Méthodes de machine learning pour la tarification des assurances.

ADOU Kouamé Mathurin (kouame.m.adou@aims-senegal.org)
African Institute for Mathematical Sciences (AIMS), Senegal
Supervisé par: Dr. Djiby Fall
Directeur Academique Aims Sénégal

20 Mars 2022

Soumission pour l'obtention du Master en Science Mathématique, Option Big Data à AIMS-Sénégal

Déclaration

Ce travail a été effectué à AIMS Sénégal en remplissant partiellement les conditions requises pour l'obtention d'un Master scientifique. Je déclare par la présente que, sauf mention contraire, ce travail n'a jamais été présenté en totalité ou en partie pour l'obtention d'un diplôme à AIMS Sénégal ou à toute autre université.

ADOU Kouamé Mathurin: 

REMERCIEMENTS

La réalisation de ce mémoire fut pour nous une expérience enrichissante et formatrice. Elle nous a notamment permis d'éprouver une détermination et forger en nous le sens de la rigueur. Toutefois, la concrétisation de ce mémoire n'aurait été possible sans l'aide et le soutien de nombreuses personnes que nous tenons à remercier. Ainsi, nos premiers mots de remerciements sont adressés à Dieu, lui qui nous a permis d'être à ce stade d'études universitaires.

Nos remerciements vont à l'endroit du Docteur **D**jiby **F**ALL, notre Encadreur de mémoire, pour la confiance qu'il nous a faite en nous confiant ce travail de recherche, pour sa disponibilité, sa rigueur, son apport scientifique et ses judicieux conseils.

Nous remercions également le coordonnateur du projet AIMS et toute l'équipe de AIMS-Sénégal en l'occurrence le Professeur Titulaire **F**all, President de AIMS-Sénégal et le Professeur Titulaire **F**ranck **K**alala **M**UTOMBO, le Directeur Académique d'avoir pris en considération notre demande de candidature pour le programme de l'année 2020-2021.

Nos remerciements vont à l'endroit des tuteurs en particulier **L**ema **L**ogamou **S**eknewna, mon tuteur pour sa disponibilité et ses remarques pertinentes.

Nous tenons à remercier aussi les membres du jury d'avoir accepté d'examiner notre travail.

Nos respects et nos remerciements vont ensuite, à tous nos enseignants qui ont montré beaucoup de générosité à partager avec nous leur savoir.

Nous souhaitons aussi témoigner toute notre reconnaissance à **K**PTA **A**. **A**bokon **B**érenger **P**., Maîtres Assistant à l'université Nangui Abrogoua (Côte d'Ivoire) et à **E**ssoh **M**ODESTE, Docteur à l'université Nangui Abrogoua qui ont accepté de nous recommander à AIMS-sénégal.

Nous tenons à remercier chaleureusement nos amis des promotions 2020-2021, en particulier **F**anta **S**OUMAHORO pour son encouragement, son aide tout au long de cette aventure.

Nous tenons à témoigner toute notre reconnaissance à **A**BO **A**dou **M**axime, **S**AKO **K**ady et **P**auline **O**rnela **M**egne **C**houdja, pour leurs apports dans la rédaction de ce mémoire.

De plus profond du coeur, nous remercions ici toutes les personnes évoluant hors de la sphère professionnelle, qui nous ont soutenu dans cette rédaction. Toute ma famille bien entendu, qui m'a toujours apporté son soutien moral et financier.

Nous remercions enfin tous ceux qui d'une manière ou d'une autre, ont contribué à la réussite de ce travail et qui n'ont pas pu être cités ici.

DEDICACE

Je dédie ce travail, un événement marquant de ma vie à mon adorable mère, **F**ODJO **K**ossia **K**ra, qui m'est chère et qui a œuvré à ma réussite de par son amour, son soutien et ses énormes sacrifices consentis. Puisse Dieu, le Tout puissant te soutenir dans toutes tes épreuves et t'accorder une longue vie!

Résumé

Pour faire face à leurs engagements futurs, les organismes d'assurance sont tenus de constituer des provisions avant la déclaration d'un sinistre. Ces provisions dites provisions d'ouverture sont évaluées différemment d'un assureur à un autre, conduisant à considérer plusieurs approches. Les assureurs envisagent de plus en plus l'approche visant à l'utilisation de leur base de sinistres. Plusieurs travaux ont notamment été menés sur le provisionnement tout en se déclinant comme une alternative aux méthodes classiques. La principale motivation de ce mémoire porte sur le calcul de la tarification en assurance non vie selon différentes caractéristiques initiales. Il est ainsi question, à terme, de proposer à des modèles de machine learning pour la prévision des sinistres. Pour mener à bien ces travaux, nous nous appuyons sur des méthodes de machine learning et les modèles linéaires généralisés comme modèle classique pas le machine learning, Decision Tree regression, le random forest regression et sur les réseaux de neurones. Une comparaison de ses méthodes est présentée pour un choix ultime d'un modèle d'implémentation.

Mots clés:

Modèles Linéaires Généralisés (GLM), Random forest regression, Decison Tree regression, Reseaux de neurones

Abstract

To meet their future commitments, insurance organizations are required to set aside reserves ahead as a claims occurrences. These so-called opening reserves are valued differently from one insurer to another, leading to the consideration of several approaches. Insurers are increasingly considering the approach aimed at using their claims base. In particular, several works have been conducted on reserving while declining as an alternative to classical methods. The main motivation of this dissertation is the calculation of non-life insurance pricing according to different initial characteristics. It is thus a question, in the long run, of proposing models for the prediction of claims. To carry out this work, we rely on machine learning methods and in particular those based on generalized linear models, Decison Trees regression , random forest and on neural networks. A comparison of these methods is presented for an ultimate choice of an implementation model.

Keywords:

Generalized Linear Model (GLM), Random forest regression, Decison Tree regression, neural networks.

Table des matières

Table des figures

Liste des tableaux

1. Introduction

Dans ce projet de recherche, nous voulons comparer la performance des méthodes de machine learning dans l'assurance avec les méthodes traditionnelles. Nous essaierons de comprendre et d'expliquer où se trouvent les succès et les pièges afin de déterminer quand et quelles techniques de machine learning sont appropriées pour un problème donné.

La tarification est l'un des principaux problèmes mathématiques auxquels sont confrontés les actuaires. Car, leurs objectifs premiers est d'estimer le coût attendu de chaque contrat d'assurance appelé **prime pure**. Mais cette valeur attendue est conditionnée par les informations disponibles sur l'assuré et sur le contrat, que nous appelons le profil d'entrée. Dans ce travail, il sera question de l'assurance automobile qui est une tâche très complexe pour les assureurs car plusieurs facteurs entre en ligne de compte et faire de hypothèse d'indépendance sur chaque facteur, cela peut être préjudiciable et souvent irréalisable si tous les facteurs sont prisent en compte et cela est due à la malédiction de la dimensionnalité [1, 3] . Le deuxième problème qui se présente est de la distribution des réclamtions: souvent asymétrique avec une grande majorité de zéros et quelques valeurs peu fiables et très grandes, c'est-à-dire des valeurs positives très élevées. La modélisation des données avec une telle distribution est essentiellement difficile car les valeurs aberrantes, qui sont échantillonnées dans la queue de la distribution, ont une forte influence sur l'estimation des paramètres et lorsque la distribution est symétrique autour de la moyenne, les problèmes causés par les valeurs aberrantes peuvent être réduits à l'aide de techniques d'estimation [13] qui visent essentiellement à ignorer ou à réduire les valeurs aberrantes. La troisième difficulté est due de la relation entre les variables explicatives et la variable a expliquée (le montant attendu des sinistres). Cela a un effet important sur la méthodologie à utiliser, en particulier en ce qui concerne la tâche de sélection du modèle. Sans oublier le niveau général des sinistres qui preuvent infuencer une année à une autre, la non qualité des données.

La prime d'assurance est la somme que doit payer, l'assuré pour bénéficier des garanties prévues au contrat d'assurance, en cas de sinistre.

Nous abordons les méthodes utilisées en tarification non-vie, et plus spécifiquement en tarification automobile. Tout l'enjeu est de comprendre les apports du machine learning par rapport aux méthodes plus classiques utilisées. Il est pour cela indispensable de se pencher sur les grands principes de la tarification afin de mettre en exergue un cadre général commun à toutes les méthodes, permettant de les comparer.

1.1 Les techniques et principes de calcul de la prime

1.1.1 Les techniques de calcul de la prime

Elles sont en général effectuées par des actuaires en se basant sur les composantes telles que:

- La prime pure : c'est le montant du sinistre moyen auquel devra faire face l'assureur pour

le risque. Mathématiquement, la prime pure est égale à l'espérance des pertes.

- Le chargement de sécurité : ce montant vient s'ajouter à la prime pure. Il permet à l'assureur de pouvoir résister à la volatilité naturelle des sinistres.
- Le chargement pour frais de gestion. Ces frais comportent aussi bien les frais de gestion des sinistres que la rémunération des apporteurs (agents généraux ou courtier)
- Les taxes

ce calcul de la prime pure a pour but d'évaluer, pour chaque assuré ou prospect, le montant attendu des sinistres pour la période d'assurance étudiée. Cette évaluation se fait le plus souvent par des méthodes statistiques, fondées par exemple sur la technique du scoring. La sinistralité est divisée en plusieurs composantes, chacune étant évaluée indépendamment :

- La probabilité d'un sinistre normal
- Le coût d'un sinistre normal
- La probabilité d'un sinistre grave
- Le coût d'un sinistre grave

Définition 1.1.1. *[21] On notera X la variable aléatoire positive représentant le montant annuel du coût des sinistres pour un risque déterminé.*

X*, l'ensemble des risques(l'ensemble des variables positives).*

F*, l'ensemble des fonctions de probabilté.*

(Ω, F, P) *est l'espace de probabilité.*

Un principe de calcul de prime est une application:

$$P : \mathrm{X} \to [0, +\infty[, X \mapsto P(X). \tag{1.1.1}$$

1.1.2 Propiétés des primcipes de prime [21]

L'assureur souhaite avoir réalisé quelques propriétés par un principe de prime technique. Notons d'emblée qu'aucun principe de changement ne les satisfait toutes, chaucn ayant ses avantages et ses inconvénients. Soient X, Y, X, voir définition 1.1.1

1. **l'indépendance:**

   $P(X)$ dépend seulement de la fonction de repartition de X, notée F_X, tel que:

   $$F_X(t) = \mathbb{P}\{\omega \in \Omega; X(\omega) \leq t\} \tag{1.1.2}$$

2. **Chargement de risque:**

$$P(X) \geq \mathbb{E}(X) \tag{1.1.3}$$

Cette propiété est satisfaite dans tous les principes de prime. Toute fois, en pratique, l'assureur peut momentanément accepter un chargement négatif à cause des conditions du marché, de la concurrence ou pour des raisons commerciales.

3. **Aucun chargement injustifié de risque:**

Si le risque X est identiquement égal à une valeur constante $c \geq 0$ alors $P(X) = c$

4. **Perte maximale (no-rip-off):**

Le principe de prime de perte maximal est un cas limité. S X est non borné(illimité), alors la prime est infinie.

5. **Invariance par translation(Uniformité):**

$\forall a \geq 0, P(X + a) = P(X) + a$

6. **Homogénéité positive (Scale Invariance):**

$\forall b \geq 0, P(bX) = bP(X)$

7. **Additivité**

$$P(X + Y) = P(X) + P(Y) \tag{1.1.4}$$

8. **Sous-additivité:**

$$P(X + Y) \leq P(X) + P(Y) \tag{1.1.5}$$

9. **Sup-additivité:**

$$P(X + Y) \geq P(X) + P(Y) \tag{1.1.6}$$

10. **Monotonie:**

$\forall \omega \in \Omega, \exists X(\omega), Y(\omega)$, telque:

$$X(\omega) \leq Y(\omega) \Longrightarrow P(X) \leq P(Y) \tag{1.1.7}$$

11. **Preserves first stochastic dominante (FSD) ordering:**

$$\forall t \geq 0, F_X(t) \leq F_Y(t) \Longrightarrow P(X) \leq P(Y) \tag{1.1.8}$$

12. **Preserves stop-loss ordering (SL):**

$$\forall d \geq 0, \mathbb{E}(X - d)_+ \leq \mathbb{E}(Y - d)_+ \Longrightarrow P(X) \leq P(Y) \tag{1.1.9}$$

13. **Continuité:**

$\lim_{a \to 0^+} P(\max(X - a, 0)) = P(X)$, et $\lim_{a \to \infty} P(\min(X, a)) = P(X)$

1.1.3 Les principes de calcul de la prime

Nous donnons ci-après quelques principes de prime.

- Principe de la prime pure: $P(X) = E(X)$. Ce principe vérifie toutes les propriétés, cependant, le niveau de la prime apparaît comme insuffisant puisque nous ne tenons pas compte du risque.
- Principe de prime de l'Espérance mathématique: $P(X) = (1+\theta)\mathbb{E}(X)$ pour $\theta > 0$. θ est le coéfficient du chargement ou le taux de chargement. le taux du chargement θ peut varier en fonction des caractéristiques des assurés.
- Principe de la variance: $P(X) = \mathbb{E}(X) + \alpha Var(X)$ avec $\alpha > 0$. Ce principe ne vérifie pas l'homogénéité.
- Principe de l'écart-type: $P(X) = \mathbb{E}(X) + \alpha\sigma(X)$ avec $\alpha > 0$. Cette fois-ci l'homogénéité est satisfaite mais la sous-additivité ne l'est plus en général même pour des variables aléatoires indépendantes. On remarque que la prime n'est pas toujours inférieure à la perte maximiale on faisant la même construction que pour le principe de la variance (pour certains α).

1.1.3.1 L'approche technique

L'approche actuarielle classique pour la modélisation des portefeuilles d'assurance non-vie utilise une variable aléatoire composée, N décrivant le nombre de sinistres survenant au cours d'une période fixe et Z période de temps fixe et $Z_1, \ldots, Z_N$ décrivant les tailles des sinistres individuels. Le montant total des sinistres au cours de cette période fixe est alors donné par

$$S = Z_1 + \cdots + Z_N = \sum_{k=1}^{N} Z_k. \tag{1.1.10}$$

La principale tâche de la modélisation de **l'assurance non-vie** est de comprendre la structure de ces modèles de montant total des sinistres.

Hypothèses du modèle .

Nous considerons un espace de probabilité filtre (Ω, P, F) comme espace de travail avec les trois hypothèses suivantes:

1. N est une variable aléatoire discrète qui prend des valeurs dans $\mathbb{N}_0$;
2. $Z_1, Z_2, \ldots$ sont indépendants et identiquement distribués (i.i.d.) ;
3. N et $(Z_1, Z_2, \ldots)$ sont indépendants.

Remarque. *1. Si S satisfait aux trois hypothèses standard $(1)-(3)$ des hypothèses du modèle, nous disons que S a une distribution composée. N est appelé nombre de sinistres et $Z_k, k \geq 1$, sont les tailles des sinistres individuels ou les sévérités des sinistres.*

2. *La distribution composée a été largement étudiée dans la littérature actuarielle. L'objectif ici est de donner plus de structure au problème afin qu'il puisse être utilisé pour répondre à des questions complexes de tarification sur des portefeuilles d'assurance hétérogènes.*

1.1.4 Modèle classique de risque

Les réserves des compagnies d'assurance sont calculés par la formules suivantes:

$$R_t(x) = x + ct - \sum_{k=1}^{N_t} Y_k, t \geq 0 \tag{1.1.11}$$

R_t modélise (Cramer-Lundberg modèle ou modèle classique de risk) réserves des compagnies d'assurance en temps $t \geq 0$. Où x est capital initial, ct est le total des primes que l'assureur reçoit pas unité de temps, N_t est un processus de Poisson d'intensité λ.

Y_k est un processus de variables i.i.d avec la fonction de répartition $F_Y(y) = \mathbb{P}[Y_k \leq y]$ et d'espérence $\mathbb{E}(Y_k) = \mu$.

On definit le temps de ruine par:

$$\tau(x) = \inf\{t \geq 0; R_t < 0\} \tag{1.1.12}$$

avec $\tau(x) = \infty$ si $R_t(x) \geq 0, \forall t \geq 0$.

La probabilté de ruine est:

$$\psi(x) = \mathbb{P}(\tau(x) < \infty) = \mathbb{P}[\inf_{s>0} R_s < 0] \tag{1.1.13}$$

L'assureur s'intéresse à minimiser cette probabilté par un ensemble d'actions dont la prime c.

Dans le modèle classique, la probabilté de ruine depend de la comparaison entre c et $\lambda\mu$.

Ceci est resumé dans le théorème suivant:

Théorème 1.1.1. *Considérons le modèle de risque classique de Cramer-Lundberg (1).*

1. *Si* $c < \lambda\mu$, *alors* $\psi(x) = 1$ *pour tout x.*
2. *Si* $c > \lambda\mu$, *alors* $\lim_{x\to\infty} \psi(x) = 0$

Remarque. *Ce théorème indique que si le taux de prime effectivement chargé est inférieur à l'espérance des risques agrégés alors la ruine est inévitable pour l'assureur quelque soit le niveau de capital. Et si* $c > \lambda\mu$, *on peut agir sur l'éventualité de faillite en augmentant le capital x.*

Preuve.

Soit τ_i le temps d'arrivé du $i^{ième}$ sinistre, alors les temps jusqu'au prochain sinistre $\tau_i - \tau_{i-1}$ suivent une loi exponentielle d'espérance $\frac{1}{\lambda}$.

Soit $Z_i = Y_i - c(\tau_i - \tau_{i-1}), i \geq 1$.

Alors $(Z_i)_i$ est un processus de variables i.i.d avec $\mathbb{P}(Z_i > 0) \neq 0 \neq \mathbb{P}(Z_i < 0)$ et $\mathbb{E}(Z_i) = \mu - \frac{c}{\lambda} < \infty$.

Soit $S_n = \sum_{i=1}^{n} Z_i$ et $M = \sup_{t\geq 0}(\sum_{i=1}^{N_t} Y_i - ct)$

Alors $M = \sup_{n\geq 1} S_n$ et $\psi(x) = \mathbb{P}(M > x)$.

1) Si $c < \lambda\mu$ alors $\mathbb{E}(Z_i) > 0$ et par la loi des grands nombres on a

$$\mathbb{P}[\lim_{n\to\infty} \frac{1}{n} S_n = \mu - \frac{c}{\lambda}] = 1 \Rightarrow \mathbb{P}[\lim_{n\to\infty} S_n = +\infty] = 1 \Rightarrow \psi(x) = 1 \textbf{pour tout x}.$$

Car $M = \sup_{n\geq 1} S_n = +\infty \Rightarrow \mathbb{P}(M > x) = 1$

2) Si $c > \lambda\mu$ alors $\mathbb{E}(Z_i) < 0$ et $\mathbb{P}(\lim n \to \infty S_n = -\infty) = 1$ par la loi des grands nombres. Donc $\mathbb{P}(M < \infty) = 1$ donc $\lim x \to \infty \mathbb{P}(M > x) = 0$. C'est-à-dire $\lim x \to \infty \psi(x) = 0$ ■

La condition $c > \lambda\mu$ est appelée la **condition de profit net** qui est crucial pour l'assurance.

ce théorème justifit le principe de l'espérance pour la détermination de la prime, avec le choix $c = \lambda\mu(1 + \theta)$, où $\theta > 0$ est une charge additionnelle de severité.

La théorie de la ruine est un champ actif de recherche en mathématiques actuarielles et financières.

1.2 Les techniques traditionnelles de la tarification

1.2.1 Modèle linéaire généralisé (GLM)

Pendant longtemps, les actuaires se sont limités au modèle linéaire gaussien classique. Toutefois la complexité des problèmes statistiques les ont poussés vers des méthodes plus complètes, les modèles linéaires généralisés. À titre d'exemple, la régression de Poisson est devenue la méthode la plus répandue en tarification automobile.

On rappelle la segmentation classique que l'on retrouve lorsque l'on veut effectuer des régressions:

1. Variables à expliquer/réponses/endogènes: Nombre de sinistres, montant de sinistres, probabilité d'avoir au moins un sinistre, etc.
2. Variables explicatives/prédictives/exogènes : âge de l'assuré, Cadre Socio Professionnel de l'assuré, type de véhicule, zone géographique, etc.

Dans le cas de GLM, on peut présenter la figure 1.1 récapitulatif suivant:

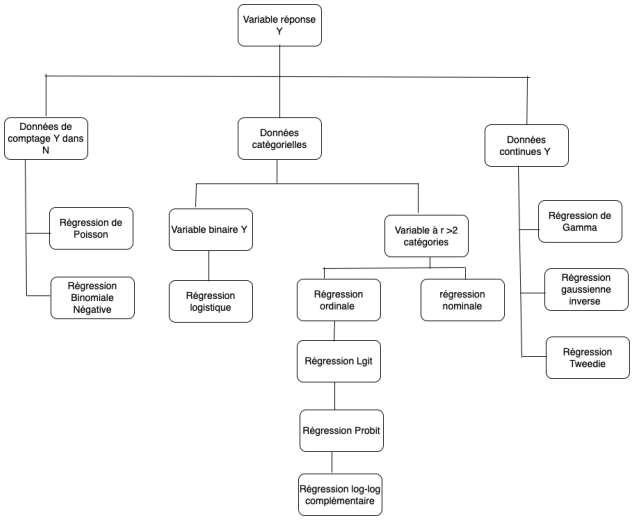

Figure 1.1: Schéma classique de choix de régression pour un GLM

Le modèle linéaire généralisé se distingue en trois composantes [19]:

1. **Composante aléatoire:** La variable à expliquer est $Y = (Y_1, \ldots, Y_n)'$ dont les densités appartiennent à la loi famille exponentielle. On dit que f_{Y_i} appartient à la loi famille exponentielle si et seulement si on peut trouver $\theta \in \mathbb{R}$ (paramètre canonique, ou de la moyenne), $\phi \in \mathbb{R}$ (paramètre de dispersion), a fonction définie sur $\mathbb{R}$ non nulle, b fonction définie sur $\mathbb{R}$ deux fois dérivable, c fonction définie sur $\mathbb{R}^2$, tels que:

$$f(y;\theta,\phi) = \exp\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y;\phi)\}. \tag{1.2.1}$$

   La densité d'une loi de Poisson satisfait ces critères, elle est donnée par:

$$f(y) = \frac{e^{-\mu}\mu^y}{y!}, \tag{1.2.2}$$
$$= \exp\{y\log\mu - \mu - \log(y!)\}, \tag{1.2.3}$$
$$= \exp\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y;\phi)\}. \tag{1.2.4}$$
$$\tag{1.2.5}$$

   avec $\phi = 1, a(\phi) = 1, c(y;\phi) = c(y) = -\log(y!), \theta = \log\mu, b(\theta) = \mu = e^\theta$

2. **Composante déterministe:** Soit $i = 1, \ldots, n$, alors pour chaque Y_i on dispose de la valeur d'un p-uplet $(X_{1i}, \ldots, X_{pi})'$, des p variables explicatives décrivant Y_i. Les vecteurs $X_j = (X_{j1}, \ldots, X_{jn})'$ pour $j = 1, \ldots, p$ sont les vecteurs explicatifs considerées non aleatoires.

3. **Fonction de lien:** C'est une fonction g déterministre strictement monotone définie sur $\mathbb{R}$ et telle que:

$$g_n\Big(\underbrace{\mathbb{E}[Y]}_{\mu}\Big) = \underbrace{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}_{\eta:score} = X\beta. \tag{1.2.6}$$

avec $g_n : \mathbb{R}^n \to \mathbb{R}^n, (x_1, \ldots, x_n) \mapsto \Big(g_1(x_1), \ldots, g_n(x_n)\Big)$.

On a avec nos notations:

$$g_n\Big(\underbrace{\mathbb{E}[Y_i]}_{\mu_i}\Big) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p = x_i^{'}\beta = \eta_i. \tag{1.2.7}$$

Chacune des lois de probabilités de la famille exponentielle possède une fonction de lien spécifique,dite $\ll$ canonique $\gg$, et définie par $\theta = \eta$. Le lien canonique est tel que $g(\mu_i) = \theta_i$ or on sait que $\mu_i = b'(\theta_i)$ ainsi formellement $g^{-1} = b'$. La fonction de lien canonique d'une loi de Poisson est la fonction logarithmique.

Dans le contexte de l'assurance automobile, nous allons mettre en œuvre **le modèle de Poisson et nous choisissons la fonction de lien canonique** $g = \log$.

Loi de probabilité	Nom de la Fonction	Fonction de lien canonique
Normale	Fonction identité	$\mathsf{g}(\mu) = \mu$
Poisson	Fonction logarithme	$\mathsf{g}(\mu) = \log \mu$
Gamma	Fonction inverse	$\mathsf{g}(\mu) = \dfrac{1}{\mu}$
Binomiale	Fonction logit	$\mathsf{g}(\mu) = \log \mu - \log(1-\mu)$

Table 1.1: Fonctions de liens.

1.2.1.1 Estimation des paramètres

L'estimation des paramètres β_k et ϕ se fait par la méthode du maximun de vraissemblance. A cet égard, la vraissemblance d'un n-échantillon de réalisation de Y s'écrit:

$$L(y_1, \ldots, y_n) = \prod_{i=1}^{n} \exp\{\frac{y_i\theta - b(\theta)}{a(\phi)} + c(y_i;\phi)\}. \tag{1.2.8}$$

Les estimateurs de b et ϕ sont obtenus en maximisant la logvraissemblance:

$$\log L = \sum_{i=1}^{p}\{y_i b^{'-1}[g^{-1}\Big(\sum_{i=1}^{p}\beta_k x_k\Big)\phi^{-1} - bb^{'-1}[g^{-1}\Big(\sum_{i=1}^{p}\beta_k x_k\Big)\phi^{-1} + c(y_i;\phi)\}. \tag{1.2.9}$$

les tableaux suivants donnent les valeurs des différents paramètres pour certaines lois discrètes et continues.

Loi de probabilité	Pr(Y=y) $=\exp\{y\log\frac{p}{1-p}+\log(1p)\}$
θ	$\log\frac{p}{1-p}$
ϕ	1
b(θ)	0
E[Y]	p
Fonction variance	V(p)=p(1-p)

Table 1.2: Estimation des paramètres d'une loi de bernoulli B(p).

Loi de probabilité	Pr(Y=y) $=\exp\{y\log\lambda-\lambda+c(y)\}$
θ	$\log\lambda$
ϕ	1
b(θ)	$\exp\theta$
E[Y]	λ
Fonction variance	V(λ) $=\lambda$

Table 1.3: Estimation des paramètres d'une loi de poisson P(λ).

Densité	f(y) $=\exp\{(\mu y\frac{\mu^2}{2})\frac{1}{\sigma^2}+c(y,\sigma^2)\}$
θ	μ
ϕ	σ^2
b(θ)	$\sigma^2/2$
E[Y]	μ
Fonction variance	V(μ) $=1$

Table 1.4: Estimation des paramètres d'une loi de normale N(μ,σ).

Densité	f(y) $=\exp\{-(\frac{y}{\mu}+\log\mu)+c(y,\nu)\}$
θ	-1/μ
ϕ	1/ ν
b(θ)	(-1/θ)
E[Y]	μ
Fonction variance	V(μ) $=\mu^2$

Table 1.5: Estimation des paramètres d'une loi de gamma Gam($\nu,\mu/\nu$).

1.2.1.2 Qualité d'ajustement, intervalle de confiance et limites de ce modèle

Il s'agit d'évaluer la qualité d'ajustement du modèle sur la base des différences entre observations et estimations. Plusieurs critères sont proposés.

1. **Déviance:** Le modèle estimé est comparé avec le modèle dit saturé, c'est-à-dire le modèle possédant autant de paramètres que d'observations et estimant donc exactement les données.

Cette comparaison est basée sur l'expression de la déviance D des log-vraisemblances $L = L(y, \hat{\mu})$ et $L_{sat} = L(y, y)$, il y a un complet overfitting.

$$D = 2(L - L_{sat}), \tag{1.2.10}$$
$$= 2(L(y, \hat{\mu}) - L(y, y)), \tag{1.2.11}$$
$$= 2\sum_{i=1}^{n}(-\hat{\mu}_i + y_i - y_i \log \frac{y_i}{\hat{\mu}}). \tag{1.2.12}$$

qui est le logarithme du carré du rapport des vraisemblances. Ce rapport remplace ou généralise l'usage des sommes de carrés propres au cas gaussien et donc à l'estimation par moindres carrés.

On montre qu'asymptotiquement, D suit une loi du χ^2 à $n - p$ degrés de liberté ce qui permet de construire un test de rejet ou d'acceptation du modèle selon que la déviance est jugée significativement ou non importante. Attention, l'approximation de la loi du χ^2 peut être douteuse. Cet indicateur global est en pratique complété par une analyse observation par observation ; cette analyse se base souvent sur l'analyse des résidus[8].

2. **L'analyse des résidus:**

   La déviance fournit des indications globales sur la qualité du modèle. L'analyse des résidus est éssentielle pour vérifier l'adéquation des modèles en ce qui concerne le choix de la fonction de variance, de la fonction de lien ou des termes du prédicteur linéaire. Les résidus permettent également de déterminer la présence des valeurs aberrantes.

3. **Le résidu de déviance:**

   Les résidus peuvent être calculés de différentes manières. Les deux principales sont les résidus de Pearson et les résidus de déviance:

   (a) **Résidus de Pearson**

   $$r_i^p = \sqrt{w_i}\frac{y_i - \mu_i}{\sqrt{V(\mu_i)}}. \tag{1.2.13}$$

   (b) **Résidus de déviance**

   $$r_i^p = \epsilon(y_i - \mu_i)\sqrt{d_i}. \tag{1.2.14}$$

   On peut noter que la somme des carrés des résidus est dans ce cas, asymptotiquement, un Khi2 à $n - p - 1$ degrés de liberté.

1.2.1.3 Choix d'un modèle GLM – Critères AIC et BIC

La comparaison de 2 ou plusieurs modèles nécessite de tenir compte de la complexité de chaque modèle. Les critères **AIC** (Akaike information criterion) et **BIC** (Bayesian information criterion) pénalisent la log-vraisemblance du modèle avec le nombre de paramètres:

$$AIC = 2 \times (p - \ln L(\hat{\mu}|y)), \tag{1.2.15}$$
$$BIC = -2 \times \ln L(\hat{\mu}|y) + p \times \ln(n)). \tag{1.2.16}$$

L'AIC est asymptotiquement optimal lorsque l'on souhaite sélectionner le modèle avec l'erreur quadratique moyenne, si l'on fait l'hypothèse que le modèle générant les données n'est pas parmi les candidats, ce qui est en fait presque toujours le cas en pratique. Ce n'est pas le cas du BIC. Ici, la vitesse de convergence de l'AIC vers l'optimum est, dans un certain sens, la meilleure possible [26].

À partir de ces critères, on peut imaginer des processus de sélection des variables à prendre en compte dans le modèle:

1. on part du modèle avec seulement la constante et on ajoute la variable qui conduit à la plus forte baisse de l'AIC ou du BIC à chaque étape ; on s'arrête lorsque l'indicateur ne baisse plus (sélection ascendante);
2. on part du modèle avec toutes les variables et on effectue des suppressions pas à pas (sélection descendante) ;
3. un mélange des deux techniques ci-dessus (sélection ascendante avec possibilité de suppression d'une variable déjà sélectionnée).

Intervalle de confiance des estimateurs des paramètres du GLM

La détermination d'intervalle de confiance pour les paramètres β_k,(k allant de 1 à n) est important dans le calcul de la prime pure. Il permet d'apprécier la marge d'erreur entre les valeurs observées et les valeurs estimées pour un niveau de seuil donné. Plusieurs méthodes sont disponibles pour ces intervalles de confiance entre autres nous avons la méthode du rapport de vraisemblance et la méthode de Wald. Ici nous utiliserons l'approximation normale des coefficients $\hat{\theta} \sim N(\theta, I^{-1})$ afin d'obtenir un intervalle de confiance au niveau $1-\alpha$ pour θ_k donné. On a donc pour un θ_k trouvé

$$IC = [\theta_k \pm \mu_{\alpha/2}\sqrt{I^{-1}}]. \tag{1.2.17}$$

I étant la matrice d'information de Fisher.

Remarque. *Lorsque la variable à expliquer dans le cas d'un modèle linéaire généralisé dépend également linéairement d'une autre variable, cette dernière est déclarée offset. Exemple, pour modéliser le nombre de sinistres déclarés par catégorie de conducteurs, la variable exposition est déclarée offset.*

1.2.1.4 Limites du GLM

Le modèle GLM est dit paramétrique, en effet il nécessite de préciser une loi pour la variable d'intérêt. Le modèle est de plus linéaire et donc l'impact des variables explicatives également. La littérature propose les modèles additifs généralisés (GAM). Les modèles sont parfois longs à s'exécuter et l'on ne peut se permettre de tester toutes les interactions envisageables pour conserver le meilleur modèle. Pour toutes ces raisons, nous étudions dans la suite du travail des méthodes non paramétriques, c'est-à-dire les méthodes de machine learning.

2. Les méthods de machine learning pour la tarification.

2.1 Généralités sur la tarification

Les compagnies d'assurances utilisent quotidiennement des modèles statistiques pour évaluer les risques auxquels elles doivent faire face. En particulier, les modèles de régression permettent de quantifier les relations entre la valeur des contrats des risques assurés et les variables décrivant ce risque [25]. Vue l'évolution massive des données, alors les entreprises d'assurance commencent à s'interroger sur ce modèle et se migrent sur des nouveaux modèles de machine learning. Ces derniers performent alors sur des données de tout type (structurées ou non) et de différentes volumétries [19, 25]. Ces modèles permettent à la fois de modéliser des comportements non linéaires et des distributions de résidus non gaussiens. Cela est particulièrement utile en assurance non-vie où les coûts des sinistres, quand ils se concrétisent, suivent une densité très asymétrique clairement non gaussienne. Ils ont permis d'améliorer la qualité des modèles de prédiction du risque et sont aujourd'hui largement utilisés par les compagnies d'assurance.

L'objectif de cette section est de montrer les enjeux mathématiques et économiques de la tarification par le machine learning.

2.2 Machine learning

Le machine learning est un domaine de recherche en informatique qui est progressivement adopté dans l'analyse des données, et il a gagné une forte demande au cours de la dernière décennie [11]. Sa diffusion rapide est due à la croissance rapide des données générées par de nombreuses sources et aux grandes quantités de données stockées dans les bases de données. Elle permet aux individus et aux organisations de comprendre leurs ensembles de données de manière plus détaillée.

2.2.1 Types de Machine Learning (ML)

Les algorithmes de Machine Learning (ML) peuvent être classés en fonction des types de données. Quatre types de ML sont décrits ci-dessous:

1. **L'apprentissage supervisé**: utilise des données étiquetées pour prédire une étiquette future, souvent appelée sortie ou cible. Il existe deux types de problèmes dans l'apprentissage automatique, à savoir les problèmes de régression et de classification. Le problème de régression se pose lorsque la cible est continue et le problème de classification se pose lorsque la cible est catégorique.

2. **L'apprentissage non supervisé**: Il permet d'apprendre des modèles à partir de données qui ne sont ni étiquetées ni classées. Il s'agit de données non structurées qui peuvent être regroupées ou associées à des types de choses similaires.

3. **L'apprentissage semi-supervisé**: Est une combinaison de l'apprentissage supervisé et non supervisé. Il utilise une petite quantité de données étiquetées avec une grande quantité de données non étiquetées pendant la formation pour ajuster un modèle. Avec cette méthode, la précision du modèle déployé est grandement améliorée.

4. **L'apprentissage par renforcement** interagit directement avec l'environnement et apprend par expérience. Il travaille par essais et erreurs pour prendre les mesures appropriées afin de maximiser la récompense dans une situation particulière.

La prédiction des risques ou sinistre est un problème d'apprentissage supervisé, plus précisément un problème de classification ou de regréssion. La figure 2.1 présente une illustration de l'application des types de ML dans la résolution de divers problèmes en économie, l'agriculture, les affaires, la technologie, etc.

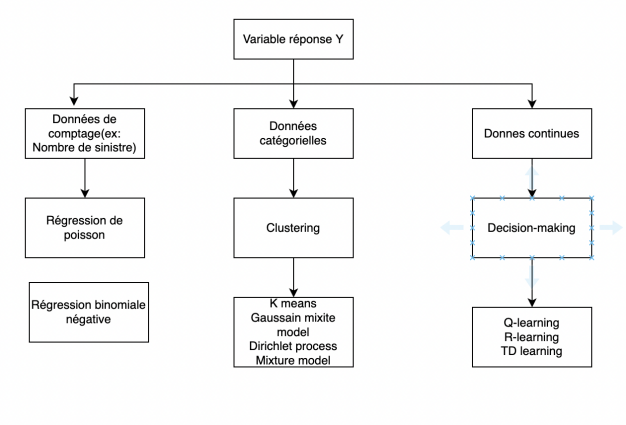

Figure 2.1: Types de machine learning et leurs differents types d'algorithme

2.3 Les méthodes utilisées dans ce travail.

2.3.1 Présentation du modèle Arbre de décision CART [12]

Les arbres de décision CART (Classification And Regression Trees) font partie de la famille des modèles supervisés non paramétriques. Leur utilisation peut se faire, soit en cas de classification, soit pour la régression.

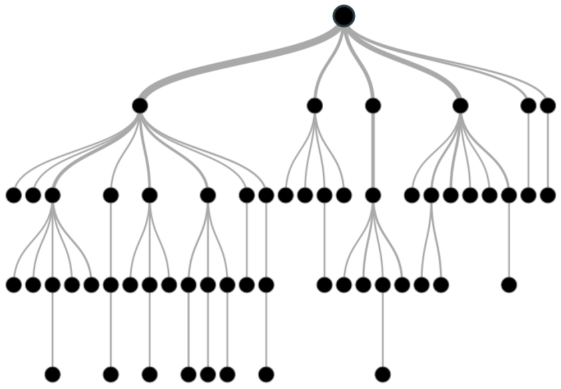

Figure 2.2: Schéma d'un arbres de décision

- **Root Nodes (Nœuds racines):** C'est le nœud présent au début d'un arbre de décision à partir de ce nœud, la population commence à se diviser selon diverses caractéristiques.
- **Decision Nodes (Nœuds de décision)**-les nœuds que nous obtenons après avoir divisé les nœuds racine sont appelés nœuds de décision
- **Leaf Nodes (Nœuds feuilles)** -les nœuds où une division supplémentaire n'est pas possible sont appelés nœuds feuilles ou nœuds terminaux
- **Sub-tree (Sous-arbre)**- tout comme une petite partie d'un graphique est appelée sous-graphe, de même, une sous-section de cet arbre de décision est appelée sous-arbre.
- **Pruning (Élagage)**- n'est rien d'autre que couper certains nœuds pour arrêter le sur-ajustement.

L'idée de base de ce modèle est simple : disposant de n variables explicatives $X_1, X_2, \ldots, X_n$ et d'une variable à expliquer Y (un réel dans le cas de la régression, une classe dans le cas de la classification), on cherche à prédire Y connaissant $X_1, X_2, \ldots, X_n$. L'algorithme va alors créer un arbre binaire, qui va séparer la population en deux sous-populations à chaque nœud selon divers critères. Par exemple, l'algorithme va chercher à maximiser la variance interclasses dans le cas d'un arbre de régression.

A chaque nœud, l'algorithme va tester si l'une des conditions d'arrêt est vérifiée (profondeur maximale, nombre d'individus minimal, etc.). Si ce n'est pas le cas, l'algorithme sépare à nouveau

la population en deux sous-populations. En revanche, si la condition est vérifiée, l'algorithme calcule la moyenne de la variable Y (ou la classe majoritaire dans le cas de la classification) pour les individus présents dans cette feuille. Ainsi, lorsque l'on voudra utiliser cet arbre pour prédire Y , les individus qui seront classés dans cette feuille se verront attribuer cette moyenne comme prédiction.

L'algoritme se résume en cinq étapes:

- **Étape 1**: Commencez l'arborescence avec le nœud racine, dit S, qui contient l'ensemble de données complet.
- **Étape 2:** Trouvez le meilleur attribut dans l'ensemble de données à l'aide de la mesure de sélection d'attribut.
- **Étape 3:** Divisez le S en sous-ensembles contenant les valeurs possibles pour les meilleurs attributs.
- **Étape 4:** Générez le nœud de l'arbre de décision, qui contient le meilleur attribut.
- **Étape 5:** Créez de nouveaux arbres de décision de manière récursive à l'aide des sous-ensembles de l'ensemble de données créé à l'étape 3. Continuez ce processus jusqu'à ce qu'un stade soit atteint où vous ne pouvez plus classer les nœuds et appelé le nœud final en tant que nœud feuille.

Les arbres de régression présentent de nombreux avantages : entre autres, ils sont simples à interpréter, rapides à mettre en place et assez économiques en temps de calcul.

Toutefois, ils sont très sensibles à l'échantillon d'apprentissage et peuvent avoir tendance à surapprendre. Ils ont alors un faible biais, mais une variance très grande. C'est pour réduire cette dernière que l'on utilise des forêts aléatoires. L'utilisation de forêts aléatoires s'accompagne d'une perte de lisibilité du modèle, mais d'une amélioration des performances.

2.3.2 Random forest

La forêt aléatoire ou Random Forest définie est composée de divers arbres binaires CART. L'idée de cette méthode est de retenir les avantages de la méthode CART tout en palliant ses limites, principalement l'effet de sur-ajustement du modèle et de ce fait, la complexité de la phase d'élagage.

Les algorithmes Random Forest sont connus pour leur robustesse et leur flexibilité. Ils sont basés sur des techniques d'échantillonnage.

Son objectif principal est de rajouter de l'aléa dans la construction des arbres de décison CART. C'est-à-dire, à partir de l'échantillon d'apprentissage contenant p variables prédictives, nous procédons à la création de N échantillons obtenus par tirage avec remise des observations et de même taille que l'échantillon initial. Sur chacune de ces N bases, nous construisons les N arbres différents sans élagage.

Ensuite, nous déterminons au hasard un nombre $k < p$ qui représentera le nombre de variables utilisées pour la segmentation de chaque arbre. Le nombre de variables prédictives k a pour but de réduire la variance du modèle final. Cette valeur k varie en fonction du type de variable. La prédiction de la variable cible Y est obtenue:

- En classification : $\hat{Y}=$ classe la plus représentée parmi les B arbres.
- En cas de régression : $\hat{Y}=$ moyenne empirique des valeurs prédites avec les B arbres.

Hormis la prédiction de la variable réponse Y , l'utilisation des forêts aléatoires fournit d'autres informations qui sont :

- l'estimation de l'erreur de prédiction

  À la différence des estimateurs qui requièrent en général une validation croisée ou une division de la base de données en entraînement-validation, l'estimation de l'erreur tient compte de l'ensemble des observations qui n'ont pas été prises en compte lors de la construction de l'échantillon bootstrap.

  Soit δ_B, l'ensemble des arbres de décisions issus des observations non présentes dans l'échantillon bootstrap. La prédiction de l'observation y_i sur cet ensemble est :

$$\hat{y}_i = \frac{1}{|\delta_B|} \sum_{i \in \delta_B} f(x_i). \tag{2.3.1}$$

  avec f représentant la fonction prédictive de chaque arbre. Ainsi, l'erreur de prédiction est alors notée :

$$\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2. \tag{2.3.2}$$

L'algorithme se résume en 4 étapes.

1. **Étape 1:** Dans la forêt aléatoire, un nombre n d'enregistrements aléatoires est extrait de l'ensemble de données ayant un nombre k d'enregistrements.
2. **Étape 2:** Des arbres de décision individuels sont construits pour chaque échantillon.
3. **Étape 3:** Chaque arbre de décision générera une sortie.
4. **Étape 4:** Le résultat final est considéré sur la base du vote à la majorité ou de la moyenne pour la classification et la régression respectivement.

Nous pouvons citer entre autre les avantages liés à cet algorithme, à savoir:

- Cet algorithme permet de pallier l'effet du sur-apprentissage du modèle;

- il permet aussi de donner un aperçu du pouvoir prédictif et permet de fournir une première idée des variables importantes à utiliser pour le modèle lors de l'analyse;
- c'est l'un des modèles les plus efficaces en termes de précision des valeurs prédites.

Mais, il présente des limites comme tout modèle:

- L'utilisation de cet modèle nécessite des ordinateurs hautement performants pour un gain de temps.
- La prédiction des variables extrêmes peut être souvent instable lors d'une régression.

2.3.3 Les Réseaux de Neurones

Les réseaux de neurones, communément appelés des réseaux de neurones artificiels sont des imitations simples des fonctions d'un neurone dans le cerveau humain pour résoudre des problématiques d'apprentissage de la machine (Machine Learning). Fondamentalement, un réseau de neuronne est composé de trois éléments qui sont : une entrée, une fonction d'activation et une sortie. La figure 2.3 présente une analogie entre les neurones biologiques et un neurone artificiel.

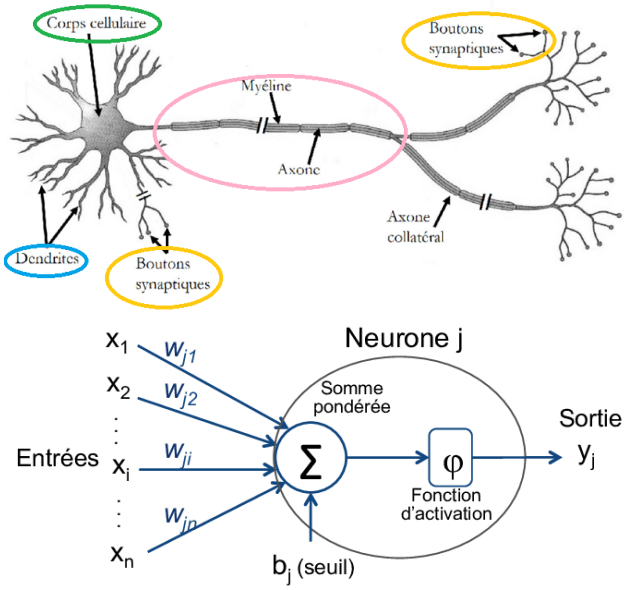

Figure 2.3: Analogie entre les neurones biologiques et un neurone artificiel

Les dendrites du neurone biologique représentent les entrées du neurone artificiel et le corps cellulaire, constitué par la fonction d'activation, envoie un signal aux autres neurones.

1. **Entrée:** Elle représente un tableau matriciel X de plusieurs types de données comme une image, un signal audio, une data frame, etc. Ces valeurs d'entrée sont ensuite une

combinaison linéaire de ces dernières avec une matrice de poids W et un vecteur de biais. Le poids est le paramètre supplémentaire d'un réseau neuronal qui transforme les données d'entrée dans les couches cachées du réseau. Il décide de la vitesse à laquelle la fonction d'activation se déclenchera. Le biais est un paramètre du réseau neuronal qui aide le modèle à s'adapter au mieux aux données.

2. **La fonction d'activation:** Une fonction d'activation $\phi : \mathbb{R} \to \mathbb{R}$. Cette fonction doit être dans l'idéal monotone, dérivable.

   Et on appellera neurone toute application $f_\omega(x) = \phi(\omega^T x)\ \forall x \in \mathbf{X}$.

   Voici une liste de fonctions d'activation ϕ définies sur $\mathbb{R}$ courament utilisées pour les réseaux de neurones.

   - **Sigmoïde:** produit une courbe en forme de S. Bien que de nature non linéaire, il ne tient toutefois pas compte des légères variations des entrées, ce qui entraîne des résultats similaires.
   - **Fonctions de tangente hyperbolique (tanh):** Il s'agit d'une fonction supérieure comparée à Sigmoid. Cependant, elle rend moins bien compte des relations et elle est plus lente à converger.
   - **Unité linéaire rectifiée (ReLu):** Cette fonction converge plus rapidement, optimise et produit la valeur souhaitée plus rapidement. C'est de loin la fonction d'activation la plus populaire utilisée dans les couches cachées.
   - **Softmax:** utilisé dans la couche de sortie car il réduit les dimensions et peut représenter une distribution catégorique.

   Elles sont définis respectivement par les figures suivantes:

Fonction	Définition	Image
Tanh	$x \to \frac{e^x - e^{-x}}{e^x + e^{-x}}$	[-1,1]
Sigmoïde,$\alpha \in \mathbb{R}^*$	$x \to \frac{1}{1 + e^{-\alpha x}}$	[0,1]
Arctan	$x \to arctan(x)$	$]-\frac{\pi}{2}, \frac{\pi}{2}[$

3. **Sortie:** C'est le résultat numérique après l'ajustement de l'entrée dans la fonction d'activation.

Les Deep Neural Network, ou réseaux de neurones profond (DNN) sont un type d'ANN construit à partir de couches multiples reliant l'entrée à la sortie. Les trois blocs de construction du DNN sont : la couche d'entrée, la couche cachée et la couche de sortie.

Le schéma suivant nous décrit l'architecture un réseau de neurone:

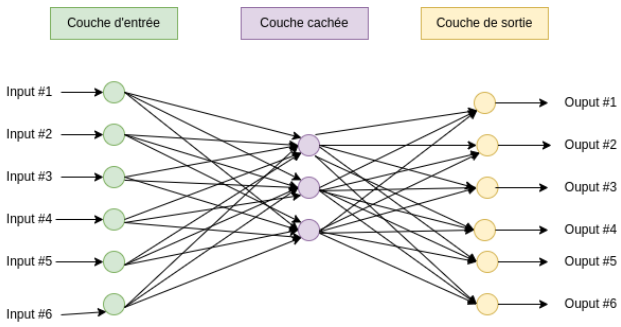

Figure 2.4: Architecture d'un neurone artificiel

La figure 2.4 nous montre les couches d'entrées qui transmetent les données initiales aux couches cachées à travers les connexions (les flèches en noir) entre elles. Après traitement, les couches cachées transmettent le résultat à l'aide de connexions à la couche de sortie.

1. **La couche d'entrée**: Elle transmet les données initiales aux couches cachées à travers les connexions (les flèches en noir) entre elles. Après traitement, les couches cachées transmettent le résultat à l'aide de connexions à la couche de sortie.

2. **Les Couche cachée:** Les couches cachées effectuent des transformations non linéaires des données transmises par la couche d'entrée pour produire la sortie. Les calculs effectués par les couches cachées et les fonctions d'activation utilisées dépendent du type de réseau neuronal utilisé, qui dépend à son tour de l'application.

3. **Couche de sortie :** La couche de sortie d'un réseau de neurone artificiel est la dernière couche de neurones responsable du résultat final.

En Deep Learning , l'objectif de l'utilisation de cette méthode est de trouver l'ensemble des paramètres (ω, b) qui prédisent le mieux la sortie. Ceci est effectué sur un processus d'apprentissage en trois étapes:

La première étape est basée sur l'approche de la propagation vers l'avant. C'est-à-dire que l'entrée est ajustée dans la couche d'entrée du DNN et une sortie est obtenue comme définie dans l'équation 2.3.3.

$$\hat{Y}^{(l)} = \sigma(\omega^{(l)} X + b^{(l)}) \tag{2.3.3}$$

Où $\hat{Y}^{(l)}, \omega^{(l)}, b^{(l)}$ sont respectivement la sortie, le poids et le biais de la couche l.

La deuxième étape consiste à comparer la sortie calculée à la sortie réelle au moyen d'une fonction de coût (fonction de perte). Cross-entropy ou Log loss et Mean Squared Error (MSE) constituent les deux principaux types de fonctions de perte, lorsque nous formons des modèles de réseaux de neurones. Elles sont représentées dans l'équation 2.3.5 (pour le problème de classification) et

dans l'équation 2.3.5 (pour le problème de régression).

$$Cross - entropy = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m} Y_{ij}.log(\hat{Y}_{ij}) \tag{2.3.4}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_i)^2 \tag{2.3.5}$$

Où N et m sont respectivement le nombre d'entrées et le nombre de classes differentes. Y_{ij} et $\hat{Y}_{ij}$ sont respectivement la valeur rélle et la valeur prédite pour la ième entrée d'entrée et la jième classe.

La dernière étape consiste à utiliser des optimiseurs pour mettre à jour les poids et le biais. Les optimiseurs sont des algorithmes utilisés pour modifier certains paramètres du réseau neuronal, tels que les poids et le taux d'apprentissage, afin de minimiser les fonctions de perte et de fournir des résultats précis. Le taux d'apprentissage est un paramètre d'ajustement ou un hyper paramètre qui détermine la taille du pas à chaque itération tout en se déplaçant vers un minimum d'une fonction de perte. L'optimiseur le plus populaire utilisé dans les réseaux neuronaux est la descente de gradient, présentée dans l'équation 2.3.6

$$W_{\text{mise à jour}} = W_{\text{précédent}} - \lambda \nabla L(W) \tag{2.3.6}$$

Où L est la fonction loss et λ est le taux d'apprentissage. L'ensemble de ces trois étapes est appelé Back Propagation résumée en figure 2.5.

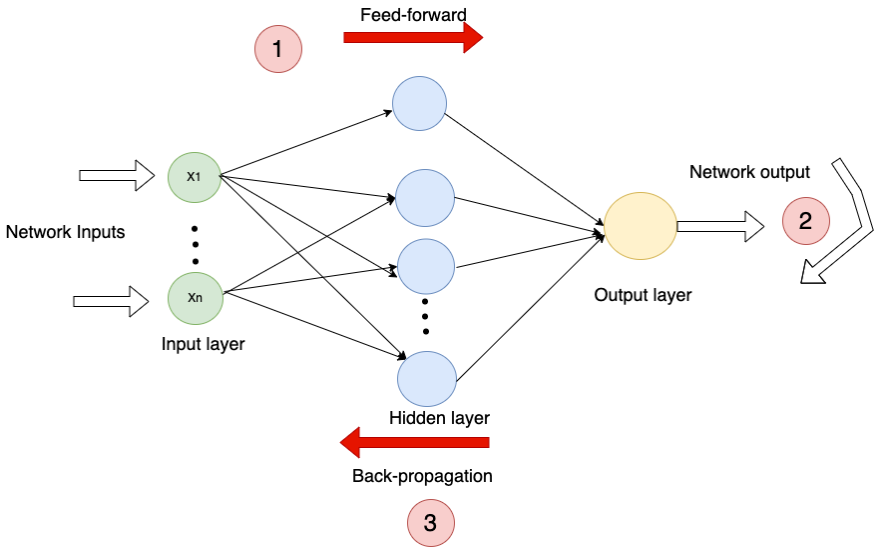

Figure 2.5: Le schéma résumant le Back-Propagation

Après la propagation vers l'avant, l'erreur entre la valeur réelle du résultat, l'estimation de la valeur prédite, les poids et le biais sont calculés selon le sens de l'indication de la figure 2.5.

3. Application aux données

Le présent chapitre va nous servir à mettre en ouvre la partie pratique théorique citée auparavant, l'application de ces méthodes: General Linear Model (GLM), pour les méthodes traditionnelles, la méthode du Decision Tree Regression , le Random Forest regression et le Réseau de Neurone Artificiel simple sous R et python après en s'appuyant sur les informations ramenées concernant l'historique du sinistre des assurés. Le choix de ses outils servant à appliquer notre études est dû à leurs flexibilités et sa facilités pour traiter les données.

3.1 Collecte de données

Il est question dans ce travail de prédire le coût de sinistre des assurances de l'Afrique de l'Ouest. Confrontés aux problèmes de données dans la zone concernée, nous nous sommes trouvés dans l'obligation de faire recours à d'autres sources à l'instar 'AutoCollision' du R package 'insuranceData'. Il s'agit des données, donc l'objectif était de calculer la prime pure en fonction du modèle de GLM.

Description des données:

Cet ensemble de données est basé sur des polices d'assurance véhicule d'un an souscrites en 2004 ou 2005. Il existe 67 856 polices, dont 4 624 (6,8% de réclamations notifiées) ont déposé des réclamations.

1. X_OBSTAT_: Identifiant du client
2. veh_value: la valeur du véhicule
3. exposure: 0-1 - Unité d'exposition (combien de temps la police a été exposée, 1 pour une année complète)
4. clm: Survenance du sinistre (0= non, 1=oui)
5. numclaims: nombre de demande
6. claimcst0: Montant du sinistre (0 s'il n'y a pas de demande)
7. : veh_body: type de véhicule
8. veh_age: l'âge du véhicule qui varie de 1 à 4
9. gender : genre
10. area: les facteurs avec des niveaux
11. agecat: Âge de l'assuré, 1,2,3,4,5,6 (moins âgé à au plus âgé)

Variable cible — Le montant du sinistre ou le montant de la demande est la variable cible.

3.1.1 Quelques statistiques descriptives

Avant de réaliser tout calcul, nous avons commencé par étudier les statistiques descriptives des différentes données. Cette étude est nécessaire et très importante car elle sert à fiabiliser la base de données, et permet de se familiariser avec la base ainsi que de relever les éventuelles données aberrantes supplémentaires qui ne sont pas forcément visibles lorsque que l'on étudie la base ligne à ligne.

Continue variables	count	mean	std	min	25%	50%	75%	max
veh_value	67478.0	1.78	1.21	0.0	1.01	1.5	2.15	34.56
exposure	67478.0	0.47	0.29	0.0	0.22	0.44	0.71	1.0
clm	67478.0	0.07	0.25	0.0	0.0	0.0	0.0	1.0
numclaims	67478.0	0.07	0.28	0.0	0.0	0.0	0.0	4.0
claimcst0	67478.0	138.04	1059.2	0.0	0.0	0.0	0.0	55922.13
veh_age	67478.0	2.67	1.07	1.0	2.0	3.0	4.0	4.0
agecat	67478.0	3.48	1.43	1.0	2.0	3.0	5.0	6.0

Table 3.1: La description statistique des données.

Nous remarquons en faisant une description de notre dataset que certaines variables ont des valeurs élévées par rapport aux autres variables.

3.1.2 Visualisation de quelques données

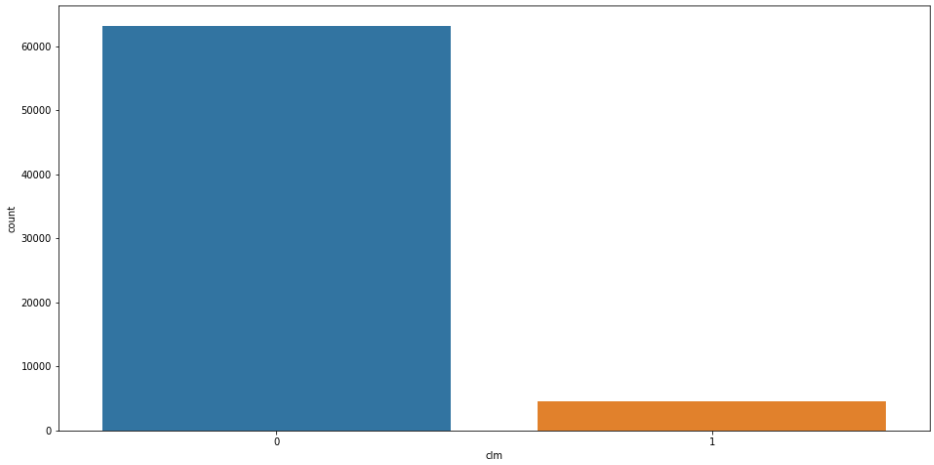

Figure 3.1: La représentation des non ou Survenance du sinistre

Sur la Figure 3.1, nous pouvez le voir, les non survenances du sinistre l'emportent de loin sur les survenances du sinistre.

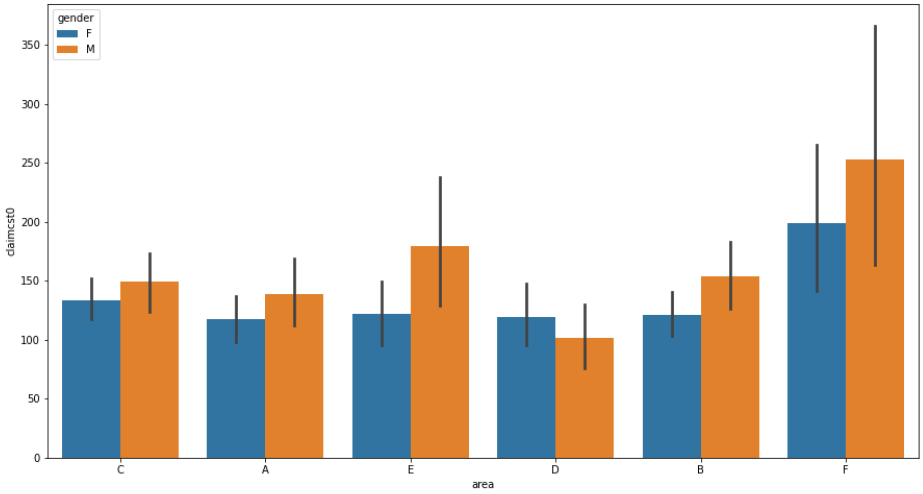

Figure 3.2: la représentation des zones en fonction des types de genre

Sur la Figure 3.2, nous remarquons que la plupart des zones, le genre **M** l'emportent de loin sur les survenances du sinistre. Et c'est la zone F qu'il y a plus de demande.

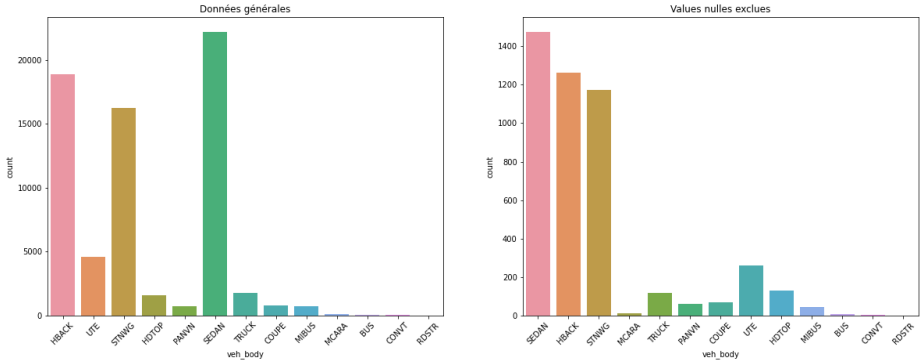

Figure 3.3: la répresentation des types de véhicules en fonction du nombre de demandes des sinistres

La figure 3.3 nous montres effectivement qu'il existe de nombreuses catégories de véhicule avec une fréquence très faible qui pourraient être un inconvénient dans la formation de la fonctionnalité en l'absence de suffisamment de points de données.

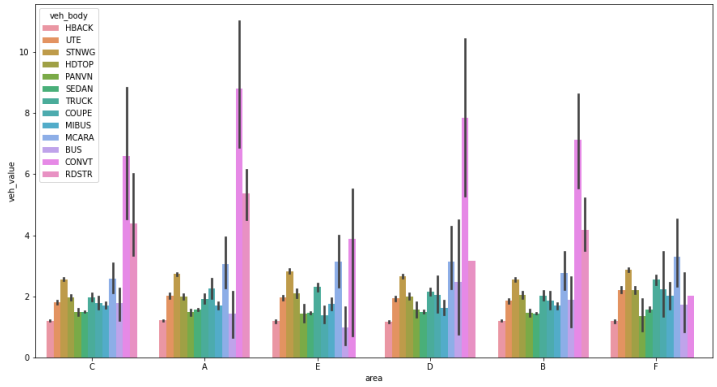

Figure 3.4: la répresentation des zones en fonction des valeurs des véhicules

D'après la figure 3.4, nous pouvons constater que dans toutes les zones, le type de véhucile RDSTR est le plus utilisé, donc cela peut avoir un impact sur le coût des sinistres qui s'explique probablement par le fait qu'il peut exister un nombre infini de raisons avec le type RDSTR pour lesquelles quelqu'un voudrait soumettre une demande d'indemnisation qui est attribuée à une collision ou à des dommages causés par les intempéries.

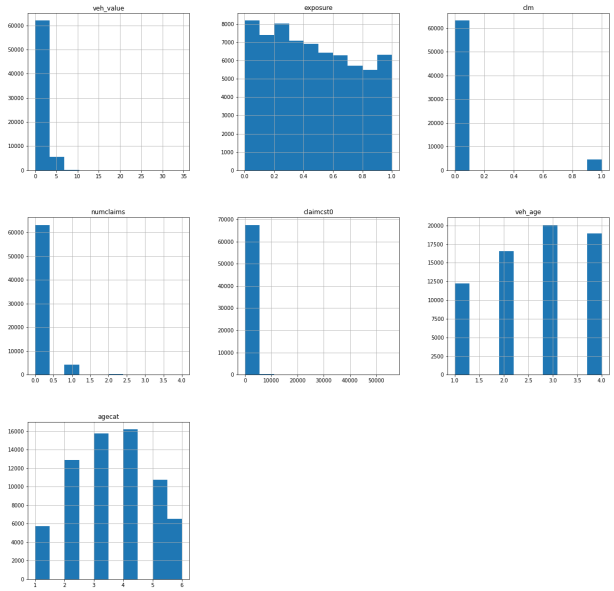

Figure 3.5: la répartition des variables continues

Les graphiques ci-dessus laisse sortir une particularité dans nos données , que nous allons étudier

dans la suite de ce travaille.

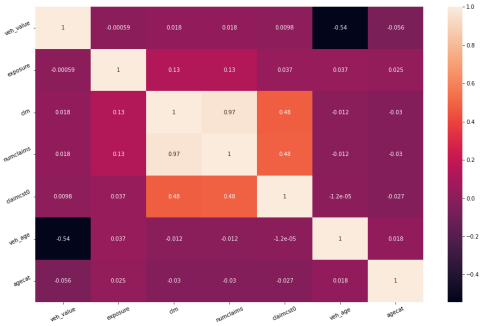

Figure 3.6: la matrices de correlation de nos varraibles continues

Sur la Figure 3.6, nous remarquons que la pluspart de nous variables continues sont moins correlées à la variable cible, donc cela peut avoir un impact aux differents modèles que nous allons utilisés à la suite de notre travail.

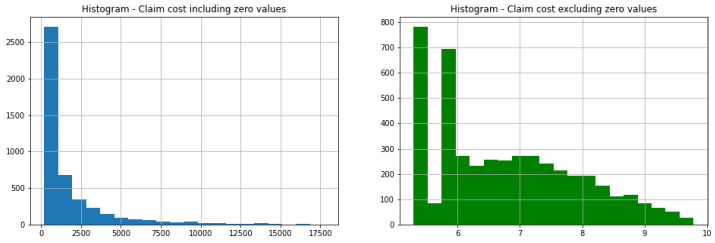

Figure 3.7: la distribution de la variable cible et son log.

Sur la Figure 3.7, Le premier histogramme (bleu) donne une intuition que les données sont très proches de la distribution gamma et l'autre (vert) suggère que nous pouvons essayer un modèle log-normal, bien qu'il soit loin d'être parfaitement normal, a deux pics au début.

Nous avons un autre défi ici, nous devons trouver une technique qui remplace mathématiquement le montant de la réclamation par une gravité de la réclamation. Nous voulions influencer les résultats du modèle sans réellement inclure cette variable (nombre de sinistres) dans la modélisation. Nous pouvons y parvenir en utilisant un terme de décalage dans le modèle.

3.2 Conception des modèles

Pour concevoir un modèle de prédiction pour ses genres de donnée, nous commençons par sélectionner les variables spécifiques. La normalisation des données était essentielle en raison des grandes valeurs des entrées de données. Nous avons utilisé le MinMaxScaler pour mettre à

l'échelle les données entre 0 et 1 pour les tous modèles. En utilisant les modèles de Randam Forest Regression, le Decision Tree Regression, nous avons trouvé les meilleurs paramètres pour lesquels ces modèles ont donné les meilleurs résultats. Le modèle Réseau de Neurone nécessite également un réglage des paramètres qui aide les modèles à fitter au mieux les données. Le Root Mean Squared Error (RMSE) pour évaluer nos modèles.

3.2.1 Split(Diviser) les données

Les données que nous utilisons ont été divisées de la manière suivante: 80 % pour la formation et 20 % pour le test.

3.2.2 Prétraitement des données

Le prétraitement des données est l'une des tâches importantes et souvent difficultes à réaliser dont l'objectif principal est de transformer les données brutes en un format compréhensible, c'est-à-dire de coder les données brutes dans une forme qui peut être facilement analysée interprétée par l'algorithme. Des données non prétraitées conduisent généralement à un mauvais modèle avec une faible précision .

D'après le tableau statistique 3.1, il apparaît clairement que les rangs des caractéristiques sont grands. Il est donc nécessaire de normaliser les données avant de mettre en œuvre une technique de prévision. le MinMaxScaler a été utilisé pour mettre à l'échelle la gamme des données entre 0 et 1, la formule du MinMaxScaler est présentée dans 3.2.1

$$X_{norm} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}. \tag{3.2.1}$$

où X_{norm} est la sortie d'une caractéristique X (X peut être Income, Number of Policies, Total Claim Amount etc) après avoir effectué la normalisation, X_{min} et X_{max} sont le minimum et le maximum de la caractéristique X.

3.2.3 Réglage des paramètres

Le réglage des paramètres est l'une des tâches importantes du modèle prédictif pour obtenir un résultat précis. La finition des meilleurs paramètres pour prévoir avec précision du coût des sinsitres définée comme suit:

- **Modèle de GLM**
- **Modèle de Décision Tree Regression**:

Le processus des modèles du début à son evalution.

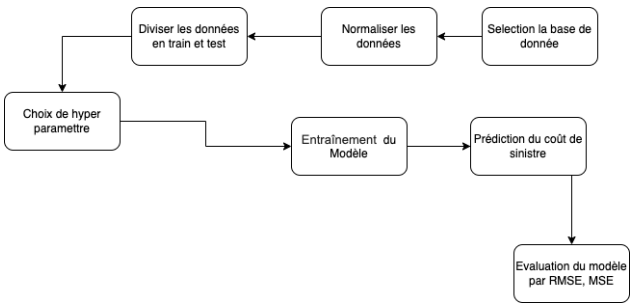

Figure 3.8: le processus du modele

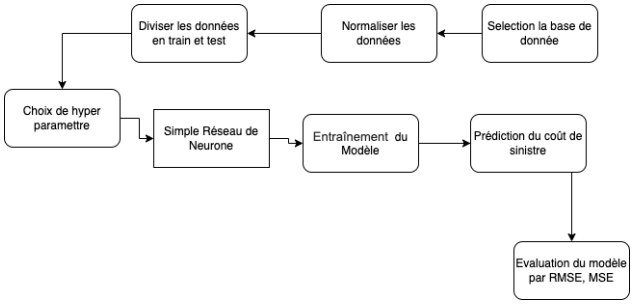

Figure 3.9: le processus du modele du Réseau de Neurone.

3.3 Résultats et discussion

3.3.1 Modèle linéaire géneralisé (GLM)

Le résultat sur le GLM avec la selection des variables importantes dans notre base de données.

Modèle Gaussian

```
glm(formula = claimcst0 ~ veh_value + veh_body + veh_age + gender +
    area + agecat, family = gaussian(link = "log"), data = train,
    offset = log(numclaims))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4610.8  -1304.9   -875.0    299.9  16196.2

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.57669    0.21119  35.877  < 2e-16 ***
veh_value    -0.02528    0.03608  -0.701  0.48353
veh_bodyHBACK -0.05346    0.17030  -0.314  0.75362
```

```
veh_bodyHDTOP -0.16224    0.21055  -0.771  0.44104
veh_bodyMIBUS  0.24768    0.23142   1.070  0.28456
veh_bodyPANVN -0.18012    0.26265  -0.686  0.49289
veh_bodySEDAN -0.19226    0.16931  -1.136  0.25621
veh_bodySTNWG -0.20204    0.17391  -1.162  0.24542
veh_bodyTRUCK -0.20665    0.21659  -0.954  0.34009
veh_bodyUTE   -0.02704    0.18370  -0.147  0.88299
veh_bodyother -0.65335    0.55756  -1.172  0.24136
veh_age2       0.03739    0.08172   0.457  0.64734
veh_age3       0.19161    0.08496   2.255  0.02417 *
veh_age4       0.20889    0.09897   2.111  0.03486 *
genderM        0.08152    0.04861   1.677  0.09365 .
areaB          0.03542    0.07163   0.494  0.62100
areaC          0.07674    0.06451   1.190  0.23429
areaD          0.01819    0.09167   0.198  0.84275
areaE          0.17592    0.08867   1.984  0.04733 *
areaF          0.27482    0.09476   2.900  0.00375 **
agecat2       -0.25420    0.07748  -3.281  0.00105 **
agecat3       -0.20354    0.07426  -2.741  0.00616 **
agecat4       -0.23316    0.07433  -3.137  0.00172 **
agecat5       -0.28524    0.08995  -3.171  0.00153 **
agecat6       -0.25702    0.10489  -2.450  0.01432 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 6380951)

    Null deviance: 2.3607e+10  on 3658  degrees of freedom
Residual deviance: 2.3188e+10  on 3634  degrees of freedom
AIC: 67743

Number of Fisher Scoring iterations: 9
```

Modèle de Gamma

```
    glm(formula = claimcst0 ~ veh_value + veh_body + veh_age + gender +
    area + agecat, family = Gamma(link = "log"), data = train,
    offset = log(numclaims))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8480  -1.2607  -0.7253   0.1411   4.3190
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    7.76128    0.24314  31.921  < 2e-16 ***
veh_value     -0.03473    0.03331  -1.043  0.29715
veh_bodyHBACK -0.20225    0.20631  -0.980  0.32698
veh_bodyHDTOP -0.33431    0.24192  -1.382  0.16708
veh_bodyMIBUS  0.04684    0.31935   0.147  0.88339
veh_bodyPANVN -0.24726    0.28887  -0.856  0.39208
veh_bodySEDAN -0.32657    0.20373  -1.603  0.10903
veh_bodySTNWG -0.37807    0.20604  -1.835  0.06660 .
veh_bodyTRUCK -0.30707    0.25091  -1.224  0.22109
veh_bodyUTE   -0.27722    0.22346  -1.241  0.21484
veh_bodyother -0.67002    0.38830  -1.726  0.08452 .
veh_age2       0.10620    0.07440   1.428  0.15352
veh_age3       0.23059    0.08096   2.848  0.00442 **
veh_age4       0.23517    0.09703   2.424  0.01542 *
genderM        0.06135    0.05036   1.218  0.22323
areaB          0.06259    0.07158   0.874  0.38199
areaC          0.08754    0.06511   1.344  0.17889
areaD         -0.03904    0.08854  -0.441  0.65931
areaE          0.16626    0.09711   1.712  0.08697 .
areaF          0.25674    0.11346   2.263  0.02371 *
agecat2       -0.26592    0.09012  -2.951  0.00319 **
agecat3       -0.22158    0.08812  -2.515  0.01196 *
agecat4       -0.24469    0.08779  -2.787  0.00535 **
agecat5       -0.31782    0.09766  -3.254  0.00115 **
agecat6       -0.31448    0.11238  -2.798  0.00516 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Gamma family taken to be 2.06352)

    Null deviance: 5108.8  on 3658  degrees of freedom
Residual deviance: 4978.6  on 3634  degrees of freedom
AIC: 61952

Number of Fisher Scoring iterations: 7
```

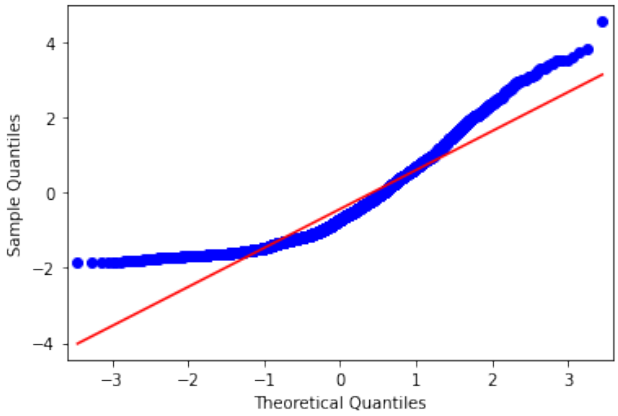

Figure 3.10: La courbe de prédiction des coûts de sinistre du random forest

Indicateur de comparaison	Gaussian avec lien log	Gamma avec lien log
AIC	67243	61952
RMSE	2.481	2.485

Nous pouvons clairement voir que l'un ou l'autre des modèles n'est pas parfait, mais quand même, le gamma avec lien log est légèrement meilleur que le gaussien. Le modèle de dispersion des résidus par rapport à l'ajustement est meilleur dans le gamma et, selon le diagramme QQ, les résidus sont très proches de la distribution normale. Il s'agit d'un GLM et non de la régression linéaire où la normalité des résidus et l'homogénéité de la variance sont une condition stricte, un certain écart est attendu mais un écart important par rapport à la normalité indique un problème avec la distribution supposée.

Enfin, l'AIC et l'erreur quadratique moyenne racine sont un autre indicateur de comparaison de modèles, les deux valeurs sont plus faibles dans le cas du gamma, ce qui suggère un meilleur ajustement.

3.3.2 Modèle de Decision Tree Regression

Le modèle de Decision Tree Regression construit de l'arbre de decision en se basant sur les hyperparamètres que nous avons trouvés et ils se décrivent comme suit:

max-depth: qui est la profondeur maximale de notre arbre de décision.

max-features: le nombre de caractéristiques à prendre en compte lors de la recherche du meilleur split (int, float or $\{auto, sqrt, log2\}$)

max-leaf-nodes: il fait croître un arbre avec max-leaf-nodes dans le meilleur des cas.

min-samples-leaf: est le nombre minimal d'échantillons requis pour être à un noeud.

min-weigth-fraction-leaf: est la fraction pondérée minimale de la somme totale des poids (de tous les échantillons d'entrée) requise pour être à un nœud.

split: est la stratégie utilisée pour choisir le fractionnement à chaque nœud (best ou random).

Le tableau 3.2 présente la spécification complète des paramètres respectivement pour le Decision Tree regression.

Paramètres	valeurs
max-depth	5
max-features	auto
max-leaf-nodes	80
min-samples-leaf	10
min-weight-fraction-leaf	0.1
split	best

Table 3.2: Spécification complète des paramètres de Decision Tree regression.

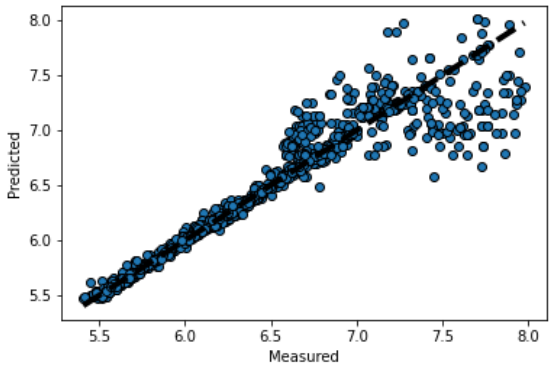

Figure 3.11: Arbre de décision

Après avoir filtré le modèle, nous faisons la performance du modèle du réseau de neurone est évaluée par RMSE comme définis dans le tableau suivant:

Modèle de la Decision Tree Regression	
RMSE	0.8634

3.3.3 Modèle de Random forest

Le modèle de Random forest à pour objectif de construire des arbres de décision C'est-à-dire, à partir de l'échantillon d'apprentissage contenant p variables prédictives, nous avons procédons à la création de N échantillons obtenus par tirage avec remise des observations et de même taille que l'échantillon initial en nous basant sur les hyperparamètres suivants:

Bootstrap: la méthode d'échantillonnage des points de données (avec ou sans remplacement)

max-features: le nombre maximum de caractéristiques prises en compte pour la division d'un nœud.

min-samples-split: le nombre minimum de points de données placés dans un nœud avant que le nœud ne soit divisé.

n-estimators: le nombre d'arbres dans le choisir la forêt.

Le tableau 3.3 présente la spécification complète des paramètres respectivement pour le Decision Tree regression.

Paramètres	valeurs
Bootstrap	True
max-features	auto
min-samples-split	8
n-estimators	20

Table 3.3: Spécification complète des paramètres du random forest.

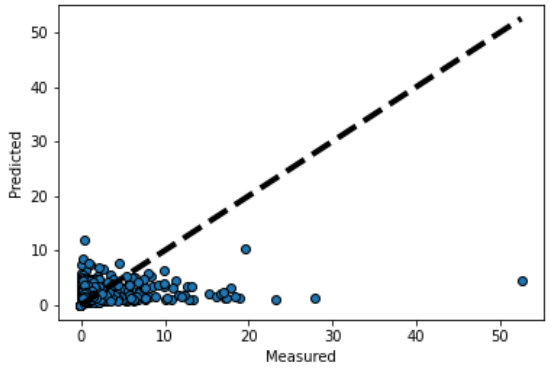

Figure 3.12: La courbe de prédiction des coûts de sinistre du random forest

Après avoir filtré le modèle, nous faisons des prédictions sur les données de test définis en 3.12 et la performance du modèle du réseau de neurone est évaluée par RMSE comme définis dans le tableau suivant:

Modèle du Random Forest	
RMSE	0.8888

3.3.4 Les Réseaux de Neurones

Le modèle de réseau de neurone est composé d'une couche d'entrée, suivie d'une couche cachée dense et d'une couche de sortie dense. Pour chaque variable, le nombre d'entrées, de couches, de fonction d'activation (Relu) appliqué à la sortie. Le nombre de caractéristiques qui constitue le nombre de mémoire est un élément de la couche d'entrée. Le tableau 3.4 présente la spécification complète des paramètres respectivement pour le réseau de neurone.

Paramètres	valeurs
Nombre de neurone	20
Nombre de couche	4
Batch size	88
fonction d'activation	ReLu
Nombre d'epoch	150
Optimizer	adam

Table 3.4: Spécification complète des paramètres du réseau de neurone simulé.

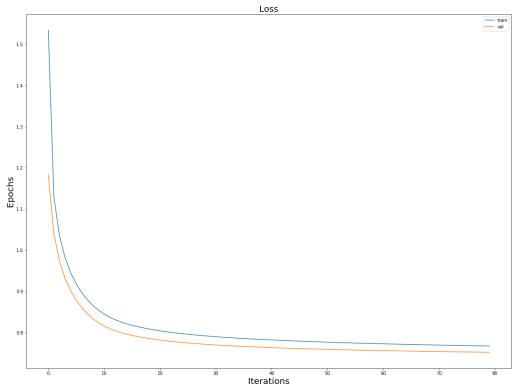

Figure 3.13: La visualisation de la fonction loss

Après avoir filtré le modèle, nous faisons des prédictions sur les données de test définis en 3.13 et la performance du modèle du réseau de neurone est évaluée par RMSE comme définis dans le tableau suivant:

Modèle du Réseau de Neurone	
RMSE	0.8673

3.4 Analyse comparative

Il s'agit d'une étude comparative entre les performances des techniques statistiques et des techniques de machine learning. Sur la base de la valeure du métrique RMSE , le Decision Tree Regression est le plus précis des modeles utilisés (RMSE=0.8634). Toutefois, les modèles peuvent être encore meilleurs si nous traitons plus profondément les variables catégorielles, car bien que ce traitement soit une tâche délicate en Machine Learning , les variables catégorielles sont souvent porteuses d'informations pertinentes.

les modèles	RMSE
GLM	2.485
Random forest regression	0.8888
Decision Tree Regression	0.8634
Réseau de Neurone	0.8673

Table 3.5: Récapitulatif de nos résultats de tous les modèles.

3.5 Calcul de la prime selon des différentes cothégories d'ages et des types de véhicules.

Âges	veh_bodyTRUCK	veh_bodyUTE	veh_bodyother	Prime
1	1552.106	1811.319	837.7243	1831.323
2	1538.082	1794.953	830.1551	1814.776
3	1529.435	1784.862	825.4880	1804.573
4	1241.966	1449.384	670.3316	1465.391
5	2091.178	2440.420	1128.6800	2467.372
6	1180.574	1377.739	637.1960	1392.954
7	1282.684	1496.901	692.3081	1513.433
8	1283.606	1497.977	692.8057	1514.521

Table 3.6: Calcul de la prime avec la méthode de GLM

Âges	veh_bodyTRUCK	veh_bodyUTE	veh_bodyother	Prime
1	1677.013	1638.384	821.3538	1977.864
2	1647.100	1609.160	806.7034	1942.585
3	1556.106	1520.262	762.1370	1835.267
4	1258.761	1229.766	616.5056	1484.578
5	1441.787	1408.576	706.1466	1700.439
6	1530.920	1495.656	749.8017	1805.563
7	1375.482	1343.799	673.6723	1622.239
8	1338.312	1307.485	655.4677	1578.401

Table 3.7: Calcul de la prime avec la méthode de decision Tree Regression

Nous constatons que les assurés dont l'âge se trouve dans la catégorie 1 reçoivent une prime moyenne plus élevée de 1977.864 dollars et les assurés dont l'âge se trouve dans la catégorie 4 reçoivent une prime moyenne plus élevée de 1484.578 dollars. Et les primes calculées avec le modèle de décision Tree Régression sont plus précises par rapport aux primes du GLM.

4. Conclusion et perspectives

Dans ce projet, l'objectif était d'évaluer la performance des techniques des méthodes de machine learning dans le domaine de l'assurance. Pour atteindre nos objectifs, nous avons comparé la performance de ces modèles à d'autres techniques appelées techniques statistiques ou traditionnelles. Les modèles étudiés étaient les suivants : les méthodes de General Linear Model (GLM), pour les méthodes traditionnelles, la méthode du Decision Tree Regression , le Random Forest regression et le Réseau de Neurone Artificiel simple pour les méthodes de machine learning.

Il en ressort au regard de nos travaux que le modèle le plus précis est le **Decision Tree Regression**, avec une erreur ('RMSE') plus faible que les autres.

Comme perspectives, nous projetons utiliser les méthodes de machine learning:

- **Sur les données télématiques** qui nous permettront de mettre à la disposition des compagnies d'assurance et des clients un système qui pourra surveiller les habitudes de conduite en temps réel pour fournir des images objectives.

  Définition: La technologie télématique est un système de suiveillance des comportements de conduite en temps réel pour fournir une image objective des habitudes de conduite. Certains assureurs utilisent la télématique pour surveiller les principaux facteurs de risque associés à la conduite d'une voiture.

- **Assurance agricole**, en s'appuyant sur les données donc fournir les systèmes d'information satellitaire en vue de rendre la qualité d'assurance en Afrique.

References

[1] Robert A Bailey and LeRoy J Simon. Two studies in automobile insurance ratemaking. *ASTIN Bulletin: The Journal of the IAA*, 1(4):192–217, 1960.

[2] Rémi Bellina. Méthodes d'apprentissage appliquées à la tarification non-vie. *Lyon: Université Claude Bernard*, 2014.

[3] RICHARD Bellman. Dynamic programming, princeton univ. *Press Princeton, New Jersey*, 1957.

[4] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.

[5] Dustin Boswell. Introduction to support vector machines. *Departement of Computer Science and Engineering University of California San Diego*, 2002.

[6] Denis Brouillet, L Delcor, D Delignières, and Marielle Cadopi. Analyse de la variabilité par les modèles arima: une source d'information pour la compréhension des processus mnésiques. *L'Année psychologique*, 108(4):699–720, 2008.

[7] A Colin Cameron and Pravin K Trivedi. *Microeconometrics: methods and applications*. Cambridge university press, 2005.

[8] Arthur Charpentier. Statistique de l'assurance. 2010.

[9] Charles Dugas, Yoshua Bengio, Nicolas Chapados, Pascal Vincent, Germain Denoncourt, and Christian Fournier. Statistical learning algorithms applied to automobile insurance ratemaking. In *CAS Forum*, volume 1, pages 179–214. Citeseer, 2003.

[10] Everette S Gardner Jr. Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1):1–28, 1985.

[11] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[12] Emmanuel BERTHELE Actuaire-André GRONDIN, Nicolas LE BERRIGAUD Actuaire-Tristan MUSCAT, and Franck VERMET. Provisionnement à l'aide de modèles de machine learning.

[13] Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics: the approach based on influence functions*, volume 196. John Wiley & Sons, 2011.

[14] Joseph M Hilbe. *Negative binomial regression*. Cambridge University Press, 2011.

[15] Mohammad Hossin and Md Nasir Sulaiman. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2):1, 2015.

[16] Keith Jefferis and Graham Smith. The changing efficiency of african stock markets. *South African journal of economics*, 73(1):54–67, 2005.

[17] Sotiris B Kotsiantis, I Zaharakis, P Pintelas, et al. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007.

[18] Sotiris B Kotsiantis, Ioannis D Zaharakis, and Panayiotis E Pintelas. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26(3):159–190, 2006.

[19] Antoine Ly. *Algorithmes de machine learning en assurance: solvabilité, textmining, anonymisation et transparence*. PhD thesis, Université Paris-Est, 2019.

[20] Peter McCullagh and JA Nelder. Generalized linear models ii, 1989.

[21] Farida Meghatria. *Etude actuarielle des principes de prime en assurance*. PhD thesis, 2010.

[22] Daniel B Nelson. Conditional heteroskedasticity in asset returns: A new approach. *Econometrica: Journal of the Econometric Society*, pages 347–370, 1991.

[23] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR, 2013.

[24] Anupam Tarsauliya, Shoureya Kant, Rahul Kala, Ritu Tiwari, and Anupam Shukla. Analysis of artificial neural network for financial time series forecasting. *International Journal of Computer Applications*, 9(5):16–22, 2010.

[25] Mario V Wuthrich and Christoph Buser. Data analytics for non-life insurance pricing. *Swiss Finance Institute Research Paper*, (16-68), 2020.

[26] Yuhong Yang. Can the strengths of aic and bic be shared? a conflict between model indentification and regression estimation. *Biometrika*, 92(4):937–950, 2005.