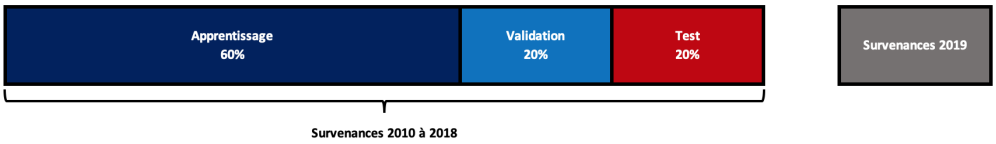

Figure 28 – Décomposition de la base de données

5.1.2 Indicateurs de performance

Les résultats des différentes méthodes que nous utiliserons seront des prédictions de classes d'appartenance pour chacune des observations de l'échantillon de test. On s'intéresse donc dans un premier temps à la matrice de confusion.

Matrice de Confusion : La matrice de confusion renvoie les résultats des prédictions d'un problème de classification, elle permet de mettre en lumière les prédictions correctes des prédictions incorrectes par classe, elle permet une comparaison entre résultats prédits et observés. Cette matrice est de taille $K \times K$ avec K le nombre de classes de la variable réponse. Il s'agit d'un élément essentiel à la compréhension des erreurs commises par le modèle. Les lignes de cette matrice correspondent aux classes prédites, les colonnes correspondent aux classes réelles. Ainsi, la diagonale de cette matrice correspond aux observations bien prédites.

Accuracy ou justesse : L'*accuracy* (ou justesse) est un indicateur permettant de mesurer la proportion des prédictions correctes effectuées par un modèle, on la note ainsi :

$$\text{Justesse} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}}$$

Il s'agît donc du rapport entre la somme de la diagonale de la matrice de confusion et de sa somme totale.

***Logarithmic Loss* (ou *Log Loss*) :** La fonction de perte logarithmique pénalise les mauvaises classifications. Dans un problème de classification multi-classes, des probabilités sont attribuées à chaque classe pour chaque observation, si l'on suppose N observations appartenant à M classes, alors le *Log Loss* est calculé ainsi :

$$LogarithmicLoss = \frac{-1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \times \ln(p_{ij})$$

Avec $\forall i \in (1, ..., N)$ et $j \in (1, ..., M)$,

- y_{ij}, indicateur d'appartenance de l'observation i à la classe j.

- p_{ij} la probabilité de l'observation i d'appartenir à la classe j.

Cet indicateur est défini dans $[0, \infty)$, en général, plus l'indicateur est proche de 0, plus la justesse augmente.

5.1.3 Optimisation des modèles

Chaque modèle possède un ou plusieurs paramètres de calibration, une analyse de leur performance selon les divers paramètres sera effectuée de manière à garantir une comparabilité des résultats entre modèles judicieusement calibrés. La justesse sera le critère à maximiser entre les différents modèles. Les paramètres de calibration α font varier les complexités respectives des modèles, l'objectif sera de trouver pour chacun d'eux, la combinaison α_{opt} de paramètres maximisant la justesse.

5.2 Arbres *CART*

5.2.1 Introduction

Les méthodes basées sur les arbres consistent à partitionner les données dans un espace en plusieurs rectangles afin de créer un modèle sur chacun de ces sous-espaces. Ils sont simples et assurent de bonnes prédictions. On considère tout d'abord un problème de régression, avec une variable réponse continue, Y et deux variables explicatives X_1 et X_2, à valeur dans $[0, 1]$. On commence tout d'abord par séparer l'espace en deux sous-espaces, et on modélise Y comme étant sa moyenne pour chacun de ces sous-espaces. Il est important de choisir des variables et points adéquats pour effectuer cette séparation. Ensuite, chacun de ces sous-espaces est séparé en d'autres sous-espaces, et ce jusqu'à application d'une règle d'arrêt.

Par exemple, dans la figure 29, on sépare tout d'abord X_1 à t_1, puis le sous-espace $X_1 \leq t_1$ est séparé en X_2 à t_2, enfin $X_1 \geq t_3$ est séparé en $X_2 = t_4$. On obtient donc les régions $R_1, R_2, ..., R_5$.

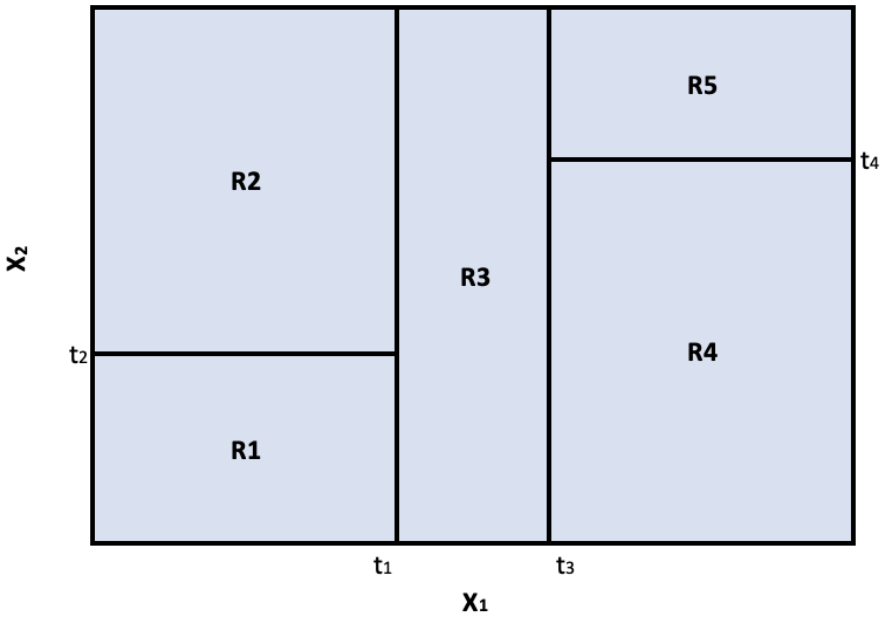

Figure 29 – Principe du *CART*

Ce modèle de régression prédit Y avec c_m constant dans la région R_m de la manière suivante :

$$\hat{f}(X) = \sum_{m=1}^{5} c_m I\left\{(X_1, X_2) \in R_m\right\}$$

Ce modèle peut être représenté dans l'arbre binaire comme suit :

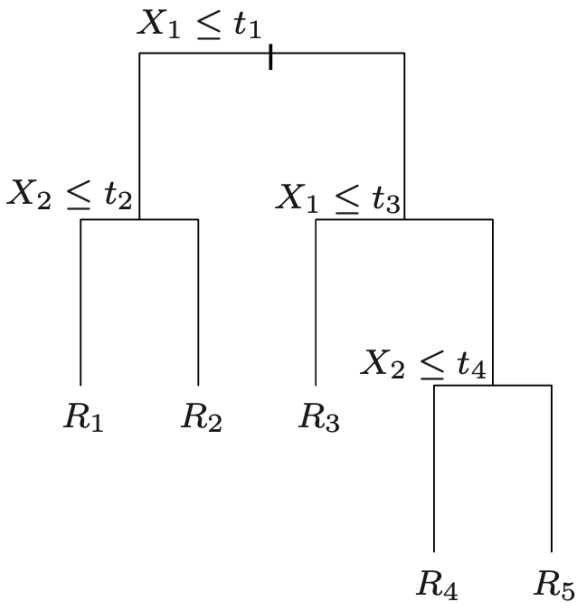

Figure 30 – Arbre *CART* Binaire

Les données complètes sont représentées en haut de l'arbre. Les observations satisfaisant la condition à chaque nœud de l'arbre sont assignées à la branche de gauche et les autres à la branche de droite. Les nœuds terminaux correspondent aux régions $R_1, R_2, ..., R_5$.

L'avantage principal de ces arbres binaires est leur facilité d'interprétation. Ces arbres permettent de représenter les partitions crées et ce, même avec plusieurs variables explicatives (là où le graphe *CART*(figure 30), ne peut représenter qu'un problème à deux variables explicatives).

5.2.2 Arbres de classification

Lors d'un problème de classification multi-classes, les indicateurs cibles pour les nœuds, ou l'élagage des arbres sont différents de ceux utilisés en régression basés sur l'erreur quadratique. Pour chaque nœud m, représentant une région R_m avec N_m observations, on a :

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I\left(y_i = k\right)$$

la proportion des observations appartenant à la classe k, dans le nœud m. Pour attribuer une classe à chacune des observations de la classe du nœud m, on choisit $k(m) = \arg\max_k \hat{p}_{mk}$ le vote majoritaire dans le nœud m. On note également trois mesures dites d'impureté, $Q_m(T)$:

L'erreur de classification :

$$\frac{1}{N_m} \sum_{i \in R_m} I\left(y_i \neq k(m)\right) = 1 - \hat{p}_{mk(m)}$$

L'indice de Gini :

$$\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{N} \hat{p}_{mk} \left(1 - \hat{p}_{mk}\right)$$

L'entropie croisée (ou déviance) :

$$-\sum_{k=1}^{K} \hat{p}_{mk} \ln \hat{p}_{mk}$$

Pour une classification à deux classes, si p est la proportion de la seconde classe, ces mesures valent respectivement $1 - \max(p, 1-p), 2p(1-p)$ et $-p \ln p - (1-p) \ln(1-p)$.

Ces mesures sont similaires mais l'indice de Gini et l'entropie croisée sont différentiables ce qui peut permettre l'optimisation numérique. Ces deux indicateurs sont plus sensibles aux changements de probabilités dans les nœuds que l'indicateur d'erreur de classification.

Par exemple, dans un problème à deux classes, avec 400 observations dans chaque classe noté $(400, 400)$, supposons qu'un nœud ait séparé cet échantillon de la manière suivante : $(300, 100), (100, 300)$, et un autre ainsi : $(200, 400), (200, 0)$, alors l'erreur de classification est dans les deux cas de $0,25$. Le second modèle produirait un nœud avec des observations d'une seule classe et serait préférable. L'indice de Gini et l'entropie croisée sont plus faibles dans le second modèle. C'est pour cela que ces deux indicateurs doivent être pris en considération dans la création des modèles.

Afin de construire les arbres et de bien effectuer leur élagage, ces trois indicateurs doivent être pris en compte, bien que l'erreur de classification soit généralement privilégiée. L'indice de Gini peut être interprété de deux manières. Plutôt que de classifier les observations selon les votes majoritaires, on pourrait utiliser leurs probabilités $\hat{p}_{mk}$, l'erreur serait alors notée $\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'}$, ce qui correspond à l'indice de Gini. De plus, si l'on attribue la valeur 1 à la classe k et 0 sinon, la variance de ce nœud pour cette variable réponse binaire serait $\hat{p}_{mk}\left(1 - \hat{p}_{mk}\right)$. La somme pour toutes ces classes serait de nouveau l'indice de Gini.

5.3 *Bagging*

Le *Bagging* se base sur une méthode de *Bootstrap* afin d'augmenter la performance d'une prédiction.

On considère tout d'abord un problème de régression : on adapte un modèle sur les données $\mathbf{Z} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, on obtient les prédictions $\hat{f}(x)$. Le *Bagging* (*Bootstrap Aggregation*) effectue la moyenne de ces prédictions sur un ensemble d'observations ré échantillonnées afin de réduire la variance des estimations. Pour chaque échantillon *Bootstrap* $\mathbf{Z}^{*b}, b = 1, 2, \ldots, B$, on crée un modèle et on obtient les prédictions $\hat{f}^{*b}(x)$.

L'estimateur du *Bagging* est alors :

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{*b}(x)$$

Soit $\hat{\mathcal{P}}$ la distribution empirique donnant la probabilité $1/N$ à chaque couple (x_i, y_i). L'estimateur du *Bagging* peut alors être également noté $\mathrm{E}_{\hat{\mathcal{P}}} \hat{f}^*(x)$ où $\mathbf{Z}^* = (x_1^*, y_1^*), (x_2^*, y_2^*), \ldots, (x_N^*, y_N^*)$ et chaque $(x_i^*, y_i^*) \sim \hat{\mathcal{P}}$.

Le *Bagging* peut également être appliqué aux arbres de régression de type *CART*. Ici, $\hat{f}(x)$ désigne la prédiction d'un arbre de régression. Chaque arbre issu du *Bootstrap* aura des nœuds terminaux différents. L'estimateur de ces arbres "baggés" sera la moyenne des prédictions de ces B arbres.

Pour un problème de classification en K-classes, on considère un estimateur de nos arbres $\hat{G}(x)$. On considère également une fonction indicatrice sous-jacente $\hat{f}(x)$, vecteur possédant une unique valeur égale à 1 et $K - 1$ valeurs égales à 0, telle que $\hat{G}(x) = \arg\max_k \hat{f}(x)$. L'estimateur du *Bagging* $\hat{f}_{\text{bag}}(x)$ sera ensuite un vecteur de taille K $[p_1(x), p_2(x), \ldots, p_K(x)]$ avec $p_k(x)$ représentant la proportion d'arbres prédisant la classe k pour l'observation x. Le modèle de classification sélectionne la classe ayant le plus de "votes" sur les B arbres. Ainsi, $\hat{G}_{\text{bag}}(x) = \arg\max_k \hat{f}_{\text{bag}}(x)$.

Il est également possible de récupérer les probabilités d'obtention de la classe k pour l'observation x au lieu de la classe en elle même. La classe prédite sera généralement égale à la classe ayant la plus grande probabilité, mais peut aussi être soumise à un certain seuil, notamment dans le cas de classes déséquilibrées par exemple. Ce vecteur de probabilité peut également être très utile dans le but d'estimer une fréquence de sinistre dans le cadre du calcul d'une prime pure individuelle à partir d'un modèle de fréquence-coût moyen.

5.4 *Boosting*

Le *Boosting* a été initialement créé pour des problèmes de classification, il s'agît d'une méthode combinant les résultats de plusieurs classifieurs faibles pour créer un modèle à fort pouvoir prédictif. Cet aspect là du modèle semble similaire aux approches de type *Bagging* mais nous verrons que ce modèle est fondamentalement différent.

*AdaBoost.M*1 (Freund et Schapire 1997[5]) fut le premier algorithme de *Boosting*, il considérait un problème de classification binaire avec une variable réponse $Y \in \{-1, 1\}$.

A partir d'un vecteur de variables explicatives X, ce classifieur $G(X)$ prédit une des deux valeurs de Y, l'erreur résultante est notée

$$\overline{\text{err}} = \frac{1}{N} \sum_{i=1}^{N} I\left(y_i \neq G\left(x_i\right)\right)$$

et l'espérance du taux d'erreur sur les futures prédictions se note $\mathrm{E}_{XY} I(Y \neq G(X))$.

Un classifieur faible est un classifieur dont l'erreur d'estimation n'est que très légèrement inférieure à une prédiction aléatoire. Le principe du *Boosting* repose alors sur l'application de classifieurs faibles de manière répétée sur des versions modifiées des données, produisant une séquence de ces classifieurs faibles : $G_m(x), m = 1, 2, \ldots, M$. La prédiction finale est obtenue à partir d'un vote majoritaire pondéré :

$$G(x) = \text{signe}\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right)$$

Les $\alpha_1, \alpha_2, \ldots, \alpha_M$ sont calculés par l'algorithme *Boosting* et sont les poids respectifs des $G_m(x)$, le but étant de donner le plus d'importance aux classifieurs les plus précis. La modification des données à chaque étape du *Boosting* consiste à appliquer des poids $w_1, w_2, \ldots, w_N$ à chacune des observations $\left(x_i, y_i\right), i = 1, 2, \ldots, N$. Les poids sont initialisés à $w_i = 1/N$ de manière à ce que le premier classifieur soit entraîné de manière habituelle. Ensuite, pour $m = 2, 3, ..., M$, les poids de chaque observation sont modifiés et l'algorithme est réappliqué à ces observations. A chaque étape, les observations mal prédites voient leurs poids augmenter, tandis que les poids des observations bien prédites diminuent. Ainsi, plus on avance dans les itérations de cet algorithme, plus les observations initialement difficiles à classifier ont un poids important.

Chacun des modèles a pour objectif de se concentrer sur ces observations mal prédites.

L'algorithme *AdaBoost.M*1 se présente ainsi :

Algorithme 4 : *AdaBoost.M1.*

1. Initialisation des poids des observations $w_i = 1/N, i = 1, 2, \ldots, N$

2. pour *m=1 à M* **faire**

(a) Entraîner un premier modèle $G_m(x)$ sur l'échantillon d'apprentissage en utilisant les poids w_i.

(b) Calculer

$$\text{err}_m = \frac{\sum_{i=1}^{N} w_i I\left(y_i \neq G_m\left(x_i\right)\right)}{\sum_{i=1}^{N} w_i}$$

(c) Calculer

$$\alpha_m = \ln\left(\left(1 - \text{err}_m\right) / \text{err}_m\right)$$

(d) Réattribuer les poids $w_i \leftarrow w_i \cdot \exp\left[\alpha_m \cdot I\left(y_i \neq G_m\left(x_i\right)\right)\right], i = 1, 2, \ldots, N$

fin

Résultat : $G(x) = \text{signe}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$

5.5 *Random Forest*

5.5.1 Introduction

L'algorithme *Random Forest* (Breiman, 2001 [2]) est un algorithme basé sur l'algorithme *Bagging*, qui construit un large ensemble d'arbres dé-corrélés, afin de construire un arbre moyen. Sa performance est très similaire à celle du *Boosting* dans de nombreux cas, mais les *Random Forests* s'entraînent et s'améliorent plus facilement. Les *Random Forests* sont donc très populaires et présents dans de nombreuses librairies R ou Python (*randomForest*, *Caret*, H2O ou encore *Scikit-Learn* pour Python).

5.5.2 Définition des *Random Forests*

L'idée générale du *Bagging* est d'agréger plusieurs modèles plus ou moins biaisés pour réduire la variance. Les arbres, de par leur capacité à déterminer des structures d'interaction complexes dans les données, présentent un faible biais s'ils sont suffisamment profonds. Le fait d'agréger plusieurs arbres est particulièrement intéressant : chaque arbre généré dans le *Bagging* est identiquement distribué (i.d.) et l'espérance de l'arbre moyen est similaire à l'espérance de chacun de ces arbres, le biais des arbres agrégés est donc similaire aux biais individuels de ces arbres, la seule perspective d'amélioration est une réduction de variance. Le *Boosting* quant à lui construit ses arbres de manière à réduire le biais, les arbres ne sont alors pas i.d.

La moyenne de B variables aléatoires i.i.d. de variance σ^2 est $\frac{1}{B}\sigma^2$. Si les variables sont simplement identiquement distribuées mais pas nécessairement indépendantes, avec un corrélation ρ positive, alors la variance de cette moyenne est :

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

Lorsque B augmente, le second terme disparaît, la corrélation des arbres limite donc les avantages de ce principe d'agrégation.

L'algorithme *Random Forest* se présente ainsi :

Algorithme 5 : *Random Forest* pour la classification

1.pour *b=1 à B* **faire**

(a) Créer un échantillon *bootstrap* Z de taille N à partir des données d'apprentissage.

(b) Créer une forêt aléatoire d'arbres T_b sur les données issues du *boostrap* en répétant les étapes suivantes pour chaque nœud terminal de l'arbre, jusqu'à atteindre la taille minimale n_{min} du nœud.

- **(i)** Choisir m variables aléatoirement parmi les p variables.
- **(ii)** Choisir la meilleure variable/point de séparation parmi les m.
- **(iii)** Séparer le nœud en deux sous-nœuds.

fin

2. Retenir l'ensemble d'arbres $\{T_b\}_1^B$.

Pour obtenir une prédiction de la classe d'une observation x :

Soit $\hat{C}_b(x)$ la classe prédite par le b^{ieme} arbre *Random Forest*, alors $\hat{C}_{\text{rf}}^B(x) =$ vote majoritaire $\left\{\hat{C}_b(x)\right\}_1^B$

L'idée de cet algorithme est d'améliorer la réduction de variance du *Bagging* en réduisant la corrélation entre les arbres sans trop augmenter la variance. C'est ce qui est réalisé dans le processus de création d'arbres par la sélection aléatoire de variables d'entrée.

5.5.3 *Random Forest* pour la classification

Lorsque l'on utilise l'algorithme *Random Forest* pour un problème de classification, le résultat est un vecteur de prédictions de classes. Pour chaque arbre, la classe retenue est issue d'un vote majoritaire de l'ensemble des arbres pour chaque observation. De plus, Breiman [2] a proposé une valeur par défaut du nombre m de variables à sélectionner qui vaut $\sqrt{p}$, ainsi que la valeur de la taille du nœud minimal, fixée à 1. En pratique, les valeurs optimales pour ces paramètres dépendent du problème traité et font l'objet de calibrage pour le modèle, afin d'en accroître son pouvoir prédictif.

5.5.4 Échantillons *Out of Bag*

Un aspect important du *Random Forest* est l'utilisation d'échantillons *out-of-bag* (OOB) : pour chaque observation, $z_i = (x_i, y_i)$, on construit un estimateur *Random Forest* en agrégeant seulement les arbres correspondant aux échantillons issus du *bootstrap* dans lesquels z_i n'apparaît pas. L'erreur OOB peut être assimilée à l'erreur obtenue par validation croisée. A l'inverse de beaucoup d'autres estimateurs non linéaires, le *Random Forest* peut être optimisé tout au long

de son exécution, grâce à la validation croisée et à la stabilisation de l'erreur OOB.

5.5.5 Importance des variables

Un avantage considérable du *Random Forest* est la possibilité de pouvoir analyser l'importance des variables sur les estimations. A chaque nœud d'un arbre, l'impact de la variable utilisée sur la qualité du modèle va définir sa mesure d'importance, cette mesure est ensuite cumulée pour chaque arbre du modèle. Les échantillons *out-of-bag* sont également utilisés pour construire un nouvel indicateur d'importance des variables, cet indicateur mesure le pouvoir prédictif de chaque variable au sein du modèle. Lors de la construction de chaque arbre, les échantillons OOB sont utilisés afin de mesurer la précision des prédictions, ensuite les valeurs de chaque variable sont réorganisées aléatoirement et la précision est à nouveau mesurée. La perte de précision est ainsi mesurée sur tout les arbres et permet de mesurer l'importance de chaque variable dans le modèle.

5.6 *Gradient Boosting Machine*

5.6.1 Introduction

Nous avons vu précédemment le *Boosting* comme une méthode visant à transformer un ensemble d'algorithmes à faible pouvoir prédictif (*weak learners*) en algorithmes à haut pouvoir prédictif (*strong learners*).

On rappelle que dans le *Boosting*, chaque nouvel arbre est modélisé sur un nouveau sous-échantillon des données globales. Le *gradient boosting* peut-être assimilé à l'algorithme *AdaBoost*.

On rappelle également que cet algorithme commence par l'apprentissage d'un arbre de décision où l'on assigne un poids identique à chaque observation. Après avoir construit le premier arbre, on augmente les poids des observations qui ont été difficiles à classifier correctement et on diminue les poids de celles qui ont été correctement classifiées. Le second arbre est ainsi créé avec ces nouvelles pondérations.

Pour le *gradient boosting*, l'idée générale est d'améliorer les prédictions du premier arbre. Le processus consiste donc à créer un modèle avec un premier puis un second arbre, calculer les erreurs de classification à partir de ce modèle à deux arbres et construire un troisième arbre pour prédire les résidus de ce modèle. Ce processus est répété un nombre prédéfini de fois. Les arbres intermédiaires permettent de classifier correctement les observations n'ayant pas été correctement prédites par les arbres précédents.

Les prédictions finales sont les sommes pondérées des prédictions faites par l'ensemble des arbres du modèle. Le *gradient boosting* entraîne de nombreux modèles de manière additive. La principale différence entre le *gradient boosting* et *AdaBoost* est sa manière d'appréhender les classifieurs faibles *«Weak Learners»*. *AdaBoost* identifie les erreurs et s'améliore en réajustant les poids tandis que le *gradient boosting* améliore l'erreur commise en utilisant le gradient pour améliorer la fonction de perte. La fonction de perte dépend du problème que nous étudions. Pour la classification, le *gradient boosting* s'intéressera à la justesse qui sera améliorée après chaque itération, mais pourra s'intéresser à l'optimisation d'autres indicateurs de performance spécifiés par l'utilisateur.

5.6.2 Principe de descente du gradient

Le principe itératif du *gradient boosting* permet d'approcher une solution optimale pour un problème donné grâce à la descente du gradient. L'optimisation de paramètres sera essentielle à cet algorithme afin d'approcher la solution optimale pour minimiser ou maximiser sa fonction

objectif.

Soit :

$$J(y, f) = \sum_{i=1}^{n} j\left(y_i, f\left(x_i\right)\right)$$

Avec $f()$, un classifieur doté de paramètres, $j()$ une fonction de coût qui compare la valeur observée et la prédiction du modèle, $J()$, une fonction de perte calculée comme étant une somme sur l'ensemble des observations. L'objectif principal sera de minimiser $J()$ en fonction des paramètres constituant $f()$, avec :

$$f_k\left(x_i\right) = f_{k-1}\left(x_i\right) - \eta \times \nabla j\left(y_i, f\left(x_i\right)\right)$$

où $f_k()$ est le classifieur à l'étape k, η est un paramètre d'apprentissage du modèle qui permet de contrôler la vitesse d'apprentissage du modèle vers la convergence de la fonction de perte, et ∇ le gradient, soit la dérivée partielle d'ordre 1 de la fonction de coût par rapport au modèle :

$$\nabla j\left(y_i, f\left(x_i\right)\right) = \frac{\partial j\left(y_i, f\left(x_i\right)\right)}{\partial f\left(x_i\right)}$$

L'algorithme *AdaBoost* optimise une fonction de coût exponentielle : à chaque modèle M_t, après chaque pondération de ses observations, M_{t-1} permet de minimiser la fonction de coût choisie.

Pour $y \in \{-1, +1\}$, on a :

$$J(f) = \sum_{i=1}^{n} \exp\left(-y_i \times f\left(x_i\right)\right)$$

Avec $J()$ la fonction de coût à minimiser et $f()$ le classifieur agrégé créé à partir d'une combinaison linéaire de classifieurs individuels M_k.

$$f_k = f_{k-1} + \frac{\alpha_k}{2} \times M_k$$

On corrige $f_k()$, le modèle agrégé à l'étape k, par le classifieur individuel M_k, créé à partir d'un échantillon repondéré ω. M_k représente ici le gradient. Pour chaque modèle intermédiaire M_k construit, le coût du modèle agrégé global est réduit. On peut noter les poids des individus à l'étape k de manière récursive en fonction de l'étape précédente :

$$\omega_i^k = \omega_i^{k-1} \times \exp\left[\alpha_{k-1} \cdot I\left(y_i \neq M_{k-1}(i)\right)\right]$$

On observe bien l'idée de correction des poids de manière itérative afin d'améliorer la fonction de perte globale.

5.6.3 Le *Gradient Boosting Machine* pour la classification

L'algorithme du *gradient boosting machine* peut s'écrire comme ceci :

Algorithme 6 : Gradient pour la classification en K-Classes

1. Initialisation de $f_{k0}(x) = 0, k = 1, 2, \ldots, K$

2. pour *m=1 à M* **faire**

(a) Soit

$$p_k(x) = \frac{e^{f_k(x)}}{\sum_{\ell=1}^{K} e^{f_\ell(x)}}, k = 1, 2, \ldots, K$$

(b) pour *k=1 à K* **faire**

(i) Calculer $r_{ikm} = y_{ik} - p_k(x_i), i = 1, 2, \ldots, N$

(ii) Construire un arbre de régression pour $r_{ikm}, \quad i = 1, 2, \ldots, N$ donnant les régions terminales $R_{jkm}, j = 1, 2, \ldots, J_m$

(iii) Calculer

$$\gamma_{jkm} = \frac{K-1}{K} \frac{\sum_{x_i \in R_{jkm}} r_{ikm}}{\sum_{x_i \in R_{jkm}} |r_{ikm}| (1 - |r_{ikm}|)}, j = 1, 2, \ldots, J_m$$

(iv) Déduire $f_{km}(x) = f_{k,m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jkm} I(x \in R_{jkm})$

fin

fin

3. Résultat : $\hat{f}_k(x) = f_{kM}(x), k = 1, 2, \ldots, K$

On note deux principaux paramètres pour cet algorithme, le nombre M d'itérations ainsi que la taille de chaque arbre $J_m, m = 1, 2, \ldots, M$.

5.6.4 Taille optimale des arbres

Le *Boosting* est une technique visant à combiner plusieurs modèles, ici des arbres. La taille optimale de chaque arbre est estimée de manière séparée. Des arbres totaux sont généralement construits puis élagués de manière à estimer un nombre optimal de nœuds terminaux. Cette approche n'est pas suffisamment performante puisque les arbres créés peuvent être trop profonds, ce qui engendre des temps de calcul trop importants et de faibles performances. Une solution simple serait de spécifier la même taille $J_m = J$, $\forall m$ à chacun des arbres du modèle. Un arbre de régression à J nœuds terminaux est ainsi créé à chaque itération. Cette taille J des arbres est donc un paramètre du modèle et peut être optimisé afin de maximiser sa performance. L'optimisation de J peut être faite en considérant la fonction objectif suivante :

$$\eta = \arg\min_f \mathrm{E}_{XY} L(Y, f(X))$$

La fonction objectif $\eta(x)$ est la fonction renvoyant la meilleure prédiction sur les futures données. Il s'agît d'une fonction à approximer. Une des propriétés de $\eta(X)$ est l'interaction entre les différentes variables $X^T = (X_1, X_2, \ldots, X_p)$, cette interaction est démontrée par son ANOVA (Analyse des Variances) :

$$\eta(X) = \sum_j \eta_j(X_j) + \sum_{jk} \eta_{jk}(X_j, X_k) + \sum_{jkl} \eta_{jkl}(X_j, X_k, X_l) + \cdots$$

La première somme correspond à la fonction objectif sur une seule variable X_j. Les fonctions $\eta_j(X_j)$ doivent approcher $\eta(X)$ pour le critère de performance utilisé. On appelle ces $\eta_j(X_j)$ l'effet principal de X_j. La seconde somme considère les deux variables qui, lorsque cette somme est ajoutée améliorent au mieux la fonction objectif. On les appelle interactions de second ordre. La troisième somme représente les interactions de troisième ordre et ainsi de suite. Les interactions de faible ordre sont en pratique les plus fréquentes, car en général, les modèles qui produisent des interactions d'ordres importants présentent des manques de justesse. Ces interactions sont limitées par la taille J des arbres. Les effets d'interaction d'ordre $J-1$ ne sont pas possibles car les modèles de *Boosting* sont additifs. Si l'on choisit une profondeur d'arbres $J = 2$, on n'obtient pas d'interactions mais seulement des effets principaux. Pour $J = 3$, on observe seulement des interactions d'ordre 2. Il est donc possible d'orienter son choix pour J vers le nombre d'interactions présumées pour $\eta(x)$. Ce nombre n'est pas toujours connu, mais est généralement faible dans de nombreux problèmes d'optimisation. $J = 2$ est souvent insuffisant, de nombreux modèles se basent sur une profondeur d'arbres $J = 10$. Ce paramètre étant généralement optimisé par validation croisée en minimisant l'erreur sur un échantillon de validation pour des valeurs de J allant de 4 à 10 par exemple.

5.6.5 Nombre d'itérations optimal

Il existe d'autres paramètres que la profondeur J des arbres pour le modèle de *gradient boosting machine*. Parmi eux, le nombre M d'itérations du *Boosting*. A chaque itération, l'erreur d'entraînement $L(f_M)$ diminue. Pour un nombre M d'itérations suffisamment grand, cette erreur peut être minime. Cependant, un trop grand nombre d'itération peut amener à du sur-apprentissage, dégradant la qualité des futures estimations. Une manière d'obtenir un nombre d'itérations optimal M^* est de mesurer l'erreur par validation croisée selon les valeurs de M, sur un échantillon de validation.

5.6.6 *Shrinkage*

Le *gradient boosting* dispose également du paramètre de *Shrinkage*, qui lui aussi peut-être optimisé par validation croisée. Ce paramètre consiste pour le *Boosting* à pondérer par ν tel que $0 < \nu < 1$ la contribution des arbres lorsqu'ils sont ajoutés dans le modèle. Il s'agît dans

l'algorithme (6) de remplacer l'étape (2.iv) par :

$$f_m(x) = f_{m-1}(x) + \nu \cdot \sum_{j=1}^{J} \gamma_{jm} I\left(x \in R_{jm}\right)$$

Ici, ν peut être interprété comme le taux d'apprentissage du *Boosting*. Plus ν est petit, plus le *Shrinkage* est grand, et donc plus le risque d'erreur est grand à nombre d'itérations équivalent. Ces deux paramètres contrôlent la qualité de prédiction du modèle. Ces deux paramètres ne doivent pas être optimisés indépendemment. En général, plus le paramètre ν est petit, plus l'erreur est importante sur un échantillon de test indépendant. Le nombre d'itérations doit par conséquent être plus grand pour garantir un niveau de prédictabilité équivalent (Friedman, 2001[6]). Empiriquement, la stratégie d'utiliser un paramètre ν et de calibrer le nombre d'itérations en utilisant un critère d'arrêt semble être le meilleur choix d'optimisation. Cette stratégie a montré ses preuves dans de nombreux problèmes de régression. Cela n'a cependant pas apporté de résultats drastiquement meilleurs lors de problèmes de classification. Utiliser un paramètre de *Shrinkage* petit implique une contrainte pratique de temps de calcul. En effet, le nombre d'itération doit être plus grand, et le temps de calcul de l'algorithme est proportionnel à ce paramètre. Cependant, ce temps de calcul reste raisonnable au regard de la faible profondeur des arbres non élagués créés par le *gradient boosting*.

5.6.7 Paramètres de sous-échantillonage

Nous avons vu précédemment que le *Bagging* pouvait améliorer la performance de certains classifieurs grâce au ré-échantillonage. Le *Bagging* introduit un mécanisme de réduction de variance qui est utilisé par le *gradient boosting* pour améliorer ses performances prédictives ainsi que son temps de calcul. En 1999, Friedman [6] a proposé l'utilisation du ré-échantillonage pour le *gradient boosting* : à chaque itération, on effectue un tirage aléatoire sans remise d'une partition η de l'échantillon d'apprentissage, et l'on construit les prochains arbres en utilisant ce sous-échantillon. On utilise classiquement un *subsample* $\eta = 21$, bien que pour de grands jeux de données, ce paramètre peut-être inférieur à 12. En plus d'améliorer le temps de calcul du *gradient boosting*, ce paramètre une fois bien calibré améliore de manière significative ses performances prédictives.

5.6.8 Interprétabilité du modèle

Les arbres de décision sont très faciles d'interprétation puisqu'ils peuvent être entièrement représentés graphiquement par un simple arbre binaire. Le *gradient boosting* utilise des combinaisons linéaires de plusieurs arbres et devient donc mois interprétable. Il est donc nécessaire de pouvoir interpréter ce modèle autrement.

L'importance des variables

Dans de nombreux modèles d'apprentissage statistique, on utilise de nombreuses variables prédictives ayant rarement le même niveau d'importance. Dans de nombreux modèles, un faible nombre de variables explicatives ont un réel effet sur la variable réponse. La plupart des variables n'ont qu'un effet minime et auraient pu ne pas être intégrées au modèle. Le *gradient boosting*, comme l'algorithme *Random Forest* propose une possibilité d'interprétation de la contribution de chacune des variables du modèle sur la prédiction de la variable réponse. Pour un unique arbre de décision, T, Breiman et al. (1984)[3] ont proposé :

$$\mathcal{I}_\ell^2(T) = \sum_{t=1}^{J-1} \hat{\imath}_t^2 I(v(t) = \ell)$$

comme mesure d'importance de chaque variable explicative X_l. Cette somme est sur l'ensemble des $J-1$ nœuds d'un arbre. A chaque nœud t, une variable $X_{v(t)}$ parmi l'ensemble des variables est utilisée pour séparer deux branches de l'arbre de décision. On retient la variable ayant amélioré le plus significativement le modèle grâce à la mesure $\hat{\imath}_t^2$. Pour un problème de classification, on considère K modèles différents $f_k(x), k = 1, 2, \ldots, K$, chacun noté comme la somme de plusieurs arbres :

$$f_k(x) = \sum_{m=1}^{M} T_{km}(x)$$

On note $\mathcal{I}_{\ell k}$ la mesure d'importance de la variable X_l, autrement dit sa capacité à séparer de manière précise les observations de la classe k des autres classes.

$$\mathcal{I}_{\ell k}^2 = \frac{1}{M} \sum_{m=1}^{M} \mathcal{I}_\ell^2 \left(T_{km}\right)$$

L'importance finale de la variable X_l est obtenue par moyenne sur l'ensemble des classes :

$$\mathcal{I}_\ell^2 = \frac{1}{K} \sum_{k=1}^{K} \mathcal{I}_{\ell k}^2$$

5.7 Algorithme *XGBoost*

5.7.1 Introduction

En 2016, Chen et Guestrin[17] ont proposé une version plus optimisée de l'algorithme de *gradient boosting*. Il s'agît de l'algorithme *XGBoost*, ou *«eXtreme gradient boosting»*. Le principe du *XGBoost* est de traiter de manière optimisée la construction du vote majoritaire comme une descente du gradient dans l'espace des fonctions $\mathcal{X} \to R^K$. On observe le résultat à chaque itération d'un apprenant faible *«weak learner»*, corrélé avec le gradient négatif de la fonction objectif. On considère dans cet algorithme que le poids de chacun de ces *«weak learners»* correspond à son impact sur la descente du gradient, alors que pour le *gradient boosting*, ce poids est optimisé à chaque itération de manière à optimiser la fonction objectif. Dans l'algorithme *XGBoost*, on considère le développement de Taylor à l'ordre 2 de la fonction objectif pour calculer le poids optimal, ce qui réduit de manière significative le temps de calcul à chaque itération.

5.7.2 Modélisation des classes

Dans le cadre de la classification, l'algorithme *XGBoost* renvoie une solution binaire pour chaque classe, il s'agît d'un vecteur y à valeurs dans $0, 1$:

$$y_k = \begin{cases} 1 & \text{si} \quad k = y \\ 0 & \text{si} \quad k \neq y \end{cases}$$

En T itérations, l'algorithme construit une fonction $f^{(T)}$ qui effectue des prédictions dans l'espace R^K. On obtient ensuite une prédiction dans $\mathcal{Y}$ avec cette relation :

$$\hat{y} = \underset{k}{\text{argmax}} \left(f_k^T(x) \right)$$

On note $f^{(T)}$ comme suit :

$$\forall x \in \mathcal{X}, f^{(T)}(x) = \sum_{t=1}^{T} h^{(t)}(x)$$

Avec $h^{(t)} : \mathcal{X} \to R^K \forall t \in \{1, \ldots, T\}$

Chaque apprenant faible *Weak Learner* est créé à partir de K arbres de régression entraînés indépendamment. On note $c_{k,j}$ les prédiction de chacune des feuilles j parmi les N feuilles de l'arbre h_k. Chacun de ces arbres peut être exprimé par $q : \mathcal{X} \to \{1, \ldots, N\}$ et un vecteur $c \in R^N$ issu des prédictions de chaque feuille. On note les prédictions d'un arbre de régression

h de la manière suivante :

$$\forall x \in \mathcal{X}, h(x) = c_{q(x)}$$

Avec $q(\mathbf{x})$ l'indice de la feuille de destination de x.

5.7.3 Différentiabilité

La fonction de perte l de l'algorithme *XGBoost* est deux fois différentiable. Cette fonction l prédit un vote majoritaire $f^{(T)}$ minimisant le risque R_S^ℓ de perte. Pour chaque itération t, on a $f_k^{(t)} = f_k^{(t-1)} + h_k^{(t)}$. On approche R_S^ℓ par un développement de Taylor d'ordre 2 de la manière suivante : $\forall k \in \mathcal{Y}$,

$$R_S^\ell\left(f^{(t)}, k\right) \simeq \frac{1}{m}\sum_{i=1}^{m} \ell\left(f_k^{(t-1)}\left(x_i\right), y_{i,k}\right) + a_{i,k}^{(t)} h_k^{(t)}\left(x_i\right) + \frac{1}{2} b_{i,k}^{(t)}\left[h^{(t)}\left(x_i\right)\right]^2$$

Avec $a_{i,k}^{(t)}$, la dérivée $\frac{\partial \ell\left(f_k^{(t-1)}(x_i), y_{i,k}\right)}{\partial f_k^{(t-1)}(x_i)}$ lors de l'itération t, et $b_{i,k}^{(t)}$ la dérivée seconde, toujours à l'itération t : $\frac{\partial^2 \ell\left(f_k^{(t-1)}(x_i), y_{i,k}\right)}{\partial f_k^{(t-1)}(x_i)^2}$.

Soient $k \in \mathcal{Y}, I_{k,j} = \{i : q_k\left(x_i\right) = j\}$ les indices des feuilles j pour chaque observation. On a, $\forall k \in \{1, \ldots, K\}$

$$\tilde{R}_S^\ell\left(f^{(t)}, k\right) \simeq \frac{1}{m}\sum_{i=1}^{m}\left[a_{i,k}^{(t)} h_k^{(t)}\left(x_i\right) + \frac{1}{2} b_{i,k}^{(t)}\left[h_k^{(t)}\left(x_i\right)\right]^2\right]$$

car $\sum_{i=1}^{m} \ell\left(f_k^{(t-1)}, y_{i,k}\right)$ ne dépend pas de $h^{(t)}$. Il s'agît ainsi de la nouvelle fonction à minimiser par chaque apprenant faible. On introduit également un terme de régularisation $\Omega\left(h^{(t)}\right)$. La fonction objectif devient alors $\forall k \in \{1, \ldots, K\}$:

$$F\left(f^{(t)}, k\right) \simeq \frac{1}{m}\sum_{i=1}^{m}\left[a_{i,k}^{(t)} h_k^{(t)}\left(x_i\right) + \frac{1}{2} b_{i,k}^{(t)}\left[h_k^{(t)}\left(x_i\right)\right]^2\right] + \Omega\left(h_k^{(t)}\right)$$

A partir de ces informations, on peut présenter le principe général du *XGBoost* de la manière suivante :

Algorithme 7 : Algorithme *XGBoost* pour la classification

Entrée : Nombre d'itérations T

Entrée : $S = \left\{(x_i, y_i), \forall i \in \{1, \dots, m\}, x_i \in \mathcal{X}, y_i \in \{0,1\}^K\right\}$

Entrée : La fonction de perte l

Entrée : Un apprenant faible *Weak Learner* WL prenant en entrée $\{(x_i, a_i, b_i), \forall i \in \{1, \dots, m\}\}$, retournant h, qui minimise $F\left(f^{(t)}, k\right)$.

1. On initialise $f^{(0)} = 0$

pour *t=1 à T* **faire**

  pour *i=1 à m* **faire**

    - **(i)** Calculer a_i et b_i.

  fin

  - **(i)** Construire $S' = \{(x_i, a_i, b_i), \forall i \in \{1, \dots, m\}, x_i \in \mathcal{X}\}$.
  - **(ii)** Construire l'arbre $h^{(t)} \leftarrow WL(S')$.
  - **(iii)** $f^{(t)} = f^{(t-1)} + h^{(t)}$

fin

Résultat : $\mathbf{f}^{(T)}$

5.7.4 Sur-apprentissage

Dans le cadre du *XGBoost*, on utilise $\Omega(h_k)$ pour contrôler la profondeur des arbres ainsi que l'amplitude des poids de chacun des arbres. On note pour chaque arbre h, muni de N feuilles et d'un vecteur c d'indices de confiance des prédictions de ses feuilles le terme Ω de la manière suivante :

$$\Omega(h) = \gamma N + \frac{1}{2}\lambda \|c\|_2^2$$

Avec $\gamma \geq 0$ et $\lambda \geq 0$.

γN permet de limiter la convergence des apprenant faibles vers des arbres de grande taille. $\frac{1}{2}\lambda\|c\|_2^2$ les empêche de renvoyer des intervalles de confiance trop élevés. On dispose également de moyens permettant d'atténuer la contribution de ces apprenant faibles dans $f(T)$, pour chaque itération t, on peut modifier la règle de mise à jour de $f(t)$ comme suit :

$\forall x \in \mathcal{X}$

$$f^{(t)}(x) = f^{(t-1)}(x) + \eta h^{(t)}(x)$$

Avec $\eta \in]0; 1]$, le taux d'apprentissage.

Un taux d'échantillonnage τ est également utilisé par *XGBoost*. Il s'agît du rapport entre le nombre d'observations sélectionnées dans l'échantillon d'apprentissage et le nombre d'observations total de ce même échantillon. Selon l'échantillonnage effectué par l'algorithme, il est possible qu'une feuille d'un des arbres de régression comporte un nombre trop faible d'observations, causant du sur-apprentissage. l'algorithme *XGBoost* permet d'éviter ce phénomène en imposant des poids minimaux lors de l'affectation des observations à chaque sous-branche des arbres. L'échantillonnage effectué par *XGBoost* via le paramètre τ permet pour chaque arbre d'apprendre sur un échantillon différent des données de manière à améliorer la justesse du modèle. Cet échantillonnage est réalisé lors de chaque itération (où les observations sont utilisées pour l'apprentissage de tous les nœuds internes des arbres), à chaque niveau de l'arbre (pour l'apprentissage des nœuds de même niveau) et à chaque tentative de création de nouveau nœud interne (les observations sont ré-échantillonnées avant chaque création de nœuds).

L'ensemble des hyper-paramètres de l'algorithme *XGBoost* peut-être optimisé par validation croisée, ce qui rend les temps de calcul globalement longs. Implémentable sur R ou encore Python, certaines librairies comme « *XGBoost* » possèdent des fonctionnalités d'optimisation de ces hyper-paramètres utilisant la puissance du GPU ou des coeurs de la machine pour paralléliser les calculs.

6 Estimation des classes de sinistralité

A partir des classes nouvellement crées par l'algorithme des *K-Prototypes*, l'utilisation d'algorithmes d'apprentissage statistique permettra l'association des nouveaux sinistres à chacune de ces nouvelles classes. L'utilisation des méthodes *Random Forest* et *XGBoost* permettra de déduire un modèle optimal parmi ces méthodes. Ce modèle optimal sera le modèle ayant obtenu la meilleure performance au sens de la justesse par validation croisée sur un échantillon de Test. Enfin, ce modèle optimisé constituera un modèle de référence et sera utilisé sur notre base de données de sinistres survenus en 2019 (figure 28).

6.1 Préparation des données

Les données ont été segmentées en un échantillon d'apprentissage, de validation et de test à, respectivement, $60\%, 20\%$ et 20%. On observe dans notre base de données segmentée la distribution des classes suivante :

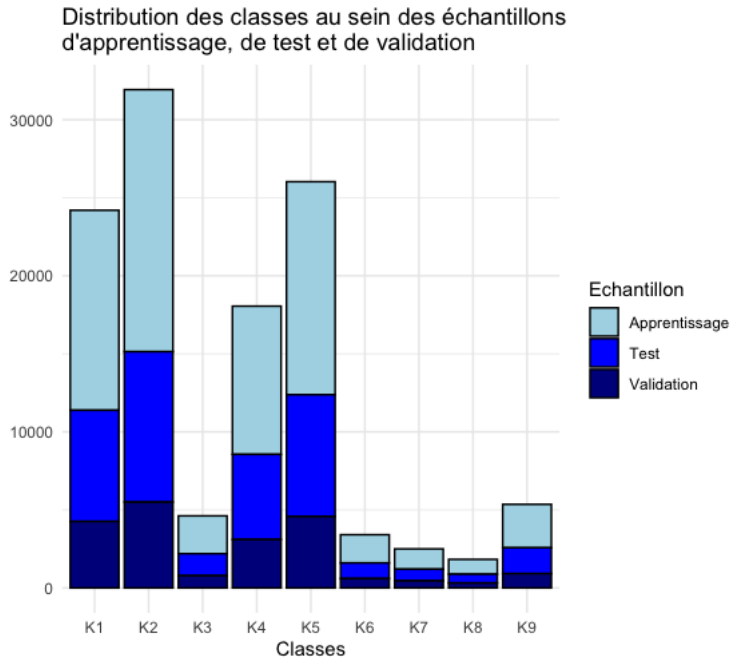

Figure 31 – Répartition des classes de sinistres au sein de chacun des échantillons partitionnés

Nous entraînerons nos différents modèles sur la base d'apprentissage, par méthode de validation croisée sur l'échantillon de validation puis l'échantillon de test.

6.2 Calibrage des modèles

Les modèles utilisés possèdent un ou plusieurs paramètres dits de tuning qui feront l'objet d'une optimisation visant à maximiser la justesse sur l'échantillon de validation.

6.2.1 Calibrage du *Random Forest*

Le nombre d'arbres ainsi que leur profondeur sont des paramètres influant sur la qualité du modèle. On pourrait tout d'abord s'intéresser à l'évolution de la performance du modèle en fonction de ces divers paramètres.

Nombre d'arbres

Tout d'abord, on se concentre sur le nombre d'arbres du modèle, avec ci-dessous l'évolution de la justesse en fonction de ce paramètre :

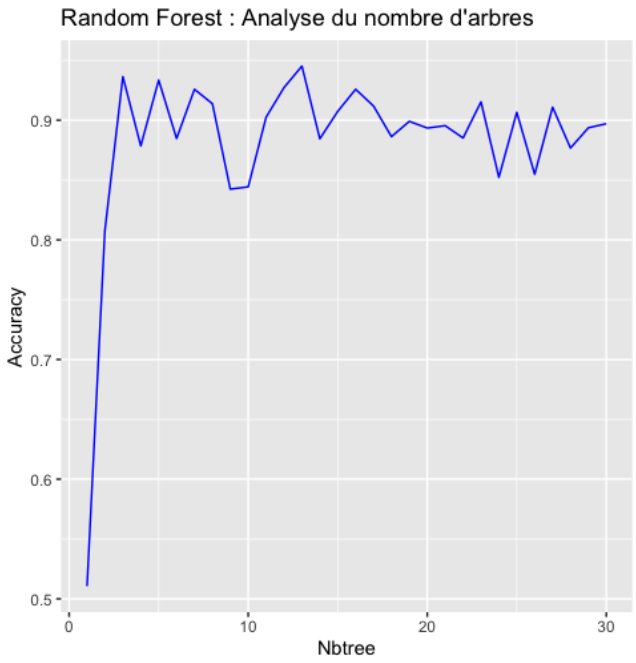

Figure 32 – Évolution de la justesse en fonction du nombre d'arbres (avec $Maxdepth = 5$)

On constate une rapide convergence vers un bon niveau de justesse pour une profondeur fixée des arbres.

Profondeur des arbres

On fixe désormais le nombre d'arbres à 13 (valeur ayant maximisé la justesse à $0,9451$), et on optimise la profondeur des arbres de la même manière.

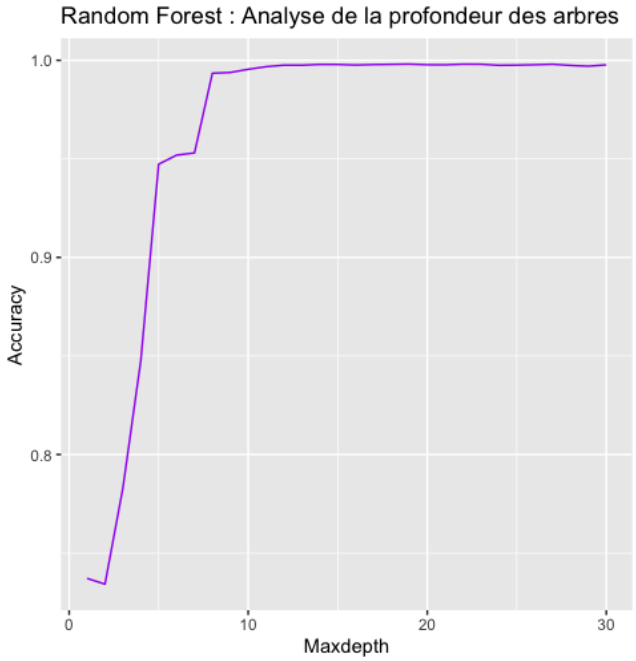

Figure 33 – Évolution de la justesse en fonction du nombre d'arbres (avec $ntree = 13$)

Avec un nombre d'arbre optimisé, la profondeur des arbres permet d'atteindre un niveau de justesse quasi-optimal sur l'échantillon de validation, et ce de manière rapide. Après optimisation séparée de ces deux paramètres, on retient un premier modèle optimisé avec une justesse de $0,9980$, avec $Ntree = 13$ arbres de profondeur maximale $Maxdepth = 19$.

Optimisation conjointe du nombre d'arbres et de la profondeur des arbres

On propose désormais de s'intéresser à l'évolution de ces deux paramètres de manière simultanée afin d'en trouver une combinaison permettant d'attribuer correctement l'ensemble des sinistres à sa classe de sinistralité. Une grille de paramètre a donc été créée représentant une combinaison de $M = 900$ modèles *Random Forest*. On observe les résultats suivants :

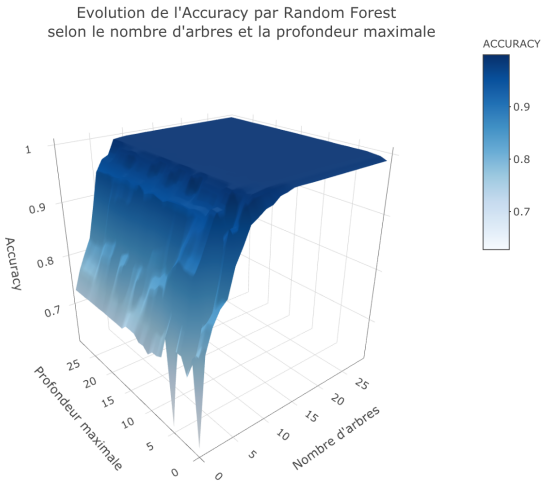

Figure 34 – Évolution de la justesse en fonction du nombre d'arbres et de leur profondeur

Le nombre d'arbres est le paramètre ayant le plus d'impact sur la qualité du modèle. On constate qu'à partir de 10 arbres, on obtient une performance optimale en terme de justesse, l'optimisation au-delà de ce seuil permet d'atteindre des niveaux de performance plus fins pour la classification. Pour un $Ntree = 21$ arbres et une profondeur maximale $Maxdepth = 15$, on obtient une justesse de $0,9986$.

L'optimisation de ces paramètres a permis de passer d'un modèle dit "Naïf" à faible pouvoir prédictif à un modèle classifiant correctement la quasi-totalité de notre échantillon de test. En effectuant une prédiction à l'aide d'un premier modèle non-optimisé puis d'un modèle optimal, sur l'échantillon de test, on peut constater l'amélioration significative de la qualité du modèle *Random Forest* à travers les matrices de confusion respectives :

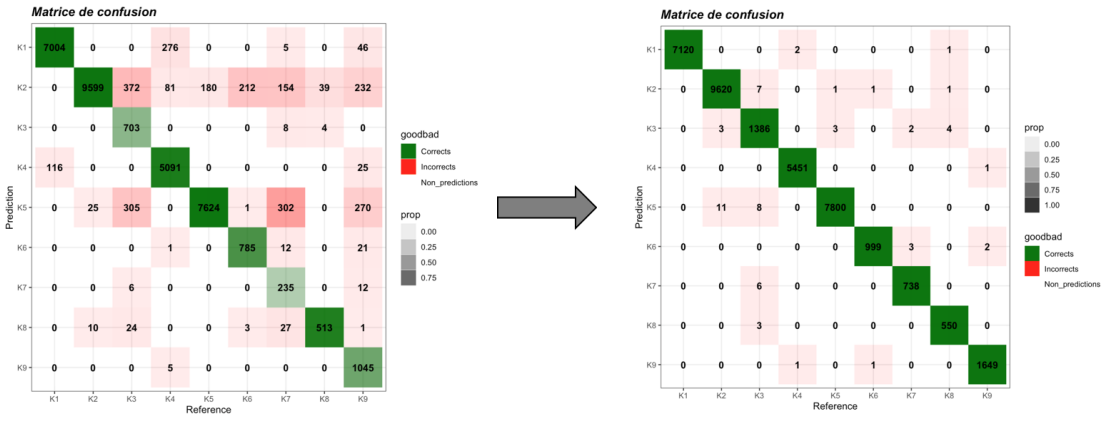

Figure 35 – Matrices de confusion de deux modèles : Modèle Naïf ($Justesse = 0,9215$) avec $Ntree = 10$ et $Maxdepth = 5$ vers un modèle optimal ($Justesse = 0,9982$) avec $Ntree = 21$ et $Maxdepth = 15$

6.2.2 Utilisation de l'algorithme *XGBoost*

Connu pour sa puissance et la qualité de ses modèles, l'algorithme *XGBoost* se présente également comme un algorithme complexe de par sa complexité d'optimisation. Cet algorithme dispose de nombreux paramètres à optimiser, le plus souvent encore par validation croisée, ce qui peut nécessiter des temps d'exécution très longs. En pratique, on procède à une recherche de paramètres optimaux à l'aide d'un maillage de paramètres plus ou moins important. Cette recherche de la combinaison de paramètres optimale qui maximise la justesse pourra être implémentée à l'aide de calcul parallèle, de manière à utiliser des grilles d'hyper-paramètres plus importantes.

Hyper-Paramètres utilisés :

Nous proposons pour notre problème de classification d'optimiser les paramètres suivants :

- ***ntrees* :** de la même manière que précédemment, le nombre d'arbres.
- ***Learning Rate :*** ou taux d'apprentissage, paramètre permettant d'éviter le sur-apprentissage, il n'est pas supposé être linéaire mais peut s'interpréter comme le pas effectué à chaque itération. A titre d'exemple, un taux d'apprentissage égal à 1 implique qu'un modèle à 50 itérations s'exécute 50 fois, là où avec un taux d'apprentissage égal à $0,5$, il s'exécuterait 100 fois. En d'autres termes, le modèle s'exécute sur un espace discrétisé du nombre d'itérations et obtiendrait davantage de chance de rencontrer un optimum. Un taux d'apprentissage plus faible permet donc d'améliorer la qualité du modèle mais peut par conséquent réduire son temps d'exécution. Il est donc conseillé de trouver le bon compromis entre nombre d'arbre et taux d'apprentissage.
- ***Maxdepth :*** ou profondeur des arbres. De la même manière que précédemment, trop augmenter la valeur de ce paramètre augmenterait la complexité du modèle et pourrait amener inévitablement vers du sur-apprentissage.
- ***Sample Rate :*** ou taux d'échantillonnage des observations. Indique le nombre d'observation de l'échantillon d'apprentissage que doivent utiliser les arbres constituant le modèle. Une valeur à $0,7$ indiquerait aux arbres de n'utiliser que 70% des observations tirées aléatoirement de la base d'apprentissage pour entraîner le modèle. Il s'agît là aussi d'un moyen d'éviter le sur-apprentissage.
- ***Col Sample Rate :*** ou taux d'échantillonnage des variables. De manière assez similaire au *sample rate*, ce paramètre vise à tirer aléatoirement parmi les variables disponibles dans la base d'apprentissage un nombre donné de variables (la moitié des variables si ce

paramètre vaut $0,5$). Ce paramètre se décline en plusieurs paramètres de la même famille, comme le *Col Sample by Tree*, tirant un nombre de variables donné pour chaque arbre du modèle, ou bien encore le *Col Sample by Node*, indiquant le nombre de variables à retenir à chaque nœud des arbres.

De nombreux autres paramètres sont disponibles selon la librairie utilisée pour implémenter l'algorithme *XGBoost*. On distingue des paramètres propres au modèle ainsi que des paramètres tournés vers les critères de performance à optimiser, ou bien la méthode d'apprentissage.

Nous optimiserons dans un premier temps chacun de ces paramètres de manière individuelle, en retenant le paramètre maximisant la justesse sur l'échantillon de validation :

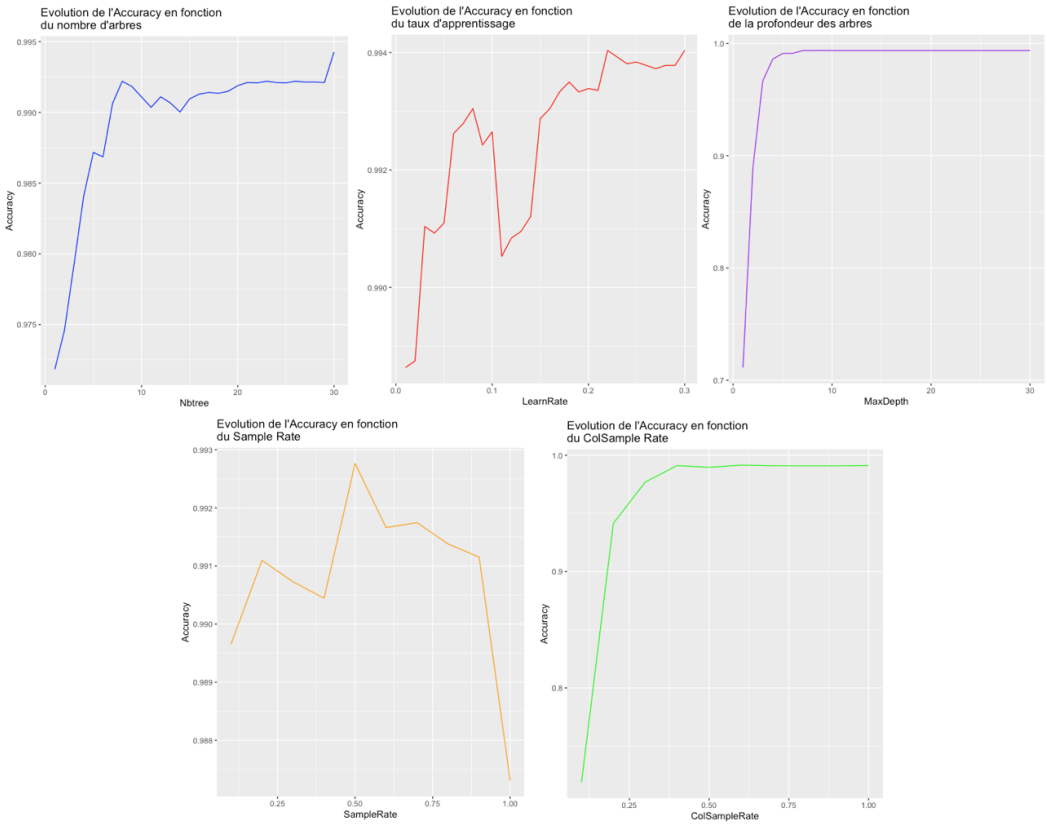

Figure 36 – Évolution de la justesse, individuellement en fonction des hyper-paramètres

En procédant à l'optimisation de manière individuelle de chacun de ces paramètres, on retiendrait une première optimisation du modèle avec les paramètres suivants :

- $Ntree = 30$
- $LearnRate = 0,22$
- $Maxdepth = 9$

- $SampleRate = 0,5$
- $ColSampleRate = 0,6$

Nous retiendrons pour la suite un taux d'apprentissage de $0,22$ et fixerons le nombre d'arbre à $ntree = 50$.

Recherche par grille de la combinaison d'hyper-paramètres optimale :

On s'intéresse maintenant au comportement de la justesse en fonction de ces hyper-paramètres, en effectuant une recherche de combinaison optimale à l'aide d'une grille de paramètre constituée de 370 combinaisons possibles.

Hyper-paramètres	Valeur
Nombre d'arbres	50
Taux d'apprentissage	0.22
Profondeur maximale	(15,20,25,30)
Echantillonnage des observations	(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)
Echantillonnage des variables	(0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)

Figure 37 – Grille d'hyper-paramètres utilisée pour le calibrage du modèle *XGBoost*

On entraîne donc chacun de ces modèles et on évalue les performances respectives au sens de la justesse sur l'échantillon de validation. Il est possible de représenter graphiquement les performances obtenues comme suit :

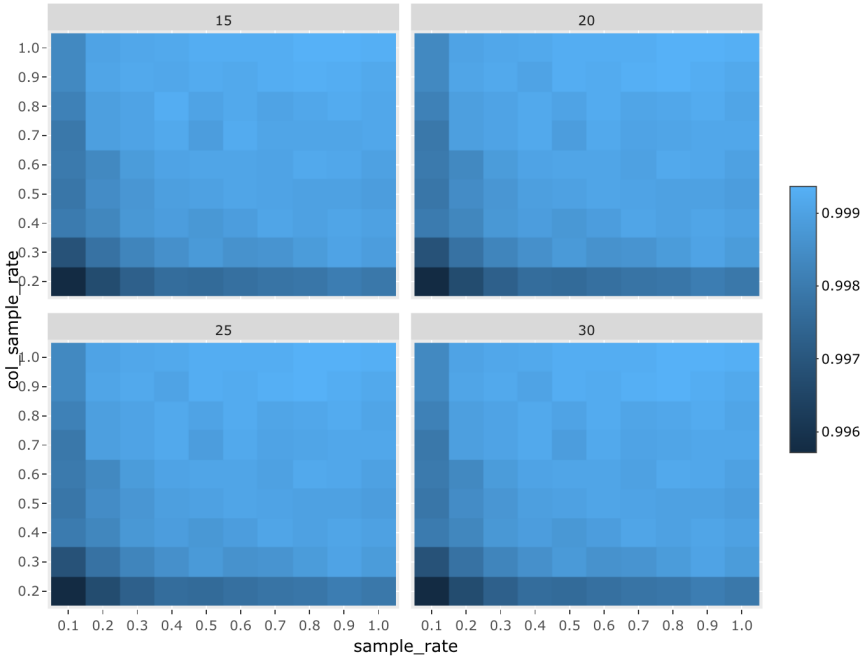

Figure 38 – Justesse en fonction de chacune des combinaisons d'hyper-paramètres

Les performances semblent dépendre principalement des deux paramètres d'échantillonnage. Quelque soit la valeur de la profondeur des arbres (représentées par les 4 secteurs de la figure 38), on observe une performance plus importante dans les parties supérieures droites, là où les taux d'échantillonage sont les plus important. Autrement dit, là où l'ensemble des variables et des observations sont conservées dans la création des arbres. On observe cependant une performance optimale avec une justesse de $0,9994$ pour la combinaison d'hyper-paramètres suivante :

- $Ntree = 50$
- $LearnRate = 0,22$
- $Maxdepth = 15$
- $SampleRate = 0,8$
- $ColSampleRate = 0,9$

En effectuant une prédiction avec un modèle *XGBoost* "naïf" puis avec ce dernier modèle optimisé sur l'échantillon de test, on obtient les matrices de confusion suivantes :

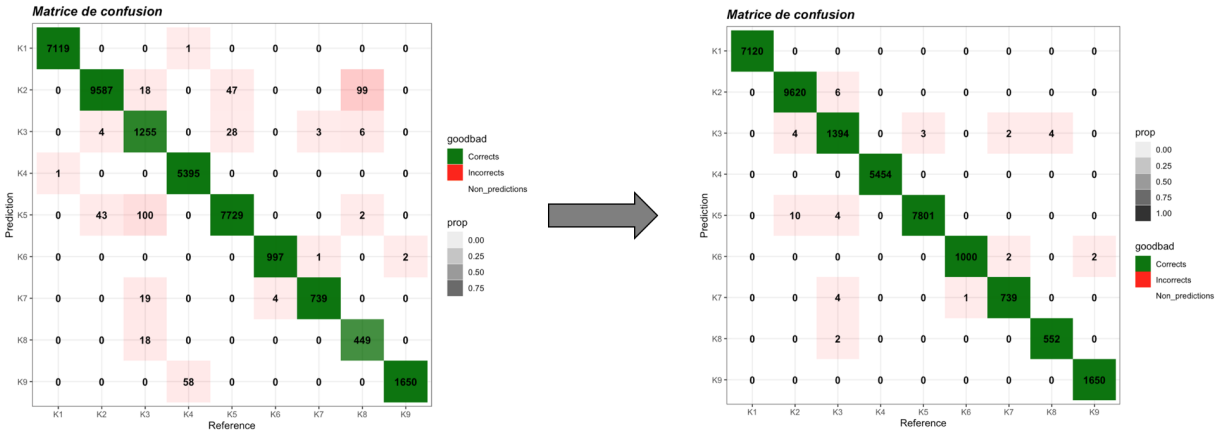

Figure 39 – Matrices de confusion de deux modèles : Modèle Naïf ($Justesse = 0,9872$) avec $Ntree = 50$, $LearnRate = 0.3$, $Maxdepth = 5$, $SampleRate = 0,5$ et $ColSampleRate = 0,5$ vers un modèle optimal ($Justesse = 0,9988$) avec $Ntree = 50$, $LearnRate = 0,22$, $Maxdepth = 15$, $SampleRate = 0,8$ et $ColSampleRate = 0,9$

Le modèle optimisé par validation croisée sur la grille d'hyper-paramètres permet d'obtenir un très bon niveau de prédiction, seules 44 observations ont été mal attribuées sur l'échantillon de test, là où 454 observations avaient été mal prédites dans le premier modèle.

6.3 Modèle retenu

Après avoir calibré les modèles *Random Forest* et *XGBoost* sur la base d'apprentissage et calculé les performances sur l'échantillon de test, on souhaite retenir le modèle optimal, à savoir celui qui attribue le plus justement chaque sinistre à sa classe de sinistralité. La calibration des modèles pour ces sinistres corporels s'est effectuée sur 900 modèles *Random Forest* et 370 modèles *XGBoost*. Les modèles retenus pour chacun de ces algorithmes ont obtenu les performances suivantes :

Modèle	Accuracy du modèle initial	Accuracy du modèle optimisé	Temps d'optimisation
Random Forest (900 modèles)	0.9215	0.9982	4 heures
XGBoost (370 modèles)	**0.9872**	**0.9988**	13 heures

Figure 40 – Résumé des performances de classification des modèles d'apprentissage statistique.

Les deux modèles parviennent à une classification de très bonne qualité sur l'échantillon de test. On retiendra le modèle optimisé *XGBoost* comme modèle final afin d'attribuer les classes aux sinistres Corporels survenus en 2019. Le modèle *Random Forest* aurait également pu être retenu car il présente l'avantage d'être moins complexe, plus facile d'interprétation et plus rapide à optimiser. Le modèle *XGBoost*, bien que difficilement interprétable et assez long à calibrer est le candidat idéal car il parvient à une classification quasi-parfaite. Il est important de remarquer qu'une fois calibrés, les modèles sont tous deux assez rapides pour prédire sur un échantillon indépendant.

A partir du modèle *XGBoost*, on estime les classes pour chacun des sinistres corporels de 2019, on retient la distribution suivante des classes :

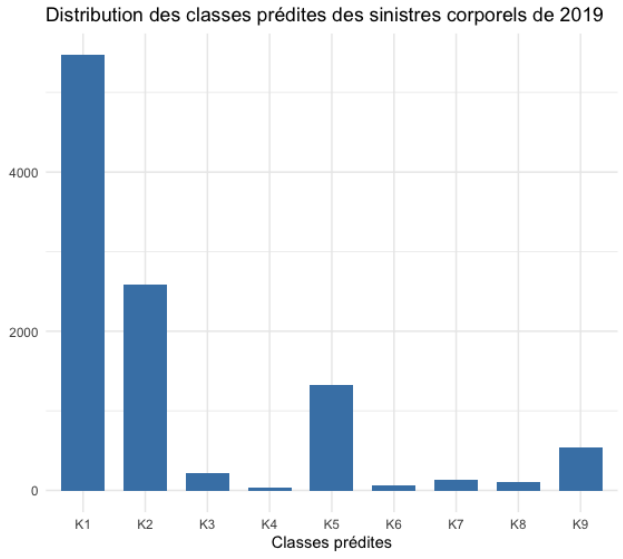

Figure 41 – Distribution des classes prédites pour les sinistres de 2019.

7 Généralisation à la sinistralité matérielle

7.1 La garantie Incendie / Vol

Les charges ultimes des sinistres incendie et vol automobiles sont estimées simultanément dans la segmentation actuelle. A partir des sinistres survenus depuis 2017, on pourrait proposer une segmentation alternative basée sur l'ensemble du processus de *clustering* puis de réaffectation des classes aux sinistres les plus récents.

Nous disposons d'un échantillon de près de 40 000 sinistres depuis début 2017 et nous les observons tous les 4 mois, de manière à avoir un développement quadrimestriel. Ce type de sinistre ayant une gestion plus rapide que les sinistres corporels, il n'est pas nécessaire de disposer d'une profondeur plus importante. Nous utiliserons comme précédemment l'algorithme des *K-Prototypes* afin de créer des sous-classes de sinistralité homogènes de la garantie Incendie/Vol. Les variables utilisées seront celles-ci :

- **Réserve d'ouverture**
- **Groupe SRA du véhicule**
- **Classe SRA du véhicule**
- **Valeur du véhicule potentiellement épargnée par l'incendie**
- **Département de survenance**
- **Présence d'un litige ou contestation**
- **Type de sinistre**
- **Indicateur véhicule neuf**
- **Gravité du sinistre**
- **Réouverture du sinistre**
- **Présence d'investigation ou non**
- **Situation du véhicule**
- **Situation du véhicule au moment du sinistre**
- **Développement Individuel Pondéré de charge**
- **Premier coefficient de passage individuel**

- **Type de véhicule**
- **Type d'usage**

7.1.1 Charges ultimes et tests de validation

On commence par estimer les charges ultimes pour cette garantie avec la segmentation actuelle, soit toute la garantie simultanément :

Triangle total Incendie Vol

	1	2	3	4	5	6	7	8
Q1_2017	25 768	31 775	31 070	30 478	30 957	31 103	30 948	**30 511**
Q2_2017	30 484	35 295	34 815	34 692	33 707	33 427	33 089	**32 622**
Q3_2017	30 917	39 057	38 833	38 194	37 390	36 892	36 611	**36 094**
Q1_2018	30 386	38 480	38 885	38 444	38 255	38 018	37 728	**37 196**
Q2_2018	35 976	42 061	43 358	42 481	42 032	41 772	41 453	**40 869**
Q3_2018	46 689	63 808	64 031	63 115	62 448	62 062	61 588	**60 719**
Q1_2019	66 119	80 071	80 236	79 089	78 253	77 769	77 175	**76 086**
Q2_2019	90 380	112 168	112 400	110 793	109 622	108 944	108 112	**106 587**

Figure 42 – Triangle de charges de la garantie Incendie/Vol projeté

On effectue ensuite un test de validation afin d'évaluer la qualité de prédiction du *Chain-Ladder* dans ce triangle. Comme précédemment, on retire les deux dernières survenances et développements puis on estime les deux dernières diagonales. On obtient alors les écarts suivants :

	1	2	3	4	5	6
Q1_2017	25 768	31 775	31 070	30 478	30 957	31 103
Q2_2017	30 484	35 295	34 815	34 692	33 707	33 865
Q3_2017	30 917	39 057	38 833	38 194	37 898	38 076
Q1_2018	30 386	38 480	38 885	38 382	38 084	
Q2_2018	35 976	42 061	41 769	41 229		
Q3_2018	46 689	56 765	56 372			

	1	2	3	4	5	6	
Q1_2017	-	-	-	-	-	-	Erreurs :
Q2_2017	-	-	-	-	-	438	438
Q3_2017	-	-	-	-	508	1 184	1 692
Q1_2018	-	-	-	(62)	(171)	-	(233)
Q2_2018	-	-	(1 590)	(1 253)	-	-	(2 842)
Q3_2018	-	(7 043)	(7 659)	-	-	-	(14 702)
						Erreur totale	(15 648)

Figure 43 – test de validation sur le triangle total Incendie/Vol

L'objectif est désormais de trouver un nombre K de classes qui obtient une meilleure performance que cette projection utilisant un unique triangle. On effectue donc un *K-Prototype* sur

l'ensemble des sinistres de cette garantie et on retiendra la segmentation en K classes ayant minimisé l'erreur sur les deux dernières diagonales.

Pour $K = 2, ..., 10$, on obtient les mesures d'intravariances suivantes :

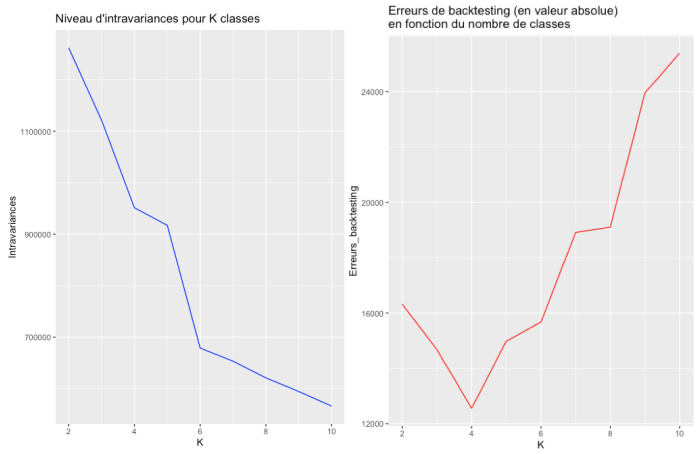

Figure 44 – *Clustering K-Prototypes* sur les sinistres Incendie/Vol

On obtient un niveau d'intravariance satisfaisant pour $K = 6$, mais l'erreur en valeur absolue obtenue par test de validation avec une segmentation en $K = 4$ classes nous oriente vers ce dernier choix. On remarque également qu'au delà de 6 classes, les erreurs augmentent fortement, ce qui est directement lié à une mauvaise estimation du modèle *Chain-Ladder* en présence de triangles avec peu d'informations. A partir de ces $K = 4$ nouvelles classes de sinistralité observées, on peut reconstituer les erreurs de provisionnement sur 4 nouveaux triangles comme suit :

Triangle K1

	1	2	3	4	5	6	
Q1_2017	-	-	-	-	-	-	Erreurs :
Q2_2017	-	-	-	-	-	517	517
Q3_2017	-	-	-	-	247	646	893
Q1_2018	-	-	-	670	218	-	889
Q2_2018	-	-	(574)	(451)	-	-	(1 025)
Q3_2018	-	(3 814)	(4 391)	-	-	-	(8 205)
						Erreur totale	(6 932)

Triangle K2

	1	2	3	4	5	6	
Q1_2017	-	-	-	-	-	-	Erreurs :
Q2_2017	-	-	-	-	-	(231)	(231)
Q3_2017	-	-	-	-	(72)	(25)	(96)
Q1_2018	-	-	-	(709)	(606)	-	(1 315)
Q2_2018	-	-	(732)	(431)	-	-	(1 163)
Q3_2018	-	333	(142)	-	-	-	191
						Erreur totale	(2 614)

Triangle K3

	1	2	3	4	5	6	
Q1_2017	-	-	-	-	-	-	Erreurs :
Q2_2017	-	-	-	-	-	236	236
Q3_2017	-	-	-	-	10	225	235
Q1_2018	-	-	-	(33)	(173)	-	(207)
Q2_2018	-	-	44	(89)	-	-	(45)
Q3_2018	-	(227)	(172)	-	-	-	(399)
						Erreur totale	(180)

Triangle K4

	1	2	3	4	5	6	
Q1_2017	-	-	-	-	-	-	Erreurs :
Q2_2017	-	-	-	-	-	(74)	(74)
Q3_2017	-	-	-	-	346	325	671
Q1_2018	-	-	-	13	410	-	423
Q2_2018	-	-	(355)	(322)	-	-	(676)
Q3_2018	-	(1 774)	(1 402)	-	-	-	(3 176)
						Erreur totale	(2 832)

Erreur Clustering TOTALE	-12 557 K

Figure 45 – *clustering K-Prototypes* sur les sinistres Incendie/Vol

A partir de la segmentation actuelle ainsi que de la nouvelle segmentation issue des 4 classes de sinistralité par *K-Prototype*, on calcule les charges ultimes sur l'ensemble des triangles et on obtient les résultats suivants :

	Q1_2017	Q2_2017	Q3_2017	Q1_2018	Q2_2018	Q3_2018	Q1_2019	Q2_2019
Triangle Global	30 511	32 622	36 094	37 196	40 869	60 719	76 086	106 587
Segmentation 4 Classes	30 511	32 741	36 099	37 262	40 789	60 813	75 890	104 740

Figure 46 – Charges ultimes obtenues par deux types de segmentation

Les charges ultimes obtenues par dates de survenances sont assez semblables jusqu'au Q1 2019 et présentent un écart de 2M€ pour le Q2 2019. Ceci est dû à l'incertitude liée au premier développement du sinistre qui est plus difficile à estimer. Lors des tests de validation, la segmentation par *K-Prototypes* permettait de réduire les erreurs pour le premier développement. On retient donc la segmentation issue du *clustering*.

7.1.2 Affectation des classes aux sinistres récents

Une fois que les classes ont été créées, il est important d'associer chaque nouveau sinistre concerné par la garantie Incendie/Vol à sa classe. De même que pour la garantie corporelle, un modèle *XGBoost* sera utilisé pour prédire ces classes sur un échantillon de test et les associer aux sinistres du Q3 2019.

Nous partitionnons l'échantillon de données de sinistres survenus entre le Q1 2017 et le Q2 2019 en trois échantillons d'apprentissage, de test et de validation avec les proportions respectives de $60\%, 20\%$ et 20%. Pour ce problème de classification à 4 classes, on calibre un modèle *XGBoost* par validation croisée avec la grille de paramètres suivante :

Hyper-paramètres	Valeur
Nombre d'arbres	50
Taux d'apprentissage	0.22
Profondeur maximale	(5,10,15,20,25,30)
Echantillonnage des observations	(0.3,0.4,0.5,0.6,0.7,0.8,0.9)
Echantillonnage des variables	(0.3,0.4,0.5,0.6,0.7,0.8,0.9)

Figure 47 – Grille d'hyper-paramètres utilisée pour la calibration d'un modèle *XGBoost* sur la garantie Incendie/Vol

Nous observons les performances suivantes sur l'échantillon de validation :

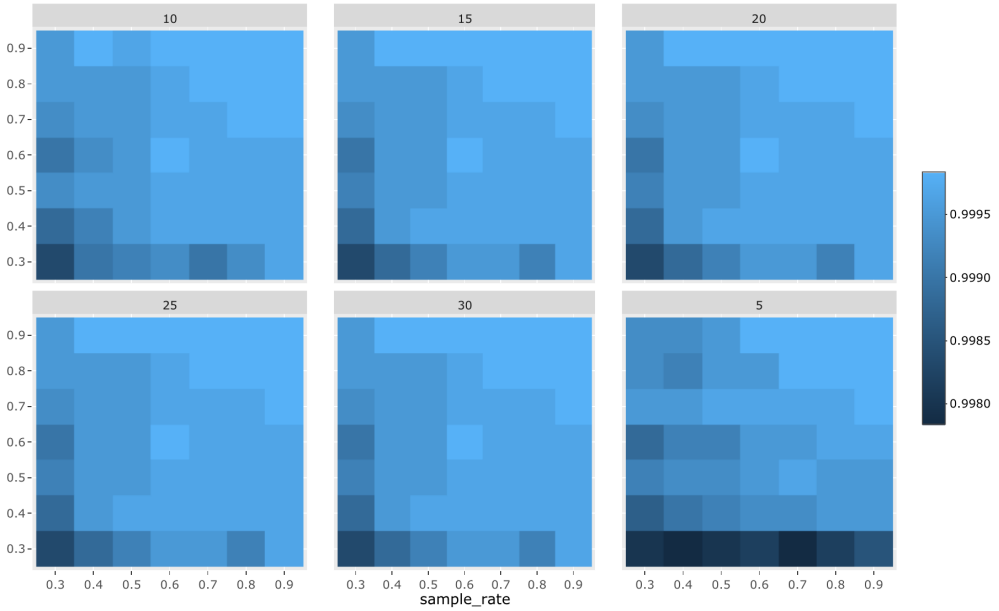

Figure 48 – Performances au sens de la justesse sur l'échantillon de validation des différents modèles de la grille d'hyper-paramètres

Une faible profondeur des arbres avec un taux bas d'échantillonnage des variables entraîne une moins bonne performance sur l'échantillon de validation (secteur inférieur droit de la figure 48). Globalement, les modèles retenus se situent dans les régions supérieures droites de chacun des secteurs du graphe de la performance des modèles. Ce qui correspond à un fort taux d'échantillonnage à la fois des variables et des observations. Le modèle optimal retenu sera le modèle avec les paramètres suivants :

- $Ntree = 50$
- $LearnRate = 0,22$
- $Maxdepth = 15$
- $SampleRate = 0,7$
- $ColSampleRate = 0,9$

Une prédiction des classes de sinistralité peut être effectuée sur l'échantillon de test et comparée avec un premier modèle naïf. La visualisation des performances peut s'effectuer à travers les matrices de confusion ci-dessous :

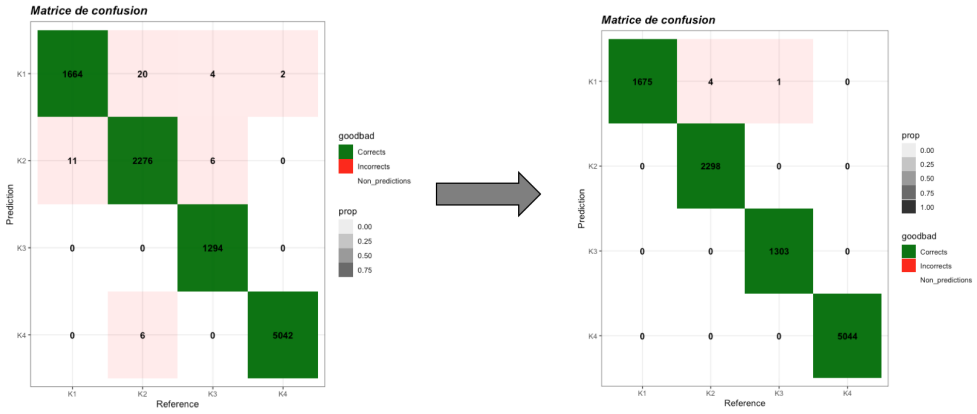

Figure 49 – Matrices de confusion de deux modèles : Modèle Naïf ($Justesse = 0,9952$) avec $Ntree = 50$, $LearnRate = 0,3$, $Maxdepth = 5$, $SampleRate = 0,5$ et $ColSampleRate = 0,5$ vers un modèle optimal ($Justesse = 0,9995$) avec $Ntree = 50$, $LearnRate = 0,22$, $Maxdepth = 15$, $SampleRate = 0,8$ et $ColSampleRate = 0.9$

A l'issue de cette classification, seuls 5 sinistres ont mal été associés aux bonnes classes. Il est désormais possible d'effectuer cette classification sur les sinistres du Q3 2019. On obtient la répartition des classes suivante.

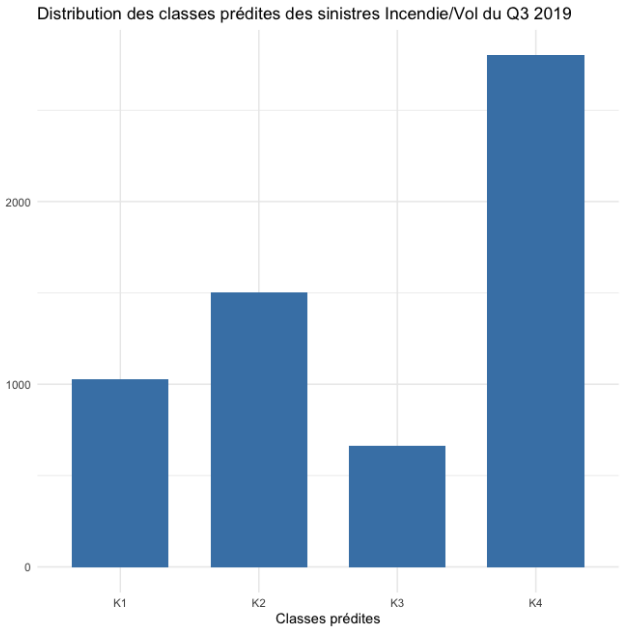

Figure 50 – Répartition des classes de sinistralité prédites pour le Q3 2019

Conclusion

Nous venons de voir qu'une bonne segmentation est primordiale dans le cadre de l'estimation des charges ultimes en provisionnement non-vie. Les modèles agrégés étant toujours fortement utilisés par les équipes actuarielles, ceux-ci nécessitent un respect d'hypothèses en vue de garantir des prédictions de qualité. Les modèles proposés dans le cadre de ce mémoire s'appuient avant tout sur une bonne gestion des données. En effet, là où l'utilisation de méthodes telles que le *Chain-Ladder* ne font intervenir que des informations financières et temporelles liées aux sinistres, les méthodes proposées ici présentent un inconvénient non négligeable qui est le biais potentiellement engendré en présence d'une base de données mal historisée, avec des valeurs manquantes ou inexactes.

Les algorithmes de *clustering* nous ont permit de créer une segmentation alternative qui améliore la performance de la méthode du *Chain-Ladder* à partir de données individuelles de sinistralité. D'un point de vue opérationnel et réglementaire, il est recommandé de segmenter au minimum par branche d'activités. Une bonne gestion du flux de données pourrait ainsi amener par la suite à classifier automatiquement les sinistres en vue d'effectuer le provisionnement sur des groupes de sinistralité différents.

L'extraction d'informations propres au sinistre et aux caractéristiques individuelles de l'assuré nécessitent un travail supplémentaire afin de s'assurer de la fiabilité des informations à disposition. Une attention particulière doit également être portée sur l'interprétabilité de ces méthodes, limite majeure de l'utilisation de tels modèles.

Les méthodes d'apprentissage statistique sont de plus en plus utilisées dans le cadre du provisionnement non-vie. Le provisionnement individuel par exemple en est un très bon exemple, il permet d'estimer un montant de charges ultimes par sinistre et de connaître la charge finale prévisible pour chaque période de survenance en sommant ces montants individuels. Le processus utilisé dans ce mémoire permet de démontrer qu'il est possible tout en utilisant une méthode très classique et utilisée qu'est la méthode *Chain-Ladder*, de trouver des caractéristiques ayant un impact direct sur le développement du sinistre. Une autre méthode aurait pu être implémentée : l'utilisation d'un algorithme supervisé directement sur les données initiales pour prévoir l'impact des variables sur les provisions. Cependant, les données n'étant pas labellisées, il était difficile de définir une variable à expliquer. Il aurait été possible d'effectuer une prédiction des coefficients de développement pour chaque année de survenance, ce qui reviendrait à créer autant de modèles que de coefficients de développement disponibles. L'algorithme des *K-Prototypes* permet donc de labelliser les données en créant une variable catégorielle à expliquer.

Nous avons utilisé ce processus de segmentation et classification de sinistres dans un premier temps pour la Responsabilité Civile Corporelle, puis sur une garantie de dommages matérielle bien précise : la garantie incendie/vol. Une différence majeure entre ces deux approches a été le format d'historisation des données. Là où les sinistres corporels nécessitent un temps de gestion bien plus long, les sinistres de la garantie incendie et vol se liquident bien plus rapidement. Il est donc important de préparer les données en fonction de cette information. Les fréquences d'observation de ces sinistres ne sont plus les mêmes entre ces garanties, il est plus opportun de regarder l'évolution des sinistres des garanties de dommages matériels plus fréquemment, et donc d'avoir des triangles de charge semestriels ou mensuels plutôt qu'annuels.

Les informations propres aux victimes de dommages corporels ont un très bon niveau de fiabilité depuis les premières survenances de l'étude au sein des bases infocentrées d'AXA France. Cependant, les informations propres aux véhicules ont subi une nette amélioration et sont de plus en plus exhaustives depuis 2015. Les variables utilisées lors du processus de segmentation sur des garanties matérielles présentaient donc de nombreuses valeurs manquantes pour les survenances plus anciennes, c'est pourquoi, le cadencement a été modifié afin de pouvoir profiter d'un maximum d'informations.

Entre garanties, les variables d'importance sont différentes. Pour des dommages matériels, les variables propres aux biens sinistrés influeront davantage sur l'homogénéité des classes tandis que pour des dommages corporels ce seront les informations propres aux personnes qui auront une plus grande importance.

Le processus d'extraction des sinistres est donc complexe et nécessite une extrême rigueur. Afin de pouvoir industrialiser de telles méthodes au sein d'une entreprise telle qu'AXA France, un travail supplémentaire serait à ajouter au processus mensuel d'extraction des sinistres. En plus d'extraire les informations financières à date, il serait nécessaire de récupérer au sein de plusieurs bases de données les informations propres à chaque garantie sinistrée.

Un inconvénient majeur de cette extraction serait un changement de gestion de l'information au sein même de ces bases de données. En effet, l'ajout de nouvelles modalités sur certaines variables pourrait impacter la performance des modèles permettant d'associer chaque sinistre à sa classe créée par *clustering*. Un recalibrage serait alors nécessaire en créant de nouvelles classes homogènes incluant ces nouvelles informations. Ce qui pourrait amener à des classes de sinistralité variables d'une année à l'autre, là où une segmentation classique par garantie sinistrée ne connaîtrait pas ce problème.

L'amélioration de la qualité d'estimation des provisions est significative sur des types de sinistralité volatiles telles que la Responsabilité Civile corporelle. Chaque sinistre pouvant se développer de manière imprévisible et s'aggraver de manière inattendue, disposer d'informations supplémentaires individuelles dès le départ peut permettre d'anticiper ces comportements et d'améliorer la qualité des modèles agrégés.

L'avenir des méthodes d'apprentissage statistique pour le provisionnement reste cependant incertain du fait de l'interprétabilité et de la potentielle complexité opérationelle de temps de calculs. Pour l'heure ces méthodes peuvent apporter une seconde opinion à l'actuaire lors de ses analyses et estimations de provisions techniques.

Références

[1] L. Breiman. "Bagging predictors". In : *Mach Learn 24* 2 (1996), p. 123-140.

[2] L. Breiman. "Random Forests." In : *Mach Learn 45* (2001), p. 5-32.

[3] L. Breiman et al. *Classification and Regression Trees*. Taylor & Francis, 1984. isbn : 9780412048418. url : https://books.google.fr/books?id=JwQx-WOmSyQC.

[4] Khalil Ben Fadhel. *Amplification d'arbres de régression compatibles avec l'encodage de la sortie, application à la reconnaissance des images de chiffres manuscrits*. 2019.

[5] Yoav Freund et Robert E Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In : *Journal of Computer and System Sciences* 55.1 (1997), p. 119-139. issn : 0022-0000. doi : https://doi.org/10.1006/jcss.1997.1504. url : https://www.sciencedirect.com/science/article/pii/S002200009791504X.

[6] J. H. Friedman. "Greedy function approximation : A gradient boosting machine". In : *Ann. Statist. 29* (2001), p. 1189-1232.

[7] J. H. Friedman. *Stochastic gradient boosting, Computational Statistics and Data Analysis*. 2002.

[8] Ian Goodfellow, Yoshua Bengio et Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[9] Friedman J. Hastie T. Tibshirani R. *The elements of Statistical Learning - Data Mining, Inference and Prediction*. Springer, 2009.

[10] S. Gey et J. M. Poggi. *Boosting and instabillity for regression trees*. 2002.

[11] M. Murty. "Cluster Analysis on Different Data Sets Using K-Modes and K-Prototype Algorithms". In : t. 249. Déc. 2013. isbn : 978-3-319-03094-4. doi : 10.1007/978-3-319-03095-1_15.

[12] Ricco Rakotomalala. *Gradient Boosting : Technique ensembliste pour l'analyse prédictive Introduction explicite d'une fonction de coût*. 2016.

[13] Brian D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996. doi : 10.1017/CBO9780511812651.

[14] Katrien Antonio Roel Henckaerts et Marie-Pier Côté. *When stakes are high : balancing accuracy and transparency with Model-Agnostic Interpretable Data-driven suRRogates*. 2020.

[15] R. Schapire. *The boosting approach to machine learning. An overview, MSRI workshop on non linear estimation and classification*. 2002.

[16] Mack T. “Distribution-free calculation of the standard error of chain-ladder reserve estimates”. In : *ASTIN Bulletin* 23 (1993), p. 213-225.

[17] Carlos Guestrin Tianqi Chen. *XGBoost : A Scalable Tree Boosting System*. 2016.

[18] A. Le Tesson A. Lenain S. Samba J. Ung. *Estimation de l'erreur de prédiction dans le cas de l'utilisation d'une combinaison de méthodes pour le calcul de provisions en assurance IARD*. 2014.

[19] ERROCHDI Zachariae. *Méthode d'optimisation du provisionnement en prévoyance ou santé collective : prise en compte des facteurs internes et du contexte économique*. Mémoire d'actuaire ISFA - AXA France, 2016.

[20] Andreas Paul Zischg et al. “Extending coupled hydrological-hydraulic model chains with a surrogate model for the estimation of flood losses”. In : *Environmental Modelling Software* 108 (2018), p. 174-185. issn : 1364-8152. doi : https://doi.org/10.1016/j.envsoft.2018.08.009. url : https://www.sciencedirect.com/science/article/pii/S1364815217306941.

Table des figures

Liste des tableaux

Liste des Algorithmes

Annexes

XGBoost : Algorithme de recherche de la subdivision optimale

Les seuils choisis pour attribuer les observations à chaque feuille dans un nœud pour les arbres de décision de l'algorithme *XGBoost* peuvent être optimisés. Une mesure de gains pour chaque seuil est évaluée à chaque itération. Il est possible de présenter ce processus à travers l'algorithme suivant :

Algorithme 8 : *XGBoost* : Algorithme de recherche de subdivision optimale

Entrée :

-Nombre T de votants faibles

-$S = \{(\mathbf{x}_i, a_i, b_i)\,, \forall i \in \{1, \ldots, m\}, \mathbf{x}_i \in \mathcal{X}, \}$, avec α_i et β_i qui contiennent les résultats des dérivées utilisées dans le développement de Taylor d'ordre 2 pour chaque arbre k

-Une matrice T de taille $m \times d$ qui stocke les indices des observations triées dans l'ordre croissant pour chaque variable

-Indices des observations acheminées à la feuille courante I

-Nombre maximal N de feuilles

-$\lambda > 0$ et $\gamma > 0$

Initialisation :

$gain = 0$, $gain^* = 0$, $\theta^* = 0$, $j^* = 0$

-$G = \sum_{i \in I} a_i, H = \sum_{i \in I} b_i$

pour *j=1 à d* **faire**

- -$G_d = 0, G_l = 0$
- -$P =$ Indice des observations permutées stockées dans la colonne j de la matrice T
- **pour** $i \in P$ **faire**
  - $G_g \leftarrow G_g + a_i, H_g \leftarrow H_g + b_i$
  - $G_d \leftarrow G - G_g, H_d \leftarrow H_d - H_g$
  - $gain = \frac{1}{m}\left(\frac{G_g^2}{H_g + \lambda \cdot m} + \frac{G_d^2}{H_d + \lambda \cdot m} - \frac{G^2}{H + \lambda \cdot m}\right) - \gamma$
  - **si** gain $>$ gain* *et* $x_{i,j} \neq x_{i+1,j}$ **alors**
    - gain* = gain, $\theta^* = \frac{x_{i,j} + x_{i+1,j}}{2}$, $j^* = j$
  - **fin**
- **fin**

fin

Résultat : Un couple $(\varphi, gain^*)$, avec $\varphi(x) = \text{signe}\,(x_{j*} - \theta^*)$, $\forall x \in \mathcal{X}$

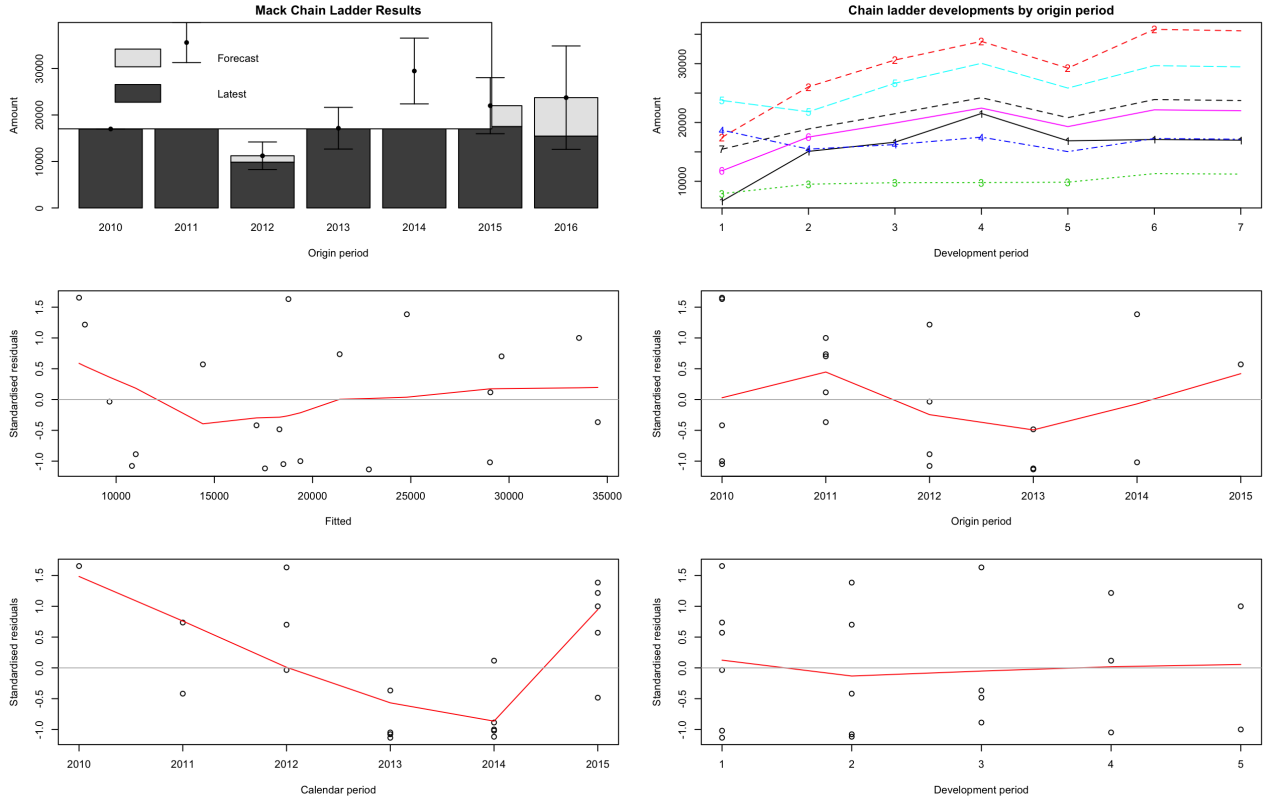

Figure 51 – Package MackChainLadder, analyse des résidus, des facteurs de développement et des estimations sur un triangle de charges de la nouvelle segmentation

Triangle total

	1	2	3	4	5	6	7	8	9
2010	83 595	120 849	147 185	154 278	147 719	152 134	151 611	151 231	155 732
2011	94 736	149 089	160 630	178 305	182 337	194 280	197 461	198 834	204 752
2012	109 583	130 649	133 520	135 818	140 414	140 900	143 696	144 104	148 393
2013	145 249	145 889	153 242	162 922	162 822	171 765	173 687	174 181	179 366
2014	164 106	161 851	166 206	170 303	182 173	189 591	191 713	192 258	197 980
2015	123 184	112 714	125 786	152 339	154 969	161 280	163 084	163 548	168 416
2016	139 956	130 508	146 454	157 587	160 308	166 836	168 703	169 182	174 218
2017	159 733	144 282	156 636	168 544	171 453	178 435	180 431	180 945	186 330
2018	159 144	170 952	185 589	199 698	203 145	211 417	213 783	214 391	220 772

Trangle total Tranches de coût

	1	2	3	4	5	6	7	8	9
2010	83 595	120 849	147 185	154 278	147 719	152 134	151 611	151 231	155 732
2011	94 736	149 089	160 630	178 305	182 337	194 280	197 461	198 834	206 648
2012	109 583	130 649	133 520	135 818	140 414	140 900	143 696	143 791	147 567
2013	145 249	145 889	153 242	162 922	162 822	171 765	173 763	174 384	180 318
2014	164 106	161 851	166 206	170 303	182 173	190 307	192 642	193 578	200 665
2015	123 184	112 714	125 786	152 339	155 015	161 362	163 168	163 637	168 891
2016	139 956	130 508	146 454	157 824	160 636	167 709	169 773	170 502	176 857
2017	159 733	144 282	156 625	168 515	171 342	178 695	180 866	181 504	188 208
2018	159 144	165 988	177 760	190 216	192 642	200 488	202 962	203 263	211 308

Trangle total Clustering

	1	2	3	4	5	6	7	8	9
2010	83 595	120 849	147 185	154 278	147 719	152 134	151 611	151 231	155 732
2011	94 736	149 089	160 630	178 305	182 337	194 280	197 461	198 834	200 908
2012	109 583	130 649	133 520	135 818	140 414	140 900	143 696	144 679	151 847
2013	145 249	145 889	153 242	162 922	162 822	171 765	175 397	174 271	182 508
2014	164 106	161 851	166 206	170 303	182 173	188 008	192 901	192 236	204 472
2015	123 184	112 714	125 786	152 339	151 206	162 754	158 565	161 712	167 288
2016	139 956	130 508	146 454	166 945	164 855	179 483	172 709	175 611	179 098
2017	159 733	144 282	160 998	183 626	180 908	197 752	189 412	192 073	193 731
2018	159 144	149 542	166 778	190 568	187 299	204 500	195 946	198 678	201 171

Figure 52 – Triangles Corporels agrégés, charges estimées selon les différentes segmentations

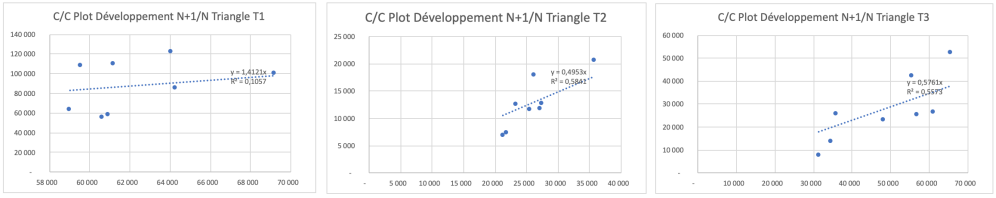

Figure 53 – Tests d'hypothèse d'indépendance des exercices de survenance - Triangles Corporels issus des tranches de coûts

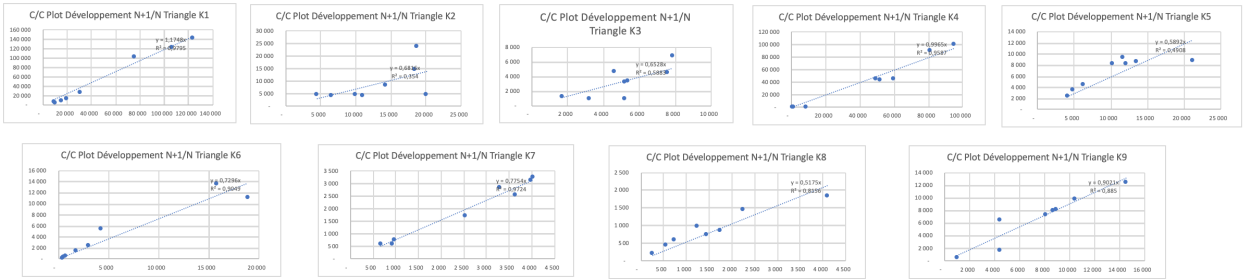

Figure 54 – Tests d'hypothèse d'indépendance des exercices de survenance - Triangles Corporels issus du *clustering*

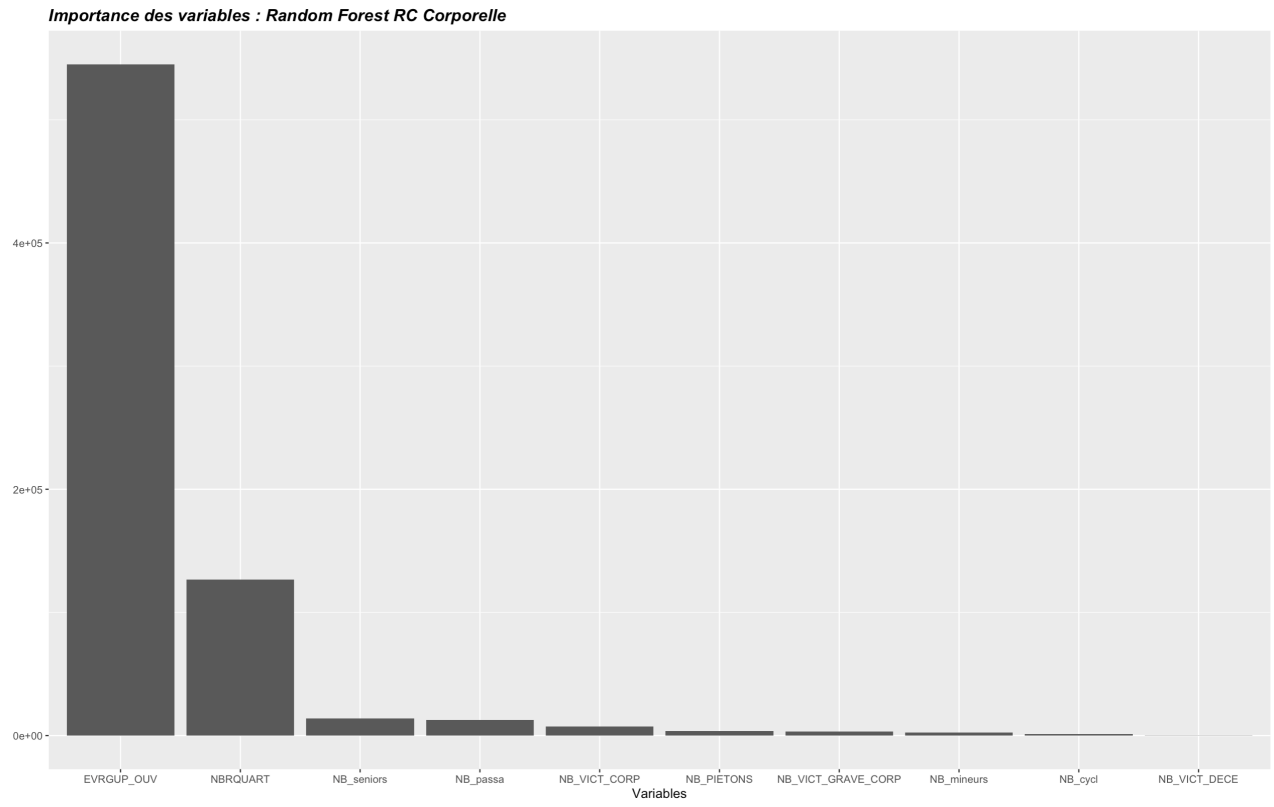

Figure 55 – Importance des variables du modèle *Random Forest* pour la responsabilité civile corporelle

Les algorithmes d'apprentissage statistique proposent une visualisation de l'importance des variables, chacun proposant une mesure différente. Ici le *Random Forest* montre que la variable la plus significative pour la bonne association des classes de sinistralité a été le montant de réserves d'ouverture des sinistres puis la responsabilité du conducteur.

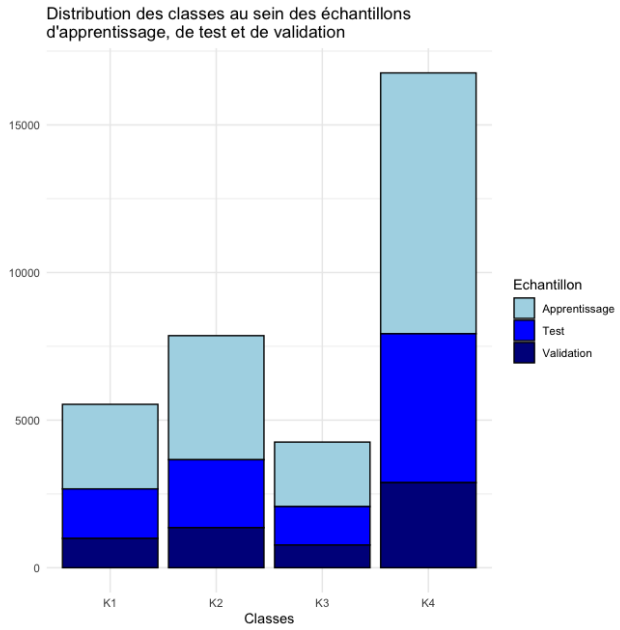

Figure 56 – Distribution des classes au sein des échantillons d'apprentissage, de test et de validation pour la garantie Incendie/Vol

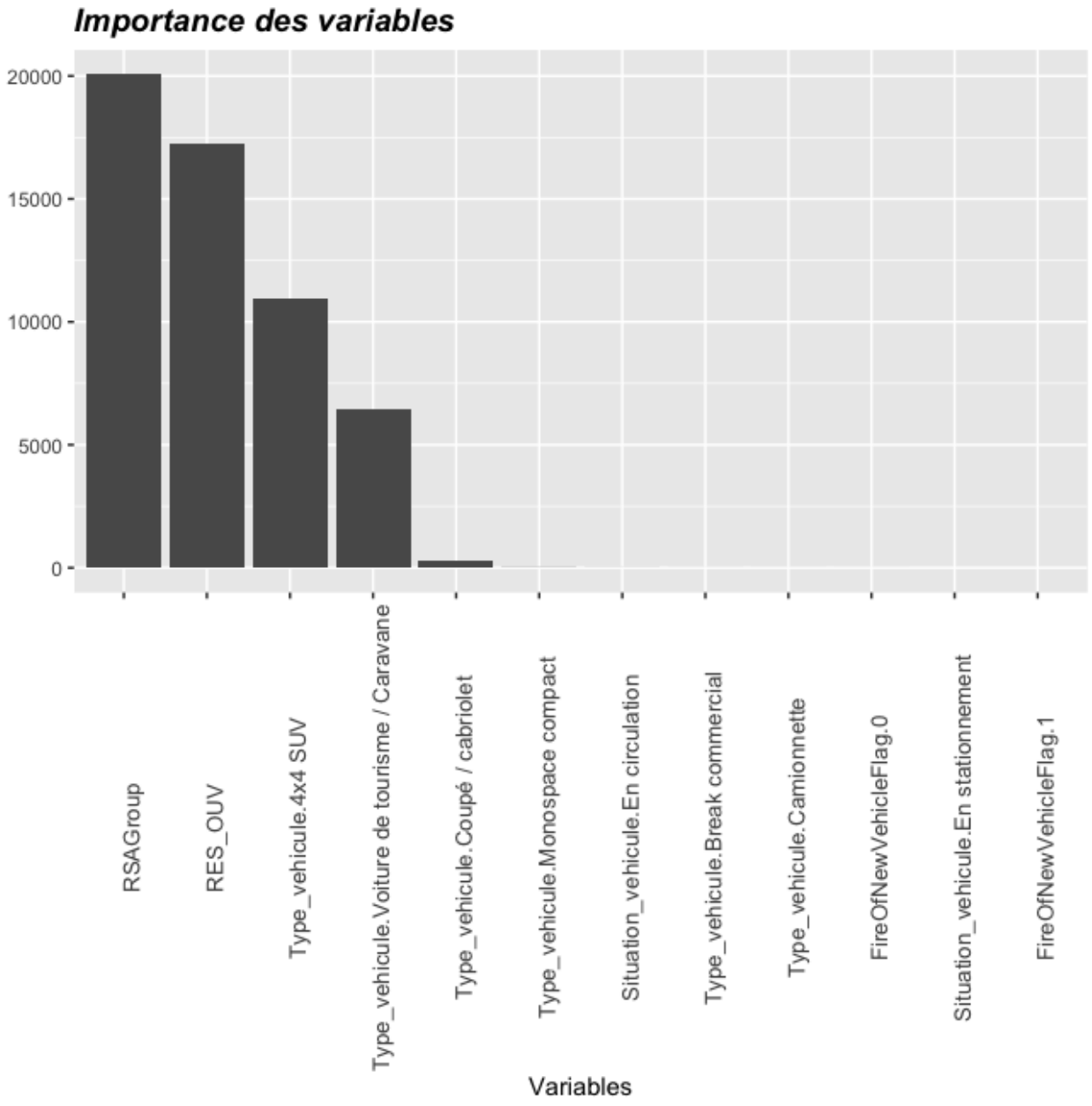

Figure 57 – Importance des variables du modèle *XGBoost* pour la garantie Incendie/Vol