UNIVERSITÉ DU QUÉBEC À MONTRÉAL

MODÉLISATION DE LA FRÉQUENCE DES SINISTRES AVEC DONNÉES TÉLÉMATIQUES PAR LE MODÈLE GBMP

MÉMOIRE
PRÉSENTÉ
COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES

PAR
NOUREDDINE MERAIHI

DÉCEMBRE 2019

REMERCIEMENTS

Je voudrais tout d'abord remercier grandement mon directeur de recherche, Jean-Philippe Boucher, pour l'entière confiance qu'il m'a accordée. «Merci pour tout le support que tu m'as donné, et surtout d'avoir cru en mes capacités dès le premier jour. Je suis ravi d'avoir travaillé en ta compagnie, car outre ton appui scientifique et financier, tu as toujours été là pour me soutenir et me conseiller.»

Je remercie mes très chers parents, Mohamed et Malika, qui ont toujours été là pour moi, «Vous avez tout sacrifié pour vos enfants. Vous m'avez donné un magnifique modèle de labeur et de persévérance. Je suis redevable d'une éducation dont je suis fier».

Enfin, je remercie ma chère épouse, «merci de ta présence à mes côtés, merci pour ta confiance dans mes choix, merci pour ton soutien inconditionnel et tes encouragements, merci de t'occuper de Sara et Amina tous ces week-ends que je travaillais. Merci pour tout.»

TABLE DES MATIÈRES

LISTE DES ALGORITHMES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES CODES

LISTE DES ABRÉVIATIONS

CARA Chaire Co-operators en analyse des risques actuariels.

CART Classification and Regression Trees.

FSA Forward Sortation Area.

GBM Gradient Boosting Machine.

GBMP Gradient Boosting Machine Poisson.

GPU Graphics Processing Unit.

IA intelligence artificielle.

RTA Région de tri d'acheminement.

SHAP SHapley Additive exPlanation.

UBI Usage Based Insurance.

VIN Vehicle identification number.

RÉSUMÉ

Les algorithmes d'apprentissage automatique sont de nos jours couramment utilisés dans plusieurs domaines comme outils de prédiction. Toutefois, certains modèles nécessitent une adaptation à la réalité de l'industrie avant de les appliquer. À l'aide de données télématiques, nous intéressons à la modélisation de la fréquence de réclamations par le modèle *Gradient Boosting Machine Poisson* (GBMP). Nous modifions sa fonction de perte afin de refléter la surdispersion dans les données d'assurance. Pour comprendre les résultats obtenus, nous examinons l'influence marginale des variables explicatives au moyen de graphiques d'importance relative et de diagrammes de dépendance partielle (PDPs). Ensuite, nous utilisons les valeurs *SHapley Additive exPlanations* (SHAP) afin de sélectionner et segmenter nos variables. Finalement, nous mesurons la performance prédictive de notre modèle à l'aide des règles de scores pour les données de comptage.

Mots-clés : actuariat, tarification en assurance automobile, modélisation de la fréquence, sélection et segmentation des variables, données télématiques, apprentissage automatique, *Gradient Boosting Machine Poisson* (GBMP), diagrammes de dépendance partielle (PDPs), *SHapley Additive exPlanations* (SHAP).

CHAPITRE I

INTRODUCTION

Le principal produit qu'un assureur transige est une prestation lors de la survenance d'un évènement incertain et aléatoire souvent appelé le « risque ». C'est pourquoi les assureurs cherchent les meilleurs outils d'évaluation de ce dernier. Les dernières années, les modèles de régression leur ont été très utiles. Pour cette raison, nous présentons dans ce chapitre quelques éléments théoriques des modèles de régression spécifiques à l'assurance non-vie. Ensuite, dans le chapitre 2, nous verrons en détail le modèle *Gradient Boosting Machine Poisson* (GBMP) et les outils théoriques d'interprétation de ce dernier. Mais avant, dans les sections 2.1 et 2.2, nous fournirons quelques éléments théoriques nécessaires concernant les arbres de régression ainsi que le *boosting*. Finalement, dans le chapitre 3, nous appliquerons notre modèle GBMP aux données télématiques d'assurance, alors que dans le chapitre 4 nous interpréterons les résultats obtenus et comparerons la prédiction obtenue par notre modèle avec des modèles de comptage.

1.1 Introduction à la tarification de l'assurance non-vie

Nous savons tous que lorsque nous assurons notre bien auprès d'un assureur, c'est que nous confions à ce dernier un besoin de protection contre certains risques auxquels nous ne pouvons faire face seuls. Nous payons ainsi une prime en échange

d'une couverture qui nous protège contre une perte partielle ou totale de notre bien. Ainsi, l'assureur s'engage à prendre le **risque** de payer la réclamation de l'assuré si un sinistre survient. On entend par sinistre tout évènement qui peut engendrer une demande de paiement en vertu des modalités d'une police d'assurance. Afin d'éviter la ruine, l'assureur doit alors mesurer ce risque. Pour ce faire, il doit mesurer l'exposition au risque. Cette **exposition** peut-être mesurée de plusieurs façons, par exemple, historiquement, dans le cas de l'assurance automobile, c'est la durée durant laquelle les assurés sont couverts. Durant cette période de couverture propre à chaque assuré, l'assureur peut faire face à un certain nombre de réclamations. Ce nombre est appelé la **fréquence** de réclamation. Chaque sinistré peut faire plus qu'une réclamation dont le montant d'argent dépend de la valeur du bien assuré. Le coût moyen de ces montants est appelé la **sévérité**. La fréquence et la sévérité des sinistres sont d'une grande importance dans la tarification de l'assurance automobile, car elles sont généralement utilisées comme variables à prédire dans les modèles de régression. Ces deux variables sont les éléments de base pour estimer le montant du sinistre moyen auquel devra faire face l'assureur, ce montant est est appelé **prime pure**. En vue de prédire cette prime pour les différents profils d'assurés, les actuaires doivent se fier à des techniques de régression comme le modèle linéaire généralisé (ou *Generalized Linear Model*, GLM).

Pour chaque assuré i, le nombre de sinistres N_i est observé, ainsi que la période totale d'exposition d_i pendant laquelle ces sinistres ont été observés. Par conséquent, la fréquence des sinistres Y_i par unité d'exposition peut être exprimée comme le rapport entre les deux. Alors que la sévérité X_i est le rapport entre le coût total encouru des sinistres sur le nombre de sinistres.

Dans le présent document, nous nous concentrons uniquement sur la modélisation de la fréquence des sinistres, en fonction des caractéristiques a priori des

assurés. Nous tenterons d'estimer la réalisation de $\hat{Y}_i$ par une technique d'apprentissage machine appelée *Gradient Boosting Machine Poisson* (GBMP). Nous interpréterons les résultats obtenus par trois techniques d'interprétation des modèles d'intelligence artificielle (IA). Finalement, nous sélectionnerons et segmenterons les variables les plus importantes dans notre base de données grâce à ces techniques. Finalement, nous estimerons à nouveau la réalisation de $\hat{Y}_i$ avec quatre modèles de comptage classiques après sélection et segmentation des variables, et comparerons les résultats obtenus.

Mais d'abord, définissons ce que c'est l'apprentissage machine et décrivons ses différents types.

1.2 Les modèles linéaires généralisés

Comme mentionné auparavant, l'objectif de la tarification de l'assurance non-vie est de déterminer la prime que chaque assuré doit payer en fonction du profil de risque de l'assuré. Pour ce faire, nous devons analyser comment cette prime varie en fonction des différents facteurs de tarification disponibles. Développé par (McCullagh et Nelder, 1989), le modèle GLM est le modèle couramment utilisé pour la tarification en assurance non-vie. Dans un modèle GLM, on suppose que la variable réponse Y suit une distribution de probabilité au sein de la famille exponentielle. Les distributions particulières appartenant à cette famille comprennent, entre autres, les distributions Normales, de Poisson et Gamma. En général, la fonction de densité de probabilité pour la variable de réponse Y peut être exprimée par :

$$f_Y(y) = \exp\Big[\frac{y\theta - b(\theta)}{\phi/w} + c(y, \phi/w)\Big], \tag{1.1}$$

où $b(\cdot)$ et $c(\cdot)$ sont des fonctions connues qui correspondent à une distribution

spécifique, θ est le paramètre naturel ou canonique et ϕ est le paramètre de dispersion. Les poids w de chaque observation sont des constantes connues. De plus, en utilisant des GLMs au lieu des modèles linéaires, nous ne supposons plus une relation strictement linéaire entre la valeur prédite de la variable réponse Y et les variables explicatives X. Cette relation est maintenant spécifiée par une fonction de lien g qui relie la moyenne de Y avec un prédicateur linéaire η.

Soit $Y = (Y_1, \ldots, Y_n)$ un vecteur de variables aléatoires indépendantes avec une distribution appartenant à la famille exponentielle et $g(\cdot)$ une fonction de lien différentiable et monotone. On pose alors la relation :

$$g(\mu_i) = \mathbf{x}_i'\boldsymbol{\beta} = \eta_i, \quad \text{pour } i = 1, \ldots, n,$$

avec $\mathbf{x}_i$ un vecteur connu de variables explicatives correspondant à la i^e observation et $\boldsymbol{\beta}$ un vecteur de paramètres inconnus. La moyenne μ_i de la variable réponse y_i peut être exprimée par

$$\mu_i = \mathbb{E}(Y_i) = g^{-1}(\eta_i).$$

Dans la tarification de l'assurance non-vie, la relation entre les résultats et les variables explicatives est souvent multiplicative plutôt qu'additive (Denuit *et al.*, 2007). Pour cette raison, une fonction de liaison logarithmique est généralement utilisée pour ce type d'analyse.

1.3 La fréquence des sinistres par la régression

Comme dans ce document nous nous concentrons sur la modélisation du nombre de sinistres, présentons le modèle du comptage des sinistres de Poisson dans un cadre GLM. Toutefois, des dérivations similaires peuvent être faites pour la modélisation de la sévérité des sinistres individuels à l'aide, par exemple, d'une distribution

gamma ou inverse-gaussienne.

Le GLM le plus couramment utilisé pour modéliser la fréquence des sinistres est un modèle de régression de Poisson loglinéaire de la forme :

$$N_i \sim \text{POI}(d_i u_i)$$

avec d_i l'exposition correspondant à l'assuré i et μ_i le nombre annuel moyen de sinistres. Pour une fonction de lien logarithmique g, le nombre annuel moyen de sinistres est donné par

$$\mu_i = \mathbb{E}(N_i) = g^{-1}(\eta_i) = \exp(\eta_i).$$

Nous supposons que le nombre de sinistres est une v.a. dont la distribution est une Poisson avec une fonction de distribution de probabilité donnée par

$$Pr(N_i = n_i) = \frac{\exp(-d_i\mu_i)(d_i\mu_i)^{n_i}}{n_i\,!}, \quad n_i = 0, 1, \ldots$$

Puisque le nombre de sinistres que nous cherchons à modéliser prend des valeurs entières non négatives, la distribution Poisson devient alors un choix naturel pour ce type de modélisation. Cependant, dans une base de données typique d'un assureur, seulement environ 5% des assurés font des réclamations, ce qui rend le vecteur de variable réponse surdispersé. Pour cela, on peut utilises ce qu'on appelle des « modèles mélange » qui sont une solution à cette surdispersion, par exemple les modèles Poisson gonflé à zéro, les régressions binomiale-négative de type I et II ainsi que d'autres modèles (voir (Boucher *et al.*, 2007)).

1.4 L'apprentissage machine

Nous avons vu que la modélisation statistique conçoit une relation entre les variables à l'aide d'une équation mathématique. Cela diffère de la modélisation en apprentissage machine. En effet, ce type de modélisation est un type d'algorithme qui peut apprendre à partir des données, sans structure explicite entre les variables. Cela permet au modélisateur d'inférer de nombreuses relations complexes et non linéaires. Mais soyons très clairs, lorsque nous disons « l'algorithme peut apprendre à partir des données », en aucun cas nous prétendons que nous concevons des modèles « intelligents » au sens de l'intelligence humaine. À vrai dire, le terme « apprendre », c'est seulement une reconnaissance d'une situation similaire, donc reconnaître que la dernière fois que nous étions dans une telle situation (même sorte de données), nous avons essayé une action particulière (produit ce résultat) et cela a fonctionné (était correct ou très semblable), donc nous allons essayer à nouveau. Là où ça n'a pas marché (résultats obtenus très différents ou taux d'erreur inacceptable), nous allons essayer autre chose.

1.5 Types d'apprentissage machine

Dans son livre, (Marsland, 2015, p. 6) a défini l'apprentissage comme le fait de « s'améliorer » dans l'accomplissement d'une tâche grâce à la pratique. Ça consiste donc à faire en sorte que les machines (ordinateurs), en suivant des recettes algorithmiques, modifient ou plutôt adaptent leurs actions pour qu'elles deviennent plus précises afin de donner des résultats justes et acceptables.

Suite à cette définition, deux questions peuvent être soulevées : comment l'ordinateur sait-il s'il s'améliore ou non, et comment sait-il comment s'améliorer ? L'auteur propose plusieurs réponses possibles à ces questions afin de cerner les

différents types d'apprentissage machine. Il est possible d'indiquer à notre algorithme la bonne réponse à un problème pour qu'il l'obtienne la prochaine fois. Nous espérons que nous n'aurons à lui donner que quelques bonnes réponses et qu'il pourra ensuite trouver comment obtenir les bonnes réponses pour d'autres problèmes similaires (généraliser). Alternativement, nous pouvons lui indiquer si la réponse était correcte ou non, mais pas comment trouver la bonne réponse, de sorte qu'il doit chercher comment trouver cette bonne réponse. Une variante de ceci est que nous donnions un score pour la réponse, en fonction de sa justesse, plutôt qu'une réponse « bonne ou mauvaise » (0, ou 1). Enfin, nous n'avons peut-être pas de bonnes réponses ; nous voulons simplement que l'algorithme trouve des entrées qui ont quelque chose en commun.

Apprentissage supervisé. Pour ce type d'apprentissage, nous donnons à notre algorithme un ensemble de données d'entraînement [1] où chaque observation est constituée de variables explicatives $\mathbf{x}_i$ où $i = 1, \ldots, n$ ainsi que la valeur attendue de la variable réponse y_i. Nous souhaitons ajuster un modèle qui relie la variable réponse y_i aux variables explicatives $\mathbf{x}_i$, dans le but de prédire avec précision une valeur associée à la réponse pour des observations futures.

Apprentissage non-supervisé. En revanche, en apprentissage non supervisé, pour chaque observation $i = 1, \ldots, n$ notre vecteur de mesures $\mathbf{x}_i$ n'est plus étiqueté (ou annoté) par les valeurs de la variable réponse y_i. Il n'est pas possible d'utiliser un modèle de régression linéaire, puisque nous n'avons plus la variable réponse à prédire. L'algorithme tente plutôt d'identifier les similitudes entre les entrées afin que si elles ont quelque chose en commun, elles soient alors catégorisées ensemble.

1. Une partie des données utilisées pour entraîner le modèle (généralement 70%) des données. Ce concept est plus détaillé à la section 2.2

Apprentissage par renforcement. C'est quelque part entre l'apprentissage supervisé et l'apprentissage non supervisé. L'algorithme est informé lorsque la réponse est fausse, mais il ne sait pas comment la corriger. Il doit explorer et essayer différentes possibilités jusqu'à ce qu'il parvienne à trouver la bonne réponse.

Le type d'apprentissage le plus commun est l'apprentissage supervisé, qui fera l'objet du prochain chapitre. Donc, avant de commencer, nous allons introduire quelques algorithmes qui s'y appliquent pour finalement présenter en détail le modèle GBM qui est le sujet principal de ce document.

CHAPITRE II

INTRODUCTION AU *GRADIENT BOOSTING MACHINE POISSON*

Dans ce chapitre, nous détaillons notre modèle GBMP. Mais pour y arriver, nous devons présenter toutes les pierres angulaires qui ont permis de construire cet algorithme. D'abord, nous introduisons la régression basée sur les arbres de décision qui est la fondation du *Gradient Boosting Machine* (GBM) et nous verrons les avantages et les lacunes de cette méthode. Ensuite, nous définissons quelques termes essentiels en apprentissage machine, afin d'introduire l'amélioration apportée aux lacunes de la régression basée sur les arbres de régressions qui a donné naissance à d'autres modèles qu'on appelle modèles adaptatifs. Par la suite, le modèle GBM sera présenté en détail. Nous adapterons ce dernier afin d'atteindre notre principal objectif, qu'est l'estimation de la fréquence de sinistre, en tenant compte de la distribution Poisson. Finalement, nous verrons plusieurs méthodes qui nous permettront d'interpréter les résultats obtenus par notre modèle GBMP.

2.1 Les arbres de régression

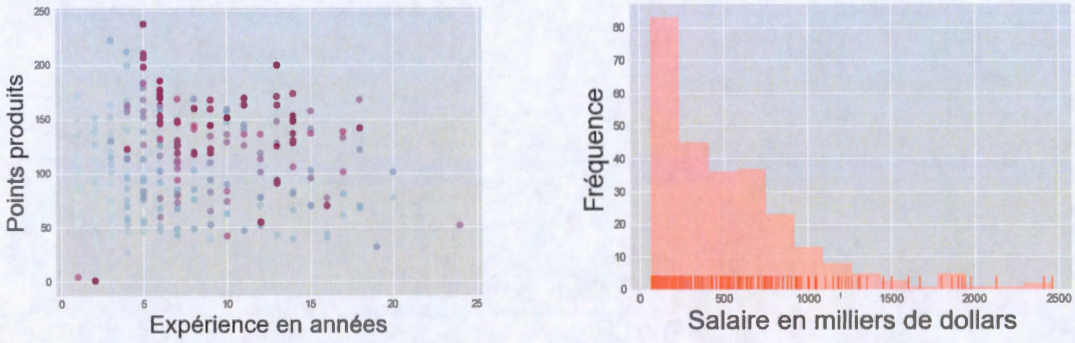
Les arbres de régression sont des méthodes de régression basées sur les arbres de décision. Ils ont été introduits par (Breiman *et al.*, 1984). Il s'agit de stratifier ou de segmenter l'espace des prédicteurs en un certain nombre de régions simples. Afin de faire une prédiction pour une observation donnée, nous utilisons généralement la

moyenne des observations dans la région à laquelle elle appartient. Cette méthode nous permet de prédire des valeurs ou une classe lorsqu'il s'agit d'un problème de classification.

2.1.1 Exemple introductif

Pour mieux comprendre l'idée derrière les arbres de régression, reprenons un exemple simple qui a été illustré dans le livre (James *et al.*, 2013, p. 304). Dans cet exemple, les auteurs tentent de prédire le salaire d'un joueur de baseball ($\mathbf{Y}$ variable réponse) en se basant sur deux variables explicatives, soient le nombre de points produits $\mathbf{X_1}$ et l'expérience au sein de la ligue de baseball $\mathbf{X_2}$ exprimée en nombre d'années jouées.

Afin de visualiser les données de ce problème, nous allons utiliser un graphique en deux dimensions dans un système de coordonnées cartésiennes, où l'axe des ordonnées représente la première variable explicative $\mathbf{X_1}$ (le nombre de points produits par un joueur donné) et l'axe d'abscisses représente le nombre d'années jouées $\mathbf{X_2}$. Dans la figure (2.1a), les salaires sont présentés par des points ayant comme coordonnées les deux variables explicatives. Nous pouvons voir la variable réponse comme une troisième dimension reliant les points produits par un joueur donné et son expérience. Ces points seront plus foncés pour les salaires les plus élevés et plus clairs pour les salaires les plus faibles.



(a) Salaire des joueurs où les points les plus foncés dont les salaires les plus élevés.

(b) Distribution de la fréquence des salaires des joueurs.

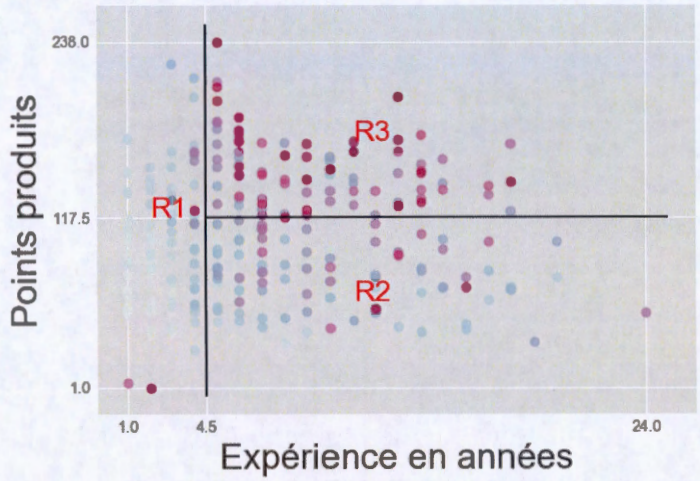
Figure 2.1: Exemple tiré du livre (James *et al.*, 2013, p. 304) où l'on tente de prédire le salaire d'un joueur de baseball en se basant sur les années d'expérience et le nombre de points produits.

Nous pouvons constater sur la figure (2.1a), que les joueurs les mieux payés sont ceux ayant le plus d'expérience et ayant produit le plus de points, donc les points situés sur le quart supérieur droit, à l'exception des deux points situés au coin inférieur gauche. Ces derniers peuvent être vus comme de jeunes joueurs considérés comme de futures vedettes.

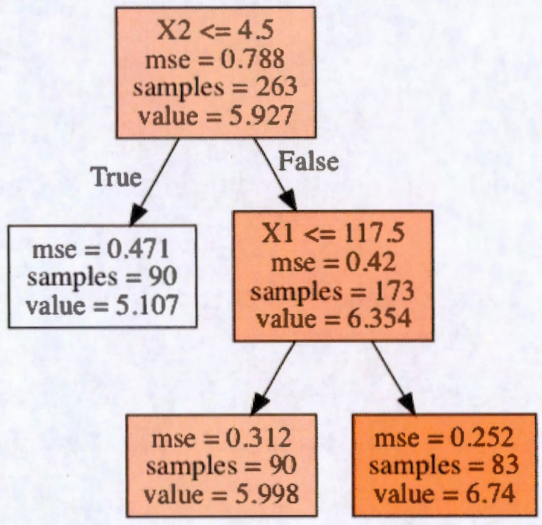
Sur cette même figure, nous séparons les points en trois groupes, de sorte à créer trois groupes de joueurs. Nous pouvons dès lors tracer une ligne verticale entre le point $\mathbf{X_2} = 4$ et $\mathbf{X_2} = 5$. Ainsi, nous avons deux groupes ; des joueurs ayant plus que 4 années d'expérience et les autres ayant moins que 5 années d'expérience.

Nous pouvons ensuite clairement séparer le premier groupe en deux sous-groupes ; des joueurs qui ont produit plus que 117 points ($\mathbf{X_1} > 117$) et qui ont plus d'années d'expérience ($\mathbf{X_1} > 4.5$), et le deuxième groupe qui est composé des joueurs ayant un nombre de points produits inférieurs à 118 points ($\mathbf{X_1} < 118$) et qui ont eux

aussi plus que 4.5 années d'expérience ($\mathbf{X_2} > 4.5$). La segmentation que nous venons de faire est illustrée à la figure (2.2a)



(a) Segmentation des salaires des joueurs en trois régions.



(b) Prédiction par l'arbre de décision.

Figure 2.2: Exemple de la régression par arbre de décision.

La séparation des points sur la figure (2.1a) que nous venons d'effectuer n'est rien d'autre qu'un arbre de décision qui est illustré à la figure (2.2b). Nous pou-

vons remarquer sur cette même figure que nous avons trois niveaux (feuilles). Les joueurs ayant moins que 4.5 années d'expérience et ceux qui ont en plus. Ensuite, ce dernier groupe se divise en deux sous-groupes ; ceux qui ont produit moins que 118 buts et les joueurs ayant produit plus que 118.

Une fois la division des groupes effectuée, il suffit de calculer la moyenne des salaires à l'intérieur de chaque groupe pour obtenir une prédiction pour de nouvelles observations. Dans le présent exemple, notre modèle prédirait les salaires suivants (en milliers de dollars) :

1. $R_1 = (\mathbf{Y}|\mathbf{X_2} \leq 4,5) = \exp(5{,}1068) = 165\ 114\ \$$
2. $R_2 = (\mathbf{Y}|\mathbf{X_2} > 4,5\ , \mathbf{X_1} \leq 117{,}5) = 402\ 783\ \$$
3. $R_3 = (\mathbf{Y}|\mathbf{X_2} > 4,5\ , \mathbf{X_1} > 117{,}5) = 845\ 307\ \$.$

Nous pouvons voir alors R_1, R_2 et R_3 comme des régions de prédiction telle qu'illustrée dans la figure (2.2a).

2.1.2 Méthodologie

Afin de construire un arbre de régression, deux étapes s'imposent :

1. Nous divisons l'espace de prédiction (l'ensemble de valeurs possibles $\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_p}$ en J régions qui ne se chevauchent pas $R_1, R_2, \ldots, R_J$.
2. Pour chaque observation dans la région R_j, la prédiction est la même car c'est simplement la moyenne des valeurs de la variable réponse R_j.

Les régions $R_1, R_2, \ldots, R_J$ peuvent avoir n'importe quelle forme géométrique. Toutefois, il a été choisi de diviser ces régions sous forme rectangulaire de deux dimensions dans l'exemple présenté précédemment. Dans un problème de classification,

le but est de trouver les régions $R_1, R_2, \ldots, R_I$ qui minimisent la somme des carrés des résidus donné par :

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} \left(\mathbf{y_i} - \hat{\mathbf{y}}_{R_j}\right)^2, \tag{2.1}$$

où $\hat{\mathbf{y}}_{R_j}$ est la moyenne des valeurs de la variable réponse des données dans la j^e région.

2.1.3 Avantages et inconvénients

Nous venons de voir que les arbres de régression sont très intuitifs. Les graphiques illustrant les arbres de décision tels que présentés à la figure (2.2b) rendent cette méthode très facile à comprendre lorsqu'il s'agit de les présenter aux personnes n'ayant pas un grand bagage mathématique. Cette méthode est basée sur une séquence récursive de règles de division [1]. Les prédictions sont acceptables lorsque les données sont homogènes. Cette méthode est très sensible aux données, surtout en présence de valeurs aberrantes. Cependant, en agrégeant [2] de nombreux arbres de décision **ensemble**, en utilisant des méthodes comme *bagging* ou *random forests*, la performance de prédiction des arbres peut être considérablement améliorée. Le principe de ces méthodes basées sur les arbres de décision est de créer plusieurs arbres de décision et faire ensuite du ré-échantillonnage de ces arbres. De cette manière, en moyenne, les prédictions deviendront de moins en moins sensibles aux données (pensons aux données aberrantes) et donneront de meilleurs résultats de prédiction.

1. Une traduction libre du terme *splits*
2. Une traduction libre du terme anglais *aggregating*

2.2 Introduction aux modèles adaptatifs

Avant de plonger dans les détails des modèles adaptatifs, définissons quelques concepts clés en apprentissage machine.

Le surapprentissage

Nous avons vu dans la section (1.4) lorsque nous avons introduit l'apprentissage machine est ses différents types, que l'apprentissage est le fait de s'améliorer dans l'exécution d'une tâche grâce à la pratique, autrement dit grâce à l'entraînement du modèle sur les données. Nous n'avons qu'à donner à notre modèle beaucoup d'exemples (données d'entraînement [3]) afin de mémoriser les actions à prendre. Toutefois, notre modèle risque de s'adapter si bien aux données qu'il mémorise les détails précis de ces données spécifiques, plutôt que de trouver des caractéristiques générales de ces données qui s'appliqueront également aux données futures sur lesquelles nous voulons calculer des prédictions. Cette mémorisation est appelée le surapprentissage.

Le principal défi de l'apprentissage machine est l'obtention de bons résultats (avec précision) sur de nouvelles observations jamais vues auparavant, et pas seulement sur ceux sur lesquels notre modèle a été entraîné. La capacité de bien performer sur des entrées précédemment non observées s'appelle le pouvoir de généralisation.

Validation croisée en K-parties

Afin d'éviter le surapprentissage, nous ne devons pas entraîner notre modèle sur toutes les données que nous possédons, nous devons cacher certaines données

3. Une traduction libre de l'expression anglaise *train data set*

afin qu'il ne puisse pas mémoriser tous les détails. Pour ce faire, nous divisons nos données en deux parties; d'abord un ensemble de données sur lequel nous entraînons notre modèle (données d'entraînement), et le second, sur lequel nous testons la performance de notre modèle (données de validation[4]). Généralement cette proportion se situe entre 50% et 80%. (Ian Goodfellow et Courville, 2016) ont montré qu'il n'y a pas vraiment de méthode pour déterminer la meilleure proportion, mais ce ratio dépend seulement du problème à résoudre et des données disponibles.

La validation croisée en K-parties utilise une partie des données disponibles pour ajuster le modèle et une partie différente pour le tester. Nous divisons les données en K parties plus ou moins égales; par exemple, lorsque $K = 5$, le scénario ressemble à ce qui illustre à la figure (2.3) :

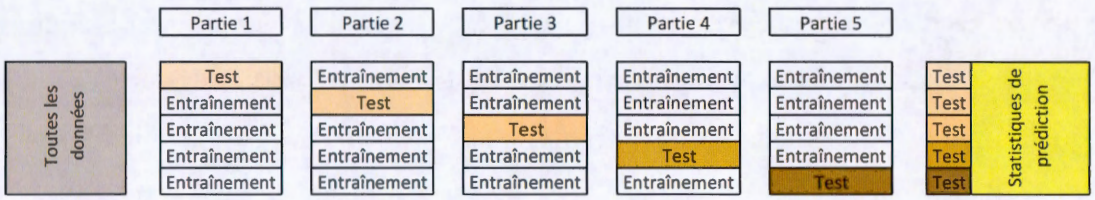


Figure 2.3: Validation croisée 5 parties.

Pour la k^e partie, nous ajustons le modèle aux $K - 1$ autres parties des données et calculons l'erreur de prédiction du modèle ajusté lorsque nous prédisons la k^e partie des données. Nous faisons cela pour $k = 1, 2, \ldots, K$ et combinons les estimations K de l'erreur de prédiction.

Soit $k : \{1, \ldots, N\} \to \{1, \ldots, K\}$, une fonction d'indexation qui indique la partition à laquelle l'observation i est allouée aléatoirement. Notons $\hat{f}^{-k}(x)$ la fonction

4. Une traduction libre de l'expression anglaise *test data set*

ajustée, calculée avec la k^e partie des données supprimée. Ensuite, l'estimation de validation croisée de l'erreur de prédiction est

$$CV(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i)). \tag{2.2}$$

Les choix typiques de K sont de 5 à 10. Le cas $K = N$ est connu sous le nom de validation croisée de type *leave-one-out*. Dans ce cas, $k(i) = i$, et pour la i^e observation, l'ajustement est calculé en utilisant toutes les données sauf la i^e.

Soit un ensemble de modèles $f(x, \alpha)$ indexés par un paramètre de calibration α, notons $\hat{f}^{-k}(x, \alpha)$ le α^e modèle ajusté avec la k^e partie des données supprimées. Ainsi, pour cet ensemble de données, nous définissons,

$$CV(\hat{f}, \alpha) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i, \alpha)), \tag{2.3}$$

où L est une fonction de perte.

La fonction $CV(\hat{f}, \alpha)$ fournit une estimation de la courbe d'erreur de test, et nous trouvons le paramètre de calibration α qui la minimise. Notre dernier modèle choisi est $f(x, \hat{\alpha})$, que nous ajustons ensuite à toutes les données (Friedman *et al.*, 2001, p. 241-242).

La méthode la plus simple et la plus largement utilisée pour estimer l'erreur de prédiction est la validation croisée. Cette méthode estime directement l'erreur extra-échantillon (*expected extra-sample error*) $Err = \mathbb{E}[L(Y, f(X))]$, l'erreur de généralisation moyenne lorsque la méthode $\hat{f}(X)$ est appliquée à un échantillon de test indépendant de la distribution conjointe de X et Y comme mentionné précédemment, nous pouvons espérer que la validation croisée évalue l'erreur conditionnelle, avec l'ensemble d'apprentissages T maintenu fixe.

Le biais de l'estimateur d'un paramètre est la différence entre la valeur de l'espérance de cet estimateur (qui est une variable aléatoire) et la valeur qu'il est censé estimer (définie et fixe). Si $\hat{\theta}$ est l'estimateur des θ, alors $\text{Biais}(\hat{\theta}) \equiv \mathbb{E}[\hat{\theta}] - \theta$. Un estimateur $\hat{\theta}$ est dit non biaisé si $\text{Biais}(\hat{\theta}) = 0$, ce qui implique que $\mathbb{E}(\hat{\theta}) = \theta$

Supposons un algorithme qui génère des modèles un grand nombre de fois, si l'on fait une moyenne sur les valeurs prédites par ces modèles et qu'on obtient une valeur de la moyenne prédite qui est asymptotiquement différente de la valeur moyenne réelle, on dit alors que notre algorithme est biaisé.

Cet algorithme génère alors des classificateurs trop simples qui ne prédisent pas parfaitement la partie de données d'entraînement. Ceci n'est pas forcément aberrant, car ce classificateur se doit d'une part de généraliser pour anticiper l'autre partie de données de validation, et d'autre part de ne pas être induit en erreur par le bruit dans les données. Pour faire le parallèle avec notre exemple de joueurs de baseball, un jeune joueur vedette qui a signé un contrat très payant avec une équipe de baseball, son grand salaire peut-être considéré comme valeur aberrante.

Lorsqu'un algorithme utilise des modèles non linéaires à des fins de prédiction, par exemple les arbres de décision, l'algorithme devient très sensible aux données, car il crée de très petits découpages d'espace, ou de plusieurs arbres en profondeur dans le cas des arbres de décision. Dans un tel cas, la prédiction devient ainsi parfaitement ajustée sur la partie des données d'entraînement et créerait potentiellement un surajustement.

2.2.1 AdaBoost

Le *boosting* est l'une des idées d'apprentissage machine les puissantes présentées dans les vingt dernières années (Friedman *et al.*, 2001, p.337-339). Il a été conçu à

l'origine pour des problèmes de classification, mais comme nous le verrons dans ce chapitre, il peut aussi être utilisé pour la régression. Le principe est très simple, il s'agit de construire une famille de modèles qui sont ensuite agrégés par une moyenne pondérée des estimations (régression) ou par un vote (classification). La façon de construire les modèles est récurrente. En effet, chaque modèle est une version adaptée du précédent en donnant plus de poids, pour le modèle naissant, aux observations mal ajustées ou mal prédites.

Nous commençons par décrire l'algorithme de *boosting* le plus populaire introduit par (Freund et Schapire, 1997) appelé AdaBoost.M1 (*Adaptive Boosting*). Supposons que nous voulons prédire la classe d'une observation quelconque dont la variable réponse est codée $Y \in \{-1, 1\}$. Étant donné un vecteur de variables prédicteurs X, un classificateur $G(X)$ produit une prédiction prenant l'une des deux valeurs $\{-1, 1\}$. Le taux d'erreur sur l'échantillon est de

$$\overline{err} = \frac{1}{N} \sum_{i}^{N} \mathbb{1}_{\{y_i \neq G(x_i)\}},$$

et l'espérance du taux d'erreur sur les prédictions futures est $E_{XY}\mathbb{1}_{(Y \neq G(X))}$.

Un classificateur faible est un classificateur dont le taux d'erreur n'est que légèrement supérieur à celui d'un classificateur aléatoire. Le but du *boosting* est d'appliquer séquentiellement l'algorithme de classification faible à des versions modifiées à plusieurs reprises sur les données, produisant ainsi une séquence de classificateurs faibles $G_m(x), m = 1, 2, \ldots, M$.

Les prédictions de chacun d'entre eux sont ensuite combinées par un vote à majorité pondérée pour produire la prédiction finale :

$$G(x) = \text{signe}\Big(\sum_{m=1}^{M} \alpha_m G_m(x)\Big). \tag{2.4}$$

Ici, $\alpha_1, \alpha_2, \ldots, \alpha_M$ sont calculés par l'algorithme Boosting et pondèrent la contribution de chaque $G_m(x)$ respectif. Leur effet est de donner une plus grande influence aux classificateurs les plus précis dans la séquence.

Les modifications de données à chaque étape du Boosting consistent à appliquer des poids $w_1, w_2, \ldots, w_N$ à chacune des observations sur les données d'entraînement $(x_i, y_i), i = 1, 2, \ldots$ initialement, tous les poids sont initialisés à $w_i = 1/N$, de sorte que la première itération entraîne simplement le classificateur sur les données sans discrimination. Pour chacune des itérations $m = 2, 3, \ldots, M$ les poids d'observation sont modifiés individuellement et l'algorithme de classification est réappliqué aux observations pondérées. À l'itération m, les observations qui ont été mal classées par le classificateur $G_{m-1}(x)$ induit à l'étape précédente ont leur poids élevé, alors que les poids sont plus petits pour celles qui ont été classées correctement. Ainsi, au fur et à mesure que les itérations avancent, les observations qui sont difficiles à classer correctement reçoivent une influence de plus en plus grande. Chaque classificateur successif est ainsi contraint de se concentrer sur les observations d'entraînement qui ne sont pas prises en compte correctement par les classificateurs précédents.

Algorithme 1 : AdaBoost

1 Initialiser les poids $w_i = 1/N$, $i = 1, 2, \ldots, N$ sur les observations.;

2 **pour** $m \leftarrow 1$ **à** M **faire**

(a) Appliquer un classificateur faible $G_m(x)$ sur les données d'entraînement en utilisant les poids w_i;

(b) Calculer

$$\text{err}_m = \frac{\sum_{i=1}^{N} w_i \mathbb{1}_{\{y_i \neq G(x_i)\}}}{\sum_{i=1}^{N} w_i}$$

(c) Calculer $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$;

(d) $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot \mathbb{1}_{\{y_i \neq G(x_i)\}}]$;

3 **fin**

Résultat :

$$G(x) = \text{signe}\Big[\sum_{m=1}^{M} \alpha_m G_m(x)\Big].$$

Dans l'algorithme 1, le classificateur $G_m(x)$ est induit sur les observations pondérées à la ligne 2a. Le taux d'erreur pondéré résultant est calculé à la ligne 2b. La ligne 2c calcule le poids α_m donné à $G_m(x)$ dans la production du classificateur final $G(x)$ (ligne 3). Les poids individuels de chacune des observations sont mis à jour pour l'itération suivante à la ligne 2d. Les observations mal classées par $G_m(x)$ ont leurs poids mis à l'échelle par un facteur $\exp(\alpha_m)$ ce qui augmente leur influence relative pour induire le classificateur suivant $G_{m+1}(x)$.

2.2.2 Modèle adaptatif pas-à-pas

Lorsque nous voulons résoudre un problème de régression, nous cherchons à obtenir une valeur (quantité) de y caractérisée par un vecteur d'entrées $\mathbf{x} = x_1, \ldots, x_p$. Soit un ensemble de M instances $(y_i, \mathbf{x}_i)$; $i = 1, \ldots M$ de valeurs connues (y, x),

le but est d'utiliser ces données pour estimer la fonction f, qui sert à mettre en correspondance le vecteur d'entrée x avec les valeurs de la sortie y. L'estimation $\hat{f}$ peut alors être utilisée pour faire des prédictions sur des instances où seulement x sont observés. Formellement, nous souhaitons former la fonction de prédiction $\hat{f} : \mathbf{x} \to y$ qui minimise l'espérance d'une fonction de perte spécifique $L(y, \hat{f})$ sur la distribution conjointe de toutes les valeurs $(y, \mathbf{x})$

$$\hat{f}(\mathbf{x}) = \underset{f(\mathbf{x})}{\operatorname{argmin}} E_{y,\mathbf{x}} L(y, f(\mathbf{x})). \tag{2.5}$$

Comme dans ce document nous voulons résoudre un problème de régression où les prédictions sont quantitatives, nous essayons d'estimer $E(y|\mathbf{x}) = g^{-1} \circ \hat{f}(x)$. Où $g(\cdot)$ est une fonction de lien équivalente à celle des GLM (1.2). Dans la régression linéaire, $\hat{f}(x) = \sum_{t=1}^{T} \beta_t x_t$. De façon plus générale

$$\hat{f}(x) = \sum_{t=1}^{T} \hat{f}_t(\mathbf{x}). \tag{2.6}$$

Cependant, cette généralité extrême est difficile à obtenir. L'un des objectifs des techniques de modélisation prédictive est de se rapprocher le plus possible de cette généralité. Par conséquent, les modèles prédictifs expriment $\hat{f}(x)$ comme suit

$$\hat{f}(\mathbf{x}) = \sum_{t=1}^{T} \hat{f}_t(\mathbf{x}) = \sum_{t=1}^{T} \beta_t h(\mathbf{x}\,;\mathbf{a}_t), \tag{2.7}$$

où les fonctions $h(\mathbf{x}\,;\mathbf{a}_t)$ sont généralement considérées comme des fonctions simples caractérisées par un ensemble de paramètres $\mathbf{a} = \{a_1, a_2, \dots\}$ et un multiplicateur β_t.

Dans le contexte du Boosting, $\beta_t h(\mathbf{x}\,;\mathbf{a}_t)$ représente le classificateur faible et $f(x)$ le vote majoritaire pondéré des classifieurs faibles individuels. L'estimation des paramètres dans l'équation (2.7) consiste à résoudre à chaque étape

$$\min_{\{\beta_t, \mathbf{a}_t\}_t^T} \sum_{i=1}^{M} L\Big(\mathbf{y}_i, \sum_{t=1}^{T} \beta_t h(\mathbf{x}_i ; \mathbf{a}_t)\Big), \tag{2.8}$$

où $L(y, f(x)$ est une fonction de perte qui définit la faiblesse de l'ajustement (Lee *et al.*, 2015). La méthode par étapes vers l'avant (*forward stepwise method*) consiste à résoudre l'équation 2.8 en ajustant séquentiellement un seul *weak learner* et en l'ajoutant à l'expansion de termes préalablement ajustés tels que décrits dans l'algorithme (2).

Algorithme 2 : Modèle adaptatif pas-à-pas

1 On initialise $f_0(\mathbf{x}) = 0$;

2 **pour** $t \leftarrow 1$ **à** T **faire**

(a) Estimer β_t et $\mathbf{a}_t$ en minimisant

$$\sum_{i=1}^{M} L(\mathbf{y}_i, f_{t-1}(\mathbf{x}_i) + \beta_t h(\mathbf{x}_i ; \mathbf{a}_t))$$

(b) Mettre à jour $f_t(\mathbf{x}) = f_{t-1}(\mathbf{x}) + \beta_t h(\mathbf{x} ; \mathbf{a}_t)$

3 **fin**

Résultat :

$$\hat{f}(\mathbf{x}) = f_T(\mathbf{x})$$

Si l'on utilise la fonction des moindres carrés comme fonction de perte, la ligne 2a de l'algorithme (2) se simplifie à

$$\begin{aligned} L(y_i, f_{t-1}\mathbf{x}_i) + \beta h(\mathbf{x}_i ; \mathbf{a}) &= (y_i - f_{t-1}(\mathbf{x}_i) - \beta h(\mathbf{x}_i ; \mathbf{a}))^2 \\ &= (r_{it} - \beta h(\mathbf{x}_i ; \mathbf{a}))^2, \end{aligned} \tag{2.9}$$

où r_{it} sont les résidus de la i^e observation à l'itération courante.

2.2.3 Gradient Boosting Machine

Dans le même esprit d'approximation adaptative pas-à-pas, (Friedman, 2002) a proposé sous l'acronyme MART (multiple additive regression trees) puis sous celui de GBM, une famille d'algorithmes basés sur une fonction perte supposée convexe et différentiable notée l.

Le principe de base est le même que pour AdaBoost : construire une séquence de modèles de sorte que chaque étape, chaque modèle ajouté à la combinaison, apparaisse comme un pas vers une meilleure solution. La principale innovation est que ce pas est franchi dans la direction du gradient de la fonction perte l, afin d'améliorer les propriétés de convergence. Une deuxième idée consiste à estimer le gradient par un arbre de régression (section 2.1).

Dans le modèle adaptatif pas-à-pas dans la section précédente, on avait :

$$\hat{f}_t(\mathbf{x}) = \hat{f}_{t-1}(\mathbf{x}) + \beta_t h(\mathbf{x};\mathbf{a}_t)$$

qu'on transforme en une décente du gradient :

$$\hat{f}_t(\mathbf{x}) = \hat{f}_{t-1} - \beta_t \sum_{i=1}^{M} \nabla_{f_{t-1}} L(y_i; f_{t-1}(\mathbf{x}_i)).$$

Plutôt que de chercher un meilleur classifieur comme avec AdaBoost, on cherche un meilleur pas de descente B :

$$\min_{\beta} \sum_{i=1}^{M} \left[L\Big(y_i, f_{t-1}(\mathbf{x}_i) - \beta \frac{\partial L(y_i, f_{t-1}(\mathbf{x_i}))}{\partial f_{t-1}(\mathbf{x_i})}\Big)\right]. \tag{2.10}$$

L'algorithme (3) décrit le GBM pour un problème de régression.

Algorithme 3 : Gradient Boosting Machine

1 We initialize $f_0(\mathbf{x})$ to a constant, $f_0(\mathbf{x}) = \text{argmin}_\beta \sum_{i=1}^{M} L(y_i, \beta)$;

2 **pour** $t \leftarrow 1$ **à** T **faire**

(a) calculate

$$r_i = -\Big[\frac{\partial L(y_i, f(\mathbf{x_i}))}{\partial f(\mathbf{x_i})}\Big]_{f(\mathbf{x})=f_{t-1}(\mathbf{x})}, i = \{1, \ldots, M\}$$

(b) adjust a regression model r_i with least squares methods using $\mathbf{x}_i$ as the observation vector and obtain an estimate $\mathbf{a}_t$ of $\beta_t h(\mathbf{x}_i ; \mathbf{a})$

(c) Estimate β_t by minimizing $L(\mathbf{y}_i, f_{t-1}(\mathbf{x}_i) + \beta_t h(\mathbf{x}_i ; \mathbf{a}_t))$

(d) Update $f_t(\mathbf{x}) = f_{t-1}(\mathbf{x}) + \beta_t h(\mathbf{x} ; \mathbf{a}_t)$

3 **fin**

Résultat :

$$\hat{f}(\mathbf{x}) = f_T(\mathbf{x})$$

Le GBM étend la capacité du Boosting à résoudre les problèmes de régression, il devient ainsi une percée en apprentissage machine. L'algorithme inclut avec succès des éléments statistiques les plus communs, tels que la modélisation additive et le maximum de vraisemblance. Ce faisant, les auteurs ont pu dériver des techniques d'évaluation de la qualité des prédictions du modèle.

Exemple introductif

Soit l'ensemble de données présenté dans la figure (2.4). Nous remarquons que nous avons cinq groupes de points bien séparés. Si nous ajustons une simple régression linéaire sur tous les points, nous obtenons alors une grande variabilité. Toutefois, essayons de découper les observations en sous-ensemble de points, et ajuster un modèle de régression linéaire ou d'arbre de décision sur chaque sous-ensemble de

points.

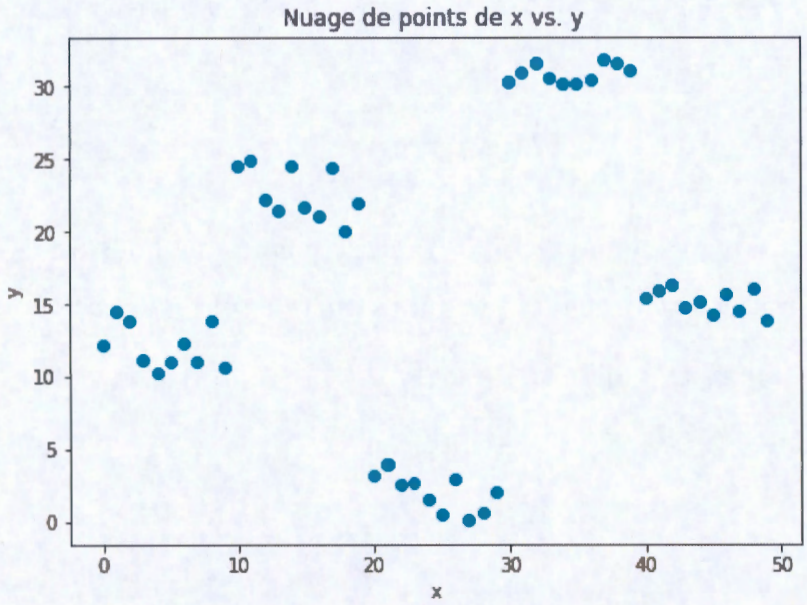


Figure 2.4: Exemple introductif du GBM.

Dans l'algorithme (3) à la ligne 1, nous commençons par initialiser les prédictions, cela se traduit par la ligne 5 du code (A.1) en annexe.

Ensuite, nous appliquons un modèle d'ajustement afin de créer une prédiction de la variable réponse, nous pouvons utiliser un modèle de régression linéaire, un arbre de décision ou tout autre modèle. Ici, nous appliquons un modèle CART comme mentionné dans (Guelman, 2012). Ce qui est équivalent à la ligne 11 dans le même code en annexe.

Puisque dans cet exemple la fonction de perte est la fonction des moindres carrés (voir la fonction 2.9), nous calculons les résidus à la ligne 29 dans le même code (A.1). Finalement, nous additionnons les résidus calculés à la prédiction à la ligne 25 du code. On peut apercevoir le résultat de la première itération à la figure (2.5) où la figure de droite représente l'ajustement aux observations, alors que la figure

de droite représente les résidus de notre modèle d'ajustement.

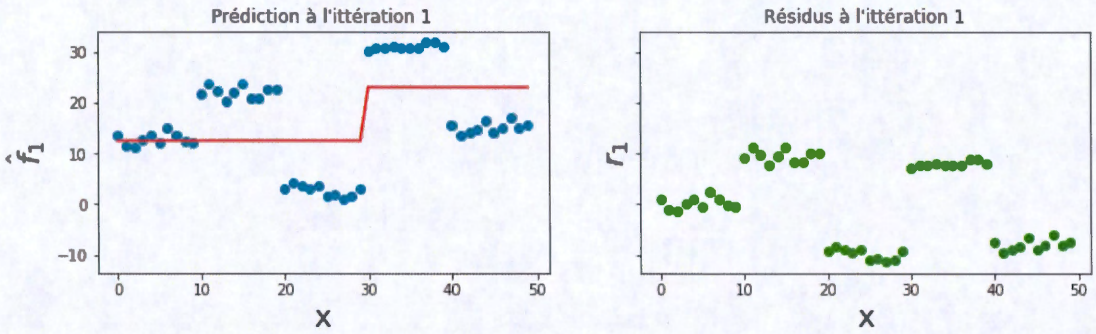
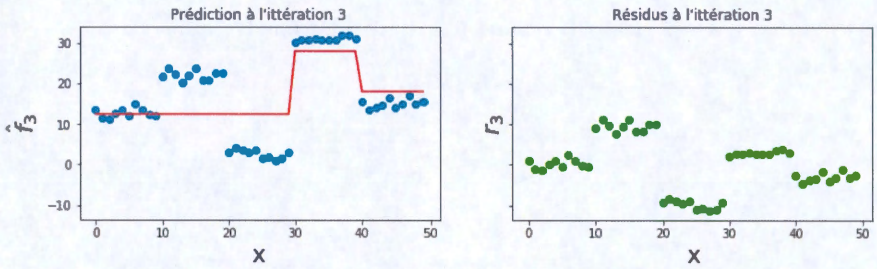
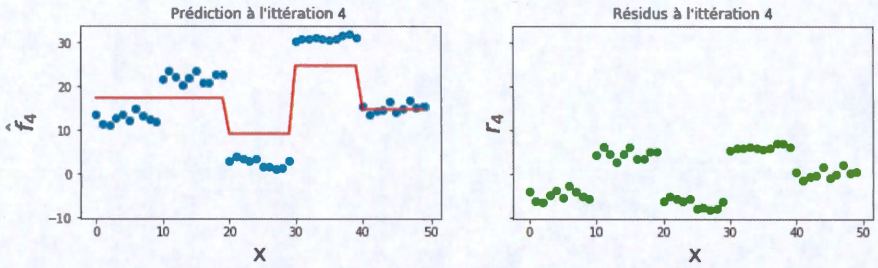


Figure 2.5: Estimation de $\hat{f}_1(x)$ par GBM.

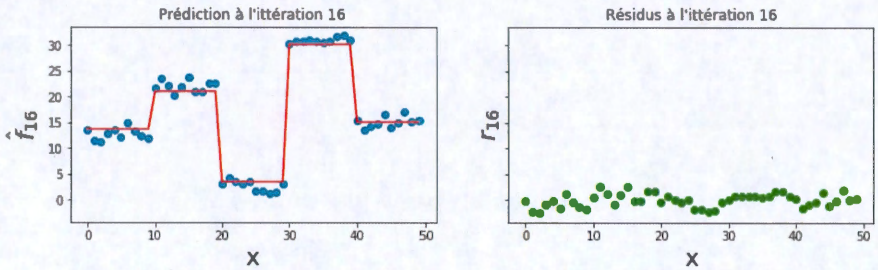
Le résultat de la troisième et quatrième itération est illustré à la figure (2.6a) et (2.6b) respectivement. À la 16^e et 30^e itération, nous obtenons les résultats présentés aux figures (2.6c) et (2.6d) respectivement.



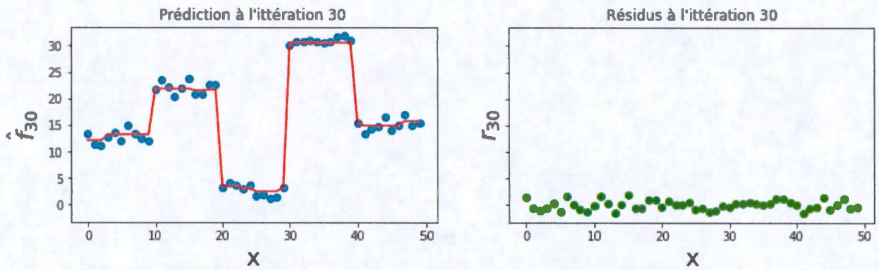
(a) Estimation de $\hat{f}_3(x)$ par GBM.



(b) Estimation de $\hat{f}_4(x)$ par GBM.



(c) Estimation de $\hat{f}_{16}(x)$ par GBM.



(d) Estimation de $\hat{f}_{30}(x)$ par GBM.

Figure 2.6: Illustration de l'ajustement du modèle GBM de la 3^e à 30^e itération.

2.3 GBM Poisson

Le GBM est reconnu comme l'un des meilleurs outils de modélisation prédictive en raison de ses performances obtenues dans plusieurs cas pratiques de prédictions. (Friedman *et al.*, 2001, p. 371) ont montré empiriquement que le GBM a été le plus prédictif sur plusieurs ensembles de données de référence. L'algorithme est intuitif et les résultats du modèle peuvent être interprétés à travers des outils d'interprétation des modèles d'apprentissage automatique tels que l'importance des variables ou de dépendance partielle qui seront détaillés dans la section (2.6). De plus, il est reconnu par sa flexibilité aux valeurs manquantes et cela permet au modélisateur de mettre moins d'effort dans le nettoyage des données pour obtenir un ajustement satisfaisant. Cependant, son application n'est pas toujours directe à tous les problèmes de modélisation. En particulier, nous trouvons difficile d'appliquer le GBM dans la modélisation en assurance IARD.

En effet, les portefeuilles d'assurance sont souvent constitués d'environ 90% des assurés ayant 0 réclamation (Boucher *et al.*, 2007). La plupart des produits d'assurance ont une fréquence de réclamation faible. Cela implique que la majorité des assurés ne font aucune réclamation au cours de la période d'exposition. En raison de cette surdispersion des donnée, nous devons adapter notre modèle GBM.

En assurance IARD, il est très commun d'utiliser une distribution Poisson pour modéliser le nombre de sinistres dans un portefeuille d'assurance. D'abord, rappelons quelques propriétés de cette distribution.

La distribution de Poisson pour une variable aléatoire Y a la fonction de masse de probabilité suivante pour une valeur donnée $Y = y$:

$$P(Y = y|\lambda) = \frac{e^{-\lambda}\ \lambda^{y}}{y\,!} \tag{2.11}$$

Notons que cette distribution est caractérisée par le seul paramètre λ, qui est le taux d'occurrence moyen d'un événement mesuré où l'on suppose que les évènements sont rares. Il est déterminé par un ensemble de $p-1$ prédicateurs $\mathbf{X} = (X_1, X_2, \ldots, X_{p-1})$. L'expression reliant ces quantités est

$$\lambda = \exp\{\mathbf{X}\beta\}.$$

Ainsi, le modèle fondamental de régression de Poisson pour l'observation i est donné par

$$P(Y_i = y_i|\mathbf{X}_i, \beta) = \frac{e^{-\exp\{\mathbf{X}_i\beta\}} \exp\{\mathbf{X}_i\beta\}^{y_i}}{y_i\,!}, \quad y_i = 0, 1, \ldots$$

Autrement dit, pour un ensemble donné de prédicteurs, le résultat suit une distribution de Poisson avec un taux $\exp\{\mathbf{X}\beta\}$. Pour un échantillon de taille n, la vraisemblance d'une régression de Poisson est donnée par :

$$\begin{aligned} L(\beta;y,\mathbf{X}) &= \prod_{i=1}^{n} f(y_i,\lambda) = \prod_{i=1}^{n} \frac{e^{-\lambda}\,\lambda^{y_i}}{y_i!} \\ \log[L(\beta;y,\mathbf{X})] &= \sum_{i=1}^{n} \log\left(\frac{e^{-\lambda}\,\lambda^{y_i}}{y_i!}\right) \\ &= \sum_{i=1}^{n} \log(e^{-\lambda}\lambda^{y_i}) - \log(y_i!) \qquad (2.12) \\ &= \sum_{i=1}^{n} \log(e^{-\lambda}) + \log(\lambda^{y_i}) - \log(y_i!) \\ &= \sum_{i=1}^{n} \lambda + y_i \log(\lambda) - \log(y_i!), \end{aligned}$$

alors que la fonction de déviance est donnée par :

$$
\begin{aligned}
D &= 2[l(y;y) - l(y;\lambda)] \\
&= 2\Big[\sum_{i=1}^{n} y_i + y_i \log(y_i) - \log(y_i!) \\
&\quad - \sum_{i=1}^{n} \lambda + y_i \log(\lambda) - \log(y_i!)\Big] \\
&= 2\Big[\sum_{i=1}^{n} \lambda - y_i + y_i\Big(\log(y_i) - \log(\lambda)\Big)\Big].
\end{aligned}
\tag{2.13}
$$

Ainsi, nous obtenons les résidus de déviance en dérivant l'expression 2.13 par rapport à $f_t(\mathbf{x})$;

$$
\begin{aligned}
r_i &= -\Big[\frac{\partial L(y_i, f_t(\mathbf{x})}{\partial f_t(\mathbf{x})}\Big] \\
&= \frac{-2\sum \partial(\lambda - y_i + y_i\ (\log(y_i) - \log(\lambda))}{\partial f_t(\mathbf{x})} \\
&= -2\sum \frac{\partial(e^{f_t(\mathbf{x})} - y_i + y_i\ (\log(y_i) - f_t(\mathbf{x}))}{\partial f_t(\mathbf{x})} \\
&= \boxed{-2\sum (y_i - e^{f(\mathbf{x_i})})}.
\end{aligned}
\tag{2.14}
$$

Lorsque nous prenons la déviance comme fonction de perte de la Poisson, CART divise les données en J noeuds (Lee *et al.*, 2015) à chaque itération. À chaque noeud, le meilleur pas de descente B tel que décrit à l'équation (2.10) pour la

Poisson est $\log\left(\frac{\sum y_t}{\sum e^{f_t(x_t)}}\right)$;

$$
\begin{aligned}
\mathrm{L}(y_i, f(\mathbf{x}_i)) &= -y_i f(\mathbf{x}_i) + \exp(f(\mathbf{x}_i)) \\
\Rightarrow \sum_{i;\mathbf{x}_i \in \mathbb{R}_2^{(t)}} \mathrm{L}(y_i, f(\mathbf{x}_i)) &= \sum_{i;\mathbf{x}_i \in \mathbb{R}_2^{(t)}} -y_i(f_{t-1}(\mathbf{x}_i) + \beta) + \exp(f_{t-1}(\mathbf{x}_i) + \beta) \\
\Rightarrow \frac{\partial}{\partial \beta}\Big[\sum_{i;\mathbf{x}_i \in \mathbb{R}_2^{(t)}} \mathrm{L}(y_i, f_{t-1}(\mathbf{x}_i) + \beta)\Big] &= \sum_{i;\mathbf{x}_i \in \mathbb{R}_2^{(t)}} y_i + \exp(f_{t-1}(\mathbf{x}_i)) \exp(\beta) \\
\Rightarrow \frac{\partial}{\partial \beta}\Big[\sum_{i;\mathbf{x}_i \in \mathbb{R}_2^{(t)}} \mathrm{L}(y_i, f_{t-1}(\mathbf{x}_i) + \beta)\Big] &= 0 \\
\Leftrightarrow \exp(\beta) \sum_{i;\mathbf{x}_i \in \mathbb{R}_2^{(t)}} \exp(f_{t-1}(\mathbf{x}_i)) &= \sum_{i;\mathbf{x}_i \in \mathbb{R}_2^{(t)}} y_i \\
\Leftrightarrow \beta &= \boxed{\log\left(\frac{\sum y_i}{\sum \exp(f_{t-1}(\mathbf{x}_i))}\right)}
\end{aligned}
\tag{2.15}
$$

Maintenant si nous remplaçons la ligne (2a) dans l'algorithme 3 par l'expression (2.14), ainsi que la ligne (2c) par l'expression 2.15 nous obtenons alors l'algorithme (4) GBMP.

Nous pouvons apercevoir ce changement dans le code (A.1) où nous exprimons $y_i - e^{f(\mathbf{x}_i)}$ par la fonction `_gradient`, ainsi que l'expression $\log\left(\frac{\sum y_i}{\sum \exp(f_{t-1}(\mathbf{x}_i))}\right)$ par la fonction `_terminal_node_estimates` dans le code (A.2).

2.4 Régularisation

Le Boosting est un algorithme qui risque de converger avec exactitude, donc éventuellement vers une situation de surapprentissage. Pour corriger ce problème, dans sa version *Stochastic gradient Boosting*, (Friedman, 2002) a proposé deux autres améliorations de l'algorithme GBM, à savoir la régularisation par rétrécissement (*shrinkage*) des facteurs affaiblis et le sous-échantillonnage (*subsampling*) aléatoire

Algorithme 4 : Gradient Boosting Machine Poisson (GBMP)

1 We initialize $f_0(\mathbf{x})$ to a constant, $f_0(\mathbf{x}) = (\frac{\sum_{i=1}^{M} y_i}{M})$;

2 **pour** $t \leftarrow 1$ **à** T **faire**

(a) calculate

$$r_i = \boxed{y_i - e^{f_{t-1}(\mathbf{x_i})}} \quad , i = \{1, \ldots, M\}$$

(b) Find the best split at to form J partitions using standard CART

(c) calculate $\beta_t = \boxed{\log\left(\frac{\sum y_i}{\sum \exp(f_{t-1}(\mathbf{x}_i))}\right)}$

(d) Update $f_t(\mathbf{x}) = f_{t-1}(\mathbf{x}) + \beta_t h(\mathbf{x};\mathbf{a}_t)$

3 **fin**

Résultat :

$$\hat{f}(\mathbf{x}) = f_T(\mathbf{x})$$

de prédicteurs à chaque étape.

2.4.1 Rétrécissement des facteurs affaiblis

L'approche adoptée par (Friedman, 2002) consiste à réduire la contribution de chaque arbre d'un facteur $\tau \in (0, 1]$ dans l'estimation $\hat{f}$ courante. La mise en œuvre la plus simple du rétrécissement dans l'algorithme GBM se traduit par la modification de la ligne (2d) de l'algorithme (3) par

$$f_t(\mathbf{x}) = f_{t-1}(\mathbf{x}) + \tau \cdot \beta_t h(\mathbf{x};\mathbf{a}_t). \tag{2.16}$$

Le paramètre τ peut être considéré comme contrôleur du taux d'apprentissage. Des valeurs plus petites de τ (plus de rétrécissement) entraînent un plus grand T, soit

le nombre d'itérations. Cependant,τ et T ne fonctionnent pas indépendamment, de plus petites valeurs de τ conduisent à des valeurs plus grandes de T pour les mêmes données d'entraînement, de sorte qu'il y a un compromis à faire.

Empiriquement, il a été montré (Friedman, 2002) que des valeurs plus petites de τ favorisent une meilleure erreur de test et requièrent des valeurs de T plus grandes. En fait, la meilleure stratégie semble être dans la réduction de τ dans un ordre très petit ($\tau < 0.1$) et donner une valeur de T que nous verrons un peu plus dans ce document comment trouver les meilleures valeurs de ces paramètres.

Presque tous les algorithmes de modèles utilisés dans l'apprentissage automatique ont un ensemble de coefficients de calibration (*Hyperparameters*) affectent la façon dont l'algorithme d'apprentissage ajuste le modèle.

2.4.2 Sous-échantillonnage

La plus simple des procédures de régularisation introduites pour les GBM est le sous-échantillonnage. La procédure de sous-échantillonnage a montré qu'elle améliorait les propriétés de généralisation du modèle, tout en réduisant les efforts de calcul requis (Friedman, 2002).

L'idée derrière cette méthode est d'introduire un certain caractère aléatoire dans la procédure d'ajustement. À chaque itération d'apprentissage, seule une partie η (tirée aléatoirement et sans remplacement) des observations est utilisée pour former un apprenant de base. Le reste de l'algorithme est identique. (Friedman *et al.*, 2001, .p365) suggèrent une valeur typique pour $\eta = 1/2$ comme taux d'échantillonnage des observations, bien que pour un grand N, η peut être substantiellement plus petit que $1/2$. Non seulement l'échantillonnage réduit le temps de calcul de la même fraction η, mais dans de nombreux cas, il produit un modèle

plus précis.

2.5 Calibration du modèle

Dans la section précédente, nous avons vu deux paramètres de contrôle qui permettent d'éviter le surajustement. Ces derniers prennent des valeurs numériques qui sont fixées au départ. L'ensemble de toutes les combinaisons de valeurs pour ces paramètres de contrôle est appelé l'espace hyperparamètre. Nous aimerions trouver un ensemble de valeurs d'hyperparamètres qui nous donne le meilleur modèle pour nos données dans un laps de temps raisonnable. Ce processus s'appelle l'optimisation hyperparamétrique. Pour savoir que, par exemple $\tau = 0.015$ est meilleur (pour les mêmes données) que $\tau = 0.010$, nous testons les deux scénarios sur la partie test de la validation du modèle comme illustré à la figure (2.3). Un τ qui donne la plus petite erreur est alors favorisé. Cependant, cette espace devient vite très large, imaginons seulement un τ prenant 11 valeurs, $\tau = 0.05, 0.010, \ldots, 0.1$ ainsi qu'un η qui peut prendre 11 valeurs,$\eta = 0.40, 0.41, \ldots, 0.50$. Cela nous donne comme espace de $11 * 11 = 121$ scénarios possibles.

Depuis son introduction, plusieurs autres hyperparamètre ont été ajoutés par la communauté. Cela a un impact sur le temps que ça prend pour converger un modèle. En effet, imaginons seulement l'ajout d'un autre paramètre pouvant prendre 10 valeurs, notre espace contiendrait alors 1210 modèles à tester. Si le temps de convergence d'un seul modèle est d'environ 5 minutes (ce qui est très rapide déjà), le temps de calcul total serait d'environ 100 heures. Heureusement, aujourd'hui la plupart des librairies des langages de programmation en apprentissage machine contiennent des fonctions qui permettent d'optimiser un choix d'hyperparamètres. Ces librairies sont basées sur des techniques d'optimisation allant des plus naïves comme forcer la recherche dans tout l'espace des paramètres, à d'autres tech-

niques plus sophistiquées comme la recherche aléatoire (Bergstra et Bengio, 2012) ou l'optimisation bayésienne (Snoek *et al.*, 2012). Sans oublier que la plupart des librairies du GBM contiennent aujourd'hui la possibilité d'utiliser un processeur graphique GPU afin de trouver les hyperparamètres optimaux. Cela réduit considérablement le temps de calcul puisqu'un GPU a le pouvoir de paralléliser les calculs sur des milliers de cœurs de calculs.

2.6 Interprétation du modèle

Les arbres de décision uniques sont hautement interprétables comme nous l'avons démontré à l'exemple de prédictions des salaires de jours de baseball. L'ensemble du modèle peut être entièrement représenté par un simple graphique bidimensionnel (arbre binaire) très facile à comprendre comme illustré à la figure (2.2b).

Et malgré la simplicité d'un arbre de décision, quand il y a des milliers d'arbres dans notre modèle comme dans les GBM, il devient quasi impossible de faire de telle interprétation aussi simpliste. Les combinaisons linéaires d'arbres perdent cette simplicité et doivent donc être interprétées différemment.

Plusieurs outils ont été conçus pour atténuer les problèmes d'interprétation des modèles GBM. Dans cette section, nous décrivons les outils les plus courants pour l'interprétation du GBM.

2.6.1 Importance relative des variables prédictives

L'importance relative d'une variable quelconque dans le modèle est le degré de participation de cette dernière à la régression. Des informations pertinentes sont obtenues par le calcul et la représentation graphique d'indices proportionnels à l'importance de chaque variable dans le modèle agrégé.

Définissons l'influence de la variable j dans un seul arbre T. Considérons que l'arbre possède L divisions, donc nous recherchons tous les nœuds non terminaux (divisons) de la racine au niveau $L-1$ de l'arbre. (Breiman *et al.*, 1984) ont proposé une définition de l'influence de la variable

$$\text{Influence}_j(T) = \sum_{t=1}^{L-1} I_t^2 \mathbb{1}_{\{S_t=j\}}. \tag{2.17}$$

Cette mesure est basée sur le nombre de fois qu'une variable est sélectionnée pour la division, c'est-à-dire que la variable de séparation courante S_i est la même que la variable interrogée j. La mesure capture également les poids de l'influence avec l'amélioration empirique au carré I_i^2, attribué au modèle à la suite de cette division. Pour obtenir l'influence globale de la variable j dans l'ensemble, il faut faire la moyenne de cette influence sur tous les arbres :

$$\text{Influence}_j = \frac{1}{M} \sum_{t=1}^{M} \text{Influence}_j(T_i). \tag{2.18}$$

Les influences sont standardisées de manière à ce qu'elles atteignent 100%. Elles sont présentées sous forme d'histogramme où l'on aperçoit cette mesure des variables les plus importantes au moins importantes. Basé sur cette idée, (Fisher *et al.*, 2018) ont proposé une version améliorée de l'importance des variables.

Le concept est très simple : nous mesurons l'importance d'une variable en calculant l'augmentation de l'erreur de prédiction du modèle après avoir permuté toutes les variables. Une variable est « importante » si le brassage de ses valeurs augmente l'erreur du modèle, car dans ce cas le modèle s'est appuyé sur la variable pour la prédiction. Une variable est « faible » si la permutation de ses valeurs laisse l'erreur de modèle inchangée, parce que dans ce cas, le modèle a ignoré la variable pour la prédiction.

L'importance des variables ne fournit aucune explication sur la manière dont la variable explicative affecte réellement la variable réponse. Toutefois, les influences résultantes peuvent ensuite être utilisées pour les procédures de sélection de variables.

Algorithme 5 : Importance des variables par permutation

1 Estimer l'erreur de modèle original $L(y, f(x))$;

2 **pour** *pour chaque variable* $j = 1, \ldots, p$ **faire**

(a) Générer la matrice $\mathbf{X}_j^r$ en permutant la variable j dans les données $\mathbf{X}$.

(b) Estimez l'erreur $\text{err}_j = L(Y, f(\mathbf{X}_j^r))$ d'après les prédictions des données permutées.

(c) Calculer l'importance de la permutation $\text{Imp\_}j = \text{err}_j - \text{err}_o$.

3 **fin**

Résultat : Calculer l'importance moyenne $\text{Imp}_j = \frac{1}{K}\sum_{t=1}^{K} \text{Imp}_j^k$ sur chacune des k partie des données de notre validation croisée k partie et trier les variables par Imp par ordre décroissant.

2.6.2 Graphique de la dépendance partielle

La visualisation est l'un des moyens d'interprétation le plus simple à comprendre. Après avoir identifié les variables les plus pertinentes, l'étape suivante consiste à tenter de comprendre la nature de la dépendance de l'estimation de $\hat{f}(x)$ sur le sous-ensemble $\mathbf{x}_S$ de taille $l < p$ du vecteur $\mathbf{x}$. La dépendance partielle implique la démonstration de l'effet d'une variable sur la réponse modélisée après avoir marginalisé toutes les autres variables explicatives.

Bien que la façon correcte d'obtenir des fractions marginales serait d'intégrer numériquement d'autres variables sur une grille de valeurs, en pratique cela peut être très gourmand en calcul. Une approche plus facile est donc couramment

utilisée, lorsque les variables marginalisées sont fixées avec une valeur constante, égale à la moyenne de leur échantillon. Par exemple, si l'on s'intéresse à l'effet du kilométrage parcouru sur la fréquence des réclamations prédite, nous devons découper le kilométrage parcouru en plusieurs segments, et voir par la suite l'effet moyen qu'a eu chaque segment sur la variable réponse.

Considérons un sous-vecteur $\mathbf{x}_l$ des prédicteurs $X^T = (X_1, X_2, \ldots, X_p)$ indexé par $S \subset \{1, 2, \ldots, p\}$. Soit C le complément où $S \cup C = \{1, 2, \ldots, p\}$. La fonction générale $f(x)$ devrait en principe dépendre de toutes les variables explicatives : $f(x) = f(X_S, X_C)$. Une façon de définir la moyenne (ou la dépendance partielle) de $f(x)$ sur X_S est

$$f_S(X_S) = E_{X_C} f(X_S, X_C). \tag{2.19}$$

C'est une espérance marginale de f, et peut servir de description utile de l'effet du sous-ensemble choisi sur $f(x)$ lorsque, par exemple, les variables dans X_S ont des interactions fortes avec celles de X_c (Friedman *et al.*, 2001, p. 369).

Les dépendances partielles peuvent être estimée par

$$\hat{f}(X_S) = \frac{1}{M} \sum_{i=1}^{M} \hat{f}(x_S, x_{iC}), \tag{2.20}$$

où $\{x_{1C}, x_{2C}, \ldots, x_{NC}\}$ sont les valeurs de X_C se trouvant dans les données d'entraînement. Pour ce faire, il faut transmettre les données pour chaque ensemble de valeurs communes de X_S pour lequel $\bar{f}_S(X_S)$ doit être évalué. Cela peut nécessiter des calculs intensifs.

Il est important de noter que les fonctions de dépendance partielle définies en (2.19) représentent l'effet de X_S sur $f(X)$ après prise en compte des effets (moyens) des autres variables X_C sur $f(X)$. Ils ne sont pas l'effet de X_S sur $f(X)$ en ignorant

les effets de X_C.

Ces graphiques peuvent ne pas être une représentation parfaite des effets capturés, en particulier si les interactions variables ont un impact significatif sur le modèle résultant. Cependant, les diagrammes de dépendance partielle peuvent fournir une base utile pour l'interprétation des modèles.

La même idée avec la visualisation peut être appliquée à des couples de variables, permettant ainsi d'inspecter et d'analyser les interactions les plus importantes. Pour identifier les interactions de l'intérêt, on peut d'abord utiliser l'influence de la variable relative et ensuite produire des courbes de dépendance par paires.

2.6.3 Les valeurs SHAP

La technique des valeurs *SHapley Additive exPlanation* (SHAP) a été proposée dans des articles très récents par (Lundberg et Lee, 2017a) et (Lundberg et Lee, 2017b). Ils sont basés sur les valeurs de Shapley (Shapley, 1953), une technique développée par le mathématicien du même nom, utilisée en théorie des jeux pour déterminer dans quelle mesure chaque joueur dans un jeu collaboratif a contribué à son succès. C'est une méthode d'attribution des paiements aux joueurs en fonction de leur contribution au paiement total. Les joueurs coopèrent au sein d'une coalition et tirent un certain profit de leurs contributions individuellement.

Afin de mieux comprendre l'idée générale derrière cette approche, reprenons un exemple introductif tiré du livre (Molnar, 2019). Supposons que nous avons entraîné un modèle d'apprentissage machine pour prédire le prix des appartements dans une région quelconque. Pour un appartement donné, notre modèle a prédit 300 000 \$. Cet appartement possède une superficie de 50 m^2, est situé au 2^e étage, a un parc à proximité et les animaux sont interdits dans les appartements.

La prédiction moyenne pour tous les appartements de notre base de données est de 310 000 $. La question qu'on pourrait se poser est : dans quelle mesure chaque valeur de caractéristique (les valeurs des variables citées ci-haut) a-t-elle contribué à la prédiction par rapport à la prédiction moyenne ?

La réponse est simple pour les modèles de régression linéaire. L'effet de chaque caractéristique est le poids de la caractéristique multiplié par sa valeur. Cela ne fonctionne qu'en raison de la linéarité du modèle. Pour les modèles plus complexes, nous avons besoin d'une solution différente. Par exemple, (Ribeiro *et al.*, 2016b) suggère des modèles locaux pour estimer les effets. Une autre solution vient de la théorie du jeu coopératif SHAP.

Pour faire le parallèle entre la théorie des jeux originalement introduite par (Shapley, 1953) et l'interprétapilité d'un modèle d'apprentissage machine, nous devons adapter certains termes utilisés dans la théorie des jeux. Le « jeu » est la tâche de prédiction pour une seule observation de l'ensemble de données. Le « gain » est la prédiction réelle pour cette observation moins la prédiction moyenne pour toutes les observations. Les « joueurs » sont les valeurs caractéristiques[5] de l'observation qui collabore pour recevoir le gain (= prédire une certaine valeur). Dans notre exemple de prédiction des prix d'appartements, les valeurs de la caractéristique `parc à proximité`, `animaux interdits`, `superficie` $= 50m^2$ et `2e étage` ont contribué ensemble pour atteindre une prédiction de 300 000 $. Notre objectif est d'expliquer la différence entre la prédiction réelle (300 000 $) et la prévision moyenne (310 000 $) : soit une différence de -10 000 $.

La réponse pourrait être : la `parc à proximité` a contribué à hauteur de 30 000

5. La valeur d'une variable d'une observation en particulier. Par exemple, la valeur caractéristique âge de la 19^e observation est 40. Donc pour la caractéristique « Âge », la valeur de la 19^e observation c'est 40

\$; la `superficie` $= 50m^2$ a contribué 10 000 \$; le `2e étage` a contribué 0 \$; les `animaux interdits` ont contribué - 50 000 \$. Le total des contributions s'élève à -10 000 \$, soit la prédiction finale moins le prix moyen prédit de l'appartement. En effet, on peut tout de suite voir que la valeur Shapley est la contribution marginale moyenne d'une valeur d'une variable dans toutes les coalitions possibles.

Supposons que nous voulons évaluer plus concrètement la contribution de la valeur de la caractéristique `animaux interdits` lorsqu'elle est ajoutée à une coalition de `parc à proximité` et de `superficie` $= 50m^2$.

Nous simulons que seules les caractéristiques `parc à proximité`, `animaux interdits` et `superficie` $= 50m^2$ font partie d'une coalition en tirant au hasard un autre appartement à partir des données et en utilisant sa valeur pour l'élément de l'étage. La valeur `2e étage` a été remplacée par la valeur `1er étage` tiré au hasard. Ensuite, nous prédisons le prix de l'appartement avec cette combinaison (310 000 \$). Dans un deuxième temps, nous retirons la caractéristique `animaux interdits` de la coalition en la remplaçant par une valeur aléatoire de la caractéristique des animaux autorisés/interdits dans l'appartement tiré au hasard. Supposons que le tirage a donné `animaux autorisés` mais il aurait pu être `animaux interdits` à nouveau. Nous prédisons le prix de l'appartement pour la coalition `parc à proximité` et de `superficie` $= 50m^2$ (320 000 \$). La contribution de la caractéristique `animaux interdits` s'est élevée à 310 000 \$ - 320 000 \$ = - 10 000 \$. Cette estimation dépend des valeurs de l'appartement tiré aléatoirement. Nous obtiendrons de meilleures estimations si nous répétons plusieurs fois cette étape d'échantillonnage et faisons la moyenne des contributions. La valeur Shapley est alors la moyenne de toutes les contributions marginales à toutes les coalitions possibles. Pour déterminer la valeur Shapley pour la variable `animaux interdits`, les coalitions suivantes auraient été nécessaires :

— Aucune autre variable (donc seulement `parc à proximité`)

— `parc à proximité`

— `superficie` $= 50m^2$

— `2e étage`

— `parc à proximité` + `superficie` $= 50m^2$

— `parc à proximité` + `2e étage`

— `superficie` $= 50m^2$ + `2e étage`

— `parc à proximité` + `superficie` $= 50m^2$+ `2e étage`.

Pour chacune de ces coalitions, nous calculons le prix prédit de l'appartement avec et sans la valeur des caractéristiques `animaux interdits` et prenons la différence pour obtenir la contribution marginale. La valeur Shapley est la moyenne (pondérée) des contributions marginales. Nous remplaçons les valeurs des caractéristiques qui ne font pas partie d'une coalition par des valeurs de caractéristiques aléatoires de l'ensemble de données de l'appartement pour obtenir une prédiction à partir du modèle d'apprentissage machine. Si nous estimons les valeurs de Shapley pour toutes les valeurs de caractéristiques, nous obtenons la distribution complète de la prédiction (moins la moyenne) entre les valeurs de caractéristiques.

Nous nous intéressons à la façon dont chaque caractéristique affecte la prédiction d'une observation donnée. Dans un modèle linéaire, il est facile de calculer les effets individuels. Voici à quoi ressemble une prédiction de modèle linéaire pour une instance de données :

$$\hat{f}(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p \tag{2.21}$$

où x est l'instance pour laquelle nous voulons calculer les contributions. Chaque

x_j est une valeur de caractéristique, avec $j = 1, \ldots, p$. Le site β_j est le poids correspondant à la caractéristique j.

La contribution ϕ_j de la j^e caractéristique de la prédiction $\hat{f}(x)$ est :

$$\phi_j(\hat{f}) = \beta_j x_j - E(\beta_j X_j) = \beta_j x_j - \beta_j E(X_j) \tag{2.22}$$

où $E(\beta_j X_j)$ est l'estimation de l'effet moyen pour la caractéristique j. La contribution est la différence entre l'effet de caractéristique moins l'effet moyen. Nous savons maintenant dans quelle mesure chaque caractéristique a contribué à la prédiction. Si nous additionnons toutes les contributions de caractéristiques pour une instance, le résultat est le suivant :

$$\begin{aligned}\sum_{j=1}^{p} \phi_j(\hat{f}) &= \sum_{j=1}^{p} (\beta_j x_j - E(\beta_j X_j)) \\ &= (\beta_0 + \sum_{j=1}^{p} \beta_j x_j) - (\beta_0 + \sum_{j=1}^{p} E(\beta_j X_j)) \\ &= \hat{f}(x) - E(\hat{f}(X)).\end{aligned}$$

Il s'agit de la valeur prédite pour une observation donnée x moins la valeur moyenne prédite. Comme nous n'avons généralement pas de poids similaires β pour d'autres types de modèles, nous ne pouvons malheureusement pas généraliser ce calcul pour tous les modèles, nous avons donc besoin d'une solution différente ; la théorie du jeu coopératif. La valeur Shapley est une solution permettant de calculer les contributions de fonctions pour des prédictions uniques pour n'importe quel modèle d'apprentissage machine.

La valeur Shapley est définie par une fonction de valeur v des joueurs dans S. La valeur Shapley d'une valeur de caractéristique est sa contribution au paie-

ment, pondérée et additionnée sur toutes les combinaisons possibles de valeurs de caractéristiques :

$$\phi_j(v) = \sum_{S \subseteq \{x_1,\ldots,x_p\} \setminus \{x_j\}} \frac{|S|\,!(p-|S|-1)\,!}{p\,!}(v(S \cup \{x_j\}) - v(S)) \tag{2.23}$$

où S est un sous-ensemble des caractéristiques utilisées dans le modèle, x est le vecteur des valeurs des caractéristiques de l'instance à expliquer et p le nombre de caractéristiques. $v(S)$ est la prédiction des valeurs des caractéristiques du jeu S qui sont marginalisés par rapport aux caractéristiques non compris dans le jeu S :

$$v_x(S) = \int \hat{f}(x_1, \ldots, x_p) d\mathbb{P}_{x \notin S} - E_X(\hat{f}(X)). \tag{2.24}$$

Nous effectuons en fait plusieurs intégrations pour chaque variable qui n'est pas contenue dans S. Par exemple, avec un modèle d'apprentissage machine fonctionne ayant 4 variables $x1$, $x2$, $x3$ et $x4$ et nous évaluons la prédiction pour la coalition S constitué des valeurs de caractéristiques $x1$ et $x3$:

$$v_x(S) = v_x(\{x_1, x_3\}) = \int_{\mathbb{R}} \int_{\mathbb{R}} \hat{f}(x_1, X_2, x_3, X_4) d\mathbb{P}_{X_2 X_4} - E_X(\hat{f}(X)).$$

On peut remarquer que lorsque l'on a un grand nombre de variables, le calcul de telles intégrales peut devenir très gourmand en temps de calcul. En effet, toutes les coalitions (ensembles) possibles de valeurs de caractéristiques doivent être évaluées avec et sans la j^e caractéristique pour calculer la valeur exacte de Shapley. Lorsque le nombre de variables est grand, la solution exacte à ce problème devient problématique à mesure que le nombre de coalitions possibles augmente exponentiellement à mesure que de nouvelles variables sont ajoutées. Heureusement,

(Štrumbelj et Kononenko, 2014) ses collaborateurs proposent une approximation par échantillonnage Monte-Carlo :

$$\hat{\phi}_j = \frac{1}{M} \sum_{m=1}^{M} \left(\hat{f}(x_{+j}^m) - \hat{f}(x_{-j}^m) \right), \tag{2.25}$$

où $\hat{f}(x_{+j}^m)$ est la prédiction pour x, mais avec un nombre aléatoire de valeurs de caractéristiques remplacées par des valeurs de caractéristiques à partir d'un point de données aléatoire z, sauf pour la valeur respective de la caractéristique j. Le vecteur x_{-j}^m est presque identique à x_{+j}^m, mais la valeur x_j^m est également prise dans l'échantillon x. Nous itérons cette prédiction M fois.

Illustrons cette approximation dans l'algorithme (6) :

Algorithme 6 : Valeurs Shapley pour la valeur d'une seule caractéristique

1 On initialise $f_0(\mathbf{x}) = 0$;

2 **pour** $m \leftarrow 1$ **à** M **faire**

1. Tirage aléatoire de l'instance z dans la matrice de données $\mathbf{X}$
2. Choisir une permutation aléatoire o des valeurs de caractéristiques
   - Ordonner l'instance x : $x_o = (x_{(1)}, \ldots, x_{(j)}, \ldots, x_{(p)})$
   - Ordonner l'instance z : $z_o = (z_{(1)}, \ldots, z_{(j)}, \ldots, z_{(p)})$
3. Construire deux nouvelles instances
   - Avec la variable j : $x_{+j} = (x_{(1)}, \ldots, x_{(j-1)}, x_{(j)}, z_{(j+1)}, \ldots, z_{(p)})$
   - Sans la variable $x_{-j} = (x_{(1)}, \ldots, x_{(j-1)}, z_{(j)}, z_{(j+1)}, \ldots, z_{(p)})$
4. Calculer la contribution marginale : $\phi_j^m = \hat{f}(x_{+j}) - \hat{f}(x_{-j})$

3 **fin**

Résultat : La moyenne des valeurs Shapley : $\phi_j(x) = \frac{1}{M} \sum_{m=1}^{M} \phi_j^m$

2.6.4 Interaction des variables par SHAP

Lorsque des variables interagissent les unes avec les autres dans un modèle de prédiction, la prédiction ne peut pas être exprimée comme la somme des effets des variables, car l'effet d'une variable dépend de la valeur de l'autre variable.

Si un modèle d'apprentissage machine fait une prédiction basée sur deux variables, nous pouvons décomposer la prédiction en quatre termes : un terme constant, un terme pour la première variable, un terme pour la deuxième variable et un terme pour l'effet d'interaction entre les deux variables. L'interaction entre deux variables est le changement dans la prédiction qui se produit en variant les variables après avoir pris en compte les effets des variables individuelles.

Dans leur article, (Lundberg et Lee, 2017b) ont montré que si l'on considère les interactions par paires, on obtient une matrice de valeurs d'attribution représentant l'impact de toutes les paires de caractéristiques sur une prédiction de modèle donnée. Puisque les valeurs Shapley sont basées sur les valeurs classiques de Shapley de la théorie des jeux, une extension naturelle des effets d'interaction peut être obtenue grâce à l'indice d'interaction Shapley plus moderne

$$\phi_{i,j} = \sum_{S \subseteq \{x_1,\ldots,x_p\} \setminus \{x_j\}} \frac{|S|\,!(p-|S|-2)\,!}{2(p-1)\,!}(\nabla_{ij}(S)), \tag{2.26}$$

quand $i \neq j$, et

$$\begin{aligned} \nabla_{ij}(S) &= v(S \cup \{x_{ij}\}) - v(S \cup \{x_i\}) - v(S \cup \{x_j\}) + v(S) \\ &= v(S \cup \{x_{ij}\}) - v(S \cup \{x_j\}) - [v(S \cup \{x_i\}) - v(S)]. \end{aligned} \tag{2.27}$$

Dans l'équation 2.26, la valeur d'interaction Shapley entre la caractéristique i et la caractéristique j est répartie également entre chaque caractéristique, de sorte que

$\phi_{i,j} = \phi_{j,i}$ et l'effet d'interaction total est $\phi_{i,j} + \phi_{j,i}$. Les principaux effets pour une prédiction peuvent alors être définis comme la différence entre la valeur Shapley et les valeurs d'interaction Shapley pour une caractéristique :

$$\phi_{i,j} = \phi_i - \sum_{j \neq i} \phi_{i,j}. \tag{2.28}$$

Alors que les valeurs d'interaction Shapley peuvent être calculées directement à partir de l'équation 2.26. Dans leur article (Lundberg et Lee, 2017b), les auteurs ont détaillé un algorithme qu'ils ont développé pour réduire considérablement leur coût en temps de calcul afin d'estimer les valeurs d'interaction Shapley.

CHAPITRE III

APPLICATION À L'ASSURANCE AUTOMOBILE

3.1 Introduction

Comme nous l'avons mentionné en introduction, l'assureur doit mesurer le risque lorsqu'il vend une couverture d'assurance à ses assurés. Cet exercice consiste essentiellement à classer les assurés en fonction de leurs caractéristiques de risque afin de leur assigner une prime d'assurance personnalisée. Ces caractéristiques sont des variables définissant chaque assuré et sont connues par l'assureur au moment de la souscription. Pour cela, on les appelle des variables a priori. Par exemple, l'âge de l'assuré, son sexe, son état civil ou toute autre caractéristique pertinente à la tarification sont des informations communiquées à l'assureur au moment même de la souscription.

Ces caractéristiques du risque observables sont généralement considérées comme des covariables non aléatoires. Par exemple, si un conducteur possède deux véhicules, nous savons très bien que le risque de collision ne peut s'appliquer que sur un seul véhicule à la fois. D'autres caractéristiques de risque sont inobservables et doivent être considérées comme des paramètres inconnus. En revanche, un assuré qui conduit régulièrement son véhicule avec des facultés affaiblies est un très mauvais risque pour l'assureur, mais ce dernier ne peut tenir compte de cette variable

dans la tarification car elle ne peut être connue qu'en cas de condamnation (perte de permis de conduire).

3.2 Données télématiques

De nos jours, la plupart des assureurs automobiles proposent à leurs clients d'installer dans leurs véhicules des modules (ou d'application mobile) pouvant capter les informations télématiques, telles que la distance parcourue ainsi qu'une panoplie d'informations sur les habitudes de conduite à chaque trajet. S'ils acceptent, les assurés pourraient ainsi bénéficier des rabais sur leurs primes d'assurance, qui pourraient être dans certains cas très importants. Par exemple, dans un document interne de (The Co-operators Group Limited, 2017), un assureur IARD canadien, peut accorder jusqu'à 25% de rabais sur la prime d'assurance automobile. Évidemment, cela engendre un important investissement financier de la part de l'assureur, pensons seulement à toute l'infrastructure informatique nécessaire à la captation et au stockage de ces données très volumineuses. En échange, l'assureur obtient des renseignements pouvant révéler de riches informations sur les habitudes de conduite des assurés à des fins de tarification.

Dans ce travail, grâce à la Chaire *Co-operators* en analyse des risques actuariels (CARA), nous avons eu la chance de mettre la main sur des données télématiques abrégées de l'assureur canadien *The Co-operators*. Ces données sommaires concernent seulement les assurés résidant dans la province de l'Ontario. Nous n'avons aucune information sur les détails du trajet, mais plutôt un sommaire sur les habitudes de conduite durant toute la période observée. Donc nous n'avons pas le kilométrage parcouru sur chaque trajet, mais nous avons le kilométrage exact ainsi que le nombre de trajets sur toute la période d'exposition. Chaque observation est identifiée par le numéro d'identification du véhicule *Vehicle identification*

number (VIN) attribué par le manufacturier automobile. Chaque colonne de notre base de données est une caractéristique qui nous donne des informations détaillées sur l'assuré, son véhicule ainsi que ses habitudes de conduite.

Afin d'enrichir notre base de données, nous y avons ajouté d'autres caractéristiques que nous avons pu extraire à partir d'une interface de programmation d'application (API) de la *National Highway Traffic Safety Administration.* Nous avons pu extraire la marque et modèle du véhicule (Thoma, 2019) qui nous manquaient dans notre base de données originale. Grâce aux codes postaux des assurés, nous avons pu ajouter les données démographiques par code postal à partir des données de Statistiques Canada (Canada, 2019), tels que le revenu moyen et la population.

Dans le tableau (3.1) nous avons quelques statistiques de sept des cent vingt et une variables explicatives de l'ensemble de données d'entraînement.

	Variable	Nombre	Moyenne	Écart-type	min	25.00%	50.00%	75.00%	max
X1	Distance parcourue observée	65489	7,994.38	7,288.81	0.1	2616	5903.9	11255.3	76271.8
X2	Nombre de jours d'exposition	65489	182.94	113.11	1	89	181	273	394
X3	Âge du véhicule	65489	5.76	4.47	-2	2	5	9	20
X4	Pourcentage d'utilisation par par jour	65489	0.80	0.19	0.03	0.72	0.86	0.95	1.00
X5	Nombre de jours d'utilisation par semaine	65489	5.63	1.32	0.2	5	6	6.6	7
X6	Moyenne du temps de conduite	65489	383.76	203.81	1	243	356	490	3141
X7	Nombre moyen d'utilisations de la voiture par semaine	65489	32.29	13.46	0	23	31	41	60
...	...	...	...	...	...	...	...	...	...

Tableau 3.1: Statistiques descriptives de quelques variables de l'ensemble de données d'entraînement.

3.3 Statistiques descriptives

Dans cette section, nous ferons quelques explorations statistiques sur notre base de données afin d'essayer de découvrir certains résultats pertinents avant l'application même d'appliquer notre modèle. Pour ce faire, nous avons divisé nos variables

explicatives en trois catégories ;

1. les caractéristiques de l'assuré,
2. les caractéristiques du véhicule et
3. les habitudes de conduite.

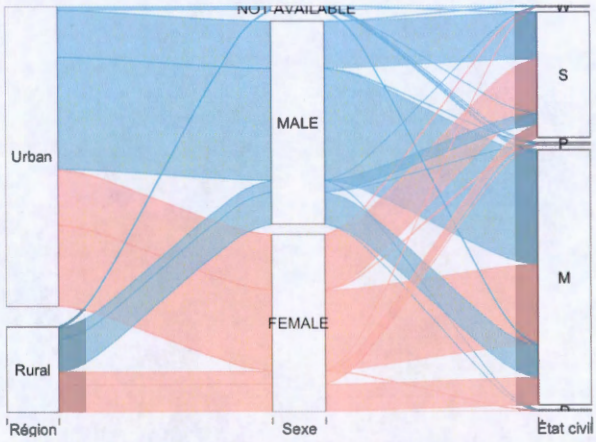
3.3.1 Caractéristiques de l'assuré

L'âge moyen de nos assurés est de 49 ans, ils habitent en majorité en région urbaine et les hommes sont à peu près aussi nombreux que les femmes. Pour la plupart, ils sont mariés tels qu'illustré à la figure (3.1a).

Comme nous l'avons mentionné auparavant, notre base de données concerne seulement les assurés habitant en Ontario, nous pouvons avoir un aperçu sur la fréquence des codes postaux dans la figure (3.1b). Afin d'avoir une meilleure idée sur la distribution du lieu de résidence de nos assurés sur une carte géographique, nous avons récupéré l'altitude et la longitude de chaque code postal, la densité de la distribution de ces codes postaux est illustrée à la figure (3.2). Nous pouvons en constater que la plupart des nos assuré se trouvent dans les grandes métropoles de l'Ontario, soit les régions de Toronto, Mississauga et Ottawa.

Afin d'alléger le nombre de catégories dans la variable « code postal » de notre base de données, nous avons regroupé les lieux de résidence de nos assurés par Région de tri d'acheminement (RTA) [1]. Cela a été possible grâce au croisement de notre base de donner avec les données publiques de Statistique Canada (Canada, 2019). Par la même occasion, nous avons enrichi notre base en y ajoutant plus d'informations telles que ; le nombre d'habitants ainsi que le score de crédit moyen par RTA.

1. De l'anglais *Forward Sortation Area (FSA)*



(a) Région de résidence, le sexe et l'état civil des assurés.

(b) Nuage des codes postaux.

Figure 3.1: Caractéristiques des assurés.

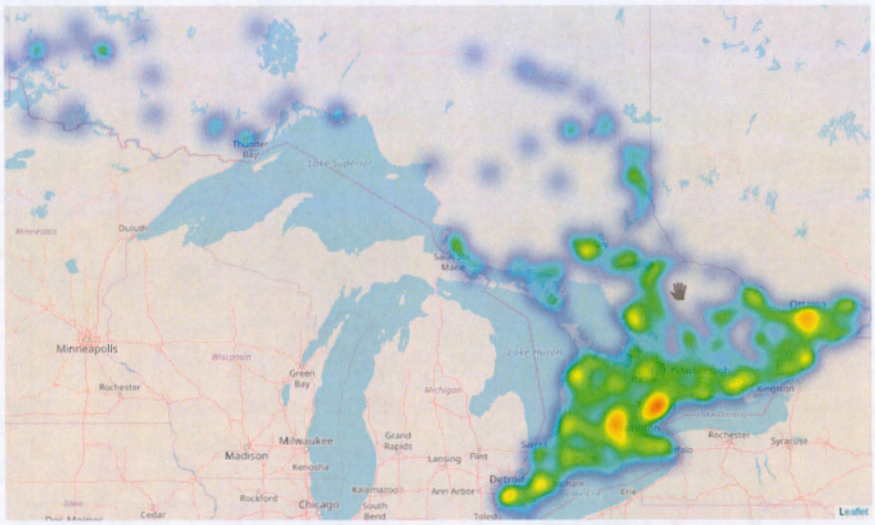


Figure 3.2: Distribution des lieux de résidence des assurés par code postal.

3.3.2 Caractéristiques du véhicule

Il a été possible de trouver beaucoup d'information sur le véhicule en utilisant le VIN de ce dernier comme référence de recherche. Pour ce faire, nous avons utilisé la librairie (Thoma, 2019) afin de se connecter à une interface de programmation applicative (souvent désignée par le terme API pour *application programming interface*). Nous avons pu récupérer beaucoup de données, nous en avons gardé seulement la marque et le type de la carrosserie. Dans la figure (3.3), nous avons tracé un graphique de barres pour illustrer la distribution de ces deux variables. Nous pouvons remarquer que la plupart des véhicules sont des voitures de tourisme (*passenger car*), les marques les plus populaires sont ; Toyota, Honda et Ford.

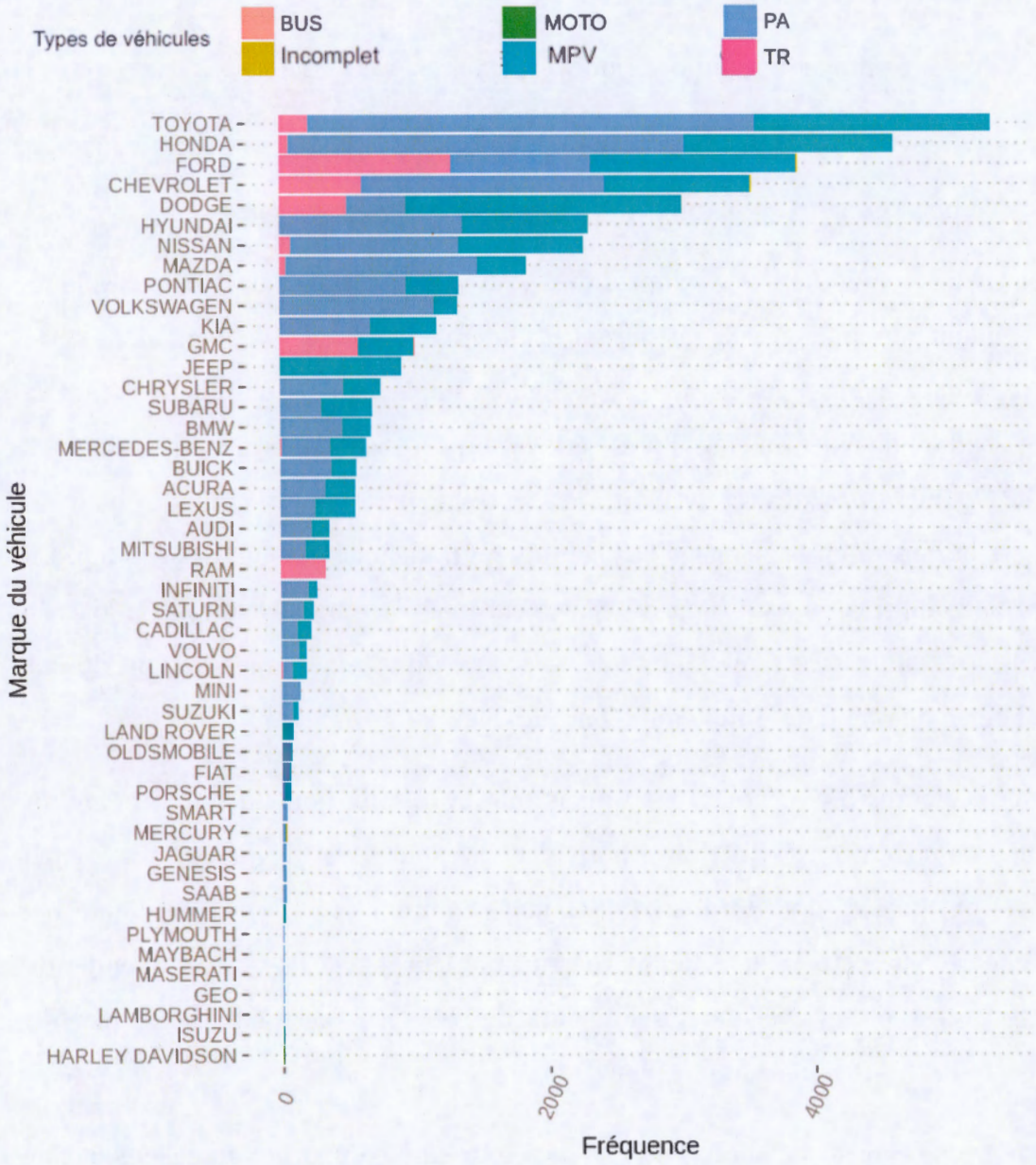


Figure 3.3: Distribution des marques de véhicule par type de carrosserie.

3.3.3 Habitudes de conduites

Grâce aux informations contenues dans les données télématiques, nous avons pu constater certains faits intéressants. Par exemple, dans la figure (3.4a), nous avons

illustré les trois variables suivantes :

— le pourcentage de conduite (jour de semaine/ week-end)

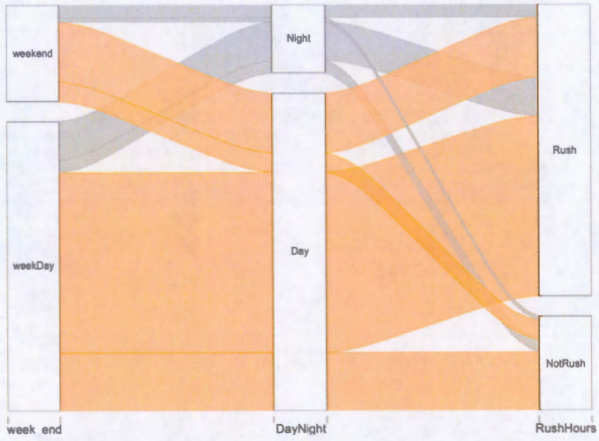
— le pourcentage de conduite (jours/ nuit)

— le pourcentage de conduite (aux heures de pointe / ou non).

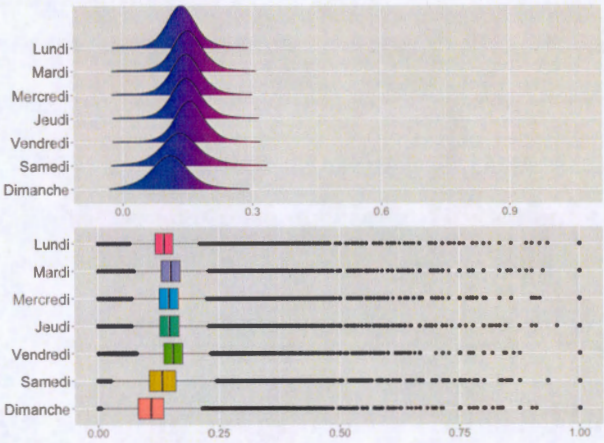
Nous avons remarqué que la plupart des assurés conduisent leur véhicule les jours de semaine, de jours et surtout durant les heures de pointe. D'ailleurs dans cette même figure, nous pouvons nous questionner sur les heures de pointe durant le week-end ou de nuit. En effet, après vérification avec notre fournisseur de données (The Co-operators), il s'est avéré que les heures de pointe sont définies comme des plages horaires qui ne tiennent pas compte de la définition jours de semaine. En plus, la définition de jours/nuit est seulement une plage horaire durant la journée sur toute l'année sans tenir compte des saisons où il commence à faire nuit durant les heures de pointe (à 16 :30 au début janvier).

Dans la figure (3.4b), nous remarquons que la distribution du pourcentage du temps de conduite est distribuée normalement pour tous les jours. La moyenne est autour de 15% pour les jours de semaine et autour de 11% pour le dimanche. D'ailleurs, sur cette même figure (partie inférieure), l'écart-type du temps de conduite est plus élevé que sur les jours de semaine, ceci peut s'expliquer du fait que les assurés font de plus longs trajets le week-end.

Nous avons également analysé la distance moyenne parcourue annuellement par RTA. Nous pouvons voir dans la figure (3.5b) qu'à Toronto, on parcourt moins de kilométrage par rapport à la région d'Ottawa ou à Mississauga. Cela pourrait s'expliquer par l'utilisation du transport en commun pour les déplacements domicile-travail. On pourrait creuser un peu plus cette analyse si l'on avait les coordonnées géographiques des trajets des assurés afin d'en ressortir une récurrence des déplacements quotidiens.



(a) Les différents moments de la journée d'utilisation du véhicule steps.



(b) Pourcentage du temps de conduite selon les jours de la semaine steps.

Figure 3.4: Les sabitudes de conduites des assurés.

3.3.4 Nombre de sinistres

Le tableau (3.2) nous donne un aperçu sur le nombre de sinistres, la distribution des sinistres, la distribution du kilométrage parcouru par nombre de sinistres ainsi

(a) Ontario (b) Région de Toronto (c) Région Ottawa

Figure 3.5: Kilométrage conduit RTA (vert clair étant un kilométrage plus élevé).

que l'exposition en jours par nombre de sinistres. On peut y lire par exemple, que 4.28% de nos assurés ont eu un seul sinistre (deuxième ligne), la distance parcourue par ces derniers représente environ 6.4% de la distance totale parcourue par tous nos assurés, et leur exposition en nombre de jours représente environ 5.52% du nombre de jours total d'exposition de tous nos assurés. On voit bien que la distribution l'exposition (en distance parcourue et en jour) des assurés reflète la distribution du nombre de sinistres.

Nb accidents	**Nb policyholder**	**accidents freq**	**distance (km)**	**expo days**
0	89504	95.54%	93.26%	94.22%
1	4013	4.28%	6.40%	5.52%
2	160	0.17%	0.32%	0.25%
3	5	0.01%	0.01%	0.01%

Tableau 3.2: La fréquence de l'exposition des assurés (en distance parcourue et en jour) par nombre de sinistres.

Nous avons également tracé la distribution des sinistres sur une carte géographique afin d'illustrer dans quelle région se trouvent les assurés ayant eus au moins un,

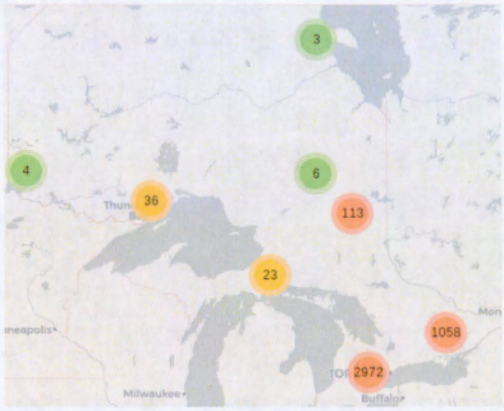
deux ou trois sinistres dans les figures (3.6a), (3.6b) et (3.6c) respectivement. On peut remarquer que la fréquence des sinistres est plus élevée dans les régions métropolitaines que dans les régions rurales.

Afin d'analyser la corrélation entre les variables explicatives, nous avons sélectionné les variables quantitatives les plus utilisées dans la tarification classique en assurance automobile ainsi que d'autres variables télématiques. Nous avons tracé la corrélation de *Pearson* entre les variables à la figure (3.7) où l'on peut remarquer une forte corrélation positive entre certaines variables. Par exemple, le nombre d'années sans réclamation (`RA_YRS_LICENSED`) avec le nombre d'années d'expérience de conduite (`RA_YRS_CLAIMS_FREE`).

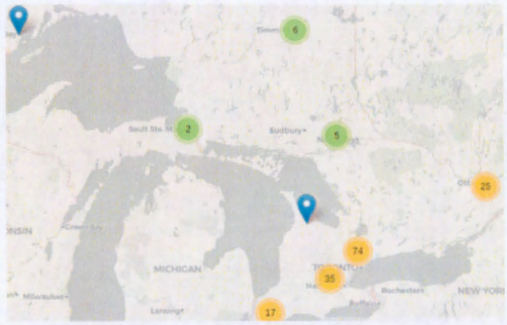
Cet exercice d'analyse statistique que nous avons effectuée nous permet d'avoir un bon aperçu global des données avant même d'effectuer d'ajuster des modèles prédictifs. Cela peut être révélateur de certaines intuitions qui nous guident dans la conception de nos modèles.

3.4 Modèle GBMP

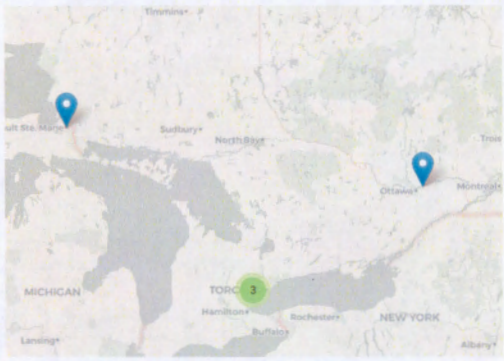
Tout d'abord, nous devons rappeler que le but de notre exercice est de modéliser la fréquence des sinistres avec un modèle GBMP. Pour ce faire, comme nous l'avons expliqué dans le chapitre précédent, nous devons entraîner notre modèle sur des données d'entraînement qui contiennent notre variable réponse observée y afin de prédire la valeur de la fréquence estimé $\hat{y}$. Dans notre base de données, nous avons choisi la variable `RA_NB_CLAIM` qui est tout simplement le nombre de sinistres (tous types confondus) qu'un assuré a eu pendant la période de couverture de sa police d'assurance.



(a) Un sinistre ou plus.



(b) Deux sinistres ou plus.



(c) Trois sinistres ou plus.

Figure 3.6: Distribution des sinistres par région.

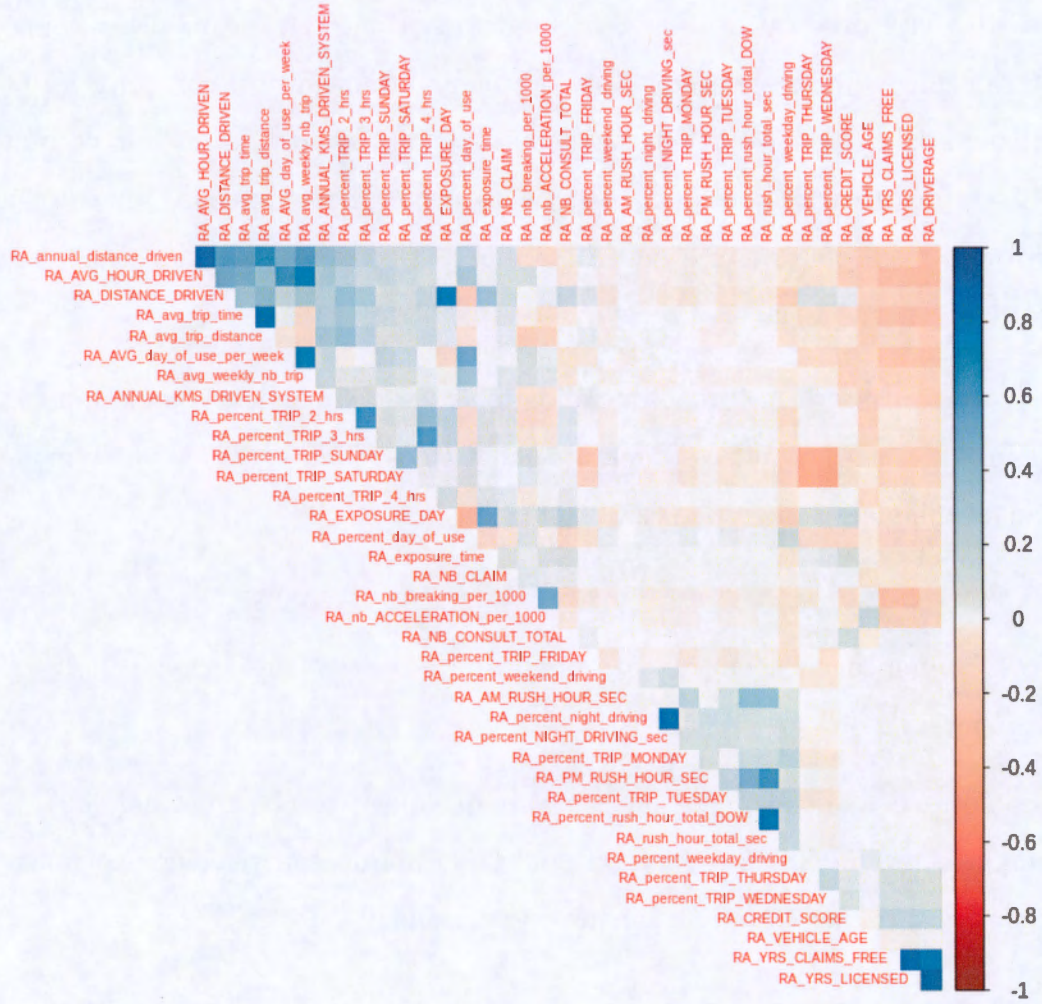


Figure 3.7: La corrélation de *Pearson* entre les variables explicatives.

3.4.1 Partitionnement des données

Un des grands avantages du GBM est qu'aucun prétraitement des données n'est requis avant l'ajustement du modèle (Friedman *et al.*, 2001, p. 352). Nous avons conservé les données qui nous ont été remises par l'assureur tel quel, nous n'avons pas eu besoin de supprimer les données manquantes ni corriger les données aber-

rantes. Nous avons remarqué une variable indiquant le partitionnement des données, 70% des observations ont été consacrées à entraîner le modèle (données d'entraînement). Le reste a été écarté et consacré seulement à comparer les prédictions produites par le modèle vs les observations de notre variable réponse tel qu'il sera décrit dans la section (4.4). Nous avons choisi de conserver ce même partitionnement pour d'une éventuelle comparaison des résultats de notre modèle avec ceux de l'assureur.

Nous avons utilisé la méthode de validation croisée 5 parties, comme illustré à la figure (2.3) sur les données d'entraînement (70%) pour entraîner notre modèle comme expliquées à la section (2.2).

3.4.2 Traitement des données catégorielles

Dans notre base de données, nous avons quelques variables catégorielles, par exemple le sexe de l'assuré, son état civil ou la marque du véhicule. Nous avons encodé ces variables en format binaire comme suit ;

$$\mathbf{X}_{\text{sexe}} = \begin{cases} 0 & \text{si l'assuré est un homme,} \\ 1 & \text{si l'assurée est une femme.} \end{cases} \tag{3.1}$$

Nous avons utilisé la librairie *One Hote Encoder* de (Pedregosa *et al.*, 2011) afin de transformer ses variables en format binaire. Il faut noter que certaines bibliothèques de calcul GBM peuvent détecter et supporter les données catégorielles sans que le modélisateur le précise. Toutefois, nous avons pris la précaution de bien faire la transformation avant d'ajuster notre modèle.

3.4.3 Modèle GBMP et sa calibration

Nous avons vu que le GBM risque de converger exactement si l'on ne détermine pas des paramètres de contrôle pour éviter le surapprentissage. Nous avons également vu que ces paramètres peuvent prendre plusieurs valeurs, et que très récemment, plusieurs méthodes d'optimisation ont été développées afin de trouver la valeur la plus optimale de ces paramètres, autrement dit, les paramètres qui minimisent l'erreur de prédiction.

Pour notre problème, nous avons entraîné notre modèle sur les données d'entraînement avec la technique de validation croisée 5-parties en précisant deux paramètres de contrôle, soient un taux d'apprentissage τ et un taux de sous-échantillonnage η. Afin d'obtenir les valeurs optimales de ces paramètres, nous avons construit un espace de paramètres restreints ($\tau = 0.005, 0.01, \ldots, 0.1$ ainsi que $\eta = 0.40, 0.41, \ldots, 0.50$), dans lequel nous avons forcé la recherche de la combinaison la plus optimale en termes d'erreur de prédiction. Donc à chacune des cinq itérations de la validation croisée, notre modèle est entraîné sur 70% des données et valide calcule l'erreur produite sur la prédiction faite sur le 30% des données restantes. Nous avons trouvé les deux paramètres les plus optimaux étant un taux d'apprentissage $\tau = 0.05$ ainsi qu'un taux de sous-échantillonnage de l'ordre de $\eta = 50\%$.

Pour ce qui du nombre d'itérations (nombre d'arbres), nous avons fixé $T = 300$ dans l'algorithme (4) et nous avons comparé les la moyenne des résidus de déviance à chaque itération entre la partie d'entraînement des et la partie test des données (de la validation croisée). Dans la figure, nous pouvons constater qu'un modèle avec 183 arbres a été suffisant afin de minimiser les résidus de déviance.

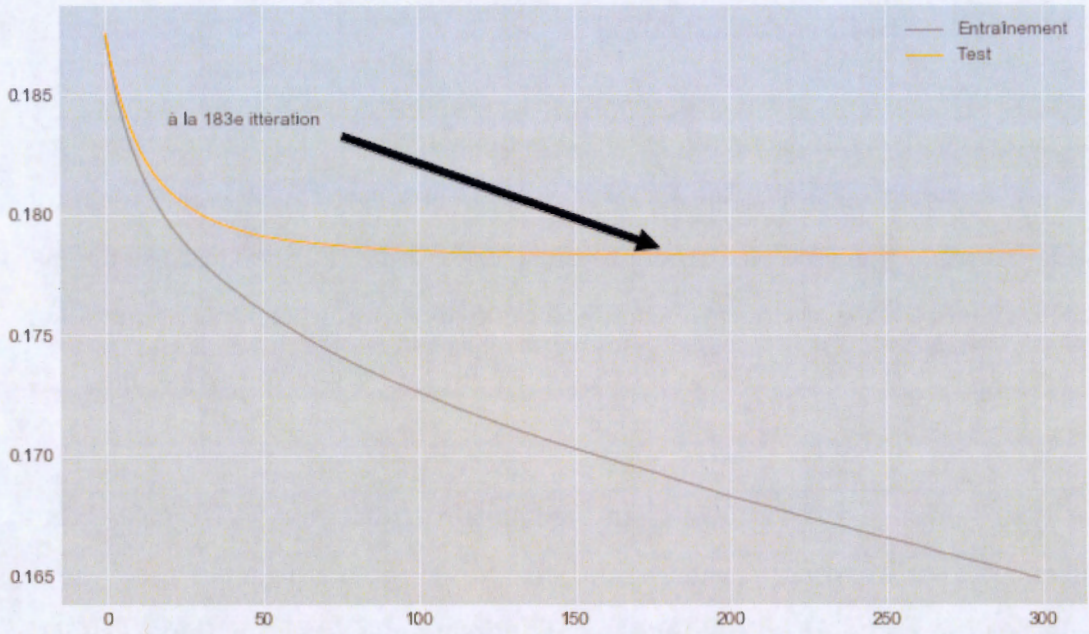


Figure 3.8: Nombre d'itérations (arbres) nécessaires pour converger notre modèle GBMP.

3.4.4 Importance des variables

Comme nous l'avons expliqué à la section (2.6.1), l'importance d'une variable est l'augmentation de l'erreur de prédiction du modèle après avoir permuté les valeurs de la variable, ce qui permet de rompre la relation entre la variable explicative et le résultat réel. Pour chacune des variables dans notre base de données (une variable à la fois), nous avons permuté aléatoirement les observations de cette variable en gardons les observations des autres variables inchangées, nous avons ajusté notre modèle et nous avons calculé l'erreur de prédiction à chacune des cinq itérations (dans validation croisée) comme décrit dans l'algorithme (5).

Avec notre modèle GBMP, nous avons sélectionné les trente variables les plus importantes telles qu'illustrées à la figure (3.9). Nous constatons que la distance

parcourue est la variable la plus sensible aux changements des valeurs de nos observations des données d'entraînement. Ci-dessous, les sept variables les plus importantes que nous avons sélectionnées ;

1. `RA_DISTANCE_DRIVEN` : la distance parcourue
2. `RA_EXPOSURE_DAY` : l'exposition en jour
3. `RA_VEHICLE_AGE` : l'âge du véhicule
4. `RA_percent_day_of_use` : le pourcentage d'utilisation du véhicule durant le jour
5. `RA_AVG_day_of_use_per_week` : le nombre d'utilisations moyen par semaine du véhicule durant le jour
6. `RA_AVG_HOUR_DRIVEN` : nombre d'heures moyen conduit
7. `RA_avg_weekly_nb_trip` : nombre de trajets moyen par semaine.

Dans le chapitre qui suit, nous reproduirons un autre modèle GBM avec les mêmes paramètres, mais cette fois, notre modèle sera basé sur seulement ces sept variables. Nous ferons l'interprétation de ce modèle avec les techniques expliquées dans les sections (2.6.2) et (2.6.3) respectivement. Nous comparerons ensuite sa performance sur le nombre de sinistres prédits avec un modèle GLM classique.

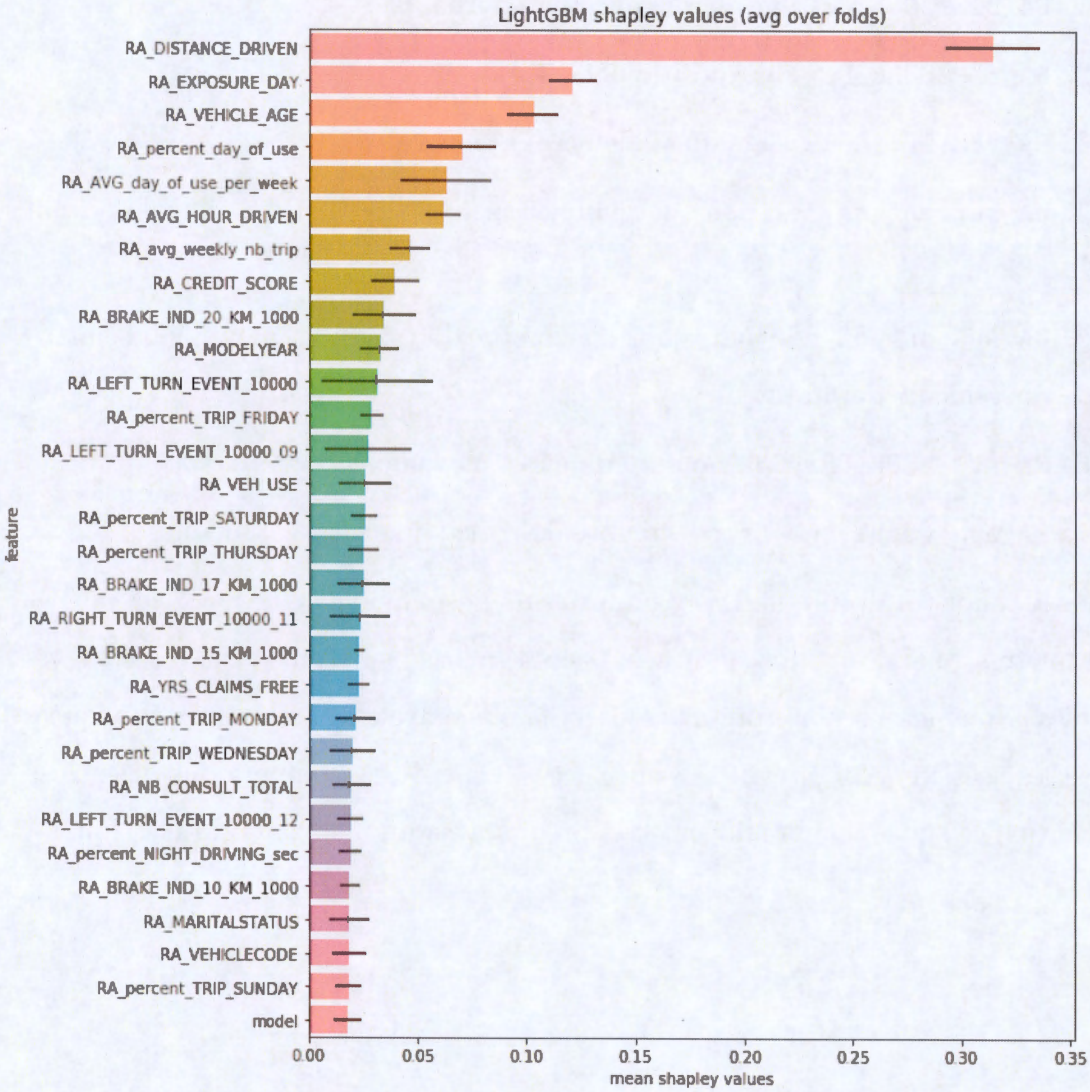


Figure 3.9: Importance de 30 des 121 variables par le modèle GBMP.

CHAPITRE IV

INTERPRÉTATION DES RÉSULTATS

Lorsque nous créerons des modèles mathématiques, il est important de faire preuve de rigueur avant de tirer rapidement des conclusions, c'est pour cette raison que nous consacrons toute une section à l'interprétation des résultats obtenus avec notre modèle GBMP. On peut définir l'interprétation comme la mesure dans laquelle nous pouvons comprendre la cause d'une décision. Plus l'interprétation d'un modèle d'apprentissage automatique est élevée, plus il nous sera facile de donner une signification aux décisions à prendre.

4.1 Dépendance partielle

Comme nous le savons déjà, la dépendance partielle implique la démonstration de l'effet d'une variable explicative sur la variable réponse modélisée, après avoir marginalisé toutes les autres variables explicatives (section 2.6.2). Nous allons analyser l'effet de chacune des sept variables importantes que nous a révélé la technique de l'importance des variables dans la section précédente.

D'abord, regardons de plus près quel effet aura la variation (augmentation ou diminution) de la variable `RA_DISTANCE_DRIVEN` sur la fréquence de sinistre pour chaque assuré. Mais avant, comme nous avons des données télématiques, les dis-

tances parcourues sont différentes pour chaque assuré, car nous avons la distance **exacte** parcourue durant toute la période d'observation de ce dernier. Il aurait été difficile d'illustrer de tels résultats, où l'on veut montrer que si un assuré qui a parcouru réellement 13556 km avait parcouru 21299 km exactement, aurait augment sa fréquence de sinistre de $x\%$. Donc au lieu de faire des variations sur le kilométrage parcouru exact, nous avons alors décidé de segmenter toutes les observations de variable `RA_DISTANCE_DRIVEN` à l'intérieur du vecteur $[0, 1000, 2000, \ldots, 20000, 22000, 24000, 30000, 80000]$. Donc un assuré qui a parcouru 8000 km se trouve dans le même groupe qu'un autre assuré ayant parcouru 8999 km durant la période observée.

Cette segmentation est illustrée à la figure (4.1) où chaque bâtonnet vert de l'histogramme représente le nombre d'assurés dans chaque groupe. Par exemple, 3080 assurés ont parcouru entre $[8000, 9000)$ km. Alors que la ligne bleue est la fréquence moyenne pour chaque groupe, où si l'on veut le nombre de sinistres moyen observé pour chaque groupe d'assuré. Par exemple, la fréquence moyenne est de 0.053 pour les assurés ayant parcouru entre $[8000, 9000)$ km. On peut remarquer d'ailleurs sur cette figure que nous nous déplaçons vers la droite, donc plus le kilométrage parcouru est élevé, plus la fréquence moyenne est élevée.

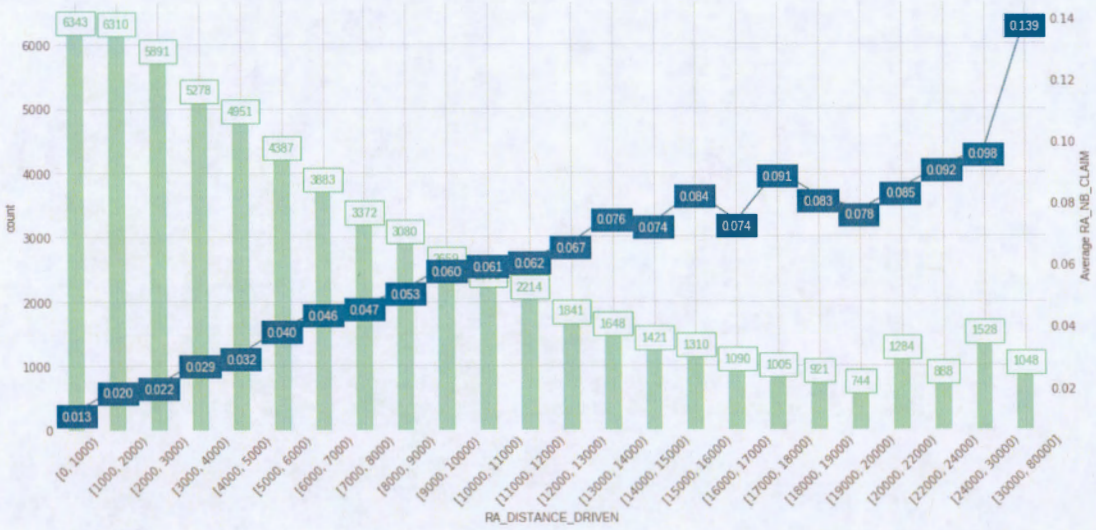
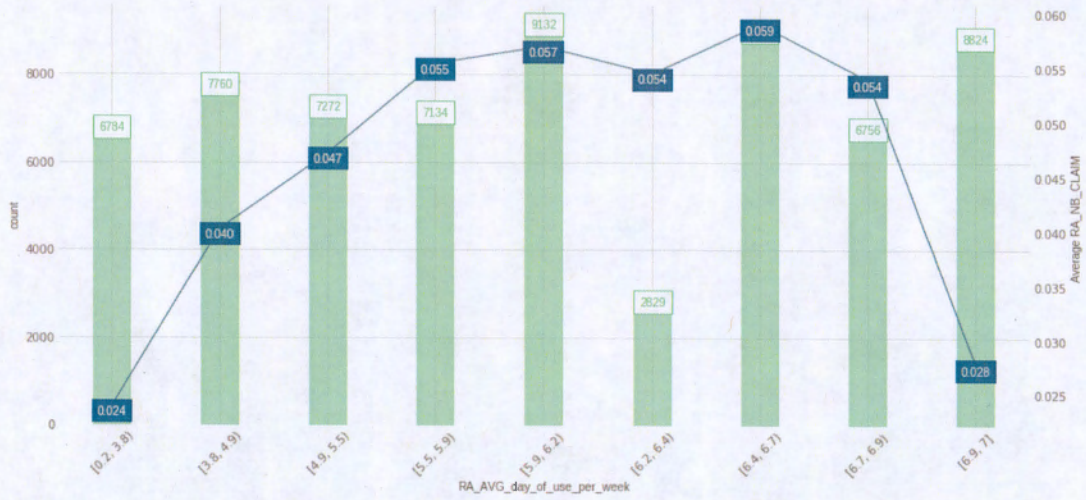


Figure 4.1: Description de la variable `RA_DISTANCE_DRIVEN`. Nous avons fait une première tentative de segmentation arbitraire où l'on illustre la fréquence moyenne des sinistres par segment de kilométrage parcouru.

(a) L'exposition en jour.

(c) Le pourcentage d'utilisation du véhicule durant le jour.



(d) Le nombre d'utilisations moyen par semaine du véhicule durant le jour.

(e) Nombre d'heures moyen conduit.

(f) Nombre de trajets moyen par semaine.

Figure 4.2: Fréquence moyenne des sinistres par segment $(10, 20, \ldots 100^e)$ pour chacune des variables sélectionnées.

Nous avons refait le même exercice de segmentation pour les six autres variables. Cependant, cette fois nous avons segmenté chacune de ces variables par le $10, 20, \ldots 100^{e}$ centile des valeurs observées dans notre jeu de données d'entraînement. Ces segmentations sont illustrées à la figure (4.2).

Maintenant que nous avons segmenté nos sept variables, nous sommes enfin prêts à voir comment les graphiques de la dépendance partielle séparent l'effet de chaque variable. Il suffit de prendre une variable à la fois, et remplacer toutes valeurs observées de cette variable par les valeurs des segmentations que nous avons effectuées. Par exemple, lorsqu'on s'intéresse à la dépendance partielle de la variable `RA_DISTANCE_DRIVEN`, nous remplaçons ces valeurs pour chaque assuré par $[0, 1000, 2000, \ldots, 20000, 22000, 24000, 30000, 80000]$, une valeur à la fois tout en gardant fixes les valeurs des autres caractéristiques de l'assuré. Les tableaux (4.1a, 4.1b et 4.1c) illustrent ce processus pour les trois premiers segments de la variable `RA_DISTANCE_DRIVEN`.

Ensuite, avec le modèle GBMP entraîné sur les données originales, nous calculons la fréquence des sinistres prédite pour chaque assuré lorsque son kilométrage parcouru a été remplacé par $[0, 1000, 2000, \ldots, 20000, 22000, 24000, 30000, 80000]$ km. Cette valeur prédite est un point sur les lignes fines de la figure (4.3) où l'on a représenté la prédiction de seulement 1000 assurés afin d'alléger le graphique. Finalement, la dépendance partielle de la variable `RA_DISTANCE_DRIVEN` est représentée par la ligne jaune de la même figure, où chaque point de cette ligne est la moyenne de la prédiction de la fréquence des sinistres pour tous les assurés ayant parcouru $[0, 1000, 2000, \ldots, 20000, 22000, 24000, 30000, 80000]$ km. Autrement dit, chaque point sur la ligne jaune est une moyenne des points sur les lignes fine pour chaque segment de distance parcouru des assurés.

$\mathbf{X}_{dist}$	$\mathbf{X}_2$	$\mathbf{X}_3$	...	$\hat{\mathbf{y}}$
1000	2705.9	152	...	$\hat{y}_1$
1000	3180.6	223	...	$\hat{y}_2$
1000	31559.2	32	...	$\hat{y}_3$
1000	2870.2	73	...	$\hat{y}_4$

(a) Première itération

$\mathbf{X}_{dist}$	$\mathbf{X}_2$	$\mathbf{X}_3$	...	$\hat{\mathbf{y}}$
2000	2705.9	152	...	$\hat{y}_1$
2000	3180.6	223	...	$\hat{y}_2$
2000	31559.2	32	...	$\hat{y}_3$
2000	2870.2	73	...	$\hat{y}_4$

(b) Deuxième itération

$\mathbf{X}_{dist}$	$\mathbf{X}_2$	$\mathbf{X}_3$	...	$\hat{\mathbf{y}}$
30000	2705.9	152	...	$\hat{y}_1$
30000	3180.6	223	...	$\hat{y}_2$
30000	31559.2	32	...	$\hat{y}_3$
30000	2870.2	73	...	$\hat{y}_4$

(c) Troisième itération

Tableau 4.1: Exemple des trois premières itérations du processus de dépendance partielle de la variable `RA_DISTANCE_DRIVEN` sur la fréquence des sinistres prédite par GBMP.

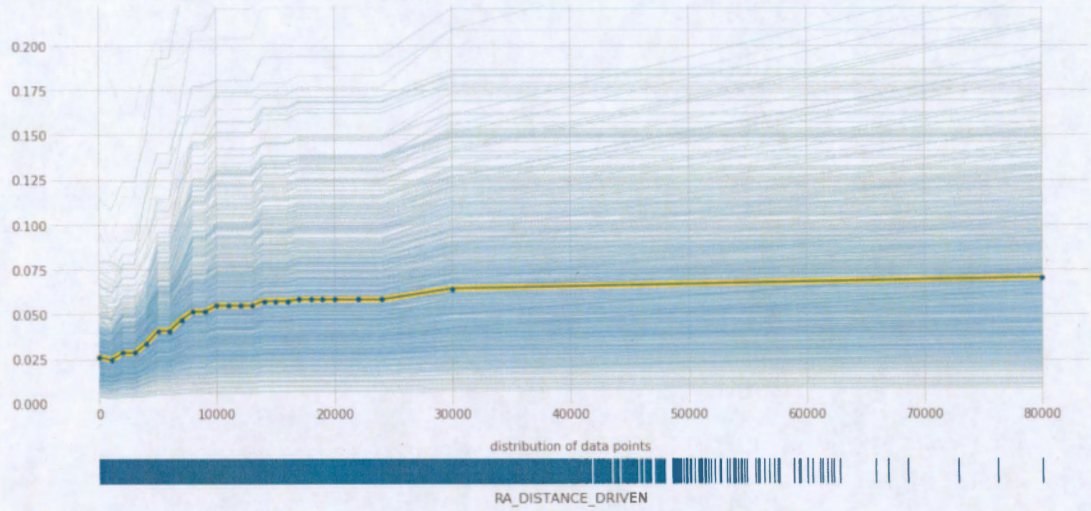


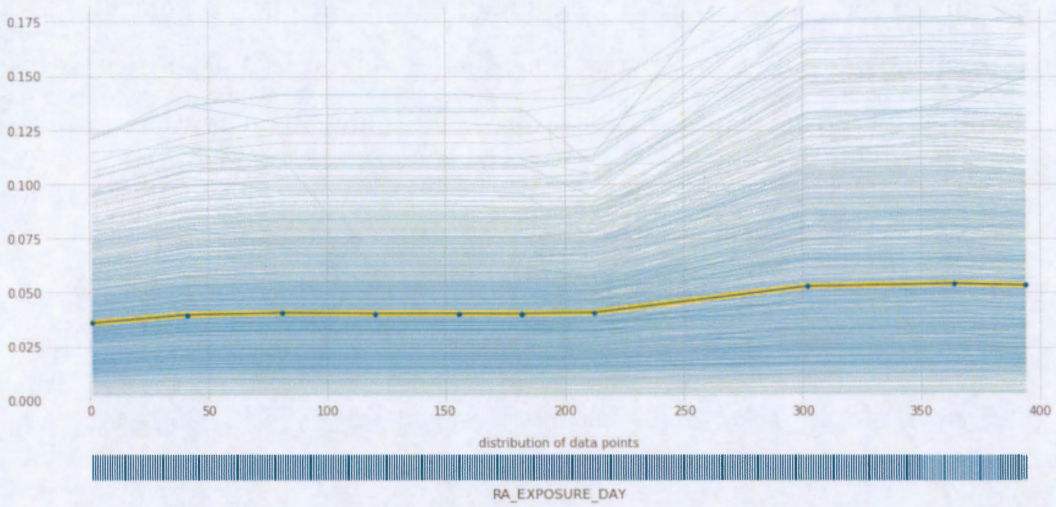
Figure 4.3: Diagramme de la dépendance partielle pour la variable `RA_DISTANCE_DRIVEN`.

Dans la figure (4.4), nous avons tracé le diagramme de dépendance partielle en faisant varier cette fois les six autres variables les plus importantes présentées à la figure (3.9).

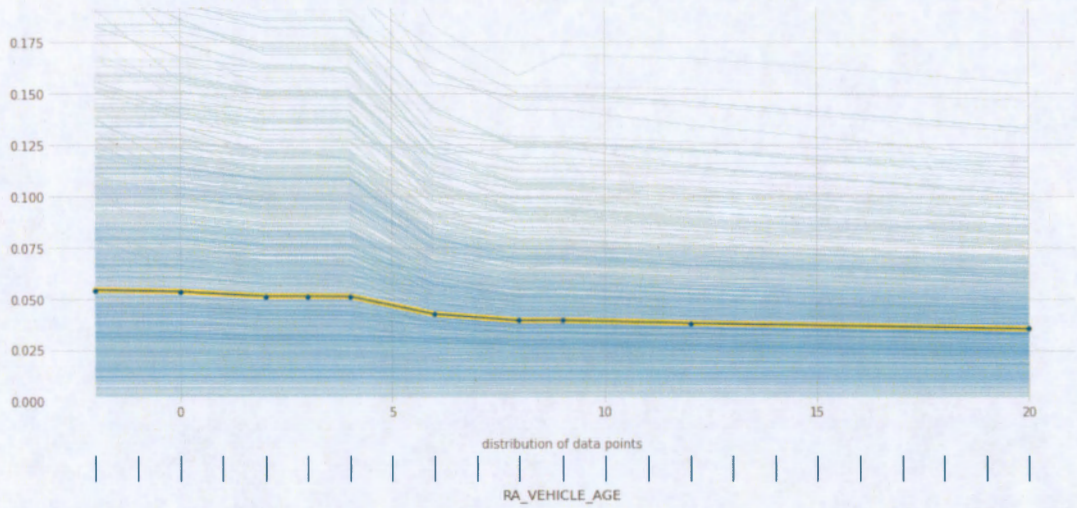
Dans ces graphiques, nous pouvons remarquer que, plus un assuré est exposé en temps ou en kilométrage parcouru, plus sa fréquence de réclamation prédite sera élevée, à l'exception des deux variables `RA_percent_day_of_use` ainsi que `RA_AVG_day_of_use_per_week`, dans les figures (4.4c) et (4.4d) respectivement. Rappelons que la première est le pourcentage d'utilisation du véhicule durant le jour, alors que la deuxième est le nombre d'utilisations moyen par semaine du véhicule durant le jour. De ces deux diagrammes, nous pouvons constater que, plus un assuré conduit durant le jour, moins il risque de faire une réclamation, cela semble suivre l'intuition que nous avons en général sur la conduite durant le

jour versus la conduite durant la nuit.

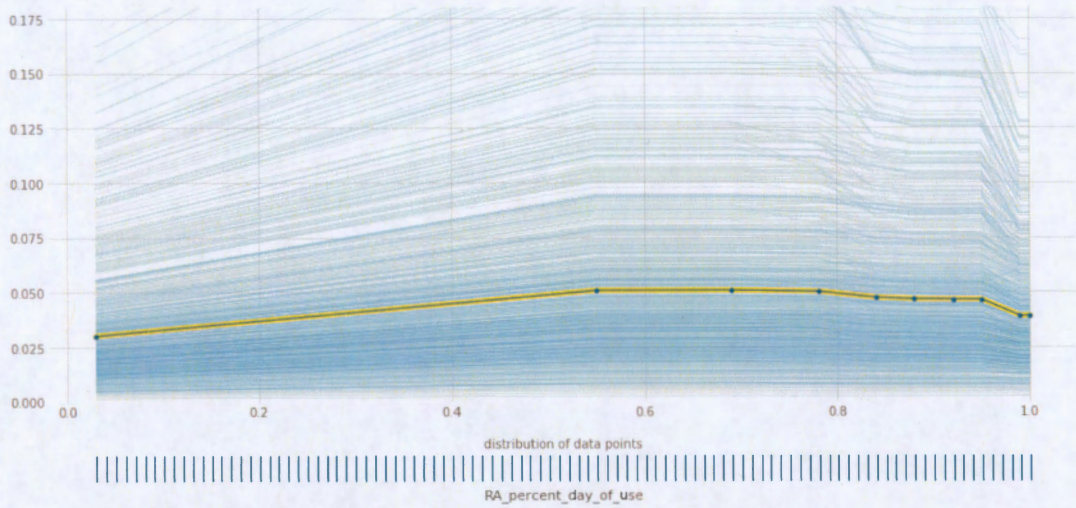
Nous apercevons sur la figure (4.4b) que la fréquence prédite de réclamation baisse avec l'âge du véhicule. Cela pourrait s'expliquer par le fait que les véhicules les plus âgés sont moins coûteux à réparer, donc l'assuré ne déclare pas son sinistre et préfère absorber le coût de réparations afin d'éviter une augmentation du prix de sa prime d'assurance.



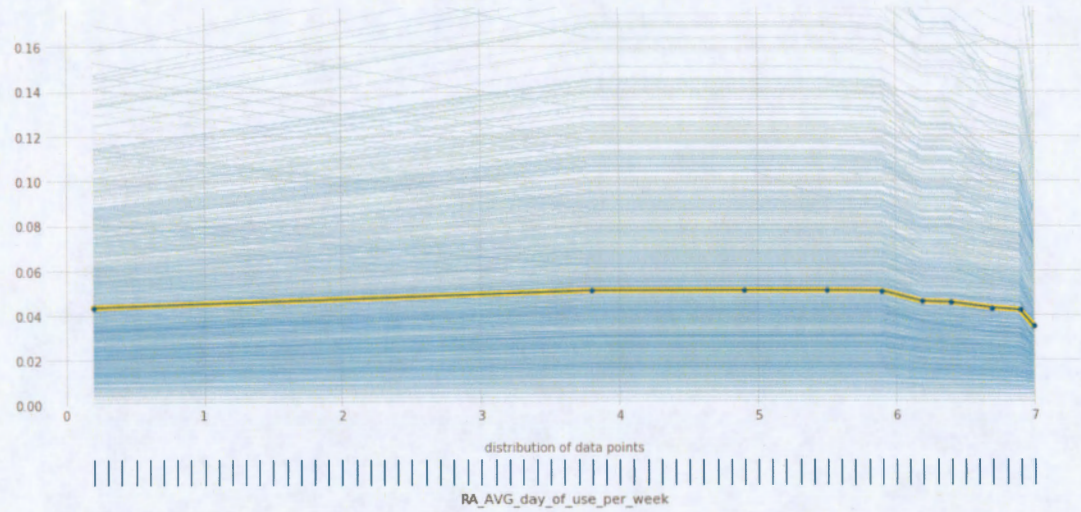
(a) L'exposition en jour.



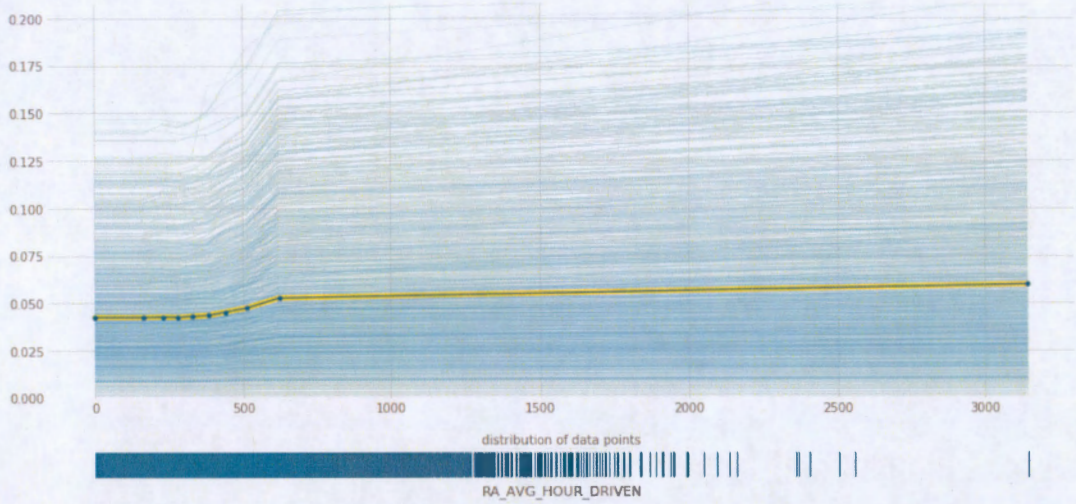
(b) L'âge du véhicule.



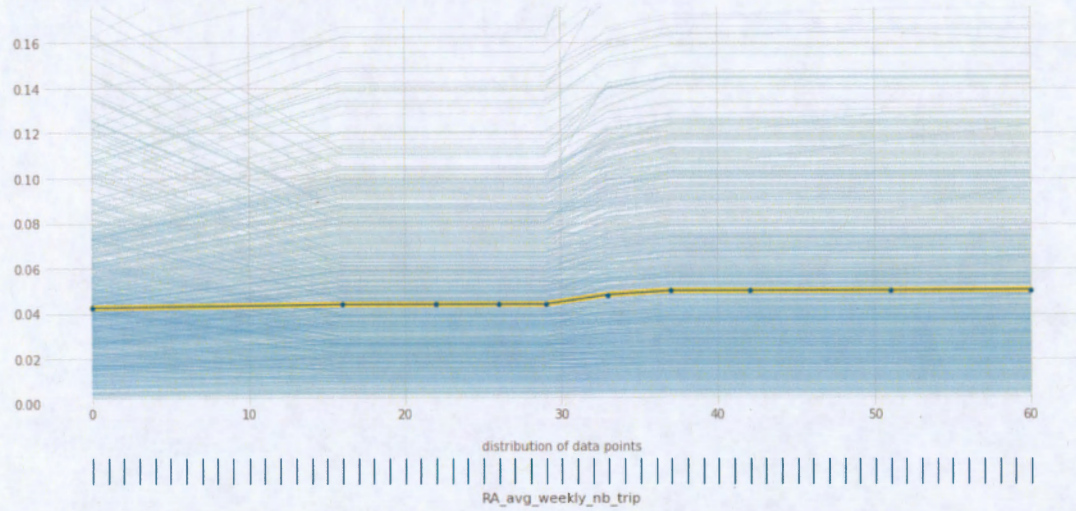
(c) Le pourcentage d'utilisation du véhicule durant le jour.



(d) Le nombre d'utilisations moyen par semaine du véhicule durant le jour.



(e) Nombre d'heures moyen conduit.



(f) Nombre de trajets moyen par semaine.

Figure 4.4: Diagramme de dépendance partielle pour chacune des variables sélectionnées.

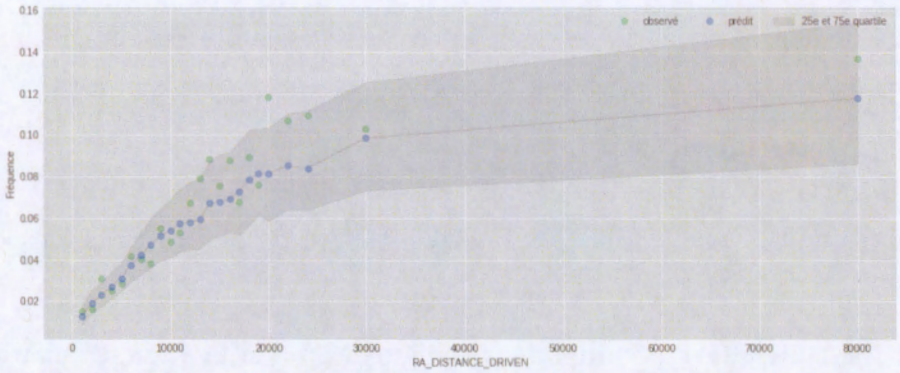
Nous venons de voir que les diagrammes de dépendance partielle peuvent nous aider à interpréter des modèles non linéaires très complexes, en plus ils nous aident à confirmer ou infirmer nos intuitions comme nous l'avons vu avec l'exposition en kilométrage ou en temps qui ont tendance à augmenter la fréquence de réclamation prédite. Toutefois, il faut prendre un peu de recul face à de tels résultats; nous avons vu que nous avons pris par exemple la variable `RA_DISTANCE_DRIVEN` et nous avons fait varier toutes les possibilités de distance tout en conservant les autres caractéristiques originales des assurées, cela peut poser un problème pour certaines observations. En effet, pensons seulement à un assuré dont l'exposition en temps est d'une durée d'une journée, alors qu'avec la technique de dépendance partielle nous avons fait varier son kilométrage 0 km jusqu'à 80 000 km. Or, il est irréaliste qu'un assuré puisse parcourir de telles distances en si peu de temps d'exposition.

4.1.1 Prédiction sur les données test

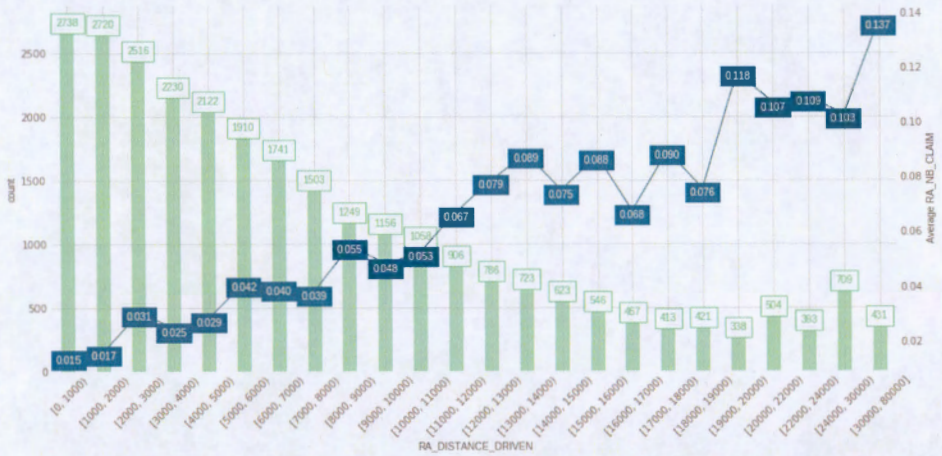
Maintenant que nous avons démontré l'effet de chacune des variables explicatives sur la fréquence de sinistre prédite par notre modèle GBMP entraîné sur les données d'entraînement (figures 4.3 et 4.4). Vérifions si nous obtiendrons les mêmes effets de dépendance partielle sur la partie test des données que, rappelons-le, notre modèle n'a jamais vues.

Tout d'abord, nous réalisons une analyse descriptive de la fréquence des réclamations en segmentant les sept variables auxquelles nous nous intéressons. La variable `RA_DISTANCE_DRIVEN` est illustrée à la figure (4.5b) et les six autres variables présentées à la figure (B.2) où les courbe blues présentent la fréquence observée lorsque les observations sont groupées par segment (bâtonnets verts). Maintenant, traçons une courbe de la fréquence **prédite** moyenne pour chacun de ces segments

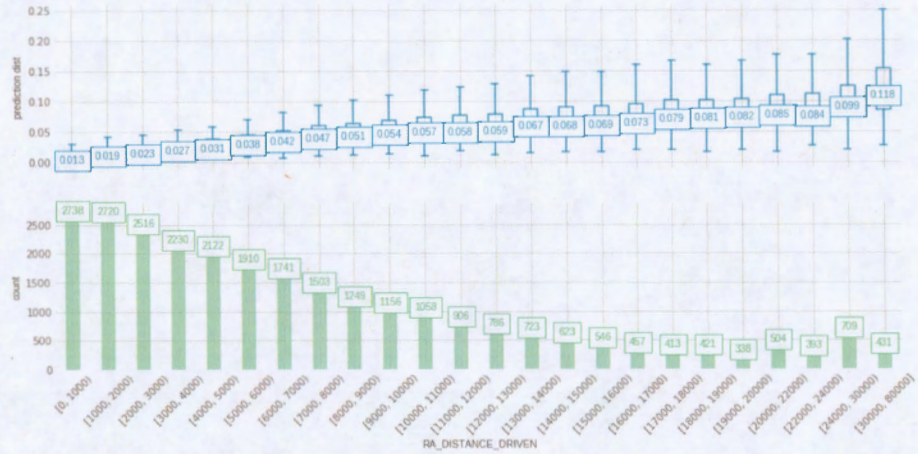
en faisant varier la variable explicative. Nous apercevons le résultat de ces prédictions aux figures (4.5c et B.4) où les valeurs des prédictions de la fréquence pour chaque segment sont illustrées par des boîtes à moustaches. Nous avons ensuite juxtaposé ces deux figures pour comparer la fréquence de sinistre prédite avec celle observée dans la figure (4.5a) pour la variable `RA_DISTANCE_DRIVEN` ainsi qu'aux figures (4.6) pour les six autres variables. Les points verts sont les fréquences observées pour chaque segment à l'intérieur de l'écart interquartile (à différence entre le troisième et le premier quartile). Les points bleus sont les fréquences prédites. Nous pouvons y remarquer que nos prédictions sont très proches de celles observées pour la plupart des segments de kilométrage parcouru. Rappelons-nous que nous avons effectué une segmentation quasi arbitraire, nous verrons dans les prochaines sections qu'il est possible d'utiliser d'autres techniques pour nous aider à mieux segmenter nos variables.



(a) Comparaison de la fréquence des sinistres prédite et observée par la technique de dépendance partielle.

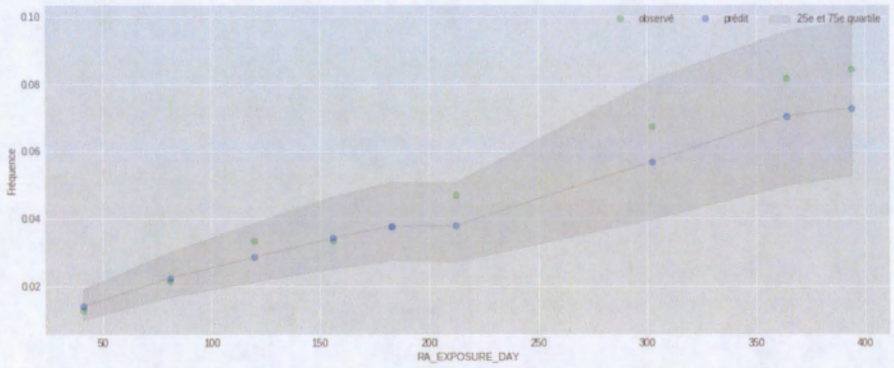


(b) Fréquence des sinistres observée.

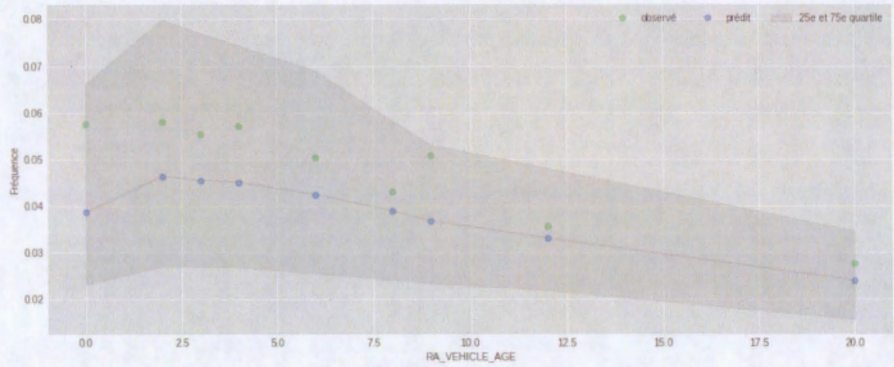


(c) Fréquence des sinistres prédite.

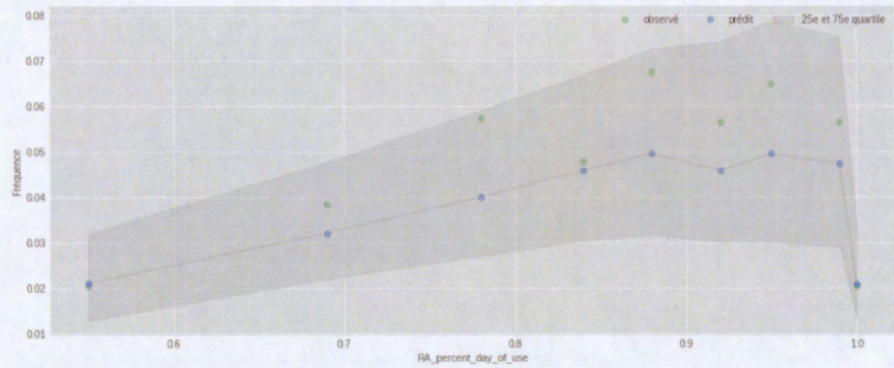
Figure 4.5: Fréquence des sinistres prédite versus observée par la technique de la fréquence partielle pour chaque segment de la variable `RA_DISTANCE_DRIVEN`.



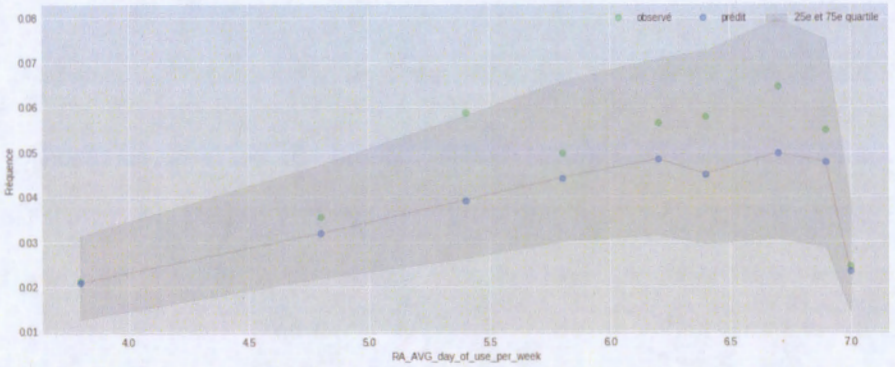
(a) L'exposition en jour.



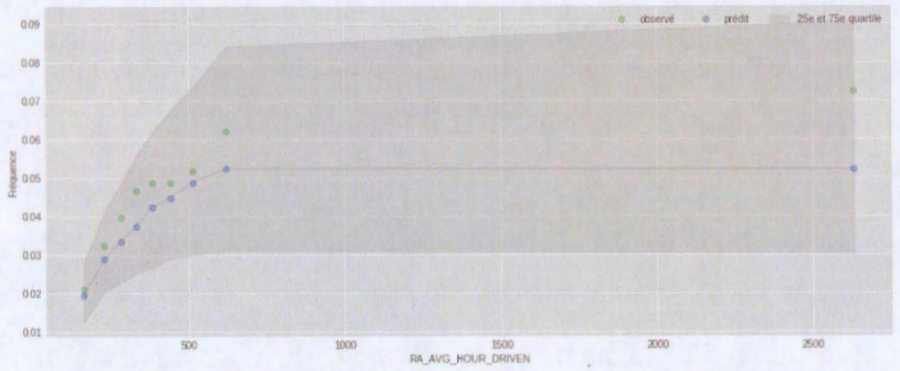
(b) L'âge du véhicule.



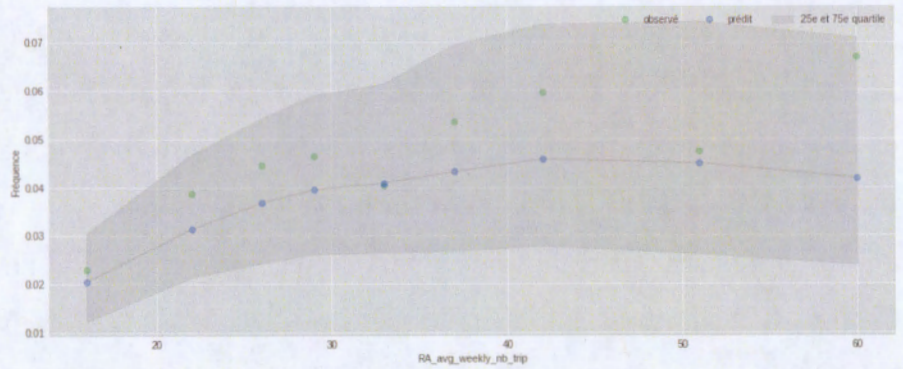
(c) Le pourcentage d'utilisation du véhicule durant le jour.



(d) Le nombre d'utilisations moyen par semaine du véhicule durant le jour.



(e) Nombre d'heures moyen conduit.



(f) Nombre de trajets moyen par semaine.

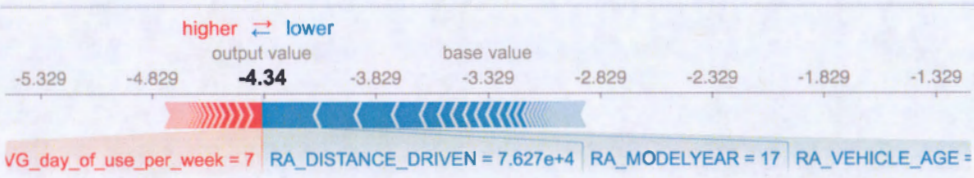
Figure 4.6: Comparaison de la fréquence des sinistres prédite et observée par la technique de dépendance partielle pour chacune des variables sélectionnées.

4.2 Les valeurs SHAP

Comme nous l'avons expliqué dans la section (2.6.3), les valeurs de Shapley sont des valeurs de la contribution marginale moyenne d'une valeur de caractéristique sur l'ensemble de la prédiction. Afin de calculer ces valeurs sur nos données télématiques, nous avons utilisé la librairie SHAP *(SHapley Additive exPlanations)* (Lundberg, 2019) de l'auteur des articles (Lundberg et Lee, 2017a) et (Lundberg et Lee, 2017b).

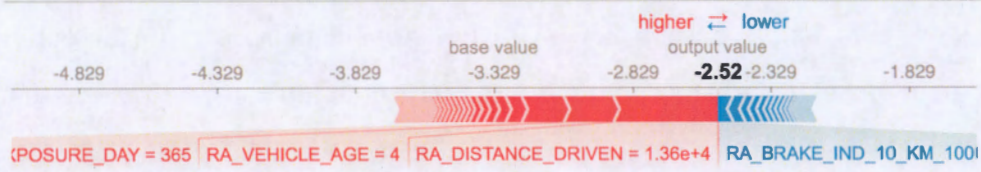
Pour mieux illustrer ces valeurs, nous avons choisi trois assurés dans notre base de données afin de voir dans quelle mesure, chacune leurs caractéristiques contribuent, positivement ou négativement, à la fréquence de réclamation prédite. Le premier ayant parcouru beaucoup de kilométrage (76 000 km), le second n'a parcouru que 0.1 km, et finalement le troisième assuré a parcouru une distance qu'un conducteur moyen parcourt habituellement, soit environ 14 000 km. Il est important de noter ici que ce kilométrage est la réelle distance observée durant la période d'exposition.

Nous avons calculé les valeurs SHAP pour ces trois assurés tels que présentés au tableau (4.2). Ces mêmes valeurs sont illustrées dans les graphiques (4.7a), (4.7b) et (4.7c) respectivement. Nous remarquons clairement que le haut kilométrage contribue négativement sur la fréquence prédite pour cet assuré (vers la gauche dans la figure (4.7a), donc augmente la fréquence prédite. Toutefois, cette valeur de contribution marginale tient compte de toutes les caractéristiques de l'assuré en question, c'est d'ailleurs très clair dans les figures (4.7b) et (4.7c), où malgré la différence du kilométrage parcouru, les deux assurés ont une contribution marginale de la distance à peu près équivalente, et ce, même si la valeur de cette caractéristique très différente pour ces deux assurés.

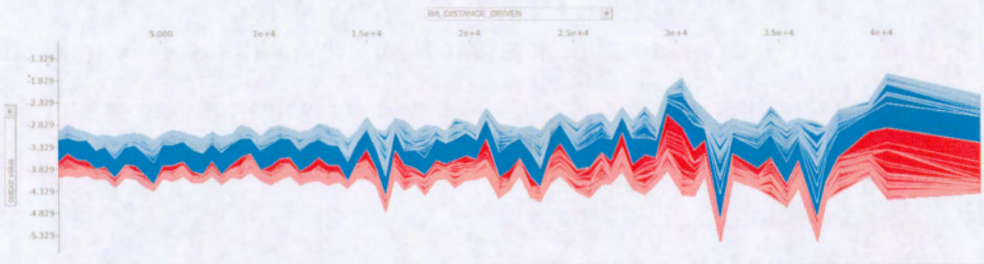


(a) Valeurs SHAP de l'assuré #1 ayant parcouru 76 271.8 km.

(b) Valeurs SHAP de l'assuré #2 ayant parcouru 0.1 km.



(c) Valeurs SHAP de l'assuré #3 ayant parcouru 13 597.7 km.



(d) Juxtaposition verticale des valeurs SHAP des 2000 [a] premiers assurés dans notre base de données.

a. Nous avons tracé les valeurs SHAP des 2000 premières observations afin d'alléger le graphique

	Assuré #1	SHAP Assuré #1	Assuré #2	SHAP Assuré #2	Assuré #3	SHAP Assuré #3
RA_DISTANCE_DRIVEN	76271.8	-0.261214	0.1	0.452481	13597.7	0.37882
RA_EXPOSURE_DAY	333	-0.057105	5	0.179411	365	0.15163
RA_VEHICLE_AGE	2	-0.167097	3	0.188622	4	0.17815
RA_percent_day_of_use	1	0.045026	0.4	0.015716	0.58	0.017949
RA_AVG_day_of_use_per_week	7	0.071812	2.8	-0.017279	4.1	-0.005084
RA_AVG_HOUR_DRIVEN	1391	-0.04968	18	-0.045674	260	-0.012379
RA_avg_weekly_nb_trip	60	-0.034987	4	-0.0333	26	-0.043829
...	...	...	...	...	...	...

Tableau 4.2: Exemple de valeurs SHAP pour trois assurés tirés des données d'entraînement.

L'importance des variables nous indique quelles caractéristiques sont les plus importantes pour l'ensemble des assurés, mais cette approche unique ne s'applique pas toujours sur chaque observation prise individuellement, car un facteur important pour un assuré donné peut ne pas l'être pour un autre. Prenons seulement l'exemple de deux assuré qui parcourt de longues distances annuellement, un conducteur habitant en banlieue qui utilise sont véhicule quotidiennement pour aller au travail le matin et retourner chez lui le soir, et un autre conducteur qui utilise son véhicule à des fins d'autopartage en milieu urbain, donc plus exposé aux accidents et risques divers. Donc, en regardant seulement les tendances globales, ces variations individuelles peuvent se perdre, avec seulement les dénominateurs les plus communs qui restent.

En plus diagramme typique à barres d'importance de variables (comme illustré à la figure (3.9)), où l'influence obtenue de chaque variable sur la prédiction est la moyenne des cinq l'influence de la validation croisée. Nous avons calculé les valeurs SHAP sur l'ensemble de la partie d'entraînement des données, nous avons obtenu alors des valeurs SHAP de chacune des variables pour chaque assuré comme au tableau (4.3) où l'on aperçoit quatre variables ainsi que les cinq premières observations.

RA_EXPOSURE_DAY	RA_ANNUAL_KMS_DRIVEN_SYSTEM	RA_DRIVERAGE	...	credit_score_incom_FSA
-0.033153	0.010829	0.345738	...	-0.000731
-0.067796	0.006381	-0.005229	...	-0.001645
0.245181	0.00744	-0.003115	...	0.001367
-0.034653	0.009009	-0.012933	...	-0.004305
-0.095674	0.010533	-0.007532	...	-0.001261

Tableau 4.3: Aperçu des valeurs SHAP pour les six premières observations la partie entraînement des données

Ensuite, nous avons utilisé un diagramme de dispersion de densité des valeurs Shapely (figure 4.8) pour chaque caractéristique afin d'identifier l'impact de chaque variable sur le résultat du modèle pour les individus de l'ensemble de données de validation. Les caractéristiques sont triées par la somme des grandeurs de valeur SHAP de tous les échantillons. Ce diagramme devient très intéressant, car non seulement il indique quelles variables sont les plus influentes, mais il indique aussi leur gamme d'effets sur l'ensemble de données. La couleur nous permet d'apparier la façon dont les changements de valeur d'une variable influent sur la variable prédite. On peut voir que la variable `RA_DISTANCE_DRIVEN` parcourue a influé sur plus de prédiction dans une large mesure, alors que la variable `RA_EXPOSURE_DAY` influe sur moins de prédictions et dans une moindre mesure. Toutefois, si l'on compare cette dernière `RA_VEHICLE_AGE`, on voit que cette dernière caractéristique influe le modèle sur plus de prédictions, mais dans une mesure plus ou moins égale lorsqu'on compare en valeur absolue sur l'axe des x (les valeurs SHAP), soit l'impact sur le modèle. Lorsque les points de dispersion ne tiennent pas sur une ligne, ils s'empilent pour montrer la densité, et la couleur de chaque point représente la valeur de la caractéristique pour chaque assuré.

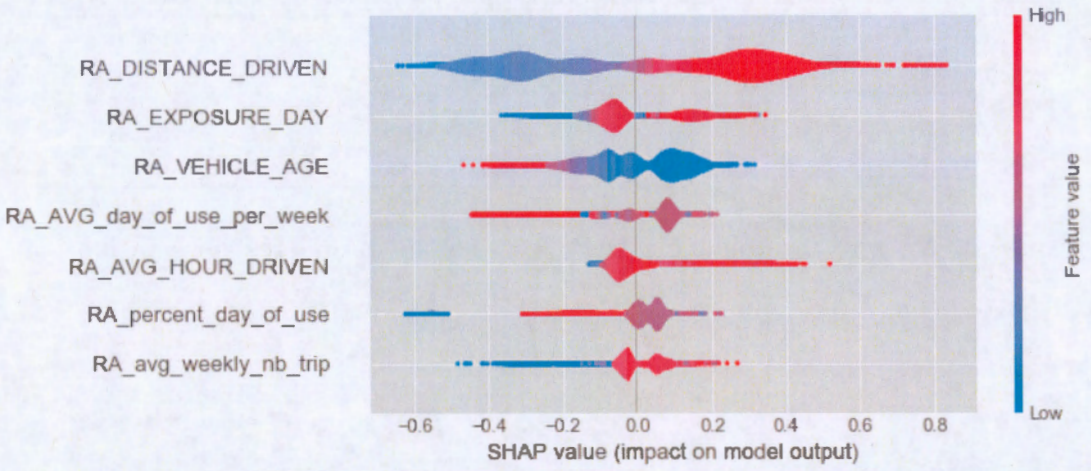


Figure 4.8: Diagramme de dispersion de densité des valeurs SHAP triées par la somme des grandeurs ces valeurs.

Lorsque compare le graphique de l'importance des variables à la figure (3.9) avec le diagramme d'importance des variables par les valeurs SHAP à la figure (4.8), on voit bien que l'ordre d'importance est similaire à l'exception des trois variables :

— `RA_percent_day_of_use`

— `RA_AVG_day_of_use_per_week`

— `RA_AVG_HOUR_DRIVEN`.

Cela est causé pour une raison bien simple, sur la première figure, nous avons une moyenne de l'importance de chacune des variables sur chacune des cinq parties de la validation croisée. Alors que dans la deuxième figure nous avons calculé les valeurs SHAP pour l'ensemble des données d'entraînement. Nous avons calculé les valeurs SHAP pour chacune des cinq partie des données de la validation croisée, ensuite nous avons fait une moyenne des valeurs SHAP qui nous a permet de tracé le graphique à la figure (B.5) où l'on constate que nous obtenons le même ordre

d'importance de variables. Toutefois, il est nécessaire de rappeler que 'objectif premier des valeurs SHAP est d'analyser la contribution marginale d'une valeur de caractéristique sur l'ensemble de la prédiction pour chaque assuré.

On voit bien qu'il s'agit d'une idée un peu similaire à celle de l'importance des variables, où nous pouvons déterminer l'impact de chaque variable en examinant l'ampleur de son coefficient sur nos valeurs prédites. Toutefois, les valeurs SHAP offrent deux avantages importants. Premièrement, elles peuvent être calculées pour n'importe quel modèle d'arbre de décision, de sorte qu'au lieu d'être limitées à des modèles de régression logistique simples, linéaires (et donc moins précis), nous pouvons construire des modèles complexes, non linéaires et plus précis. En plus, nous pouvons identifier les facteurs qui ont le plus d'impact pour chaque assuré pris individuellement, ce qui nous permet de personnaliser nos prochaines actions en conséquence. Par exemple, augmenter ou diminuer la franchise d'une prime pour une catégorie d'assurés donnée.

Bien que les valeurs SHAP puissent être un excellent outil, elles présentent des lacunes. D'une part, elles sont sensibles aux fortes corrélations entre les différentes caractéristiques. Lorsque des variables sont corrélées, leur impact sur le score du modèle peut être réparti entre elles d'un grand nombre de façons. Cela signifie qu'elles seront inférieures à ce qu'elles auraient été si toutes les caractéristiques corrélées, sauf une, avaient été supprimées du modèle. Le risque est que le fait de diviser les impacts de cette façon les fasse paraître moins importants que si leurs impacts restaient indivisibles. D'ailleurs, cette lacune n'est pas propre qu'aux valeurs SHAP, elle est aussi connue dans la technique d'importance de variables.

4.3 Interaction des variables par SHAP

Comme nous l'avons vu théoriquement à la section (2.6.4), il est possible de capter les effets d'interaction de variables par paire avec les valeurs SHAP. Nous obtenons ainsi une matrice pour chaque prédiction, où les principaux effets sont sur la diagonale et les effets d'interaction sont hors diagonale. Les principaux effets sont similaires aux valeurs SHAP que nous obtenons pour un modèle linéaire, et les effets d'interaction capturent toutes les interactions d'ordre supérieur et les divisent entre les termes d'interaction par paires. Notons que la somme de l'ensemble de la matrice d'interaction est la différence entre la production actuelle et la production attendue du modèle, de sorte que les effets de l'interaction sur le hors diagonale sont divisés en deux (puisqu'il y en a deux de chaque).

Cette matrice de valeurs d'interaction SHAP est illustré à la figure (4.9) où l'on aperçoit une matrice de graphiques récapitulatifs avec les principaux effets sur la diagonale et les effets d'interaction hors diagonale.

Nous avons exécuté un diagramme de dépendance partielle sur les valeurs d'interaction SHAP, cela nous a permis d'observer séparément les effets principaux pour la variable `RA_DISTANCE_DRIVEN` ainsi que les effets d'interaction entre cette même variable la variable `RA_VEHICLE_AGE` comme illustrée aux figures (4.10a) et (4.10b) respectivement. Dans la première figure, nous remarquons clairement que pour les conducteurs ayant parcouru une distance plus petite qu'envieront 8 000 km, l'effet principal de cette variable a contribué négativement à la prédiction de la fréquence des réclamations. Jusqu'à maintenant, ce résultat ne fait que confirmer notre intuition initiale ainsi que les résultats obtenus précédemment.

Dans la deuxième figure (4.10b), où contrairement aux effets principaux, les effets d'interactions saisissent une dispersion verticale qui permet de voir l'effet des deux

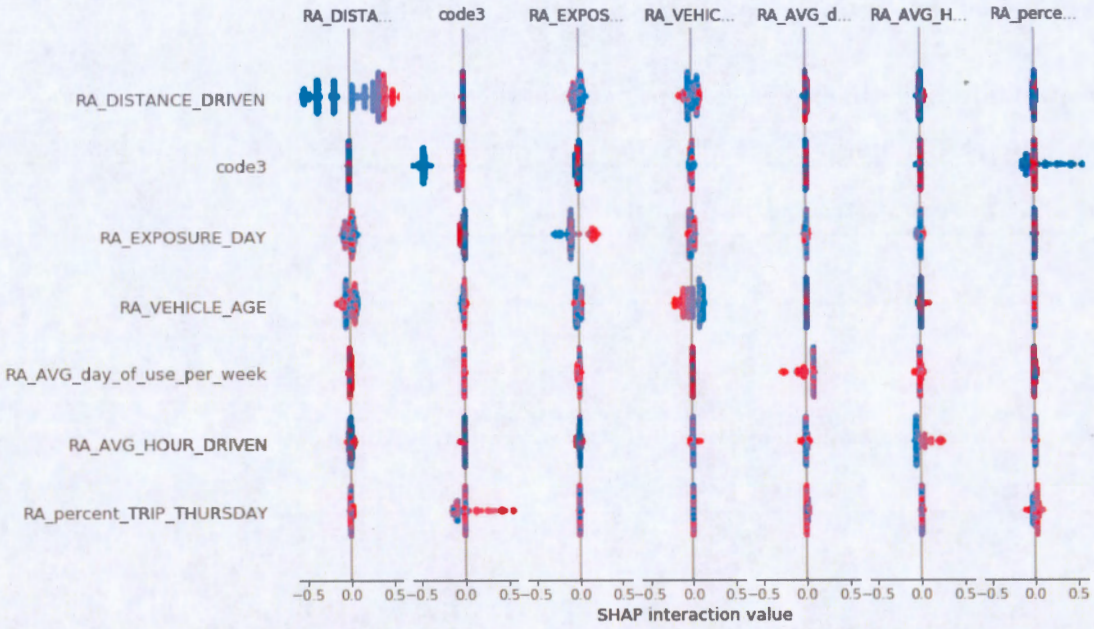
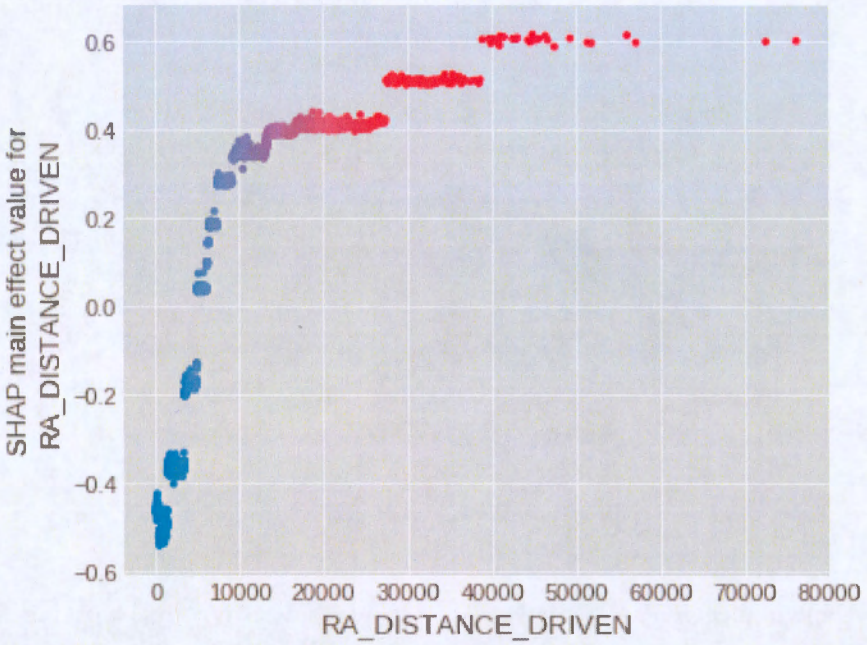


Figure 4.9: Matrice de prédiction par valeurs SHAP dont les principaux effets sont sur la diagonale et les effets d'interaction sont hors diagonale.

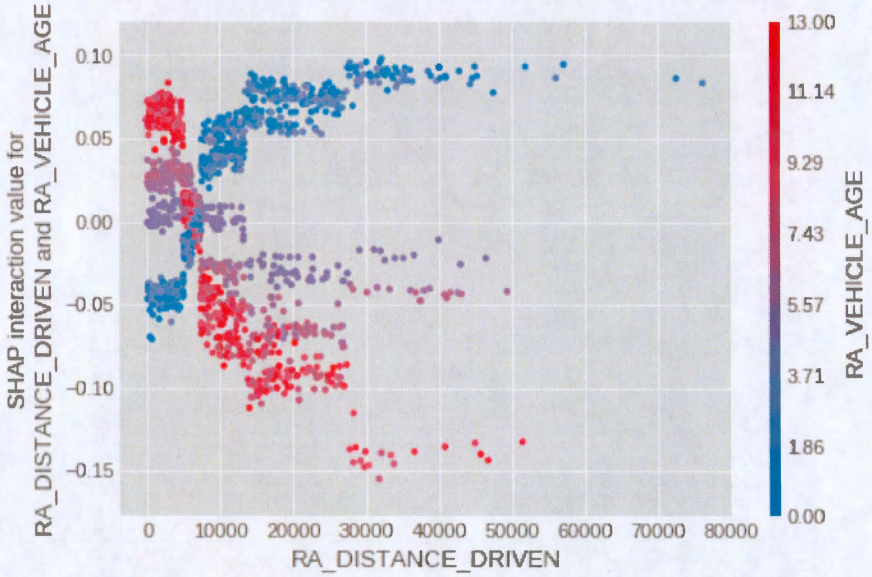
variables sur la prédiction du modèle. D'ailleurs, pour les exemples de ces deux variables, on peut remarquer que les vieux véhicules contribuent positivement à la valeur prédite lorsque le kilométrage parcouru est inférieur à 8 000 km, alors que peu de véhicules récents contribuent négativement sur la fréquence prédite dans cette même catégorie de kilométrage parcouru (inférieur à environ 8 000 km). Au-delà de cette distance parcourue, c'est l'effet contraire que nous obtenons. Cela peut s'expliquer par le fait que les propriétaires de véhicules récents ont tendance à réclamer plus fréquemment à cause du coût de réparation qui peut être plus souvent plus élevé que la franchise de la police d'assurance pour cette catégorie de véhicule. Nous avions déjà fait une telle constatation dans le graphique (4.4), sauf qu'avec les effets de l'interaction des deux variables, nous pouvons apercevoir à quel kilométrage cela peut survenir. Nous avons également tracé un graphique de dépendance partielle sur les effets d'interaction entre ces deux variables afin de

voir l'impact sur la prédiction dans les figures (4.10c) et (4.10d).

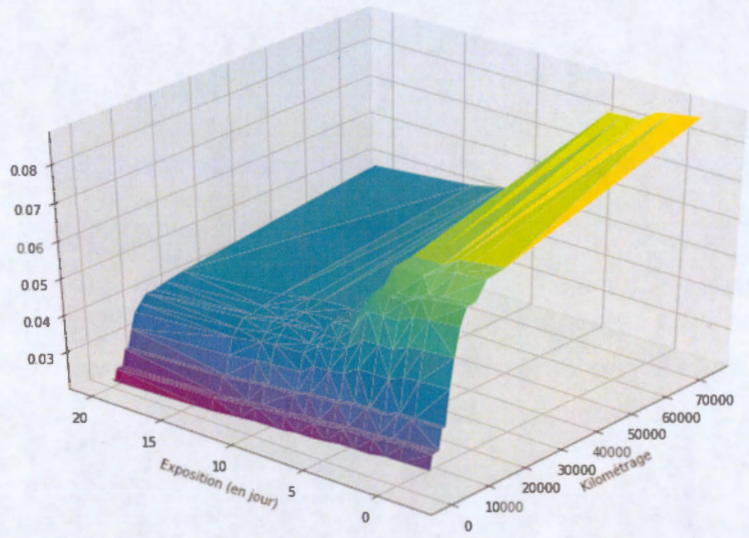
Les graphiques détaillés des effets principaux des six autres variables sélectionnées sont illustrés à la figure (4.11) alors que les vingt et une paires d'effets d'interaction hors diagonale sont tracées dans la figure (B.7).



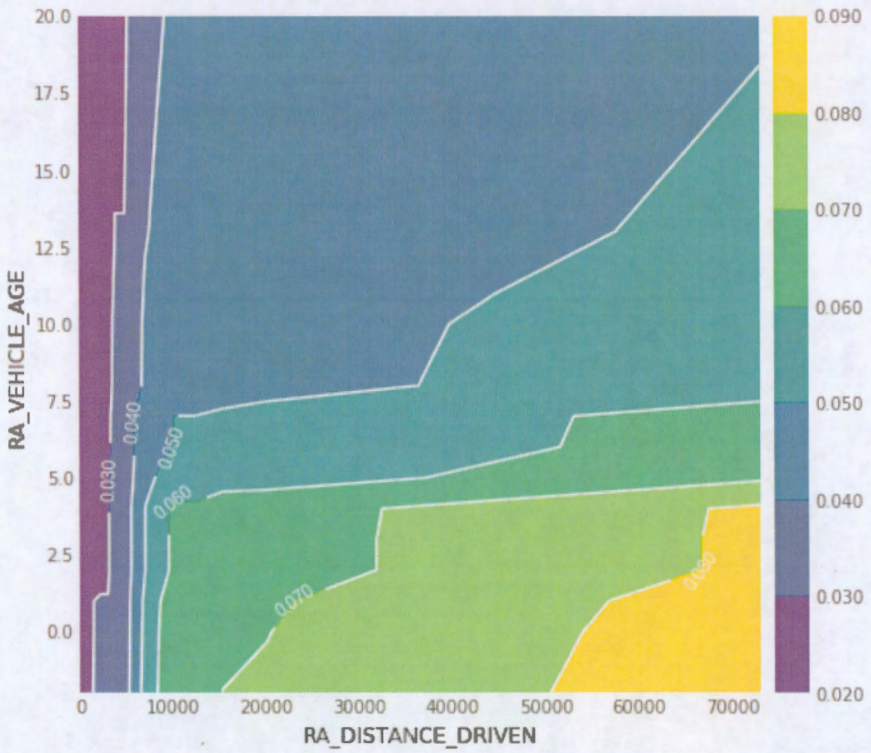
(a) Effets principaux pour la distance parcourue.



(b) Effets d'interaction entre la variable distance parcourue et l'âge du véhicule.



(c) Dépendance partielle sur les effets d'interaction entre la variable distance parcourue et l'âge du véhicule l'exposition en jour.



(d) Dépendance partielle sur les effets d'interaction entre la variable distance parcourue et l'âge du véhicule.

Appliquer la technique de dépendance partielle sur les valeurs SHAP est très avantageux, elle nous permet d'analyser la variation de la contribution des valeurs individuelle de chaque caractéristique de l'assuré à la prédiction de la variable réponse. Il ne s'agit plus de faire une moyenne sur des segments de la variable en question. On voit bien que cette technique nous permet de repousser les limites de la complexité et de l'exactitude du modèle, tout en nous permettant d'obtenir des explications intuitives pour chaque prédiction individuelle.

4.4 Comparaison des modèles

Maintenant que nous avons utilisé ces outils d'interprétation de notre modèle, nous nous en sommes servi afin de segmenter nos sept variables les plus importantes, au lieu d'appliquer une segmentation arbitraire comme à la section 4.1 où nous avons coupé nos variables aux $10, 20, \ldots 100^{e}$ centiles. Nous nous sommes servi plus précisément des effets d'interaction principaux (figure 4.10a et 4.11) comme suit :

1. `RA_DISTANCE_DRIVEN` : $[0, 5000, 8000, 12000, 25000, 80000]$
2. `RA_EXPOSURE_DAY` : utilisée comme *offset* (expo/max(expo))
3. `RA_VEHICLE_AGE` : $[-2, 0, 5, 8, 20]$
4. `RA_percent_day_of_use` : $[0, .2, .5, .8, 1]$
5. `RA_AVG_day_of_use_per_week` : $[0, 2, 4, 5, 7]$
6. `RA_AVG_HOUR_DRIVEN` : $[0, 230, 410, 610, 3500]$
7. `RA_avg_weekly_nb_trip` : $[0, 10, 20, 30, 40, 60]$.

Nous avons utilisé certaines méthodes d'estimations de modèles de données de comptage (Boucher *et al.*, 2007). D'abord, nous avons utilisé la théorie des GLM

afin d'estimer les paramètres d'une Poisson. Ensuite nous avons utilisé la loi binomiale négative 2 (NB2) qui est définie par :

$$\mathrm{P}(Y = y|\mu, \sigma) = \frac{\Gamma(y + \frac{1}{\sigma})}{\Gamma(y+1)\,\Gamma(\frac{1}{\sigma})} \left(\frac{\sigma\mu}{1+\sigma\mu}\right)^{y} \left(\frac{1}{1+\sigma\mu}\right)^{\frac{1}{\sigma}} \tag{4.1}$$

pour $y = 0, 1, 2, 3$ sinistres, et $\mu > 0$ et $\sigma > 0$ deux paramètres estimés.

Et la loi binomiale négative 1 (NB1) dont la fonction de distribution est donnée par :

$$\mathrm{P}(Y = y|\mu, \sigma) = \frac{\Gamma(y + \frac{\mu}{\sigma})\sigma^{y}}{\Gamma(\frac{\mu}{\sigma})\,\Gamma(y+1)\;(1+\sigma)^{y+\frac{\mu}{\sigma}}} \tag{4.2}$$

pour $y = 0, 1, 2, 3$ sinistres, et $\mu > 0$ et $\sigma > 0$ deux paramètres estimés.

Et finalement, la distribution Poisson gonflée à zéro (ZIP) ;

$$\mathrm{P}(Y = y) = \begin{cases} \sigma + (1-\sigma)e^{-\mu}, & \text{si } y = 0. \\ (1-\sigma)\frac{e^{-\mu}\mu^{y}}{y!}, & \text{si } y = 1, 2, 3. \end{cases} \tag{4.3}$$

pour $y = 0, 1, 2, 3$ sinistres, et $\mu > 0$ et $\sigma > 0$ deux paramètres estimés.

Les résultats des prédictions de la fréquence des sinistres au tableau (4.4). Le modèle GBM en utilisant les 121 variables sans aucun prétraitement des données nous a donné des prédictions très proches du nombre de sinistres observés. Notre première tentative a été d'ajuster les modèles de comptage (GLM Poisson, NB2, NB1 et ZIP) sur nos données d'entraînement, et d'estimer les paramètres de chacune des distributions afin de prédire les probabilités d'avoir $0, 1, 2$ et 3 sinistres pour chaque assuré se trouvant dans la partie test (30%) des données. Toutefois, aucun de ces modèles n'a convergé correctement. En plus, cela a nécessité un

grand prétraitement des données afin de corriger les valeurs manquantes ce qui a réduit le nombre d'observations de nos deux parties des données.

Ensuite, nous avons utilisé un autre modèle GBMP avec les mêmes hyperparamètre que le modèle original, mais cette fois, nous avons entraîné notre modèle que nous avons appelé GBM-7-Bin sur seulement les sept variables les plus importantes en les segmentant comme décrit précédemment. Évidement, ce modèle n'a pas été très performant, car comme nous l'avons décrit dans l'introduction de ce document, plus nous avons de données plus nos modèles non paramétriques sont performant. Toutefois, nous nous sommes servis des mêmes données (sept variables importantes segmentées) afin d'ajuster les modèles de comptage. Cette fois, ces modèles ont tous convergé, et nous avons obtenu des résultats très intéressants tels que décrits au tableau (4.4).

La performance prédictive de ces modèles est évaluée à l'aide de règles de scores (Verbelen *et al.*, 2018) pour les données de comptage présentées au tableau (4.5). Les règles de scores évaluent la qualité des prédictions probabilistes à l'aide d'un score numérique $s(P, n)$ basé sur la distribution prédictive P et le nombre n observé. Des scores plus faibles indiquent une meilleure qualité des prédictions. Dans ce tableau, nous définissons par $p_k = \mathbb{P}(N = k)$ et $P_k = \mathbb{P}(N \leq k)$ la fonction de masse de probabilité et la fonction de probabilité cumulative de la distribution prédictive P pour la variable de comptage N. La masse de probabilité évaluée à n observé est désignée par pn. La moyenne et l'écart-type de P sont écrits comme μ_p et σ_p respectivement. Finalement, nous avons définis, $\|p\| = \sum_{k=0}^{\infty} {p_k}^2$.

À partir du tableau (4.4), nous remarquons que grâce à la sélection et la segmentation de variables effectuée par GBMP et SHAP, nous avons conçu des modèles de comptage très performants (POISS, NB1, NB2 et ZIP). La prédiction sur les données test (*out of sample*) a été très proche des valeurs observées pour les

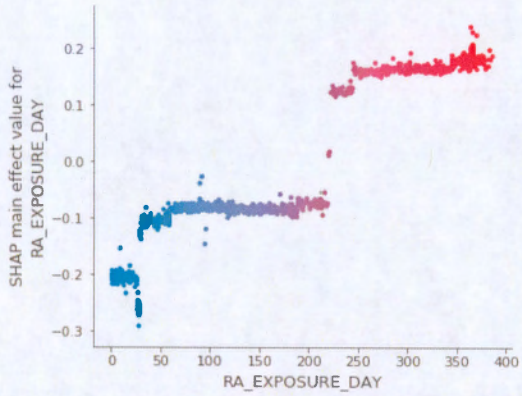
Nb claims	observed	GBM-121	GBM-7-Bin	POISS	NB1	NB2	ZIP
0	26932	26936	27156	26759	26758	26759	26567
1	1202	1213	964	1337	1338	1337	1514
2	57	41	26	49	49	49	64
3	2	1	0	2	2	2	2
aic	–	–	–	23941.27	**23940.27**	23940.50	23940.42
logs	–	–	–	2259.06	2271.81	**2221.54**	2257.88
qs	–	–	–	-56545.77	-56760.77	-56549.50	**-57342.58**
sphs	–	–	–	-28272.89	-28380.39	-28274.75	**-28671.29**
rps	–	–	–	1246.60	1246.63	**1246.60**	1251.90
dss	–	–	–	**-38008.46**	-36086.10	-37996.93	-36106.51
ses	–	–	–	1363.72	1363.74	**1363.71**	1369.67

Tableau 4.4: Comparaison de la fréquence prédite et certains scores de comparaison de modèles pour les données de comptage par quelques modèles, dont l'ajustement a été effectué sur des variables sélectionnées par le modèle GBMP et segmentées grâce aux méthodes d'interprétation des modèles d'apprentissage automatique.

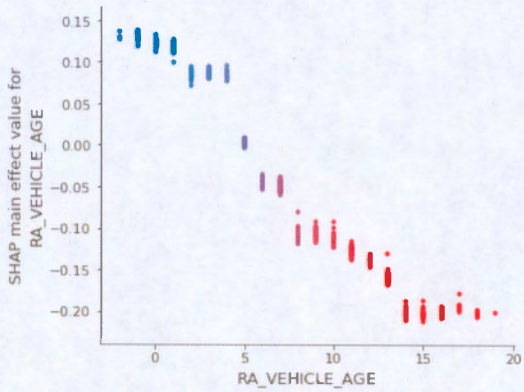
quatre modèles. Les modèles les plus prédictifs sont les modèles POISS et NB2. Le meilleur AIC a été celui de la NB1. Toutefois, le modèle NB2 l'emporte pour trois autres scores (logs, rps et ses).

Score	**Formula**
Logarithmique	$\text{logs}(P, n) = -\log(p_n)$
Quadratique	$\text{qs}(P, n) = -2p_n + \|\|p\|\|$
Spherique	$\text{sphs}(P, n) = -p_n / \|\|p\|\|$
Probabilité classée	$\text{rps}(P, n) = \sum_{k=0}^{\infty} \{P_k - \mathbb{1}(n \leq k)\}^2$
Dawid–Sebastiani	$\text{dss}(P, n) = \left(\frac{n-\mu_p}{\sigma_p}\right)^2 + 2\log(\sigma_p)$
Erreur classée	$\text{ses}(P, n) = (n - \mu_p)^2$

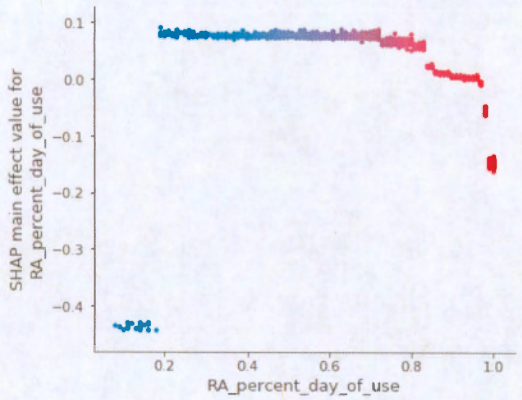
Tableau 4.5: Règles de comparaison de modèles pour les données de comptage.



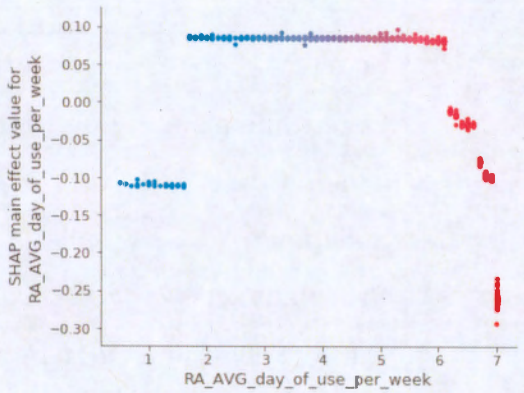
(a) L'exposition en jour.



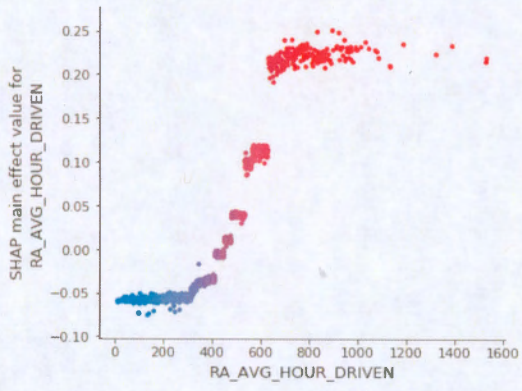
(b) L'âge du véhicule.



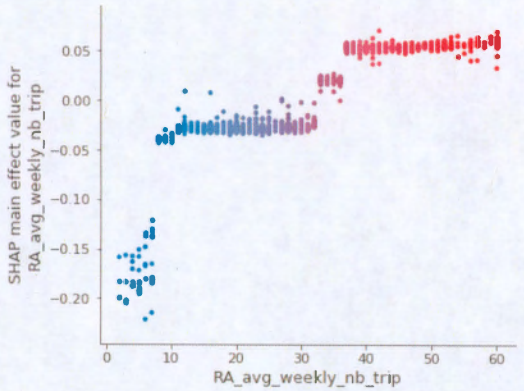
(c) Le pourcentage d'utilisation du véhicule durant le jour.



(d) Le nombre d'utilisations moyen par semaine du véhicule durant le jour.



(e) Nombre d'heures moyen conduit.



(f) Nombre de trajets moyen par semaine.

Figure 4.11: Les effets principaux pour les six variables sélectionnées.

CHAPITRE V

CONCLUSION

L'assurance est une entreprise ancienne, l'une des plus anciennes entreprises financières. Traditionnellement, de larges tables actuarielles sont utilisées pour classer les demandeurs de polices dans une catégorie de risque. Le groupe est ensuite ajusté de manière à ce qu'un nombre suffisant de personnes soient regroupées pour que, dans l'ensemble, les produits d'assurance soient rentables pour l'entreprise. Bien entendu, cette approche fait en sorte que certaines personnes paient plus qu'elles ne le devraient en fonction du niveau de base des données utilisées pour regrouper les gens.

L'arrivée des systèmes télématiques a un impact considérable sur l'industrie automobile. À la différence de la méthode traditionnelle plutôt stagnante de mesure de la façon dont un conducteur conduit, en utilisant des informations génériques et datées associées à son âge, son sexe, voire son état civil. L'assurance basée sur l'utilisation (ou Usage Based Insurance (UBI)) utilise le comportement au volant réel de la personne pour déterminer une prime d'assurance précise et équitable.

C'est une approche qui enthousiasme les clients et les compagnies d'assurance. En fait, c'est une aubaine pour les conducteurs prudents qui, depuis des années, voire des générations, sont injustement tarifiés à cause de leur âge, lieu de résidence ou autre caractéristique. Les assurés ne sont pas les seuls à pouvoir bénéficier de

la popularité croissante de la télématique de l'assurance automobile et de l'assurance basée sur l'utilisation. Les compagnies d'assurance se lancent aussi dans la recherche d'un gain de profit.

Cette approche a donné naissance à un tout nouveau domaine en assurance appelé l'assurtech, une combinaison des mots « assurance » et « technologie », inspirée du terme fintech. L'assurtech cherche à s'attaquer de front au problème de données ou d'analyse. En utilisant des données massives provenant de toutes sortes d'appareils, y compris celle générée par les modules d'UBI jusqu'aux traqueurs d'activité physique sur nos poignets.

Nous avons vu qu'avec des méthodes de modélisation comme les GBMP, nous avons pu sélectionner les variables les plus influentes à la prédiction de la fréquence de réclamation. Ensuite, nous avons pu segmenter ces variables en utilisant des méthodes d'interprétation des modèles complexe d'IA afin d'ajuster des modèles de comptage qui nous permis d'obtenir des prédictions de la fréquence de réclamations sur un ensemble de données test sur lesquels nos modèles n'ont jamais été entraînés.

En plus de meilleurs modèles de tarification, les méthodes d'IA permettent à l'assureur de jouer un rôle de prévention. Pensons seulement au rabais qu'il peut accorder aux assurés pour un comportement exemplaire de conduite automobile. Ainsi, l'assureur et l'assuré peuvent accéder potentiellement en temps réel, à l'information leur permettant de détecter l'apparition de facteurs de risque importants et d'intervenir afin de prévenir certaines réclamations.

Il est certain que les assureurs (dans leurs plans d'affaires) ainsi que les actuaires (dans leurs méthodes de modélisation) doivent s'adapter aux nouvelles technologies. Pensons seulement à l'arrivée de la voiture pleinement autonome, il serait intéressant de réfléchir à la conception des modèles prédictifs plus sophistiqués

qui tiendraient compte de l'ensemble des données massives enregistré par plusieurs capteurs et caméras de ces véhicules.

ANNEXES A

CODE PYTHON DE L'ALGORITHME (4) ET ADABOOST

```
    def _gradient(self, y, log_prediction):
        return y - np.exp(log_prediction)
```

Code A.1: Fonction permettant de calculer les résidus de déviance Poisson (algorithme 4 ligne 2-a)

```
def _terminal_node_estimates(self, X, y, log_prediction, tree):
....
    estimates[idx] = np.log(y_in_node / prediction_in_node)
return estimates
```

Code A.2: Fonction permettant d'estimer les β du GBMP (algorithme 4 ligne 2-c)

```
xi = x # initialisation de la variable d'entrée x
yi = y # initialisation de la variable réponse y
ei = 0 # initialisation de l'erreur (résidus)
n = len(yi)
predf = 0 # Un prédiction initiale à 0
nbiter=30
itteration=np.array([0,1,2,3,round(nbiter/2,0),nbiter-1])

```

```
for i in range(nbiter):
#     Ajuster un arbre de décision sur les données
    tree = DecisionTree(xi,yi)
    tree.find_better_split(0)

    r = np.where(xi == tree.split)[0][0]

    left_idx = np.where(xi <= tree.split)[0]
    right_idx = np.where(xi > tree.split)[0]

#     Calculer les prédictions
    predi = np.zeros(n)
    np.put(predi, left_idx, np.repeat(np.mean(yi[left_idx]), r))
    np.put(predi, right_idx, np.repeat(np.mean(yi[right_idx]), n-r))

    predi = predi[:,None]
    predf = predf + predi  # La prédiction finale est égale à
    # la prédiction initiale plus les résidus

    # Calcul des résidus à partir des données originales y
    ei = y - predf
    yi = ei # mettre à jours les yi

    # Code pour le graphique
    xa = np.array(x.x)
    order = np.argsort(xa)
    xs = np.array(xa)[order]
    ys = np.array(predf)[order]

    if i in itteration:
        f, (ax1, ax2) = plt.subplots(1, 2, sharey=True, figsize =
    (17,1.5))

```

```
41          ax1.plot(x,y, 'o')
42          ax1.plot(xs, ys, 'r')
43          ax1.set_xlabel('x')
44          ax1.set_ylabel('y / y_pred ' +str(i))
45          ax1.set_title("HM GBM code at "+str(i)+" itteration")
46
47          ax2.plot(x, ei, 'go')
48          ax2.set_xlabel('x')
49          ax2.set_ylabel('Residuals')
```

Code A.3: Code Python de l'algorithme adaBoost

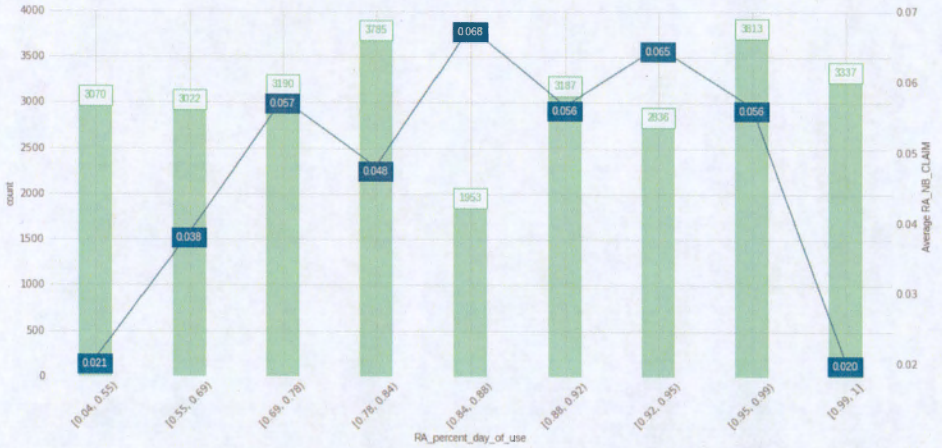
ANNEXES B

ANALYSE GRAPHIQUE

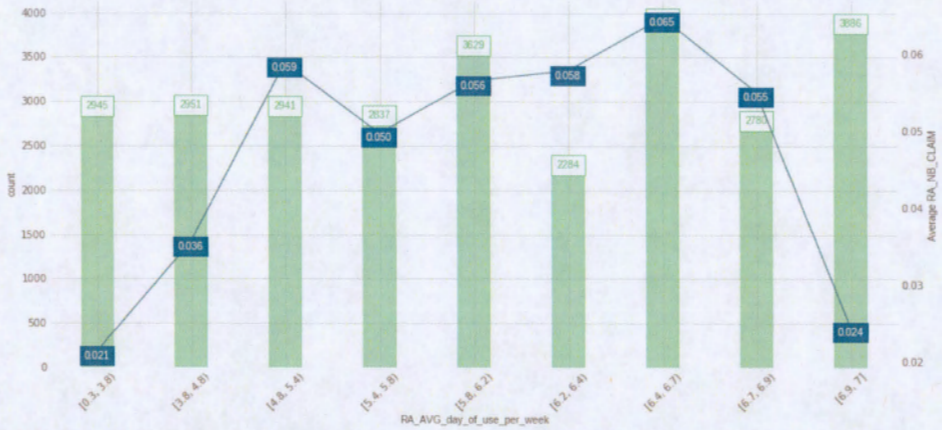
B.1 Dépendance partielle observée (données test)

(a) L'exposition en jour

(a) L'âge du véhicule

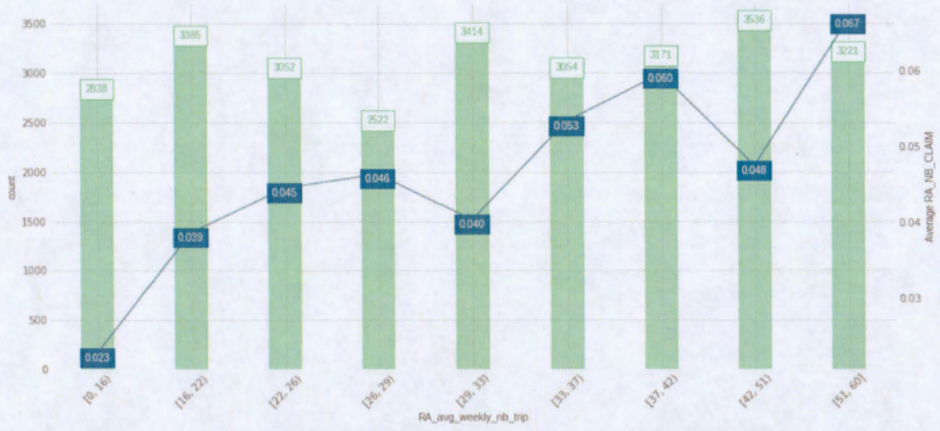


(b) Le pourcentage d'utilisation du véhicule durant le jour



(c) Le nombre d'utilisations moyen par semaine du véhicule durant le jour

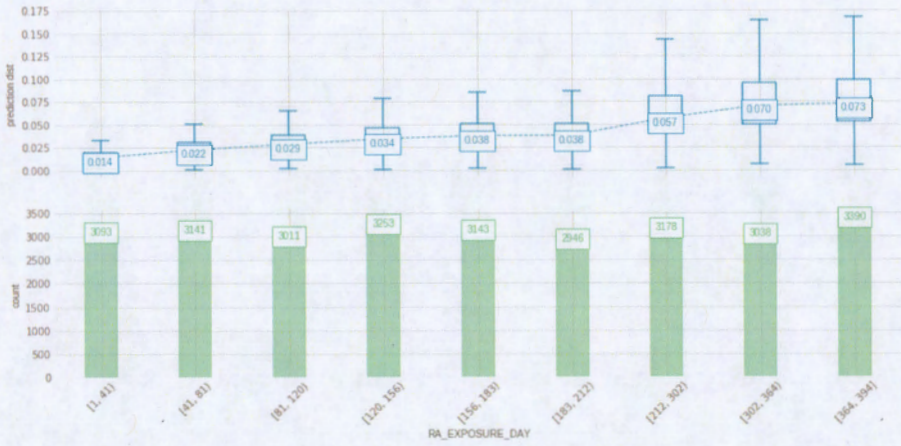
(d) Nombre d'heures moyen conduit



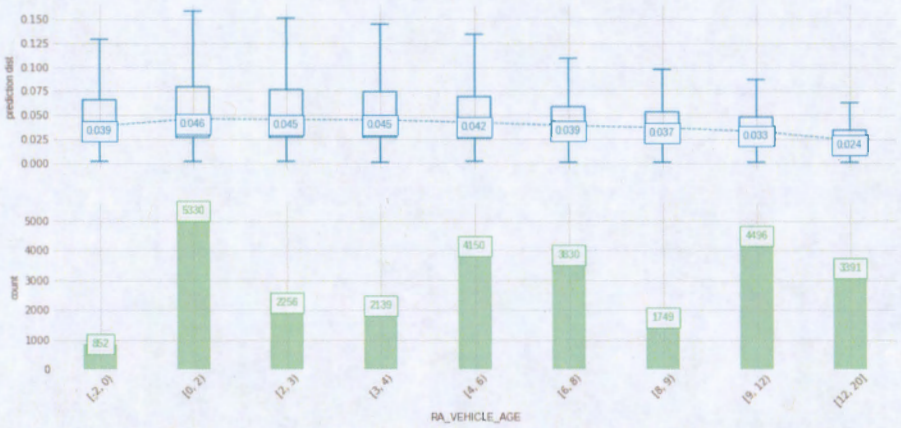
(e) Nombre de trajets moyen par semaine

Figure B.2: Diagramme de dépendance partielle observée sur la partie test des données

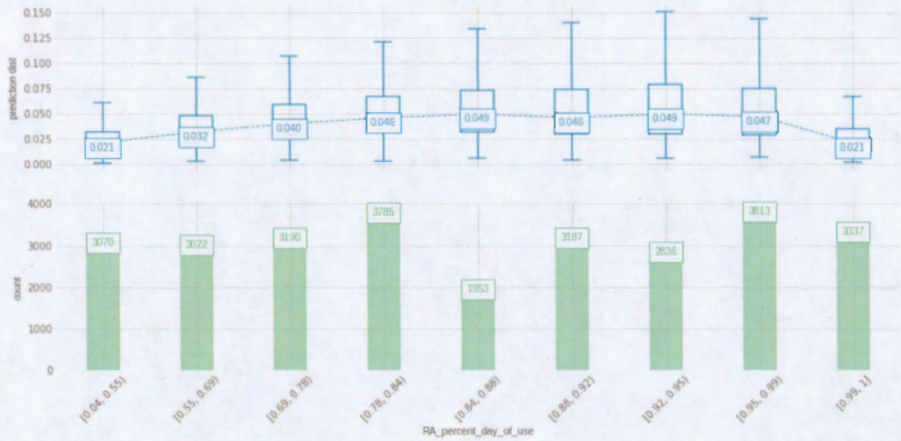
B.2 Dépendance partielle prédite (données test)



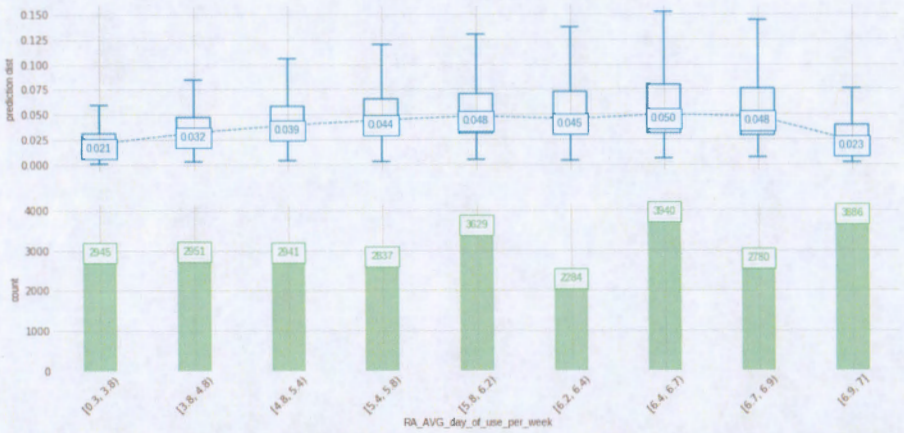
(a) L'exposition en jour



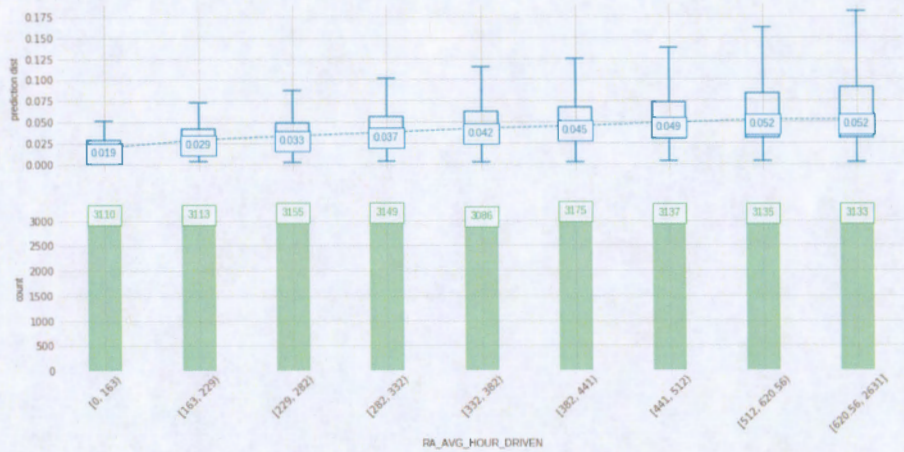
(a) L'âge du véhicule



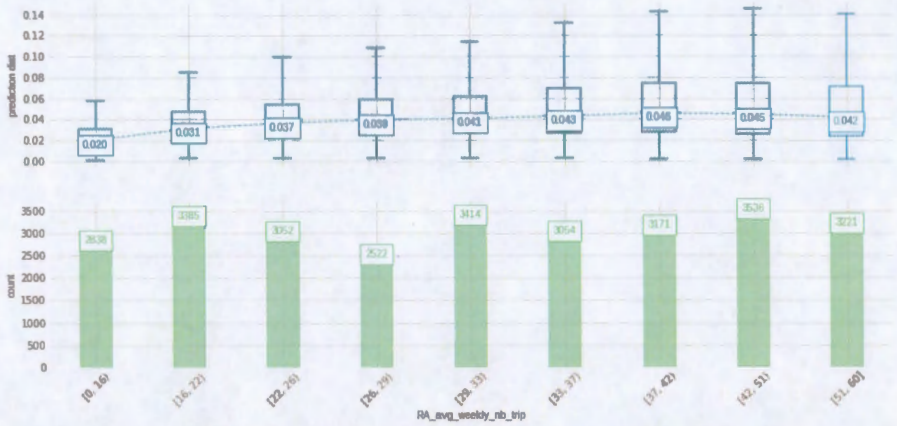
(b) Le pourcentage d'utilisation du véhicule durant le jour



(c) Le nombre d'utilisations moyen par semaine du véhicule durant le jour



(d) Nombre d'heures moyen conduit



(e) Nombre de trajets moyen par semaine

Figure B.4: Diagramme de dépendance partielle prédite sur la partie test des données

B.3 Valeurs SHAP

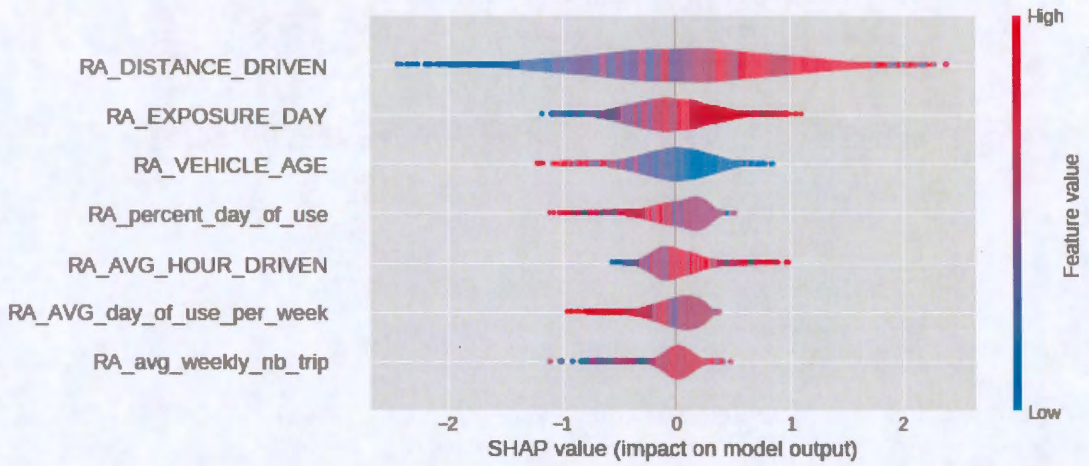


Figure B.5: Moyenne des valeurs SHAP sur les cinq partie des données de la validation croisée.

B.4 Effets d'interaction SHAP par paire

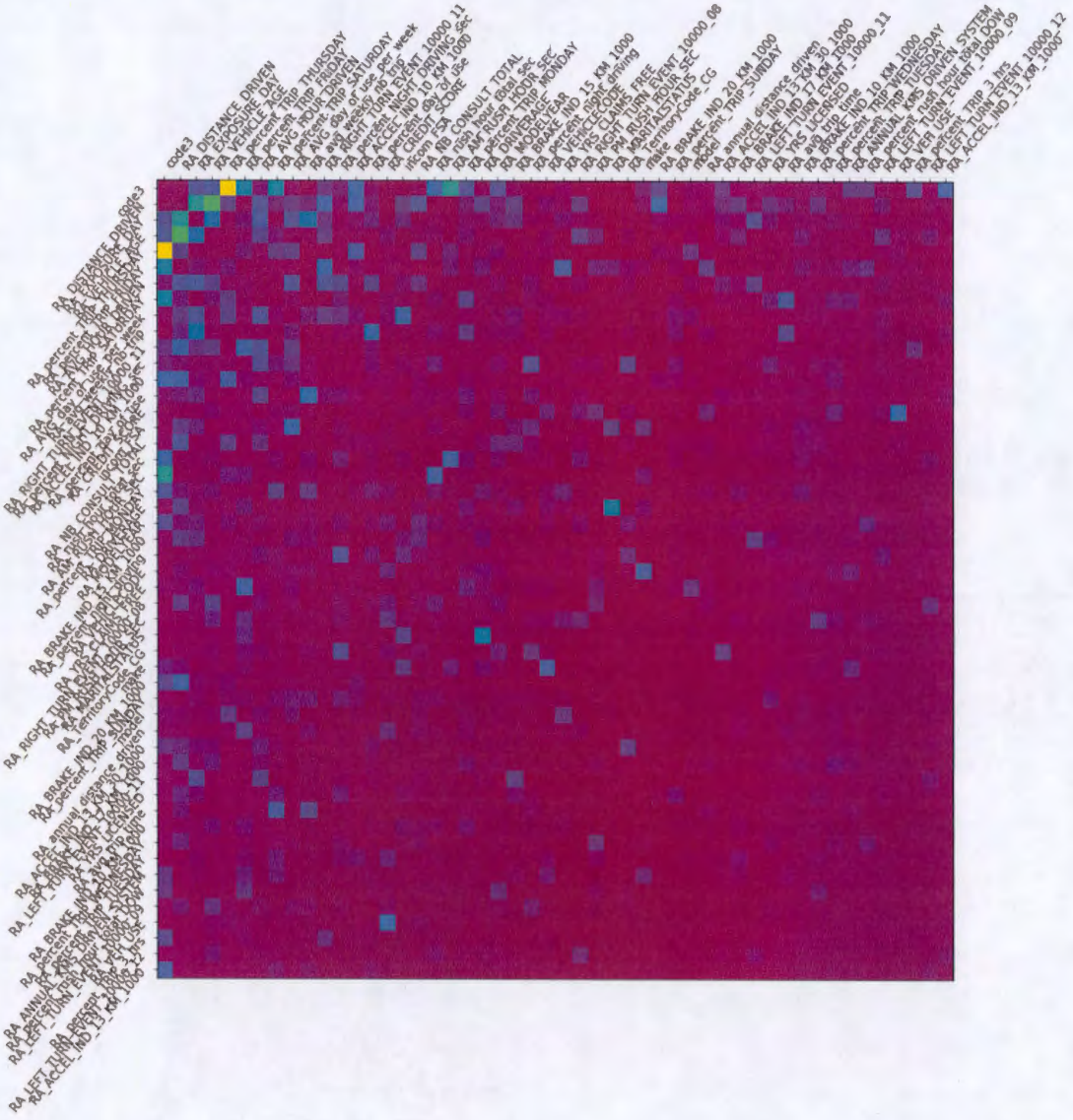
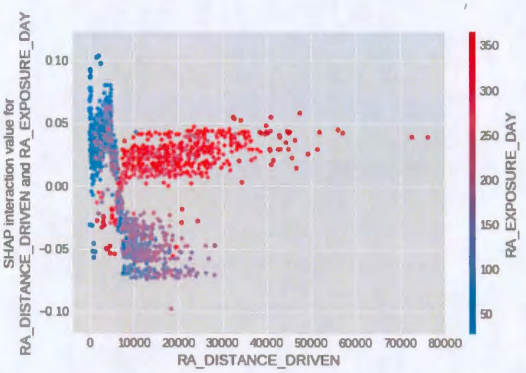
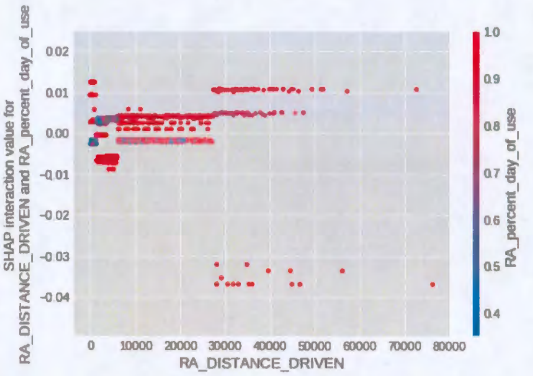
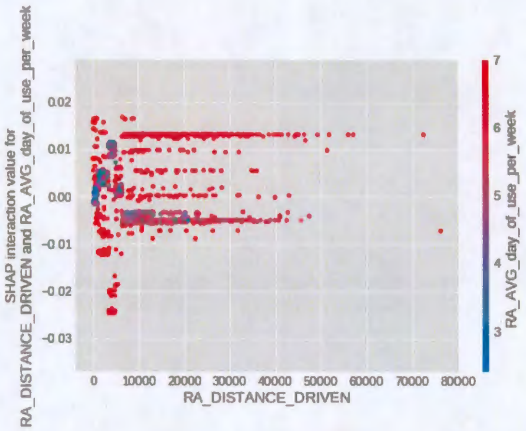
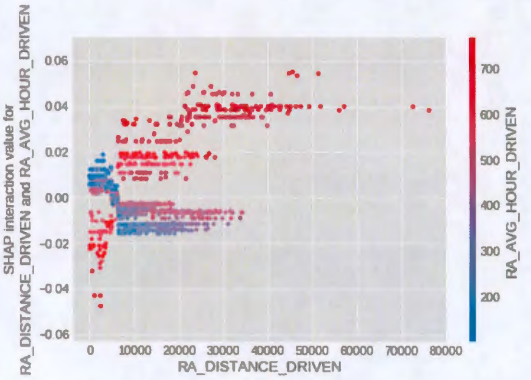


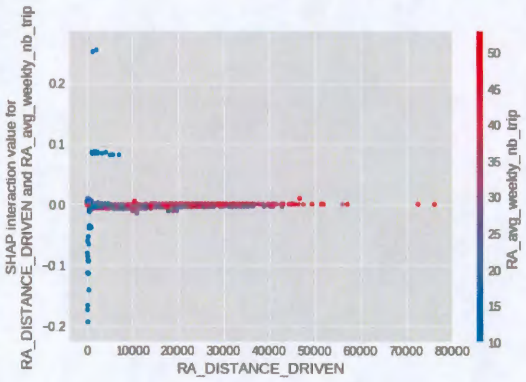
Figure B.6: Graphique de carte thermique représentant le niveau d'interaction par paire pour 50 des 121 variables explicatives.

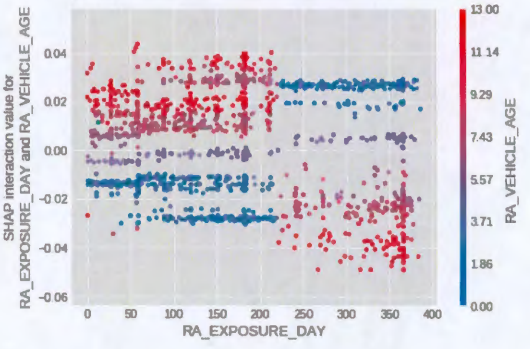

SHAP interaction value for
RA_DISTANCE_DRIVEN and RA_EXPOSURE_DAY
RA_DISTANCE_DRIVEN
RA_EXPOSURE_DAY

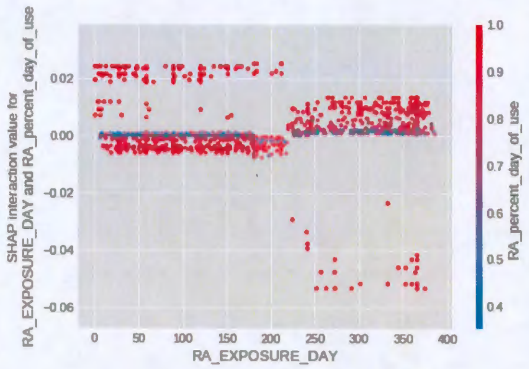

SHAP interaction value for
RA_DISTANCE_DRIVEN and RA_percent_day_of_use
RA_DISTANCE_DRIVEN
RA_percent_day_of_use

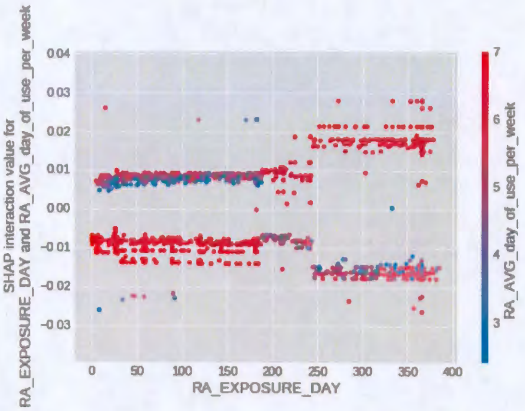

SHAP interaction value for
RA_DISTANCE_DRIVEN and RA_AVG_day_of_use_per_week
RA_DISTANCE_DRIVEN
RA_AVG_day_of_use_per_week

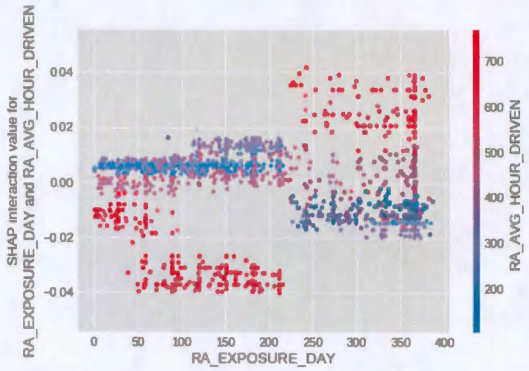

SHAP interaction value for
RA_DISTANCE_DRIVEN and RA_AVG_HOUR_DRIVEN
RA_DISTANCE_DRIVEN
RA_AVG_HOUR_DRIVEN

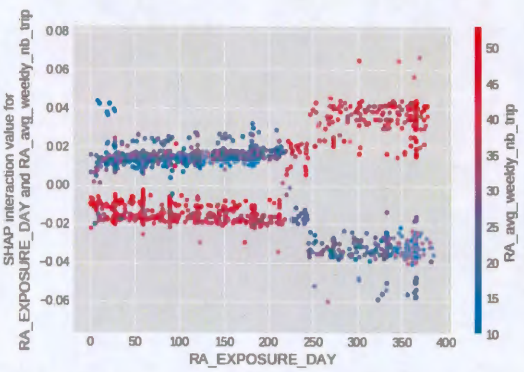

SHAP interaction value for
RA_DISTANCE_DRIVEN and RA_avg_weekly_nb_trip
RA_DISTANCE_DRIVEN
RA_avg_weekly_nb_trip

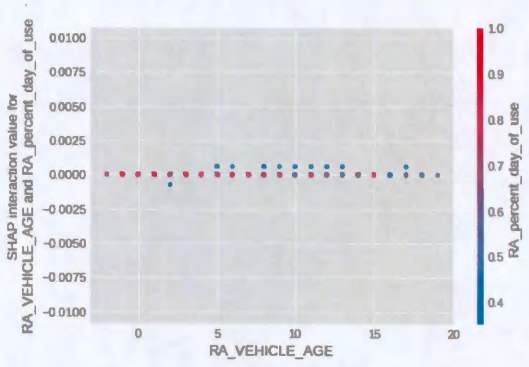

SHAP interaction value for
RA_EXPOSURE_DAY and RA_VEHICLE_AGE
RA_EXPOSURE_DAY
RA_VEHICLE_AGE

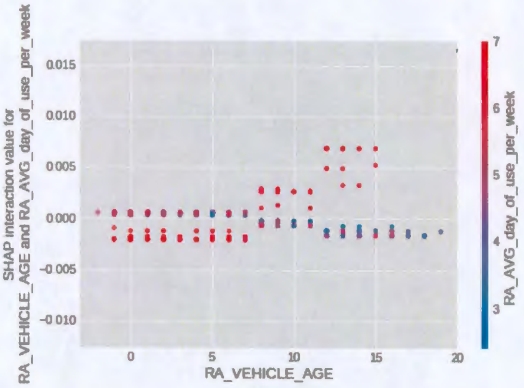

SHAP interaction value for
RA_EXPOSURE_DAY and RA_percent_day_of_use
RA_EXPOSURE_DAY
RA_percent_day_of_use

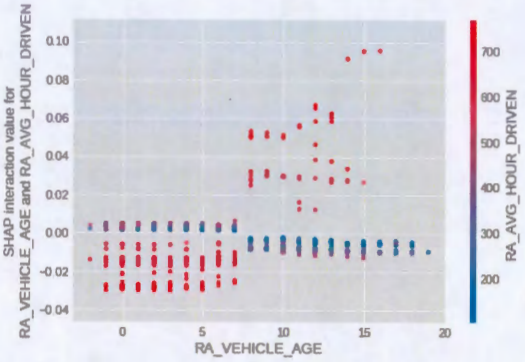

SHAP interaction value for
RA_EXPOSURE_DAY and RA_AVG_day_of_use_per_week
RA_EXPOSURE_DAY
RA_AVG_day_of_use_per_week

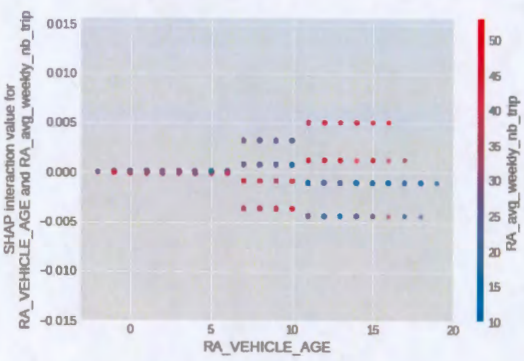

SHAP interaction value for
RA_EXPOSURE_DAY and RA_AVG_HOUR_DRIVEN
RA_EXPOSURE_DAY
RA_AVG_HOUR_DRIVEN

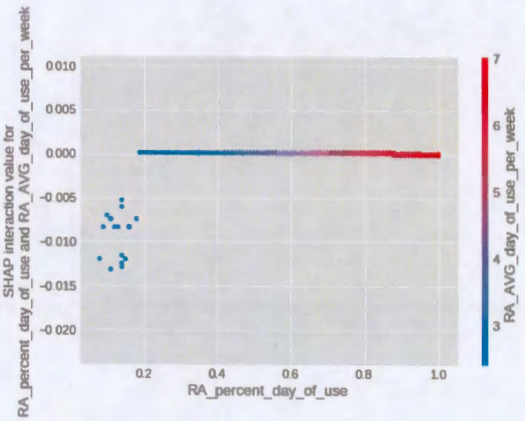

SHAP interaction value for
RA_EXPOSURE_DAY and RA_avg_weekly_nb_trip
RA_EXPOSURE_DAY
RA_avg_weekly_nb_trip

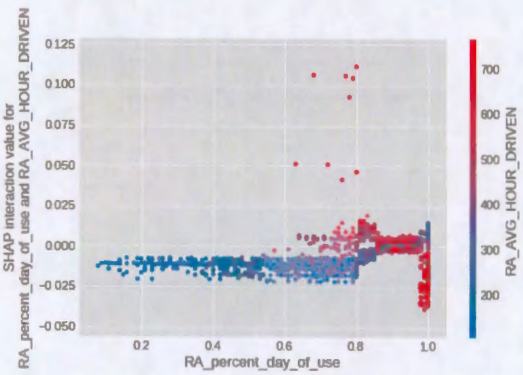

SHAP interaction value for
RA_VEHICLE_AGE and RA_percent_day_of_use
RA_VEHICLE_AGE
RA_percent_day_of_use

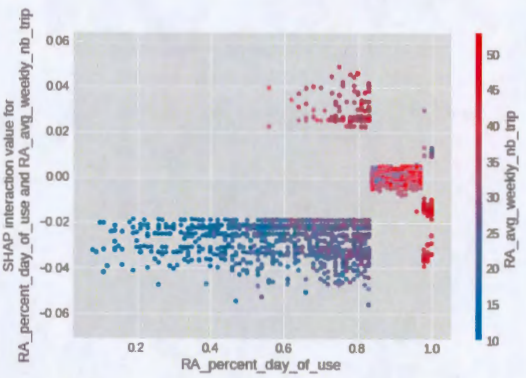

SHAP interaction value for
RA_VEHICLE_AGE and RA_AVG_day_of_use_per_week
RA_VEHICLE_AGE
RA_AVG_day_of_use_per_week

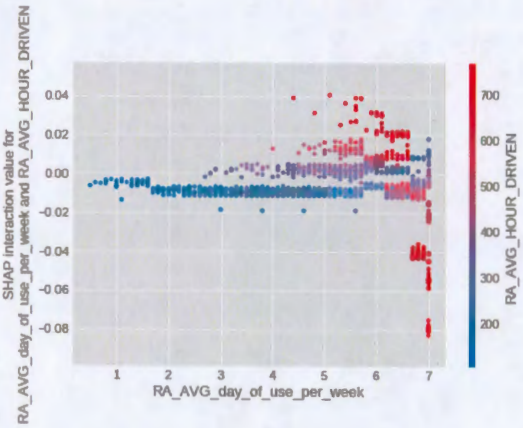

SHAP interaction value for
RA_VEHICLE_AGE and RA_AVG_HOUR_DRIVEN
RA_VEHICLE_AGE
RA_AVG_HOUR_DRIVEN


SHAP interaction value for
RA_VEHICLE_AGE and RA_avg_weekly_nb_trip
RA_VEHICLE_AGE
RA_avg_weekly_nb_trip


SHAP interaction value for
RA_percent_day_of_use and RA_AVG_day_of_use_per_week
RA_percent_day_of_use
RA_AVG_day_of_use_per_week


SHAP interaction value for
RA_percent_day_of_use and RA_AVG_HOUR_DRIVEN
RA_percent_day_of_use
RA_AVG_HOUR_DRIVEN


SHAP interaction value for
RA_percent_day_of_use and RA_avg_weekly_nb_trip
RA_percent_day_of_use
RA_avg_weekly_nb_trip


SHAP interaction value for
RA_AVG_day_of_use_per_week and RA_AVG_HOUR_DRIVEN
RA_AVG_day_of_use_per_week
RA_AVG_HOUR_DRIVEN

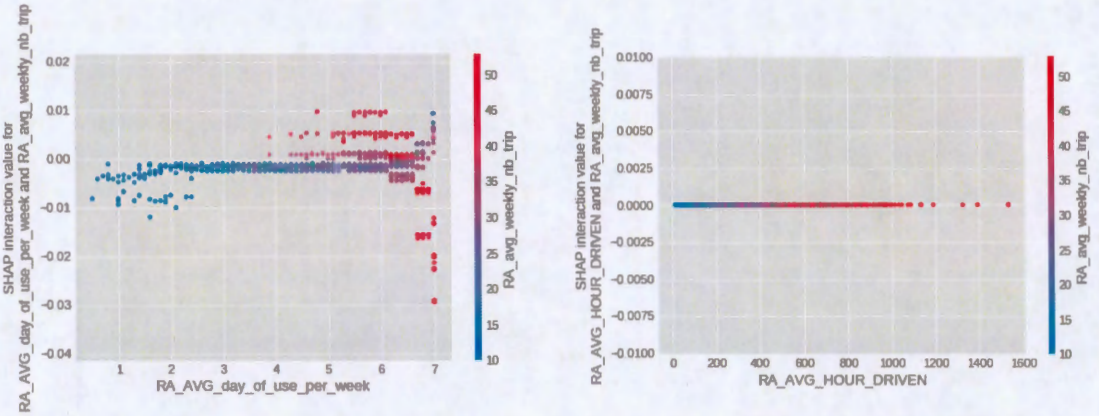


Figure B.7: Les effets d'interaction par paire entre les sept variables explicatives sélectionnées

ANNEXES C

DICTIONNAIRE DES DONNÉES

Tableau C.1: Dictionnaire des données

Variable	Description	Catég	mean	std	min	25%	50%	75%	max
RA_EXPOSURE_DAY	Nombre de jours d'exposition	0	182.94	113.11	1.00	89.00	181.00	273.00	394.00
RA_ANNUAL_KMS_DRIVEN_SYSTEM	Kilométrage déclaré	0	14,789.55	6,245.25	0.00	10000.00	12000.00	20000.00	200000.00
RA_DRIVERAGE	Âge de l'assuré	0	48.72	17.19	16.00	34.00	49.00	62.00	103.00
RA_GENDER	Sexe de l'assuré	1							
RA_VEHICLE_AGE	Âge du véhicule	0	5.76	4.47	-2.00	2.00	5.00	9.00	20.00
RA_YRS_LICENSED	Nombre d'années du permis de conduire	0	27.88	16.82	0.00	13.00	27.00	41.00	83.00
RA_YRS_CLAIMS_FREE	Nombre d'années sans réclamations	0	26.04	17.64	-2.00	9.00	25.00	40.00	83.00
RA_MARITALSTATUS	Statut matrimonial	1							
RA_VEH_RATE_TERRITORY	Cote du territoire	0	26.56	17.43	0.00	10.00	26.00	43.00	54.00
RA_VEH_USE	Utilisation du véhicule	1							
RA_MODELYEAR	Année du modèle du véhicule	0	13.45	4.49	0.00	10.00	14.00	17.00	21.00
RA_VEHICLECODE	Code du véhicule	0	699.93	505.78	0.00	226.00	667.00	1120.00	1751.00
RA_CREDIT_SCORE	Cote de crédit du conducteur	0	781.14	584.32	0.00	666.00	773.00	821.00	8000.00
RA_DISTANCE_DRIVEN	Distance parcourue observée	0	7,994.38	7,288.81	0.10	2616.00	5903.90	11255.30	76271.80
RA_LEFT_TURN_EVENT_10000_08		0	1,597.32	26,184.37	0.00	15.00	134.00	692.00	1388909.00
RA_RIGHT_TURN_EVENT_10000_08		0	1,753.48	24,139.68	0.00	23.00	254.00	1280.00	1521517.00
RA_LEFT_TURN_EVENT_10000_09		0	1,237.70	25,152.74	0.00	3.00	46.00	291.00	1359766.00
RA_RIGHT_TURN_EVENT_10000_09		0	1,254.16	22,950.14	0.00	6.00	94.00	627.00	1521517.00
RA_LEFT_TURN_EVENT_10000		0	924.86	23,195.68	0.00	0.00	7.00	64.00	1223109.00
RA_RIGHT_TURN_EVENT_10000		0	823.21	21,282.21	0.00	0.00	17.00	170.00	1521517.00
RA_LEFT_TURN_EVENT_10000_11		0	775.97	21,599.68	0.00	0.00	1.00	21.00	1201910.00
RA_RIGHT_TURN_EVENT_10000_11		0	656.53	20,201.10	0.00	0.00	4.00	61.00	1521503.00
RA_LEFT_TURN_EVENT_10000_12		0	673.73	20,126.73	0.00	0.00	0.00	6.00	1201910.00
RA_RIGHT_TURN_EVENT_10000_12		0	554.38	19,479.09	0.00	0.00	0.00	19.00	1521503.00
RA_ACCEL_IND_10_KM_1000	Nombre d'accélération brusques (10 km/h/s) par 1000km	0	84.16	121.82	0.00	14.00	42.00	102.00	1000.00
RA_ACCEL_IND_13_KM_1000	Nombre d'accélération brusques (13 km/h/s) par 1000km	0	9.51	38.14	0.00	0.00	2.00	6.00	1000.00
RA_ACCEL_IND_15_KM_1000	Nombre d'accélération brusques (15 km/h/s) par 1000km	0	3.69	27.85	0.00	0.00	0.00	1.00	1000.00
RA_ACCEL_IND_17_KM_1000	Nombre d'accélération brusques (17 km/h/s) par 1000km	0	1.97	23.44	0.00	0.00	0.00	0.00	1000.00
RA_ACCEL_IND_20_KM_1000	Nombre d'accélération brusques (20 km/h/s) par 1000km	0	1.15	20.68	0.00	0.00	0.00	0.00	1000.00

RA_ACCEL_IND_23_KM_1000	Nombre d'accélération brusques (23 km/h/s) par 1000km	0	0.79	18.34	0.00	0.00	0.00	0.00	1000.00
RA_BRAKE_IND_10_KM_1000	Nombre de freinages brusques (10 km/h/s) par 1000km	0	155.97	156.98	0.00	54.00	106.00	201.00	1000.00
RA_BRAKE_IND_13_KM_1000	Nombre de freinages brusques (13 km/h/s) par 1000km	0	19.04	37.26	0.00	5.00	10.00	22.00	1000.00
RA_BRAKE_IND_15_KM_1000	Nombre de freinages brusques (15 km/h/s) par 1000km	0	6.33	25.77	0.00	1.00	3.00	6.00	1000.00
RA_BRAKE_IND_17_KM_1000	Nombre de freinages brusques (17 km/h/s) par 1000km	0	2.72	22.10	0.00	0.00	1.00	2.00	1000.00
RA_BRAKE_IND_20_KM_1000	Nombre de freinages brusques (20 km/h/s) par 1000km	0	1.24	20.07	0.00	0.00	0.00	1.00	1000.00
RA_BRAKE_IND_23_KM_1000	Nombre de freinages brusques (23 km/h/s) par 1000km	0	0.78	17.91	0.00	0.00	0.00	0.00	1000.00
RA_ACCEL_IND_10_KM_30_1000	Nombre d'accélération brusques (10 km/h/s) par 1000km captés à une vitesse de 30	0	83.92	121.38	0.00	14.00	42.00	102.00	1000.00
RA_ACCEL_IND_13_KM_30_1000	Nombre d'accélération brusques (13 km/h/s) par 1000km captés à une vitesse de 30	0	9.22	36.14	0.00	0.00	2.00	6.00	1000.00
RA_ACCEL_IND_15_KM_30_1000	Nombre d'accélération brusques (15 km/h/s) par 1000km captés à une vitesse de 30	0	3.39	25.01	0.00	0.00	0.00	1.00	1000.00
RA_ACCEL_IND_17_KM_30_1000	Nombre d'accélération brusques (17 km/h/s) par 1000km captés à une vitesse de 30	0	1.65	19.90	0.00	0.00	0.00	0.00	1000.00
RA_ACCEL_IND_20_KM_30_1000	Nombre d'accélération brusques (20 km/h/s) par 1000km captés à une vitesse de 30	0	0.82	16.53	0.00	0.00	0.00	0.00	1000.00
RA_ACCEL_IND_23_KM_30_1000	Nombre d'accélération brusques (23 km/h/s) par 1000km captés à une vitesse de 30	0	0.44	12.89	0.00	0.00	0.00	0.00	1000.00
RA_BRAKE_IND_10_KM_30_1000	Nombre de freinage brusques (10 km/h/s) par 1000km captés à une vitesse de 30	0	155.97	156.98	0.00	54.00	106.00	201.00	1000.00
RA_BRAKE_IND_13_KM_30_1000	Nombre de freinage brusques (13 km/h/s) par 1000km captés à une vitesse de 30	0	19.04	37.26	0.00	5.00	10.00	22.00	1000.00
RA_BRAKE_IND_15_KM_30_1000	Nombre de freinage brusques (15 km/h/s) par 1000km captés à une vitesse de 30	0	6.33	25.77	0.00	1.00	3.00	6.00	1000.00
RA_BRAKE_IND_17_KM_30_1000	Nombre de freinage brusques (17 km/h/s) par 1000km captés à une vitesse de 30	0	2.72	22.10	0.00	0.00	1.00	2.00	1000.00
RA_BRAKE_IND_20_KM_30_1000	Nombre de freinage brusques (20 km/h/s) par 1000km captés à une vitesse de 30	0	1.24	20.07	0.00	0.00	0.00	1.00	1000.00
RA_BRAKE_IND_23_KM_30_1000	Nombre de freinage brusques (23 km/h/s) par 1000km captés à une vitesse de 30	0	0.78	17.91	0.00	0.00	0.00	0.00	1000.00

RA_ACCEL_IND_10_KM_60_1000	Nombre d'accélération brusques (10 km/h/s) par 1000km captés à une vitesse de 60	0	0.40	14.72	0.00	0.00	0.00	0.00	1000.00
RA_ACCEL_IND_13_KM_60_1000	Nombre d'accélération brusques (13 km/h/s) par 1000km captés à une vitesse de 60	0	0.40	14.72	0.00	0.00	0.00	0.00	1000.00
RA_ACCEL_IND_15_KM_60_1000	Nombre d'accélération brusques (15 km/h/s) par 1000km captés à une vitesse de 60	0	0.40	14.72	0.00	0.00	0.00	0.00	1000.00
RA_ACCEL_IND_17_KM_60_1000	Nombre d'accélération brusques (17 km/h/s) par 1000km captés à une vitesse de 60	0	0.40	14.72	0.00	0.00	0.00	0.00	1000.00
RA_ACCEL_IND_20_KM_60_1000	Nombre d'accélération brusques (20 km/h/s) par 1000km captés à une vitesse de 60	0	0.40	14.72	0.00	0.00	0.00	0.00	1000.00
RA_ACCEL_IND_23_KM_60_1000	Nombre d'accélération brusques (23 km/h/s) par 1000km captés à une vitesse de 60	0	0.40	14.72	0.00	0.00	0.00	0.00	1000.00
RA_BRAKE_IND_10_KM_60_1000	Nombre de freinage brusques (10 km/h/s) par 1000km captés à une vitesse de 60	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_BRAKE_IND_13_KM_60_1000	Nombre de freinage brusques (13 km/h/s) par 1000km captés à une vitesse de 60	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_BRAKE_IND_15_KM_60_1000	Nombre de freinage brusques (15 km/h/s) par 1000km captés à une vitesse de 60	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_BRAKE_IND_17_KM_60_1000	Nombre de freinage brusques (17 km/h/s) par 1000km captés à une vitesse de 60	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_BRAKE_IND_20_KM_60_1000	Nombre de freinage brusques (20 km/h/s) par 1000km captés à une vitesse de 60	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_BRAKE_IND_23_KM_60_1000	Nombre de freinage brusques (23 km/h/s) par 1000km captés à une vitesse de 60	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_ACCEL_IND_10_KM_90_1000	Nombre d'accélération brusques (10 km/h/s) par 1000km captés à une vitesse de 90	0	0.09	5.07	0.00	0.00	0.00	0.00	593.00
RA_ACCEL_IND_13_KM_90_1000	Nombre d'accélération brusques (13 km/h/s) par 1000km captés à une vitesse de 90	0	0.09	5.07	0.00	0.00	0.00	0.00	593.00
RA_ACCEL_IND_15_KM_90_1000	Nombre d'accélération brusques (15 km/h/s) par 1000km captés à une vitesse de 90	0	0.09	5.07	0.00	0.00	0.00	0.00	593.00
RA_ACCEL_IND_17_KM_90_1000	Nombre d'accélération brusques (17 km/h/s) par 1000km captés à une vitesse de 90	0	0.09	5.07	0.00	0.00	0.00	0.00	593.00
RA_ACCEL_IND_20_KM_90_1000	Nombre d'accélération brusques (20 km/h/s) par 1000km captés à une vitesse de 90	0	0.09	5.07	0.00	0.00	0.00	0.00	593.00
RA_ACCEL_IND_23_KM_90_1000	Nombre d'accélération brusques (23 km/h/s) par 1000km captés à une vitesse de 90	0	0.09	5.07	0.00	0.00	0.00	0.00	593.00
RA_BRAKE_IND_10_KM_90_1000	Nombre de freinage brusques (10 km/h/s) par 1000km captés à une vitesse de 90	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00

RA_BRAKE_IND_13_KM_90_1000	Nombre de freinage brusques (13 km/h/s) par 1000km captés à une vitesse de 90	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_BRAKE_IND_15_KM_90_1000	Nombre de freinage brusques (15 km/h/s) par 1000km captés à une vitesse de 90	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_BRAKE_IND_17_KM_90_1000	Nombre de freinage brusques (17 km/h/s) par 1000km captés à une vitesse de 90	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_BRAKE_IND_20_KM_90_1000	Nombre de freinage brusques (20 km/h/s) par 1000km captés à une vitesse de 90	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_BRAKE_IND_23_KM_90_1000	Nombre de freinage brusques (23 km/h/s) par 1000km captés à une vitesse de 90	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_ACCEL_IND_10_KM_200_1000	Nombre d'accélération brusques (10 km/h/s) par 1000km captés à une vitesse de 200	0	0.04	2.03	0.00	0.00	0.00	0.00	272.00
RA_ACCEL_IND_13_KM_200_1000	Nombre d'accélération brusques (13 km/h/s) par 1000km captés à une vitesse de 200	0	0.04	2.03	0.00	0.00	0.00	0.00	272.00
RA_ACCEL_IND_15_KM_200_1000	Nombre d'accélération brusques (15 km/h/s) par 1000km captés à une vitesse de 200	0	0.04	2.03	0.00	0.00	0.00	0.00	272.00
RA_ACCEL_IND_17_KM_200_1000	Nombre d'accélération brusques (17 km/h/s) par 1000km captés à une vitesse de 200	0	0.04	2.03	0.00	0.00	0.00	0.00	272.00
RA_ACCEL_IND_20_KM_200_1000	Nombre d'accélération brusques (20 km/h/s) par 1000km captés à une vitesse de 200	0	0.04	2.03	0.00	0.00	0.00	0.00	272.00
RA_ACCEL_IND_23_KM_200_1000	Nombre d'accélération brusques (23 km/h/s) par 1000km captés à une vitesse de 200	0	0.04	2.03	0.00	0.00	0.00	0.00	272.00
RA_BRAKE_IND_10_KM_200_1000	Nombre de freinage brusques (10 km/h/s) par 1000km captés à une vitesse de 200	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_BRAKE_IND_13_KM_200_1000	Nombre de freinage brusques (13 km/h/s) par 1000km captés à une vitesse de 200	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_BRAKE_IND_15_KM_200_1000	Nombre de freinage brusques (15 km/h/s) par 1000km captés à une vitesse de 200	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_BRAKE_IND_17_KM_200_1000	Nombre de freinage brusques (17 km/h/s) par 1000km captés à une vitesse de 200	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_BRAKE_IND_20_KM_200_1000	Nombre de freinage brusques (20 km/h/s) par 1000km captés à une vitesse de 200	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_BRAKE_IND_23_KM_200_1000	Nombre de freinage brusques (23 km/h/s) par 1000km captés à une vitesse de 200	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA_NB_CONSULT_TOTAL	Nombre consultation totales	0	4.02	19.01	0.00	0.00	0.00	1.00	1003.00
RA_zone	La zone d'habitation	1							
RA_exposure_time	Temps d'exposition % qui est le nb jours/365	0	0.50	0.31	0.00	0.24	0.50	0.75	1.08
RA_avg_weekly_nb_trip	Nombre moyen d'utilisations de la voiture par semaine	0	32.29	13.46	0.00	23.00	31.00	41.00	60.00

RA_avg_trip_time	Temps moyen d'utilisation de la voiture	0	13.66	4.68	0.00	10.00	13.00	16.00	60.00
RA_avg_trip_distance	Kilométrage moyen par utilisation	0	9.71	5.17	0.00	6.00	9.00	12.00	60.00
RA_AVG_HOUR_DRIVEN	Moyenne du temps de conduite	0	383.76	203.81	1.00	243.00	356.00	490.00	3141.00
RA_AVG_day_of_use_per_week	Nombre d'utilisation moyen par semaine durant le jour du véhicule	0	5.63	1.32	0.20	5.00	6.00	6.60	7.00
RA_percent_day_of_use	Pourcentage d'utilisation du véhicule durant le jour	0	0.80	0.19	0.03	0.72	0.86	0.95	1.00
RA_nb_breaking_per_1000	Nombre de freinage par 1000km	0	18.95	37.15	0.00	5.00	10.00	21.00	1000.00
RA_nb_ACCELERATION_per_1000	Nombre d'accélération par 1000km	0	9.47	38.01	0.00	0.00	2.00	6.00	1000.00
RA_percent_NIGHT_DRIVING_sec	Pourcentage de conduite durant la nuit	0	0.19	0.13	0.00	0.09	0.17	0.27	1.00
RA_AM_RUSH_HOUR_SEC	Pourcentage de conduite durant les heures de trafic matinales	0	0.10	0.09	0.00	0.03	0.08	0.15	1.00
RA_PM_RUSH_HOUR_SEC	Pourcentage de conduite durant les heures de trafic en soirée	0	0.14	0.08	0.00	0.09	0.13	0.18	1.00
RA_rush_hour_total_sec	Pourcentage de conduite durant les heures de trafic	0	0.24	0.14	0.00	0.15	0.21	0.31	1.00
RA_percent_rush_hour_total_DOW	Pourcentage de conduite durant les heures de trafic (pas certain) day of week	0	0.23	0.11	0.00	0.15	0.21	0.29	1.00
RA_percent_weekend_driving	Pourcentage de conduite durant les jours de semaine	0	0.25	0.09	0.00	0.20	0.25	0.29	1.00
RA_percent_weekday_driving	Pourcentage de conduite durant les jours du week-end	0	0.75	0.09	0.00	0.71	0.75	0.80	1.00
RA_percent_night_driving	Pourcentage de conduite la nuit	0	0.18	0.12	0.00	0.08	0.16	0.25	1.00
RA_percent_TRIP_MONDAY	Pourcentage du nombre d'utilisations du véhicule le lundi	0	0.14	0.05	0.00	0.12	0.14	0.16	1.00
RA_percent_TRIP_TUESDAY	Pourcentage du nombre d'utilisations du véhicule le mardi	0	0.15	0.05	0.00	0.13	0.15	0.17	1.00
RA_percent_TRIP_WEDNESDAY	Pourcentage du nombre d'utilisations du véhicule le mercredi	0	0.15	0.05	0.00	0.13	0.15	0.17	1.00
RA_percent_TRIP_THURSDAY	Pourcentage du nombre d'utilisations du véhicule le jeudi	0	0.15	0.05	0.00	0.13	0.15	0.17	1.00
RA_percent_TRIP_FRIDAY	Pourcentage du nombre d'utilisations du véhicule le vendredi	0	0.16	0.05	0.00	0.14	0.16	0.18	1.00
RA_percent_TRIP_SATURDAY	Pourcentage du nombre d'utilisations du véhicule le samedi	0	0.14	0.06	0.00	0.11	0.13	0.16	1.00
RA_percent_TRIP_SUNDAY	Pourcentage du nombre d'utilisations du véhicule le dimanche	0	0.11	0.05	0.00	0.08	0.11	0.14	1.00

RA_percent_TRIP_2_hrs	Pourcentage du nombre d'utilisations pour une utilisation du véhicule sur une durée de deux heures ou moins	0	0.00	0.01	0.00	0.00	0.00	0.00	0.41
RA_percent_TRIP_3_hrs	Pourcentage du nombre d'utilisations pour une utilisation du véhicule sur une durée de trois heures ou moins	0	0.00	0.00	0.00	0.00	0.00	0.00	0.32
RA_percent_TRIP_4_hrs	Pourcentage du nombre d'utilisations pour une utilisation du véhicule sur une durée de quatre heures ou moins	0	0.00	0.00	0.00	0.00	0.00	0.00	0.27
RA_annual_distance_driven	Distance parcourue annualisée	0	16,079.24	10,083.28	2.35	8980.52	14284.11	21082.52	279261.50
RA_TerritoryCode_CG	Code du territoire	0	26.50	17.43	0.00	10.00	26.00	43.00	54.00
VehicleType	Type du véhicule	1							
make	Marque du véhicule	1							
model	Modèle du véhicule	1							
code3	FSA du code postal	1							
incom_FSA	Revenu moyen du ménage par région FSA	0	29,119.53	17,858.21	70.00	16760.00	25090.00	36220.00	90100.00
credit_score_incom_FSA	Cote de crédit moyenne par région FSA	0	0.04	0.19	0.00	0.02	0.03	0.05	42.86

RÉFÉRENCES

Delta Boosting Machine with Application to General Insurance | SOA. Récupéré le 2017-06-17 de https: //www.soa.org/research-reports/2017/delta-boosting-machine/

(2016). Complete Guide to Parameter Tuning in Gradient Boosting (GBM) in Python. Récupéré le 2017-06-20 de https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/

Bergstra, J. et Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(Feb), 281–305.

Boucher, J.-P., Denuit, M. et Guillén, M. (2007). Risk classification for claim counts : a comparative analysis of various zeroinflated mixed poisson and hurdle models. *North American Actuarial Journal*, *11*(4), 110–131.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. http://dx.doi.org/10.1007/BF00058655. Récupéré de https://doi.org/10.1007/BF00058655

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Stone, C. J. et Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Canada, S. (2019). Census Profile, 2016 Census. https://www12.statcan.gc.ca/census-recensement/2011/geo/bound-limit/bound-limit-2016-eng.cfm. [Dernier accès le 2018-11-10].

Denuit, M., Maréchal, X., Pitrebois, S. et Walhin, J.-F. (2007). *Actuarial modelling of claim counts : Risk classification, credibility and bonus-malus systems*. John Wiley & Sons.

Dorogush, A. V., Ershov, V. et Gulin, A. (2018). Catboost : gradient boosting with categorical features support. *arXiv preprint arXiv :1810.11363*.

Efron, B. et Tibshirani, R. J. (1994). *An Introduction to the Bootstrap*. CRC Press. Google-Books-ID : gLlpIUxRntoC.

Feller, W. (1971). Law of large numbers for identically distributed variables. *An introduction to probability theory and its applications*, *2*, 231–234.

Fisher, A., Rudin, C. et Dominici, F. (2018). All Models are Wrong but many are Useful : Variable Importance for Black-Box, Proprietary, or Misspecified Prediction Models, using Model Class Reliance. Récupéré de `http://arxiv.org/abs/1801.01489`

Freund, Y. et Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, *55*(1), 119–139.

Freund, Y., Schapire, R. E. *et al.* (1996). Experiments with a new boosting algorithm. Dans *Icml*, volume 96, 148–156.

Friedman, J., Hastie, T. et Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Friedman, J., Hastie, T., Tibshirani, R. *et al.* (2000). Additive logistic regression : a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, *28*(2), 337–407.

Friedman, J. H. (2001). Greedy Function Approximation : A Gradient Boosting Machine. *The Annals of Statistics*, *29*(5), 1189–1232. Récupéré le 2017-06-27 de `http://www.jstor.org.proxy.bibliotheques.uqam.ca:2048/stable/2699986`

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367–378.

Guelman, L. (2012). Gradient boosting trees for auto insurance loss cost modeling and prediction. *Expert Systems with Applications*, *39*(3), 3659–3667. `http://dx.doi.org/10.1016/j.eswa.2011.09.058`. Récupéré le 2017-06-17 de `http://www.sciencedirect.com/science/article/pii/S0957417411013674`

Ian Goodfellow, Y. B. et Courville, A. (2016). Deep learning. Book in preparation for MIT Press

James, G., Witten, D., Hastie, T. et Tibshirani, R. (2013). *An introduction to statistical learning*, volume 6. Springer.

Lee, S., Lin, S. et Antonio, K. (2015). Delta boosting machine and its application in actuarial modeling.

Lundberg, S. (2019). SHAP : Open source shapley additive explanations Python. [Dernier accès le 2019-01-11]. Récupéré le 2019-01-11 de `https://github.com/slundberg/shap`

Lundberg, S. M. et Lee, S.-I. (2017a). A Unified Approach to Interpreting Model Predictions. Récupéré de `http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions`

Lundberg, S. M. et Lee, S.-I. (2017b). Consistent feature attribution for tree ensembles. Récupéré de `http://arxiv.org/abs/1706.06060`

Marsland, S. (2015). *Machine learning : an algorithmic perspective*. CRC press.

McCullagh, P. et Nelder, J. A. (1989). *Generalized linear models*, volume 37. CRC press.

Molnar, C. (2019). *Interpretable Machine Learning*. https ://christophm.github.io/interpretable-ml-book/. `https://christophm.github.io/interpretable-ml-book/`.

Natekin, A. et Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, *7*. `http://dx.doi.org/10.3389/fnbot.2013.00021`. Récupéré le 2017-06-27 de `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3885826/`

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. et Duchesnay, E. (2011). Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Ribeiro, M. T., Singh, S. et Guestrin, C. (2016a). Model-agnostic interpretability of machine learning. *arXiv preprint arXiv :1606.05386*.

Ribeiro, M. T., Singh, S. et Guestrin, C. (2016b). Why should i trust you ? : Explaining the predictions of any classifier. Dans *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144. ACM.

Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games*, *2*(28), 307–317.

Snoek, J., Larochelle, H. et Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. Dans *Advances in neural information processing systems*, 2951–2959.

Štrumbelj, E. et Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, *41*(3), 647–665.

The Co-operators Group Limited. (2017). *TELEMATICS RESEARCH PROJECT, Documentation and Findings Version 3.0.* The Co-operators Group Limited.

Thoma, M. (2019). VIN : Open source vin decoder Python. [Dernier accès le 2019-01-09]. Récupéré le 2019-01-09 de `https://github.com/MartinThoma/vin_decoder`

Tseng, G. Interpreting Complex Models with SHAP. [Dernier accès le 2019-03-14]. Récupéré le 2019-01-11 de `https://canopylabs.com/resources/interpreting-complex-models-with-shap/`

Verbelen, R., Antonio, K. et Claeskens, G. (2018). Unravelling the predictive power of telematics data in car insurance pricing. *Journal of the Royal Statistical Society : Series C (Applied Statistics)*, *67*(5), 1275–1304.

Wuthrich, M. V. et Buser, C. (2017). Data analytics for non-life insurance pricing.