Mémoire présenté le : 4 mai 2021

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

Par : Marie Hainneville

Titre : Reverse Engineering dans la tarification Automobile

Confidentialité : ☒ NON ☐ OUI (Durée : ☐ 1 an ☐ 2 ans)

Les signataires s'engagent à respecter la confidentialité indiquée ci-dessus

Membre présents du jury de l'Institut des Actuaires :

M. Anthony Derien

M. Michel Louchard

Membres présents du jury de l'ISFA :

Mme Esterina Masiello

M. Denis Clot

Entreprise : Reacfin

Directeur de mémoire en entreprise :

Nom : Michaël Lecuivre

Signature :

Invité :

Nom :

Signature :

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

Signature du responsable entreprise :

Signature du candidat :

Remerciements

Tout d'abord, je tiens à remercier toute l'équipe de Reacfin pour son accueil chaleureux et son encadrement. Tout particulièrement le pôle Non-Vie qui a toujours su se rendre disponible quand j'avais des interrogations, et m'a immédiatement intégrée et fait partager leurs connaissances ! Un merci tout particulier à Julie, Sam, Michaël et Xavier pour leur disponibilité !

Merci à Rose-Marie Gallon (groupe 8, le meilleur !) sans qui ce mémoire n'aurait jamais existé puisque sans elle, le monde de l'actuariat serait demeuré un mystère pour moi.

Merci également à Mr Picard, qui, rétrospectivement, a été un des meilleurs professeurs de mathématiques que j'ai eu la chance d'avoir en cours, et qui m'a fait prendre conscience de mon appétence pour les statistiques.

Et bien sûr, tout ce temps passé à l'ISFA n'aurait pas été le même si ce n'était pour la merveilleuse aventure humaine qui s'est déroulée avec mes camarades. On pensera également à saluer le groupe des Gagnantes, que le Ninkasi regrette déjà pour ses blind-tests, mais qui saura demeurer dans le temps.

Finalement, n'oublions pas également de remercier deux amis indispensables dans cette entreprise que constitue la rédaction d'un mémoire. Ordinateur et internet, ces sauveurs, toujours disponibles et prêts à aider. Bien sûr, ils ne fonctionnent pas seuls, et des dizaines de contributeurs à StackOverFlow ont contribué sans le savoir, à cet ouvrage.

"All the world is made of faith, and trust, and pixie dust" - J.M. Barrie

Résumé

Dans un contexte de compétitivité accrue au sein du marché de l'assurance non-vie, et avec l'évolution constante des capacités de calcul des ordinateurs, il devient de plus en plus nécessaire pour un assureur de connaître sa place sur le marché. Afin de répondre à cette problématique, il est important de bénéficier de suffisamment de données pour mener des analyses pertinentes. Pour cela ce mémoire se base sur le principe de *reverse engineering,* permettant de retrouver le modèle de tarification utilisé par un assureur, et propose une comparaison des performances de différents modèles, des plus classiques (GLM, GAM) aux plus raffinés (Random Forest, GBM, réseaux de neurones). Une fois ces modèles définis, il est plus simple d'obtenir de nouvelles données sur un segment d'intérêt, ou d'exploiter directement ces modèles de manière à pouvoir étudier la position stratégique d'un assureur. En complément de cette analyse comparative, ce mémoire ouvre également la voie vers des solutions pour pallier à différents problèmes rencontrés lors de l'implémentation de cette méthodologie de *reverse engineering*. On citera notamment les problèmes liés à la sélection des variables et à une multicollinéarité éventuelle, ou ceux découlant d'interactions entre les variables, qui sont souvent utilisées pour répliquer au mieux les phénomènes réels. Finalement, la problématique d'un changement de tarif aléatoire au sein d'un portefeuille sera également évoquée, puisque dans notre processus de *reverse engineering*, il s'agit d'un évènement que l'on peut rencontrer et qui peut altérer défavorablement le modèle s'il n'est pas pris en compte.

Abstract

In a context of increased competitiveness within the non-life insurance market, combined with the constant evolution of the capacities of computers, it becomes more and more necessary for an insurer to be aware of its position on the market. In order to overcome this problem, it is important to have enough data to conduct relevant analyzes. This thesis uses the principle of *reverse engineering*, allowing to find the pricing model used by an insurer, and compares the performances of different models, from the most classic (GLM, GAM) to the most refined (Random Forest, GBM, neural network). Once these models are defined, it is quite simple to obtain new data on a segment of interest, or to directly exploit these models; it enables thus to study the strategic position of an insurer. In addition to this comparative analysis, this thesis also offers solutions to stun various problems related to this *reverse engineering* methodology. These include the problems associated with variable selection and possible multicollinearity, or those arising from interactions between variables, which are often used to replicate real phenomena. Lastly, the problem of a random price change within a portfolio will also be mentionned, since in our reverse engineering process, this is an event that can be encountered and which can alter in an unfortunate way our models if not taken into account.

Note de synthèse

Avec l'évolution constante des réglementations et des technologies, utilisées la plupart du temps dans le but de favoriser l'assuré, le marché de l'assurance non-vie, et plus particulièrement de l'assurance automobile, est devenu de plus en plus concurrentiel. En effet, au cours des dernières années le phénomène de segmentation tarifaire, qui se veut donner un tarif au plus près de la réalité pour chaque client, s'est fortement développé. Cette segmentation est également importante au niveau même de l'assureur, puisqu'elle limite le risque de sélection adverse et de rachat.

Pour réaliser cette segmentation, les assureurs non-vie s'appuient généralement sur une approche modèle-fréquence et sur des méthodes traditionnelles telles que les modèles linéaires généralisés (GLM). Pour cela, les assureurs exploitent les données que l'assuré leur a fournies via son questionnaire de souscription. Grâce à ces données et à leur exploitation sur le long terme, l'assureur peut, en fonction de la sinistralité survenue, déterminer le juste prix à facturer à l'assuré pour sa couverture de risque. Néanmoins ces méthodes classiques, issues de l'économétrie traditionnelle, ont leurs limites : avec elles, il est compliqué de prendre en compte certains phénomènes non-linéaires comme les interactions entre les variables ou l'abondance des variables tarifantes.

Une manière de pallier à ces difficultés est d'utiliser d'autres types de modèle que les GLM. Mais ce changement a un prix : celui de l'interprétabilité et de la traçabilité. Les assureurs ont de plus en plus tendance à raffiner leurs modèles, au risque d'une perte de lisibilité claire sur les faits entrant réellement en jeu. Cela complexifie donc l'étude du marché de l'assurance non-vie. En effet, chaque assureur distille lui-même les variables qu'il va utiliser, et choisit également comment il va les prendre en compte. Une comparaison segmentarisée des assureurs présents sur une branche non-vie est désormais compliquée à mettre en place rapidement.

L'objectif de ce mémoire est donc, à l'aide de différents profils clients accompagnés des tarifs appropriés, de reconstruire le modèle utilisé par un assureur. Cela permettra ainsi d'obtenir de nouvelles données sur des segments sur lesquels nous ne disposons pas forcément de tarifs, et de développer une comparaison tarifaire au cas par cas. Il sera ainsi plus facile pour un assureur d'évaluer sa position sur le marché, et d'adapter son offre.

Pour cela, nous étudierons différents types de modélisation qui peuvent s'appliquer au secteur de l'assurance non-vie. La méthode de marché traditionnelle de la modélisation linéaire généralisée, évoquée plus haut, est très utilisée depuis de nombreuses années. Néanmoins elle reste peu précise et est particulièrement sensible aux variables et au choix potentiel de leurs combinaisons. Afin d'améliorer les performances prédictives de cette méthode, des assureurs se tournent vers des modèles affines généralisés (GAM) ou des améliorations de GLM, avec des transformations polynomiales par morceaux (splines cubiques) par exemple. Ces méthodes permettent notamment à l'actuaire, à travers l'analyse de graphiques de tendances, de mieux appréhender les données disponibles ; cela conduit, in fine, à une meilleure définition des frontières des classes tarifaires commercialisées.

Pour parvenir à reconstituer ces modèles dont la création se fait sur une sélection arbitraire de différentes variables parmi la pléthore de choix offerte, plusieurs méthodes seront employées. Ainsi une première étape sera de mettre en lumière la complexité de reconstruire un tarif en ayant peu d'informations sur les variables utilisées ou les regroupements de modalités qui ont été effectués. Cela amènera donc une réflexion sur comment sélectionner de manière optimale les variables qui ont pu potentiellement servir à l'élaboration du modèle, tout en s'affranchissant d'une éventuelle redondance d'une partie de l'information. En effet, un trop grand nombre de régresseurs conduit à

des gênes statistiques et informatiques qui imposent de réduire la dimensionnalité. Comme il s'agit d'un travail réalisé en amont, il est nécessaire pour la fiabilité du modèle reconstruit de retrouver cette réduction de la dimensionnalité. Différentes méthodes seront ainsi présentées, des méthodes statistiques comme les filtres, les méthodes wrapper et embedded, ou des méthodes de régularisation des coefficients (Lasso et Ridge), aux méthodes moins traditionnelles issues du machine learning.

Une fois cette sélection faite, un modèle sera développé dans le but de reconstruire le tarif présent dans les données. Pour analyser la pertinence de cette démarche et la véracité de notre modèle ainsi reconstitué, nous comparerons ses résultats à d'autres types de méthode. En effet, avec l'évolution des capacités des ordinateurs, des alternatives à l'apprentissage statistique se sont développées. De nombreux algorithmes ont ainsi vu leur popularité s'accroître aux cours des dernières années (Random Forest, GBM, réseau de neurones…). Leur efficacité n'est plus à prouver, puisqu'ils se sont montrés plus qu'efficaces dans de nombreux challenges telles que les compétitions Kaggle. Ce mémoire tâchera donc de comparer ces modèles de pointe aux modèles plus classiques. Ci-dessous, les résultats obtenus sur la base de données liée à un seul assureur sont exposés :

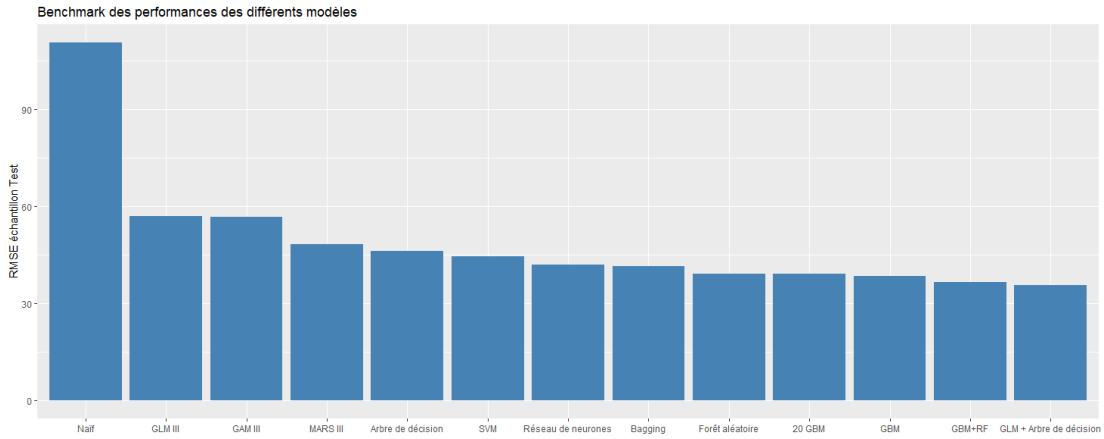


Figure 1 : Benchmark des performances des différents modèles

L'étude approfondie des divers modèles montre que la sélection de variables est justifiée. Le GLM, même modifié, peine à atteindre les performances des modèles issus de l'apprentissage supervisé (Random Forest, GBM). Cela vient probablement du fait que ces modèles sont plus aptes à transcrire des relations non-linéaires et des effets croisés entre deux (ou plus) variables explicatives dus à une éventuelle multicollinéarité. De plus ces modèles présentent l'avantage non-négligeable de ne pas nécessiter une pré-sélection des variables dans leur mise en place. Le gain de temps de l'actuaire est donc indiscutable.

Toutefois, ces modèles ont l'ennuyeuse tendance à être considérés comme des modèles 'black box', c'est-à-dire difficilement exploitables. Hors, au-delà de parvenir à obtenir de nouvelles données et des comparaisons individuelles de tarif, il est important pour l'actuaire de s'assurer que le modèle est pertinent ; un modèle doit pouvoir être contrôlé et assuré de sa bonne technicité. On s'attardera donc dans ce mémoire sur l'interprétabilité des modèles de machine learning, et ce qu'ils peuvent nous enseigner. En effet, de nombreux outils apparaissent désormais dans le but de 'blanchir' ces modèles, de manière à produire des informations aussi limpides que celles des méthodes plus classiques.

Néanmoins, dans la réalité, d'autres paramètres de tarification peuvent intervenir (promotion éventuelle, détention de multi-contrats…) et ceux-ci sont difficilement décelables sans informations a priori. Que ce soit par les méthodes classiques ou de machine learning, il est compliqué de déceler un changement de tarif inhérent aux données, sans savoir à quel moment celui-ci se produit. Ce mémoire

proposera donc quelques pistes à ce type de problématique dans sa dernière partie. Différents types de modèles seront ainsi à nouveau comparés dans le but de parvenir au modèle le plus efficient possible.

Il est cependant nécessaire de nuancer le fait que les performances obtenues tout au long de cette étude ne sont pas intrinsèques mais grandement liées à la base de données utilisée. En effet, puisque les techniques de modélisation varient selon les assureurs, les différentes modélisations peuvent voir leur performance modifiée selon la technique employée originellement. Il a néanmoins été confirmé, dans un cadre plus global s'établissant sur une base de données plus importante, avec plus d'assureurs, que les modèles d'apprentissage supervisé ont tiré leur épingle du jeu en comparaison à des modèles plus classiques. Ces modèles d'apprentissage supervisé permettent également une offre plus poussée des risques sous-jacents au sein du portefeuille, puisqu'ils prennent en compte divers phénomènes qui ne transparaissent pas toujours lors de l'utilisation de modèles plus classiques.

Summary

With the constant evolution of regulations and technologies the non-life insurance market – and especially the segment of car insurance – has become increasingly competitive. The main goals of these technologies are to improve the insured's experience, by providing him more accurate and tailor-made pricing. In fact, over the past few years, the phenomenon of price segmentation, which aims to provide a price as close as possible to the real risk premium for each customer, has developed significantly. Segmentation is also important at the level of the insurer itself, since it limits the risk of adverse selection and lapse.

To achieve this segmentation, non-life insurers generally rely on a model-frequency approach and on traditional methods such as generalized linear models (GLM). Insurers use the data that the insured has provided to them via their subscription questionnaire to give them a price estimation close to what they have observed historically. By having at its disposal a large amount of historic data, the insurer can determine the fair price to charge the insured for its risk coverage. However, these classic methods, derived from traditional econometrics theory, have their limits: with them, it is tricky to take into account certain non-linear phenomena such as interactions between variables or the multiplication of pricing variables.

One way to overcome these difficulties is to use other types of models than GLMs. But this change comes at a price: a loss of interpretability and traceability. So one needs to set a balance to make sure that the gain does compensate for the sacrifice made. Nevertheless, we observe through insurers that they have an increasing tendency to refine their models, at the risk of losing clear readability of the real effects. This therefore complicates the study of the non-life insurance market. Indeed, each insurer itself distils the variables it uses, and also chooses how to take them into account. A segmented comparison of insurers of a non-life branch is now quite complicated to perform quickly.

The objective of this mémoire is thus to reconstruct the model used by an insurer, by using different customer profiles with their proper quotes. This will allow us to obtain new data on segments for which we do not necessarily have tariffs, and to develop a price comparison on a case-by-case basis. This would make it easier for an insurer to assess its position in the market, and to adapt its offer.

For this, we will study different types of modeling that can be applied to the non-life insurance industry. The traditional market method mentioned above, generalized linear modeling, has been widely used for many years. However, it remains imprecise and is particularly sensitive to the variables and their eventual combinations. In order to improve the predictive performance of this method, insurers are shifting to generalized affine models (GAM) or are making upgrades to GLM, with piecewise polynomial transformations (splines) for example. These methods allow the actuary, through the analysis of trend charts, to better understand the available data; it ultimately leads to a better definition of the boundaries of the classes of tariffs marketed.

To succeed in reconstituting these models, several methods will be used. Thus a first step will be to highlight the complexity of reconstructing a tariff with little information on the variables used or the groupings of modalities that have been made. This will therefore lead to a reflection on how to optimally select the variables that could potentially be used in the development of the model, while avoiding a possible redundancy of a part of the information. Indeed, using too many regressors leads to statistical and computing problems which require a reduction of dimensionality. Since this kind of work is typically done upstream, it is necessary for the reliability of the reconstructed model to match this reduction of dimensionality. Different methods will thus be presented, from statistical methods

such as filters, wrapper and embedded methods, or coefficient regularization methods (Lasso and Ridge), to less traditional methods derived from machine learning.

Once this selection is made, a model will be developed in order to reconstruct the tariff present in the data. To analyze the relevance of this approach and the truthfullness of our reconstituted model, we will compare its results to other types of method. Indeed, with the evolution of computer capacities, alternatives to statistical learning have arisen. Many algorithms have seen their popularity increase in recent years (Random Forest, GBM, neural network, etc.). Their effectiveness is no longer to be proven. This mémoire will therefore try to compare these cutting-edge models to more classic models. Below, the results obtained on the database associated to a single insurer are presented:

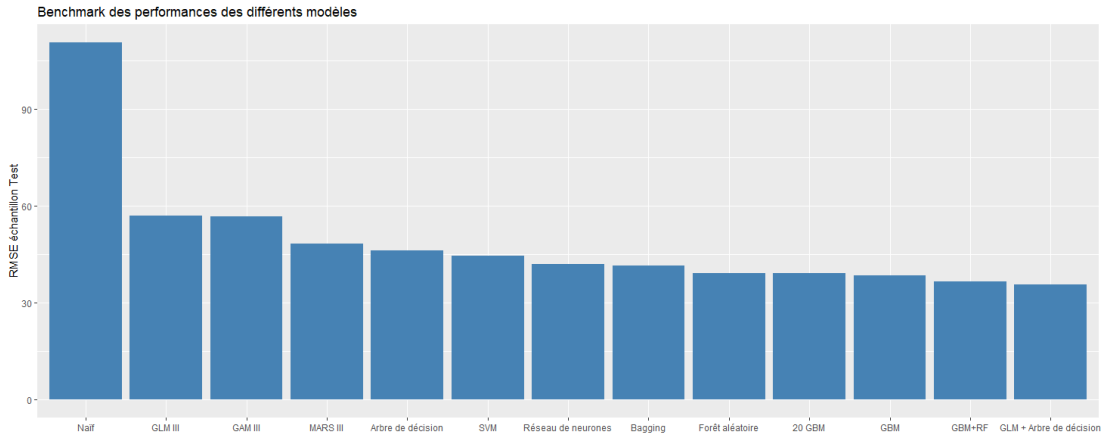


Figure 2 : Performance Benchmark of the different models

The in-depth study of the various models shows that the selection of variables realized upstream is justified. The GLM, even modified/upgraded, struggles to achieve the performance of models resulting from supervised learning (Random Forest, GBM). This probably comes from the fact that these models are more apt to transcribe non-linear relations and cross-effects between two (or more) explanatory variables due to a possible multicollinearity. In addition, these models have the significant advantage of not requiring a pre-selection of the variables in their implementation. The time saved for the actuary is therefore indisputable.

However, these models have the annoying tendency to be seen as "black box" models, which is a short way to say they are difficult to exploit. However, beyond obtaining new data and individual rate comparisons, it is important for the actuary to ensure that the model is relevant; a model should be easy to monitor and to assess. We will therefore focus in this mémoire on the interpretability of machine learning models, and on what they can teach us. Indeed, many tools are now appearing with the aim of "whitewashing" these models, so as to produce information as clear as that of more traditional methods.

However, in a real world, other pricing parameters may intervene (possible promotion, holding of multi-contracts, etc.) and these phenomena are difficult to detect without a priori information. Whether by traditional or machine learning methods, it is complicated to detect a change in price inherent in the data, without knowing when it occurs. This mémoire will therefore try to propose some leads to this type of problem in its last part. Different types of models will thus be compared once again in order to come up with the most efficient model possible.

Finally, it is necessary to precise the fact that the performances obtained throughout this study are not intrinsic but largely linked to the database used. Indeed, since the modeling techniques vary according to the insurers, the different models can see their performance modified according to the technique used. However, it has been confirmed, in a more global framework based on a larger

database, with more insurers, that supervised learning models have performed well compared to more traditional models. These supervised learning models also allow a deeper offering of the underlying risks within the portfolio, since they take into account various phenomena that are not always reflected when using more traditional models.

Sommaire

Résumé 3

Abstract 3

Note de synthèse 4

Summary 7

Préambule 13

I Introduction 15

I.1. Contexte 15

I.1.1 Quelques chiffres *16*
I.1.2 Etat du marché et avenir *18*

I.2. Tarification 18

I.2.1 Décomposition de la prime *19*
I.2.2 Détermination de la prime pure *21*
I.2.3 Présentation de la base de données *22*

II Traitement des données 27

II.1. Analyse préliminaire 28

II.1.1 Analyse des données *29*
II.1.2 Données aberrantes *31*

II.2. Nettoyage des données 32

II.2.1 Création et transformation de variables *33*
II.2.2 Elimination des variables redondantes (corrélation) *36*
II.2.3 Utilisation de la variable réponse *41*

III Sélection des variables pertinentes 44

III.1. Modèles linéaires généralisés 45

III.1.1 Présentation et limites du modèle linéaire *46*
III.1.2 Principe général *48*
III.1.3 Mise en oeuvre dans le cas saturé *50*

III.2. Méthodes classiques de sélection : wrapper, embedded, filtres 52

III.2.1 Filtres *53*
III.2.2 Wrapper méthodes *54*
III.2.3 Embedded Méthodes *56*

III.3. Méthodes Ridge et Lasso 60

III.3.1 Régression ridge *60*
III.3.2 LASSO *63*
III.3.3 Variantes *65*

III.4. Autres méthodes 67

III.4.1 SMuRF méthode et opérateurs proximaux *68*
III.4.2 Friedman H-Statistique *69*
III.4.3 Permutation *72*
III.4.4 ACP *73*

III.5. Bilan sur la sélection de variables 75

IV Reverse Engineering du modèle 77

IV.1. Etat des lieux 77

IV.1.1 Partition des données *78*
IV.1.2 Modèle de référence *79*
IV.1.3 Critère de sélection du modèle et validation croisée *80*

IV.2. Méthodes statistiques 82

IV.2.1 Modèles linéaires généralisés (GLM) *83*
IV.2.2 Modèles additifs généralisés (GAM) *85*
IV.2.3 Régression multivariée par splines adaptatifs (MARS) *86*

IV.3. Algorithmes de Machine Learning 88

IV.3.1 Introduction aux modèles supervisés *89*
IV.3.2 Méthodes avec arbres *90*
IV.3.3 Autres méthodes *95*

IV.4. Benchmark 98

IV.4.1 Méthodes statistiques *98*
IV.4.2 Méthodes de machine learning *99*
IV.4.3 Combinaison statistique et machine learning *100*
IV.4.4 Limites et problématiques liées *101*

V Pour aller plus loin : jugements, interprétation et limites 103

V.1. Intervalles de confiance 104

V.1.1 Partie Statistique : intervalles sur paramètres et sur prédictions *105*
V.1.2 Autres méthodes communes : quantiles, bootstrap et clustering *107*
V.1.3 Prédictions conformes *110*

V.2. Interprétation et confiance dans le modèle 112

V.2.1 Effet des variables *113*
V.2.2 Détection Outlier / Observations influentes *115*
V.2.3 Interprétation de l'outcome *118*
V.2.4 Confiance dans le modèle *124*

V.3. Détection d'un changement de modèle inhérent 127

V.3.1 Création des données *128*
V.3.2 Modélisation basique *128*
V.3.3 Détection *129*
V.3.4 Remarques *131*

Conclusion 133

Annexes **134**

A.1. Opérateurs Proximaux 134
A.2 Lois de la famille exponentielle et correspondance 135
A.3 ADMM 136
A.4 Tuning des hyper-paramètres 137
A.5 Intervalle autour des paramètres (GLM) 139

Bibliographie **140**

Table des figures **1402**

Préambule

"Si les gens ne croient pas que les mathématiques sont simples, c'est simplement parce qu'ils ne réalisent pas combien la vie est compliquée" - John Louis Von Neumann

La tarification est au cœur du métier de l'actuaire non-vie. Elle représente un défi de taille, puisque si un contrat est mal-tarifé, alors c'est toute la compagnie d'assurance qui est impactée. Un contrat trop cher n'attirera que peu de clients et le principe de mutualisation, sur lequel repose une grande partie de l'activité d'assurance, n'aura pas lieu d'être. A contrario, si le prix du contrat est sous-estimé, alors les primes ne suffiront pas à payer les sinistres engendrés par les assurés. L'assureur se verra donc dans l'obligation de revoir sa stratégie, ou de manière plus grave, d'arrêter ses activités.

La tarification d'un contrat automobile repose généralement sur une approche coût-fréquence, dont les valeurs sont directement liées aux caractéristiques de l'assuré et de son véhicule. Néanmoins, au fil du temps les méthodes économétriques se sont complexifiées alors que le nombre de données disponibles et exploitables a fortement augmenté. Cela a bouleversé de manière indéniable les modèles tarifaires développés par les actuaires. Ceux-ci sont devenus de plus en plus raffinés et complexes pour pouvoir coller au mieux à la réalité observée. Dans certains cas on a même pu observer une hyper segmentation de portefeuille, qui n'est toutefois pas une solution miracle. En effet, avec une segmentation trop importante, on perd une fois de plus le principe de mutualisation, primordial en assurance non-vie.

Les modèles statistiques traditionnels se font toutefois de plus en plus concurrencés par les algorithmes d'apprentissage statistique (*statistical learning*), issues de l'informatique à fondements statistiques. Ces derniers permettent d'obtenir de meilleures approximations grâce à une puissance de calcul étendue. Un engouement se porte donc vers ses méthodes depuis le début des années 2010, avec la 'hype' de la data science, qui se veut utiliser ces techniques dans différents milieux, aussi divers que variés. La démocratisation de ces techniques a permis, principalement dans le milieu actuariel et biologique, de mettre aux points différentes méthodes d'application et d'interprétabilité, grâce à de nombreuses recherches et articles sur ce sujet.

Néanmoins, beaucoup de personnes restent encore hermétiques face à ces méthodes, jugées 'black box', c'est-à-dire difficilement exploitables du fait de leurs résultats pratiques limités et l'impossibilité de vérifier ce qui s'est passé entre l'entrée des données et la sortie attendue. Cette réticence a ralenti la transition escomptée vers ces méthodes, pourtant adulées dans le milieu de la data science, du fait de leur efficience dans des compétitions de traitement de données comme Kaggle.

Ce mémoire s'inscrit dans ce contexte de mise en lumière de ces algorithmes et de la réalité de leur applicabilité opérationnelle. En effet, ce mémoire se veut proposer, à travers différentes étapes, un *reverse engineering* poussé et raffiné, qui allie à la fois la puissance des algorithmes de l'apprentissage statistique et l'intelligibilité de l'économétrie classique. Une telle utilisation des algorithmes, quand elle est bien menée, peut être bénéfique à un assureur et lui accorder un véritable avantage vis-à-vis de la concurrence. En effet, en connaissant les modèles de ses concurrents un assureur peut s'adapter rapidement et modifier ses propres tarifs sur les profils qu'il souhaite conquérir.

Comment retrouver le modèle de tarification d'un assureur en s'assurant de son efficacité, mais également de son exploitabilité ?

Ce mémoire abordera le processus de *reverse engineering* à travers une large variété de modélisations possibles. Tout d'abord, une étude classique sera conduite, mais celle-ci nécessite une étape préliminaire. En effet, les précédents rapports ont montré que les algorithmes classiques réagissent mal à une surabondance de variables explicatives. Cela signifie que, parmi toutes les données exploitables liées à un profil client, certaines ne sont pas pertinentes ou sont redondantes dans la tarification. C'est pourquoi ce mémoire s'attardera premièrement sur le processus de sélection des variables explicatives. Il s'agit d'un enjeu majeur pour l'actuaire, puisque pour obtenir un modèle performant et intelligible, il est nécessaire de maintenir un nombre de variables et de modalités décents. Cela est directement lié au phénomène d'hyper segmentation mentionné plus haut.

Or dans un processus de *reverse engineering*, les données ne présentent que les caractéristiques d'un client et le tarif appliqué en conséquence. Aucune indication n'est donnée quant à l'utilisation ou non de telles ou telle variable. Un enjeu majeur sera donc de s'assurer de la bonne sélection des variables. Une autre problématique est encore soulevée lors de la fusion des différentes modalités au sein d'une variable (par exemple les codes postaux) ou de la création de nouvelles variables (transformer une variable numérique comme l'âge en intervalles par exemple). Ceci requiert une véritable étude poussée de la base de données et ne peut s'improviser. Néanmoins, comme mentionné au début de ce paragraphe, les données utilisées ne présentent aucun indice quant à telle ou telle utilisation, et cela représente une véritable gêne à l'utilisation de méthodes classiques dans le *reverse engineering.*

Dans un second temps, différentes alternatives seront proposées, principalement basées sur l'apprentissage statistique. Une comparaison des méthodes est ainsi présentée en partie IV. La pertinence de la sélection des variables réalisée en III sera ainsi mise en exergue. Afin de comparer les performances opérationnelles de ces différentes approches, une base de données réduite sera mise de côté et utilisée pour assurer la mission d'échantillon test.

Comme l'interprétation des modèles d'apprentissage automatique peut sembler floue pour des néophytes, ce mémoire s'évertuera à proposer également une pléiade d'outils pour mieux comprendre leurs aboutissants et le degré de confiance à leur accorder. En effet, quel est l'intérêt de disposer d'un modèle performant si on ne peut pas lui faire confiance ? Il s'agit d'une réelle problématique soulevée récemment dans le monde de la data science, puisque si la question du comment trouve sa solution dans la programmation, celle du pourquoi demeure ouverte.

Finalement, d'autres questionnements seront également évoqués, comme la pertinence du choix de la fenêtre de temps de certaines données ; en effet, alors que les promotions et les soldes éphémères se multiplient, comment s'assurer que notre tarif prend bien en compte ce phénomène ? Ainsi, un modèle de *reverse engineering* se doit de faire la part des choses entre véritable tarif et tarif promotionnel. Cela s'avère toutefois complexe et se rapproche fortement de la problématique de franchissement de seuil que l'on peut retrouver dans divers domaines actuariels, comme la théorie de la ruine ou la finance mathématique.

La première partie de ce mémoire propose donc une introduction au processus de tarification automobile et place le contexte dans lequel elle s'inscrit. Une seconde partie permet d'introduire la base de données et les modifications qui ont été effectuées. Dans sa troisième partie, ce mémoire propose différentes alternatives de sélections de variables, dont les résultats se verront comparés et évalués. Puis la quatrième partie rentrera réellement dans le processus de *reverse engineering* puisqu'elle présentera succinctement les méthodes de modélisation les plus répandues, avant de proposer un benchmark de ces différentes méthodes. Finalement une dernière partie traitera des différentes limites et jugements à apposer à la fois aux modèles d'apprentissage supervisés, mais aussi au processus de *reverse engineering* en lui-même.

I Introduction

"A problem well-stated is a problem half-solved" - Charles Kettering

Dans un premier temps, il est indispensable de définir dans quel contexte s'opère l'assurance automobile. En effet, ce mémoire repose sur l'utilisation d'une base d'assurés automobile et les résultats obtenus ne sont pas forcément transposables dans tous les secteurs de la non-vie (multirisques habitation, accidents, maladie, juridique…).

Dans ce mémoire, l'attention sera dirigée vers le contrat d'assurance automobile. Il s'agit d'un contrat en vertu duquel, moyennant le paiement d'une prime fixe ou variable, un assureur s'engage envers son assuré à fournir une prestation définie à l'avance dans le contrat, dans le cas où interviendrait un évènement incertain que l'assuré (ou le bénéficiaire) a intérêt à ne pas voir se réaliser.

Puisque le but de ce mémoire est également d'apporter une ouverture sur la compétitivité d'un assureur, il est important de rappeler pourquoi il est si difficile d'obtenir des données fiables et facilement exploitables, justifiant ainsi la démarche de *reverse engineering*.

Il est tout autant primordial que le lecteur s'assure de sa bonne compréhension du processus de tarification opéré par les différents acteurs de l'assurance. En effet, si l'actuaire est chargé de calculer une prime faisant en sorte que la compagnie ait une espérance nulle ou quasi-nulle (la compétition féroce obligeant les assureurs à ne pas appliquer une trop large marge de profit), cette prime n'est pas celle payée par l'assuré. En effet, la prime calculée par l'actuaire, qualifiée de 'prime pure' doit passer entre les mains du commercial, qui doit non seulement la 'charger', c'est-à-dire ajouter les différents frais fixes liés à l'exploitation du contrat, mais doit aussi s'assurer que le montant final de la prime n'est pas non plus 'hors du marché'. C'est alors un jeu d'équilibriste qui prend place. En effet, pour devenir compétitif, un assureur aura tendance à vouloir baisser ses tarifs, et donc à diminuer la valeur de la 'prime pure'. Cela engendre donc une espérance négative, et donc un contrat déficitaire.

La tâche de tarification est complexe et doit être prise dans son ensemble. En effet, si un contrat est déficitaire, alors cette perte peut être compensée par un autre contrat fortement bénéficiaire, et ainsi conduire à un bilan positif de l'assureur. Les assureurs cherchent donc de plus en plus à fidéliser leur clientèle et à les encourager vers la politique de multi-contrats. Via des prix d'appel bas pour une assurance donnée, les assureurs cherchent ensuite à leur faire souscrire une assurance ou des options plus prolifiques pour eux.

L'optique d'étude de marché est alors encore plus importante puisqu'elle conditionne non plus une seule branche d'assurance, mais une myriade de produits additionnels. Savoir où se positionner, et comment, est donc plus important que jamais pour un assureur.

I.1. Contexte

L'histoire de l'assurance automobile Belge est assez similaire à celle Française. Dès le début des années 1900, le nombre de véhicules circulant sur les routes en Belgique augmente progressivement. Cela a conduit inexorablement à une hausse des accidents, qui a poussé les autorités à instaurer une réglementation pour encadrer le secteur automobile, protégeant par ailleurs également les victimes. Une première ébauche de Fonds Commun de Garantie (similaire au FGA

français, désormais FGAO : Fonds de Garantie des Assurances Obligatoires de dommage) voit le jour en 1906 mais le gouvernement n'y donne pas suite.

Ce n'est qu'en 1937 qu'une première loi établit une première protection légale aux victimes de la route. Le Fonds Commun de Garantie Belge (FGCB) ne verra le jour qu'en 1957 et vise à régler les indemnités dues aux victimes d'accidents corporels causés par des véhicules dont les conducteurs demeurent inconnus, ou non assurés et insolvables. La création de ce fonds succède à la législation rendant la Responsabilité Civile (RC) obligatoire, loi promulguée en 1956. Cette loi, revisitée au fil des ans, n'adoptera sa formulation définitive qu'en 1989, créant ainsi la loi du 21 novembre 1989, encore en application à l'heure actuelle. Un arrêté Royal en 1992 viendra fixer les critères requis concernant le contrat-type d'assurance obligatoire de la responsabilité en matière de véhicules automoteurs ; cet arrêté sera par la suite modifié pour suivre diverses directives européennes et prendre la forme qu'on lui connaît aujourd'hui.

Contrairement à la France qui adoptera en 1986 l'adoption de la vignette d'assurance, les Belges jugent cette mesure inutile et coûteuse, préférant à ceci l'amélioration d'un système informatique centralisant toutes ses informations via la plaque d'immatriculation du véhicule. Aucun contrôle n'est donc opéré en Belgique concernant la détention ou non d'un contrat de RC automobile, exception faite de contrôles inopinés opérés par des agents de police (contrôle des papiers du véhicule avec présentation de la 'carte verte'). Malgré l'obligation légale de disposer à minima d'une assurance RC, aujourd'hui près de 100 000 véhicules belges rouleraient sans assurance, soit environ 1,7% des véhicules du parc automobile belge.

Cela laisse néanmoins près de 6 millions d'automobiles à couvrir, et tout autant de potentiels assurés. C'est pourquoi les assureurs sont particulièrement attirés par ce marché, car même si la RC est peu profitable pour un assureur, elle est la porte d'entrée pour vendre d'autres contrats au souscripteur. Ce secteur s'est pourtant vite saturé du fait de la faible évolution du parc automobile. Il y a donc nécessité d'appliquer une bonne tarification automobile, car la compétition entre les assureurs est très acharnée et les petits acteurs du marché ont des difficultés à tirer leur épingle du jeu, faute de volume assez important pour pouvoir fournir des prix attractifs.

On trouve en Belgique trois grands types d'assurance automobile : la RC, la mini Omnium et la full Omnium. La garantie RC automobile a pour but de réparer les dommages que l'assuré pourrait causer à autrui avec sa voiture, et à les indemniser, quels que soient les dommages causés (matériels et/ou corporels). Néanmoins cela ne couvre pas les dommages liés au conducteur ou à son véhicule. Pour cela l'assuré doit souscrire des assurances optionnelles (assurance conducteur ou assurance Omnium). La mini omnium (ou omnium partielle) est une assurance couvrant les dégâts causés par l'assuré à son propre véhicule et comprend généralement une garantie incendie, vol, bris de vitre et forces de la nature. Lorsque l'on y ajoute l'option dégâts matériels, la garantie mini omnium devient alors une garantie full omnium, généralement 3 fois plus chère que la mini omnium pour un même véhicule.

I.1.1 Quelques chiffres

Le marché de la non-vie couvre de nombreux risques : assurances accidents, maladies, véhicules (auto, train, avion, bateaux…), maison, responsabilités civiles… La plupart de ces activités d'assurance sont excédentaires, exceptées deux d'entre elles : les garanties catastrophe naturelle et responsabilité civile automobile. Attardons-nous plus en détail sur le marché de la Responsabilité Civile automobile en Belgique. Comme mentionné plus tôt, le segment RC est souvent peu profitable pour les assureurs mais sert plus de produit d'appel. Cela se vérifie dans les chiffres.

En 2018, les différents assureurs belges collectaient des primes pour l'assurance RC automobile à hauteur de 2250.5 millions d'euros, chiffre assez stable au cours des 5 dernières années. La charge avérée de sinistre s'élevait elle à 59.8% des primes, amenant pour la première fois en cinq ans le ratio en-dessous des 60%. Couplée aux 23.8% des frais généraux et aux 15.9% de frais liés aux commissions, cela conduit le ratio combiné sur l'année 2018 à se tenir à 99.5%, passant ainsi sous la barre des 100% pour la première fois depuis 2012. Cela signifie que le segment RC a globalement été profitable pour les assureurs en 2018.

Néanmoins, ce n'est généralement pas le cas. En effet, en France, le segment RC est déficitaire sur les résultats techniques des organismes d'assurance non-vie (-29%). Mais même si la RC reste globalement peu intéressante pour les assureurs, ces derniers compensent sur d'autres catégories d'assurance, qui sont cette fois-ci optionnelles.

On mentionnera notamment la branche corps de véhicule, qui a collecté en 2018 1469.5 millions d'euros en Belgique (avec un montant collecté en constante augmentation), qui dispose d'un ratio combiné de 90 à 95% grâce à une charge avérée de sinistre à hauteur de 50 à 55% des primes.

La branche garantie juridique automobile doit également être mentionnée, car même si son volume de primes est moins important (333 millions d'euros en 2018, mais également en constante augmentation), son ratio combiné est très bon, atteignant généralement les 85%. On notera également un taux de commission moyen plus important (25%), transcrivant une réelle motivation des assureurs d'encourager la souscription d'une telle assurance.

Du point de vue des assurés, on trouve qu'en Belgique moins de 60% des automobilistes déclarent un sinistre chaque année. Sinistre dont le montant moyen s'élève à 3513€. La prime moyenne d'une assurance RC auto, elle, s'élève à 318€ et les jeunes conducteurs voient généralement leur montant d'assurance augmenter de 10 à 20% suite à leur inexpérience. Une forte disparité régionale des primes est également remarquée, puisque les Flamands et les Bruxellois ont tendance à payer plus cher leur assurance automobile. On note également qu'en Belgique, de moins en moins d'assureurs utilisent une échelle de bonus-malus, contrairement à beaucoup de ses pays voisins, particulièrement friands de cette fonctionnalité.

Il est donc clair que la branche automobile est importante pour les assureurs non-vie, puisqu'elle leur permet de dégager des bénéfices importants. Il permet aussi une certaine fidélisation de la clientèle, puisque l'assuré aura tendance à faire jouer le multi-contrat pour obtenir des réductions sur ses différents contrats d'assurance. Ainsi chez certains assureurs, cumuler une assurance automobile couplée avec une assurance habitation ou une GAV (Garantie Accidents de la Vie) permet d'obtenir une remise de quelques pourcents sur son contrat automobile.

Cela encourage donc l'assuré à chercher le meilleur prix pour son assurance. L'assurance automobile représente un poste de dépense important pour les ménages à faibles revenus. Certains intermédiaires l'ont bien compris et de nombreuses plateformes de comparaison en ligne, comme assurances.be, topcompare.be ou mefirst.be, sont apparues sur internet. Cela a créé un mini-séisme dans le secteur de l'assurance qui a dû rapidement s'adapter. Les plus prompts à réagir à cette révolution digitale ont été, sur le marché belge, Axa et Yuzzu, qui font partie de la courte liste des assureurs belges proposant de réaliser un devis ou une souscription en ligne.

Le marché digital belge reste néanmoins encore balbutiant, contrairement au marché français dont presque tous les acteurs se sont adaptés à cette nouvelle technique de démarchage. En effet, le CCM Benchmark estimait en 2017 que 20% des internautes français avaient déjà souscrit à une assurance en ligne, dont 52% pour de l'assurance auto-moto.

I.1.2 Etat du marché et avenir

Assez traditionnel jusque dans les années 2000, le marché de l'assurance automobile doit maintenant faire face à de nouveaux besoins, notamment avec les chauffeurs de plateformes telles que Uber ou Lyft, ou les protections en cas de covoiturage. Avec la démocratisation de nouvelles méthodes de déplacement, le milieu de l'assurance doit constamment s'adapter pour satisfaire et répondre aux besoins de ses assurés. Des partenariats se créent ainsi, comme celui d'Axa avec Uber (2018) ou d'Axa avec Blablacar (2015, puis BlaBlaSure en 2018). Certains experts parlent même du futur marché des véhicules autonomes. Celui-ci devrait représenter 81 milliards d'euros d'ici à 2025, et remet en question pas mal d'acquis. En effet, les conducteurs ne seront plus automatiquement considérés comme responsables du sinistre : qui sera alors nommé responsable ? Le constructeur, l'équipementier ou l'exploitant du logiciel ? Ainsi, à terme, un changement radical des modes de consommation devrait s'opérer : les véhicules personnels pourraient laisser place au développement de services de transport (taxi) reposant sur une flotte de véhicules autonomes, transformant l'industrie automobile en une industrie de services. Cela soulève de nouveaux défis et des nouvelles problématiques chez les assureurs non-vie.

Mais ce n'est pas le seul virage que doit prendre l'assurance automobile. Les assureurs doivent également chercher à prendre en compte les nouvelles données disponibles, notamment avec le "quantified self' (ensemble de pratiques consistant à analyser son activité physique ou son mode de vie : poids, tension, calories consommées, rythme cardiaque...). En assurance auto, on parle de télématique embarquée. Le 'Pay-as-you-drive' ou l'assurance au kilomètre, reste malgré tout assez peu répandu en Europe. Rendue obligatoire aux Emirats Arabes Unis, cette assurance tarde à convaincre les occidentaux. Elle permet pourtant une tarification sur-mesure du contrat selon l'utilisation réelle du véhicule, déterminée par un système de GPS embarqué dans la voiture de l'assuré. L'utilisation d'une telle méthode permettrait aux assureurs de révolutionner leur système bonus-malus, en évaluant en temps réel le comportement de l'assuré, au lieu de se baser sur des données historiques. D'autres utilisations du boîtier télématique sont également envisagées, comme des détecteurs de freinage d'urgence ou de vitesse moyenne, qui impacteraient immédiatement la prime mensuelle de l'assuré.

Malgré la tentative d'Axa en 2005, qui essuya un premier refus de la CNIL (Commission Nationale de l'Informatique et des Libertés) avant de lancer ses premiers produits de télématique embarquée en 2009, les français demeurent frileux face à ce nouvel outil de tarification, qui, dans les faits, permet de faire économiser jusqu'à 100€ aux bons conducteurs, principalement car les assurés craignent l'usage abusif qui pourrait être fait de leurs données personnelles. Les belges sont également récalcitrants vis-à-vis de cette nouvelle technique.

Une autre question pourrait se poser : avec l'intérêt porté vers les transports durables et écologiques, comme les transports en commun ou l'utilisation du vélo, le marché n'est-il pas condamné ? Les chiffres ne montrent pour l'instant pas d'avancée dans ce sens, ni de ralentissement du secteur puisque, par exemple, en Belgique le nombre de voiture du parc automobile national évolue d'environ +5% par an depuis 2016. Il semblerait donc que l'assurance automobile ait encore de beaux jours devant elle ; et que par extension, la compétition entre les différents assureurs n'est pas près de perdre sa vigueur.

I.2. Tarification

Les contrats assurances automobiles ont la particularité de s'appliquer à ce que la terminologie de Durkheim qualifie de 'faits sociaux'. En effet, les accidents qu'ils couvrent ne peuvent pas être prédits à un niveau individuel, mais peuvent toutefois être approximés à travers un modèle collectif. En effet, par effet d'étoffement, plus la population est grande moins l'aléatoire est présent,

créant ainsi une certaine régularité. Une individualisation du tarif est alors difficilement envisageable, car hautement volatile.

L'étape de tarification est donc primordiale à toute création de contrats. L'établissement d'une prime 'juste' est essentielle pour s'assurer qu'un portefeuille soit équilibré, surtout en tarification automobile. En effet, si un contrat est mal tarifé alors l'assureur peut se trouver en défaut. Mais même sans cela, il est important de tenir compte du fait qu'il y ait des 'bons' risques et des 'mauvais' risque dans notre portefeuille (entité contenant des contrats d'assurance). En effet, se contenter de faire la moyenne des montants des sinistres multipliée par la fréquence moyenne d'occurrence d'un sinistre, puis de partager le montant obtenu entre tous les assurés n'est pas indiqué ; le tarif produit est alors :

- Plus avantageux pour les 'mauvais' conducteurs, sujets à plus d'accidents que les 'bons' conducteurs, puisque la part de risque supplémentaire qu'ils portent normalement est partagé sur les 'bons' assurés
- Néfaste pour les 'bons' conducteurs qui doivent alors payer autant qu'un 'mauvais' assuré alors que leur espérance de coût est bien plus faible

Un phénomène d'antisélection est alors observé, où les 'mauvais' risques restent dans le portefeuille (puisqu'ils payeront plus chers chez un assureur proposant cette segmentation entre 'bons' et 'mauvais') et où les 'bons' risques s'en vont chez la concurrence.

De la même manière, une segmentation trop forte n'est pas une solution. En effet, on remarque souvent dans les cas d'hypersegmentation que les tarifs ne sont pas assez robustes ; cela veut dire que l'on ne dispose pas de suffisamment de données pour pouvoir construire un tarif répondant bien au risque porté par le segment, qui peut être très variable d'une année à l'autre.

Il est donc primordial de construire intelligemment son tarif pour éviter ce genre de phénomène. Pour cela il faut étudier la corrélation entre fréquence et sévérité du sinistre et les caractéristiques de l'assuré pour parvenir à un tarif plus juste. C'est pourquoi l'assureur demande toujours des renseignements à l'assuré sur sa personne et sur son véhicule lors de l'établissement du contrat d'assurance. Un jeune assuré avec une voiture puissante, plus susceptible d'avoir un accident, payera donc plus cher qu'un assuré avec une petite voiture et plusieurs années d'assurance derrière lui.

Le but de la tarification est donc de séparer les assurés en différentes classes, de telles manière qu'au sein de ces catégories les risques puissent être considérés comme homogènes. Comme indiqué par Ohlsson, la pratique du marché consiste à discrétiser les variables continues afin d'identifier des classes de risque homogène. Il faut toutefois garder en tête qu'une segmentation des risques trop fine, à l'échelle même de l'individu, n'est pas suffisamment robuste pour être effective.

I.2.1 Décomposition de la prime

La plupart des compagnies d'assurance établissent leur tarification automobile en se basant sur des modèles économétriques. Ces modèles lient le phénomène de sinistralité avec divers facteurs de risques. Ces facteurs de risques sont apportés par le signataire du contrat et concernent le bénéficiaire de l'assurance, son véhicule et les modalités du contrat. Des exemples de modalités seront donnés dans la section II.

Le principe de la prime pure chargée est la méthodologie de tarification dominante sur le marché. Il est utilisé autant dans le domaine IARD qu'en assurance maladie. La prime pure représente la meilleure estimation de la prime à appliquer à un assuré, c'est-à-dire la perte moyenne attendue. Cela reste pourtant, malgré sa facilité apparente, une opération complexe. En effet, de nombreuses

difficultés affèrent à cette tâche comme la multiplicité des facteurs de risque, la répartition de la distribution à queue épaisse (certains sinistres peuvent coûter très cher en assurance automobile, surtout lorsque l'on traite un accident conduisant à l'incapacité ou la mort d'une tierce-personne), relation de dépendance non-linéaire, influence temporelle (effet d'inflation sur des sinistres plus anciens dans la base de données), données manquantes…

D'autres paramètres peuvent également jouer, comme le choix entre tarification a priori ou a posteriori : la tarification a posteriori dispose d'informations additionnelles, comme la sinistralité réelle inhérente à l'assuré. En assurance auto, la tarification a posteriori transparaît notamment dans l'utilisation de coefficient bonus-malus, ou de prime à l'ancienneté. La tarification a priori concerne elle un assuré sur lequel on n'aurait aucune information sur la conduite : on doit donc le juger par défaut comme ni bon ni mauvais conducteur. Néanmoins, que ce soit en tarification a priori ou a posteriori, l'idée est de séparer les différents contrats et individus de la base de données en plusieurs classes ou catégories de sorte qu'en leur sein les risques puissent être considérés comme homogènes. L'hétérogénéité pose en effet des problèmes en assurance, notamment à cause du phénomène d'antisélection qui a été précisé plus haut.

Une fois ces distinctions prisent en considération, on peut déterminer une prime pure en prenant l'espérance de la distribution de la sinistralité. A cette prime pure est rajoutée un chargement dû directement à la théorie de la ruine ; ainsi au sein d'une mutualité de risque, il existe une volatilité résiduelle de sa sinistralité. En effet, la prime pure n'est qu'une espérance ; la réalité de la perte peut donc être bien supérieure, comme inférieure. Si une compagnie encaisse durant plusieurs années consécutives une perte supérieure à celle estimée, alors les primes pures ne peuvent couvrir le manque de fonds de l'assureur. En tarifant les contrats au niveau de la prime pure (et en supposant une distribution des pertes symétriques), l'assureur perd de l'argent une année sur deux. En l'absence de fonds propres, cette situation conduit immédiatement à la faillite. Le lecteur intéressé par plus de détails pourra se référer au chapitre 4 du livre de Denuit et Charpentier.

Différentes méthodes de chargement existent, mais aucune n'a à ce jour supplanté les autres, car chacune dispose de ses forces et de ses faiblesses :

- Chargement proportionnel à la prime pure. Le coefficient de proportionnalité reflète la volatilité du risque
- Chargement dépendant de l'écart-type des pertes, et donc de la sinistralité ; légèrement problématique puisque le chargement de sécurité dépend alors des cas de gains (perte réelle inférieure à la prime pure)
- Chargement dépendant d'un certain quantile des pertes ; cela permet de garantir que la prime sera suffisante dans un nombre de cas déterminé à l'avance

La détermination de la prime pure est donc le point central de la tarification automobile. Elle se calcule généralement avec l'approche fréquence-sévérité, considérant ainsi pour chaque type contrat le nombre et le montant individuel de chaque sinistre. Néanmoins, avec l'affluence de données historiques, il est de plus en plus fréquent de voir cette prime être calculée sur des groupes de plus en plus restreints d'assurés.

Pourtant la méthodologie classique fait appel à la notion de modèle collectif. Avec les notations traditionnelles, on définit ainsi une variable aléatoire discrète N, variable de comptage, représentant le nombre de sinistres survenus durant une période (généralement un an). Le montant du $k^{ème}$ sinistre, sa sévérité, est également une variable aléatoire, notée W_k. On a alors :

$$S = \sum_{k=1}^{N} W_k$$

Où S est le risque transmis, soit la charge totale de l'assureur. S suit une loi composée.

Dans des hypothèses très simples, souvent appliquées par les assureurs, où l'on considère les $(W_k)_{k=1,\dots,N}$ une suite de variables aléatoires indépendantes de N, mutuellement indépendantes et identiquement distribuées, on peut montrer que :

$$E(S) = E(N)E(W_1)$$

La prime pure est alors égale au produit entre la fréquence moyenne $E(N)$ et le coût moyen engendré par un sinistre $E(W_1)$.

Déterminer les lois de N et de W est un processus complexe. Comme Dugas et collab. le soulignent dans leur article de 2003 '*Statistical Learning Algorithms Applied to Automobile Insurance Ratemaking*', de nombreuses difficultés viennent alourdir l'estimation comme : la pluralité des facteurs de risque, la corrélation éventuelle entre ces derniers, la relation de dépendance entre facteurs et variable réponse non linéaire, l'influence temporelle, les enjeux commerciaux, les données parfois manquantes ou erronées… Cela vient entacher la pertinence des résultats et force l'actuaire à prendre des décisions et à porter un jugement sur sa méthodologie.

I.2.2 Détermination de la prime pure

Pour déterminer la prime pure, il faut donc trouver une expression de N et de W. En effet, il n'y a pas de raisons particulières de penser que les variables expliquant la fréquence suivront la même loi que celles décrivant la sévérité. Il est donc plus pertinent de les modéliser séparément. Plusieurs modélisations sont envisageables pour chacune des variables. Pour N, du fait qu'il s'agisse d'une variable de comptage, on se restreint généralement à l'utilisation de trois lois :

- La loi Binomiale, où on a $Var(N) < E(N)$, on a donc une sous-dispersion
- La loi de Poisson, où on a , on a $Var(N) = E(N)$, ce qui amène une équi-dispersion
- La loi Binomiale négative, où on a $Var(N) > E(N)$, on a donc sur-dispersion

D'autres méthodes adaptées de celles-ci peuvent également être utilisées, comme la ZIP (Zero-Inflated Poisson), la ZINB (Zero-Inflated Negative Binomial) ou une distribution Tweedie. Ces méthodes permettent de prendre en compte la prépondérance de 0 dans la fréquence des sinistres présente dans les données. En effet (et fort heureusement pour les compagnies d'assurance), seuls quelques contrats sont effectivement sinistrés au cours d'une année, et constituent donc une faible part des contrats présents dans le portefeuille.

Pour la variable aléatoire W, la tâche est plus complexe. Il faut d'abord séparer les sinistres attritionnels des sinistres graves (coût dépassant un certain seuil). La sinistralité attritionnelle est souvent modélisée par un modèle linéaire généralisé, qui sera développé plus en détail dans la suite de ce mémoire. Le coût d'un sinistre grave est lui, généralement moins finement modélisé en raison du faible nombre de données disponibles d'une part, et de la faible efficience prédictive des variables explicatives d'autres part. Ainsi, certains phénomènes de sinistralité non-vie suivent parfois une distribution de probabilité à queue épaisse (en comparaison, la loi normale a une distribution de probabilité à queue légère). Les sinistres de la garantie Responsabilité Civile corporelle en sont des exemples typiques ; ces sinistres peuvent parfois se chiffrer en millions d'euros et affectent ainsi très fortement la distribution sous-jacente.

Il est donc préférable d'écrêter les sinistres extrêmes, selon la théorie des valeurs extrêmes, dans un modèle de tarification. On se retrouve ainsi à modéliser deux types de sinistralité différentes

au sein de la variable W. Différentes techniques peuvent alors être appliquées, dont notamment celles développées dans ce mémoire.

En effet, la modélisation d'un modèle de fréquence et d'un modèle de sinistre s'inscrivent dans le cadre des problèmes de régression. Les modèles traditionnels répondant à ce type de problématique sont historiquement les régressions linéaires, qui se sont progressivement améliorées et adaptées à travers les régressions linéaires multiples puis les modèles linéaires généralisés (GLM).

Ces modèles sont populaires dans le milieu actuariel car ils sont simples à interpréter et apportent de manière très naturelle une quantification de l'impact de chaque variable explicative sur la variable réponse. De plus ils s'adaptent facilement à différentes hypothèses de loi (et particulièrement bien aux distribution de la famille exponentielle comme les lois binomiales, de Pareto, exponentielle…).

Ces modèles rencontrent toutefois plusieurs limites, notamment dans le fait que les variables explicatives utilisées sont rarement décorrélées entre elles, causant ainsi des termes d'interaction, qui seront évoqués plus en détails dans la partie III. De plus les modèles économétriques classiques sont incapables de modéliser des phénomènes non-monotones.

D'autres difficultés entrent également en jeu mais ne seront pas développées plus avant dans le cadre de ce mémoire, car non lié directement à la problématique exposée dans ce dernier. On peut par exemple citer :

- L'exposition, soit la durée de présence dans le portefeuille du contrat, relativisée par rapport à la durée d'observation. En effet, plus un contrat est récent, moins il est susceptible d'être l'objet d'un sinistre durant la période d'observation. Cela impacte donc de manière plus générale la probabilité de survenance d'un sinistre. Une première méthode naïve pourrait consister à pondérer chaque observation par son exposition, mais cela introduit un biais en faveur des contrats les plus exposés. Il s'agit donc d'une véritable problématique qui est encore peu développée actuellement.
- L'impact temporel. Lorsque l'étude d'un portefeuille se déroule sur plusieurs années, alors des paramètres comme l'inflation influent sur la sévérité des sinistres.
- Les classes fortement déséquilibrées. Cela apparaît surtout lorsque l'on essaye de trouver un modèle pour la variable N, puisque les polices ne présentant aucun sinistre sur l'ensemble de la période d'observation sont largement majoritaires. Or les modèles traditionnels, qui reposent sur une maximisation de la vraisemblance, sont peu robustes en présence de classes fortement minoritaires, la vraisemblance étant une quantité moyenne. Cet enjeu est particulièrement important lorsque l'on se place dans une optique de prédiction

I.2.3 Présentation de la base de données

Cette étude repose sur la base de données d'un agrégateur d'assurances belge. A travers des requêtes formulées par un particulier, l'agrégateur recueille ses caractéristiques et lui fournit en réponse un tarif correspondant pour chaque assureur présent dans sa base de données. Toutes les simulations présentes dans la base de données sont réalisées sur des véhicules appartenant à des particuliers. Les véhicules de flottes, les camions ou les véhicules spécialisés et de transports ne sont donc pas concernés ici.

Le prospect cherchant une comparaison entre les offres du marché se connecte donc sur le site internet de l'agrégateur et répondra à un questionnaire structuré se déroulant selon une séquence de questions de type :

- Le conducteur est-il un homme ou une femme ?
- Quelle est la puissance de la voiture ?

- Quel est le type de véhicule ?
- Etc.

A la fin du questionnaire, l'agrégateur fournira ainsi un tarif conseillé vu le type de profil. On remarquera néanmoins que le maillage est suffisamment fin pour que deux prospects aux caractéristiques similaires sans pour autant être identiques ne se voient jamais proposer exactement le même tarif ; le processus est donc loin d'être standardisé et est complètement personnalisé. Ainsi, deux assurés qui diffèrent seulement par le type de résiliation du précédent contrat ne se verront pas proposer exactement le même tarif, même si cette variable n'intervient pas à proprement parler dans le modèle de l'assureur : cela est lié au fait que l'agrégateur demeure complètement opaque sur son traitement des données.

Trois garanties sont présentées sur le site : responsabilité civile, mini omnium et la full omnium ; garanties présentées plus en détail dans la partie I.1. de ce mémoire. Dans cette étude, seule la garantie mini omnium sera développée. Cette garantie a été privilégiée car elle est plus complète que la garantie Responsabilité Civile, qui ne prend pas forcément en compte les caractéristiques du véhicule, et permet une étude plus simple que la garantie full omnium, où de multiples combinaisons de franchise sont présentes, et donc diminuent le nombre de profils différents éventuels. En effet certains assureurs ne présentent qu'un certain type de franchise, ce qui rend la comparaison plus complexe.

Le cheminement suivi s'applique toutefois aux autres types de garantie et fournit des résultats similaires à ceux trouvés pour la mini omnium. Le travail sera conduit sur des données extraites sur 1 an, du 6 janvier 2019 au 5 janvier 2020 pour la catégorie mini omnium. Les données pour les assureurs proposant une franchise ont été enlevées car elles ne représentaient que deux assureurs, et donc une minorité des offres disponibles. Notre base de données comprend alors 659 183 lignes, qui peuvent se résumer à 98 188 profils, puisqu'un même profil se voit souvent proposé plusieurs offres de différents assureurs, avec dans notre cas, différents niveaux de dépréciations (valeur réelle ou valeur agréée).

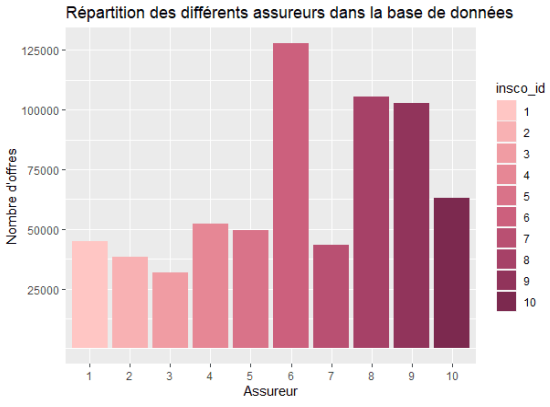


Figure 3: Répartition des différents assureurs dans la BDD

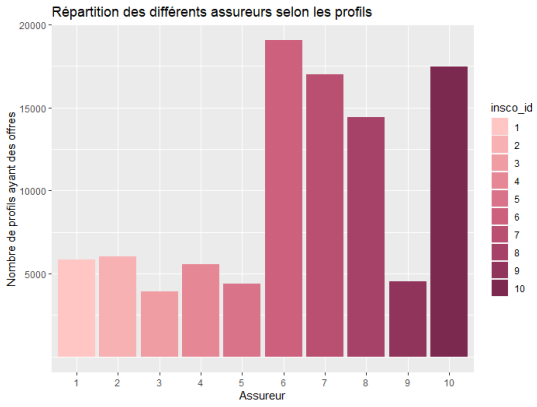


Figure 4: Réparition des assureurs selon le nombre de profils

Néanmoins, l'agrégateur n'est pas totalement transparent, et l'on ne sait donc pas si les prix fournis aux prospects sont issus de l'encodage direct des modèles des assureurs, ou si le tarif est récupéré directement via un API disponible sur le site de l'assureur. En effet, les assureurs belges restent assez réfractaires à la souscription de contrats en ligne ; peu d'assureurs rendent disponible la possibilité d'effectuer un devis directement via leur site web. La première hypothèse de modèles directement encodés est donc privilégiée.

Dans un souci de conformité au Règlement Général de Protection des Données (RGPD, 2016), il est impossible de tracer l'assureur puisqu'il apparaît sous un identifiant qui est généré aléatoirement

par le site pour chaque extraction de données. De la même manière, les profils clients ne comportent aucune donnée permettant de remonter jusqu'à eux, le maillage s'arrêtant au ZIP code et le numéro d'immatriculation étant masqué.

Le secteur assurantiel est impacté directement par cette nouvelle directive, puisqu'ils possèdent une grande quantité de données. Les données historiques sont souvent primordiales pour estimer le coût/la fréquence des sinistres, et pour ensuite pouvoir tarifer de manière légitime leurs assurés. Désormais, avec le nouveau règlement, les assureurs doivent non seulement protéger au maximum les données personnelles de leurs clients mais ils doivent également tenir un registre interne du traitement de données (consignant l'ensemble des traitements ayant eu lieu sur les données personnelles des prospects et clients) et s'engager à ne traiter que celles réellement nécessaires à leur activité. En effet, si les données ont été collectées exclusivement afin de payer les allocations retraite, alors elles ne doivent pas être utilisées pour faire de la prospection commerciale (cf Malakoff-Médéric). En cas de non-conformité, de lourdes sanctions peuvent être infligées, de l'ordre de la centaine de milliers d'euros, et pouvant atteindre jusqu'à 20 millions d'euros.

Un autre changement majeur touche à la durée de conservation des données personnelles; si, postérieurement, un courtier en assurance pouvait conserver les données personnelles de leurs clients sans limitation dans le temps, désormais, les compagnies d'assurance doivent systématiquement effacer les données de leurs prospects dès lors que ces derniers n'ont pas signé de contrat dans les 3 ans suivant un dernier contrat. Pour les assureurs, cette durée s'étend jusqu'à 10 ans après la fin de la relation contractuelle (sauf si le prospect demande explicitement la suppression totale de leurs données personnelles).

Dans une optique de protection des données personnelles, les données stockées doivent être anonymisées, c'est-à-dire que l'on ne doit pas être capable de retrouver la personne compte tenu des informations sur son profil (nom, adresse, numéro d'identité… ne doivent pas apparaître). Le consentement explicite de la personne concernée doit être également recueilli et sauvegardé, en cas de contrôle de la CNIL.

Avant traitement on dispose d'une base de données comprenant 62 variables. Certaines ne seront pas mentionnées puisqu'elles servent de référence (identifiant client, identifiant profil…). Les autres variables, sont des caractéristiques du profil client. On distingue notamment trois sous-catégories : celles liées à l'assuré, celles liées au contrat et celles liées au véhicule.

On fait également une distinction selon le type de la variable. En effet, selon son type, une variable ne sera pas traitée de la même manière ; on dissocie donc les variables qualitatives (ordinales ou nominales) des variables quantitatives (discrètes ou continues). Ci-dessous, un tableau regroupant par type et par catégorie toutes les variables présentes dans la base de données.

	Assuré	Véhicule	Contrat
Catégorielle	- Sexe - Type de permis - Status marital - Région - Province - ZIP code - Profession - Age du plus jeune conducteur - Mode de transport utilisé pour aller au travail - Cause de résiliation de la police précédente	- Marque - Modèle - Série - Type de carburant - Est sportive ? - Est cabriolet ? - Est 4x4 ? - Est monospace ? - Est break ? - Date de l'achat - Type de voiture (perso, société, leasing) - Usage (privé, chemin du travail, professionnel…) - Type de garage (ouvert, fermé, voie publique) - Type d'alarmes - Associé ou non à une remorque - Type d'assistance de conduite	- Type de garantie - Type de franchise - Type de dépréciation - Première souscription à un contrat (Y/N) - Franchise à l'anglaise (Y/N) - Assureur actuel
Numérique	- Age du conducteur - Nombre d'années sans accidents - Nombre d'accidents responsables - Nombre de dommages responsables - Nombre d'accidents en cours - Nombre de tentative de vols - Nombre de révocation de permis - Nombre d'années assurées consécutives sans accidents - Bonus-malus - Identifiant du profil	- Âge de la voiture - Nombre de chevaux - Valeur catalogue - Valeur facturée - Valeur des options - Valeur des extras - Valeur assurée - Nombre de kilomètres - Nombre de sièges - Emission de CO^2 - Poids du véhicule	- Prime annuelle - Prime annuelle si paiement semestriel - Prime annuelle si paiement trimestriel

On retrouve donc toutes les caractéristiques de l'assuré (âge, lieu de vie, utilisation de sa voiture, passé assurantiel…), les caractéristiques de la voiture (âge, puissance, valeur, catégorie…) et celles du contrat (type, choix de paiement…). Un détail complet des variables sera donné plus en détail au fur et à mesure de notre cheminement. En effet, leur compréhension complète et totale n'est pour le moment pas nécessaire.

Néanmoins, on sait déjà que l'on se séparera de certaines variables qui ne sont pas des variables explicatives à proprement parler. On citera par exemple :

- Les primes semestrielles et trimestrielles qui ne sont pas connues en entrée du modèle dans la réalité
- L'identifiant du profil
- Le type de franchise, et la présence ou non d'une franchise à l'anglaise, qui ne nous concernent pas dans notre étude

- La date d'achat puisqu'on connaît l'âge du véhicule
- L'assureur actuel, puisque anonymisé et n'ayant pas de correspondance avec l'assureur proposant la prime
- Les prix catalogue, la valeur des extra et la valeur des options : ces trois valeurs sont réunies dans la valeur assurée, qui est leur somme
- Le prix d'achat, car il peut être biaisé si la voiture a été négocié entre particuliers par exemple

D'autres variables peuvent inciter à l'antisélection, mais seront tout de même maintenues, comme le kilométrage, qui n'était traditionnellement pas utilisé par les assureurs puisqu'invérifiable ; l'assuré peut être amené à sous-estimer ses biens couverts ou à dissimuler des informations sur sa déclaration, de manière à pouvoir profiter d'un tarif plus avantageux.

Dans le cadre de ce mémoire la variable d'intérêt sera la prime annuelle. Il s'agit ici d'une prime totale. Elle résulte de la somme de la prime commerciale (ou prime brute / chargée) et des taxes dues sur le contrat d'assurance (27.1% pour une responsabilité civile, 26.75% pour les assurances omnium). Ces taxes diffèrent selon les garanties concernées et sont redistribuées à différents organismes tels que le Fonds Commun de Garantie Belge, la Croix-Rouge, le Fonds des Handicapés, l'Institut National d'Assurance Maladie-Invalidité (INAMI)...

La prime commerciale est, elle, composée, en assurance automobile, à environ 70% de la prime pure, à laquelle on ajoute des dépenses administratives (frais d'établissement du contrat et des quittances exposées à l'origine), des frais d'acquisition du contrat (commission des parties prenantes/intermédiaires), une marge de sécurité et une marge de profit.

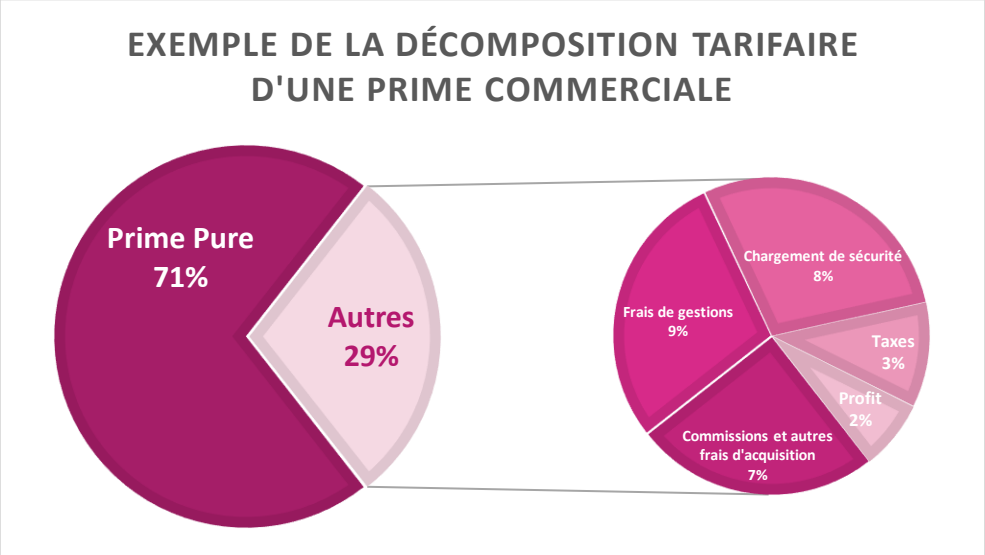


Figure 5: Décomposition tarifaire d'une prime commerciale

II Traitement des données

"On fait la science avec des faits, comme on fait une maison avec des pierres ; mais une accumulation de faits n'est pas plus une science qu'un tas de pierre n'est une maison" - Charles Kettering

Avec l'émergence de la big data et des mégadonnées, de nouvelles contraintes apparaissent. Auparavant, lorsque l'assureur réalisait son étude statistique basée sur son historique de sinistres, il ne disposait que de données restreintes. Avec l'apparition des data warehouse et le passage au numérique, tous les nouveaux sinistres sont désormais consignés en détail. Certains assureurs offrent même de réaliser le constat par photos, ce qui leur permet de développer de manière encore plus précise leur base de données. Les informations peuvent également être croisées avec des facteurs exogènes, telles que la météo ou le trafic à l'instant de l'incident. L'explosion quantitative de la donnée conduit donc à une nouvelle vision du monde et son analyse doit elle aussi évoluer. Avec les performances des ordinateurs qui ne cessent d'augmenter, le traitement des données est de plus en plus facilité ; les grandes dimensionnalités ne sont plus un rempart désormais mais restent néanmoins compliquées à appréhender.

Ainsi, les assureurs peuvent évaluer de mieux en mieux le coût probable d'un client en fonction de ses caractéristiques. Mais comment prédire ce coût efficacement ? En effet, grande dimensionnalité implique que le traitement de la base de données est plus long et coûteux en terme de ressources. De nombreux algorithmes prédictifs souffrent donc de ce que Richard Bellman avait nommé la 'Curse of dimensionality' : une augmentation linéaire de la dimensionnalité conduit à une augmentation exponentielle de l'espace de recherche. Plus l'espace est grand, plus il y a besoin de données pour estimer un nombre de paramètres croissant.

De plus, que l'on traite des données numérisées ou rentrées automatiquement, il se peut que certaines données soient manquantes voire aberrantes. En effet, que les erreurs soient de source humaine ou dues à une évolution du processus de traitement, leur trop grand nombre peut affecter involontairement une étude statistique. Un traitement est donc requis au préalable, car de nombreux algorithmes gèrent avec difficultés ces irrégularités. Si le nombre de données manquantes est trop important, l'algorithme peut alors totalement choisir d'ignorer complètement une variable. Des variables peuvent également être fortement corrélées entre elles, voire redondantes, ce qui peut perturber certains algorithmes comme les GLM.

Ces deux problèmes peuvent être palliés par le nettoyage préalable de la base de données, ce qui rend le traitement des données une étape essentielle de la tarification. Sans 'bonnes' données il est tout simplement impossible de réaliser un modèle pertinent et exploitable. Comment peut-on déduire de son histoire si l'on ne dispose pas d'informations sur son passé ?

C'est pourquoi dans cette partie le processus sera présenté pas à pas ; toutefois toutes les démarches réalisées dans le cadre de l'obtention d'une base de donnée propre et apte à être étudiée de manière à limiter au maximum les incohérences dues au matériel de base ne seront pas présentées.

Il est tout d'abord primordial d'avoir une bonne idée des données sur lesquelles on se base : les comprendre est indispensable pour pouvoir les exploiter. Une fois la base de données et ses variables appréhendée, il faut s'assurer qu'elle ne présente pas d'incohérences, telles que des niveaux de modalités qui n'existent pas ou des observations irréalisables dans la vie réelle (un assuré de 130 ans par exemple).

Le travail d'un actuaire en tarification est également de s'assurer que les variables à exploiter sont pertinentes. Pour cela il peut être amené à créer ou fusionner des classes d'assurés déjà existantes. Il s'agit d'un travail important réalisé en amont de la tarification. Discrétiser des variables continues permet notamment de :

- Neutraliser les outliers, qui se trouvent intégrés à la première ou la dernière modalité
- Prendre en compte la non-monotonie et la non-linéarité, avec autant de coefficients que de modalités
- Obtenir une plus grande robustesse
- Prendre en compte simultanément les variables qualitatives et quantitatives

Comme il l'a été précisé dans le préambule de ce mémoire, la manière dont l'actuaire a travaillé est une donnée dont nous ne disposons pas. Hors le but de ce mémoire est de proposer à un assureur une vision au plus juste de sa concurrence. Cela revient donc à reconstruire non seulement le modèle de l'assureur concurrent, mais également les différentes modifications faites par l'actuaire quant à la sélection de variables. Il est aussi important de garder à l'esprit que pour le bon fonctionnement de la modélisation, il est nécessaire de limiter les redondances d'informations imputée à une corrélation trop forte entre deux ou plusieurs variables.

II.1. Analyse préliminaire

Tout d'abord il est essentiel de comprendre la constitution et le fonctionnement de la base de données sur laquelle se basera notre étude. On rappelle au lecteur qu'il s'agit d'une base de données de 659 183 lignes, comprenant 98 188 profils, et proposant uniquement de la garantie mini omnium sans franchise. Même si le processus de tarification évolue d'un assureur à l'autre, le principe de base reste tout de même très similaire, c'est pourquoi cette étude sera réalisée sur un unique assureur, qui sera détaillé dans la suite de cette partie.

Le traitement des données se fait désormais quasiment systématiquement par ordinateur. Dans notre base plus particulièrement, aucune des données n'est antérieure à 2019. Néanmoins, il ne faut pas perdre de vue qu'elle est issue d'un agrégateur en ligne. Les prospects sont donc libres d'inscrire n'importe quelle information, sans que cette dernière ne soit vérifiée par la suite. Même s'il est improbable qu'une mauvais remplissage soit effectué volontairement, on n'est jamais à l'abri d'une erreur de typographie ou d'une erreur dans une date. Ce genre d'erreurs peut notamment conduire à des incohérences entre âge du permis et âge du conducteur, ou encore entre modèle et valeur du véhicule.

Ce genre d'erreur peut être limitée par une saisie automatique adaptée. Certains systèmes opèrent alors une liaison entre la sélection des dates. D'autres outils, comme le renseignement d'une plaque d'immatriculation fournissant automatiquement marque et modèle d'un véhicule peuvent également réduire cette erreur systématique. Néanmoins aucune de ces fonctionnalités n'est implémenté par l'agrégateur utilisé.

Malgré tous ces désagréments éventuels, le tarif final devrait prendre en compte ces erreurs et refléter un tarif les prenant en compte, quitte à ce qu'il soit dans la réalité incohérent voire impossible. Cela devrait également faciliter le travail des algorithmes à déceler la pertinence, via leur influence, de certaines variables. Pourtant, pour assurer un travail plus rigoureux, ces données seront supprimées dans l'étude ; il est difficile de connaître l'incidence d'une certaine combinaison de variables, potentiellement traitée comme impossible par l'assureur fournissant le tarif. Cela implique d'opérer une différence de traitement entre données influentes et données aberrantes.

II.1.1 Analyse des données

Il est important de connaître les données sur lesquelles on base toute une étude ; d'appréhender leur répartition ainsi que leur subtilités pour pouvoir en faire ressortir un maximum d'informations. L'une des premières étapes dans cette découverte peut se faire via des statistiques globales basiques. Dans notre cas, cela amène à réaliser quelques points méritant d'être soulevés, car impactant l'étude :

- Un graphe représentant le montant des primes maximum montre que les primes sont plafonnées pour tous les assureurs à un montant de 561.44€. Néanmoins, cela touche moins de 1% des données de chaque assureur
- Une observation de la fréquence de certaines variables montrent des apparitions trop fréquentes de certaines modalités. Ainsi, on observe 38 426 observations ayant un âge de 44 ans contre 12 859 ayant un âge de 43 ans… On a également un pic pour une valeur assurée de 19 561.98€. Les observations présentant ces deux caractéristiques ceteris paribus seront donc supprimées pour ne pas surreprésenter un profil particulier, surreprésentation qui n'a pas lieu d'être dans la réalité et qui pourrait nuire à toute notre démarche. En effet ce pic de fréquence est sûrement dû à une valeur par défaut prise par l'agrégateur lorsque le champ correspondant est vide ; en supprimant ces associations de valeurs on se sépare donc des requêtes des profils par défaut. Après cette modification, on n'observait plus le pic de fréquence sur ces valeurs.
- Des valeurs assurées sont négatives, et sont en valeur absolue, supérieures à 8 millions d'euros. Ces observations, au nombre de 95 sont supprimées, puisqu'elles ne concernent que 95 observations, soit moins de 0.015% des données

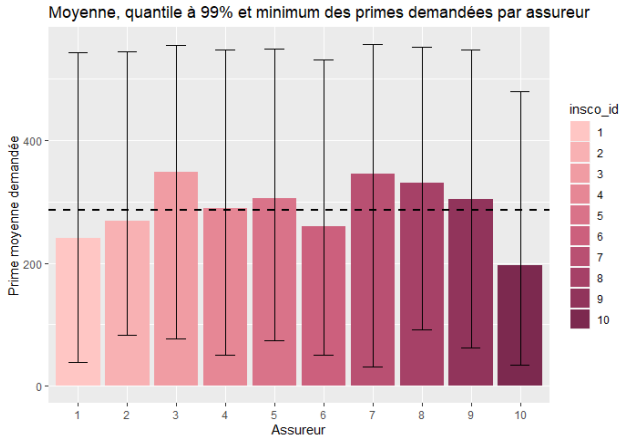


Figure 6 : Statistiques de la prime par assureur

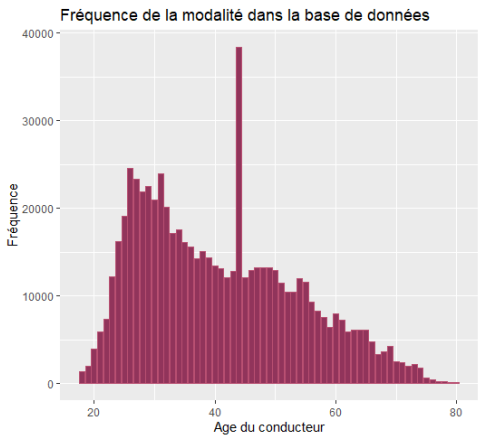


Figure 7 : Fréquence du profil par âge du conducteur

Il est aussi fondamental de se poser la question sur la manière de traiter correctement les valeurs manquantes. Une réflexion sur ce sujet suffirait à combler un chapitre entier. En effet, à partir de quel pourcentage de valeurs manquantes vaut-il mieux supprimer la variable? Faut-il mieux remplacer les valeurs manquantes par la valeur la plus fréquente, ou espérer pouvoir en toute cohérence la remplacer avec une distribution similaire à celle observée sur les modalités non manquantes ?

Dans un ouvrage consacré à cette problématique, "*Statistical Analysis with missing Data*", R.J.A Little et D.B.Rubin déclarent que "L'idée de l'imputation est à la fois séductrice et dangereuse".

En effet, puisque la base de données provient d'un tiers, il est compliqué d'avoir une transparence totale sur le traitement qui a été réalisé avant la mise à disposition. Les valeurs NA correspondent-elles à des informations non complétées ? Ou à une erreur momentanée de l'algorithme de traitement ? Si l'information n'a pas été complétée, était-ce volontairement car aucune option ne correspondait, ou tout simplement un oubli du prospect ? Autant de questions pour lesquelles obtenir une réponse se révèle être compliqué. C'est pourquoi les valeurs manquantes sont un véritable obstacle pour l'actuaire, qui doit comprendre leur présence, mais aussi faire la part des choses dans leur interprétation.

Dans notre cas, nous avons décidé d'opter pour les choix suivants, selon le pourcentage que représentaient les données manquantes :

- 3.5% des données manquaient à la colonne demandant le moyen de commutation. Les NA ont été remplacés par la valeur la plus fréquente, le déplacement en voiture
- 35.1% des données n'étaient pas fournies pour le coefficient bonus-malus. Comme leur usage est controversé en Belgique, il ne sera pas utilisé dans notre étude
- 0.1% des données ne comprennent pas de renseignement pour le statut marital. Après une étude graphique, ces données étant similaires à celles des personnes divorcées, ces dernières ont été regroupées

Ce traitement des données manquantes peut bien sûr être remis en question, mais est un choix nécessaire pour la poursuite de l'étude. Une imputation par tirage conditionnel ou par analyse factorielle aurait pu être réalisée pour les valeurs manquantes du statut marital, mais semblait peut pertinente dans notre cas, compte tenu qu'un assuré divorcé ne compte pas forcément de statistiques particulières, et du fait que le nombre de données nécessitant une imputation était relativement faible compte tenu du nombre total de lignes de la base de données.

Un autre point à ne pas négliger lorsque l'on s'assure de la bonne fiabilité des données, est de vérifier que les données sont globalement en accordance avec ce que l'on s'attend à trouver au niveau de la répartition des primes, ou ce qui a existé par le passé :

- Des primes en moyennes plus chères pour les jeunes conducteurs
- Une augmentation de la prime moyenne en accord avec la puissance du véhicule
- Etc...

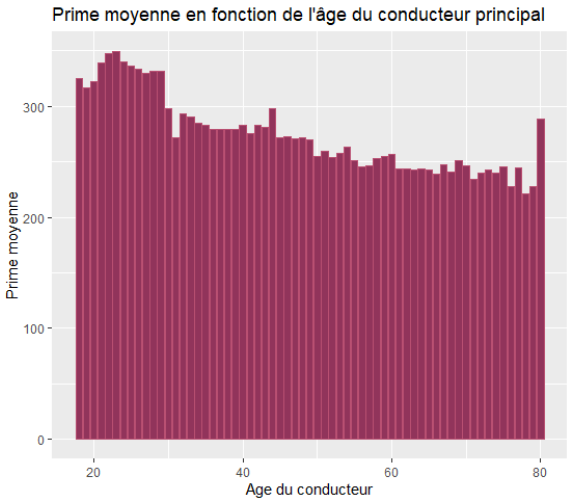


Figure 8 : Prime moyenne en fonction de l'âge du conducteur

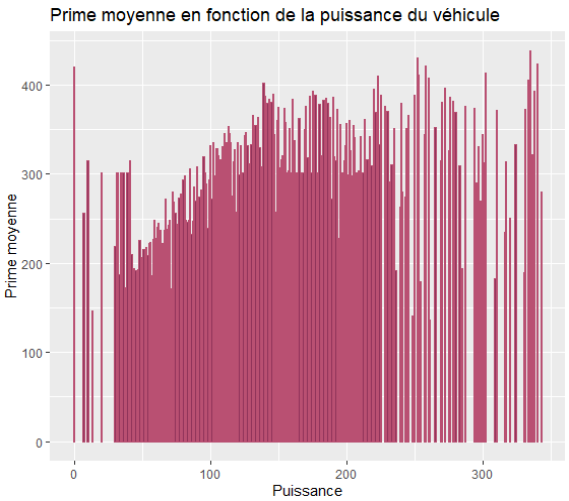


Figure 9 : Prime moyenne en fonction de la puissance

Ces types de graphique permettent également de mettre en exergue quelques anomalies : ici, par exemple, on observe la présence de voitures ayant une puissance de 0 à 50 chevaux, ce qu'il est quasi impossible de trouver en pratique. De telles anomalies, usuellement peu fréquentes, mais nécessitant pourtant de l'attention, verront leur traitement développé dans la partie suivante.

II.1.2 Données aberrantes

Dans cette partie, nous tenterons de détecter des erreurs dans les lignes de notre base de données. Ce processus repose sur le bon sens et est à distinguer du processus de recherche d'outliers qui sera développé dans la partie V.2.2.

Une donnée aberrante est définie comme une observation se trouvant 'loin' et isolée des autres observations du dataset : de manière plus mathématique, les valeurs d'une observation "non-aberrante' doivent plus ou moins correspondre aux valeurs prises par d'autres observations du dataset.

Ainsi il est important de vérifier la cohérence de nos données. Il est impossible qu'un assuré de 20 ans ait 10 ans de permis derrière-lui. De même, si un assuré se dit habiter dans la province du Brabant Wallon alors la région renseignée (Flandres, Bruxelles-Capitale, Wallonie) ainsi que son ZIP code doivent être remplis en accordance avec cette donnée.

C'est pourquoi des observations où la valeur assurée et catalogue est inférieure à 0, où l'âge du permis est de 100 ans, ou où l'émission de CO^2 est supérieure à 400g doivent être étudiées avec attention. En effet, il s'agit de voir si il s'agit d'une erreur récurrente, et si oui, de comprendre pourquoi elle a été faite et sa provenance. Ici, leur survenance est sporadique, et ces données seront donc retirées, du fait qu'elles sont biaisées.

Ensuite il est important de vérifier que toutes les variables ont été encodées correctement. En effet, une variable catégorielle comme le statut marital, encodée en 1 = Marié, 2 = Divorcé, 3 = Veuf, 5 = Célibataire, 6 = Cohabitant, ne devrait pas être une variable numérique, mais une variable catégorielle.

De plus certaines variables, comme les classes de valeur assurées, partitionnées en classes de 5k€ doivent être ordonnées pour éviter des contresens logiques éventuels ; il serait illogique qu'en mini omnium, un assuré ayant une voiture assurée pour 10k€ paye plus cher son assurance qu'un usager avec les mêmes caractéristiques mais une voiture de 100k€.

Il ne s'agit pas de données aberrantes à proprement parler, mais revenons sur ce que nous avions remarqué dans la partie précédente : les primes sont comme bornées pour leur valeur maximale. Cela peut à priori sembler une remarque anodine et de faible importance, mais si l'on cherche à retrouver les modèles des différents assureurs dans une optique de les comparer par la suite, cela à toute son importance.

En effet, admettons qu'un assureur A soit en moyenne 10% plus cher que l'assureur B. Alors un très mauvais client sur le papier, par exemple un jeune assuré avec un an de permis et déjà deux accidents responsables, peut obtenir par le biais de cette tarification le même tarif chez A et chez B, ce qui n'arriverait pas dans la réalité. De plus cela peut influer sur le modèle dans le sens où l'effet marginal de chacun de ces paramètres (jeune âge, permis récent, accidents responsables), sera atténué par la nécessité de diminuer le tarif par rapport à ce qu'il est vraiment.

Puisque nous ne savons pas pourquoi ces données apparaissent comme limitées, la décision a été prise de supprimer les observations ayant atteint ce seuil et celles en étant proches. Il est

toutefois évident que réaliser un tel écrêtement supprime certains profils, et que pour ces profils à forte prime, la modélisation sera plus complexe.

On voit donc que notre jeu de données peut parfois contenir des variables aberrantes puisque impossibles à trouver dans la réalité. Néanmoins, même si certaines combinaisons de variables peuvent sembler incongrues (un véhicule de 200 000€ avec seulement 50 chevaux par exemple), ce n'est pas pour cela qu'elles doivent être supprimées de l'étude. Une telle combinaison peut arriver dans la réalité, et alors il faudra que le modèle choisi sache prédire pour celle-ci. En effet, il arrive souvent qu'une erreur soit faite systématiquement, c'est-à-dire qu'une certaine combinaison en entrée permet d'atteindre une autre combinaison en sortie, mais toujours la même. Bien sûr, la meilleure solution reste de détecter la cause du problème en amont, mais il peut être possible de la corriger plus tard. Néanmoins, il faut faire attention à ce que ce genre de données n'influe pas sur l'effet marginal de chaque variable, risquant ainsi d'altérer tout le modèle ; leur proportion doit donc être contrôlée et surveillée.

C'est ainsi qu'on peut introduire la notion d'observation influente. Puisque ces observations sont influentes sur un modèle, et non de manière générale, leur cas sera évoqué une fois que nous serons entré dans la phase de modélisation, c'est-à-dire dans la partie V.

Pour l'instant, il nous faut encore que adapter notre base de données au travail qui l'attend.

II.2. Nettoyage des données

A partir de maintenant, pour des raisons pratiques nous n'utiliserons plus la base de données entière. En effet, le but étant de déterminer un tarif pour chaque assureur, nous ne nous baserons désormais que sur un assureur en particulier. Cela permettra de détecter des particularités et des outliers propres à chaque assureur. De plus, chaque assureur accorde une importance différente à l'utilisation de chaque variable renseignée.

Arbitrairement, l'assureur 8 a été sélectionné. Nous mènerons donc l'étude sur 76 998 observations et 29 748 profils. Nous verrons plus tard que la quantité de données dans notre cas n'est pas forcément un indicateur de la performance du modèle.

Dans la suite de cette étude, nous ferons donc référence à la base de l'assureur 8 comme notre base de données, qui ne sera dès lors plus la base de données contenant tous les assureurs. Bien sûr, toutes les étapes réalisées à partir de maintenant sont donc à reproduire en cas de changement d'assureur sujet d'étude.

Nous en profitons également pour introduire quelques notations qui seront utilisées tout au long de ce mémoire, et seront régulièrement modifiées pour s'adapter aux différentes amélioration apportées à nos modèles :

- n représentera le nombre total d'observations. Dans notre cas, $n = 76\,998$
- p représentera le nombre de variables explicatives. Pour l'instant $p = 47$
- $Y = (y_1, \ldots, y_n)$ est le vecteur réponse, c'est-à-dire le vecteur contenant les n primes des prospects
- $\hat{Y} = (\widehat{y_1}, \ldots, \widehat{y_n})$ est le vecteur des prédictions, c'est-à-dire le vecteur obtenu après prédiction de notre modèle
- $\mathrm{X}^{(\mathrm{i})} = (x_{i,1}, \ldots, x_{i,p})$est la représentation vectorielle des caractéristiques d'un assuré i
- $X_j = (x_{1,j}, \ldots, x_{n,j})$ représente les valeurs prises par la variable j, c'est-à-dire que si par exemple $j = Age$ alors $X_{Age} = (25, 24, 42, \ldots.)$

- $\ddot{X} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,p} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,p} \end{pmatrix} = \begin{pmatrix} X^{(1)} \\ \vdots \\ X^{(n)} \end{pmatrix} = (X_1 \quad \cdots \quad X_p)$ est la matrice contenant toutes les informations contenues à propos des p variables dans la base de données pour les n profils

Notre objectif dans ce mémoire sera d'obtenir un estimateur $\hat{\phi}(X)$ tel que :

$$\hat{\phi}(X_i) = E[Y|X = X_i]$$

Toutefois estimer $\phi(X)$ n'est pas simple car cela revient à un problème d'estimation non paramétrique et on ne sait rien de la fonction ϕ. Des hypothèses doivent donc être faites sur la loi de Y pour transformer ce problème en un problème d'estimation paramétrique. Cela sera développé plus avant dans la partie IV, une fois la base de données prête à être utilisée, et un prétraitement de sélection des variables ayant été réalisé.

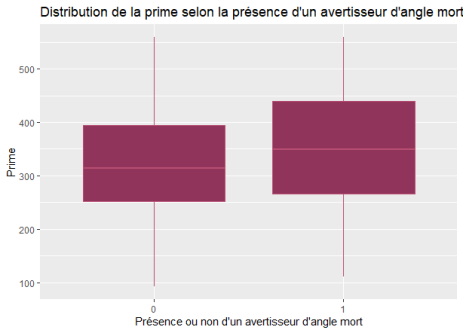
Cette partie est donc consacrée à la modification de notre base de données, de manière à la rendre plus intelligible et plus maniable pour la suite de cet ouvrage.

II.2.1 Création et transformation de variables

Les données récoltées sont brutes. C'est-à-dire qu'elles n'ont pas été retraitées par les assureurs de manière à obtenir un profil de risque cohérent. Comme mentionné plus haut, les actuaires s'occupant du processus de tarification regroupent souvent les variables de manière à obtenir des classes homogènes de risque. Ces classes font partie intégrante du modèle de l'assureur et sont donc confidentielles. De même, si l'assureur a décidé de combiner deux variables entre elles, comme 'usage' et le code postal, il est difficile de savoir si la distinction s'est faite ou non.

Toutefois, dans le cadre de notre étude, il est important de s'affranchir de certaines modalités qui peuvent porter à confusion notre algorithme, du fait de leur trop grand nombre. On va donc s'intéresser aux différentes modalités comprises dans les variables ZIP, profession, système d'assistance au conducteur et marque/modèle/série.

Commençons avec la variable du système d'assistance au conducteur. Dans nos données, le prospect se voit proposer de renseigner quels systèmes sont installés sur le véhicule à assurer afin de faire diminuer le coût de la prime. Huit systèmes sont implémentés : avertisseur d'angle mort, régulateur de vitesse adaptatif, détecteur de fatigue, assistance parking et d'autres encore. Ces systèmes peuvent être aussi bien séparés que additionnés entre eux. On compte donc dans notre base 255 combinaisons différentes de ces modalités. 47 449 observations ne disposent d'aucun de ces dispositifs, soit 45% des observations. Dans la majorité des autres équipements, le taux moyen de possession est de 21%. Néanmoins on remarque que leur effet sur la prime est sensiblement le même:



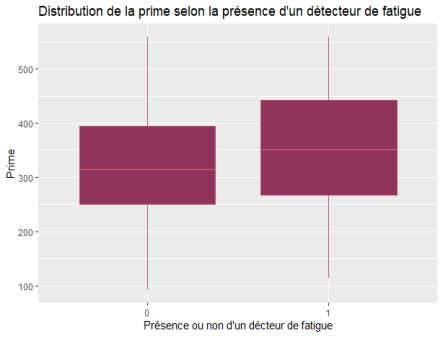


Figure 10 : Distribution de la prime selon présence d'un avertisseur d'angle mort *Figure 11 : / d'un détecteur de fatigue*

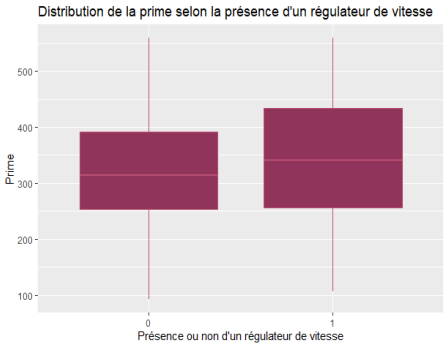


Figure 12 : / d'un régulateur de vitesse

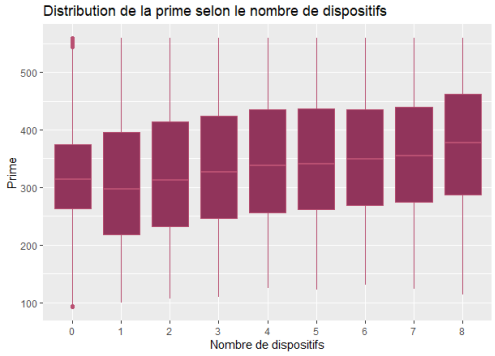


Figure 13: / Selon le nombre de dispositifs d'aide à la conduite

Par contre on observe clairement que la prime augmente avec le nombre de dispositifs, ce qui peut sembler contre-intuitif au premier abord, avant que l'on ne se rappelle qu'on parle ici de la garantie mini omnium et que ces dispositifs ne sont pas disposés dans les voitures bas de gamme. On a donc une relation entre prix de la voiture et nombre d'équipements. On gardera donc le nombre de dispositifs comme variable explicative. Ceci est confirmé, si l'on fait abstraction des voitures au prix assuré extrêmement élevé (soit supérieur à 100k€), dans le graphique d'analyse bivariée présenté ci-dessous :

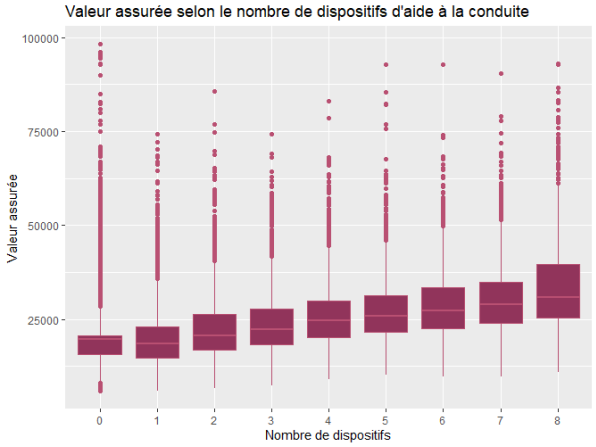


Figure 14 : Valeur assurée selon le nombre de dispositifs d'aide à la conduite

Pour ce qui est du ZIP, on a 1145 niveaux. Compte tenu du fait que nous avons déjà des informations sur la région et sur la province, nous allons nous affranchir de cette variable tarifante, même s'il est fort probable qu'elle applique une plus-value aux variables susmentionnées, dans une ère où la segmentation peut se faire à la rue même, avec calcul de la distance à l'autoroute la plus proche.

Intéressons-nous maintenant à la marque, au modèle et à la série du véhicule. Il y a respectivement 65, 703 et 12256 modalités reliées à ces variables. Une étude statistique montre que cette segmentation est trop importante pour être pertinente dans notre étude. En effet, nous ne disposons pas de classes assez représentatives, même dans la segmentation de la marque seule, pour pouvoir conclure sur un lien éventuel. De plus, il serait incongru de mélanger des marques au sein d'une nouvelle classe. Nous nous abstiendrons donc d'utiliser ces éléments dans nos modèles. Néanmoins, certaines des informations contenues dans les références de la voiture peuvent se retrouver dans la puissance, le prix catalogue, le type de carburant… Une utilisation plus performante à faire de ses données serait de pouvoir utiliser un véhiculier. Hors, tous comme les zoniers, ces

informations sont strictement réservées à l'usage d'un assureur ; il est donc difficile d'appréhender une telle notion sans connaissance interne.

Pour la profession, nous avons à nouveau une profusion de classes : 27 niveaux. Une étude statistique permet de réaliser un regroupement de classes selon leurs similarités sur la distribution des primes, permettant de descendre à un nombre de 7 classes de profession. Cependant, certaines classes présentent peu de profils, et une telle approche doit être réalisée avec prudence puisque peu de données dans une classe peut conduire à une mauvaise représentation de ce segment, et donc par la suite à une mauvaise assimilation par le futur.

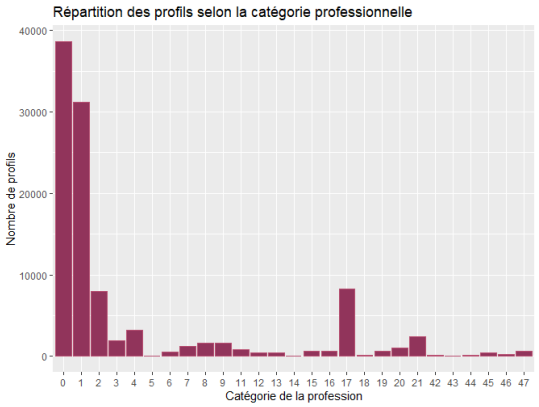


Figure 15 : Répartition des profils selon catégorie professionnelle

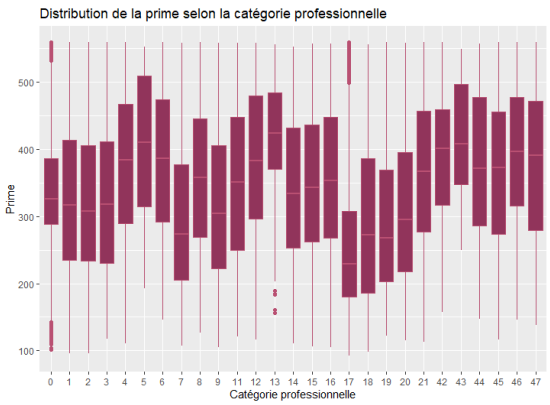


Figure 16 : Prime selon selon catégorie professionnelle

Cela nécessite donc une étude statistique plus approfondie. En effet si une classe contient peu d'observations et a un tarif similaire à une classe du même acabit, alors il peut être envisagé de les regrouper. Cela nuit également à l'interprétation du modèle, puisque au fur et à mesure que l'on mélange les classes, leur dénomination tend à perdre son sens, mais permet de gagner en pouvoir de prédiction, surtout chez les modèles gérant mal une multitude de variables/modalités.

On pourrait également se poser la question de la discrétisation d'une variable numérique pour se retrouver avec des classes. On pense néanmoins à la valeur assurée où à l'âge du conducteur. Néanmoins, ce sujet n'étant pas l'intérêt principal de cet ouvrage, on laissera le lecteur intéressé se référer aux oeuvres d'Arthur Charpentier.

Nous arrivons ainsi à un modèle où nous nous retrouvons avec des variables comptant au plus 11 modalités (pour la variable région). Le modèle compte désormais 40 variables. 22 variables catégorielles et 18 variables numériques. La prochaine étape va d'être d'essayer de faire diminuer ce nombre, et ainsi la dimensionnalité du problème.

En effet, à plus forte mesure, lorsque le nombre de variables devient trop important (centaines, voir milliers ou millions de variables) le problème de p-hacking peut se poser. En effet, avec la multiplication de variables, il devient plus facile d'obtenir un résultat statistiquement significatif (dans la plupart des cas on veut un résultat tel que $p < 0.05$). Un tel phénomène peut conduire à la détection de fausses corrélations, obtenues par p-hacking ou triturage des données en français (c'est à dire en cherchant, cherchant, jusqu'à trouver une corrélation significative). Ces corrélations ne sont pas réelles et sont juste une conséquence du bruit présent dans les données initiales. Elles doivent donc être traitées avec prudence.

Néanmoins, avec seulement 39 variables, nous estimerons que ce phénomène de p-hacking n'est pas présent dans nos données.

II.2.2 Elimination des variables redondantes (corrélation)

Un des écueils majeur de la modélisation se trouve dans l'autocorrélation. En effet, plus on a de variables, plus on a de risque que ces variables soient corrélées entre elles. A première vue, il serait facile de trouver cela propice ; inclure les deux variables dans le modèle ne le lèsera pas, puisque, les deux variables n'étant pas fondamentalement les mêmes, cela revient à rajouter toujours un minimum d'information nouvelle. Mais ce postulat est faux. En incluant deux variables, on pénalise le modèle.

En effet, dans les méthode statistiques (modèles linéaires, GLM, GAM) que nous détaillerons dans la suite, le principe général consiste à estimer des paramètres traduisant l'importance des variables explicatives. Si on a deux variables ou plus apportant la même information, alors, non seulement la méthode aura du mal à estimer correctement leur paramètre, mais en plus l'importance relative de l'information sera partagée entre les variables. Ainsi, pour une information apportant 10 au modèle, son importance sera par exemple répartie entre 7 et 3. Avec ceci, on observe une recrudescence de l'erreur associée, à la fois aux paramètres mais aussi au modèle. Cela peut être dangereux dans l'étape de sélection des variables que nous verrons dans le chapitre III.

Considérons que pour une importance de 20 on ait une répartition à 9 et 11. Hors, pour avoir un modèle simplifié, on décide de ne garder que les variables ayant une importance supérieure à 15. On se sépare de cette information, alors qu'au final elle respecte la condition, puisque son importance globale est de 20. Cela amène donc à un modèle loin d'être optimal qui peut même devenir médiocre si ce phénomène survient à plusieurs occasions. Il est donc primordial d'identifier et de traiter ces éventuels problèmes de multicolinéarité.

Prenons par exemple les variables "driver_age" et "license_age" représentant respectivement l'âge du conducteur et sa durée de possession du permis. Ces deux variables sont évidemment corrélées. En effet, une augmentation de 1 unité de l'âge du conducteur conduit la plupart du temps à la même augmentation dans la durée de possession du permis. Cela vient du fait que généralement les gens obtiennent leur permis à 18, 19 ou 20 ans.

On peut donc se demander s'il n'y a pas redondance d'une des deux variables. En effet, logiquement, puisque l'on connaît l'âge du conducteur, on pourrait facilement en déduire depuis quand il possède son permis. Toutefois, on ne peut pas *vraiment* se permettre cet amalgame puisque certains assurés ne respectent pas le cas général. Néanmoins on pourrait se demander à quel point "driver_age" et "license_age" sont liés.

Une manière de détecter cette relation est via une 'heatmap' ou un tableau de corrélation. On utilise pour cela le coefficient de corrélation de Pearson qui s'obtient, pour deux variables X et Z, via la formule suivante :

$$\widehat{r_p} = \frac{\widehat{\sigma_{XZ}}}{\widehat{\sigma_X}\widehat{\sigma_Z}} = \frac{\sum_{i=1}^{n}(x_i - \underline{x})(z_i - \underline{z})}{\sqrt{\sum_{i=1}^{n}(x_i - \underline{x})} \times \sqrt{\sum_{i=1}^{n}(z_i - \underline{z})}}$$

Où $\underline{x}$ et $\underline{z}$ sont les moyennes arithmétiques des variables X et Z.

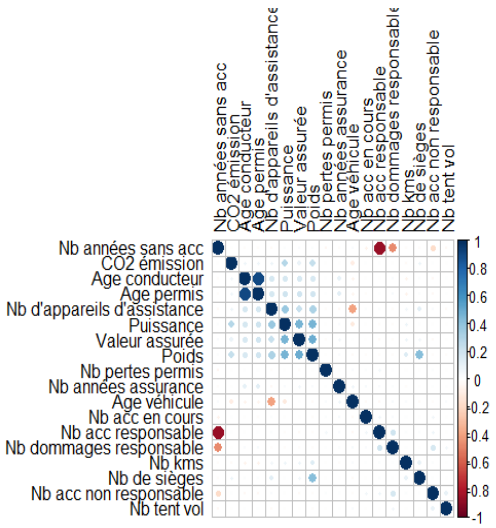


Figure 17 : Graphe de la corrélation de Pearson sur les variables numériques

Plusieurs blocs de corrélations importantes apparaissent mais ils semblent relativement intuitifs :

- Une forte corrélation existe entre l'âge du conducteur et la durée de permis, comme présumé
- On voit également une corrélation entre le nombre d'années sans accidents et le nombre d'accidents responsables et le nombre de sinistres avec des dommages responsables. Plus ces derniers augmentent plus le nombre d'années sans accidents est impacté négativement, ce qui est logique
- Une plus faible corrélation existe entre la puissance, la valeur assurée et le poids, ce qui est sensé vu que ce sont des variables caractéristiques du véhicule
- Notre hypothèse du nombre d'appareils d'assistance présent lié à la valeur assurée du véhicule est également validée. On voit aussi que ce nombre est corrélé négativement avec l'âge du véhicule
- On remarque aussi que l'âge du conducteur est aussi corrélé, tout comme la durée de permis, à la puissance, la valeur assurée et au poids du véhicule. Les personnes plus âgées ont ainsi tendance à avoir des véhicules assurés plus puissants et plus lourds, donc plus chers

Au vu de ces résultats, il semble logique de laisser de côté certaines variables comme la durée de permis et le nombre d'années sans accidents puisqu'elles peuvent se retrouver redondantes avec d'autres variables. De plus leur corrélation avec d'autres variables est suffisamment importante pour que la majorité de la donnée soit conservée. La conservation de la variable poids au vu de sa corrélation avec les variables 'valeur assurée' et 'puissance' est également questionnable.

On remarque néanmoins qu'aucune des variables qualitatives n'est présente. On peut toutefois obtenir le même type de graphique pour les variables qualitatives en mesurant leur interdépendance deux à deux à travers la statistique du test du χ^2de Pearson. On a alors la formule suivante :

$$\sum_{i,j}\left(\frac{\frac{n_{i,j}}{n}-\frac{n_i n_j}{n}}{\frac{n_i n_j}{n}}\right)^2 \sim \chi^2$$

Où $n_{i,j}$est le nombre d'observations présentant la $i^{ème}$valeur pour la première variable à comparer et la $j^{ème}$ valeur pour la seconde valeur. On a donc $n_i = \sum_j n_{i,j}$ et $n_j = \sum_i n_{i,j}$. Et donc la proportion théorique de la fréquence $E_{i,j}$sous l'hypothèse d'indépendance s'écrit comme ;

$$E_{i,j} = n \times \frac{n_i}{n} \times \frac{n_j}{n} = \frac{n_i n_j}{n}$$

On obtient alors le tableau suivant, où sont représentées les p-values du modèle, c'est-à-dire les probabilités d'obtenir, sous notre hypothèse d'indépendance deux à deux, les mêmes valeurs présentes dans notre base de données. On cherche donc cette fois-ci les valeurs les plus proches de 1, ce qui signifie que l'hypothèse d'indépendance est 'validée' :

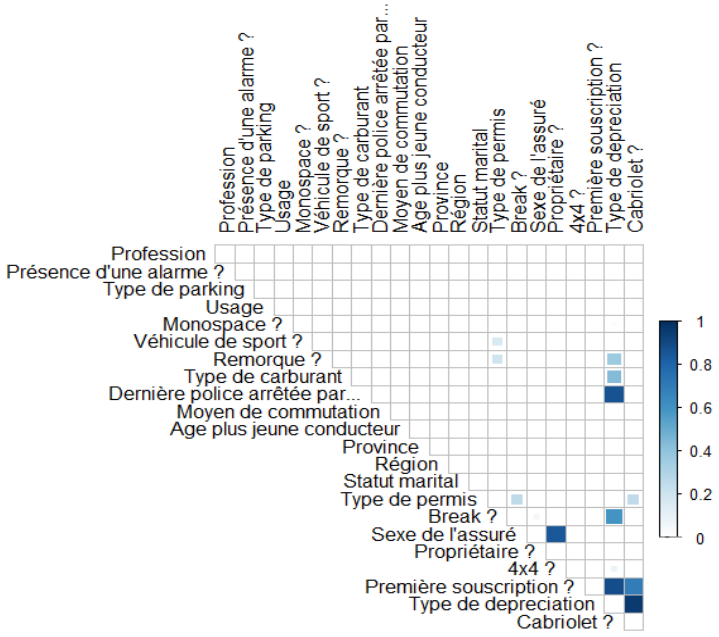


Figure 18 : Graphe de la corrélation du X^2de Pearson sur les variables catégorielles

Selon notre hypothèse, on peut rejeter l'hypothèse d'indépendance deux à deux pour la plupart des combinaisons de variables, sauf pour le type de dépréciation choisi (ce qui est logique vu que chaque profil se voit proposer différents types de dépréciations par notre assureur). On voit également que le sexe de l'assuré est indépendant du statut de propriétaire du véhicule. Néanmoins ces résultat sont peu exploitables. En effet, d'après nos résultats, on peut rejeter l'hypothèse d'indépendance pour la plupart des couples de variables, ce qui ne nous permet pas de faire diminuer le nombre total de variables, et d'ainsi réduire la dimensionnalité de notre problème.

L'un des défauts de cette statistique est qu'elle tient compte du nombre total d'observations, ce qui peut porter désavantage aux modalités d'une variable peu représentée, et donc par extension, expliquer nos résultats peu concluants. Une alternative, le V de Cramer, permet de s'affranchir de ce fait et permet ainsi d'obtenir une mesure absolue de l'intensité de l'association entre les deux variables.

Cette nouvelle statistique normalise le χ^2 par le χ^2 maximal théorique, qui correspondrait à un tableau de contingence comportant une seule valeur non nulle par ligne (l lignes) et par colonne (c colonnes) :

$$V = \sqrt{\frac{\kappa}{\chi^2_{max}}} = \sqrt{\frac{\kappa}{n \times min(l-1, c-1)}}, \quad avec\ \kappa = \sum_{i,j} \left(\frac{\frac{n_{i,j}}{n} - \frac{n_i n_j}{n}}{\frac{n_i n_j}{n}}\right)^2$$

On obtient alors un nouveau tableau de p-values. Cette fois-ci, une valeur de 1 indique une association totale entre les variables et une valeur de 0 une absence de corrélation :

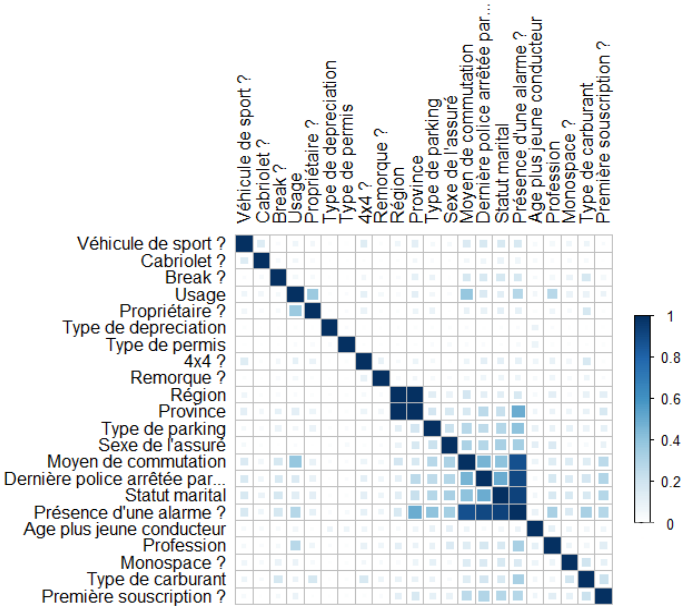


Figure 19 : Graphe du V de Cramer sur les variables catégorielles

On retrouve cette fois-ci un graphe plus similaire à celui trouvé pour les variables purement numériques. Plusieurs blocs apparaissent à nouveau :

- Région et province sont parfaitement corrélées
- La présence d'une alarme est très fortement liée au statut marital, à la raison de fin du dernier contrat et au sexe de l'assuré ainsi qu'à la province de résidence
- Le statut marital est également lié au moyen de commutation, au sexe de l'assuré et au type de parking, ce qui paraît logique puisqu'un assuré marié dispose plus probablement de sa propre résidence et donc d'un parking

Au vu de ces résultats, bien plus concluants que ceux donnés par la statistique de Pearson, il semble nécessaire de retirer plusieurs variables catégorielles. On exclura donc de notre étude les variables suivantes : Présence d'une alarme, Région, Type de parking et Usage puisque ces variables sont redondantes.

Mais on pourrait se demander s'il n'y a pas de corrélation entre les variables qualitatives et quantitatives. Pour ce faire, l'une des méthodes les plus simples est de discrétiser les variables qualitatives en les regroupant par classes. On se retrouve ainsi dans le cas précédent où toutes les variables sont qualitatives. On peut alors obtenir une 'heatmap' représentant la statistique du V de Cramer.

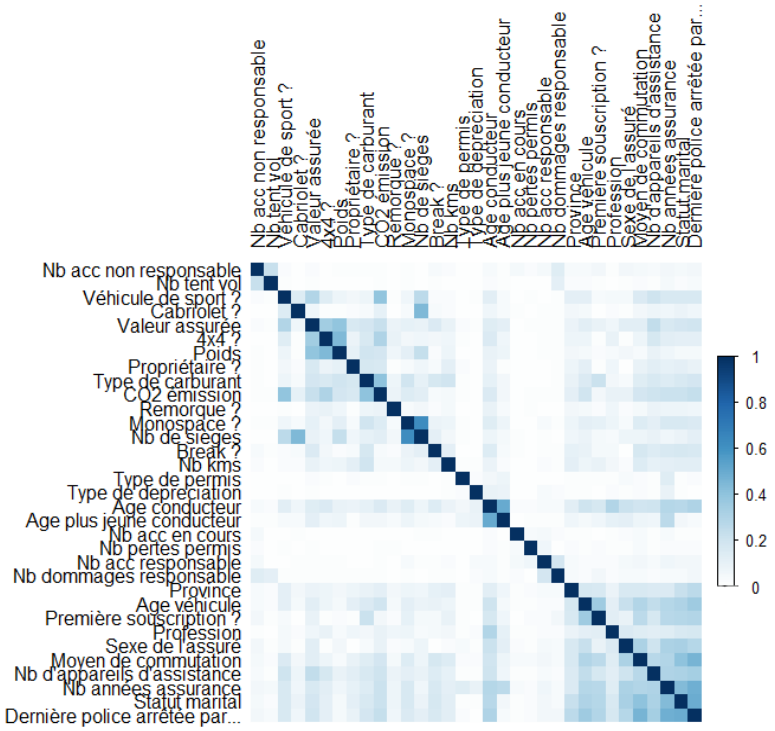


Figure 20 : Graphe du V de Cramer sur toutes les variables

On observe alors de nouvelles corrélations par rapport à celles observées indépendamment entre les variables qualitatives et quantitatives. Par exemple, l'âge du plus jeune conducteur est lié à l'âge du conducteur principal. Le fait que le véhicule soit un 4x4 est lui lié avec la valeur assurée et le poids.

On voit également que le nombre de sièges est relié avec le fait que le véhicule soit ou non un monospace.

Suite à ces nouvelles découvertes, nous décidons d'exclure les variables suivantes du modèle : nombre de sièges, si le véhicule est un 4x4 ou non et l'émission de CO^2 du véhicule.

Néanmoins, malgré une certaine proximité de certaines variables (moyen de commutation principal, statut marital, raison de l'arrêt de la dernière police, âge du plus jeune conducteur), il s'agit de critères clé du contrat d'assurance, et il paraît donc difficile d'écarter arbitrairement l'une d'entre elles.

Nous nous retrouvons donc avec 30 variables explicatives (13 numériques, 17 catégorielles) utilisées pour déterminer la prime proposée à un prospect. Elles sont récapitulées dans le tableau ci-dessous.

	Assuré	Véhicule	Contrat
Catégorielle	- Sexe - Type de permis - Status marital - Province - Profession - Age du plus jeune conducteur - Mode de transport utilisé pour aller au travail - Cause de résiliation de la police précédente	- Type de carburant - Est sportive ? - Est cabriolet ? - Est monospace ? - Est break ? - Type de voiture (perso, société, leasing) - Associé ou non à une remorque	- Type de dépréciation - Première souscription à un contrat (Y/N)
Numérique	- Age du conducteur - Nombre d'accidents responsables - Nombre de dommages responsables - Nombre d'accidents en cours - Nombre de tentative de vols - Nombre de révocation de permis - Nombre d'années assurées consécutives sans accidents	- Âge de la voiture - Valeur assurée - Nombre d'options d'aide à la conduite - Nombre de kilomètres - Poids du véhicule	

II.2.3 Utilisation de la variable réponse

Maintenant que nous avons fait le tour des variables explicatives, concentrons-nous sur la variable réponse. Il s'agit de la prime fournie au prospect par l'assureur, basée sur toutes les caractéristiques rentrées dans le questionnaire l'agrégateur.

Dans notre base de données (celle de l'assureur 8), la prime, toutes catégories de dépréciation confondues, comporte ces statistiques :

	Nombre	Moyenne	Médiane	Min	Max	Quantile à 25%	Quantile à 75%	Ecart-type
Prime	105 361	330.7	319.3	91.9	560	254.6	403.7	104.7

Les différentes catégories de dépréciation sont équiréparties dans le portefeuille dans le sens où la dépréciation à valeur réelle compte 27933 profils différents, à valeur agréée 27061 profils et à valeur agréée 22004 profils (un profil étant une liste de caractéristiques spécifiques à un particulier). Pour un même assuré se voyant proposer une offre agréée et une offre agréée +, l'offre agréée représente 87% du tarif de l'assurance agréée. La différence est plus ténue entre la valeur réelle et la valeur agréée, dont la prime représente 105% de la prime à valeur réelle.

La question de scinder une fois de plus la base de données selon les différentes dépréciations s'est posée durant cette étude. Néanmoins, puisque cette variable a un impact linéaire avec la prime, et qu'il serait dommage de se priver des 2/3 des données disponibles, il a été décidé de garder pour le moment la base de données telle quelle, c'est-à-dire sans opérer cette distinction.

Toutefois, pour montrer l'impact de la réalité sur la théorie, les prochains graphes de cette section ont été réalisés sur la partie à dépréciation réelle de nos données, puisqu'elles sont censées représenter exactement la même garantie. Ainsi, comme nous l'avons vu dans l'introduction, un modèle utilise souvent la présomption que les montants des sinistres suit une loi normale ou log-normale. La prime doit donc contenir cette information et la transcrire. On utilisera ainsi $\mu = 321.77$, $\sigma = 106.4$ pour notre simulation de la loi normale et $\mu_{ln} = 5.71$, $\sigma_{ln} = 0.35$ pour notre simulation de la loi log-normale.

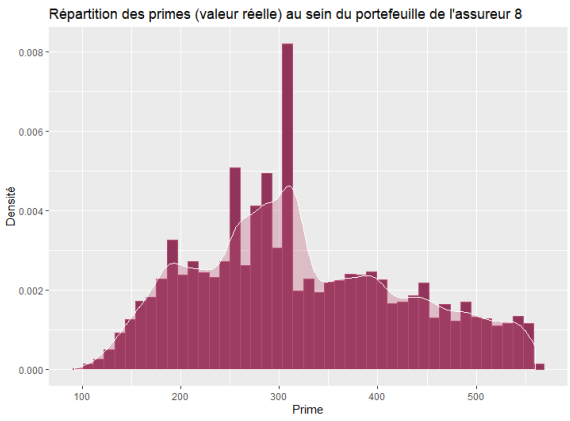


Figure 21 : Répartition des primes (Assureur 8)

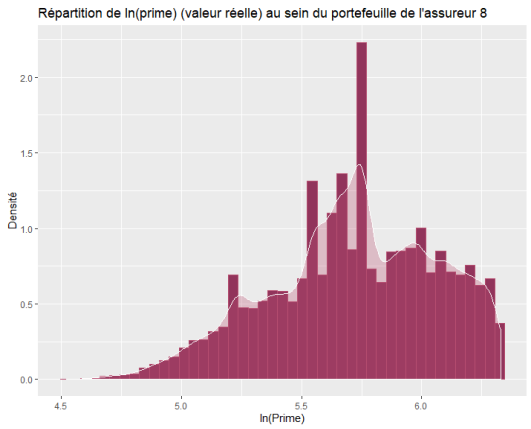


Figure 22 : Répartition de ln(Prime) (Assureur 8)

On remarque dans un premier temps que notre portefeuille ne semble pas respecter cette dynamique. Certains profils, surtout en queue de peloton, donc les 'mauvais' assurés semblent surreprésentés. Cela peut s'expliquer par le fait que ce sont eux qui ont le plus d'intérêt à comparer les prix chez un agrégateur.

On remarque toutefois que la distribution, si l'on néglige la surreprésentation dans la queue des profils à fortes primes, peut s'apparenter à celle d'une loi normale ou log-normale. La loi log-normale semble même être la plus adaptée.

Si on reprend notre portefeuille complet, c'est-à-dire avec toutes les formes de dépréciation, et en réadaptant bien sûr la loi normale à fitter, on obtient des graphes similaires :

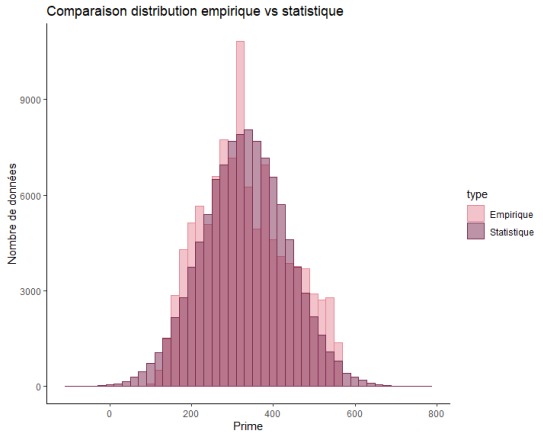


Figure 23 : Empirique vs Statistique (Prime)

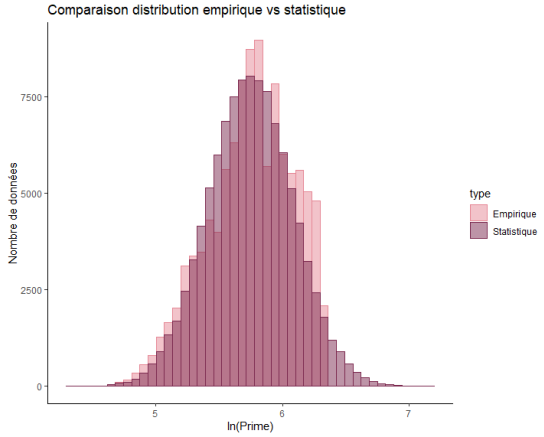


Figure 24 : Empirique vs Statistiqu (ln(Prime))

En effet, utiliser le logarithme népérien des primes permet d'éviter trop de primes plus faibles que ce qu'attendrait une loi normale de paramètre μ_{ln}et σ_{ln}.

Pour conforter notre choix, nous effectuons le test de Kolmogorov-Smirnov. En effet, compte tenu de notre nombre important d'observations (76998), le test de Shapiro-Wilk, qui compare directement une loi empirique à une loi normale, ne peut pas s'appliquer puisque lorsqu'il est utilisé pour un nombre trop important de données, les chances que l'hypothèse de normalité soit rejetée devient trop grande. En effet, avec un grand nombre de données, même les plus petites variations à une distribution normale peuvent être détectés, amenant ainsi à rejeter une hypothèse de normalité, même lorsque les données sont normales. Le test de Kolmorog-Smirnov permet de comparer deux distributions en utilisant leur fonction de répartition.

Sur nos données, on obtient avec la loi normale une p-value de 0.0467 et avec la loi log-normale une p-value de 0.0874. Avec un indice de confiance de 5% on ne peut donc pas rejeter l'hypothèse de log-normalité de nos données.

Toutefois il n'y a aucune certitude sur la loi suivie par nos données. Il est même fort probable qu'elles ne suivent pas de distribution spécifique. Nous verrons que cela peut être notamment problématique dans la mise en place de certains modèles statistiques.

III Sélection des variables pertinentes

"All things are subject to interpretation, whichever interpretation prevails at a given time is a function of power and not truth" - Friedrich Nietzche

Comme cela a été abordé dans le chapitre précédent, il est important que la base de données soit propre. En effet, puisque le but de ce mémoire est de reconstruire un modèle actuariel, il est important de se poser les bonnes questions sur quelles variables ont été utilisées pour tarifer tel ou tel assuré. Ainsi, on comprendra mieux le modèle, et à terme, le possesseur de telles informations pourra les utiliser à profit, en adaptant son propre modèle aux critères prédéfinis par la concurrence, ou en prenant en compte de nouveaux paramètres qui pourraient lui permettre de se montrer plus compétitif.

La sélection de variable permet, entre autres, d'avoir une première idée du poids respectif que devrait occuper chaque variable dans la proposition d'un tarif. Cela pourra, à terme, permettre de valider le modèle, en s'assurant qu'il représentera la tendance que l'on cherche à déceler. Une variable qui sera jugée par la suite 'importante' devra avoir un fort coefficient, ou à défaut, une forte présence/représentation dans notre modèle final.

On notera que certaines variables ne devraient pas avoir un impact dans notre base de données. On pense notamment à la variable 'sexe de l'assuré' qui ne devrait pas influer le tarif final proposé au prospect. En effet la cour de Justice de l'Union Européenne a jugé l'utilisation de cette variable discriminante et a interdit son utilisation par les assureurs dans un processus de tarification depuis le 21 décembre 2012. Cette réflexion devra, par la suite, attester de la légitimité de nos modèles. Néanmoins, on ne peut qu'espérer que les assureurs jouent le jeu ; si notre modèle détecte une légère inclinaison vers l'une ou l'autre des deux modalités, nous ne pouvons assurer avec certitude le phénomène. Certaines variables sont toutefois attendues à n'avoir qu'un effet minime dans la reconstruction de la prime.

Dans cette partie nous aborderons également une nouvelle problématique. En effet, si on décide de lancer un modèle en deux dimensions, basé sur deux variables, alors on peut décomposer la prédiction en 4 termes : une constante, un terme pour la première variable, un autre pour la seconde variable, et un dernier terme pour l'interaction entre ces deux variables. Le terme d'interaction représente la variation dans la prédiction après avoir modifié les valeurs des variables et considéré les effets individuels. Cette notion est très importante dans la constitution d'un modèle de tarification, c'est pourquoi nous lui consacrerons une partie de notre ouvrage.

Dans notre cheminement, nous passerons à travers la sélection de variables standardisée des modèles linéaires généralisés, ainsi que la sélection de variables par les méthodes classiques utilisées en tarifications : filtres, wrapper et embedded méthodes. Dans un second temps, les régression RIDGE et LASSO seront également introduites. Il s'agit de méthodes de contraction des coefficients qui fournissent une liste de variables minimisant la contrainte imposée et sont donc pertinents dans notre contexte ; notre contrainte étant de minmiser l'utilisation de variables.

On abordera également à nouveau la problématique des corrélations entre variables, mais cette fois-ci de manière plus poussée, puisqu'on s'intéressera à leur intérprétation et leur signification.

Nous verrons donc dans ce chapitre différents outils de sélection de variables, que ce soit les filtres ou les méthodes *wrapped* et *embedded*, ou même d'autres méthodes moins répandues telles

que les méthodes SmuRF, la statistique H de Friedman ou encore l'utilisations de méthodes de machine learning .

Nous rappelons également au lecteur les notations introduites en II.2. ci-dessous, de manière à faciliter la compréhension de cette partie :

- n représentera le nombre total d'observations. Dans notre cas, $n = 76\,998$
- p représentera le nombre de variables explicatives. Pour l'instant $p = 40$
- $Y = (y_1, \ldots, y_n)$ est le vecteur réponse, c'est-à-dire le vecteur contenant les n primes des prospects
- $\hat{Y} = (\widehat{y_1}, \ldots, \widehat{y_n})$ est le vecteur des prédictions, c'est-à-dire le vecteur obtenu après prédiction de notre modèle
- $X^{(i)} = (x_{i,1}, \ldots, x_{i,p})$ est la représentation vectorielle des caractéristiques d'un assuré i
- $X_j = (x_{1,j}, \ldots, x_{n,j})$ représente les valeurs prises par la variable j
- $\ddot{X} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,p} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,p} \end{pmatrix} = \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix} = \begin{pmatrix} \underline{X_1} & \cdots & \underline{X_p} \end{pmatrix}$ est la matrice contenant toutes les informations contenues à propos des p variables dans la base de données pour les n profils

III.1. Modèles linéaires généralisés

Dans cette partie, nous allons nous intéresser aux modèles linéaires généralisés (GLM). Généralisation plus souple des régressions linéaires, grâce à l'intromission des fonctions liens et des fonctions caractérisant la variance de chaque mesure, les GLM sont aujourd'hui grandement utilisés dans la construction de modèles de calcul de coût ou de fréquence. La force des GLM réside dans le fait qu'ils s'intéressent à d'autres distributions que la distribution normale et permettent de simuler, par exemple, des lois de Poisson pour la fréquence ou des log-normales pour la sévérité. Cette particularité ne permet toutefois pas de modéliser fidèlement certains phénomènes comme :

- Les impacts de signe contraire aux deux extrémités du domaine de définition d'un certain prédicteur (effets d'extrêmes)
- Les ruptures de la relation de dépendance en des points de discontinuité précis (effets de seuil)
- Les bornes de la variable de sinistralité aux extrêmes (plancher et plafond)
- Un comportement convexe (la fonction de lien logarithmique étant par définition concave)

De plus, comme nous l'avons vu dans la partie II.2.3., notre variable réponse ne semble pas suivre de distribution particulière, mais nous tenterons tout de même de la modéliser à travers une distribution log-normale. Cela s'explique par le fait que, même si un modèle GLM a probablement été développé en amont par des actuaires, le tarif est souvent exprimé sous la forme d'une variable aléatoire suivant une loi composée, qui elle, ne suit pas une loi simplement définie.

Néanmoins de nombreux modèles sont non linéaires et sont tout de mêmes traités linéairement par les méthodes GLM pour plus de simplicité ; il est difficile en effet de sélectionner un modèle en amont lorsque les relations ne sont pas linéaires et que le nombre de variables explicatives est particulièrement important. Ainsi, le 'vrai' modèle est souvent approximé par un modèle linéaire, qui ne représente pas forcément la réalité.

Le but de ce mémoire est d'exposer les limites de cette modélisation, en comparant cette technique communément utilisée à d'autres techniques plus innovatrices. Seront également présentées, les méthodes classiques de sélection des variables dans le cas du GLM et leurs manquements.

III.1.1 Présentation et limites du modèle linéaire

Nous introduirons ici la notion de régression linéaire. Les modèles de régressions sont depuis longtemps utilisés par les statisticiens, les informaticiens, ou tout autre personne qui chercherait à construire un modèle facilement traduisible en expressions non-mathématiques. En effet, l'un des atouts majeurs de la régression linéaire est sa lisibilité ; ces modèles sont facilement interprétables pour toute personne, qu'elle ait ou non un background en mathématiques.

Un modèle de régression prédit la variable réponse comme une somme pondérée des variables d'entrée. C'est cette linéarité qui rend le modèle facile à interpréter. Mais c'est également cette linéarité qui est son majeur défaut ; il est impossible de modéliser des interactions non linéaires avec un tel modèle. Hors ces interactions sont naturellement présentes au sein de la structure de dépendance de la sinistralité. Ces effets non-linéaires peuvent s'exprimer sous diverses formes telles que : des paliers (plancher ou plafond) bornant les valeurs prédites, des seuils marquant une rupture du phénomène, des extrêmes indiquant un caractère non monotone, de la convexité…

D'une manière plus formelle, si on considère un prospect i, on a y_i sa variable réponse et $\mathrm{X}^{(\mathrm{i})} = (x_{i,1}, \ldots, x_{i,q})$ le vecteur constitué de toutes ses variables explicatives. On voit ici que le vecteur X_i n'est pas exactement le même que celui présenté plus haut, qui contenait p observations. Par abus de notation, nous continuerons d'appeler X_i le vecteur du prospect i même si il serait plus légitime d'écrire $\mathrm{X}^{(\mathrm{i})} = (x_{i,1}, \ldots, x_{i,p})$ le vecteur originel et $\mathrm{X}^{(\mathrm{i})\prime} = (1, x_{i,1}', \ldots, x_{i,q}')$ le vecteur obtenu à partir de $X^{(i)}$, permettant une cohérence avec le vecteur $\beta = (\beta_0, \beta_1, \ldots, \beta_q)$ du modèle.

On a alors pour le prospect i cette valeur réponse :

$$y_i = X^{(i)}\beta^T = \beta_0 + \beta_1 x_{i,1} + \ldots + \beta_q x_{i,q} + \varepsilon_i$$

Avec ε_i l'erreur réalisée au niveau du prospect.

q représente le nombre total de paramètres du modèle ; il ne dépend pas de p dans le sens où une seule variable comprenant plusieurs modalités peut amener k paramètres dans le modèle. Une hypothèse forte est déjà faite à ce niveau du modèle ; les erreurs ε sont supposées suivre une distribution gaussienne, c'est à dire normale. Il est présumé que l'on fait beaucoup de 'petites' erreurs et peu de 'larges' erreurs.

Avec cette hypothèse, on peut écrire de manière plus générale :

$$Y = \beta \times \ddot{X} = \beta_0 + \beta_1 X_1 + \cdots + \beta_q X_q$$

On a alors :

$$E(Y) = E\left(\beta_0 + \beta_1 X_1 + \ldots + \beta_q \mathrm{X_q} + \varepsilon\right) = \beta_0 + \beta_1 E(\mathrm{X_1}) + \ldots + \beta_q E\left(\mathrm{X_q}\right)$$

On voit donc que le terme d'erreur ε disparaît lorsque l'on prend l'espérance de la variable réponse. Cela implique que nos données appliquées au modèle doivent conduire à ce que l'espérance de la variable $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)$ soit nulle.

De nombreuses méthodes peuvent être employées pour déterminer le vecteur $\beta = (\beta_0, \beta_1, \ldots, \beta_q)$ mais la plus répandue demeure la méthode des moindres carrés ordinaires ; cette

méthode présente l'avantage d'être sans biais, c'est-à-dire $E(Y) = E(\hat{Y})$ où $\hat{Y}$ est l'estimation de Y obtenue avec nos données.

Mais pour pouvoir appliquer la méthode des moindres carrés ordinaires, il faut faire de nouvelles hypothèses : les termes d'erreur $(\varepsilon_i)_{i=1,\dots,n}$ doivent être homoscédastiques et non corrélés ; sinon il vaut mieux utiliser la méthode des moindres carrés généralisés. Dans notre cas, on supposera les erreurs homoscédastiques ($Var(\varepsilon_i) = \sigma^2, \forall i \in [1, n]$) et non corrélées.

On comprend donc pourquoi les modèles linéaires sont très répandus ; très simples à mettre en place et à comprendre, l'importance des différentes variables pouvant être interprétée à un niveau modulaire via les poids β_j. Les effets sont de plus additifs et faciles à séparer les uns des autres. Néanmoins, comme on l'a vu au cours de notre construction progressive, utiliser ce type de modèle nécessite l'utilisation de nombreuses hypothèses, qui ne s'appliquent pas toujours dans la réalité. L'emploi d'hypothèses fausses nuit évidemment à la qualité du modèle. Dans l'optique d'utiliser raisonnablement un modèle de régression linéaire, les données doivent donc obéir à quelques règles rappelées ci-dessous :

- Linéarité : ce modèle force la prédiction à être une combinaison linéaire des variables explicatives, ce qui est à la fois un atout puisque cela facilite l'interprétation, mais est également une faiblesse car l'hypothèse est peu vraisemblable dans la plupart des cas
- Normalité : la variable réponse est supposé suivre une distribution normale. Cela vient du fait que pour avoir une idée de ce que vaut la variable réponse, il faut que celle-ci varie peu et/ou qu'elle puisse s'étendre sur un intervalle infini avec peu de densité dans les queues. Si cette hypothèse n'est pas vérifiée alors les intervalles de confiance sur les paramètres sont faux
- Homoscédasticité : la variance de l'erreur ($Var(\varepsilon)$) est supposée constante. Cela signifie qu'on se trompera d'autant sur une prime à 200€ que sur une prime de 1200€, ce qui peut être problématique dans l'estimation des petites primes
- Indépendance : chaque observation est présumée indépendante des autres
- Variables fixes : les variables d'entrée sont supposées données : ce ne sont pas des variables aléatoires
- Absence de multicolinéarité : on suppose que les variables explicatives ne sont pas corrélées ou très peu

Comme on l'a vu dans la partie II, certaines hypothèses ont déjà été quelque peu remises en questions de par nos données. On peut d'ores et déjà mentionner que celles d'absence de multicolinéarité est violée. Celle relative à la normalité de la variable réponse est questionnable.

Quant aux autres hypothèses, on ne peut, avant la création du modèle, attester de l'hypothèse d'homoscédasticité ou de linéarité. Concernant l'hypothèse sur les variables fixes, puisque les données sont réputées fiables, alors on peut estimer qu'elle est vérifiée.

On voit donc que même si on peut, dans les faits, utiliser un modèle linéaire, son assertion ne sera pas forcément valide. En effet ce modèle est assez rigide et offre peu de libertés dans son adaptation, ce qui peut poser problème dans la pratique.

C'est pourquoi les GLM (Generalized Linear Model) ont été proposés par John Nelder et Robert Wedderburn en 1972, dans une optique d'unifier les modèles statistiques utilisés à l'époque, comme la régression linéaire, la régression logistique ou encore la régression de Poisson.

III.1.2 Principe général

Le GLM diffère de sa prédécesseure, la régression linéaire ordinaire, dans le sens où il permet un certain affranchissement des hypothèses nécessaires à son application. En effet, les GLM offrent la possibilité d'évaluer d'autres distributions que la distribution gaussienne. Là même où les régressions linéaires ordinaires prédisent l'espérance d'une quantité inconnue (la variable réponse) comme une combinaison linéaire des valeurs des variables observées (les variables prédictives), basé sur le fait que la distribution est normale, le GLM permet, tout en conservant le même principe, d'utiliser d'autres modélisations.

En effet, si l'on reste sur une régression linéaire classique et que l'on présuppose nos données suivant une loi gaussienne, cela signifie qu'un changement 'a' chez une variable prédictive impliquera toujours un changement 'b' chez la variable réponse, 'a' et 'b' étant des constantes. Ce type de modèle est donc approprié seulement dans les cas où la variable réponse varie indéfiniment dans le sens positif ou négatif, ou alors varie par de petites variations (taille d'une plante d'espèce donnée par exemple).

Ainsi, si un individu appartient à la classe de référence (généralement la classe la plus densément peuplée de l'échantillon) pour toutes ses caractéristiques, alors il se verra facturer la prime moyenne. Ainsi, chaque variable β_j peut être vue comme un facteur de correction au prix de référence lorsque l'individu s'éloigne de cette classe de référence en présentant la caractéristique $x_{i,j}$. Fitter un GLM revient donc à estimer ces classes ainsi que la valeur de l'espérance sur chacune d'elles.

Néanmoins dans le cas actuariel, si l'on s'intéresse à la tarification automobile et aux montants que devraient payer les assurés pour être couvert, cette distribution est inadaptée, car dans la réalité il est impossible d'obtenir des valeurs négatives (l'assureur devrait de l'argent à un assuré après un sinistre) ; de plus les variations de tarifs peuvent être assez conséquentes, et donc les données un peu dispersées. De même, toujours dans cet exemple de tarification, si un assuré se voit augmenter son âge de permis de dix ans, ceteris paribus, on s'attendrait à ce que son tarif baisse de 25%. Au contraire, si l'on diminue de dix ans son âge de permis, alors il devrait s'attendre à payer quelque chose comme 25% de plus. Il s'agit alors d'un modèle à réponse exponentielle, ou modèle log-linéaire, puisque le logarithme de la réponse est prédit pour varier linéairement.

Les GLM permettent de prendre en compte cet aspect de notre distribution. Mais il est également performant dans d'autres domaines de l'actuariat, comme par exemple pour traiter les risques binaires (risque de défaut d'un assuré, basé sur ses caractéristiques propre). Là encore il permet d'aller plus loin qu'une régression linéaire classique.

Si les GLM peuvent s'adapter à ces différents cas et traiter de nombreuses distributions (Gamma, Poisson, Normale, Logistique, Binomiale, Gaussienne inverse, quasi-Poisson…), c'est grâce à l'utilisation de fonctions liens. Ces fonctions liens permettent de traiter linéairement les prédictions après transformation.

Ainsi, selon le type de problèmes rencontrés, on peut trouver sa variante adaptée de la régression linéaire : multiples imputations pour des données manquantes, inférence Bayésienne pour inclure des connaissance a priori dans le modèle, modèle de Poisson pour un modèle de comptage, Zero-Inflated Poisson (ZIP) régression si la valeur 0 est surreprésentée…

Néanmoins, dans le cadre de notre problématique, seule le cas de la réponse numérique nous intéresse. C'est pourquoi nous ne développerons en détail que le modèle traitant de la distribution exponentielle, qui s'appliquera ainsi sur le logarithme népérien de notre prime.

En effet, comme vu dans la partie II, le logarithme de la prime est plus adepte à respecter le principe de 'normalité' cité plus haut, et est nécessaire à l'implémentation d'un modèle linéaire.

III.1.2.1. Description mathématique

Un modèle GLM est défini par trois éléments : une distribution de probabilités appartenant à la famille exponentielle, un prédicteur linéaire et une fonction lien.

La famille exponentielle a été introduite par George Darmois, E.J.G. Pitman et Bernard Koopman en 1935. Il s'agit d'une classe de distributions de probabilités dont la forme générale est $f_T(t\ ;\ \theta, \phi) = exp\ \{\frac{t\theta - b(\theta)}{a(\phi)}\ +\ c(t; \phi)\}$, avec T une variable aléatoire.

a et b sont deux fonctions de $\mathbb{R}$, définies comme deux fois dérivables. c est une fonction définie sur $\mathbb{R}^2$. $\theta\ \in \mathbb{R}$ est le paramètre canonique, ou de la moyenne ; $\phi \in \mathbb{R}$ est lui le paramètre de dispersion

De multiples lois courantes appartiennent à la famille exponentielle, telles que les lois Normale, Exponentielle, Gamma, Chi-2, de Dirichlet, de Bernoulli, de Poisson, de Wishart… Toutefois certaines lois, comme la t-Student, les lois mélanges, ou les lois uniformes avec borne non fixée ne font pas partie de la famille exponentielle. Il est facile de mettre sous forme générale une distribution appartenant à la famille exponentielle grâce au tableau de la famille exponentielle (voir Annexe A.2).

Intéressons-nous maintenant au prédicteur linéaire. Il s'agit de la fonction qui incorpore l'information des variables prédictives. On pose $\ddot{X}\beta = \eta$ notre prédicteur linéaire avec $\ddot{X} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,q} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,q} \end{pmatrix}$ et $\beta = (\beta_0, \beta_1, \ldots, \beta_q)$

Le dernier élément constitutif d'un GLM est la fonction lien. Cette fonction permet de relier le prédicteur linéaire à la moyenne de la fonction de distribution. Chaque loi de la famille exponentielle dispose d'une fonction lien canonique, dérivée de l'exponentielle de la fonction de densité de la variable réponse. Toutefois l'usage de cette fonction canonique n'est pas imposé, car il est parfois plus intéressant d'utiliser une autre fonction lien, plus en accord avec le domaine de définition ou pour un but algorithmique. La seule différence est alors que, lorsque l'on utilise la fonction lien canonique, on a $\theta\ =\ \eta$, ce qui permet à $\ddot{X}^T Y$ d'être une statistique suffisante pour β (c'est-à-dire que l'on peut déterminer θ seulement avec les informations dont on dispose via $\ddot{X}^T Y$). Les fonctions liens canoniques peuvent se retrouver grâce au tableau de la famille exponentielle (voir Annexe A.2).

Maintenant que tout a été défini, nous pouvons revenir à l'expression générale d'un modèle linéaire généralisé.

La moyenne observée de la variable réponse Y est similaire à l'expression vue pour les modèles linéaires classiques, à l'exception faite de l'introduction de la fonction lien :

$$g : E(Y)\ =\ \mu\ = g^{-1}(\ \beta_0\ + \beta_1 \mathrm{X}_1 + \ldots + \beta_q X_q)\ = g^{-1}(\eta).$$

Réciproquement, on a $g(\mathrm{E}_\mathrm{Y}(y)\ )\ =\ \beta_0\ +\ \beta_1 \mathrm{X}_1 + \ldots + \beta_\mathrm{q} \mathrm{X}_\mathrm{q}$.

On a également, pour l'expression de la variance, $Var(Y)\ =\ V(\mu)\ = V(g^{-1}(\eta))$, où V désigne la fonction variance, dépendante de μ, et Var la variance comme définie statistiquement, c'est-à-dire $Var(Y) = E[(Y - \bar{Y})^2]$.

Si on considère comme fonction lien la fonction log alors on a :

$$ln(\mathrm{E}_Y(Y \mid \ddot{X})) = \beta_0 + \beta_1 X_1 + \ldots + \beta_q X_q$$

Et donc $\mathrm{E}_Y(y \mid x) = \exp(\beta_0 + \beta_1 \mathrm{X}_1 + \ldots + \beta_q \mathrm{X}_\mathrm{q})$. L'effet d'un changement de valeur d'une des variables est alors multiplicatif et non plus additif comme avec la régression linéaire traditionnelle.

III.1.3 Mise en oeuvre dans le cas saturé

Dans le contexte de notre base de données automobile, nous allons mettre en oeuvre la régression log-normale. En effet les lois log-normale et gamma sont les plus répandues dans la modélisation des coûts. Néanmoins les tarifs sont souvent construits comme un mélange de loi entre celle de coût et de fréquence (modélisée par une loi de Poisson, une binomiale négative ou tout autre type de lois de comptage), ce qui explique que nous ne retrouvions pas parfaitement une de ces lois dans nos données.

Nous utiliserons également la fonction lien log de manière à obtenir un tarif multiplicatif. Dans cette partie, nous ne nous attacherons pas à déterminer le modèle optimal de reverse engineering ; ce travail sera réalisé dans la partie IV de ce mémoire. Le seul objectif de cette partie est d'exposer la nécessité d'opérer une sélection de variables avant la création du modèle, de manière à obtenir un modèle performant.

Pour démontrer ceci, nous réaliserons différents modèles sur un échantillon d'entraînement (70% des observations de notre base de données) et nous fournirons également les résultats de performance obtenus sur des observations indépendantes (30% des observations, soit l'échantillon test).

Dans le but de pouvoir comparer légitimement les différents modèles produits par le GLM, et tout autre de ses variantes, nous utiliserons le critère d'information d'Akaïke (critère AIC) qui cherche à maximiser la vraisemblance du modèle, tout en pénalisant l'utilisation d'un trop grand nombre de variable afin de satisfaire au critère de parcimonie ('les hypothèses suffisantes les plus simples doivent être préférées'). Le critère de parcimonie sert à limiter l'effet de sur-ajustement qui pourrait exister. L'AIC est asymptotiquement optimal lorsque l'on souhaite sélectionner le modèle avec l'erreur quadratique moyenne ; de plus, contrairement au BIC (critère d'information Bayésien), le critère AIC permet de choisir le meilleur modèle quand ce modèle n'est pas forcément présent dans le set de modèles essayés.

Le critère d'AIC s'exprime sous la forme suivante :

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

Où k représente le nombre de variables/modalités utilisées et $\ln(L)$ la log-vraisemblance du modèle. La fonction de vraisemblance d'un modèle est une fonction des paramètres de ce modèle (ici, notre paramètre est $\beta = (\beta_0, \beta_1, \ldots, \beta_q)$) calculée à partir des données observées. Ces fonctions de vraisemblances jouent un rôle clé dans l'inférence statistique fréquentiste, notamment pour les méthodes statistiques d'estimation de paramètres.

On écrit alors la fonction vraisemblance comme $L(\beta|\ddot{X}) = \prod_{i=1}^{n} L(\beta|X^{(i)}) = \prod_{i=1}^{n} f_\beta(X^{(i)})$ où $f_\beta(X)$ est la fonction de densité de probabilité de la variable X. La log-vraisemblance nécessaire dans le calcul du critère d'Akaïke se présente alors comme $\ln\left(L(\beta|\ddot{X})\right) = \sum_{i=1}^{n} f_\beta(X^{(i)})$.

Dès lors on obtient l'équivalence suivante au sujet de la performance de nos modèles :

$$Modèle\ 1 > Modèle\ 2 \Leftrightarrow AIC_1 < AIC_2$$

III.1.3.1. Evaluation de la performance de différents modèles

Nous allons donc calculer une régression linéaire classique ainsi que des GLMs sur nos données d'entraînement. En utilisant ces modèles sur notre échantillon de test, nous serons capables d'évaluer la performance desdits modèles. Nous performerons dans un premier temps le modèle sur toutes les variables présentes à l'origine dans la base de données.

Nous obtenons ainsi ces résultats pour un modèle linéaire classique :

```
Residual standard error: 61.77 on 73804 degrees of freedom
  (3113 observations deleted due to missingness)
Multiple R-squared:  0.681,     Adjusted R-squared:  0.6807
F-statistic:  1970 on 80 and 73804 DF,  p-value: < 2.2e-16


AIC is 819077.69033
```

Et ces résultats avec un modèle linéaire généralisé :

```
---------------------------------------------
No. of observations in the fit:  73885
Degrees of Freedom for the fit:  82
      Residual Deg. of Freedom:  73803
                      at cycle:  2

Global Deviance:     815118.1
            AIC:     815282.1
            SBC:     816037.3
*********************************************
```

On voit que le modèle linéaire généralisé est meilleur que la simple régression linéaire (critère AIC plus petit). Cela conforte donc notre choix de ne plus utiliser par la suite la régression linéaire classique.

En s'attardant sur le modèle GLM, on peut voir que pour certaines variables on obtient une non-significativité. En effet, R nous indique la significativité d'une variable ou d'une modalité par un code visuel. *** signifie que que la p-value est inférieure à 0.0001, ** à 0.001, * à 0.05, . à 0.1.

La p-value d'un coefficient est calculée comme :

$$2 \times P[Z < (-1) \times |\frac{Estimate}{Standard\ Error}|]$$

Avec Z une loi normale centrée réduite. Si cette p-value est petite alors on peut rejeter H_0, c'est-à-dire l'hypothèse que le coefficient déterminé n'est pas asymptotiquement normal. Si le coefficient est asymptotiquement normal alors il est stable donc significatif.

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	1.716e+00	1.339e-03	1281.690	< 2e-16	***
depreciation2	5.457e-03	2.640e-04	20.672	< 2e-16	***
depreciation3	2.516e-02	2.788e-04	90.246	< 2e-16	***
driver_age	-2.487e-03	1.387e-05	-179.252	< 2e-16	***
sex2	-3.487e-03	2.822e-04	-12.359	< 2e-16	***
license_type1	-4.017e-02	9.147e-03	-4.392	1.13e-05	***
license_type2	2.134e-02	2.021e-02	1.056	0.291080	
license_type3	5.316e-03	1.726e-03	3.079	0.002075	**
license_type4	2.134e-02	2.021e-02	1.056	0.291080	
license_type5	2.134e-02	2.021e-02	1.056	0.291080	
marital_status2	-1.005e-02	1.483e-03	-6.772	1.28e-11	***
marital_status3	8.870e-05	4.673e-03	0.019	0.984855	
marital_status5	-5.704e-03	5.855e-04	-9.743	< 2e-16	***
marital_status6	-1.794e-03	9.762e-04	-1.838	0.066127	.
province1	2.255e-02	5.660e-04	39.850	< 2e-16	***
province2	-4.944e-03	3.821e-04	-12.940	< 2e-16	***
province3	2.315e-02	4.853e-04	47.710	< 2e-16	***
province4	3.433e-02	4.796e-04	71.582	< 2e-16	***
province5	3.299e-02	4.620e-04	71.400	< 2e-16	***
province6	-1.091e-02	4.449e-04	-24.523	< 2e-16	***
province7	-1.070e-02	5.142e-04	-20.815	< 2e-16	***
province8	2.568e-02	6.647e-04	38.635	< 2e-16	***
province9	1.492e-02	8.828e-04	16.900	< 2e-16	***
province10	4.611e-02	4.868e-04	94.721	< 2e-16	***
travail2	-1.043e-03	4.365e-04	-2.391	0.016824	*
travail3	2.023e-03	4.433e-04	4.563	5.05e-06	***
travail4	1.348e-02	1.732e-03	7.778	7.42e-15	***
travail5	4.153e-03	2.385e-03	1.741	0.081698	.
travail6	-5.006e-04	6.156e-04	-0.813	0.416121	
travail7	1.980e-04	9.250e-04	0.214	0.830469	
travail9	1.064e-02	1.637e-03	6.496	8.28e-11	***
nr_accident_not_at_fault	3.476e-03	5.473e-04	6.351	2.15e-10	***

Figure 25 : Coefficients obtenus par GLM

De nombreuses variables sont donc étiquetées comme non-significative. Cela est compréhensible, compte tenu de l'analyse réalisée dans la partie II. Le principe de parcimonie n'est pas mis en application ici. On va donc déterminer un nouveau modèle, cette fois-ci uniquement sur les 30 variables sélectionnées à l'issue de la partie II. Les résultats sont présentés ci-dessous :

```
---------------------------------------------
No. of observations in the fit:  73885
Degrees of Freedom for the fit:  71
      Residual Deg. of Freedom:  73814
                      at cycle:  2

Global Deviance:     825560.3
            AIC:     825702.3
            SBC:     826356.2
*********************************************
```

La performance du GLM a régressé de manière très légère (AIC : +1.3%) mais cela n'est pas très important, puisque l'on a tout au moins, réduit le nombre de variables utilisées. Ce genre de démarche est particulièrement utile pour obtenir un modèle plus synthétique et plus interprétable. Néanmoins, une analyse plus poussée du modèle montre que de nombreuses variables/modalités demeurent non-significatives. Certaines variables présentent également trop de modalités, ce qui nuit à leur significativité globale. Il faut pallier à ce problème si on veut obtenir un modèle aux performances optimales.

La suite de ce chapitre présente donc différentes méthodes de sélection de variables visant à améliorer les performances des algorithmes qui seront performés dans la partie IV de ce mémoire.

III.2. Méthodes classiques de sélection : wrapper, embedded, filtres

Nous allons maintenant passer à travers différentes méthodes de sélection de variable. Le but de ces méthodes est de réduire le set de variables de manière à, en finalité, obtenir le meilleur modèle, tout en évitant un sur-ajustement (ou overfitting) trop notable.

En effet, l'overfitting est un défi majeur en modélisation. Son but est que le modèle développé soit le meilleur à prédire une prime sur un échantillon quelconque de clients. Mais chaque client est différent et le nombre de combinaisons possibles de variables est supérieur à plusieurs billions. Il est donc fort peu probable qu'un profil connu réapparaisse à l'identique dans notre échantillon test.

Ceci en tête, il arrive parfois que le modèle soit confronté à des exceptions ou des 'erreurs' lorsqu'il est entraîné. Le modèle, qui cherche à copier au mieux ce qu'il voit dans les données d'entraînement, est alors influencé par les exceptions, et essaye alors de les comprendre et de s'y adapter. Quand cela arrive de manière régulière, le modèle devient alors trop spécifique et perd de son pouvoir de prédiction dans le cas général. Plus l'algorithme dispose de données, plus de cas 'exceptionnels' sont susceptibles d'apparaître et plus le modèle sera prompt à tenter de déceler chaque subtilités présentes.

A contrario, même si nous cherchons in fine à limiter le nombre de variables, il est important de prendre conscience que nos variables peuvent induire plus que leur simple contribution personnelle. En effet on peut assister à des effets quadratiques (prise en compte de l'âge au carré dans le tarif par exemple) ou des effets croisés (une jeune assurée peut être moins risquée que ce que l'effet 'jeune' et 'femme' peut laisser présumer). Cela revient à ajouter des 'variables' à notre modèle. Il faut donc trouver un juste milieu entre ces deux effets.

Nous allons parcourir pour cela différentes méthodes. On citera les familles des méthodes wrapper, embedded et des filtres. Les methodes wrapper et embedded sont à différencier des filtres

dans le sens où elles sélectionnent les variables en fonction de la procédure de prédiction utilisée en aval de la sélection ; les filtres eux peuvent se calculer une unique fois et sont indépendants de l'algorithme utilisé dans la prédiction. Leurs différences majeures résident dans leur temps de calcul et leur faculté ou non à détecter les interactions entre variables.

III.2.1 Filtres

Les filtres sont un moyen d'affiner les variables utiles dans la constitution d'un modèle efficace. Ces techniques se basent généralement sur des outils statistiques bien connus. Les filtres sont ainsi souvent utilisés comme traitement préliminaire rapide, pour dégrossir le travail de sélection, afin d'appliquer ensuite d'autres méthodes plus raffinées. Ils ont également l'avantage de ne pas nécessiter de modèle défini en amont, ce qui signifie que les données n'ont pas à présenter de pattern caractéristique pour qu'on puisse les utiliser. Cela leur permet également d'être performant au niveau du temps de calcul. De plus les résultats fournis par les filtres sont généralement faciles à interpréter.

Les méthodes utilisées dans la partie II de ce travail sont en réalité des filtres. Nous avons ainsi successivement filtré selon les valeurs manquantes de $\ddot{X}$, puis sur la corrélation / dépendance des X_j entre eux.

D'autres filtres existent également, comme certains éliminant les variables ayant un faible coefficient de variation ($\frac{\sqrt{Var(X_j)}}{E(X_j)}$ petit) pour les variables numériques ou un trop grand nombre de données de la même classe. Ces filtres n'ont pas été appliqués ici car il est difficile d'établir un seuil pour lequel la variable ne semble pas apporter suffisamment d'informations. De même, il est possible que la base contienne des variables dont une ou plusieurs des classes est très peu représentée, mais a, au final, une grande influence sur la variable réponse. On peut notamment penser à la variable déterminant si une voiture est dite sportive ou non. Ces modèles de véhicule ont en effet tendance à être plus chers que ceux plus classiques.

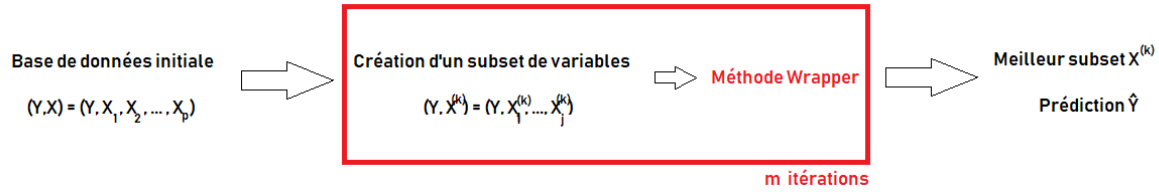
Dans la partie II, nous aurions pu également étudier la relation entre nos variables explicatives et notre variable réponse. Comme il l'a été montré dans la partie III.1, toutes les variables (mais pas forcément toutes leurs modalités) sont significatives au sens du GLM. Cela veut dire qu'elles sont liées à notre variable réponse. Un tel filtre serait donc superflu dans cet état des choses. L'un des désavantages de ce genre de filtre est aussi sa difficulté, lorsque les classiques tests de Spearman et de Pearson sont utilisés, à détecter des interactions qui dérogent à de simples relations linéaires. Ces tests ont en effet du mal à détecter des relations plus complexes, telles que des liaisons quadratiques.

Ces relations quadratiques sont assez dangereuses dans le sens où elles ne sont jamais évidentes, à la fois à mettre en place, mais aussi à détecter. En effet, à mesure que le degré pris par une variable augmente dans un modèle, on perd en interprétabilité. Ces interactions sont de plus, rarement testées et envisagées, car trop nombreuses.

Les filtres sont également impuissants à détecter des interactions d'ordre du second degré entre deux variables et la variable réponse. Il est impossible de déterminer l'impact d'une combinaison de deux variables sur notre variable réponse, sans faire abstraction des effets indépendants de ces variables sur la réponse. C'est pourquoi on a souvent recours à des méthodes de sélection plus performantes, comme nous le verrons par la suite.

Ainsi les filtres sont au final moins raffinés que les deux autres méthodes de sélection qui seront développées par la suite. Ils restent néanmoins performants dans leur travail préliminaire d'éclaircissement.

III.2.2 Wrapper méthodes



Les méthodes wrapper (méthode d'enveloppe) entraînent un large ensemble de modèles. Ils sont donc assez fastidieux à mettre en place, car exigeant beaucoup de ressources.

Le principe des méthodes wrapper est de construire un grand panel de modèles, mais nécessite utiliser différentes combinaisons de variables à chaque fois. On ne conserve ensuite que les modèles ayant la plus forte performance au regard du critère choisi. Parmi ce sous-ensemble, on peut soit opter pour le modèle ayant le fort pouvoir prédictif, soit opter pour le modèle dont la combinaison de variables nous semble la plus plausible.

Cette approche est efficace si tous les sous-ensembles de données sont considérés ; dans la réalité, cette approche est très coûteuse en temps et en ressource puisqu'elle implique d'entraîner autant de modèles que de sous-ensemble de variables (2^n modèles, soient 16 modèles pour 4 variables, 32 pour 5 variables…). Elle l'est encore plus lorsque l'on considère même tester des regroupements au sein de modalités. C'est pourquoi on parle souvent d'algorithmes gloutons, quand on se réfère à ces méthodes, qui permettent de diminuer le nombre de sous-ensembles. Un algorithme glouton étant un algorithme qui permet de réaliser, étape par étape, un choix optimum local.

De plus, si on en vient à considérer les interactions au sein d'une même variable, on se retrouve avec des modèles à croissance exponentielle. Il est donc peu recommandé d'utiliser les méthodes wrapper sur des bases de données de plus d'une dizaine de variables.

Les méthodes wrapper se contentent donc souvent d'approximation sur un échantillon de modèles, à défaut de pouvoir tester tous les modèles possibles. Elles ont également le défaut de pouvoir overfitter un modèle.

Les méthodes wrapper ont besoin d'une méthode de recherche prédéfinie pour être appliquées ; les plus connues sont les méthodes stepwise de Forward feature selection (ajout d'une variable une à une), Backward feature elimination (on part d'un modèle complet et on retire au fur et à mesure une des variables) et Exhaustive Feature Selection (on essaye toutes les combinaisons possibles).

Cependant, les résultats de la régression stepwise sont fréquemment exploités de manière impropre, c'est-à-dire qu'ils ne prennent pas en compte les implications de la procédure de sélection. En effet, les estimateurs et les intervalles de confiance liés à ces derniers sont propres au modèle final, issu de cette procédure, et ne sont donc pas ajustés pour refléter convenablement l'incertitude liée à cette sélection. Les tests de significativité entre deux itérations sont biaisés, puisqu'ils sont basés sur les mêmes données, et donc leurs résultats sont donc mécaniquement améliorés. Cela invalide les hypothèses de distribution sur lesquelles s'appuient ces tests, puisque la sélection de variable conduit

à produire des modèles dont l'erreur est davantage corrélée aux régresseurs, en comparaison du modèle final.

Beaucoup d'algorithmes utilisent les méthodes wrapper. C'est ainsi que sont traditionnellement construits les GLM. Les algorithmes génétiques ainsi que la recherche locale itérée font également partie de cette grande famille utilisant les wrappers. En pratique, n'importe quelle combinaison d'une stratégie de recherche avec un algorithme de machine learning ou de modélisation peut être utilisée comme méthode wrapper.

Le processus d'une méthode wrapper est assez simple :

- On produit à l'aide d'une méthode de recherche un sous-ensemble de variables explicatives parmi celles dont on dispose
- On construit un modèle
- On évalue le modèle sur ses performances
- On répète toutes ces étapes

On s'arrête lorsque la condition d'arrêt est respectée (pas d'amélioration des performances sur les $k^{èmes}$ derniers modèles, nombre de modèles pré-définis…).

Au-delà des problèmes de temps de calcul, les méthodes traditionnelles de recherche ont quelques défauts de construction. On peut notamment souligner le fait que, pour les méthodes Forward, il est possible qu'une variable utile au début, quand le modèle est constitué de peu de variables, ne le soit plus après en avoir ajouté quelques autres. Le même problème peut se poser avec la sélection Backward. Des alternatives existent, on pourra citer la recherche bidirectionnelle (combinaison de Forward et Backward, on évalue l'ajout / la soustraction de chaque variable à chaque itération) ou la méthode LRS.

Nous avons donc testé cette méthode sur notre GLM introduit dans la partie III.1. On a choisi comme critère de sélection l'AIC et comme critère d'arrêt 1000 étapes, dans le cas où l'algorithme ne converge pas plus tôt.

Suivants les méthodes de recherche utilisées, on trouve donc que les variables suivantes ont été retirées du modèle :

	Méthode Stepward	Méthode Backward	Both (Bidirectionnelle)
Variables originelles	- Emission de CO^2	- Région - Kilométrage - Est un break ? - Emission de CO^2 - Nombre d'accidents en cours	- Région - Emission de CO^2 - Est un break ? - Type de parking - Nombre d'accidents en cours
Variables après filtres réalisés en partie II		- Nombre d'accidents en cours	- Nombre d'accidents en cours

On s'aperçoit donc que les résultats obtenus sont concordants avec ceux obtenus à l'aide de filtres. Néanmoins, on remarque que la méthode stepward, c'est-à-dire avec l'ajout successif d'une variable par une variable, ne filtre quasiment pas les variables. On voit également quelques disparités

selon la méthode employée (backward ou stepward). Avec la méthode backward on présume que le kilométrage de la voiture n'est pas important dans la déduction du tarif incombé à l'assuré, ce qui est peu probable en réalité. Cette méthode nécessite donc toujours un jugement humain avant de pouvoir prendre des décisions. On remarquera également, que contrairement à ce qui a été statué au début de ce chapitre, la variable 'sexe' continue d'être sélectionnée comme variable discriminante quel que soit la méthode employée.

De plus, pour des raisons de temps de calcul, les interactions entre variables n'ont pas été introduites ici. On verra dans la partie III.2.3. comment les introduire de façon optimisées et sans nuisance pour le temps de calcul.

On remarque également que la méthode de recherche influe sur la performance finale du modèle. On va donc essayer de trouver des méthodes alternatives permettant de réduire cette volatilité.

III.2.3 Embedded Méthodes

Dans la présentation des méthodes wrapper nous avons vu que celles-ci demandaient l'entraînement de nombreux modèles, et étaient, de fait, très gourmandes en temps de calcul. En pratique, ces méthodes sont donc peu envisageables à grande échelle, sur de grosses bases de données comportant de nombreuses variables explicatives. Dans notre cas, avec nos 30 variables, la dimension est encore exploitable, mais si l'on rajoute des interactions, même en se limitant au niveau 2, on se retrouve avec $\binom{30}{2} + 2 * \binom{30}{1} = 435 + 2 * 30 = 495$ variables à prendre en compte.

C'est pour cela que l'on a décidé d'envisager d'inclure la partie de sélection des variables à l'intérieur même du modèle ; il s'agit de l'essence même des méthodes embedded (méthode intégrée). Les méthodes embedded tentent ainsi de sélectionner au cours de l'entraînement même du modèle prédictif les variables les plus intéressantes. Un exemple typique en est le Random Forest, qui sera développé plus avant dans la partie IV.3.2.

La méthode embedded fonctionne comme suit :

- Un modèle de machine learning est entraîné
- L'importance des variable est calculée, c'est-à-dire à quel point chaque variable contribue à la prédiction
- Les variables contribuant peu ou pas du tout sont supprimées

Les méthodes les plus souvent appliquées sont les méthodes de régularisation, qui seront développées dans la partie suivante de ce chapitre, et les méthodes reposant sur des arbres. On notera que l'utilisation des arbres permet de ne pas inclure manuellement toutes les interactions entre les variables, car elles sont directement prise en compte dans la structure de l'arbre. Néanmoins, il est difficile d'avoir une vue globale de la contribution individuelle de chaque variable, puisque cette information est dissimulée dans sa contribution globale, formée de cette contribution, et de celle apportée lorsque cette variable est combinée.

III.2.3.1. Importance des variables

Nous allons ici nous concentrer sur les fonctionnalités offertes par les algorithmes de machine learning. En effet, de plus en plus d'algorithmes offrent la possibilité d'obtenir un diagramme ou des statistiques sur l'utilisation des variables qui a été faites lorsque le modèle a été créé.

Par exemple, si l'on prend des modèles basés sur des arbres (Arbre de décision, forêt d'arbres aléatoires, GBM…), l'importance de chaque variable est déterminée en calculant leur influence relative : c'est-à-dire si la variable a été sélectionnée durant le processus de construction de l'arbre, et si elle l'a été, de combien l'erreur au carré a diminué par son utilisation. Cela revient à soustraire l'erreur au carré des nœuds-fils à l'erreur au carré du nœud-père. Cette erreur au carré étant la réduction de la variance de la variable réponse à l'intérieur de ce nœud. Ou plus simplement :

$$Squared\ Error = MSE * N = VAR * N \text{ où } VAR = \frac{1}{N} * \sum_{i=1}^{N} (y_i - \hat{y_i})^2$$

Ces différentes méthodes, utilisant des algorithmes à base d'arbres, seront expliquées plus en détail dans la partie IV de cet ouvrage. Ici, nous n'utiliserons que leur aspect résultat sur l'importance des variables, sans nous attarder sur le fonctionnement de ces différents algorithmes. En effet, ce n'est pas notre propos dans cette section. Ainsi, nous ne rentrerons pas dans la phase de fine-tuning, c'est-à-dire de sélection des meilleurs paramètres, et nous nous contenterons d'effectuer une validation croisée à 5 folds (technique développée en IV.1.3.) pour limiter le phénomène d'overfitting des données.

Ci-dessous, on trouvera ainsi un graphique d'importance des variables de notre base de données, réalisée avec une forêt d'arbre aléatoires et avec un GBM (Gradient Boosting Method, algorithme détaillé dans la partie IV.3.2), sur la base de données comprenant toutes les variables originelles. Dans le but d'avoir une meilleure visibilité, seules les 20 variables les plus influentes ont été conservées :

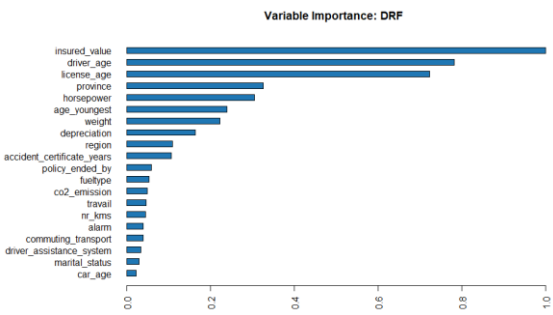


Figure 26: Importance des variables par forêt aléatoire

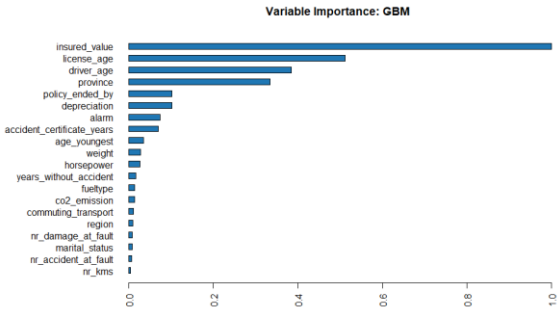
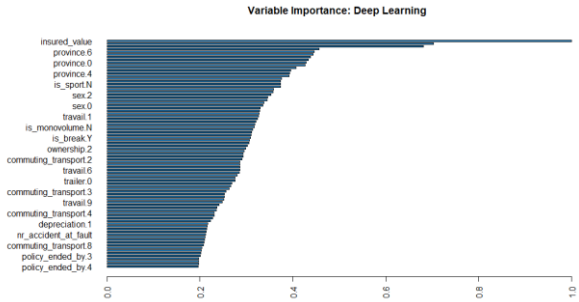


Figure 27: Importance des variables par GBM

Si l'on considère d'autres modèles, qui ne reposent pas cette fois-ci sur l'utilisation d'arbres, comme les réseaux de neurones, on peut obtenir un parallèle de cette importance des variables via les poids utilisés dans chaque couche de neurones. Ici, on a utilisé la méthode de Gédéon, qui utilise les poids des deux premières couches cachées du réseau pour déduire l'importance relative des différentes variables.



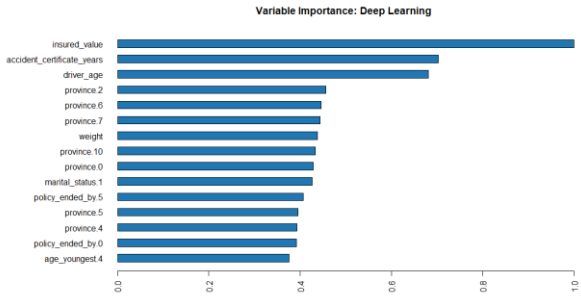


Figure 28: Importance des variables par réseau de neurones *Figure 29: Importance des variables par réseau de neurones (15)*

On voit sur ces graphes qu'il y a très clairement trois variables prépondérantes dans la justification du tarif proposé à un prospect : la valeur assurée du véhicule, le nombre d'années sans accidents et l'âge de l'assuré. Si on en croit le premier graphique, présenté à gauche, presque toutes les variables ont toutefois une importance non négligeable. Cela vient en partie du fait que la méthode de Gédéon utilise les poids des deux premières couches cachées, et donc plus ou moins toutes les variables d'entrées. Quand on s'intéresse aux 15 variables ayant la plus grande importance au sens de la méthode de Gédéon, on retrouve néanmoins les mêmes variables que celles obtenues à travers l'algorithme de forêt aléatoire ou celui de GBM.

Ainsi cette méthode ne fournit pas toujours des résultats 100% fiables, c'est pourquoi il est préférable de se tourner vers d'autres modèles pour avoir une idée de l'importance des différentes variables.

On peut voir à nouveau que notre sélection réalisée par filtres était justifiée ; néanmoins, tout comme avec les méthodes wrapper, les différentes méthodes n'accordent pas le même poids à chaque variable. Cela peut provenir du fait que les algorithmes ne sont pas affinés de manière à être les plus optimaux possibles, mais sont juste utilisés à titre indicatif.

Nous avons donc fixé un seuil relatif de 0.01. Toute variable n'atteignant pas cette importance (normalisée par rapport à 1) sera donc soustraite du modèle. On retrouve alors le rejet de différentes variables, résumé ci-dessous :

	Random Forest	GBM
Variables originelles	- Sexe - Première souscription ? - Voiture de sport ? - Voiture cabriolet ? - Voiture de type monospace ? - Voiture 4x4 ? - Propriétaire ou non - Nb d'accidents non responsables - Présence d'une remorque ? - Catégorie du permis - Nombre de retraits de permis - Nb d'accidents en cours - Nb de tentatives de vol	- Sexe - Première souscription ? - Voiture de sport ? - Voiture cabriolet ? - Voiture 4x4 ? - Propriétaire ou non - Nb d'accidents non-responsables - Présence d'une remorque ? - Catégorie du permis - Nombre de retraits de permis - Nb d'accidents en cours - Nb de tentatives de vol - Région - Emission de Co^2 - Travail - Kilométrage - Présence d'une alarme - Moyen de transport pour aller au travail - Nombre d'appareils d'aide à la conduite - Statut marital - Age de la voiture - Situation du parking - Utilisation

		- Nb de dommages responsables - Nb de sièges
Variables après filtres réalisés en partie II	- Voiture type monospace ? - Voiture type break ? - Voiture cabriolet ? - Propriétaire ou non - Nb d'accidents non-responsables - Présence d'une remorque ? - Type de permis - Nombre de retraits de permis - Nb d'accidents en cours - Nb de tentatives de vol	- Voiture type monospace ? - Voiture type break ? - Voiture cabriolet ? - Propriétaire ou non - Nb d'accidents non-responsables - Présence d'une remorque ? - Type de permis - Nombre de retraits de permis - Nb d'accidents en cours - Nb de tentatives de vol - Travail - Kilométrage - Nombre d'appareils d'aide à la conduite - Age de la voiture - Voiture de sport ? - Première souscription ? - Sexe

On notera que la colonne pour le réseau de neurones ne figure pas dans le tableau précédent. Cela vient du fait que, puisque non seulement le réseau de neurones ne traite pas de la même manière les variables (l'utilisation d'un réseau de neurones nécessite que toutes les modalités soient encodées séparément), causant alors une répartition malvenue de l'importance globale d'une variable, son interprétation diffère des méthodes avec arbres.

Ainsi, l'étude du graphe précédent ne permet pas réellement de conduire au rejet de certaines variables / modalités, excepté la variable de la valeur de l'émission de CO^2 du véhicule.

On s'aperçoit également que le modèle du GBM est beaucoup plus 'sélectif' que l'algorithme de forêt aléatoire, dans le sens où il nécessite moins de variables que son alternative. On verra dans la partie IV de ce mémoire si cette sélectivité vient avec une perte de performance ou non. En effet, si l'on en croit le GBM, la variable 'Age de la voiture' apporte peu d'informations sur la constitution du tarif assurantiel. Cela nécessite un jugement humain post-étude pour vérifier cette conjecture faite par l'algorithme.

III.2.3.2. Comparaison des différentes méthodes

Pour terminer sur cette première partie présentant les méthodes les plus classiques de sélection de variables, celles-ci ont été regroupées dans un tableau répertoriant leurs forces et leurs faiblesses. Les méthodes embedded étant les méthodes présentant les avantages les plus intéressants et le moins d'inconvénients, elles seront étudiées plus en profondeur et présentées sous diverses formes dans le prochain chapitre.

Type de méthode	Avantages	Inconvénients	Exemples de méthode
Filtres	- Rapide - Indépendant de la méthode de prédiction - Robuste à l'overfitting	- Difficulté à déceler les interactions du second ordre ou au-dessus	- Tests de corrélations - Tests de vraisemblance - Test du χ^2
Wrapper	- Interagit avec la méthode de prédiction - Modélise la dépendance des variables - Produit le sous-set de variables optimal	- Risque d'overfitting - Sélection dépendante de la méthode de prédiction - Temps de calcul	- Sélections forward, backward, bidirectionnelle - Arbres de décision

Embedded	- Interagit avec la méthode de prédiction - Tient compte des interactions entre les variables explicatives - Plus rapide que les wrapper - Plus performante que les filtres - Peu enclines à l'overfitting	- Sélection dépendante de la méthode de prédiction	- Random Forest - LASSO - Ridge regression - Elastic-net

On ajoutera également que cette analyse aurait pu se faire sur une base de données plus restreinte, c'est-à-dire en utilisant une base de données où seule un type de dépréciation aurait été maintenu. Néanmoins, après une étude menée de manière plus limitée, il s'est avéré que faire cette segmentation à ce niveau n'influait pas sur les résultats donnant les variables à négliger.

III.3. Méthodes Ridge et Lasso

Nous allons maintenant nous intéresser à deux processus, qui sont des cas particuliers des méthodes intégrées, et à leurs dérivés. Ces processus sont basés sur le concept de régularisation dans le but de diminuer le nombre de paramètres à inclure dans un modèle ; cela permet de rendre ce dernier plus simple et de réduire le risque d'overfitting.

Dans la plupart des modèles il est nécessaire d'aboutir à un compromis entre performance et reproductibilité (bias-variance trade-off). En effet, plus un modèle dispose d'informations, plus il aura matière à s'entraîner et plus son erreur tendra vers 0. Ce qui est en théorie ce que l'on cherche à obtenir. Mais dans les faits, ce modèle sur-entraîné est trop complexe pour la réalité et ce dernier a appris et mémorisé des patterns et du bruit qui n'apparaîtront probablement pas dans les autres échantillons sur lesquels on veut exercer notre modèle dans le futur. De plus un modèle qui overfitte est dépendant de l'échantillon sur lequel on l'a entraîné. On ne peut donc pas en conclure une généralité ; hors dans notre cas, le but est de pouvoir déployer les modèles trouvés à long terme, et sur toute une gamme de clients.

Une des méthodes pour limiter cet overfitting consiste à sélectionner notre modèle, et donc nos variables, en ajoutant une pénalité aux différents paramètres, de manière à réduire le nombre de degrés de libertés du modèles (le nombre de valeurs pouvant être prises par chaque variable). Cette pénalité est appliquée à tous les coefficient de notre modèle linéaire ; cela rend le modèle plus robuste au bruit éventuel, améliorant ainsi sa généralisation.

Les deux principales méthodes de régularisation, les régressions LASSO et Ridge, seront ici développées. Une fois le lecteur familier avec ces notions, certaines de leurs dérivées seront également étudiés. On précisera également que d''autres méthodes de régression existent, comme les algorithmes LARS (Least Angle Regression) ou OLS (Ordinary Least Squares) ; néanmoins par souci de simplicité, ceux-ci ne seront pas développés plus avant dans cet ouvrage.

Dans tout ce chapitre les calculs se feront sur la base de données présentant toutes les variables originelles (c'est-à-dire $p = 40$, et on a toujours $n = 78\,998$).

III.3.1 Régression ridge

La première méthode de contraction des coefficients présentée ici est la méthode dite de régression d'arête (ridge regression) ou régression l_2 car relatif à la norme 2. Proposée par le mathématicien russe Andreï Nikolaïevitch Tikhonov dans les années 60, elle est connue comme étant

la méthode la plus employée pour résoudre les problèmes qui ne sont pas bien posés (c'est-à-dire qui ne respectent pas la définition de Hadamard) ou les problèmes inverses (où on détermine la cause par les effets).

Si l'on s'abstient de rentrer dans sa formulation mathématique, on peut ramener une régression ridge à une contraction des variables ayant la plus petite variance au sein de leur plage de données. De manière similaire à l'analyse en composantes principales (ACP, qui sera développée dans la partie III.4.4.), la régression ridge projette les données dans un espace $\mathbb{R}^p$et réduit la valeur des coefficients des composantes à basse variance.

La régression ridge est particulièrement utilisée dans l'analyse des données de régressions multiples qui souffrent de la multicolinéarité. En effet, lorsque les variables sont fortement corrélées entre elles, la méthode des moindres carrés ordinaires (Ordinary Least Squares, OLS), largement utilisée dans les problèmes d'optimisation, n'est plus aussi efficace. Ainsi, même si l'estimateur reste non biaisé, la variance a tendance à grandir de manière démesurée à cause des corrélations. L'estimateur donné par OLS n'est alors plus fiable.

La régression ridge s'appuie sur la méthode des moindres carrés ordinaires ; elle ajoute un degré de biais, de manière à réduire la variance de l'estimateur. La méthode OLS consiste à minimiser la valeur de :

$$\sum_{i=1}^{n} (\mathrm{X}^{(\mathrm{i})}\beta - y_i)^2 = \sum_{i=1}^{n} (\hat{y_i} - y_i)^2$$

Cela revient donc, en d'autres termes, à minimiser la distance moyenne entre les valeurs prédites et les vraies valeurs.

De manière plus concrète, la régression ridge est une méthode des moindres carrés ordinaires sous la contrainte :

$$\sum_{i=1}^{p} \beta_i^2 \leq c$$

Elle vise donc à limiter le poids et/ou le nombre des coefficients utilisés. De cette manière, l'impact de la multicolinéarité est fortement réduit.

La régression ridge ajoute un terme dans l'expression à minimiser : Γ la matrice de Tikhonov. La formule à optimiser devient alors :

$$||\ddot{X}\beta - Y||^2 + ||\Gamma\ddot{X}||^2 = (Y - Z\beta)^T(Y - Z\beta) + \lambda(\beta^T\beta - c)$$

La matrice Z correspondant à la matrice $\ddot{X}$ centrée et normalisée ; cela permet aux variables à fortes variances (comme l'âge du conducteur par exemple) d'avoir une influence limitée. Y doit alors également être centré. λ correspond au paramètre ridge et est directement lié à la matrice de Tikhonov.

L'estimateur donné par la méthode Ridge admet une forme analytique :

$$\hat{\beta}_R = \left(\ddot{X}^T\ddot{X} + \lambda I\right)^{-1}\ddot{X}^T Y$$

Toutefois, sans Γ, il est impossible de déterminer $\hat{\beta}$. Pour calculer Γ plusieurs méthodes existent mais aucune ne garantit la fiabilité du résultat.

L'usage le plus courant est alors de trouver Γ de manière empirique, en faisant varier ses valeurs et en observant les répercussions sur la valeur de $\hat{\beta}$. Lorsque les coefficients de $\hat{\beta}$ se sont

stabilisés, on peut alors supposer que l'on a trouvé la valeur de Γ optimale. Sinon, on peut calculer Γ, mais cela revient à admettre des hypothèses simplificatrices. Parmi ces méthodes, on peut citer celles de Hoerl, Kennard et Baldwin $\lambda_{HKB} = \frac{(p-2)\hat{\sigma}_\varepsilon^2}{\sum_{j=1}^{p}\hat{\beta}_j^2}$, ou encore Lawless et Wang $\lambda_{LW} = \frac{n(p-2)\hat{\sigma}_\varepsilon^2}{\sum_{j=1}^{n}\hat{y}_i^2}$. p est ainsi le nombre de paramètres du modèles et $\hat{\sigma}_\varepsilon^2$ la variance du modèle.

L'un des principaux défauts de cette méthode réside dans le fait que la qualité de prédiction du modèle n'est pas prise en compte dans l'établissement de Γ. De plus, déterminer la matrice de Tikhonov est assez compliqué, et repose sur une étude empirique, basée elle-même sur des estimations établies à partir de la méthode des moindres carrés ordinaires.

III.3.1.1. Application de la méthode ridge

La régression ridge a été appliquée sur l'ensemble de nos données. Dans un premier temps, il a fallu déterminer λ. Pour ce faire trois méthodes ont été utilisées. Ces méthodes sont celles brièvement introduites en amont, auxquelles on a ajouté, pour comparaison, le λ trouvé par validation croisée sous R, technique qui sera développée dans la partie IV.1.

Méthode	Hoerl, Kennard, Baldwin	Lawless, Wang	Validation croisée
Valeur de λ	1.06	1.15	0.79

Une fois λ obtenu, un nouveau GLM est lancé, prenant en compte cette pénalisation. Nous allons comparer nos résultats selon le critère RMSE (le choix des critères sera expliqué plus en détail dans la partie IV.1.3). Ici on cherche à minimiser notre RMSE (racine carrée de l'erreur carrée moyenne). A titre de comparaison, avec un GLM classique, on obtient un RMSE de 65.0132. On remarque également que certaines variables ont disparu de notre set. On a donc peu gagné en efficience, mais on respecte plus le principe de parcimonie.

	Hoerl, Kennard, Baldwin	Lawless, Wang	Validation croisée
RMSE	64.73571	64.74353	64.71252
Variables disparues du set originel	- Nb accidents en cours - Nb de tentatives de vol - Nb de retraits de permis - Moyen de transport pour aller au travail - Voiture de type break ? - Remorque - Propriétaire ou non - Travail	- Nb accidents en cours - Nb de tentatives de vol - Nb de retraits de permis - Moyen de transport pour aller au travail - Voiture de type break ? - Remorque - Propriétaire ou non - Travail	- Nb accidents en cours - Nb de tentatives de vol - Nb de retraits de permis - Moyen de transport pour aller au travail - Voiture de type break ? - Remorque - Propriétaire ou non - Travail

III.3.2 LASSO

La méthode LASSO (Least Absolute Shrinkage Selection Operator) ou régression l_1 a elle été proposée par Robert Tibshirani en 1996. Son but était de conserver les variables les plus importantes d'un modèle, déterminées à la fois par la sélection en sous-set et par régularisation. La méthode est très similaire à celle de la régression ridge, la seule différence résidant dans la contrainte, qui devient $\sum_{i=1}^{p} |\beta_i| \leq c$ en lieu et place de $\sum_{i=1}^{p} \beta_i^2 \leq c$.

La fonction de pénalité est alors :

$$\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \beta_j z_{i,j})^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

On retrouve les mêmes notations que précédemment avec Z la version centrée et normalisée de $\ddot{X}$. Y doit de nouveau être au moins centrée. Le paramètre de régularisation est ici $\lambda \geq 0$.

L'avantage de la méthode LASSO est qu'elle permet la mise à 0 de certains coefficients. Ainsi, l'algorithme LASSO choisira en priorité les variables les plus influentes. Un autre de ses avantages est que cette méthode est peu coûteuse en temps de calcul ou en espace de stockage.

Toutefois la méthode LASSO est peu performante dans les modèles à grande dimension ($n \ll p$) car elle sélectionne au plus n variables prédictives. De plus, cette méthode réagit assez mal face à la situation de multicolinéarité ; en effet, elle a tendance à privilégier de manière arbitraire une des variables corrélées et à mettre à 0 les autres ; la consistance du modèle n'est alors plus assurée. En effet, selon l'ordre dans lequel les variables seront évaluées, le résultat final ne sera pas forcément identique, et donc stable.

Contrairement à la méthode Ridge, LASSO n'admet pas de réponse analytique. En effet la norme l_1 n'est pas différentiable en 0. Des algorithmes ont donc été développés pour obtenir des estimations du vecteur $\hat{\beta}$. Citons par exemple la méthode LARS (Least-Angle Regression), les méthodes de sous-gradients ou encore l'algorithme FISTA (Fast Iterative Shrinkage-Thresholding Algorithm). La méthode des opérateurs proximaux, qui sera également évoquée en III.4.1 tend également à gagner en popularité.

L'algorithme LARS, le plus fréquemment utilisé, est un algorithme itératif. Au début chaque $\widehat{\beta_j}$ est initialisé à 0. On calcule ensuite les résidus $(y - \hat{y})$ et on regarde quel X_j est le plus corrélé avec les résidus. On fait bouger son coefficient jusqu'à ce qu'un nouveau X_j devienne plus corrélé avec les résidus que notre ancien X_j. On réitère ensuite ce procédé jusqu'à ce qu'on ait parcouru p prédicteurs. Cet algorithme adapté au LASSO conduit à ce que si on se trouve dans la situation où un $\widehat{\beta_j}$ atteint 0 alors qu'il ne partait pas de 0, alors la variable j disparaît de la liste des prédicteurs.

D'autres algorithmes existent, notamment la méthode de gradient qui est, elle, particulièrement intéressante lorsque l'on s'intéresse à des problèmes de fortes dimensionnalités ($n \ll p$).

III.3.2.1. Application de la méthode LASSO

Comme précédemment avec la pénalisation ridge, la régularisation LASSO a été appliquée sur nos données. Le processus est assez similaire. Il faut d'abord commencer par déterminer le coefficient de régularisation, λ. Contrairement à la méthode ridge, le paramètre de régularisation pour LASSO n'a pas de formule mathématique conférée. La méthode la plus répandue est à nouveau une méthode empirique : un large panel de valeurs de λ est testé, et on finit par sélectionner λ tel que le critère AIC

est minimisé. Il faut toutefois faire attention à ce que λ ne soit pas trop petit ; en effet, si la régularisation devient trop forte, des variables importantes peuvent être écartées et les coefficients peuvent être rétrécis excessivement, ce qui peut nuire à la fois au pouvoir de prédiction mais également aux inférences déduites.

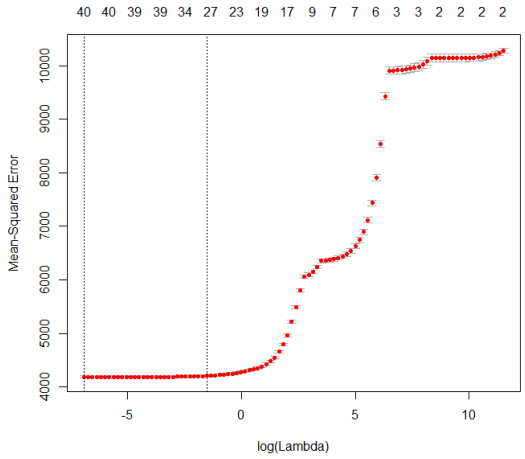


Figure 30: Evolution de la MSE selon la valeur de ln(lambda)

L'erreur moyenne au carré est ici minimisée pour $\lambda = 0.001$. Toutefois, suivant ce qui a été mentionné plus haut, il est préférable que λ ne soit pas trop petit. C'est pourquoi, tout en respectant les valeurs de la courbe ci-dessus, nous testerons également la pénalisation pour $\lambda = 0.1$ et $\lambda = 1$. De plus, sur le graphique ci-dessus, il apparaît que le nombre optimal de variables sélectionnées soit proche de 29.

Une fois λ determiné, la méthode LARS a été appliqué pour déterminer le vecteur β. On relance alors un GLM, en supprimant alors les variables dont le coefficient est 0 ou proche de zéro et on obtient alors les résultats suivants.

	$\lambda = 0.001$	$\lambda = 0.1$	$\lambda = 1$
RMSE	64.66106	64.66528	64.99903
Variables supprimées du set originel	- Type de permis - Nb d'accidents non-responsables - Nb d'accidents en cours - Moyen de transport pour se rendre au travail - Type break ? - Propriétaire - Rejet de CO^2 - Travail	- Type de permis - Nb d'accidents non-responsables - Nb d'accidents en cours - Moyen de transport pour se rendre au travail - Type break ? - Propriétaire - Rejet de CO^2 - Travail - Première souscription - Remorque ou non	- Type de permis - Nb d'accidents non-responsables - Nb d'accidents en cours - Moyen de transport pour se rendre au travail - Type break ? - Propriétaire - Rejet de CO^2 - Travail - Première souscription - Remorque ou non - Voiture type sport ? - Nb retraits de permis - Nb de tentatives de vol - Statut marital

On voit donc que plus λ est grand, plus la pénalisation sur le nombre de variables se fait ressentir. Pourtant, le RMSE du modèle n'en pâtit pas fortement pour autant. Le principe de parcimonie semble donc effectif.

III.3.3 Variantes

Ces différentes méthodes ont été développées il y a maintenant plusieurs décennies, des variantes ont donc été développées, cherchant à corriger et améliorer ces méthodes classiques. On parlera notamment d'Elastic-Net ou de LASSO adaptatif. On a également assisté à des modifications de l'algorithme originel pour s'intéresser à des problématiques légèrement différentes (Fused-LASSO et Group-LASSO) ou permettre l'introduction d'informations externes (Prior LASSO).

La variante la plus largement utilisée est la méthode Elastic-Net. Elle vise à combiner les avantages des méthodes ridge et LASSO. Pour cela la contrainte d'optimisation est modifiée et devient:

$$\alpha \sum_{i=1}^{p} |\beta_i| + (1-\alpha) \sum_{i=1}^{p} \beta_i^2 \leq c$$

Ce qui amène également une altération du programme d'optimisation qui devient :

$$\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \beta_j z_{i,j})^2 + \lambda \left[\alpha \sum_{i=1}^{p} |\beta_i| + (1-\alpha) \sum_{i=1}^{p} \beta_i^2 \right]$$

Elastic-Net additionne alors l'avantage de la méthode LASSO de sélection des variables (et donc d'exclusion des variables non pertinentes), et de ridge en partageant les poids des variables corrélées (et donc n'en conserve plus une seule de manière arbitraire). Elastic Net permet de plus, la sélection de plus de n variables, même si $n \ll p$.

On notera que si $\alpha = 0$ on retombe sur une sélection Ridge, et si $\alpha = 1$ on fait face à une sélection LASSO.

D'autres méthodes dérivées des deux principales méthodes introduites dans ce mémoire existent. Par exemple, Fused-Lasso permet de prendre en compte la spatialité des variables, en attribuant des coefficients similaires aux variables 'proches' spatialement ; pour cela la distance entre les coefficients sous la norme l_1 est également pénalisée. La méthode Generalized Fused-Lasso conserve la même méthodologie, mais cette fois-ci l'applique aux variables déjà disponibles sous forme de graphes.

Group-Lasso, développée en 2006, est une méthode qui étend la sélection de variables à la sélection de groupes. Ces groupes, fournis à priori, contiennent un ensemble de variables, qui auront alors toutes le même coefficient. La méthode Adaptive-LASSO a été développée pour assurer une consistance en toute circonstance de l'estimation donnée par LASSO. Adaptive-LASSO peut se résoudre par les mêmes algorithmes que ceux utilisés pour LASSO.

La dernière méthode, Prior-LASSO développée en 2016 par Jiang et Al., permet l'introduction d'informations à priori. Ainsi, un poids différent est accordé aux données et à ces informations, issues d'un avis extérieur ou d'observations constatées.

Toutefois ces méthodes ne s'appliquent pas directement à notre problématique, c'est pourquoi on laissera le lecteur intéressé par plus de renseignements consulter l'article 'Regularization and Variable Selection via the Elastic Net' par Zou, Hui, Hastie et Trevor. Seule la méthode Elastic-Net nous intéresse et c'est pourquoi ce sera la seule à être implémenté dans la suite.

III.3.3.1. Application de la méthode Elastic-Net

On va donc tester différentes valeurs de λ et α et sélectionner le meilleur résultat après validation croisée. On obtient donc $\alpha = 0.106$ et $\lambda = 0.323$. On teste également deux autres combinaisons avec $\alpha = 0.461$ et $\lambda = 0.00108$, et $\alpha = 0.437$ et $\lambda = 0.0256$ puisque comme nous testons à chaque fois des couples de combinaisons aléatoire (α, λ), il est difficile d'obtenir une solution stable. Les variables supprimées et les RMSE obtenus depuis les différents modèles sont regroupés dans le tableau ci-dessous.

	$\alpha = 0.106$ et $\lambda = 0.323$	$\alpha = 0.461$ et $\lambda = 0.00108$	$\alpha = 0.437$ et $\lambda = 0.0256$
RMSE	64.6918	64.66106	64.66255
Variables supprimées du set originel	- Type de permis - Nb d'accidents non-responsables - Nb d'accidents en cours - Moyen de transport pour se rendre au travail - Type break ? - Propriétaire - Rejet de CO^2 - Nb retraits de permis - Remorque ?	- Type de permis - Nb d'accidents non-responsables - Nb d'accidents en cours - Moyen de transport pour se rendre au travail - Type break ? - Propriétaire - Rejet de CO^2	- Type de permis - Nb d'accidents non-responsables - Nb d'accidents en cours - Moyen de transport pour se rendre au travail - Type break ? - Propriétaire - Rejet de CO^2 - Nombre d'années sans accidents

On remarque que la régression de type LASSO (régression de type l_1) a un impact bien plus fort sur le pouvoir de pénalisation de l'algorithme. En effet c'est lorsque λ est grand que l'on observe une plus grande restriction, alors que le paramètre α semble peu jouer. C'est donc principalement la valeur de λ qui drive le nombre de variables à ôter du modèle.

III.3.3.2. AVENGER méthode / Ridge Effect

Une dernière méthode alternative aux régressions ridge et LASSO sera présentée ici succinctement. Cette méthode, dénommée méthode AVENGER pour Accurate Variable Extraction using the support Norm, Grouping and Extreme Randomization, se base sur la norme k-support, qui a su dépasser les performances de LASSO et d'Elastic-Net dans certaines études (*Spectral k-Support Norm Regularization,* Mc Donald et al.).

La norme k-support, développée par Andreas Argyriou, Rina Foygel et Nathan Srebro au début des années 2010, revient à pénaliser les poids les plus lourds du modèle à l'aide de la norme l_2 tandis que le principe de parcimonie est imposé sur le reste du vecteur de poids à travers la norme l_1. Elle s'exprime mathématiquement pour une dimension au maximum de d, sous la forme :

$$||w||_k^p = \left(\sum_{i=1}^{k-r-1} (|w|_i^{\downarrow})^2 + \frac{1}{r+1} \left(\sum_{i=k-r}^{p} |w|_i^{\downarrow})^2 \right) \right)^{\frac{1}{2}}$$

Où r est le seul entier dans $\{0, \dots, k-1\}$ vérifiant $|w|_{k-r-1}^{\downarrow} > \frac{1}{r+1} \sum_{i=k-r}^{p} |w|_i^{\downarrow} \geq |w|_{k-r}^{\downarrow}$. $w_i^{\downarrow}$ désigne lui le $i^{ème}$ plus grand élément de w avec la convention $|u|_0^{\downarrow} = +\infty$.

Sur le principe, il s'agit donc d'un compromis entre les régressions LASSO et Ridge, similairement à Elastic-Net. Néanmoins, là où Elastic-Net peut être vu comme un cas particulier de la sélection par groupe avec des groupes prédéfinis (singletons et $\{1, \ldots, p\}$), la norme k-support considère des groupes implicites dont seules les tailles sont connues.

La méthode AVENGER, qui a été introduite dans la thèse de Anne-Claire Haury en 2012, permet de contrer le manque de stabilité induite parfois par certaines méthodes dû au fait que les variables renvoyées dépendent de l'échantillon de départ. On cherche ici à obtenir les q variables les plus importantes. Pour cela, AVENGER utilise une randomisation extrême, bootstrappant à la fois les données mais également la sélection de variables disponibles pour le choix de l'algorithme. C'est un processus que l'on retrouvera notamment dans l'algorithme de Random Forest. Ensuite des poids sont tirés au hasard dans une distribution de Bernoulli. La fréquence de sélection de la variable j parmi les q premières variables est ainsi calculée conditionnellement au poids déterminé précédemment. Les listes produites par chaque itération sont ensuite agrégées.

Dans la mise en place de cette méthode, la méthode ADMM (voir Annexe A.3) a été utilisée. Pour 500 rééchantillonnage et une limitation à $q = 20$ paramètres, nous trouvons finalement comme variables supprimées :

	Variables supprimées du set originel	RMSE
Limitation à $q = 20$ paramètres	- Sexe - Première souscription ? - Voiture de sport ? - Voiture cabriolet ? - Voiture 4x4 ? - Propriétaire ou non - Nb d'accidents non-responsables - Présence d'une remorque ? - Catégorie du permis - Nb d'accidents en cours - Nb de tentatives de vol - Emission de Co^2 - Travail - Présence d'une alarme - Moyen de transport pour aller au travail - Statut marital - Age de la voiture - Situation du parking - Utilisation - Nb de sièges	65.632450

III.4. Autres méthodes

Dans ce paragraphe nous présenterons d'autres méthodes de sélection de variables, mais cette fois-ci dans un contexte plus large. Même si la méthode SMuRF (qui n'a rien à voir avec les petits êtres bleus) est liée à la méthode LASSO, elle sera développée dans cette partie car elle nécessite la compréhension d'un nouvel outil mathématique, les opérateurs proximaux.

Nous verrons également dans cette partie comment certains algorithmes de machine learning peuvent aider les méthodes statistiques dans la sélection de variables, à la fois lorsque le modèle est compilé, mais également en amont de cette compilation.

Nous nous attarderons enfin sur le concept de permutations, que nous serons amenés à revoir par la suite (notamment avec le concept de valeurs de Shapley). Sera présenté également, un des

outils les plus fréquemment utilisés dans la réduction de dimensions, l'Analyse en Composantes Principales (ACP).

III.4.1 SMuRF méthode et opérateurs proximaux

Nous allons ici introduire la méthode SMuRF (Sparse Regression with Multy-type Regularized Feature Modeling), qui est à distinguer de la méthode SMuRFS (Sequential Multi-Response Feature Selection). Cette dernière méthode, qui permet également de parvenir à trouver le meilleur sous-set de variables en utilisant une succession de forêts aléatoires, ne sera pas plus détaillée dans cet ouvrage.

La méthode SMuRF est une régression LASSO multi-type qui somme les sous-pénalités de plusieurs types de prédicteurs (catégoriel, numérique). Elle permet de réaliser en même temps que l'estimation des paramètres, une sélection des variables et des fusions au sein même d'un groupe de modalités. En effet, lorsque l'on effectue une estimation des différents coefficients à affecter aux variables, les termes de régularisation (ou de pénalisation) imposent une contrainte de 'budget' sur l'espace des paramètres, dans le but de réduire la dimensionnalité pour éviter des problèmes tels que l'overfitting ou la multicollinéarité.

Pour résoudre les sous-problèmes posés par type de variables, SMuRF utilise la théorie des opérateurs proximaux. Grâce à cela, résoudre le problème par sous-problème revient à résoudre le problème dans son ensemble. Une description des opérateurs proximaux se trouve dans l'Annexe A1.

La méthode SMuRF s'adapte donc particulièrement bien sur des bases de données conséquentes. Elle permet notamment de réconcilier les méthodes de machine learning avec les méthodes statistiques plus traditionnelles.

La méthode SMuRF se veut réconcilier deux écoles. Puisque l'utilisation de méthodes de régularisation s'est de plus en plus développée pour répondre aux contraintes susmentionnées, de nombreuses méthodes ont vu le jour. Du côté de la communauté de machine learning, différentes techniques sont utilisées, telles que l'algorithme pour la régression des moindres angles (LARS, Efron), le flow paramétrique (Xin et al.) ou des algorithmes de sous-gradient. Ces méthodes sont performantes sur les data sets ne contenant qu'un unique type de prédicteur (catégoriel / numérique), mais sont particulièrement difficiles à adapter dans les autres cas.

Du côté des statisticiens, Oelker et Tutz ont proposé en 2017 une approximation quadratique locale des pénalités, de manière à pouvoir adapter l'algorithme PIRLS (Penalized Iteratively Reweighted Least Squares) aux GLM régularisés. Cette approche permet de résoudre la plupart des problèmes multi-types, mais l'approximation quadratique conduit à des erreurs de sélection de prédicteurs / de fusion des modalités. De plus, l'algorithme PIRLS nécessite un grand temps de calcul puisqu'il travaille sur l'inverse de grandes matrices, ce qui peut poser problèmes dans le cas de matrices larges / longues.

C'est pourquoi l'utilisation des opérateurs proximaux s'est démocratisée ; cette utilisation est légitimée par le fait que selon les résultats sur l'optimisation convexe de Boyd et Vandenberghe, minimiser la formule de régularisation entière revient à minimiser itérativement ses différentes parties.

Une fois le problème divisé en sous-problèmes avec des opérateurs proximaux, il suffit de résoudre les problèmes de minimisation avec les différents modèles de machine learning / statistiques disponibles.

Tout comme les algorithmes classiques du LASSO, la méthode SMuRF nécessite la donnée d'un paramètre de régularisation λ. Pour trouver celui-ci, pas de solution miracle. Il est difficile de le déterminer à priori. Le meilleur moyen de procéder est de répéter l'algorithme pour différentes valeurs de λ ; le meilleur λ est ensuite sélectionné comme étant celui minimisant le critère choisi, dans notre cas le critère AIC.

Une fois λ défini, et les résultats obtenus via la méthode SMuRF, on applique le modèle sur les variables dont le coefficient obtenu est différent de 0. Cette ré-estimation sans l'algorithme de pénalité, mais avec une matrice réduite $\ddot{X}''$, permet de réduire le biais dans les coefficients des variables obtenus.

Ci-dessous, les résultats obtenus avec λ estimé égal à 0.034 par validation croisée :

	$\lambda = 0.034$
RMSE	65.37293
Variables supprimées du set originel	- Type de permis - Nb d'accidents non-responsables - Nb d'accidents en cours - Moyen de transport pour se rendre au travail - Type break ? - Propriétaire - Rejet de CO^2 - Sex

Cette méthode, compliquée à mettre en place, du fait de ses nombreuses contraintes techniques, notamment dues au fait du partage de la base de données en sous-bases de données, qui ne présentent pas alors forcément toutes les modalités d'une variable, n'apporte pas de découverte majeure relativement aux méthodes proposées en amont.

Il est donc préférable, tout du moins dans notre cas, d'utiliser les méthodes traditionnelles.

III.4.2 Friedman H-Statistique

La statistique H a été proposée par Friedman et Popescu en 2008 et permet de mesurer de combien la variation de la prédiction dépend de l'interaction des variables. En effet, comme il l'a été mentionné plus haut, il ne suffit pas d'additionner les effets individuels de chaque variable pour obtenir la prédiction ; d'autres facteurs rentrent en jeu, comme notamment la dépendance entre variables. Ainsi, le but ici n'est pas de supprimer des variables, mais d'en inclure : particulièrement des interactions de degré 2.

La statistique H repose sur des fondements mathématiques solides, puisqu'elle se base sur la décomposition de la fonction de dépendance partielle, c'est-à-dire sur les marginales de chaque variable ; elle définit alors l'interaction comme la part de variance expliquée par celle-ci.

Pour mesurer l'intensité de l'interaction entre deux variables, deux méthodes existent. La première, mesure si la variable X_j interagit avec d'autres variables ; on peut alors trouver les variables montrant les plus fortes interactions dans le modèle.

Pour faire cela, on regarde la valeur de la prédiction avec le modèle original, appelons-la $f(X^{(i)})$, $X^{(i)} = (x_{i,1}, x_{i,2}, \ldots, x_{i,p})$.

On calcule ensuite $PD(\mathrm{X_j})$ la dépendance partielle de la variable X_j, puis $PD(\mathrm{X_{-j}})$ la dépendance partielle de toutes les variables exceptées X_j.

On obtient alors pour l'observation i : $\rho_j^{(i)} = \frac{(f(X^{(i)}) - PD(x_i,j) - PD(x_{i,-j}))}{Var(f(\mathrm{X^{(i)}}))}$

Ensuite, on obtient $H_j^2 = \sum_{i=1}^{n} \rho_j^{(i)}$

La seconde méthode consiste à observer avec quelles variables une variable $\mathrm{X_j}$ est la plus corrélée. La formule est similaire à la précédente, sauf que $f(x)$ est remplacé par $pd(\mathrm{X_k}, \mathrm{X_j})$, l'interaction partielle entre notre variable $\mathrm{X_k}$ et une autre variable X_j appartenant au modèle.

Identifier ces interactions peut permettre de comprendre quelles variables créent une codépendance dans le comportement de nos modèles. Cela permet également de pré-sélectionner les interactions à étudier via les graphes PDPs qui seront développés dans la partie V de ce mémoire.

La statistique H est assez gourmande en temps de calcul. Une alternative pourrait se trouver dans la limitation de profils étudiés durant son calcul ; toutefois il est difficile de se séparer de certaines observations puisque les calculs impliquent d'estimer des distributions marginales, qui peuvent avoir une certaine variance qui ne peut être décelée que lorsque toutes les observations sont utilisées.

Ici, nous avons utilisé comme fonction f un GBM quelconque (c'est-à-dire non tuné de manière à donner le meilleur modèle possible). Ce GBM nous a donné les résultats d'interaction suivants, réalisé de prime abord sur toutes les variables, puis sur toutes les combinaisons de variables avec la variable 'valeur assurée' :

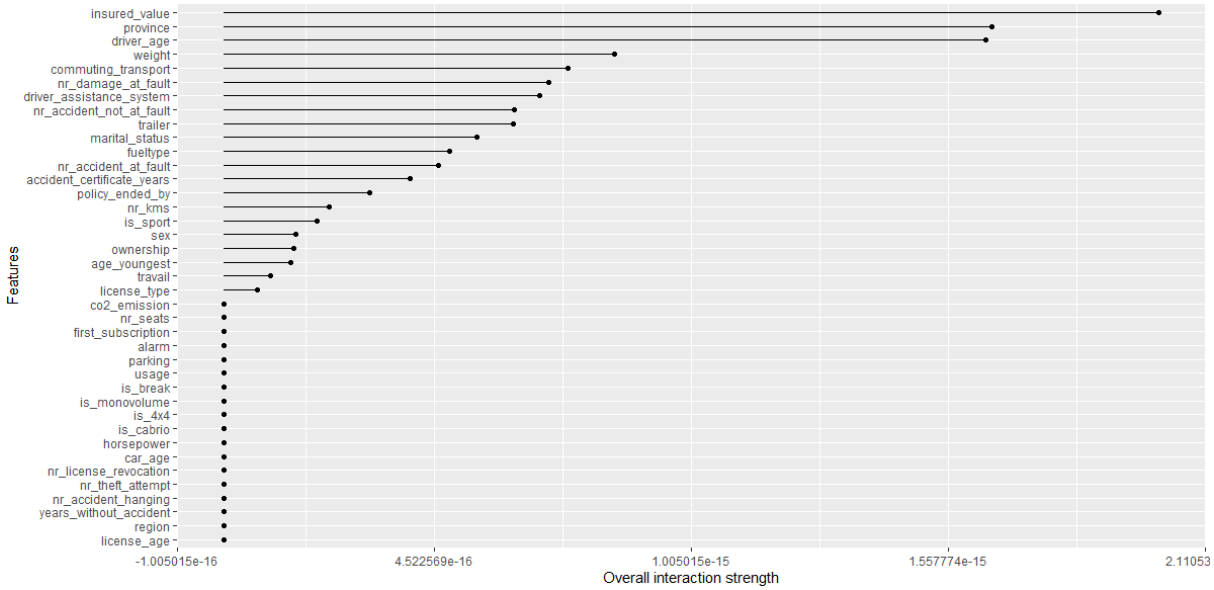


Figure 31 : Statistique H de Friedman pour toutes les variables

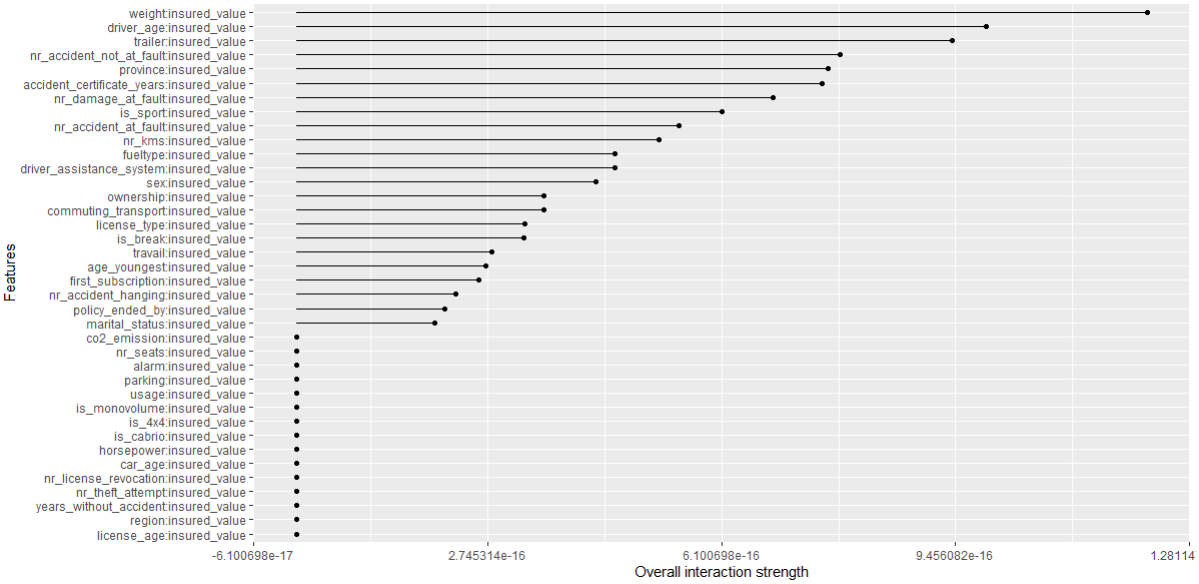


Figure 32: Statistique H de Friedman pour toutes les variables, corrélées avec la valeur assurée

Fort de ces résultats, nous avons alors rajouté les interactions les plus fortes, c'est-à-dire :

- Poids – Valeur assurée
- Age du conducteur – Valeur assurée
- Age du conducteur – Sexe

Pour prouver l'efficacité de l'ajout de telles combinaisons de variables, nous avons ensuite développé des modèles GLM avec et sans interactions, dont la performance finale est regroupée dans le tableau ci-dessous.

	Variables interagissants	RMSE avant interaction	RMSE après interaction
Variables originelles	- Poids – Valeur assurée - Age du conducteur – Valeur assurée - Age du conducteur – Sexe	61.19235	60.70688
Variables conservées à l'issue de la phase II	- Poids – Valeur assurée - Age du conducteur – Valeur assurée - Age du conducteur – Sexe - Type de carburant - Age du conducteur	63.25122	61.45743

Même si l'usage de la statistique H semble satisfaisant, une de ses faiblesses est néanmoins notable : la statistique H repose sur l'hypothèse (comme les PDPs (Partial Dependence Plots) ou les permutations) que les valeurs des variables peuvent être modifiées indépendamment. Hors dans notre cas, certaines combinaisons de variables sont impossibles.

III.4.3 Permutation

Ici le concept est assez simple : on mesure l'importance d'une variable en calculant l'augmentation dans l'erreur de prédiction du modèle une fois la variable permutée. Une variable est ainsi importante si mélanger aléatoirement ses valeurs amène une augmentation de l'erreur du modèle ; cela veut dire que le modèle se base sur cette variable pour donner ses prédictions. Si au contraire le modèle reste inchangé, alors cela signifie que le modèle a ignoré cette variable pour établir sa prédiction.

Cette méthode a été proposée par Breiman en 2001 puis revue de manière à devenir modèle-agnostique, par Fisher, Rudin et Dominici.

Dans la pratique, on calcule une première fois un modèle sur l'échantillon d'entraînement avec les vraies valeurs de la variable d'intérêt, puis on l'applique deux fois à l'échantillon de test (une fois avec les vraies valeurs, une seconde fois avec des valeurs permutées de la variable d'intérêt). Ainsi, via le ratio de l'erreur établi sur l'échantillon test on obtient la contribution de chaque variable à la performance du modèle sur des données non vues.

Pour obtenir des résultats fiables il faut souvent répéter l'opération avec différentes permutations, de manière à stabiliser la mesure. Néanmoins, comme pour les PDPs, que l'on verra plus en détail dans la section V.3.3., on peut se retrouver avec des données aberrantes si la variable d'intérêt est corrélée à d'autres variables du dataset. Ainsi, réaliser une telle permutation sur la colonne de l'âge du conducteur conduira à une suite de profils aberrants ; le modèle peut alors se perdre entre les différentes variables qui lui sont corrélées.

Une version corrigée de cet algorithme a été développée par Altman, Tolosi, Sander et Lengauer, de manière à corriger le biais en utilisant des p-values.

La permutation des caractéristiques permet de prendre en compte toutes les interactions possibles dans le modèle. Cette méthode est également performante, dans le sens où elle ne nécessite pas de recréer un nombre important de fois le modèle, mais se repose juste sur la prédiction faite par un modèle initial.

Sur nos données, nous avons donc réalisé une permutation de chaque colonne, et avons testé le GLM entraîné sur toutes les variables, et le GLM avec uniquement les variables restantes après la partie II.

Le graphe d'importance par permutations suivant est ainsi obtenu selon les variables présentes originellement dans le set de données :

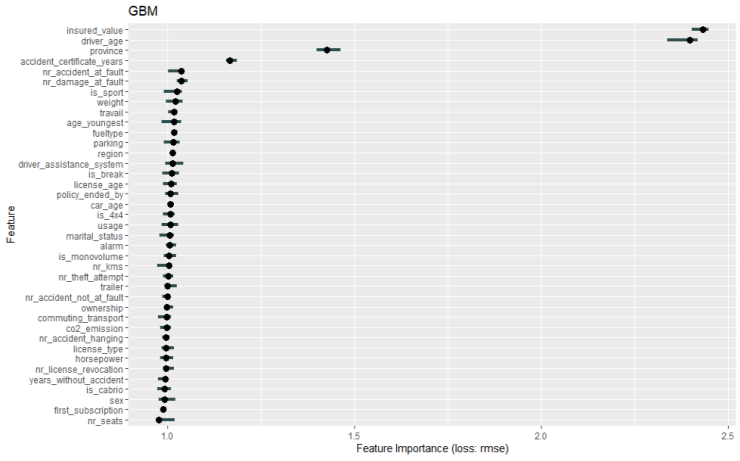


Figure 33 : Graphe d'importance par permutation par GBM

Et à nouveau, comme dans la partie précédente, nous avons regroupé les performances des modèles obtenus à l'issue de cette sélection de 'nouvelles' variables.

	Variables interagissants	RMSE avant interaction	RMSE après interaction
Variables originelles	- Age du conducteur – Valeur assurée - Age du conducteur – Province - Valeur assurée - Province	61.19235	56.95752
Variables conservées à l'issue de la phase II	- Age du conducteur – Valeur assurée - Age du conducteur – Province - Valeur assurée - Province - Certificat d'accidents – Age du conducteur	63.25122	57.7333

III.4.4 ACP

Dans cette partie nous allons introduire le concept d'ACP (Analyse en Composantes Principales). D'autres variantes de cette analyse existent également comme l'analyse en composantes binaires (ACOBI) ou l'analyse factorielle de correspondances multiples (AFCM), mais nous n'introduirons ici que la plus fréquemment utilisée, l'ACP.

L'ACP permet d'obtenir des représentations et des réductions de l'information contenue dans des bases de données volumineuses. Avec les notations introduites précédemment, l'espace naturel pour représenter les données est $\mathbb{R}^p$. Le nuage de points de toutes les observations peut être alors tracé dans un second espace, $\mathbb{R}^n$.

Dans l'espace $\mathbb{R}^p$, les points représentent ainsi les p caractéristiques de l'individu i. Dans l'espace $\mathbb{R}^n$, les points représentent les valeurs prises par la variable j sur l'ensemble des n individus.

Néanmoins il est difficile de se rendre compte de la forme des nuages de points en grande dimension. C'est pourquoi il est fréquent de les projeter sur des droites ou sur des plans. En pratique, on représente un cercle, le cercle des corrélations, qui à chaque point-variable associe un point dont la coordonnée sur un axe factoriel est une mesure de la corrélation entre cette variable et le facteur. Ces axes factoriels, qui sont formés par les vecteurs propres de la matrice de corrélation entre les variables explicatives, engendrent ainsi les sous-espaces de projection maximisant la distance.

Ces composantes principales sont décorrélées entre elles et ordonnées de manière à ce que les premières incluent la plus grande partie de l'information comprise dans l'ensemble des variables initiales. Les composantes principales contiennent les coordonnées des individus sur les axes factoriels qui correspondent aux directions de l'espace expliquant au mieux la variance de l'échantillon.

Nous avons représenté ici différents graphiques d'une ACP réalisée sur les variables numériques de notre base de données.

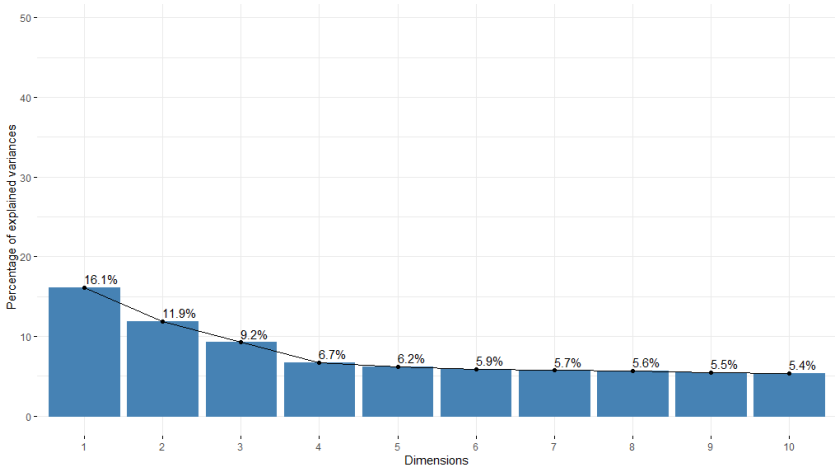


Figure 34 : Pourcentage de variance expliquée selon l'ajout progressif de dimension

Sur ce premier graphique, sont représentées les différentes valeurs propres de nos variables numériques –c'est-à-dire la quantité de variance expliquée par une variable-. On peut ainsi voir qu'avec seulement 8 variables, 70% de la variance totale est expliquée. Il n'y a pas de valeur 'officielle' pour déterminer le nombre optimal de composantes principales. Néanmoins, il ne faut pas perdre de vue, qu'ici seules les variables à valeurs numériques sont représentées.

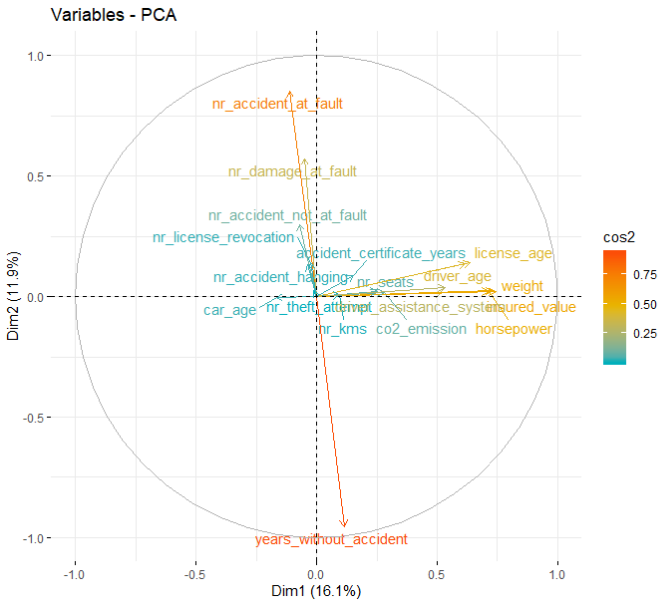


Figure 35 : ACP réalisée sur les variables numériques

Le graphique ci-dessus est un graphique de corrélation des variables. Les variables positivement corrélées sont regroupées ; celles qui le sont négativement sont dans des quadrants opposés. La distance entre les variables et l'origine mesure la qualité de la représentation de la variable. Plus elles en sont loin, plus elles sont bien représentées au sein du modèle. L'interprétation des axes est toutefois relativement compliquée et d'un point de vue opérationnel, il est difficile d'expliquer la présence des composantes principales en tant que variables explicatives dans le modèle si ces composantes ne sont pas totalement interprétables.

Finalement nous avons décidé de regrouper au sein d'un tableau de corrélation les variables les plus contributives pour chaque dimension :

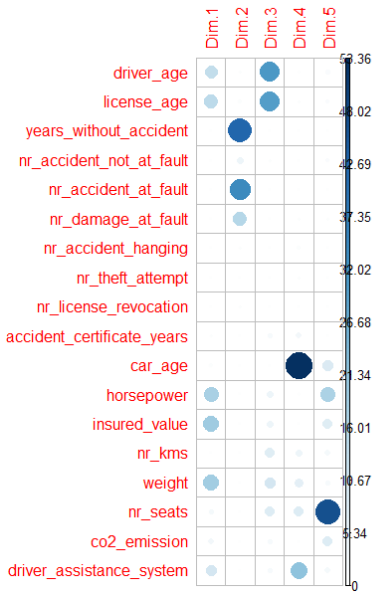


Figure 36 : Tableau représentant la contribution de chaque variable numérique dans chaque dimension (5 premières)

Nous pouvons voir que les résultats ne se recoupent pas forcément avec ceux obtenus à travers les différentes méthodes évoquées dans ce chapitre. Il est donc compliqué de décider quelles variables conserver et quelles variables évincer. En grande dimension, il est particulièrement difficile de se repérer. Ici, si l'on se base sur l'analyse des 5 premières dimensions, on pourrait décider de supprimer les variables numériques nombre d'accidents en cours, nombre de retraits de permis et nombre d'années sans accidents. Mais avec quel degré d'assurance pouvons-nous vraiment nous séparer de ces variables ?

Dans le cas présent, nous n'avons pu mener l'analyse ACP que sur une partie de nos variables (voir figures 34, 35 et 36 ci-dessus), c'est-à-dire celles étant de type numérique. D'autres méthodes, comme l'ACM (Analyse en Composantes Multiples) ou l'AFDM (Analyse Factorielle des Données Mixtes) permettent de traiter également les données de type qualitatif, mais elles ne seront pas développées plus avant dans ce mémoire.

III.5. Bilan sur la sélection de variables

On l'a vu tout au long de cette partie, bien sélectionner nos variables est primordial pour pouvoir entraîner par la suite un modèle performant, en partant sur des bases solides. Néanmoins les techniques pour la sélection de variables sont aussi diverses que variées. Et au final, il est compliqué d'accorder une confiance aveugle à un des algorithmes présenté ci-dessus, suite à la difficulté d'obtenir un consensus solide quant au choix de variable à effectuer.

Le principe de parcimonie est né en économie mais a rapidement trouvé sa place en statistiques. En effet, si les statisticiens et les actuaires développent des modèles en interne, ces modèles ont vocation à être présentés à des personnes qui ne sont pas forcément familières avec les algorithmes et les processus de modélisation associés. C'est pourquoi un modèle ne doit pas être trop complexe. Mais la complexité ne doit pas nuire à la performance du modèle. La frontière est pourtant facile à traverser.

C'est pourquoi dans cette partie on s'est également intéressé à l'intégration de nouvelles variables, combinaisons de variables figurant déjà dans nos données, et censées représenter des interactions au sein même des variables. Cela a été rendu notamment possible grâce à la statistique

H de Friedman. Ces interactions auraient pu être tout autant détectées par les autres algorithmes introduits, mais le nombre de combinaisons dont on doit tester l'inclusion reste toutefois prohibitif.

Ainsi, à l'issue de cette partie, nous sommes parvenus à un set de variables 'optimal' présenté ci-dessous :

Variables ajoutées	Variables supprimées
- Age du conducteur – Valeur assurée - Age du conducteur – Province - Valeur assurée - Province - Certificat d'accidents – Age du conducteur - Age du conducteur - Sexe	- Type de permis - Nb d'accidents non-responsables - Nb d'accidents en cours - Moyen de transport pour se rendre au travail - Type break ? - Propriétaire - Rejet de CO^2 - Travail - Première souscription - Remorque ou non

On peut voir que par rapport à notre première sélection à l'aide de filtres (fin du chapitre II), peu de variables diffèrent, si ce n'est bien sûr pour les nouvelles variables d'interactions inclues.

IV Reverse Engineering du modèle

"Fools say that they learn by experience. I prefer to profit by other experience" - Otto Von Biskmarck

Maintenant que nous conaissons les variables qui se révèleraient pertinentes dans notre étude, et que nous avons nettoyé notre base de données, intéressons-nous au coeur du sujet de ce mémoire. Comment obtenir, à partir des caractéristiques d'un portefeuille d'assuré et du tarif proposé, le modèle de tarification utilisé par un assureur ?

Ce chapitre s'articulera autour de la présentation de différentes méthodes, permettant de répondre à la problématique ci-dessus. Tout d'abord les méthodes classiques, qui devraient théoriquement répondre le mieux à cette problématique, puisque ce sont elles qui ont été utilisée de prime abord pour déterminer le tarif. Néanmois, il ne faut pas perdre de vue que l'on dispose pour chaque assuré de la prime commerciale et non de la prime pure, et c'est cette dernière qui a été estimée par la méthode statistique.

Si les statistiques permettent de déduire des inférences sur une population à partir d'un échantillon, le but du machine learning est de trouver des schémas prédictifs facilement généralisables. Certains algorithmes des deux partis peuvent cependant se rejoindre et parvenir à trouver un juste milieu entre ces objectifs. Ainsi, le principe de bootstrap largement utilisé en statistiques inférentielles, se trouve être la base de l'algorithme de forêt aléatoire. Les deux méthodes semblent donc être complémentaires.

Une courte introduction aux différents algorithmes de machine learning utilisés dans le cadre de cette étude sera également proposée. Nous verrons ensuite leur application dans notre cas ainsi que leurs limites, telles que l'interprétabilité et la justesse, qui seront discutées plus avant dans le chapitre V de ce mémoire.

Un benchmark récapitulant toutes les méthodes utilisées sera présenté en fin de chapitre.

IV.1. Etat des lieux

Une des premières questions à se poser est : De quoi dispose-t-on à la fin de chaque étape ?

En fin de chapitre II, nous disposons d'une base de données propre et scindée de telle sorte à ne représenter qu'un seul assureur. Après le chapitre III, nous possédons une sélection de variables réfléchie par rapport à la relation entre variable réponse et variables explicatives. Nous avons donc opéré une pré-sélection des variables les plus enclines à expliquer la prime de notre assureur.

Cette présélection sera particulièrement utile dans le cas des modèles statistiques, mais nous verrons dans la partie sur les algorithmes de machine learning que cette sélection ne fait pas vraiment sens et contribue peu à l'amélioration des modèles.

Avant de se lancer dans la création des différents modèles, il est important de savoir comment seront organisées les différentes modélisations, de manière à pouvoir proposer une comparabilité entre chacune. Pour cela il est important de définir les bases de données qui seront utilisées ainsi que le critère de sélection qui permettra d'attester de l'efficacité d'un modèle.

Lors de la phase de modélisation, une nouvelle question s'est posée. Etait-il plus judicieux de conserver ou non la variable de dépréciation comme variable explicative ? En effet, la conserver permet de maintenir un large nombre de données ; néanmoins, comme vu dans la partie II, son effet n'est pas fixe mais relatif à la prime. Ainsi, certains algorithmes comme les modèles linéaires, peuvent

rencontrer des difficultés pour traiter cette information, puisqu'ils se contentent d'appliquer un coefficient donné pour telle ou telle valeur de dépréciation. Après étude plus approfondie de la problématique, il s'est avéré que garder la variable dépréciation telle quelle conduisait à la surreprésentation de certains profils dans notre portefeuille (ceux pour qui plusieurs offres avaient été proposés), ce qui n'est pas représentatif de la réalité. De cette manière, si un profil est considéré comme un 'outlier' car son pricing n'a pas été réalisé selon le même schéma que celui appliqué aux autres, il sera surreprésenté, ce qui pourrait induire en erreur l'algorithme.

C'est pourquoi à partir de maintenant, les modèles seront réalisés seulement sur 26 872 profils, c'est-à-dire tous les profils auxquels ont été présentés une offre comprenant une dépréciation à valeur agréée. Au final, cela représente 90.3% des profils totaux de la base. En théorie, une telle scission peut conduire la compagnie d'assurance à développer trois modèles internes selon le type de dépréciation, ce qui est très coûteux en temps et en ressources.

Néanmoins, nous verrons à la fin de la partie IV qu'au final, appliquer seulement les coefficients de proportionnalité mentionnés dans la partie II sur le prix trouvé pour la valeur agréée, permet de fournir des prédictions proches des vraies valeurs.

IV.1.1 Partition des données

La partition des données est une étape cruciale de la modélisation. En effet, il n'est pas judicieux de se servir de la totalité de sa base de données pour entraîner son modèle. Faire cela revient à mesurer les performances du modèle seulement sur une plage spécifique de données ; on néglige alors sa reproductibilité. Hors dans notre cas, et dans la plupart des problématiques de modélisation, on veut pouvoir juger de la capacité de notre modèle à bien réagir sur différents échantillons.

Un algorithme est configuré pour que le modèle s'améliore, sur la base d'un critère choisi au préalable, à chaque itération. Hors, sans limite temporelle ni restrictions, le modèle s'améliore indéfiniment. Mais on fait alors face à un modèle qui ne peut pas se répliquer, car il overfitte le jeu de données d'entraînement, c'est-à-dire que notre modèle a appris, non seulement le pattern qu'il était censé déceler, mais aussi le bruit qui entourait les observations.

On doit donc faire face à ce qui est souvent considéré comme le compromis efficacité / biais dans la littérature. Ce phénomène a déjà été mentionné plus tôt, puisqu'il résulte aussi de la présence d'un trop grand nombre de variables / modalités.

Compte tenu de ce phénomène, nous avons décidé de partitionner notre modèle sur une base 70/30, c'est-à-dire que 70% des données sont réparties aléatoirement dans un échantillon dit d'entraînement, et les 30% restants forment l'échantillon de test.

La littérature actuelle ne s'accorde pas sur un pourcentage optimal, mais compte tenu du nombre important de nos observations, nous pouvons nous permettre ce partage, qui permet de maintenir un échantillon, que l'on espère représentatif, dans la base de données test.

Nous aurions également pu conserver une partie de la base de données pour constituer un échantillon de validation, utilisé durant la phase de création du modèle pour justement limiter le phénomène d'overfitting. Nous avons préféré utiliser la méthode de la validation croisée pour endiguer ce phénomène. Cette méthode sera détaillée plus en détail dans la partie IV.1.3. de cet ouvrage.

IV.1.2 Modèle de référence

En amont, nous avons souvent parlé de notre quête *d'amélioration* de notre modèle, mais que signifie vraiment *améliorer notre modèle* ? En quelques mots, cela revient à revoir et corriger notre modèle pour que celui-ci nous fournisse un résultat plus proche du réel.

Mais quel est notre point de départ ? Quel est le premier modèle à considérer ? Si nous en revenons au prémisse de ce travail, nous mentionnions dans la partie I.2. que pour un assureur, le meilleur moyen pour un assureur de tarifer le coût d'un contrat était de partager l'espérance des coûts des sinistres entre ses assurés.

Bien sûr cette méthode nécessite un portefeuille d'assurés homogène, pour éviter les problématiques de sélection adverse ou d'antisélection mentionnés plus tôt.

Mais partons de cette hypothèse de base ; supposons que notre portefeuille d'assuré est homogène. Il est en réalité peu probable que ce soit vraiment le cas, sauf cas rare où nous ne disposerions que de données clients n'appartenant qu'à un cluster bien particulier.

Alors, notre modèle est très simple : un nouveau client se verra appliqué la moyenne des tarifs proposés à ce jour par un assureur. Bien sûr ce modèle comporte de nombreux défauts : si notre portefeuille n'est pas assez diversifié alors un nouveau client n'appartenant à aucune des catégories présentes (par exemple un portefeuille de personnes d'une quarantaine d'années et un nouvel assuré de 18 ans, nouvellement conducteur) se verra appliqué un tarif qui n'a probablement rien à voir avec son risqué de sinistralité. A petite échelle, pour un assuré parmi des milliers d'autres, cela n'est pas grave ; toutefois, si cet assuré remarque la brèche, il encouragera son entourage à en profiter également, et l'assureur, s'il n'a pas encore détecté le problème, se retrouvera dans une situation compliquée lorsqu'il assistera à une augmentation non planifiée de sa sinistralité. De plus, cette technique d'application d'un tarif unique demande un ajustement régulier pour éviter qu'une augmentation intrinsèque du risque, du fait de l'ajout de nouveaux assurés, n'augmente le risque de défaut de la compagnie, dû à la perte de mutualisation du risque.

C'est pourquoi, les compagnies n'optent en pratique jamais pour ce modèle ; un modèle qualifié en théorie de la crédibilité de modèle avec une prime adoptée 100% collective. Néanmoins dans la pratique il s'agit du modèle le plus simple à mettre en oeuvre, c'est pourquoi ce modèle sera toujours utilisé comme référentiel d'amélioration.

Mathématiquement, le modèle, que l'on appellera modèle de la prime moyenne, s'exprime sous la forme :

$$Prime = \frac{1}{n}\sum_{i=1}^{n} P_i$$

Pour faciliter nos comparaisons avec ce modèle, un outil mathématique existe. Il s'agit du R^2, soit le coefficient de détermination. Il permet de mesurer la qualité de la prédiction d'une régression, en établissant le rapport de la variance expliquée par la régression par rapport à la variance totale.

Il est calculé selon la formule suivante :

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y_i})^2}$$

Pour notre modèle de la moyenne, on a donc par la formule que le R^2 du modèle est égal à 0.

IV.1.3 Critère de sélection du modèle et validation croisée

Maintenant que l'on sait *quoi* comparer, on va s'intéresser au *comment*. Il existe de multiples façons d'évaluer la performance d'un modèle et nous verrons ci-dessous les trois principales, ainsi que les raisons qui ont guidé notre choix final. On remarquera également que nous utilisons uniquement un critère de performance, et non pas un critère combiné performance-interprétabilité. Ce choix sera développé dans la partie V de ce mémoire.

Le premier critère mentionné est basé sur la théorie de l'information : il s'agit du critère d'Akaike (Akaike Information Criterion, AIC). Ce critère permet de pénaliser les modèles en fonction du nombre de paramètres k afin de satisfaire le critère de parcimonie. Il repose sur l'évaluation de la fonction de log-vraisemblance du modèle :

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

Il offre ainsi un compromis entre la qualité de l'ajustement et la complexité du modèle. Une version corrigée existe (AICc) lorsque le nombre d'observations n est relativement petit. Cela ne s'appliquerait pas notre cas. Ce critère a notamment été introduit dans la partie III.1 de ce mémoire, et utilisé tout au long de la partie III pour évaluer la performance des divers sets de variables déterminés.

Un deuxième critère de discrimination potentiel est l'erreur absolue moyenne (Mean Absolute Error, MAE). La MAE mesure l'amplitude moyenne de l'erreur dans le set de prédictions, sans considérer leur direction (négative ou positive). Elle est calculée comme suit :

$$MAE = \frac{1}{n}\sum_{j=1}^{n} |y_j - \widehat{y_j}|$$

Une dernière alternative à ces deux mesures précédentes est de prendre la racine de l'erreur moyenne au carré (Root Mean Squared Error). Comme la MAE, elle permet de mesurer l'amplitude moyenne de l'erreur. Elle s'exprime sous la forme :

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n} (y_j - \widehat{y_j})^2}$$

Comme la MAE, la RMSE exprime dans une unité similaire à la variable réponse, l'erreur moyenne du modèle prédictif. Leur plage de valeurs s'étend de 0 à $+\infty$.

Alors comment choisir entre ces trois mesures ?

Si on avait seulement dû sélectionner le meilleur modèle sur base de modèles statistiques, alors on aurait privilégié le critère AIC. En effet, grâce à son effet de parcimonie, ce critère permet de limiter un over-fitting en évaluant quel est le meilleur compromis entre taille du modèle et qualité de prédiction. Toutefois, dans notre étude nous serons confrontés à des modèles de machine learning, où le calcul du critère AIC est impossible, puisqu'il n'existe pas de fonction de vraisemblance explicite. C'est pourquoi nous rejetons dans notre étude ce critère, malgré le fait qu'il soit éminemment pertinent pour les modèles GLM et leurs dérivés.

Il nous reste donc à choisir entre la MAE et la RMSE. Ces deux critères sont très similaires, mais ont pourtant la particularité de ne pas traiter les erreurs larges de la même manière. Prendre la racine carrée de l'erreur carrée moyennée permet de donner un poids relativement important aux

grandes erreurs puisque l'erreur est mise au carré avant d'être moyennée. La RMSE doit donc être privilégiée lorsque les grandes erreurs sont particulièrement indésirables.

On doit également faire la remarque que la RMSE n'augmente pas en même temps que la variance intrinsèque des erreurs ; elle augmente avec la variance de la fréquence de distribution de la taille des erreurs.

Dans notre cas, puisque l'on peut considérer que faire une erreur de 150€ sur le tarif est bien plus que deux fois plus grave que de faire une erreur de 75€, on considérera la RMSE comme mesure discriminante dans le choix du modèle. En effet, si le reverse engineering est utilisé dans une optique de classements de tarifs entre assureurs, l'ordre sera plus profondément chamboulé si les tarifs varient dans une marge d'erreur plus grande.

La RMSE sera donc utilisée dans la suite de cette étude et on aura donc l'équivalence suivante:

$$Modèle\ 1 > Modèle\ 2 \ \Leftrightarrow RMSE_1 < RMSE_2$$

Toutefois, l'erreur utilisée dans la comparaison des modèles ne sera pas la résultante de la RMSE calculée sur le modèle final. En lieu et place, on utilisera une méthode de validation croisée, appelée "k-fold cross-validation" ; utiliser cette méthode permet de réduire les problèmes d'échantillonnage. En effet, lorsque l'on crée un modèle, la pratique veut que l'on sépare les données en deux ou trois parties (selon que l'on veuille ou non un échantillon de validation). Or, même si l'on se restreint à faire un échantillonnage aléatoire (choix de 80% de lignes par exemple), il se peut que le hasard fournisse tout de même un échantillon biaisé. Bien sûr, plus le nombre de lignes est important, plus il y a de chance que l'échantillon soit cohérent.

L'utilisation d'un set de validation permet, en parallèle de la création du modèle, de comparer l'erreur faite sur l'échantillon de validation. Cela signifie que même si l'erreur sur l'échantillon d'entraînement continue de diminuer au fil des itérations, l'algorithme s'arrêtera lorsque l'erreur sur le set de validation commencera à augmenter, ce qui correspond au moment où l'algorithme commence à sur-apprendre.

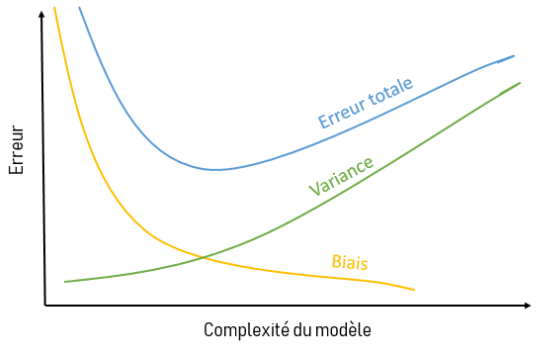


Figure 37 : Compromis Biais / Variance

Le processus de validation croisée en k-fold permet de répéter k fois le processus de partage du set de données. Ainsi, le set d'entraînement fourni en entrée est divisé en k sous-échantillons. Une fois ce partage en k sous-échantillons réalisés, le modèle est ensuite entraîné k+1 fois, k fois sur k-1 sous-sets, le $k^{ème}$ sous-set servant de dataset de contrôle (aka de validation). Chaque sous-dataset est donc utilisé au moins une fois comme set de validation. On obtient pour chacune de ces itérations une erreur, $RMSE_k$.

Le modèle final est ensuite entraîné sur le set d'entraînement complet, mais son erreur est l'erreur moyenne des k modèles :

$$RMSE_{finale} = \sum_{j=1}^{k} RMSE_j$$

On a donc, compte tenu de tout ce qui précède, ces métriques à battre :

	R^2 du modèle	RMSE Éch.Entr	RMSE Éch.Test	Nombre de variables
Modèle naïf	0	111.384	110.6228	0

Un point important, qui devrait également intervenir dans la sélection du modèle, est la robustesse de ce dernier. En effet, même si la théorie mathématique de la robustesse est souvent délaissée par les actuaires, du fait qu'elle est souvent difficile à mettre en place, son formalisme sous-jacent est néanmoins digne d'intérêt.

Dans la pratique, on qualifie souvent un modèle de peu robuste lorsque ce dernier est fortement impacté par un changement mineur dans les données. Dans la théorie, la robustesse d'un modèle implique d'autres points. En effet un modèle est robuste lorsqu'il fait abstraction des erreurs dans le résultat dues à une réalité déviant des hypothèses déclarées. En d'autres termes, cela signifie que même si les hypothèses ne sont qu'approximativement remplies, le modèle robuste aura toujours une efficacité raisonnable et un biais relativement faible.

Hors dans notre cas, et en particuliers avec les modèles statistiques, les hypothèses faites sont assez lourdes et rarement vérifiées dans la réalité. Ainsi, s'il y a une mauvaise spécification du modèle, alors il est probable que les méthodes classiques auront une piètre performance. Plusieurs méthodes ont été développées pour quantifier la robustesse d'une méthode. Les plus communes sont le point de rupture (breaking point), la fonction de sensibilité ou encore la fonction d'influence. Ces méthodes seront développées dans la partie V. 2.2 de ce mémoire, puisqu'on les utilisera pour tester le bien fondé de nos modèles.

IV.2. Méthodes statistiques

Maintenant que les fondements de notre étude ont été posés, nous pouvons nous atteler à la lourde tâche de retrouver le modèle de notre assureur. Pour cela nous commencerons par, dans un premier temps, retrouver le modèle de pricing via les méthodes statistiques. En effet ces dernières sont encore largement utilisées par les compagnies d'assurance dans leurs modèles de pricing malgré le développement de nouvelles méthodes de tarification bien plus efficaces. Néanmoins, leur usage demeure car leur base statistique et leur interprétabilité leurs permettent de se distinguer des nouvelles méthodes, souvent qualifiées de plus 'obscures'.

Toutefois, nous verrons dans la partie V que cette qualification n'a plus vraiment lieu d'être, puisque le caractère 'black box' des méthodes de machine learning a tendance à s'amoindrir, alors que de nouveaux travaux exposent de plus en plus de manières inédites de considérer les résultats donnés par ces techniques. La performance redevient alors le critère de décision fondamental, offrant alors aux modèles de machine learning toutes leurs chances contre les modèles classiques.

Nous reprendrons ainsi dans cette partie les modèles linéaires généralisés, notion abordée dans la partie III.1 de cet ouvrage. Néanmoins les GLM présentent plusieurs limites, dont leur manque

d'élasticité au niveau des variables numériques. En effet, comme son nom l'indique, le GLM traite ses variables linéairement. Hors dans la réalité, on peut faire face à des épisodes de rupture au niveau de certaines valeurs (par exemple une valeur seuil de 30000€ de valeur assurée qui conduit à une augmentation automatique de 50€ du tarif). Ces phénomènes sont impossibles à modéliser avec un GLM classique. De même, une inversion de la tendance du tarif ou une évolution non linéaire de l'évolution de la variable, comme notamment avec l'âge du conducteur, peut conduire à des inepties avec l'utilisation d'un GLM mal paramétré.

C'est pourquoi nous présenterons dans cette partie des algorithmes inspirés du GLM classique, mais permettant une plus grande liberté au niveau du traitement des variables. On évoquera ainsi les modèles additifs généralisés (Generalized Additive Model, GAM) et la modélisation MARS (Multivariate Adaptive Regression Splines), qui utilise des splines pour modéliser les variables non-linéaires.

IV.2.1 Modèles linéaires généralisés (GLM)

Comme expliqué dans la partie III.1, le but d'un GLM est de prédire le résultat comme une somme pondérée des variables placées en entrée. L'un de ses défauts majeurs est son nombre de paramètres à prendre en compte.

En effet, plus le nombre de paramètres augmente, plus le modèle est performant, car plus il est 'aidé' à comprendre le phénomène par l'utilisation des variables explicatives. Néanmoins, une explosion de variables explicatives (notamment dû à l'introduction de variables quantitatives qui nécessitent un one-hard-encoding, c'est-à-dire l'introduction de manière binaire de chaque modalité dans des nouvelles variables), conduit non seulement à une perte d'interprétabilité lorsqu'il s'agit de variables avec interactions, mais aussi à une perte de fiabilité dans l'estimation des paramètres.

Ainsi, un modèle avec une vingtaine de paramètres se révèlera souvent être bien plus performant qu'un modèle prenant en compte plus de trente paramètres. Cette remarque a justifié le chapitre III de ce mémoire.

A la fin de ce chapitre, nous sommes d'ailleurs arrivés à la construction d'un set de variable 'optimal', comprenant des variables interagissant entre elles. Dans le but de prouver l'efficacité de l'introduction d'un tel phénomène, nous appliquerons les modèles de ce chapitre sur deux sets de variables : celui sélectionné à la fin de la partie II et celui sélectionné à la fin de la partie III.

Pour faire référence au set de données utilisée lors de la modélisation, nous utiliserons en indice 'II' pour référer au set de fin de la partie II, et 'III' pour référer au set de fin de la partie III.

IV.2.1.1. GLM Sans interaction

Dans un premier temps, nous nous concentrerons sur les GLM sans interaction. Cela veut dire que chaque variable est traitée individuellement, comme si toutes les variables étaient indépendantes. Ce GLM est similaire à celui introduit en III.1.2.

Faisons donc cette hypothèse. Nous obtenons donc les résultats suivants :

	R^2 du modèle	RMSE Éch.Entr	RMSE Éch.Test	Nombre de variables
GLM_{II}	0.6478731	66.09556	64.41719	29
Modèle naïf	0	111.384	110.6228	0

Le RMSE est définitivement meilleur que celui obtenu sur notre modèle de base, mais reste néanmoins élevé. Nous allons donc tester d'autres alternatives.

IV.2.1.2. GLM Avec interactions quadratiques

Les modèles de régression linéaires (traditionnels et GLM) présument que l'effet d'une variable est le même quelles que soient les valeurs des autres variables.

Or ce n'est pas ce que nous trouvons dans la réalité. En effet, avant que l'interdiction du sexe comme variable tarifante ne soit mise en place, les assureurs avaient remarqué une corrélation entre ces deux variables. Une jeune femme aura donc moins de risque d'accidents qu'un jeune homme du même âge. Cette corrélation diminue avec l'âge. L'âge n'est donc pas indépendant du sexe.

D'autres couples de variables entre également en compte comme âge du conducteur et valeur assurée ou valeur assurée et poids, comme l'a démontré la partie III.4.2 sur la statistique H de Friedman.

Compte tenu de ce fait, il est assez dangereux d'utiliser le GLM dans ces conditions, ou du moins faire aveuglément confiance en ses prédictions. Le GLM réagit mal lorsque plusieurs de ses variables sont corrélées entre elles. Cela cause une grande variance sur les estimateurs, qui seront alors partagés par une information délivrée par deux variables. Cela peut conduire, comme on l'a vu précédemment, à retirer deux variables explicatives, qui indépendamment expliquent peu le résultat, mais ensemble sont importantes dans l'élaboration de la prime.

Une des méthodes pour améliorer le GLM contenant des variables non-indépendantes, est d'inclure des interactions quadratiques. Cela signifie que désormais, on détermine via le GLM, non seulement des coefficients pour les variables introduites, mais également des coefficients de corrélation. Si l'on reprend notre exemple avec le sexe et l'âge, cela veut dire que l'on aura un B1 = 0.5 dans le cas fille de 21 ans et un B2 = 0.75 dans le cas d'un garçon âgé de 21 ans. Et ceci pour tous les âges/catégories disponibles. On voit donc que même si l'on introduit une nouvelle information, le modèle est fortement pénalisé par l'ajout de toutes ces nouvelles variables (dont le nombre est égal à toutes les combinaisons possibles entre les deux variables). Ce nombre peut rapidement devenir rédhibitoire, c'est pourquoi il faut bien choisir les variables à croiser. A nouveau on est face à un choix de modélisation.

Ce problème a déjà été mentionné dans la partie III de ce mémoire, mais il est important de le garder en tête. De même, effectuer une pré-sélection des variables est utile et nécessaire, mais conduit forcément à une perte d'information plus ou moins importante.

On retrouve donc ci-dessous les résultats obtenus avec le set de variable contenant des interactions déterminées à la fin du chapitre III :

	R^2 du modèle	RMSE Éch.Entr	RMSE Éch.Test	Nombre de variables
GLM_{III}	0.7410656	56.67839	56.85238	34
GLM_{II}	0.6478731	66.09556	64.41719	29

Le RMSE du modèle de degré 2 est inférieur au RMSE obtenu dans le cas sans interactions. On voit donc la pertinence de rajouter ces interactions dans notre modèle. Néanmoins, cela reste compliqué à mettre en place, du fait de la multiplication de combinaisons disponibles au fur et à

mesure que l'on augmente le degré possible. L'analyse faite dans la partie III de ce mémoire est alors primordiale.

On notera que l'utilisation de la statistique H de Friedman en partie III.4.2. n'a pas permis l'introduction de variables de degré 2 (c'est à-dire l'introduction de la variable Age^2 par exemple). Hors en pratique, ce genre d'utilisation des variables est assez répandu. Néanmoins, dans notre étude, nous avons décidé de négliger ce type d'interaction, comme l'introduction un à un de variables au carré n'apportait à chaque fois que peu d'informations en elle-même au modèle.

IV.2.2 Modèles additifs généralisés (GAM)

Les GAM (Generalized Additive Model) se veulent comme une amélioration des GLM. Contre une légère perte d'interprétabilité, les GAM permettent de traiter les variables numériques de manière moins controversée. Développé par Trevor Hastie et Rob Tibshirani, le modèle GAM a pour vocation de rassembler les propriétés du modèle linéaire généralisé (GLM) avec celles du modèle additif. Les GAM ressemblent beaucoup aux GLM, à la différence près que le prédicteur linéaire dépend cette fois de fonctions de lissage inconnues des variables prédictives.

Ces fonctions de lissage sont généralement soit des fonctions polynômiales (permettant ainsi de prendre en compte des particularités telles qu'une variable au carré), soit des fonctions en escalier (c'est-à-dire ayant une valeur fixe particulière selon l'intervalle dans lequel se trouve la valeur de la variable X_j) soit des splines (fonctions polynômiales définies sur des intervalles).

IV.2.2.1. Introduction mathématique

Contrairement aux GLM, avec une fonction lien prédéterminée g, où on cherche $\beta_1, \beta_2, \ldots, \beta_q$ tels que :

$$g\big(E(Y)\big) = g(\mu) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_q x_q$$

Dans un GAM, on cherche plutôt les fonctions $fi : R -> R$ telles que :

$$g(E(Y)) = \beta_0 + f_1(\mathrm{X}_1) + f_2(\mathrm{X}_2) + \ldots + f_q(X_q)$$

Les fonctions f_i peuvent être ajustées en utilisant les moyennes non paramétriques ou paramétriques. Différentes méthodes existent pour estimer les f_i mais la plus courante est l'algorithme de backfitting.

L'algorithme de backfitting, introduit en 1985 par Leo Breiman et Jerome Friedman, permet de résoudre le problème susnommé. Il repose sur le fait que, si le GAM est correct, alors :

$$E[g(Y) - \beta_0 - \textstyle\sum_{k \neq j} f_k(X_k) \big| X_j] = f_j(X_j) \quad pour\ tout\ k = 1, \ldots, q$$

L'algorithme de backfitting résout ces q équations itérativement. A chaque étape, l'espérance conditionnelle des résidus partiels est modélisée avec un lissage univarié.

On initialise ainsi $\beta_0 = g(\mathrm{Y})$. Tant que tous les f_jn'ont pas convergés, on boucle sur $k = 1, \ldots, q$ et on pose $f_k = S\,(g(Y) - \widehat{\beta_0} - \sum_{j \neq k} f_j(X_j) \mid X_j)$. Si lorsqu'on arrive à f_q les f_j n'ont pas tous convergés, on reboucle sur $k = 1, \ldots, q$.

La solution apportée par cet algorithme n'est pas unique, sauf s'il n'y a pas de concurvité dans le modèle (c'est à dire qu'aucun terme de lissage ne peut être approximé par un ou plus des autres termes de lissage du modèle, c'est-à-dire qu'il n'y a pas de corrélation entre les variables).

Néanmoins, pour des soucis de simplicité, nous avons décidé d'opter pour une autre méthode, construite autour de l'approche de lissage de rang réduit. L'idée ici est de remplacer les fonctions de lissage inconnues par des extensions de bases.

On pose alors :

$$f_j(X_j) = \sum_{k=1}^{K_j} \beta_{jk} b_{jk}(X_j)$$

Où les $b_{jk}(X_j)$ sont des fonctions de bases, choisies pour leur bonnes propriétés théoriques d'approximation (splines de base par exemple). Les β_{jk} sont eux des coefficients à estimer lors de la phase de modélisation. K_j est choisi pour être suffisamment large pour pouvoir couvrir toutes les données, mais également suffisamment petit pour conserver l'efficience des calculs. Le résultat est unique à une constante près. On estime ensuite ces différentes fonctions en utilisant une version pénalisée de la méthode des moindres carrés itérativement repondérés.

On notera que des interactions peuvent être facilement ajoutées au modèle, sous la forme $f_i(X_j, X_k)$, sans que cela ne nuise à la structure de la fonction.

L'un des problèmes majeurs des GAM reste néanmoins le surapprentissage. Le nombre de paramètres lissants doit être petit pour contrecarrer ce problème ; mais on est alors probablement en dessous des degrés de liberté offerts par les données.

IV.2.2.2. Résultats

On va ici s'intéresser au résultat fourni par cet algorithme sur notre modèle. On a voulu répliquer les résultats obtenus par GLM, avec interactions et sans interaction. On trouve donc les métriques suivantes pour chaque modèle :

	R^2 du modèle	RMSE Éch.Entr	RMSE Éch.Test	Nombre de variables
GAM_{II}	0.651495	61.75477	64.44902	29
GAM_{III}	0.7410364	55.6816	56.82359	34
GLM_{III}	0.7410656	56.67839	56.85238	34

Les GAM offrent ici une performance similaire au GLM. Cela peut s'expliquer par le fait que les fonctions de bases sont des paramètres supplémentaires à estimer et peuvent amener un certain surapprentissage.

IV.2.3 Régression multivariée par splines adaptatifs (MARS)

La régression multivariée par splines adaptifs est également une forme de régression pouvant être considérée comme une extension des régressions linéaires. Elle est proposée en 1991 par J.H. Friedman, et a la particularité de modéliser automatiquement la non-linéarité des modèles et leurs interactions intrinsèques.

MARS étant une marque déposée par Salford Systems, la plupart des scientifiques ont contourné ce problème en produisant des implémentations nommées 'Earth'. Une réaction plutôt terre à terre.

IV.2.3.1. Introduction mathématique

Un modèle MARS s'exprime sous la forme :

$$\widehat{f_{MARS}}(x) = \sum_{i=1}^{k} c_i B_i(x)$$

Où les B_i sont des fonctions de base et les c_i sont des constantes déterminées par l'algorithme. Les B_i ne peuvent être que de trois formes :

- Une constante (étape 0 de l'algorithme)
- Une fonction charnière, c'est-à-dire une fonction qui s'exprime sous la forme $max(0, x - constante)$ ou $max(0, constante - x)$.
- Un produit de deux fonctions charnières ou plus. Une telle combinaison permet de prendre en compte les interactions éventuelles entre deux variables ou plus

Les c_i et les B_i sont déterminés par une régression pas à pas (stepwise regression). Le premier modèle créé par l'algorithme comprend juste la constante d'intersection, égale à la moyenne de la variable réponse. MARS ajoute ensuite à chaque étape une fonction de base et sa réciproque (toujours une paire), choisie comme étant la fonction permettant de diminuer le plus la somme des carrés des résidus (algorithme glouton).

Cette fonction est créée comme une combinaison des termes déjà existants avec toutes les variables (pour la nouvelle fonction de base) et toutes les valeurs de celle-ci. Les c_i sont ensuite déterminé par une régression linéaire.

Généralement, au bout de ce processus on obtient un modèle qui overfitte les données. C'est pourquoi on le couple généralement avec une élongation, c'est-à-dire qu'on retire une à une les fonctions contribuant le moins au modèle. Si le premier processus gère les fonctions par paires, la suppression peut se faire indépendamment. Il n'y a donc pas forcément de paires complètes dans le modèle final.

Il en résulte donc que les modèles MARS sont plus flexibles que les modèles de régression linéaire et gardent leur caractère essentiel d'interprétation. Contrairement aux GLM, MARS permet de gérer efficacement à la fois les variables catégorielles et numériques.

Une des différences notables est également que les modèles MARS ne nécessitent pas une préparation excessive des données en amont. Une pré-sélection des variables n'est donc pas primordiale, comme nous le verrons dans les résultats.

Les modèles MARS peuvent être également combinés avec les modèles GLM, en appliquant une fonction lien après que le modèle MARS ait été déterminé. Néanmoins, vu nos données, nous nous contenterons d'un modèle MARS unique.

IV.2.3.2. Résultats

Comme il l'a été mentionné plus haut, les modèles MARS sont programmés de telle sorte qu'il n'est pas forcément nécessaire que les données aient été pré-sélectionnées. Nous allons donc appliquer l'algorithme MARS sur trois sets différents de données, l'un comportant toutes les variables originelles, et les deux autres sets introduits précédemment. Pour les distinguer, nous caractériserons le set originel par l'indice 'I'.

	R^2 du modèle	RMSE Éch.Entr	RMSE Éch.Test	Nombre de variables
$MARS_I$	0.8023335	49.52098	50.28073	39
$MARS_{II}$	0.7853995	51.59862	52.12685	29
$MARS_{III}$	0.8157264	47.81391	48.39449	34
GAM_{III}	0.7410364	56.6816	56.82359	34

Le modèle MARS se révèle donc être bien plus performant que les modèles linéaires/additifs généralisés. Cette technique, simple à mettre en place, pourrait donc être implémentée plus fréquemment car son pouvoir prédictif est bien meilleur que celui des algorithmes traditionnels. De plus, l'utilisation de fonctions de bases permet, entre autres, de conserver une certaine interprétabilité.

IV.3. Algorithmes de Machine Learning

Évoquons maintenant les algorithmes de Machine Learning. Il y a 25 ans, les compétences pour réaliser de l'analyse prédictive se restreignaient à la connaissance du simple modèle de régression (linéaire et logistique). Les modèles de régression statistiques étaient alors les maîtres indiscutés de l'analyse prédictive. Les temps ont néanmoins changé avec le développement du 'big data' et l'évolution des capacités calculatoires des ordinateurs : le machine learning, et ses techniques d'apprentissage ont su révolutionner les pratiques. Mais que comprend l'appellation machine learning ? Il s'agit d'un ensemble de méthodes que les ordinateurs utilisent pour créer et améliorer des prédictions ou des comportements basés sur des données.

Cependant, comme le précise Dan Ariely, professeur de psychologie et d'économie comportementale israëlo-américain, 'Big data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it…'. En d'autres termes, il est compliqué de vraiment savoir ce que comprend ce mot-valise. C'est pourquoi la suite de ce mémoire se veut tenter d'éclairer ce vaste sujet que constituent le big data et le machine learning.

Lorsque l'on parle d'apprentissage ou plus généralement de 'data mining', l'une des grandes familles de techniques employées repose sur le précepte algorithmique : "diviser pour régner". Concrètement,il s'agit de procéder à des subdivisions récursives d'un problème initial de grande taille, afin de parvenir à de multiples problèmes de taille réduite. Les méthodes d'apprentissage par arbres de décision s'en inspirent. L'apprentissage permet d'identifier les sous-ensembles où les solutions

sont identiques. Dès lors, lorsqu'un nouveau cas est soumis au système, l'attribution de la solution revient à l'attribution des données d'entrée dans un sous-ensemble pré-établi.

Afin de gagner en efficacité et en précision, la tâche est parfois distribuée ou répétée plusieurs fois, comme confiée à différents experts. La solution générale s'écrit alors comme la combinaison des réponses de chacun. On parle de combinaison statique si elle ne dépend pas de l'entrée (typiquement la moyenne des différentes réponses) ; on parlera de combinaison dynamique le cas contraire. Ces approches seront abordées sous l'appellation de "méthodes ensemblistes" ou "méthode d'agrégation".

L'intelligence artificielle est souvent associée à un grand nombre de promesses et d'attentes. Internet est apparu dans les années 90 mais le monde espère déjà une nouvelle révolution informatique. Avec la recrudescence de films futuristes, les hommes sont dans l'attente de voir se matérialiser ce qu'ils peuvent voir sur un écran. Les machines surpassent aujourd'hui les humains dans de nombreuses tâches et de nombreux domaines, comme jouer aux échecs (ou même au Go), ou à prédire la météo. Toutefois, même si la science ne cesse de développer des outils de plus en plus performants, les entreprises et les institutions sont encore réticentes à leur utilisation.

Et cela non pas parce qu'elles n'ont pas confiance en leur capacité, mais parce que cela prend du temps et de l'argent d'intégrer ces nouvelles méthodes à leurs processus actuels. Même si la demande en recrutement de data scientists ne cesse de croître, les entreprises ne sont souvent pas prêtes à les intégrer et à leur confier des tâches de simplification / amélioration de leur process.

On a notamment soulevé tout au long de ce document le fait que les modèles de régression linéaires se trouvent souvent en défaut dans les cas où le lien entre les variables explicatives et la variable à expliquer est non linéaire, ou si il y a présence de multicolinéarité. C'est pourquoi différents algorithmes ont été développés pour parer à ces problèmes. Ces techniques, se basant notamment sur les avancées récentes en machine learning, commencent toutefois doucement à faire leur apparition dans les différents groupes assurantiels et bancaires.

Dans cette troisième partie les principales méthodes de machine learning développées à l'heure actuelle seront introduites. Dans le cadre de notre étude, seules les méthodes dites "d'apprentissage supervisé" sont pertinentes; en effet, nous connaissons la sortie que l'algorithme doit produire (la prime) ; notre but étant de s'en rapprocher le plus possible. L'apprentissage non-supervisé, qui n'a pas de résultats étiquetés, ne s'applique donc pas à notre problématique.

On passera également, pour chaque méthode développée, dans la procédure habituelle de modélisation à travers les techniques de machine learning :

1. Choix de la technique de modélisation : arbre de décision, méthode d'ensemble, réseaux de neurones…
2. Spécification d'un protocole d'évaluation : division de la base de donnée et définition des critères employés (minimisation des écarts au carré, indice de Gini…)
3. Construction du modèle et optimisation des paramètres
4. Evaluation de la performance du modèle selon le critère choisi sur l'échantillon de test

IV.3.1 Introduction aux modèles supervisés

En machine learning ils existent deux grandes familles de modèles :

- Les modèles supervisés, qui consistent à apprendre une fonction de prédiction à partir d'exemples annotés, c'est-à-dire dont on dispose de la sortie. On y retrouve des problèmes de classification et de régression

- Les modèles non-supervisés, où les données ne sont pas étiquetées (on ne connaît pas la sortie sur nos données) ; il s'agit donc de découvrir les structures sous-jacentes à ces données

Les modèles supervisés sont à distinguer des modèles non-supervisés, dans lesquels on trouve majoritairement des algorithmes de classifications en clusters comme les k-means, les agrégateurs ou des algorithmes permettant de détecter des outliers (Isolation forest) ou encore de réduire la dimensionnalité des données (Generalized Low Rank Models, Principal Component Analysis). Dans notre problématique, ces algorithmes ne sont pas adaptés ; on se concentrera donc uniquement sur les modèles supervisés.

Pour créer un modèle supervisé, on donne à l'algorithme une entrée et une sortie, là où le modèle non-supervisé n'aura lui besoin que d'une entrée. A partir de cette entrée, l'algorithme va chercher un pattern qui va lui permettre d'établir des règles.

On distingue au sein des algorithmes supervisés deux grandes familles :

- Les méthodes utilisant des arbres : arbres de régression, bagging, forêt d'arbres aléatoires, la méthode du gradient boosting (GBM)
- Les autres méthodes comme : réseaux de neurones, machine à support vectoriel (SVM)…

Ces méthodes, apparues dans les années 70, sont sans cesse améliorées. Ainsi, on remarquera que si l'arbre CART a été développé dans les années 1970 par Léo Breiman, il a été amélioré à travers les Random Forest par Tin Kam Ho dont la méthode a été proposée pour la première fois en 1995. Quelques années plus tard, le terme de gradient boosting fera son apparition dans la communauté de la data science.

On présentera donc par la suite les différents types de méthodes, selon le regroupement fait ci-dessus, c'est-à-dire modèles utilisant des arbres contre modèles n'utilisant pas d'arbres. Notre but via l'algorithme étant toujours, comme ce qui a été introduit avec les GLM, de trouver une fonction $f : \ddot{X} -> Y$ où $\ddot{X}$ représente l'espace engendré par les variables explicatives $X_1, X_2, \ldots, X_p$ et Y est l'espace de la variable réponse.

IV.3.2 Méthodes avec arbres

De nombreux modèles sont non linéaires mais sont tout de même traités linéairement par les méthodes GLM pour plus de simplicité car il est difficile de préciser un modèle lorsque les relations ne sont pas linéaires et que le nombre de variables explicatives est important. Ainsi souvent le "vrai" modèle est approximé par un modèle linéaire. Or, lorsque l'on veut prédire des données différentes de celles qui ont servi à construire le modèle, les prévisions risquent d'être très mauvaises. Les arbres sont des algorithmes non linéaires, ce qui permet de relâcher cette hypothèse forte des GLM qu'est la linéarité.

Les arbres ne sont pas monotones, c'est à dire qu'une même variable explicative continue peut avoir différents effets, par exemple s'il existe une valeur seuil en-dessous de laquelle cette variable a un effet positif sur la variable d'intérêt et au-dessus un effet négatif, un noeud pourrait être construit en cette valeur seuil pour partager la population en deux groupes plus homogènes et partageant ce même effet. A contrario, le modèle GLM estime un unique β_j exprimant l'effet moyen de cette variable explicative, ce qui conduit souvent à une mauvaise estimation si l'on est dans ce cas.

A contrario, les arbres de régression sont peu performants lorsqu'il s'agit de traiter de relations linéaires car elles ne pourront, au mieux, qu'être approximées.

Les méthodes avec arbres sont les plus connues parmi les méthodes de machine learning. En effet la distribution visuelle d'un arbre permet facilement de comprendre la répartition des données. De plus, appréhender un arbre de décision ou de régression est rapide et peut être fait par n'importe quel non initié.

Les modèles avec arbres divisent ainsi à plusieurs reprises les données selon certaines valeurs seuil, homogénéisant ainsi les données dans des sous-database. Les sous-databases terminales sont appelées 'feuilles'. Pour obtenir la prédiction pour une observation dans le cas d'une régression, on moyenne la réponse de toutes les observations présentes dans la feuille à laquelle elle appartient.

Nous ne verrons ici que les arbres utilisant la méthode CART (Classification And Regression Trees). D'autres méthodes existent comme l'algorithme ID3 (Iterative Dichotomiser 3, où un split peut avoir plus de deux enfants), CHAID (Chi-Square Automatic Interaction Detection, particulièrement compliqué à paramétrer) ou ASSISTANT (descendant de ID3, permet une hiérarchie à l'intérieur des classes au lieu de les forcer à être disjointes). Le lecteur intéressé pourra consulter le chapitre 4.7 de l'ouvrage 'A compendium of Machine Learning, Volume 1' par Garry Briscoe et Terry Caelli, pour plus d'informations sur ces techniques.

IV.3.2.1. Arbres de régression

On se contentera ici d'expliquer succinctement l'algorithme d'arbre de régression.

Un arbre de régression permet de lier y_i la variable réponse avec $X^{(i)} = (x_{i,1}, x_{i,2}, \dots, x_{i,p})$ les variables explicatives. Pour cela une fonction f est modélisée telle que :

$$\widehat{y_i} = f(X^{(i)}) = \sum_{m=1}^{M} c_m \, I_{\{X_i \in R_m\}}$$

Où R_m représente un sous-ensemble feuille et I est la fonction indicatrice. Les c_m sont alors les réponses moyennées des observations présentes dans la feuille R_m.

A chaque scission, l'algorithme CART sélectionne une variable parmi toutes celles disponibles ; pour chacune de ces variables, il détermine ensuite quelle valeur de seuil minimise la variance de Y. Cette valeur peut être de deux sortes, selon que la variable traitée est de type numérique ou de type catégorielle. Si la variable est numérique, alors l'algorithme cherchera à 'couper' la variable en deux de manière à optimiser la scission ; si la variable est catégorielle, alors l'algorithme regroupera en deux groupes les différentes modalités de la variable.

Une fois le seuil déterminé, l'algorithme effectue la scission la plus intéressante, c'est-à-dire celle permettant de réduire au mieux l'hétérogénéité présents dans les feuilles de l'arbre résultant de cette division. Cette étape est ensuite répétée sur chaque feuille, jusqu'à ce que le critère d'arrêt soit atteint. Ce dernier peut se présenter sous diverses formes telles qu'un nombre d'observations minimum dans un nœud pour pouvoir continuer la division, un critère d'amélioration minimale à dépasser pour autoriser une scission, un nombre maximum de feuilles…

Un arbre CART est donc intéressant pour sa capacité d'interprétabilité. Contrairement au GLM, il permet de prendre en compte les interactions entre les variables à tout niveau. En plus de parvenir plutôt décemment à capter les interactions, l'arbre CART permet ainsi de regrouper les données en des groupes distincts qui sont souvent plus faciles à comprendre que des points dans un

hyperplan multidimensionnel (cf régression linéaire). On obtient également de bonnes explications, facilement compréhensible par un humain car elles sont contrastées et souvent sélectives.

Avec le graphique ci-dessous, on peut voir les premiers niveaux de l'arbre entraîné sur notre échantillon sont montrés. On retrouve bien notre prime moyenne de 325€ sur l'échantillon d'entraînement. Ensuite une première segmentation forte se fait entre les assurés ayant le permis depuis plus de 13 ans : ces derniers payent en moyenne 290€ (69% des assurés), alors que ceux ayant eu le permis depuis plus récemment payent en moyenne 403€. La segmentation se poursuit ensuite parmi les valeurs assurées et l'âge du conducteur. On remarquera néanmoins que pour un même niveau, les variables segmentantes ne sont pas les mêmes ; cela signifie que certaines variables ont un impact plus fort chez une certaine catégorie de population. Hors ce genre de phénomène (interactions entre variables) est difficile à déceler avec des modèles statistiques.

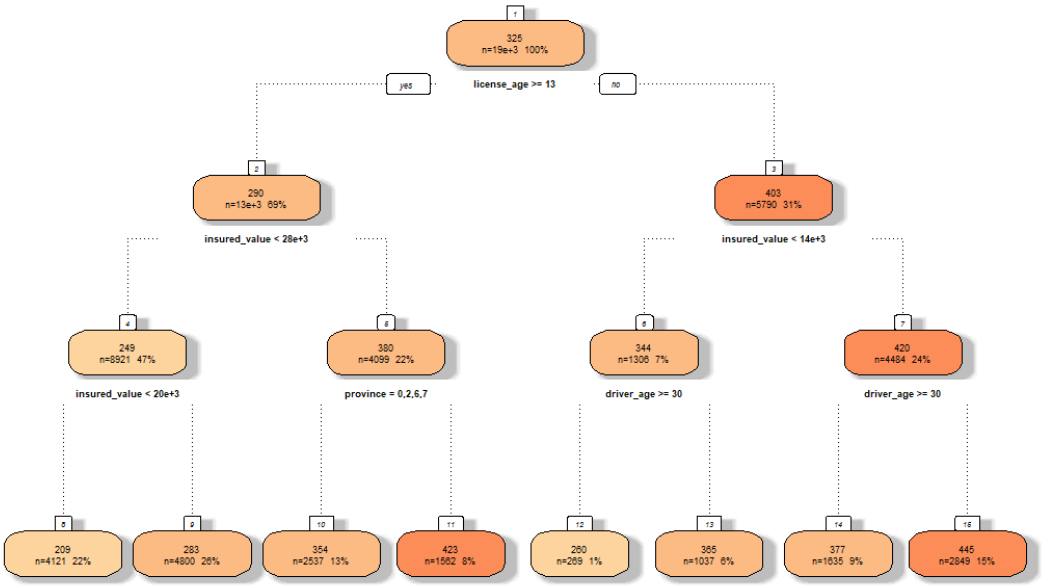


Figure 38 : Arbre de régression réalisée sur les données (profondeur = 3)

L'un des défauts des arbres de régression est leur forte variabilité. En effet, pour peu que le nombre de données soit insuffisant pour obtenir des résultats stables (grandes valeurs discriminantes), ou que l'on veuille aller assez loin dans l'arbre, il est peu probable que l'on obtienne de manière successive deux fois le même arbre ; ainsi si l'on altère légèrement l'échantillon d'entraînement fourni à l'arbre cela peut causer de grandes modifications. Il est ainsi rare d'obtenir un résultat fiable et stable avec un seul arbre de régression sur un même modèle originel.

Ainsi, si l'on tombe 'mal' et que l'on fait maladroitement confiance à notre arbre, on peut en tirer des conclusions complètement erronées, pour peu que l'échantillon d'entraînement ne soit pas assez représentatif du modèle sous-jacent. Sa faible fiabilité n'en fait donc pas un candidat optimal pour notre problématique de reverse engineering. C'est d'ailleurs pourquoi l'arbre de décision est souvent classifié comme un apprenant faible, c'est-à-dire seulement faiblement corrélé avec le vrai résultat. Par opposition, un apprenant fort produit des résultats arbitrairement bien corrélés à ces derniers.

Une alternative a été développée pour répondre à ce problème : l'ensachage (plus communément appelé par sa dénomination anglaise, le bagging). On notera que la traduction française n'est pas la plus pertinente puisqu'à l'origine le mot bagging vient de la contraction de bootstrap aggregating, étymologie que l'on ne retrouve pas dans la version française.

IV.3.2.2. Bagging et Random Forest

Il a été vu dans la sous-partie précédente que l'un des défauts des arbres de régressions était de parfois parvenir à des conclusions complètement erronées, pour peu que l'échantillon de données ne soient pas assez représentatif des données, ou que celui-ci comprenne de nombreux outliers.

Le bagging a ainsi été développé par Léo Breiman en 1994 pour répondre à ce problème. Même s'il est plus communément appliqué avec des arbres, le principe de bagging peut néanmoins être répliqué avec n'importe quelle méthode.

Le bagging se base sur une méthode d'inférence statistique bien connu des statisticiens : le bootstrap. Le fondement du boostrap est de répliquer de nombreuses fois les données dans différents jeux (rééchantillonnage) pour arriver à une certaine stabilité du modèle. Cela permet également de réduire la variance que l'on obtiendrait avec la création d'un unique arbre, ainsi que les risques d'overfitter le modèle. Plus généralement, le bootstrap est assez similaire à une procédure de 'moyenne de modèles'.

Le bagging appliqué aux arbres de régression, consiste donc à entraîner k arbres sur k échantillons différents de données issus de notre échantillon d'entraînement. L'échantillonnage se fait généralement avec replacement, c'est-à-dire qu'une même observation peut apparaître plusieurs fois dans un même sous-échantillon. Ensuite, lorsque l'on passe à la phase de prédiction, les observations sont fournies à l'algorithme qui obtient k prédictions et renvoie finalement la moyenne de ces k prédictions.

Mathématiquement, si on appelle f_k les différentes fonctions produites par les arbres réalisés sur les échantillons $\ddot{X_k}$, on obtient pour prédicteur final :

$$\hat{f}(X^{(i)}) = \frac{1}{k}\sum_{i=1}^{k} f_k(X^{(i)})$$

L'algorithme de bagging est toutefois loin d'être parfait. En effet, dans notre cas, puisque nous utiliserons la technique de bagging sur des arbres de décision CART, on peut être confronté à une grande corrélation dans nos prédictions. En effet, puisque les arbres CART reposent sur des algorithmes gloutons (c'est-à-dire que le choix d'un optimum local est fait à chaque étape), cela peut conduire à beaucoup de similarités structurelles au sein des arbres.

Hors, combiner des prédictions provenant de modèles multiples fonctionne mieux lorsque les prédictions réalisées par ces sous-modèles sont décorrélées, ou au pire, seulement légèrement corrélées. L'algorithme de forêt aléatoire a ainsi été conçu pour pallier à cette contrainte, puisqu'il permet de réduire la corrélation entre tous ces sous-modèles.

Pour cela les algorithmes de forêts aléatoires (random forest) imposent une contrainte supplémentaire aux arbres CART ; à chaque fois que l'arbre se divise, la scission ne se fait non plus sur une des variables présente dans le data set, mais parmi une des variables présentes dans un sous-data set issu de ce data-set. La taille de ce sous-data-set est généralement fixé à 1/3 pour les problèmes de régression et ce sous-data-set varie aléatoirement à chaque nœud. La taille de ce sous-set est néanmoins un hyper-paramètre qui peut être tuné lors de la modélisation (voir partie IV.4.2).

Les arbres sont donc différents à chaque itération, puisqu'aucun arbre (ou presqu'aucun) ne peut réaliser ses scissions sur les mêmes sous-sets de variables. On limite donc le phénomène de corrélation entre les arbres.

Néanmoins, l'un des revers des algorithmes de bagging et de forêt aléatoire est que l'on perd rapidement l'aspect visuel des arbres de décision unique.

IV.3.2.3. Boosting

Petit dernier des algorithmes utilisant des arbres de régression, le boosting est relativement récent. Particulièrement puissant et efficace, il a su montrer son potentiel lors des compétitions Kaggle où il est toujours très plébiscité comme algorithme de résolution. Similairement aux algorithmes de bagging et de forêt aléatoire développés précédemment, le but originel du boosting est réduire la variance due à l'utilisation d'un seul arbre de régression. Contrairement aux deux autres méthodes, le boosting ne cherche pas à moyenner les prédictions de différents arbres ; à la place, il cherche à créer un apprenant fort basé sur des apprenants faibles.

Pour cela l'algorithme de boosting utilise itérativement des apprenants faibles (typiquement des arbres de régression) et les ajoute ensuite à son apprenant 'fort'. A chaque ajout d'un apprenant faible, les données sont repondérées ; les observations dont les prédictions sont les plus éloignées de la réalité se voient donner un poids plus fort, alors que les prédictions proches de la réalité voient leur poids diminuer. Ainsi, le prochain apprenant faible devra se focaliser davantage sur la correction des données dont la prédiction est peu pertinente. Cela peut causer une non-robustesse de l'algorithme lorsque les données sont bruyantes et/ou contiennent de multiples outliers.

L'algorithme de boosting repose donc sur un ajout successif d'arbres de régression, qui sont traités en série plutôt que parallèlement. En reprenant le formalisme mathématique introduit plutôt, on retrouve qu'à chaque étape on doit construire un arbre de régression $h_m(\ddot{X})$ tel que :

$$h_m(\ddot{X}) = \sum_{j=1}^{J_m} b_{jm} I_{R_{jm}}(\ddot{X})$$

Où J_m représente le nombre de feuilles de l'arbre h_m, et R_{jm} la région définie par la feuille j de l'arbre h_m.

Ces arbres h_m se construisent sur les pseudos-résidus :

$$r_{im} = -\left[\frac{\partial L(y_i,\ F(X^{(i)})}{\partial F(X^{(i)})}\right]_{F(\ddot{X})=F_{m-1}(\ddot{X})}$$

Ils cherchent donc à prédire les r_{im} en se basant sur les différentes variables X_j. $F(\ddot{X})$ correspond ici à l'apprenant fort.

L'apprenant fort à l'étape m s'écrit donc :

$$F_m(\ddot{X}) = F_{m-1}(\ddot{X}) + \gamma_m h_m(\ddot{X})$$

Où $\gamma_m = arg\min_{\gamma} \sum_{i=1}^{n} L(y_i,\ F_{m-1}(X^{(i)}) + \gamma h_m(X^{(i)}))$, L étant la fonction de perte différentiable définie en amont.

Différentes versions de boosting existent, et dans notre étude nous avons utilisé la méthode de boosting par gradient (GBM, Gradient Boosting Method). Le GBM a été imaginé par Leo Breiman qui avait alors comparé le boosting à un algorithme d'optimisation soumis à une fonction de coût adaptée. Mais c'est Jerome H. Friedman qui proposa le premier un algorithme de régression basé sur le boosting par gradient.

Cela revient à évincer l'utilisation d'un unique γ_m en utilisant plutôt un γ_{jm} dépendant de chaque feuille de l'arbre h_m. Cela revient à se séparer des coefficients b_{jm} et l'algorithme devient alors :

$$F_m(\ddot{X}) = F_{m-1}(\ddot{X}) + \sum_{j=1}^{J_m} \gamma_{jm} I_{R_{jm}}(\ddot{X}) \quad ; \quad \gamma_m = arg\min_{\gamma} \sum_{i=1}^{n} L(y_i,\ F_{m-1}(X^{(i)}) + \gamma)$$

Les algorithmes de boosting sont ainsi vus comme des algorithmes optimisant une fonction de coût en choisissant itérativement une fonction qui pointe dans la direction négative du gradient. Ces fonctions sont en réalité de nouveaux arbres de régression.

De nombreux hyper-paramètres permettent de raffiner le GBM pour que celui-ci s'approche le plus possible des données. On notera notamment, la possibilité de faire un GBM stochastique s'appuyant sur les motivations qui menèrent Breiman à proposer le bagging (sous-échantillonnage de l'échantillon d'entraînement pour la construction de chaque arbre) ou bien encore l'introduction d'un paramètre de rétrécissement permettant de limiter l'impact de chaque nouvel arbre sur l'apprenant fort.

IV.3.3 Autres méthodes

Les arbres de régression sont toutefois de plus en plus délaissés dans les secteurs de la data science au profit des réseaux de neurones sont très populaires. Par leur architecture linéaire, ils sont particulièrement présents dans les secteurs de reconnaissance visuelle, de diagnostics médicaux ou encore de jeux impliquant un processus de décision (échec ou backgammon par exemple).

D'autres méthodes tirent également leur épingle du jeu comme les machines à vecteurs de support (SVM) ou les méthodes ensemblistes

On présentera ainsi succinctement le principe des réseaux de neurones, des SVM et de la méthode d'empilement (stacking) appelée également algorithme Super Learning.

IV.3.3.1 Réseau de neurones

Les réseaux de neurones ont la particularité de pouvoir être à la fois des algorithmes d'apprentissage supervisé mais aussi non-supervisé. Ils s'inspirent directement du fonctionnement des neurones biologiques simplifiés, désignés également sous l'appellation neurones formels.

Le neurone formel est conçu comme un automate doté d'une fonction de transfert qui lui permet de transformer ses entrées en sorties selon des règles précises. Généralement ces fonctions de transfert sont appelées fonctions d'activation et peuvent prendre diverses formes. Les plus répandues sont :

- Maxout. Un réseau maxout est un réseau de neurones multi-couches qui emploie des unités maxout dans ses couches cachées. Une unité maxout sortant le maximum d'un set de k produits scalaires entre k vecteurs pondérés et le vecteur de la couche précédente. En combinant cette approche avec l'abandon de neurones, on obtient une bonne approximation de la moyenne du modèle avec des paramètres partagés à travers un large nombre de réseaux.
- Tanh (tangente hyperbolique). Cette fonction d'activation sigmoïdale permet de garder les entrées quasi nulles proches de zéro. Les entrées négatives y sont amplifiées négativement. Tanh possède également l'avantage d'être

différentiable et monotone. Elle convient néanmoins plus à des problématiques de classification

- ReLU (Rectified Linear Unit). Il s'agit de la fonction d'activation la plus utilisée. Très simple, il s'agit de prendre le maximum de la valeur de l'entrée avec zéro.
- Heavyside. Il s'agit de la fonction indicatrice de $\mathbb{R}^+$

Une fonction d'activation, comme son nom l'indique, compare la somme des entrées avec une valeur seuil. Si cette valeur est égalée ou dépassée, alors le neurone est activé et renvoie un signal en sortie. La valeur de ce signal dépend de la fonction d'activation choisie. L'efficacité de la transmission des signaux d'un neurone à l'autre peut varier : on parle de 'poids synaptique', et ces poids sont déterminés lors de l'apprentissage du modèle.

La génération de réseaux de neurones utilisée le plus souvent aujourd'hui est le perceptron multicouche. Ces systèmes permettent le traitement de phénomènes non-linéaires, en se reposant sur la rétropropagation du gradient de l'erreur dans des systèmes à plusieurs couches.

Un réseau de neurones se compose d'une succession de couches dont chacune prend ses entrées sur les sorties de la précédente. Chaque couche i est composée de N_i neurones prenant leurs entrées sur les N_{i-1} neurones de la couche précédente. A chaque synapse est associé un poids synaptique, de sorte que les N_{i-1} neurones sont multipliés par ce poids, puis additionnés par les neurones du niveau i. Les poids sont définis par un phénomène de rétropropagation ; cela signifie qu'après que les poids aient été définis une première fois, le réseau est parcouru dans le sens inverse pour corriger l'erreur commise au niveau des poids qui y ont le plus contribué. Ce processus, associé à l'entièreté du set d'entraînement est appelé Epoch, et est répété autant de fois que spécifié en amont dans les hyper-paramètres.

Les réseaux de neurones constituent une méthode complexe à mettre en oeuvre en raison des calculs qu'elle nécessite. De plus ils sont relativement difficiles à configurer puisqu'il n'existe pas de méthode explicite pour déterminer le nombre de neurones ni le nombre de couches cachées nécessaires au fonctionnement optimal de l'algorithme. Les réseaux de neurones ont également le défaut de ne pas permettre de mesurer l'influence d'une variable sur la variable de sortie dès lors qu'une couche cachée est présente (aspect « boîte noire »), et la fonction de prédiction produite n'est pas interprétable en termes de règles simples.

Néanmoins, si les données sont trop complexes pour les méthodes d'apprentissage traditionnelles, les réseaux neuronaux sont à privilégier, en ce sens qu'ils utilisent non seulement une pondération des variables mais également une pondération des liaisons entre celles-ci. Ils permettent ainsi de mettre en évidence des relations de causalité que les autres méthodes ne voient pas, en particulier dans les situations non-linéaires.

IV.3.3.2 Support Vector Machine

Les Support Vector Machines (SVM) sont une alternative non négligeable aux méthodes d'apprentissage les plus performantes, y compris les modèles agrégés. Le principal reproche qui leur est adressé est de n'être directement applicable que pour des variables à prédire à deux modalités. Même dans ces cas-là, les paramètres du modèle solution restent néanmoins difficilement interprétables. Cependant, puisque la construction et le choix des fonctions noyaux est laissé libre à l'utilisateur, ces méthodes s'avèrent être très flexible et s'adaptent à de nombreuses problématiques.

Les SVM sont peu utilisés en régression, mais préférés en classification. Ils se basent sur la théorie développée par Vapnik et Chervonenkis dans les années 1990. Ils offrent la capacité de

travailler avec des données de grandes dimensions, tout en étant facilement applicable, du fait du nombre faible d'hyperparamètres qu'ils requièrent.

Dans certains domaines, les SVM se montrent même plus performants que d'autres méthodes comme les réseaux de neurones ou les mélanges gaussiens. Le principe du SVM est simple. Prenons un exemple de classification en dimension 2. Utiliser le SVM revient à trouver la droite dans le plan maximisant la marge, c'est-à-dire la distance entre la droite et les échantillons les plus proches, tout en séparant les données en deux groupes distincts.

Pour les problèmes à plus grande dimension, ou pour traiter les cas où les données ne sont pas linéairement séparables, les SVM transforment l'espace de représentation des données d'entrée en un espace de plus grande dimension dans lequel il est probable qu'il existe une séparation linéaire. Ceci se fait via une fonction noyau, respectant les conditions du théorème de Mercer, qui a l'avantage de ne pas nécessiter la connaissance explicite de la transformation à appliquer pour le changement d'espace.

La dénomination du SVM s'appliquant au problème de régression est SVR (Support Vector Regression). Le but du SVR est différent d'une régression traditionnelle, où le but est de minimiser le taux d'erreur ; pour le SVR, l'objectif est de contenir l'erreur dans une certaine marge. On cherche donc à positionner notre droite séparatrice de telle sorte que le maximum de points soient contenus entre cette droite plus ou moins ε.

IV.3.3.3 Super Learning

Le Super Learning apparaît pour la première fois dans le journal *'American Statistical Application in Genetics and Molecular Biology'* dans un article rédigé par Mark Van der Laan, Eric Polley et Alan Hubbard. Les algorithmes Super Learner impliquent d'entraîner un modèle metalearner (c'est-à-dire un autre modèle parmi ceux présentés dans cette section) de deuxième niveau, permettant ainsi de trouver la combinaison optimale des apprenants de base.

Contrairement au bagging, qui permet de rassembler des apprenants faibles, le stacking rapproche des apprenants forts et divers. Bien que ce concept ait été évoqué dès 1992, ses garanties théoriques n'ont été prouvées qu'en 2007 par Van der Laan, Polley et Hubard. Dans leur article, ces statisticiens ont démontré que l'ensemble Super Learner représente un système d'apprentissage asymptotiquement optimal.

L'algorithme Super Learner utilise la validation croisée pour estimer les performances de différents modèles de machine learning (ou de mêmes modèles mais avec des hyper-paramètres différents). Il crée ensuite une moyenne pondérée optimale de ces modèles, en utilisant la performance du set de données.

Ainsi, si on entraîne L forêts aléatoires, basés sur des hyper-paramètres différents, il est possible de les regrouper dans un modèle Super Learning. Pour cela, chaque forêt aléatoire doit être entraînée sur le même set d'entraînement. Une validation croisée est également opérée sur les mêmes échantillons. Les prédictions sont ainsi sauvées dans une matrice de taille $n \times L$, qui, couplée avec le vecteur réponse Y, forme la donnée de 'niveau 1'.

Un modèle dit 'metalearner' est ensuite entraîné sur cette donnée de 'niveau 1'. Pour obtenir des prédictions de ce modèle Super Learner sur le set de données test il faut d'abord générer les prédictions via les L forêts aléatoires, puis fournir ces résultats au metalearner.

On expérimentera ainsi cette méthode de deux manières différentes :

- En l'utilisant sur une vingtaine de GBM tunés de manière aléatoire pour mesurer son efficacité contre la sélection du modèle produisant le meilleur RMSE
- En l'utilisant sur les différents modèles optimaux trouvés pour tous les algorithmes mentionnés plus haut (GLM, GBM, forêt aléatoire, réseau de neurones)

IV.4. Benchmark

Maintenant que nous sommes passés à travers les différentes méthodes pouvant être utilisées pour répondre à notre problématique, nous allons pouvoir nous attacher à comparer leurs différentes performances, tout en rappelant brièvement les avantages et inconvénients de chaque méthode.

Pour cela nous détaillerons dans chaque section, trouvant leur équivalent dans la partie IV.3, les différents paramètres qui ont été modifiés dans l'optique d'obtenir le meilleur modèle. En effet, même s'il ne l'a été mentionné que de manière laconique par le passé, l'une des phases primordiales lors de la modélisation se trouve dans le paramétrage des différents modèles. Cette phase, souvent référée comme phase de tuning, permet d'encadrer au mieux le modèle pour que celui parvienne à obtenir les meilleurs résultats. Cela évite également quelques soucis, notamment le fait d'être bloqué au niveau d'un optimum local ou de coller de trop près les données d'entraînement.

Une section a également été rajoutée pour évoquer brièvement une méthode combinant à la fois statistique et algorithmique de machine learning. Finalement, un commentaire et une critique sera faite envers les différents résultats obtenus, permettant également de faire un lien avec la partie V de cet ouvrage, sur l'interprétation et le jugement pouvant être tenu à propos de ces modèles.

IV.4.1 Méthodes statistiques

Suite aux différentes phases de calibrage des modèles retenus, et de sélection de variables (partie III), nous avons abouti à l'élaboration de modèles finaux susceptibles d'être comparés. Les résultats obtenus, qui ont également été présentés dans la partie IV.2, sont rassemblés ci-dessous. Seuls ont été conservés les meilleurs modèles dans chaque catégorie. On notera que tous les modèles sélectionnés l'ont été vis-à-vis de l'échantillon de variables issues de la partie III de ce mémoire ; cela souligne à quel point cette étape est importante en statistiques.

	R^2 du modèle	RMSE Éch.Entr	RMSE Éch.Test	Nombre de variables
Modèle naïf	0	111.384	110.6228	0
GLM_{III}	0.7410656	56.67839	56.85238	34
GAM_{III}	0.7410364	56.6816	56.82359	34
$MARS_{III}$	0.8157264	47.81391	48.39449	34

Grâce à ce tableau on peut voir la puissance de l'algorithme MARS et de l'utilisation de splines ; néanmoins, malgré ses bonnes performances, il reste encore négligé par les équipes actuarielles de nos jours.

IV.4.2 Méthodes de machine learning

Avant de s'intéresser aux résultats fournis par ces méthodes, rappelons de manière simple et imagée le fonctionnement des principaux algorithmes par arbres. Vishal Morde propose ainsi une analogie entre les différents algorithmes et un entretien d'embauche :

- Arbre de décision : Le responsable recrutement interviewe le candidat selon ses propres critères personnels. La d: écision finale est celle du responsable.
- Bagging : Le responsable recrutement n'est plus seul, mais fait partie d'un panel, dont chaque membre dispose d'un droit de vote. La décision finale est donc prise démocratiquement selon le vote de chaque membre
- Forêt aléatoire : Processus similaire au bagging, sauf que cette fois chaque membre du panel testera le candidat seulement selon certaines de ses qualifications (une interview technique et une interview relationnelle par exemple, pour un panel de deux personnes)
- Boosting : Chaque membre du panel modifie ses critères d'évaluation en se basant sur le feedback donné par l'interviewer précédent. L'efficacité du process d'entretien est alors 'boostée' par le déploiement d'un process plus dynamique
- Gradient Boosting : Les candidats les moins qualifiés sont d'office retirés après étude de leur interviews par un cabinet de conseil en stratégie

Comme il l'a été précisé plus haut, le tuning est une phase importante du processus de modélisation, et encore plus quand il en vient aux algorithmes de machine learning. De nombreux hyper-paramètres ont été configurés pour pouvoir prendre une pléiade de valeurs. Le lecteur souhaitant plus de détails sur cette étape pourra se référer à l'annexe A.4.

Il est également important de préciser qu'après avoir fait opérer les différents algorithmes sur la base de données contenant toutes les variables présentes en II, et sur une base de donnée ne contenant que celles sélectionnées dans la partie III, il a été choisi de demeurer sur la base de données complète, c'est-à-dire celle de la fin du chapitre II. En effet les résultats étaient légèrement meilleurs avec toutes les variables disponibles. Puisque les algorithmes de machine learning traitent en leur sein toute problématique liée aux interactions, la sélection faite en partie III ne s'est pas révélée plus évocatrice que ses autres versions.

Pour chaque type de modélisation, cent modèles ont été testés et les meilleurs sur critère du RMSE ont fini par être sélectionnés. Pour les méthodes ensemblistes, vu que le GBM offrait le meilleur résultat, nous avons souhaité comparer la méthode présentée en IV.3.3.3 sur les 20 modèles GBM ayant été auparavant modélisés.

Un modèle ensembliste comprenant le meilleur GBM et la meilleure forêt aléatoire a également été testé ; il est présenté ci-dessous sous l'appellation 'Ensembliste RF + GBM'.

	R^2 du modèle	RMSE Éch.Entr	RMSE Éch.Test	Nombre de variables
Modèle naïf	0	111.384	110.6228	0
Arbre de décision	0.8587748	41.85803	46.10937	39
Bagging	0.8910788	36.76028	41.46338	39
Random Forest	0.8742860	39.48341	39.16074	39
GBM	0.8792211	38.700382	38.54347	39
Réseau de neurones	0.8663378	40.705265	41.8712	39
SVM	0.8517107	43.10318	44.51136	39
Ensembliste RF + GBM (metalearner : GLM)	0.9293785	29.60642	36.65624	39
Ensembliste 20 GBM	0.9098932	33.44711	39.09865	39

On voit donc que les modèles utilisant des arbres sont particulièrement performants. On notera la bonne performance de l'algorithme ensembliste des deux meilleurs modèles (GBM + RF), qui nous offre dans cette étude l'alternative la plus viable à notre projet. En effet ses performances sur le set d'entraînement et le set test donne de très bons résultats.

En conclusion, tous les algorithmes de machine learning réagissent bien à nos données, et tous offrent des performances supérieures aux modèles statistiques. Bien sûr, ces derniers ne sont pas optimaux dans le sens où toutes les combinaisons (simples ou multiples) de variables n'ont pas été testées, faute de temps. Mais dans un monde où tout doit toujours aller plus vite, les algorithmes de machine learning constituent un atout non négligeable dans la pratique de la modélisation.

IV.4.3 Combinaison statistique et machine learning

Dans cette partie nous allons adresser une alternative combinant deux méthodes vues dans la partie III. Il s'agit de combiner un arbre CART sur les résidus d'un modèle GLM. Ainsi, on ne perd aucune interprétabilité, puisque le GLM et l'arbre CART sont tous les deux facilement lisibles et intelligibles.

Cette alternative permet une maîtrise totale de ce qu'il se passe dans le processus de tarification. Les résultats finaux obtenus sur cette méthode sont :

	R^2 du modèle	RMSE Éch.Entr	RMSE Éch.Test	Nombre de variables
GLM_{II} + Decision Tree	0.9021598	32.74927	36.62424	29

Cette méthode semble donc fournir une alternative solide aux algorithmes 'black box' de la partie IV.4.2. Néanmoins cette méthode est sujette à une forte possibilité de surapprentissage comme en témoigne le large écart entre le RMSE de l'échantillon test et de l'échantillon d'entraînement.

IV.4.4 Limites et problématiques liées

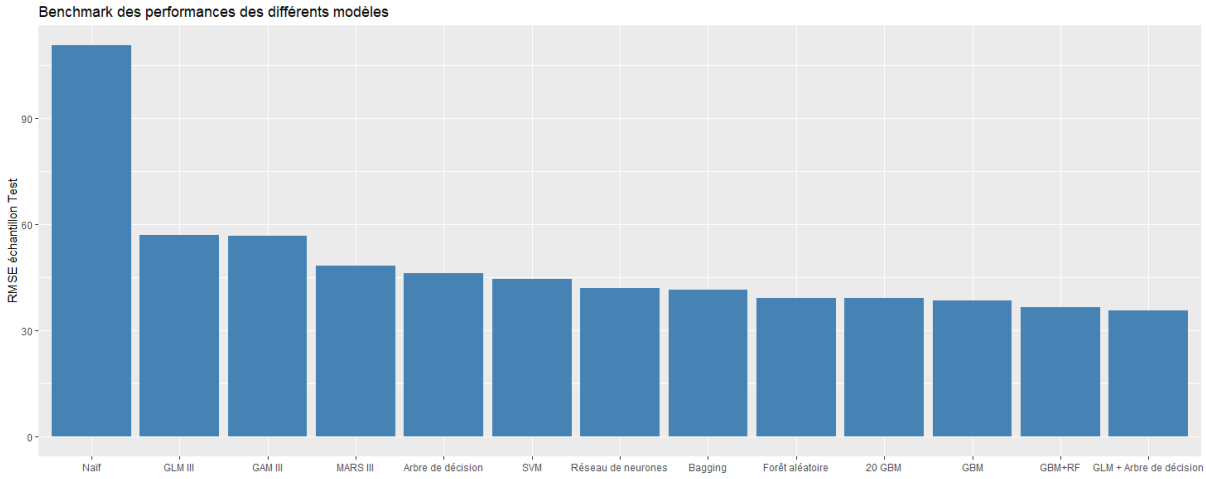


Figure 39 : Benchmark des performances (RMSE) des différents modèles

Comme l'a démontrée la partie IV.4.3, il est indiscutable que les modèles proposés par les techniques de machine learning sont plus performants que ceux plus classiques. Toutefois, on ne mentionnera jamais assez que cela implique de renoncer à de nombreux avantages des méthodes statistiques : interprétabilité, compréhension et traçabilité.

On se permet alors d'ouvrir une parenthèse. Dans la partie II, le choix avait été fait de seulement opérer sur la catégorie de dépréciation 'valeur agréée'. Il s'est avéré que le pourcentage appliqué détecté était assez similaire à ce qui était appliqué en réalité ; le développement de modèles 'personnels' pour chaque type de dépréciation n'apportait que peu de changement aux performances globales du modèle. Néanmoins, pour d'autres assureurs, il s'est avéré que la relation n'était pas aussi simple. Les modèles de machine learning se sont alors révélés beaucoup plus avantageux dans le sens où ils permettaient de prendre directement en compte cette particularité, et ne nécessitaient pas l'entraînement propre de trois modèles différents – un pour chaque type de dépréciation – là où les modélisations de type statistique nécessitaient explicitement la division du set de données d'entrées en 3 parties différentes.

Cette partie IV a donc également montré les limites des modèles linéaires généralisés, pourtant largement utilisés en actuariat. Il semble donc que nous sommes face à un dilemme : que privilégier entre interprétabilité et efficience ?

Cela crée donc de nouvelles problématiques directement liée à cette utilisation renforcée du machine learning. En effet, l'une des faiblesses du machine learning est son côté 'black box' qui le rend difficile à vérifier ; cela veut dire que notre modèle pourrait sembler particulièrement bon sur le traitement de nos données, mais en réalité se tromper complètement dans son raisonnement, et aboutir à des conclusions absurdes. Ce genre de comportement peut être décelé immédiatement dans des modèles plus classiques, comme les GLM.

En effet, rien qu'en étudiant le signe / l'amplitude d'un coefficient, on peut inférer ou non sur nos hypothèses.

Une autre question se pose également, mais cette question pourrait également se poser avec une moindre mesure pour des modèles plus classiques ; comment l'algorithme fait-il pour extrapoler sur des données non vues durant l'entraînement, et assez éloignées de celles-ci ? En effet, puisque les modèles de machine learning se veulent être des approximations universelles (ou tout du moins autant que possible), ils ont peu de restriction. Nous ne possédons donc un contrôle que très relatif de la fonction sur les régions sur dont nous avons peu ou pas de données. Suite à cela, il est difficile de s'assurer du comportement approprié de nos modèles dans les zones d'extrapolation, et encore plus de détecter avec efficacité les zones où l'algorithme réagit mal.

Dans une optique allant au-delà d'un simple benchmark, mais où l'on veut réellement pouvoir se servir des modèles acquis via cette étape de modélisation pour déterminer le positionnement plus global des assureurs contre lesquels nous sommes en compétition, notre modèle brut nous est inutile. En effet, à part si l'on utilise un simple arbre de prédiction (dont les résultats sont assez aléatoires et volatiles), il est difficile de déterminer l'impact des différentes variables sur la tarification finale.

C'est pourquoi nous nous intéresserons, dans la prochaine partie, à tenter d'apporter un peu de clarté dans ces eaux troubles.

V Pour aller plus loin : jugements, interprétation et limites

"I have always thought the action of men the best interpreters of their thoughts" - John Locke

A la vue des résultats présentés dans la partie IV.4, il est indéniable que les modèles de Machine Learning sont plus performants que les méthodes statistiques classiques. Ils sont rapides à mettre en oeuvre et ne nécessitent quasiment pas de traitement de la base de données en amont. Toutefois, comme il l'a souvent été mentionné tout au long de cet ouvrage, les modèles de Machine Learning ont ce gros défaut d'être quasiment des "black box".

Le terme quasiment est ainsi employé car au fil des années et des avancées de la recherche, ces méthodes ont tendance, sans néanmoins atteindre la transparence des modèles 'white box', à devenir de plus en plus intelligible. Ainsi, si au début des années 2000, ce qui se cachait à l'intérieur d'un modèle, hormis sa structure inhérente, était un véritable mystère, ce dernier tend à s'éclaircir avec l'apparition de nouvelles techniques de jugements et d'interprétation.

Même si nos modèles utilisant les algorithmes de machine learning semblent sur le papier bien meilleur à ceux fournis par les méthodes statistiques, il ne faut pas oublier que 'It is one thing to have a model with which one can make accurate predictions; it is another to have a model that makes accurate predictions for the right reasons", comme l'écrit Michael Weisberg dans son livre Simulation and Similarity. Les ordinateurs n'expliquent en effet généralement pas leur prédictions, ce qui est un véritable frein à leur utilisation de manière standardisée.

Il ne faut également pas perdre de vue que le Machine Learning apporte une vision stochastique et non déterministe des problèmes posés. Ainsi, toute contrainte physique n'a pas lieu d'être quand on parle d'un modèle de machine learning ; un réseau de neurones ne pourra donc pas comprendre qu'une aire ou une densité ne peut être négative, tout comme les phénomènes physiques qui nous entourent et constituent des règles inviolables.

La littérature est assez pauvre sur la manière de s'assurer du pouvoir prédictif d'un modèle de machine learning. Toutefois, ces dernières années, quelques chercheurs se sont penchés sur la question, de comment améliorer les solutions déjà existantes, souvent peu fiables et / ou trop gourmandes en temps de calcul.

C'est pourquoi nous allons dans cette partie nous attacher à comprendre les modèles développés dans la partie IV de ce mémoire. Tout d'abord, nous verrons comment nous pouvons construire des intervalles de confiance autour de notre prédiction, pour savoir à quel point notre modèle peut être fiable, et la marge d'erreur que nous avons lorsque nous l'utilisons.

Dans un second temps, nous nous attarderons sur la compréhension profonde de notre modèle : l'importance des variables, leurs interactions et leur répercussions sur notre variable réponse. Nous aborderons également une force des modèles de machine learning, qui leur permet d'obtenir une vision ciblée de l'évolution de la variable réponse à un endroit local.

Finalement, dans une optique d'ouverture sur les possibilités que laissent entrevoir les modèles de machine learning, nous tenterons de détecter des changements de tarifs qui peuvent se produire à un certain moment dans notre base de données. Car, à nouveau, nous n'avons aucune information exogène sur nos données, et nous devons nous assurer à terme, que le modèle que nous construisons est le plus performant.

V.1. Intervalles de confiance

Par ce qui précède, on voit bien que les algorithmes de Machine Learning sont bien plus performants que les méthodes statistiques traditionnelles. Toutefois, l'avantage des méthodes statistiques, est qu'il est possible d'obtenir des informations supplémentaires sur les estimateurs : biais éventuel, efficacité ($Var(\theta)$), intervalle de confiance…, puisqu'elles sont très encadrées mathématiquement (on sait comment le processus est réalisé puisqu'on est capable d'en répliquer toutes les étapes),

Prenons l'exemple du GLM : via ce modèle, on peut à la fois déterminer une erreur d'échantillonnage et une erreur paramétrique due à l'estimation des différents β_j. On obtient alors un intervalle de confiance sur la valeur finale rendue par le GLM ; cet intervalle est interprétable et fiable.

Pour les algorithmes de Machine Learning la problématique est toute autre. Nous n'obtenons pas par leur biais d'estimateurs à proprement parler, mais un modèle plus global. Même s'il est possible de récupérer via des graphiques d'importance les contributions de chaque variable au modèle, cela ne peut que donner une idée d'ensemble de comment le modèle fonctionne. Les méthodes traditionnelles comme le Théorème Central Limite (TCL) ne s'appliquent pas. On ne dispose donc généralement d'aucun moyen de savoir de manière directe à quel point notre modèle est efficace.

Il est en effet primordial de savoir si nos prédictions sont à 95% vraies dans une différence de tarifs de plus ou moins 40€, et non avec une erreur potentielle de 250€. Il est donc essentiel de pouvoir mesurer le niveau de certitude d'une prédiction.

On peut ainsi présumer de l'efficacité d'un modèle suivant la taille moyenne de l'intervalle de confiance ou de l'intervalle de prédiction (un grand intervalle signifiant que l'algorithme a eu du mal à associer une valeur aux données ; le modèle est donc peu efficace).

Dans cette partie, nous essayerons donc de trouver un compromis entre les différents types de confiance pouvant être appliquée à un modèle pour pouvoir parvenir à une adéquation entre partie statistique et partie machine learning.

Il n'existe pas de manière explicite d'obtenir des intervalles de confiance en machine learning. C'est pourquoi on s'intéressera par la suite aux intervalles de prédictions. En effet, puisque dans ces modèles il nous est impossible d'obtenir directement l'effet de telle ou telle variable, la seule valeur de confiance qui nous intéresse devient à quel point notre prédiction est pertinente.

Les intervalles de confiance différent des intervalles de prédiction dans le sens où l'intervalle de confiance donne la justesse de l'estimation d'un paramètre, comme une moyenne ou un coefficient de régression. La taille de l'intervalle de confiance dépend de l'écart-type ; plus la taille de l'échantillon est importante, plus la moyenne convergera vers la moyenne de la population car l'écart-type tendra vers 0.

Les intervalles de prédictions sont différents dans le sens où ils annoncent où l'on peut s'attendre à voir le prochain point échantillonné ; pour un intervalle de prédiction à 95%, on s'attend à ce que la valeur prédite tombe 95% du temps dans l'intervalle de prédiction.

Les intervalles de prédiction prennent en compte à la fois l'incertitude dans l'estimation de la moyenne de la population ainsi que la variation aléatoire des valeurs individuelles. Un intervalle de prédiction est donc toujours plus large qu'un intervalle de confiance. Les intervalles de prédiction

témoignent de la distribution des valeurs individuelles, ce qui permet d'obtenir des intervalles de prédiction différents selon les valeurs entrées en début d'algorithme.

Mathématiquement, un intervalle de confiance donne une gamme de valeurs pour $E[y|x]$ alors qu'un intervalle de prédiction donne une gamme de valeurs pour y lui-même.

V.1.1 Partie Statistique : intervalles sur paramètres et sur prédictions

De manière générale, les intervalles d'erreur sont produits en ajoutant / soustrayant à la valeur fittée un certain nombre d'écart-type. Découvert en 1718 par Abraham de Moivre, l'écart-type sera ensuite utilisé par Karl Pearson dès 1893. Défini comme la moyenne quadratique des écarts par rapport à la moyenne, l'écart-type est homogène à la variable mesurée et est un indicateur de dispersion. Il mesure ainsi la variabilité d'une série statistique.

Dans les domaines scientifiques, on fait fréquemment l'hypothèse que les mesures d'une grandeur se répartissent selon une distribution gaussienne, par accumulation d'erreurs de mesure ou d'interférences indépendantes avec d'autres phénomènes, en application du théorème central limite.

L'histogramme des valeurs observées se rapproche alors d'une courbe en cloche caractéristique de la loi normale ; la courbe étant complètement définie par la donnée de la valeur moyenne et de l'écart-type, on peut donc définir un intervalle de fluctuation concentrant l'essentiel des observations.

On a donc dans un intervalle de taille 2σ 68,27% des valeurs, dans un intervalle de taille 4σ 95,43% des valeurs, et dans un intervalle de taille 6σ 99,73% des valeurs.

On cherchera par la suite à obtenir un intervalle de confiance à 95% ; si la valeur réelle n'est pas dans cet intervalle, alors la probabilité a priori du résultat de l'observation que l'on a obtenue était inférieure à 5%.

V.1.1.1. Intervalle de confiance sur les paramètres

Il existe différentes manières d'obtenir des intervalles de confiance sur les paramètres. Obtenir un estimateur de confiance sur les paramètres permet, non seulement de savoir si l'utilisation de la variable dans le modèle est pertinente. Si l'intervalle de confiance comprend 0 ou est trop large, alors cela signifie que le modèle n'a pas réussi à trouver la réelle contribution de cette variable dans l'explication de la variable réponse.

Nous exposerons ici les deux principales méthodes : la méthode de Wald ainsi que la méthode du profil de vraisemblance (Profile Likelihood method).

La méthode de Wald présente l'avantage de se calculer sur un seul modèle (on n'a pas à calculer de second modèle où la contrainte est relâchée, c'est-à-dire où on met notre estimateur à 0). La détermination de l'intervalle de confiance est simple et s'obtient via la méthode décrite précédemment, où l'on ajoute/soustrait un certain nombre d'écart-type à la valeur du paramètre. On prend toutefois les percentiles de la distribution t-Student lorsqu'il s'agit de coefficients de régression.

Cette méthode se base sur une hypothèse de normalité assez forte et peut se montrer peu robuste si la distribution du paramètre est nettement asymétrique ou si l'erreur standard est une mauvaise estimation de l'écart-type de l'estimateur. Les erreurs standards étant basées, dans les GLM, sur les variances asymptotiques obtenues via la matrice d'information de Fischer, la méthode de Wald réalise une mauvaise performance sur les échantillons petits ou de taille réduite.

Un des effets moins connus de l'utilisation du test de Wald est l'effet Hauck-Donner ; une statistique de test de Wald n'augmente pas de façon monotone en fonction de l'augmentation de la distance entre l'estimation du paramètre et la valeur nulle. Il en résulte une valeur de p (p-value) biaisée puisque surestimée; cela amène également une perte de puissance de test. Tous ces faits peuvent résulter en des conséquences très dommageables, notamment dans la sélection de variables.

Une alternative se trouve dans l'utilisation des intervalles de confiance de profils. Ceux-ci, plus fiables que les intervalles de Wald, ne supposent pas que l'on connaît l'erreur standard. Dans cette méthode, on réalise pourtant toujours une approximation asymptotique, mais cette fois-ci on utilise la distribution asymptotique du Chi-Deux de ratio de la statistique de log-vraisemblance.

L'idée derrière cette méthode, est qu'une fois le profil de maximum de vraisemblance obtenu (chaque point de cette fonction est la valeur maximisée de la fonction de vraisemblance, pour un profil fixé, c'est à dire pour un paramètre βi pouvant prendre comme valeur θ), on a une suite de points :

$$L_i(\beta_i) = max(\delta)\, L(\beta_i, \delta)$$

Où $\delta = \beta_{-i} = (\beta_1, \beta_2, \ldots, \beta_{i-1}, \beta_{i+1}, \ldots, \beta_n)$, avec $L(\theta, \delta)$ la fonction de maximum de vraisemblance. Avec cette fonction il est facile d'obtenir un intervalle de confiance.

En effet, un intervalle de confiance pour β_i est l'ensemble des valeurs β_{i_0} telles qu'un test bilatéral de l'hypothèse nulle $H_0 : \beta_i = \beta_{i_0}$ ne serait pas rejetée au niveau de significativité 5%. Or on peut calculer facilement cette statistique via la test de ratio statistique qui est égal à :

$2\,[log(\hat{L}, \hat{\delta}) - log(L_{\theta_0}, \widehat{\delta_0})] < \chi^2_{95\%}$ puisque $log(L(\beta_{i_0}, \widehat{\delta_0}) = log\,(L_i(\beta_{i_0}))$

Cette méthode présente aussi l'avantage que les intervalles produits sont invariants par transformation monotone d'un paramètre (si $[c_1, c_2]$ est un intervalle pour θ alors $[g(c_1), g(c_2)]$ est un intervalle pour $g(\theta)$).

Le test du multiplicateur de Lagrange peut également être utilisé pour obtenir des intervalles de confiance sur les paramètres, mais ne sera pas développé ici.

Dans notre cas, seuls les GLM sont sujets à ces intervalles de confiance. Puisque cela ne peut pas en faire une mesure globalement utilisée, nous nous contenterons de mettre les résultats de l'analyse des paramètres avec ces deux méthodes dans l'annexe A.5.

V.1.1.2. Intervalle de confiance sur la prédiction

Nous allons maintenant nous intéresser à développer un intervalle de confiance sur notre prédiction. La technique des écarts-types présentée en amont, et qui s'applique très bien sur les estimateurs β_j ne fonctionne pas dans le cas du GLM ; elle ne marche que pour les modèles linéaires. Dans le cas d'un modèle de Poisson où l'espérance serait de 0, la variance ne peut être de 0 (la variance est égale à l'espérance pour une loi de Poisson), puisque le modèle ne retournera jamais des valeurs fittées exactement égales à 0; il y a donc une certaine incertitude sur la valeur retournée.

Il faut donc créer l'intervalle sur l'échelle de la fonction lien et non sur l'échelle de la réponse. En effet, lorsque la variable réponse n'a pas encore été retransformée pour revenir à son échelle originelle, le modèle est considéré comme un modèle linéaire. On estime alors que le modèle est approximativement Gaussien (si l'échantillon est de taille suffisamment importante). On peut alors utiliser la méthode d'ajout/soustraction de l'écart-type sur la réponse non-retransformée. En appliquant l'inverse de la fonction lien on peut alors obtenir un intervalle de confiance sur les valeurs fittées.

On obtient alors pour notre modèle des intervalles de prédiction de largeur 260 € (+ 130€ / -130€ par rapport à la valeur prédite). Ces intervalles, en plus d'être larges et de ce fait, de ne pas fournir de manière 'sûre' la bonne valeur, ont le défaut d'être homogènes, c'est-à-dire qu'ils ne dépendent ni de la valeur à prédire, ni des caractéristiques du client. Pourtant, comme le montre le graphique ci-dessous, le modèle a particulièrement du mal à prédire les larges primes, c'est-à-dire les valeurs au-dessus de 400€ ; dans ces cas-là, le modèle tend à minimiser le tarif proposé. Il s'agit donc d'un problème à pallier.

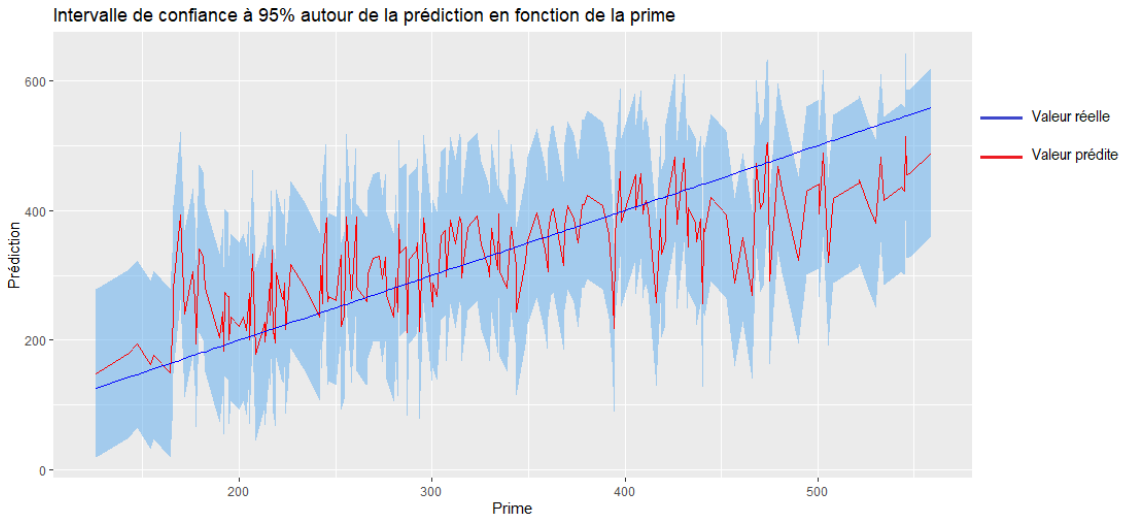


Figure 40: Intervalle de prédictionà 95% en fonction de la prime

V.1.2 Autres méthodes communes : quantiles, bootstrap et clustering

Comme nous l'avons vu dans la partie précédente, la méthode standard de création d'intervalles nécessite l'hypothèse forte que la variable réponse suive une loi normale. Or comme nous l'avions entrevu dans la partie II.2.3., ce n'est pas vraiment le cas pour notre variable réponse.

Dans ce cas, il faut trouver des alternatives à la méthode statistique classique. On verra ici trois méthodes différentes. Une, le bootstrap, est souvent utilisée pour garantir la robustesse d'un modèle. Les deux autres sont rarement utilisées, du fait qu'elles sont peu robustes, ou tout simplement car elles sont difficiles à mettre en place.

Nous noterons toutefois que ces trois méthodes peuvent s'appliquer aussi bien à des modèles statistiques qu'à des modèles de machine learning.

V.1.2.1. Méthode des quantiles

L'une des premières idées est de s'intéresser aux quantiles du modèle, de manière à créer des intervalles de prédiction. Le but étant d'estimer les quantiles conditionnels de la variable réponse, et non la moyenne conditionnelle comme le permet la méthode des moindres carrés.

L'utilisation des quantiles pour construire un intervalle de prédiction est assez simple : en combinant deux quantiles, il est possible d'obtenir un intervalle borné par les deux sets de prédiction produits par les deux modèles. Un quantile de niveau q représentant la valeur x_q telle que $P(X \leq x_q) = q$.

Ainsi, si l'on utilise le quantile 97.5% combiné avec le quantile 2.5%, on obtient théoriquement une probabilité à 95% que la valeur appartienne à l'intervalle.

Même si elle est très simple à implémenter, la méthode de régression des quantiles porte quelques défauts. L'un d'entre eux est que, lorsque les quantiles sont extrêmes (très grands ou très petits), les intervalles ont tendance à être très grand, les bornes étant assez éloignée des vraie valeurs. Cela ne veut pas dire que les quantiles calculés sont mauvais ; juste que l'essence même de la construction de l'intervalle perd son sens.

L'un des problèmes majeur de cette méthode et que nous avons rencontré, est que chaque quantile nécessite sa propre régression. Cela veut dire que la valeur associée au quantile 97.5% est parfois plus petite que celle associée au quantile 2.5%. En effet, dans les modèles trop fragiles (peu de variables, peu de données…) les ajustements peuvent être hautement volatiles, c'est pourquoi il n'est pas surprenant d'observer de tels résultats.

On a donc appliqué un tel intervalle sur notre modèle GBM trouvé en IV.4.2. Ci-dessous, on peut ainsi retrouver les caractéristiques de l'intervalle de confiance, ainsi que le nombre de données se trouvant finalement dans l'intervalle.

Critère	Quantile	Quantile ordonnés
Taille moyenne de l'intervalle	90.20	90.4390
Taille médian de l'intervalle	89.30	89.2999
Taille maximale de l'intervalle	348.75	348.7458
% de valeurs appartenant à l'intervalle	71.38427	71.68196

On voit ainsi que la valeur de 95% des données contenues dans l'intervalle est loin d'être observée, alors que nous travaillons tout de même sur un échantillon de grande taille (8062 observations dans notre dataset test). On remarque aussi que, même si les tailles moyennes et médianes des intervalles sont correctes, dans certains cas, les intervalles sont très larges (au moins 25% des observations ont un intervalle de taille > 120€) et les mesures associées ont donc une pertinence moindre. En effet, peut-on réellement croire un modèle si l'on sait que sa valeur peut varier du simple au double ?

V.1.2.2. Bootstrap

Une autre méthode d'obtention d'un intervalle de prédiction se trouve dans les méthodes d'inférence statistique. Basé sur la réplication multiple des données à partir du jeu de données étudiée, le bootstrap, successeur du "jackknife", s'est développé dans les années 70 avec l'augmentation des capacités de calcul des ordinateurs. Le bootstrap consiste à estimer une caractéristique d'une population (comme sa moyenne) en sous-échantillonnant de manière répétée l'échantillon de base. Ainsi, on obtient des estimations en s'affranchissant d'utiliser les distributions de probabilité théorique. Le lecteur attentif pourra faire le parallèle entre cette utilisation et celle qui a été réalisée dans la partie IV.3.2.2 lors de l'introduction du bagging et de l'algorithme de forêt aléatoire.

Le bootstrap repose sur des simulations stochastiques, comme les méthodes de Monte-Carlo, les méthodes numériques bayésiennes (échantillonneur de Gibbs, algorithme de Metropolis-Hastings), à la différence près que le bootstrap ne nécessite pas d'autre information que celle disponible sur les individus de l'échantillon originel.

Le bootstrap tire son nom de son principe ; en anglais le bootstrap désigne le fait de "se hisser en tirant sur ses propres lacets, ou plus précisément sur ses "bootstraps"", les bootstraps étant les anneaux de cuir ou tissus cousus au rebord des bottes pour y passer les doigts afin de les enfiler plus facilement. Cela fait écho au caractère itératif et autocentré : le bootstrap crée des "nouveaux échantillons" statistiques par tirage avec remise, à partir de l'échantillon initial.

Plus formellement, le bootstrap traite l'inférence de la vraie distribution de probabilité P, étant donné l'échantillon originel, comme étant analogue à l'inférence de la distribution empirique $\hat{P}$, étant donné un sous-échantillon. L'exactitude des inférences concernant $\hat{P}$ peut être évaluée car $\hat{P}$ est connue. Si $\hat{P}$ est une approximation raisonnable de P, alors la qualité de l'inférence sur P peut à son tour être déduite.

L'intervalle de prédiction déterminé par le bootstrap est assez similaire à la méthode des quantiles. En effet, on identifie à nouveau les quantiles, mais cette fois-ci on s'intéresse aux quantiles de la distribution bootstrap et non aux quantiles du modèle. On reprend à nouveau les quantiles d'ordre 97.5% et d'ordre 2.5% pour obtenir un intervalle à 95%.

Pour obtenir un intervalle de prédiction bootstrap, il faut réaliser un grand nombre de simulations (généralement de l'ordre du millier). Or c'est là que le bât blesse. Les modèles prennent en effet à eux seuls un certain temps à se calculer. Ici, pour obtenir une estimation fiable, il faut en réaliser un grand nombre, ce qui peut, en pratique, être inconvenant, voire bloquant.

De plus, cela implique de garder en mémoire toute la distribution bootstrap pour obtenir pour chaque prédiction un intervalle adapté : pour une distribution de 1000 modèles, il faut donc créer 1001 modèles (le modèle originel plus la distribution bootstrap) et faire tourner ces 1001 modèles sur chaque prédiction. Cela est très demandeur, à la fois en terme de temps et de mémoire. C'est pourquoi dans la pratique, on cherche souvent des alternatives au bootstrap ; dans notre étude, nous avons renoncé à cette technique qui aurait nécessité des heures de run de notre machine.

V.1.2.3. Clustering

Une autre piste pour obtenir un intervalle de prédiction pour contrôler l'efficacité du modèle fut d'essayer de regrouper les profils par similarités, de manière à obtenir des groupes homogènes quant à la taille de l'intervalle de prédiction.

L'idée étant que, par exemple, pour une certaine catégorie de la population comme les gens vivant à Liège et ayant une voiture avec une puissance supérieure à 150 chevaux, on pouvait juger la prédiction comme fiable car l'intervalle de prédiction était particulièrement petit.

Toutefois, la base de données comprenait 30 variables, une fois le pré-tri de la partie II effectué. Malgré l'implémentation de différentes méthodes de clustering (k-means, DBSCAN ou les méthodes de regroupement hiérarchiques), la base de données s'est montrée inadaptée à un traitement de ce genre. Ce n'est pas en soi aberrant, dans le sens où la plupart de ces algorithmes fonctionnent bien en petites dimensions ; ici, il y avait tout simplement trop de segmentations possibles pour parvenir à une issue pertinente.

La plupart des algorithmes ne formaient pas des clusters contenant plus d'une dizaine de lignes ; lorsque le nombre de clusters était imposé, ils étaient très hétérogènes et donc totalement inutiles.

C'est pourquoi cette piste a été abandonnée, faute de fournir des résultats concluants.

V.1.3 Prédictions conformes

La dernière méthode présentée ici, a été développée par G. Shafer, V.Vovk et A.Gammerman [2005]. H.Papadopoulos l'a ensuite adaptée à différents algorithmes de machine learning [2006]. Le concept repose sur le fait que nous attendons des performances passées qu'elles aiguillent les performances futures.

Ainsi, la plupart du temps, on estime que si le modèle est 71% fiable sur l'échantillon test, alors il est également 71% fiable sur de nouvelles données. Mais à quel point cela est-il vrai?

Avec les prédictions conformes (conformal predictions), on obtient des probabilités bien calibrées (80% signifie 80%). De plus, il n'y a pas besoin de connaître les probabilités à priori : on suppose seulement que les données sont interchangeables (donc indépendantes et identiquement distribuées, et l'ordre des observations n'est pas pertinent).

Dans un cadre global, considérons une distribution Z : X x Y et une fonction f : Z -> R . On sélectionne 4 données aléatoires de Z et on leur applique f : on obtient alors 4 valeurs réelles $\alpha_1, \alpha_2, \alpha_3$ et α_4. Sans perte de généralité, on peut considérer $\alpha_1 \leq \alpha_2 \leq \alpha_3 \leq \alpha_4$. De manière statistique, on a alors :

20%	20%	20%	20%	20%
α_1	α_2	α_3	α_4	
$< \alpha_1$	$[\alpha_1; \alpha_2]$	$[\alpha_2; \alpha_3]$	$[\alpha_3; \alpha_4]$	$> \alpha_4$

En effet, étant donné que l'on a supposé toutes les données interchangeables, si on pioche une nouvelle donnée de Z et qu'on lui applique f, alors on peut estimer la distribution des scores, relativement à $\alpha_1, \alpha_2, \alpha_3$ et α_4.

Ainsi $P[f(z) \leq \alpha_3] = 0.6$ et $P[f(z) \leq \alpha_4] = 0.8$ par exemple.

Maintenant, on pose $f(z_i) = |y_i - h(x_i)|$, où i représente la i-ème observation, yi la prime associée (la valeur réponse) et xi les caractéristiques de l'assuré i. h étant la fonction liée à l'algorithme de machine learning déterminé en partie IV.

On a toujours $P[|y_i - h(x_i)| \leq \alpha_3] = 0.6$ et $P[|y_i - h(x_i)| \leq \alpha_4] = 0.8$.

Or $\alpha_1, \alpha_2, \alpha_3$ et α_4 sont des valeurs connues, puisque l'on dispose de la fonctoin h donc :

$$P[|y_i - h(x_i)| \leq \alpha_3] = 0.6 \Leftrightarrow P[y_i \in h(x_i) \pm \alpha_3] = 0.6$$

Ainsi, on obtient un intervalle de confiance pour l'observation i, de variable réponse y_i.

La fonction $f(z)$ est appelée fonction de non-conformité; elle mesure "l'étrangeté" d'un schéma (x_i, y_i).

Dans le cas de la régression, on commence par partitionner Z (l'ensemble des données) en deux data sets : un dataset propre à l'entraînement et un dataset dit de calibration, de taille q. On utilise ensuite le dataset d'entraînement pour obtenir un modèle (voir IV) ; ce modèle sera appelé modèle sous-jacent.

On pose ensuite $f(z_i) = |y_i - h(x_i)|$ et on applique $f(z)$ à toutes les lignes de la base de données de calibration On trie ensuite les valeurs du dataset de calibration par valeur décroissante de $f(z_i)$. On obtient une suite de réels α_j tels que $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_q$.

Pour un niveau de prédiction à 95%, on pose $s = [5\% \times (q+1)]$; s est la position du 95%-percentile du score de non-conformité α_s.

L'intervalle de prédiction à 95% pour une donnée non-vue est alors :

$$\Gamma_i^{95\%} = h(x_i) \pm \alpha_s$$

Cet intervalle contient y_i avec 95% de probabilité.

Néanmoins cette démarche suppose que chaque intervalle de prédiction a la même taille, α_s. Pour avoir des intervalles plus individuels, il faut continuer la démarche en normalisant la fonction de non-conformité.

$f(z_i)$ devient alors $f(z_i) = \frac{|y_i - h(x_i)|}{\sigma_i}$. σ_i modélise la difficulté à prédire y_i.

Pour trouver σ_i on le décompose en deux parties :

$$\sigma_i = g(x_i) + \beta$$

β est un paramètre de sensibilité qui détermine l'impact de la normalisation. La fonction g est elle obtenue en essayant d'expliquer les résidus de $h(x_i)$ à l'aide des x_i. Pour cela on applique un nouveau modèle de machine learning sur nos résidus issus du set de training. Dans notre cas on a :

$$f(z_i) = \frac{Erreur\ Absolue}{Erreur\ Absolue\ Prédite} = \frac{|Valeur\ réelle - prédiction|}{g(x_i) + \beta}$$

On applique ensuite cette nouvelle fonction f au set de calibration, que l'on trie à nouveau par ordre décroissant de $f(z_i)$. Comme précédemment, on pose $s = [5\% \times (q+1)]$.

L'intervalle de prédiction à 95% pour une donnée non-vue est alors :

$$\Gamma_i^{95\%} = h(x_i) \pm \alpha_s \times (g(x_i) + \beta)$$

Cette méthode se révèle la plus fiable et la plus performante dans un temps restreint. Elle ne nécessite que le calcul de 2 algorithmes de machine learning. Elle est valide puisque par construction, elle respecte une cohérence entre la définition entre l'erreur et le niveau de confiance. Son efficacité est liée au modèle sous-jacent.

Ci-dessous un tableau présentant les caractéristiques de la distribution de la taille des intervalles de prédiction selon la méthode, par rapport au modèle GBM utilisé en IV.4.2.

	Quantile à 25%	Médiane	Moyenne	Quantile à 75%	Min	Max	Valeurs réelles dans l'intervalle
Quantile ordonnés	55.5067	89.2999	90.4390	121.9022	0.0033	348.74	71.681%
Conformal	19.82	26.62	29.48	35.26	0.80	275.78	92.769%

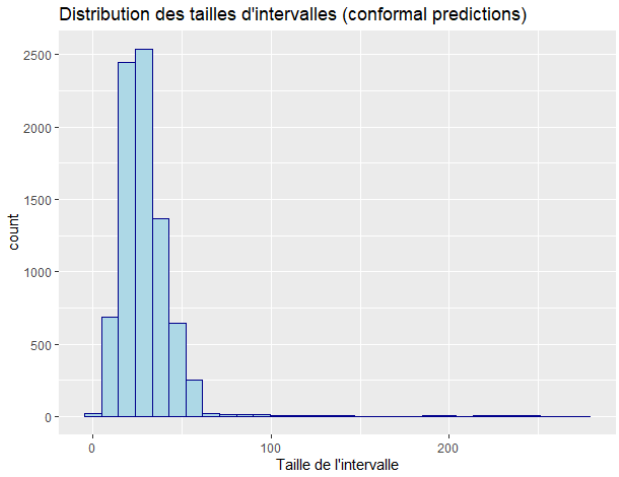


Figure 41 : Distribution de la taille des intervalles, issus de la méthode conformal predictions

Ci-dessous, la mise en œuvre de cet intervalle de prédiction dans la tarification via plusieurs assureurs, d'un profil quelconque en assurance à valeur agréée +. On voit que cela permet de donner une vue plus complète de la qualité de prédiction, surtout lorsque l'on cherche à interpréter cette dernière pour établir un ranking entre assureurs. Il est alors difficile d'attester avec certitude si l'assureur 7 est plus cher que le 6, alors que c'est ce que l'on conclurait sans l'intervention de ces intervalles de confiance.

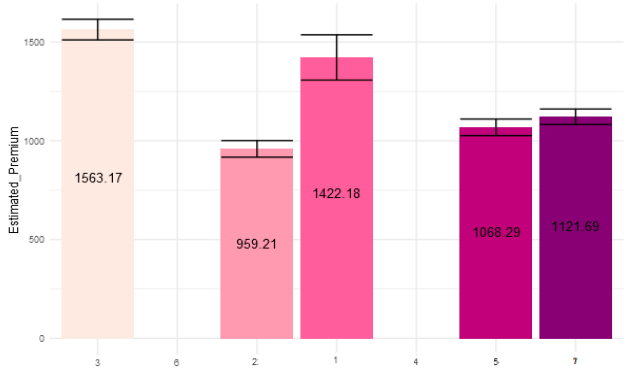


Figure 42: Tarification d'un assuré, avec intervalles de confaince autour des valeurs

V.2. Interprétation et confiance dans le modèle

Alors que les modèles de machine learning pénètrent des secteurs de plus en plus complexes et sérieux comme la médecine, la justice (jugement par algorithme) et les marchés financiers, l'incapacité de l'homme à comprendre ces modèles apparaît problématique. Malgré l'absence d'une définition claire de ce qu'est réellement l'interprétabilité, de nombreux articles font régulièrement appel à ce concept. Toutefois sa portée n'est pas toujours la même ; il s'agit parfois de regarder l'interprétabilité comme une manière d'engendrer la confiance, d'autres fois comme une connexion entre un modèle interprétable et un modèle qui découvre la structure causale dans les données.

S'il est facile de comprendre pourquoi le machine learning a eu un impact profond sur les sciences, il est plus compliqué d'appréhender ses capacités, et encore plus important, ses limitations. Yuval Noah Harari a alors proposé le terme 'dataïsme' pour qualifier la nouvelle étape supposée de la civilisation dans laquelle nous entrons, et où nous faisons davantage confiance aux algorithmes et aux données qu'à notre propre jugement et logique.

Mais à nouveau, le concept de confiance est peu clair ; cela fait-il appel à la performance, à la robustesse, ou à d'autres des propriétés de ce modèle ? D'où tirons-nous exactement cette confiance?

Comme il l'a déjà été mentionné plus tôt dans ce travail, l'un des freins majeurs à l'abandon des modèles statistiques au profit des modèles dits de machine learning, est la difficulté à les interpréter. Cela revient à échanger un modèle 'white box' contre un modèle 'black box', dont on n'est pas vraiment certain d'obtenir les bons résultats pour les bonnes raisons.

Même si on peut aisément comprendre le fonctionnement d'un arbre de décision et le classement qu'il opère, cela devient plus complexe dès lors que l'on passe à des modèles plus évolués. Spécifiquement, dans le monde de l'actuariat, les modèles doivent être communiqués, compris et approuvés par un public non expert en statistiques et travaillant quelques fois dans d'autres domaines.

Pendant longtemps, les chercheurs et utilisateurs de ces techniques se sont demandés où devait être placé le curseur entre interprétabilité et justesse. En effet, plus le modèle est complexe, plus il y a de chances qu'il parvienne à en capturer toute les spécificités.

Il n'y a pas de définition mathématique de l'interprétabilité. Tim Miller (2017) l'a défini comme 'le degré auquel un humain peut comprendre la cause d'une décision' dans son papier 'Explanation in artificial intelligence: insights from the social sciences'. Cette définition correspond bien à ce que l'on va s'attacher à faire dans la suite de ce mémoire.

Interpréter un modèle GLM avec un degré d'interaction bas (1 ou 2) est simple : il suffit de regarder le coefficient β_i associé à la variable ou à l'interaction i qui nous intéresse. On obtient ainsi l'effet qu'à cette variable lorsqu'elle 'active' $\beta_i$si on parle de variables catégorielles, ou en multipliant sa valeur avec β_i dans le cas d'une variable numérique. Même avec un modèle GAM il reste simple d'obtenir l'impact d'une variable.

Toutefois, cette notion d'interprétabilité doit être prise avec des pincettes et ne doit pas être acceptée aveuglément. En effet, même si l'interprétation des poids peut sembler intuitive, leur valeur peut se révéler fragile selon la manière dont les variables ont été pré processées. Par exemple, la relation entre prime payée et âge du conducteur peut se révéler positive ou négative selon que notre modèle inclue ou non les variables valeur assurée ou âge du plus jeune conducteur.

V.2.1 Effet des variables

On va donc premièrement s'intéresser à l'effet des variables. Cette démarche est liée à celle effectuée en III, même si cette fois-ci on s'intéresse de prime abord aux variables dont l'importance est forte, et non à celles dont l'impact est quasiment négligeable. Tout d'abord représentons nous les variables présentes dans notre modèle GLM. On peut visualiser facilement l'effet de chaque variable avec un 'weight' plot ou un 'effect' plot.

Le weight plot permet de représenter les poids $\beta_j$avec leur variance, formant ainsi un intervalle Γ_j. Si Γ_j comprend la valeur 0, alors la variable X_j n'est pas significative, car sa valeur oscille autour de 0, et peut être aussi bien négative que positive.

Toutefois l'un des défauts des weight plots est que les valeurs des variables sont mesurées via des échelles différentes. C'est là qu'intervient l'effect plot ; en multipliant le poids par la valeur de la variable on peut avoir l'effet global de la variable sur la prédiction. On a $effect_j^{(i)} = w_j x_i j$.

On peut visualiser ces effets avec différents types de diagramme. On peut voir ci-dessous un weight plot obtenu sur notre modèle final :

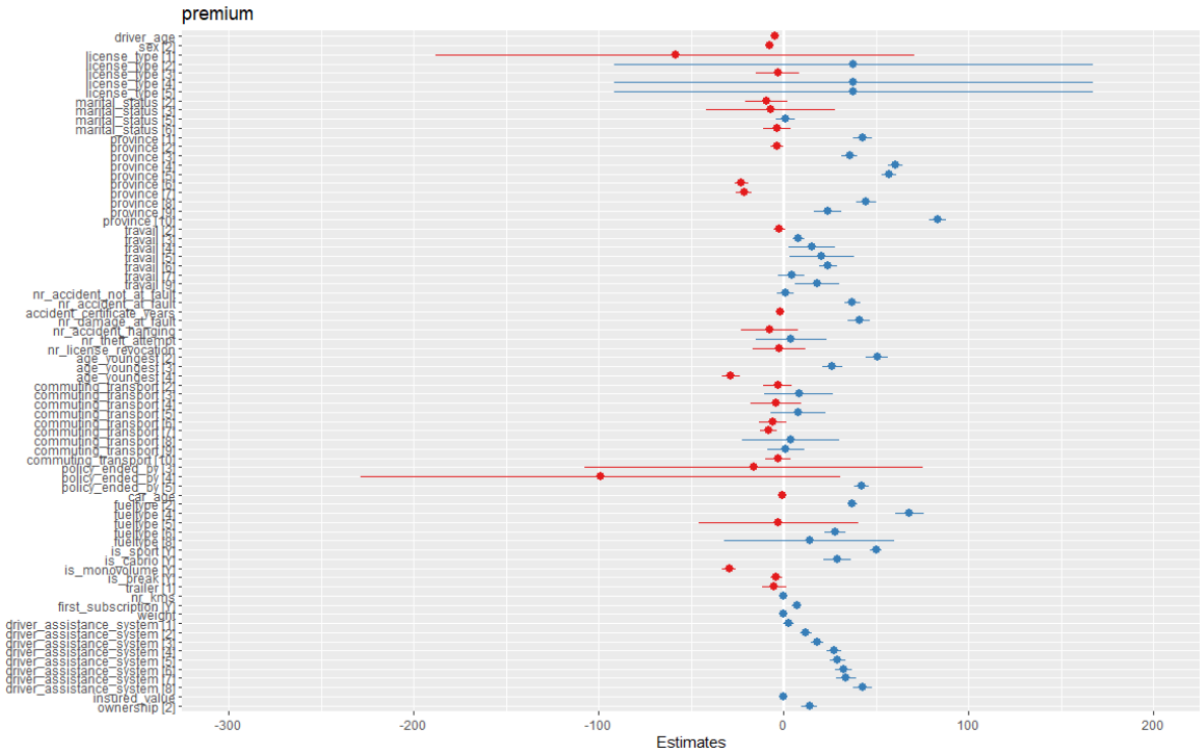


Figure 43 : Effect plot des variables et de leur modalié (GLM)

On peut également obtenir un graphe d'importance des variables pour nos modèles de machine learning. Pour les modèles se basant sur les arbres de régression, l'importance des variables est déterminée en calculant leur influence relative ; c'est-à-dire si la variable a été sélectionnée comme variable segmentante, et si elle l'a été, à quel point l'utilisation de cette variable a contribué à améliorer le résultat (en diminuant l'erreur au carré du modèle, c'est-à-dire en diminuant l'hétérogénéité des feuilles). Pour les modèles neuronaux, la méthode Gédéon est implémentée ; cette méthode ne se base toutefois que sur les deux premières couches neuronales.

L'importance des variables ne suffit généralement pas à conclure quant aux raisons expliquant une prédiction. En effet, un événement peut souvent s'expliquer par des causes multiples : on appelle cela l'effet Rashomon. Rashomon est, à la base, un film japonais d'Akira Kurosawa, racontant quatres versions différentes et contradictoires de témoins oculaires, à propos de la mort d'un samouraï et traduit le phénomène psychologique qui veut qu'un même événement peut avoir plusieurs causes.

Cela s'avère être un avantage en machine learning car les méthodes ensemblistes, c'est-à-dire qui combinent plusieurs modèles, sont en général plus performantes car en moyennant ces différentes 'explications' elles rendent la prédiction plus robuste et précise. Mais cela rend encore plus compliqué de comprendre pourquoi une certaine prédiction a été faite.

Avec un arbre CART, on peut obtenir l'importance globale d'une variable. Elle se calcule de la manière suivante : on parcourt tous les noeuds pour lesquels la variable a été utilisée et on mesure de combien la variance a baissé suite à cette utilisation. La somme de toutes ces importances est ensuite rééchelonnée sur 100 et on obtient alors une contribution en parts de chaque variable.

Pour les modèles utilisant les arbres CART (Random Forest, GBM, bagging), c'est cette méthode que nous avons utilisée. Pour le réseau de neurones, l'importance des variables se traduit immédiatement via les poids qui leur sont accordés.

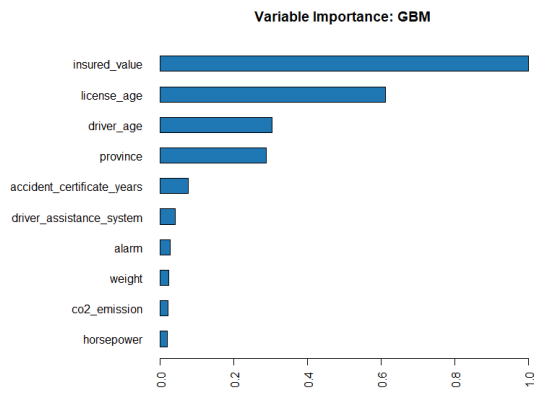


Figure 44: Variable importance par GBM

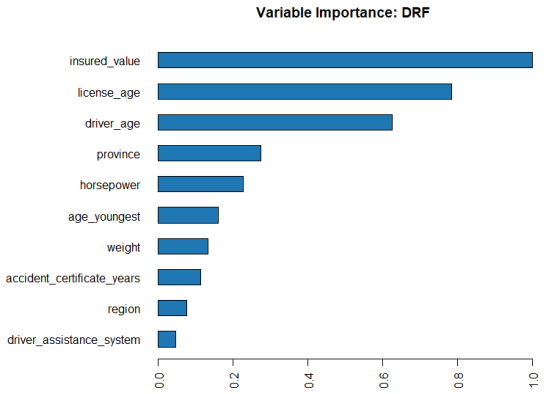


Figure 45 : Variable importance par forêt aléatoire

On a donc à travers ces graphiques une idée des variables qui ont le plus d'importance dans le modèle. Si les conclusions reflétées par ces graphiques sont peu cohérentes avec la théorie, alors le modèle doit être remis en question.

V.2.2 Détection Outlier / Observations influentes

Nous avons mentionné dans la partie II.1.2 un phénomène qui pouvait survenir assez fréquemment dans nos données : les observations aberrantes. Nous avions à l'époque abordé le fait que certaines de ces observations pouvaient influer négativement sur notre modèle. La résistance aux outliers est aussi un phénomène lié à la robustesse du modèle. Ceci avait été également abordé plus tôt. Dans notre processus de sélection de modèle, nous ne sommes pas intéressés outre mesure à ce phénomène pourtant important. C'est pourquoi il a été jugé nécessaire d'introduire cette partie permettant de mettre en lumière différentes méthodes permettant de soulever le problème d'utilisation d'un modèle peu robuste.

On fait en général une distinction entre valeur aberrante et outlier. Le traitement des outliers repose lui-même sur une théorie que nous n'aborderons qu'en surface. En effet, dans certains cas, il est bon de garder un outlier car il représente une donnée possible et n'affecte pas le pouvoir du prédiction du modèle.

Prenons pour exemple une détection visuelle d'observations influentes sur un bête modèle de régression linéaire :

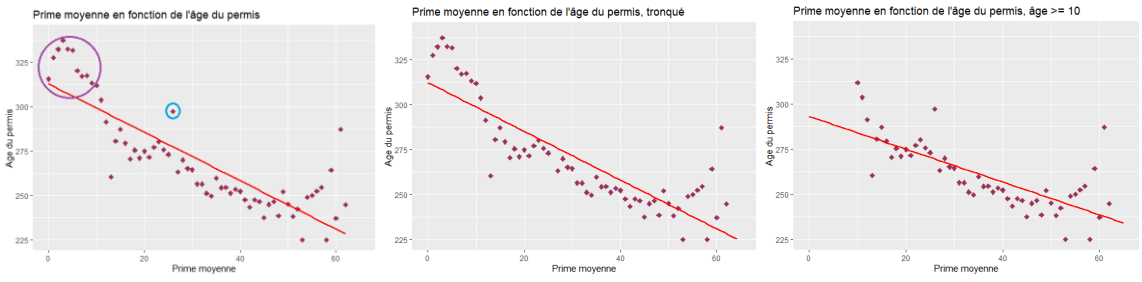


Figure 46 : Prime moyenne en fontion de l'âge du permis selon différents sets de training

On peut voir sur les graphes ci-dessus des régressions linéaires de l'âge du permis sur la prime moyenne. Le premier graphe contient toutes les observations. Sur le deuxième graphe on a retiré l'observation entourée en bleue ; on voit alors que la courbe de tendance n'est quasiment pas altérée par cette observation. On peut donc en déduire, que même si elle apparaît comme aberrante car loin de la valeur prédite, cette observation n'est pas influente et peut donc simplement être considérée

comme étant un outlier. Sur le troisième graphique, on a cette fois-ci retiré 10 observations, correspondant à un âge du permis entre 0 et 10 ans. On peut observer cette fois-ci que la pente de la droite est altérée ; ces observations sont donc influentes. Leur traitement doit donc être réfléchi si l'on cherche à exprimer une relation linéaire entre âge du permis et prime moyenne. Ici on ne connaît pas la véritable relation entre ces deux paramètres, mais cet exemple permet une illustration claire de ce que sont ou non des outliers.

La plupart du temps, si une observation a une influence néfaste sur le modèle (évolution de la droite assez radicale) alors il vaut mieux la supprimer. En effet une altération profonde des paramètres et du modèle dû à une seule observation annonce un modèle très volatile et peu fiable. Pour cela on va donc tester l'influence de chaque observation pour déterminer si oui on non on devrait utiliser cette observation dans l'élaboration de notre modèle.

Deux méthodes sont généralement utilisées pour la détection d'observations influentes :

- Supprimer une observation du set de données, réentraîner le modèle et regarder la variation des paramètres (s'il y en a) et la mesure d'erreur du modèle
- Attribuer des poids à certaines observations en approximant le changement dans les paramètres en se basant sur les gradients du modèle (fonctions d'influence)

V.2.2.1. Suppression d'une observation

Cette première méthode est la plus intuitive. En effet, quoi de mieux pour tester l'influence d'une observation que de comparer les performances des modèles avec ou sans en tenir compte ?

Cette méthode est assez restrictive puisqu'elle est assez difficile à mettre en place, surtout lorsque l'on dispose de nombreuses données. Elle ne s'applique ainsi généralement qu'aux modèles linéaires généralisés (GLM).

On rappellera ici de manière très épurée le fonctionnement des GLM. Les GLM sont des modèles statistiques qui cherchent à écrire la prime, aka notre variable réponse Y sous la forme

$$\mathrm{Y} = \beta_0 + \sum_{j=1}^{q} \beta_j X_j$$

Où les β_j sont à déterminer par le modèle. On présentera ici deux mesures majeures de l'influence, la mesure DFBeta et la distance de Cook. La première mesure l'effet de suppression sur les paramètres du modèles. La distance de Cook, elle, mesure cet effet sur les prédictions faites par le modèle. Dans les deux cas, il faut entraîner un premier modèle contenant toutes les observations, puis n modèles où l'on retire à chaque fois là $i^{ème}$observation pour i allant de 1à n.

La mesure $DFB\hat{e}ta$ pour l'observation i, $i \in [1, n]$, s'exprime sous la forme

$$DFB\hat{e}ta = \beta - \beta^{(-i)}$$

Où $\beta = (\beta_1, \dots, \beta_q)$ est le vecteur des paramètres associé au modèle avec toutes les observations et $\beta^{(-i)} = (\beta_1^{(-i)}, \dots, \beta^{(-i)}{}_q)$ le vecteur associé au modèle privé de l'observation i.

La distance de Cook pour une observation i s'exprime alors sous cette forme :

$$\mathrm{D_i} = \frac{\sum_{j=1}^{n} (\hat{y}_j - \hat{y}_j^{(-i)})^2}{q \times MSE}$$

La distance de Cook communique combien le résultat prédit change lorsque la $i^{ème}$observation est retirée du set d'entraînement.

Une variable peut être influente au sens de $DFB\hat{e}ta$ mais pas de la distance de Cook et vice-versa. On remarque que cette méthode est coûteuse en temps de calcul vu que l'on doit réentraîner n modèles pour chaque mesure. C'est pourquoi il est difficile de l'appliquer lorsque de larges sets de données sont employés. De plus ces mesures ne fonctionnent que pour des régressions linéaires et ne s'appliquent pas pour des algorithmes plus complexes, comme ceux de Machine Learning.

V.2.2.2. Fonctions d'influence

Ainsi, c'est à la fois pour gagner en temps et en généralité que l'on se tourne vers les fonctions d'influence. Cette méthode est relativement récente puisqu'elle a été proposée par Koh et Liang en 2017.

Cette fois-ci, au lieu de supprimer l'observation, on décide simplement de lui accorder un poids plus fort dans la fonction de perte utilisée tout au long de l'algorithme. Globalement cela revient à estimer la perte autour des paramètres actuels du modèle en utilisant la méthode du gradient (dérivées d'ordre 1) et la matrice Hessienne (dérivées d'ordre 2) du modèle.

Cette méthode nécessite un accès au gradient de perte lié aux paramètres du modèle ; cela n'est pour l'instant possible que pour certains algorithmes de machine learning (régression logistique, réseau de neurones, SVM…) et pour les méthodes plus traditionnelles de modèles linéaires. Néanmoins on note que cette méthode ne s'applique pas aux modèles construits sur des arbres ; en effet ces derniers ne donnent pas accès au gradient au cours de la modélisation.

Malgré son champ d'action restreint, présentons quand même de manière succincte cette méthode. Si l'on appelle $\theta = (\theta_1, , \ldots, \theta_t)$ le vecteur de paramètres du modèle, alors en posant ε la variation de poids infinitésimale ajoutée à l'observation $X^{(i)}$ dont on veut simuler la suppression, et L la fonction de perte, alors on a comme nouveau paramètre $\hat{\theta}_{\varepsilon,\ddot{X}} = arg\min_{\theta\epsilon\Theta} (1-\varepsilon) \times \frac{1}{n}\sum_{i=1}^{n} L(\ddot{X}, \theta) + \varepsilon\, L(X^{(i)}, \theta)$.

On obtient alors la fonction d'influence par $I(x_i) = \frac{d\hat{\theta}_{\varepsilon,\ddot{X}_i}}{d\varepsilon}|_{\varepsilon=0} = -H_{\hat{\theta}}^{-1} \times \nabla_\theta L(X^{(i)}, \hat{\theta})$

$H_{\hat{\theta}}$ est la matrice Hessienne, soit la seconde dérivée de la fonction de perte liée aux paramètres du modèle ; il s'agit de la vitesse de changement du gradient de perte (à quel point la courbe de perte est incurvée à un certain point). $\nabla_\theta L(X^{(i)}, \hat{\theta})$ correspond à la perte de gradient respective aux paramètres modifiés dûs à l'observation pondérée.

Cela permet, à terme, de comparer des modèles de machine learning, qui peuvent accorder différentes importances à une même observation ; cela donne une meilleure compréhension de leur comportement et permet d'aller plus loin que la seule performance prédictive. Dans une autre optique, cela peut également aider à développer des données divergente qui peuvent ensuite être utilisée pour tester l'algorithme à des fins d'amélioration.

Toutefois, aucune implémentation simple des fonctions d'influence n'a encore été réalisée, c'est pourquoi nous nous contenterons ici de regarder les observations influentes au sens du ratio de Cook et de la mesure DFBêta.

Ci-dessous une comparaison du nombre d'observations qualifiées d'outliers, par les deux méthodes à travers nos données :

Modèle	Distance de Cook	DFFits
GLM	17	116

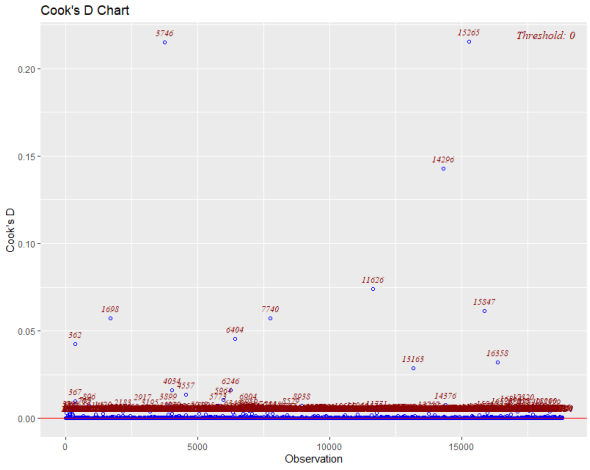


Figure 47 : Distance de Cook

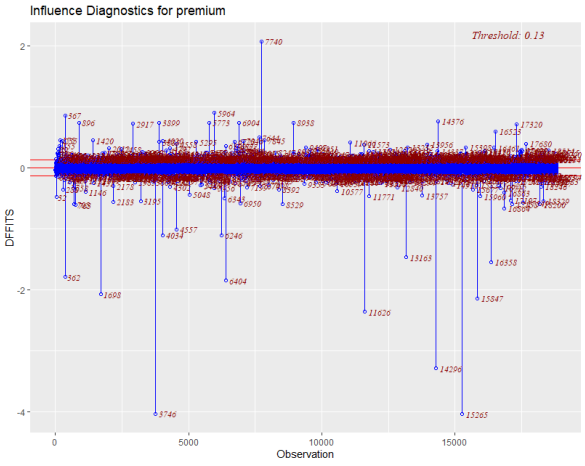


Figure 48 : Graphe DFFIT

On peut donc voir que de nombreuses observations ont influées sur notre modèle. Elles sont représentées par les points bleus. Néanmoins, dans notre décompte présenté dans le tableau plus haut, seules les observations se démarquant vraiment des statistiques des autres observations ont été dénombrées.

Dans une optique générale de modélisation, les fonctions d'influence et de suppression d'observation sont les meilleurs outils pour débugger des modèles de machine learning. En effet, elles donnent une indication directe sur quelles observations devraient être contrôlées pour erreur. Néanmoins ces méthodes requièrent un long temps de calculs puisqu'elles nécessitent d'entraîner de multiples fois des modèles. De plus la communauté statistique n'est pas encore parvenue à s'accorder sur la détermination d'un seuil au-delà duquel on considère une observation comme influente. On retrouve également cette problématique dans d'autres domaines statistiques, comme l'utilisation de la valeur p.

L'un des reproches pouvant également être fait quant à l'utilisation des fonctions d'influence, est que ces dernières ne prennent en compte la délétion que d'une observation à la fois. Ainsi, si plusieurs observations sont corrélées, alors elles peuvent, en tant que groupe, avoir une influence plus forte lors de l'entrainement du modèle ; ce genre de phénomène ne serait donc pas pris en compte avce des mesures telles que le DFFIT ou la distance de Cook.

V.2.3 Interprétation de l'outcome

Maintenant nous allons nous intéresser à à quel point on peut gagner à comprendre l'interprétabilité de notre modèle de machine learning. En R, avec une recrudescence de personnes cherchant à résoudre cette problématique, de nombreux package se développent. Avec ceux-ci, une littérature tend à grossir derrière.

Dans cette partie nous proposerons donc quelques unes des nouvelles techniques d'interprétabilité des modèles de machine learning. Nous verrons ainsi ce qu'expriment chacunes de ces techniques sur nos données. On citera par exemple :

- les Partial Dependence Plot (PDP) et les ICE (Individual Conditional Expectation)
- les graphes ALE
- les valeurs de Shapley

Cette partie est assez similaire avec la paragraphe trouvé en III.4, puisque nous cherchons à nouveau à comprendre nos données. Néanmoins les enjeux ne sont plus les mêmes. Dans la partie III nous effectuions une analyse a priori de nos données ; désormais nous souhaitons mener une étude a posteriori, et comprendre notre modèle de manière plus complète. De plus, la démarche menée en III avait pour but de restreindre le nombre de variables utilisées de façon à parvenir, dans le cas des modèles statistiques classiques, au modèle le plus performant. Hors, puisqu'il a été démontré en IV que ces modèles avaient une performance inférieure à ceux de machine learning, nous pouvons désormais nous affranchir de cette contrainte et tenter d'appréhender le modèle dans sa globalité.

L'objectif de cette partie est donc d'étudier et d'interpréter l'impact de différentes variables sur nos données, grâce à l'aide de notre modèle. Ainsi, seront mis en lumière d'autres effets que ceux qui auraient pu être révélés par une analyse préliminaire. Toutes ces méthodes ont un lien direct avec le modèle mis en oeuvre, et de ce fait transfigurent totalement ce qui peut ressortir de leur étude.

Dans cette partie, le modèle utilisé sera un de nos modèle 'classique' optimal, soit notre modèle GBM déterminé en IV.3.

V.2.3.1. PDP / ICE

Commençons avec les graphes de dépendance partielle (PDPs) et les graphes d'espérance conditionnelle individuelle (ICE, Individual Conditional Expectation). Ces méthodes ont été documentées par Friedman (2001) et Goldstein et al (2014).

Ces deux méthodes sont présentées dans une même sous-partie car elles sont intimement liées. Les PDPs ne sont en réalité qu'un graphe moyennant les résultats fournis par les courbes ICE. Mais intéressons nous à la manière dont ces graphes sont déterminés pour mieux comprendre leur intérêt dans notre démarche.

Pour une variable X_j, les graphes ICE sont évalués en chaque valeur $x_{i,j}$ prise par X_j. En d'autres termes, cela signifie que pour chaque valeur différente prise par X_j (par exemple, pour chaque type de carburant de l'étude), disons k valeurs, on force toutes les observations de notre set de données à prendre cette valeur x_l, $l \in [1, k]$. On obtient alors un nouveau set de données, où la variable X_j ne prend qu'une seule valeur, x_l. On évalue notre modèle sur ce nouveau set d'observations et on passe à une autre valeur pour x_j. On dispose alors de nos n observations, qui ont chacune k primes prédites.

Les graphes ICE présentent donc n courbes, une pour chaque observation. Dans un souci de lisibilité, nous avons ici modélisé un graphe ICE sur un échantillon de $n_0 = 1000$ profils. Ce graphe n'est donc pas forcément représentatif de notre jeu de données, puisque l'on ne peut pas être sûr que notre échantillon soit un bon échantillon de notre population globale.

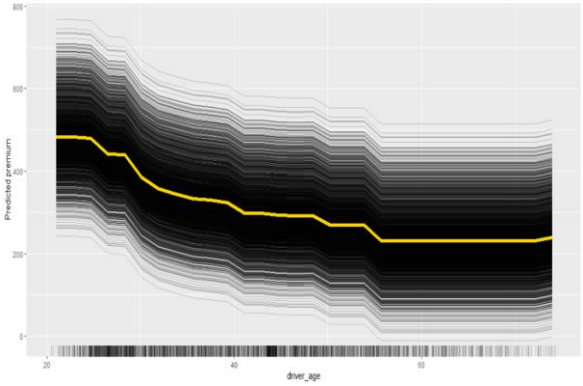


Figure 49 : ICE Plot par âge du conducteur

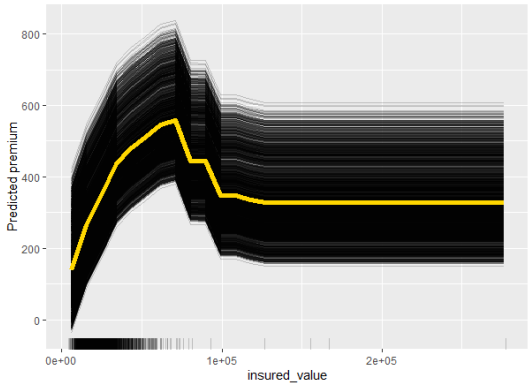


Figure 50 : ICE Plot par valeur assurée

On remarque sur les graphes ci-dessus que le faible nombre de données renseignées pour certaines valeurs (valeur assurée très importante par exemple), cause un phénomène inattendu sur les courbes ; ces dernières doivent donc être traitées avec une prudence encore plus grande dans ces cas de figure.

Cela explique que, dans certains cas, il vaut mieux utiliser leur acolyte, le graphe de dépendance partielle (courbe jaune présentée ici). Les PDPs sont en effet des généralisations des graphes ICE, qui ne présentent qu'une seule courbe qui se trouve être la moyenne de toutes les courbes ICE. Ils représentent donc la prédiction moyenne à travers toutes les observations. On peut ainsi percevoir l'effet marginal qu'a une variable sur notre modèle. A travers ce type de graphe, on peut voir si la relation entre une variable explicative et la variable réponse est linéaire, monotone ou plus complexe.

On introduit ainsi quelques notations : X_S comprend les variables d'intérêt, X_C les autres variables présentes dans le modèle ; l'ensemble de ces variables est représenté par $X_{tot} = X_S :: X_C$. Similairement, $x_S \in X_S$ représente la modalité dont nous voulons connaître l'impact et $x_C^{(i)} \in X_C$ les caractéristiques du prospect i, privées de la variable X_S. La fonction de dépendance partielle s'écrit alors $\widehat{f_{X_S}}(x_S) = E_{X_C}\left[\hat{f}(x_S :: X_C)\right] = \int \hat{f}(x_S :: x_C)\, d\mathbb{P}(x_C)$. Cette fonction, prise en une valeur de X_S particulière représente la prédiction moyenne que l'on obtiendrait si l'on forçait toutes les observations à avoir cette valeur de X_S.

On estime généralement $\widehat{f_{X_S}}(x_S)$ par la méthode de Monte-Carlo, c'est-à-dire $\widehat{f_{X_S}}(x_S) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}\left(x_S :: x_C^{(i)}\right)$.

Le calcul des PDP a une interprétation causale ; on analyse la relation cause à effet entre la variable et la prédiction. Toutefois, cela ne représente pas vraiment le monde réel et peut créer des cas aberrants (par exemple, si on modifie l'âge du permis sans modifier l'âge du conducteur). Une des hypothèse des PDP est donc que les variables présentent dans le support C sont non-corrélées avec celles du support S.

Néanmoins, par rapport aux graphes ICE, on se se prive de la détection éventuelle d'effets croisés. En effet, s'il existe un effet hétérogène, dû à une interaction entre les deux variables étudiées, alors en moyennant toutes les prédictions, on peut perdre de vue cet effet.

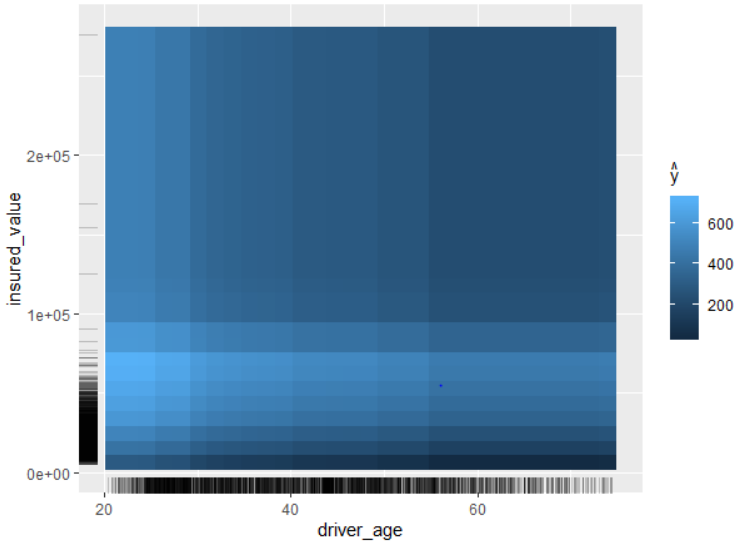


Figure 51 : PDP à deux dimensions sur l'âge du conducteur et la valeur assurée

Des PDP peuvent également exister en deux dimensions pour permettre une vision claire de l'interaction de deux variables sur la détermination de la prime finale. Ici on voit que l'évolution de la valeur assurée est celle vue sur le graphique à une dimension. Néanmoins on voit aussi que l'effet de la valeur assurée (plus de couleurs dans le dégradé) est plus important lorsque le conducteur principal est jeune.

V.2.3.2. Graphes ALE

Il existe toutefois une alternative aux PDPs : les ALE plots (Accumulated Local Effects). L'esprit est le même que celui des graphes PDPs et ICE. On cherche à nouveau à obtenir l'effet de la variation d'une variable sur la valeur de notre variable réponse. Sauf que cette fois cette alternative est non-biaisée ; c'est à dire que des profils non réalistes n'apparaîtront pas lors du calcul de la courbe.

Pour cela, les graphs ALE calculent la différence entre les prédictions plutôt que les moyennes (pour obtenir la prédiction pour un âge 30 ans, on soustrait la prédiction obtenue pour un âge remplacé par 29 à celle obtenue pour un âge remplacé par 31).

On a alors, en reprenant les mêmes notations introduites dans la partie précédente :

$$\widehat{f_{X_S}}(x_S) = \int_{z_{0,1}}^{x_S} E_{X_C|X_S}[\hat{f}^S(X_S :: X_C) \mid X_S = z_S]\, dz_S - constante$$

Avec $\hat{f}^S(x_S :: x_C) = \frac{\delta \hat{f}(x_S :: x_C)}{\delta x_S}$. Ce qui donne :

$$\widehat{f_{X_S}}(x_S) = \int_{z_0}^{x_S} \int_{x_C} \hat{f}^S(z_S :: x_C)\mathbb{P}(x_C|z_S) dx_C dz_S - constante$$

z_0 étant une valeur de X_S choisie de telle sorte qu'elle soit proche de la borne basse (si X_S numérique). Néanmoins ce choix est peu important, puisqu'il n'affecte que la translation verticale du graphe ALE, effet qui sera corrigé par la constante. Généralement, on essaye de centrer le graphe ALE, de manière à ce que l'effet moyen soit de 0, permettant ainsi d'avoir une vision plus claire de l'effet des caractéristiques de la variable par rapport à la moyenne. On voit ainsi que si l'effet sur la prime peut monter jusqu'à +150€ pour les conducteurs jeunes, à partir de 55 ans, l'effet sur la prime est proche de – 100€. On voit également qu'après 65 ans on observe une légère remontée de l'effet moyen de l'âge sur la prime.

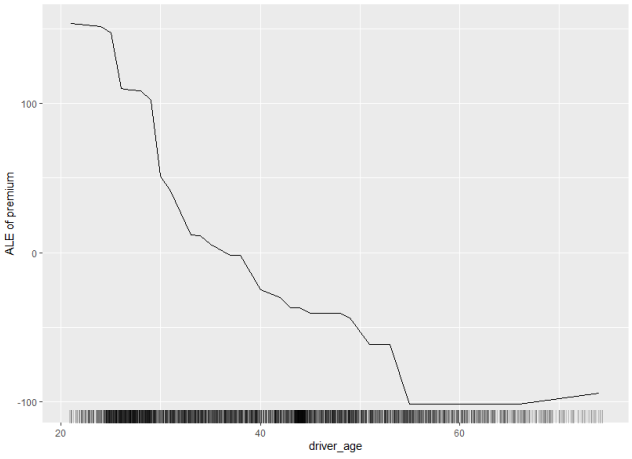


Figure 52: Graphe ALE sur l'âge du conducteur

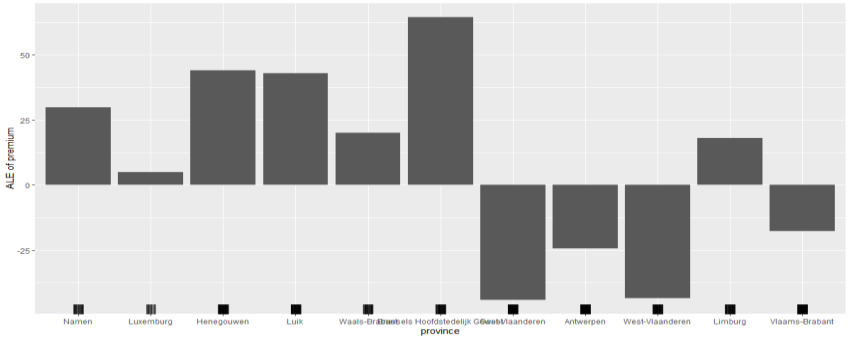


Figure 53 : ALE Plot sur une variable catégorielle, la province

Si on utilise les estimateurs ALE pour deux variables, alors on voit l'effet de second-ordre de ces variables ; c'est-à-dire l'effet d'interaction additionnel produit par ces variables, une fois que les effets principaux ont été retirés. On voit ainsi que cet effet est important chez les jeunes ($-3e^{-12}$), et chez les personnes de 30 à 35 ans ($+1e^{-12}$).

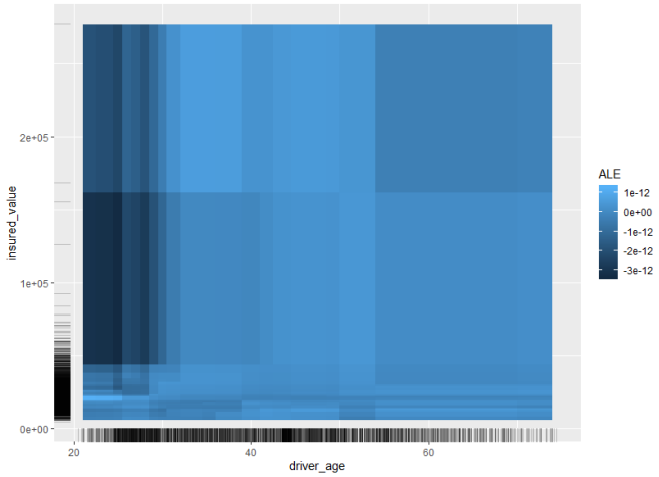


Figure 54 : ALE Plot à deux dimensions sur l'âge du conducteur et la valeur assurée

Les graphes ALE montrent donc l'interaction entre deux variables, alors qu'un graphe PDP montre lui l'effet combiné des deux variables. Néanmoins, vu que l'on se retrouve à nouveau dans le cas où l'effet est moyenné, on perd la possibilité de détection d'une hétérogénéité éventuelle.

V.2.3.3. Valeurs de Shapley

Une autre technique d'interprétation des modèles de machine learning consiste à utiliser les valeurs de Shapley (Shapley, 1953), issues de la théorie des jeux et aux applications multiples. Proposée par Lundberg et Lee en 2016, l'idée derrière les valeurs de Shapley adaptées au machine learning, est de tester toutes les combinaisons possible de prédicteurs afin de déterminer leur impact. La valeur de Shapley correspond à la contribution de chaque caractéristique d'une variable envers la valeur prédite, comparée à la prédiction moyenne du data set.

La valeur de Shapley nécessite néanmoins une valeur de base, réelle. C'est-à-dire qu'elle s'appuie sur une observation et un modèle, et non seulement sur un modèle. Elle détermine ainsi l'impact sur la prédiction de la variable X_j avec cette valeur, et non l'impact de la variable $X_j$dans son ensemble. Néanmoins, son application est surtout à but individuel, puisque les valeurs de Shapley s'appliquent sur un set de combinaison bien particulier.

Intéressons-nous à la variable X_j et à sa valeur x_l, $l \in [1, k]$, où k est le nombre de valeurs différentes possibles pouvant être prises par X_j. On va tout d'abord évaluer toutes les combinaisons possible de variables ne comprenant pas x_l. Pour cela on va utiliser des coalitions de tailles différentes (on va fixer une variable et moyenner, puis une autre… Puis fixer deux variables et moyenner la prédiction…). A chaque fois, la valeur prédite sur les observations de la coalition est calculée avec et sans la valeur x_l (la valeur différente de x_l étant tirée aléatoirement dans les valeurs possibles).

On va ensuite calculer la différence entre les deux valeurs de prédictions trouvées pour chaque coalition. La valeur de Shapley correspond à la moyenne pondérée de ces différentes contributions marginales.

La valeur de Shapley est la contribution de la valeur d'une variable à la différence entre la prédiction réelle et la prédiction moyenne. La valeur de Shapley d'une observation donnée, avec val la fonction valeur (c'est-à-dire le modèle 'black-box), pour la variable X_j de modalité x_l s'écrit :

$$\phi_j(val) = \sum_{S \subseteq \{x_1,\dots,x_k\} \setminus \{x_j\}} \frac{|S|! \times (k - |S| - 1)!}{k!} \times (val(S \cup \{\mathrm{x_j}\} - val(S))$$

La valeur de Shapley est la seule méthode d'attribution qui satisfait des propriétés d'efficacité, de symétrie, de mutisme et d'additivité. On retrouve donc, grâce par exemple à sa propriété d'efficacité que $\sum_{j=1}^{p} \phi_j = \hat{f}\left(X^{(i)}\right) - E_X[\hat{f}(\ddot{X})]$; c'est-à-dire que la somme des effets doit être égale à la différence entre la prédiction pour un prospect i et la moyenne.

Les valeurs de Shapley sont considérées plus robustes que les résultats obtenus avec la méthode LIME, qui seront présentées dans la partie V.2.4. En effet elles garantissent que la différence entre la prédiction moyenne et la prédiction est distribuée équitablement parmi les valeurs des variables de l'observation.

A titre de comparaison, nous allons ainsi comparer les résultats obtenus sur un profil aléatoire de notre base de données. L'assuré sélectionné a une prime réelle de 262.03€ et une prime prédite de 278.60€.

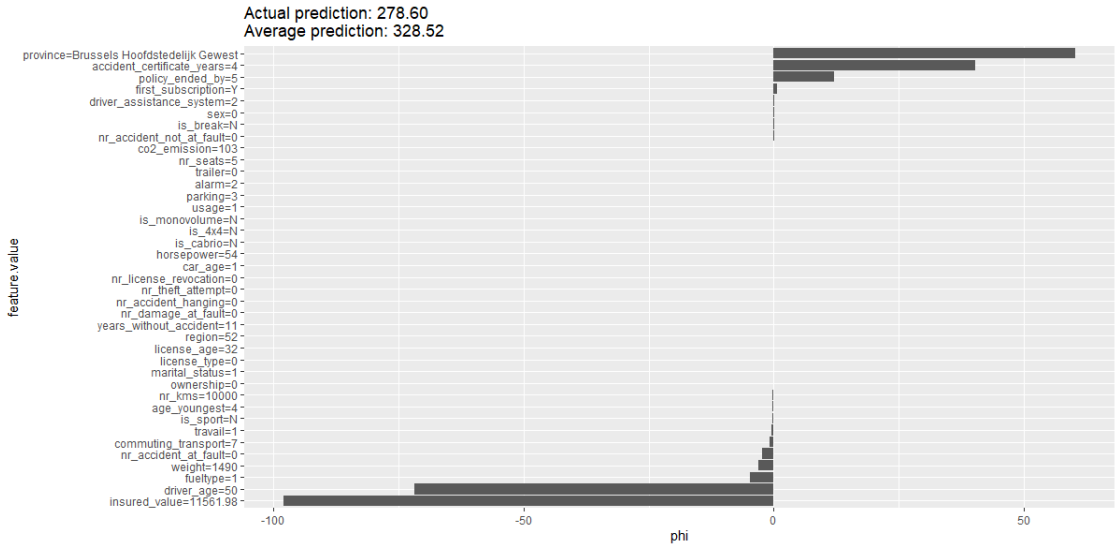


Figure 55 : Valeurs de Shapley sur un prospect donné

On obtient donc un lien direct avec les variables jugées comme les plus influentes par notre algorithme de prédiction. Néanmoins, on observe que peu de variables ont contribuées à l'explication

du tarif. La prédiction moyenne donnée par les valeurs de Shapley sont assez éloignées de la prédiction réelle.

Cela vient du fait qu'une fois de plus on se heurte au problème que l'on peut rencontrer des observations peu probables dans la réalité. Il s'agit en effet d'un défaut récurrent des méthodes basées sur des techniques de permutation.

Il existe également un autre algorithme basé sur les valeurs de Shapley : l'algorithme SHAP (Shapley Additive exPlanations). Cet algorithme permet, grâce à l'aggrégation de différentes valeurs de Shapley, de produire des méthodes d'interprétation globales d'un modèle de machine learning. Néanmoins, cet algorithme ne sera pas développé dans ce mémoire. On mentionnera néanmoins que la version de l'algorithme TreeSHAP permet d'améliorer le temps de calcul des valeurs de Shapley.

V.2.4 Confiance dans le modèle

"I'm not upset that you lied to me, I'm upset that from now on I can't believe you" est une citation célèbre du philosophe allemand Friedrich Nietzche. Cette citation s'applique particulièrement dans notre cas. En effet, nous avons tendance à surestimer les pouvoirs des ordinateurs et de leurs calculs ; leur accordant même parfois une confiance aveugle. Mais il ne faut pas oublier que ces derniers ne sont pas parfaits et peuvent commettre des erreurs. Une fois ce fait réalisé, une question se pose : 'S'il s'est trompé une fois, pourquoi cela ne se reproduirait-il pas une seconde fois ? Et une troisième fois ?'.

D'autres questions peuvent suivre. Ces erreurs sont-elles si impactantes ? Si non, pourquoi ne ferions-nous pas simplement confiance à notre modèle ? Pourquoi essayer de comprendre pourquoi il a pris une telle décision ? Vivre dans le déni des erreurs qui peuvent être commises peut parfois être particulièrement séduisant.

En effet, dans certains cas, seul le résultat importe. Mais quand on s'intéresse à l'obtention d'une marge d'erreur, que l'on essaye de comprendre quelles sont les variables expliquant ce résultat, on a besoin d'interprétabilité. Ainsi, si l'on comprend pourquoi un modèle réussit, on comprend également pourquoi il se trompe ; on gagne ainsi en pertinence et on peut essayer de corriger ce défaut. Des chercheurs ont ainsi montré que parfois les décisions prises prises par un algorithme grâce à des vertus comme l'originalité ou le leadership déguisaient souvent de la discrimination sociale ou sexuelle (Mounk, 2014). C'est pourquoi il faut toujours se montrer prudent avec les résultats obtenus, de manière à ne pas répliquer à grande échelle un comportement pathologique.

De plus, dans d'autres circonstances que les nôtres, l'interprétabilité d'un modèle permet de détecter un biais qui aurait été observé dans l'échantillon d'entraînement. Encore dans un autre domaine, l'interprétabilité peut permettre aux gens ou aux programmes de manipuler le système ; par exemple, dans un système de scoring de crédits. La banque et le particulier ne partagent pas le même objectif : la banque ne souhaite donner des crédits que si le score du particulier est bon, alors que le particulier cherche à obtenir un prêt quelque soit son score. Si le particulier sait qu'avoir plus de trois cartes de crédit nuit à son score, alors il fera en sorte de résilier ses autres cartes avant la demande du prêt, pour ensuite les recontracter une fois le prêt obtenu. Son score s'est amélioré, mais sa capacité de remboursement reste inchangée. Il a ainsi pu faire jouer le système en sa faveur car il en connaissait les rouages.

Il est notamment possible de tromper le système lorsque les variables utilisées dans la détermination du résultat sont des proxys d'une variable causale, mais ne provoquent pas directement le résultat. On peut citer un exemple de ce mécanisme avec le système Google Flu Trends

développé par Google. Le but de ce système était de corréler les recherches Google avec les épisodes de grippe. Au final, les résultats n'étaient pas concluants car le système avait manqué de nombreux épisodes à cause des changements des mots-clés recherchés et une corrélation seulement partielle (recherche du mot "fièvre") loin d'être unique.

Différentes méthodes d'interprétation existent donc, et peuvent être regroupées dans des catégories telles que : sommaire statistique des variables, résumé visuel des variables, modèles internes, data point, modèles interprétables intrinsèquement…

L'interprétabilité pourrait être alors confondue avec la transparence du modèle, mais ce sont deux notions totalement différentes. La transparence d'un algorithme demande seulement des connaissances sur l'algorithme et non sur les données. On comprend comment l'algorithme marche indépendamment des données mais on ne sait pas comment il fonctionne dans le cas particulier de nos données. L'interprétabilité, elle, cherche à nous procurer des explications post-hoc.

Nous allons donc explorer dans ce chapitre plusieurs méthodes, comme les interprétations gloables / locales du modèle, qui veulent rejoindre machine learning et méthodes statistiques pour faciliter la compréhension des modèles 'black box'.

V.2.4.1. Global Surrogate

Une première manière de rendre un modèle "black box" plus interprétable est de le remplacer par un modèle plus simple (modèle "white box"), comme un arbre de décision. Une fois le premier modèle "black box" trouvé, on calcule ses prédictions sur notre dataset originel.

On entraîne alors un second modèle, cette fois-ci "white box" sur le dataset originel, mais en essayant cette fois-ci de prédire, non pas la variable réponse, mais les prédictions de cette réponse faites par le modèle précédent. Si ce second modèle parvient à bien répliquer ces prédictions, et si notre premier modèle était cohérent, alors on peut se contenter d'interpréter notre modèle 'white box' pour comprendre le modèle global.

Cette méthode permet de simplifier le processus d'interprétation. Néanmoins, il faut garder en mémoire qu'il faut quand même que le résultat reste cohérent ; c'est-à-dire que cette méthode ne marche pas dans toutes les situations. Il faut toujours s'assurer que le modèle 'white box' arrive à représenter de manière suivie le modèle "black box".

Les critères principaux devant être observés chez un modèle ayant pour but de blanchir un modèle 'black box' sont :

- La justesse : la précision du modèle approximant devrait être du même ordre que celle du modèle 'black box'
- La fidélité : le modèle approximant devrait avoir des prédictions proches de celles du modèle boîte noire
- La consistance : les deux modèles devraient avoir une certaine cohérence dans leur utilisation des variables
- La stabilité : des observations similaires devraient avoir les mêmes explications
- La compréhensibilité : un humain devrait être capable de comprendre les explications fournies par le modèle approximant
- La certitude : un modèle approximant donnant une information sur la confiance du modèle est un plus

V.2.4.2. Local Surrogate

L'interprétabilité globale d'un modèle permet de comprendre la distribution de la variable résultat à partir des différentes variables. Mais en pratique elle est difficile à atteindre car un modèle avec plus de quelques paramètres a du mal à être conçu et visualisé par le cerveau humain, souvent mal à l'aise au-delà d'espaces de dimension supérieure à 3.

Une autre manière de comprendre un modèle est alors de se concentrer sur l'interprétabilité locale d'une prédiction unique. Le but est ainsi de comprendre pourquoi on a obtenu ce résultat et comment. On peut alors trouver des dépendances linéaires qui ne s'appliquent que dans des cas spécifiques ; par exemple, que pour les hommes de 30 ans, la puissance du véhicule est linéaire avec la prime.

Les techniques d'interprétations locales fournissent des méthodes pour expliquer pourquoi une prédiction individuelle a été donnée pour une observation donnée.

Une de ces techniques d'interprétation est la technique LIME (local interpretable model-agnostic explanations). Développée par Ribeiro et al (2016); elle permet d'ordonner les variables selon leurs contributions à la variable réponse. On obtient ainsi l'effet, positif ou négatif et plus ou moins marqué, de chaque variable du modèle. On peut donc vérifier, en comparant deux observations aux résultats finaux complètement différents, l'importance à leurs échelles des différentes variables.

L'algorithme LIME teste l'effet d'une modification des données d'entrées sur les prédictions. Il génère un nouveau dataset constitué de l'observation d'intérêt altérée ; dans les faits, l'algorithme modifie pour chaque ligne chaque variable en se basant sur une distribution normale centrée sur la valeur générale de la variable. Le modèle 'black box' est alors appliqué sur ce dataset pour obtenir une prédiction sur le tarif. Un nouveau modèle, cette fois-ci interprétable, est alors entraîné sur ce dataset, pondéré par un poids représentant la proximité des données avec l'observation initiale, de manière à obtenir une prédiction de la prime.

Or, à partir de nos données il est difficile d'établir une relation de proximité compte tenu du nombre de variables entrant en jeu dans le modèle. Les résultats dépendent donc fortement de la forme du noyau choisi pour faire baisser la dimensionnalité, et de la taille de ce dernier.

On reprend ainsi le prospect sélectionné dans la partie V.2.3.3. concernant les valeurs de Shapley et on obtient alors les résultats suivants :

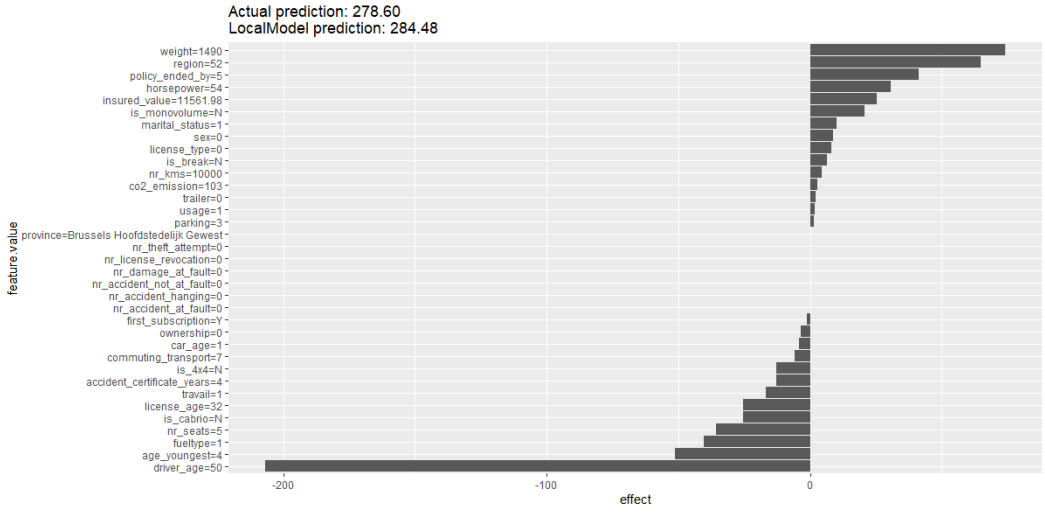


Figure 56 : Graphe LIME pour un prospect donné (même que pour valeurs de Shapley)

On voit donc que le modèle local surestime la prime proposée. De plus, certaines variables qui ne sont pas forcément jugées importantes dans le modèle de prédiction, apparaissent comme particulièrement importantes dans notre modèle local (Modèle linéaire). Néanmoins, cela permet, à moindre mesure, d'obtenir des informations plus claires sur les variables qui ont joué dans la prédiction, surtout lorsque l'on compare cette prédiction à des profils similaires.

Ribeiro, Singh et Guestrin ont également proposé les règles de portées, suivant le même cheminement que l'algorithme LIME, mis à part le fait que cette fois-ci l'algorithme renvoie des règles SI-ALORS, appelées 'ancres'. Mais ce genre d'algorithme reste difficilement applicable à une base de données de grande taille.

V.3. Détection d'un changement de modèle inhérent

Dans la réalité, les données récupérées peuvent être altérées par des changements inhérents. En effet, sur une période, le tarif proposé par un assureur peut être sujet à des soldes flottants, permettant sur une certaine période d'augmenter le taux de souscription ou d'attirer un certain segment de la population. Ou, même sans cette problématique, un changement tarifaire peut survenir à une quelconque période, sans que cela soit notifié dans la base de données.

Ces changements dans le tarif sont dificiles à repérer. Quand il agit au niveau global, ce genre de phénomène peut être détecté plus ou moins facilement, selon l'amplitude de la réduction appliquée ; à contrario, lorsque seule une partie de la population est affectée par ce changement de prix, alors il est quasiment impossible de déceler un changement de tarif sans informations à priori. En effet, sans connaître ni la date, ni les caractéristiques de la population touchée, il est quasiment impossible de remarquer un changement dans l'algorithme de tarification initial.

Nous nous intéresserons donc ici principalement à des changements de tarifs globaux, impactant de manière similaire chaque prospect. Le but étant, de parvenir à déceler la date et le niveau de changement afin de pouvoir redécouper la base de données de manière à ce que celle-ci puisse être utilisée à fortiori dans la démarche qui a précédé ce chapitre. En effet, si un changement de tarif a affecté 1/3 de nos données, alors le modèle réalisé sur ces données est faux ; en effet, il sera biaisé puisqu'il aura tenté de matcher à la fois les tarifs avec et sans promotion. La démarche est alors complètement apocryphe. Un assureur peut aussi décider, suite à une nouvelle tarification, ou après comparaison avec le marché d'abaisser (ou d'augmenter) de manière permanente ses tarifs.

On notera que certaines techniques utilisées classiquement dans la détection de phénomènes de rupture, comme l'utilisation de technique bootstrap ou de la fonction CUSUM s'appliquent difficilement à notre problématique, puisque nous ne nous attaquons pas à une problématique de changement de moyenne. L'utilisation d'une technique telle que le CUSUM pré-supposerait que la moyenne des observations soit globalement égale dans le temps. Hors aucune information sur la distribution des profils ne permet d'assurer cela.

C'est pourquoi, la prochaine section cherchera à proposer une solution à cette problématique, fort similaire à celle de la détection d'une rupture en finance quantitative ou dans la fréquence de survenance de sinistre d'un assuré. Nous cherchons en effet à détecter dans nos données un changement de modèle.

V.3.1 Création des données

Ne disposant pas d'une base de données adaptée à cette problématique, il nous avons été amenés à construire une nouvelle base de données, basée sur la base originelle, mais agrémentée de quelques changements. En effet la base de données originelle ne dispose pas de la data de proposition du tarif au prospect. De plus, nous ne savons pas en amont, si un changement de tarif a été réalisé ou non.

Nous travaillons ici encore sur la base de données définie en II. C'est-à-dire sur une base de données comprenant 105 361 observations et 35 146 profils.

Dans cette optique de création d'une base de données adaptée, une colonne 'date de proposition du contrat' a été ajoutée à notre base. Celle-ci a établi de manière arbitraire une distribution des observations au cours de l'année. Les données ont ensuite été ordonnées et la date du 10 septembre a été choisie comme date de changement de tarif. Sachant que notre base de données débute au 6 janvier 2019, ce qui veut dire qu'environ 68% des données bénéficient de l' 'ancien modèle de tarification et 30% du 'nouveau' tarif.

Nous avons décidé, pour tester la robustesse de nos moyens de détection d'appliquer deux méthodes de soldes différents :

- Une remise de 50€ toutes caractéristiques confondues pour chaque prospect (BDD 1)

- Une remise de 15% sur le tarif proposé avec l'ancienne tarification (BDD2)

Dans la réalité, l'assureur cherchant une analyse de ses compétiteurs dispose de données a priori sur l'établissement de soldes flottants (mais pas sur un changement de modèle de tarification). La démarche peut néanmoins lui être utile dans ces circonstances, en lui permettant d'établir un lien entre chaque assureur présent dans la base de données (anonymisé) et sa réelle dénomination.

V.3.2 Modélisation basique

Nous reprenons ici la démarche appliquée en IV pour parvenir à l'obtention de la meilleure modélisation. Mais nous introduisons cette fois-ci la variable 'date de proposition du contrat' comme variable explicative, pour donner une chance à l'algorithme de détecter par lui-même ce changement de tarif.

Nous développons ainsi différents modèles, que nous raffinons de la même manière que vue en IV pour parvenir au modèle le plus pertinent possible.

Toujours sur le critère de la RMSE, nous sélectionnons ainsi la modélisation réalisée par le GBM pour la BDD 1.

	R^2 du modèle	RMSE Éch.Entr	RMSE Éch.Test	Nombre de variables
Modèle naïf	0	118.5755	119.516	0
GBM	0.896363	38.15821	37.7545	40

Pour la BDD 2, l'algorithme le plus performant est également le GBM.

	R^2 du modèle	RMSE Éch.Entr	RMSE Éch.Test	Nombre de variables
Modèle naïf	0	113.3653	113.946	0
GBM	0.892391	37.53422	37.6413	40

On peut voir sur ces tableaux que les algorithmes GBM restent quand même bien supérieurs à l'utilisation d'un modèle naïf. Nous allons dans la suite nous attarder à le rendre plus adéquat, puisque le but est de n'utiliser que les données valides pour construire le tarif.

Par la suite, les algorithmes GBM et Random Forest seront utilisés dans les phases de modélisation, du fait de leur rapidité et de leur efficacité.

V.3.3 Détection

La détection d'une rupture nécessite à la fois une date de survenance (ou date de rupture) ainsi qu'un changement inhérent au processus. Ici, nous ne disposons d'information sur aucun de ces éléments.

Le but de ce paragraphe est donc, à travers deux méthodes simples, de détecter à la fois la date de rupture mais également la correction qui a été appliquée au tarif. Pour cela le but est simple. Une fois la date de rupture détectée, deux modélisations seront réalisées indépendamment ; une sur la partie antérieure à la date de rupture, et une autre sur la partie postérieure. Les prédictions réalisées sur la base entière des deux modèles, sans égard de la date de cotation, seront ensuite comparées pour estimer la différence de prix entre les deux tarifications.

Pour cela deux stratégies seront déployées : une première se basant sur la méthode dite de dichotomie, et une seconde, plus artisanale, consistant à tester une à une chaque date pour déterminer celle la plus susceptible d'être une date de rupture.

V.3.3.1. Dichotomie

La première stratégie employée se base sur le principe simple de la dichotomie. Le but est simple. Après avoir créé un premier modèle sur l'entièreté des données, sans prise en compte de la date :

- On divise le segment temporel en deux segments de taille équivalente
- On crée un modèle sur les données postérieures à la date de division, et on compare son critère de performance avec celui obtenu sur les données antérieures à la date de division. Si on remarque que le modèle s'est significativement amélioré, alors on répète la démarche sur le segment de gauche puisque cela signifie que la date de rupture est antérieure à celle déterminée. Si la performance du modèle s'est peu améliorée, alors cela veut dire que l'on prend toujours en compte des données qui sont incohérentes ; on répète donc l'opération sur le segment de droite
- On interrompt le processus lorsque l'amélioration du modèle est à peine décelable.

Avec cette méthodologie, nous obtenons pour nos deux bases de données ces facteurs :

	Date de rupture	RMSE Éch.Entr.Avant	RMSE Éch.Entr.Après
BDD 1	D - 182	36.864304	42.258976
	D - 273	46.480137	41.90995
	D - 227	42.604885	46.81295
	D - 250	48.58738	55.109283
BDD 2	D - 182	36.935593	37.99647
	D - 273	39.87691	38.72403
	D - 227	44.235332	42.087234

Pour la base de données 1, on finit par obtenir une date de changement qui se situerait entre + 250 jours et + 273 jours. Néanmoins, à ce niveau d'observations, les modèles ne se font que sur 1000 observations et sont donc peu stables. Si on s'arrête à l'étape précédente, on obtient un intervalle de + 227 jours à + 273 jours pour la date de changement de modèle, ce qui est cohérent avec la réalité (changement à date 247).

Pour la base de données 2, le modèle a eu plus de mal à déterminer la date de changement de modèle avec précision. Il la situe entre +182 et +227 jours, alors qu'il se trouve à la date +247.

Cette méthode semble donc peu pertinente, dans le cas de la recherche d'une date précise de changement. On notera également d'autres gros défauts, comme le fait que cette méthode ne peut trouver efficacement qu'un unique point de rupture et est donc peu exploitable dans la détection de phénomène saisonnier. On voit également, par sa configuration, que la date de rupture déterminée demeure peu robuste et très volatile, du fait que l'ajout de quelques données 'erronées' ne nuit pas forcément énormément au modèle, si elles font partie d'un plus grand ensemble.

V.3.3.2. Recul progressif

Théoriquement cette méthode est plus robuste que l'algorithme de dichotomie, puisqu'elle permettrait, si nécessaire, de détecter plusieurs changements dans le processus de tarification.

Cette méthode, tout comme la précédent est assez intuitive. En effet, en partant de la dernière date disponible (dans notre cas le 5 janvier 2020), nous allons petit à petit nous déplacer dans le passé, ajoutant ainsi de plus en plus de données à notre base de données temporaires, et sur laquelle nous allons construire une succession de modèle.

Le fait de rajouter des données améliore les performances de l'algorithme, mais est néanmoins dangereux dans les premières itérations car on dispose de peu de données, ce qui peut conduire l'algorithme à sur-apprendre, et à ainsi produire des résultats sur évalués. Il faut donc généralement décaler la date de début de l'algorithme de quelques jours pour éviter ce problème.

Le principe est donc simple. Au fur et à mesure du temps, on rajoute des données quotidiennes dans notre modélisation, et nous comparons ses résultats aux modélisations précédentes. Lorsque le modèle cesse de s'améliorer du fait de l'introduction d'observations incohérentes, alors cela veut dire que la date de rupture est survenue.

Cette date de rupture peut être déterminée plus précisément à l'aide d'études statistiques, qui ne seront pas menées ici, mais qui pourraient permettre d'estimer la variance de cette date, compte tenu de l'arrêt de l'amélioration du modèle.

Néanmoins, un des défauts majeurs de cette technique est son temps de calcul. En effet, elle nécessite l'entraînement d'autant de modèles que de jours qui se sont écoulés depuis la date de rupture. On peut cependant pallier à ce problème en choisissant par défaut, au début, un grand pas, qui se fait de plus en plus petit au fur et à mesure que l'on recule dans le passé.

Avec cette technique, nous avons obtenu les résultats suivants :

	Date de début du nouveau modèle	RMSE Éch.Entr
BDD 1	D - 335	53.968456
	D - 305	49.52151
	D - 375	46.950863
	D - 245	45.490276
	D - 215	43.48938
BDD 2	D – 335	46.60596
	D – 305	43.57085
	D – 275	44.235332
	D – 245	40.476177
	D - 215	39.581505

On peut donc voir que les performances de cette méthode sont inférieures à celles obtenues par dichotomie. En effet, on ne parvient ni dans le cas de la base de données 1, ni dans celui de la base de données 2, à obtenir une idée de la date de rupture. En effet, l'effet de l'ajout d'observations erronées ne semble pas avoir un effet si impactant que cela sur la création du modèle.

V.3.4 Remarques

Suite à l'étude des résultats obtenus, arrive à la conclusion que la méthode dichotomique semble relativement plus adaptée que celle du 'recul'. Néanmoins aucune des méthodes n'a montré une grande performance. Le problème demeure donc entier, et notre problématique n'a pas pu trouver de résolution.

La démarche pourrait toutefois être encore développée, notamment en introduisant une recherche d'un phénomène non plus définitif mais plutôt saisonnier. Toutefois il faut garder en mémoire que les détections de rupture ont vocation à déceler un changement durable, et non un phénomène exceptionnel court. On pourrait également tester le pouvoir de détection des différentes méthodes sur des changements tarifs plus restreints, pour voir leur capacité à s'adapter à de petites variations.

Ici la problématique s'appliquait dans un cadre statique ou détection hors-ligne (nous disposions de toutes les données), mais la même question pourrait se poser dans un cadre dynamique,

c’est-à-dire un traitement progressif des données au fur et à mesure de leur arrivée dans le système. Un tel traitement permettrait une réactivité d’autant plus rapide de l’assureur dans l’étude de sa politique concurrentielle, puisqu’un changement de tarif pourrait être décelé quasi-instantanément.

On retiendra également que le phénomène de ‘fausse alarme’, c’est-à-dire lorsque l’algorithme détecte une date de rupture qui n’en est pas une, n’est dans notre cas pas très important. Tant que la base de données à utiliser reste d’une taille assez conséquente (plusieurs milliers d’observations), les modèles de machine learning demeurent performants. Si le modèle détecte une fausse alarme au-delà d’une certaine date restreignant la taille de données nécessaires au minimum, la future modélisation n’en sera pas affectée outre mesure.

Conclusion

Les statistiques sont la science de changer son avis sous couvert d'incertitude. Par définition il s'agit donc d'une science subjective ; au départ notre esprit est conditionné par une action par défaut ou une croyance a priori. Le but des statistiques est de provoquer suffisamment de confiance pour que le choix final se fasse de manière plus ou moins certaine. Le côté analytique des statistiques revient à raisonner sur ce que l'on sait au vu des données. Le but de ce mémoire était d'allier ce côté analytique à des tactiques plus poussées pour conforter les croyances.

On a donc vu tout au long de cet ouvrage comment implémenter de manière structurée et efficiente le reverse engineering. Cette étude a développé l'utilisation d'outils dépassant le cadre de la statistique traditionnelle, permettant ainsi d'ouvrir le panel de méthodes pouvant être employées en tarification non-vie. Les GLMs ont ainsi été comparés à de multiples algorithmes, que ce soient des algorithmes basés sur des arbres, ou sur d'autres techniques. La nécessité de sélectionner ses variables en amont avec les modélisations traditionnelles a également été soulignée ; néanmoins, du point de vue l'actuaire tarificateur, cela permet de surveiller ce qu'il se passe à l'intérieur du modèle. Cependant, dans notre optique de réplication tarifaire, cette méthode peut rapidement devenir subsidiaire, selon l'objectif recherché : performance ou interprétation ?

Il a ainsi été mis en exergue que ces techniques issues de l'apprentissage statistique améliorent non seulement les performances prédictives des différentes méthodologies traditionnelles, mais permettent également à l'utilisateur d'appréhender différemment leurs résultats. La perte d'interprétabilité qui peut parfois se manifester peut néanmoins être palliée par l'utilisation de méthodes s'appuyant sur diverses théories, comme celle de la permutation ou celle des jeux.

Néanmoins, comme l'a dit Leo Breiman, l'homme qui a démocratisé l'usage des arbres de régression, le but d'un modèle est d'obtenir de l'information utile sur la relation entre la variable réponse et les variables prédictives ; l'interprétabilité est une manière d'obtenir de l'information, mais un modèle n'a pas à être simple pour fournir de l'information fiable.

Une ouverture quant aux démarches qui auraient pu être également réalisées, se trouve dans l'alliance entre modèles traditionnels et modèles de machine learning. Comme nous l'avons vu dans la dernière partie de ce mémoire, combiner deux de ces méthodes permet d'allier performance et interprétabilité.

Terminons sur une citation du statisticien britannique George Box, titulaire de nombreux prix en statistiques, qui résume parfaitement le bien-fondé de ce mémoire :

"All models are approximations. Assumptions, whether implied or clearly stated, are never exactly true. All models are wrong, but some models are useful. So the question you need to ask is not 'Is the model true?' (it never is) but 'Is the model good enough for this particular application?" - George Box

Annexes

A.1. Opérateurs Proximaux

L'un des avantages des méthodes proximales réside dans leur pouvoir de traiter des problèmes d'optimisation dans lesquels la fonction à minimiser est la somme de deux fonctions dont l'une est 'lisse' (différentiable) et l'autre simplement sous-différentiable mais d'opérateur proximal connu. Cela est notamment le cas de la somme d'un terme quadratique et d'une norme l_1.

On définit l'opérateur proximal d'une fonction $J : \mathbb{R}^n \to \mathbb{R}$ par $\wp_J(u) = \underset{v\in\mathbb{R}^n}{\arg\min}(J(v) + \frac{1}{2}\|v-u\|_2^2)$ où $\|.\|_2$ est la norme usuelle (euclidienne) de $\mathbb{R}^n$.

On souhaite résoudre à l'aide des opérateurs proximaux un problème d'optimisation semblable au suivant :

$$\min_{u\in\mathbb{R}^n} F(u) + G(u)$$

Avec pour hypothèses que F est convexe, différentiable et définie sur tout $\mathbb{R}^n$; G est convexe, semi-continue inférieurement et propre (c'est-à-dire que l'image réciproque par G d'un ensemble compact reste compact). La fonction $\wp_G$ doit également pouvoir se résoudre facilement.

La méthode du gradient proximal consiste à mettre en œuvre l'algorithme suivant, où $\varepsilon^{(k)} > 0$ représente la longueur d'un pas :

$$u^{(k+1)} = \wp_{\varepsilon^{(k)}G}(u^{(k)} - \varepsilon^{(k)}\nabla F(u^{(k)}))$$

Par définition de l'opérateur proximal, la méthode du gradient proximal revient à minimiser la somme de la fonction G et d'une fonction quadratique représentant F au voisinage de $u^{(k)}$:

$$u^{(k+1)} = \underset{u\in\mathbb{R}^n}{\arg\min}(G(u) + \langle\nabla F(u^{(k)}), u-u^{(k)}\rangle + \frac{1}{2\varepsilon^{(k)}}\|u-u^{(k)}\|_2^2)$$

A.2 Lois de la famille exponentielle et correspondance

De nombreuses lois usuellement utilisées en statistiques font partie de la famille exponentielle. La famille exponentielle est une classe de distributions de probabilité dont la forme générale est $f_X(x|\theta) = a(x)\ b(\theta)\ \exp[\eta(\theta).T(x)]$ où X est la variable aléatoire et θ un paramètre. $T(x)$ est une statistique suffisante (ou exhaustive) de la distribution ; cela veut dire que $T(x)$ résume parfaitement les données mesurées x dans le cadre d'un échantillon tiré de cette distribution. Ainsi si un autre jeu de données y diffère complètement de x mais $T(x) = T(y)$ alors la densité estimée à partir des observations sera la même.

Les lois appartenant à cette famille présentent quelques caractéristiques propres, notamment leur reformulation à l'aide de paramètres naturels et leur capacité à obtenir facilement des statistiques suffisantes de la famille, permettant ainsi de résumer un échantillon de données à l'aide d'un nombre réduit de valeurs.

On peut réécrire les lois appartenant à cette famille sous une autre forme, plus intuitive : $f_X(x_i, \theta_i, \varphi, \omega_i) = \exp(\frac{x_i\theta_i - b(\theta_i)}{a(\varphi)}\omega_i + c(x_i, \varphi, \omega_i))$. Sous cette forme, θ_i est le paramètre canonique, fonction de l'espérance $\theta_i = g(\mu_i)$, avec g la fonction lien et $\mu_i = E(X_i)$. Le paramètre de dispersion φ_i est lui supposé connu. w_i est un poids. Néanmoins, on précisera que les fonctions a et b de cette formule ne sont pas les mêmes que dans la formulation précédente.

Ci-dessous, les lois de la famille exponentielle les plus fréquentes et leurs paramètres :

Distribution	$\boldsymbol{\theta(\mu)}$	$\boldsymbol{b(\theta)}$	$\boldsymbol{a(\varphi)}$	$\boldsymbol{E(Y) = b'(\theta)}$	$\boldsymbol{V(Y)}$
Normale $\boldsymbol{N(\mu, \sigma)}$	μ	$\frac{\theta^2}{2}$	σ^2	$\mu = \theta$	σ^2
Bernoulli $\boldsymbol{B(1, \mu)}$	$\log(\frac{\mu}{1-\mu})$	$\log(1 + e^\theta)$	1	$\mu = \frac{e^\theta}{1+e^\theta}$	$\mu(1-\mu)$
Poisson $\boldsymbol{P(\mu)}$	$\log(\mu)$	e^θ	1	$\mu = e^\theta$	μ
Gamma $\boldsymbol{G(\mu, \nu)}$	$-\frac{1}{\mu}$	$-\log(-\theta)$	$\frac{1}{\nu}$	$\mu = -\frac{1}{\theta}$	$\frac{\mu^2}{\nu}$
Gauss Inv $\boldsymbol{IG(\mu, \sigma^2)}$	$-\frac{1}{2\mu^2}$	$-\sqrt{-2\theta}$	σ^2	$\mu = \frac{1}{\sqrt{-2\theta}}$	$\mu^3\sigma^2$

D'autres lois appartiennent également à la famille exponentielle, on citera notamment de manière non-exhaustive : la loi binomiale, la loi exponentielle, la loi de Pareto, la loi de Weibull, la loi de Laplace, la loi du X^2, la loi log-normale, la loi Gamma inverse, la loi multinomiale…

A.3 ADMM

L'algorithme des directions alternées (ou ADMM pour Alternating Direction Method of Multipliers) est un algorithme de résolution de problèmes d'optimisation décomposables. Cet algorithme se base sur l'algorithme du lagrangien augmenté. Il est typiquement utilisé pour minimiser la somme de deux fonctions dépendant de variables différentes couplées par une contrainte affine :

$$\begin{cases} \inf\limits_{(x,y)} f(x) + g(y) \\ Ax + By = c \end{cases}, \text{où } f: \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\} \; et \; g: \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$$

On a également $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$ et $c \in \mathbb{R}^p$.

Le lagrangien augmenté associé à ce problème est la fonction $l_t: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$, dont la valeur en $x, y \; et \; \lambda$ s'écrit comme ci-dessous, avec r le paramètre d'augmentation (si $r = 0$ on retrouve un Lagrangien classique) :

$$l_t(x, y, \lambda) = f(x) + g(y) + \lambda^T(Ax + By - c) + \frac{r}{2}\|Ax + By - c\|_2^2$$

L'inconvénient de l'algorithme du lagrangien augmenté est que pour minimiser le lagrangien augmenté, il faut le minimiser simultanément en x et en y.

En effet pour obtenir le triplet $(x_{k+1}, y_{k+1}, \lambda_{k+1})$ il faut tout d'abord obtenir (x_{k+1}, y_{k+1}) tel que $(x_{k+1}, y_{k+1}) \in \underset{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m}{\arg\min} l_t(x, y, \lambda_k)$. Ensuite on détermine $\lambda_{k+1} = \lambda_k + r_k(Ax_{k+1} + By_{k+1} - c)$.

L'algorithme des directions alternées résout le problème en minimisant successivement en x et en y. On a alors :

$$x_{k+1} \in \underset{x \in \mathbb{R}^n}{\arg\min}\, l_t(x, y_k, \lambda_k) \quad et \quad y_{k+1} \in \underset{y \in \mathbb{R}^m}{\arg\min}\, l_t(x_{k+1}, y, \lambda_k)$$

On obtient ensuite λ_{k+1} par la même formule que précédemment : $\lambda_{k+1} = \lambda_k + r_k(Ax_{k+1} + By_{k+1} - c)$.

L'algorithme des directions alternées est généralement un algorithme lent si l'on désire une solution précise (on préfèrera alors utiliser des algorithmes tenant compte des dérivées secondes ou d'une approximation de ces dernières).

A.4 Tuning des hyper-paramètres

Dans ce mémoire, nous avons traité de nombreux modèles de machine learning. Dans l'optique de recherche du meilleur modèle, nous avons raffiné chacun des modèles que nous avons produits. Pour cela, nous avons parcouru un large nombre de combinaisons de paramètres pour chaque algorithme. Cette partie vise à lister les différents paramètres qui ont pu être raffinés durant cette phase de tuning, ainsi que la plage de valeurs qu'ils ont pu adopter et la raison de leur modification.

Paramètre	Plage de valeur	Explication
Nombre d'arbres	100 ; 300 ;500 ; 750	Réduire le nombre d'arbres permet d'améliorer la vitesse de l'algorithme ; mais on risque alors d'arrêter le modèle avant qu'il n'ait pu nous livrer tous ses secrets
Profondeur maximum	3 ; 5 ; 7 ; 9 ; 13 GBM / DRF 5 ; 10 ; 15 ; 25	Spécifie la profondeur maximale de l'arbre. Pour les algorithmes comme le GBM, ce nombre est généralement bas. Si la profondeur de l'arbre est trop grande alors le modèle risque d'être sur-appris
Nb min obs feuille	50 ; 500 ; 1500	Plus ce nombre est bas plus l'algorithme aura tendance à aller profondément dans l'arbre et à créer des groupes 'petits', pouvant amener un sur-apprentissage et une mauvaise généralisation de l'algorithme
Nb cases (numérique)	3 ; 7 ; 12 ; 18	Le nombre de cases est défini à chaque split et non globalement. Augmenter le nombre de cases permet de rendre un modèle plus spécifique ; le baisser revient à améliorer sa généralisation
Nb cases (catégoriel)	3 ; 7 ; 12 ; 18	Même chose que pour *Nb cases (numérique)*
Taux d'échantillonnage des observations	0.5 ; 0.7 ; 0 .8 ; 0.9 ; 1	Prendre un échantillon des observations à chaque split permet d'améliorer la pertinence du modèle sur l'échantillon de test ; cela améliore la généralisation du modèle
Taux d'échantillonnage des variables	0.5 ; 0.7 ; 0 .8 ; 0.9 ; 1	Prendre un échantillon des variables à chaque split permet d'améliorer la pertinence du modèle sur l'échantillon de test ; cela améliore la généralisation du modèle et permet de limiter le risque de se retrouve dans un minimum local
Type d'encodage des variables catégorielles	One Hot ; LabelEncoder	Avec One Hot, l'algorithme crée une nouvelle colonne pour chaque modalité présentes par variable et l'encode par 0 ou 1. Avec Label Encoder, chaque modalité reçoit un entier correspondant ; cela permet de limiter le nombre de colonnes créées
Valeur minimale d'amélioration	0.00001 ; 0.0001 ; 0.001 ; 0.01	Permet de limiter le suraprentissage avec des valeurs grandes ; néanmoins l'algorithme s'arrêtera plutôt avec de telles valeurs et peut donc ne pas livrer toute l'information disponible
Type d'histogramme utilisé pour caser	Uniforme ; Quantile ; RoundRobin	Impacte la sélection du point de split optimal. Uniforme crée des intervalles de taille égale, ce qui peut poser problèmes si les données sont biaisées. Quantile crée des intervalles où les populations sont également distribuées. RoundRobin utilise successivement sur chaque arbre, les techniques Uniforme, Quantile et Aléatoire (split réalisés aléatoirement)
Fonction de perte	Huber ; Quadratic	La fonction de perte Huber permet de combiner la fonction de perte quadratique et la fonction de perte absolue ; la première est appliquée sur 90% des erreurs les plus petites, et la deuxième sur les 10% les plus larges
Taux d'apprentissage	0.001 ; 0.005 ; 0.01 ; 0.05	Généralement les meilleurs modèles sont construits avec les taux d'apprentissage les plus faibles, mais alors plus d'arbres sont requis pour obtenir un même niveau de fit. Cela permet de l'imiter le surapprentissage
Taux de réduction	0.95 ; 0.99	Permet d'accélérer la convergence de l'algorithme en réduisant progressivement le taux d'apprentissage
Fonction d'activation	Tanh ; Rectifier ; Maxout	La fonction d'activation influence grandement les décisions prises par le réseau de neurones ; il est donc important de voir laquelle correspond le mieux à notre problématique
Nb couches cachées	1 ; 3 .; 5 ; 10	Spécifie le nombre de couches cachées : plus ce nombre est grand plus il y a de chances que l'algorithme comprenne toutes les subtilités du modèle ; mais le risque de sur-apprentissage est également plus important
Nb epochs	1 ; 3 ; 5 ; 10	Une époche est réalisée lorsque le dataset entier est passé une fois (en avant et en arrière) à travers le réseau de neurones. Plus on augmente le nombre d'époches, plus on recalcule de poids, et plus on a des chances d'obtenir les poids les plus proches de la réalité. Toutefois on peut faire face à un phénomène de sur-apprentissage

Ci-dessous un tableau récapitulant la liste des paramètres qui ont été tunés pour chacun des algorithmes :

Paramètre	Arbre de régression	Forêt aléatoire	Bagging	GBM	Réseau de neurones
Nombre d'arbres		X	X	X	
Profondeur maximum	X	X	X	X	
Nb min obs feuille	X	X	X	X	
Nb cases (numérique)	X	X		X	
Nb cases (catégoriel)	X	X		X	
Taux d'échantillonnage des observations	X	X		X	
Taux d'échantillonnage des variables	X	X		X	
Type d'encodage des variables catégorielles		X		X	X
Valeur minimale d'amélioration		X		X	
Type d'histogramme utilisé pour caser		X		X	
Fonction de perte				X	X
Taux d'apprentissage				X	X
Taux de réduction				X	X
Fonction d'activation					X
Nb couches cachées					X
Nb epochs					X

On notera également que dans les dernières années, une nouvelle fonction de perte a été développée sur le modèle d'un apprentissage sensible au coût ; cela signifie que lors du calcul de l'erreur, l'algorithme n'accorde pas autant de poids aux erreurs positives que négatives. En effet, dans la réalité il est rare que le coût effectif des erreurs soit asymétrique. Dans notre cas, il est toutefois difficile de décider si sur-estimer ou sous-estimer la prime est plus ou moins néfaste. Le lecteur intéressé pourra toutefois trouver plus d'informations sur ce sujet dans l'article 'An extended tuning method for cost-sensitive regression and forecasting' par Zhao, Sinha et Bansal.

C'est pourquoi nous nous sommes restreints dans nos modèles à l'utilisation de fonction de perte telles que l'erreur quadratique ou la fonction de perte d'Huber.

A.5 Intervalle autour des paramètres (GLM)

Avec les méthodes statistiques on peut souvent distinguer les erreurs d'estimations des erreurs de prédiction. Dans cette partie nous nous attarderons sur l'erreur d'estimation commise.

Les GLM sont des modèles paramétriques estimés par maximum de vraisemblance. En utilisant cette méthode, on dispose de résultats de convergence qui fournissent une information sur l'écart entre l'estimateur et la vraie valeur :

$$\sqrt{n}\left(\widehat{\beta_n} - \beta\right) \xrightarrow{Loi} \mathcal{N}(0, I(\beta)^{-1})$$

On dispose alors également d'un estimateur de la variance asymptotique :

$$\widehat{Var}\left(\hat{\beta}\right) = -\left[\frac{\partial^2 \ln L\left(y|x;\hat{\beta}\right)}{\partial\beta\partial\beta'}\right]^{-1}$$

Exemple d'intervalles de confiance autour de quelques variables sélectionnées par le GLM optimal.

	Lower bound 2.5 %	Upper bound à 97.5 %
(Intercept)	2.894537e+02	3.113286e+02
driver_age	-4.994067e+00	-4.764939e+00
sex2	-9.461298e+00	-5.191163e+00
license_type1	-1.880919e+02	7.072066e+01
license_type2	-9.140903e+01	1.669542e+02
license_type3	-1.524315e+01	8.608927e+00
license_type4	-9.140903e+01	1.669542e+02
license_type5	-9.140903e+01	1.669542e+02
marital_status2	-2.081469e+01	2.137347e+00
marital_status3	-4.172613e+01	2.753479e+01
marital_status5	-4.434718e+00	6.271531e+00
marital_status6	-1.087510e+01	3.773659e+00
province1	3.756291e+01	4.752351e+01
province2	-6.881797e+00	-6.508964e-02
province3	3.135347e+01	3.995136e+01
province4	5.622932e+01	6.442057e+01

Bibliographie

ARGYRIOU A., FOYGEL R. & SREBRO N., [2012], *Sparse Prediction with the k-support norm*

ASSURALIA

AWAD M. & KHANNA R., [2015], *Support Vector Regression,* Efficient Learning Machines, pp 67-80

BELLINA R., [2014], *Méthodes d'apprentissage appliquées à la tarification non-vie*

BOYD S. & VANDENBERGHE L., [2004], *Convex Optimization*

BREIMAN L., [1996], *Bagging predictors,* Machine Learning, 8(2), 209-218

BREIMAN L., [2001], *Random forests,* Machine Learning, 45(1), 5-32

BREIMAN L. & FRIEDMAN J. & STONE C.J. & OLHSEN R.A., [1984], *Classification and Regression Trees,* The Wadsworth and Brooks-Cole statistics-probability series, Taylor & Francis

BÜLHMANN P., [2019], *Causality – In a wide sense,* Seminar for Statistics

CARPENTIER P., [2019], *Méthodes proximales*

CHARPENTIER A., BARRY L., [2020], *Personalization as a promise: Can Big Data change the practice of insurance?,* Big Data & Society

DEVRIENDT S. & ANTONIO K. & REYNKENS T. & VERBELEN R., [2018], *Sparse Regression with Multi-type Regularized Feature Modeling*

DUGAS C. & BENGIO Y. & CHAPADOS N. & VINCENT P. & DENONCOURT G. & FOURNIER C., [2003], *Statistical Learning Algorithms Applied to Automobile Insurance Ratemaking*

EFRON B., HASTIE T., JOHNSTONE I. & TIBSHIRANI R., [2004], *Least Angle Regression (LARS),* The Annals of Statistics, 32(2), 407-499

FRIEDMAN J., [1991], *Multivariate adaptive regression splines,* The Annals of Statistics, 19(1), 1-67

FRIEDMAN J., [2001], *Greedy function approximation : A gradient boosting machine*

FRIEDMAN J. & POPESCU B., [2008], *Predictive learning via rule ensembles*

GELMAN A. & HILL J., [2012], *Missing-data imputation,* Data Analysis Using Regression and Multilevel/Hierarchical Models, pp 529-544

GRENWELL B. & BOEHMKE B. & MCCARTHY A., [2018], *A Simple and Effective Model-Based Variable Importance Measure*

HASTIER T. & TIBSHIRANI R. & FRIEDMAN J., [2009], *The elements of statistical learning : data mining, inference and prediction,* Springer, 2nd edition

HAURY A-C., [2012], *Feature selection from gene expression data: molecular signatures for breast cancer prognosis and gene regulation network inference*

HENCKAERTS R. & CÔTÉ M-P. & ANTONIO K. & VERBELEN R., [2019], *Boosting insights in insurance tariff plans with tree-based machine learning methods*

HILL C. & GRIFFITHS W. & LIM G., [2011], *Principles of Econometrics, Fourth edition*

LINUSSON H., [2017], *An introduction to conformal prediction*

LIPTON Z., [2019], *The Mythos of Model Interpretability*

LITTLE R. & RUBIN D., [2019], *Statistical Analysis with Missing Data, 3rd edition*

MAYO M., [2017], *The current state of automated machine learning*

MCDONALD A. & PONTIL M. & STAMOS D., [2014], *Spectral k-support Norm Regularization,* Advances in Neural Information Processing System 27

MOLNAR C., [2019], *Interpretable Machine Learning : A Guide for Making Black Box Models Explainable*

MOLNAR C. & KÖNIG G. & HERBINGER J. & FREIESLEBEN T. & DANDL S. & SCHOLBECK C. & CASLICCHIO G. & GROSSE-WENTRUP M. & BISCHL B., [2020], *Pitfalls to Avoid when Interpreting Machine Learning Models*

NATEKIN A. & KNOLL A., [2013], *Gradient boosting machines, a tutorial*

OELKER M. & TUTZ G., [2017], *A uniform framework for the combination of penalties in generalized structure models,* Advances in Data Analysis and Classification (11), 97-120

OHLSSON E. & JOHANSSON B., [2010], *Non-life Insurance Pricing with Generalized Linear Models*, EAA Series, Springer Berlin Heidelberg

PARIENTE J., [2017], *Modélisation du risqué géographique en assurance habitation*

PAPADOPOULOS H., [2004], *Qualified predictions for large datasets*

PLANCHET F., [2017], *Tarification IARD : Introduction aux techniques avancées*

RIBEIRO M. & SINGH S. & GUESTRIN C., [2016], *Why should I trust you? Explaining the predictions of any classifier,* Proceedings of the 22nd ACM SIGKDD International Conference, pp 1135-1144

SHAFER G., GAMMERMAN A. & VOVK V., [2005], *Algorithmic Learning in a random world,* New York Springer

SHAFER G. & VOVK V., [2008], *A tutorial on conformal prediction,* The Journal of Machine Learning Research, vol.9, pp 371-421

THOMAS C., [2017], *Sélection de variables*

TIBSHIRANI R., [1994], *Regression shrinkage and selection via the LASSO,* Journal of the Royal Statistical Society, Series B, pp 267-288

TUFFERY S., [2019], *Améliorer les performances d'un modèle prédictif : perspectives et réalité*

VAN DER LAAN M., POLLEY E. & HUBBARD A., [2007], *Super Learner,* Statistical Applications in Genetics and Molecular Biology 6(1) : Article 25

WOLPERT D.H., [1992], *Stacked Generalization,* Neural Networks, 5(2), pp 241-259

WU J. & ROY J. & STERWART W., [2010], *Prediction modeling using ehr data: challenges, strategies, and a comparison of machine learning approaches,* Medical Care, pp S106-S113

ZHAO H. & SINHA A. & BANSAL G., [2011], *An extended tuning method for cost-sensitive regression and forecasting*

Table des figures

Figure 1 : Benchmark des performances des différents modèles 5
Figure 2 : Performance Benchmark of the different models 8
Figure 3: Répartition des différents assureurs dans la BDD 8
Figure 4: Réparition des assureurs selon le nombre de profils 23
Figure 5: Décomposition tarifaire d'une prime commerciale 26
Figure 6 : Statistiques de la prime par assureur 26

Figure 7 : Fréquence du profil par âge du conducteur 29
Figure 8 : Prime moyenne en fonction de l'âge du conducteur 30
Figure 9 : Prime moyenne en fonction de la puissance 30
Figure 10 : Distribution de la prime selon présence d'un avertisseur d'angle mort 33
Figure 11 : Distribution de la prime selon présence d'un détecteur de fatigue 33
Figure 12 : Distribution de la prime selon présence d'un régulateur de vitesse 34
Figure 13 : Distribution de la prime selon le nombre de dispositifs d'aide à la conduite 34
Figure 14 : Valeur assurée selon le nombre de dispositifs d'aide à la conduite 34
Figure 15 : Répartition des profils selon catégorie professionnelle 35
Figure 16 : Prime selon selon catégorie professionnelle 35
Figure 17 : Graphe de la corrélation de Pearson sur les variables numériques 37
Figure 18 : Graphe de la corrélation du X^2 de Pearson sur les variables catégorielles 38
Figure 19 : Graphe du V de Cramer sur les variables catégorielles 39
Figure 20 : Graphe du V de Cramer sur toutes les variables 40
Figure 21 : Répartition des primes (Assureur 8) 42
Figure 22 : Répartition de ln(Prime) (Assureur 8) 42
Figure 23 : Empirique vs Statistique (Prime) 42
Figure 24 : Empirique vs Statistiqu (ln(Prime)) 42

Figure 25 : Coefficients obtenus par GLM 51
Figure 26: Importance des variables par forêt aléatoire 57
Figure 27: Importance des variables par GBM 57
Figure 28: Importance des variables par réseau de neurones 58
Figure 29: Importance des variables par réseau de neurones (15) 58
Figure 30: Evolution de la MSE selon la valeur de ln(lambda) 64
Figure 31 : Statistique H de Friedman pour toutes les variables 70
Figure 32: Statistique H de Friedman pour toutes les variables, corrélées avec la valeur assurée 71
Figure 33 : Graphe d'importance par permutation par GBM 72
Figure 34 : Pourcentage de variance expliquée selon l'ajout progressif de dimension 74
Figure 35 : ACP réalisée sur les variables numériques 74
Figure 36 : Tableau représentant la contribution de chaque variable numérique dans chaque dimension (5 premières) 75

Figure 37 : Compromis Biais / Variance 81
Figure 38 : Arbre de régression réalisée sur les données (profondeur = 3) 92
Figure 39 : Benchmark des performances (RMSE) des différents modèles 101

Figure 40: Intervalle de prédictionà 95% en fonction de la prime 107
Figure 41 : Distribution de la taille des intervalles, issus de la méthode conformal predictions 112
Figure 42: Tarification d'un assuré, avec intervalles de confaince autour des valeurs 112
Figure 43 : Effect plot des variables et de leur modalié (GLM) 114
Figure 44: Variable importance par GBM 115
Figure 45 : Variable importance par forêt aléatoire 115
Figure 46 : Prime moyenne en fontion de l'âge du permis selon différents sets de training 115
Figure 47 : Distance de Cook 118
Figure 48 : Graphe DFFIT 118
Figure 49 : ICE Plot par âge du conducteur ..... .119
Figure 50 : ICE Plot par valeur assurée 119
Figure 51 : PDP à deux dimensions sur l'âge du conducteur et la valeur assurée 120
Figure 52: Graphe ALE sur l'âge du conducteur 121
Figure 53 : ALE Plot sur une variable catégorielle, la province 122
Figure 54 : ALE Plot à deux dimensions sur l'âge du conducteur et la valeur assurée 122
Figure 55 : Valeurs de Shapley sur un prospect donné 123
Figure 56 : Graphe LIME pour un prospect donné (même que pour valeurs de Shapley) 126