Mémoire présenté devant l'Université de Paris-Dauphine

pour l'obtention du Certificat d'Actuaire de Paris-Dauphine

et l'admission à l'Institut des Actuaires

le 12/09/2022

Par : Sylvain EYRAUD

Titre : Les apports de l'apprentissage statistique dans le provisionnement 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

Entreprise : AxaXL, a division of Axa
Nom : Paul Henri Rastoul
Signature :

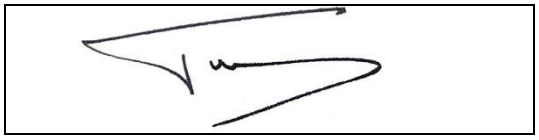
Membres présents du jury du Master Actuariat de Paris Dauphine

Directeur de mémoire en entreprise :
Nom : Paul Henri Rastoul
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)*

Secrétariat :

Signature du responsable entreprise



Bibliothèque :

Signature du candidat

Résumé

Les méthodes classiques de provisionnement, telles que Chain Ladder et Bornhuetter Ferguson, demeurent largement utilisées par les assureurs. Mais de nouvelles techniques basées sur l'apprentissage statistique automatisé ont fait leur apparition ces dernières années dans la littérature actuarielle. Cette étude se propose d'en explorer les apports pour le provisionnement ligne à ligne de la branche dommage d'AxaXL. La première partie est consacrée à la nécessaire préparation des données (chapitre 1). Les principales méthodes de provisionnement seront ensuite étudiées : agrégées d'abord (chapitre 2) puis celles issues de l'apprentissage statistique automatisé (chapitre 3). La modélisation du problème ligne à ligne ainsi que les contraintes qui en découlent y seront exposées et une application concrète sera conduite pour chaque algorithme. Enfin, les dernières techniques d'interprétabilité seront expliquées et appliquées (chapitre 4), ce qui permettra d'identifier les variables explicatives ayant contribuées le plus à la prédiction de la charge totale. Les résultats de l'étude confortent la position de leader du Chain Ladder mais identifient des pistes intéressantes pour la prédiction des développements lointains avec le XGBoost.

Mots clés : non vie, apprentissage statistique automatisé, machine learning, dommage, branche courte, provisionnement ligne à ligne, Python.

Abstract

Aggregated reserving methods, such as Chain Ladder and Bornhuetter Ferguson, continue to be widely used across the insurance industry. But state of the art technics based on Machine Learning emerged a few years back in the actuarial literature. This study aims at exploring what they can bring to the table for the transactional (line by line) reserving of the Property line of business in AxaXL, starting with the necessary data engineering (chapter 1). The most iconic methods will then be reviewed: the aggregated ones first (chapter 2), then those emanating from the Machine Learning world (chapter 3). The line by line modelling along with its constraints will be touched on and practical application will be carried out for each. Lastly, the latest interpretability methods will be explained & applied to the dataset (chapter 4), which will lead to the identification of the most useful variables in predicting the ultimate loss reserve. The study confirms the undisputed status of the Chain Ladder but unveils interesting takeaways from the XGBoost when it comes to predicting development years farther in time.

Key words: non life, Machine Learning, Property, short tail, individual reserving, Python.

Note de synthèse

Contexte

Cette étude a été effectuée sur les données de la ligne « dommage » d'AXA XL, la division grands risques d'AXA. Le faible volume de données disponibles (comparativement à l'assurance personnelle) et la volatilité élevée des sinistres graves ont eu un impact significatif sur la qualité des prédictions. Le caractère volatile a d'ailleurs donné lieu à l'ajout d'une donnée d'exposition (les primes) et une gestion spécifique des valeurs extrêmes (seuil majeurs vs attritionnels et retraitement des loss ratios aberrants) afin d'obtenir un jeu de données plus uniformément distribuées- comme illustré ci contre.

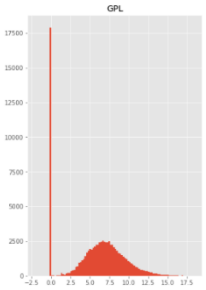


Figure 2: données avant retraitements

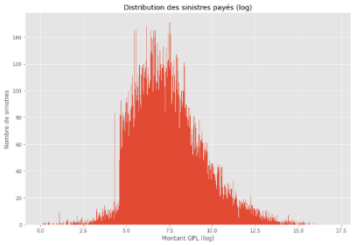


Figure 1: données après retraitements

Des méthodes agrégées à l'apprentissage statistique ligne à ligne

Les méthodes agrégées reposent sur l'utilisation de triangles de développement. Ces triangles reflètent la dynamique de sinistres et les données sont essentiellement de deux natures : soit des montants de sinistres (paiements, charges ou réserves), soit des nombres de sinistres. Leur succès tient à une implémentation simple et des temps de calculs quasi-instantanés.

Les assureurs génèrent néanmoins une quantité importante d'information à la maille sinistre. Leur agrégation en deux dimensions, tel qu'imposée par l'utilisation de triangles agrégés, empêche la prise en compte des caractéristiques individuels du sinistre. L'objectif du provisionnement ligne à ligne est de mettre à profit un ensemble varié de variables explicatives.

La science des données permet de répondre à cet enjeu en conjuguant l'inférence statistique et l'algorithmie. Elle a pour objectif l'exploration, l'analyse et la transformation des données à des fins de prédiction. L'apprentissage statistique automatique quant à lui est un des composants de la science des données et consiste à choisir parmi un ensemble de modèles et hyperparamètres concurrents celui dont la prédiction est la meilleure. A cette fin, les données sont allouées en deux parties disjointes : l'une, usuellement la plus riche en termes de données, sert à entraîner le modèle tandis que la seconde est utilisée pour vérifier la performance du modèle sur des données non vues lors de l'entraînement. Contrairement à la statistique classique, l'apprentissage statistique ne nécessite pas de formuler des hypothèses sur la structure et la distribution des données. Une seule hypothèse est nécessaire : les données à prédire sont générées de façon identiques et indépendantes par un processus aléatoire à partir du vecteur des variables explicatives. Le résultat de cet apprentissage est une fonction qui fait intervenir des variables explicatives et devient de plus en plus complexe à mesure que l'algorithme "apprend", permettant ainsi de capturer les singularités de la structure des données – *par exemple des interactions ou comportements non linéaires*.

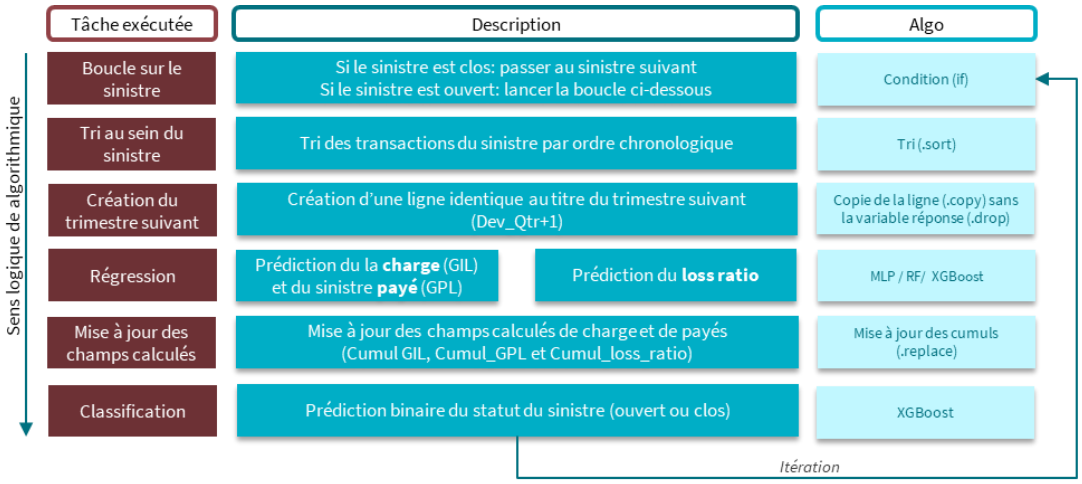
Présentation du modèle

La modélisation utilisée dans cette étude est une adaptation du « cascading », rendu publique dans les travaux ASTIN (B. Harej & al, 2017). Elle consiste, à partir du triangle supérieur initial contenant les données réelles, à prédire itérativement la prochaine diagonale jusqu'à compléter entièrement le triangle inférieur.

La **phase d'apprentissage** couvre 80% des sinistres rattachés aux années de survenance 2010 à 2020. Les trois variables réponses sur lesquelles sont entraînés les modèles sont le montant trimestriel de sinistres payés, la charge totale et le loss ratio. Chaque algorithme est hyperparamètré différemment selon qu'il s'agisse de la forêt aléatoire, du XGBoost ou du réseau de neurones.

La **phase de prédiction** est effectuée sur un échantillon couvrant la seule période 2010-2014 afin d'éviter le phénomène de censure. Le modèle commence par vérifier le statut du sinistre : si ce dernier est ouvert, la boucle est lancée. Les transactions de chaque sinistre sont classées par ordre chronologique, puis les 3 algorithmes de prédiction que sont le réseau de neurones, la forêt aléatoire et le XGBoost sont appliqués pour chaque variable réponse. Les montants cumulés sont ensuite mis à jour, l'ensemble est donné en entrée du classificateur binaire afin de déterminer si le sinistre est toujours ouvert. Puis la boucle passe à la transaction du trimestre suivant jusqu'à extinction du sinistre ou complétion du triangle inférieur.

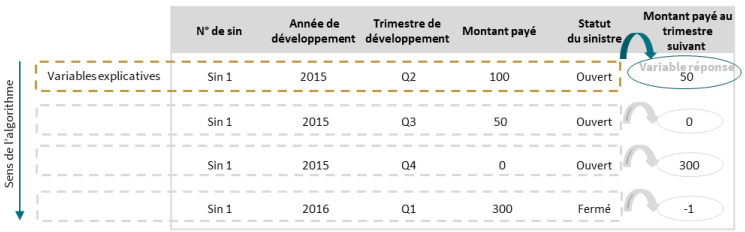
Figure 3: vue générale de la modélisation de l'étude



Construction du modèle

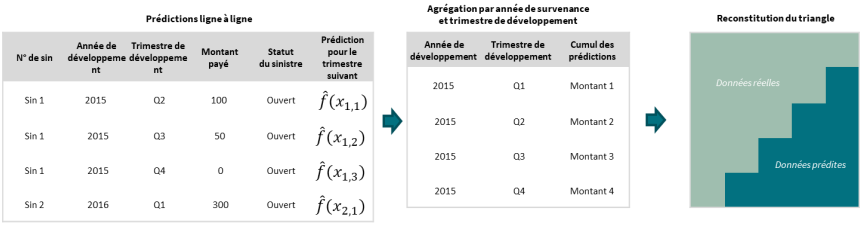
La granularité ligne à ligne du modèle a fortement contraint les phases d'apprentissage et de prédiction. L'étape de « features engineering » a eu pour objectif de (re)générer un niveau important d'information au niveau de la transaction. Le résultat est un algorithme qui se sert de toutes les variables explicatives de la transaction afin de prédire le montant de sinistre au trimestre suivant, comme illustré dans le Tableau 1.

Tableau 1: prédiction schématique de la transaction trimestrielle suivante



Une fois les prédictions à la maille transactionnelle effectuées, les sinistres sont regroupés par année de survenance et agrégés dans le triangle inférieur (zone verte foncée de la Figure 4).

Figure 4:agrégation des prédictions dans le triangle inférieur



Résultats

La performance de chaque algorithme est mesurée à l'aide d'un score de proximité calculé entre les valeurs prédites et les valeurs réelles de chaque transaction sur la période de survenance 2010-2014:

$$Score\ de\ proximité\ (Prox) = \log D_M + |ER_M| - C_M$$

Il est la synthèse de 3 mesures :

- La moyenne arithmétique de la racine des écarts au carré entre les prédictions et les observations autrement dit la RMSE (D_M),
- L'erreur relative du modèle (ER_M),
- La corrélation entre valeurs réelles et valeurs prédites (C_M),

	MAE	Percentage	RMSE	ER	C_M	Prox
Genins	1332157.135	1042.125	1772690.422	0.171	0.811	13.748
Linear Regression	1782986.805	1228.840	2779592.834	0.893	0.249	15.483
XGBoost	1799133.224	891.448	2813136.187	1.364	0.478	15.736
MLP	1919339.938	936.736	3058334.210	1.575	-0.082	16.591
Random Forest	2209858.778	1135.559	3251461.673	2.350	-0.145	17.490

Tableau 2: score de proximité sur les sinistres payés par algorithme (classé du plus au moins performant)

Le Tableau 2 classe les algorithmes par score de proximité, du meilleur au moins bon. Premier constat : le Chain Ladder (ici « Genins ») continue de régner en maître dans la prédiction des sinistres. Il combine à la fois l'écart relatif le plus faible (17%) et le meilleur score de proximité. Le XGBoost est troisième au classement : l'étude des prédictions par année de développement montre notamment qu'il est capable de capter des liens temporels que les autres algorithmes ne perçoivent pas. La régression linéaire performe bien au global grâce à une bonne prédiction sur les 2 premières années de développement mais se fait distancer les années suivantes tout comme le Chain Ladder.

Interprétabilité

Malgré une démocratisation rendue possible par l'explosion des puissances de calcul, l'apprentissage statistique automatique en assurance ne s'est pas encore généralisé. L'actuaire doit prévenir le biais et contenir la variance de son modèle tout en s'assurant de l'adhésion de ses parties prenantes. Malheureusement, les meilleures prédictions effectuées sur de larges bases de données sont souvent le fruit d'une grande complexité, inhérente aux algorithmes ensemblistes ou d'apprentissage profond, que les experts eux-mêmes ont du mal à interpréter.

(Ribero & al, 2016) détaillent la méthode SHAP[1], qui assigne à chaque variable une valeur d'importance pour une prédiction particulière et permet d'auditer les prédictions de modèle boîtes noire.

La Figure 5 ci contre illustre la contribution de chaque prédicteur pour la variable réponse. Il est composé de toutes les données utilisées lors de l'apprentissage et permet de classer les variables prédictives par ordre décroissant. La place occupée par chaque point sur l'axe horizontal détermine son effet sur la prédiction (positif ou négatif). Enfin, la couleur (bleue ou rouge) détermine si la valeur de Shap associée à l'observation est faible ou élevée (respectivement).

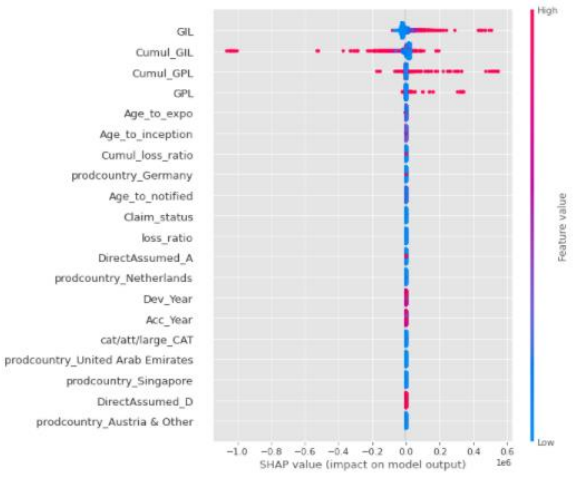


Figure 5: SHAP value pour la prédiction MLP des sinistres payés

L'étude a permis ainsi d'identifier que les cumuls de sinistres payés passés étaient les meilleurs prédicteurs de charge à l'ultime, devant la géographie du risque ou encore l'âge du sinistre.

Perspectives

L'écart séparant le processus consistant à reprendre dans un fichier Excel les données du trimestre précédent afin de mettre à jour les facteurs de développement et un modèle de prédiction entièrement automatisé développé sur Python est immense. Ce dernier inclut la requête, le nettoyage et l'import des données, la prédiction et la production de rapports d'interprétation, autant d'étapes complexes à programmer avant d'envisager une mise en production.

A ce jour, aucune publication actuarielle ne permet de déployer un modèle d'apprentissage statistique à grande échelle au sein d'une compagnie d'assurance non-vie. Les concepts présentés sont intéressants, souvent innovants, mais l'incertitude demeure grande sur ce qui marche ou ne marche pas. Cette étude ne fait pas exception. Il ne faut cependant pas sous-estimer la vitesse à laquelle la technologie progresse. Une publication scientifique ou une librairie révolutionnaire, suivie par l'adoption de quelques acteurs clés de l'assurance non-vie permettront le bond quantique tant attendu dans le domaine de l'estimation des IBNR[2]. Il semble d'ailleurs qu'une masse critique ait d'ores et déjà été atteinte en termes de compétences et de prise de conscience autour de l'apprentissage statistique automatisé.

[1] Shapley Additive exPlanations
[2] Incurred But Not Reported

Executive summary

Context

This study was performed using the Property dataset of AxaXL, the large risk division of Axa. The low volume of available data (as opposed to personal lines) and the volatility of large losses has had a significant impact on the quality of the predictions. Volatility notably led to the inclusion of an exposure vector (premiums) and specific attention was paid to managing outliers (attritional vs major, loss ratio) in order to get a well distributed dataset- as illustrated hereafter.

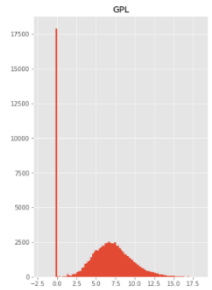


Figure 7: distribution before data engineering

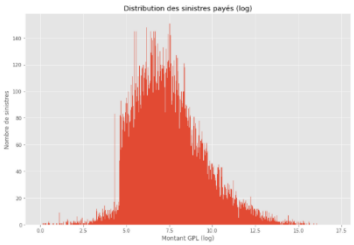


Figure 6: distribution after data engineering

From aggregated methods onto line by line Machine Learning

The aggregated methods are based on the use of triangles that are meant to reflect the dynamic of the claims. Inputs are of two kinds: either claims amount (be it paid, incurred or case by case reserves), or number of claims. Their success can be explained by a straightforward implementation and near real time computation.

However, insurers generate a much higher volume of claims data. Their aggregation into 2 dimensions, as for the use of triangles, prevents them from taking into account the individual characteristics of the claims. The objective of the individual reserving is to take advantage of all predictors available.

Data science can help achieving this goal by combining statistics and algorithmic. It was design to explore, analyse and transform data in order to predict. Machine Learning is a subset of this field and consists in selecting the one that delivers the best prediction among a set of competing models and hyper parameters. To that end, the dataset is allocated into 2 subsets: one, usually richer in terms of data points, is used to train the algorithm while the other acts as a sanity check to ensure that the model continues to perform on an unseen set of data. Contrary to the statistics, Machine Learning does not require to formulate any hypothesis on the distribution of the data. Only one assumption needs to hold true: data to predict must be independent and identically distributed via a random process from a set of explanatory variables. The result of the learning is a function which includes predictors and becomes more complicated as it progresses through the learning phase, allowing it to capture idiosyncrasies from the data themselves- *for example interactions or non linear behaviours*.

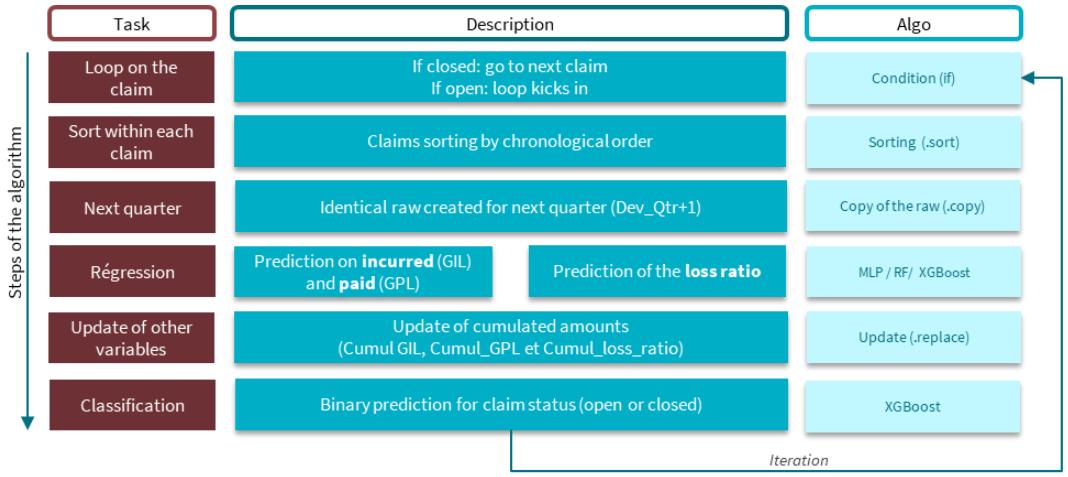
The model

Modelling used in this study was derived from the “cascading” method made public in ASTIN workshops (B. Harej & al, 2017). It aims at predicting iteratively the next diagonal of the lower triangle based on real data included in the upper triangle, until completion.

The **learning phase** covers 80% of the claims embedded in the dataset related to occurrence years ranging from 2010-2020. Models are trained on 3 dependant variables that are the quarterly paid transaction, total incurred & the quarterly loss ratio. Each algorithm is lightly finetuned using hyper parameters.

The **prediction phase** is performed on a sample covering 2010-2014 occurrence years, in order to avoid censorship phenomenon. The model starts with a check on the claim status: if still opened, the loop kicks in. All quarterly transactions within a said claim is ordered chronologically, then the 3 predicting algorithms that are the Neural Network, the Random Forest and the XGBoost are applied to the two dependant variables (claims paid and loss ratio).

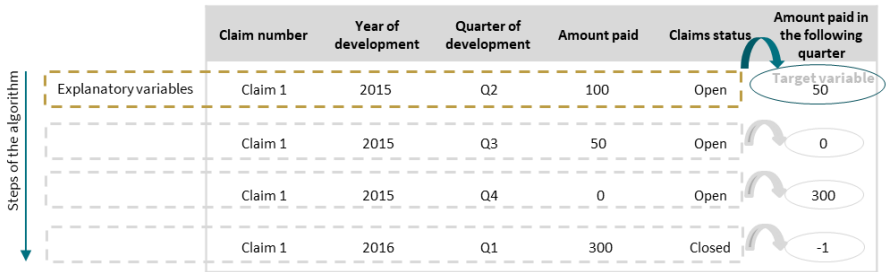
Figure 8: overview of the model presented in this study



Building the model

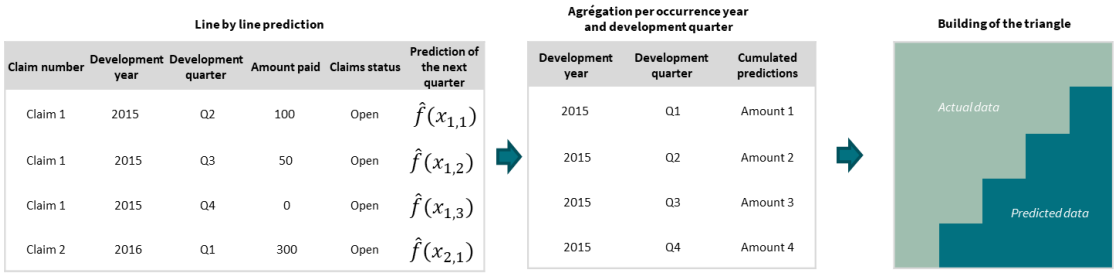
The granularity required by a line by line model gave birth to a whole bunch of constraints prior to running the learning and testing phases. The features engineering had notably to (re) create as much information as possible at the level of the transaction. Outcomes is an algorithm which uses all available explanatory variables in its prediction of the next quarterly transaction paid, as illustrated below.

Tableau 3: Illustrative mechanics of the prediction at transaction level



Once prediction at the level of the transaction is made, all claims are aggregated per occurrence year and allocated accordingly within the lower triangle illustrated in dark green below.

Figure 9: aggregation of the prediction into the lower triangle



Results

Eeach algorithm performance is assessed thanks to a tailor made proximity score computed between actuals and predicted data of each transaction across the 2010-2014 period:

$$Proximity\ score\ (Prox) = \log D_M + |ER_M| - C_M$$

It reflects 3 different measures:

- The arithmetic average of the square root of squared gaps between prediction and actual observations (RMSE),
- The relative error of the model (ER),
- Correlation between actual and predicted data (C_M),

	MAE	Percentage	RMSE	ER	C_M	Prox
Genins	1332157.135	1042.125	1772690.422	0.171	0.811	13.748
Linear Regression	1782986.805	1228.840	2779592.834	0.893	0.249	15.483
XGBoost	1799133.224	891.448	2813136.187	1.364	0.478	15.736
MLP	1919339.938	936.736	3058334.210	1.575	-0.082	16.591
Random Forest	2209858.778	1135.559	3251461.673	2.350	-0.145	17.490

Tableau 4: proximity score for claims paid per algorithm (from highest to lowest performing)

The above table ranks each algorithm per proximity score, from the highest to the lowest. First takeaway: Chain Ladder (referred as “Genins”) continues to be the best in predicting claims paid. It combines the lowest relative error (17%) and the best proximity score. The XGBoost ranks third: a focus per development year shows that it captures time as no other can. Linear Regression performs well across the board thanks to accurate predictions the first 2 years of developments, but loses steam in subsequent years as in the Chain Ladder.

Interpretability

Despite the widespread of computational power, Machine Learning is not widely used in insurance. Actuaries must prevent bias and control variance of their model while ensuring the buy-in of their stakeholders. Unfortunately, the best predictions run on large dataset are often made at the expense of a high complexity, inherent to aggregated or deep learning algorithms, that experts themselves struggle to explain.

(Ribero & al, 2016) explain the SHAP[3] methodology, which gives an importance value for a said prediction to each variable and helps reading through black box models.

Figure 10 illustrate the positive or negative contribution a predictor can have on the dependant variable. It gathers all available explanatory variables and rank them by descending order. Each data point on the horizontal axis shows the impact on the prediction (positive or negative). The colour coding (blue or red) shows if the SHAP value associated with the observation is low or high (respectively).

The study led to the conclusion that cumulated amounts, be it paid or loss ratios, were the best predictors of all, as opposed to location of the risk or even the age of the claim.

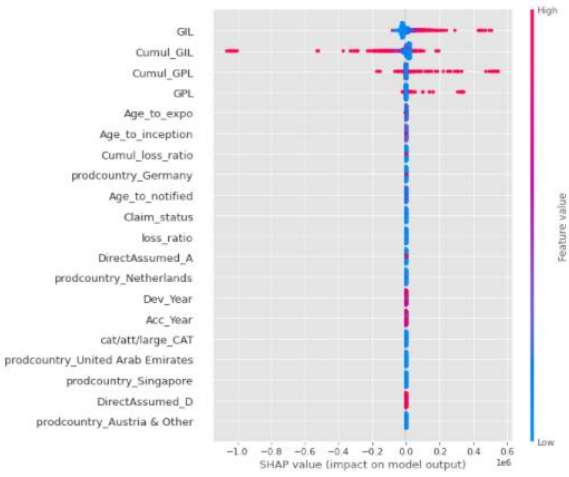


Figure 10: SHAP values given by the MLP for paid claims

Perspectives

The gap between the use of an Excel file to update your Chain Ladder for the next quarter and a fully fledged Machine Learning model developed in Python is immense. The latter requires to plan for the query, cleaning and import of data, plus the run of prediction and interpretation reports: all are complex steps getting in the way of a possible go live in a company.

To date, no actuarial research is able to explain how to deploy a Machine Learning model at scale in a non life insurer. Concepts touched upon are useful, often innovative, but uncertainty around what works and don't work remains high. This study does not make exception. However one should not underestimate the pace at which things can change. One revolutionary actuarial publication or open source library, followed by a few key actors in the non life insurance space, can be enough for a quantum leap in the field of IBNR[4] reserving. A critical mass seems to have been reached anyway, be it in terms of skills or awareness of the role Machine Learning can play in the future.

[3] Shapley Additive exPlanations
[4] Incurred But Not Reported

Remerciements

Je remercie Frédéric Fisher et Paul Henri Rastoul chez Axa, ainsi que toute l'équipe pédagogique de l'Université de Paris-Dauphine et de l'Institut du Risk Management,

Je remercie Alexis pour ses explications durant la formation et Martin pour son aide précieuse sur Python,

Je remercie mes parents et Sophie pour leur soutien indéfectible.

Table des matières

1. Introduction

Les provisions pour sinistres à payer constituent l'essentiel du passif des assureurs non-vie. Leur juste évaluation est un élément clé pour la santé financière et la profitabilité de l'assureur. C'est également une donnée en entrée du processus de tarification, donc un facteur compétitivité.

Historiquement, la méthode de provisionnement utilisée par les assureurs repose sur l'agrégation de données sous forme de triangles de développement, sur lesquels sont appliquées des méthodes actuarielles d'estimation des provisions dont les plus emblématiques sont le Chain Ladder et le Bornhuetter Ferguson. Ces méthodes font néanmoins l'hypothèse d'une stabilité qui n'est pas toujours vérifiée. Des travaux d'actuaires (Gibello, H. et Lebrun, B, 2011) ont par ailleurs démontré la faiblesse du Chain Ladder sur des branches dont les années de liquidation sont insuffisamment développées.

Parallèlement, l'avènement du numérique et la hausse continue de la puissance de calcul ont initié une mutation dans l'exploitation et l'interprétation des données en grande dimension. L'intelligence artificielle et l'apprentissage statistique automatisé ont ainsi permis de révolutionner les algorithmes de prédiction. La littérature actuarielle ne fait pas exception et s'est emparée du sujet à travers de nombreux articles scientifiques. L'application en entreprise tarde néanmoins pour plusieurs raisons : d'abord, la bataille des compétences fait rage et le secteur de la tech tend à capter l'essentiel des talents dans ce domaine. Plus fondamentalement, les tentatives théoriques et pratique en assurance donnent des résultats souvent intéressants, mais au détriment de la simplicité et l'interprétabilité du modèle.

Cette étude a pour objectif d'explorer les techniques les plus avancées du provisionnement en non-vie et d'évaluer les performances des modèles d'apprentissage statistique les plus emblématiques sur le cas concret du dommage en grand risque[5]. Elle comporte trois parties : une présentation des méthodes agrégées les plus utilisées, les principales méthodes d'apprentissage automatique et une introduction à l'interprétabilité des modèles boîtes noires.

Afin de nous permettre de conclure, chaque étape du second chapitre est accompagnée d'une application concrète, fruit d'une modélisation sur mesure pour la projection de la charge au détail de chaque transaction du sinistre.

La question à laquelle ce mémoire tente de répondre est la suivante :

- ⇨ **Est-ce que l'apprentissage statistique ligne à ligne peut améliorer le provisionnement d'une branche courte mais volatile comme le dommage en grands risques ?**

Nous verrons que la réponse n'est pas binaire, mais des enseignements intéressants émergent néanmoins.

« LE PROVISIONNEMENT N'EST PAS UN PROBLEME DE MODELISATION COMPLEXE ET SOPHISTIQUE, MAIS UN EXERCICE DE CHOIX DE MODELE ». Hans Bühlmann.

[5] Grand risque= société dont le chiffre d'affaire est supérieur à deux milliards de dollars et dont les effectifs sont supérieurs à 5000,

1. Objectifs et périmètre de l'étude

1.1 Généralités

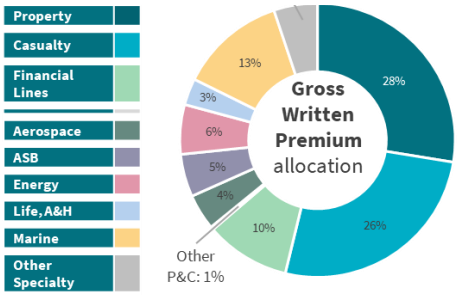
Cette étude a été effectuée sur des données d'AXA XL, la division grands risques d'AXA spécialisée dans les risques d'entreprise de moyennes et grandes tailles[6].

La majorité de son offre est composée du dommage et de la Responsabilité Civile- *28% et 26% du portefeuille respectivement*- le reste étant des branches de spécialité.

Deux spécificités liées aux grands risques commerciaux vont structurer cette étude :

- le faible volume de données disponibles comparativement à l'assurance personnelle
- la volatilité élevée des sinistres graves (risque de pic et exposition Cat[7])

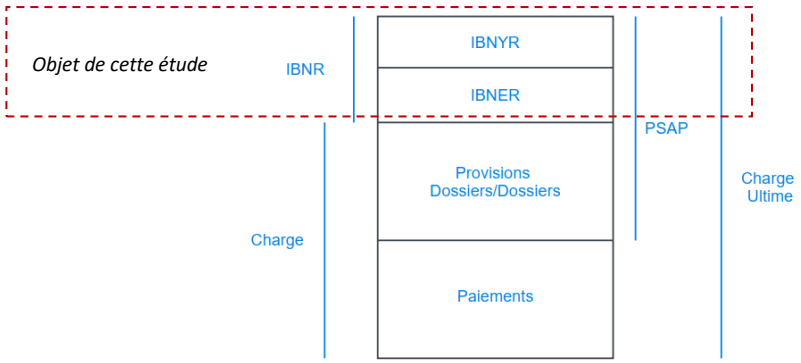
Figure 11: mix portefeuille AxaXL



Avant de passer à l'étude concrète des données, il est utile de définir la provision puis de décrire le processus de provisionnement.

La provision pour sinistre à payer, également appelée PSAP, est la principale provision portée au passif de l'assureur non-vie. Elle est définie par le code des assurance (Code des assurances, 2013) comme « LA VALEUR ESTIMATIVE DES DEPENSES EN PRINCIPAL ET EN FRAIS, TANT INTERNES QU'EXTERNES, NECESSAIRES AU REGLEMENT DE TOUS LES SINISTRES SURVENUS ET NON PAYES (…) ».

Figure 12: décomposition de la charge ultime en assurance non-vie



[6] Seuil fixé à partir du chiffre d'affaires des clients : 150m€ pour les moyennes, 2M€ pour les grandes tailles
[7] Cat = catastrophe naturelle

La Figure 12 décrit les trois grandes composantes de la charge ultime.
La première composante est le paiement effectué à l'assuré. La seconde est l'évaluation par le gestionnaire sinistre des sommes restantes à payer - *le sinistre est alors connu*- communément appelé « dossier dossier ». Le niveau de précision varie du coût forfaire (usuellement basée sur le coût moyen, organisé par tranche de sévérité) à l'estimation établie par un rapport d'expert. La troisième composante est une estimation des sinistres « tardifs », encore inconnus du gestionnaire car survenus mais pas encore déclarés à l'assureur, que l'actuaire doit anticiper. Il s'agit des IBNR, acronyme pour « *incurred but not reported* ». Ces derniers sont la somme de deux natures d'IBNR :

> « *Incurred but not enough reported* » (IBNER) : il s'agit de provisions permettant d'augmenter la provision d'un sinistre préalablement déclaré à l'assureur mais considéré comme insuffisamment provisionné au moment de la clôture des comptes (les causes peuvent être multiples : erreur du gestionnaire, éléments tardifs portés au dossier…)
>
> « *Incurred but not yet reported* » (IBNyR) : il s'agit de provisions permettant de couvrir les coûts des sinistres survenus mais pas encore portés à la connaissance de l'assureur (c'est-à-dire non déclarés)

C'est cette troisième et dernière composante de la charge ultime qui fera l'objet de cette étude.

D'autres provisions peuvent se trouver au bilan des assureurs non vie. Bien que ne faisant pas l'objet de ce mémoire, elles sont listées ci-dessous pour information (Ortiz, 2019):

«

- La **provision pour frais de gestion de sinistres**, c'est-à-dire les coûts internes que génèrera le sinistre jusqu'à sa clôture. Il s'agit généralement de frais généraux incluant les frais de structure de l'assureur,
- Une estimation des **recours** à encaisser,
- **provision mathématique des rentes** (PM) qui correspond à la valeur actuelle des engagements de l'entreprise en ce qui concerne les rentes mises à sa charge. La **provision pour risques croissants** (PRC) représente quant à elle une provision pour les opérations d'assurance contre les risques de maladie et d'invalidité et est égale à la différence des valeurs actuelles des engagements pris par l'assureur et par les assurés.
- La **provision pour égalisation** (PE) est plus transverse car destinée à faire face aux charges exceptionnelles
  (catastrophes naturelles, risque atomique, RC pollution, risques spatiaux, assurance-crédit...).
- Enfin, la **provision pour risque d'exigibilité** (PRE) a pour but de faire face aux engagements dans le cas de la moins-value de certains actifs.

»

1.2 Les données

La base utilisée dans cette étude ne comporte aucune donnée à caractère personnel[8], son utilisation n'est donc pas soumise au Règlement Général Européen sur la Protection des Données (RGPD).

La modélisation a été effectué sur Python, un logiciel open source généraliste orienté objet. Tout comme R, il présente de nombreux avantages :

- Capacité à gérer de nombreuses tâches différentes, sorte de couteau Suisse de l'apprentissage statistique,
- Nécessité de construire un modèle récursif d'apprentissage pour cette étude (voir explications dans « 3.2 La modélisation »),
- Utilisation limitée de la ponctuation (lisibilité du code),
- Interopérabilité avec d'autres langages de programmation dans l'hypothèse d'une mise en production,

La base de données inclue 10 ans d'historique AxaXL, brutes de réassurance et dont l'exposition est monde entier. Elle contient 18 variables. La fréquence de développement des sinistres est trimestrielle (variable « Dev_Qtr ») et seules 3 variables initiales sont dynamiques[9]. Les variables explicatives sont listées ci-dessous, accompagnées de leur signification métier.

Tableau 5: liste des variables de la base de données

	Variable	Signification
	PFKPolNbr	Numéro de police de l'assuré
	PFKClmNbr	Numéro de sinistre
	FKTradeOfBusiness	Secteur d'activité du client
	FKCovrgTypeCode	Couverture d'assurance
	InceptionDate	Date d'effet de la police
	ExpiryDate	Date d'expiration de la police
	LossAttachmentDate	Date de rattachement du sinistre
	DateNotified	Date de notification du sinistre
	DirectAssumed	Nature du business (assurance directe ou acceptée en Facultative)
	USStates	Code postal US du risque
	prodcountry	Pays où le contrat a été souscrit
	Exposure_Qtr	Trimestre d'exposition au risque
	ChartField1/AccidentYear	Date de survenance du sinistre
	cat/att/large	Segmentation du sinistre (par défaut: attritionnel)
	ri_type	Type de réassurance (aucune dans les données de l'étude)
Données dynamiques	Dev_Qtr	Trimestre d'enregistrement de la transaction
	GIL	Charge sinistre
	GPL	Sinistre payé
Variables créés pour les besoins de l'étude (features engineering)	Cumul_GIL	Cumul des sinistres payés
	Cumul_GPL	Cumul des provisions
	Claim_status	Statut du sinistre
	Age_to_notified	Nombre de trimestres écoulés depuis la notification du sinistre
	Age_to_inception	Nombre de trimestres écoulés depuis la date d'effet de la police
	Age_to_expo	Nombre de trimestres écoulés depuis la date d'exposition au sinistre
	Acc_Year	Année de survenance du sinistre
	Dev_Year	Année d'enregistrement de la transaction
	Notif_Year	Année de notification du sinistre
	Loss_Year	Année de rattachement du sinistre
	Expo_Year	Année d'exposition au risque
	GPL_trim_suivant	Transaction de provision enregistrée le trimestre suivant
	GIL_trim_suivant	Transaction de sinistre payé enregistrée le trimestre suivant
	loss_ratio	GIL divisé par primes à l'ultime de l'année de survenance

[8] Qui se définit comme « toute information se rapportant à une personne physique identifiable directement ou indirectement » (JO de l'Union Européenne, 2016)

[9] Dynamique = variable dont les catégories évoluent au cours du temps

Parmi les variables crées pour les besoins de l'étude (partie basse du Tableau 5), quatre doivent faire l'objet d'un focus particulier pour comprendre ce qui a été fait à l'étape de modélisation:

1. le cumul des transactions (« Cumul_GIL » et « Cumul_GPL »),
2. l'âge du sinistre (« Age_to_notified »),
3. le montant du trimestre suivant (« GIL_trim_suivant » et « GPL_trim_suivant »)
4. le ratio sinistres sur primes (« loss_ratio »).

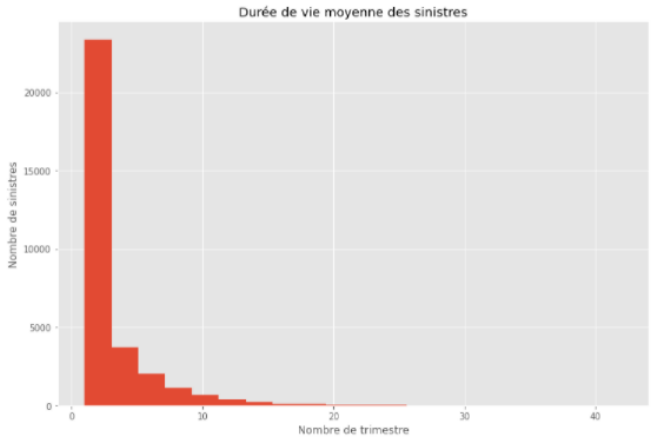
L'objectif de ces variables supplémentaires est d'ajouter de l'information à la maille transaction afin de proposer en entrée du modèle un maximum de variables possiblement explicatives. Le cumul des transactions pour chaque sinistre va ainsi permettre à l'algorithme de connaître le niveau de paiements/charge totale [10] à chaque étape de la vie du sinistre (illustration Tableau 6).

Tableau 6: illustration simplifiée de "Cumul_GIL'

Sinistre	Trimestre	GIL	Cumul
1	Q1	5	5
1	Q2	10	15
1	Q3	0	15
1	Q4	100	115

La création de l'âge du sinistre part de l'intuition que plus un sinistre est âgé, plus sa sévérité peut être élevée. Une variable est donc créée afin de compter le nombre de trimestres écoulés depuis la déclaration du sinistre (« Age_to_notified »). Sa distribution est illustrée ci-dessous. Une variable identique est créée avec la date d'effet de la police (« Age_to_inception ») et la date d'exposition au risque (« Age_to_expo »).

Figure 13: Histogramme de la variable « Age_to_notified"



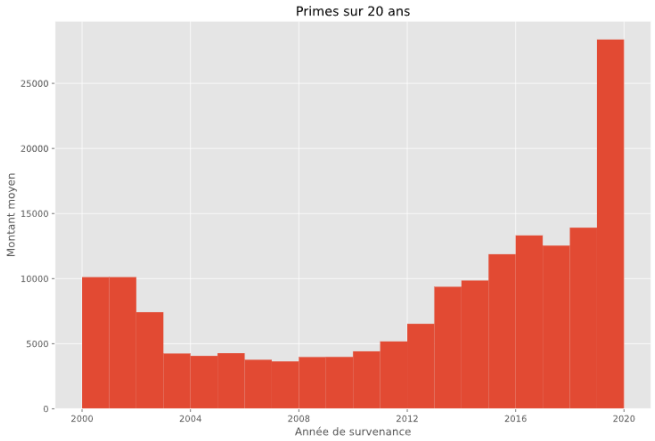
[10] GPL et GIL respectivement dans la base de données

Le montant de sinistre payé/provision au trimestre suivant est expliqué en détail à la section 3.2 La modélisation. A ce stade, et toujours dans un souci d'ajouter un maximum d'information à la maille transaction, retenons que la variable permet de ramener sur la transaction du trimestre courant le montant payé/provisionné du trimestre suivant afin de préparer l'étape d'apprentissage ligne à ligne.

1.2.1 Le vecteur de primes

Les primes collectées par l'assureur fournissent des informations sur le portefeuille qu'il peut être intéressant d'inclure dans un processus de provisionnement. Il existe de nombreux types de primes : émises, acquises, encaissées, rattachées à des exercices de souscription, de survenance… Les primes acquises sont utilisées dans cette étude et correspondent à la fraction de la prime couvrant la période durant laquelle la police est en vigueur. C'est cette portion qui sert à l'indemnisation des sinistres survenus sur la même période.

Figure 14: prime moyenne en dommage chez AxaXL par année de survenance



La période 2010-2020, utilisée pour les besoins de cette étude, montre une forte croissance du montant moyen de la prime. Deux facteurs explicatifs : le premier est l'augmentation annuel du taux de prime (« hardenning market ») due à la constante augmentation de la sinistralité CAT et large[11]. Le second facteur est la fusion de deux portefeuilles sensiblement différents lors du rachat de l'assureur XL par Axa[12].

Le ratio sinistres sur primes est un moyen d'ajouter la donnée d'exposition que sont les primes à l'ultime pour l'année de survenance considérée. De nombreux papiers actuariels conseillent l'utilisation de données d'exposition afin de ne pas introduire de biais lié à la croissance du portefeuille. Le loss ratio à la maille transaction sera donc calculé ainsi :

$$loss\ ratio\ (\text{à}\ la\ maille\ transaction) = \frac{montant\ payé\ de\ la\ transaction}{cumul\ des\ primes\ de\ l'année\ de\ survenance\ concernée}$$

Le loss ratio (loss_ratio) sera la troisième variable réponse que cette étude tentera de prédire, au côté des sinistres payés (GPL) et de la charge sinistre (GIL).

[11] Un sinistre est considéré comme large chez AxaXL lorsque la charge atteint $5m
[12] Opération annoncée le 12 Septembre 2018

1.2.2 Nettoyage des données

La base est a été extraite grâce à une requête SQL dont le détail se trouve en annexe. L'extraction comporte 193 211 lignes.

Un premier coup d'œil sur les données permet d'identifier une proportion de 1% de valeurs manquantes sur les polices. Par ailleurs, la colonne « USStates » contient des valeurs sur 3% des lignes : les 97% restants correspondent aux cas dans lesquels le risque ne se situe pas sur le territoire Américain. Choix est fait à ce stade de conserver la donnée car les Etats Unis connaissent une sévérité et une fréquence plus élevées que dans le reste du monde compte tenu de leur exposition CAT ainsi que leur système judiciaire procédurier. L'information peut être utile à la prédiction. Un traitement spécifique lui est appliqué afin de remplacer les valeurs vides par « Non_US ».

Figure 16: Etat des données brutes

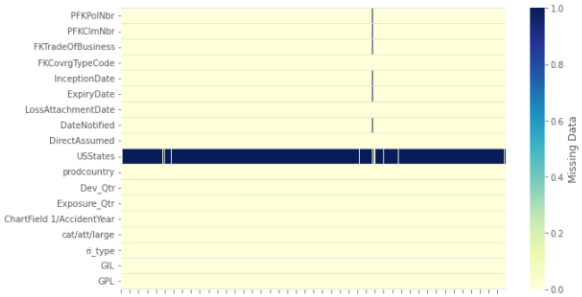


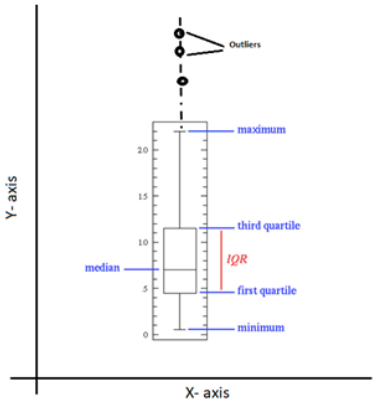
Figure 15: Données après retraitements

1.2.3 Analyse descriptive

Le premier outil utilisé pour visualiser les données est un box plot. Cette méthode permet de montrer graphiquement des groupes de données numériques à travers leurs quartiles (25-50-75%). Les lignes verticales, appelées « moustaches », sont une indication de la variabilité des données au-delà de la borne inférieure et supérieure des quartiles. Cela permet d'identifier visuellement les éventuelles valeurs extrêmes.

L'écart inter quartile (« IQR[13] » de la Figure 17) mesure l'écart au centre de la base de données : il s'obtient par la différence entre le premier et le troisième quartile.

Figure 17: Illustration du box plot



[13] IQR est l'acronyme anglais pour « Inter Quartile Range »

Le box plot sur les données brutes des transactions de sinistres payés (« GPL » dans la base) et charge sinistre («GIL » dans la base) montre un nombre élevé de valeurs extrêmes, suffisamment pour écraser visuellement l'écart inter quartile.

Soit Q_1 le quartile inférieur (25ème percentile) et Q_3 le quartile supérieur (75ème percentile), alors toute donnée inférieur à $Q_1 - 1.5(IQR)$ ou supérieur à $Q_3 + 1.5(IQR)$ est considéré comme « valeur extrême ». La Figure 18 montre les mêmes données une fois les outliers retirés. Un grand nombre de transactions négatives apparaît également sur les sinistres payés (« GPL »).

Figure 18: box plot des sinistres payés et charge totale

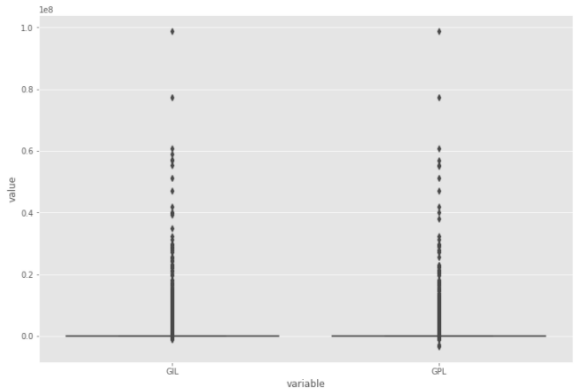
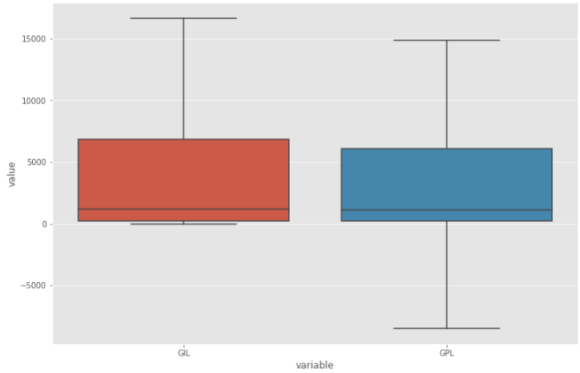


Figure 19: box plot sans les outliers



L'histogramme sur ces mêmes données met à jour une forte concentration des données proches de 0. Un passage au logarithme népérien à la maille sinistre permet d'y voir plus clair et confirme la nécessité d'un retraitement sur les données avant la modélisation (voir la section suivante « La gestion des valeurs extrêmes »). Il est à noter qu'un passage log à la maille transaction n'est pas possible car certaines sont négatives, contrairement à la maille sinistre.

Figure 20: Histogramme des sinistres payés et provisions

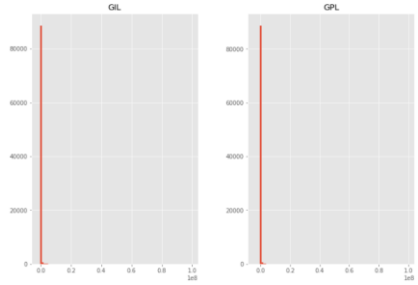
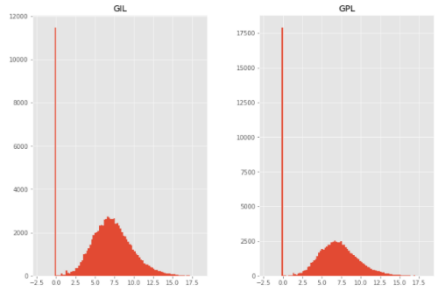


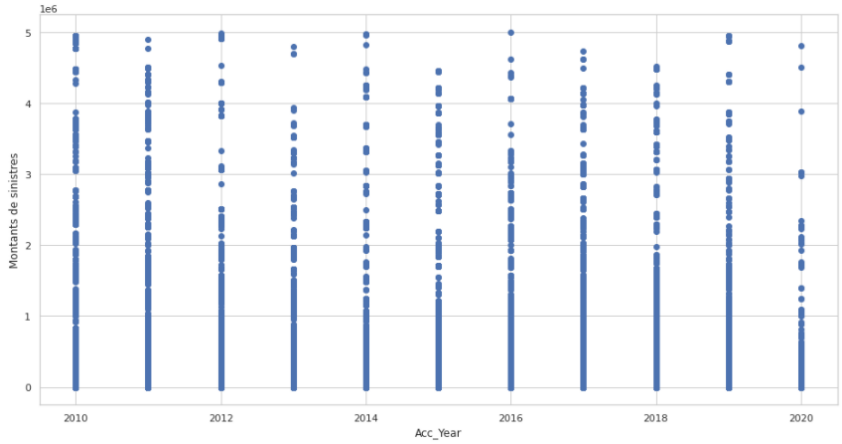
Figure 21: application de la transformation log



1.2.4 La gestion des valeurs extrêmes

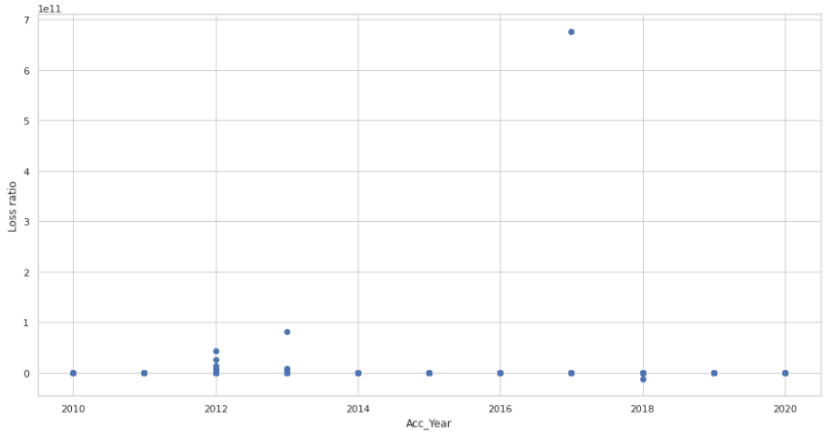
La segmentation des sinistres permet d'appréhender l'impact des sinistres extrêmes et de limiter l'instabilité des estimations. En provisionnement, elle permet de séparer les sinistres en deux catégories homogènes (attritionnels et graves) afin de les modéliser séparément. Il est possible de déterminer ce seuil de plusieurs façons : un simple montant forfaitaire basé sur l'historique de sinistralité, une charge sur-crête représentant un pourcentage arbitraire de la charge, un quantile de la distribution historique (usuellement compris entre 0.5% et 2%) ou encore grâce à la théorie des valeurs extrêmes. Cette dernière offre un cadre mathématique permettant la modélisation de la fréquence et de la sévérité de phénomènes rares ce qui, en termes probabilistes, équivaut à évaluer l'épaisseur de la queue de distribution. AxaXL utilise un seuil de $5m, tout sinistre passant au-delà conserve le statut « grave » tout sa vie. La segmentation n'étant pas le cœur de cette étude[14], tous les sinistres dont la variable « Cumul_GIL » dépassent le seuil de $ 5m à un instant T sont exclus de la base de travail, comme illustré en Figure 22.

Figure 22: Montants de sinistres par année de survenance



La Figure 23 illustre les loss ratios par année de survenance à l'aide de nuages de points. Elle permet de constater que de nombreux outliers subsistent. Même si la somme des loss ratio transactionnels par année de survenance (tous sinistres confondus) n'a pas de sens assurantiel, l'échelle à 10^{11} fait office d'alerte.

Figure 23: Nuage de points du loss ratio avant retraitements



[14] Pour des informations plus détaillées sur la définition qualitative et quantitative d'un seuil de gravité, voir le mémoire d'actuaire (Ortiz, 2019)

L'analyse graphique par itérations (compromis entre perte de données et abaissement du seuil) permet d'établir un plafond à 10, comme le montre le graphique ci-dessous. La perte de données se limite à 3 415 lignes et préserve l'homogénéité de la distribution des transactions par sinistre comme l'illustrent les Figure 24 et Figure 25.

Figure 24: nuage de points du loss ratio avant application du plafond à 10

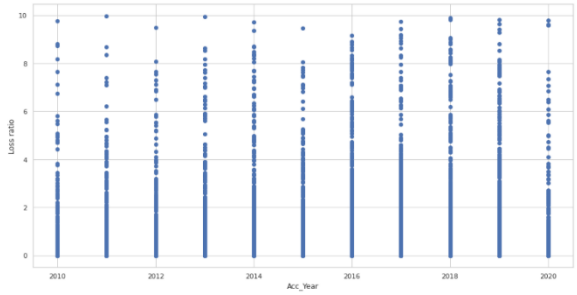
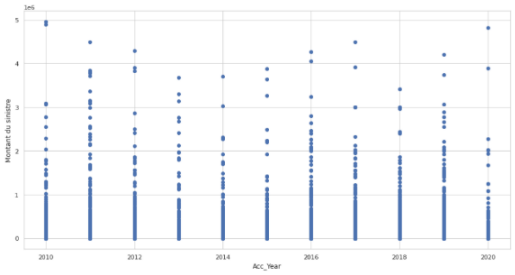


Figure 25: nuage de points des transactions de payés après application du plafond de $5m



Il est alors possible de comparer les box plots du loss ratio par année de survenance avant et après retraitements grâce aux Figure 26 et Figure 27.

Figure 26: box plot du loss ratio avant retraitements

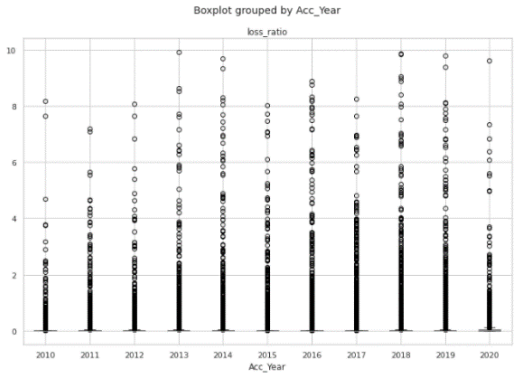
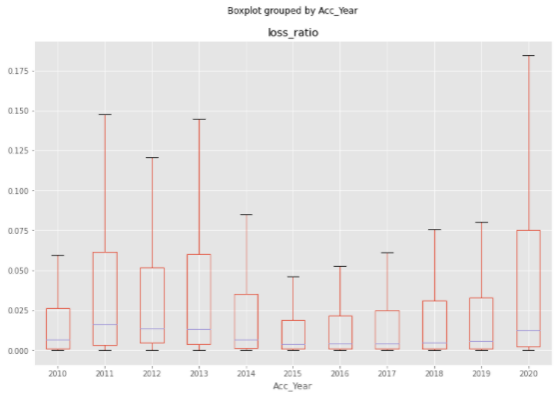
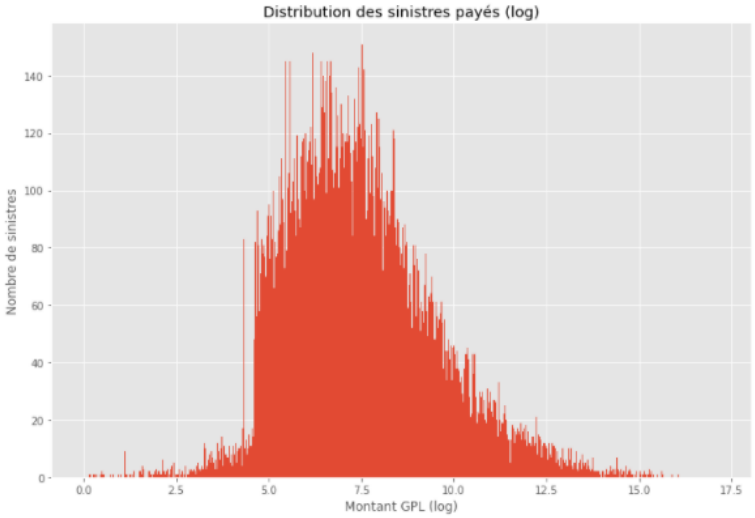


Figure 27: box plot du loss ratio après retraitements



Tous ces retraitements permettent d'obtenir un jeu de données homogène et prêt à alimenter les algorithmes d'apprentissage statistique du second chapitre, comme le montre la Figure 28 ci-dessous.

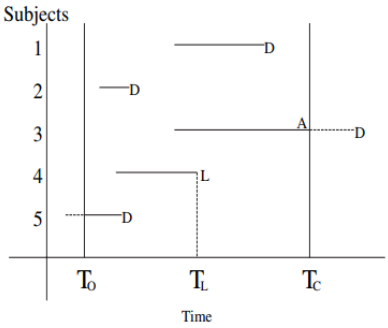
Figure 28: histogramme des transactions de sinistres payés (passées au log)



1.3 Troncature et censure

Les variables temporelles se caractérisent par un point de départ et une durée de vie à la fin de laquelle un évènement déterminé survient. Selon le domaine étudié, il peut s'agir de la mort d'un patient, de l'arrêt d'une machine ou, dans le cas assurantiel non vie, de la clotûre du sinistre. Les données liées à une temporalité possèdent deux composantes : elles sont strictement positives et généralement asymétriques. Lorsque des sujets peuvent « survivre » au-delà de la période étudiée, les données sont alors dites « censurées ». La Figure 29 ci-dessous illustre ce phénomène[15].

Figure 29: Type de censure (Barnett et Dobson, 2008)



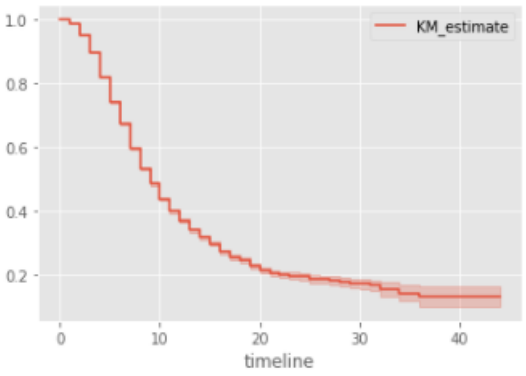
[15] Dans le schéma, les lettres D = mort, L= vie du sujet étudié et T représente les étapes de l'étude

La base de données est composée de sinistres en cours et clos. L'application d'un modèle de provisionnement sur les seuls dossiers clos introduirait un biais de sélection car les sinistres clôturés sont majoritairement ceux dont les développements sont les plus courts et les montants de charge les plus faibles. (Duval, F et Pigeon, M, 2019) démontrent ainsi grâce au « modèle B »[16] que l'apprentissage sur des sinistres trop simples conduit à une sous estimation des nouveaux sinistres, sans compter la perte importante d'information induite par l'exclusion d'une partie des données. Afin de corriger ce biais, (Lopez, O. et al, 2016) proposent d'implémenter une stratégie qu'ils baptisent « IPCW »[17] et qui consiste à allouer des poids aux observations afin de compenser la censure sur une partie des données. C'est l'estimateur Kaplan-Meier de la distribution censurée qui permet de déterminer l'attribution des poids.

La base de données utilisée dans la présente étude ne comporte pas de troncature car tous les sinistres commencent à l'année de survenance 2010. Cinq années suffisent pour une liquidation totale des sinistres dommage : la base reprenant un historique de 10 ans, l'ajout de poids n'est pas nécessaire pour les 5 premières années (2010-2015).

Pour s'en convaincre, la fonction de survie des sinistres de la base est tracée en Figure 30 selon l'âge du sinistre, ce dernier correspondant au nombre de mois séparant la date d'effet de la police de la clôture du sinistre. La probabilité de survie du sinistre au 44ème mois (ou 3 ans et demi) est de 13%. La clôture est presque certaine au 60ème mois (ou 5 ans), ce qui justifie l'utilisation de la période 2010-2015 pour la mesure de performance des modèles d'apprentissage statistique décrite à la section « 3.2.3 L'évaluation des modèles ».

Figure 30: fonction de survie du statut du sinistre (ouvert/fermé) en nombre de mois (avec intervalle de confiance)



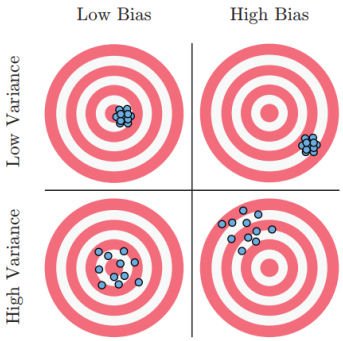
[16] Modèle dans lequel la censure n'est pas corrigée
[17] Pour **I**nverse **P**robability of **C**ensoring **W**eighting

1.4 Le compromis biais-variance

Le dilemme biais-variance est une problématique classique de modélisation. Il décrit le difficile équilibre entre complexité et généralisation du modèle.

Soit un ensemble de points $(x_1, \ldots, x_n)$ utilisé pour l'apprentissage et $(y_1, \ldots, y_i)$ les valeurs réelles associées à chaque point. La modélisation suppose la relation fonctionnelle bruitée suivante : $y_i = f(x) + \epsilon_i$ dans laquelle ϵ_i est l'erreur irréductible de moyenne nulle et de variance constante σ^2. La modélisation va consister à trouver $\hat{f}(x)$ tel que $\hat{f}(x) \simeq y_i$.

Figure 31: Dilemme biais/variance, extrait de (Anwar, 2021))



Le biais est l'erreur entre la moyenne des prédictions du modèle et les valeurs réelles. Il peut être vu comme l'erreur d'estimation due aux hypothèses formulées lors de la modélisation.

$$Biais\left[\hat{f}(x)\right] = \mathrm{E}\left[\hat{f}(x) - f(x)\right]$$

La variance est la variabilité des prédictions entre différents ensembles de données en entrée. Elle peut être vue comme la variation de la décision du modèle selon les données d'entraînement, quantifiant de combien le modèle se déplace autour de sa moyenne.

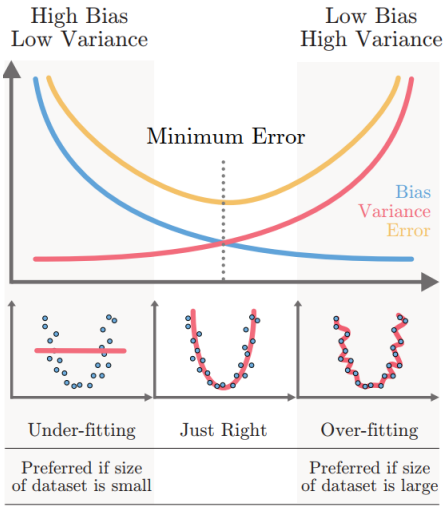
$$Var\left[\hat{f}(x)\right] = \mathrm{E}\left[\left(\hat{f}(x) - \mathrm{E}\left|\hat{f}(x)\right|\right)^2\right]$$

La décomposition biais-variance est une façon d'analyser l'espérance de l'erreur de prédiction comme une somme de trois termes : le biais, la variance et l'erreur irréductible. Ainsi que l'erreur quadratique attendue de la fonction $\hat{f}$ sur un échantillon de test $(x_1, \ldots, x_n)$ se décompose comme suit :

$$\mathrm{E}\left[(y - \hat{f}(x))^2\right] = Biais\left[\hat{f}(x)\right]^2 + Var\left[\hat{f}(x)\right] + Var[\epsilon]$$

Plus le modèle sera complexe, plus le biais sera faible mais plus sa généralisation à de nouvelles données sera délicate, donc de variance élevée. La Figure 32 ci-dessous illustre cette quadrature à laquelle l'actuaire est confronté lors de la modélisation. Ces concepts seront régulièrement mentionnés dans cette étude.

Figure 32: minimisation de l'erreur, extrait de (Anwar, 2021))



2. Présentation des méthodes agrégées déterministes

Ce chapitre s'attache à décrire le corpus théorique des principales méthodes agrégées non stochastiques. Ces méthodes sont dites déterministes car conçues en dehors du cadre probabiliste.

2.1 Triangle de développement

Les méthodes agrégées reposent sur l'utilisation de triangles de développement. Ces triangles reflètent la dynamique de sinistres et les données sont essentiellement de deux natures: soit des montants de sinistres (paiement, charges ou réserves), soit des nombres de sinistres.

Notations (Charpentier et Denuit, 2005) :

- i : indice des années de survenance des sinistres où i $\in [1\,;n]$
- j : indice des années de développement où j $\in [1;n]$
- $Y_{i,j}$: correspond au montant des sinistres survenus l'année i et réglés l'année i + j (dis autrement : après j année de développement)
- $C_{i,j}$: correspond aux paiements agrégés des sinistres survenus l'année i, en j années de développement, en d'autres termes : $C_{i,j} = Y_{i,1} + Y_{i,2} + \cdots + Y_{i,j}$

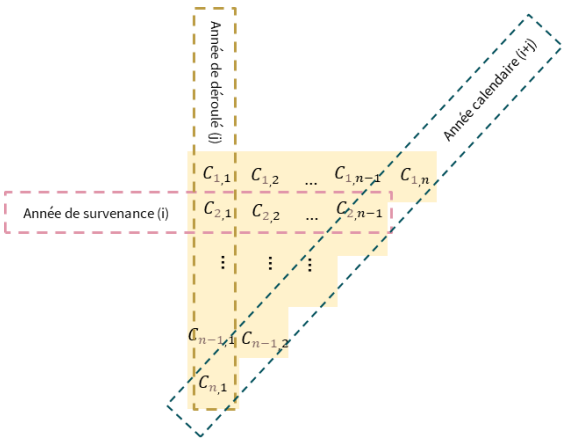
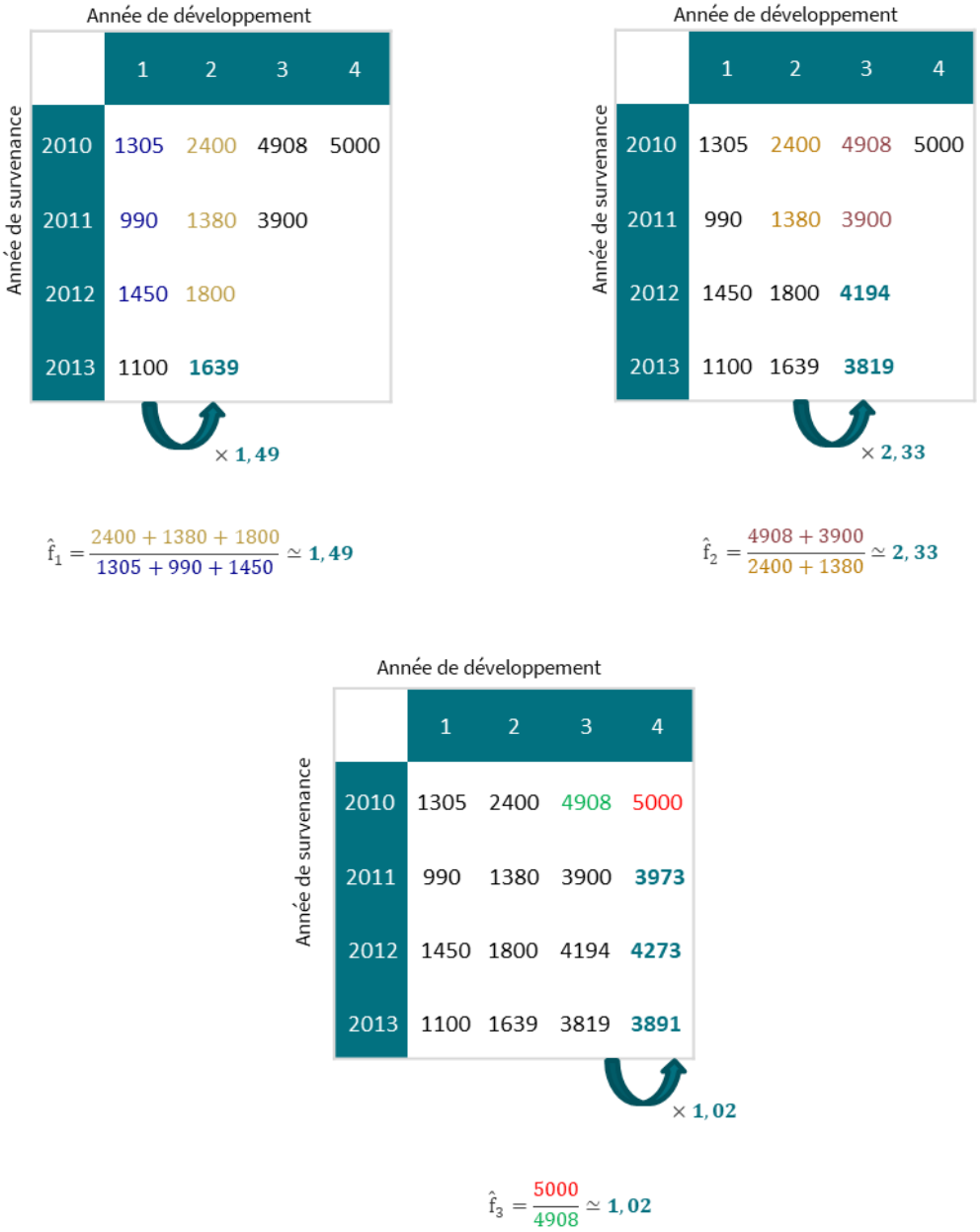


Figure 33: Visualisation du triangle de règlements,
adapté de (Charpentier et Denuit, 2005)

2.2 Le Chain Ladder

La méthode Chain Ladder est une méthode déterministe d'estimation des réserves basée sur un triangle de règlements ou de charges[18]. Elle repose sur l'idée que les sinistres survenus lors d'années de survenance différentes se développent de façon similaire mais indépendante. La charge ultime est calculée en appliquant à la diagonale du triangle des facteurs de développement (notés λ_j) qui ne dépendent que de l'année de déroulement.

Figure 34: illustration du mécanisme de Chain Ladder



[18] Egalement appelé link ratios

La très large adoption de cette méthode tient à sa simplicité de mise en œuvre et son interprétabilité. (Charpentier et Denuit, 2005) rappellent par ailleurs qu'elle nécessite de formuler une hypothèse de stabilité du délai s'écoulant entre la survenance d'un sinistre et les règlements, ce qui n'est pas aisé à maintenir dans un temps long. Les complications pour l'actuaire sont multiples : inflation, changement de structure du portefeuille, changement dans l'offre contrat, changement de process dans la gestion des sinistres (etc.). La probabilité de tels changements augmente avec le temps : il semble donc raisonnable de penser que cette méthode se comporte mieux sur les branches courtes comme le dommage, objet de cette étude, que sur une branche longue comme la responsabilité civile.

2.2.1 Cadre formel

Si les hypothèses (H1) et (H2) sont satisfaites, la méthode de Chain Ladder suppose que les $C_{i,j}$ sont liés par un modèle de la forme :

$$\mathrm{C_{i,k+1}} = \lambda_k \mathrm{C_{i,k}}\,,\ \forall \mathrm{i}, \mathrm{k} = 1, \dots, \mathrm{n}$$

- (H_1) *les années de survenance sont indépendants entre elles,*
- (H_2) *les années de développement sont les variables explicatives du comportement des sinistres futurs*

Les facteurs de développement sont estimés à l'aide des observations :

$$\hat{\lambda}_k = \frac{\sum_{i=1}^{n-k} C_{i,k+1}}{\sum_{i=1}^{n-k} C_{i,k}} \text{ pour } \mathrm{k} = 1, \dots, \mathrm{n}$$

Il devient alors possible de compléter la partie inférieure du triangle des montants cumulés pour chaque exercice de survenance et chaque année de développement :

$$\hat{\mathrm{C}}_{\mathrm{i},j} = \left(\hat{\lambda}_{\mathrm{n+1-i}} \ \dots\ \hat{\lambda}_{\mathrm{j-1}}\right) \mathrm{C_{i,n+1-i}} = \mathrm{C_{i,n+1-i}} \prod_{k=n+1-i}^{j-1} \hat{\lambda}_k \qquad (1)$$

Nous obtenons alors une estimation de tout ce dont l'actuaire provisionnement a besoin :

- La charge ultime de sinistre par exercice de survenance (1)
- Les provisions par exercice de survenance : $\widehat{\mathrm{R}}_{\mathrm{i}} = \hat{\mathrm{C}}_{i,\mathrm{j}} - \mathrm{C_{i,j-i}}$
- Les provisions totales : $\widehat{\mathrm{R}} = \sum_{i=1}^{n} \widehat{\mathrm{R}}_{\mathrm{i}}$

2.2.2 Forces et faiblesses

Cette méthode présente l'avantage d'être simple car elle ne fait aucune hypothèse quand à la loi suivie par le coût des sinistres, ou leur fréquence.

L'inconvénient principal du Chain Ladder réside dans le fait que le schéma de développement est identique pour toutes les années de survenance, ce qui pose problème lors d'éventuels changements de jurisprudence ou de gestion. (Charpentier et Denuit, 2005) ajoutent que l'incertitude est très importante pour les années récentes : le coefficient multiplicateur de la dernière année est le produit de $(n-1)$ estimations de λ_k. Cette incertitude est d'autant plus grande pour les risques longs où les premiers paiements commencent au bout de quelques années: une forte variation en pourcentage sur un petit montant lors des premières années peut conduire à de fortes variations à l'ultime.

D'autres inconvénients, souvent liés au point précédent, ont été soulignés dans divers mémoires d'actuaires:

- L'estimation des facteurs de développement les plus récents repose sur une quantité limitée de données[19],
- S'agissant d'une méthode déterministe, il n'est pas possible d'obtenir une mesure de précision sur les estimations[20],
- Impossibilité de prendre en compte une inflation non constante, de repérer un changement de jurisprudence,
- Ne détecte pas les irrégularités potentielles du triangle,
- Risque de sur-paramétrisation dans le cas où un facteur de queue doit être ajouté,
- Impossibilité de séparer les IBNER des IBNyR,
- Manque de robustesse pour les valeurs extrêmes,
- Propagation des erreurs à travers les facteurs de développement pouvant conduire à une importante erreur d'estimation sur les dernières périodes de développement,
- Impossibilité d'obtenir une estimation de la loi de probabilité de la provision totale (pas de calcul de volatilité, value at risk…)

2.2.3 Application

Avant toute chose, il est important de vérifier les hypothèses d'indépendance telle qu'exposée à la section précédente.

Figure 35: vérification des hypothèses d'indépendance

```
clrd.valuation_correlation().z_critical
```

	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
2010	False	False	False	False	False	False	False	False	False	False

Le triangle des sinistres payés sur 10 ans est présenté en Figure 36 ci-dessous. La projection du triangle inférieur sera effectuée au second chapitre par souci de concision.

Figure 36: triangle supérieur des sinistres payés cumulés

	12	24	36	48	60	72	84	96	108	120	132
2010	49,061,414	22,036,862	8,778,983	3,087,128	1,041,890	1,943,543	403,425	129,692	32,918	56,662	76,329
2011	29,771,722	76,174,978	19,045,325	4,481,712	1,685,402	631,468	1,896,104	229,912	44,364	15,741	
2012	36,347,606	20,631,134	13,807,387	1,629,506	749,488	347,446	473,640	44,320	15,678		
2013	26,765,949	44,715,388	5,647,316	7,708,857	948,041	2,501,169	70,045	11,491			
2014	9,673,901	34,242,379	11,111,263	2,740,081	1,181,056	225,061	105,737				
2015	16,315,070	35,309,994	17,342,016	2,396,509	766,979	423,102					
2016	37,838,863	76,584,017	16,721,007	12,656,656	4,344,993						
2017	30,390,173	90,362,803	23,475,550	5,561,057							
2018	26,259,196	51,838,275	37,461,298								
2019	42,996,756	59,696,522									
2020	17,909,692										

[19] des méthodes complémentaires de pondération existent pour en limiter les effets

[20] des méthodes comme le « Bootstrap » permettent néanmoins d'estimer la variabilité des estimations grâce à un re-échantillonnage par tirage aléatoire avec remise (Efron, B. et Tibshirani, R., 1994)

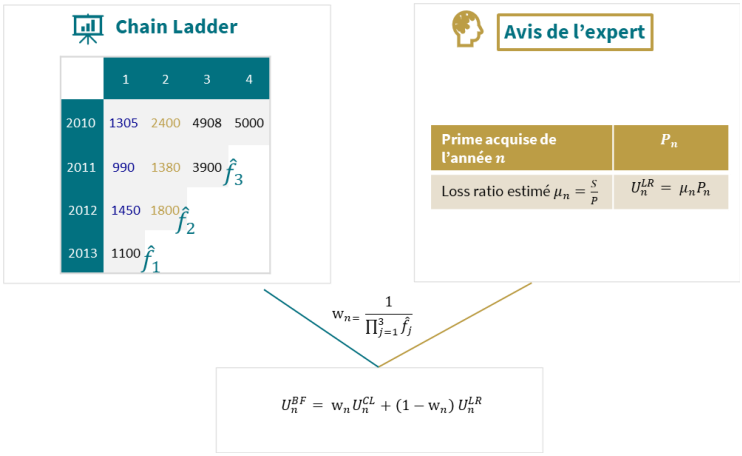
2.3 Bornhuetter-Ferguson

Cette méthode (Bornhuetter & al, 1972) utilise également les triangles de développement tout en permettant de s'affranchir du lien entre la charge ultime et le dernier règlement cumulé. Le but est de solutionner le manque de robustesse du Chain Ladder sur les années de survenances récentes, notamment lorsqu'il s'agit d'une branche longue.

Pour cela, elle introduit une donnée exogène d'exposition en ajoutant au Chain Ladder une estimation du loss ratio[21]. Ce dernier s'obtient généralement « à dire d'expert », ce qui présente l'avantage de lier le calcul des provisions à la stratégie connue de souscription.

La projection à l'ultime est donc l'addition des pertes à l'instant présent (connues) auxquelles sont ajoutés les développements futurs estimés grâce à la combinaison crédibilisée des cadences de développement du Chain Ladder et d'un loss ratio estimé. Ce faisant, le modèle permet de projeter un montant de charge ultime strictement positif même si la dernière position connue est nulle[22].

Figure 37: Illustration du Bornhuetter Ferguson



Ainsi plus le facteur de développement est petit, plus l'actuaire donnera de la crédibilité à l'estimé déterminé par le Chain Ladder. Ce sera le cas des années matures cumulant plusieurs périodes de développement. Inversement, plus le développement espéré est élevé, plus l'actuaire donnera de poids au loss ratio estimé à dire d'expert. Ce sera notamment le cas des années récentes dont les périodes de développement sont peu nombreuses.

2.3.1 Le cadre formel

L'hypothèse fondamentale est l'indépendance entre la projection et le dernier montant connu des règlements $C_{i,I+1-i}, i \in \{1,\dots,I\}$. L'estimation à priori des montants de charge ultime $\widehat{U}_i, i \in \{1,\dots,I\}$ s'obtient ainsi : $\widehat{U}_i = \Phi_i \times P_i$,

[21] Le loss ratio, également appelé S/P, est le rapport des sinistres sur primes

[22] Ce qui peut arriver dans les premières années de développement d'une branche longue telle que la RC

Avec :

- Φ_i le loss ratio attendu pour l'année de survenance concernée i,
- P_i le montant de primes acquises ultimes pour l'année de survenance concernée i,

L'estimateur $\hat{C}_{i,J}^{BF}$ de la charge à l'ultime, avec cette méthode, pour une année de survenance donnée i est :

$$(1)\ \hat{C}_{i,J}^{BF} = C_{i,I-i} + (1 - \hat{z}_{I+1-i})\,\widehat{U}_i,$$

Avec :

- $\hat{z}_j$ la cadence après j années de développement des règlements cumulés, calculée par le Chain Ladder

On en déduit alors l'estimateur de la provision à constituer pour l'année de survenance concernée :

$$\hat{R}_i^{BF} = (1 - \hat{z}_{I+1-i})\,\widehat{U}_i,$$

Vient ainsi l'estimateur de la provision globale :

$$\hat{R}^{BF} = \sum_{i=1}^{I} \hat{R}_i^{BF}.$$

2.3.2 Forces et faiblesses

Cette méthode est utile lorsque les pertes connues ne sont pas un bon indicateur des IBNR. C'est souvent le cas des branches à faible fréquence mais à haute intensité. Elle peut être implémentée malgré un volume faible de données, par exemple lors de la création d'un nouveau produit. Elle permettra de lisser la variance des projections due à des sinistres majeurs survenus tôt dans le développement de l'année de survenance, ce qui peut être adapté à des branches longues et volatiles comme la Responsabilité Civile.

A contrario, l'inertie inhérente à l'application d'un loss ratio à priori ne permet pas de refléter des changements récents d'expérience. Ce même loss ratio est d'ailleurs nécessairement sujet à caution, car arbitraire.

Enfin, un inconvénient technique peut survenir lorsque le facteur de développement est plus petit que 1. La méthode étant une moyenne pondérée selon la crédibilité des résultats, elle peut également se résumer ainsi :

$$(2)\ \hat{C}_{i,J}^{BF} = C_{i,I-i} \times \hat{\lambda}_k \times \hat{z}_{I+1-i} + (1 - \hat{z}_{I+1-i})\,\widehat{U}_i\ ,^{23}$$

Or $\hat{z}$ doit donc être compris entre $[0;1]$ ce qui n'est vrai que si $\frac{1}{\hat{\lambda}_k} \leq 1$ ou encore $\hat{\lambda}_k \geq 1$. Pour cette raison, il est préférable de prendre le maximum entre $\hat{\lambda}_k$ et 1 lors des calculs.

Il est à noter que (1) *et* (2) sont algébriquement équivalent : dans la première équation les sinistres survenus mais partiellement développés sont ajoutés aux sinistres attendus via le loss ratio à priori, lui-

[23] *Pour rappel,* $\hat{\lambda}_k$ *est le facteur de développement du Chain Ladder, avec* $\hat{\lambda}_k = \frac{\sum_{i=1}^{n-k} C_{i,k+1}}{\sum_{i=1}^{n-k} C_{i,k}}$ pour $k = 1, \dots, n$

même pondéré par un facteur crédibilisé de sous développement $(1 - \hat{z}_{I+1-i})$. Dans la seconde équation, ces mêmes sinistres survenus sont d'abord projetés à l'ultime grâce au Chain Ladder, le total est pondéré par un facteur de développement crédibilisé $\hat{z}_{I+1-i}$ et ajouté aux sinistres attendus pondérés par le facteur crédibilisé de sous développement décrit à (1).

2.4 Le modèle linéaire généralisé (GLM)

(Nelder & al, 1972) ont les premiers introduits cette classe de modèles qui regroupe les lois appartenant à la famille des exponentielles[24]. Le GLM est utilisé pour évaluer et quantifier la relation entre une variable réponse et des variables explicatives. L'idée est de généraliser les modèles linéaires classiques en termes de loi de probabilité et de lien à la linéarité. L'hypothèse sur la distribution est ainsi remplacée par une propriété de linéarité et une relation espérance-variance.

Le GLM diffère de la régression linéaire classique sur deux aspects :

- La distribution de la variable réponse est choisie parmi la famille des exponentielles. Ainsi, elle n'a pas besoin d'être une Normale- *ou même de l'approcher*. La variable réponse peut alors être hétéroscédastique[25], permettant ainsi à sa variance d'être fonction de sa moyenne,
- La moyenne de la variable réponse est elle-même liée linéairement aux variables explicatives,

Cette classe de modèles a pris une importance importante dans le provisionnement car l'hypothèse de normalité est rarement applicable dans la réalité assurantielle : le montant des sinistres ou leur fréquence ne suivent pas nécessairement une distribution normale. Par ailleurs, la relation entre la variable dépendante et les variables de risque est souvent multiplicative, par opposition à la relation additive.

Un GLM est caractérisé par trois hypothèses, reprises une à une dans le cadre formel : la première a trait à la distribution de la variable explicative, la seconde concerne l'expression de la linéarité des variable(s) explicative(s) et la troisième concerne le lien à la linéarité entre variable(s) explicative(s) et valeur prédite.

La prochaine section rappelle le cadre général de la modélisation d'un GLM dans le cadre d'une régression puis présente les pénalisations Ridge et Lasso.

2.4.1 Cadre formel

Soit Y la variable dépendante, fonction d'une combinaison linéaire des variables explicatives $\mathrm{X} = \left(X_1, X_2, \ldots, X_p\right)$ pouvant être qualitatives ou quantitatives. L'objectif est de trouver :

$$\mathbb{E}(Y|X = x),$$

Si la variable dépendante appartient bien à la famille des exponentielles, la densité de sa distribution est vérifiée par l'équation suivante :

$$f_{Y_i}(y_i, \theta_i) = exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + w(y_i, \phi)\right\}$$

[24] Lois discrètes : exponentielle, binomiale, Poisson… et lois continues : Gaussienne, Gamma, Weibull…
[25] La variance des résidus n'est pas constante

Avec :

- θ_i le paramètre naturel permettant de trouver $\mathbb{E}(Y_i)$ grâce à la relation $\mathbb{E}(Y_i) = \mathrm{b}'(\theta_\mathrm{i})$
- ϕ le paramètre de dispersion, qui servira à ajuster la variance du modèle à celle observée,
- b : son choix détermine la distribution de f,

Le prédicteur linéaire, qui est une composante déterministe du modèle, est le vecteur à n composantes suivant :

$$\eta = \mathrm{X}\beta = \beta_0 + \sum_{i=1}^{p} \mathrm{X}_i\beta_i$$

Avec :

- $\beta = (\beta_0, \beta_1, \ldots, \beta_p)$ le vecteur des coefficients inconnus à estimer,
- β_0 une constante
- X_i pouvant être des variables explicatives, ou leurs transformations (logarithme, raciné carré ou une transformation polynomiale), ou leurs interactions (par exemple $\mathrm{X}_k = \mathrm{X}_i \times \mathrm{X}_j \; avec \; \forall i, j \in [\![1, p]\!]$), ou encore l'encodage de variables qualitatives.

La relation fonctionnelle exprime la relation entre la composante aléatoire $f_{Y_i}(y_i, \theta_i)$ et le prédicteur linéaire grâce à une fonction de lien appelée g monotone et différentiable au moins une fois :

$$g\big(\mathbb{E}(Y|X = x)\big) = \eta = \mathrm{X}\beta$$

Et sa réciproque,

$$\mathbb{E}(Y|X = x) = g^{-1}(\eta) = g^{-1}(\mathrm{X}\beta)$$

2.4.2 Forces et faiblesses

Les modèles GLM permettent une exécution rapide et son facilement interprétables. Ils permettent également de donner une estimation de l'erreur (écart type) et d'observer l'effet marginal des variables explicatives sur la variable dépendante. En cela, ils sont plus puissants que la régression linéaire classique. Un GLM peut même être déployé dans une feuille Excel.

Les modèles GLM nécessitent néanmoins des hypothèses fortes quant à la distribution et la variabilité du terme d'erreur et ne permettent pas de capter les interactions non linéaires. Les variables explicatives doivent être non corrélées et la prédiction est sensible aux valeurs aberrantes.

2.4.3 Application

L'application concrète d'un GLM nécessite tout d'abord de choisir une distribution $f_{Y_i}(y_i, \theta_i)$ et par là même de choisir $b(\theta_i)$. La distribution de la variable doit être adaptée aux données.
Ensuite, la fonction de lien η est sélectionnée : les principaux modèles linéaires généralisés en assurance non vie sont présentés en Tableau 7.

Les variables explicatives X_i vont permettent la modélisation de η. Il faut ensuite collecter les observations $(y_1, y_2 \ldots y_n)$ de la variable réponse y ainsi que les valeurs correspondantes $(x_1, x_2 \ldots x_n)$ des variables

explicatives. Les observations sont présumées indépendantes[26]. Le modèle est ensuite ajusté grâce à l'estimation du maximum de vraisemblance[27] : l'objectif est de trouver le vecteur β ou, si inconnu, le paramètre de dispersion ϕ.

La qualité d'ajustement du modèle sera mesurée par les écarts entre les valeurs prédites par le modèle et les observations initiales, en prenant garde de séparer ce qui est attribuable au modèle lui-même du simple hasard[28].

Tableau 7: les principaux GLM en assurance IARD, extrait du mémoire d'actuaire (Yufei Luo, 2016)

Loi de $Y\|X=x$	**Type de** $Y\|X=x$	**Problème**	**Modèle IARD**	**Nom du lien canonique**	**Fonction de lien canonique**
Bernoulli	Discrète : oui/non	Régression logistique	taux de transformation/résiliation et modèle de propension	lien logit	$g(\mu) = log(\mu/1-\mu)$
Poisson	Discrète : comptage	Régression de poisson	fréquence des sinistres et nombre de sinistres	lien log	$g(\mu) = log(\mu)$
Gamma	Continue positive	Régression sur variable dépendante quantitative asymétrique	coût moyen des sinistres	lien réciproque	$g(\mu) = -1/\mu$
Log-normal	Continue positive	Régression sur variable dépendante quantitative asymétrique	coût moyen des sinistres	lien identité	$g(\mu) = \mu$

Dans la mesure où le réseau de neurones est un GLM à plusieurs couches, le GLM simple ne sera pas appliqué dans cette étude.

2.5 GLM régularisés (Lasso, ridge regression)

La régularisation d'un GLM consiste à limiter le nombre de variables explicatives tout en préservant la qualité de généralisation du modèle qui devient alors parcimonieux. L'intérêt est de limiter le temps de calcul ainsi que les risques de sur apprentissage. En effet, un modèle trop précis se comportera de façon instable lorsque présenté avec de nouvelles données. Réduire la dimension du modèle peut réduire la variance.

L'aperçu des techniques de régularisation et de sélection de variables donné ci-dessous a pour but l'acquisition de l'intuition nécessaire à la compréhension des algorithmes développés au second chapitre, comme l'élagage dans la forêt aléatoire. Les régularisations Ridge et Lasso ont pour objectif de minimiser le maximum de vraisemblance pénalisé. Toutes deux sont optimisées par validation croisée.

[26] L'échantillon sera considéré comme un tirage aléatoire en provenance de la population concernée,
[27] Il s'agira de résoudre des équations non linéaires grâce à l'utilisation de méthodes itératives,
[28] La déviance et les résidus de Pearson serviront à mesurer le modèle tandis que l'analyse des résidus déterminera l'effet du hasard,

2.5.1 La régularisation Ridge

La régularisation Ridge conserve l'ensemble des variables mais contraint la norme des paramètres β_j en limitant la grandeur des valeurs qu'il peut prendre, donc sa variance. Ainsi l'estimateur ridge de β dans le modèle

$$Y = \mathrm{X}\beta + \varepsilon$$

Se définit par un critère des moindres carrés auquel est ajouté une pénalité l_2 (dernier terme):

$$\hat{\beta}^{Ridge} = \arg\min_{\beta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_i^j \beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Avec :

- λ le paramètre de complexité.

Ce dernier, selon la pénalité imposée, baissera le biais mais augmentera la variance et inversement. Une valeur à zéro obligera le modèle à passer par tous les points : il oscillera donc fortement entre chaque points (surajustement). Le dernier terme agit donc bien comme une contrainte.

La régularisation Ridge doit être utilisée sur des variables centrées et réduites afin d'appliquer un paramètre de complexité unique à toutes les variables. Elle permet ainsi de régler le sujet posé par la colinéarité des variables explicatives mais ne permet pas de limiter le nombre de variables explicatives. La régularisation Lasso le permet.

2.5.2 La régularisation Lasso

La régularisation Lasso[29] est identique à Ridge mais contraint le modèle par une pénalité l_1. Ainsi,

$$\hat{\beta}^{Lasso} = \arg\min_{\beta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_i^j \beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Avec :

$$\sum_{j=1}^{p} |\beta_j| = \|\beta\|_1$$

Le paramètre de complexité contrôlera ici la parcimonie du modèle, c'est-à-dire le nombre de variables explicatives prises en compte par le modèle, en leur appliquant un coefficient (si valeur nulle : la variable est ignorée). Tout comme la régularisation Ridge, le dernier terme est à optimiser de façon à minimiser l'erreur de prédiction du modèle. Cette méthode est utile pour sélectionner des variables dans des problèmes de grandes dimensions. Néanmoins elle ne fonctionnera pas bien si des variables non influentes sont corrélées avec celles qui le sont.

La compréhension théoriques de ces deux types de régularisation sera utile lors de la troisième partie consacrée à l'apprentissage statistique automatique, également appelé Machine Learning.

[29] Acronyme pour "Least Absolute Shrinkage and Selection Operator"

3. Les apports du Machine Learning

Les méthodes présentées jusqu'ici restent très largement utilisées par les compagnies d'assurance. Leur succès tient à une implémentation simple, des temps de calculs quasi-instantanés, un traitement global de l'information et une interprétabilité totale. Elles ne permettent néanmoins pas d'optimiser l'information à disposition de l'assureur : il est pourtant raisonnable de penser qu'une information plus détaillée permettrai un provisionnement plus précis.

La science des données permet de répondre à ce nouveau défi en conjuguant l'inférence statistique et l'algorithmie. Elle a pour objectif l'exploration, l'analyse et la transformation des données à des fins de prédiction.

L'apprentissage statistique automatique[30] quant à lui est un des composants de la science des données et consiste à choisir parmi un ensemble de modèles et hyperparamètres concurrents celui dont la prédiction est la meilleure[31]. A cette fin, les données sont allouées en deux parties disjointes : l'une, usuellement la plus riche en termes de données, sert à entraîner le modèle tandis que la seconde est utilisée pour vérifier la performance du modèle sur des données non vues lors de l'entraînement.

Contrairement à la statistique classique, l'apprentissage statistique ne nécessite pas de formuler des hypothèses sur la structure et la distribution des données. Une seule hypothèse est nécessaire : les données à prédire sont générées de façon identiques et indépendantes par un processus aléatoire à partir du vecteur des variables explicatives. Le résultat de cet apprentissage est une fonction qui fait intervenir des variables explicatives et devient de plus en plus complexe à mesure que l'algorithme "apprend", permettant ainsi de capturer les singularités de la structure des données – par exemple des interactions ou comportements non linéaires.

Plus formellement, pour le calcul des réserves pour le triangle Δ^{k}, l'actuaire doit trouver le modèle M de sorte que $\widehat{R_{i,j}^{k,M}} = M(X = \Delta^{k})$, qui correspond à une sélection de modèles dans l'espace $\mathcal{M}$ contenant tous les modèles possibles.

La sélection de $\mathrm{M} \in \mathcal{M}$ se fait alors sur la performance attendue de la prédiction des montants futurs de sinistres, mais comment l'évaluer ? Plutôt que l'utilisation de maximum de vraisemblance qui nécessite une hypothèse à priori sur la distribution des montants de sinistres, le scoring des modèles s'effectue par la confrontation des valeurs ultimes prédites vs réelles.

[30] Également appelé Machine Learning

[31] De nombreux indicateurs existent afin d'objectiver la signification de « meilleur » (voir suite de l'étude)

3.1 Définition du provisionnement individuel

Les assureurs génèrent une quantité importante d'information à la maille du sinistre. Or leur agrégation en deux dimensions, telle qu'étudiée en première partie, empêche la prise en compte de leurs caractéristiques individuels. L'objectif du provisionnement individuel, à l'instar des méthodes classiques, est de projeter la prochaine diagonale du triangle de run-off. La différence réside néanmoins dans l'absence d'agrégation des sinistres, permettant ainsi l'utilisation des données individuelles comme variables explicatives.

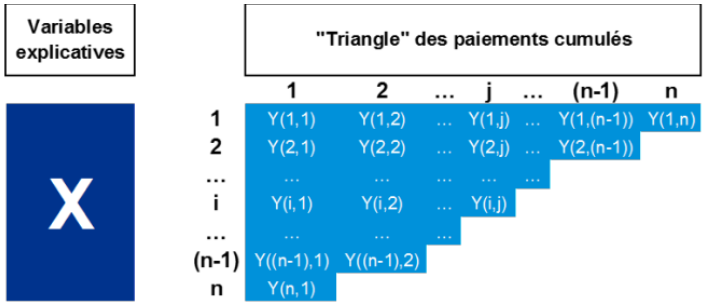
Soit:

- N une base de données contenant des sinistres survenus entre les années j=1 et n,
- N_i le nombre de sinistres survenus l'année i
- T le triangle de paiements connus
- $T_{\alpha,j}$ le développement des paiements[32] en j du sinistre α
- p le nombre de variables explicatives
- $Y_\alpha(n)$ les paiements cumulés du sinistre α

Chaque élément du « triangle des paiements cumulés » - *voir* Figure 38 *ci dessous*- est un vecteur de taille N_i. Le vecteur $Y_{i,j}$ contient le développement de l'année j des sinistres survenus l'année i.

L'objectif de l'algorithme d'apprentissage statistique de provisionnement sera de trouver la régression f tel que : $f(X_\alpha, T_\alpha) = Y_\alpha(n)$

Figure 38: représentation donnée par (Fabre Rudelle, 2018)



[32] Les provisions dossier-dossier ou la charge des sinistres sont des alternatives tout aussi acceptables du point de vue du modèle

3.2 La modélisation

Un modèle de provisionnement adopte nécessairement l'une de ces deux approches, selon la granularité des données à disposition:

1. Une approche agrégée, dans lequel les données sont consolidées par période (trimestriel ou annuel)
2. Une approche individuelle, dans lequel l'actuaire dispose de l'essentiel des informations du contrat et/ou des sinistres et tente d'en traiter tout ou partie.

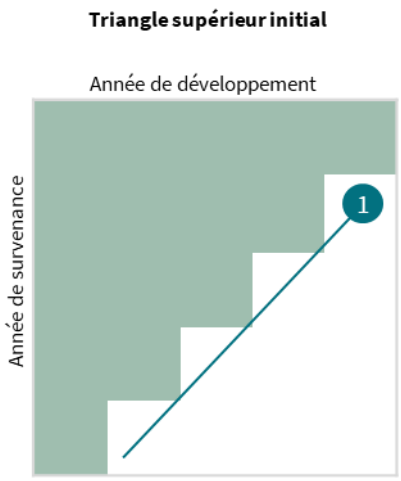
L'approche agrégée- *objet de la première partie de cette étude*- a été largement traitée dans la littérature actuarielle et est utilisée dans l'immense majorité des compagnies d'assurance (ASTIN, 2016). L'approche individuelle quant à elle est plus récente et nécessite l'utilisation de techniques d'apprentissage automatique.

3.2.1 Fonctionnement de la prédiction ligne à ligne

La modélisation utilisée dans la seconde partie de ce mémoire est une adaptation du « cascading », rendue publique dans les travaux ASTIN (B. Harej & al, 2017). Elle consiste, à partir d'un triangle supérieur initial, à prédire itérativement la prochaine diagonale jusqu'à compléter entièrement le triangle inférieur.

Les schémas suivants illustrent la démarche à chaque étape du processus. L'initialisation de la prédiction (diagonale n°1) s'effectue sur l'ensemble des valeurs réelles à disposition (zone colorée de la Figure 39).

Figure 39: Initialisation de la première diagonale



L'algorithme intègre ensuite cette première diagonale de prédiction au triangle supérieur des valeurs réelles : la boucle suivante utilise ainsi les zones colorées des illustrations ci-dessous, jusqu'à compléter le triangle inférieur, formant ainsi un jeu complet de projections à l'ultime.

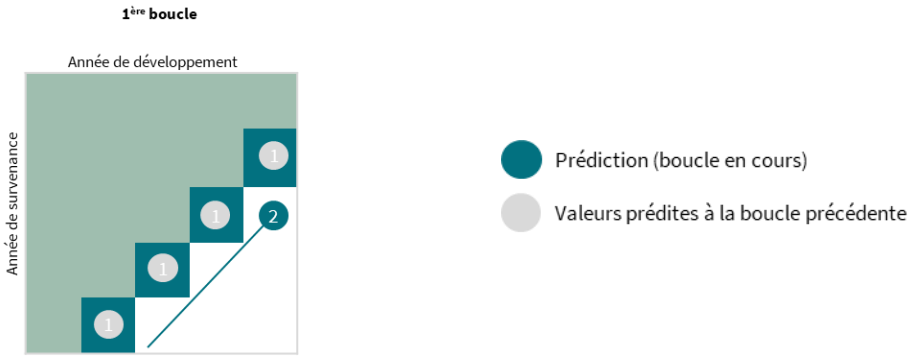
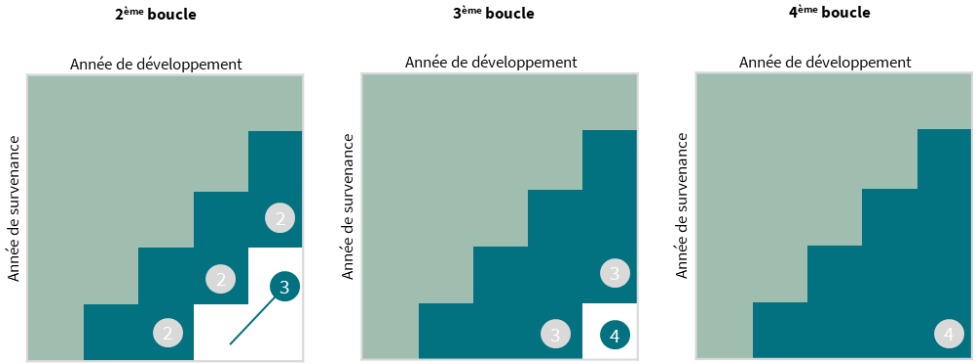


Figure 40: boucles suivantes jusqu'à complétude du triangle inférieur



Ces projections se font transaction par transaction au sein de chaque sinistre, on parle alors de « ligne à ligne ». Il est impératif d'établir un maximum de liens entre la prédiction itérative des diagonales et les opérations algorithmiques effectuées à la maille transactionnelle. C'est pour cela que la charge de sinistres payés est cumulée par transaction : la nouvelle variable permet ainsi d'apporter de l'information issue des transactions précédentes. En procédant ainsi, le risque de rupture de l'hypothèse H_3 stipulant que les transactions de chaque sinistre doivent être indépendantes les unes des autres s'accroit. Sa contribution à la prédiction, décrite à la section 4.3 de « Interprétation des modèles de type boîte noire », révèlera que ce choix était judicieux.

Comme décrit à la section « Les données», la fréquence à laquelle les sinistres se développent est ici trimestrielle. Néanmoins, chaque trimestre ne donne pas automatiquement lieu à une transaction : si aucun mouvement en payé ou provision n'a été généré, aucune transaction n'est enregistrée. L'étape d'apprentissage va néanmoins nécessiter que chaque ligne dispose d'une variable réponse : il s'agira du montant de la transaction du trimestre suivant. Pour cela, chaque trimestre doit contenir au moins une transaction par trimestre de sa durée de vie : si un trimestre est absent, une ligne factice est ajoutée (montant à « 0 ») comme l'illustre la ligne verte du tableau ci-dessous[33].

Tableau 8: ajout des transactions trimestrielles manquantes

Illustration sur un sinistre

N° de sin	Année de développement	Trimestre de développement	Montant payé	Statut du sinistre	Montant payé au trimestre suivant
Sin 1	2015	Q2	100	Ouvert	*Colonne inexistante à ce stade du processus.*
Sin 1	2015	Q3	50	Ouvert	
Sin 1	2015	Q4	0	Ouvert	
Sin 1	2016	Q1	300	Fermé	

Chaque sinistre disposant à présent d'au moins une transaction par trimestre tout au long de sa vie, il est possible d'ajouter la variable réponse ligne à ligne en copiant la valeur de la transaction du trimestre suivant. La valeur -1 est arbitrairement appliquée lorsque le sinistre est clos.

Tableau 9: Création de la colonne paiement du trimestre suivant

Illustration sur un sinistre

N° de sin	Année de développement	Trimestre de développement	Montant payé	Statut du sinistre	Montant payé au trimestre suivant
Sin 1	2015	Q2	100	Ouvert	50
Sin 1	2015	Q3	50	Ouvert	0
Sin 1	2015	Q4	0	Ouvert	300
Sin 1	2016	Q1	300	Fermé	-1

[33] L'illustration vaut à la fois pour les montants payés et provisionnés

L'apprentissage ligne à ligne peut débuter : les valeurs des 18 variables explicatives de chaque ligne doivent permettre de prédire le montant de la transaction au prochain trimestre selon l'illustration simplifiée ci-dessous.

Tableau 10: apprentissage ligne à ligne

Sens de l'algorithme

	N° de sin	Année de développement	Trimestre de développement	Montant payé	Statut du sinistre	Montant payé au trimestre suivant
Variables explicatives	Sin 1	2015	Q2	100	Ouvert	Variable réponse 50
	Sin 1	2015	Q3	50	Ouvert	0
	Sin 1	2015	Q4	0	Ouvert	300
	Sin 1	2016	Q1	300	Fermé	-1

L'algorithme classe les transactions par ordre croissant de développement (du plus ancien au plus récent) et utilise les variables explicatives pour estimer la variable réponse $\hat{f}(x)$. Le modèle est ensuite appliqué sur l'ensemble de la base, puis les prédictions sont agrégées par année de survenance et année de développement afin de reconstituer le triangle total, comme illustré ci-dessous.

Tableau 11: Prédictions puis reconstitution du triangle agrégé

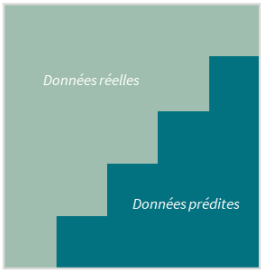
Prédictions ligne à ligne

N° de sin	Année de développement	Trimestre de développement	Montant payé	Statut du sinistre	Prédiction pour le trimestre suivant
Sin 1	2015	Q2	100	Ouvert	$\hat{f}(x_{1,1})$
Sin 1	2015	Q3	50	Ouvert	$\hat{f}(x_{1,2})$
Sin 1	2015	Q4	0	Ouvert	$\hat{f}(x_{1,3})$
Sin 2	2016	Q1	300	Ouvert	$\hat{f}(x_{2,1})$

Agrégation par année de survenance et trimestre de développement

Année de développement	Trimestre de développement	Cumul des prédictions
2015	Q1	Montant 1
2015	Q2	Montant 2
2015	Q3	Montant 3
2015	Q4	Montant 4

Reconstitution du triangle



De manière plus générale, la modélisation fait les hypothèses suivantes :

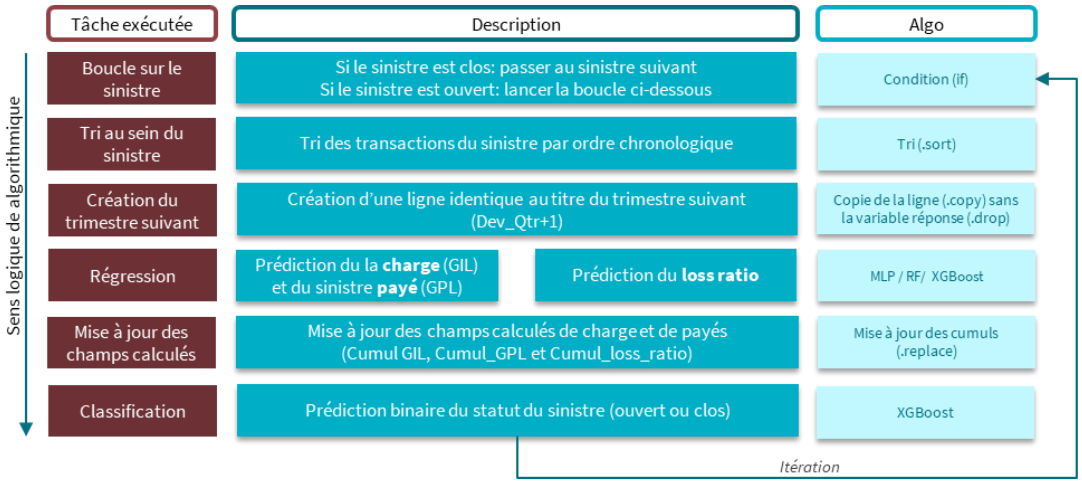
- H_1 : un sinistre clos ne peut être réouvert (valeur -1),
- H_2 : les sinistres sont indépendants les uns des autres,
- H_3 : les transactions de chaque sinistre sont indépendantes les unes des autres,
- H_4 : un sinistre ne peut pas être négatif,
- H_5 : une transaction de sinistre peut être négative (recours ou annulation),

3.2.2 Le modèle

Le modèle est une combinaison de deux sous-modèles fonctionnant conditionnellement l'un avec l'autre : un classificateur binaire de modélisation du statut du sinistre, couplé à un régresseur pour la prédiction du montant de chaque transaction trimestrielle du sinistre tant que le statut du sinistre reste ouvert.

Le schéma ci-dessous résume les étapes successives effectuées lors de la construction du modèle.

Figure 41: Vision globale du modèle de prédiction



Après les étapes de features engineering et nettoyage, la base de données est utilisée telle quelle en entrée du modèle. Les variables catégorielles sont transformées en valeurs binaires 0 et 1 dans des colonnes séparées. Cela fait d'autant plus sens que les algorithmes choisis dans l'étude utilisent des poids dans leurs calculs (voir «Le Random Forest » et « Le Gradient Boosting »). Cette commande, appelée OneHotEncoding, est appliquée à la fois au problème de classification et de régression, le statut du sinistre étant la seule différence s'agissant de la variable réponse de la classification.

Ensuite chaque modèle d'apprentissage automatique est entraîné tour à tour sur 80% des sinistres[34] rattachés aux années de survenance 2010 à 2020. La variable réponse sur laquelle sont entraînés les modèles est le montant trimestriel de sinistres payés. Deux raisons à cela : l'essentiel de la base étant entièrement liquidée, hypothèse est faite que le montant cumulé des sinistres est équivalent à la charge ultime, donc redondante avec la charge totale (« GIL » dans la base). Ensuite, le loss ratio n'est que la version « mise à l'échelle[35] » du montant des sinistres payés : elle permet de prendre en compte l'évolution de l'exposition du portefeuille au cours du temps mais son comportement est présumé identique.

Chaque algorithme est ensuite hyperparamètré différemment selon qu'il s'agisse de la forêt aléatoire, du XGBoost ou du réseau de neurones[36]. Rappelons que les hyperparamètres permettent d'estimer les paramètres du modèle :

[34] La séparation entre base d'apprentissage et de test est faite à la maille sinistre (et non à la maille transaction), après mélange aléatoire (« random.seed » puis « random.shuffle »)
[35] Le jargon consacré préfèrera le mot « scalée »
[36] L'étape d'hyperparamétrage est détaillée à la section « application » des algorithmes concernés

- Les paramètres sont inférés à partir des données: il s'agit des estimateurs d'une loi paramétrique[37] ou du modèle lui-même. La pente dans une régression linéaire ou les poids dans un réseau de neurones en sont quelques exemples.
- Les hyperparamètres quant à eux ne peuvent être déduis des données : ils doivent être calibrés manuellement par l'actuaire, souvent par itération. Le taux d'apprentissage dans une descente de gradient ou encore le nombre de couches dans un réseau de neurones en sont les illustrations.

La phase de prédiction est inspirée de (Fabre Rudelle, 2018) : elle est effectuée sur un échantillon couvrant la seule période 2010-2014 afin d'éviter le phénomène de censure. Cette sous partie du triangle est appelée « triangle inclus », comporte 4006 lignes/58 colonnes[38] et sert de base à l'évaluation des algorithmes.

Le triangle inclus réel est présenté dans les tableaux ci-dessous (zone rouge). Il sera utilisé pour comparer les prédictions de chaque application lors des sections suivantes.

Il est à noté que ces chiffres sont issus d'un échantillonnage et ne constituent pas une indication de la profitabilité du portefeuille global d'AxaXL en dommage. Par ailleurs, les primes utilisées pour le loss ratio sont uniquement celles des polices concernées par l'étude : les polices non sinistrées n'ont pas été prises en compte tandis que d'autres ont été retirées lors de l'étape de nettoyage des données.

Tableau 12: triangle inclus des payés (vision incrémentale)

	12	24	36	48	60
2010	1,291,265	5,425,344	3,158,739	3,483,945	1,651,422
2011	1,965,273	2,823,327	3,563,193	930,431	1,783,167
2012	433,741	3,575,189	2,160,406	1,738,190	380,923
2013	676,290	5,475,618	2,486,931	1,164,828	1,106,796
2014	1,040,862	1,940,948	775,287	753,579	758,139

Tableau 13: triangle inclus des loss ratio (vision incrémentale)

	12	24	36	48	60
2010	33.82	18.71	12.61	0.84	1.11
2011	16.07	27.79	11.07	0.41	9.02
2012	37.34	45.42	4.39	9.00	5.90
2013	66.17	41.65	14.97	7.80	7.60
2014	59.36	9.31	3.66	3.45	3.45

La phase d'évaluation contient les sinistres « non vus » lors de la phase d'entraînement. Les années de développement prédites sont comparées une à une (2015 à 2018) ainsi qu'en cumulé afin d'avoir une idée de la capacité du modèle à prédire la charge ultime indépendamment de la variance annuelle.

[37] Par exemple la moyenne ou l'écart type
[38] Après OneHotEncoding

Enfin, le modèle « Explainable Boosting Machine » de Microsoft sera lancé sur 80% des données afin de permettre l'interprétabilité des boîtes noires. Il n'est pas représenté sur le schéma général de la section « 3.2 Le modèle » car il est redondant avec XGBoost mais il donne accès à un ensemble de mesures globales et locales de façon native.

3.2.3 L'évaluation des modèles

Le montant de charge à l'ultime à provisionner peut-être évalué de deux façons : à partir des seuls mouvements de paiements ou à partir de la charge totale (paiement + provisions dossier-dossier). Grâce à l'hypothèse selon laquelle les montants de sinistres payés et de charge sont égaux à l'ultime, l'étude se limitera à prédire le montant des seuls paiements. Cette règle est en ligne avec le type de branche considérée : il est fréquent de modéliser la charge ultime de branches longues à l'aide de triangles de charges et à l'aide de triangle des paiements les branches courtes.

Pour rappel, la phase d'apprentissage est conduite sur 80% des sinistres dont l'année de survenance se situe entre 2010 et 2014. Les 20% restant sont utilisés pour le test.

Le protocole de test consiste à réagréger les prédictions de chaque modèle afin de les comparer à la charge ultime réelle. La qualité prédictive sera mesurée pour chaque modèle à l'aide des indicateurs suivants: l'erreur de prédiction, l'erreur du modèle au global et la corrélation entre valeurs prédites et valeurs réelles. Soit M le modèle de prédiction et C les valeurs réelles :

L'erreur de prédiction est la RMSE[39], c'est-à-dire la moyenne arithmétique de la racine des écarts au carré entre les prédictions et les observations. Elle permet d'évaluer la proximité des prédictions avec la réalité. Elle donne ainsi un poids important aux erreurs importantes, ce qui correspond aux enjeux de l'assurance grands risques.

$$D_n(M) = \frac{1}{n}\sqrt{\sum_{i=1}^{n}\left(C_{i,n} - M_{i,n}\right)^2}.$$

Avec :

- $C_{i,n}$ la charge ultime pour l'année de survenance i donnée par les valeurs réelles,
- $M_{i,n}$ la charge ultime pour l'année i donnée par le modèle d'apprentissage statistique automatique M,

L'erreur relative du modèle permet de quantifier l'erreur relative entre les deux charges. Plus ER_M est proche de zéro, plus le modèle prédit une charge proche de la réalité.

$$ER_M = \frac{R_M - R}{R}$$

[39] Root Mean Square Error = moyenne de la racine des erreurs au carré

Avec :

- R_M le montant de la charge prédite par le modèle,
- R le montant de la charge réelle,

Afin de mesure à quel point les charges prédites par le modèle évoluent comme les charges réelles, nous utiliserons la corrélation suivante :

$$C_M = \frac{1}{m} \sum_{i=1}^{m} cor\ (M_i, C_i)$$

Avec :

- m le nombre d'années de survenance,
- M_i les valeurs prédites par le modèle M,
- C_i les valeurs réelles,
- $cor\ (M_i, C_i)$ la corrélation entre l'évolution des valeurs prédites et celle des valeurs réelles,

Il est à noter que ce calcul ne peut pas fonctionner sur la dernière année de survenance car seule une donnée réelle est disponible (correspond au 1er développement de la dernière année de survenance) : dans ce cas précis, l'algorithme applique arbitrairement la valeur 1.

Enfin, un score de proximité est calculé afin de faire la synthèse des 3 mesures précédentes :

$$Score\ de\ proximité = \log D_M + |ER_M| - C_M$$

3.3 La modélisation du statut du sinistre

Le modèle de regression seul, sans la modélisation du statut du sinistre, surestime grandement les montants de charge à l'ultime. L'intuition initiale était que ces modèles non linéaires capteraient automatiquement l'information liée à la durée de vie du sinistre : les variables « time_to_notified » et « time_to_inception » décrites à la section « Les données » étaient censées apporter cette information au modèle. Mais les premiers résultats ont démontré que c'était insuffisant.

Le statut du sinistre a donc été modélisé : le projection du trimestre suivant devient ainsi conditionnelle à la prédiction binaire de fermeture du dossier. Si le statut est projeté « clos » à la boucle précédente, la prédiction de la charge s'arrête pour le sinistre concerné.

L'entraînement du modèle a été simplifié comparativement au régresseur : après l'étape OneHot Encoding, seuls la forêt aléatoire et le XGBoost ont été testés. Les premiers résultats sans hyperparamétrage ont donné le XGBoost largement devant en termes de F_1 Score. Rappel synthétique à propos de la façon dont il se calcule :

$$F_1\ score\ =\ 2 \times \frac{Précision\ \times\ Recall}{Précision\ +\ Recall}$$

Avec :

$$Précision = \frac{Nombre\ de\ vrais\ positifs}{Nombre\ de\ vrais\ positifs\ +\ Nombre\ de\ faux\ \boldsymbol{positifs}}$$

Autrement dit, la précision dénombre le pourcentage de prédictions correctes parmi toutes les valeurs que le modèle a prédit « positives » (à tort ou à raison).

Et :

$$Rappel = \frac{Nombre\ de\ vrais\ positifs}{Nombre\ de\ vrais\ positifs\ +\ Nombre\ de\ faux\ \boldsymbol{négatifs}}$$

Autrement dit, le rappel dénombre le pourcentage de valeurs correctement prédites parmi toutes les valeurs positives contenues dans la base.

Après hyperparamétrage, le XGBoost donne un $F_1\ score$ de 79% ainsi qu'une précision de 72%. Il a donc été intégré dans la boucle de prédiction sans hyperparamétrage et les résultats se sont améliorés significativement.

3.4 Le Random Forest

L'algorithme d'apprentissage automatique de la forêt aléatoire est une méthode ensembliste agrégeant plusieurs arbres de décisions décorrélés. Chaque arbre découpe l'espace des variables explicatives en groupes homogènes et ceci à la maille sinistre. Il s'agit d'une modélisation ligne à ligne.

On appelle respectivement arbre de régression/classification un arbre de décision dont la variable d'intérêt est quantitative ou qualitative.

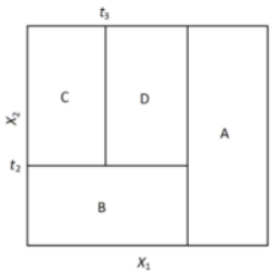
Cette section se concentre sur la prédiction de valeurs quantitatives mais l'algorithme permet également de traiter des problèmes de classification.

3.4.1 Arbre de décision : le chaînon élémentaire

L'arbre de décision partitionne l'espace des valeurs prises par les variables explicatives en rectangles, dans lesquels la variable d'intérêt est constante. Chaque élément de la base est un vecteur multidimensionnel $(x_1, x_2 \dots x_n)$ regroupant les variables descriptives de chaque point. Le nœud de l'arbre correspond, pour des variables numériques, à un test par intervalles de valeurs. Les feuilles de l'arbre spécifient quant à elles les classes.

La classification d'un nouveau candidat se fait par la descente de l'arbre, en commençant par la racine et à travers les feuilles, dont le résultat est donné en Figure 42. A chaque nœud intermédiaire, une variable x_i est testée afin de décider du chemin à prendre pour continuer la descente.

Figure 42: illustration de la construction de l'arbre décisionnel par partition récursive



Au début de l'algorithme, tous les points de la base se situent à la racine de l'arbre : ce dernier se construit alors par partition récursive de chaque nœud, en fonction de l'homogénéité des descendants par rapport à la variable cible (Figure 43). La variable testée sera celle qui maximise cette homogénéité. Le process s'arrête lorsque les éléments d'un nœud ont la même valeur pour la variable cible.

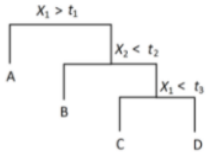


Figure 43: illustration de la construction d'un arbre décisionnel

3.4.2 Algorithme CART de régression

Il existent plusieurs types de modèles utilisant les arbres de décision, le plus utilisé par la recherche actuarielle étant le CART (Breimann,L. et al, 1984). Ce dernier crée des « partitions binaires récursives ».

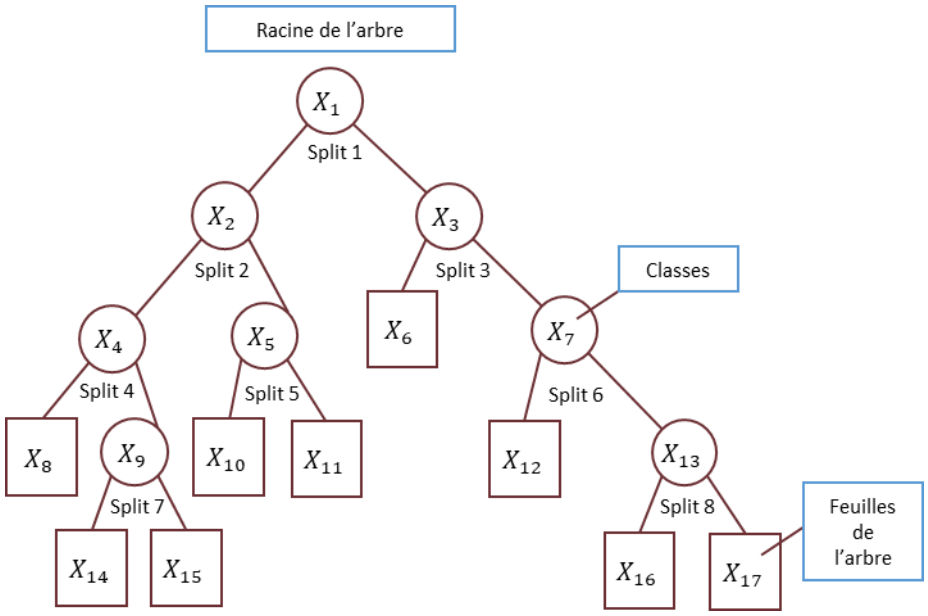
Les deux régions nouvellement crées sont à leur tour séparées en deux sous-régions et ceci jusqu'à ce qu'un critère d'arrêt soit déclenché. L'algorithme consiste à séparer l'espace des observations en deux régions simples et homogènes appelées « nœud » : à chaque étape, l'objectif est de minimiser la variance intra groupe tout en maximisant la variance inter groupe.

Les points intermédiaires au sein de l'arbre sont appelés « nœuds internes », le premier split étant le facteur le plus important dans la détermination de la variable d'intérêt. Les régions restantes non séparées sont appelées « nœuds terminaux » ou feuilles de l'arbre, dans lesquels la valeur attribuée correspondra à la moyenne des observations. Enfin, la profondeur de l'arbre désigne le nombre d'étapes de séparations (8 dans l'illustration de la Figure 44 ci-dessus).

La séparation en régions simples est déterminée par un critère de coupure décrit dans la section suivante. L'essentiel de l'algorithme se résume donc en 3 éléments:

1. Le critère de coupure, étudié en détail dans la prochaine section
2. Le critère d'arrêt, qui peut être une profondeur fixée, l'atteinte d'un nombre de feuilles maximal, l'atteinte du seuil minimal d'observations par nœud ou encore l'absence de gain prédictif marginal.
3. L'allocation des nœuds terminaux à une classe,

Figure 44: Illustration d'un arbre à 6 nœuds, adapté de (Breiman, 1996)



3.4.2.1 *Le critère de coupure*

Soit C_i les variables explicatives et X un nœud quelconque dans l'arbre CART appelé espace d'états. On note N_X le nombre d'observations appartenant au nœud X. Une coupure est un couple (j, z) tel que :

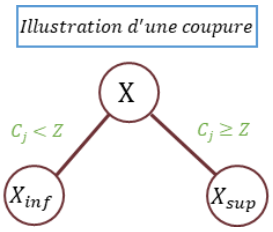
- $j \in \{1, \ldots, p\}$ désigne la variable sur laquelle s'opère la coupure
- z est la valeur du critère de coupure

Le schéma ci-contre illustre la coupure d'un nœud quelconque avec les notations suivantes:

$X_{inf} = \{c \in \mathrm{X};\ c_j < z\}$

$X_{sup} = \{c \in \mathrm{X};\ c_j \geq z\}$

Figure 45: coupure à la $i^{ème}$ coordonnée (par rapport à Z)



Le critère de coupure s'écrit alors ainsi:

$$L_n(j,z) = \frac{1}{N_x} \sum_{i=1}^{n} (Y_i - \bar{Y}_X)^2 \mathbb{1}_{C_i \in X} - \frac{1}{N_x} \sum_{i=1}^{n} \left(Y_i - \bar{Y}_{X_{inf}} \mathbb{1}_{X_{inf}} - \bar{Y}_{X_{inf}} \mathbb{1}_{X_{sup}}\right)^2 \mathbb{1}_{C_i \in X}$$

Avec :

- $\bar{Y}_X$: la moyenne des $Y_i \in X$
- $\bar{Y}_{X_{inf}}$: la moyenne des $Y_{X_{inf}} \in X_{inf}$
- $\bar{Y}_{X_{sup}}$: la moyenne des $Y_{X_{sup}} \in X_{sup}$

La séparation optimale $(\hat{j}_n, \hat{z}_n)$ est celle qui maximise $L_n(j,z)$.

3.4.2.2 *Elagage*

La profondeur de l'arbre est un facteur de complexité : l'élagage agit sur les paramètres de l'arbre décisionnel afin de supprimer les branches peu prédictives et rendre le modèle généralisable. Cette étape repose sur l'idée que la performance n'augmente pas forcément avec la profondeur de l'arbre. Par ailleurs, une trop grande profondeur peut entraîner le sur-apprentissage.

Si l'on suppose un sous-arbre T inclus dans T_0, alors le coût de complexité est :

$$C_\alpha(t) = \sum_{x=1}^{|\mathrm{T}|} Q_x + \alpha |\mathrm{T}|$$

Avec :

- $|\mathrm{T}|$ le nombre de feuilles de T,
- Q_x est l'erreur quadratique dans le nœud x,
- $\alpha > 0$ le paramètre à estimer, qui représente le compromis entre la profondeur de l'arbre et son adéquation aux données (paramètre de pénalisation),

L'actuaire paramètrera les éléments suivants :

- La profondeur maximale de l'arbre,

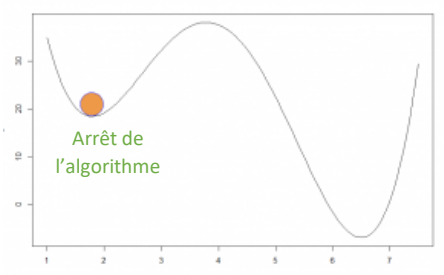
- Le gain minimum pour déclencher la coupure,
- Le nombre d'observations minimal sur une feuille,

3.4.2.3 Avantages et inconvénients

Les arbres de régression présentent l'avantage de l'interprétabilité des résultats grâce au découpage de l'espace, gèrent bien les données manquantes, se comportent bien face aux valeurs extrêmes et leur coût d'utilisation est logarithmique. Enfin, le nombre de paramètres à estimer est limité : la fonction d'impureté qui mesure la qualité de la séparation (souvent la MSE en régression), la profondeur de chaque arbre, le nombre d'observations minimal afin de constituer un nœud et une feuille, le nombre maximum de feuilles, le seuil à partir duquel la fonction d'impureté autorise ou non la séparation d'un nœud.

Néanmoins, le risque de sur-apprentissage augmente à mesure que l'arbre grandit. Les arbres peuvent également ne pas être équilibrés si la base de données de l'est pas : une classe se met alors à dominer excessivement dans la prédiction. La structure de l'algorithme, qui consiste à parcourir les variables explicatives une à une sans y revenir, est sujet aux minimas locaux (voir Figure 46 ci contre).

Figure 46: illustration du risque lié à un minimum local de la fonction de perte



Enfin, les arbres de décision peuvent être instables si des changements, même légers, se produisent dans les données. Plus ces changement sont proches du nœud racine plus la prédiction de l'arbre sera affectée : on dit que les arbres produisent des estimateurs de variance élevée. La prochaine section décrit des méthodes plus sophistiquées permettant d'améliorer la robustesse[40] du modèle CART.

[40] Par robustesse, il est entendu : variabilité des résultats et risque de sur apprentissage

3.4.3 Les méthodes ensemblistes

Une méthode ensembliste est un paradigme d'apprentissage dans lequel des modèles basiques[41] sont entraînés simultanément puis combinés afin d'obtenir un meilleur compromis biais / variance. Le choix du modèle basique doit être cohérent avec l'agrégation qui en sera faite. Par exemple, un modèle basique à faible biais mais forte variance devra être agrégé de manière à réduire la variance (et vice versa). Il existe ainsi deux principaux types d'agrégation : le Bagging et le Boosting.

Le Bagging va considérer des modèles basiques homogènes, c'est-à-dire identique les uns aux autres, et va les entraîner indépendamment les uns des autres, en parallèle. Leurs prédictions sont ensuite moyennées, selon un processus déterministe dont l'objectif sera plutôt l'obtention d'un modèle ensembliste à plus faible variance.

Le Boosting va également considérer des modèles basiques homogènes, mais cette fois ci l'apprentissage est séquentiel, rendant ainsi le processus adaptatif (le modèle basique suivant apprend du précédent). L'agrégation est elle aussi déterministe et aura plutôt pour objectif de baisser le biais.

Le Stacking est équivalent au Bagging mais considère des modèles basiques hétérogènes, c'est-à-dire différents les uns des autres. A l'instar du boosting, l'objectif de ce type d'agrégation sera plutôt de baisser le biais du modèle.

Les prochaines sections décrivent les éléments constitutifs des forêts aléatoires avant d'expliquer la notion elle-même.

3.4.3.1 Bootstrap

Le Bootstrap est l'ancêtre du Bagging et consiste à créer de nouveaux échantillons statistiques par tirage avec remise à partir d'un échantillon initial. Sous les bonnes hypothèses, ces nouveaux échantillons peuvent être à leur tour considérés comme quasi indépendants et identiquement distribués.

Les hypothèses à satisfaire :

- (H_1) l'échantillon initial doit être suffisamment représentatif de la vraie distribution,
- (H_2) l'indépendance entre échantillon initial et Bootstrap doit être préservée (la taille du premier doit être significativement plus importante que celle du second),

Le Bootstrap permet ainsi d'évaluer la précision statistique et peut être appliqué à une grande variété de procédures statistiques. Pour plus de détails, voir (Efron, B. et Tibshirani, R., 1994).

L'illustration du Bootstrap dans le cas paramétrique (Wikipedia, 2021) permet d'acquérir l'intuition nécessaire à sa compréhension en non paramétrique :

- Soit un échantillon $\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n$ de n observations indépendantes et identiquement distribuées selon une loi inconnue F.
- Soit un échantillon Bootstrap $X_1^*, X_2^*, \ldots, X_n^*$ selon $\hat{\mathrm{F}}$ et B le nombre d'échantillons.

Une statistique simple à calculer est la moyenne empirique à partir de l'échantillon Bootstrap :

$$\hat{\theta}_b = \frac{(X_1^* + X_2^* + \ldots + X_n^*)}{n}$$

[41] Appelé également « Weak Learner » ou régresseur faible

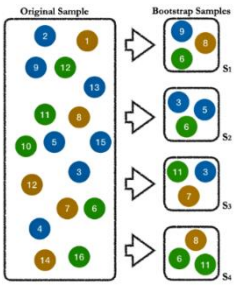
La moyenne empirique $\bar{\hat{\theta}}$ est la moyenne des moyennes empiriques des échantillons Bootstrap $\hat{\theta}_b$. La variance de l'estimateur de la moyenne empirique $\hat{\sigma}^2\left(\bar{\hat{\theta}}\right)$ est approchée par la variance empirique de la population Bootstrap des $\hat{\theta}_b$:

$$\hat{\sigma}^2\left(\bar{\hat{\theta}}\right) = \frac{1}{B}\sum_{b=1}^{B}\left[\hat{\theta}_b - \bar{\hat{\theta}}\right]^2$$

avec $\bar{\hat{\theta}} = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}_b$ qui constitue une alternative à l'estimateur classique $\hat{\theta} = \frac{1}{n}\sum_i X_i$

L'intuition est donc la suivante (Sarkar, 2019): en l'absence d'information complète sur l'ensemble de la population étudiée, sachant que l'échantillonnage est le meilleur moyen d'approximer sa distribution, il est logique de rééchantillonner l'échantillon. Une illustration est donnée en Figure 47 ci dessous.

Figure 47: illustration du Bootstrap (Sarkar, 2019)

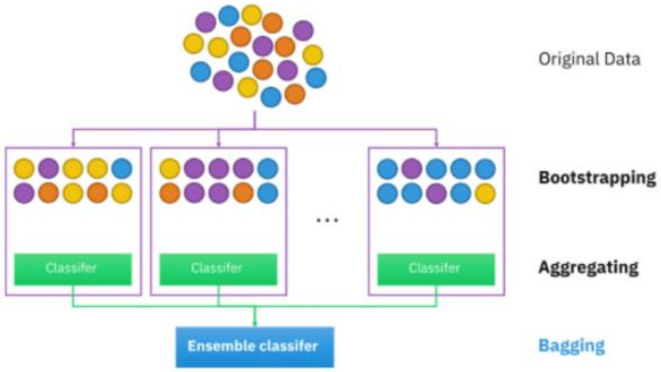


3.4.3.2 *Le Bagging*

La méthode du bagging a été introduite par (Breiman, 1996), dont le nom est la contraction de « bootstrap » et « aggregating ». Elle consiste à lancer l'apprentissage de modèles basiques indépendamment puis à moyenner les prédictions afin d'obtenir un modèle ensembliste à plus faible variance. La limite des données ne permettant pas d'entraîner complètement chaque modèle basique, la méthode utilise les propriétés statistiques conférées par le Bootstrap.

Soit L_n un échantillon d'apprentissage, le bagging va consister à sortir B échantillon Bootstrap $(L_{n,1}, L_{n,2} \dots L_{n,B})$ puis calibrer un modèle basique[42] sur chacun afin de construire B estimateurs $\hat{\phi}_1(.) \dots \hat{\phi}_B(.)$.

Figure 48: illustration du bagging (Joshi, 2020)



L'estimateur final est alors la moyenne des différents estimateurs :

$$\hat{\phi}(.) = \frac{1}{B}\sum_{b=1}^{B} \hat{\phi}_b(.)$$

Moyenner les résultats issus des modèles basiques ne change pas la qualité de l'estimateur mais baisse la variance des résultats. (Biau, 2008) a par ailleurs démontré la consistance universelle du bagging en prenant comme exemple la méthode des plus proches voisins (qui, comme CART, n'est pas universellement consistant) et des échantillons d'apprentissage Bootstrap L_n avec :

- $\lim_{n\to\infty} L_n = +\infty$

Et

- $\frac{\lim_{n\to\infty} L_n}{n} = 0$

Il a ainsi démontré que l'estimateur ensembliste asymptotique $\hat{\phi}(.) = \lim_{B\to\infty} \frac{1}{B}\sum_{b=1}^{B} \hat{\phi}_b(.)$ est universellement consistant.

[42] Modèle basique = modèle à faible stabilité et précision, sujet au sur-apprentissage (KNN, CART etc.)

3.4.4 Le Boosting

Le boosting est une méthode d'amélioration des modèles basiques où chaque nouvel arbre apprend du précédent. C'est une technique ensembliste qui agrège des régresseurs faibles en corrigeant séquentiellement leurs poids : chaque régresseur est ajusté en fonction de l'erreur du précédent. Ce faisant, chaque nouveau modèle se concentre sur l'observation la plus difficile à prédire. L'objectif de cette méthode étant principalement la baisse du biais, le modèle basique considéré sera généralement de variance faible et de biais élevé : lors de l'utilisation de l'arbre de décision, la profondeur adéquate sera généralement faible. L'utilisation de modèles basiques de faible variance et de biais élevé autorise également des temps de calculs plus raisonnables (quelque degrés de liberté lors du paramétrage) compte tenu de l'impossibilité de paralléliser les calculs comme dans le Bagging.

Ce type d'apprentissage séquentiel détermine l'information que le nouveau modèle va utiliser en provenance du précédent et la manière dont les modèles basiques sont agrégés. Ce point est décrit à travers deux principaux algorithmes de Boosting que sont l'AdaBoost[43] et le gradient Boosting.

Le Boosting adaptatif définit le modèle ensembliste par la somme des modèles basiques :

$$S_L(.) = \sum_{l=1}^{L} C_l \times w_l(.)$$

avec C_l les coefficients et $w_l(.)$ les modèles basiques.

Trouver les coefficients et modèles basiques optimaux sous cette forme est impossible : l'algorithme utilise donc une approche itérative d'optimisation, comme décrit ci-dessous (algorithme originel AdaBoost).

On note δ la fonction de discrimination, m les modèles de prédiction et $z = \{(x_1, y_i), \ldots, (x_n, y_n)\}$ un échantillon :

- Initialisation des poids : $w = \left\{w_i = \frac{1}{n}, i \in [\![1, n]\!]\right\}$
- Pour m allant de 1 à M :
  - Estimation de δ_n sur l'échantillon pondéré par w,
  - Calcul du taux d'erreur apparent du modèle : $\hat{\xi}_p = \frac{\sum_{i=1}^{m} w_i \mathbb{1}_{\delta_m(x_1) \neq y_i}}{\sum_{i=1}^{m} w_i}$,
  - Calcul des logit : $C_m = -\ln \frac{1-\hat{\xi}_p}{\hat{\xi}_p}$,
  - Mise à jour des pondérations : $w_i \leftarrow w_i \, e^{C_m \, \mathbb{1}_{\delta_m(x_1) \neq y_i}}$,
  - Prédiction du résultat : $\Phi_M(x) = sign\big(C_m \ \delta_m(x)\big) = \pm 1$

S'agissant d'un problème de classification à deux classes, la prédiction $\hat{Y} \in \{-1, 1\}$.

[43] Pour Adaptive Boosting ou encore Boosting Adaptatif

3.4.5 La forêt aléatoire

La forêt aléatoire combine les notions explicitées précédemment : il s'agit en effet d'une méthode de Bagging dans laquelle les arbres de décisions CART sont entraînés sur des échantillons Bootstrap puis combinés afin d'obtenir des résultats à plus faible variance. L'échantillonnage n'est pas effectué uniquement sur les observations, mais également sur les variables explicatives. Ainsi, les arbres ne sont pas entraînés sur les mêmes informations ce qui réduit la corrélation des résultats issus des modèles basiques. Ce type d'échantillonnage présente également l'avantage de rendre le processus d'apprentissage plus robuste aux données manquantes, une observation incomplète sur une variable pourra toujours être exploitée sur une autre variable.

Chaque arbre est construit de manière identique : si l'échantillon est composé de N observations, alors N observations sont échantillonnées avec remise. Au sein de M variables, un nombre constant m de variables est sélectionnée à chaque nœud (sachant que $m < M$) afin que la meilleure séparation du nœud soit effectuée sur ces variables. C'est cette dernière étape qui différencie la forêt aléatoire du Bagging : le premier ne sélectionne pas toutes les variables lors de la séparation du nœud en deux feuilles (contrairement au Bagging).

Le taux d'erreur de la forêt aléatoire dépend de la corrélation entre les arbres de décision et la force de chaque arbre (qui est le taux d'erreur individuel). Ainsi la réduction de m fera baisser la corrélation mais aussi la force des arbres.

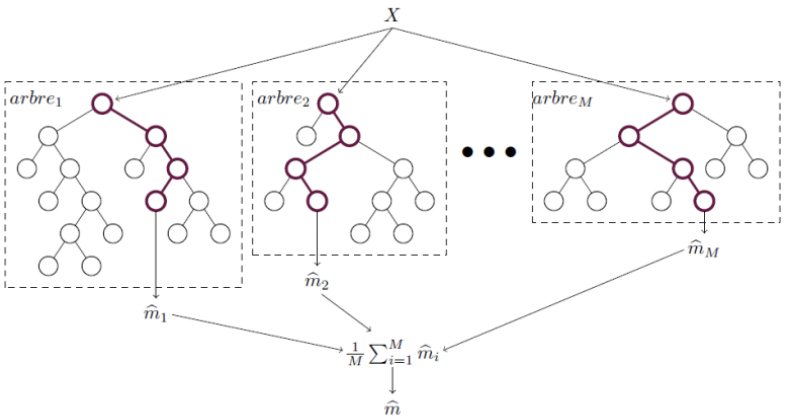
L'algorithme de construction est le suivant :

- Pour chaque arbre M aléatoire :
    - q observations sont sélectionnées avec remise parmi les n observations initiales (Bootstrap)
    - A chaque nœud :
        - un seul sous ensemble de variables explicatives est utilisé,
        - la séparation s'effectue lorsque le critère CART de coupure est maximisé,
- la boucle itère sur chaque arbre jusqu'à ce que le critère d'arrêt soit atteint,

La Figure 49 illustre l'agrégation des estimateurs, l'estimateur agrégé final étant :

$$\hat{m}(x) = \frac{1}{M}\sum_{j=1}^{M} \hat{m}_j$$

Figure 49: agrégation des estimateurs CART, extrait de (C, Bilore, 2016)



3.4.5.1 *L'erreur out of bag*

L'un des avantages de la forêt aléatoire est qu'elle permet d'effectuer une sélection de variables et d'en mesurer l'importance. L'indépendance des arbres permet de mesurer l'erreur par validation croisée, tirant ainsi parti de sa composante Bootstrap. Ce faisant, il devient possible de classer les variables explicatives de manière ordinale en fonction de l'information marginale apportée par chacune. Dans le cas de la régression c'est la variation marginale de l'erreur out of bag, suite à la permutation des observations de la variable, qui déterminera son importance dans la prédiction finale.

Soit F_i^* la sous forêt construite avec un échantillon excluant l'observation (x_i, y_i) se trouvant dans la base d'apprentissage. L'erreur out of bag est alors l'erreur de prédiction moyenne des arbres composant F_i^* sur l'observation (x_i, y_i).

Seule une fraction de la base de données est utilisée pour la mesure de cette erreur, ce qui conduit à une réduction de l'effet d'agrégation du Bagging. La validation croisée est donc souvent préférable, néanmoins il arrive que le nombre d'observations soit insuffisant : dans ce dernier cas, l'erreur out of bag offre une alternative et permet de conserver l'entièreté de la base pour les besoins de l'apprentissage – même si le calcul d'un score est calculé différemment et n'est pas nécessairement comparable à l'erreur out of bag.

Il est facilement démontrable qu'environ 36% de la base d'apprentissage est nécessaire pour une erreur out of bag fiable. Pour N lignes au sein de la base, la probabilité de ne pas choisir l'une d'entre elles est $\frac{N-1}{N}$.

Or en échantillonnant avec remise, cette probabilité devient $\left(\frac{N-1}{N}\right)^N$ ce qui, à la limite, devient:

$$\lim_{N\to\infty}\left(1-\frac{1}{N}\right)^N = e^{-1} \simeq 0.368$$

soit 36% de la base d'apprentissage disponible pour un calcul de l'erreur out of bag fiable.

3.4.6 Avantages et inconvénients

L'intérêt de la forêt aléatoire réside dans sa précision. Elle fonctionne efficacement sur de larges bases de données (observations ou nombre de variables), peut traiter des problèmes de classification et de régression et permet une gestion efficace des données manquantes. Par ailleurs, le biais de son estimateur réduit au fur et à mesure que la forêt grandit.

Seuls quelques paramètres – constants tout au long du processus algorithmique- sont nécessaires à son optimisation: le nombre d'estimateurs (c'est-à-dire le nombre d'arbres de décision qui composent la forêt), le nombre de variables utilisées pour la séparation des nœuds, l'utilisation ou non de l'échantillon Bootstrap et quelques paramètres propres aux arbres CART.

A l'inverse, la forêt aléatoire ne sera pas pertinente si les variables utiles à la prédiction sont en nombre restreint. Son processus de décision sera plus compliqué à visualiser qu'un simple arbre CART sans que le risque de sur-apprentissage ne disparaisse pour autant. Enfin, son déploiement est plus compliqué que les méthodes agrégées vu au chapitre précédent et son temps de calcul peut parfois être long.

3.4.7 Application

Pour rappel, le triangle inférieur est constitué des années de survenance 2010-2014 projetées sur les années de développement 2015-2018 (échelle en mois). Les algorithmes sont lancés un à un sur le triangle supérieur puis les résultats sont présentés par année de développement pour 2 variables cibles : les sinistres payés et le loss ratio. La charge totale n'est pas présentée ici car elle est identique à la charge des payés (développement quasi terminé).

Avant de présenter les résultats, notons que les principaux hyperparamètres de la forêt aléatoire sont la fonction de perte, le taux d'apprentissage de chaque arbre (plus il sera faible, plus il faudra d'arbres), le nombre d'étapes à parcourir, la taille de l'échantillon, la proportion de variables disponibles à chaque étape pour la découpe, le Bootstrap (utilisation oui/non) et les paramètres applicables à chaque arbre. Pour les besoins de cette étude, seuls 3 hyperparamètres ont été optimisés et font l'objet d'une brève description ci-dessous.

- "max_depth"

Il décrit le chemin le plus long entre le nœud racine et la dernière feuille. Il permet ainsi de limiter la profondeur de la forêt aléatoire et par là même le risque de sur apprentissage.

- n_estimators"

Cet hyperparamètre décrit le nombre d'arbres que l'algorithme doit considérer. Plus ce nombre est grand, plus il est précis mais complexe et lent computationnellement.

- "min_sample_split"

C'est un critère de coupure. Il détermine le nombre minimal d'observations nécessaire au sein de chaque nœud avant d'autoriser l'arbre de décision à procéder à une coupure.

RESULTATS POUR LA VARIABLE REPONSE DES SINISTRES PAYES

Tableau 14: données réelles échantillonnées

	12	24	36	48	60
2010	1,291,265	5,425,344	3,158,739	3,483,945	1,651,422
2011	1,965,273	2,823,327	3,563,193	930,431	785,060
2012	433,741	3,575,189	2,160,406	-9,214	283,604
2013	676,290	5,475,618	2,912,984	3,830,183	1,216,597
2014	1,040,862	7,824,497	6,363,272	678,412	519,012

Tableau 15: données prédites par Random Forest

	12	24	36	48	60
2010	1,291,265	5,425,344	3,158,739	3,483,945	1,651,422
2011	1,965,273	2,823,327	3,563,193	930,431	1,687,041
2012	433,741	3,575,189	2,160,406	1,571,090	290,389
2013	676,290	5,475,618	1,595,079	372,636	372,694
2014	1,040,862	766,017	209,584	209,689	209,742

Le Tableau 14 représente les valeurs réelles échantillonnées : il s'agit du cumul des sinistres dont l'année de survenance est comprise entre 2010 et 2014. Chaque année de survenance se développe sur 5 années, les colonnes faisant le compte des mois de développement. Le Tableau 15 consolide les valeurs du triangle supérieur, identiques aux données réelles, avec le triangle inférieur, somme de l'ensemble des transactions futures prédites.

Le Tableau 16 ci-dessous récapitule quant à lui le score de proximité et ses composantes par année de développement, la première étant 2015.

Tableau 16: score de proximité du Random Forest

Random Forest evolution

	diagonals_MAE	diagonals_percentage	diagonals_RMSE	diagonals_ER	diagonals_C_M	diagonals_prox
2015	2714667.813	4350.296	3703712.877	1.049	-0.933	17.106
2016	3206006.695	63.123	4075232.181	11.007	-0.410	26.637
2017	656313.074	69.229	682595.676	2.254	1.000	14.688
2018	309270.294	59.588	309270.294	1.475	1.000	13.116

Plus précisément, il représente la performance de la Forêt Aléatoire sur chaque année de développement, toutes années de survenance cumulées. Une boucle a été crée afin d'effectuer les calculs décrits à la section « 3.2.3 L'évaluation des modèles » sur base du détail de chaque transaction, afin d'établir le score de proximité par diagonale[44]. Ce travail n'est généralement pas nécessaire car un actuaire raisonne à l'ultime (par année de survenance, toutes diagonales cumulées), mais il apporte des éclairages intéressants dans le cadre de cette étude.

La Forêt Aléatoire est l'algorithme qui performe le moins de tous : les scores de proximité sont élevés et l'écart relatif moyen est mauvais sur toutes les années. L'absence d'hyper paramétrage poussé est assurément une des raisons à cette sous performance.

Comme expliqué à la section « 3.2.3 L'évaluation des modèles », les deux dernières années de développement ne peuvent avoir un score de corrélation et la valeur 1 est donc arbitrairement appliquée.

Une boucle a été crée afin de permettre au lecteur de vérifier le nombre de sinistres fermés par le classificateur à chaque étape de calcul. Regardons l'année 2015 à titre d'illustration. Cette analyse ne sera par reproduite pour chaque algorithme par souci de synthèse.

Tableau 17: Prédiction de fermeture des dossiers Q1 2015

```
Predicting for model :  random_forest
Claim Values is true, cased closed :  42
Current year prediction is :  2015-03-31 00:00:00
Applying prediction on :  441 data.
Length of resulting data :  3022
Average Accident Year of predicted data :  2012.1700680272108
```

Tableau 18: Prédiction de fermeture des dossiers Q2 2015

```
Predicting for model :  random_forest
Claim Values is true, cased closed :  21
Current year prediction is :  2015-06-30 00:00:00
Applying prediction on :  399 data.
Length of resulting data :  3421
Average Accident Year of predicted data :  2012.265664160401
```

Tableau 19: Prédiction de fermeture des dossiers Q3 2015

```
Predicting for model :  random_forest
Current year prediction is :  2015-09-30 00:00:00
Applying prediction on :  378 data.
Length of resulting data :  3799
Average Accident Year of predicted data :  2012.2380952380952
--------------------
```

Tableau 20: Prédiction de fermeture des dossiers Q4 2015

```
Predicting for model :  random_forest
Current year prediction is :  2015-12-30 00:00:00
Applying prediction on :  378 data.
Length of resulting data :  4177
Average Accident Year of predicted data :  2012.2380952380952
--------------------
```

[44] Une diagonale est la forme imagée d'une année de développement (même signification)

Les tableaux 13 à 15 montrent que l'algorithme ferme 42 sinistres au premier trimestre 2015, 21 au second puis aucun les 3ème et 4ème trimestres.

VARIABLE REPONSE : LOSS RATIO

Tableau 22: loss ratio sur base des données réelles échantillonnées

	12	24	36	48	60
2010	33.82	18.71	12.61	0.84	1.11
2011	16.07	27.79	11.07	0.41	0.25
2012	37.34	45.42	4.39	0.18	0.27
2013	66.17	41.65	5.46	1.49	1.28
2014	59.36	48.88	7.82	1.84	0.89

Tableau 21: loss ratios prédits par Random Forest

	12	24	36	48	60
2010	33.82	18.71	12.61	0.84	1.11
2011	16.07	27.79	11.07	0.41	4.16
2012	37.34	45.42	4.39	4.98	0.72
2013	66.17	41.65	9.75	1.03	1.03
2014	59.36	5.73	0.64	0.64	0.64

Tableau 23: score de proximité du Random Forest sur les loss ratios

Random Forest evolution

	diagonals_MAE	diagonals_percentage	diagonals_RMSE	diagonals_ER	diagonals_C_M	diagonals_prox
2015	14.036	1100.027	21.899	1.224	-0.010	4.321
2016	2.693	95.614	4.158	2.990	-0.537	4.952
2017	0.725	42.273	0.867	0.864	-1.000	1.722
2018	0.242	27.288	0.242	0.375	1.000	-2.045

Les résultats de prédiction sur le loss ratio ne sont pas plus satisfaisants que sur les sinistres payés: il est néanmoins à noter que le score de proximité s'améliore tout au long du cycle de développement des sinistres, jusqu'à 37.5% d'écart relatif en 2018.

3.5 Le Gradient Boosting

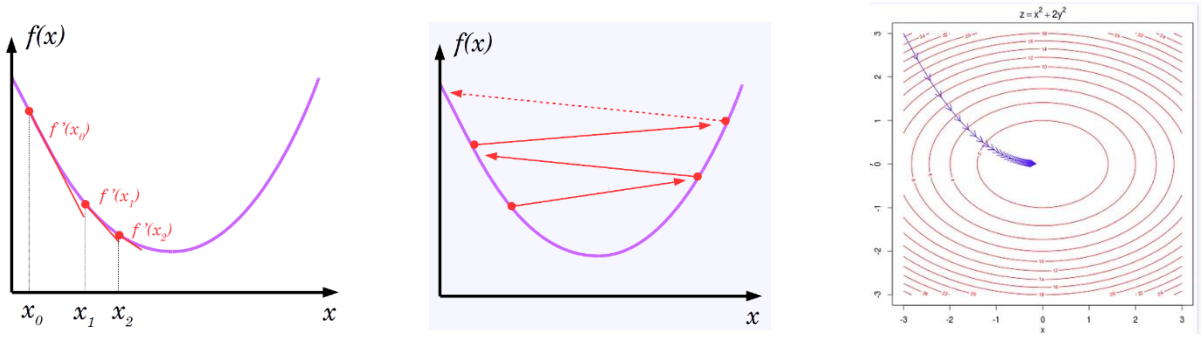
Le Gradient Boosting combine la descente de gradient avec le Boosting. Il généralise et étend la notion introduite par AdaBoost en entraînant des modèles basiques de façon séquentielle, graduelle et additive. Sa principale différence avec le Boosting Adaptatif tient à la mesure qu'il fait de l'erreur faite par le modèle basique : alors qu'AdaBoost utilise le poids de chaque observation, le Gradient Boosting utilise le gradient de la fonction de coût. La fonction de coût mesure la différence, ou encore l'erreur, entre les valeurs prédites et réelles. Il s'agit d'une mesure de la justesse des estimateurs du modèle, elle variera donc en fonction de la nature du processus que le modèle cherche à optimiser.

3.5.1 La descente de gradient

La descente de gradient est un algorithme d'optimisation différentiable utilisé dans de nombreux modèles d'apprentissage automatique comme le réseau de neurones ou la forêt aléatoire.

Pour minimiser cette fonction de coût, la descente de gradient a besoin de quelques paramètres qui vont permettre de déterminer le calcul itératif de dérivée partielle et l'obtention d'un minimum local, au premier rang desquels la fonction de coût elle-même et un taux d'apprentissage. La Figure 50 en donne une illustration, avec x_0 est le point de départ de l'algorithme, α est le taux d'apprentissage et $x_{n+1} = x_n - \alpha \frac{df}{dx}(x_n)$ est la descente de gradient. Le troisième graph schématise la descente de gradient en grande dimension.

Figure 50: illustrations de la descente de gradient, extrait de (Lucidar)



3.5.2 Le Gradient Boosting

Contrairement aux forêts aléatoires qui font une moyenne aléatoire et indépendante des résultats donnés par chaque arbre de décision, le Gradient Boosting combine l'utilisation d'une fonction de perte quadratique L, qu'il essaie de minimiser comme suit avec la méthode de descente de gradient :

$$f(x) = \text{argmin}\frac{1}{n}\sum_{i=1}^{n} L\big(y_i, \Phi_M(x_i)\big)$$

L'avantage de cette méthode réside dans le choix qu'il offre quant à la fonction de coût. Elle permet également de s'adapter aux spécificités du problème de prédiction, évalue elle-même l'erreur et mesure naturellement l'importance des variables. A l'inverse, ses inconvénients sont ceux des boîtes noires : ses choix ne sont pas explicites tandis que ses paramètres sont complexes à ajuster.

3.5.3 Application

Le triangle inférieur utilisé pour l'application est identique à celui présenté à la section « Le Random Forest ».

Les principaux hyperparamètres du XGBoost sont la fonction de coût, la pondération de la contribution de chaque arbre au suivant (appelé taux d'apprentissage), le nombre d'estimateurs, de variables disponibles à chaque séparation de nœud, l'utilisation du Bootstrap et les paramètres des arbres de décision.

Pour les besoins de cette étude, seuls 3 hyperparamètres spécifique au Boosting [45] ont été optimisés et font l'objet d'une brève description ci-dessous.

- « Learning rate »

C'est le taux d'apprentissage décrit plus haut. Il contrôle l'amplitude de la mise à jour en provenance de chaque arbre. Plus il sera petit, plus il généralisera bien mais sera lent à entraîner.

- « min_child_weight »

Il définit la somme minimale des poids de toutes les observations se trouvant au sein d'une feuille enfant. Un hyperparamètre similaire existe pour la forêt aléatoire, mais ici il s'agit du poids total minimum des observations (et non pas du simple nombre minimum d'observations avant coupure). Son utilité est en revanche identique : contrôler le sur apprentissage.

- « gamma »

Un nœud n'est coupé que si le suivant permet une réduction de la fonction de coût : cet hyperparamètre permet de fixer la réduction minimale attendue. Plus sa valeur est élevée, plus l'algorithme est prudent.

[45] Max_depth et min_depth ainsi que n_estimators sont des hyperparamètres que l'on retrouve pour la forêt aléatoire

RESULTATS POUR LA VARIABLE REPONSE : SINISTRES PAYES

Tableau 25: données réelles échantillonnées

	12	24	36	48	60
2010	1,291,265	5,425,344	3,158,739	3,483,945	1,651,422
2011	1,965,273	2,823,327	3,563,193	930,431	785,060
2012	433,741	3,575,189	2,160,406	-9,214	283,604
2013	676,290	5,475,618	2,912,984	3,830,183	1,216,597
2014	1,040,862	7,824,497	6,363,272	678,412	519,012

Tableau 24: données prédites par le Gradient Boosting

	12	24	36	48	60
2010	1,291,265	5,425,344	3,158,739	3,483,945	1,651,422
2011	1,965,273	2,823,327	3,563,193	930,431	813,595
2012	433,741	3,575,189	2,160,406	1,230,344	242,446
2013	676,290	5,475,618	1,920,113	909,561	711,349
2014	1,040,862	1,497,086	1,114,077	996,817	887,341

Même présentation des résultats que pour la forêt aléatoire, avec la confrontation des résultats par année de survenance ci-dessous :

Tableau 26: score de proximité du Gradient Boosting

XGBoost evolution

	diagonals_MAE	diagonals_percentage	diagonals_RMSE	diagonals_ER	diagonals_C_M	diagonals_prox
2015	2147094.046	3392.839	3261872.573	1.108	0.451	15.655
2016	2736991.478	57.752	3468226.151	3.623	0.980	17.703
2017	411826.434	44.232	422289.822	0.109	-1.000	14.063
2018	368328.756	70.967	368328.756	-0.415	1.000	12.232

Le XGBoost fait preuve de volatilité dans la précision de ses prédictions : l'erreur relative est remarquablement faible sur l'année de développement 2017 (10% d'écart), mais se détériore en 2018 (-41%). Les deux premières années sont très mal prédites (+362% en 2016), les scores de proximité s'améliorant les années suivantes.

L'algorithme clôture par ailleurs d'avantage de sinistres que la forêt aléatoire : 198 contre 63, ce qui permet une moindre sur estimation des sinistres.

RESULTATS POUR LA VARIABLE REPONSE : LOSS RATIO

Tableau 28: loss ratio sur base des données réelles échantillonnées

	12	24	36	48	60
2010	33.82	18.71	12.61	0.84	1.11
2011	16.07	27.79	11.07	0.41	0.25
2012	37.34	45.42	4.39	0.18	0.27
2013	66.17	41.65	5.46	1.49	1.28
2014	59.36	48.88	7.82	1.84	0.89

Tableau 27: loss ratios prédits par XGBoost

	12	24	36	48	60
2010	33.82	18.71	12.61	0.84	1.11
2011	16.07	27.79	11.07	0.41	12.36
2012	37.34	45.42	4.39	12.71	11.50
2013	66.17	41.65	21.00	13.55	13.76
2014	59.36	11.99	7.72	7.51	8.36

Tableau 29: score de proximité du XGBoost sur le loss ratio

XGBoost evolution

	diagonals_MAE	diagonals_percentage	diagonals_RMSE	diagonals_ER	diagonals_C_M	diagonals_prox
2015	19.267	3039.806	21.828	-0.057	-0.290	3.430
2016	7.794	1638.577	9.510	-0.708	-0.875	3.835
2017	9.071	639.498	9.690	-0.853	-1.000	4.124
2018	7.473	843.973	7.473	-0.894	1.000	1.905

Les prédictions sur le loss ratio sur estiment considérablement la charge dès 2015. La RMSE s'améliore tout au long de la prédiction mais c'est insuffisant pour que le score de proximité s'améliore de manière continue sur l'ensemble de la période.

3.6 Les réseaux artificiels de neurones

Cet algorithme, également appelé Perceptron Multicouche (Pitts), utilise des méthodes de résolution numérique empruntée à la neurobiologie, introduisant ainsi les concepts d'apprentissage adaptatif et de mémorisation. Il s'agit d'un modèle statistique non-linéaire.

L'illustration ci dessous décrit le fonctionnement d'un neurone biologique autour de trois régions : les dendrites qui captent les signaux envoyés au neurone, l'axone qui se connecte aux dendrites des autres neurones et les synapses qui font la jonction entre deux neurones.

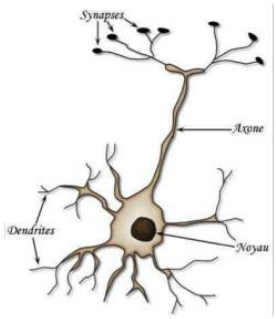
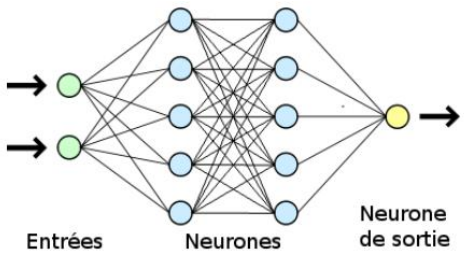


FIGURE 7.1 – *Un neurone biologique*

Le réseau artificiel de neurones peut être représenté comme un ensemble d'unités appelés neurones ou nœuds, connectées par des synapses. En d'autres termes, il peut être vu comme une chaîne de plusieurs GLM successifs.

Les réseaux de neurones constituent un vaste domaine au sein de la recherche scientifique, cette section se limitera aux deux algorithmes présentant des possibilités applicatives intéressantes pour l'apprentissage supervisé en assurance : le perceptron et le réseau multicouches.

Figure 51: représentation simplifiée du réseau de neurones

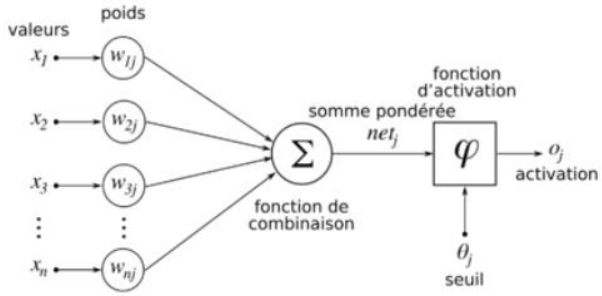


3.6.1 Le Perceptron ou neurone formel

Un neurone artificiel est une fonction de transition entre deux états. Il reçoit en entrée un ensemble d'information et génère une valeur unique en sortie. Son application mathématique se définit par un espace de départ X et d'arrivée Y, d'un ensemble de poids noté W et d'une fonction d'activation ϕ.

Schéma d'un neurone artificiel en Figure 52 :

Figure 52: fonction activation phi (Wikipedia)



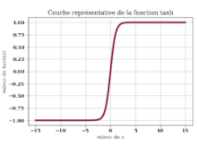
L'espace de départ X_i est l'ensemble auquel appartient l'information d'entrée. On considère qu'il existe $n \in N^*$ tel que X soit isomorphe à un sous-espace de $\mathbb{R}^n$.

L'espace d'arrivée Y correspond à la décision finale et peut être vue comme une transformation par une fonction pondérée des X_i. Lorsque Y est infini, le problème est de type regression[46].

L'ensemble de poids $W = (\omega_i)_{0 \leq i \leq n} \in \mathbb{R}^{n+1}$ est la pondération de chaque observation X_i en entrée dont la prédiction va dépendre. Ce sont les poids du neurone.

La fonction d'activation ϕ: $\mathbb{R} \to \mathbb{R}$ génère la réponse et doit vérifier les hypothèses de monotonie croissante, différentiable presque partout et de valeurs finies dans un intervalle fermé ou un ensemble fini. En dessous d'un certain seuil le neurone sera inactif, proche du seuil il sera en phase de transition et au dessus le neurone sera en activité. Il existe un grand nombre de fonctions d'activation et les 3 plus courantes sont représentées dans le Tableau 30 ci-dessous[47].

Tableau 30: trois fonctions d'activation classiques et leur représentation graphique

Fonction	sigmoïde, $\alpha \in \mathbb{R}_+$	ReLU	tanh
Domaine de valeurs	[0,1]	$\mathbb{R}_+$	[-1,1]
Expression	$z \mapsto \frac{1}{1+e^{-\alpha z}}$	$z \mapsto max(0,z)$	$z \mapsto \frac{e^z - e^{-z}}{e^z + e^{-z}}$
			

A noter que (B. Harej & al, 2017) retiennent les fonctions d'activation sigmoïde et tangente hyperbolique pour le réseau de neurones dont ce mémoire s'inspire: la sigmoïde donne une réponse dans l'espace $[0,1]$ et sa dérivée est facile à calculer :

$$g'(x) = \alpha \frac{e^{-\alpha x}}{1 + e^{-\alpha x}} = \alpha . g(x) . \big(1 - g(x)\big)$$

[46] Lorsqu'il est fini, il s'agit d'un problème de classification

[47] A noter que (B. Harej & al, 2017) ajoutent à cette liste la fonction Gaussienne $g(x) = \frac{1}{\sqrt{2\Pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

La fonction tangente hyperbolique (tanh) permet quant à elle une réponse dans l'espace $[-1, +1]$. Toutes deux sont adaptées au principe de descente de gradient sur lequel la retro-propagation est basée car l'hypothèse de monotonie permet l'obtention d'une direction fiable. Le neurone opère ainsi une somme pondérée des informations :

$$y = \phi\left(\sum_{i=0}^{n} \omega_i \times X_i\right)$$

Et le neurone renvoie le résultat suivant :

$$y = \phi(\omega^t X - \omega_0)$$

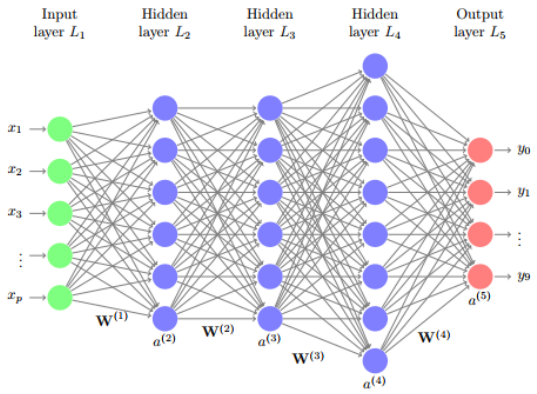
3.6.2 Le réseau de neurones

Le perceptron multicouche est une extension du simple neurone artificiel : il se compose de plusieurs couches successives dont les fonctions d'activations sont différentes.

Un réseau de neurones à propagation avant[48], appelé également « perceptron multicouches », est un ensemble de connexions entre plusieurs neurones formels organisés en couches successives. Les neurones au sein d'une même couche ne sont pas connectés.

Le lien entre deux neurones s'appelle le poids synaptique. Ce poids caractérise l'information partagée entre les neurones. Au niveau de l'espace d'entrée X, chaque nœud est connecté aux poids synaptiques. Ces poids sont initialisés aléatoirement, s'ajustent grâce au processus d'apprentissage puis se lient aux nœuds de la couche suivante via la fonction d'activation. La Figure 53 en illustre le fonctionnement.

Figure 53: réseau de neurones à 3 couches cachées (T.Hastie, 2017)



[48] Ou « feed-forward »

Plus formellement, on considère l'espace de variables $\{Y, X_1, \ldots, X_p\}$ avec p variables explicatives, Y la variable cible et θ le paramètre de biais scalaire[49]. Le perceptron multicouches sera alors défini par la fonction $f_{(w,\theta)}$ tels que:

$$f_{(\omega,\theta)} : \begin{cases} X \to y \\ x \to \phi({}^t\omega x + \theta) \end{cases}$$

L'architecture du réseau dépendra de sa connectivité. Elle sera récurrente si elle comporte des boucles, c'est-à-dire : $\exists\, i\ tel\ que\ \omega_{i,i} \neq 0$. On parlera de réseau profond s'il comporte plusieurs couches cachées. L'actuaire doit donc calibrer plusieurs paramètres en fonction de la nature du problème à résoudre : l'architecture (nombre de neurones, couches et connectivité), la matrice de poids W et les fonctions d'entrée et d'activation. L'étape d'apprentissage est consacrée à cette calibration.

L'algorithme d'apprentissage a pour objectif de minimiser la fonction de coût, qui représente la différence entre les valeurs observées et les valeurs prédites par le modèle. Dans le cas de la régression, objet de ce mémoire, elle se définie par l'erreur moyenne au carré : $C(\omega) = \sum_\alpha (y_\alpha - \hat{y}_\alpha)^2$ où ω est le vecteur des poids, $\hat{y}_\alpha$ la sortie du $\alpha^{ème}$ neurones et y_α les valeurs réellement observées. Dans l'espace de paramètres de la fonction de coût, l'algorithme recherche la direction descendante qui minimise la fonction et répète cette étape jusqu'à ce qu'un critère d'arrêt soit déclenché. La fonction de coût peut être l'erreur de corrélation croisée : $C(\omega) = -\sum_{i=1}^{n} (y_\alpha \ln \hat{y}_\alpha + (1 - y_\alpha) \ln(1 - \hat{y}_\alpha))$ dont l'avantage est de converger plus rapidement que l'erreur quadratique.

Le processus algorithmique de rétropropagation se base quant à lui sur a) une étape de propagation dans laquelle chaque neurone est activé en fonction des neurones de la couche précédente puis b) une étape de rétropropagation dans laquelle un calcul récursif du gradient partiel est effectué et dont la somme permet d'obtenir la fonction globale de coût.

3.6.3 Forces et faiblesses

Les limites du perceptron multicouches tiennent à sa complexité algorithmique : la rétropropagation est connue pour sa lenteur de convergence, dont le temps d'apprentissage augmente exponentiellement en fonction du nombre de connexions neuronaux[50].

La seconde limite est le sur-apprentissage, qui augmente avec le nombre de passages sur les données. Le modèle retient alors les caractéristiques exceptionnels de la base de données et ne peut plus généraliser. La validation croisée ainsi que l'agrégation de modèles sont deux outils permettant de limiter ce risque.

[49] Notion similaire à l'ordonnée à l'origine dans le cas d'une régression linéaire

[50] Hinton a démontré empiriquement un temps d'apprentissage $\mathrm{O}(N^3)$ avec N le nombre de connexions neuronaux

3.6.4 Application

Le triangle inférieur utilisé pour l'application est identique à celui présenté aux sections « Le Random Forest » et « Le Gradient Boosting ».

Les principaux paramètres disponibles pour optimiser le réseau de neurones sont les suivants : le nombre de couches cachées et de neurones composant chaque couche, la fonction d'activation (qui peut être la fonction identité, logistique, hyperbolique tangente ou ReLu), le type d'optimisation (Newtonnien, descente de gradient ou autre), le taux d'apprentissage et enfin le nombre maximum d'itérations pour l'optimisation. Pour les besoins de cette étude, seuls 3 hyperparamètres ont été optimisés.

- "activation"

La fonction d'activation définit la manière dont la somme pondérée des données en entrée d'un nœud est transformé en sortie. Elle est usuellement différentiable[51].

- "Dropout"

Le terme se réfère à l'abandon aléatoire de neurones visibles et invisibles pendant la phase d'entraînement. Ainsi un certain nombre de neurones ne sont pas considéré par l'algorithme lors de la phase forward ou backward. L'objectif est d'obtenir un réseau réduit, limitant ainsi le risque de sur apprentissage.

- "learning rate"

Contrôle le niveau de modification du modèle en réponse à l'erreur estimée à chaque mise à jour du réseau. Il s'agit de la taille du pas vers lequel la fonction de coût, à chaque itération, se dirige vers son minimum.

RESULTATS POUR LA VARIABLE REPONSE : SINISTRES PAYES

Tableau 32: données réelles échantillonnées

	12	24	36	48	60
2010	1,291,265	5,425,344	3,158,739	3,483,945	1,651,422
2011	1,965,273	2,823,327	3,563,193	930,431	785,060
2012	433,741	3,575,189	2,160,406	-9,214	283,604
2013	676,290	5,475,618	2,912,984	3,830,183	1,216,597
2014	1,040,862	7,824,497	6,363,272	678,412	519,012

Tableau 31: données prédites par le Multi Layer Perceptron

	12	24	36	48	60
2010	1,291,265	5,425,344	3,158,739	3,483,945	1,651,422
2011	1,965,273	2,823,327	3,563,193	930,431	1,149,592
2012	433,741	3,575,189	2,160,406	1,308,817	719,375
2013	676,290	5,475,618	1,756,417	1,032,514	1,032,514
2014	1,040,862	877,240	533,184	533,184	533,184

[51] La première dérivée peut être calculée pour une donnée d'entrée donnée

Tableau 33: score de proximité du MLP

MultiLayered Perceptron evolution

	diagonals_MAE	diagonals_percentage	diagonals_RMSE	diagonals_ER	diagonals_C_M	diagonals_prox
2015	2446596.929	3619.838	3587203.796	1.261	-0.459	16.813
2016	3021176.048	106.106	3741959.300	3.585	-0.278	18.998
2017	164656.037	18.269	165798.139	0.210	1.000	11.229
2018	14171.449	2.730	14171.449	-0.027	1.000	8.586

Le MLP est l'algorithme qui s'améliore franchement au fur et à mesure des années de développement, tant du point de vue de l'erreur relative que du score de proximité. C'est aussi l'algorithme qui ferme le plus de sinistres : 258 sinistres, significativement plus que les deux algorithmes vus précédemment. Ce dernier point contribue indéniablement à l'amélioration de la qualité des prédictions au cours du temps.

RESULTATS POUR LA VARIABLE REPONSE : LOSS RATIO

Tableau 35: loss ratio sur base des données réelles échantillonnées

	12	24	36	48	60
2010	33.82	18.71	12.61	0.84	1.11
2011	16.07	27.79	11.07	0.41	0.25
2012	37.34	45.42	4.39	0.18	0.27
2013	66.17	41.65	5.46	1.49	1.28
2014	59.36	48.88	7.82	1.84	0.89

Tableau 34: loss ratio prédits par le MLP

	12	24	36	48	60
2010	33.82	18.71	12.61	0.84	1.11
2011	16.07	27.79	11.07	0.41	10.73
2012	37.34	45.42	4.39	11.58	10.41
2013	66.17	41.65	21.73	15.48	15.46
2014	59.36	11.49	8.60	8.64	8.60

Tableau 36: score de proximité du MLP

MultiLayered Perceptron evolution

	diagonals_MAE	diagonals_percentage	diagonals_RMSE	diagonals_ER	diagonals_C_M	diagonals_prox
2015	18.883	2723.250	21.808	-0.013	-0.203	3.299
2016	8.300	1551.916	9.982	-0.722	-0.589	3.612
2017	10.484	736.188	11.115	-0.870	-1.000	4.278
2018	7.714	871.219	7.714	-0.897	1.000	1.940

Même constat pour les loss ratios, même si la sur estimation de la charge est ici plus importante que sur les sinistres payés. L'écart relatif se détériore lors de la seconde année mais la RMSE s'améliore, atteignant ainsi un score de proximité excellent sur la dernière année de développement.

3.7 Comparaison des résultats

Après avoir vu chaque algorithme un à un, un récapitulatif cumulant toutes les années de développement est proposé ci-dessous pour chaque variable réponse (sinistres payés et loss ratio).

Tableau 37: prédictions des sinistres payés

	MAE	Percentage	RMSE	ER	C_M	Prox
Genins	1332157.135	1042.125	1772690.422	0.171	0.811	13.748
Linear Regression	1782986.805	1228.840	2779592.834	0.893	0.249	15.483
XGBoost	1799133.224	891.448	2813136.187	1.364	0.478	15.736
MLP	1919339.938	936.736	3058334.210	1.575	-0.082	16.591
Random Forest	2209858.778	1135.559	3251461.673	2.350	-0.145	17.490

Tableau 38: prédictions des loss ratio

	MAE	Percentage	RMSE	ER	C_M	Prox
Genins	2.083	48.321	3.548	-0.193	0.980	0.479
Linear Regression	9.326	897.250	13.959	-0.078	0.198	2.517
XGBoost	12.607	1540.464	15.559	-0.432	0.010	3.167
MLP	12.912	1470.643	15.837	-0.443	-0.031	3.236
Random Forest	6.592	316.301	14.042	1.329	0.343	3.628

Tableau 37 et Tableau 38 classent les algorithmes vus précédemment par score de proximité. Deux algorithmes sont ajoutés à des fins de comparaison :

1. le Chain Ladder classique tel que décrit dans la section « Le Chain Ladder » (appelé « Genins » dans les tableaux)
2. la régression linéaire, un algorithme simple qui nous permettra de mesurer l'intelligence des autres modèles,

Premier constat : le Chain Ladder continue de régner en maître dans la prédiction des sinistres. Il combine à la fois, et sur les deux variables réponses étudiées, l'écart relatif le plus faible (17-19%) et le meilleur score de proximité. Second constat : le classement des algorithmes est identique quelle que soit la variable réponse étudiée. L'ajout d'une donnée d'exposition ne s'est donc pas révélée décisive dans cette étude.

La régression linéaire, XGBoost et MLP sont dans un mouchoir de poche au sens du score de proximité et surestiment tous grandement la charge de sinistres payés. A l'inverse, ils sous estiment la charge du loss ratio, un phénomène également constaté sur le Chain Ladder qui confirme que l'apprentissage n'est pas en cause.

L'étude par année de développement dans le Tableau 39 est également intéressante : elle met à jour une qualité de prédiction différente selon les années de développement. Le Chain Ladder est par exemple le meilleur sur les deux premières années, mais son score se dégrade au fur et à mesure de la propagation de l'erreur, comme décrits dans le chapitre «Le Chain Ladder». Les algorithmes non linéaires prennent les premières places lors des développements suivants.

Tableau 39: classement des algo par année de développement (sinistres payés)

2015

	diagonals_MAE	diagonals_percentage	diagonals_RMSE	diagonals_ER	diagonals_C_M	diagonals_prox
Genins	1668308.708	3907.502	2079331.469	0.140	0.780	13.908
XGBoost	2263778.210	4795.327	3116380.388	0.448	0.257	15.143
Linear Regression	2147094.046	3392.839	3261872.573	1.108	0.451	15.655
Random Forest	2446596.929	3619.838	3587203.796	1.261	-0.459	16.813
MLP	2714667.813	4350.296	3703712.877	1.049	-0.933	17.106

2016

	diagonals_MAE	diagonals_percentage	diagonals_RMSE	diagonals_ER	diagonals_C_M	diagonals_prox
Genins	1594411.264	72.640	2007391.478	0.619	0.963	14.169
Linear Regression	2736991.478	57.752	3468226.151	3.623	0.980	17.703
XGBoost	2783553.188	63.907	3574873.799	3.514	0.584	18.020
Random Forest	3021176.048	106.106	3741959.300	3.585	-0.278	18.998
MLP	3206006.695	63.123	4075232.181	11.007	-0.410	26.637

2017

	diagonals_MAE	diagonals_percentage	diagonals_RMSE	diagonals_ER	diagonals_C_M	diagonals_prox
XGBoost	92484.149	10.053	94091.513	0.019	1.000	10.471
Random Forest	164656.037	18.269	165798.139	0.210	1.000	11.229
Linear Regression	411826.434	44.232	422289.822	0.109	-1.000	14.063
MLP	656313.074	69.229	682595.676	2.254	1.000	14.688
Genins	709903.539	102.560	959866.146	-0.428	-1.000	15.203

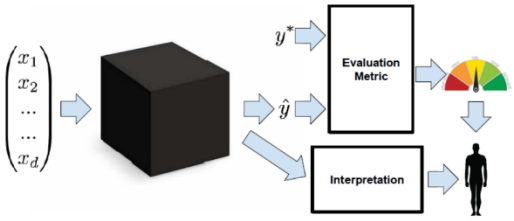
2018

	diagonals_MAE	diagonals_percentage	diagonals_RMSE	diagonals_ER	diagonals_C_M	diagonals_prox
Random Forest	14171.449	2.730	14171.449	-0.027	1.000	8.586
XGBoost	239127.347	46.074	239127.347	-0.315	1.000	11.700
Linear Regression	368328.756	70.967	368328.756	-0.415	1.000	12.232
Genins	445295.651	85.797	445295.651	-0.462	1.000	12.468
MLP	309270.294	59.588	309270.294	1.475	1.000	13.116

4. Interprétation des modèles de type boîte noire

Malgré une démocratisation rendue possible par l'explosion des puissances de calcul et l'accès à des librairies optimisées [52], l'apprentissage statistique automatique en assurance ne s'est pas encore généralisée. L'actuaire doit prévenir le biais et contenir la variance de son modèle tout en s'assurant de l'adhésion de ses parties prenantes[53]. Malheureusement, les meilleures prédictions effectuées sur de larges bases de données sont souvent le fruit d'une grande complexité, inhérente aux algorithmes ensemblistes ou d'apprentissage profond, que les experts eux-mêmes ont du mal à interpréter. La Figure 54 ci-dessous illustre ce besoin de performance et d'interprétation.

Figure 54: Desiderata de la recherche sur l'interprétabilité (Z.C, Lipton, 2017)



La forêt aléatoire requiert de nombreux arbres de décision répartis sur plusieurs niveaux et des millions de nœuds. Les réseaux de neurones sont encore plus compliqués (couches cachées, variables en entrée des neurones, paramètres etc). Une tension émerge entre précision et interprétabilité. Pour autant, les contraintes règlementaires couplées aux impératifs financiers des compagnies d'assurance exigent une compréhension complète du modèle, que ce soit pour vérifier l'absence de données sensibles (GDPR, données interdites etc.) ou simplement valider les comptes de fin d'année.

(Ribero & al, 2016) détaillent deux méthodes afin d' interpréter les prédictions de modèles complexes :

- la méthode SHAP (Shapley Additive exPlanations), qui assigne à chaque variable une valeur d'importance pour une prédiction particulière,
- la méthode LIME (Local Interpretable Model-Agnostic Explanation), qui approxime localement le modèle en introduisant une perturbation des données à l'entrée afin d'en mesurer l'effet à la sortie.

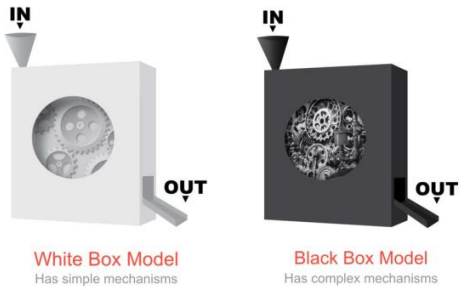
[52] Citons Python et R
[53] Actionnaires, régulateur et auditeurs entre autres

La prochaine section pose le cadre général puis discute chacune d'entre elles en détails. A noter que les modèles opaques ou encore boîtes noires se réfèrent à un système algorithmique dans lequel les entrées et sorties sont observables mais la transformation ne l'est pas.

Leur mécanisme peut être décrit mais demeure difficile à comprendre. En d'autres termes, les paramètres sont optimisés pour la fonction de coût mais ne se fondent pas sur la théorie statistique classique. Il en découle une non reproductibilité des résultats[54] sur un même jeu de données ainsi que des difficultés de compréhension liées aux grandes dimensions[55]. Les modèles transparents ou « glass box » sont quant à eux intrinsèquement interprétable, ils autorisent une compréhension quasi-totale du modèle.

La Figure 55 ci dessous illustre visuellement ce qui les différencie.

Figure 55: Comparaison visuelle du modèle opaque et transparent (Masis, 2021)



[54] Diverses raisons à cela : chiffre aléatoire lors de l'initialisation (poids & hyperparamètres) ou encore discrimination stochastique (dans le cas de la forêt aléatoire).
[55] On parle de malédiction de la dimension (« curse of dimensionality »)

4.1 Les critères d'interprétabilité

Il est difficile de définir mathématiquement l'interprétabilité. Une définition non mathématique est donnée par (Been Kim & al, 2016) selon laquelle elle serait « le degré auquel un humain peut prédire de manière constante les résultats d'un modèle ». Un modèle est donc considéré meilleur *du point de vue de l'interprétabilité* si ses décisions sont mieux comprises par un humain comparativement à celles prises par un autre modèle. Pour en juger, la littérature distingue les notions d'interprétabilité et d'explicabilité.

(Masis, 2021) définit L'INTERPRETABILITE comme la mesure par laquelle un humain, même non expert, peut comprendre les causes et conséquences, entrée et sorties d'un modèle d'apprentissage statistique automatique. L'objectif est d'expliquer les prédictions et justifier la pertinence du modèle étudié. Une autre notion importante est la complexité du modèle, elle-même liée à son architecture[56] et au niveau de mathématiques utilisés. Elle est composée de 3 facteurs : la non linéarité, l'interactivité entre variables explicatives et la non-monotonie – chacun augmentant la complexité globale du modèle en la dimension.

Le tableau ci dessous résume en couleurs ces facteurs par type de modèle, dont ceux étudiés à la section précédente.

Tableau 40: évaluation des propriétés d'interprétabilité par type de modèle, extrait de (Masis, 2021)

White Box?	Model Class	Properties that Increase Interpretability					Task	
		Expl.	Linear	Monotone	Non-Interactive	Regul.	Regr.	Classif.
✓	Linear Regression	🟢	🟢	🟢	🟢	🟢	✓	✗
✓	Regularized Regression	🟢	🟢	🟢	🟢	🟢	✓	✓
✓	Logistic Regression	🟢	🟡	🟢	🟢	🟢	✗	✓
✓	Gaussian Naïve Bayes	🟢	🔴	🟢	🟢	🟢	✗	✓
✓	Polynomial Regression	🟡	🟡	🟢	🟡	🟢	✓	✓
✓	RuleFit	🟢	🟢	🔴	🔴	🟢	✓	✓
✓	Decision Tree	🟢	🔴	🟡	🔴	🟢	✓	✓
✓	k-Nearest Neighbors	🟡	🔴	🔴	🟢	🔴	✓	✓
✗	Random Forest	🔴	🔴	🔴	🔴	🟢	✓	✓
✗	Gradient Boosted Trees	🔴	🔴	🔴	🔴	🟢	✓	✓
✗	Multi-layer Perceptron	🔴	🔴	🔴	🔴	🟢	✓	✓

Sans pour autant pouvoir élucider précisément le fonctionnement intrinsèque du modèle opaque, (Z.C, Lipton, 2017) rappelle que des informations importantes peuvent être récupérées grâce à une explication en langage naturel, une représentation de l'apprentissage ou encore une illustration par l'exemple. C'est l'analyse du compromis entre complexité et interprétabilité, appelé « interprétabilité après coup[57] », qui va permettre d'interpréter des modèles opaques sans sacrifier la précision des prédictions.

L'EXPLICABILITE englobe quant à elle tout ce qui compose l'interprétabilité mais va plus loin : elle exige de la transparence sur les rouages du modèle et son apprentissage. Il ne s'agit plus d'inférer, il faut expliquer chaque étape.

La clarification de ces deux notions a pour objectif de préserver l'actuaire d'une erreur fréquente en apprentissage supervisé : l'illusoire maîtrise du biais. La sur optimisation des paramètres peut conduire à une excellente précision et fait croire au contrôle de la prédiction. Mais la complexité d'un problème ne se

[56] L'architecture ensembliste sera considérée plus complexe qu'une régression linéaire ou un GLM
[57] Préférer « post-hoc interpretability » pour des recherches à ce sujet

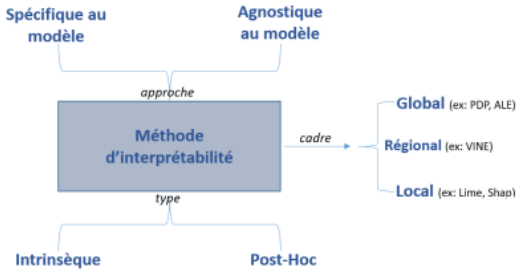
résout pas avec des prédictions « justes » suffisamment souvent, l'examen minutieux des faiblesses du modèle se fait grâce au travail d'interprétation décrit dans les sections suivantes.

4.2 Techniques d'interprétation

La meilleure interprétation d'un modèle simple est le modèle lui-même : il se représente parfaitement et est simple à comprendre. Pour les modèles complexes, comme les méthodes ensemblistes ou les réseaux profonds, le modèle original est trop complexe pour être interprétable. Il faut alors utiliser des modèles explicatifs simples dont l'objectif sera d'approximer le modèle afin de pouvoir l'interpréter.

Les deux grandes techniques d'interprétation présentées par (Z.C, Lipton, 2017) sont les méthodes globales et locales. La première tente de comprendre le modèle dans son intégralité et peut être soit holistique (compréhension de l'entièreté du modèle), soit modulaire (compréhension des variables les plus importantes). La seconde a pour objet de comprendre la prédiction à partir d'une observation en particulier ou d'un groupe d'observations. Qu'elle soit globale ou locale, chacune peut être spécifique ou agnostique au modèle, comme l'illustre la Figure 56.

Figure 56: extrait du mémoire d'actuaire "réseau de neurones, contrôle et transparence"



Cette section et les suivantes se limitent aux méthodes locales qui tentent d'expliquer une prédiction $f(x)$ à partir d'une seule entrée x, avec f la prédiction originale du modèle et g son explication. (S.Lee et al, 2017) expliquent que presque tous les modèles explicatifs utilisent des entrées simplifiées x' liées à l'entrée originale grâce à la fonction de lien $x = h_x(x')$. La méthode locale tente alors de s'assurer que $g(z') \approx f\left(h_x(z')\right)$ à chaque fois que $z' \approx x'$.

Les modèles explicatifs sont donc une fonction linéaire de variables binaires. Ils attribuent un effet ϕ_i à chaque variable. Sommer l'effet de toutes les variables permet d'approximer la sortie $f(x)$ du modèle original, on parle alors de « modèle additif d'attributions de variables » :

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z_i'$$

où $z' \epsilon \{0,1\}^M$, *M est le nombre total de variables d'entrées simplifiées et* $\phi_i \epsilon \mathbb{R}$.

4.2.1 « Local Interpretable Model Agnostic Explanation » (LIME)

La méthode LIME interprète la prédiction individuelle du modèle en approximant localement le modèle à proximité[58] d'une prédiction donnée. Elle appartient à la famille des modèles additifs d'attribution de variables et explique pourquoi telle prédiction a été classée dans telle catégorie, ceci en considérant les x' comme des « entrées interprétables ». Pour trouver ϕ, LIME minimise la fonction suivante :

$$\xi = \text{argmin}_{g \epsilon G}[\mathrm{L}(f, g, \pi_{x'}) + \Omega(g)]$$

avec :

- L la fonction de coût,
- f la fonction associée au modèle initial d'apprentissage,
- g la fonction associée au modèle explicatif de substitution que l'on souhaite optimiser,
- $\pi_{x'}$ une mesure de proximité définissant la taille du voisinage autour de x,
- Ω la fonction traduisant la complexité du modèle et qui, comme dans la régularisation Ridge, fait office de pénalisation et oblige le modèle à rester simple,

Le modèle explicatif $g(z')$ approche le modèle $f\left(h_x(z')\right)$ grâce à la fonction de coût L sur un échantillon dans l'espace des entrées simplifiées pondérées par le noyau local $\pi_{x'}$ tandis que la fonction Ω pénalise la complexité de g. La méthode est agnostique au modèle d'apprentissage statistique: elle ne requiert pas de comprendre comment le modèle fait ses prédictions.

La fonction de coût L se définit comme suit :

$$\mathrm{L}(f, g, \pi_{x'}) = \sum_{z \in Z} \left(f(z) - g(z')\right)^2$$

En d'autres termes, pour chaque prédiction, LIME va échantillonner des données sur la distribution d'apprentissage (les valeurs catégorielles seront échantillonnées sur base de leur fréquence d'apparition). Ce nouvel échantillon « perturbé » va permettre le calcul d'une distance avec les observations initiales du modèle boite noire permettant ainsi de calculer un score de similarité. Le modèle boite noire est ensuite appliqué aux données perturbées et les variables expliquant le mieux les prédictions sont sélectionnées : un modèle simple est alors ajusté et les variables précédemment sélectionnées sont pondérées par leur similarité avec les prédictions initiales. Les poids des variables sont enfin extraits.

La méthode est rapide à calculer et permet une interprétation sélective grâce à la pénalisation Ω. Elle est néanmoins sensible à $\pi_{x'}$: pour deux mesures proches, LIME peut fournir un modèle explicatif fort différent. Enfin, et contrairement à la mesure SHAPE, LIME ne garantit ni les propriétés de précision ni celle de cohérence (voir section suivante).

[58] Au sens de l'espaces de Hilbert et plus précisément du RKHS (Reproducing Kernel Hilbert Space)

4.2.2 "Shapley additive explanation" (SHAP)

SHAP est une mesure d'importance des variables explicatives qui unifie un ensemble de mesure dont LIME. Elle vérifie également certaines propriétés statistiques utiles pour l'interprétation. Elle nécessite néanmoins des approximations afin de conserver des temps de calculs raisonnables.

4.2.3 La valeur de Shapley

L'interprétation de la valeur de Shapley est la suivante: compte tenu des valeurs prises par les variables explicatives, la contribution de l'une d'entre elles à la différence entre la prédiction réelle et sa valeur moyenne est évaluée par la valeur de Shapley.

Elle est la moyenne de la contribution marginale d'une variable explicative j comparativement à toutes les autres combinaisons de variables possibles, comme décrit ci après (C. Molnar, 2019) :

$$\phi_j(val) = \sum_{S \subseteq \{1,\dots,p\} \setminus \{j\}} \frac{|S|!\ (p - |S| - 1)!}{p!} \left(val\ (S \cup \{j\}) - val\ (S)\right)$$

Avec :

- S la combinaison de variables explicatives,
- p le nombre de variables,
- x le vecteur des valeurs prises par les variables explicatives pour la prédiction à expliquer,
- $val_x(\mathrm{S})$ la prédiction donnée par S et qui se définit comme suit lorsqu'il n'y a qu'une seule variable explicative étudiée:

$$val_x(\mathrm{S}) = \int \hat{f}(x_1, \dots, x_p)\, d\mathbb{P}_{x \notin S} - E_X\left(\hat{f}(X)\right),$$

Lorsqu'une combinaison de variables explicatives est étudiée, une intégrale multiple est calculée pour chaque variable qui n'est pas dans S. Dans son ouvrage (C. Molnar, 2019) prend l'exemple d'un modèle à 4 variables explicatives (X_1, X_2, X_3, X_4) et dont on veut évaluer la prédiction pour la combinaison $\mathrm{S} = (x_1, x_3)$ représentant les valeurs prises par les variables explicatives $X_1\ et\ X_3$ respectivement.

$$val_x(\mathrm{S}) = \iint \hat{f}(x_1, X_2, x_3, X_4)\, d\mathbb{P}_{X_2 X_4} - E_X\left(\hat{f}(X)\right),$$

La valeur de Shapley satisfait par ailleurs les propriétés d'efficacité, symétrie, nullité et additivité. Elles assurent une explicabilité totale et non biaisée, contrairement à la méthode LIME. Chacune est brièvement décrit ci-dessous.

Propriété d'**efficacité** : la contribution des variables explicatives doit être égale à la différence entre la prédiction de x et la moyenne.

$$\sum_{j=1}^{p} \phi_j = \hat{f}(x) - E_X\left(\hat{f}(X)\right)$$

Propriété de **symétrie** : la contribution de deux valeurs j et k doivent être les mêmes si elles contribuent de manière équivalente à toutes les autres combinaisons possibles. Autrement dit, si : $val\ (S \cup \{j\}) = val\ (S \cup \{k\})\ \forall\ S\ \subseteq \{1, \ldots, p\} \backslash \{j, \forall\ k\}$, alors $\phi_j = \phi_k$.

Propriété de **nullité** : une valeur j qui n'a pas d'influence sur la valeur prédite doit avoir une valeur de Shapley nulle. Autrement dit, si : $val\ (S \cup \{j\})\ = val\ (S)\ \forall\ S\ \subseteq \{1, \ldots, p\}$ alors $\phi_j = 0$.

Propriété **d'additivité** : pour les modèles ensemblistes, la valeur de Shapley pourra être calculée pour chaque modèle basique puis sommée ainsi : $\phi_{(j)}(n + m) = \phi_j(n) + \phi_j(m)$. Illustration avec la forêt aléatoire : pour une valeur donnée, il est possible de calculer la valeur de Shapley pour chaque arbre de décision puis en faire la moyenne pour l'obtenir au titre de la forêt aléatoire.

L'un des principaux inconvénients liés au calcul de cette valeur tient au temps de calcul requis. En effet, le calcul exact de $\phi_j(val)$ nécessite de calculer la prédiction du modèle avec et sans la variable X_j pour toutes les conditions possibles de S. Le nombre possible de combinaisons est donc de 2^p à chaque ajout de valeur. Il est donc recommandé d'approximer la valeur de Shapley en échantillonnant des combinaisons de variables et en limitant le nombre d'itérations effectuées par la méthode. Il s'agira de trouver le juste milieu entre approximation et temps de calcul.

4.3 Application sur le modèle

« Interpret ML » est un package open source qui inclus les dernières techniques d'interprétabilité et fournit une explicabilité totale des modèles « white box », permettant ainsi d'auditer la manière dont le modèle est parvenu à telle décision plutôt qu'une autre. Mais au-delà des modèles classiques disponibles dans scikit-learn, le package contient également un modèle à part entière pour interpréter les « boites noires » : Explainable Boosting Machine (EBM).

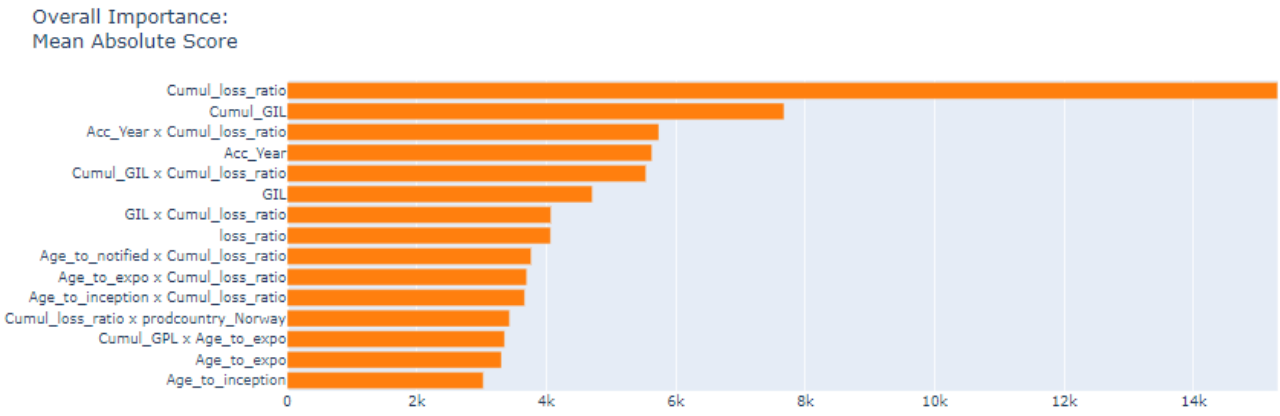
Il s'agit d'un modèle interprétable développé par Microsoft Research. Il utilise les techniques de machine learning développées dans les sections précédentes (Le Bagging et Le Gradient Boosting). Il est aussi précis que la Forêt Aléatoire ou le Gradient Boosting mais donne des explications interprétables à l'actuaire.

Son fonctionnement a été pensé pour cela : EBM crée en effet un arbre de décision uniquement à partir de la variable n°1, puis transfère le résidu à l'arbre suivant en utilisant l'algorithme de boosting. Idem pour toutes les autres variables. Un grand nombre d'itérations est effectué sur chaque variable afin d'obtenir la meilleure fonction pour la variable concernée. Une fois toutes les variables itérées[59], il devient possible de comparer la contribution de chaque variable à la prédiction globale.

4.3.1 Application à l'étude

EBM est lancé sur le modèle de l'étude, en utilisant 60% des données. Le tableau ci-dessous résume l'importance globale des variables explicatives du modèle.

Tableau 41: Importance globale (Mean Absolute Score)



Le Tableau 41 classe les variables les plus significatives pour la prédiction par ordre décroissant. Ainsi, « Cumul_loss_ratio » contribue le plus à la force prédictive du modèle. L'abscisse du graphe est la moyenne des valeurs absolues de la Shap Value : en d'autres termes, elle mesure l'impact moyen de la variable concernée sur l'output global du modèle. Les variables explicatives les plus significatives sont classées par ordre croissant sur l'ordonnée.

[59] Les itérations sont faites en parallèle afin d'optimiser le temps de calcul

Le cumul du loss ratio est de loin la variable explicative la plus importante : cette variable cumul en effet les sinistres payés par le passé, normalisés par l'exposition. Cette première place valide la conviction métier qui consiste à attendre d'un sinistre un comportement futur similaire à celui constaté dans le passé.

Il est intéressant de noter que le top 5 des variables les plus importantes dans la prédiction sont identiques – bien que parfois dans le désordre- quelle que soit la variable réponse : sinistre payés, loss ratios ou charge totale. Une illustration est donnée ci-dessous.

Tableau 42: variable réponse des payés (GPL)

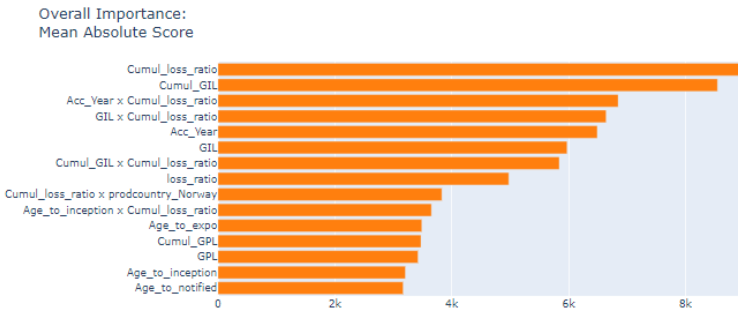


Tableau 43: variable réponse des loss ratios

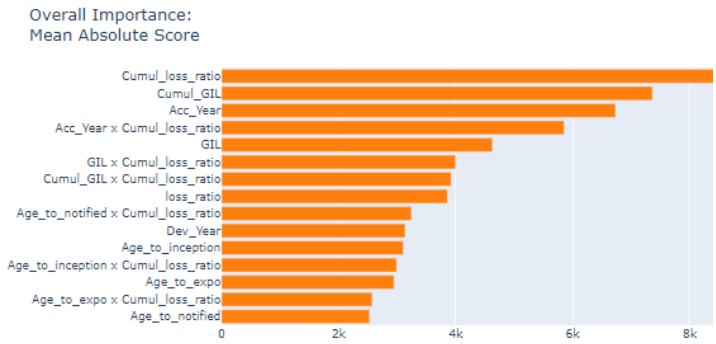


Tableau 44: variable réponse de la charge totale

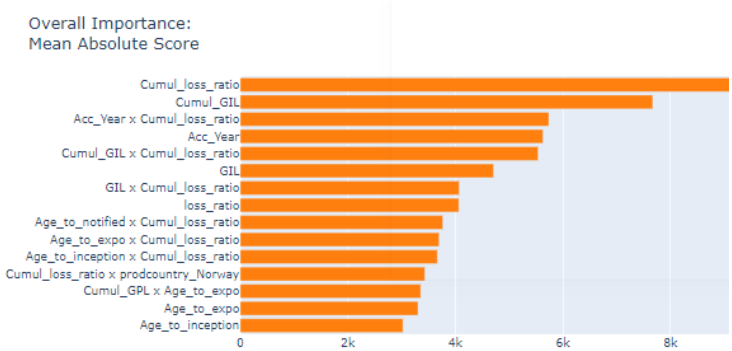
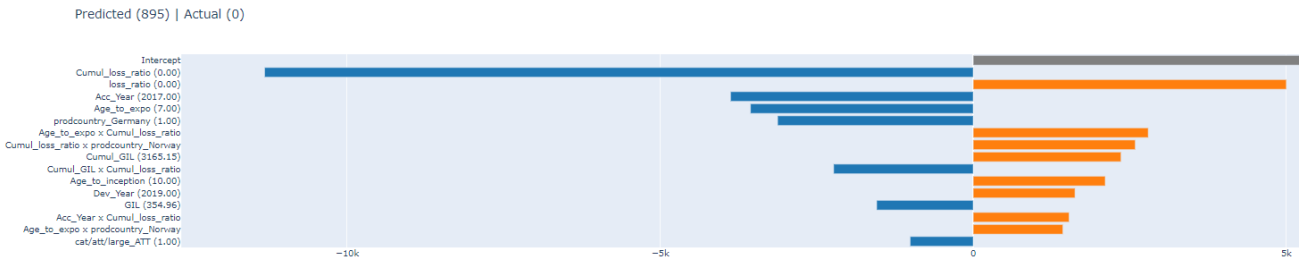


Figure 57 ci-dessous permet d'analyser la prédiction à un niveau local. Elle représente sous forme d'histogramme la contribution de chaque variable explicative à la prédiction de la transaction située sur une ligne en particulier. Choix est fait de sélectionner la ligne 76 (choix arbitraire).

Figure 57: Analyse locale de la transaction en ligne 76 (pour la variable réponse « charge totale »)



Les scores sont présentés dans l'unité de sortie du modèle (ici des dollars). Ainsi, chaque barre de l'histogramme correspond au montant que la variable explicative concernée a ajouté ou soustrait à la prédiction finale.

Dans le cas présent, le modèle à prédit une valeur négative de 895$ alors que la valeur réelle de la transaction est nulle. Les variables dont la contribution respective est la plus élevée sont en ligne avec l'importance globale du modèle présentée en page précédente : le cumul du loss ratio, l'année de survenance du sinistre ainsi que l'âge du sinistre (« age_to_expo »).

Il est à noter que la procédure de « One Hot Encoding » sur les variables catégorielles ne permet pas à Interpert ML de restituer des résultats totalement fiables : ceci est lié au fonctionnement de LIME qui, comme expliqué en section 4.2.1, définit la puissance contributive d'une variable par permutation des données. LIME n'étant pas en mesure de reconnaître les colonnes appartenant à la même variable initiale, il crée donc un point de donnée incohérent avec les données initiales.

L'analyse individuelle d'une vingtaine de transactions a permis de mettre à jour la redondance des valeurs réelles nulles. Ceci est dû aux traitements décrits à la section « 3.2 La modélisation » : l'ajout des trimestres manquants était nécessaire à la bonne marche de l'algorithme mais a crée un grand nombre de transactions nulles.

Le graphique de la SHAP Value permet d'illustrer la contribution positive ou négative de chaque prédicteur à la variable réponse. Il est composé de toutes les données utilisées lors de l'apprentissage et permet de classer les variables prédictives par ordre décroissant. La place occupée par chaque point sur l'axe horizontal détermine son effet sur la prédiction (positif ou négatif). Enfin, la couleur (bleu ou rouge) détermine si la valeur de Shap associée à l'observation est faible ou élevée (respectivement).

Figure 59: SHAP value pour chaque variable prédictive (prédictions MLP des sinistres payés)

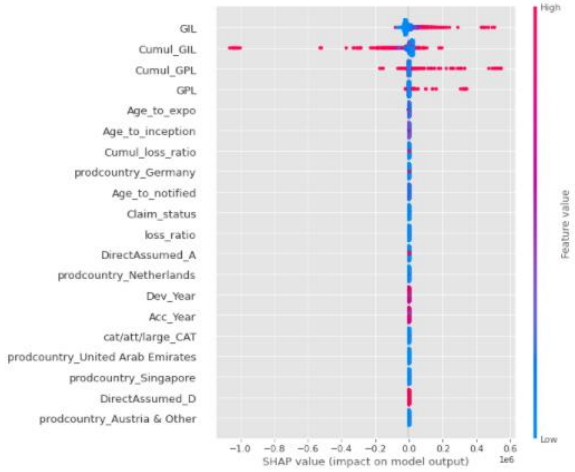
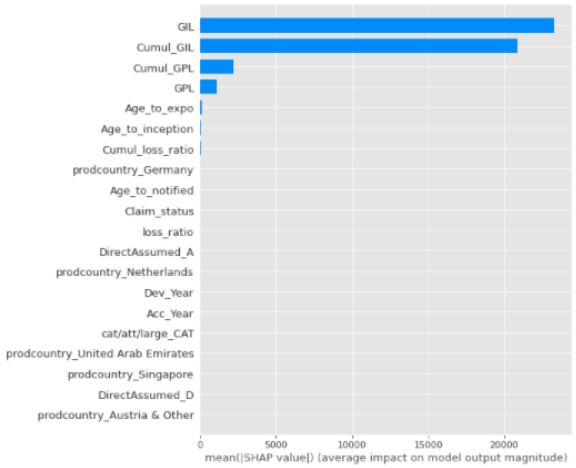


Figure 58: Importance globale des variables prédictives (MLP)



L'analyse sur le MLP de la Figure 59 permet d'identifier 4 variables prédictives significatives : « GIL », « Cumul_GPL » et « GPL » ont un impact positif fort sur les prédictions du modèle tandis que « Cumul_GIL » impact négativement la prédiction du modèle. La Figure 58 permet de visualiser différemment les informations de la Figure 59.

Les mêmes graphiques sur XGBoost (Figure 61 et Figure 60) donne des résultats plus bruités où un plus grand nombre de variables explicatives affecte la prédiction. Le top 5 reste néanmoins inchangé versus le MLP, validant ainsi les choix effectués à l'étape de « features engineering ».

Figure 61: SHAP value pour le XGBoost (prédiction des sinistres payés)

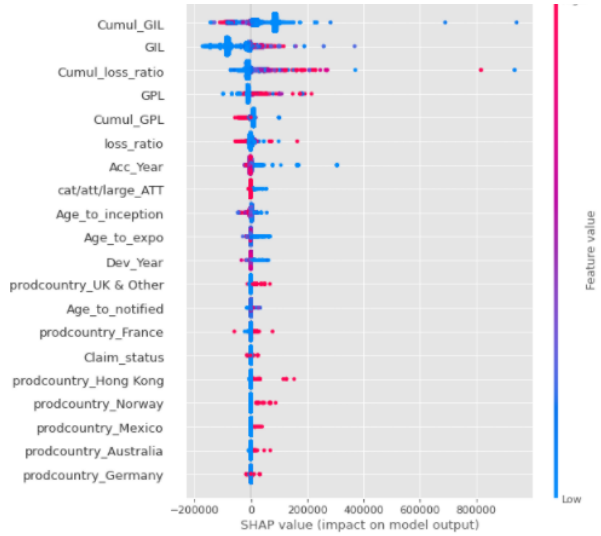
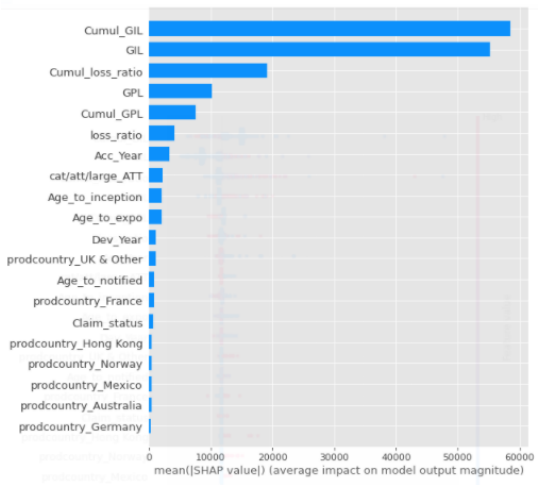


Figure 60: Importance globale des variables prédictives (XGBoost)



5. Conclusion

Cette étude détaille l'implémentation de divers modèles - *incluant la forêt aléatoire, le Gradient Boosting et le réseau de neurones*- afin de prédire la charge sinistre totale des années de survenance 2010-2014 sur la branche dommage, soit un horizon temporel limité à 5 ans. Afin de ne perdre aucune variable explicative, choix a été fait d'effectuer la prédiction au niveau transactionnel. De ce choix est né une myriade de contraintes techniques à l'étape de modélisation : création de transactions à 0, récupération de la transaction suivante pour l'étape d'apprentissage, ajout d'un classificateur binaire pour la fermeture du sinistre etc. Toutes ces étapes ont permis d'améliorer progressivement les prédictions et parvenir à un niveau acceptable au sens du score de proximité, mais au détriment de la simplicité.

Les techniques récentes d'interprétabilité offrent par ailleurs un cadre formel afin de mieux comprendre les modèles boites noires. Le classement SHAP a par exemple validé l'intuition que le meilleur prédicteur pour le coût final d'un sinistre à l'instant t de sa vie $(avec\ t < T)$ est le cumul de ses flux passés, validant ainsi notre intuition lors de l'étape de « features engineering[60] » développée en section 1.2. Il a également permis d'identifier les variables explicatives dont la force prédictive était anecdotique, permettant ainsi de réduire les données en entrée du modèle et gagner en temps de calcul.

Les résultats de l'étude apportent plusieurs enseignements intéressants. Comme attendu, le Chain Ladder règne en maître sur les premières années de développement, avec des scores de proximité et une erreur relative raisonnablement faible. Mais les modèles ligne à ligne s'avèrent plus précis dans les années de développements lointaines, notamment le XGBoost qui semble capturer une relation temporelle au sein des données alors que les autres n'y parviennent pas.

L'une des faiblesses du modèle est la prédiction sur le statut du sinistre. Ce classificateur a été entraîné séparément des modèles de prédiction de charge : des interactions fructueuses auraient pu naître d'un entraînement conjoint, même s'il s'agit d'une simple conjecture à ce stade. La classification aurait également pu être plus précise, avec la prédiction d'une probabilité (de fermeture) au lieu d'un résultat binaire ouvert/fermé. Cela aurait permis de mesurer la sensibilité des résultats en fonction de plusieurs seuils à tester itérativement.

Plus généralement, l'écart séparant le processus consistant à reprendre dans un fichier Excel les données du trimestre précédents afin de mettre à jour les facteurs de développement et un modèle de prédiction entièrement automatisé développé sur Python est immense. Ce dernier inclut la requête, le nettoyage et l'import des données, la prédiction et la production de rapports d'interprétation, autant d'étapes complexes à programmer avant d'envisager une mise en production.

A ce jour, aucune publication actuarielle ne permet de déployer un modèle d'apprentissage statistique à grande échelle au sein d'une compagnie d'assurance non-vie. Les concepts présentés sont intéressants, souvent innovants, mais l'incertitude demeure grande sur ce qui marche ou ne marche pas. Il ne faut cependant pas sous-estimer la vitesse à laquelle la technologie progresse. Une publication scientifique ou une librairie révolutionnaire, suivie par l'adoption de quelques acteurs clés de l'assurance non-vie permettront le bond quantique tant attendu dans le domaine de l'estimation des IBNR. Il semble d'ailleurs qu'une masse critique ait d'ores et déjà été atteinte en termes de compétences et de prise de conscience autour de l'apprentissage statistique automatisé.

[60] Travail effectué sur les variables explicatives (modification ou ajout de variables)

6. Bibliographie

Anwar Aqeel Machine Learning Cheat sheets [En ligne]. - 2021. - https://medium.com/swlh/cheat-sheets-for-machine-learning-interview-topics-51c2bc2bab4f.

ASTIN Non life reserving practices [Rapport]. - [s.l.] : International Actuarial Association, 2016.

B. Harej & al Individual claim development with Machine Learning [Revue]. - [s.l.] : ASTIN report, 2017.

Balona, C. et Richman, R The actuary and IBNR Techniques: a machine learning approach [Livre]. - 2020.

Barnett et Dobson An introduction to Generalized Linear Models [Livre]. - [s.l.] : CRC Press, 2008.

Been Kim & al Examples are not enough, learn to criticize! Criticism for interpretability [Revue]. - [s.l.] : Advances in Neural Information Processing Systems , 2016.

Biau Gérard et Devroye, Luc Consistency of random forest and other averaging classifiers [Livre]. - [s.l.] : UPMC and McGill University, 2008.

Bornhuetter & al The actuary and IBNR [Rapport]. - 1972.

Breiman Leo Bagging Predictors [Rapport]. - [s.l.] : Berkley University, 1996.

Breimann,L. et al Classification and regression trees [Livre]. - 1984.

C, Bilore Application de l'apprentissage automatique au provisionnement ligne à ligne en assurance vie [Rapport]. - [s.l.] : Mémoire d'actuaire CEA, 2016.

C. Molnar Interpretable machine learning. A Guide for Making Black Box Models Explainable [Livre]. - 2019.

Charpentier et Denuit Mathématiques de l'assurance non-vie [Livre]. - [s.l.] : Economica, 2005. - Tome 2.

Code des assurances Legifrance, Art R331-6. - 2013.

Duval, F et Pigeon, M Individual loss reserving using gradient boosting-based approach [Conférence]. - 2019.

Efron, B. et Tibshirani, R. An introduction to the Bootstrap [Livre]. - [s.l.] : Springer Science Business Media, 1994.

Fabre Rudelle D Apport des méthodes d'apprentissage statistique pour le provisionnement individuel en assurance non vie [Rapport]. - 2018.

Gibello, H. et Lebrun, B Crédibilisation des méthodes de provisionnement non vie [Rapport]. - [s.l.] : Mémoire d'actuaire, 2011.

Harej, B et al, Individual claim development with machine learning [Livre]. - 2017. - ASTIN.

JO de l'Union Européenne Règlement relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel [En ligne]. - 2016. - https://eur-lex.europa.eu/legal-content/.

Joshi Prateek What is Bootstrap Sampling in Statistics and Machine Learning? [En ligne]. - 2020. - https://www.analyticsvidhya.com/blog/2020/02/what-is-bootstrap-sampling-in-statistics-and-machine-learning/.

Lopez, O. et al A tree-based algorithm adapted to microlevel reserving. - [s.l.] : Electronic Journal of Statistics, 2016.

Lucidar Gradient descent for neural networks [En ligne]. - https://lucidar.me/en/neural-networks/single-layer-gradient-descent/.

Masis Serg Interpretable Machine Learning with Python [Livre]. - [s.l.] : Packt, 2021.

Nelder & al Generalized linear models [Revue]. - [s.l.] : Journal of the Royal Statistical Society, 1972. - Vol. Series A.

Ortiz Loup Eléments d'Intelligence Artificielle faible en provisionnement non vie [Rapport]. - 2019.

Pitts W. McCullogh & W. A logical calculus of the ideas immanent in nervous activity [Livre]. - [s.l.] : The Bulleting of mathematical biophysics .

Ribero & al Why should I trust you? Explaining the predictions of any classifiers [Rapport]. - 2016.

S.Lee et al A unified approach to interpreting Model Predictions [Conférence] // 31st Conference on Neural Information Processing Systems. - 2017.

Sarkar Dipayan Ensemble Machine Learning Cookbook [Livre]. - [s.l.] : Packt Publishing , 2019.

T.Hastie B.Efron & Computer age statistical inference [Livre]. - 2017.

Tim Miller Explanation in artificial intelligence: Insights from the social sciences [Revue]. - 2017. - arXiv Preprint arXiv:1706.07269.

Wikipedia [En ligne]. - https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_artificiels.

Wikipedia Bootstrap: principe général [En ligne]. - 2021. - https://fr.wikipedia.org/wiki/Bootstrap_(statistiques).

Yufei Luo Amélioration de la modélisation des sinistres graves à l'aide d'une approche d'apprentissage [Rapport]. - [s.l.] : Institut des actuaires, 2016.

Z.C, Lipton the Mythos of Model Interpretability [Rapport]. - 2017.

7. Index des illustrations

8. Index des tableaux

Annexes

8.1 Requête SQL d'extraction

```
select PFKPolNbr, PFKClmNbr, FKTradeOfBusiness, FKCovrgTypeCode, InceptionDate, ExpiryDate,
LossAttachmentDate, DateNotified, ClmDesc, DirectAssumed, USStates, prodcountry, Dev_Qtr,
Exposure_Qtr, [ChartField 1/AccidentYear], [cat/att/large], ri_type,

sum ([Gross EP]) as 'GEP',
sum ([Ceded EP]) as 'CEP',
sum ([Gross Incurred Loss] + [Gross Incurred ALAE]) as 'GIL',
sum ([Ceded Incurred Loss] + [Ceded Incurred ALAE]) as 'CIL',
sum ([Gross Paid Loss] + [Gross Paid ALAE]) as 'GPL',
sum ([Ceded Paid Loss] + [Ceded Paid ALAE]) as 'CPL'
from dbo.[Super ActSubDB_swg_2020Q4_bookings_restated/restructured]
where
[OperatingUnit/OBU] in ('I - Property Multinational',
'I - Property Domestic', 'I - Property Captives', 'I - Property OAPS') and
[ChartField 1/AccidentYear] > 2015

group by PFKPolNbr, PFKClmNbr, FKTradeOfBusiness, FKCovrgTypeCode, InceptionDate, ExpiryDate,
LossAttachmentDate,
DateNotified, ClmDesc, DirectAssumed, USStates, prodcountry, Dev_Qtr, Exposure_Qtr, [ChartField
1/AccidentYear],
[cat/att/large], ri_type
```