UCLouvain

FACULTE DES SCIENCES

Faculté des sciences

AGLM : Une méthode de modélisation hybride combinant GLM et techniques de science des données

Auteur: **Caroline LENGELÉ**
Promoteur: **Michel DENUIT**
Lecteur: **Florian PECHON**
Année académique 2023–2024
Master [120] en sciences actuarielles

Résumé

Ces dernières années ont été marquées par des avancées technologiques en science des données. L'actuaire doit apprendre à les utiliser intelligemment et à trouver un moyen pour les intégrer dans le processus de modélisation prédictive. Les modèles les plus simplistes sont encore les plus répandus grâce à leur propriété d'interprétabilité. Cette caractéristique est essentielle dans le secteur de l'assurance où la transparence est exigée par les normes réglementaires. Toutefois, la précision de la prédiction devrait être tout aussi importante que l'interprétabilité du modèle. Il faut réussir à trouver un compromis efficace entre une forte précision prédictive et une bonne interprétabilité. L'idée de ce mémoire est de développer la méthodologie des Accurate GLM. C'est une méthode qui évolue dans le cadre des GLM et qui, en combinant avec des techniques de science des données, cherche à fournir une meilleure prédiction. Une fois les concepts théoriques étudiés, nous appliquerons cette méthode sur une base de données réelle et nous la comparerons aux méthodes classiques.

Summary

Recent years have been marked by technological advances in data science. Actuaries must learn to use them efficiently and find effective ways to incorporate them into predictive modeling practices. Simpler models remain widely used, primarily due to their interpretability. This characteristic is essential in the insurance sector, where transparency is required by regulatory standards. predictive accuracy is just as critical as model interpretability. This thesis aims to develop the Accurate GLM methodology, a model that builds upon the conventional GLM framework while integrating data science techniques to enhance predictive performance. After exploring the theoretical foundations, we will apply this approach to a real dataset and evaluate its effectiveness against traditional methods.

Remerciement

Je tiens tout d'abord à remercier mon promoteur, le Professeur Michel Denuit, pour son écoute et sa disponibilité tout au long de la rédaction de ce mémoire. Ces conseils m'ont permis d'améliorer mon travail.

Je remercie tous les professeurs que j'ai eus la chance d'écouter au cours de ces années de master. Ils m'ont fait réfléchir et m'ont donné l'envie d'approfondir mes connaissances actuarielles.

Je remercie ma famille de m'avoir soutenue durant mes études, et en particulier durant ces années de master. Ils m'ont toujours encouragée à faire de mon mieux et à ne rien lâcher.

Je remercie mes amis et l'ensemble des personnes qui m'ont soutenue tout au long de mes études universitaires. C'est grâce à ce soutien sans faille que j'ai pu mener à bien ce projet.

Je remercie également toutes les personnes qui ont contribué, de près ou de loin, à ce travail de longue haleine. Je suis particulièrement reconnaissante envers celles qui ont acceptées de relire mon mémoire.

Table des matières

Introduction

Dans le secteur des assurances, et en particulier en assurance non-vie, le rôle de l'assureur est de fournir une prestation au preneur d'assurance, moyennant le payement d'une prime, lorsqu'un évènement incertain se produit. Contrairement au cycle classique de production, le coût pour l'assureur n'est pas connu lors de la vente du contrat d'assurance. L'actuaire a donc pour objectif la tarification de ces polices d'assurance. Afin de déterminer le montant adéquat de la prime, des méthodes de modélisations sont utilisées pour prédire la fréquence des sinistres ainsi que leurs coûts.

Les méthodes initialement utilisées sont les modèles linéaires généralisés. Ils permettent de modéliser la relation existante entre une variable réponse et des variables explicatives afin de prédire la réponse. La structure simple de leur score permet au modèle d'être facilement interprétable. La performance du modèle n'est pas toujours idéale lorsque des relations complexes entre les variables apparaissent. Les modèles additifs généralisés ont été mis en place pour garder l'interprétabilité du GLM tout en améliorant la précision de prédiction. Les dernières années ont néanmoins été marquées par une avancée technologique importante et une augmentation considérable des informations disponibles. Le machine learning s'est alors développé et a permis la création de nouvelles méthodes de modélisation telles que le Gradient Boosting et les réseaux neurones. Ces méthodes sont flexibles et capables de capturer des relations complexes entre les variables ce qui permet au modèle d'atteindre une performance élevée. Elles sont cependant difficiles à interpréter et considérées comme des boites noires.

L'idée de développer une méthode de modélisation qui serait un entre-deux des méthodes précédentes est apparue. En effet, la précision prédictive d'un modèle est un point essentiel dans l'analyse des risques d'assurance. Par ailleurs, le modèle sélectionné doit respecter des normes et rester suffisamment simple afin que les choix puissent être justifiés auprès des institutions réglementaires et des utilisateurs. Le Accurate Generalized Linear Model a été imaginé afin d'atteindre une grande précision de prédiction tout en gardant un niveau d'interprètabilité important.

Dans un premier temps, ce mémoire a pour objectif de développer le modèle AGLM d'un point de vue méthodologique. Pour ce faire, une brève remise en contexte de la structure du GLM sera abordée et une explication détaillée des méthodes de régularisation sera fournie. Ensuite, la structure d'un AGLM ainsi que les techniques de science de données utilisées seront développées. À savoir, les différents types de codage disponibles selon la nature des variables explicatives. Pour finir, une comparaison qualitative entre les AGLM et les autres méthodes de modélisation classiques sera établie. La deuxième partie de ce mémoire consistera en une application numérique sur un jeu de données concret. Le modèle AGLM sera utilisé pour de prédire la fréquence des sinistres. Une comparaison quantitative avec les autres modèles permettra d'évaluer les points forts des AGLM. Finalement, nous évoquerons brièvement comment appliquer cette méthode dans le cadre d'un modèle des sévérités.

Chapitre 1

Structure des GLM et méthodes de régularisation

Les AGLM sont fondés sur le modèle linéaire généralisé, ce qui garantit un niveau d'interprétabilité nécessaire. Pour maintenir une précision prédictive élevée, diverses techniques sont intégrées aux GLM. Celles-ci incluent la discrétisation, la transformation des variables explicatives via un codage, ainsi qu'une méthode de régularisation. En combinant ces approches de machine learning, les AGLM parviennent à offrir une interprétabilité similaire aux GLM, facilitant ainsi leur explication même pour ceux qui ne sont pas experts dans le domaine. De plus, ils permettent d'améliorer la précision des prédictions. C'est pourquoi les actuaires Fujita, Tanaka, Kondo et Iwasawa ayant développé cette nouvelle méthode l'ont appelée "Accurate GLM".

Avant d'entrer dans la formalisation des AGLM, il est essentiel de revenir brièvement sur le modèle linéaire généralisé et les méthodes utilisées pour estimer ses paramètres. Ensuite, nous nous pencherons plus en détail sur les différentes techniques de régularisation, en examinant attentivement leurs avantages et leurs inconvénients, qui joueront un rôle clé dans la définition des AGLM.

1.1 Modèle Linéaire Généralisé :

Le modèle GLM est une généralisation de la régression linéaire gaussienne. Le but est d'utiliser des variables explicatives afin d'établir une relation linéaire entre ces variables explicatives et une variable expliquée, également nommée la variable réponse. Ainsi, en intégrant des données futures dans le modèle, cela permet d'obtenir une prédiction de la variable réponse.

Considérons les éléments suivants :
- Y_i avec $i \in \{1,...,n\}$ la variable réponse pour chaque individu i
- $x_i = (1, x_{i1}, x_{i2}, ..., x_{ir})$ avec $i \in \{1,...,n\}$ l'ensemble des variables explicatives pour chaque individu i.
- $\beta = (\beta_0, \beta_1, \beta_2, ..., \beta_r)$ les coefficients de la régression linéaire avec β_0 l'intercept

avec n le nombre d'observations du jeu de données et r le nombre de variables explicatives. Par définition d'une régression linéaire, la variable réponse Y_i satisfait à l'équation suivante :

$$Y_i = \beta_0 + \sum_{j=1}^{r} x_{i,j}\beta_j + \varepsilon_i, \text{ for } i \in \{1,...,n\}$$

où ε_i est un terme d'erreur de loi Normale centrée réduite. Notons que les modèles de régression linéaire ne s'appliquent que sous certaines conditions, notamment la continuité de la variable réponse, l'homoscédasticité des variances et la normalité du terme d'erreur. Cependant, il est fréquent de faire face à d'autres types de variable réponse. Par exemple, les variables binaires ou les variables de comptage que nous retrouvons régulièrement dans le secteur de l'assurance. Le modèle linéaire généralisé a donc été mis en place pour étendre les conditions d'application assez restrictives de la régression linéaire simple.

La distribution de la variable réponse Y est choisie parmi la famille Exponentielle de dispersion. Pour appartenir à cette famille, la fonction de masse d'une loi discrète ou la fonction de densité d'une loi continue doit être telle que

$$\left.\begin{matrix} P[Y=y] \\ f_Y(y) \end{matrix}\right\} = exp(\frac{y\theta - a(\theta)}{\phi/\mu})\ c(y,\phi/\mu) \tag{1.1}$$

où θ, ϕ et $a(.)$ sont des paramètres qui dépendent de la distribution choisie. Cette famille contient les distributions classiques utilisées pour définir la variable réponse dans le secteur de l'assurance. La table 1.1 fournit, pour des exemples de distributions couramment utilisées appartenant à la famille ED, le paramètre canonique θ, la fonction cumulante $a(.)$ et le paramètre de dispersion ϕ utilisés dans la formule (1.1).

Distribution	θ	ϕ	$a(.)$	μ	$Var[Y]$
Poi(μ)	$\ln(\mu)$	1	$\exp(\theta)$	μ	μ
Bin(m,q)	$\ln(\frac{q}{1-q})$	1	$m\ \ln(1+\exp(\theta))$	mq	$\mu(1-\frac{\mu}{m})$
Nor(μ,σ^2)	μ	σ^2	$\frac{\theta^2}{2}$	μ	1
Exp(μ)	$\frac{-1}{\mu}$	1	$-ln(-\theta)$	μ	μ^2
Gam(μ,α)	$\frac{-1}{\mu}$	$\frac{1}{\alpha}$	$-ln(-\theta)$	μ	μ^2

Table 1.1. Exemples de distribution appartenant à la famille ED et leurs paramètres

Le modèle linéaire généralisé permet d'obtenir une relation linéaire entre le score, i.e. une version transformée de la moyenne μ_i de la variable réponse Y_i, et les variables explicatives. Le score de la variable réponse dans un modèle linéaire généralisé est défini comme

$$score_i = \beta_0 + \sum_{j=1}^{r} x_{i,j}\beta_j, \text{ for } i \in \{1,...,n\}$$

Afin d'obtenir la moyenne μ_i de la variable réponse Y_i, l'actuaire doit définir une fonction lien g monotone et différentiable. Par exemple, la fonction logarithme pour les variables de comptage ou la fonction logit pour les variables binaires. Remarquons que prendre la fonction identité comme fonction lien revient à faire une régression linéaire simple. Cette fonction lien satisfait la relation suivante

$$g(\mu_i) = score_i = \beta_0 + \sum_{j=1}^{r} x_{i,j}\beta_j$$

$$\Leftrightarrow\ \mu_i = g^{-1}(\beta_0 + \sum_{j=1}^{r} x_{i,j}\beta_j)$$

1.2 Estimation des paramètres :

Les paramètres β_j du modèle sont estimés à l'aide de la méthode du maximum de vraisemblance. Cette méthode a pour objectif de fournir les estimations des paramètres qui nous donneront les valeurs observées les plus probables pour la variable réponse Y_i.

Considérons le vecteur aléatoire $Y = (Y_1, Y_2, ..., Y_n)$ de nos variables réponses. La fonction de vraisemblance $L(y_1, ..., y_n; \beta)$ correspond à la fonction de densité jointe des différentes variables aléatoires Y_i du modèle supposées indépendantes, i.e. le produit des fonctions de densité, et est donnée par

$$L(y_1, ..., y_n; \beta) = \Pi_{i=1}^{n} \, f_i(y_i)$$

où $f_i(y_i)$ est la fonction de densité et y_i les valeurs observées de la variable Y_i pour $i \in \{1, ..., n\}$. Si la distribution de la variable réponse est discrète, par exemple la loi de Poisson, la fonction de densité est remplacée la fonction de masse $P[Y_i = y_i]$. Dans le cadre du modèle linéaire généralisé, la distribution des variables réponse est incluse dans la famille Exponentielle de dispersion. Ainsi, la fonction de vraisemblance peut se réécrire comme

$$L(y_1, ..., y_n; \beta) = \Pi_{i=1}^{n} exp(\frac{y_i\theta_i - a(\theta_i)}{\phi/\nu_i}) \; c(y_i, \phi/\nu_i).$$

Après avoir obtenu la fonction de vraisemblance, il suffit de la dériver par rapport au paramètre β du modèle et regarder pour quelles valeurs du paramètre cette dérivée s'annule. Le vecteur des estimations par maximum de vraisemblance est donc obtenu en résolvant les équations suivantes

$$\frac{\partial L(y_1, ..., y_n; \beta)}{\partial \beta_j} = 0 \quad \text{for } j \in \{1, ..., r\}$$

Par ailleurs, remarquons que maximiser une fonction revient à minimiser son opposé. Il est également fréquent de transformer la fonction de vraisemblance en une log-vraisemblance de sorte que le produit des fonctions de densité soit converti en une somme. En réécrivant les équations précédentes, cela nous donne

$$\frac{\partial(-log(L(y_1, ..., y_n; \beta)))}{\partial \beta_j} = 0 \quad \text{for } j \in \{1, ..., r\}$$

Dans le cadre du modèle linéaire généralisé, cette équation devient

$$\begin{aligned}
& -\sum_{i=1}^{n} \frac{\partial}{\partial \beta_j} log(exp(\frac{y_i\theta_i - a(\theta_i)}{\phi/\nu_i}) \; c(y_i, \phi/\nu_i)) = 0 \\
\Leftrightarrow & -\sum_{i=1}^{n} \frac{\partial}{\partial \beta_j} (\frac{y_i\theta_i - a(\theta_i)}{\phi/\nu_i}) \; + \underbrace{\frac{\partial}{\partial \beta_j} log(c(y_i, \phi/\nu_i))}_{= 0 \; (constante)} = 0 \\
\Leftrightarrow & \sum_{i=1}^{n} \frac{(y_i - \mu_i) x_{ij}}{V(\mu_i)\nu_i} \frac{\partial \mu_i}{\partial s_i} = 0
\end{aligned}$$

avec $s_i = score_i$ et $\theta_i = g^{-1}(score_i)$.

Ce système d'équations se résout numériquement à l'aide d'algorithme itératif comme la méthode des moindres carrés itératifs re-pondérés. À chaque étape, on initialise nos paramètres et on ajuste la

variable réponse ainsi que les poids de nos variables explicatives via les moindres carrés pondérés. L'algorithme prend fin lorsque les règles d'arrêt sont satisfaites. Par exemple, lorsque la différence entre les estimations de l'itération l et $l+1$ devient infime. Sous forme matricielle, les paramètres de la régression à l'étape itérative $l+1$ sont obtenus par

$$\hat{\beta}^{l+1} = (X^T W^{(l)} X)^{-1} X^T W^{(l)} z^{(l)} \tag{1.2}$$

avec X la matrice contenant les variables explicatives dont la première colonne est reliée à l'intercept et donc constituée uniquement de 1. La matrice $W^{(l)}$ est une matrice diagonale où le $i$ème élément coïncide au poids v_i, c'est à dire

$$W^l = \begin{pmatrix} v_1^l & 0 & 0 & ... & 0 \\ 0 & v_2^l & 0 & ... & 0 \\ 0 & 0 & v_3^l & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & ... & v_n^l \end{pmatrix}$$

et $Z^{(l)}$ contient les réponses transformées obtenues à l'itération précédente également appelées réponses de travail. Elles sont obtenues en faisant une approximation linéaire locale de $g(y_i)$ à l'aide d'un développement de Taylor du premier degré, à savoir

$$z_i = g(y_i) \approx g(\mu_i) + (y_i - \mu_i) g'(\mu_i).$$

L'estimation du vecteur des paramètres β s'obtient donc à l'aide du problème de minimisation suivant

$$\hat{\beta} = \min_{\beta}(-log(L(y_1, ..., y_n; \beta))). \tag{1.3}$$

1.3 Régularisation :

Aujourd'hui, grâce aux avancées technologiques, nous sommes inondés d'une multitude de données diverses. Dans le domaine de l'assurance, cela se traduit par la disponibilité de grandes bases de données contenant de nombreuses variables explicatives. Les GLM restent efficaces sur des bases de données contenant un vaste nombre d'observations, ils ont cependant des difficultés à performer lorsque le nombre de variables explicatives devient trop élevé ou lorsqu'il y a de la colinéarité entre les variables. Intuitivement, une solution serait de sélectionner les variables explicatives les plus intéressantes et de n'utiliser qu'elles pour entraîner le modèle. Or, il peut être difficile de choisir quelle variable mérite d'être sélectionnée.

Lorsqu'on se trouve dans une de ces situations, il serait donc intéressant de trouver une solution permettant de réduire la complexité du modèle. Elle représente la capacité du modèle à décrire des relations complexes entre les différentes variables explicatives et la variable réponse, ainsi que sa capacité à s'adapter aux données d'entraînement. En effet, un modèle de machine learning est adéquat s'il est capable de fournir de bonnes prédictions pour les données d'entraînement mais aussi pour de nouvelles données.

Concrètement, la complexité d'un GLM va varier selon le nombre de paramètres du modèle, i.e. le nombre de variables explicatives. Lorsque celui ci devient trop élevé, la complexité du modèle

explose. Afin de contrôler cette complexité, des techniques de régularisation vont être mises en place. Cela se traduit par une modification du problème de minimisation (1.3) qui permet d'obtenir les estimations des paramètres du modèle. Le problème de minimisation sous la forme lagrangienne devient

$$\hat{\beta} = \underset{\beta}{min}(-log(L(\beta)) + \lambda\ R(\beta)) \tag{1.4}$$

où le paramètre de réglage λ contrôle la force du terme de régularisation $R(\beta)$.

Il n'existe pas de formule fermée pour l'estimation des paramètres β du modèle. Cela va dépendre de la distribution de probabilité de la variable réponse Y et de la fonction lien choisie qui définiront la fonction de vraisemblance. Ensuite, des algorithmes itératifs sont utilisés pour ajuster les valeurs des paramètres β du modèle jusqu'à ce qu'ils convergent vers une solution du problème d'optimisation 1.4.

En vertu de la dualité des problèmes d'optimisation avec pénalité et avec contrainte, le problème d'optimisation (1.4) peut se réécrire à l'aide de la formulation suivante

$$\hat{\beta} = \underset{\beta}{min}(-log(L(\beta)) \qquad \text{soumis à la contrainte } R(\beta) \leq t \tag{1.5}$$

où t est une limite supérieure du terme $R(\beta)$ qui sera égal aux normes L1 ou L2 pour les régularisations de Ridge ou de Lasso définies aux sections 1.3.1 et 1.3.2. D'après la théorie des multiplicateurs de Lagrange, le vecteur des estimations $\hat{\beta}$ est solution du problème sous contrainte (1.5) avec $t \geq 0$ si et seulement si il est solution du problème d'optimisation pénalisé (1.4) avec $\lambda \geq 0$. L'ensemble des solutions est donc équivalent et les paramètres sont inversement reliés. Lorsque la pénalité est forte, i.e. quand λ est grand, la contrainte sera plus stricte, i.e. le t sera petit, et inversement. Cette dualité a été discutée et établie dans de nombreux ouvrages, notamment dans Hastie, T., Tibshirani, R. et Wainwright, M. (2015), dans le cadre de la régularisation.

La formulation pénalisée (1.4) est généralement préférée à la version (1.5) dans le cadre de l'apprentissage automatique lorsqu'on utilise la validation croisée pour déterminer les paramètres du modèle. En effet, les performances du modèle sont robustes à des petites variations du paramètre λ tandis que ce n'est pas le cas pour des petites variations du paramètre t. Néanmoins, la version sous contrainte (1.5) a l'avantage de fournir une expression explicite du domaine de la pénalité et donc, d'obtenir facilement une représentation graphique. Nous illustrerons les pénalités de chaque type de régularisation dans les sections 1.3.1, 1.3.2 et 1.3.3 correspondantes.

Il existe différentes méthodes de régularisation. Parmi les plus connues, Ridge, Lasso ou encore Elastic Net. Chacune d'elles a une pénalité différente et agit différemment sur le modèle. Avant de les analyser plus en détail dans les sections suivantes, remarquons que les pénalités ne seront appliquées uniquement sur les coefficients $\beta_1, ..., \beta_r$ et non sur l'intercept β_0. Pour rappel, l'intercept représente la valeur moyenne de la réponse lorsque toutes les variables explicatives x_{ij} sont mises à zéro. Or, la régularisation a pour objectif de contrôler l'impact des variables explicatives sur la réponse et non de biaiser l'estimation de la réponse moyenne lorsque toutes les variables sont nulles.

1.3.1 Régularisation de Ridge

La méthode de régularisation de Ridge consiste à ajouter une pénalité L2 à l'opposé de la log-vraisemblance du modèle. Dans un modèle linéaire généralisé, le vecteur des estimations β^R des paramètres du modèle obtenu via Ridge est solution du problème d'optimisation suivant

$$\beta^R = \min_{\beta}(-log(L(\beta)) + \lambda \sum_{j=1}^{r} |\beta_j|^2) \tag{1.6}$$

avec λ le paramètre de réglage. Il a pour but de contrôler l'impact relatif de ces deux termes sur les estimations des paramètres de la régression et est tel que $\lambda \geq 0$.

En d'autres mots, pour éviter que les coefficients de la régression ne prennent des valeurs trop élevées, ce qui causerait une augmentation significative de la variance, il est nécessaire d'imposer une contrainte à ces coefficients, le problème d'optimisation est donc équivalent à

$$\beta^R = \min_{\beta}(-log(L(\beta))) \qquad \text{avec la contrainte } \sum_{j=1}^{r} |\beta_j|^2 \leq t \tag{1.7}$$

où $t \geq 0$ est une limite supérieure pour la norme $L2$ du vecteur β.

Ce type de régularisation est une méthode de shrinkage continu. Cela signifie que les coefficients du modèle sont réduits de manière progressive en fonction de l'augmentation de la contrainte de régularisation. Si la pénalité est forte, alors les paramètres β_j de la régression sont proches de zéro. Lorsque le paramètre de réglage est nul, cela revient à utiliser la méthode du maximum de vraisemblance sans y appliquer de pénalité.

Cette méthode a pour but de permettre au modèle d'atteindre une meilleure performance de prédiction grâce à un compromis biais-variance. Ces deux composantes représentent deux sources d'erreur d'un modèle. En statistique, l'absence de biais indique que, en moyenne, la valeur estimée est correcte et que les cas où le modèle sous-estime et sur-estime la valeur réelle vont se compenser. Si le biais du modèle est élevé, cela signifie que cette compensation n'a pas lieu. L'ajustement sur le jeu d'entraînement n'est donc pas adéquat. D'autre part, la variance correspond aux erreurs causées par une sensibilité extrême du modèle aux fluctuations d'un jeu de données. Le modèle capte alors les tendances ainsi que le bruit. Ainsi, il devient incapable de performer sur de nouvelles données. L'ajustement du paramètre de réglage λ fixe le niveau du compromis biais-variance. Si il est trop élevé, les coefficients sont fortement réduits ce qui augmente le biais et réduit la variance. A l'inverse, un λ trop petit ne pénalise pas suffisamment. Le biais est donc réduit mais la variance augmente.

Cette méthode permet de gérer les problèmes qui peuvent apparaître lors de l'estimation des paramètres β du modèle. En effet, ils sont obtenus au moyen d'un algorithme itératif décrit à l'équation (1.2). Afin de résoudre ce système d'équations, il est nécessaire de pouvoir inverser la matrice $X^T WX$. Or, il existe des scénarios qui mènent à une inversion de matrice mal conditionnée ou bien la matrice devient singulière.

Cela est causé par la colinéarité entre les variables explicatives. A la base du modèle linéaire généralisé, les colonnes de la matrice X sont supposées linéairement indépendantes. Cependant, cette hypothèse est contredite lorsqu'une des variables explicatives est une combinaison linéaire des autres.

Autrement dit, lorsque l'information disponible à travers cette variable explicative peut être déduite grâce aux autres caractéristiques et n'apporte donc aucune plus-value. Dans ce cas, si la corrélation entre les variables est exacte, la matrice devient singulière et l'algorithme itératif échoue. Le plus souvent, nous n'avons pas à faire à une corrélation parfaite mais plutôt à une colinéarité proche. La matrice ne sera peut-être pas parfaitement singulière mais elle s'y rapprochera de sorte que le calcul de l'inverse fera face aux difficultés. Cette situation est de plus en plus fréquente avec l'augmentation de la taille des bases de données. En effet, lorsque le nombre de variables explicatives devient trop élevé, l'information disponible est conséquente et la redondance d'information est donc plus fréquente.

Si le calcul de l'inverse aboutit malgré l'instabilité numérique, les estimations des paramètres β risquent de mener à des erreurs importantes et des variances élevées. L'ajout de la pénalité de Ridge modifie le problème d'optimisation et transforme l'expression des estimations des paramètres β de l'équation (1.2). Mathématiquement, la matrice X^TWX devient $X^TWX + \lambda I_p$ avec I_p la matrice identité de dimension p. L'ajout du second terme rend la matrice non singulière et, par conséquent, inversible. En effet, la diagonale de la matrice X^TWX est augmentée de la constante λ strictement positive avant que l'inversion ne soit effectuée. Les valeurs propres de cette matrice sont donc strictement positive ce qui rend la matrice inversible. L'algorithme itératif peut alors performer correctement et rendre des estimations cohérentes des paramètres β. Lors de la première étude de la méthode de Ridge par Hoerl et Kennard (1970), cette transformation matricielle constituait le fondement de leur analyse.

A l'inverse de la méthode de Lasso, décrite dans la section 1.3.2, la pénalité *L2* n'impacte pas l'interprétabilité du modèle. En effet, elle n'a pas d'impact sur la sélection des variables pertinentes au modèle. Ainsi, l'ensemble des p variables explicatives se retrouveront dans le modèle. Nous développerons ce point dans la section suivante.

Graphiquement, la contrainte de la pénalité L2 définie à l'équation (1.7) est représentée à la figure 1.1 en dimension 2 par un cercle. Chaque axe représentant un des coefficients du modèle. Ils sont d'abord estimés via MLE à l'extérieur du cercle puis forcer de s'en rapprocher afin de satisfaire la contrainte. Ils seront alors réduits mais coïncideront rarement avec un des axes. Les estimations des paramètres ne seront donc pas fixées à zéro.

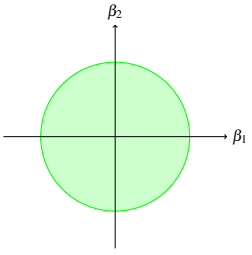

Figure 1.1. Domaine de la pénalité L2

1.3.2 Régularisation de Lasso

La méthode de régularisation de Lasso (i.e. Least Absolute Shrinkage and Selection Operator) consiste à ajouter une pénalité L1 à la fonction objective du modèle. Elle a été proposée par Tibshirani (1996) pour tenter de surpasser la méthode de Ridge. Dans un modèle linéaire généralisé, le vecteur des estimations β^L des paramètres du modèle obtenu via Lasso est solution du problème

d'optimisation suivant

$$\beta^L = \min_{\beta}(-log(L(\beta)) + \lambda \sum_{j=1}^{r} |\beta_j|) \tag{1.8}$$

avec λ le paramètre de réglage tel que $\lambda \geq 0$. Ce paramètre contrôle la complexité du modèle en pénalisant multiplicativement la norme L1. Il est possible d'obtenir le même résultat en appliquant une contrainte explicite sur la norme L1 ce qui fixe une limite stricte aux coefficients du modèle. Cela s'écrit sous la forme

$$\beta^L = \min_{\beta}(-log(L(\beta))) \quad \text{avec la contrainte } \sum_{j=1}^{r} |\beta_j| \leq t \tag{1.9}$$

où $t \geq 0$ est une limite supérieure pour la norme $L1$ du vecteur β. Cette contrainte impose que la somme des valeurs absolues des coefficients de régression soit inférieure à une constante fixée. Ainsi, les paramètres β resteront relativement petits, et certains d'entre eux seront nuls.

De façon analogue à la régularisation de Ridge, la régularisation de Lasso est une méthode de réduction. Elle va diminuer la valeur de certaines estimations des paramètres β mais, contrairement à la régularisation de Ridge, elle fait également une sélection automatique des variables explicatives les plus pertinentes du modèle en mettant les estimations des paramètres correspondant à zéro. Les variables explicatives dont le paramètre est égal à zéro vont automatiquement disparaître du score et n'auront donc pas d'effet sur la réponse. Par conséquent, appliquer la régularisation de Lasso permet d'obtenir des modèles plus parcimonieux que ceux obtenus sans régularisation ou via la régularisation de Ridge. Pour que cette propriété de sélection de variables apparaisse également dans la régularisation de Ridge, il aurait fallu que le paramètre de réglage λ précédant la pénalité $L2$ soit égal à l'infini, ce qui est théoriquement impossible.

Comme pour la méthode de Ridge, il y a un trade-off entre biais et variance nécessaire dans le choix du paramètre λ. En réduisant et mettant certains coefficients à zéro, le biais du modèle est augmenté et la variance diminuée. Il faut donc trouver le bon compromis afin que la régularisation de Lasso aide le modèle à être le plus fidèle aux données sans pour autant induire un sous ou sur ajustement par rapport au jeu d'entraînement.

Par ailleurs, il est évident que l'interprétabilité du modèle sera plus aisée une fois la régularisation de Lasso appliquée. En effet, le modèle final contiendra moins de prédicteurs ce qui permettra de comprendre de manière plus directement les relations qu'ils ont avec la réponse et de connaître directement lesquels sont significatifs et ceux qui ne le sont pas.

Bien que la régularisation de Lasso prime sur celle de Ridge concernant l'interpretabilité du modèle, elle se heurte à certaines limitations. Considérons notamment le problème de dimension du modèle lorsque le nombre de variables explicatives est conséquent ainsi que la multi-colinéarité entre ces variables qui sont deux obstacles que Ridge peut surmonter. Imaginons que plusieurs variables explicatives soient corrélées les unes aux autres, Lasso aura du mal à les distinguer et à comprendre laquelle d'entre elles est la plus importante dans le modèle. Il aura tendance dans un premier temps à vouloir fixer les estimations des paramètres de régression de ces variables à des valeurs relativement semblables. Ensuite, lorsque le paramètre de réglage λ va pousser le modèle à réduire et annuler certains de ces coefficients, Lasso sélectionnera arbitrairement une variable parmi les variables corrélées. Le modèle final sera alors basé sur une variable explicative éventuellement non pertinente alors

qu'une autre, qui aurait pu l'être, a été exclue.

La contraire de la pénalité L1 se représente en dimension 2 par un losange comme illustré à la figure 1.2. Chaque axe représentant un des coefficients du modèle. Contrairement au cercle de la régularisation de Lasso, le losange dispose de plusieurs coins alignés avec les axes ce qui annule le paramètre du modèle. En effet, lorsque les paramètres du modèle vont être réduits, il est plus probable qu'ils commencent par croiser un des coins du losange plutôt que l'arête. En dimension 3, la contrainte devient un rhomboïde qui dispose à nouveau de nombreux coins.

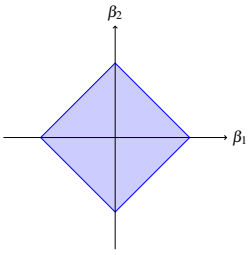

Figure 1.2. Domaine de la pénalité L1

1.3.3 Régularisation Elastic Net

En tenant compte des avantages et des inconvénients des régularisations de Lasso et de Ridge, il serait souhaitable d'avoir une méthode de régularisation de type shrinkage, permettant une sélection automatique de variables, tout en traitant efficacement la colinéarité au sein d'un ensemble de variables explicatives ainsi que les problèmes de haute dimension du modèle.

La méthode de régularisation Elastic Net a été proposée par Zou et Hastie (2005) et permet de répondre aux limitations des méthodes de Ridge et Lasso. Elle est définie comme la combinaison des pénalisations L1 et L2. Son terme de régularisation est donné par

$$R(\beta,\lambda,\alpha) = \lambda((1-\alpha)\sum_{j=1}^{r}|\beta_j|^2 + \alpha\sum_{j=1}^{r}|\beta_j|) \tag{1.10}$$

avec $\lambda \geq 0$ le paramétrage de réglage et $0 \leq \alpha \leq 1$ le paramètre qui contrôle la part de pénalité L1 et L2 dans le modèle.

Cette méthode de régularisation possède plusieurs avantages provenant des méthode de Ridge et Lasso :

1) Sélection automatique des variables : au travers de la pénalité de Lasso, les coefficients des variables les moins pertinentes seront mis à zéro afin de garder pour l'entraînement de notre modèle les variables explicatives les plus significatives. Cette caractéristique est particulièrement importante étant donné que l'interprétabilité est un des critères principaux de la qualité d'un modèle.
2) Gestion de la multi colinéarité entre les variables explicatives : la régularisation Elastic Net a la capacité de regrouper les variables fortement corrélées ensemble en fixant les coefficients de régression de ce groupe à des valeurs similaires grâce à la pénalité L2. Si ces variables ne sont pas pertinentes au modèle alors la pénalité L1 les exclura.

3) Robuste en haute dimension : la qualité du modèle est maintenue lorsque le nombre de variables explicatives devient élevé par rapport aux nombres d'observations. La combinaison des deux pénalités permet de stabiliser les estimations des paramètres du modèle tout en continuant de sélectionner les variables pertinentes.

Néanmoins, cette méthode a quand même certains inconvénients dont le choix des paramètres. Elle est basée sur deux paramètres, λ et α, dont les valeurs vont fortement influencer le modèle. Il est donc nécessaire d'avoir à sa disposition des critères de sélection efficace ainsi que connaître les propriétés que nous voulons apporter à notre modèle. La méthodologie classique utilisée pour le choix de ces paramètres est la validation croisée. Cette méthode est détaillée à la section 1.4.

Le paramètre α est sélectionné en fonction des propriétés recherchées pour le modèle. Il se rapprochera de 1 lorsque l'on souhaite avoir un modèle le plus parcimonieux possible et il sera proche de 0 lorsque l'on s'inquiète des colinéarités entre les variables explicatives. En effet, lorsque le paramètre α est nul, on retombe sur la régularisation de Ridge et quand α est égale à 1, on retombe sur celle de Lasso.

Afin d'analyser graphiquement la pénalité, nous regardons le problème d'optimisation sous contrainte. Dans le cas des modèles linéaires généralisés régularisés à l'aide de la méthode Elastic Net, les estimations $\beta^{E_{net}}$ des paramètres du modèle sont obtenues via le problème d'optimisation suivant

$$\beta^{E_{net}} = \min_{\beta}(-log(L(\beta)) \quad \text{avec la contrainte} \quad \alpha \sum_{j=1}^{r} |\beta_j| + (1-\alpha) \sum_{j=1}^{r} |\beta_j|^2 \leq t \tag{1.11}$$

où $t \geq 0$ est une limite supérieure pour la combinaison des normes L1 et L2.

La pénalité Elastic Net est représentée sur le graphique 1.3 par la contrainte du problème 1.11. Le losange bleu décrit le périmètre du cas où le paramètre α est égal à 1. Le cercle vert décrit le cas où le paramètre α est égal à 0. Plus α s'éloigne de 1, plus les arêtes du losange vont se courber pour former un arc de cercle lorsque α devient proche de 0. Le cas intermédiaire avec $\alpha = 0.5$ est représenté en rouge.

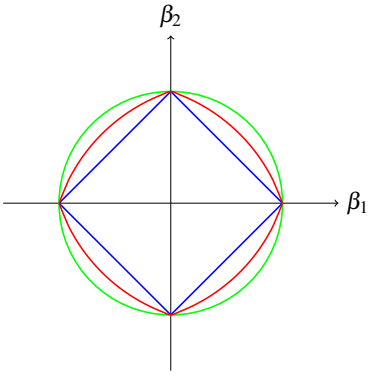

Figure 1.3. Pénalité L1 (bleu), pénalité L2 (vert), pénalité Elastic Net avec $\alpha = 0.5$ (rouge)

1.4 Validation croisée

Dans la section 1.3, différentes méthodes de régularisation ont été définies. Chacune d'entre elles dépend d'un paramètre de réglage λ. Il doit être choisi de sorte à obtenir les meilleures prédictions pour la variable réponse du modèle. Pour ce faire, la validation croisée à K-pli est une technique régulièrement utilisée. Le paramètre K varie en général entre 5 et 10 selon la taille du jeu de données. Il peut néanmoins être plus élevé et, dans le cas limite, être égal à n le nombre d'observations. Dans ce cas, nous parlerons de validation croisée leave-one-out.

L'approche générale de la validation croisée à K-pli se base sur une division aléatoire du jeu de données en K sous-groupes de taille approximativement égale. Un sous-groupe est sélectionné pour jouer le rôle de l'ensemble de validation, i.e. pour mesurer les performances prédictives des modèles, tandis que les $K-1$ restants constituent l'ensemble d'entraînement, i.e. qu'ils sont utilisés pour ajuster les différents modèles proposés.

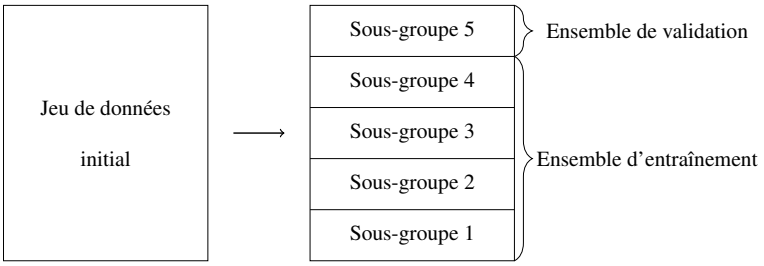

Figure 1.4. Validation croisée à K-plis avec K=5

Une fois que cette division est définie, nous ajustons le modèle sur l'ensemble d'entraînement de sorte à obtenir les estimations des paramètres de régression et, par conséquent, la prédiction de la variable réponse $\hat{y}_i$ pour chaque individu $i = 1,...,n$. Ensuite, la performance du modèle est calculée sur le sous-groupe qui joue le rôle de l'ensemble de validation. Il existe différentes méthodes qui seront plus ou moins adaptées selon la distribution de la variable réponse. En général si la distribution est gaussienne, l'erreur quadratique moyenne est un choix cohérent. Elle est définie comme étant la moyenne des carrés des différences entre les vraies valeurs y_i et les valeurs prédites $\hat{y}_i$. Mathématiquement, on a

$$MSE_k = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

avec $k = 1,...,K$ l'indice du sous-groupe correspondant à l'ensemble de validation. Une alternative est l'erreur absolue moyenne qui mesure la moyenne des valeurs absolues des erreurs comme suit

$$MAE_k = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Lorsque la réponse suit une loi de Poisson, il est préférable d'utiliser la log-vraisemblance, comme définie dans la section 1.2, ou la déviance réduite qui mesure la qualité de l'ajustement d'un modèle

en comparant le modèle ajusté au modèle saturé. Mathématiquement, on a

$$\begin{aligned} D &= 2(log(L_{satur}) - log(L)) \\ &= 2\sum_{i=1}^{n}(y_i \ln \frac{y_i}{\hat{y}_i} - (y_i - \hat{y}_i)) \end{aligned}$$

Ensuite, le processus est répété $K-1$ fois, jusqu'à ce que chaque sous-groupe ait été considéré une fois comme l'ensemble de validation. Nous obtenons ainsi K estimations MSE_1, MSE_2, MSE_K de l'erreur quadratique moyenne sur l'ensemble de validation. L'erreur de validation croisée à K-pli est ensuite calculée en prenant la moyenne des MSE_l, $l = 1, ..., K$. Autrement dit, elle est définie par la formule suivante :

$$CV = \frac{1}{K}\sum_{i=1}^{K} Mesure_i$$

où K est le nombre de plis choisi de manière arbitraire et *Mesure* correspond à la mesure de performance du modèle sélectionnée en fonction des caractéristiques de la réponse. Cette procédure est répétée pour chaque modèle envisagé, afin d'obtenir une estimation de l'erreur de validation croisée pour chaque modèle analysé. Celui qui possède la plus petite erreur est alors sélectionné comme le modèle le plus adéquat.

Dans le contexte des méthodes de régularisation de Ridge et Lasso, nous utilisons la validation croisée pour déterminer la valeur optimale du paramètre de régularisation λ. Pour ce faire, nous commençons par définir un ensemble de valeurs potentielles pour λ. Ensuite, pour chaque valeur de λ, nous appliquons la méthode de validation croisée, ce qui nous fournit une estimation de l'erreur de validation croisée. Le paramètre λ sélectionné est alors celui qui minimise l'erreur de validation croisée. Cette méthode est également utilisée dans d'autres situations pour déterminer d'autres paramètres du modèle ou, simplement pour choisir parmi plusieurs modèles proposés lequel aurait tendance à ajuster le mieux les données.

Une procédure similaire est utilisée dans le cas de la régularisation Elastic Net à la différence que, dans ce cas-ci, la validation croisée est utilisée simultanément pour les paramètres λ et α. On définit aléatoirement deux séquences de valeurs pour ces paramètres. Ensuite, pour chaque combinaison possible (λ, α) nous appliquons la méthode de validation croisée et nous sélectionnons le modèle avec le couple de paramètres qui minimise l'erreur de validation croisée.

La méthode de validation croisée leave-one-out est moins utilisée, principalement en raison de sa charge computationnelle plus élevée. Contrairement à la validation croisée à K-plis où le modèle est ajusté K fois, la méthode leave-one-out nécessite d'ajuster le modèle n fois, où n est le nombre d'observations dans le jeu de données. Étant donné que les ensembles de données contiennent généralement un nombre considérable d'observations, cette approche peut entraîner une charge de calcul importante. De plus, retirer une seule observation dans un large jeu de données a généralement peu d'impact sur les estimations, ce qui rend la méthode leave-one-out plus adaptée aux ensembles de données de petite taille.

Chapitre 2

Formulation des Accurate GLM

Maintenant que les bases du modèle ont été établies, nous pouvons définir formellement la méthode des Accurate GLM proposée par Fujita, S., Tanaka, T., Kondo, K. et Iwasawa, H. (2020). Nous commencerons par formuler le modèle en lui-même, puis nous explorerons en détail ses différentes composantes.

2.1 Définition des AGLM :

La méthode des Accurate GLM est définie comme un modèle linéaire généralisé régularisé. Ainsi, la moyenne de la variable réponse Y_i est dérivée de la définition du score d'un GLM. Cependant, les variables explicatives x_{ij} y sont transformées en des variables multidimensionnelles z_{ij}. Par conséquent, les paramètres de régression β_j sont également convertis en paramètres multidimensionnels β_j'. La réponse moyenne est formulée par la relation

$$E[y_i] = g^{-1}(\beta_0 + \sum_{j=1}^{r} z_{ij}\beta_j') \quad \text{pour i=1,...,n} \tag{2.1}$$

où g est la fonction lien définie dans la section 1.1 sur les modèles linéaires généralisés. De plus, une méthode de régularisation est appliquée à ce modèle. La méthode retenue est la pénalité Elastic Net définie à la section 1.3.3. Le terme de régularisation se réécrit comme suit :

$$R(\beta, \lambda, \alpha) = \lambda((1-\alpha) \sum_{j=1}^{r}\sum_{k=1}^{m_j} |\beta_{jk}|^2 + \alpha \sum_{j=1}^{r}\sum_{k=1}^{m_j} |\beta_{jk}|) \tag{2.2}$$

avec λ le paramètre de réglage, α le paramètre de contrôle et β_{jk} les coefficients de la régression associés à la variable explicative z_{ij}.

Il existe deux types de variables explicatives, elles peuvent être catégorielles ou continues. Une variable catégorielle est, par définition, une variable qui prend ces valeurs dans un nombre fini de catégories. Ces différentes catégories peuvent être ordonnées ou non ordonnées. Par exemple, la variable "marque" est une variable catégorielle non ordonnée tandis que la variable "Bonus Malus" est généralement une variable catégorielle ordonnée composée de niveaux qui sont progressivement parcourus selon l'évolution de chaque assuré. A contrario, une variable continue peut prendre une infinité de valeurs sur une plage spécifique et peut être ordonnée de manière significative. Prenons la variable

âge du conducteur en exemple.

Selon la nature de la variable explicative x_{ij}, qu'elle soit catégorielle ou continue, nous effectuerons une transformation en un vecteur multidimensionnel z_{ij}. Voici les trois scénarios possibles qui seront définis à la section 2.3 :

- Transformation des variables catégorielles non ordonnées en variables fictives usuelles $d_j^U(x)$.
- Transformation des variables catégorielles ordonnées en variables fictives ordinales $d_j^O(x)$.
- Discrétisation des variables continues et transformation en variables fictives ordinales $d_j^O(x)$ ou en variables L $l_j(x)$.

2.2 Discrétisation des variables numériques

Les variables x_{ij} continues considérées dans la structure des GLM ne peuvent généralement pas être introduite dans le modèle. En effet, dans la majorité des cas, leur effet ne sera pas linéaire sur le score. Prenons l'âge du conducteur. Dans le domaine des assurances auto, il est établi que les jeunes conducteurs sont généralement plus susceptibles de causer des accidents par rapport aux conducteurs plus expérimentés. Il est également courant de rencontrer un pic d'accidents aux alentours de 45-55 ans et une forte augmentation de la fréquence aux âges avancés. Il n'est donc pas adéquat de l'introduire telle quelle dans le score car l'effet de la variable sur la réponse sera uniquement croissant ou décroissant. Ainsi, lorsqu'une des variables explicatives est continue, il est nécessaire de la discrétiser afin qu'elle devienne catégorielle. La discrétisation est un processus simple consistant à découper l'ensemble des valeurs de la variable en plusieurs sous-catégories. Par exemple, si l'âge du conducteur est compris entre 18 et 80 ans, nous catégoriserons cette variable en créant différentes classes d'âge. Nous prendrons les conducteurs de 18 à 25 ans puis ceux de 25 à 30 et ainsi de suite jusqu'à avoir partitionné l'ensemble des valeurs initiales.

Considérons une variable numérique x définie sur $]b_0, b_m]$. Fixons $m-1$ bornes intermédiaires $b_1, b_2, ..., b_{m-1}$ et découpons l'intervalle de sorte à obtenir m tranches $B_1 =]b_0, b_1]$, $B_2 =]b_1, b_2]$, ..., $B_m =]b_{m-1}, b_m]$. Le nombre et la taille de ces tranches sont choisis arbitrairement afin que les m classes contiennent un nombre d'observations similaire. Ce choix a des conséquences non négligeables sur le modèle.

D'une part, si le paramètre m est trop élevé, cela peut causer du sur-ajustement. En effet, en découpant une variable explicative continue en un nombre élevé de variables discrètes, le nombre de paramètres du modèle augmente de manière significative ainsi que la complexité du modèle. En situation de sur-ajustement, le modèle risque de coller trop précisément à l'ensemble de données d'entraînement. Il captera la relation entre les variables explicatives et la réponse mais également le bruit aléatoire présent dans la base de donnée étudiée. Le modèle fournira une très bonne prédiction pour les données d'entraînement mais aura des difficultés à se généraliser sur d'autres données.

D'autre part, si le paramètre m est trop petit, cela peut causer un sous-ajustement. En effet, en choisissant peu de tranches contenant chacune un large ensemble de données, le nombre de paramètres, et donc, la complexité du modèle est réduit. A l'inverse du sur-ajustement, le modèle ne colle pas assez aux données et n'arrive pas à capter les relations entre les variables explicatives et la réponse. Par conséquent, il pourra fournir une prédiction de mauvaise qualité que ce soit sur le jeu de données

d'entraînement ou sur un nouvel ensemble.

Le choix du paramètre m est donc crucial. Il doit être ni trop grand ni trop petit afin d'éviter le sur et le sous ajustement. Lorsque ce choix est fait manuellement, comme dans les GLM, il nécessite une connaissance actuarielle et une analyse descriptive approfondie des variables. Afin de rendre ce choix de paramètre moins arbitraire et d'automatiser le processus, les méthodes de régularisation définies dans la section 1.3 possédant la propriété de sélection de variables peuvent être utilisées. En effet, elles vont permettre de limiter la capacité du modèle à mémoriser le bruit dans les données d'entraînement. Ce processus est automatisé dans les AGLM afin de discrétiser les variables continues de la meilleure façon qu'il soit.

Exemple 2.2.0.1 *Dans le cadre d'une assurance auto, une variable fréquemment utilisée est l'âge du conducteur. Supposons qu'il varie entre 18 et 80 ans et discrétisons cet ensemble en 5 tranches. Après avoir choisi arbitrairement les 4 bornes intermédiaires, nous obtenons les 5 tranches suivantes*

$$B_1 = [18,28], B_2 =]28,40], B_3 =]40,55], B_4 =]55,65], B_5 =]65,80].$$

2.3 Codage des variables explicatives à l'aide de variables fictives

Suite à la discrétisation, nous procédons au codage des variables numériques en utilisant des variables fictives binaires. Il est également utilisé pour les variables catégorielles. Commençons par définir les variables fictives usuelles. Ensuite, nous les modifierons de sorte à intégrer le caractère ordinal entre les différentes catégories des variables.

2.3.1 Variables fictives usuelles

Considérons la variable x divisée en m niveaux. Elle est soit une variable numérique définie sur $]b_0, b_m]$ et discrétisée en m classes, soit une variable catégorielle composée de m catégories. Une variable fictive usuelle $d_k^U(x)$ est définie pour chaque classe k allant de 1 à m. Elle sera égale à 1 si la caractéristique x de l'individu est contenue dans la k^{ieme} classe et 0 sinon. Mathématiquement, elle est définie par la variable binaire suivante

$$d_k^U(x) = \begin{cases} 1 & \text{si } x = k \text{ pour } k \in \{1,...,m\} \\ 0 & \text{sinon} \end{cases}$$

Remarquons que lorsque la variable explicative x est divisée en m classes, m variables fictives usuelles sont définies. En d'autres termes, une nouvelle variable est créée pour chaque catégorie correspondante. Cela représente l'une des différences notables par rapport au modèle linéaire généralisé. En effet, un niveau de référence y est défini, ce qui conduit à ne construire que $m-1$ variables fictives usuelles pour m catégories. Ce concept sera examiné en détail dans la section 3.2.

Par définition, pour chaque valeur de x, le vecteur des variables fictives usuelles $(d_1^U(x),...,d_m^U(x))$ ne contiendra qu'une seule variable égale à 1 et toutes les autres seront nulles. Sous forme matricielle, cela nous rend une matrice diagonale ne contenant que des 1 sur la diagonale principale i.e. la matrice

identité,

$$U = \begin{pmatrix} 1 & 0 & 0 & ... & 0 \\ 0 & 1 & 0 & ... & 0 \\ 0 & 0 & 1 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & ... & 1 \end{pmatrix}$$

Cela implique également que

$$\sum_{k=1}^{m} d_k^U(x) = 1 \quad \Longrightarrow d_1^U(x) = 1 - \sum_{k=2}^{m} d_k^U(x).$$

Cela signifie qu'une combinaison linéaire des variables explicatives $d_k^U(x)$ est égal à une constante. Autrement dit, cela induit de la colinéarité entre les variables $d_k^U(x)$ du modèle. Cela risque de créer des instabilités dans le modèle comme expliqué dans la section 1.3.1. Dans les AGLM, l'ajout d'une méthode de régularisation et, en particulier, celle de Ridge, permettra de résoudre ce problème. Dans les GLM, il est également traité grâce à la mise en place du niveau de référence.

Ce codage est principalement utilisé pour les variables catégorielles et va impacter la structure de notre modèle. La variable explicative catégorielle x_{ij} initiale est convertie en un vecteur de m variables fictives $(d_1^U(x_{ij}), ..., d_m^U(x_{ij}))$. De même pour le coefficient de régression β_j associé à la variable x_{ij} qui est, lui aussi, converti en un vecteur $(\beta_{j_1}, ..., \beta_{j_m})$. Ainsi, le terme de la régression linéaire initiale

$$x_{ij}\beta_j$$

se transforme en combinaison linéaire de m nouveaux termes dépendant des nouvelles variables fictives

$$\sum_{k=1}^{m} d_k^U(x_{ij})\beta_{j_k}.$$

Lorsque le modèle de régression linéaire généralisé est réalisé, les coefficients β_{j_k} sont estimés indépendamment les uns des autres. Les relations entre les différents niveaux de la variable ne sont donc pas représentées par le modèle. Ainsi, un des coefficients β_{j_k} pourrait être nul alors que les coefficients adjacents $\beta_{j_{k-1}}$ et $\beta_{j_{k+1}}$ ne le sont pas. Cela pourrait mener à des incohérences dans la prédiction de la variable réponse. Par exemple, deux individus appartenant à des groupes adjacents pourraient avoir une réponse moyenne totalement différente en raison du fait que la courbe de contribution des variables explicatives n'est pas lisse. Ce problème n'apparaît pas dans le cas du modèle additif généralisé. En effet, les β ne sont pas estimés indépendamment. C'est une fonction $f(x)$ qui est estimée sur base des données avec comme seule restriction qu'elle soit lisse. La méthode la plus connue est l'utilisation des splines de lissage ou, en d'autres termes, des fonctions polynomiales par morceaux.

Notons que lorsqu'une variable fictive est nulle, cela implique la disparition de cette variable dans la structure de notre modèle. Lorsqu'une variable fictive est égale à 1, le coefficient de régression qui lui est associé est alors ajouté à l'intercept β_0 du modèle. C'est trivial puisque chaque individu i ne peut appartenir qu'à un seul groupe de la variable catégorielle.

Ce codage est inefficace pour les variables numériques. En effet, lorsqu'une variable numérique est discrétisée et transformée en classes, il y a toujours une notion d'ordre entre ces variables. Si le

codage par variables fictives usuelles est utilisé, cette information se perd. Un individu appartiendra à la classe k si la variable $d_k^U(x)$ égale à 1 mais le modèle ne sera pas capable de comprendre où se situe cette catégorie par rapport aux autres. Il capture le fait que l'individu appartienne à la catégorie k mais pas qu'elle se situe entre les catégories $k-1$ et $k+1$. Un nouveau codage est donc exploité à la section suivante pour intégrer cette notion d'ordre dans le modèle.

Exemple 2.3.1.1 *Repartons de l'exemple précédent. Dans ce cas-ci, nous devons créer 5 variables fictives*

$$d_1^U(x) = \begin{cases} 1 & si\ x \in B_1 = [18,28] \\ 0 & sinon \end{cases}$$

$$\vdots$$

$$d_5^U(x) = \begin{cases} 1 & si\ x \in B_5 = [65,80] \\ 0 & sinon \end{cases}$$

Considérons un individu âgé de 36 ans, sa variable explicative âge est représentée par le vecteur de variables fictives $(0,1,0,0,0)$. *Pour un GLM, nous avons seulement 4 variables binaires qui sont définies. Le vecteur de dimension 4 contiendra pour chaque valeur de x trois variables nulles et une égale à 1, sauf pour le niveau de référence qui sera représenté par un vecteur nul.*

2.3.2 Variables fictives ordonnées

Afin d'obtenir un codage adapté aux variables numériques, nous définissons de nouvelles variables fictives qui prennent en compte la notion d'ordre. Considérons la variable x divisée en m niveaux $L_1, L_2, ..., L_m$ une variable numérique définie sur $]b_0, b_m]$ et discrétisée en m classes ou directement une variable catégorielle composée de m sous-groupes. Une variable fictive ordinale $d_k^O(x)$ est définie pour chaque niveau k allant de 1 à m. Mathématiquement, pour $x = L_l,\ l \in \{1,...,m\}$, elle est définie par la variable binaire suivante

$$d_k^O(x) = \begin{cases} 1 & \text{si } l < k \text{ pour } k \in \{1,...,m\} \\ 0 & \text{sinon} \end{cases} \tag{2.3}$$

Par définition, pour chaque variable x, le vecteur des variables fictives ordinales $(d_1^O(x), ..., d_m^O(x))$ contiendra autant de 1 qu'il n'y a de k strictement supérieur à la valeur de x tandis que les autres seront nulles. En d'autres mots, si la valeur de la variable explicative pour l'individu i appartient à une classe strictement inférieure à la $k$ième classe alors la variable $d_k^O(x)$ est égale à 1, sinon elle est nulle. Exprimé sous forme matricielle, cela nous donne une matrice triangulaire supérieure ne contenant que des zéros sur la diagonale principale,

$$O = \begin{pmatrix} 0 & 1 & 1 & ... & 1 \\ 0 & 0 & 1 & ... & 1 \\ 0 & 0 & 0 & ... & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & ... & 0 \end{pmatrix}$$

Ce codage est utilisé pour les variables catégorielles ordonnées ainsi que pour les variables numériques. De la même manière que pour les variables fictives usuelles, la variable x_{ij} est convertie en un vecteur de m variables fictives ordonnées et le coefficient de régression β_j est remplacé par un vecteur de coefficients de dimension m. Ainsi, le terme initial de la régression

$$x_{ij}\beta_j$$

se transforme en une combinaison linéaire de nos nouvelles variables fictives ordinales $d_k^O(x)$ et de leurs coefficients de régression associés $(\beta_{j_1},...,\beta_{j_m})$ pour chaque variable explicative x_{ij}, à savoir

$$\sum_{k=1}^{m} d_k^O(x_{ij})\beta_{j_k}. \tag{2.4}$$

C'est une fonction en escalier. Lorsqu'une des variables fictives ordinales $d_k^O(x_{ij})$ est nulle, le terme $d_k^O(x_{ij})\beta_{j_k}$ correspondant disparaît du modèle. Lorsque cette variable est non nulle, i.e. quand $k > x$, le terme $d_k^O(x_{ij})\beta_{j_k}$ correspondant est pris en compte dans l'impact sur la variable réponse.

Ce type de codage permet de lisser les coefficients de régression et de tenir compte des relations d'ordre qui existent entre les différentes catégories de la variable. En effet, la première variable fictive ordinale $d_1^O(x)$ capture les observations de la variable x qui appartiennent à une classe strictement inférieure à la classe 1. La deuxième variable $d_2^O(x)$ capture les observations de x qui appartiennent à une classe strictement inférieure à la classe 2. Ce processus se répète pour chaque variable fictive ordinale $d_k^O(x)$ avec $k = 1,...,m$ de sorte qu'une fois les variables définies, elles captent correctement la progression ordonnée des niveaux de la variable explicative x. A la place d'être estimé indépendamment, chaque coefficient β_{j_k} dépend des catégories inférieures et supérieures.

Exemple 2.3.2.1 *Considérons le même individu âgé de 36 ans que dans l'exemple précédent. Sa variable explicative âge est désormais représentée par le vecteur de variables fictives ordinales* $(0,0,1,1,1)$. *Grâce à cette représentation, nous pouvons en déduire que l'âge de l'individu est strictement inférieur aux bornes* b_2, b_3, b_4 *et supérieur ou égal aux bornes* b_0 *et* b_1. *Par déduction, l'âge de l'individu appartient à la classe* $B_2 =]28,40]$ *et nous avons connaissance de la position de cette classe par rapport aux autres.*

Il est possible que, lors de la discrétisation de la variable numérique, plusieurs sous-catégories se soient créées alors qu'elles auraient pu se regrouper en une seule. En ajoutant à ce codage une technique de régularisation ayant comme propriété la sélection automatique des variables, cela va produire un effet de groupement. Autrement dit, nous allons regrouper certaines catégories adjacentes afin de créer une nouvelle sous division contenant un nombre de groupes strictement inférieur au nombre de tranches initiales m. Pour ce type de régularisation, il faut se référer à la pénalité de type L1 comme par exemple la régularisation de Lasso ou Elastic Net avec un paramètre α strictement supérieur à 0. Ces méthodes sont détaillées dans la section 1.3. Les avantages de la régularisation Elastic Net, en particulier sa capacité à combiner les avantages de celle de Lasso et de Ridge, y ont été évoqués, ce qui justifie son application dans le cadre des AGLM.

Approfondissons ce concept en fournissant une explication plus détaillée. Considérons les variables explicatives x_{ij} de l'individu $i = 1,...,n$ avec j allant de 1 à r. Il existe deux possibilités. Soit

x_{ij} est une variable catégorielle ordonnée composée de m_j tranches. Soit x_{ij} est une variable numérique discrétisée en m_j classes. Quelque soit la situation, les variables sont transformées en variables fictives ordonnées $d_k^O(x_{ij})$. Ensuite, afin d'estimer les coefficients de régression β_{jk} du modèle (2.4), le problème d'optimisation suivant doit être résolu

$$\hat{\beta} = \min_{\beta}(-log(L(\beta)) + R(\beta, \lambda, \alpha)) \tag{2.5}$$

où le terme de régularisation Elastic Net est

$$R(\beta, \lambda, \alpha) = \lambda((1-\alpha) \sum_{j=1}^{r} \sum_{k=1}^{m_j} |\beta_{jk}|^2 + \alpha \sum_{j=1}^{r} \sum_{k=1}^{m_j} |\beta_{jk}|)$$

La régularisation a pour principal effet de réduire ou d'annuler les coefficients de régression β_{jk} du modèle. Pour une variable explicative fixée, deux scénarios sont possibles :

- Tous les coefficients β_{jk} $(k = 1, ..., m)$ associés aux variables $(d_1^O(x_{ij}), ..., d_m^O(x_{ij}))$ sont estimés à zéro, alors la variable explicative correspondante est exclue du modèle.
- Un nombre $l < m$ parmi les coefficients β_{jk} $(k = 1, ..., m)$ sont fixés à zéro, tandis que les $m - l$ autres coefficients restent non nuls. Les catégories adjacentes sont alors regroupées de manière à former $m - (l-1)$ nouvelles catégories. Prenons par exemple $m = 8$ et supposons que les paramètres β_1, β_4, β_6 et β_7 soient fixés à zéro tandis que β_2, β_3, β_5 et β_8 restent non nuls. Dans cette configuration, les 8 sous-groupes initiaux sont regroupés en 5 nouveaux sous-groupes. Le premier contient la première classe, le deuxième la deuxième classe, le troisième la troisième et quatrième classe, le quatrième la cinquième, sixième et septième classe, et le dernier la huitième classe.

Mathématiquement, l'ajustement du modèle en utilisant un codage avec des variables fictives ordonnées et une technique de régularisation qui sélectionne les variables est analogue à l'ajustement du modèle en utilisant un codage simplifié avec des variables fictives classiques et la technique de régularisation Fused Lasso. Cette méthode a été proposée par Robert Tibshirani and Michael Saunders (2005). Elle se base sur le problème d'optimisation défini dans le cadre de la régularisation de Lasso auquel une contrainte supplémentaire est ajoutée. Les estimations sont obtenues à l'aide du problème d'optimisation suivant

$$\beta^{Fused\ Lasso} = \min_{\beta}(-log(L(\beta)))$$

soumis aux contraintes

$$\sum_{j=1}^{r} \sum_{k=1}^{m} |\beta_j| \leq t_1 \qquad \text{et} \qquad \sum_{j=1}^{r} \sum_{k=2}^{m} |\beta_{jk} - \beta_{j_{k-1}}| \leq t_2$$

avec t_1 et t_2 les paramètres de réglage. La pénalité L1 sélectionne les variables de manière globale tandis que la deuxième pénalité force les différences entre coefficients à tendre vers zéro. Autrement dit, Lasso pénalise la somme des valeurs absolues des coefficients pour encourager la parcimonie dans les coefficients tandis que Fused Lasso introduit une pénalisation supplémentaire qui encourage la continuité ou la similarité entre les coefficients adjacents. Effectivement, lorsque les groupes de la variable explicative sont ordonnés, on s'attend à ce que la réponse varie faiblement entre deux catégories adjacentes du prédicteur x_j. Autrement dit, nous cherchons à éviter les changements brusques et

favorisons des sous-vecteurs de coefficients β plus réguliers. Ainsi, les écarts entre les coefficients des niveaux adjacents sont pénalisés. Les paramètres t_1 et t_2 contrôlent respectivement la parcimonie des coefficients et la similarité entre les coefficients adjacents. Les paramètres optimaux à utiliser peuvent être obtenus à l'aide de la validation croisée. La méthode des AGLM est basée sur le codage et la régularisation Elastic Net, nous ne développerons donc pas plus en détail le Fused Lasso.

Notons qu'une approche alternative à la définition (2.3) existe. En effet, on peut définir une nouvelle variable fictive ordonnée en ajustant la frontière. Voici sa définition :

$$d'^{O}_{k}(x) = \begin{cases} 1 & \text{si } x > k \text{ pour } k \in \{1,...,m\} \\ 0 & \text{sinon} \end{cases}$$

La forme matricielle devient alors une matrice triangulaire inférieure contenant des 0 sur la diagonale principale,

$$O' = \begin{pmatrix} 0 & 0 & 0 & ... & 0 \\ 1 & 0 & 0 & ... & 0 \\ 1 & 1 & 0 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & ... & 0 \end{pmatrix}$$

2.3.3 Variables L

Les variables fictives décrites dans la section précédente permettent au modèle de capturer les relations ordinales entre les différentes valeurs d'une variable explicative. Cependant, un aspect peut poser problème dans certaines situations : la courbe de contribution des variables fictives ordonnées définie par (2.4) dans la section précédente n'est pas continue. Pour résoudre ce problème, nous allons introduire une nouvelle fonction qui, une fois intégrée dans le modèle de régression linéaire, rendra la courbe de contribution de ces variables continue.

Soit x une variable continue définie sur $]b_0, b_m]$ discrétisée en m classes. Les variables L, notées $l_k(x)$ pour $k = 1,...,m$, sont définies par

$$l_k(x) = \begin{cases} |x - b_k| & \text{si } k = 1,...,m-1 \\ x & \text{si } k = m \end{cases}$$

En particulier, la fonction $l_k(x) = |x - b_k|$ mesure la distance entre la variable explicative x et la borne supérieure b_k de la tranche k.

La courbe de contribution de x indique la relation entre cette variable et la réponse du modèle, en gardant les autres variables explicatives constantes. Elle est définie comme la somme de chaque variable L multipliée par son coefficient de régression, à savoir

$$L(x_{ij}) = \sum_{k=1}^{m} l_k(x_{ij})\beta_{j_k}$$

avec $i = 1,...,n$ l'individu et $j = 1,...,r$ l'indice du prédicteur. Cette fonction relie les couples de points $(b_k, L(b_k))$, pour k=0,...,m, passant par les bornes choisies lors de la discrétisation.

La fonction $L(x)$ est une ligne polygonale, c'est à dire une fonction linéaire par morceaux. Elle relie les points $(b_k, L(b_k))$ pour $k \in 0,...,m$. Autrement dit, dans chaque intervalle $]b_{k-1}, b_k]$ nous pouvons observer un segment de droite qui relie $(b_{k-1}, L(b_{k-1}))$ et $(b_k, L(b_k))$. Notons que cette fonction a des caractéristiques semblables aux splines utilisées dans les modèles additifs généralisés. La section 3.3 du chapitre suivant fournira une comparaison plus approfondie entre ces méthodes. La propriété de continuité de la courbe de contribution est souvent préférable, surtout pour l'interprétation. Une courbe continue offre une compréhension plus claire de la relation entre la variable explicative et la réponse du modèle.

Nous avons repris le même exemple que précédemment afin de visualiser l'aspect de cette courbe $L(x)$ en pratique. Par souci de simplicité, et pour avoir un visuel explicite, nous avons fixé arbitrairement la valeur des estimations β. Le graphique 2.1 montre que la fonction est effectivement linéaire par morceaux et continue. A l'intérieur d'un intervalle $]b_{k-1}, b_k]$, la pente est constante. De plus, si un des coefficients de régression est nul alors les pentes des intervalles adjacents seront constantes. Cette situation est représentée sur le graphique 2.2.

Exemple 2.3.3.1 *Reprenons la même situation que dans les exemples précédents avec un individu âgé de $x = 36$ ans. La variable explicative âge peut être représentée par des variables L, elle est alors transformée en le vecteur $(l_1(x), l_2(x), l_3(x), l_4(x), l_5(x)) = (8, 4, 19, 29, 36)$. La courbe de contribution évaluée à 36 ans est obtenue comme suit, les paramètres ayant été sélectionnés manuellement :*

$$\begin{aligned} L(36) &= \beta_1\, l_1(x) + \beta_2\, l_2(x) + \beta_3\, l_3(x) + \beta_4\, l_4(x) + \beta_5\, l_5(x) \\ &= 0.2 \times 8 + 0.01 \times 4 - 0.15 \times 19 - 0.09 \times 29 + 0.13 \times 36 \end{aligned}$$

Le point correspondant est représenté en rouge sur le graphique de gauche ci-dessous. Le graphique de droite est construit de la même manière à la différence que le paramètre β_3 a été fixé à zéro. Les bornes $(b_0, b_1, b_2, b_3, b_4, b_5)$ de notre modèle ont été renommées (A, B, C, D, E, F). La mise à zéro du paramètre β_3 lié à l'intervalle $]b_2, b_3]$ implique une pente constante aux intervalles adjacents, donc une pente constante pour l'intervalle $]b_1, b_4]$. On observe effectivement une droite de pente constante entre B et E sur le graphique 2.2.

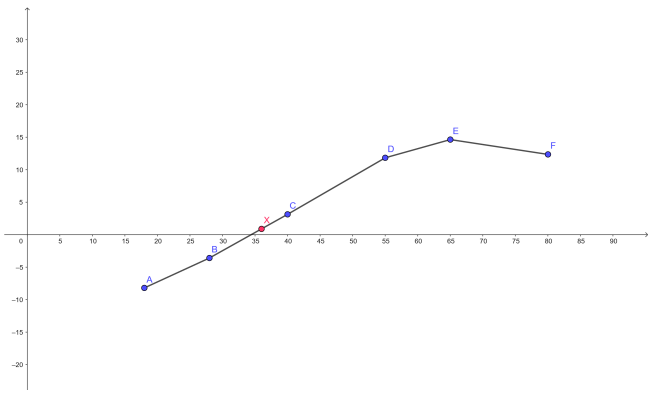

Figure 2.1. Courbe de contribution basée sur les variables L

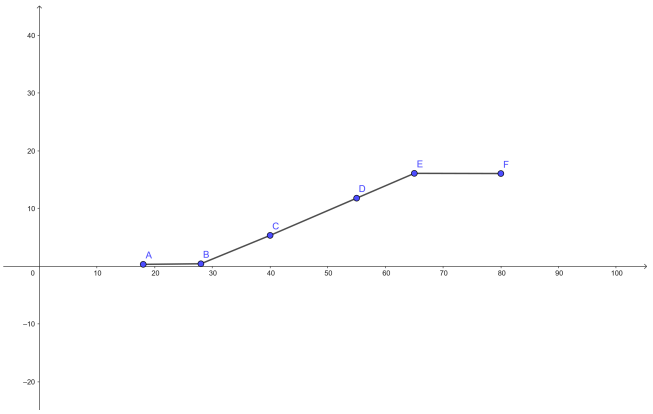

Figure 2.2. Courbe de contribution basée sur les variables L, avec $\beta_3 = 0$

Chapitre 3

Avantages et inconvénients des AGLM

Les Accurate GLM sont structurés pour trouver un équilibre optimal entre le niveau d'interprétabilité désiré et la qualité des prédictions du modèle. Cette balance est réalisée grâce à la combinaison des diverses techniques de machine learning présentées dans le chapitre 2. Dans ce chapitre, nous commencerons par mettre en avant les avantages des Accurate GLM. Ensuite, nous les comparerons à d'autres méthodes de machine learning telles que les GLM, les GAM ou encore les méthodes basées sur les arbres, qui sont déjà largement connues et utilisées par les actuaires.

3.1 Points forts des AGLM

Par définition, l'Accurate GLM repose sur le modèle linéaire généralisé, ce qui garantit une relation explicite entre les variables explicatives et la variable réponse. Cette approche permet d'interpréter facilement l'effet qu'un changement d'une variable explicative peut avoir sur la réponse, en supposant que les autres variables explicatives restent constantes. Dans le secteur de l'assurance, des informations telles que l'âge du preneur d'assurance sont particulièrement importantes. Il est donc primordial de pouvoir déduire les effets de cette variable sur la réponse, ce qui peut s'avérer complexe lorsque nous utilisons des modèles plus sophistiqués comme les GBM ou les réseaux neurones. De plus, ce type de modèle est couramment utilisé dans la tarification des primes d'assurance. Avoir un score facilement interprétable permet une transparence envers le client, facilitant ainsi l'explication du montant de sa prime en fonction de ses caractéristiques personnelles. Cela permet également de répondre aux exigences réglementaires fortement présentes dans le secteur des assurances.

Outre l'interprétabilité d'un modèle, sa précision de prédiction est fondamentale. Le Accurate GLM est construit de manière à améliorer la précision de prédiction du GLM classique. L'introduction des techniques de science des données, abordées dans le chapitre 2, permet aux AGLM d'éviter à la fois le sous-ajustement et le sur-ajustement des données, des problèmes qui surviennent fréquemment lors de l'utilisation de modèles basiques sur des bases de données complexes. En effet, avec l'évolution des technologies, les bases de données deviennent de plus en plus volumineuses, le nombre de variables explicatives disponibles ne cesse d'augmenter, entraînant ainsi une redondance d'informations de plus en plus fréquente. En utilisant une méthode de régularisation pour sélectionner uniquement les variables les plus pertinentes du modèle, nous évitons que le modèle ne s'adapte trop aux données d'entraînement. De plus, la discrétisation et la transformation des variables numériques avant leur intégration dans le modèle permettent de mieux refléter leur relation avec la réponse.

3.2 Comparaison avec les GLM

Depuis des années, le modèle linéaire généralisé est considéré comme une référence dans le secteur de l'assurance, en grande partie en raison de sa facilité d'interprétation. Le Accurate GLM partage cette caractéristique, comme détaillé dans la section 3.1. La principale différence entre ces modèles réside dans leur capacité de prédiction.

Le GLM classique aura des difficultés à atteindre le niveau de prédiction fourni par le modèle AGLM. Pour s'en approcher, des manipulations manuelles des données peuvent être nécessaires. Par exemple, afin de sélectionner les variables explicatives pertinentes dans l'ajustement du modèle, il pourrait être nécessaire d'ajuster plusieurs modèles en sélectionnant à chaque fois un sous-groupe de variables différent, puis de choisir celui offrant la meilleure qualité de prédiction. De même, pour regrouper certaines catégories d'une variable, il faut tester différents modèles basés sur des regroupements différents et choisir le meilleur. Ces processus requièrent une expertise actuarielle non négligeable afin de comparer les modèles et de sélectionner celui répondant le mieux aux critères choisis. C'est réalisable lorsque le nombre de prédicteurs reste relativement petit et connu des actuaires. Le nombre de caractéristiques disponibles devient de plus en plus grand et les nouvelles informations disponibles sont moins maîtrisées par les experts. Ces manipulations peuvent donc être coûteuses en termes de temps. Les AGLM offrent l'avantage d'automatiser ces processus et requièrent ainsi une connaissance moins précise des différentes caractéristiques étudiées.

Comparons plus en détail le codage des variables explicatives dans le GLM par rapport à celui dans le Accurate GLM. Considérons une variable explicative x catégorisée en m classes. Dans un modèle linéaire généralisé, une variable indicatrice, identique aux variables fictives usuelles définies à la section 2.3.1, est définie pour chaque classe k, allant de 1 à $m-1$, différente de la classe de référence. Cette classe doit être choisie par l'actuaire. Généralement il est préférable de la définir comme étant celle qui contient le plus d'observations. La variable explicative x est alors transformée en un vecteur de variables binaires $d_k^U(x)$ pour $k = 1, ..., m-1$. Si elles sont toutes nulles, cela signifie que la valeur de la variable se trouve dans la catégorie de référence. Le choix de cette catégorie est une étape importante étant donné que les analyses qui suivent se font par rapport à ce niveau de référence. Pour rappel, la mise en place de ce niveau de référence est essentiel dans les GLM. Cette approche évite la colinéarité entre les variables fictives usuelles. Dans les AGLM, un codage complet à l'aide des m variables fictives est utilisé. Cela crée de la colinéarité mais, comme discuté dans les sections 1.3.1 et 2.3, la technique de régularisation incluse dans le modèle permet de résoudre le problème. Dans les GLM classique, la régularisation n'est pas utilisée. Ce codage à l'aide des variables fictives usuelles est efficace pour les variables catégorielles non ordonnées. Néanmoins, lorsqu'il y a une notion d'ordre entre les catégories, l'information n'est pas retranscrite à travers les variables indicatrices. En utilisant les variables fictives ordonnées $d^O(x)$ dans un AGLM, les relations ordinales au sein des catégories d'une variable explicative sont maintenues.

Considérons une variable explicative continue x. Son intégration directe dans le score linéaire du GLM limite la capacité du modèle à capturer les relations non linéaires entre cette variable et la réponse. Lorsqu'on inclut la variable directement dans le modèle, on le force à considérer la relation linéaire uniquement. Pour remédier à cela, il est nécessaire de discrétiser la variable en la transformant en une série de variables indicatrices. Cependant, la sélection adéquate des catégories de discrétisation n'est pas toujours évidente et nécessite souvent des tests manuels, comme mentionné précédemment.

De plus, cette approche soulève le même problème que celui évoqué dans le paragraphe précédent, à savoir la perte de la relation ordinale entre les catégories de la variable explicative. En outre, les estimations des paramètres de chaque catégorie de la variable continue peuvent contenir des sauts importants. Il n'y a aucune contrainte de lissage.

La construction des AGLM est basée sur une automatisation des processus de discrétisation et de codage. De plus, ils gagnent en flexibilité en autorisant 3 types de codage selon la nature de la variable explicative. Les variables L utilisées pour le codage des variables continues offrent au modèle la capacité de lisser les résultats. Cela permet au modèle de capter les relations d'ordre entre les catégories des variables explicatives ainsi que l'effet non-linéare de certaines variables.

En bref, bien que les GLM et les AGLM partagent des similitudes dans leur structure de base, les AGLM offrent une plus grande souplesse dans la modélisation du jeu de données. Cela permet de mieux capturer l'effet des variables explicatives sur la réponse et ainsi atteindre une qualité de prédiction supérieure.

3.3 Comparaison avec les GAM

Le modèle additif généralisé a été proposé par Trevor Hastie et Robert Tibshirani (1986). Ils ont élaboré une extension des GLM de sorte à gérer correctement l'introduction des variables continues dans le modèle. La partie du score d'un GAM pour les variables catégorielles est identique à celui d'un GLM. Ensuite, un deuxième terme dédié aux variables continues est ajouté. Mathématiquement, le score se calcule comme suit :

$$score_i = \beta_0 + \sum_{j=1}^{r_{cat}} x_{ij}\beta_j + \sum_{j=r_{cat}+1}^{r} f_j(x_{ij}) \tag{3.1}$$

où r_{cat} est le nombre de variables catégorielles, $r_{con} = r - r_{cat}$ est le nombre de variables continues et les $f_j,\ j = r_{cat+1}, ..., r$ sont des fonctions non spécifiées qui devront être estimées à partir des données avec pour hypothèse qu'elles doivent être lisses. Grâce à cette nouvelle définition du score, le modèle est capable de prendre en compte les relations non linéaires entre les variables et la réponse.

Le GAM maintient le haut niveau d'interprétabilité des GLM. En effet, le score est toujours bien défini de sorte à garder une relation claire entre les variables explicatives et la réponse. Le Accurate GLM a un niveau d'interprétabilité équivalent.

La différence entre les modèles réside dans la structure du score. Pour les variables catégorielles, l'approche est similaire excepté pour celles qui sont ordonnées. Le score d'un GAM pour la partie concernant les variables catégorielles non ordonnées correspond donc au score linéaire du GLM qui est identique à celui du AGLM, au niveau de référence près. Cela a déjà été explicité dans la section 3.2. Si il y a une relation ordinale entre les catégories de la variable, le AGLM utilise le codage par variables fictives ordinales définies à la section 2.3.2. Le AGLM permet au modèle de capter les relations ordinales qui existent entre les catégories de la variable tandis que le GAM n'en tiendra pas nécessairement compte.

Contrairement aux GLM, les scores des GAM et des AGLM autorisent le modèle à considérer les variables continues directement et sont capables de capturer les relations complexes existantes. Afin

d'estimer les fonctions non paramétriques f_j, le GAM utilise des splines de lissage ou des méthodes GLM locales. Pour une étude détaillée de cette seconde méthode, le lecteur est invité à se référer à Denuit, M., Hainaut, D., et Trufin, J. (2019). La méthode des splines est plus couramment utilisée. Ce sont des polynômes par morceaux lissés en choisissant un ensemble de points appelés nœuds. Les splines cubiques sont généralement utilisées dans les GAM. En pratique, des B-splines sont linéairement combinées afin d'obtenir la fonction lissée f du score :

$$f(x) = \sum_{k=1}^{m} \beta_k \, B_k^l(x)$$

où les B-splines d'ordre l sont construites récursivement sur bases- des noeuds sélectionnés. Un GLM peut alors être ajusté sur base des nouvelles variables $B_k^l(x)$ une fois les noeuds définis.

Par la suite, les P-splines sont apparues afin d'avoir plus de flexibilité en considérant un nombre de noeuds plus élevés tout en ajoutant une pénalité sur la somme des carrés des différences de second ordre des coefficients des B-splines successives. L'ajout de cette pénalité offre un compromis entre le lissage de la fonction f et la qualité de l'ajustement. Cette méthode est devenue la norme à utiliser dans le secteur de l'assurance lors de l'ajustement des GAM. Pour avoir une explication plus détaillée des diverses avantages des P-splines, le lecteur peut se référer à l'article "Why P-splines ?" écrit par Paul Eilers and Brian Marx (2021) suite à la publication de leur livre.

Dans les AGLM, les variables continues x sont transformées en variables fictives ordinales $d_k^O(x)$ ou en variables L $l_k(x)$. Au vu des définitions de ces fonctions décrites aux sections 2.3.2 et 2.3.3, il est évident que nous faisons face à un cas particulier des GAM où des P-splines de degré 1 sont générées. La différence réside dans le choix de la pénalité. Le modèle AGLM utilise la méthode de régularisation Elastic Net au lieu d'appliquer une pénalité sur les différence entre les coefficients adjacents des P-splines. L'idée est de prendre un nombre de noeuds initial suffisamment grand pour autoriser une flexibilité suffisante dans le modèle. Ensuite, la partie liée à la pénalité Lasso incluse dans Elastic Net contraint certains coefficients à zéro. Le modèle est alors plus parcimonieux, avec un nombre de noeuds raisonnable. Cette approche est similaire à celle des A-splines proposée par Goepp, V., Bouaziz, O. et Nuel, G. (2018). Une recherche sur leur aptitude à performer en pratique a été menée par Seck, N. A., et Denuit, M. (2022).

De plus, le score du GAM défini à l'équation (3.1) peut contenir un terme supplémentaire autorisant le modèle à considérer les interactions entre les différentes variables du jeu de données. Le score est redéfini par la formule suivante

$$\sum_{j=1}^{r_{cat}} x_{ij}\beta_j + \sum_{j=r_{cat}+1}^{r} f_j(x_{ij}) + \sum_{j,k} f_{jk}(x_{ij}, x_{ik})$$

Lorsqu'on parle d'interaction entre variables, cela signifie que l'effet d'une variable sur la réponse sera différent selon la variation d'une autre variable. Actuellement, la notion d'interaction n'a pas encore été développée en profondeur pour les AGLM.

De ce fait, le AGLM échoue lorsque l'on veut analyser des variables de dimension 2. Dans le secteur de l'assurance, il n'est pas rare que la réponse de notre modèle reste constante à l'intérieur d'une même zone géographique. Pour tarifier les produits d'assurance, il est donc intéressant de regarder

la variable `zone géographique`. L'introduction des fonctions multivariées à l'intérieur du score permet de capturer l'effet des variables géographiques. Il suffit de considérer les variables `longitude` et `latitude` et d'analyser l'effet de leur combinaison sur la variable réponse. Dans le Accurate GLM, le score n'introduit pas automatiquement de fonctions multivariées qui analysent ce type de combinaison de variables. Il n'est dans pas possible de mettre un place un zoning.

Remarquons également que, dans les GAM, si certains niveaux d'une variable catégorielle ne sont pas significativement différents alors il est peut être utile de procéder à un regroupement manuel des catégories afin d'améliorer l'ajustement de notre modèle. Pour ce faire, il faut analyser la base de données. Tester plusieurs regroupements possibles. Mesurer la performance de chaque modèle et sélectionner le regroupement qui optimise cette mesure de performance. Grâce au processus de régularisation, les Accurate GLM sont capables de capturer automatiquement les niveaux qui sont pertinents au modèle.

3.4 Comparaison avec les modèles basés sur les arbres

Les modèles linéaires généralisés et les modèles additifs généralisés sont des approches globales qui fournissent une interprétation claire des résultats. Cependant, leur qualité de prédiction n'est pas toujours idéale. C'est pourquoi d'autres méthodes basées sur les arbres de régression existent comme les forêts aléatoires ou la méthode du gradient boosting. L'arbre de régression est une approche séquentielle dans laquelle l'estimateur de la réponse est construit de façon récursive et non pas optimisé globalement.

Un arbre seul est très instable. La forêt aléatoire combine donc un grand nombre d'arbres et nécessite de nombreux paramètres ce qui permet de rendre la prédiction du modèle plus robuste. La méthode du gradient boosting se base sur des arbres de régressions multiples. L'idée est qu'à chaque itération le modèle va construire un nouvel arbre sur base de l'itération précédente en corrigeant les erreurs commises. C'est donc un modèle très puissant qui, en termes prédictif, à tendance à surpasser les autres méthodes de machine learning.

L'utilisation des nombreux paramètres dans ces méthodes permet d'être flexible et d'ajuster au mieux le modèle. Cela permet d'éviter le sur-ajustement et le sous-ajustement. Par conséquent, le modèle est capable de capturer les relations complexes entre les variables explicatives. Néanmoins, sélectionner les paramètres adéquats dans l'ajustement du modèle n'est pas une tâche aisée. Généralement, les paramètres optimaux découlent d'un processus par validation croisée basé sur une grille aléatoire de paramètres. L'association des concepts développés au chapitre 2 et de la régularisation permet aux Accurate GLM de capturer ces relations complexes. Contrairement aux GBM, il ne faut déterminer que les deux paramètres de la régularisation.

Cependant, la structure du gradient boosting implique une prise en compte implicite au travers des arbres de décision des interactions entre les variables explicatives du modèle. La présence d'une interaction entre deux variables explicatives implique que l'impact d'une d'entre elle sur la réponse varie selon le niveau de l'autre. L'effet de la variable sur la réponse n'est donc plus indépendant par rapport aux autres caractéristiques. Il est important qu'un modèle puisse capturer ces relations complexes afin d'adapter sa prédiction. Les GLM et GAM ne sont pas capables de le faire automatiquement. Il est possible de créer ces interactions manuellement en créant de nouvelles variables explicatives. Si

les variables x_{j1} et x_{j2} interagissent, alors une nouvelle variable bivariée dépendant de x_{j1} et x_{j2} est ajoutée. De façon analogue, les Accurate GLM ne tiennent pas automatiquement compte des interactions. Il reste possible de les ajouter manuellement. Si l'interaction est représentée par le produit des variables, l'interaction risque d'être trop simpliste. Elle pourrait être quadratique ou plus complexe encore.

Contrairement aux GLM, GAM et Accurate GLM, les méthodes basées sur les arbres ne détiennent pas de formule explicite permettant d'avoir une relation claire entre les variables explicatives et la variable réponse. En effet, il n'existe pas de formule explicite comme c'est le cas pour le score des autres méthodes. Il existe néanmoins des moyens pour représenter l'impact de chaque variable explicative sur la réponse moyenne. Les graphiques de dépendances partielles montrent l'effet marginal moyen d'une variable sur la prédiction de la réponse. Si un profil de risque est définit, le graphique de dépendance partielle de chaque variable montre la tendance de l'effet de cette variable sur la réponse. Cependant, les prédictions fournie dépendront toujours des interactions introduites automatiquement dans le modèle. L'effet d'une variable ne peut donc pas être isolé. Dans un Accurate GLM, la structure est claire. Les estimations des paramètres du modèle sont donc directement interprétable. C'est un avantage non négligeable lorsque le modèle doit être expliquer à un public inexpérimenté. Un modèle transparent est également préféré pour remplir les exigences réglementaires du secteur de l'assurance.

Au sein de la communauté actuarielle, les GLM restent largement utilisés, malgré l'émergence croissante de nouvelles techniques. Passer d'un GLM à un GBM requiert beaucoup de nouvelles connaissances. L'avantage de passer à un AGLM est qu'il garde le même cadre qu'un GLM. Le passage de l'un à l'autre peut donc se faire plus rapidement.

Chapitre 4

Application numérique

Dans les chapitres précédents, nous avons exploré en détail la méthodologie des AGLM ainsi que leurs avantages et inconvénients d'un point de vue théorique. Dans ce chapitre, nous allons mettre en pratique ce modèle en l'appliquant à une base de données réelle, puis analyser les résultats obtenus. Cette démarche nous permettra d'illustrer les points forts de cette nouvelle méthode, tout en mettant en lumière ses éventuelles limitations. Nous procéderons également à une comparaison quantitative avec d'autres méthodes de machine learning couramment utilisées, telles que les GLM, les GAM et la méthode de gradient boosting.

4.1 Description du jeu de données

Dans le secteur de l'assurance, les méthodes de machine learning sont couramment utilisées, notamment pour anticiper le nombre de sinistres futurs d'un assuré ou pour estimer les coûts associés. Nous allons commencer en appliquant notre modèle à un jeu de données portant sur des polices de responsabilité civile liées à l'assurance automobile. Le jeu de données sélectionné `freMTPLfreq` provient du package `R CASdatasets`. Il est constitué de 413 169 observations et nous offre des informations pertinentes dans le cadre de la tarification d'assurance. De plus, il contient des variables explicatives continues, catégorielles et catégorielles ordonnées. Ainsi, les différents concepts et codages définis à la section 2.3 seront représentés dans cette application numérique.

Avant toute chose, commençons par une brève analyse descriptive de ce jeu de données. Des graphiques sont disponibles en annexe. Il contient 3 variables explicatives continues :

- L'âge du conducteur en années qui varie entre 18 ans, à savoir l'âge minimum légal du permis de conduire, et 99 ans.
- L'âge du véhicule en années qui varie entre 0 et 100 années.
- La densité d'habitants, i.e. le nombre d'habitants par kilomètre carré, dans la ville où vit le conducteur de la voiture.

Il contient 1 variable catégorielle ayant une relation d'ordre entre les différentes catégories de la variable :

- La puissance du véhicule classée en 12 catégories ordonnées allant de "d" à "o".

Il contient aussi 3 variables catégorielles non ordonnées :

- Le type de carburant classé en 2 catégories : essence ou diesel
- La région catégorisée en 10 groupes
- La marque du véhicule en 7 catégories

Ce jeu de données contient les trois types de variables analysés lors de chapitres précédents. Il permettra donc d'appliquer les différents concepts définis dans la modélisation de l'Accurate GLM. De plus, le jeu de données contient également certaines autres informations. L'identifiant de la police d'assurance qui sert à relier notre jeu de données à celui contenant la sévérité des sinistres. Cette information ne nous est pas utile ici, nous pouvons donc enlever cette information du jeu de données. L'exposition au risque de l'assuré en années polices est aussi disponible ainsi que le nombre de sinistres pour chaque police durant la période d'exposition.

En analysant de plus près les variables `Density` et `CarAge`, il ressort que garder uniquement un sous-ensemble du jeu de données semble plus approprié. En effet, la dernière colonne de la table 4.1 indique les quantiles à 95% des variables explicatives. Par définition, cela représente la valeur de la variable telle que 95% des observations de la distribution ont une valeur inférieure ou égale et, par conséquent, 5% des observations sont supérieures. Le jeu de données initial `freMTPLfreq` est filtré sur base de ces deux variables et est réduit aux observations satisfaisant aux deux conditions suivantes :

$$\begin{cases} \texttt{Density} \leq 10000 \\ \texttt{CarAge} \leq 25 \end{cases}$$

Le filtrage de ce jeu de données réduit le nombre d'observations de 413 169 à 393 386.

Variable	Minimum	Maximum	Moyenne	Quantile 95%
`Density`	2	27000	1985	8023
`CarAge`	0	100	7.532	17

Table 4.1. Analyse descriptive des variables `Density` et `CarAge`

Dans les sections suivantes, nous allons analyser ce jeu de données filtré et ajuster différents modèles de machine learning. Cela permettra de prédire la variable réponse Y, à savoir le nombre de sinistres. La distribution de la réponse est supposée de loi de Poisson. En effet, c'est un processus de comptage d'un nombre de sinistres assez faible compris entre 0 et 4. Cette loi appartient à la famille Exponentielle de dispersion, le cadre des GLM, et par conséquent des AGLM, est donc bien respecté. Par ailleurs, le logarithme est la fonction lien sélectionnée afin d'imposer le caractère positif de la réponse moyenne, un nombre de sinistres ne pouvant pas être négatif. Le graphique 4.1 associe à chaque nombre de sinistres possibles, le nombre de polices d'assurance associées.

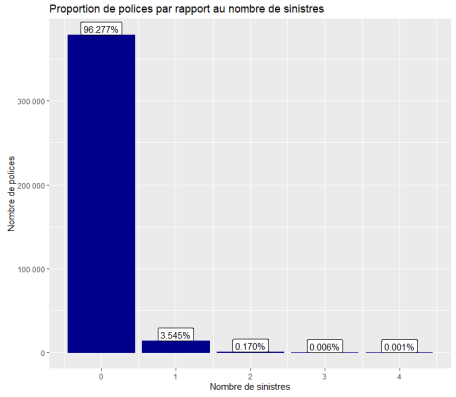

Figure 4.1. La proportion de polices d'assurance selon le nombre de sinistres

Nous pouvons calculer la fréquence moyenne des sinistres dans ce portefeuille en divisant le nombre total de sinistres du portefeuille par la somme des expositions au risque. On obtient :

$$\frac{15\ 374}{222\ 450.6} = 6.911196\%$$

Dans le modèle que nous allons ajuster, nous voulons prédire la fréquence des sinistres selon les caractéristiques spécifiques d'un individu.

4.2 Accurate GLM

Nous allons ajuster un Accurate GLM sur le jeu de données décrit à la section 4.1. Pour ce faire, nous allons utiliser le package `aglm` disponible depuis 2019 sur GitHub, une plateforme de développement qui permet aux développeurs de créer, stocker, gérer et partager leur code, et actuellement disponible sur CRAN depuis 2021.

4.2.1 Description du package R

Le souhait des actuaires, Fujita, Tanaka, Kondo et Iwasawa, ayant mis en lumière la nouvelle méthode de modélisation, les Accurate GLM, est de la faire connaître au sein de la communauté des sciences actuarielles et encourager les actuaires à l'adopter plus systématiquement. Pour ce faire, ils ont créé un package R intitulé `aglm`, destiné à promouvoir l'utilisation des Accurate GLM. Ces derniers visent à offrir une haute précision dans les prédictions tout en conservant l'interprétabilité des modèles classiques comme les GLM. Un des avantages de ce package est qu'il se base sur un autre package déjà connu dans le secteur des actuaires, à savoir `glmnet`. Les fonctions disposent de la même structure ce qui permet aux utilisateurs une maîtrise plus rapide.

Ce package R est composé de diverses fonctions semblables à celles que nous pouvons retrouvons pour d'autres méthodes de modélisation et qui permettent d'ajuster les modèles, de les visualiser et de prédire la variable réponse à partir de nouvelles données. La fonction principale nommée `aglm` a pour but d'ajuster un modèle sur le jeu de données.

```
model <- aglm(x, y, family = "", offset = , alpha = , lambda = ,
                  use_LVar = , nbin.max = )
```

Pour ajuster le modèle, le jeu de données initial est divisé en deux blocs, *x* un data frame contenant les variables explicatives et *y* un vecteur contenant la variable réponse. Ensuite, différents paramètres peuvent être fixés dont les paramètres de régularisation λ et α définis dans l'équation (2.5).

Dans la définition de cette fonction `aglm`, les paramètres de régularisations λ et α sont définis arbitrairement par l'actuaire. Les valeurs prises par défaut sont α égal à 1, i.e. la pénalité de Lasso, et une séquence construite aléatoirement pour λ selon deux paramètres fixés, `nlambda` égal à 100 et `lambda.min.ratio` la valeur minimale possible de λ qui dépend de la taille du nombre d'observations et du nombre de variables explicatives.

Afin de rendre ce choix moins arbitraire, deux fonctions basées sur la méthode de la validation croisée, explicitée à la section 1.4, ont été créées. La fonction `cv.aglm` est une fonction d'ajustement

avec validation croisée pour le paramètre de régularisation λ en fixant α égal à 1 comme valeur par défaut. Il existe également la fonction `cva.aglm` qui applique la validation croisée pour les paramètres λ et α. Cependant, la complexité computationnelle de cette dernière entraîne des temps de calcul significatifs. Il est donc préférable de l'éviter si cela n'apporte pas de plus-value à nos modèles. Ces deux fonctions ont la même structure de base que la fonction `cv.glmnet` du package `glmnet` qui utilise, par défaut, la déviance comme mesure de performance. Selon la distribution de la réponse, le choix de la mesure de performance peut différer, comme expliqué à la section 1.4.

4.2.2 Préparation des données

Les différentes variables explicatives que nous allons utiliser dans l'ajustement de nos modèles sont définies à la section 4.1. Rappelons que le modèle `aglm` a pour but d'automatiser l'ingénierie des caractéristiques définie au chapitre 2. Selon le type de la variable explicative, de nouvelles variables fictives usuelles ou ordinales vont être créées ainsi que les variables L. La table 4.2 décrit le type des variables explicatives définies par R et le codage appliqué par la fonction `aglm`. Remarquons que la variable `Power` est considérée par le logiciel R comme une variable catégorielle non ordonnée. Or, la définition de cette variable dans le package `CASdatasets` nous indique que les catégories sont ordonnés de *d* à *o*. Nous allons donc redéfinir cette variable comme une variable catégorielle ordonnée à l'aide de la fonction `clean factor` du package `cleaner` de sorte à ce que le codage crée des variables fictives ordinales à la place des variables fictives usuelles.

Variables explicatives	Type	Codage dans R
`Power`	Factor	Variable U
`CarAge`	Integer	Variable O ou Variable L
`DriverAge`	Integer	Variable O ou Variable L
`Brand`	Factor	Variable U
`Gas`	Factor	Variable U
`Region`	Factor	Variable U
`Density`	Integer	Variable O ou Variable L

Table 4.2. Variables explicatives provenant de `freMTPLfreq`

Avant d'ajuster nos modèles, nous allons diviser le jeu de données en deux sous-ensembles, le jeu d'entraînement et le jeu de vérification. Le premier est utilisé dans la calibration de nos modèles. Ensuite, le deuxième est utilisé pour vérifier le modèle et sa qualité de performance. Un des critères de la qualité de la performance du modèle est le calcul de la déviance sur le jeu de validation. Ces deux ensembles sont créés aléatoirement, 80% des données du jeu de données de base constituent l'ensemble d'entraînement et 20% le jeu de vérification.

Au vu de la section 1.4, notre réponse est supposée suivre une loi de Poisson, nous utiliserons donc la déviance de Poisson définie comme suit

$$\frac{1}{n}\sum_{i=1}^{n}(y_i \ln\frac{y_i}{\hat{\mu}_i} - (y_i - \hat{\mu}_i))$$

avec $\hat{\mu}_i$ l'estimation de la moyenne de la réponse, y_i la valeur observée de la réponse et n le nombre

d'observations incluses dans la jeu d'entraînement en cas d'ajustement ou bien dans le jeu de validation lorsqu'on compare la performance de différents modèles.

4.2.3 Ajustement du modèle AGLM

Contrairement aux GLM, les Accurate GLM sont directement ajustés sur le jeu d'entraînement sans nécessiter une transformation des variables préalable. La discrétisation des variables continues et le codage se font directement dans l'ajustement du modèle. Afin de déterminer les paramètres optimaux de la régularisation, nous ajustons notre modèle à l'aide de la fonction `cv.aglm` définie à la section 4.2.1 en se basant sur x un data frame contenant les 7 variables explicatives et y un vecteur contenant la variable réponse. Dans le cadre de cette application numérique et au vu du nombre d'observations, il semble que l'utilisation de la fonction `cva.aglm` n'est pas adéquate. En effet, le temps nécessaire à l'exécution de cette fonction est de plusieurs heures. Par souci de rigueur, nous la testerons quand même. Les résultats obtenus sont indiqués à la première ligne de la table 4.3. Le paramètre optimal α étant fixé à 1, cela revient à se restreindre à une pénalité de Lasso. Or, cette méthode de régularisation a tendance à rendre des résultats moins réguliers dû à sa caractéristique de sélection de variables. De plus, comme nous avons pu le voir dans la section 2.3, le codage utilisé pour les variables catégorielles dans un modèle `aglm` fait apparaître de la colinéarité entre les variables fictives. La régularisation de Ridge étudiée à la section 1.3.1 doit donc être utilisée afin de gérer ce problème. C'est pourquoi le modèle aglm a été construit de sorte à pouvoir mixer les pénalités de Lasso et de Ridge au travers de la régularisation Elastic Net. L'ajout de la pénalité de Ridge permet de mieux gérer la colinéarité entre les variables et rend alors des résultats plus réguliers. Dans le cas du modèle `aglm_cva`, le caractère de sélection des variables issu de la pénalité de Lasso est trop marqué. Une grande partie des coefficients β du modèle sont fixés à 0 et cela nous donne des résultats qui n'arrivent pas à capter les tendances dans les données. Les estimations des β sont données dans les annexes.

En outre, il est nécessaire de fixer d'autres paramètres afin que l'ajustement soit le plus adéquat possible. Au vu du caractère de comptage de la variable réponse, la distribution est supposée de loi de Poisson et la fonction lien utilisée est le logarithme. Dans un modèle de fréquence, il est également important de prendre en compte l'exposition au risque représentée par la partie de l'année durant laquelle le véhicule est assuré. Il rentre dans le modèle à l'aide de l'offset que nous fixons égal au logarithme de l'exposition. Par ailleurs, la fonction `aglm` crée, par défaut, des variables fictives ordinales pour les variables numériques. Or, nous avons défini à la section 2.3.3 les variables L qui assurent la continuité de la courbe de contribution et sont préférables dans le cas des variables continues. Afin que ces variables soient utilisées, il suffit d'ajouter l'option `use_LVar = TRUE` dans l'ajustement de notre modèle, les autres paramètres étant égaux à leurs valeurs par défaut. Ainsi, le modèle discrétise chaque variable numérique automatiquement le plus finement possible en sélectionnant les intervalles $[b_{k-1}, b_k]$ dont la longueur est minimum. Ensuite, la régularisation se charge de déterminer si le niveau doit être inclus ou non dans le modèle. Ce processus est décrit à la section 2.2. Il est également possible de contrôler le nombre maximum de niveaux désiré dans la discrétisation en fixant le paramètre `nbin.max`. Dans ce cas-ci, le choix est fait arbitrairement par l'actuaire. Notons que ce paramètre est unique et attribué à chaque variable continue.

Commençons par appliquer la fonction *cv.aglm* dans 3 situations particulières du paramètre α à savoir la pénalité de Lasso, la pénalité de Ridge et la pénalité Elastic Net avec $\alpha = 0.5$. Le nombre de plis utilisés dans la validation croisée est fixé à 5 et la mesure de performance est la déviance

de la loi de Poisson. Les résultats sont indiqués à la table 4.3. Le paramètre λ du modèle cv_lasso est plus petit que celui du modèle aglm_cva. La pénalité étant réduite, moins de coefficients β sont contraints à zéro. Le modèle cv_ridge ne fixe aucun paramètre à zéro étant donné que cette méthode de régularisation ne permet pas de sélectionner des variables. Ainsi, les résultats obtenus pour Ridge sont nettement plus lisses que ceux de Lasso. Les courbes de contribution dans le cas de Ridge seront donc plus lisses, et donc, se rapprocheront plus des courbes obtenues avec l'ajustement d'un GAM. En termes de performance, le modèle basé sur Lasso minimise la déviance définie à la section 1.4. Celui basé sur la régularisation de Ridge le succède en termes de déviance mais rend des résultats plus réguliers.

Modèle	λ	α	Déviance
`aglm_cva`	0.0002519597	1	0.127039
`cv_lasso`	5.494939e-06	1	0.1269164
`cv_ridge`	0.000646651	0	0.1269610
`cv_elasticnet`	2.987695e-06	0.5	0.1269731

Table 4.3. Résultat de la validation croisée, utilisation des fonctions du package R

Le modèle `cv_elasticnet` utilise quant à lui les deux pénalités ce qui permet de trouver un compromis entre une haute qualité prédictive et une régularité des résultats. Les modèles de la table 4.3 échouent à chaque fois sur l'un de ces deux critères. Nous allons étendre le champ de recherche en ajustant des modèles pour différentes combinaisons de paramètres $alpha - lambda$ afin d'obtenir le compromis le plus adéquat possible. Définissons manuellement une grille de paramètres α et λ et

α \ λ	0.00	0.01	0.04	0.09	0.16	0.25
0	0.124906	0.1251157	0.1253959	0.1256386	0.1258040	0.1259275
0.1	0.124906	01257477	0.1267229	0.1272331	0.1272331	0.1272331
0.2	0.124906	0.1262389	0.1272331	0.1272331	0.1272331	0.1272331
0.3	0.124906	0.1264953	0.1272331	0.1272331	0.1272331	0.1272331
0.4	0.124906	0.1266961	0.1272331	0.1272331	0.1272331	0.1272331
0.5	0.124906	0.1269208	0.1272331	0.1272331	0.1272331	0.1272331
0.6	0.124906	0.1271645	0.1272331	0.1272331	0.1272331	0.1272331
0.7	0.124906	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331
0.8	0.124906	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331
0.9	0.124906	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331
1	0.124906	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331
α \ λ	0.36	0.49	0.64	0.81	1.00	1.5
0	0.1260275	0.1261138	0.1261908	0.1262591	0.1263219	0.1264453
0.1	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331
0.2	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331
0.3	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331
0.4	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331
0.5	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331
0.6	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331
0.7	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331
0.8	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331
0.9	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331
1	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331	0.1272331

Table 4.4. Résultat de la validation croisée, effectuée manuellement

utilisons la validation croisée à 5 plis pour déterminer la zone optimale pour ces paramètres avec comme mesure de performance la déviance du pli faisant office de jeu de validation. La table 4.4 regroupe les résultats obtenus. Notons que cette procédure de validation croisée a eu un temps de run extrêmement conséquent. Elle a également crée une surcharge de mémoire dans le logiciel R utilisé. Il est préférable de s'assurer que l'ordinateur utilisé pour faire tourner cette partie du code est suffisamment puissant afin d'éviter ce type de problème.

La déviance minimale de la table 4.4 est obtenue lorsque λ est égal à 0 et ce quelque soit la valeur du paramètre α. Cette situation revient à ne pas appliquer de pénalité à nos coefficients. Le but des AGLM est d'utiliser ces techniques de science des données pour améliorer le modèle, il semble donc logique de ne pas sélectionner λ égal à 0 comme paramètre optimal. Les résultats nous permettent néanmoins de tester de nouveaux modèles dans une zone de paramètre plus affinée là où les déviances sont minimales, à savoir lorsque les paramètres sont relativement petits. Les résultats sont indiqués dans la table 4.5.

α \ λ	0	0.0001	0.001	0.002
0	0.124906	0.1248672	0.1249169	0.1249583
0.001	0.124906	0.1248650	0.1249278	0.1249647
0.005	0.124906	0.1248666	0.1249363	0.1249571
0.01	0.124906	0.1248686	0.1249284	0.1249783

Table 4.5. Résultat de la validation croisée sur une grille de paramètres plus affinée

Le modèle qui minimise la déviance est celui ayant comme paramètres α égal à 0.001 et λ égal à 0.0001, nommons le `aglm1`. Vérifions que ce modèle est acceptable en termes de régularité. Les graphiques 4.2, 4.3 et 4.4 représentent les courbes de contribution des variables explicatives continues du modèle. Ils suivent les mêmes tendances que les fréquences observées représentées aux graphiques 4.43, 4.44 et 4.45 des annexes. Les courbes possèdent néanmoins des irrégularités, pics et creux, qui sont difficiles à justifier. Les déviances des autres modèles sont relativement proches. Nous allons donc chercher parmi les autres qui ont une performance similaire, un modèle qui fournit des résultats plus lisses.

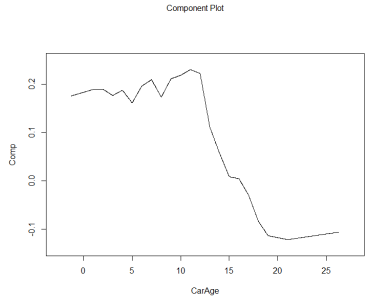

Figure 4.2. Courbe de contribution du modèle `aglm1` pour la variable `CarAge`

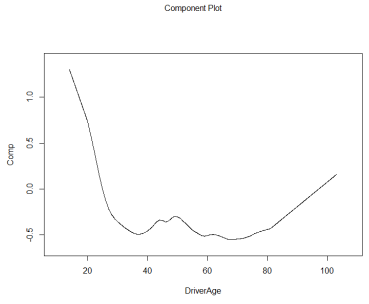

Figure 4.3. Courbe de contribution du modèle `aglm1` pour la variable `DriverAge`

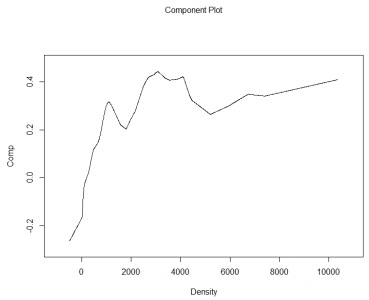

Figure 4.4. Courbe de contribution du modèle `aglm1` pour la variable `Density`

Les modèles les plus proches en termes de performance sont ceux ayant un paramètre λ égal à 0.0001. Les résultats de ces modèles sont semblables à ceux du modèle `aglm1` en termes de lissage, le critère de régularité n'est donc pas optimisé. Regardons les modèles ayant le paramètre λ égal à 0.001, et en particulier celui avec α égal 0.01. Ce modèle, que nous noterons `aglm2`, maintient une bonne performance tout en améliorant la régularité des prédictions. Par conséquent, le modèle `aglm2` ressort être le modèle le plus adéquat pour notre base de données.

4.2.4 Résultat du modèle AGLM

Analysons plus en détail le processus et les résultats du modèle `aglm2`. Les 3 variables catégorielles x_j du jeu de données ont été transformées respectivement en m_j variables fictives usuelles où m_j est le nombre de catégories de la variable x_j. La variable catégorielle ordonnée a été transformée en m_j variables fictives ordinales. La fonction `plot` nous fournit la représentation graphique de la contribution de chacune de ces variables. Elles sont illustrées sur les graphiques 4.5 à 4.8. Les valeurs numériques des estimations des paramètres du modèle sont énumérées dans les annexes. Remarquons que les estimations des paramètres de la variable `Power` ne sont pas aussi régulières que ce à quoi on se serait attendu grâce à l'utilisation des variables fictives ordonnées. Néanmoins si nous avions utilisé le codage par variables fictives usuelles, la variabilité entre les catégories aurait été nettement plus marquée. Les estimations des paramètres pour ces deux cas de figure sont énumérées dans l'annexe à la figure 4.46.

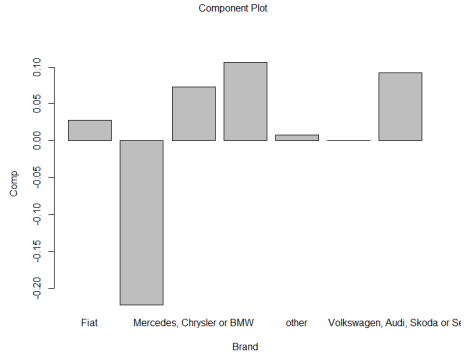

Figure 4.5. Estimation des coefficients β du modèle `aglm2` pour la variable `brand`

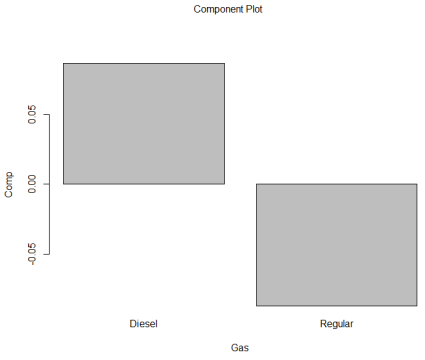

Figure 4.6. Estimation des coefficients β du modèle `aglm2` pour la variable `gas`

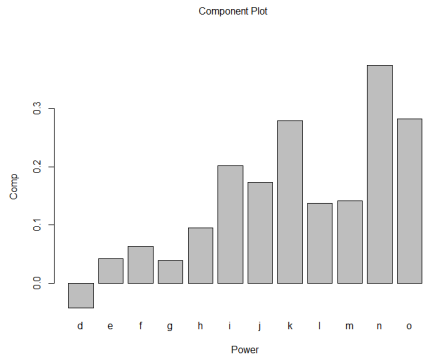

Figure 4.7. Estimation des coefficients β du modèle `aglm2` pour la variable `power`

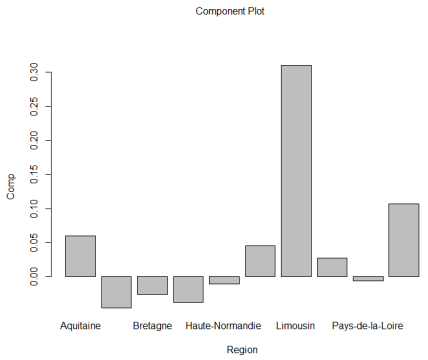

Figure 4.8. Estimation des coefficients β du modèle `aglm2` pour la variable `region`

Concernant les variables continues, elles ont été discrétisées avant d'être transformées en variables L. Le tableau 4.6 indique le nombre de catégories créées pour chacune de ces variables. Ces choix sont cohérents avec les distributions des variables continues représentées aux graphiques 4.9, 4.10 et 4.11. Pour l'âge du véhicule, chaque âge de 0 à 19 constitue une borne b_k, $k \in \{0, 22\}$. Le modèle considère que chacune de ces catégories a une importance dans l'ajustement du modèle car le nombre de polices par âge est suffisant. Ensuite, la quantité de polices entre 19 et 25 ans est réduite. Le modèle fixe donc une dernière borne à 21 ans.

Variable	**m = nombre de catégories**
`CarAge`	22
`DriverAge`	56
`Density`	98

Table 4.6. Nombre de catégories fixé dans la discrétisation des variables continues

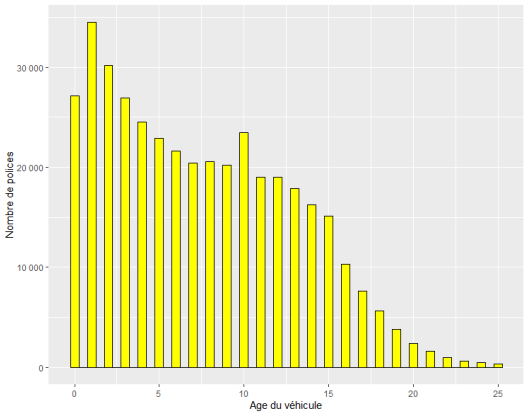

Figure 4.9. Distribution du nombre de polices par rapport à l'âge du véhicule

Des conclusions similaires sont obtenues pour l'âge du conducteur. Jusqu'à 70 ans, le nombre de polices par âge est suffisant pour que chaque âge représente une catégorie significative dans le

modèle. Ensuite de 70 à 80, le modèle regroupe les âges par 2 ou 3 et finalement il sépare le reste en 2 catégories.

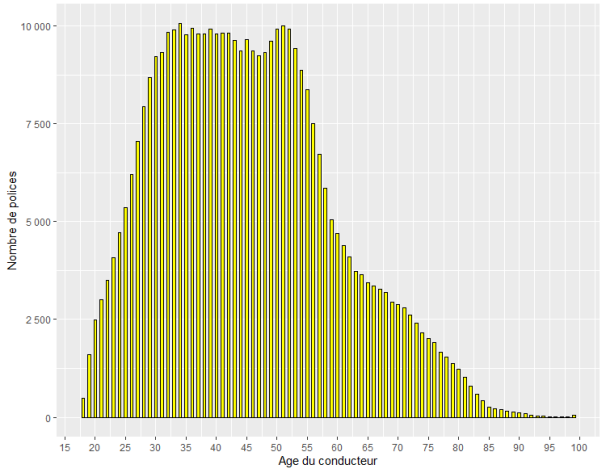

Figure 4.10. Distribution du nombre de polices par rapport à l'âge du conducteur

La distribution de la variable densité de population montre que la majorité des polices se situent dans les faibles densités. Le premier quartile est égal à 64. Cela signifie qu'un quart des observations ont une densité inférieure ou égale à 64. C'est pourquoi les premières bornes sont proches les unes des autres de 2 à 100. Elles s'éloignent ensuite de plus en plus car la quantité de polices est moindre.

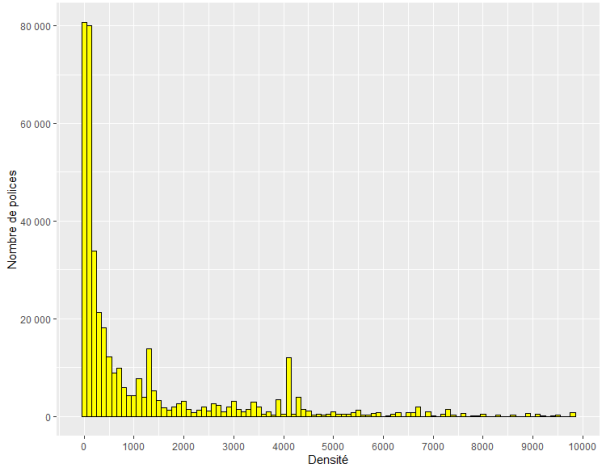

Figure 4.11. Distribution du nombre de polices par rapport à la densité de population

Les graphiques 4.12, 4.13 et 4.14 représentent les contributions de chaque variable continue dans le modèle `aglm2`. Le graphique 4.12 indique qu'au plus la voiture est récente, au plus la probabilité qu'un sinistre se produise est élevée. Le graphique 4.13 montre qu'au plus le conducteur est jeune, au plus le risque qu'un sinistre se produise est élevé. Remarquons également qu'un pic est présent entre 40 et 60 ans, cela correspond en général aux parents qui laissent la voiture à leur enfant jeune conducteur. Ensuite, passé 80 ans il y a à nouveau une augmentation du risque de sinistre.

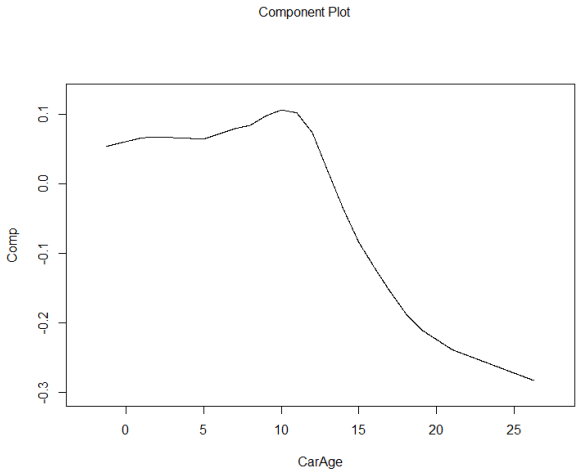

Figure 4.12. Courbe de contribution du modèle `aglm2` de la variable `CarAge`

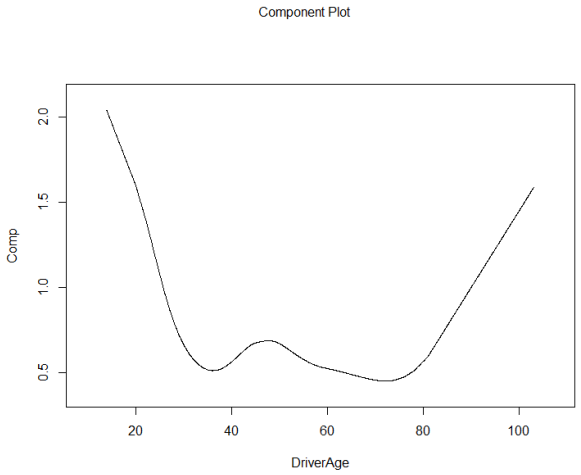

Figure 4.13. Courbe de contribution du modèle `aglm2` de la variable `DriverAge`

Le graphique 4.14 indique qu'au plus la population est dense, au plus les risques de sinistre sont élevés. C'est cohérent avec la réalité observée. À partir d'une densité de 5000, les observations disponibles sont moins présentes d'où la stabilité de la courbe au-delà de cette valeur.

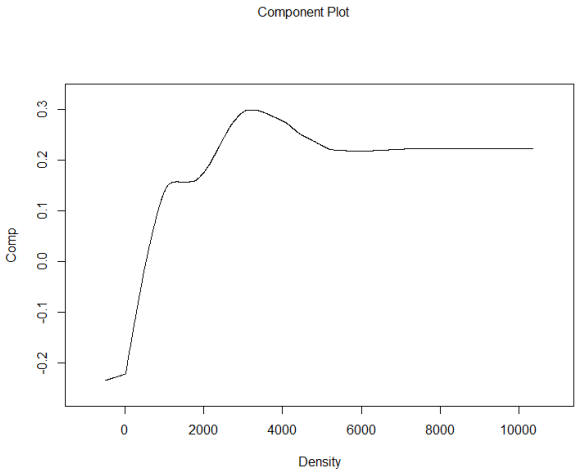

Figure 4.14. Courbe de contribution du modèle `aglm2` de la variable `Density`

4.2.5 Justification des options utilisées dans notre modèle

L'utilisation de la discrétisation et du codage a été justifiée théoriquement. Regardons si, en pratique, un choix différent conduit nécessairement à un modèle moins performant. Par exemple, ajustons le modèle `aglm3` en utilisant les mêmes valeurs de paramètres que le modèle de fréquence `aglm2` sans utiliser le codage en variable L. Ajustons également le modèle `aglm4` en limitant le nombre maximum de niveaux de la discrétisation. Les résultats obtenus sont présentés dans la table 4.7.

Modèle	λ	α	Option	Deviance
aglm2	0.001	0.01	use_LVar = TRUE	0.1269741
aglm3	0.001	0.01	use_LVar = FALSE	0.1268554
aglm4	0.001	0.01	use_LVar = TRUE et nbin.max = 10	0.1270845

Table 4.7. Résultat des ajustements des Accurate GLM

Le modèle aglm3 surpasse légèrement le modèle aglm2 en terme de performance. Cependant, le modèle aglm3 échoue au niveau du lissage des courbes de contribution des variables continues CarAge, DriverAge et Density. En effet, en utilisant uniquement des variables fictives ordinales pour coder les variables continues, la courbe de contribution devient instable. Graphiquement, il est évident que les courbes de contribution du modèle aglm2 représentées sur les graphiques 4.13 et 4.14 sont plus adéquates que celles du modèle aglm3 représentées sur les graphiques 4.15 et 4.16. Par conséquent, l'utilisation des variables L est justifiée et l'option use_LVar = TRUE doit être utilisée afin d'avoir des résultats plus lisses avec des sauts moins importants entre les différentes adjacentes de la variable.

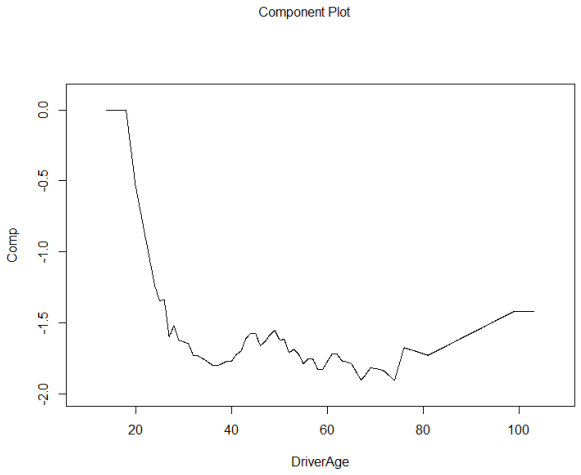

Figure 4.15. Courbe de contribution du modèle aglm3 de la variable DriverAge

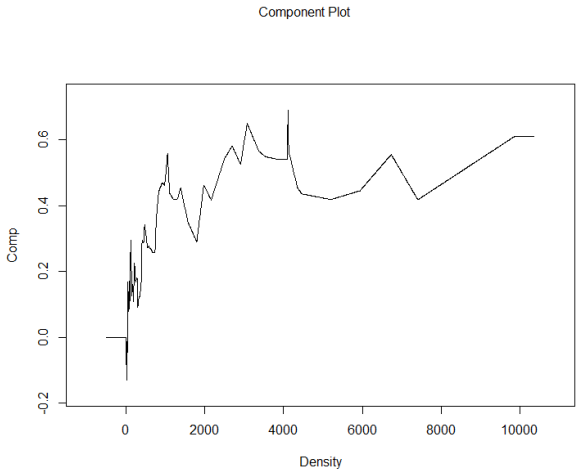

Figure 4.16. Courbe de contribution du modèle aglm3 de la variable Density

Le modèle aglm2 surpasse le modèle aglm4 en terme de performance. De plus, déterminer de manière arbitraire le nombre maximal de niveaux autorisé pour la discrétisation des variables continues est une tâche complexe. Comme discuté dans la section 2.2, le paramètre *m* doit être ni trop grand, ni trop petit afin d'éviter un sur-ajustement ou un sous-ajustement des données. Le graphique 4.17 représente la courbe de contribution de la variable DriverAge discrétisée en 10 niveaux. Le lissage n'est pas assez précis en comparaison du graphique 4.13. De plus, le paramètre nbin.max est appliqué sur toutes les variables. Or, certaines variables disposent d'un ensemble de valeur conséquent et ont donc besoin d'une discrétisation plus fine. Par exemple, la contribution de la variable Density, illustrée sur le graphique 4.18 avec un nombre maximum de 10 niveaux est complètement insuffisante.

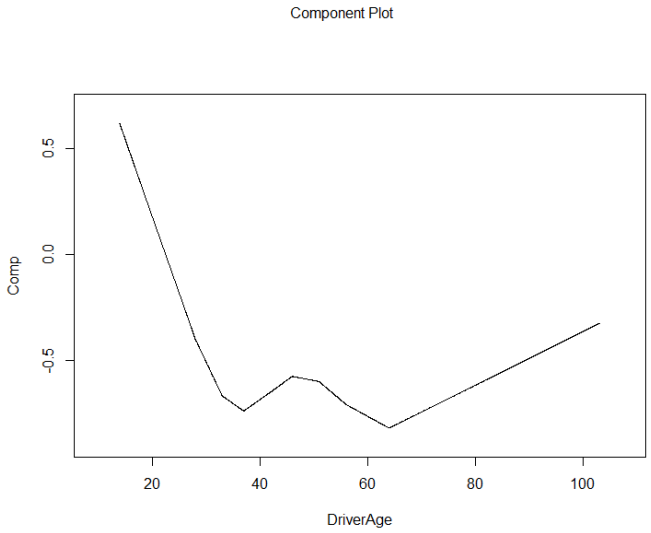

Figure 4.17. Courbe de contribution du modèle `aglm4` de la variable `DriverAge`

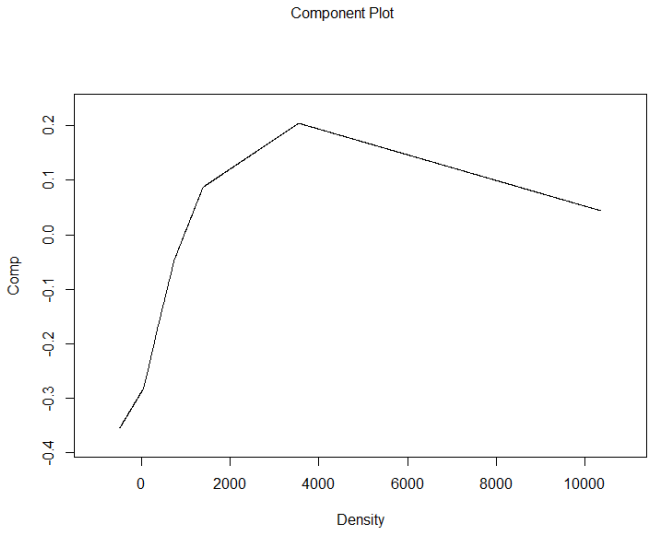

Figure 4.18. Courbe de contribution du modèle `aglm4` de la variable `Density`

4.3 Comparaison avec les modèles classiques

Nous allons ajuster différents modèles sur le même jeu de données défini à la section 4.1. Nous gardons la même sous division en jeu d'entraînement et jeu de vérification. Afin de déterminer quantitativement la qualité de performance du modèle Accurate GLM par rapports aux autres, nous allons calculer la déviance de chaque modèle sur le jeu de données de validation.

Les modèles que nous allons utiliser sont indiqués dans le tableau 4.8. Chacun de ces modèles nécessite le chargement du package R approprié. Nous avons supposé dans l'ajustement de l'Accurate GLM, l'utilisation de la distribution de Poisson ainsi que le logarithme comme fonction lien. Ces deux hypothèses sont maintenues pour les ajustements des modèles suivants.

Modèle	**Package**
Modèle Linéaire Généralisé	stats (chargé par défaut)
Modèle Additif Généralisé	mgcv
Méthode du Gradient Boosting	gbm3
Modèle Linéaire Généralisé régularisé	glmnet, glmnetUtils

Table 4.8. Modèles ajustés dans l'application numérique et leur package R

4.3.1 Ajustement des modèles

Commençons par ajuster les modèles sur les données brutes. Cela signifie que nous allons directement ajuster les modèles sans faire de modification préalable des variables. Les modèles sont construits comme suit :

- Le premier modèle linéaire généralisé `glm1` inclut les variables continues sans les catégoriser et n'introduit pas d'interaction entre les variables ni de regroupement de catégories. La fonction `step` est utilisée afin de sélectionner les variables. Si le modèle amputé d'une variable rend un AIC (Akaike Information Criterion) plus faible alors elle exclut la variable du modèle. Dans notre cas, il n'y a pas d'amélioration de l'AIC donc les 7 variables sont incluses dans l'ajustement du modèle. Il est prévisible que le GLM échouera en termes de prédiction. Les analyses effectuées dans les sections 4.2.3 et 4.2.4 nous indiquent clairement que les variables continues n'impactent pas linéairement la réponse. Nous ajustons donc un deuxième modèle `glm2` qui inclus les variables continues discrétisées. La discrétisation utilisée est fournie dans les annexes.
- Le modèle additif généralisé prend en compte les variables catégorielles initiales et utilise la fonction `s(...)`, qui se base sur les splines, pour lisser les variables continues.
- Le modèle de boosting par gradient est sélectionné par validation croisée à 5 plis. Les différents modèles ajustés sont indiqués dans la table 4.9. Chaque paramètre joue un rôle important dans l'ajustement du modèle. Une grille de paramètres est crée et le GBM le plus prédictif est celui minimisant la déviance de Poisson. C'est donc le premier modèle qui est choisi.

Modèle	`num_trees`	`interaction_depth`	`min_num_obs_in_node`	`shrinkage`	`best_iter`	`déviance`
1	700	10	500	0.02	564	0.1266837
2	600	10	500	0.01	585	0.1266983
3	400	5	450	0.01	400	0.1268015
4	500	5	500	0.001	500	0.1280666
5	350	10	500	0.01	585	0.1266983
6	600	5	1000	0.01	600	0.1267327
7	300	5	500	0.01	300	0.1268828

Table 4.9. Ajustement des GBM et choix des paramètres

- Le modèle linéaire généralisé régularisé fixe les paramètres α et λ de la régularisation à l'aide de la fonction *cv.glmnet* exécutée pour plusieurs valeurs du paramètre α. La table 4.10 indique les paramètres λ optimaux obtenus pour chaque modèle selon le α qui a été fixé. Le modèle qui atteint les meilleures performances en termes de déviance est celui avec $\alpha = 1$ et $\lambda = 0.0001986852$. Par ailleurs, remarquons que l'ajustement de ce modèle ne permet pas de transformer automatiquement les variables catégorielles en variables fictives usuelles. Il est nécessaire de transformer le data frame *x* en une matrice de variables fictives usuelles préalablement à l'ajustement du modèle.

α	λ	**Déviance**
0	0.001353626	0.1290003
0.01	0.001353626	0.1289632
0.3	0.000662284	0.1288503
0.5	0.0003973704	0.128817
0.7	0.000283836	0.1288004
0.8	0.0002483565	0.1287949
1	0.0001986852	0.1287869

Table 4.10. Validation croisée à 5 plis pour le modèle linéaire généralisé régularisé

Le code utilisé pour ajuster ces modèles ainsi que les estimations des paramètres sont fournis en

annexe.

4.3.2 Importance des variables

Lors de l'ajustement d'un modèle, il est important de comprendre quelles variables ont le plus d'influence sur les prédictions. Le modèle gradient boosting est capable de nous fournir immédiatement l'importance des variables. Le graphique 4.19 montre que les variables `DriverAge` et `Density` sont les plus importantes du modèle. Les estimations des paramètres associés à ces variables sont indiqués en annexe. Ils sont effectivement significatifs dans les différents modèles étudiés. Par ailleurs, lors de l'ajustement d'un GLM régularisé par Lasso, ce sont les uniques variables considérées dans le modèle.

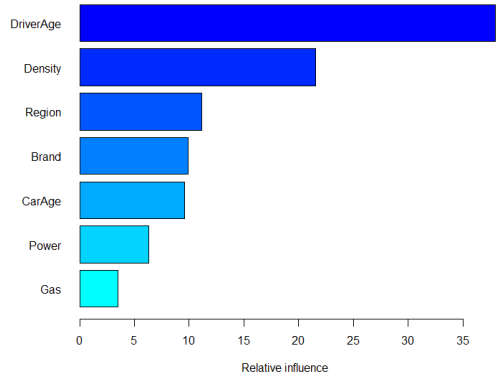

Figure 4.19. Importance des variables, modèle GBM

4.3.3 Résultat et comparaison des modèles

Les résultats de la mesure de performance des modèles sont indiqués dans la table 4.11. Le modèle Accurate GLM se situe en troisième position, après le GBM et le GAM. Ensuite, nous retrouvons les GLM en quatrième place et cinquième place et, pour finir, le GLM régularisé. En termes de déviance, l'Accurate GLM n'apparaît pas comme le premier choix. Toutefois, il se positionne devant le GLM ce qui est essentiel étant donné que le but premier de l'Accurate GLM est de garder les avantages d'interprétabilité des GLM tout en augmentant sa qualité de prédiction. Il aurait cependant été opportun que le modèle Acuurate GLM arrive en tête de classement dans la mesure de performance. Cela aurait permis de conclure directement qu'en terme de qualité de prédiction, il apparaît être le meilleur.

Modèle	Déviance
`glm1`	0.1282058
`glm2`	0.1272804
`gam`	0.1268885
`gbm`	0.1266837
`glm régularisé`	0.1284160
`aglm2`	0.1269741

Table 4.11. Performance des ajustements

Comparons de manière générale l'impact des variables explicatives sur la réponse pour le GLM, le GLM régularisé, le GAM et le GBM avec celui de l'Accurate GLM. Les comparaisons par rapport au GLM seront basées sur le modèle `glm1`. Bien que le modèle `glm2` améliore la déviance, l'idée est d'opposer à l'AGLM un modèle sans transformation de variable et d'analyser quantitativement les écarts de prédiction qui apparaissent.

Au vu du caractère linéaire du score d'un GLM, les courbes de contributions sont représentées graphiquement par une droite de pente $\hat{\beta}$. Cette hypothèse empêche le modèle de capturer les relations non linéaires des variables par rapport à la réponse. Les estimations des coefficients de la régression sont respectivement $-1.038e-02$ et $-9.787e-03$ pour les variables `CarAge` et `DriverAge`. Il s'agit de droites décroissantes. Plus les âges sont élevés, plus la fréquence de sinistre est faible. Pour la variable `Density`, la pente est de $8.211e-05$ ce qui mène à une droite croissante. Plus la densité est élevée, plus les sinistres seront fréquents. Bien que ces courbes ne soient pas lisses par rapport aux données et empêchent de capturer certaines relations, ces caractéristiques sont cohérentes avec les croissances et décroissances observées sur les courbes de contribution 4.12, 4.13 et 4.14 des AGLM.

Pour visualiser au mieux et comparer les impacts des variables de chaque modèle, nous allons superposer les courbes de l'impact des variables continues de tous les modèles sur un même graphique. Nous ferons de même pour les variables catégorielles en obtenant une valeur par catégorie au lieu d'une courbe. Autrement dit, nous analysons de plus près les différences entre les prédictions de chaque modèle comme décrit ci-après.

Pour chaque variable explicative, nous allons calculer la fréquence de sinistre pour un sous groupe particulier variant selon la variable explicative analysée. Nous ferons de même pour un individu de référence. Le rapport de ces fréquences nous permettra d'examiner l'effet de la variable explicative tout en ayant neutralisé l'influence du profil sélectionné. Refaire le même exercice avec un nouveau profil nous mènera donc au même résultat de sorte que l'impact de la variable ait été correctement isolé, excepté pour le GBM. La table 4.12 indique la valeur de chaque variable explicative dans les sous-groupes utilisés pour les comparaisons. Nous fixerons l'exposition au risque à 1 an.

Variable analysée	**DriverAge**	**CarAge**	**Density**	**Gas**	**Power**	**Region**	**Brand**
DriverAge	{18,...,99}	3	200	Diesel	f	Centre	Renault, Nissan or Citroen
CarAge	35	{0,...,25}	200	Diesel	f	Centre	Renault, Nissan or Citroen
Density	35	3	{2,...,9850}	Diesel	f	Centre	Renault, Nissan or Citroen
Gas	35	3	200	...	f	Centre	Renault, Nissan or Citroen
Power	35	3	200	Diesel	...	Centre	Renault, Nissan or Citroen
Region	35	3	200	Diesel	f	...	Renault, Nissan or Citroen
Brand	35	3	200	Diesel	f	Centre	...

Table 4.12. Sous-groupe sélectionné pour la comparaison des modèles

La table 4.13 indique pour chaque variable explicative la catégorie de référence choisie. Pour les variables catégorielles, elle correspond à la catégorie contenant le plus d'observations. Pour les variables continues, nous choisissons une valeur qui se situe dans une zone avec une forte concentration d'observations.

DriverAge	**CarAge**	**Density**	**Gas**	**Power**	**Region**	**Brand**
35	3	200	Diesel	f	Centre	Renault, Nissan or Citroen

Table 4.13. Choix du niveau de référence

Les effets des variables explicatives continues sont représentés aux graphiques 4.20, 4.22 et 4.24 avec une courbe par modèle. Concrètement, le modèle GBM, indiqué en pointillé vert, ne peut pas être superposé et comparé via cette courbe aux autres modèles. En effet, comme expliqué précédemment, le gradient boosting introduit automatiquement les interactions au vu de sa structure d'arbres de décision successifs. Ainsi, si la variable dont l'effet est étudié est influencée par une autre variable explicative, alors l'effet de cette variable ne sera pas isolé. Le choix d'un autre profil impliquera une modification dans l'effet de la variable sur la réponse. La tendance de l'effet de la variable peut toutefois être observé sur les graphiques de dépendance partielle représentés aux figures 4.21, 4.23 et 4.25.

En ce qui concerne la variable `DriverAge`, la tendance qui ressort des modèles sur le graphique 4.20 est une fréquence nettement plus élevée avant 35 ans et un pic aux alentours des 50 ans. L'hypothèse de linéarité du GLM l'empêche de représenter ce pic. Les effets obtenus à l'aide du modèle GAM et AGLM arrivent à capter la tendance générale et le pic. La différence entre ces modèles apparaît aux grands âges. Or, nous savons qu'aux valeurs extrêmes, le modèle additif généralisé est moins performant. Cela est dû au manque de données et, par conséquent, une extrapolation trop importante. Le graphique 4.21 suit la même tendance bien que le pic des 50 ans est un peu moins marqué. Le GLM régularisé rend un résultat assez différent des autres modèles. Le Lasso ne garde en réalité que deux variables avec un coefficient non nul, l'âge du conducteur et la densité. Nous ajoutons l'impact de ces deux variables pour le modèle GLM régularisé avec Ridge sur les graphiques 4.20 et 4.22. La modèle basé sur Ridge suit la tendance décroissante de l'effet de l'âge du conducteur contrairement au Lasso. Par contre, il est le modèle avec la moins bonne mesure de performance dans la table 4.10.

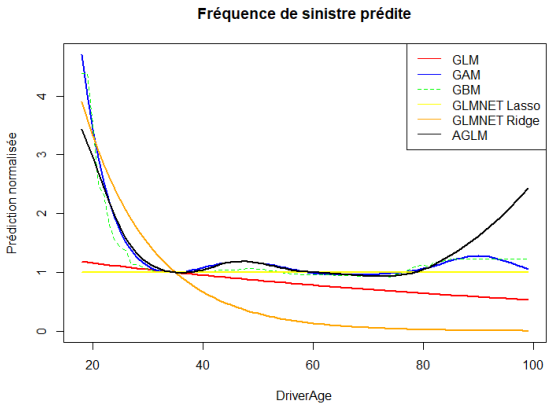

Figure 4.20. Impact de la variable `DriverAge` sur la fréquence de sinistre prédite pour un sous groupe donné

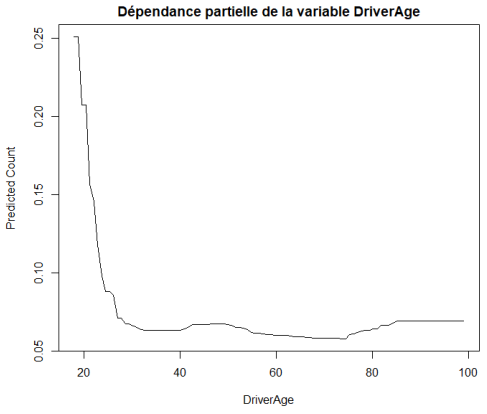

Figure 4.21. Graphique de dépendance partielle du modèle GBM, variable `DriverAge`

Pour la variable `CarAge`, le graphique 4.22 montre que la fréquence est plus élevée aux alentours des 10 ans. Avant elle reste relativement stable par rapport à l'âge de référence et après elle chute fortement. C'est ce qui est observé en pratique. À nouveau, les modèles GAM et AGLM prédisent un effet similaire pour la variable `CarAge` sans décrochage et ce, même aux valeurs extrêmes. Notons que l'âge du véhicule dans notre base de données s'étendait jusqu'à 100 ans et que nous avons filtré pour garder les âges inférieurs à 25 ans. Cela a permis au modèle de maintenir un niveau d'observations acceptable aux nouvelles valeurs extrêmes. Le GBM montre les mêmes tendances sur le graphique 4.23, i.e. une fréquence élevée au début et faible aux grands âges. Le GLM régularisé via Lasso

s'éloigne complètement de la tendance générale. Celui basé sur Ridge rend un résultat similaire au GLM.

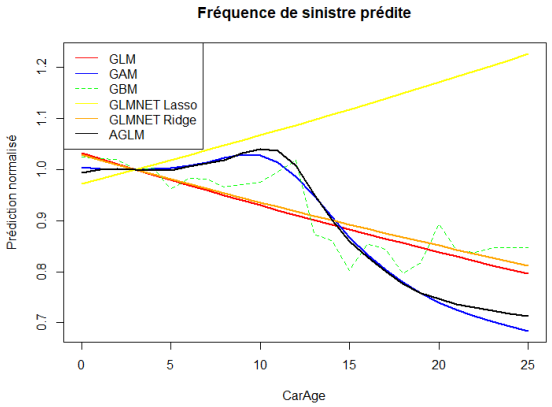

Figure 4.22. Impact de la variable `CarAge` sur la fréquence de sinistre prédite pour un sous groupe donné

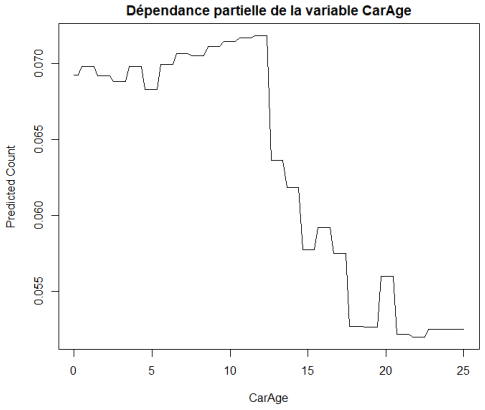

Figure 4.23. Graphique de dépendance partielle du modèle GBM, variable `CarAge`

Pour la variable `Density`, le graphique 4.24 indique que lorsque le nombre d'habitants par mètre carré est plus élevé alors la fréquence de sinistre sera plus élevée que celle du niveau de référence. Les modèles GAM et AGLM sont les plus réguliers et les plus semblables. On retrouve la même tendance sur le graphique 4.25 du GBM bien que la courbe y soit plus irrégulière. Le GLM régularisé est incohérent avec la tendance observée. Par ailleurs, celui basé sur Ridge subit un tel décrochage qu'il n'est pas pertinent de le représenter sur ce graphique.

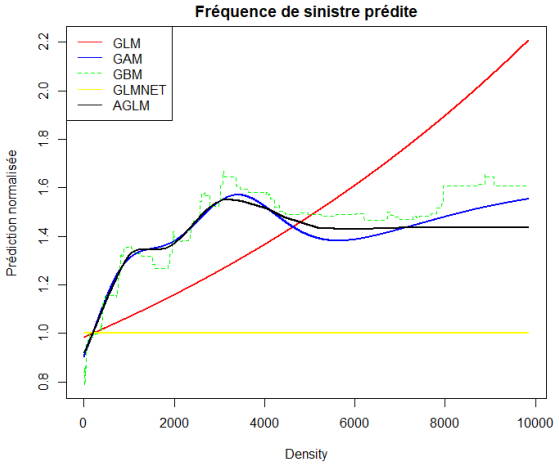

Figure 4.24. Impact de la variable `Density` sur la fréquence de sinistre prédite pour un sous groupe donné

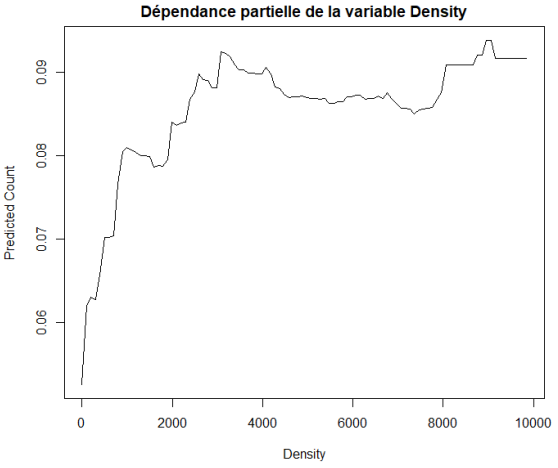

Figure 4.25. Graphique de dépendance partielle du modèle GBM, variable `Density`

Remarquons que les 3 graphiques de dépendance partielle du GBM ont une allure plus similaire aux courbes de contribution du premier modèle Accurate GLM considéré, `aglm1`, alors que le `aglm2` est plus semblable au GAM. Le `aglm1` surpassait le `aglm2` en termes de déviance et, parallèlement,

le GBM surpasse le GAM. En termes de régularités des résultats on a le résultat inverse. La nature du modèle gradient boosting est la cause du non lissage de ces courbes. En effet, contrairement aux GAM qui utilisent des fonctions lissées, comme les splines, les GBM sont des modèles séquentiels qui divisent l'ensemble des valeurs des variables explicatives et fournissent donc des prédictions par morceaux. Contrairement aux GLM qui imposent une hypothèse de linéarité dans la relation entre réponse et variable explicative, les GBM n'en imposent pas et sont construits sur base des données de sorte à pouvoir capturer les relations non linéaires et complexes.

Les effets des variables explicatives catégorielles sont indiqués dans la table 4.14. Lorsque l'effet est égal à 1, cela signifie que la fréquence de sinistre prédite pour le niveau considéré est égale à celle de la catégorie de référence. Un effet supérieur (inférieur) à 1 signifie que la fréquence de sinistre prédite est plus élevée (faible) que celle du niveau de référence.

Variable	Catégories	glm	gam	aglm	gbm	glmnet
Gas	Diesel	1	1	1	1	1
	Regular	0.8803901	0.8345553	0.8405353	0.8430258	1.084422
Region	Centre	1	1	1	1	1
	Aquitaine	1.131014	1.109708	1.102965	1.158224	1.005455
	Basse-Normandie	1.031836	0.9865497	0.9924908	1.022933	1.089175
	Bretagne	1.039025	1.006258	1.012242	1.043771	0.8914217
	Haute-Normandie	1.020825	1.024298	1.027928	1.074535	0.992274
	Ile-de-France	1.061014	1.093083	1.086696	1.130478	0.9462403
	Limousin	1.367633	1.439342	1.417158	1.481694	1.005455
	Nord-Pas-de-Calais	1.139122	1.071473	1.067818	1.110083	1.005455
	Pays-de-la-Loire	1.059587	1.029043	1.032194	1.039951	1.223613
	Poitou-Charentes	1.183074	1.158534	1.155929	1.238725	1.051837
Power	d	0.9131157	0.8992740	0.9001557	0.9196303	0.9904360
	e	0.9836467	0.9776367	0.9788041	0.9886933	0.9909674
	f	1	1	1	1	1
	g	0.9579761	0.9759700	0.9765248	0.9449591	1.3277675
	h	1.0032864	1.0327291	1.0326972	0.9991877	0.9999213
	i	1.1237977	1.1516171	1.1491786	1.0377436	0.9999213
	j	1.0975665	1.1217563	1.1166945	1.1268372	0.9999213
	k	1.203738	1.251028	1.241534	1.236738	1.030051
	l	1.040673	1.075184	1.076709	1.175775	0.950125
	m	1.0314685	1.0788153	1.0817869	1.2208931	0.9999213
	n	1.346722	1.385912	1.365641	1.298979	1.013024
	o	1.1829319	1.2592415	1.2452604	1.2989793	0.9784492
Brand	Renault, Nissan or Citroen	1	1	1	1	1
	Fiat	1.038510	1.030376	1.028209	1.007231	1.000000
	Mercedes, Chrysler or BMW	1.0754405	1.0745432	1.0752775	1.0524249	0.7729334
	Opel, General Motors or Ford	1.121365	1.113858	1.112501	1.106109	1.039810
	Volkswagen, Audi, Skoda or Seat	1.108680	1.097335	1.096844	1.087022	0.989166
	other	0.9867832	1.0100362	1.0079260	1.0165871	1.0926853

Table 4.14. Impact sur la fréquence des variables catégorielles

De manière générale, les effets obtenus avec le GAM et l'AGLM sont très similaires. Ceux du modèle GLM s'en écartent davantage, mais restent néanmoins cohérents avec la tendance observée pour le GAM et l'AGLM. Pour le modèle GLMNET basé sur Lasso, nous remarquons un décrochage assez significatif par rapport aux tendances. Ce sont des conclusions similaires à celles obtenues pour

les variables continues. Nous avons également calculé de la même façon un effet approximatif des variables catégorielles pour le GBM. Cependant, ce résultat n'est pas correct pour les mêmes raisons qu'évoquer dans le cas des variables continues.

4.4 Interactions

De manière générale, nous faisons l'hypothèse que les variables explicatives sont indépendantes. Cependant, il se peut que l'effet d'une variable sur la réponse ne soit pas le même suivant une autre variable. C'est ce qu'on appelle une interaction. La statistique de Friedman nous permet de représenter sur la Figure 4.26 la force d'interaction qui existe entre chaque couple de variables explicatives.

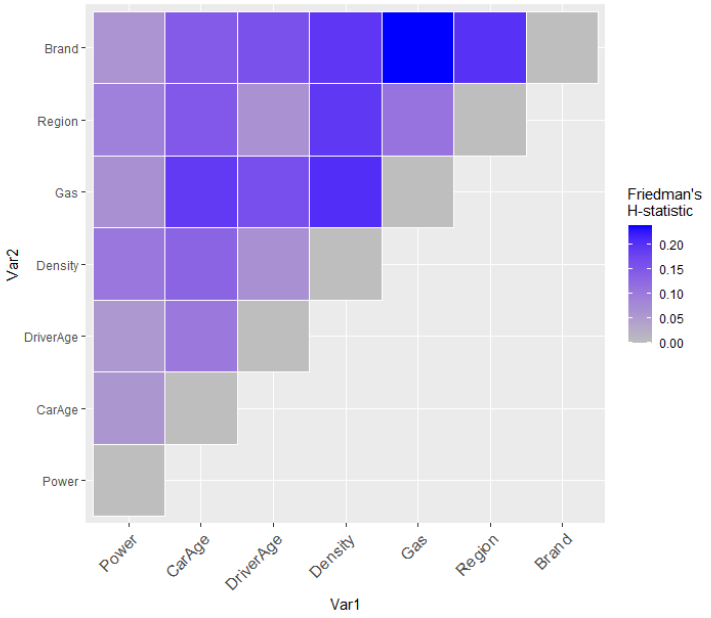

Figure 4.26. Analyse des interactions, statistique de Friedman

Force est de constater qu'il existe certaines interactions dans notre base de données. Il aurait été intéressant de les ajouter lors de l'ajustement de nos modèles. Dans un Accurate GLM, il est possible d'ajouter ces interactions aux modèles à l'aide de l'option `add_interaction_columns=TRUE`. Le problème est qu'il n'est pas encore possible de sélectionner certaines variables. En effet, cette option crée des variables supplémentaires $x_{j1} * x_{j2}$ pour chaque paire de variables (x_{j1}, x_{j2}). Le but n'est pas d'alourdir et de complexifier le modèle avec des interactions inutiles. Nous n'avons donc pas utilisé cette option et, par conséquent, nous n'avons pas intégré les interactions dans le GLM et le GAM afin de garder des modèles comparables. Le fait de ne pas les avoir inclus dans notre modèle peut, en partie, expliquer la raison pour laquelle le GBM surpasse le AGLM au niveau de la déviance.

4.5 Application numérique pour la prédiction des sévérités

Les sections précédentes ont permis de construire un modèle de fréquence sur base d'un jeu de données en assurance automobile. Nous allons appliquer une méthodologie similaire afin de prédire le coût moyen des sinistres de ce même jeu de données. La variable réponse n'est donc plus une variable

de comptage. La loi de Poisson est alors généralement remplacée par une loi Gamma. Elle appartient également à la famille Exponentielle de dispersion et est citée dans la table 1.1. La loi Gamma est appropriée pour modéliser les coûts. Ce sont des variables positives qui présentent généralement une dissymétrie à droite dans leur distribution. En effet, nous avons tendance à rencontrer beaucoup de sinistres à faible coût et peu de sinistres à coût élevé.

Néanmoins, le package `aglm` se limite à un ensemble de lois pré définies. En effet, actuellement il est restreint aux lois de Poisson, Binomial et Gaussienne. La loi Binomiale est généralement utilisée pour modéliser l'occurrence des sinistres. La loi Gaussienne peut être utilisée pour établir le modèle des coûts. En effet, une alternative à la loi Gamma est la loi log-normale. Il suffit alors d'ajuster un `aglm` à l'aide de la famille gaussienne en modifiant préalablement la variable réponse en son logarithme.

Le jeu de données utilisé est `freMTPLsev` du package `CASdataset`. Il contient uniquement le numéro d'identité de la police et le coût du sinistre associé à 15 390 polices. Dans un premier temps, il faut agréger les sinistres par police et diviser par le nombre de sinistres de la police. Ensuite, il faut aller rechercher les informations correspondantes des variables explicatives dans la base `freMTPLfreq`. Afin d'analyser les mêmes contrats que ceux considérés dans le modèle de fréquence, nous allons appliquer le même filtre. Les observations avec une densité supérieure à 10 000 et un véhicule de plus de 25 ans sont exclues.

Le graphique 4.27 montre les coûts moyens des sinistres du plus petit au plus élevé. Le jeu de données contient certaines valeurs extrêmes. Il sera tronqué pour éviter qu'elles n'influencent le modèle de manière inadéquate en faussant les estimations des paramètres. La loi log-normal ne permet pas de représenter correctement les valeurs extrêmes. Dans le cadre de notre travail, nous gardons les sinistres avec un coût inférieur à 15 000 € et nous ne traiterons pas des sinistres extrêmes.

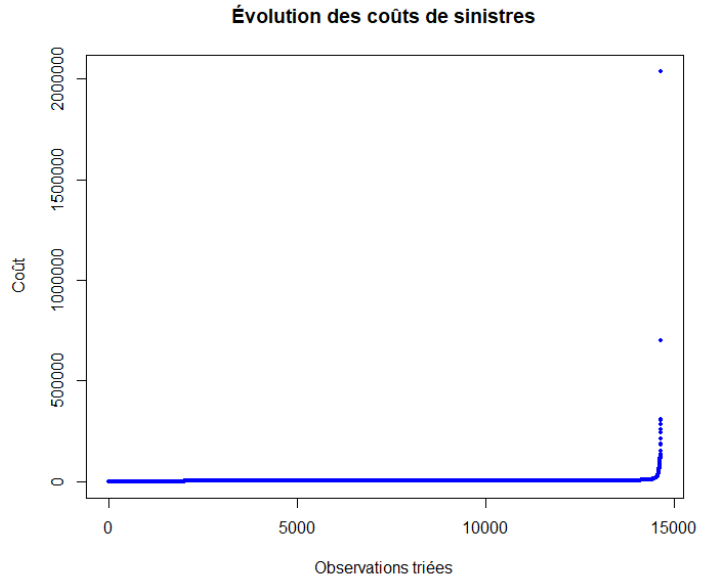

Figure 4.27. Coût moyen des sinistres, du plus petit au plus élevé

Plusieurs AGLM sont ajustés à l'aide de la fonction `cv.aglm`. Le nombre d'observations est for-

tement réduit donc la fonction tourne rapidement. Les résultats sont dans la table 4.15. Dès que le paramètre α s'éloigne de 0, la pénalité de Lasso prend le dessus. De nombreuses estimations des paramètres du modèle sont fixées à zéro. Afin d'analyser des résultats relativement lisses, nous choisissons de garder le modèle avec $\alpha = 0$.

α	λ	**Déviance**
0	0.08196678	1.674063
0.05	0.001639336	1.631357
0.3	0.0002732226	1.624094
0.5	0.0008748641	1.625272
0.7	0.001198508	1.625996
1	0.0009207525	1.621323

Table 4.15. Validation croisée à 5 plis pour l'AGLM, modèle des sévérités

Les résultats du modèle sont représentés graphiquement aux figures 4.28 à 4.34. Ainsi, nous pouvons conclure quelle catégorie impacte les coûts des sinistres à la hausse ou à la baisse.

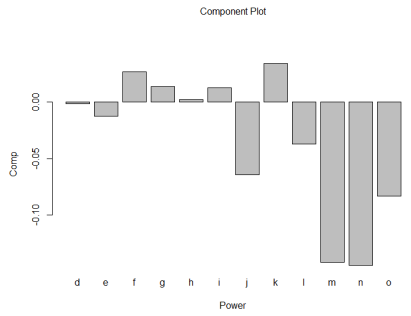

Figure 4.28. Estimation des coefficients β du modèle `aglm` pour la variable `power`

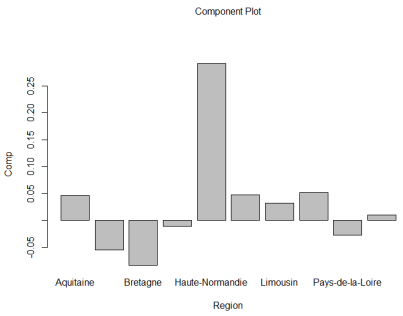

Figure 4.29. Estimation des coefficients β du modèle `aglm` pour la variable `region`

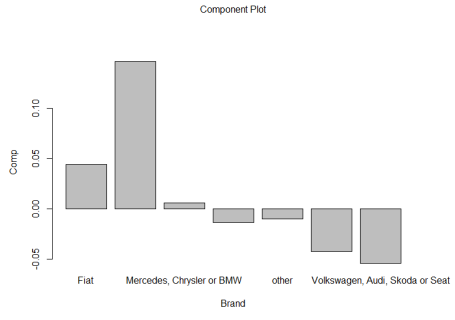

Figure 4.30. Estimation des coefficients β du modèle `aglm` pour la variable `brand`

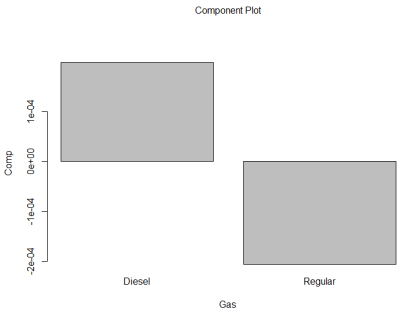

Figure 4.31. Estimation des coefficients β du modèle `aglm` pour la variable `gas`

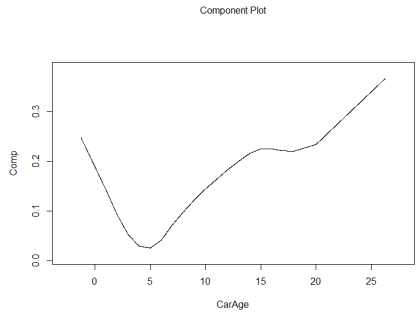

Figure 4.32. Courbe de contribution du modèle `aglm` pour la variable `CarAge`

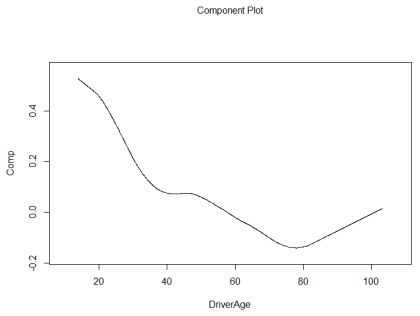

Figure 4.33. Courbe de contribution du modèle `aglm` pour la variable `DriverAge`

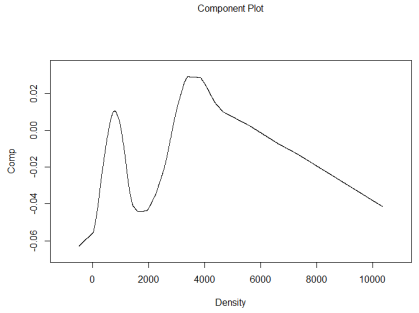

Figure 4.34. Courbe de contribution du modèle `aglm` pour la variable `Density`

Lorsqu'un modèle de fréquence et un modèle de sévérité sont ajustés, il est possible d'en déduire un modèle pour le calcul de la prime pure. Pour un profil déterminé, la fréquence et la sévérité peuvent être prédites. La prime pure du preneur d'assurance sera égal au produit de la fréquence prédite avec la sévérité prédite.

Conclusion

Dans ce travail nous avons mis en lumière une nouvelle méthode de modélisation : les Accurate GLM. Cette approche vise à fournir un modèle qui combine efficacement interprétabilité et précision des prédictions. Les modèles couramment utilisés, ont tendance à remplir un seul de ces deux critères. Or, au vu de la croissance continue des bases de données disponibles dans le secteur de l'assurance, il est crucial d'apprendre à utiliser des méthodes qui maintiennent un haut niveau de prédiction et qui, simultanément, sont capables de simplifier le modèle en repérant les variables les plus pertinentes.

Dans les chapitres 1 et 2, la structure des Accurate GLM ainsi que les techniques de sciences des données utilisées ont été définies. Brièvement, un Accurate GLM possède la même structure qu'un GLM en fournissant une flexibilité plus importante au modèle grâce à la régularisation et au codage des variables explicatives. Le maintien de la structure du score équivalent à celle d'un GLM permet à l'Accurate GLM d'avoir une relation claire entre les variables explicatives et la réponse moyenne. D'autre part, la modification dans le codage des variables ainsi que l'utilisation des méthodes de régularisation permet au modèle de capturer les relations complexes et non linéaires présentes entre les variables explicatives et la réponse moyenne.

Dans le chapitre 3, nous avons illustré qualitativement les points forts des Accurate GLM en les comparant avec les GLM, GAM et GBM. Essentiellement, il en ressort que l'Accurate GLM est supérieur aux GLM. L'Accurate GLM utilise les variables L alors que le GAM génère des P-splines mais globalement les deux méthodes semblent être sur un pied d'égalité en termes d'interprétation et de lissage des courbes de contribution des variables explicative. Contrairement aux Accurate GLM, le GBM est capable d'inclure automatiquement les interactions dans l'ajustement du modèle. Par contre, il est souvent considéré comme une boite noire et rend l'interprétation des résultats moins évidente.

Après avoir étudié la méthodologie de ce nouveau modèle, nous avons effectué une application numérique dans le chapitre 4, à savoir la mise en oeuvre d'un modèle de fréquence de sinistre d'une assurance automobile. Outre montrer la viabilité de ce modèle dans un cadre réel, des comparaisons quantitatives avec les autres modèles ont pu être établies. Il en a découlé que le AGLM surpasse effectivement le GLM en termes de qualité de prédiction. Les déviances du GAM et du AGLM sont relativement proches. En réalité, nous retombons sur les conclusions obtenues de façon qualitative. Ces deux modèles sont semblables en termes d'interprétation et de qualité de prédiction. Notre application numérique n'a pas été capable de mettre en avant une supériorité prédictive de l'Accurate GLM par rapport au Gradient Boosting. La différence entre les analyses des impacts des variables explicatives dans chacun de ces modèles a pu être soulignée numériquement et graphiquement. Par ailleurs, notons que le choix des paramètres de régularisation à inclure dans le modèle AGLM a été une étape majeur de l'ajustement. Une discussion a permis de comprendre qu'il était nécessaire de trouver le bon compromis entre qualité de prédiction et régularité des résultats.

Pour conclure, l'Accurate GLM est un modèle avec du potentiel. Contrairement aux Gradient boosting et aux réseaux neurones, souvent considérés comme des boîtes noires, il offre un bon compromis entre interprétabilité et prédiction. Dans le secteur de l'assurance, la transparence des modèles est primordial, notamment pour répondre aux exigences réglementaires. Cependant, ce besoin de clarté ne doit pas prendre le dessus sur la précision des prédictions. L'Accurate GLM combine ces deux caractéristiques.

Il serait intéressant de calibrer les modèles sur une nouvelle base de données. Ainsi nous pourrions observer si l'Accurate GLM maintient une qualité de prédiction similaire aux GAM sur d'autres types de données. Cela peut être fait sur une nouvelle base de données d'assurance automobile contenant de nouvelles variables explicatives. À priori, ce modèle pourrait également être appliquer dans d'autres domaines comme en assurance vie.

Dans le futur, il serait instructif de développer les interactions dans la méthode des Accurate GLM. Cela pourrait permettre au modèle d'améliorer sa qualité de prédiction et de se rapprocher de celle des GBM. Continuer à développer des méthodes de modélisation hybride, en combinant des modèles déjà répandus dans le domaine actuariel avec de nouvelles techniques innovantes de science des données, ne peut être que bénéfique.

Bibliographie

[1] Denuit, M., Hainaut, D. & Trufin, J. (2019). *Effective Statistical Learning Methods for Actuaries I.* 1st ed. Springer International Publishing.

[2] Destrero, A., Mosci, S., De Mol, C., Verri, A., & Odone, F. (2009)."Feature selection for high-dimensional data". *Computational Management Science*.

[3] Fujita, S., Tanaka, T., Kondo, K., & Iwasawa, H. (2020). "AGLM : A hybrid modeling method of GLM and data science techniques".

[4] Gertheiss, J. & Tutz, G. (2010). "Sparse modeling of categorical explanatory variables". *The Annals of Applied Statistics*, 4(4), pp. 2150-2180.

[5] Goepp, V., Bouaziz, O., & Nuel, G. (2018). “Spline Regression with Automatic Knot Selection”.

[6] Hastie, T., & Tibshirani, R. (1986). "Generalized Additive Models". *Statistical Science*, 1(3), pp. 297-310.

[7] Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity : The Lasso and Generalizations*. CRC Press.

[8] Hoerl, Arthur E., & Kennard, Robert W. (1970). "Ridge Regression : Biased Estimation for Nonorthogonal Problems." *Technometrics*, 12(1), pp. 55-67.

[9] Kenji Kondo. "Aglm : Accurate Generalized Linear Model." (Juin 2021). URL : https ://cran.r-project.org/web/packages/aglm/index.html (visité le 15/05/2024).

[10] Eilers, P. & Marx, B. (2021). “Why P-splines ?”. URL : https ://psplines.bitbucket.io/ (visité le 20/07/2024).

[11] Seck, N. A. & Denuit, M. (2022). "Adaptive splines for continuous features in risk assessment". CAS E-Forum Summer (September).

[12] Tibshirani, R. (1996). "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society : Series B*, 58(1), pp. 267-288.

[13] Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., & Knight, K. (2005). “Sparsity and Smoothness via the Fused Lasso.” *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 67(1), pp. 91–108.

[14] Zou, H., & Hastie, T. (2005). "Regularization and Variable Selection via the Elastic Net." *Journal of the Royal Statistical Society : Series B*, 67(2), pp. 301-320.

Annexe

Analyse descriptive de la base de données

Analysons de plus près les variables catégorielles. Les graphiques de gauche permettent de déduire la catégorie de référence utilisé dans certains modèles.

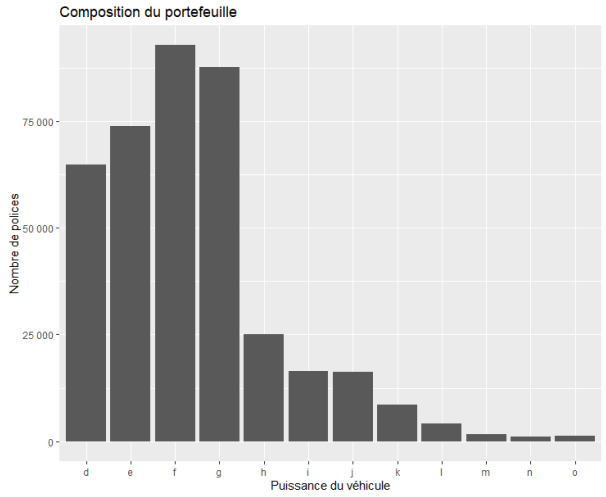

Figure 4.35. Nombre de polices par catégorie, variable `Power`

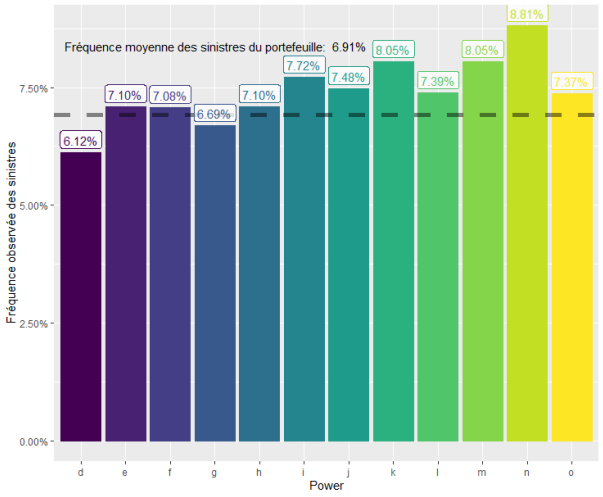

Figure 4.36. Fréquence observée moyenne par catégorie, variable `Power`

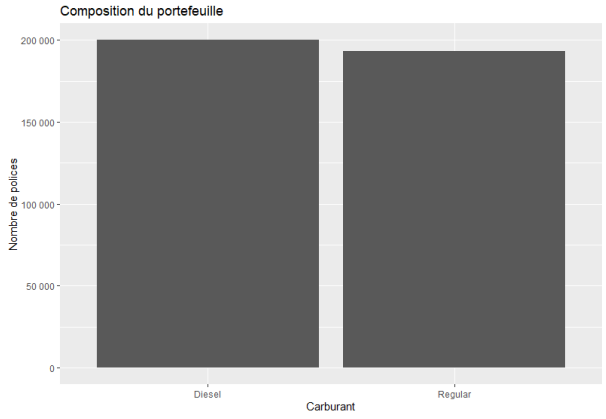

Figure 4.37. Nombre de polices par catégorie, variable `Gas`

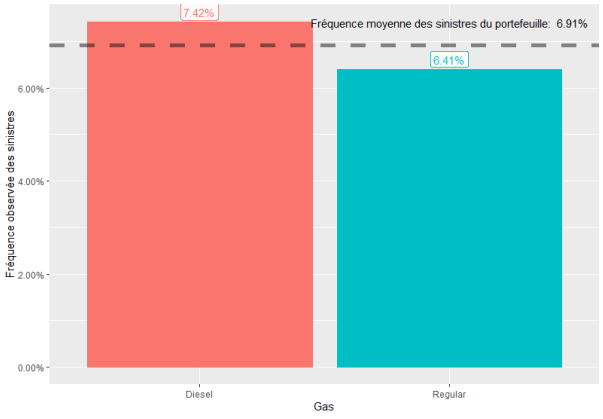

Figure 4.38. Fréquence observée moyenne par catégorie, variable `Gas`

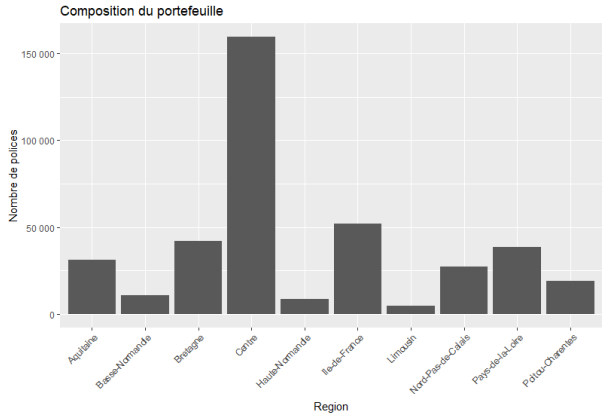

Figure 4.39. Nombre de polices par catégorie, variable `Region`

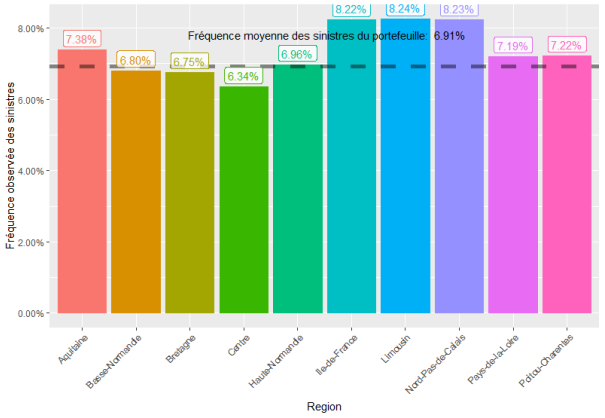

Figure 4.40. Fréquence observée moyenne par catégorie, variable `Region`

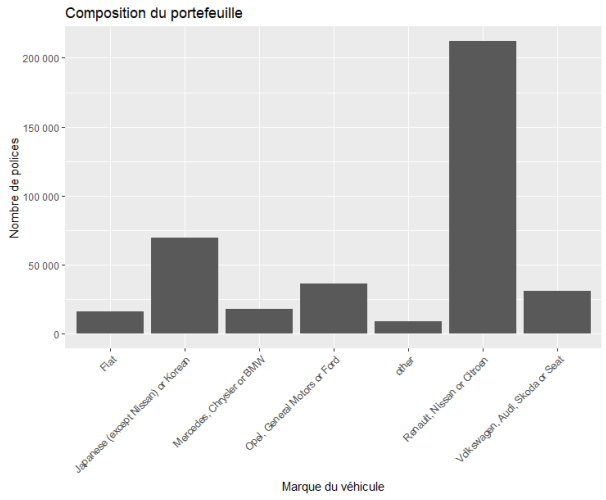

Figure 4.41. Nombre de polices par catégorie, variable `Brand`

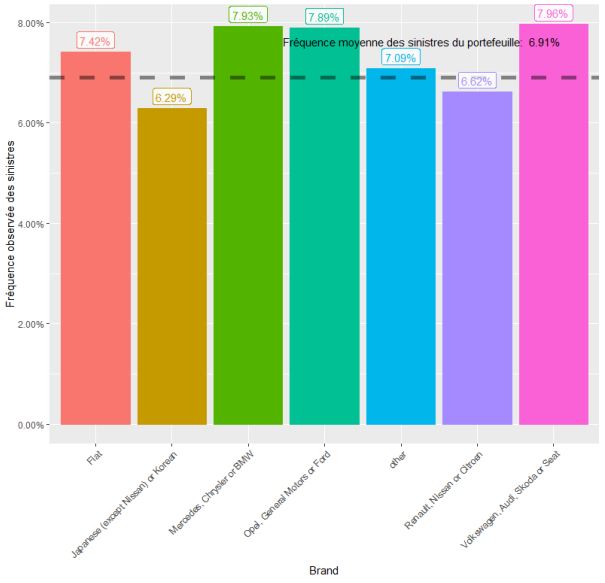

Figure 4.42. Fréquence observée moyenne par catégorie, variable `Brand`

Représentons graphiquement l'évolution des fréquences observées dans notre base de données par rapport à chacune des variables continues sur les figures 4.43, 4.44 et 4.45.

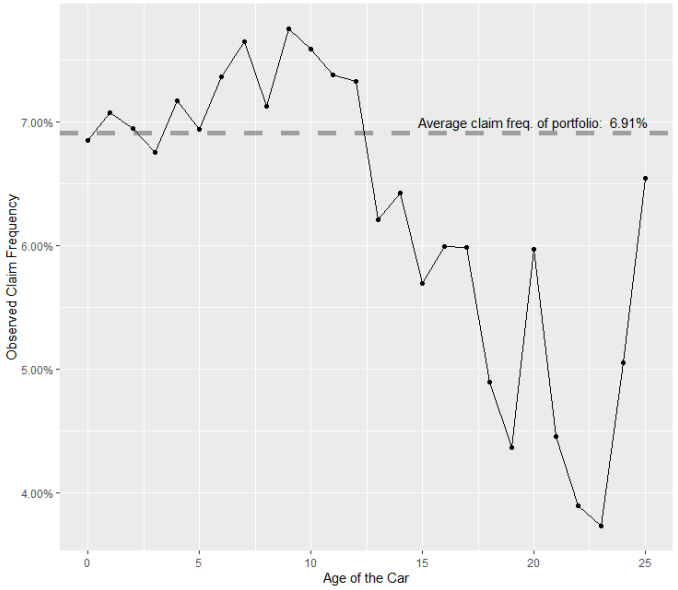

Figure 4.43. Fréquence observée pour l'âge de la voiture

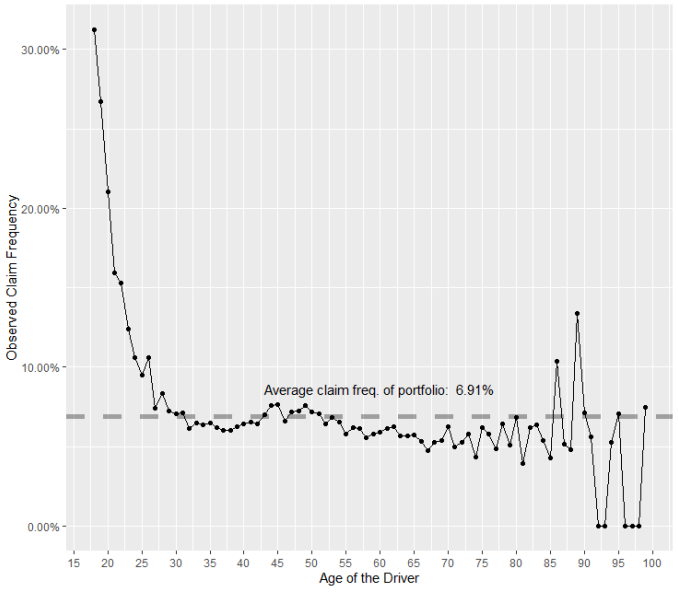

Figure 4.44. Fréquence observée pour l'âge du conducteur

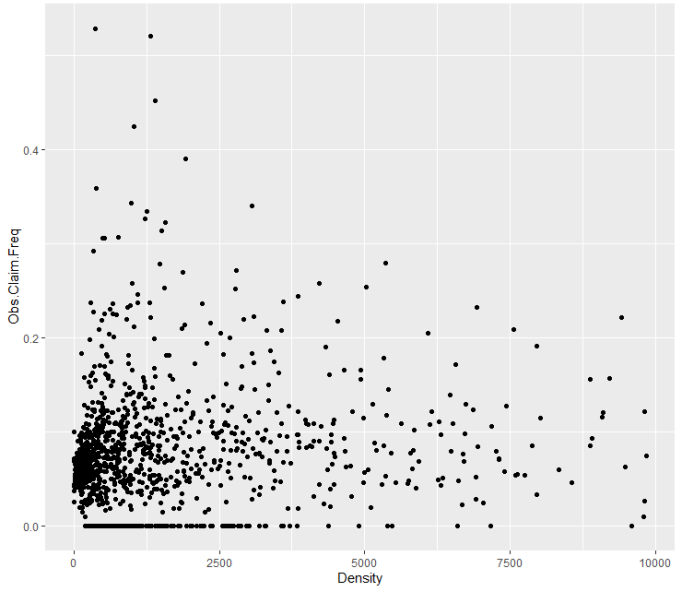

Figure 4.45. Fréquence observée de la densité de population

Préparation des données - Code R

```
# Chargement des packages :
library(CASdatasets)
library(aglm)
library(MASS)
library(mgcv)
library(gbm3)
library(xts)
library(gridExtra)
library(caret)
library(reshape2)
library(ggplot2)
library(visreg)
library(stringr)
library(scales)
library(stringi)
library(lme4)
library(glmnet)
library(jtools)
library(forcats)
library(plyr)
library(mgcViz)
library(dplyr)
library(parallel)
library(cleaner)
```

```
# Chargement du jeu de donnees :
data("freMTPLfreq")
dataset_0 = freMTPLfreq
# Filtre :
dataset_0 <- freMTPLfreq %>% filter(Density <= 10000 & CarAge <=
   25)
# Retrait du police ID
dataset = dataset_0[,-1]
# Variable puissance = variable categorielle ordonnee
dataset$Power=clean_factor(dataset$Power, levels = c("d","e","f",
   "g","h","i","j","k","l","m","n","o"), ordered = TRUE)
str(dataset)
```

```
#Separation en training set (80%) et testing set (20%) :
set.seed(21)
in_training = createDataPartition(dataset$ClaimNb, times = 1, p =
    0.8, list=FALSE)
training_set = dataset[in_training,]
testing_set  = dataset[-in_training,]

#Definition du data frame pour le AGLM
x <- training_set[3:ncol(dataset)]
y <- training_set$ClaimNb
newx <- testing_set[3:ncol(dataset)]
observed_values <- testing_set$ClaimNb
```

Ajustement des AGLM - Code R et résultats

```
# Fonction pour la deviance de Poisson
deviance_poisson <- function(y, y_pred) {
y_nonzero <- ifelse(y == 0, 1e-10, y)
deviance <- (1/length(y)) * sum(y * log(y / y_pred)-(y - y_pred))
return(deviance)
}
```

```
# Utilisation de la fonction cva.aglm (couteux en temps de calcul
   )
set.seed(21)
aglm_cva <- cva.aglm(x,y, family="poisson", offset=log(training_
   set$Exposure), alpha=seq(0,1,len=11)^3, type.measure="deviance
   ", nfolds = 5)
model_cva@alpha.min
model_cva@lambda.min
```

```
predicted_values <- predict(aglm_cva,newx = newx, s=aglm_
   cva@lambda.min, type="response",newoffset=log(testing_set$
   Exposure))
deviance_poisson(observed_values, predicted_values)
```

```
# Utilisation de la fonction cv.aglm
set.seed(21)
cv_lasso <- cv.aglm(x,y,use_LVar=TRUE, family="poisson", offset=
   log(training_set$Exposure), alpha=1, type.measure="deviance",
   nfolds=5)
cv_lasso@lambda.min
predicted_values <- predict(cv_lasso,newx = newx, s=cv_
   lasso@lambda.min , type="response",newoffset=log(testing_set$
   Exposure))
deviance_poisson(observed_values, predicted_values)
# Meme procedure avec alpha=0 et alpha=1
```

```
# Validation croisee manuelle (couteux en temsp de calcul) :
alpha_vect <- seq(0, 1, by = 0.1) #grille de alpha
lambda_vect <- seq(0, 1, length = 11)^2 #grille de lambda

#Separation du jeu de donnees en 5 plis
set.seed(21)
folds <- createFolds(training_set$ClaimNb, k = 5)
results <- matrix(NA, nrow = length(alpha_vect), ncol = length(
   lambda_vect))

for (i in 1:length(alpha_vect)) {
  for (j in 1:length(lambda_vect)) {
    fold_deviance <- sapply(folds, function(x) {
      x_train <- training_set[-x,][,3:length(dataset)]
      y_train <- training_set[-x,][,1]
      x_test <- training_set[x,][,3:length(dataset)]
      y_test <- training_set[x,]$ClaimNb

      mod <- aglm(x_train, y_train, use_LVar = TRUE, family = "
         poisson", offset = log(training_set[-x,]$Exposure),
         lambda = lambda_vect[j], alpha = alpha_vect[i])

      pred <- predict(mod, newx = x_test, s= lambda_vect[j], type
          = "response",newoffset=log(training_set[x,]$Exposure))

      deviance <- (1/length(y_test))*(sum(dpois(x = y_test,
         lambda = y_test, log = TRUE)) -
        sum(dpois(x = y_test, lambda = pred, log = TRUE)))
      return(deviance)
```

```
    })
    results[i, j] <- mean(fold_deviance)
  }
}
resultats

# Recherhe plus affinee : meme processus sur base de la grille
   suivante
alpha_vect2 <- c(0,0.001,0.005)
lambda_vect2 <- c(0,0.0001,0.002)
```

```
# Modele retenu :
set.seed(21)
aglm2 <- aglm(x,y, use_LVar = TRUE, family="poisson", offset=log(
   training_set$Exposure), alpha = 0.01, lambda =  0.001)
predicted_values <- predict(aglm2,newx = newx, s=0.001, type="
   response",newoffset=log(testing_set$Exposure))
deviance_poisson(observed_values, predicted_values)
```

Ci dessous, nous affichons les estimations des paramètres β de l'ajustement du modèle `aglm2`.

```
> coef(essai8,index=1)
$coef.OD
 Power_OD_1  Power_OD_2  Power_OD_3  Power_OD_4  Power_OD_5  Power_OD_6  Power_OD_7  Power_OD_8  Power_OD_9 Power_OD_10
 0.00000000  0.04178217  0.01481210  0.00000000  0.03878565  0.05162823  0.02594240  0.01718297  0.00000000  0.04627874

$coef.UD
   Power_UD_1    Power_UD_2    Power_UD_3    Power_UD_4    Power_UD_5    Power_UD_6    Power_UD_7    Power_UD_8    Power_UD_9
-0.041981608  0.000000000  0.006611635 -0.017143506  0.000000000  0.055245211  0.000628373  0.089420425 -0.053018301
 Power_UD_10  Power_UD_11  Power_UD_12
-0.048313389  0.184696571  0.046138332
```

Figure 4.46. Résultat du AGLM, coefficients de la variable `Power`

```
> coef(essai8,index=2)
$coef.linear
[1] -0.001521965

$coef.LV
   CarAge_LV_1    CarAge_LV_2    CarAge_LV_3    CarAge_LV_4    CarAge_LV_5    CarAge_LV_6    CarAge_LV_7    CarAge_LV_8    CarAge_LV_9
-0.0019822314 -0.0012330064  0.0000000000  0.0000000000  0.0042675756  0.0000000000 -0.0013230277  0.0042958403 -0.0023402040
 CarAge_LV_10   CarAge_LV_11   CarAge_LV_12   CarAge_LV_13   CarAge_LV_14   CarAge_LV_15   CarAge_LV_16   CarAge_LV_17   CarAge_LV_18
-0.0065434123 -0.0121205926 -0.0136863379  0.0008302932  0.0030952273  0.0060252343  0.0009417884  0.0013179704  0.0041062520
 CarAge_LV_19   CarAge_LV_20
 0.0044056050  0.0030362428
```

Figure 4.47. Résultat du AGLM, coefficients de la variable `CarAge`

```
> coef(essai8,index=3)
$coef.linear
[1] -0.01389014

$coef.LV
 DriverAge_LV_1  DriverAge_LV_2  DriverAge_LV_3  DriverAge_LV_4  DriverAge_LV_5  DriverAge_LV_6  DriverAge_LV_7  DriverAge_LV_8
  -1.175036e-02   -5.651376e-03   -1.846730e-03    1.961810e-04    2.851658e-03    4.518238e-03    6.238387e-03    6.186392e-03
 DriverAge_LV_9 DriverAge_LV_10 DriverAge_LV_11 DriverAge_LV_12 DriverAge_LV_13 DriverAge_LV_14 DriverAge_LV_15 DriverAge_LV_16
   6.181232e-03    5.544105e-03    4.872652e-03    4.653876e-03    4.164848e-03    3.871764e-03    3.772499e-03    3.832051e-03
DriverAge_LV_17 DriverAge_LV_18 DriverAge_LV_19 DriverAge_LV_20 DriverAge_LV_21 DriverAge_LV_22 DriverAge_LV_23 DriverAge_LV_24
   3.758797e-03    3.253676e-03    2.629199e-03    1.972908e-03    3.300720e-04   -1.153491e-05   -2.536853e-03   -3.823721e-03
DriverAge_LV_25 DriverAge_LV_26 DriverAge_LV_27 DriverAge_LV_28 DriverAge_LV_29 DriverAge_LV_30 DriverAge_LV_31 DriverAge_LV_32
  -3.385798e-03   -1.063903e-03   -1.526329e-03   -3.050077e-03   -3.916814e-03   -2.299897e-03   -1.334105e-03    0.000000e+00
DriverAge_LV_33 DriverAge_LV_34 DriverAge_LV_35 DriverAge_LV_36 DriverAge_LV_37 DriverAge_LV_38 DriverAge_LV_39 DriverAge_LV_40
   2.670963e-04    5.364796e-04    1.279366e-03    1.063498e-03    1.190835e-03    1.537695e-03    8.226434e-04    0.000000e+00
DriverAge_LV_41 DriverAge_LV_42 DriverAge_LV_43 DriverAge_LV_44 DriverAge_LV_45 DriverAge_LV_46 DriverAge_LV_47 DriverAge_LV_48
  -1.273127e-04   -3.516988e-04   -1.306086e-04   -3.661845e-05    0.000000e+00    2.263037e-04    5.147270e-04    5.931709e-04
DriverAge_LV_49 DriverAge_LV_50 DriverAge_LV_51 DriverAge_LV_52 DriverAge_LV_53 DriverAge_LV_54
   1.416240e-03    2.327806e-03    3.470383e-03    4.007304e-03    5.501088e-03    8.241651e-03
```

Figure 4.48. Résultat du AGLM, coefficients de la variable `DriverAge`

```
> coef(essai8,index=4)
$coef.UD
   Brand_UD_1    Brand_UD_2    Brand_UD_3    Brand_UD_4    Brand_UD_5    Brand_UD_6    Brand_UD_7
 0.027818150 -0.222078986  0.072578795  0.106610469  0.007894706  0.000000000  0.092436535
```

Figure 4.49. Résultat du AGLM, coefficients de la variable `Brand`

```
> coef(essai8,index=5)
$coef.UD
   Gas_UD_1    Gas_UD_2
 0.08673181 -0.08698447
```

Figure 4.50. Résultat du AGLM, coefficients de la variable `Gas`

```
> coef(essai8,index=6)
$coef.UD
 Region_UD_1  Region_UD_2  Region_UD_3  Region_UD_4  Region_UD_5  Region_UD_6  Region_UD_7  Region_UD_8  Region_UD_9
 0.060030290 -0.045508878 -0.025804056 -0.037971351 -0.010426580  0.045170146  0.310682145  0.027645827 -0.006284715
Region_UD_10
 0.106932703
```

Figure 4.51. Résultat du AGLM, coefficients de la variable `Region`

```
> coef(essai8,index=7)
$coef.linear
[1] 1.263747e-05

$coef.LV
 Density_LV_1  Density_LV_2  Density_LV_3  Density_LV_4  Density_LV_5  Density_LV_6  Density_LV_7  Density_LV_8  Density_LV_9
 1.035147e-05  1.013284e-05  2.167096e-05  2.473638e-05  2.624686e-05  2.639472e-05  2.592273e-05  2.425903e-05  2.039484e-05
Density_LV_10 Density_LV_11 Density_LV_12 Density_LV_13 Density_LV_14 Density_LV_15 Density_LV_16 Density_LV_17 Density_LV_18
 1.536968e-05  9.538913e-06  4.653894e-06  1.525873e-06  3.420397e-08  0.000000e+00  3.064238e-07  3.697751e-07  2.680714e-07
Density_LV_19 Density_LV_20 Density_LV_21 Density_LV_22 Density_LV_23 Density_LV_24 Density_LV_25 Density_LV_26 Density_LV_27
 0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
Density_LV_28 Density_LV_29 Density_LV_30 Density_LV_31 Density_LV_32 Density_LV_33 Density_LV_34 Density_LV_35 Density_LV_36
 0.000000e+00 -7.838440e-07 -1.935823e-06 -1.978824e-06 -1.682894e-06 -2.004393e-06 -1.473609e-06 -1.026577e-06 -1.202794e-06
Density_LV_37 Density_LV_38 Density_LV_39 Density_LV_40 Density_LV_41 Density_LV_42 Density_LV_43 Density_LV_44 Density_LV_45
-1.545718e-06 -9.507141e-07 -5.158475e-07 -6.111022e-07 -4.904979e-07 -4.764039e-07 -4.475098e-07 -3.647665e-07 -5.080628e-07
Density_LV_46 Density_LV_47 Density_LV_48 Density_LV_49 Density_LV_50 Density_LV_51 Density_LV_52 Density_LV_53 Density_LV_54
-8.191843e-07 -4.729601e-07 -1.597390e-07 -2.941697e-08 -9.740430e-08 -5.073524e-08  0.000000e+00  0.000000e+00 -5.519372e-07
Density_LV_55 Density_LV_56 Density_LV_57 Density_LV_58 Density_LV_59 Density_LV_60 Density_LV_61 Density_LV_62 Density_LV_63
-8.266291e-07 -2.626001e-06 -3.346056e-06 -5.662523e-06 -7.837903e-06 -8.364619e-06 -6.967377e-06 -6.317504e-06 -4.552395e-06
Density_LV_64 Density_LV_65 Density_LV_66 Density_LV_67 Density_LV_68 Density_LV_69 Density_LV_70 Density_LV_71 Density_LV_72
-4.208691e-06 -5.741884e-06 -8.101510e-06 -1.661942e-05 -2.385594e-05 -2.772631e-05 -3.303947e-05 -2.629579e-05 -1.596650e-05
Density_LV_73 Density_LV_74 Density_LV_75 Density_LV_76 Density_LV_77 Density_LV_78 Density_LV_79 Density_LV_80 Density_LV_81
-5.255211e-06 -3.264957e-06  0.000000e+00  0.000000e+00  8.363426e-06  3.132702e-05  1.805739e-05  1.709991e-05 -4.314952e-06
Density_LV_82 Density_LV_83 Density_LV_84 Density_LV_85 Density_LV_86 Density_LV_87 Density_LV_88 Density_LV_89 Density_LV_90
-2.152147e-05 -1.910370e-05 -2.971686e-05 -1.098789e-05 -4.948056e-06 -1.836983e-06 -4.080879e-06 -5.138278e-06 -3.749216e-06
Density_LV_91 Density_LV_92 Density_LV_93 Density_LV_94 Density_LV_95 Density_LV_96
 3.400716e-06  8.915909e-06  1.779482e-05  4.492185e-06 -4.947896e-08 -2.017254e-06
```

Figure 4.52. Résultat du AGLM, coefficients de la variable `Density`

Ajustement des GLM - Code R et résultats

```
# GLM avec les variables initiales
set.seed(21)
glm <-step(glm(ClaimNb~ offset(log(Exposure))+Gas+Brand+Region+
   Power+DriverAge+CarAge+Density,
          data=training_set, family=poisson(link="log")))
summary(glm)
```

```
> summary(glm)

Call:
glm(formula = ClaimNb ~ offset(log(Exposure)) + Gas + Brand + 
    Region + Power + DriverAge + CarAge + Density, family = poisson(link = "log"), 
    data = training_set)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-0.6820  -0.3416  -0.2657  -0.1494   6.5224  

Coefficients:
                                      Estimate Std. Error z value Pr(>|z|)    
(Intercept)                         -1.966e+00  6.949e-02 -28.293  < 2e-16 ***
GasRegular                          -1.274e-01  1.980e-02  -6.433 1.25e-10 ***
BrandJapanese (except Nissan) or Korean -3.155e-01  5.339e-02  -5.908 3.46e-09 ***
BrandMercedes, Chrysler or BMW       3.494e-02  6.154e-02   0.568  0.57018    
BrandOpel, General Motors or Ford    7.676e-02  5.180e-02   1.482  0.13835    
Brandother                          -5.109e-02  7.252e-02  -0.705  0.48110    
BrandRenault, Nissan or Citroen     -3.779e-02  4.549e-02  -0.831  0.40619    
BrandVolkswagen, Audi, Skoda or Seat 6.538e-02  5.297e-02   1.234  0.21708    
RegionBasse-Normandie               -9.177e-02  6.330e-02  -1.450  0.14713    
RegionBretagne                      -8.483e-02  4.353e-02  -1.949  0.05133 .  
RegionCentre                        -1.231e-01  3.800e-02  -3.240  0.00119 ** 
RegionHaute-Normandie               -1.025e-01  8.345e-02  -1.228  0.21931    
RegionIle-de-France                 -6.389e-02  4.632e-02  -1.379  0.16776    
RegionLimousin                       1.900e-01  8.742e-02   2.173  0.02978 *  
RegionNord-Pas-de-Calais             7.143e-03  5.053e-02   0.141  0.88759    
RegionPays-de-la-Loire              -6.524e-02  4.488e-02  -1.454  0.14604    
RegionPoitou-Charentes               4.500e-02  5.228e-02   0.861  0.38935    
Power.L                              2.855e-01  1.117e-01   2.555  0.01061 *  
Power.Q                             -1.334e-02  9.807e-02  -0.136  0.89176    
Power.C                              2.209e-02  9.788e-02   0.226  0.82148    
Power^4                              2.357e-02  9.844e-02   0.239  0.81079    
Power^5                             -5.635e-04  9.318e-02  -0.006  0.99517    
Power^6                             -1.649e-01  8.988e-02  -1.835  0.06657 .  
Power^7                             -1.245e-01  8.910e-02  -1.397  0.16249    
Power^8                             -4.580e-02  8.456e-02  -0.542  0.58804    
Power^9                             -4.132e-02  7.438e-02  -0.556  0.57851    
Power^10                             3.837e-02  6.106e-02   0.628  0.52971    
Power^11                             6.652e-02  4.937e-02   1.347  0.17790    
DriverAge                           -9.787e-03  6.573e-04 -14.890  < 2e-16 ***
CarAge                              -1.038e-02  1.905e-03  -5.450 5.03e-08 ***
Density                              8.211e-05  6.197e-06  13.249  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 80081  on 314708  degrees of freedom
Residual deviance: 79284  on 314678  degrees of freedom
AIC: 103018

Number of Fisher Scoring iterations: 6
```

Figure 4.53. Résultat du GLM

```
# Discretisation des variables continues
```

```
#Age du conducteur :
seuil_driverage <- c(17,28,34,40,45,50,55,65,99)
dataset$DriverAge_CAT <- cut(dataset$DriverAge, breaks = seuil_
   driverage)
dataset$DriverAge_CAT %>% fct_count(sort=TRUE, prop=TRUE)
dataset$DriverAge_CAT = dataset$DriverAge_CAT %>% fct_infreq()

#Age du vehicule :
seuil_carage <- c(-1,5,10,15,25)
dataset$CarAge_CAT <- cut(dataset$CarAge, breaks = seuil_carage)
dataset$CarAge_CAT %>% fct_count(sort=TRUE, prop=TRUE)
dataset$CarAge_CAT = dataset$CarAge_CAT %>% fct_infreq()

#Densite :
seuil_densite <- c(1,25,50,100,200,400,1000,2000,4000,9850)
dataset$densite_CAT <- cut(dataset$Density, breaks = seuil_
   densite)
dataset$densite_CAT %>% fct_count(sort=TRUE, prop=TRUE)
dataset$densite_CAT = dataset$densite_CAT %>% fct_infreq()

#Une procedure identique est appliquee au training set et testing
    set

set.seed(21)
glm_cat <-glm(ClaimNb~ offset(log(Exposure))+Gas+Brand+Region+
   Power+DriverAge_CAT+CarAge_CAT+densite_CAT,
            data=training_set, family=poisson(link="log"))
summary(glm_cat)

#Comparaison :
pred_glm <- predict(glm,newdata = testing_set, type="response")
deviance_poisson(observed_values, pred_glm)

pred_glm_cat <- predict(glm_cat,newdata = testing_set, type="
   response")
deviance_poisson(observed_values, pred_glm_cat)
```

```
glm(formula = ClaimNb ~ offset(log(Exposure)) + Gas + Brand +
    Region + Power + DriverAge_CAT + CarAge_CAT + densite_CAT,
    family = poisson(link = "log"), data = training_set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7319  -0.3361  -0.2622  -0.1481   6.6269

Coefficients:
                                    Estimate Std. Error z value Pr(>|z|)
(Intercept)                        -2.46518    0.07093 -34.754  < 2e-16 ***
GasRegular                         -0.17307    0.01993  -8.686  < 2e-16 ***
BrandJapanese (except Nissan) or Korean -0.26120    0.05360  -4.873 1.10e-06 ***
BrandMercedes, Chrysler or BMW      0.04134    0.06164   0.671 0.502478
BrandOpel, General Motors or Ford   0.08572    0.05181   1.655 0.098022 .
Brandother                         -0.01511    0.07253  -0.208 0.834956
BrandRenault, Nissan or Citroen    -0.02217    0.04554  -0.487 0.626283
BrandVolkswagen, Audi, Skoda or Seat  0.07316    0.05301   1.380 0.167491
RegionBasse-Normandie              -0.11618    0.06341  -1.832 0.066936 .
RegionBretagne                     -0.10872    0.04403  -2.469 0.013546 *
RegionCentre                       -0.08359    0.03826  -2.185 0.028893 *
RegionHaute-Normandie              -0.08220    0.08355  -0.984 0.325197
RegionIle-de-France                -0.02635    0.04479  -0.588 0.556375
RegionLimousin                      0.28959    0.08807   3.288 0.001009 **
RegionNord-Pas-de-Calais           -0.04713    0.05091  -0.926 0.354510
RegionPays-de-la-Loire             -0.06987    0.04511  -1.549 0.121378
RegionPoitou-Charentes              0.05344    0.05260   1.016 0.309641
Power.L                             0.32643    0.11189   2.918 0.003528 **
Power.Q                            -0.02412    0.09808  -0.246 0.805744
Power.C                             0.03253    0.09788   0.332 0.739604
Power^4                             0.03906    0.09850   0.397 0.691711
Power^5                             0.01370    0.09323   0.147 0.883185
Power^6                            -0.14220    0.08990  -1.582 0.113713
Power^7                            -0.10846    0.08913  -1.217 0.223694
Power^8                            -0.04706    0.08457  -0.556 0.577872
Power^9                            -0.03713    0.07439  -0.499 0.617682
Power^10                            0.04446    0.06107   0.728 0.466570
Power^11                            0.07481    0.04939   1.515 0.129837
DriverAge_CAT(28,34]                0.10054    0.03602   2.791 0.005248 **
DriverAge_CAT(55,65]                0.02414    0.03665   0.659 0.510007
DriverAge_CAT(40,45]                0.16256    0.03604   4.511 6.45e-06 ***
DriverAge_CAT(45,50]                0.17506    0.03579   4.892 1.00e-06 ***
DriverAge_CAT(50,55]                0.08989    0.03649   2.463 0.013767 *
DriverAge_CAT(17,28]                0.62960    0.03444  18.280  < 2e-16 ***
DriverAge_CAT(65,99]               -0.01330    0.03867  -0.344 0.730808
CarAge_CAT(5,10]                    0.03582    0.02283   1.569 0.116658
CarAge_CAT(10,15]                  -0.02216    0.02522  -0.879 0.379649
CarAge_CAT(15,25]                  -0.19325    0.03810  -5.072 3.93e-07 ***
densite_CAT(50,100]                -0.19757    0.03544  -5.575 2.47e-08 ***
densite_CAT(100,200]               -0.16378    0.03545  -4.620 3.83e-06 ***
densite_CAT(200,400]               -0.14789    0.03623  -4.082 4.46e-05 ***
densite_CAT(25,50]                 -0.29187    0.03781  -7.718 1.18e-14 ***
densite_CAT(1e+03,2e+03]            0.12121    0.03499   3.464 0.000532 ***
densite_CAT(4e+03,9.85e+03]         0.20375    0.04028   5.059 4.22e-07 ***
densite_CAT(1,25]                  -0.42456    0.04270  -9.942  < 2e-16 ***
densite_CAT(2e+03,4e+03]            0.22821    0.03916   5.827 5.64e-09 ***
```

Figure 4.54. Résultat du GLM avec discrétisation

Il aurait été intéressant de regrouper les catégories non significativement différentes et regarder si le modèle en aurait été amélioré.

Ajustement du GAM - Code R et résultats

```
set.seed(21)
gam <- bam(ClaimNb ~ offset(log(Exposure)) + Power + s(DriverAge)
    + s(CarAge) + s(Density) + Brand + Gas + Region,
           data = training_set, family = poisson(link = "log"))
summary(gam_cr)
pred_gam <- predict(gam,newdata = testing_set, type="response")
deviance_poisson(observed_values, pred_gam)
```

```
#Visualisation
viz <- getViz(gam)
j=1 #2,...,7
plot(viz,select=j)
```

```
> summary(gam)

Family: poisson
Link function: log

Formula:
ClaimNb ~ offset(log(Exposure)) + Power + s(DriverAge) + s(CarAge) +
    s(Density) + Brand + Gas + Region

Parametric coefficients:
                                        Estimate Std. Error z value Pr(>|z|)
(Intercept)                            -2.400094   0.062092 -38.654  < 2e-16 ***
Power.L                                 0.351571   0.111923   3.141  0.00168 **
Power.Q                                -0.024652   0.098087  -0.251  0.80156
Power.C                                 0.036103   0.097883   0.369  0.71225
Power^4                                 0.033708   0.098490   0.342  0.73217
Power^5                                 0.002737   0.093235   0.029  0.97658
Power^6                                -0.150854   0.089919  -1.678  0.09341 .
Power^7                                -0.112467   0.089155  -1.261  0.20714
Power^8                                -0.048151   0.084583  -0.569  0.56917
Power^9                                -0.034326   0.074397  -0.461  0.64452
Power^10                                0.047654   0.061081   0.780  0.43529
Power^11                                0.071573   0.049395   1.449  0.14734
BrandJapanese (except Nissan) or Korean -0.266226   0.054188  -4.913 8.97e-07 ***
BrandMercedes, Chrysler or BMW          0.041972   0.061639   0.681  0.49591
BrandOpel, General Motors or Ford       0.077906   0.051811   1.504  0.13267
Brandother                             -0.019937   0.072526  -0.275  0.78340
BrandRenault, Nissan or Citroen        -0.029923   0.045541  -0.657  0.51114
BrandVolkswagen, Audi, Skoda or Seat    0.062961   0.053013   1.188  0.23497
GasRegular                             -0.180856   0.020017  -9.035  < 2e-16 ***
RegionBasse-Normandie                  -0.117638   0.063457  -1.854  0.06377 .
RegionBretagne                         -0.097858   0.044060  -2.221  0.02635 *
RegionCentre                           -0.104097   0.038307  -2.717  0.00658 **
RegionHaute-Normandie                  -0.080089   0.083506  -0.959  0.33751
RegionIle-de-France                    -0.015094   0.046824  -0.322  0.74718
RegionLimousin                          0.260090   0.087666   2.967  0.00301 **
RegionNord-Pas-de-Calais               -0.035062   0.051323  -0.683  0.49450
RegionPays-de-la-Loire                 -0.075467   0.045067  -1.675  0.09402 .
RegionPoitou-Charentes                  0.043059   0.052523   0.820  0.41232
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
               edf Ref.df Chi.sq p-value
s(DriverAge) 7.887  8.518 958.04  <2e-16 ***
s(CarAge)    4.336  5.340  61.92  <2e-16 ***
s(Density)   6.550  7.670 338.16  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.00843   Deviance explained = 3.11%
fREML = 5.7108e+05  Scale est. = 1         n = 314709
```

Figure 4.55. Résultat du GAM

Ajustement des GBM - Code R et résultats

```
# Modele retenu :
set.seed(89)
```

```
gbm = gbmt(ClaimNb ~ offset(log(Exposure))+DriverAge+CarAge+
   Density+Power+Brand+Gas+Region,
          data = training_set,
          distribution = gbm_dist("Poisson"),
          train_params = training_params(num_trees = 700,
                                         shrinkage = 0.02,
                                         interaction_depth = 10,
                                         min_num_obs_in_node =
                                            500,
                                         num_train = 1*nrow(
                                            training_set)),
          is_verbose = TRUE,
          cv_folds = 5,
          par_details = gbmParallel(num_threads = 3))

best_iter = gbmt_performance(gbm, "cv")

plot(gbmt_performance(gbm, method="cv"),main="Cross validation")
legend("bottomleft",legend=c("Training set","Testing set","Best
   iteration"),col=c(1,3,4),lty=c(1,1,2),lwd=2)
#on atteint un plateau

predicted_values <- predict(gbm,newdata = testing_set,n.trees=
   best_iter, type="response")*testing_set$Exposure
deviance_poisson(observed_values, predicted_values)

#Le graphique suivant nous montre l'importance des variables
summary(gbm,normalize = TRUE)

#Garphique de dependance partielle de la variable j = 1,...,7
plot(gbm, var_index = j, num_trees = best_iter, type="response",
   main="Dependance partielle de la variable DriverAge")
```

GLM régularisé - Code R et résultats

```
# Modification du data frame
x_matrix <- as.matrix(x)
x_matrix <- model.matrix(~ . - 1, data = x)
newx_matrix <- as.matrix(newx)
newx_matrix <- model.matrix(~ . - 1, data = newx)
```

```
#Validation croisee pour determiner les parametres de
   regularisation :
#Lasso :
set.seed(54)
```

```
folds = createFolds(training_set$ClaimNb, 5, list=FALSE)
set.seed(21)
glmreg_lasso <- cv.glmnet(x_matrix, y , family = "poisson",
   offset=log(training_set$Exposure), alpha = 1,
  nfolds = 5,
  foldid = folds,
  maxit=10^4,
  nlambda = 25)
predicted_values <- predict(glmreg_lasso,newx = newx_matrix,
   newoffset=log(testing_set$Exposure), type="response")
deviance_poisson(observed_values, predicted_values)
```

```
> coef(glmreg_lasso)
32 x 1 sparse Matrix of class "dgCMatrix"
                                                      s1
(Intercept)                                 -2.530704e+00
(Intercept)                                  .
DriverAge                                   -3.396153e-03
CarAge                                       .
Density                                      1.292731e-05
Power.L                                      .
Power.Q                                      .
Power.C                                      .
Power^4                                      .
Power^5                                      .
Power^6                                      .
Power^7                                      .
Power^8                                      .
Power^9                                      .
Power^10                                     .
Power^11                                     .
BrandJapanese (except Nissan) or Korean      .
BrandMercedes, Chrysler or BMW               .
BrandOpel, General Motors or Ford            .
Brandother                                   .
BrandRenault, Nissan or Citroen              .
BrandVolkswagen, Audi, Skoda or Seat         .
GasRegular                                   .
RegionBasse-Normandie                        .
RegionBretagne                               .
RegionCentre                                 .
RegionHaute-Normandie                        .
RegionIle-de-France                          .
RegionLimousin                               .
RegionNord-Pas-de-Calais                     .
RegionPays-de-la-Loire                       .
RegionPoitou-Charentes                       .
>
```

Figure 4.56. Résultat du GLM régularisé avec la pénalité Lasso, $\alpha = 1$

```
#Ridge :
set.seed(21)
glmreg_ridge <- cv.glmnet(x_matrix, y , family = "poisson",
   offset=log(training_set$Exposure), alpha = 0,
                          nfolds = 5,
                          foldid = folds,
                          maxit=10^4,
```

```
                            nlambda = 25)
predicted_values <- predict(glmreg_ridge,newx = newx_matrix,
   newoffset=log(testing_set$Exposure), type="response")
deviance_poisson(observed_values, predicted_values)
```

```
> coef(glmreg_ridge)
32 x 1 sparse Matrix of class "dgCMatrix"
                                                  s1
(Intercept)                                -2.638560e+00
Powerd                                     -7.879088e-03
Powere                                      2.418816e-03
Powerf                                      2.785401e-03
Powerg                                     -3.387335e-03
Powerh                                      9.315596e-04
Poweri                                      6.200873e-03
Powerj                                      4.706685e-03
Powerk                                      9.253135e-03
Powerl                                      2.185272e-03
Powerm                                      4.146549e-03
Powern                                      1.782509e-02
Powero                                      6.313307e-03
CarAge                                     -5.091524e-04
DriverAge                                  -6.467068e-04
BrandJapanese (except Nissan) or Korean    -5.833766e-03
BrandMercedes, Chrysler or BMW              8.564030e-03
BrandOpel, General Motors or Ford           9.055829e-03
Brandother                                  1.161721e-03
BrandRenault, Nissan or Citroen            -5.100368e-03
BrandVolkswagen, Audi, Skoda or Seat        1.129014e-02
GasRegular                                 -9.216679e-03
RegionBasse-Normandie                      -1.907831e-03
RegionBretagne                             -1.682760e-03
RegionCentre                               -8.562265e-03
RegionHaute-Normandie                      -1.272916e-04
RegionIle-de-France                         8.428592e-03
RegionLimousin                              1.077517e-02
RegionNord-Pas-de-Calais                    7.744060e-03
RegionPays-de-la-Loire                      1.565381e-03
RegionPoitou-Charentes                      5.432835e-03
Density                                     4.418079e-06
```

Figure 4.57. Résultat du GLM régularisé avec la pénalité Ridge, $\alpha = 0$

```
#Elastic Net alpha = 0.5
set.seed(21)
glmreg_en <- cv.glmnet(x_matrix, y , family = "poisson", offset=
   log(training_set$Exposure), alpha = 0.5,
                           nfolds = 5,
                           foldid = folds,
                           maxit=10^4,
                           nlambda = 25)

predicted_values <- predict(glmreg_en,newx = newx_matrix,
   newoffset=log(testing_set$Exposure), type="response")
deviance_poisson(observed_values, predicted_values)
```

```
> coef(glmreg_en)
32 x 1 sparse Matrix of class "dgCMatrix"
                                                      s1
(Intercept)                                 -2.544458e+00
Powerd                                       .
Powere                                       .
Powerf                                       .
Powerg                                       .
Powerh                                       .
Poweri                                       .
Powerj                                       .
Powerk                                       .
Powerl                                       .
Powerm                                       .
Powern                                       .
Powero                                       .
CarAge                                       .
DriverAge                                   -3.073718e-03
BrandJapanese (except Nissan) or Korean      .
BrandMercedes, Chrysler or BMW               .
BrandOpel, General Motors or Ford            .
Brandother                                   .
BrandRenault, Nissan or Citroen              .
BrandVolkswagen, Audi, Skoda or Seat         .
GasRegular                                   .
RegionBasse-Normandie                        .
RegionBretagne                               .
RegionCentre                                 .
RegionHaute-Normandie                        .
RegionIle-de-France                          .
RegionLimousin                               .
RegionNord-Pas-de-Calais                     .
RegionPays-de-la-Loire                       .
RegionPoitou-Charentes                       .
Density                                      1.160862e-05
```

Figure 4.58. Résultat du GLM régularisé avec la pénalité Elastic Net, $\alpha = 0.5$

Comparaison des modèles

```
#Analyse de l'effet de la variable DriverAge :

#On definit un sous groupe particulier avec l'age qui varie
ens_fix =data.frame()
for (age in seq(18,99)) {
  tmp<-data.frame(DriverAge=age,Power="f",CarAge=3,
                  Brand="Renault, Nissan or Citroen",Region="
                     Centre",Density=200, Gas="Diesel"
                  ,Exposure=1)
  ens_fix<-rbind(ens_fix,tmp)
}

#On definit un individu avec l'age de reference
ens_fixR = data.frame(DriverAge=35,Power="f",CarAge=3,
                  Brand="Renault, Nissan or Citroen",Region="
                     Centre",Density=200, Gas="Diesel"
                  ,Exposure=1)
```

```
ens_fix$Power <- factor(ens_fix$Power, levels = levels(testing_
   set$Power), ordered = TRUE)
ens_fix$Brand <- factor(ens_fix$Brand, levels = levels(testing_
   set$Brand))
ens_fix$Region <- factor(ens_fix$Region, levels = levels(testing_
   set$Region))
ens_fix$Gas <- factor(ens_fix$Gas, levels = levels(testing_set$
   Gas))

ens_fixR$Power <- factor(ens_fixR$Power, levels = levels(testing_
   set$Power), ordered = TRUE)
ens_fixR$Brand <- factor(ens_fixR$Brand, levels = levels(testing_
   set$Brand))
ens_fixR$Region <- factor(ens_fixR$Region, levels = levels(
   testing_set$Region))
ens_fixR$Gas <- factor(ens_fixR$Gas, levels = levels(testing_set$
   Gas))

#On predit la reponse pour chaque modele sur les deux ensembles :

pred_glm<-exp(predict(glm,ens_fix))
pred_glmR<-exp(predict(glm,ens_fixR))

pred_gam<-exp(predict(gam,ens_fix))
pred_gamR<-exp(predict(gam,ens_fixR))

newx_ens <- model.matrix(~ . - 1, data = ens_fix[,-8])
pred_glmnet <- predict(glmreg_lasso,newx = newx_ens, s=glmreg_
   lasso$lambda.min, newoffset=log(ens_fix$Exposure), type="
   response")
newx_ensR <- model.matrix(~ . - 1, data = ens_fixR[,-8])
pred_glmnetR <- predict(glmreg_lasso,newx = newx_ensR, s=glmreg_
   lasso$lambda.min, newoffset=log(ens_fixR$Exposure), type="
   response")
#idem pour glmreg_ridge

ens_fix_new <- ens_fix[,-8]
ens_fix_new <- ens_fix_new %>%
  select(Power, CarAge, DriverAge, Brand, Gas, Region, Density)
pred_aglm <- predict(essai8,newx = ens_fix_new, s=6.46651e-05 ,
   type="response",newoffset=log(ens_fix$Exposure))
ens_fix_newR <- ens_fixR[,-8]
ens_fix_newR <- ens_fix_newR %>%
  select(Power, CarAge, DriverAge, Brand, Gas, Region, Density)
pred_aglmR <- predict(essai8,newx = ens_fix_newR, s=6.46651e-05 ,
    type="response",newoffset=log(ens_fixR$Exposure))
```

```
#GBM : pas correct mais utilise pour avoir une idee approximative
pred_gbm<-predict(gbm,newdata=ens_fix,n.trees=best_iter,type="
   response")*ens_fix$Exposure
pred_gbmR<-predict(gbm,newdata=ens_fixR,n.trees=best_iter,type="
   response")*ens_fixR$Exposure

#Graphique
matplot(seq(18,99), cbind(pred_glm/pred_glmR,pred_gam/pred_gamR
   [1],pred_gbm/pred_gbmR,pred_glmnet/pred_glmnetR[1],pred_aglm/
   pred_aglmR[1]),type="l",col=c("red","blue","green","yellow","
   black"),
        lty=c(1,1,2,1,1),lwd=c(2,2,1,2,2),xlab="DriverAge",ylab="
           Prediction normalisee",main="Frequence de sinistre
           predite")
legend("topright",c("GLM", "GAM","GBM","GLMNET","AGLM"),lty=c
   (1,1,2,1,1), col=c("red","blue","green","yellow","black"))

#Meme procedure pour les autres variables continues
```

```
#Analyse de l'effet de la variable Gas :

#Un profil est defini pour chaque categorie
profil1_gas = data.frame(DriverAge=35,Power="f",CarAge=3,
                         Brand="Renault, Nissan or Citroen",
                            Region="Centre",Density=200, Gas="
                            Regular"
                         ,Exposure=1)

#Profil de reference
profil2_gas = data.frame(DriverAge=35,Power="f",CarAge=3,
                         Brand="Renault, Nissan or Citroen",
                            Region="Centre",Density=200, Gas="
                            Diesel"
                         ,Exposure=1)

profil1_gas$Power <- factor(profil1_gas$Power, levels = levels(
   testing_set$Power), ordered = TRUE)
profil1_gas$Brand <- factor(profil1_gas$Brand, levels = levels(
   testing_set$Brand))
profil1_gas$Region <- factor(profil1_gas$Region, levels = levels(
   testing_set$Region))
profil1_gas$Gas <- factor(profil1_gas$Gas, levels = levels(
   testing_set$Gas))

profil2_gas$Power <- factor(profil2_gas$Power, levels = levels(
```

```
   testing_set$Power), ordered = TRUE)
profil2_gas$Brand <- factor(profil2_gas$Brand, levels = levels(
   testing_set$Brand))
profil2_gas$Region <- factor(profil2_gas$Region, levels = levels(
   testing_set$Region))
profil2_gas$Gas <- factor(profil2_gas$Gas, levels = levels(
   testing_set$Gas))

#On predit la reponse pour chaque modele sur chaque profil :

pred_glm4<-exp(predict(glm,profil1_gas))
pred_glm4R<-exp(predict(glm,profil2_gas))
pred_glm4/pred_glm4R

pred_gam4<-exp(predict(gam,profil1_gas))
pred_gam4R<-exp(predict(gam,profil2_gas))
pred_gam4/pred_gam4R

newx_ens4 <- model.matrix(~ . - 1, data = profil1_gas[,-8])
pred_glmnet4 <- predict(glmreg_lasso,newx = newx_ens4,s=glmreg_
   lasso$lambda.min, newoffset=log(profil1_gas$Exposure), type="
   response")
newx_ens4R <- model.matrix(~ . - 1, data = profil2_gas[,-8])
pred_glmnet4R <- predict(glmreg_lasso,newx = newx_ens4R,s=glmreg_
   lasso$lambda.min, newoffset=log(profil2_gas$Exposure), type="
   response")
pred_glmnet4/pred_glmnet4R

ens_fix_new4 <- profil1_gas[,-8]
ens_fix_new4 <- ens_fix_new4 %>%
  select(Power, CarAge, DriverAge, Brand, Gas, Region, Density)
pred_aglm4 <- predict(essai8,newx = ens_fix_new4, s=6.46651e-05 ,
    type="response",newoffset=log(profil1_gas$Exposure))
ens_fix_new4R <- profil2_gas[,-8]
ens_fix_new4R <- ens_fix_new4R %>%
  select(Power, CarAge, DriverAge, Brand, Gas, Region, Density)
pred_aglm4R <- predict(essai8,newx = ens_fix_new4R, s=6.46651e-05
    , type="response",newoffset=log(profil2_gas$Exposure))
pred_aglm4/pred_aglm4R

#GBM : pas correct mais idee approximative
pred_gbm4<-predict(gbm,newdata=profil1_gas,n.trees=best_iter,type
   ="response")*profil1_gas$Exposure
pred_gbm4R<-predict(gbm,newdata=profil2_gas,n.trees=best_iter,
   type="response")*profil2_gas$Exposure
pred_gbm4/pred_gbm4R
```

Modèle pour les sévérités

```
#Chargement du jeu de donnees
data("freMTPLsev")
dataset_sev = freMTPLsev

#Aggregation par police
aggregated <- dataset_sev %>%
  group_by(PolicyID) %>%
  summarise(TotalClaimCost = sum(ClaimAmount))

#Merge avec le dataset contenant les variables explicatives
combined_data <- merge(dataset_0, aggregated, by = "PolicyID",
   all.y = TRUE)
#Calcul du cout moyen par police
combined_data$MeanClaimCost = combined_data$TotalClaimCost/
   combined_data$ClaimNb

#On applique le meme filtre que celui utilise dans le modele de
   frequence
combined_data <- combined_data %>% filter(Density <= 10000 &
   CarAge <= 25)

#On exclu les valeurs extremes
sev_data <- combined_data %>%
  filter(MeanClaimCost <= 15000)

#Transformation logarithmique de la reponse
sev_data$MeanClaimCost <- log(sev_data$MeanClaimCost)
```

```
#Deviance pour la loi normale
deviance_gaussian <- function(y, y_pred) {
  residuals <- (y - y_pred)^2
  deviance <- (1/length(y))*sum(residuals)
  return(deviance)
}
```

```
sev_data$Power=clean_factor(sev_data$Power, levels = c("d","e","f
   ","g","h","i","j","k","l","m","n","o"), ordered = TRUE)
str(sev_data)

#Separation du jeu de donnees en deux groupes, le training set et
    le testing set
set.seed(21)
in_training_sev = createDataPartition(sev_data$MeanClaimCost,
   times = 1, p = 0.8, list=FALSE)
```

```
training_set_sev = sev_data[in_training_sev,]
testing_set_sev  = sev_data[-in_training_sev,]

#Creation du dataframe utilise dans l'ajustement du modele
x_sev <- training_set_sev[5:ncol(sev_data)-1]
y_sev <- training_set_sev$MeanClaimCost
newx_sev <- testing_set_sev[5:ncol(sev_data)-1]
observed_values_sev <- testing_set_sev$MeanClaimCost
```

```
#Utilisation de la fonction cv.aglm
set.seed(21)
mod_sev <- cv.aglm(x_sev,y_sev, use_LVar=TRUE, family="gaussian",
    offset=log(training_set_sev$Exposure),alpha=0,nfolds=5)
pred <- predict(mod_sev,newx = newx_sev, s=mod_sev@lambda.min,
   type="response",newoffset=log(testing_set$Exposure))
deviance_gaussian(observed_values_sev, pred)
```

```
#Modele retenu
set.seed(21)
m5sev <- aglm(x_sev,y_sev, use_LVar=TRUE, family="gaussian",
   offset=log(training_set_sev$Exposure),alpha=0, lambda
   =0.08196678)
pred <- predict(m5sev,newx = newx_sev, s=m5sev@lambda.min, type="
   response",newoffset=log(testing_set$Exposure))
deviance_gaussian(observed_values_sev, pred)

#Information sur l'ajustement (i=1,...,7):
m5sev@vars_info[i]
coef(m5sev,index=i)
plot(m5sev,vars=i)
```

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Faculté des sciences
Place des Sciences, 2 bte L6.06.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/sc