Institut des ACTUAIRES

Dauphine | PSL
UNIVERSITÉ PARIS

Mémoire présenté devant l'Université de Paris-Dauphine pour l'obtention du Certificat d'Actuaire de Paris-Dauphine et l'admission à l'Institut des Actuaires

le

Par : Paul DE ARCE
Titre : Analyse des déterminants de la marge client d'un portefeuille d'assurance

Confidentialité : ☐Non ☑ Oui (Durée : ☐ 1 an ☑ 2 ans)

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

Membres présents du jury de l'Institut des Actuaires :

Entreprise :
Nom : Prim'Act
Signature :

Directeur de Mémoire en entreprise :
Nom : Frédéric PLANCHET
et Sugiban RATNASOTHY
Signature :

Membres présents du Jury du Certificat d'Actuaire de Paris-Dauphine :

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

Secrétariat :

Signature du responsable entreprise

Signature du candidat

Bibliothèque :

Résumé

Une compagnie d'assurance, proposant différents types de contrats vie et non-vie, a construit un modèle de prédiction de la marge future de son portefeuille d'assurés sur les 31 prochaines années. Ce mémoire a pour objectif de reconstituer cette marge à l'aide de différents modèles statistiques interprétables, afin d'en expliquer les déterminants et de la rendre interprétable.

Dans un premier temps, on a effectué différentes modélisations en prenant en compte uniquement l'information de détention des différents contrats. Un modèle linéaire généralisé (GLM) a été appliqué : ses résultats sont facilement interprétables, et ses métriques de performances seront utilisées comme valeurs étalon dans la suite du mémoire. On a ensuite modélisé des arbres de régression (CART) afin d'améliorer les performances du modèle, tout en restant compréhensibles pour tout lecteur non initié aux statistiques. D'un point de vue métier, ces modèles sont intéressants de par leur lecture facile. Enfin, on a utilisé les modèles segmentés MOB, combinant une arborescence avec plusieurs régressions. Ces modèles performent mieux que ceux les GLM et CART, et restent facilement interprétables.

On a ensuite ajouté l'information dont dispose la compagnie d'assurance sur les caractéristiques individuelles des assurés, telle que l'ancienneté dans le portefeuille, l'âge de l'assuré ou le type de véhicule possédé par exemple. Cet ajout de données améliore fortement les performances des différents modèles, mais rend leur interprétabilité plus compliquée, car de nombreuses variables rentrent en jeu. L'optimisation des modèles devra donc prendre en compte la facilité d'interprétation, afin d'éviter d'obtenir des modèles très performants mais peu compréhensibles.

Mots-clés : *marge*, *GLM*, *CART*, *MOB*, *Modèles segmentés*, *Interprétabilité*.

Abstract

An insurance company, offering different types of life and non-life policies, has built a model to predict the future margin of its portfolio of policyholders over the 31 coming years. The objective of this thesis is to reconstruct this margin using different interpretable statistical models, in order to explain its determinants.

First, different models have been performed, considering only the information about the holding of the different policies. A generalized linear model (GLM) was applied: its results are easily interpretable, and its performance metrics will be used as benchmarks in the rest of the thesis. Regression trees (CART) were then modeled in order to improve the performance of the model, while remaining understandable to any reader not familiar with statistics. From a practical point of view, these models are useful because they are easy to understand. Finally, we used the segmented models MOB, which combine a tree structure with several regressions. These models perform better than the GLM and CART models and remain easily interpretable.

We then added the information available to the insurance company on the individual characteristics of the policyholders, such as the seniority in the portfolio, the policyholder's age or the type of vehicle owned for example. This addition of data significantly improves the performance of the different models, but makes their interpretability more complex, as many variables come into play. The optimization of the models must therefore consider the easiness of interpretation, in order to avoid obtaining models that are highly efficient but poorly understandable.

Keywords : *margin*, *GLM*, *CART*, *MOB*, *segmented models*, *Interpretability.*

Note de Synthèse

Une compagnie d'assurance disposant de produits IARD et de produits vie (épargne et prévoyance) a construit des modèles de mesure de la marge de chacun de ces produits. Ce modèle estime la marge individuelle future du stock de contrats (à la date de calcul) sur les 31 prochaines années. Une actualisation est effectuée, et l'assureur obtient ensuite une valeur de la marge future estimée pour les différents contrats des assurés du portefeuille.

Cadre de l'étude

Cette étude a pour objectif de reconstituer la marge future à l'aide de différents modèles statistiques, tout en conservant une certaine interprétabilité des résultats. On cherchera à obtenir des modèles interprétables afin de faire ressortir les déterminants de la marge.

Les travaux présentés dans ce mémoire se baseront sur les données transmises par la compagnie d'assurance, qui ont été retraitées dans la suite. Elles correspondent aux assurés ayant souscrit un contrat avant 2020. La base de données finale est composée des caractéristiques individuelles existantes pour tous les assurés (telles que l'âge, l'ancienneté dans le portefeuille ou le canal de souscription par exemple), ainsi que des caractéristiques relatives au différents contrats. Ainsi, un assuré détenant un contrat `AUTO` aura des caractéristiques relatives à ce contrat (la puissance du véhicule assuré, ou le Bonus/Malus du conducteur par exemple), tandis qu'un assuré détenant un contrat MultiRisques Habitation aura les caractéristiques relatives à ce contrat (le nombre de pièce du bien assuré par exemple).

Les assurés peuvent détenir plusieurs type de contrats parmi ceux proposés par la compagnie. Le produit `MRH` est un contrat Multirisques Habitation, le produit `AUTO` correspond à un contrat Automobile, le produit `CORP` est un contrat d'assurance corporelle et le produit `PREV` est un contrat de Prévoyance. Par ailleurs, deux autres produits `EPARGNE` et `DECES` seront mentionnés, correspondant respectivement à un contrat d'épargne et un contrat d'assurance-vie.

La base de donnée a été séparée en deux échantillons : une base d'apprentissage sur laquelle les différents modèles seront entraînés, et une base de test, utilisée pour comparer les performance des modèles. On évaluera ces performances à l'aide de deux métriques, la MAE (*Mean Absolute Error*) et la MSE (*Mean Squared Error*). Afin de reconstituer la marge future des assurés, on a utilisé 3 familles de modèles : les modèles linéaires, les arbres CART et les modèles segmentés MOB.

Modèles linéaires généralisés

Dans un premier temps, on a modélisé la marge future à l'aide d'un modèle linéaire généralisé, ou GLM (*Generalized Linear Model* en anglais). Ce modèle est facilement interprétable, il suffit d'étudier les coefficients affectés à chaque covariable par le modèle. La performance du GLM sera utilisée dans la suite de l'étude comme valeur de référence pour comparer les différents modèles.

Présentation du modèle

Dans notre étude, Y correspond à la marge totale que l'on souhaite reconstruire, et X correspond à la matrice des covariables. Le modèle linéaire généralisé est une extension du modèle linéaire classique, permettant d'élargir la famille de lois que Y peut suivre. On ne suppose plus qu'il existe une relation linéaire entre Y et les covariables (X_i), ni que Y appartient à $\mathbb{R}$: la variable à prédire peut être discrète ou strictement positive par exemple. On introduit donc un nouveau modèle :

$$g(\mathrm{E}[Y|x]) = x^T\beta. \tag{1}$$

La méthode s'effectue en 3 étapes :

1. Choisir une loi de probabilité pour $Y|x$ parmi la famille exponentielle naturelle $\mathcal{F}^{NAT}$;
2. Choisir une "bonne" fonction de lien $g(.)$, en général le lien canonique ;
3. À partir de $(Y_i, x_i)_{1\leq i\leq n}$, estimer β par $\widehat{\beta}$ tel que $\widehat{\mathrm{E}(Y_i|x_i)} = g^{-1}(x_i^T\widehat{\beta})$.

Dans notre cas, la marge future pouvant être positive ou négative, la loi retenue est la loi normale, et la fonction de lien est la fonction identité.

Afin de vérifier que notre modèle ne fait pas de surapprentissage, on effectue une validation croisée (*CV*, ou *cross-validation* en anglais) pendant la phase d'apprentissage. Cette méthode consiste à séparer notre base d'apprentissage en k blocs de même taille, puis de les sélectionner un à un à tour de rôle. Le bloc sélectionné sera utilisé comme base de validation, et les $k-1$ autres blocs constituerons la base d'apprentissage. On répète la méthode k fois au total, ce qui donne k valeurs des métriques choisies, calculées à chaque fois sur le bloc utilisé comme échantillon de test. Si les métriques obtenues diffèrent trop d'un bloc à l'autre, alors le modèle ne prédit pas bien les données.

GLM

Une première modélisation est effectuée en conservant uniquement les variables indiquant si un assuré détient ou non un type de contrat. Ces variables prenant la forme d'indicatrices (valant 0 ou 1), l'interprétation du modèle est très facile : l'estimation de la marge future correspond à la somme des coefficients relatifs aux contrats détenus. Par exemple, pour un assuré détenant un contrat `AUTO1`, un contrat `MRH` et un contrat `CORP`, la valeur prédite de sa marge sera :

$$\begin{aligned}\text{Marge totale} &= \beta_0 + \beta_1 + \beta_3 + \beta_4 \\ &= -417,280 + 433,284 + 696,293 + 423,641 \\ &= 1135,938.\end{aligned}$$

avec β_0 correspondant à l'intercept, β_1, β_3 et β_4 correspondant respectivement aux coefficients relatifs à la détention du contrat `AUTO1`, `MRH` et `CORP`.

Afin de s'assurer de la qualité du modèle, une analyse des résidus de Pearson a été effectuée. Ceux-ci appartenant à plus de 88% à l'intervalle $[-2;2]$, on en conclut une bonne adéquation du modèle.

GLM avec interactions

Le modèle GLM précédent considère seulement les covariables indiquant si un assuré possède ou non un contrat. L'information sur la possession de 2 contrats en même temps n'est pas prise en compte, alors que cela peut avoir un fort impact sur la marge. En effet, on peut supposer qu'un assuré possédant un contrat `AUTO` et un contrat `MRH` n'aura pas à la fois un sinistre automobile et un sinistre MRH la même année, ces 2 risques étant indépendants. Sa marge estimée sera donc probablement plus élevée que celle de 2 individus distincts, l'un possédant uniquement un contrat `AUTO` et l'autre uniquement un contrat `MRH`.

On va donc rajouter les effets d'interactions du second ordre entre les covariables. L'inconvénient de cet ajout est que le nombre de covariables du modèle augmente considérablement. Pour un modèle composé de n covariables, le nombre de covariables d'un modèle avec interactions du $2^{ème}$ ordre est $\frac{n*(n-1)}{2} + n$. Dans notre cas, on dispose de 7 covariables (`AUTO1`, `AUTO2`, `MRH`, `CORP`, `PREV`, `EPARGNE`, `DECES`), ce qui donne un modèle avec interactions du $2^{ème}$ ordre de 28 variables, plus l'intercept.

LASSO

Afin d'améliorer l'interprétabilité du modèle avec interactions, on se tourne vers une régression pénalisée LASSO, effectuée à l'aide du package `R glmnet`. L'intérêt de la pénalisation LASSO est qu'elle fixe certains coefficients à 0, ce qui facilite l'explicabilité du modèle. La pénalisation LASSO permet de garder seulement les covariables significatives, rendant notre modèle parcimonieux. L'algorithme LASSO fixe 7 covariables à 0, ce qui permet de passer d'un modèle comportant 29 covariables à 22, améliorant son interprétabilité.

Utilisation de la loi Gamma

Afin d'améliorer le pouvoir prédictif du modèle, on peut séparer la base de données en 2 parties, l'une avec les valeurs de marge positives, et l'autre les négatives. Cela nous permet de modéliser 2 modèles linéaires généralisés avec la loi Gamma et la fonction de lien réciproque, puis d'utiliser ces 2 modèles dans un modèle composé Bernoulli-Gamma, la paramètre p de la loi Bernoulli correspondant à la proportion empirique d'être négatif. Cette proportion vaut 0.10399.

La performance de ce modèle Bernoulli-Gamma est bien moins bonne que celles des modèles précédent, la MAE valant 1044 et la MSE 2 570 512. Cette méthode n'est en revanche pas applicable à une prédiction d'un nouvel assuré, car on ne connaît pas *a priori* le signe de sa marge, et donc le sous-modèle Gamma adéquat à appliquer.

Performances

Les différentes métriques de performances des modèles linéaires sont présentées dans le Tableau 1 ci-dessous. L'ajout des interactions améliore la performance de la MSE de 2,4% et celle de la MAE de 2,9%. Le modèle LASSO n'entraîne pas d'amélioration significative, mais réduit le nombre de variables explicatives, améliorant l'interprétabilité du modèle.

	MSE	MAE
GLM	1 871 561	884
GLM avec interactions	1 826 465	858
LASSO	1 826 618	858

TABLE 1 : Performances des modèles linéaires

Arbres CART

Les arbres CART (*Classification And Regression Trees*) font partie de la catégorie des arbres de décision. Un arbre de décision est une méthode permettant de prédire ou d'expliquer une variable (qui peut être quantitative ou qualitative) à l'aide d'autres variables explicatives. On l'appelle *arbre de régression* si la variable à expliquer est quantitative, et *arbre de classification* si elle est qualitative. Les arbres de décision sont très populaires car ils sont simples, intuitifs et visuels.

Présentation du modèle

Le modèle fonctionne sur un principe de partitionnement binaire des individus. Le modèle commence à partir d'un premier noeud composé de l'ensemble des individus, puis sépare cette population en deux sous-ensembles selon une variable (choisie par le modèle de façon à créer des sous-ensembles homogènes). Ce partitionnement est réitéré jusqu'à ce qu'un critère d'arrêt soit atteint. Une phase d'*élagage coût-complexité* de l'arbre est ensuite appliquée par le modèle, afin de sélectionner l'arbre optimal. Chaque noeud terminal (*i.e.* un noeud qui n'est pas séparé) est appelé feuille, et la valeur prédite au sein de cette feuille correspond à la moyenne des valeurs de marge des individus dans ladite feuille.

CART

On applique maintenant le modèle CART à nos données. Cette modélisation a été effectuée à l'aide du package `R rpart`. Comme pour les modèles précédents, on se restreint à la détention ou non d'un type de contrat, sans prendre en compte les caractéristiques individuelles. L'arbre de régression CART va donc effectuer des coupes dans notre base d'apprentissage en fonction de la détention d'un type de contrat ou non, ce qui en fait un modèle très simple à expliquer. Les modèles CART ont déjà pour avantage d'être facilement interprétables, mais dans le cas de variables explicatives valant 1 ou 0, ils le sont encore plus : il suffit de savoir si un assuré possède ou non le contrat utilisé par CART pour la coupe afin de savoir dans quelle branche va l'assuré. De même que pour les modèles GLM et LASSO précédents, une validation croisée a été effectuée, afin de s'assurer qu'il n'y a pas de surapprentissage.

L'arbre créé obtient une performance moins bonne que le GLM (MAE supérieure de 1,2%). Cela vient du fait que le paramètre de complexité *cp* (*Complexity Parameter* en anglais) est fixé à 0,01 par défaut par l'algorithme, et que ce paramètre rentre en compte lors du choix de l'arbre optimal. On a modifié la valeur de ce paramètre afin d'améliorer la performance de CART.

Optimisation de CART

On a modélisé différents arbres CART en faisant varier la valeur du paramètre de complexité. On observe que plus l'arbre est profond, plus il devient performant. Cela se fait au détriment de l'interprétabilité et de la lisibilité du modèle, car l'arbre devient trop complexe. Une valeur de $cp = 0.001$ conduit à un bon compromis entre performance et interprétabilité de l'arbre CART (présenté sur la

Figure 1). Les performances de cet arbres sont présentées dans le Tableau 2 ci-dessous.

	MSE	MAE
GLM	1 871 561	884
CART	1 843 805	865

TABLE 2 : Performances du modèle CART avec cp=0.001

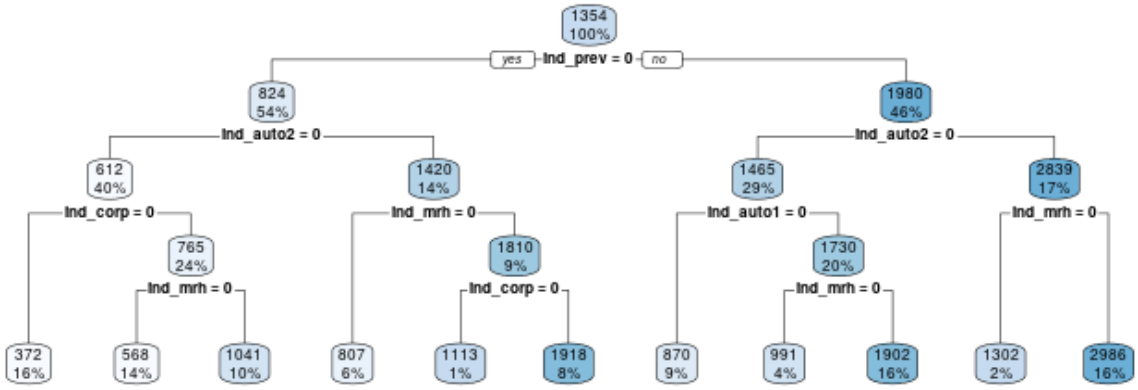


FIGURE 1 : Arbre CART - cp=0.001

Modèles segmentés MOB

Les modèles MOB (*MOdel Based recursive partitionning* en anglais) appartiennent à la classe des modèles segmentés. Ces modèles fonctionnent selon le principe suivant : on partitionne la population selon certaines variables afin de créer des groupes homogènes, puis on applique des modèles à chaque sous-population ainsi créée. Un modèle MOB simple est présenté sur la Figure 2 ci-dessous

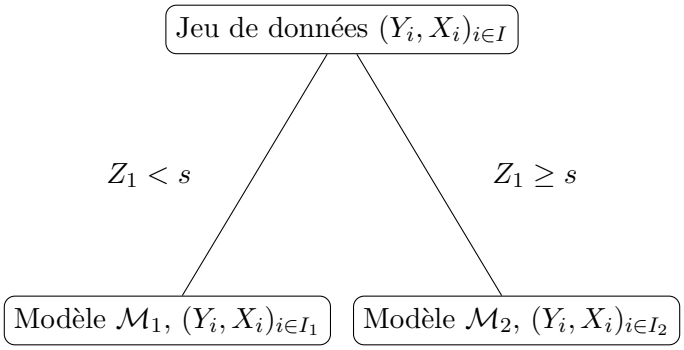


FIGURE 2 : Modèle MOB simplifié

Dans le cadre de cette étude, le partitionnement sera effectué avec un arbre CART, et on appliquera des modèles linéaires aux feuilles de l'arbre. Les covariables sont séparées en deux groupes : les variables de partitionnement, utilisées pour séparer la population en sous-populations, et les variables de régression, utilisées pour l'ajustement des modèles dans les feuilles.

MOB en fonction de la détention des contrats

Une première méthode consiste à utiliser toutes les variables indiquant si un assuré détient ou non un type de contrat, à la fois en tant que variables de partitionnement et de régression. En effet, dans une modélisation MOB classique, l'algorithme peut utiliser une variable dans ces deux rôles : par exemple, si une variable représente la taille d'un individu, MOB peut segmenter selon que l'individu mesure plus ou moins de 176 cm, puis utiliser la taille comme variable de régression en lui affectant un coefficient. Il pourrait même segmenter de nouveau selon que les individus de moins de 176 cm mesurent plus ou moins de 160 cm.

Notre modélisation est différente, car les variables explicatives sont des indicatrices, prenant les valeurs `1` ou `0`. Si l'algorithme segmente sur la variable `ind_auto1` par exemple, représentant la détention du contrat `AUTO1`, alors toutes les observations de ce segment posséderont un contrat `AUTO1`, i.e. `ind_auto1=1`. L'algorithme ne pourra plus utiliser cette variable comme variable de partitionnement (puisque toutes les observations ont la même valeur `ind_auto1=1`), ni comme variable de régression, car cela reviendrait à ajouter le même coefficient à toutes les observations de ce segment, autrement dit créer un second intercept. Dans ce cas, la fonction `glmtree` assigne une valeur NA au coefficient de la régression, et l'intercept contient l'information de cette variable.

Cette méthodologie possède un inconvénient lorsque plusieurs variables indicatrices sont utilisées comme variables de partitionnement. Elles seront toutes fixées à NA dans la régression, et l'intercept sera modifié en conséquence, mais on perd l'interprétation de la variable. En effet, prenons un exemple dans lequel l'algorithme a segmenté une première fois sur la variable `ind_auto1` puis une seconde fois sur `ind_mrh`. L'intercept prend la valeur 267, mais on ne sait pas dans quelles proportions `ind_auto1` et `ind_mrh` ont impactées cette valeur.

La Figure 3 représente le résultat du modèle MOB lorsque la profondeur de l'arbre est fixée à 3. On obtient donc 4 segments, sur lesquels une régression linéaire a été effectuée, donnant les coefficients affichés. Cet exemple illustre les paragraphes précédents, des valeurs NA apparaissant lorsque les variables sont utilisées comme variables de partitionnement. On observe aussi que la variable `ind_auto1` vaut NA dans les nœuds 6 et 7, alors qu'elle n'a pas été utilisée comme variable de partitionnement. Cela vient de la totale corrélation de `ind_auto1` avec `ind_auto2`. En effet, si un assuré possède le contrat `AUTO2`, alors il possède nécessairement le contrat `AUTO1` par construction du portefeuille. Les nœuds 6 et 7 ne comportant que des assurés détenant le contrat `AUTO2` (et par extension le contrat `AUTO1`), la variable `ind_auto1` vaut 1 pour tous les assurés de ces segments.

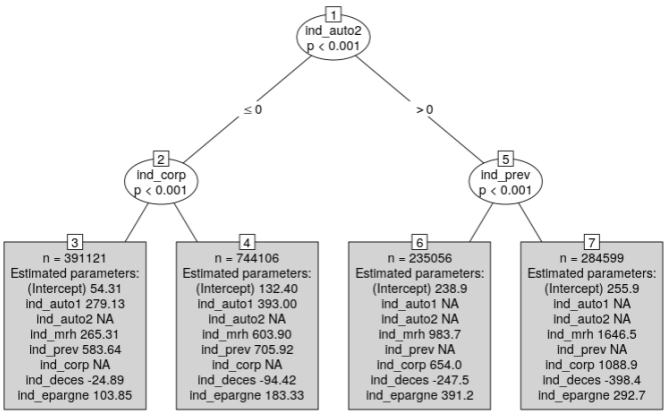


FIGURE 3 : Arbre MOB de profondeur 3

Le modèle MOB avec une profondeur de 3 performe mieux que le GLM classique, mais moins bien que d'autres modèles testés précédemment, comme le GLM avec interactions. On va donc augmenter la profondeur de l'arbre afin de rendre notre modèle plus performant. Les métriques de performances des arbres MOB avec différentes profondeurs sont présentées dans le Tableau 3.

	MSE	MAE	Temps	Nombre de feuilles
MOB profondeur 3	1 832 250	861	67 sec	4
MOB profondeur 4	1 827 421	858	80 sec	8
MOB profondeur 5	1 825 924	857	91 sec	14
MOB profondeur 6	1 825 905	857	95 sec	16

TABLE 3 : Métriques selon la profondeur de l'arbre MOB

L'arbre MOB de profondeur 4 obtient le meilleur compromis entre performance et interprétabilité. Il est composé de 8 feuilles, ce qui permet d'avoir une bonne performance sans être trop complexe. Ainsi, la combinaison d'arbres de décision et de régressions au sein des modèles MOB permet d'améliorer le pouvoir prédictif des modèles. Les différentes modélisations précédentes ont été effectuées en prenant uniquement en compte la détention ou non d'un type de contrat. On va maintenant essayer d'améliorer la prédiction des modèles en rajoutant les caractéristiques individuelles, sans diminuer trop fortement l'interprétabilité.

Ajout des caractéristiques individuelles

Notre base de données étant très riche, on dispose d'une grande information sur les assurés. On peut donc améliorer la qualité de prédiction de nos modèles en prenant en compte cette information. Cependant, afin de garder l'aspect interprétable de nos modèles, on devra faire attention à ne pas obtenir de modèles trop compliqués, au risque de ne plus pouvoir les interpréter.

GLM et CART

On a, dans le même esprit que pour la première partie du mémoire, modélisé la marge future à l'aide de modèles linéaires, en incorporant les variables représentant les caractéristiques individuelles des assurés. Les résultats sont en phase avec l'amélioration attendue suite à l'ajout des caractéristiques individuelles : la MAE diminue de 27%. En revanche, le modèle n'est pas très interprétable, le GLM étant composé de 151 coefficients, car les modalités des variables qualitatives prennent chacune un coefficient différent.

Le GLM avec interactions n'a pas été modélisé, car il aurait été composé de plus de 11 000 coefficients (le nombre de covariables d'un modèle avec interactions du $2^{ème}$ ordre étant de $\frac{n*(n-1)}{2} + n$, notre modèle aurait possédé 11 476, n valant 151). En revanche, on a modélisé un modèle LASSO, réduisant le nombre de coefficients à 80. La performance du LASSO est donc légèrement moins bonne que le GLM, et il reste assez complexe à interpréter.

Une sélection des variables par la méthode *backward* a donc été effectuée, diminuant fortement le nombre de coefficients de régression pour une faible baisse de performance en contrepartie.

Le modèle CART permet d'obtenir une bonne interprétabilité grâce à son système de branches, compréhensible par tout le monde. Néanmoins, les arbres ne sont pas adaptés à des jeux de données possédant de nombreuses covariables. L'algorithme séparant les observations en fonction d'une seule covariable, il devra nécessairement avoir une très grande profondeur si l'on souhaite utiliser un grand

nombre de covariables différentes. En effet, à l'inverse d'un modèle linéaire pour lequel ajouter une variable revient, en terme d'interprétabilité, à ajouter un coefficient, pour le modèle CART, cela revient à ajouter un étage à l'arbre final (dans le cas où la fonction objectif à optimiser est améliorée). Faire cela avec plusieurs variables devient très vite illisible sur le graphique de l'arbre final. Les performances des modèles CART sont meilleures que pour les modèles linéaires, mais ils deviennent rapidement peu interprétable car trop profonds.

L'autre inconvénient de l'utilisation des modèles CART avec ce jeu de données, composé de nombreuses covariables, est que l'on ne peut pas utiliser toutes ces covariables dans la modélisation (ou alors le modèle serait extrêmement complexe et long à calculer). La valeur prédite dans les feuilles étant la moyenne des observations appartenant à cette feuille, on perd l'information contenue dans les variables non utilisées pour la création de l'arbre. Cela justifie l'utilisation des modèles MOB, car la valeur prédite dans les feuilles de l'arbre sera le résultat d'un modèle linéaire calibré sur les variables non utilisées pour la segmentation des observations.

MOB

On rappelle que dans le cadre d'une modélisation MOB, on fournit en paramètre à l'algorithme une formule spécifiant les variables utilisables (et utilisées) pour la segmentation de l'échantillon, et celles utilisées pour les régressions au sein des feuilles. La recherche de la meilleure variable pour segmenter l'échantillon est coûteuse en temps de calcul, c'est pourquoi, dans un premier temps, on se limitera aux variables de détention des différents contrats pour les variables de segmentation. Pour les variables de régression, l'ensemble des variables (détention des contrats non utilisées pour la segmentation et variables caractéristiques individuelles) seront utilisées.

Le second paramètre important de la modélisation MOB est la profondeur de l'arbre : elle indique le nombre de coupes que devra effectuer l'algorithme pour créer les différentes feuilles sur lesquelles seront effectuées les régressions. Le temps de calcul est directement lié à ce paramètre : une profondeur de l'arbre égale à 2 impliquera une seule coupe, donc une seule recherche de meilleure variable de segmentation. Une profondeur de 3 impliquera 3 coupes (1 coupe au début, puis 2 coupes à l'étape suivante), et donc 3 recherches de la meilleure variable de segmentation, et ainsi de suite. On se limitera donc à une profondeur faible, ce qui permettra à la fois de ne pas avoir un temps de calcul trop long, et d'obtenir une forte interprétabilité.

Le premier modèle a donc été paramétré avec une profondeur égale à 2 (ce qui revient à n'effectuer qu'une seule coupe et obtenir 2 feuilles). Ce modèle n'est pas très intéressant et reste assez simpliste, il ne fait que séparer l'échantillon selon que l'assuré possède un contrat `MRH` ou non. On a ensuite augmenté la profondeur d'un niveau (*i.e.* profondeur = 3) afin d'améliorer le pouvoir prédictif du modèle, tout en restant très interprétable (le modèle est visuellement compréhensible par tout le monde).

Puis on a augmenté la profondeur à 4, ce qui ajoute un étage à l'arbre de segmentation. Le modèle reste encore très interprétable, chaque feuille étant le résultat de 3 segmentations en fonction de la détention d'un type de contrat par l'assuré. L'arbre est visible sur la Figure 8. Pour une profondeur de 5, le modèle a de nouveau été lancé, mais les résultats montrent que la profondeur maximale autorisée n'est pas atteinte pour toutes les feuilles. Cela signifie que le modèle estime que segmenter une feuille n'améliore pas significativement le modèle, et qu'il est plus optimal de garder la feuille telle quelle. Cette non-segmentation d'une feuille montre que la complexité du calcul n'est pas le facteur limitant d'une modélisation MOB optimale (si cela avait été le cas, on devrait continuer à incrémenter la profondeur, conduisant à des calculs toujours plus longs), ce qui nous conforte sur la qualité du modèle.

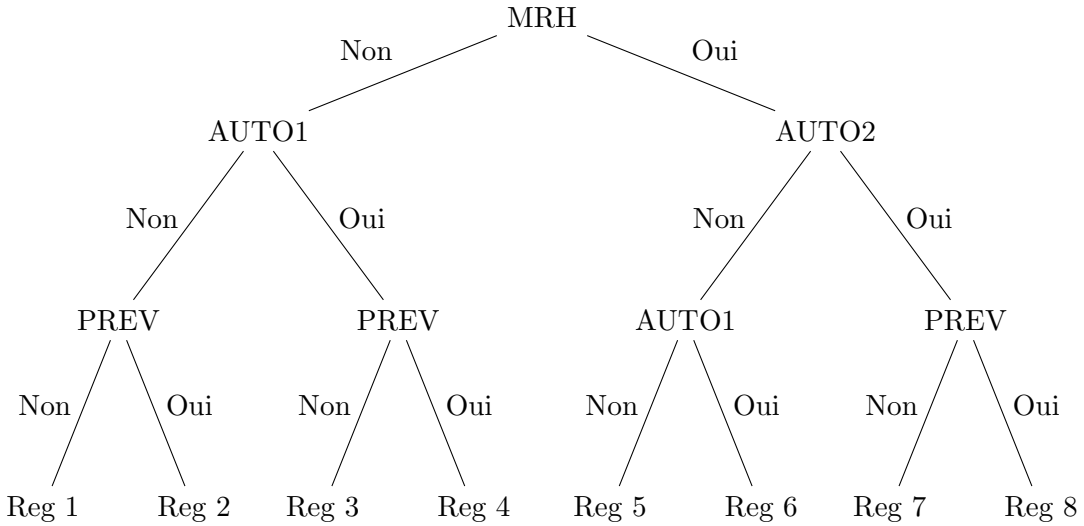


FIGURE 4 : Modèles MOB - Indiv avec profondeur 4

Suite à l'ajout des variables individuelles à notre modélisation, le pouvoir prédictif des modèles s'est considérablement amélioré par rapport aux modèles sans l'information sur les caractéristiques individuelles des assurés, comme on pouvait l'imaginer. Les améliorations en termes de MAE sont de 27% pour le modèle GLM, 24% pour le LASSO, 30% pour le CART (à paramètre *cp* identique) et 28% pour le MOB (avec un paramètre de profondeur égal à 5). On obtient donc en moyenne 25% d'amélioration du pouvoir prédictif de nos modèles : l'ajout des variables individuelles a donc un impact significatif sur la qualité de nos modèles.

Le Tableau 3.9 ci-dessous présente les métriques des différents modèles retenus pour chaque classe de modèle, ainsi que leur temps de calcul. Le modèle MOB est donc le plus performant, malgré un temps de calcul bien supérieur aux autres modèles.

	MSE	MAE	Temps
GLM - Indiv	1 166 117	648	54 sec
LASSO - Indiv	1 175 718	650	5 min
CART - Indiv - cp=0.0001	1 127 294	634	6 min
MOB - Indiv - 5	1 105 757	614	54 min

TABLE 4 : Métriques de performance des différents modèles avec les données individuelles

Ajout de variables de segmentation

On pourrait techniquement ajouter toutes les variables individuelles à la liste des variables de segmentation. Cependant, cela n'aurait pas trop de sens pour certaines d'entre elles. En effet, notre base de données contient des variables spécifiques à certains contrats (la variable `auto2_energie` par exemple, qui vaut `NA` pour tous les assurés ne possédant pas le contrat `AUTO2`), une segmentation selon ces variables serait donc peu pertinente. En revanche, certaines variables sont communes à tous les assurés, comme l'âge ou l'ancienneté du contrat. On effectue donc de nouvelles modélisations en ajoutant les variables `age`, `anc`, `canal`, `enfant` et `couple` comme potentielles variables de segmentation.

Une première modélisation est effectuée en fixant le paramètre de profondeur maximale de l'arbre à 3. Le modèle a sélectionné la variable `canal` comme première variable de segmentation : cette variable a donc une grande influence sur la marge (la MAE passe de 620 à 614 avec l'ajout de la variable `canal`, soit une amélioration d'environ 1%). Les 2 feuilles ainsi créées (l'une composée des assurés ayant souscrit via le canal interne, l'autre via le canal externe) sont ensuite segmentées selon la variable `ind_mrh`, qui était la première variable de segmentation dans les modélisations MOB - Indiv précédentes. On en conclut que le canal de souscription a une plus grande influence sur la marge que la détention d'un contrat MRH, tandis que l'âge, l'ancienneté, la présence d'enfants ou d'un couple sont moins impactant que la détention d'un contrat MRH. On augmente ensuite la profondeur maximale de l'arbre, dans le but de déterminer si ces nouvelles variables de segmentation ont un impact sur la marge supérieur à la détention des autres contrats.

Avec une profondeur maximale égale à 4, l'arbre obtient une meilleure performance (présentée dans le Tableau 5) que le modèle MOB - Indiv - 4. Cela vient de l'utilisation de la variable `canal` comme variable de segmentation, qui a une forte influence sur la marge. Il en est de même pour le modèle avec une profondeur maximale égale à 5 : la MAE passe de 614 à 601, soit une amélioration d'un peu plus de 2%. La variable `canal` a donc un grand impact sur la marge future.

	MSE	MAE	Temps
MOB - Indiv - 4	1 109 851	616	46 min
MOB - Indiv - 3 - canal	1 077 173	614	36 min
MOB - Indiv - 4 - canal	1 058 625	604	50 min
MOB - Indiv - 5 - canal	1 052 689	601	75 min
MOB - Indiv - 6 - canal	1 049 209	599	85 min

TABLE 5 : Métriques de performance des modèles MOB

Sélection des variables

Afin de réduire le nombre de variables explicatives au sein de chaque feuille, la modélisation MOB a été lancée de nouveau en utilisant les variables sélectionnées par la méthode *backward* comme variables de régression. Les performances des modèles diminuent peu par rapport aux modèles MOB utilisant toutes les variables comme variables de régression (1 point de MAE en moyenne). Pour les variables de partitionnement, on a utilisé les indicatrices de détention des types de contrat, ainsi que les variables individuelles communes aux assurés (`age`, `anc`, `canal`, `couple` et `enfant`).

Ces modèles deviennent donc bien plus lisibles et interprétables que les précédents, car les feuilles sont composées d'un nombre réduit de coefficients. En effet, le nombre de variables de régression passe de 42 à 28. Celles-ci pouvant posséder entre 2 et 12 modalités, on réduit fortement le nombre de coefficients de régression au sein des feuilles.

Utilisation de la loi Gamma

La régression linéaire avec une loi Gamma nécessitant que les valeurs cibles soient de même signe, on ne peut pas appliquer la méthode utilisée précédemment, qui consistait à créer un modèle Bernoulli-Gamma en fonction du signe de la marge

En effet, l'arbre de partitionnement obtenu sera potentiellement différent en fonction de l'échantillon sélectionné. L'arbre étant créé en fonction des caractéristiques des individus base d'apprentissage, les variables utilisées pour les coupes successives ne sont pas nécessairement identiques, car les échantillons

sont différents selon que la marge est positive ou négative. On ne peut donc pas simplement créer de modèle Bernoulli-Gamma et effectuer les régressions dans les feuilles, puisque les structures des 2 arbres ne sont pas identiques.

En revanche, cette approche permet de faire ressortir les déterminants de la marge en fonction de son signe. En créant deux arbres de partitionnement, l'algorithme MOB segmente les échantillons selon différentes variables, ce que les modélisations précédentes ne nous permettaient pas de faire, étant effectuées sur l'ensemble du jeu de données.

On observe donc que la variable `canal` est la plus importante pour les assurés dont la marge future estimée est négative, alors qu'il s'agit de la variable `ind_auto2` pour ceux dont la marge future estimée est positive.

Conclusion

Dans cette étude, on a cherché à reconstituer la marge future d'un portefeuille d'assurés, afin de la rendre plus interprétable. En se basant sur les valeurs de marge future individuelles calculées par le modèle de l'assureur sur les 31 prochaines années, on a construit différents modèles statistiques permettant de reconstituer cette marge et d'en faire ressortir les déterminants. Cette analyse apporte une meilleure interprétabilité des résultats, permettant différentes applications métier (quels segments d'assurés privilégier pour la souscription d'un nouveau contrat par exemple).

Les modèles MOB semblent être les meilleurs modèles pour répondre à la problématique de ce mémoire, ils obtiennent de meilleures performances que les différentes régressions linéaires étudiées, ainsi que les arbres CART. Ils sont par ailleurs assez facilement interprétable, à l'inverse des modèles "boîtes noires" évoqués au chapitre 1, très performant mais peu interprétables. Le choix de la profondeur maximale de l'arbre permet de satisfaire les différents besoins des utilisateurs : on augmentera la profondeur si une meilleure performance est recherchée, tandis qu'on la diminuera si l'on souhaite obtenir une meilleur interprétabilité des résultats.

Synthesis note

An insurance company with P&C products and life products has built models to measure the margin of each of these products. This model estimates the future individual margin of the stock of contracts (at the calculation date) over the next 31 years. A discounting is performed, and then the insurer obtains a value of the estimated future margin for the different contracts of the policyholders of the portfolio.

Study environment

The objective of this study is to reconstruct the future margin using several statistical models, while maintaining some interpretability of the results. Interpretable models will be sought in order to highlight the drivers of the margin.

The results presented in this thesis are based on the data provided by the insurance company, which has been reprocessed thereafter. They correspond to policyholders who have taken out a contract before 2020. The final database is made of individual characteristics existing for all policyholders (such as age, seniority in the portfolio or the subscription channel for example), as well as characteristics related to the different contracts. Thus, a policyholder with a car policy (`AUTO`) will have characteristics relating to that policy (the power of the insured car, or the driver's no-claim bonus, for example), while a policyholder with a multi-risk home insurance policy will have the characteristics related to that policy (the number of rooms in the insured property, for example).

The insured can hold several type of contracts among those offered by the company. The product `MRH` is a multi-risk home policy, the product `AUTO` corresponds to a car policy, the product `CORP` is a complementary car insurance policy and the product `PREV` is a provident policy. Moreover, two other products `SAVINGS` and `DEATH` will be mentioned, corresponding respectively to a contract of saving and a contract of life insurance.

The database has been split into two segments: a training database on which the different models will be trained, and a test database, used to compare the performance of each model. The performance will be evaluated using two metrics, the MAE (*Mean Absolute Error*) and the MSE (*Mean Squared Error*). In order to reconstruct the future margin of the policyholders, 3 families of models have been used: linear models, CART trees and MOB segmented models.

Generalized Linear Models

First, the future margin was modelized using a Generalized Linear Model (GLM). This model is easily interpretable, we only need to examine the coefficients assigned to each covariable of the model. The performance of the GLM will be used in the remainder of the study as a benchmark to compare the various models.

Model presentation

In our study, Y corresponds to the total margin that we wish to reconstruct, and X corresponds to the covariate matrix. The generalized linear model is an extension of the linear model, allowing to extend the family of laws that Y can follow. We no longer assume that there is a linear relationship between Y and the covariates (X_i), nor that Y belongs to $\mathbb{R}$: the variable to be predicted can be discrete or strictly positive for example. We therefore introduce a new model:

$$g(\mathrm{E}[Y|x]) = x^T\beta. \tag{2}$$

The method is performed in 3 steps:

1. Choose a probability distribution for $Y|x$ from the natural exponential family $\mathcal{F}^{NAT}$;
2. Choose a "good" link function $g(.)$, in general the canonical link ;
3. From $(Y_i, x_i)_{1\leq i\leq n}$, estimate β by $\widehat{\beta}$ such that $\widehat{\mathrm{E}(Y_i|x_i)} = g^{-1}(x_i^T\widehat{\beta})$.

In our case, since the future margin can be positive or negative, the distribution chosen is the normal distribution, and the link function is the identity function.

In order to check that our model does not overfit the training data, we perform a cross-validation (CV) during the learning phase. This method consists in separating our learning base into k blocks of the same size, and then selecting them one by one in succession. The selected block will be used as a validation base, and the $k-1$ other blocks will constitute the learning base. The method is repeated k times in total, which gives k values of the selected metrics, calculated each time on the block used as a test sample. If the metrics obtained differ too much from one block to another, we conclude that the model does not predict the data well.

GLM

A first modeling is done by only keeping the variables indicating whether or not an policyholder holds a type of contract. Since these variables take the form of indicators (worth 0 or 1), the interpretation of the model is straightforward: the estimate of the future margin corresponds to the sum of the coefficients relating to the contracts held. For example, for a policyholder holding a contract `AUTO1`, a contract `MRH` and a contract `CORP`, the predicted value of his margin will be :

$$\begin{aligned}\text{Total margin} &= \beta_0 + \beta_1 + \beta_3 + \beta_4 \\ &= -417,280 + 433,284 + 696,293 + 423,641 \\ &= 1135,938.\end{aligned}$$

with β_0 corresponding to the intercept, β_1, β_3 and β_4 corresponding respectively to the coefficients relating to the holding of the contract `AUTO1`, `MRH` and `CORP`.

To ensure the quality of the model, an analysis of the Pearson residuals was performed. The residuals are more than 88% in the interval $[-2;2]$, we conclude that the model is well fitted.

GLM with interactions

The previous GLM model considers only the covariates indicating whether or not an insured owns a policy. The information on the possession of 2 contracts at the same time is not taken into account, although it can have a strong impact on the margin. Indeed, we can suppose that an insured with

an `AUTO` policy and a `MRH` policy will not have both a car claim and a MRH claim in the same year, these 2 risks being independent. Its estimated margin will therefore probably be higher than that of two distinct individuals, one with only an `AUTO` policy and the other with only a `MRH` policy.

We will therefore add the effects of second order interactions between the covariates. The disadvantage of this addition is that the number of covariates in the model increases considerably. For a model composed of n covariates, the number of covariates of a model with 2^{nd} order interactions is $\frac{n*(n-1)}{2}+n$. In our case, we have 7 covariates (`AUTO1`, `AUTO2`, `MRH`, `CORP`, `PREV`, `SAVINGS`, `DEATH`), which gives a model with interactions of the 2^{nd} order of 28 variables, plus the intercept.

LASSO

In order to improve the interpretability of the model with interactions, we turn to a penalized LASSO regression, performed with the package `R glmnet`. The interest of the LASSO penalization is that it sets some coefficients to 0, which facilitates the explicability of the model. LASSO penalization allows us to keep only the significant covariates, making our model parsimonious. The LASSO algorithm sets 7 covariates to 0, which allows us to go from a model with 29 covariates to 22, improving its interpretability.

Use of the Gamma distribution

In order to improve the predictive power of the model, we can separate the database into 2 parts, one with positive margin values, and the other with negative ones. This allows us to model 2 generalized linear models with the Gamma distribution and the reciprocal link function, and then to use these 2 models in a composite Bernoulli-Gamma model, the parameter p of the Bernoulli distribution corresponding to the empirical proportion of being negative. This proportion is 0.10399.

The performance of this Bernoulli-Gamma model is much worse than the previous models, the MAE being 1044 and the MSE 2,570,512. This method is not applicable to the prediction of a new policyholder, because we do not know the sign of its margin *a priori*, and thus the adequate Gamma sub-model to apply.

Performances

The different performance metrics of the linear models are presented in the Table 6 below. Adding the interactions improves the performance of the MSE by 2.4% and the MAE by 2.9%. The LASSO model does not lead to a significant improvement, but reduces the number of explanatory variables, improving the interpretability of the model.

	MSE	MAE
GLM	1 871 561	884
GLM with interactions	1 826 465	858
LASSO	1 826 618	858

Table 6: Performances of linear models

CART trees

CART trees (*Classification And Regression Trees*) belong to the category of decision trees. A decision tree is a method used to predict or explain a variable (which can be quantitative or qualitative) using other explanatory variables. It is called a *regression tree* if the variable to be explained is quantitative, and a *classification tree* if it is qualitative. Decision trees are popular because they are simple, intuitive and visual.

Model presentation

The model works on the principle of binary partitioning of individuals. The model starts from a first node composed of all the individuals, and then separates this population into two subsets according to a variable (chosen by the model in order to create homogeneous subsets). This partitioning is repeated until a stopping criterion is reached. A phase of *cost-complexity pruning* of the tree is then applied by the model, in order to select the optimal tree. Each terminal node (i.e. a node that is not split) is called a leaf, and the predicted value within this leaf corresponds to the average of the margin values of the individuals in this leaf.

CART

We now apply the CART model to our data. This modeling has been done with the package `R rpart`. As for the previous models, we restrict ourselves to the possession or not of a type of contract, without taking into account the individual characteristics. The CART regression tree will therefore make cuts in our learning base according to whether or not one holds a type of contract, which makes it a very simple model to explain. CART models are already easy to interpret, but in the case of explanatory variables worth 1 or 0, they are even easier to interpret: we only need to know if an insured owns or not the contract used by CART for the cut in order to know in which branch the insured goes. As for the previous GLM and LASSO models, a cross-validation was performed to ensure that there is no over-learning.

The created tree performs worse than the GLM (the MAE is 1.2% higher). This is due to the fact that the complexity parameter *cp* is set to 0.01 by default by the algorithm, and that this parameter is taken into account when choosing the optimal tree. We have modified the value of this parameter in order to improve the performance of CART.

CART Optimization

Different CART trees were modeled by changing the value of the complexity parameter. We observe that the deeper the tree is, the better it performs. This occurs at the detriment of the interpretability and readability of the model, as the tree becomes too complex. A value of $cp = 0.001$ leads to a good compromise between performance and interpretability of the CART tree (presented on Figure 5). The performances of this tree are presented in Table 7 below.

	MSE	MAE
GLM	1 871 561	884
CART	1 843 805	865

Table 7: CART performance with cp=0.001

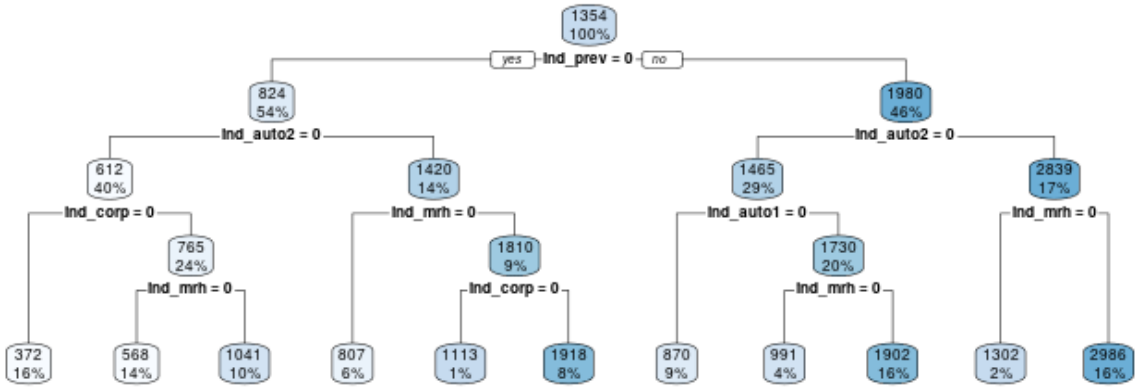


Figure 5: CART tree - cp=0.001

MOB segmented models

MOB models (*MOdel Based recursive partitioning* in English) belong to the class of segmented models. These models work on the following idea: the population is partitioned according to certain variables in order to create homogeneous groups, and then models are applied to each sub-population that has been created. A simple MOB model is shown in Figure 6 below.

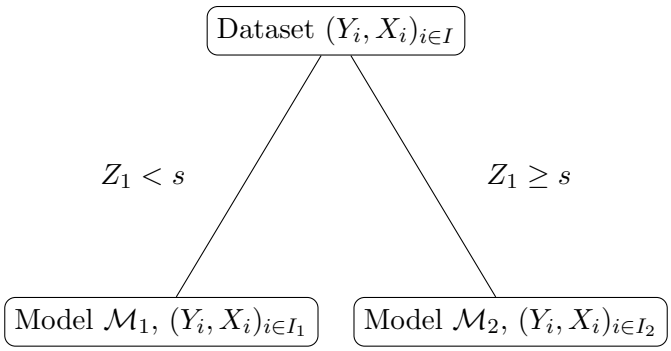


Figure 6: Simplified MOB model

In this study, partitioning will be performed with a CART tree, and linear models will be applied to the leaves of the tree. The covariates are separated into two groups: partitioning variables, used to separate the population into subpopulations, and regression variables, used to fit the models in the leaves.

MOB based on contract ownership

A first method is to use all variables indicating whether or not an insured holds a type of contract, both as partitioning and regression variables. Indeed, in a classical MOB modeling, the algorithm can use a variable for both roles: for example, if a variable represents the height of an individual, MOB can segment according to whether the individual is taller or shorter than 176 cm, and then use the height as a regression variable by assigning a regression coefficient to it. It could even segment again by whether individuals under 176 cm are taller or shorter than 160 cm.

Our modeling is different, because the explanatory variables are indicators, taking the values `1` or `0`. If the algorithm segments on the variable `ind_auto1` for example, representing the holding

of the contract `AUTO1`, then all the observations of this segment will possess a contract `AUTO1`, i.e. `ind_auto1=1`. The algorithm will no longer be able to use this variable as a partitioning variable (since all the observations have the same value of `ind_auto1=1`), nor as a regression variable, because this would mean adding the same coefficient to all the observations of this segment, in other words creating a second intercept. In this case, the function assigns an NA value to the regression coefficient, and the intercept carries the information of this variable.

This methodology has a drawback when several indicator variables are used as partitioning variables. All these variables are set to NA in the regression, and the intercept is adjusted accordingly, but the interpretation of the variable is lost. Indeed, let's take an example in which the algorithm has segmented a first time on the variable `ind_auto1`, then a second time on `ind_mrh`. The intercept takes the value 267, but we don't know in which proportions `ind_auto1` and `ind_mrh` have impacted this value.

Figure 7 represents the results of the MOB model when the depth of the tree is fixed at 3. We thus obtain 4 segments, on which a linear regression has been performed, giving the displayed coefficients. This example illustrates the previous paragraphs, NA values appearing when the variables are used as partitioning variables. We also observe that the variable `ind_auto1` is NA in nodes 6 and 7, even though it was not used as a partitioning variable. This is due to the total correlation of `ind_auto1` with `ind_auto2`. Indeed, if a policyholder owns the contract `AUTO2`, then he necessarily owns the contract `AUTO1` by portfolio construction. Since nodes 6 and 7 only contain policyholders with the contract `AUTO2` (and by extension the contract `AUTO1`), the variable `ind_auto1` is 1 for all the policyholders in these segments.

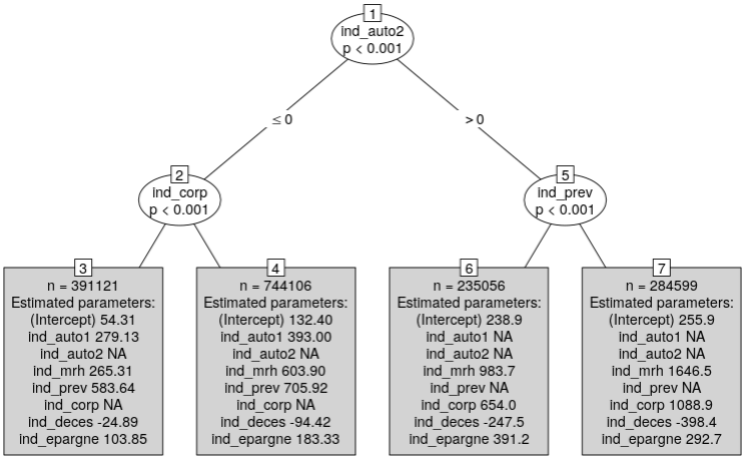


Figure 7: MOB tree with a depth of 3

The MOB model with a depth of 3 performs better than the classical GLM, but worse than other models tested previously, such as the GLM with interactions. We will therefore increase the depth of the tree to make our model perform better. The performance metrics of the MOB trees with different depths are presented in Table 8.

	MSE	MAE	Time	Number of leaves
MOB with a depth of 3	1 832 250	861	67 sec	4
MOB with a depth of 4	1 827 421	858	80 sec	8
MOB with a depth of 5	1 825 924	857	91 sec	14
MOB with a depth of 6	1 825 905	857	95 sec	16

Table 8: Metrics according the the depth of MOB trees

The MOB tree of depth 4 obtains the best compromise between performance and interpretability. It is made of 8 leaves, which allows to have a good performance without being too complex. Thus, the combination of decision trees and regressions in MOB models improves the predictive power of the models. The various previous models were carried out by taking into account only the possession or not of a type of contract. We will now try to improve the predictive power of the models by adding the individual characteristics, without reducing too much the interpretability.

Addition of individual characteristics

Since our database is very deep, we have a lot of information about the policyholders. We can therefore improve the prediction quality of our models by taking this information into account. However, in order to keep the interpretable aspect of our models, we will have to be careful not to obtain too complicated models, at the risk of not being able to interpret them.

GLM and CART

In the same spirit as for the first part of the thesis, the future margin was modeled using linear models, incorporating variables representing the individual characteristics of policyholders. The results are in line with the expected improvement following the addition of individual characteristics: the MAE decreases by 27%. On the other hand, the model is not very interpretable, as the GLM is composed of 151 coefficients, because the modalities of the categorical variables take a different coefficient each.

The GLM with interactions was not modeled, because it would have been composed of more than 11,000 coefficients (the number of covariates in a model with 2^{nd} interactions being $\frac{n*(n-1)}{2} + n$, our model would have had 11,476, n being 151). Instead, we modeled a LASSO model, reducing the number of coefficients to 80. The performance of the LASSO is thus slightly worse than the GLM, and it remains rather complex to interpret.

A selection of the variables by the *backward* method was thus carried out, strongly decreasing the number of regression coefficients for a small decrease in performance in return.

The CART model allows for good interpretability due to its branch system, which can be understood by everyone. Nevertheless, trees are not adapted to data sets with many covariables. Since the algorithm separates observations according to a single covariate, it will necessarily have to be very deep if a large number of different covariates are to be used. Indeed, contrary to a linear model for which the addition of a variable means, in terms of interpretability, adding a coefficient, for the CART model, it means adding a stage to the final tree (in the case where the objective function to be optimized is improved). Doing this with several variables becomes quickly unreadable on the graph of the final tree. The performances of CART models are better than for linear models, but they quickly become uninterpretable because of their depth.

The other disadvantage of using CART models with this dataset, composed of many covariates, is that we cannot use all these covariates in the modeling (or else the model would be extremely complex

and long to compute). The predicted value in the leaves being the average of the observations belonging to this leaf, one loses the information contained in the variables not used for the creation of the tree. This justifies the choice of MOB models, because the predicted value in the leaves of the tree will be the result of a linear model calibrated on the variables not used for the segmentation of the observations.

MOB

We recall that in the context of MOB modeling, we provide the algorithm with a formula specifying the variables that can be used (and are used) for the segmentation of the sample, and those used for the regressions within the leaves. The search for the best variable to segment the sample is costly in terms of computation time, which is why, at first, we will limit ourselves to the variables of possession of the different contracts for the segmentation variables. For the regression variables, all the variables (contract ownership not used for segmentation and individual characteristics variables) will be used.

The second important parameter of the MOB modeling is the depth of the tree: it indicates the number of cuts that the algorithm will have to perform to create the different leaves on which the regressions will be performed. The computation time is directly linked to this parameter: a tree depth of 2 will imply only one cut, thus only one search for the best segmentation variable. A depth of 3 will imply 3 cuts (1 cut at the first step, then 2 cuts at the second step), and thus 3 searches for the best segmentation variable, and so on. We will therefore limit ourselves to a low depth, which will allow us to avoid a long computation time, and to obtain a high interpretability.

The first model was therefore configured with a depth equal to 2 (which means that only one cut was made and 2 leaves were obtained). This model is not very interesting and remains rather simplistic, it only separates the sample according to whether the policyholder has a contract `MRH` or not. We then increased the depth of the model by one level (i.e. depth = 3) in order to improve the predictive power of the model, while remaining very interpretable (the model is visually understandable by everyone).

Then we increased the depth to 4, which adds a stage to the segmentation tree. The model is still very interpretable, each leaf being the result of 3 segmentations according to the policyholder's possession of a type of contract. The tree is visible on Figure 8. For a depth of 5, the model was run again, but the results show that the maximum allowed depth is not reached for all the leaves. This means that the model considers that segmenting a leaf does not significantly improve the model, and that it is more optimal to keep the leaf as it is. This non-segmentation of a leaf shows that the complexity of the computation is not the limiting factor of an optimal MOB modeling (if it were the case, we would have to continue to increment the depth, leading to ever longer computations), which confirms the quality of the model.

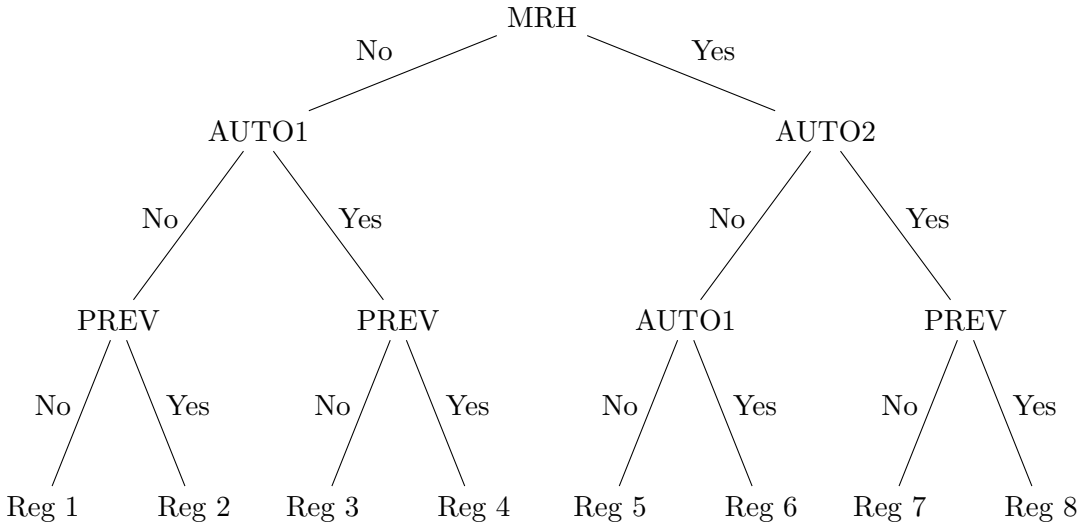


Figure 8: MOB Model - Indiv with a depth of 4

Following the addition of individual variables to our modeling, the predictive power of the models improved considerably compared to the models without the information on individual policyholder characteristics, as one would expect. The improvements in terms of MAE are 27% for the GLM model, 24% for the LASSO, 30% for the CART (with the same depth parameter) and 28% for the MOB (with a depth parameter of 5). We thus obtain on average 25% improvement in the predictive power of our models: the addition of individual variables has a significant impact on the quality of our models.

The table below shows the metrics of the different models selected for each class of model, as well as their calculation time. The MOB model is therefore the best performer, despite a much higher computation time than the other models.

	MSE	MAE	Time
GLM - Indiv	1 166 117	648	54 sec
LASSO - Indiv	1 175 718	650	5 min
CART - Indiv - cp=0.0001	1 127 294	634	6 min
MOB - Indiv - 5	1 105 757	614	54 min

Table 9: Performance metrics of the different models with individual data

Addition of segmentation variables

One could technically add all the individual variables to the list of segmentation variables. However, this would not make much sense for some of them. Indeed, our database contains variables specific to certain contracts (the variable `auto2_energy` for example, which is worth `NA` for all the policyholders who do not own the contract `AUTO2`), so a segmentation according to these variables would not be relevant. On the other hand, some variables are common to all policyholders, such as age or policy duration. We therefore carry out new modelling by adding the variables `age`, `anc`, `canal`, `enfant` and `couple` as candidate segmentation variables.

An initial modeling is performed by setting the maximum tree depth parameter to 3. The model selected the variable `canal` (representinf the distribution channel of a contract) as the first segmentation variable: this variable has therefore a great influence on the margin (the MAE goes from 620

to 614 with the addition of the variable `canal`, that is to say an improvement of about 1 %). The two leaves thus created (one composed of policyholders having subscribed via the internal channel, the other via the external channel) are then segmented according to the `ind_mrh` variable, which was the first segmentation variable in the previous MOB - Indiv models. We conclude that the subscription channel has a greater influence on the margin than the possession of a MRH contract, while age, seniority, the presence of children or a couple have less impact than the possession of a MRH contract. We then increase the maximum depth of the tree, in order to determine whether these new segmentation variables have a greater impact on the margin than the possession of other contracts.

With a maximum depth of 4, the tree performs better (shown in Table 10) than the MOB - Indiv - 4 model. This is due to the use of the variable `canal` as a segmentation variable, which has a strong influence on margin. The same applies to the model with a maximum depth equal to 5: the MAE goes from 614 to 601, an improvement of more than 2%. The variable `canal` thus has a large impact on future margin.

	MSE	MAE	Time
MOB - Indiv - 4	1 109 851	616	46 min
MOB - Indiv - 3 - canal	1 077 173	614	36 min
MOB - Indiv - 4 - canal	1 058 625	604	50 min
MOB - Indiv - 5 - canal	1 052 689	601	75 min
MOB - Indiv - 6 - canal	1 049 209	599	85 min

Table 10: Performance metrics of MOB models

Variable selection

To reduce the number of explanatory variables within each leaf, MOB modeling was run again using the variables selected by *backward* as regression variables. Model performance decreases little compared to MOB models using all variables as regression variables (1 point of MAE on average). For the partitioning variables, the policyholder possession indicators were used, as well as the individual variables common to the policyholders (`age`, `anc`, `canal`, `couple` and `enfant`).

These models become much more readable and interpretable than the previous ones, because the leaves are composed of a reduced number of coefficients. Indeed, the number of regression variables is reduced from 42 to 28. As these variables can have between 2 and 12 modalities, the number of regression coefficients within the leaves is greatly reduced.

Use of the Gamma distribution

Since linear regression with a Gamma distribution requires that the target values have the same sign, we cannot apply the method used previously.

Indeed, the partitioning tree obtained will be potentially different depending on the sample selected. As the tree is created according to the characteristics of the learning base individuals, the variables used for the successive cuts are not necessarily identical, as the samples are different according to whether the margin is positive or negative. Therefore, we cannot simply aggregate 2 MOB models and perform the regressions in the leaves according to the sign of the future margin, since the structures of the 2 trees are not identical.

On the other hand, this approach allows us to highlight the determinants of the margin according to its sign. By creating two partitioning trees, the MOB algorithm segments the samples according

to different variables, which the previous models did not allow us to do, as they were carried out on the whole data set.

We thus observe that the variable `canal` is the most important for policyholders whose estimated future margin is negative, whereas it is the variable `ind_auto2` for those whose estimated future margin is positive.

Conclusion

In this study, we attempted to reconstruct the future margin of a portfolio of policyholders, in order to make it more interpretable. Based on the individual future margin values calculated by the insurer's model over the next 31 years, different statistical models were built to reconstruct this margin and highlight its determinants. This analysis provides a better interpretation of the results, allowing different business applications (which policyholder segments to favour for the underwriting of a new contract, for example).

The MOB models seem to be the best models to answer the problematic of this thesis, as they obtain better performances than the different linear regressions studied, as well as the CART trees. They are also quite easily interpretable, unlike the "black box" models mentioned in chapter 1, which are very efficient but not very interpretable. The choice of the maximum depth of the tree allows to satisfy the different needs of the users: we will increase the depth if a better performance is sought, while we will decrease it if we wish to obtain a better interpretability of the results.

Remerciements

Je remercie tout d'abord Frédéric Planchet, associé chez Prim'Act et tuteur de mon mémoire, pour son accompagnement et les différents conseils prodigués tout au long de la rédaction de ce mémoire.

Je remercie également Quentin Guibert, professeur associé à l'université Paris-Dauphine et consultant chez Prim'Act, pour les recommandations apportées dans le cadre de ce mémoire.

Je tiens aussi à remercier chaleureusement Pierre Cardaliaguet, professeur à l'université Paris-Dauphine, que j'ai eu la chance d'avoir comme enseignant lors de ma L1 dans cette même université, et qui m'a accompagné dans la rédaction de ce mémoire.

Je souhaite remercier le cabinet Prim'Act et tous ses consultants pour leur accueil pendant mon stage, et en particulier Sugiban Ratnasothy, mon tuteur au cabinet et avec qui j'ai réalisé de nombreuses missions.

Enfin, je remercie ma famille pour leur soutien indéfectible depuis toujours, que ce soit dans ma scolarité ou dans ma vie personnelle.

Table des matières

Résumé **3**

Abstract **4**

Note de Synthèse **5**

Synthesis note **17**

Remerciements **29**

Table des matières **31**

Introduction **33**

1 Données et notions mathématiques **35**

1.1 Contexte de l'étude 35

1.2 Segmentation et mutualisation 37

1.3 Présentation des données 39

1.4 Interprétabilité et explicabilité 52

2 Reconstitution de la marge **59**

2.1 Modèle linéaire généralisé 59

2.2 Modélisation GLM 66

2.3 CART 72

2.4 Modélisation CART 77

3 Modèle de partitionnement récursif **81**

3.1 Théorie mathématique 81

3.2 Application des MOB 86
3.3 Amélioration de la prédiction 91

Conclusion **105**

Bibliographie **107**

A Compléments visuels sur les modèles peu inteprétables **109**

Introduction

Une compagnie d'assurance disposant de produits IARD et de produits vie (épargne et prévoyance) a construit des modèles de mesure de la marge de chacun de ces produits. Ce modèle estime la marge individuelle future du stock de contrats (à la date de calcul) sur les 31 prochaines années. Une actualisation est effectuée, et l'assureur obtient ensuite une valeur de la marge future estimée pour les différents contrats des assurés du portefeuille.

Les approches utilisées en IARD et en assurance-vie sont très différentes, du fait de la nature différente des risques : par exemple, le risque financier est déterminant en épargne, faible en IARD. L'assureur souhaite disposer d'une vision agrégée et unifiée de sa marge, donc en fournissant une mesure globale, considérant les produits vie et non-vie.

Les modèles mathématiques du calcul de la marge future ne seront pas directement étudiés dans ce mémoire. Néanmoins, on peut présenter la méthode appliquée dans ses grandes lignes : le modèle estime la marge à l'aide d'un modèle de Markov composé de nombreux états, chaqu'un correspondant à une combinaison des caractéristiques. Le modèle est calibré sur l'expérience de l'assureur, et les probabilités de transition ainsi que le montant des marges futures estimées sont ajustés continuellement. Nous nous intéresserons à reconstituer les marges obtenus par ces modèles, à l'aide des caractéristiques individuelles des assurés.

A l'inverse de ces caractéristiques individuelles, qui sont des données réelles et observables, la marge future est une quantité construite, c'est une estimation sur les 31 prochaines années. Elle a été estimée en se basant sur les caractéristiques individuelles des assurés, ainsi que sur les informations dont dispose l'assureur sur l'historique de ses différents contrats. Il existe donc un fort lien entre caractéristiques individuelles et marge estimées, et c'est ce lien que nous allons exploiter dans cette étude, dans le but de reconstituer la marge le plus fidèlement possible.

L'estimation de la marge réalisée par la compagnie prenant en compte l'historique de l'assureur, on ne peut pas affirmer avec certitude que le lien établit aujourd'hui entre les caractéristiques des assurés et l'estimation de la marge reste le même lors des 31 prochaines années. Des biais pourraient donc apparaître dans le futur, mais cela ne concerne pas cette étude.

Lorsque l'assureur souhaite étudier un segment en particulier, la démarche utilisée actuellement consiste à prendre la moyenne de la marge estimée sur ce segment, car le modèle est complexe et opaque. C'est donc une approche simple mais peu précise. Le but de cette étude est de reconstituer cette marge à l'aide de plusieurs modèles statistiques afin d'en extraire les déterminants tout en restant performant dans sa prédiction. Pour ce faire nous poursuivons les travaux de (Sorochynskyi, 2020) notamment.

Dans un premier temps, nous présenterons le contexte de l'étude et présenterons les données. Nous traiterons aussi de l'importance de l'interprétabilité en assurance. Le chapitre 2 sera consacré à la modélisation de la marge à l'aide de modèles simples, afin d'avoir des points de comparaisons pour

nos modèles plus complexes. Enfin, le chapitre 3 sera consacré aux modèles MOB et aux différentes améliorations de prédiction que nous pouvons mettre en place.

Les modèles MOB ont été introduits par (Zeileis et al., 2008) et offrent un bon compromis entre performance du modèle et interprétabilité des résultats. Ils sont basés sur l'idée suivante : on peut améliorer le pouvoir prédictif d'un modèle en divisant la population en segment homogènes, puis appliquer un modèle différent pour chacun des segments ainsi créés. Les résultats seront donc plus précis qu'en ajustant un unique modèle à l'ensemble des observations. Ces modèles sont appelés "modèles segmentés".

Ils ont été utilisés récemment dans le domaine de l'assurance automobile, pour la tarification dans les travaux de (Boutahar, 2021), et pour la prédictions des coûts automobile et l'influence du choix d'un réseau d'expert dans (Clément, 2022).

Chapitre 1

Données et notions mathématiques

Dans ce chapitre, nous présenterons le contexte de l'étude qui sera effectuée dans ce mémoire. Dans un premier temps, nous donnerons une vision globale du marché de l'assurance non-vie en France, puis les données utilisées dans le mémoire seront présentées. Une bonne compréhension des données est importante dans le cas de cette étude, car les modélisations qui seront effectuées en dépendent fortement. En effet, on cherchera à retrouver la marge future estimée, qui est une quantité construite à partir de ces données. Enfin, nous présenterons certaines notions mathématiques utilisées dans les chapitres suivants.

1.1 Contexte de l'étude

1.1.1 L'assurance non-vie

On peut séparer le secteur de l'assurance en deux catégories :

- L'assurance de biens et de responsabilités (aussi appelée *assurance non-vie*), qui offre un moyen de protection aux particuliers, entreprises ou toute autre entité contre les aléas ne relevant pas de la vie humaine. Elle comprend par exemple l'assurance automobile, habitation ou construction. On l'appelle aussi assurance IARD (Incendie, Accident et Risques Divers). Cette assurance fonctionne sur un principe indemnitaire.

- L'assurance de personnes, qui couvre les risques relevant de la vie humaine. On retrouve dans cette catégorie l'assurance vie, l'assurance santé, la prévoyance ou encore l'assurance emprunteur. Cette assurance fonctionne, sauf exception, sur un principe forfaitaire.

Un contrat d'assurance regroupe plusieurs acteurs :

- l'assuré, sur qui repose le risque,
- le souscripteur, qui paye la prime,
- le bénéficiaire, qui reçoit la prestation,
- l'assureur, qui couvre le risque.

Un même contrat peut être couvert par plusieurs assureurs, on parle alors de co-assurance. L'assuré, le souscripteur et le bénéficiaire peuvent être la même personne. Chaque contrat contient des conditions générales (catalogue de couverture, exclusions, éléments couverts, etc.) et des conditions particulières (montant de la prime, période de couverture, garanties souscrites, etc.).

En France, les cotisations du secteur de l'assurance IARD s'élève à 94,7 Mrd € (FranceAssureur, 2020), classant le pays au sixième rang mondial en terme de montants de cotisations. La Figure 1.1 représente le marché de l'assurance non-vie dans le monde.

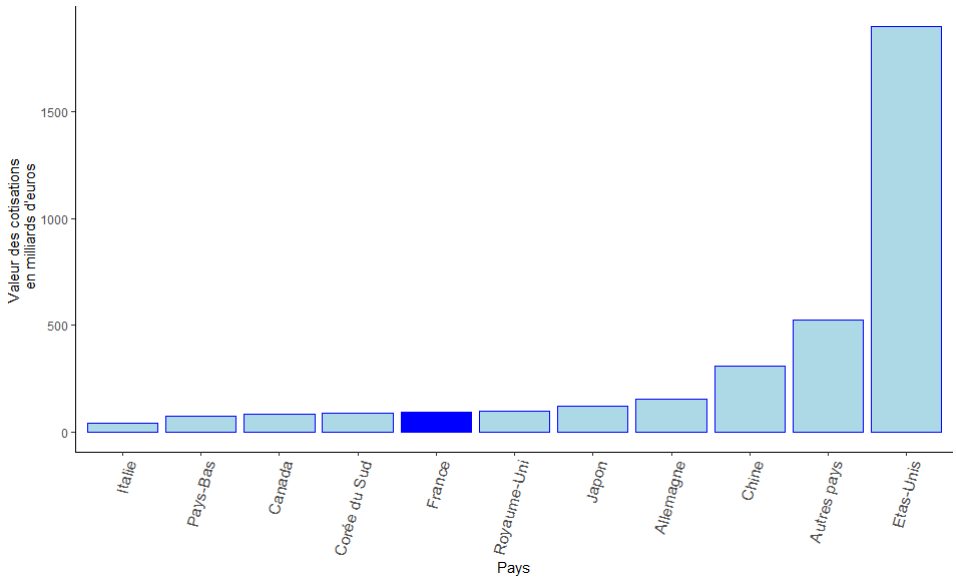


FIGURE 1.1 : Cotisations du marché assurantiel mondial (FranceAssureur, 2020)

En 2018, le secteur de l'assurance de biens et de responsabilités représentait un peu plus d'un quart des cotisations d'assurance en France (Statista, 2018). Cette proportion reste constante depuis quelques années. La Figure 1.2 représente les proportions des différents secteurs de l'assurance en France en 2018, l'assurance de personnes ayant été divisées en deux secteurs.

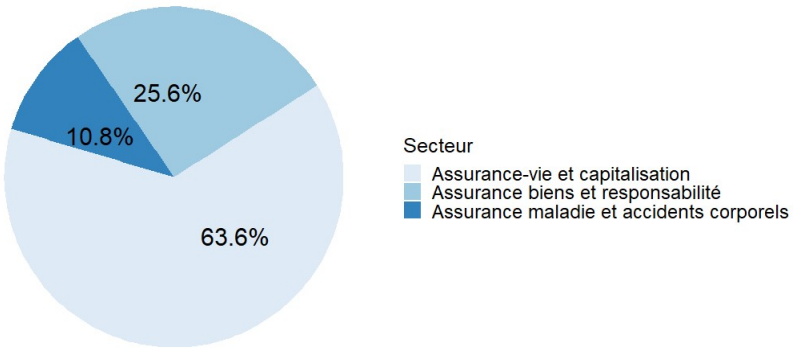


FIGURE 1.2 : Répartition des secteurs de l'assurance en 2018 (Statista, 2018)

La mesure de la rentabilité est différente entre un produit d'assurance vie et non-vie. En vie, la marge vient essentiellement d'une stratégie de volume. En effet, l'assureur devant reverser un minimum de 85% des gains financiers réalisés en plaçant l'épargne des assurés, sa marge dépend fortement de la

quantité de contrats qu'il possède dans son portefeuille. En assurance non-vie, la marge vient plutôt de la justesse des modèles mathématiques qui ont estimé la sinistralité future des assurés. Ainsi, la qualité de souscription des risques assurés est un enjeu majeur en IARD.

1.1.2 Estimation de la marge

L'assureur ayant fournit les données de cette étude a créé un modèle d'estimation de la marge future de son stock de contrat. Afin de justifier la fiabilité des données, on indiquera seulement que l'assureur, qui souhaite rester anonyme, fait partie des 10 plus grosses compagnies d'assurance dommages en France, avec un chiffre d'affaire supérieur à 2 Mrd d'€ en 2020. Ainsi, les données utilisées sont basées sur un très grand nombre d'assurés, lissant le risque d'avoir des données atypiques. On ne possède pas le modèle mathématique utilisé, mais il peut être brièvement résumé : la marge est estimée selon un modèle markovien avec de nombreux états, chaque état correspondant à une combinaison de caractéristiques. Les montants de marge estimés ainsi que les probabilités de transitions inter-états sont calibrés sur l'expérience de l'assureur, et ajustés en permanence. La marge ayant été estimée en fonction des caractéristiques individuelles des assurés, il existe une forte corrélation entre ces caractéristiques et la marge fournie par l'assureur. Ainsi, si des biais existaient lors de la construction de la marge par l'assureur, ces biais seront probablement aussi existants dans notre modélisation.

L'assureur a fourni une estimation de la marge annuelle de son stock de contrats, et ce sur un horizon de 31 ans. On possède donc une base de données avec les 31 valeurs annuelles de marge estimées, pour chaque *model point* du portefeuille. Cette marge n'a pas non plus été actualisée, elle le sera au taux de 6% dans le cadre de cette étude (cette valeur a été fixée arbitrairement en se basant sur le mémoire d'Oleksandr Sorochynskyi (Sorochynskyi, 2020))

Nous ne nous attarderons pas sur le calcul de cette marge, effectué par le modèle de la compagnie, car ce n'est pas l'objectif de cette étude. Nous chercherons à la reproduire à l'aide des caractéristiques des assurés, et ne nous intéresserons pas à savoir si cette estimation est fiable ou non. Néanmoins, il est à noter que ce modèle d'estimation de la marge est utilisé depuis plusieurs années par l'assureur afin de quantifier l'état de son portefeuille ; on peut donc supposer une certaine fiabilité du modèle.

L'assureur a estimé la marge en se basant sur des *model points*. Cela signifie que la marge n'a pas été estimée au niveau des individus, mais plutôt sur des regroupements d'individus ayant des caractéristiques proches. Ce choix de regrouper les individus sous des *model points* permet de diminuer la taille de la base de données ainsi que le temps de calcul. Néanmoins, dans le cadre de notre étude, nous modéliserons la marge au niveau de chaque individu. Ainsi, 2 individus appartenant au même *model point* auront une marge estimée par la compagnie identique (puisqu'un seul calcul a été effectué, ces 2 individus ayant été regroupés ensemble), mais pourront avoir une marge estimée différente l'un de l'autre, car elles auront été estimées individuellement par nos modèles.

1.2 Segmentation et mutualisation

Principe de mutualisation

En assurance, le principe de mutualisation des risques (Charpentier, 2015) est fondamental. Tous les assurés paient une prime à l'assureur, qui en échange les protège contre un potentiel sinistre. Si un sinistre se déclare, l'assureur indemnise l'assuré sinistré, à l'aide des primes récoltées auprès de tous les assurés. Ainsi, un assuré qui ne subit pas de sinistre aura payé une prime pour rien, mais aurait

été indemnisé au-delà du montant de cette prime s'il avait déclaré un sinistre. Les assurés se partagent donc les risques : c'est le principe de mutualisation.

Le profit (et la solvabilité) de l'assureur sont basés sur un aléa : la survenance ou non d'un sinistre. S'il n'y pas survenance, l'assureur réalise un petit bénéfice en encaissant la prime, mais s'il y a survenance du sinistre, il réalise une forte perte en indemnisant l'assuré. Ainsi, il a intérêt à posséder dans son portefeuille de nombreux contrats afin d'appliquer le principe de mutualisation. Cette stratégie est basé sur deux théorèmes "asymptotiques" ou "limites" : la Loi des Grands Nombres (LGN) et le Théorème Central Limite (TCL), énoncés ci-dessous (Charpentier, 2011).

Théorème 1 (Loi des Grands Nombres). Soit (X_n) une suite de variables aléatoires indépendantes identiquement distribuées, d'espérance μ finie. Alors

$$\forall \epsilon > 0, \mathbf{P}\left(\left|\bar{X}_n - \mu\right| > \varepsilon\right) \underset{n\to+\infty}{\longrightarrow} 0,$$

avec

$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i. \tag{1.1}$$

Ce théorème peut s'appliquer à de nombreux domaines ; dans le cas d'un assureur, les variables aléatoires X_i correspondent aux montants des sinistres (qui peuvent être nuls s'il n'y a pas de sinistre), et le théorème s'interprète comme suit : pour un grand nombre de risques indépendant et de même espérance, la moyenne empirique converge en probabilité vers l'espérance μ des sinistres.

Théorème 2 (Théorème Central Limite). Soit (X_n) une suite de variables aléatoires indépendantes identiquement distribuées, d'espérance μ finie et de variance σ^2 finie. Alors

$$\sqrt{n}\frac{\left(\bar{X}_n - \mu\right)}{\sigma} \underset{n\to+\infty}{\xrightarrow{\mathcal{L}}} \mathcal{N}(0,1).$$

Dans le cas de l'assurance, on interprète ce théorème comme suit : si l'assuré possède un grand portefeuille de risques de même espérance finie et de même variance finie, alors la charge globale des sinistres converge en loi vers une loi gaussienne. Ce théorème est la base du principe de mutualisation présenté plus haut.

Segmentation en assurance

En assurance, la segmentation permet d'affecter une prime pure plus adaptée aux différents assurés, selon qu'ils appartiennent à un segment ou un autre. En effet, un portefeuille d'assurés peut être très hétérogène, avec de "bons risques" et de "mauvais risques". La prime pure étant l'espérance d'un sinistre en moyenne, les bons risques se retrouvent à payer pour les mauvais risques. La segmentation permet de séparer ce portefeuille hétérogène en segments homogènes, et ainsi de tarifer une prime pure plus juste pour assurés, variant en fonction du segment. Les bons risques possédant une probabilité plus faible d'avoir un sinistre, leur prime pure sera donc légitimement plus faible (et inversement pour les mauvais risques).

Un assureur a tout intérêt à segmenter sa population, comme on peut le voir dans les deux cas de figure suivants :

- *L'assureur A ne segmente pas son portefeuille.* Il applique donc le principe de mutualisation à l'ensemble du portefeuille. Tous les assurés paient la même prime pure μ. Les bons risques paient

donc une prime supérieure à l'espérance de leur sinistre, et les mauvais risques paient une prime inférieure à l'espérance de leur sinistre. Les bons risques vont donc se tourner vers un autre assureur (assureur B) proposant une segmentation, et donc un tarif plus faible que μ. L'assureur A se retrouve donc avec les mauvais risques pour une prime pure μ trop faible par rapport à sa nouvelle population. Ce cas illustre le phénomène d'anti-sélection.

– *L'assureur B segmente son portefeuille.* L'assureur B sépare son portefeuille hétérogène en plusieurs segments homogènes (2 ou plus). Il propose donc un tarif inférieur à μ à ses bons risques, et un tarif supérieur à ses mauvais risques. Il se protège ainsi contre l'anti-sélection : les bons risques restent chez lui pour profiter du principe de mutualisation sur un segment homogène, et les mauvais risques peuvent soit rester et appliquer ce même principe de mutualisation sur leur segment homogène (avec un tarif supérieur à μ), soit partent chez l'assureur A qui ne segmente pas, ce qui est bénéfique à l'assureur B.

Ces deux situations illustrent l'intérêt pour un assureur de segmenter sa population, afin de proposer un tarif concurentiel et de limiter l'anti-sélection. Néanmoins, cette méthode de segmentation présente certains limites.

Limites de la segmentation

Au regard de l'exemple précédent, on pourrait penser que l'assureur a intérêt à segmenter son portefeuille de nombreuses fois, afin de proposer la prime la plus juste possible aux assurés et d'éviter que les bons risques ne s'en aillent. Cette segmentation à outrance a des limites, notamment celle de ne pas respecter l'hypothèse d'un nombre élevé d'individus dans la LGN et le TCL. Le principe de mutualisation ne sera donc plus applicable. Une autre limite est qu'une grande segmentation complexifie les modèles mathématiques utilisés, ainsi que leur interprétabilité. Dans le cas des modèles MOB, que nous présenterons dans le chapitre 3, une trop forte segmentation de la population implique de créer un nouveau modèle de régression pour chaque segment, ce qui augmente fortement le temps de calcul.

L'assureur doit donc trouver un bon compromis entre l'absence de segmentation et une segmentation trop grande, en prenant en compte l'aspect opérationnel que cette segmentation implique, en terme d'interprétabilité et de temps de calcul.

1.3 Présentation des données

Les travaux présentés dans ce mémoire se baseront sur les données transmises par la compagnie d'assurance, qui ont été retraitées dans la suite. Elles correspondent aux assurés ayant souscrit un contrat avant 2020. Ces données ont été transmises sous la forme de 6 tables, décrites ci-dessous :

1. une table contenant les marges futures estimées pour le produit `MRH`,
2. une table contenant les marges futures estimées pour le produit `AUTO`,
3. une table contenant les marges futures estimées pour le produit `PREV`,
4. une table contenant les marges futures estimées pour le produit `CORP`,
5. une table permettant de faire la jointure entre les contrats et les assurés, que nous appellerons "ASSURÉS" dans la suite,
6. une table contenant les caractéristiques individuelles des assurés.

Le produit `MRH` est un contrat Multirisques Habitation, le produit `AUTO` correspond à un contrat Automobile, le produit `CORP` est un contrat d'assurance corporelle et le produit `PREV` est un contrat de Prévoyance. Ces produits seront explicités plus en détails dans la suite du chapitre. De même, deux autres produits `EPARGNE` et `DECES` seront mentionnés, correspondant respectivement à un contrat d'épargne et un contrat d'assurance-vie.

La Figure 1.3 représente ces différentes tables. La table 6, une fois enrichie des autres tables, devient la base finale utilisée pour les calculs dans la suite du mémoire.

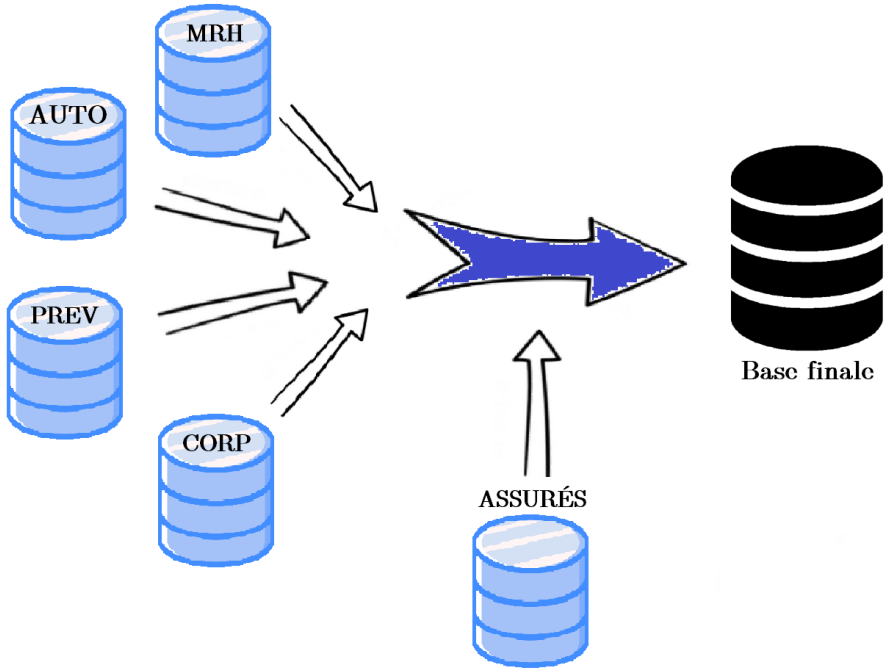


FIGURE 1.3 : Représentation des différentes tables et de la base de données

Nous allons maintenant présenter les variables des différents contrats. Certaines variables ne sont pas spécifiques à un type de contrat et seront donc présentées à part, afin de ne pas les réécrire 4 fois (c'est le cas par exemple de l'`âge` de l'assuré). Certaines variables classiquement quantitatives (l'âge par exemple) sont renseignées comme variables qualitatives, cela signifie que la variable a été divisée en plusieurs tranches.

Une analyse des données plus poussée a déjà été effectuée dans le mémoire d'Oleksandr Sorochynskyi (SOROCHYNSKYI, 2020). Cette section présentera uniquement les différentes modalités des variables de la base de données ainsi que leurs proportions respectives, afin que le lecteur puisse se faire une idée de la population du portefeuille d'assurés.

Pré-traitement des données

Les pré-traitements sur la base de données ont aussi été effectués par Oleksandr Sorochynskyi, et sont brièvement explicités.

Pour chaque contrat, la compagnie a fourni une base de données contenant une colonne identifiant et 31 colonnes de marge (correspondant aux valeurs de marge future estimées par le modèle de la compagnie). On rappelle que la marge future est estimée au niveau des *model points*, et non au niveau des assurés. La table "ASSURÉS", contenant les identifiants des assurés ainsi que ceux des *model*

points, permet d'effectuer la jointure et d'affecter les marges des *model points* aux identifiants assurés correspondants.

Les données contenant les marges futures estimées des contrats ont été actualisées et cumulées par la compagnie. Il a été décidé de les "désactualiser" et les "décumuler" pour faciliter le traitement informatique. Les marges ont ensuite été actualisées à 6% (Sorochynskyi, 2020) et cumulées.

On obtient ainsi 4 tables retraitées (correspondant aux contrats `AUTO`, `MRH`, `PREV` et `CORP`) composées d'un identifiant assuré et de la marge future estimée sur le contrat.

La table numéro 6 a aussi été retraitée avant d'être enrichie des autres tables. En effet, les assurés ne possédant pas tous des contrats de chaque nature, il y a des informations manquantes dans cette table. Par exemple, si un assuré ne possède pas de contrat `MRH`, toutes les variables relatives à ce contrat auront la modalité `NA`. Cependant, la table reçue contient aussi, pour certaines variables, une modalité `Indéterminé`, qui peut être expliquée par le fait que l'assureur ne possède pas toujours la totalité des informations relatives à ces assurés. Il a été décidé, lorsque la modalité `NA` était due au fait que l'assuré ne possédait pas le contrat en question, de recoder ces `NA` par une modalité `Absence de MRH` (ou `Absence de AUTO/CORP/PREV` pour les autres contrats). Lorsqu'une variable possède un ordre naturel (variable ordinale), il a été décidé de placer la modalité `Absence de MRH` en premier, par convention. Par exemple, si l'on avait une variable (imaginaire) `surface du bien assuré`, prenant les modalités `0-50`, `151-200`, `51-100`, `200 et plus` et `101-150`, l'ordre aurait été { `Absence de MRH` → `0-50` → `51-100` → `101-150` → `151-200` → `200 et plus`}.

Les modalités `Indéterminé` ont ensuite été transformées en `NA` afin de traiter l'absence de données. On ne perd pas en information sur les données en effectuant cette manipulation, car si une observation possédait la modalité `Indéterminé` pour une variable relative au contrat `AUTO1` par exemple, on a toujours l'information sur la détention de ce contrat via la variable `indauto1`.

Une analyse de la proportion des modalités `NA` a été effectuée, et un peu moins de 5% des observations en possèdent une. Ces modalités `NA` sont regroupées sur les variables suivantes

Nom de la variable	Nombre de valeurs NA	Description de la variable
auto2ancpermis	75 004	Ancienneté du permis
auto2agecond	56 345	Age du conducteur
auto1ancpermis	33 126	Ancienneté du permis
auto1agecond	15 887	Age du conducteur
mrh2lieuutil	11 182	Présence d'un lieu utilitaire
mrh2naturelri	7 829	Nature du bien assuré
auto1energie	18	Type d'énergie du moteur du véhicule assuré
auto2energie	15	Type d'énergie du moteur du véhicule assuré

Table 1.1 : Variables contenant des modalités NA dans la table ASSURÉS

Pour les 4 premières variables du tableau 1.1, il a été décidé de créer une modalité `(NA)`, car la proportion de `(NA)` est significative par rapports aux autres modalités de ces variables.

A l'aide la fonction `missRanger` du package `ranger` (Wright et Ziegler, 2017), on a imputé des valeurs pour les modalités `NA` des variables `mrh2_lieuutil` et `mrh2_naturelri`, afin de ne pas supprimer toutes les observations contenant une modalité `NA`. Sans entrer dans le détail de la fonction `missRanger`, celle-ci crée une forêt aléatoire en se basant sur les observations complètes, et prédit une valeur pour remplacer les `NA` en fonction des autres covariables. Plusieurs itérations sont effectuées

jusqu'à ce que l'erreur de prédiction atteigne un seuil prédéfini.

Il reste ensuite seulement 33 observations possédant une modalité `NA`, qui ont été supprimées de la base de données.

Répartition du nombre de contrats possédés

Au sein du portefeuille, les assurés peuvent détenir entre 1 et 7 contrats (`AUTO1`, `AUTO2`, `MRH`, `CORP`, `PREV`, `EPARGNE` et `DECES`). Le Tableau 1.2 présente les proportions d'assurés en fonction du nombre de contrats possédés.

Nombre de contrats détenus	Proportion
1 contrat	15,4 %
2 contrats	22,0 %
3 contrats	20,2 %
4 contrats	22,3 %
5 contrats	15,9 %
6 contrats	3,7 %
7 contrats	0,5 %

TABLE 1.2 : Proportions d'assurés en fonction du nombre de contrats

Variables communes à tous les contrats

Nom de la variable	Contenu	Type
id	Identifiant de l'assuré	Quantitatif
anc	Ancienneté de l'assuré dans le portefeuille	Qualitatif
age	Age de l'assuré	Qualitatif
canal	Canal par lequel l'assuré a souscrit son contrat	Binaire
enfant	L'assuré a des enfants ou non	Booléen
couple	L'assuré est en couple ou non	Booléen
detention_epargne	L'assuré possède un contrat EPARGNE dans la compagnie	Booléen
detention_deces	L'assuré possède un contrat DECES dans la compagnie	Booléen

TABLE 1.3 : Variables communes à tous les contrats

id : Identifiant anonyme de l'assuré. Permet de faire la jointure entre les tables contenant les caractéristiques des assurés et celles contenant la marge estimée des différents contrats.

anc : Ancienneté de l'assuré dans le portefeuille, divisée en sept tranches. Les proportions des différentes modalités sont présentées dans le Tableau 1.4, ainsi que des boîtes à moustache représentant la marge future en fonction de l'ancienneté dans la Figure 1.4.

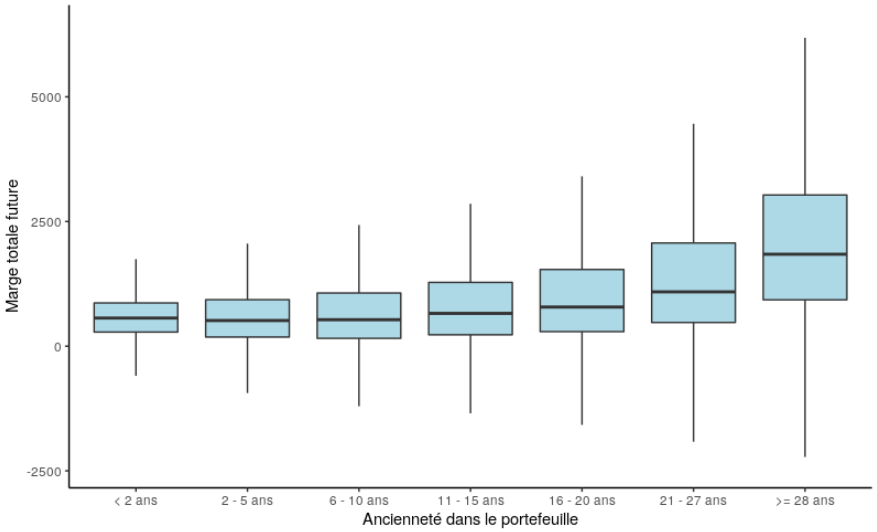


FIGURE 1.4 : Marge future estimée en fonction de l'ancienneté dans le portefeuille

Modalité	Proportion
$<$ 2 ans	0,03
2 - 5 ans	0,15
6 - 10 ans	0,13
11 - 15 ans	0,10
16 - 20 ans	0,10
21 - 27 ans	0,14
$\geqslant$ 28 ans	0,35

TABLE 1.4 : Ancienneté de l'assuré dans le portefeuille

âge : Âge de l'assuré, divisé en huit tranches de 5 ans, ainsi que deux tranches plus larges pour les assurés plus jeunes et plus vieux. La première tranche commence à 18 ans car les assurés doivent être majeurs. La figure 1.5 résume les proportions de ces modalités.

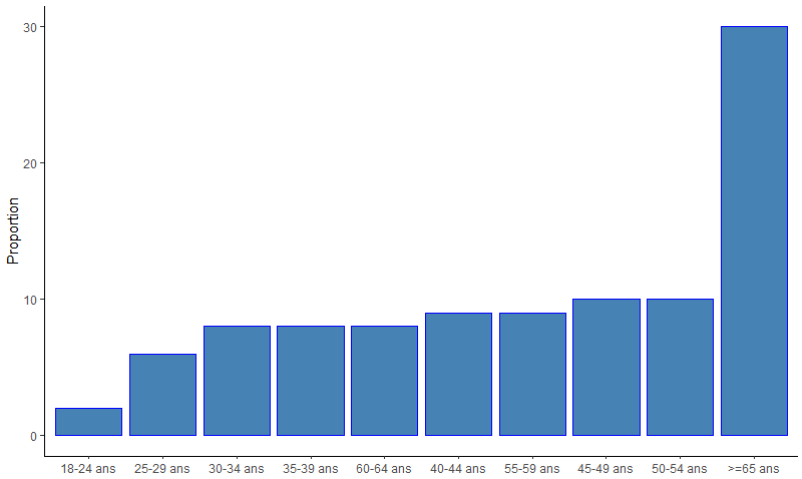


FIGURE 1.5 : Age des assurés

couple et enfant : Indique si l'assuré est en couple (marié, Pacsé, etc.) et s'il a au moins un enfant. La figure 1.6 récapitule les proportions de ces modalités.

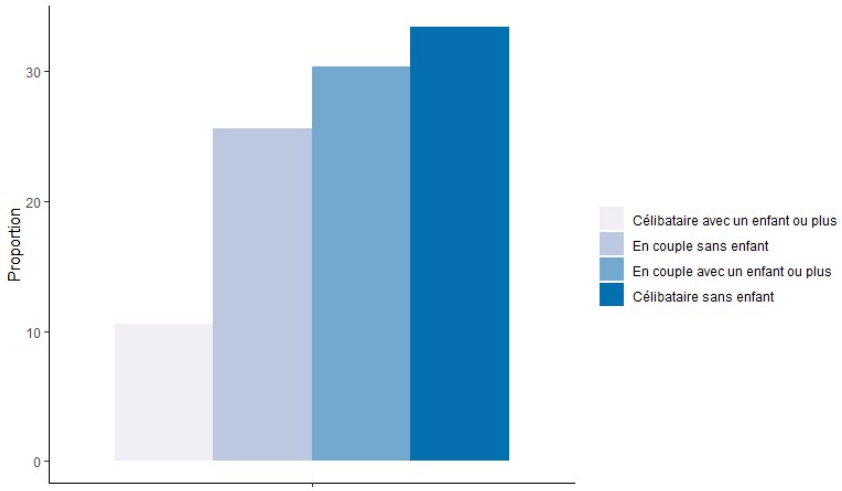


FIGURE 1.6 : Répartition Couple et Enfant

canal : Canal de souscription utilisé par l'assuré. $29,6\%$ des assurés ont souscrit via le canal externe, et $70,4\%$ des assurés ont souscrit via le canal interne. Le canal externe est disponible pour tout le monde, c'est le canal "grand public", tandis que le canal interne est seulement disponible pour une population précise de clients, avec un tarif préférentiel. Cette variable a une certaine importance sur la marge : la marge future des assurés passant par le canal externe est en moyenne de 1091, tandis que celle des assurés passant par le canal interne est en moyenne de 1464, soit une augmentation de plus de 34%.

detention_epargne et detention_deces : Variables de type booléen indiquant si l'assuré détient un contrat EPARGNE ou DECES dans la compagnie. La compagnie n'a pas communiqué de données quant à l'estimation de la marge future de ces 2 contrats, mais le fait d'en posséder un (ou les deux) peut être un déterminant de la marge future des autres contrats. En effet, savoir qu'un assuré possède un contrat épargne ou un contrat décès est une information sur son niveau de vie, qui peut avoir une influence sur le type de contrats souscrit, et donc sur la marge future. $8,2\%$ des assurés possèdent le contrat EPARGNE et $5,7\%$ possèdent le contrat DECES. Le tableau 1.5 présente une répartition plus précise de la détention de ces contrats.

		DECES	
		Oui	Non
EPARGNE	Oui	$1,3\%$	$6,9\%$
	Non	$4,4\%$	$87,5\%$

TABLE 1.5 : Détention contrat EPARGNE et DECES

1.3.1 Variables contrat `AUTO`

La compagnie propose plusieurs types de formules de contrats automobile. Un assuré possédant deux voitures devra souscrire 2 contrats automobiles différents, avec potentiellement des garanties différentes. Les données transmises par la compagnie nous informent sur les caractéristiques des 2 plus gros contrats de l'assuré, sous la dénomination `AUTO1` pour le premier contrat et `AUTO2` pour le se-

cond. Si l'assuré ne possède qu'un seul contrat automobile, celui-ci est appelé `AUTO1`. Enfin, si l'assuré possède plus de 2 contrats, nous n'avons pas d'informations détaillées sur ceux-ci, et possédons seulement une information sur le nombre de contrats supplémentaires à travers la variable `auto_vehsup`.

Dans le tableau qui suit, un "#" dans le nom d'une variable signifie que cette variable existe 2 fois, et le "#" est à remplacer par les chiffres "1" et "2" selon que l'on veut la variable correspondant au contrat `AUTO1` ou `AUTO2`.

Nom de la variable	Contenu	Type
auto_vehsup	Nombre de véhicule supplémentaire assurés	Booléen
detention_auto#	Détention du contrat `AUTO#`	Booléen
auto#_energie	Type d'énergie du véhicule	Qualitatif
auto#_agecond	Age du conducteur	Qualitatif
auto#_formule	Type de formule souscrite	Qualitatif
auto#_crm	Coefficient Bonus/Malus du conducteur	Qualitatif
auto#_categorie	Catégorie du véhicule assuré	Qualitatif
auto#_puissance	Puissance du véhicule assuré	Qualitatif
auto#_ancpermis	Ancienneté de permis de conduire de l'assuré	Qualitatif

TABLE 1.6 : Variables relatives au contrat `AUTO`

auto_vehsup : Cette variable indique le nombre de contrats supplémentaires aux 2 contrats `AUTO1` et `AUTO2` des assurés. La variable prend les modalités `0` et `1` dans notre jeu de données. Si `auto_vehsup=1`, cela signifie donc que l'assuré possède 3 contrats auto (`AUTO1`, `AUTO2` et un troisième contrat sans dénomination). Si `auto_vehsup=0`, alors l'assuré possède 0, 1 ou 2 contrats, et on retrouve cette information à l'aide des variables booléennes `detention_auto1` et `detention_auto2`, présentées ci-après.

detention_auto# : Cette variable indique si l'assuré détient les contrats `AUTO1` et/ou `AUTO2`. Ce sont des contrats identiques mais portant sur des véhicules différents. Par convention, pour l'estimation des marges futures, si un assuré possède ces 2 contrats, `AUTO1` désigne le contrat du véhicule le plus fréquemment utilisé. Une répartition des différents contrats est présentée dans la figure 1.7.

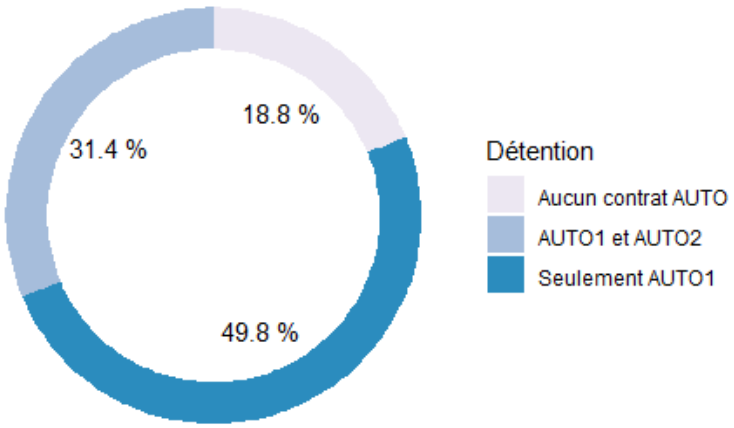


FIGURE 1.7 : Détention des contrats `AUTO1` et `AUTO2`

auto#energie : Variable indiquant le type d'énergie utilisée dans le moteur du véhicule assuré. Au total, sur tous les véhicules assurés sous les contrats `AUTO1` et `AUTO2` (on rappelle qu'on ne possède pas d'informations détaillées pour les contrats supplémentaires), 49,1% des véhicules fonctionnent au diesel, 43,1% à l'essence, 7,4% ne sont pas des 4 roues et 2,4% sont renseignés par `Autres`.

auto#_agecond : Cette variable donne l'âge du conducteur principal du véhicule assuré par le contrat. Les proportions de cette variables sont indiquées dans la Figure 1.8.

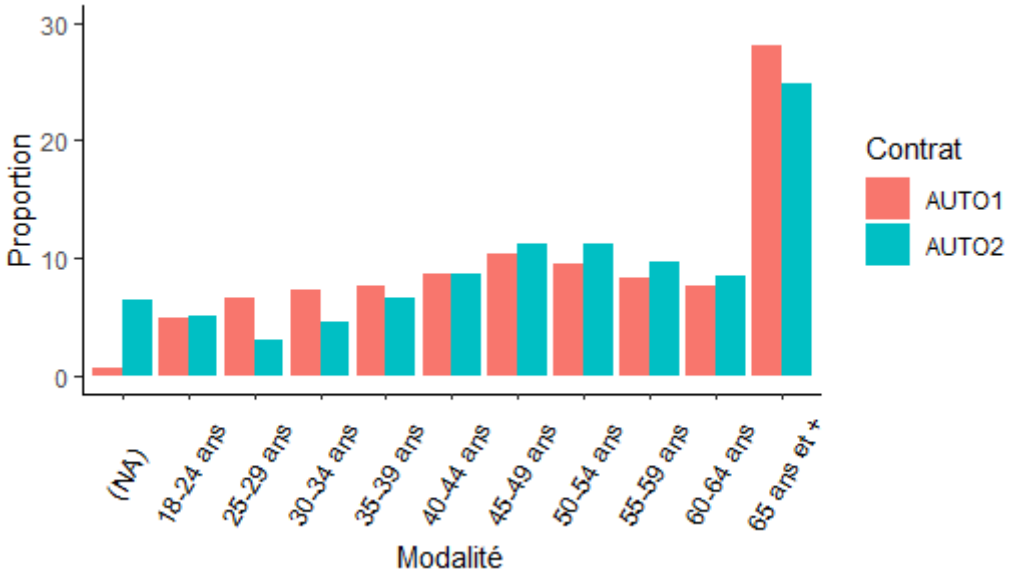


FIGURE 1.8 : Âge du conducteur principal par contrat

auto#_formule : Cette variable indique la formule souscrite par l'assuré. Elle prend les modalités `1` à `8`, qui représentent les différentes formules proposées par la compagnie, mais qui ont été anonymisées dans le cadre de ce mémoire.

auto#_crm : Indique le "Bonus/Malus" de l'assuré (CRM est l'acronyme de Coefficient de Réduction de Majoration). Le tableau 1.7 représente les proportions des différentes modalités. Le CRM commence à 100, augmente en cas d'accident et baisse en leur absence. La majorité des assurés ont la modalité `50 TBR`, qui signifie qu'ils ont atteint la valeur de 50 depuis longtemps, et qu'ils sont ainsi de Très Bons Risques.

Modalité	Proportion `AUTO1`	Proportion `AUTO2`
50 TBR	58, 4%	63, 8%
50	7, 2%	6, 4%
51 - 84	22, 6%	12, 9%
85 - 99	6, 6%	4, 1%
100	1, 6%	1%
plus_de_100	0, 5%	0, 5%
Aucun	2, 2%	11, 3%

TABLE 1.7 : Répartition des CRM par contrat

auto#_categorie : Variable indiquant la catégorie du véhicule, avec les modalités `4 roues`, `2 roues≤125`, `2 roues>125` et `Autres`. Pour le contrat `AUTO1`, 95,7% sont des véhicules à `4 roues`, tandis que seulement 84,4% des véhicules du contrat `AUTO2` sont des `4 roues`.

auto#_puissance : Indique la puissance du véhicule assuré. Le tableau 1.8 présente la répartition des différentes modalités de la variable en fonction du contrat souscrit.

Modalité	Proportion `AUTO1`	Proportion `AUTO2`
Non 4 roues	4,3%	15,6%
< 50	0,3%	1,1%
50 - 59	0,9%	1,9%
60 - 79	23,3%	30,6%
80 - 99	24%	20%
100 - 119	24,4%	17,2%
120 - 149	15,3%	9,5%
⩾ 150	7,5%	4,2%

TABLE 1.8 : Puissance du véhicule par contrat

auto#_ancpermis : Cette variable indique l'ancienneté de permis de l'assuré. Pour les 2 contrats `AUTO1` et `AUTO2`, plus de 70% des assurés possèdent leur permis depuis plus de 21 ans, dont 56% depuis plus de 28 ans.

1.3.2 Variables contrat `MRH`

Un contrat `MRH` (MultiRisques Habitation) est un contrat proposant un large choix de garanties qui permettent de protéger le patrimoine familial de l'assuré. La compagnie propose 2 contrats `MRH`, qui sont exclusifs : un assuré ne peut pas posséder à la fois le contrat `MRH1` et le contrat `MRH2`. Ce sont deux contrats différents, avec des variables différentes en fonction du contrat. Dans le portefeuille étudié, 25,8% des assurés détiennent le contrat `MRH1`, 42,2% le contrat `MRH2` et 32% ne détiennent ni `MRH1` ni `MRH2`. Cela ne signifie pas que ces assurés ne détiennent pas de contrat auprès de la compagnie. Dans la suite du mémoire, pour la partie modélisation, on considérera uniquement la détention d'un contrat `MRH`, indépendemment de `MRH1` ou `MRH2`.

Les statistiques descriptives effectuées dans cette partie ne prennent pas en compte les assurés ne possédant pas le contrat concerné. Ainsi, pour calculer la répartition des modalités de la variable `mrh1_nbprop`, les assurés détenant le contrat `MRH2` ne sont pas pris en compte.

Nom de la variable	Contenu	Type
mrh2_uhsup	Nombre de contrat MRH supplémentaires détenus par l'assuré	Quantitatif
mrh1_nbprop	Nombre de propriétés couvertes par le contrat	Qualitatif
mrh1_nbloc	Nombre de propriétés assurées en location	Qualitatif
mrh1_tranchemob	Tranche mobilière de la propriété assurée	Qualitatif
mrh_#lieuutil	Présence d'un lieu utilitaire (ex : garage)	Qualitatif
mrh2_formule	Type de formule souscrite	Qualitatif
mrh2_patmob	Patrimoine mobilier couvert par le contrat	Qualitatif
mrh2_nbpieces	Nombre de pièces de la propriété	Qualitatif
mrh2_sitjur	Situation juridique de l'assuré vis-à-vis de la propriété	Qualitatif
mrh2_naturelri	Exposition aux risques naturels	Qualitatif

mrh2_uhsup : Variable indiquant le nombre de contrats MRH2 détenus en plus du premier. C'est l'équivalent de la variable auto_vehsup pour le contrat AUTO. Parmi les détenteurs du contrat MRH2, $28,6\%$ possèdent un second contrat MRH2, et aucun ne possède 3 contrats MRH2.

mrh1_nbprop : Cette variable indique le nombre de propriétés couvertes par le contrat MRH. Le Tableau 1.9 présente les proportions de cette variable.

Modalité	Proportion
0 propriété	$12,3\%$
1 propriété	$56,8\%$
2 propriétés	$21,2\%$
3 ou plus	$9,7\%$

TABLE 1.9 : Nombre de propriétés assurées par le contrat MRH1

mrh1_nbloc : Cette variable indique si l'un des biens assurés est en location. $23,2\%$ des contrats MRH1 possèdent au moins un bien en location.

mrh1_tranchemob : Indique la tranche mobilière couverte par le contrat. Cette variable est importante dans la détermination de la marge car la valeur du bien assuré impacte fortement la prime payée par l'assuré, mais aussi le montant des remboursements en cas de sinistre. Le Tableau 1.10 représente les différentes tranches ainsi que leur répartition.

Modalité	Proportion
A	$12,4\%$
B	$25,4\%$
C	$30,5\%$
D	$17,4\%$
E	$7,9\%$
F ou plus	$6,4\%$

TABLE 1.10 : Tranches mobilières du contrat MRH1

mrh#_lieuutil : Variable indiquant la présence d'un lieu utilitaire dans le bien assuré. Le lieu utilitaire peut être un garage ou un abri de rangement par exemple. $89,7\%$ des biens assurés par le contrat `MRH1` ne possèdent pas de lieu utilitaire, $92,5\%$ pour le contrat `MRH2`.

mrh2_formule : Variable indiquant la formule souscrite par l'assuré au sein du contrat `MRH2`. 4 formules sont proposées, représentées par des chiffres de 1 à 4, la formule 4 étant la plus complète et la formule 1 la moins complète. Le Tableau 1.11 représente les proportions des différentes formules parmi les détenteurs du contrat `MRH2`.

Modalité	Proportion
1	$5,7\%$
2	$17,9\%$
3	$58,0\%$
4	$18,4\%$

TABLE 1.11 : Formules du contrat `MRH2`

mrh2_patmob : Indique la tranche du patrimoine mobilier couvert par le contrat `MRH2`. Le patrimoine mobilier est défini comme l'ensemble des biens matériels possédés par l'assuré : meubles, biens électronique, électroménager, matériel de sport, etc. Les valeurs de patrimoine mobilier ont été découpées en tranches, dont la répartition est représentée dans le Tableau 1.12.

Modalité	Proportion
Pas de patrimoine mobilier	$5,1\%$
A1	$14,7\%$
A2	$18,9\%$
AA	$6,1\%$
B1 ou B2	$25,1\%$
C1 ou C2	$19,8\%$
D ou plus	$10,3\%$

TABLE 1.12 : Tranches du patrimoine mobilier assuré par le contrat `MRH2`

mrh2_sitjur : Cette variable indique la situation juridique de l'assuré par rapport au bien assuré. Les modalités possibles sont `Locataire`, `Assurés ailleurs`, `Propriétaire occupant` et `Propriétaire bailleur`. La modalité `Propriétaire occupant` représente $64,5\%$ du portefeuille, et `Locataire` $30,1\%$ du portefeuille.

mrh2_naturelri : Variable indiquant la nature du bien assuré, qui peut être déterminante en cas de risque climatique. $58,6\%$ des biens du contrat `MRH2` sont des `Maisons`, $40,6\%$ sont des `Appartements` et $0,8\%$ sont catégorisés comme `Autres`.

mrh2_nbpieces : Indique le nombre de pièces du bien assuré. La répartition au sein du contrat `MRH2` est représenté dans la Figure 1.9.

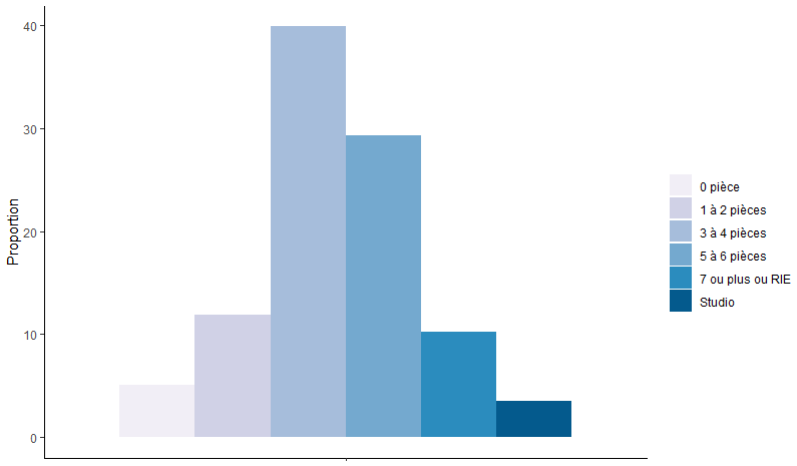


FIGURE 1.9 : Nombre de pièces du bien assuré

1.3.3 Variables contrat `PREV`

Les contrat `PREV1` et `PREV2` sont 2 contrats exclusifs de PREVoyance proposés par la compagnie. Ce sont des contrats avec des garanties différentes. Le produit `PREV1` a 2 options, tandis que le produit `PREV2` n'en a aucune. Ainsi, les statistiques présentées dans la suite de cette section se rapportent aux détenteurs du contrat `PREV1`.
Au total, sur le portefeuille d'assurés, $32,2\%$ des assurés possèdent le contrat `PREV1` et $13,6\%$ le contrat `PREV2`. Dans la suite du mémoire, pour la modélisation, on ne retiendra que le fait qu'un assuré possède un contrat `PREV`, indépendemment duquel. $45,8\%$ des assurés seront donc considérés comme détenteurs d'un contrat PREVoyance.

Nom de la variable	Contenu	Type
prev1_detmrh	Détention d'un contrat `MRH`, pas forcément dans la compagnie	Qualitatif
prev1_formule	Formule souscrite par l'assuré	Qualitatif

prev1_detmrh : Indique si l'assuré possédant le contrat `PREV1` possède aussi un contrat `MRH`. Le contrat `MRH` peut être possédé dans une autre compagnie. En effet, 219 991 assurés possèdent un contrat `MRH` mais ne possèdent ni le contrat `MRH1` ni le contrat `MRH2` : ils ont donc souscrit un contrat `MRH` chez un autre assureur.

prev1_formule : Indique la formule souscrite par l'assuré pour le contrat `PREV1`. Les formules `1` ou `2` sont possibles. 62,9% des détenteur d'un contrat `PREV1` ont choisi la formule `1`, et 37,1% la formule `2`.

1.3.4 Variables contrat CORP

A l'inverse des 3 autres types de contrats étudiés dans ce mémoire, le contrat CORP est unique. C'est une assurance CORPorelle qui prend effet lorsque l'assuré est reconnu responsable d'un accident dans lequel il est blessé. En effet, la responsabilité civile (RC) répare seulement les dommages causés à autrui, et non ceux causés à soi-même. Le contrat CORP permet donc de couvrir le conducteur en cas de préjudice corporel lorsqu'il est responsable de l'accident. 74,3% des assurés du portefeuille détiennent le contrat CORP.

Nom de la variable	Contenu	Type
corp_facteuraggr	Présence de facteurs aggravants (utilisation d'un 2 roues)	Qualitatif
corp_nbveh	Nombre de véhicules possédés par l'assuré	Qualitatif

corp_facteuraggr : Cette variable indique si l'assuré possède un véhicule à 2 roues, considéré comme un facteur de risque aggravant pour une garantie dégâts corporels. 8% des assurés possèdent un 2 roues.

corp_nbveh : Indique le nombre de véhicules possédés par l'assuré. Le tableau 1.10 présente les proportions de cette variable. 2,9% des assurés ne possèdent pas de véhicule, mais ont néanmoins choisi de souscrire un contrat CORP, qui les protège en cas d'accident corporel.

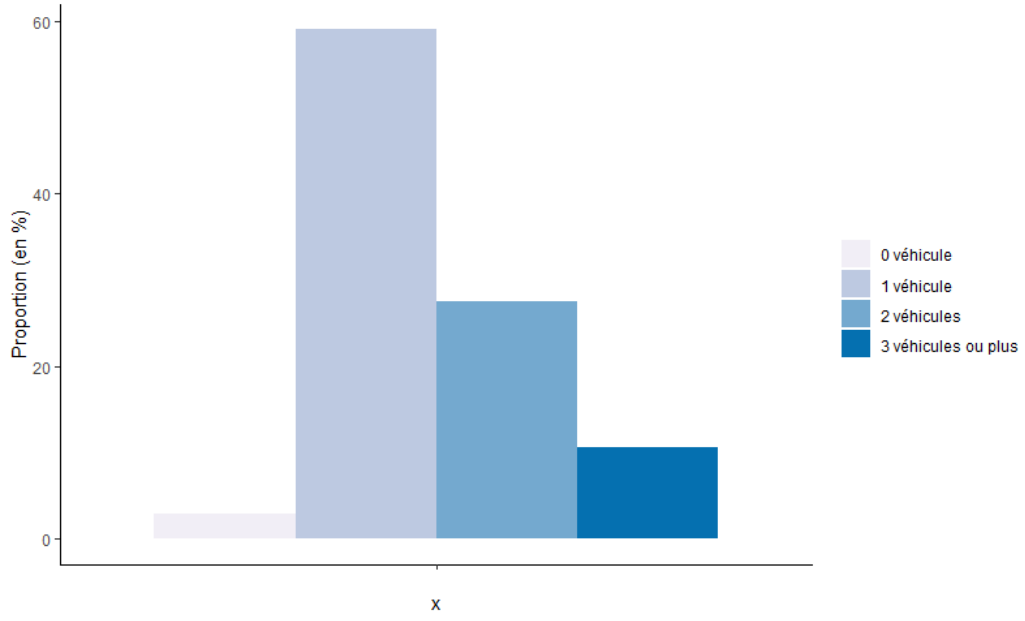


FIGURE 1.10 : Nombre de véhicules des détenteurs d'un contrat CORP

1.4 Interprétabilité et explicabilité

1.4.1 Présentation

La raison 71, liée à l'article 22 du RGPD (Règlement Général sur la Protection des Données), énonce le principe suivant :

La personne concernée devrait avoir le droit de ne pas faire l'objet d'une décision, qui peut comprendre une mesure, impliquant l'évaluation de certains aspects personnels la concernant, qui est prise sur le seul fondement d'un traitement automatisé et qui produit des effets juridiques la concernant ou qui, de façon similaire, l'affecte de manière significative, tels que le rejet automatique d'une demande de crédit en ligne ou des pratiques de recrutement en ligne sans aucune intervention humaine. [...] En tout état de cause, un traitement de ce type devrait être assorti de garanties appropriées, qui devraient comprendre une information spécifique de la personne concernée ainsi que le droit d'obtenir une intervention humaine, d'exprimer son point de vue, d'obtenir une explication quant à la décision prise à l'issue de ce type d'évaluation et de contester la décision.

Les algorithmes de machine learning prennent une part de plus en plus importante dans les processus de décision qui impactent nos vies. Les premiers algorithmes étaient régis par des règles de décision et des équations mathématiques simples, implémentées par le statisticien et pouvant être facilement expliquées. Néanmoins, depuis quelques années et l'avènement de puissants algorithmes de machine learning (tels que les réseaux de neurones), la transparence derrière les résultats produits par ces algorithmes a diminué, au profit des performances de ces algorithmes. Par ailleurs, l'aspect "boîte noire" de ceux-ci peut aussi être un frein à leur mise en pratique, car il est plus compliqué de faire accepter une décision que l'on ne peut pas expliquer.

Prenons l'exemple de l'octroi d'un crédit immobilier. Une femme célibataire, touchant un revenu de 37k€ annuel et ayant 2 enfants à charge, souhaite souscrire un crédit immobilier. Là où le conseiller bancaire aurait évalué sa solvabilité manuellement il y a quelques années, celui-ci rentre aujourd'hui les caractéristiques individuelle de sa cliente dans un outil, et un algorithme calcule un score de solvabilité de cette personne. Cet algorithme se base sur l'historique de données de la banque et compare les caractéristiques de la cliente avec d'anciens clients, pour lesquels on dispose de l'information concernant le remboursement ou non du crédit.

Supposons que le modèle fournisse une probabilité de solvabilité de $0,6$. Comment justifier cette valeur de 60 % ? Est-ce parce que la cliente est célibataire avec des enfants à charge ? Est-ce en rapport avec son salaire ? Une légère augmentation de salaire aurait-elle beaucoup changé sa probabilité de solvabilité ? Toutes ces questions pourraient être posées par la cliente, et il revient au conseiller bancaire de pouvoir interpréter et expliquer la valeur produite par l'algorithme.

Une autre difficulté que rencontrent ces algorithmes est qu'ils peuvent être biaisés à cause des données d'apprentissage. Le modèle s'adaptant aux données d'apprentissage, si celles-ci sont biaisées et ne représentent pas l'ensemble d'une population, le modèle reproduira ce biais lors des futures prédictions. En 2016, le moteur de prédiction COMPAS, servant à expliquer la propension à la récidive des condamnées, utilisait indirectement l'origine ethnique des individus, car ayant appris sur un échantillon biaisé (InstitutdesActuaires, 2019).

Un autre exemple de ce biais est celui du moteur de recrutement d'Amazon. Le modèle avait été entraîné sur les CV reçus lors des 10 dernières années ; ces CV étaient principalement masculins, car les hommes prédominent dans le secteur technologique. Le modèle en a déduit que les hommes étaient

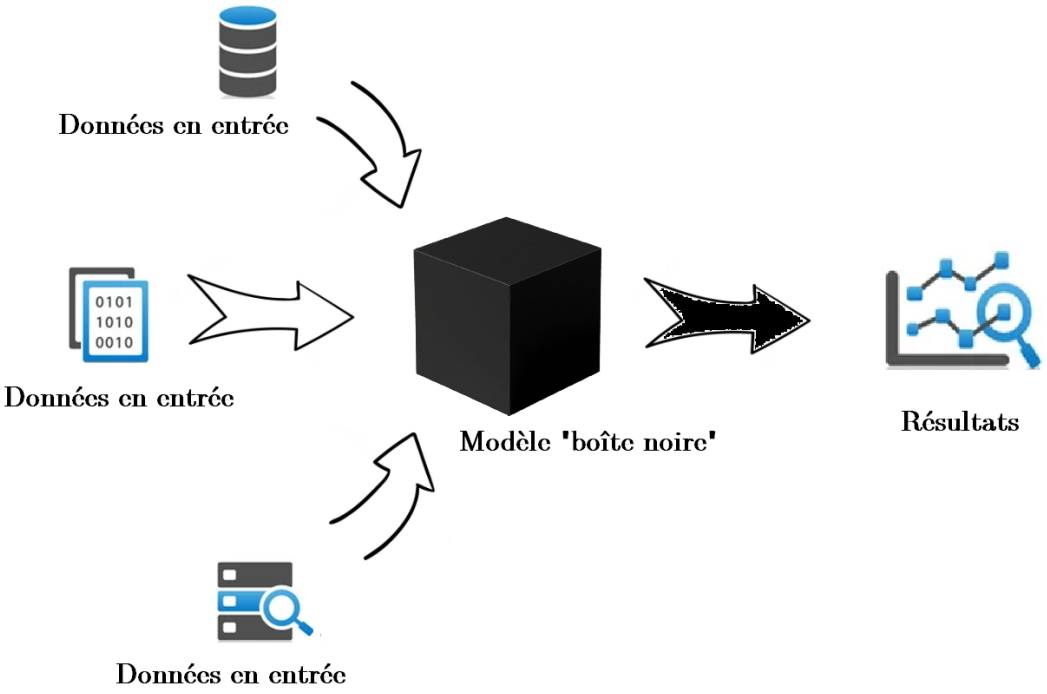


FIGURE 1.11 : Modèle “boîte noire”

de meilleurs candidats pour les postes à pourvoir, et a donc indirectement discriminé le recrutement des femmes. Sans interprétabilité, il est impossible de repérer ce genre de biais.

Ce biais lié aux données d'apprentissage peut aussi apparaître lorsque de nouvelles modalités sont ajoutées aux variables explicatives. Prenons l'exemple de la variable `auto1energie` de notre jeu de données : il n'existe pas de modalité représentant une voiture électrique, celles-ci étant comptabilisés sous la modalité `Autres`. On peut légitimement supposer que la proportion de voitures électriques va exploser dans les prochaines années, et qu'une modalité `Electrique` sera créée. Néanmoins, le modèle ayant été entraîné sans cette modalité, il ne saura pas comment interpréter cette nouvelle information, et son pouvoir prédictif pourrait baisser, sans que l'utilisateur ne comprenne d'où vient cette diminution.

Ce type d'algorithmes boîtes noires pourrait aussi mener à des problèmes d'ordre éthique. Prenons le cas d'un algorithme médical qui calcule les chances de réussite d'un acte chirurgical. Le patient se verrait refuser une opération car un algorithme a déterminé que les chances de réussites étaient trop faibles, et le chirurgien ne serait pas en mesure d'en expliquer les raisons.

Différence sémantique

Les concepts d'interprétabilité et d'explicabilité sont souvent associés, mais ne sont pas synonymes pour autant.

L'**interprétabilité** peut être définie comme la facilité de compréhension du modèle, l'évaluation globale du processus de prise de décision de l'algorithme. Elle représente le raisonnement de l'algorithme. Il faut retenir que plus un modèle est performant, moins il sera interprétable, car il sera basé sur des équations mathématiques complexes.

L'**explicabilité** fournit quant à elle des informations sur les variables déterminantes pour les résultats de l'algorithme. Elle permet à un utilisateur "métier" (c'est-à-dire ne possédant pas les capacité techniques nécessaires à la création de l'algorithme statistique) de comprendre pourquoi l'algorithme produit tel résultat.

En d'autres termes, l'interprétabilité répond à la question "comment" un algorithme de Machine Learning prend une décision, alors que l'explicabilité répond à la question "pourquoi" l'algorithme a pris cette décision.

Interprétabilité agnostique ou spécifique

L'un des axes de classification des méthode d'interprétabilité est sa dimension agnostique ou spécifique.

Une méthode **agnostique** est indépendante du modèle utilisé. Elle considère chaque modèle comme une boîte noire, et peut donc être théoriquement appliquée sur tous les modèles. L'avantage des ces méthodes est qu'elles sont flexibles, applicables sans connaissance particulière du modèle. Elles se basent uniquement sur les observations en entrée x et la valeur de sortie y. Les PDP (*Partial Dependency Plot*), qui estiment les lois marginales des variables du modèles sous l'hypothèse d'indépendance entre les variables, sont l'une des premières méthodes agnostiques. De nouvelles méthodes ont été introduites récemment, telles que LIME (*Local Interpretable Model-Agnostic Explanations*) ou Kernel SHAP. Elles prennent en compte certaines faiblesses des premières méthodes agnostiques et les adaptent pour des modèles plus complexes.

Une méthode **spécifique** dépend du modèle utilisé. Elle est donc moins flexible qu'une méthode agnostique, mais produit de bien meilleurs résultats, car elle a été spécifiquement créée pour un modèle particulier. Ces méthodes ne se basent pas seulement sur les valeurs d'entrée x et la valeur de sortie y, mais aussi sur les propriétés et méthodes de construction du modèle. Ainsi, pour un modèle donné, lorsqu'une méthode spécifique existe, elle aura toujours une meilleure performance qu'une méthode agnostique, puisqu'elle dispose d'informations supplémentaires. Deux méthodes spécifiques populaires sont DeepLIFT, utilisée pour les réseaux de neurones, et Tree SHAP, utilisée pour les arbres de décision.

Interprétabilité globale ou locale

L'interprétabilité d'un modèle peut prendre deux granularités différentes : globale ou locale.

Une méthode **globale** va expliquer les tendances globales du modèle sur l'ensemble des prédictions. Elle concerne le fonctionnement global de l'algorithme, toutes valeurs d'entrée confondues.

Une méthode **locale** se focalisera sur une prédiction en particulier. Elle permet d'être plus précis pour une observation donnée, et d'expliquer la prédiction en fonction des variables en entrée.

Le choix de la granularité de la méthode dépend fortement de l'interprétation que l'on souhaite faire. Si l'on reprend l'exemple du crédit immobilier, le conseiller bancaire sera intéressé par une méthode locale, afin d'expliquer à sa cliente les raisons qui ont conduit l'algorithme à prédire une solvabilité de 60%. En revanche, dans le cas d'un algorithme prédisant une stratégie d'investissement, on préférera une méthode globale, qui expliquera la sensibilité du modèle aux variables en entrée de l'algorithme.

1.4.2 Application à l'assurance

Historiquement, les modèles linéaires généralisés (*GLM* en anglais) ont été un choix très populaire parmi les assureurs, de part leur pouvoir prédictif et leur forte interprétabilité. En effet, dans un GLM, les coefficients de régression sont facilement interprétables et on peut rapidement en déduire l'importance des différentes variables. L'assureur étant soumis à de nombreuses contraintes réglementaires, il lui faut pouvoir justifier le prix proposé pour ses contrats, d'où un fort besoin d'interprétabilité de ses modèles de tarification. Néanmoins, avec l'arrivée de nouveaux modèles (*forêts aléatoires*, *Gradient Boosting Machine (GBM)*, *réseaux de neurones*), les assureurs ont à disposition de nouveaux algorithmes plus performants qu'un simple modèle linéaire. En revanche, contrairement au modèle linéaire, on ne peut pas facilement expliquer le lien entre les données en entrée et la valeur de sortie du modèle.

Prenons l'exemple du *Gradient Boosting Machine (GBM)*, en l'expliquant brièvement. Ce modèle est construit en suivant un processus itératif : l'algorithme commence par créer un modèle simple (un arbre de décision par exemple, que nous appelons *Arbre 1*). On évalue ce modèle et on obtient une prédiction *Prédiction 1*, puis l'algorithme calibre un nouvel arbre *Arbre 2* en donnant un poids plus fort aux observations pour lesquelles *Prédiction 1* était erronée, et un poids plus faible aux bonnes prédictions. Le modèle devient donc *Arbre 1 + Arbre 2*, et on évalue une nouvelle fois le modèle, pour obtenir *Prédiction 2*. On réitère l'opération jusqu'à ce qu'un critère d'arrêt ou un nombre fixé d'itérations soit atteint. La prédiction finale est donc la somme pondérée des prédictions effectuées par les modèles précédents.

Ainsi, pour revenir à l'interprétabilité du modèle, on n'a aucune information sur les déterminants de la valeur de sortie de l'algorithme. Le GBM a potentiellement été construit à partir d'une centaine d'arbres de décision, et il est impossible de dire quelle variable a été importante, et quelle variable a eu peu d'impact. Le GBM ne donne pas de coefficients aux variables, contrairement au GLM.

Les assureurs sont donc à la recherche de méthodes permettant l'explicabilité de ces modèles "boîtes noires", afin de pouvoir profiter de leurs performances tout en respectant les différentes réglementations et les besoins commerciaux d'interprétation.

Dans le cas de notre étude, c'est l'interprétabilité globale qui nous intéresse. En effet, on ne cherche pas à expliquer la valeur de marge future d'un assuré en particulier, mais plutôt à trouver dans quelle mesure les variables contribuent à la valeur de la marge future estimée.

Méthode globale agnostique

A l'inverse des méthodes locales, les méthode globales décrivent le comportement moyen d'un modèle de machine learning. Elles sont souvent exprimées comme l'espérance des valeurs de sortie, se basant sur la distribution des données. Deux méthodes globale agnostiques populaires sont les *Partial Dependence Plot (PDP)*, qui sont utilisées lorsque les variables sont non-corrélées, et les *Accumulated Local Effects (ALE)*, utilisés lorsque les variables sont corrélées. La méthode des PDP est explicitée dans la section suivante.

Partial Dependence Plot

Les *Partial Dependence Plot* (aussi appelés *PDP* ou *PD plot*) permettent de voir l'effet d'une ou deux variables sur la valeur en sortie de l'algorithme. La théorie mathématique derrière cette méthode n'est

pas limitée à une ou deux variables, mais le besoin de lisibilité des résultats implique de se restreindre à une (résultats en 2D) ou deux (résultats en 3D) variables à la fois.

Les *PDP* fonctionnent selon le principe suivant : on sélectionne la variable X_1 dont on veut obtenir l'effet sur le résultat final du modèle. On définit une grille de valeurs pour cette variable, puis pour chaque valeur x de la grille, on fixe $X_1 = x$ pour chaque observation de la base de données, sans toucher aux autres variables. Ainsi, on obtient une nouvelle base de données modifiée, où toutes les observations sont identiques à celles de la base initiale, à l'exception de la valeur de la variable X_1, qui vaut x. On prédit ensuite selon cette nouvelle base de données et on prend la moyenne des prédictions. On réitère pour toutes les valeurs de la grille et on trace la courbe représentant l'effet de X_1 sur la valeur en sortie de l'algorithme. L'algorithme fonctionne de la même manière si l'on regarde l'effet de 2 variables en même temps, il suffit de fixer $X_1 = x$ puis de faire varier X_2 à travers la grille de valeur de X_2, puis de réitérer pour toutes les valeurs de la grille de X_1. On obtient donc une carte de chaleur au lieu d'une courbe.

Le principal inconvénient de la méthode *PDP* est qu'elle nécessite l'indépendance de la variable sur laquelle on calcule la dépendance partielle (i.e. la variable dont on veut expliquer l'effet sur le résultat du modèle) par rapport aux autres variables du modèle. En effet, la méthode va tester des combinaisons qui peuvent être irréalistes s'il y a une corrélation entre 2 variables.
Par exemple, imaginons que l'on cherche à modéliser le prix d'un appartement en fonction de sa taille et du nombre de chambres, sur la base fictive décrite au tableau 1.13.

taille (en m^2)	nombre de chambres	prix (en €)
49	2	416 304
80	3	679 680
72	3	611 712
15	0	127 440
134	5	113 8464
95	4	807 120
26	1	220 896
34	1	288 864

TABLE 1.13 : Base fictive pour la prédiction du prix d'un appartement

On souhaite connaître l'impact de la variable `taille` sur le prix de l'appartement. La méthode *PDP* va donc créer une grille de valeurs pour la variable `taille`, puis prédire le prix des appartements en fixant la variable `taille`, et en laissant la variable *nombre de chambres* à ses valeurs initiales pour chaque observation. Ainsi, pour la valeur `taille` $= 35$, on obtient la base suivante :

taille (en m^2)	nombre de chambres	prix (en €)
35	2	416 304
35	3	679 680
35	3	611 712
35	0	127 440
35	5	113 8464
35	4	807 120
35	1	220 896
35	1	288 864

TABLE 1.14 : Base fictive modifiée

La méthode *PDP* va ensuite appliquer le modèle à cette base et comparer les prédictions avec les valeurs de `prix`. On obtient donc des combinaisons peu probables, telles qu'un appartement de $35m^2$ avec 5 chambres. Lorsque cette situation arrive et que la variable dont on cherche l'effet est corrélée avec les autres, on préférera utiliser la méthode *Accumulated Local Effects*, qui ne sera pas explicitée en détail dans ce mémoire. Le lecteur peut se reporter au livre de Christoph Molnar (MOLNAR, 2022) s'il souhaite en savoir plus sur cette méthode.

Interprétabilité des MOB

Dans le cadre de cette étude, l'interprétabilité sera relativement aisée. Les différents modèles que nous verrons sont en majorité directement interprétables. Pour les GLM, il suffit d'observer la valeur des coefficients de régression pour comprendre le résultat final. Pour les arbres de décision CART, la valeur finale d'une feuille est la moyenne des observations appartenant à cette feuille, et on peut facilement expliquer pourquoi une observation est dans une feuille en particulier, en suivant les différentes coupes effectuées par l'algorithme CART. Enfin, les MOB, qui seront présentés plus en détail dans le chapitre 3, sont eux aussi assez facilement interprétables. Pour faire simple, ceux que nous utiliserons seront un mélange d'arbre de décision CART et de GLM. Ainsi, il est facile de trouver une interprétation locale pour un individu : il suffit de suivre les différentes coupes de l'arbre, puis d'expliquer les coefficients du GLM associé.

Chapitre 2

Reconstitution de la marge

On souhaite modéliser la marge future des assurés en fonction de la détention ou non d'un type de contrat. Dans un premier temps, nous présenterons la théorie sous-jacente différents modèles mathématiques, puis nous les appliquerons à notre base de données afin de reconstituer la valeur de la marge future estimée par le modèle de la compagnie.

2.1 Modèle linéaire généralisé

Afin d'expliquer les déterminants de la marge, nous appliquerons les modèles MOB (*MOdel Based recursive partioning*) à nos données. Ceux-ci seront présentés plus en détails dans une future section. Nous allons dans un premier temps expliquer le fonctionnement des modèles linéaires généralisés (Generalized Linear Model en anglais, ou GLM), que nous utiliserons dans les MOB. Les GLM sont des modèles assez simples permettant d'étudier le lien (pas forcément linéaire, à l'inverse du modèle linéaire) entre les variables explicatives de la base de donnée et la variable que l'on souhaite prédire. Ces modèles font partie des premiers à avoir été utilisés par les assureurs, car ils sont faciles à implémenter et à interpréter. Nous avons donc choisi de les appliquer à nos données afin d'avoir un modèle simple de comparaison avec les autres modèles utilisés dans cette étude.

2.1.1 Modèle de régression linéaire multivarié

Le modèle de régression linéaire (multivarié ou non) est basé sur l'hypothèse qu'il existe un lien linéaire entre la variable à prédire et les covariables. On dispose de n observations indépendantes $(y_i, x_i)_{1\leq i\leq n}$, $y_i \in \mathbb{R}, x_i \in \mathbb{R}^p$. Ainsi, chaque individu i est représenté par p variables explicatives. Sous forme matricielle, on définit la *matrice design* Xde taille $n \times (p+1)$ par

$$X = \begin{pmatrix} 1 & x_1^1 & \cdots & x_1^p \\ \vdots & \vdots & & \vdots \\ 1 & x_n^1 & \cdots & x_n^p \end{pmatrix}. \tag{2.1}$$

On dit que $Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ suit un modèle de régression linéaire multivarié si on peut écrire

$$Y = X\beta + \epsilon \tag{2.2}$$

où $\beta \in \mathbb{R}^{p+1}$ est le vecteur des paramètres du modèle et $\epsilon \in \mathbb{R}^n$ est le terme d'erreur.
On observe les 4 postulats suivants :

- [P1] : $\mathrm{E}[\epsilon_i] = 0 \, \forall i$,
- [P2] : $\mathrm{Cov}(\epsilon_i, \epsilon_{i'}] = 0 \, \forall i \neq i'$,
- [P3] : $\mathrm{Var}[\epsilon_i] = \sigma^2 > 0 \, \forall i$,
- [P4] : le vecteur ϵ est un vecteur gaussien.

Cela implique que $\epsilon \sim \mathcal{N}_n(0, \sigma^2 I_n)$.

Estimation du paramètre β

$\beta \in \mathbb{R}^{p+1}$ est le paramètre représentant le *poids* de chaque variable explicative. Afin de trouver un estimateur de β, que nous noterons $\widehat{\beta}$, on cherche à minimiser la somme des carrés des résidus : c'est le principe des moindres carrés. Mathématiquement, cela correspond à

$$\begin{aligned} \widehat{\beta} &\in \underset{\beta \in \mathbb{R}^{p+1}}{\arg\min} ||Y - X\beta||^2 \\ &\in \underset{\beta \in \mathbb{R}^{p+1}}{\arg\min} \sum_{i=1}^{n} (y_i - (\beta_0 + \sum_{j=1}^{p} \beta_j x_i^j))^2. \end{aligned} \tag{2.3}$$

On appellera l'estimateur ainsi trouvé Estimateur des Moindres Carrés Ordinaires (EMCO). En supposant que X est une matrice de rang plein ($rg(X) = p + 1$), alors l'EMCO s'écrit

$$\widehat{\beta} = \widehat{\beta}(Y) = (X^T X)^{-1} X^T Y. \tag{2.4}$$

Sélection des variables

On ne désire pas toujours garder toutes les variables explicatives dans notre modèle. En effet, un grand nombre de variables explicatives sélectionnées améliorera l'ajustement du modèle aux données, mais entraînera un risque d'*over-fitting* (ou *surapprentissage* en français), ce qui rendra le modèle peu efficient sur de nouvelles données. On cherche donc à sélectionner le meilleur modèle, en comparant les performances de différents modèles entre eux, à l'aide d'un ou plusieurs critères. Nous détaillons ci-après le critère d'information d'Akaike (AIC), le critère d'information bayésien (BIC) et le coefficient de détermination linéaire (R^2), qui peuvent être utilisés comme métriques pour comparer les modèles.

Critère AIC

Le Critère d'Information d'Akaike (AIC) repose sur la log-vraisemblance et le nombre de paramètres du modèle. Il pénalise les modèles avec un grand nombre de variables. Il est défini par

$$AIC = 2 \times p - 2 \times log(\tilde{L})$$

avec

- $\tilde{L}$ la vraisemblance du modèle,
- p le nombre de paramètres du modèle.

On cherche à obtenir le modèle avec l'AIC le plus faible.

Critère BIC

Le Critère d'Information Bayésien (BIC) repose lui aussi sur la log-vraisemblance et le nombre de variables du modèle. Il est défini par

$$BIC = -2 \times log(\tilde{L}) + p \times ln(n)$$

avec

- $\tilde{L}$ la vraisemblance du modèle,
- p le nombre de paramètres du modèle,
- n le nombre d'observations.

Comme pour l'AIC, on cherche à minimiser le BIC. On remarque que le BIC pénalise plus fortement le nombre de paramètres du modèle que l'AIC.

Critère R^2

Le coefficient de détermination linéaire, noté R^2, permet de mesurer la qualité de la prédiction du modèle. Il est compris entre 0 et 1, et on souhaite le maximiser. Il est défini par

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$$

avec

- n est le nombre d'observations,
- $\widehat{Y}_i$ la valeur prédite pour l'individu i,
- $\bar{Y}$ la moyenne des Y_i.

Sélection de modèle

Il existe un très grand nombre de modèles différents. Si l'on devait les calculer tous, puis les comparer 2 à 2, cela engendrerait une complexité informatique trop importante. Il existe 3 méthodes de sélection de variables, qui permettent de comparer les meilleurs modèles entre eux, sans avoir à tous les calculer.

Méthode Forward : Cette méthode consiste part du modèle réduit à l'intercept (qui correspond à la valeur moyenne de la variable réponse lorsque tous les coefficients de régression sont fixés à 0). On le compare ensuite à tous les modèles contenant 1 variable, et on garde celui qui minimise le plus notre critère (AIC par exemple). On réitère l'opération avec une variable de plus dans le modèle précédemment sélectionné, jusqu'à ce que le critère ne soit plus amélioré, ou qu'on obtienne le modèle complet (contenant toutes les variables).

Méthode Backward : Cette méthode est l'opposée de la méthode Forward. On part du modèle complet, et on compare tous les modèles auxquels on a retiré une variable. On choisit celui qui minimise le plus notre critère. On réitère l'opération jusqu'à ce que le critère ne soit plus amélioré, ou que l'on arrive au modèle réduit à l'intercept.

Méthode Stepwise : Cette méthode est un mélange des deux méthodes précédentes : on part soit du modèle réduit à l'intercept, s'il s'agit d'une sélection *stepwise forward*, soit du modèle complet (contenant toutes les variables explicatives), s'il s'agit d'une sélection *stepwise backward*. On ajoute ou supprime la variable explicative qui minimise le critère à chaque étape. Comme précédemment, on s'arrête lorsque le critère n'est plus amélioré.

Mesure d'erreur

Afin de quantifier l'erreur du modèle lors de la phase de prédiction, on utilise différents indicateurs, qui calculent l'écart entre la valeur attendue et la valeur prédite. C'est en comparant ces mesures d'erreur (aussi appelées métriques d'évaluation) que l'on peut quantifier la performance prédictive d'un modèle. Nous allons en expliciter deux, que nous utiliserons par la suite : la MSE et la MAE.

MSE

La MSE (*Mean Squared Error*) d'un estimateur mesure la moyenne des erreurs quadratiques du modèle. Autrement dit, c'est la moyenne des carrés des différences entre la valeur attendue et la valeur prédite par le modèle. Elle est définie par

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y_i} - y_i)^2, \tag{2.5}$$

où $\hat{y_i}$ est la valeur prédite par le modèle et y_i est la valeur attendue pour la i^{eme} observation, .

On peut aussi utiliser la RMSE (*Root Mean Squared Error*), qui est la racine carrée de la MSE. Ces 2 métriques sont équivalentes, car la fonction racine carrée est croissante sur $\mathbb{R}^+$. On utilisera la MSE par la suite.

MAE

Une autre métrique d'évaluation est la MAE (*Mean Absolute Error*) : elle est définie comme la moyenne des écarts en valeur absolue entre la valeur attendue et la valeur prédite par le modèle. Elle s'écrit

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_i - y_i|, \tag{2.6}$$

avec $\hat{y}_i$ et y_i comme pour la MSE.

2.1.2 Modèle linéaire généralisé

Le modèle linéaire généralisé est une extension du modèle linéaire présenté précédemment : il permet de se passer des postulats [P1], [P2] et [P3] et d'obtenir une meilleure prédiction de la variable réponse Y, en élargissant la famille de lois que Y peut suivre. On ne suppose plus qu'il existe une relation linéaire entre Y et les covariables (X_i), ni que Y appartient à $\mathbb{R}$: la variable à prédire peut être discrète ou strictement positive par exemple. On introduit donc un nouveau modèle :

$$g(\mathrm{E}[Y|x]) = x^T\beta. \tag{2.7}$$

La méthode s'effectue en 3 étapes :

1. Choisir une loi de probabilité pour $Y|x$ parmi la famille exponentielle naturelle $\mathcal{F}^{NAT}$;
2. Choisir une "bonne" fonction de lien $g(.)$, en général le lien canonique ;
3. À partir de $(Y_i, x_i)_{1\leq i\leq n}$, estimer β par $\widehat{\beta}$ tel que $\widehat{\mathrm{E}(Y_i|x_i)} = g^{-1}(x_i^T\widehat{\beta})$.

Famille exponentielle naturelle

Soit f_Y la densité de Y. $f_Y \in \mathcal{F}^{NAT}$ si elle s'écrit

$$f_Y(y) = \exp\left(\frac{y\theta - b(\theta)}{\gamma(\phi)} + c(y,\phi)\right), \tag{2.8}$$

où c est une fonction dérivable, b est trois fois dérivable et b' est inversible.
θ est appelé *paramètre naturel* de la loi, ϕ est appelé *paramètre de nuisance* ou de *dispersion.*

Si la densité f_Y appartient à la famille exponentielle naturelle, alors

$$\mathrm{E}[Y] = \mu = b'(\theta) \tag{2.9}$$
$$\mathrm{Var}[Y] = \gamma(\phi)b"(\theta). \tag{2.10}$$

On appelle fonction de *lien canonique* la fonction $g(.)$ associée à la loi f_Y de $Y \in \mathcal{F}^{NAT}$ telle que

$$g(.) = (b')^{-1}(.). \tag{2.11}$$

Voici un tableau présentant les différentes fonctions de lien canonique :

Choix de la loi de $Y\|x$	Bernoulli/Binomiale	Poisson	Gamma	Gaussienne
Fonction de lien canonique	$g(\mu) = \log(\frac{\mu}{1-\mu})$	$g(\mu) = \log(\mu)$	$g(\mu) = -\frac{1}{\mu}$	$g(\mu) = \mu$
Nom du lien	logit	log	réciproque	identité

Le choix de la loi de probabilité de $Y|x$ est déterminé par l'ensemble de définition de Y. En effet, cela n'a pas de sens de poser une loi normale sur une variable qui est toujours positive. Voici quelques exemples de domaines de définition de Y motivant le choix dde la loi de probabilité :

Domaine de définition de Y	Loi de probabilité retenue
$\mathbb{R}$	$\mathcal{N}(\mu, \sigma^2)$
$\mathbb{R}_+$ ou $\mathbb{R}_-$	Gamma
$\mathbb{N}$	Poisson
E avec #E $< \infty$	Bernoulli/Binomiale

Estimation du paramètre β

On estime le paramètre β par la méthode du maximum de vraisemblance. Comme Y suit une loi appartenant à la famille exponentielle naturelle, le log de sa vraisemblance est donnée par

$$\mathcal{L}(Y, \theta) := \sum_{i=1}^{n} \frac{Y_i\theta_i - b(\theta_i)}{\phi} + c(Y_i, \phi) \tag{2.12}$$

$$= \sum_{i=1}^{n} \mathcal{L}_i(\beta), \tag{2.13}$$

où $\mathcal{L}_i(\beta)$ est la contribution de l'observation i au log de la vraisemblance.
Ainsi, on obtient $\widehat{\beta}$, appelé estimateur du maximum de vraisemblance (EMV), en résolvant

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^{p+1}}{\arg\max}\, \mathcal{L}(Y, \theta). \tag{2.14}$$

Résidus

Afin de valider la qualité de la régression, on peut analyser les résidus du modèle. Il en existe plusieurs types : les plus fréquemment utilisés sont les résidus de déviance et les résidus de Pearson.

– Les résidus de déviance sont définis par

$$r_i = \sqrt{d_i} * sign(Y_i - \hat{Y}_i) \tag{2.15}$$

où d_i est la contribution de l'individu i à la déviance totale. L'expression de d_i dépend de la loi utilisée par le modèle. Par ailleurs, on a $D = \sum_{i=1}^{n} d_i$, autrement dit la déviance totale D est la somme du carré des résidus de déviance.

– Les résidus de Pearson sont définis par

$$r_i^p = \frac{Y_i - \hat{Y}_i}{\sqrt{Var(\hat{Y}_i)}}. \tag{2.16}$$

Afin de valider le modèle, on peut effectuer un test statistique utilisant les résidus de Pearson. On prend pour statistique de test la somme des résidus de Pearson élevés au carré :

$$\chi^2 = \sum_{i=1}^{n} (r_i^p)^2. \tag{2.17}$$

Cette statistique suit asymptotiquement une loi du khi-deux à $n-(p+1)$ de liberté. On acceptera le modèle au niveau de risque α si $\chi^2 \leq q_{1-\alpha}$, avec $q_{1-\alpha}$ le quantile tel que $\mathbb{P}(\chi^2(n-(p^+1)) > q_{1-\alpha}) = \alpha$.

2.1.3 Estimateur LASSO

Cette partie se base sur le cours d'*Apprentissage statistique* d'Angelina Roche, dispensé en M1 à l'Université Paris-Dauphine, lui-même basé sur le livre *An Introduction to Statistical Learning* de (James et al., 2014).

On se place dans le modèle linéaire gaussien multivarié

$$Y = X\beta^* + \epsilon \tag{2.18}$$

avec

$$Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix} \ ; \ X = (x_i^j)_{\substack{1 \leq i \leq n \\ 1 \leq j \leq d}} \ ; \ \beta^* \in \mathbb{R}^d \ ; \ \epsilon \sim \mathcal{N}_n(0, \sigma^2 I_n). \tag{2.19}$$

L'estimateur LASSO (Least Absolute Shrinkage and Selection Operator) a pour objectif de sélectionner des variables tout en estimant β^*. La sélection de variable se fait en fixant $\beta_j = 0$ pour certains $j \in [1, ..., p]$. On dira que la $j^{ème}$ variable est sélectionnée si $\beta_j \neq 0$ et qu'elle est non sélectionnée sinon.
En effet, la prédiction $\widehat{Y}_i(\widehat{\beta})$ de Y_i à partir de l'observation de x_i et de $\widehat{\beta}$ est :

$$\begin{aligned} \widehat{Y}_i(\widehat{\beta}) &= x_i^T \widehat{\beta} \\ &= x_i^1 \widehat{\beta}_1 + ... + x_i^d \widehat{\beta_d} \end{aligned} \tag{2.20}$$

Donc s'il existe j tel que $\widehat{\beta}_j = 0$, x_i^j ne sera pas utilisé pour prédire Y_i.

On appellera **estimateur LASSO** toute solution au problème L suivant

$$\begin{aligned} &\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} (Y_i - x_i^T \beta)^2 \\ &\text{sous la contrainte } ||\beta||_1 \leq M \end{aligned} \tag{L}$$

où M est un paramètre de l'estimateur.

L'estimateur LASSO, noté $\widehat{\beta}^{(L)}$, est solution du problème L' suivant

$$\min_{\beta \in \mathbb{R}^d} \underbrace{\sum_{i=1}^{n} (Y_i - x_i^T \beta)^2}_{\text{terme d'attache aux données}} + \underbrace{\lambda ||\beta||_1}_{\text{pénalisation dite } l^1} . \tag{L'}$$

Dans l'hypothèse où $\frac{1}{n}X^TX = Id$, l'estimateur LASSO est unique et s'écrit

$$\widehat{\beta}_\lambda^{(L)} = \frac{1}{n}X^TY(1 - \frac{\lambda}{2|X^TY|})_+ \tag{2.21}$$

où $(x)_+ = \max(0, x)$.

2.2 Modélisation GLM

Dans un premier temps, on met de côté les caractéristiques individuelles des assurés et on s'intéresse au fait de posséder un type de contrat ou non. Un tableau de donnée contenant les marges totales ainsi que les indicatrices représentant la détention des contrats `AUTO1`, `AUTO2`, `CORP`, `PREV`, `EPARGNE`, `DECES` a été construit. La table 2.1 est un extrait de ce tableau de données.

id	marge_totale	ind_auto1	ind_auto2	ind_mrh	ind_corp	ind_prev	ind_epargne	ind_deces
100007	1085,44	1	0	1	1	1	0	0
100090	2098,75	1	1	1	1	0	0	0
100145	67,65	0	0	1	0	1	0	0

TABLE 2.1 : Tableau de données de détention des contrats

On a ensuite décomposé cette base en deux échantillons : une base `train` (base d'apprentissage) pour entraîner le modèle et une base `test` pour comparer les modèles. Le modèle va apprendre sur la base d'apprentissage et calculer des coefficients en fonctions des observations de cette base. On calculera ensuite les métriques de performance sur la base de test.

Notre modèle GLM s'écrit comme suit :

$$\text{Marge totale} = \beta_0 + \mathbb{1}_{\text{auto1}}\beta_1 + \mathbb{1}_{\text{auto2}}\beta_2 + \mathbb{1}_{\text{mrh}}\beta_3 + \mathbb{1}_{\text{corp}}\beta_4 + \mathbb{1}_{\text{prev}}\beta_5 + \mathbb{1}_{\text{epargne}}\beta_6 + \mathbb{1}_{\text{deces}}\beta_7 \tag{2.22}$$

Pour la modélisation GLM, on doit choisir la loi et la fonction de lien canonique à utiliser. La marge future, qui est notre valeur à prédire, pouvant prendre des valeurs positives et négatives, on choisit donc d'utiliser la loi gaussienne avec la fonction de lien identité. Elle appartient bien à la famille exponentielle, car elle s'écrit

$$f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}exp(-\frac{(x-\mu)^2}{2\sigma^2}) \tag{2.23}$$

que l'on peut réécrire sous la forme

$$f(x;\mu,\sigma) = \exp(-\frac{1}{2}\ln(2\pi\sigma^2))exp(-\frac{(y^2-2y\mu+\mu^2}{2\sigma^2}) \tag{2.24}$$

$$= exp(\frac{y\mu - \frac{\mu^2}{2}}{\sigma^2} - \frac{1}{2}(\frac{y^2+\ln(2\pi\sigma^2)}{\sigma^2})) \tag{2.25}$$

$$= \exp\left(\frac{y\theta - b(\theta)}{\gamma(\phi)} + c(y,\phi)\right) \tag{2.26}$$

avec $\theta = \mu$, $\phi = \sigma^2$, $\gamma(\phi) = \phi$, $b(\theta) = \frac{\theta^2}{2}$ et $c(y,\phi) = -\frac{1}{2}(\frac{y^2}{\sigma^2} + ln(2\pi\sigma^2))$.

Validation croisée

Afin de vérifier que notre modèle ne fait pas de surapprentissage, on effectue une validation croisée (*CV*, ou *cross-validation* en anglais) pendant la phase d'apprentissage. Cette méthode consiste à séparer notre base d'apprentissage en k blocs de même taille, puis de les sélectionner un à un à tour de rôle. Le bloc sélectionné sera utilisé comme base de validation, et les $k-1$ autres blocs constituerons la base d'apprentissage. On répète la méthode k fois au total, ce qui donne k valeurs des métriques choisies, calculées à chaque fois sur le bloc utilisé comme échantillon de test. Si les métriques obtenues diffèrent trop d'un bloc à l'autre, alors le modèle ne prédit pas bien les données.

On effectue une validation croisée sur 5 blocs et on obtient les métriques suivantes, présentées dans le tableau 2.2 :

	MSE	MAE
Bloc 1	1 861 165	884
Bloc 2	1 820 160	882
Bloc 3	1 833 680	884
Bloc 4	1 861 207	886
Bloc 5	1 863 683	882

TABLE 2.2 : Métriques de la validation croisée

Les valeurs de MSE et MAE étant très similaires d'un bloc à l'autre, on en conclut que le modèle ne fait pas de surapprentissage. On va maintenant l'entraîner sur la base d'apprentissage entière, puis calculer les métriques sur la base de test.

Le tableau 2.3 représente les coefficients obtenus pour le modèle entraîné sur l'ensemble de la base d'apprentissage :

	Estimate
(Intercept)	$-417,280$
ind_auto1	$433,284$
ind_auto2	$783,067$
ind_mrh	$696,293$
ind_corp	$423,641$
ind_prev	$819,475$
ind_epargne	$245,743$
ind_deces	$-191,856$

TABLE 2.3 : Coefficients du GLM

Résidus de Pearson

Afin de s'assurer de la qualité du modèle, on calcule les résidus de Pearson de notre modèle. On rappelle qu'ils sont définis par

$$r_i^p = \frac{Y_i - \hat{Y}_i}{\sqrt{Var(\hat{Y}_i)}} \tag{2.27}$$

La Figure 2.1 donne une représentation graphique de ces résidus et la Figure 2.2 leur distribution.

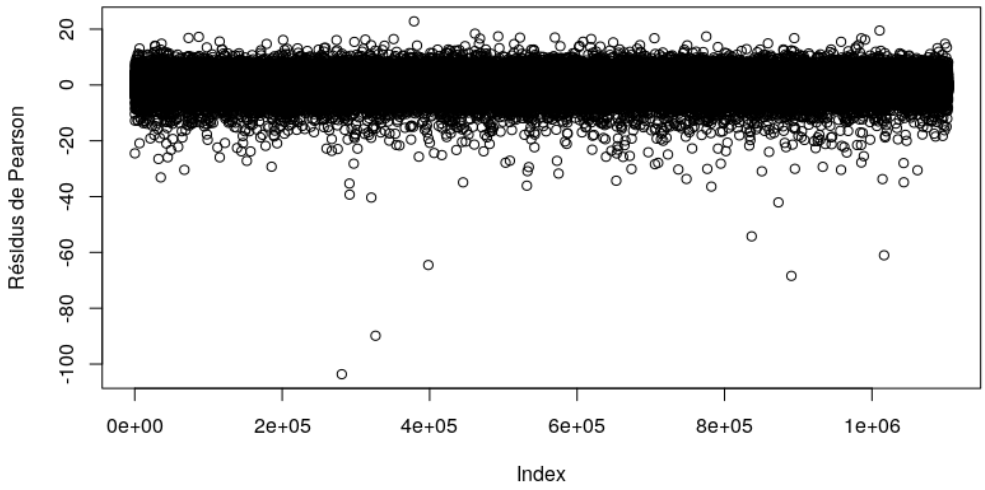


FIGURE 2.1 : Répartition des résidus de Pearson

La Figure 2.1 semble montrer que les résidus sont répartis entre -15 et 15, ce qui n'est pas un très bon résultat : ils doivent être compris principalement entre -2 et 2. Cela vient du fait que notre base de test est très grande, et que les points se superposent de nombreuses fois les uns sur les autres. De fait, 88% des résidus appartiennent à l'intervalle [-2 ;2] (ce que l'on voit aussi en regardant la Figure 2.2), 5% sont inférieurs à -2 et 7% sont supérieurs à 2.

On voit sur la Figure 2.2 que les résidus sont bien centrés en 0, mais que la densité est plus forte à gauche de l'origine qu'à droite : cela signifie que le modèle surestime la marge plus qu'il ne la sous-estime.

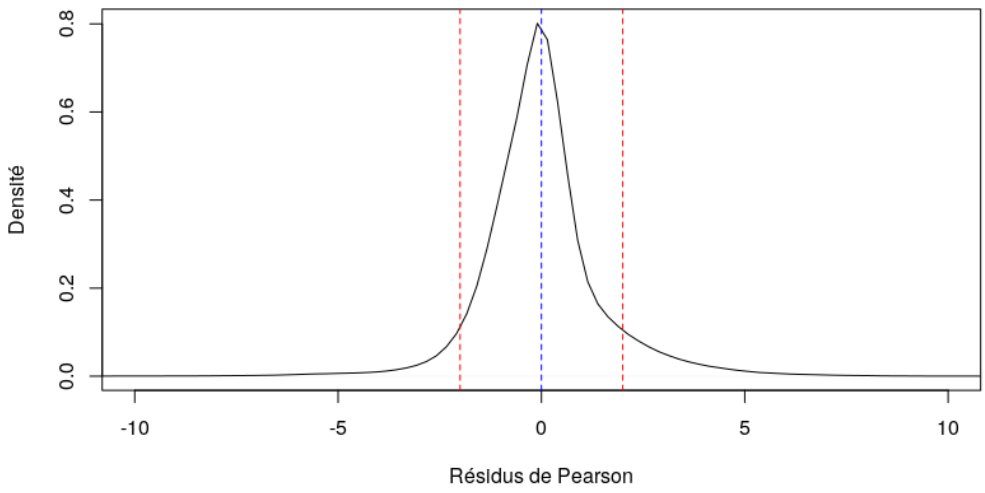


FIGURE 2.2 : Densité des résidus de Pearson

On compare maintenant le modèle GLM ainsi obtenu au modèle naïf prédisant la moyenne des observations pour toutes les valeurs de marge, *i.e.* le modèle affectant la valeur $1353,277$ à tous les individus. Le tableau 2.4 résume les valeurs des métriques de comparaison du modèle naïf et du modèle GLM.

	MSE	MAE
Modèle naïf	2 624 516	1 159
GLM	1 871 561	884

TABLE 2.4 : Métriques du modèle naïf et du GLM

Le GLM permet une diminution de plus de 40% de la MSE et de plus de 31% de la MAE. Par la suite, on comparera les autres modèles étudiés à ce GLM.

Les variables explicatives de notre modèle sont des indicatrices, valant `1` si l'assuré détient le contrat, et `0` sinon. Les coefficients du GLM sont donc très facilement interprétables : on part de l'intercept, et on ajoute la valeur du coefficient si l'assuré détient le contrat.

Par exemple, pour un assuré détenant un contrat `AUTO1`, un contrat `MRH` et un contrat `CORP`, la valeur prédite de sa marge sera :

$$\begin{aligned}\text{Marge totale} &= \beta_0 + \beta_1 + \beta_3 + \beta_4 \\ &= -417,280 + 433,284 + 696,293 + 423,641 \\ &= 1135,938.\end{aligned}$$

2.2.1 GLM avec interactions

Le modèle GLM précédent considère seulement les covariables indiquant si un assuré possède ou non un contrat. L'information sur la possession de 2 contrats en même temps n'est pas prise en compte, alors que cela peut avoir un fort impact sur la marge. En effet, on peut supposer qu'un assuré possédant un contrat `AUTO` et un contrat `MRH` n'aura pas à la fois un sinistre automobile et un sinistre MRH la même année, ces 2 risques étant indépendants. Sa marge estimée sera donc probablement plus élevée que celle de 2 individus distincts, l'un possédant uniquement un contrat `AUTO` et l'autre uniquement un contrat `MRH`.

On va donc rajouter les effets d'interactions du second ordre entre les covariables. L'inconvénient de cet ajout est que le nombre de covariables du modèle augmente considérablement. Pour un modèle composé de n covariables, le nombre de covariables d'un modèle avec interactions du $2^{ème}$ ordre est $\frac{n*(n-1)}{2} + n$. Dans notre cas, on dispose de 7 covariables (`AUTO1`, `AUTO2`, `MRH`, `CORP`, `PREV`, `EPARGNE`, `DECES`), ce qui donne un modèle avec interactions du $2^{ème}$ ordre de 28 variables, plus l'intercept.

Le tableau 2.5 présente la MSE et la MAE du modèle avec interactions du $2^{ème}$ ordre. On obtient une diminution de ces deux métriques, ce qui signifie que l'ajout des interactions améliore le pouvoir prédictif de notre modèle.

	MSE	MAE
GLM	1 871 561	884
GLM avec interactions	1 826 465	858

TABLE 2.5 : Métriques du modèle GLM et GLM avec interactions

L'amélioration de la prédictivité du modèle se fait au détriment de son explicabilité. En effet, le modèle s'écrit maintenant comme la somme de 29 coefficients, ce qui ne le rend pas très interprétable. Afin de pallier ce problème, on va pénaliser le modèle, de sorte que le nombre de variables retenues diminue.

2.2.2 Modèle LASSO

On effectue une régression pénalisée LASSO, présentée au chapitre 1. Cette régression a été effectuée à l'aide du package `R glmnet`.
L'intérêt de la pénalisation LASSO est qu'elle fixe certains coefficients à 0, ce qui facilite l'explicabilité du modèle. La pénalisation LASSO permet de garder seulement les covariables significatives, rendant notre modèle parcimonieux.

De même que pour les GLM précédents, on entraîne notre modèle sur la base d'apprentissage, puis on le teste sur la base de validation, et on compare les métriques obtenues avec celles du GLM. Une validation croisée a été effectuée ($k = 10$), puis on a sélectionné la valeur de lambda qui minimisait le plus la MSE. Les coefficients obtenus sont présentés dans le tableau 2.6.

	Estimate
(Intercept)	482,036
ind_auto1	−111,987
ind_mrh	−159,520
ind_corp	12,877
ind_prev	132,409
ind_auto1:ind_mrh	335,645
ind_auto1:ind_corp	189,233
ind_auto1:ind_prev	158,878
ind_auto1:ind_epargne	83,893
ind_auto2:ind_mrh	596,944
ind_auto2:ind_corp	280,417

	Estimate
ind_auto2:ind_prev	183,983
ind_auto2:ind_epargne	124,970
ind_auto2:ind_deces	−243,516
ind_mrh:ind_corp	291,412
ind_mrh:ind_prev	451,378
ind_mrh:ind_epargne	172,231
ind_mrh:ind_deces	−64,131
ind_corp:ind_prev	120,072
ind_corp:ind_deces	−12,469
ind_corp:ind_epargne	−30,805
ind_epargne:ind_deces	−193,114

TABLE 2.6 : Coefficients du LASSO

Les covariables n'apparaissant pas dans le tableau 2.6 sont celles qui ont été fixées à 0 par la pénalisation LASSO. On passe donc de 29 covariables à 22, ce qui simplifie l'explicabilité du modèle, sans pour autant rendre le modèle très simple.

Le tableau 2.7 présente les valeurs de MSE et MAE obtenues avec le modèle LASSO sur l'échantillon de validation. La pénalisation LASSO sur les variables et les interactions permet de diminuer ces 2 métriques, de 2,4% pour la MSE et 3% pour la MAE.

	MSE	MAE
GLM	1 871 561	884
LASSO	1 826 618	858

TABLE 2.7 : Métriques du modèle GLM et LASSO

2.2.3 Utilisation de la loi Gamma

Nous avons utilisé la loi gaussienne (ainsi que la fonction de lien identité) dans les modélisations précédentes, car la marge future peut prendre des valeurs positives ou négatives au sein de notre portefeuille.

Néanmoins, le nombre d'assurés possédant une marge future estimée négative est assez faible (10% des assurés). Il s'agit en général d'assurés ayant souscrit récemment, pour lesquels la marge future est faible ou négative. Le Tableau 2.8 suivant représente la marge future moyenne en fonction de l'ancienneté de l'assuré dans le portefeuille, ainsi que les effectifs associés. On observe ainsi que les plus petites marges sont obtenues pour les assurés possédant la plus faible ancienneté, puis que la marge croît avec l'ancienneté.

Ancienneté dans le portefeuille	Marge future	Proportion
$<$ 2 ans	596	3%
2 - 5 ans	610	15%
6 - 10 ans	676	13%
11 - 15 ans	823	10%
16 - 20 ans	999	10%
21 - 27 ans	1394	14%
$>=$ 28 ans	2213	35%

TABLE 2.8 : Marge future moyenne en fonction de l'ancienneté dans le portefeuille

Afin d'améliorer le pouvoir prédictif du modèle, on peut séparer la base de données en 2 parties, l'une avec les valeurs de marge positives, et l'autre les négatives. Cela nous permet de modéliser 2 modèles linéaires généralisés avec la loi Gamma et la fonction de lien réciproque, puis d'utiliser ces 2 modèles dans un modèle composé Bernoulli-Gamma, la paramètre p de la loi Bernoulli correspondant à la proportion empirique d'être négatif. Cette proportion vaut 0.10399.

La performance de ce modèle Bernoulli-Gamma est bien moins bonne que celles des modèles précédent, la MAE valant 1044 et la MSE 2 570 512.

En revanche, à titre informatif, on a aussi calculé les performances des 2 sous-modèles GLM avec la loi Gamma, l'un sur les marges positives et l'autre sur les marges négatives. Lorsque l'on prédit les valeurs de marge partir des deux sous-échantillons de test (séparés selon le signe de la marge), ces deux modèles sont très performants. Cette méthode n'est en revanche pas applicable à une prédiction d'un nouvel assuré, car on ne connaît pas *a priori* le signe de sa marge, et donc le sous-modèle Gamma adéquat à appliquer.

2.3 CART

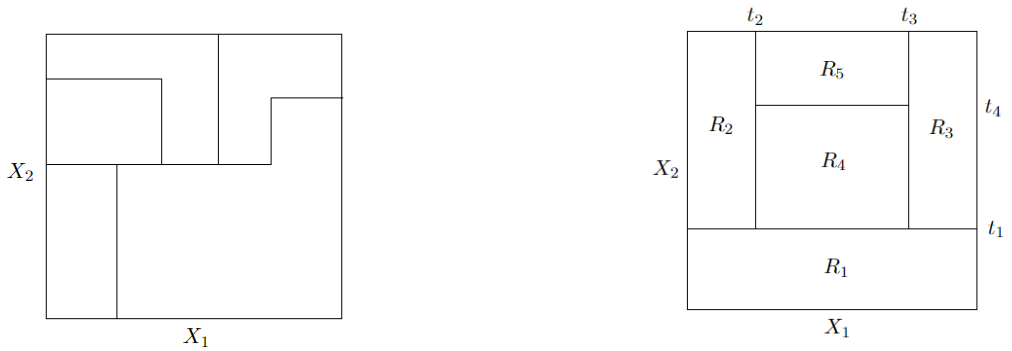
2.3.1 Introduction

Un arbre de décision est une méthode permettant de prédire ou d'expliquer une variable (qui peut être quantitative ou qualitative) à l'aide d'autres variables explicatives. On l'appelle *arbre de régression* si la variable à expliquer est quantitative, et *arbre de classification* si elle est qualitative. Les premiers arbres ont été introduits par (Morgan et Sonquist) en 1963, et ont été popularisés par l'algorithme CART (*Classification And Regression Trees*) en 1984, par (Breiman et al). Les arbres de décision sont très populaires car ils sont simples, intuitifs et visuels. On peut les comprendre et les utiliser sans avoir besoin de posséder de grandes connaissances en statistiques.

Cette section présentera l'aspect théorique de l'algorithme CART, et est basée sur le cours d'*Analyse de données* de Patrice Bertrand, dispensé en M2 Actuariat à l'Université Paris-Dauphine.

Construction d'un arbre

On considère l'exemple suivant où Y est une variable aléatoire continue que l'on explique par X_1 et X_2, deux variables quantitatives continues à valeurs dans $[0, 1]$. L'espace est découpé en plusieurs partitions dans lesquelles Y est modélisé par sa moyenne. L'algorithme CART se limite aux partitions rectangulaires (figure 2.3(b)), car la règle de prédiction d'une partition non rectangulaire (figure 2.3(a)) est très complexe. Lorsque l'espace possède plus de 2 dimensions, CART se limite aux hyperrectangles.



((a)) Partitions non rectangulaires

((b)) Partitions rectangulaires

FIGURE 2.3 : Deux partitions de [0,1]x[0,1]

Pour créer ces partitions, l'algorithme CART procède de la façon suivante :

1. CART commence par diviser l'espace entier en 2 régions rectangulaires et modélise Y par sa moyenne dans chacune des régions
2. CART choisit une variable $X_j, (j \in 1, 2)$ et un seuil s tel que la région soit séparée en 2 sous-régions selon que $X_j > s$ ou $X_j \leq s$. j et s sont choisies de sorte à optimiser le critère d'ajustement de Y.
3. CART réitère l'étape 2 jusqu'à ce qu'un critère d'arrêt soit atteint.

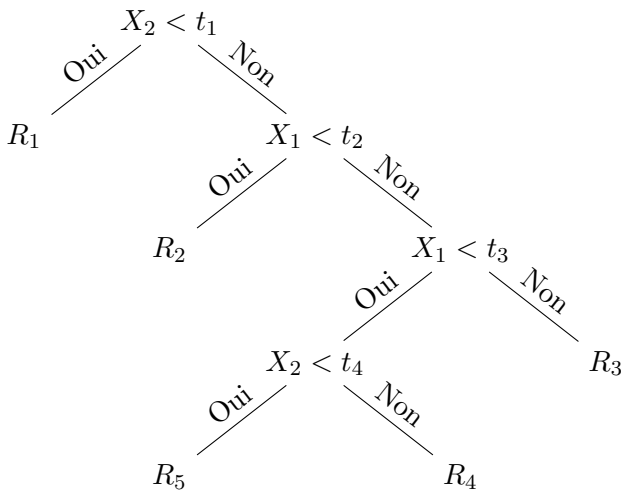


FIGURE 2.4 : Exemple d'arbre binaire

Le modèle de régression associé s'écrit donc

$$\hat{f}(X) = \sum_{m=1}^{n} c_m \mathbb{1}_{R_m}(X) \tag{2.28}$$

avec $X = (X_1, X_2)$, n le nombre de partitions, R_m la m^{eme} région et c_m une constante réelle associée à la région R_m. La constante c_m s'obtient en prenant la moyenne des observations $y_i \in R_m$.

Par exemple, dans le cas de la Figure 2.3(b) précédente, CART a commencé par séparer l'espace selon la variable X_2 et la valeur $s = t_1$. La partie supérieure a ensuite été divisé en 2 selon que $X_1 > t_2$ ou non, puis la partie droite de cette division a été divisée selon que $X_3 > t_3$. Enfin, la partition droite de cette division a été divisée selon que $X_2 > t_4$ ou non. On obtient ainsi le partitionnement R_1, R_2, R_3, R_4, R_5. On peut représenter ce modèle sous la forme d'un arbre binaire (Figure 2.4 ci-dessous), qui est facilement interprétable.

En pratique, la représentation par arbre est la seule viable lorsque le nombre de variables explicatives dépasse 2.

2.3.2 Cas général

On se place maintenant dans le cas général, où la variable quantitative Y est expliquée par p variables $X_1, ..., X_p$. Ces variables peuvent être quantitatives ou qualitatives. On dispose d'un échantillon de N individus représentés par des couples $(x_1, y_1), ..., (x_N, y_N)$, avec $x_i = (x_i^1, ..., x_i^p)$, pour $i \in 1, ..., N$. On pose le vecteur $y := (y_1 ... y_N)'$. On utilise le même algorithme récursif que dans la partie précédente : on choisit une variable X_j et une valeur de coupe s, et on divise l'espace en 2 selon cette valeur de coupe, pour $j \in 1, .., p$. On construit ainsi une partition de M régions $R_1, ..., R_M$, avec une valeur constante c_m dans chacune d'entre elles. Avec ces notations, l'ajustement du modèle s'écrit

$$f(X) = \sum_{m=1}^{M} c_m \mathbb{1}_{R_m}(x). \tag{2.29}$$

L'objectif de CART est de minimiser la somme des carrés des erreurs

$$S_M = \sum_{i=1}^{N} (y_i - f(x_i))^2. \tag{2.30}$$

La meilleure estimation de c_m, noté $\hat{c}_m$, est

$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i, \tag{2.31}$$

où $N_m = \#\{i \mid c_i \in R_m\}$, $m \in 1, ..., M$.

En pratique, à cause de sa complexité algorithmique, il est presque impossible de trouver la partition qui minimise S_M. CART utilise donc une heuristique algorithmique, c'est-à-dire une méthode de calcul qui donne des résultats réalisables et acceptables pour un problème d'optimisation difficile, mais qui ne sont pas nécessairement optimaux. L'algorithme CART sélectionne donc une variable X_j et une valeur s servant à la division de l'espace en 2 régions R_1 et R_2. On distingue deux cas, selon que X_j soit une variable quantitative ou qualitative.

- Si X_j est quantitative, alors R_1 et R_2 sont définies par :

$$R_1(j,s) = \left\{ i \mid x_i^j \leq s \right\} \quad \text{et} \quad R_2(j,s) = \left\{ i \mid x_i^j > s \right\}. \tag{2.32}$$

- Si X_j est qualitative, alors R_1 et R_2 sont définies par :

$$R_1(j,s) = \left\{ i \mid x_i^j \in s \right\} \quad \text{et} \quad R_2(j,s) = \left\{ i \mid x_i^j \notin s \right\}. \tag{2.33}$$

Dans les deux cas, X_j et s sont sélectionnées de façon à minimiser la quantité suivante :

$$\min_{j,s} \left[\min_{c1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right]. \tag{2.34}$$

Pour j et s fixées, les valeurs optimales des c_i sont :

$$\hat{c}_1 = \frac{1}{N_1} \sum_{x_i \in R_1} y_i \quad \text{et} \quad \hat{c}_2 = \frac{1}{N_2} \sum_{x_i \in R_2} y_i. \tag{2.35}$$

On calcule donc la valeur s optimale pour chaque variable, et on en déduit le couple optimal (j^*, s^*) qui minimise (2.34). On coupe le nœud m selon X_j^* et s^*, et on obtient deux nouvelles régions R_1 et R_2. On réitère le découpage pour ces 2 régions ainsi trouvées, et ainsi de suite jusqu'à ce qu'un critère d'arrêt soit atteint.

Problème du choix de la taille de l'arbre

La détermination de la taille de l'arbre est une étape cruciale de la modélisation : un arbre trop grand apprendra trop les données et peut conduire à du sur-apprentissage (overfitting), tandis qu'un arbre trop petit risque de ne pas apprendre assez les données et de faire du sous-apprentissage (underfitting).

La taille optimale de l'arbre est un paramètre déterminé de façon expérimentale, en fonction du jeu de données (tuning).

Une première approche, naturelle, serait d'arrêter le découpage pour chaque nœud si la décroissance du critère S_M n'excède pas un certain seuil, fixé à l'avance. Cette approche n'est pas optimale, car un découpage inutile peut être suivi d'un découpage pertinent.

Une seconde approche, serait de partir d'un arbre profond (i.e. avec beaucoup de feuilles), et d'examiner chaque feuille en évaluant l'effet de sa suppression à l'aide d'un échantillon test. Ce nœud (feuille) est supprimé si sa suppression entraîne une amélioration significative de la fonction objectif à optimiser. On arrête de supprimer des nœuds lorsque aucune amélioration ne peut être effectuée.

L'algorithme CART utilise une troisième approche, plus élaborée, se basant sur la précédente : l'*élagage coût-complexité*. On construit tout d'abord un arbre de grande taille noté T_0, obtenu en arrêtant le découpage lorsqu'un nœud atteint une taille limite (5 au plus dans un nœud terminal). On applique ensuite à T_0 la procédure d'élagage du type coût-complexité : on cherche des sous-arbres $T \subset T_0$ obtenus par élagage de T_0, i.e. obtenus en supprimant des nœuds internes (i.e. non terminaux) de T_0. La Figure 2.5 illustre la suppression d'un nœud interne. Un nœud interne supprimé devient donc un nœud terminal (une feuille).

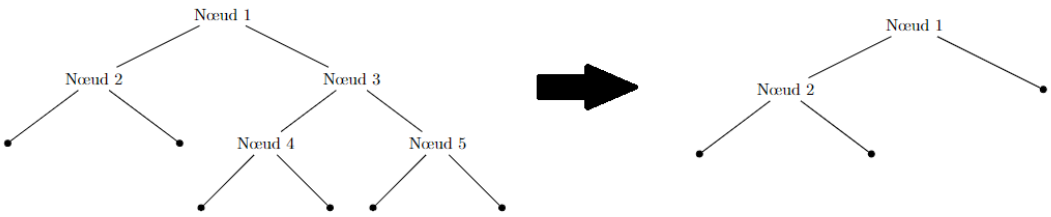


FIGURE 2.5 : Suppression du nœud 3

Soit $\tilde{T}$ l'ensemble des nœuds terminaux de l'arbre T, et $|\tilde{T}|$ le nombre de ces nœuds terminaux. Pour tout m et tout $\alpha \in \mathbb{R}_+$:

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - c_m)^2, \tag{2.36}$$

$$C_\alpha(T) = \left(\sum_{m=1}^{|T|} N_m Q_m(T) \right) + \alpha |\tilde{T}|. \tag{2.37}$$

Le paramètre $\alpha \geq 0$ pénalisant la taille de l'arbre, il règle le compromis entre adéquation de l'ajustement aux données et taille de l'arbre. Si α prend de grandes valeurs, l'arbre sera de petite taille, et inversement, si α prend de petites valeurs, l'arbre sera de grande taille. Enfin, si $\alpha = 0$, la solution sera l'arbre T_0. On détermine α par une procédure de tuning, en cherchant à minimiser le critère $C_\alpha(T)$. Pour tout $\alpha \geq 0$, on peut montrer qu'il existe un unique plus petit sous arbre de T minimisant $C_\alpha(T)$.

Méthode de détermination de T_α

On utilise l'élagage du "plus faible lien", c'est-à-dire qu'on supprime le nœud interne (et donc tous les descendants de ce nœud) qui produit le plus petit accroissement de coût $\sum_{m=1}^{|T|} N_m Q_m(T)$. On réitère ce processus d'élimination jusqu'à obtenir un arbre réduit à un seul nœud, la racine de T_0. On obtient ainsi une suite emboîtée d'arbres

$$T_0 \subsetneq T_1 \subsetneq T_2 \subsetneq \ldots \subsetneq T_L = \{r\} \tag{2.38}$$

où r est la racine de T_0, et la notation $T_p \subsetneq T_q$ signifie que T_p est un sous-arbre de T_q.

On peut montrer que la suite emboîtée d'arbres $T_0 \subsetneq T_1 \subsetneq T_2 \subsetneq \ldots \subsetneq \{r\}$ contient nécessairement T_α. Plus précisément, pour tout $l = 0, 1, ..., L$, T_l est le plus petit sous-arbre qui minimise $C_\alpha(T)$ pour tout $\alpha \in [\alpha_l, \alpha_{l+1}[$. Pour estimer la meilleure valeur de α, on évalue, pour chaque valeur de l (associée à α_l), la performance de T_l par validation croisée à k plis (5 ou 10 en général) ; on en déduit l'arbre optimal $T_{\hat{\alpha}_l}$ qui minimise $C_{\hat{\alpha}_l}(T)$. C'est l'arbre retenu par la méthode CART.

2.3.3 Avantages et limites de l'algorithme CART

L'algorithme CART, comme toutes les méthodes d'apprentissage, possède des avantages et des inconvénients. Les connaître permet de l'utiliser dans des situations adaptées, afin de maximiser ses performances.

Avantages

- l'algorithme CART est une méthode non paramétrique : il n'y a pas besoin d'inférer sur une loi de probabilité représentant la variable à expliquer.
- CART permet d'utiliser la même variables plusieurs fois, à différents endroits de l'arbre. Cela permet d'expliquer une partie du jeu de données en fonction d'une variable, et ce même si une autre partie du jeu de données ne dépend pas de cette variable.
- Les valeurs extrêmes n'ont pas d'effet significatif sur CART.
- Il n'y a pas besoin de préparer les données en amont de la modélisation : CART n'a pas besoin de données normalisées et/ou mise à la même échelle pour fonctionner (pre-processing).
- CART fonctionne avec des variables numériques et catégorielles.
- Les résultats de l'algorithme sont faciles à interpréter, et le principe de l'algorithme est intuitif. L'arbre final est facilement visualisable, et toute personne peut comprendre pourquoi telle ou telle valeur a été prédite, selon les variables explicatives données.

Inconvénients

- L'algorithme CART est instable : un petit changement dans les données d'apprentissage peut fortement impacter l'arbre finale, et donc les prédictions de CART. Ainsi, en fonction du partitionnement du jeu de données entre la base d'apprentissage et la base de test, on peut obtenir des résultats très différents.

- L'algorithme CART prend souvent plus de temps à être modélisé que les autres méthodes d'apprentissage.
- CART a tendance à faire du sur-apprentissage, malgré la procédure d'élagage présentée précédemment.
- L'algorithme CART ne peut pas prédire des valeurs qui ne sont pas dans la base d'apprentissage. Si les valeurs de Y de certains individus diffèrent fortement de celles de la base d'apprentissage, CART ne pourra pas prédire ces individus. De même, pour une variable catégorielle, si une modalité n'était pas présente dans la base d'apprentissage mais qu'elle existe dans la base de test, CART ne pourra pas utiliser cette information.
- La sensibilité des variables explicatives s'interprète moins facilement que dans un GLM.

2.4 Modélisation CART

2.4.1 Modèle CART de base

On applique maintenant le modèle CART à nos données. Cette modélisation a été effectuée à l'aide du package `R rpart`. Comme pour les modèles précédents, on se restreint à la détention ou non d'un type de contrat, sans prendre en compte les caractéristiques individuelles. L'arbre de régression CART va donc effectuer des coupes dans notre base d'apprentissage en fonction de la détention d'un type de contrat ou non, ce qui en fait un modèle très simple à expliquer. Les modèles CART ont déjà pour avantage d'être facilement interprétables, mais dans le cas de variables explicatives valant 1 ou 0, ils le sont encore plus : il suffit de savoir si un assuré possède ou non le contrat utilisé par CART pour la coupe afin de savoir dans quelle branche va l'assuré.

Ce premier modèle CART a pour paramètre de complexité 0,01 (*cp* pour *Complexity Parameter* en anglais), qui est la valeur par défaut dans le package `rpart`. Ce paramètre *cp* correspond au α de l'algorithme CART présenté dans le chapitre 1. Dans la suite, on modifiera ce paramètre afin d'améliorer notre modèle CART.

De même que pour les modèles GLM et LASSO précédent, une validation croisée a été effectuée sur la base d'apprentissage ($k = 5$). Le tableau 2.9 donne les résultats obtenus :

	MSE	MAE
Bloc 1	1 852 634	891,571
Bloc 2	1 874 850	894,383
Bloc 3	1 891 061	894,160
Bloc 4	1 880 584	896,342
Bloc 5	1 889 135	896,514

TABLE 2.9 : Métriques de la validation croisée CART

On obtient des valeurs de MSE et MAE assez proches selon les différents blocs de validation croisée, on peut donc entraîner notre modèle sur la base d'apprentissage entière, sans risquer de sur-apprentissage.

Un autre avantage de CART est que c'est un modèle très visuel : on peut prédire n'importe quel individu en regardant seulement l'arbre estimé, et ce sans avoir besoin de connaissances en statistiques.

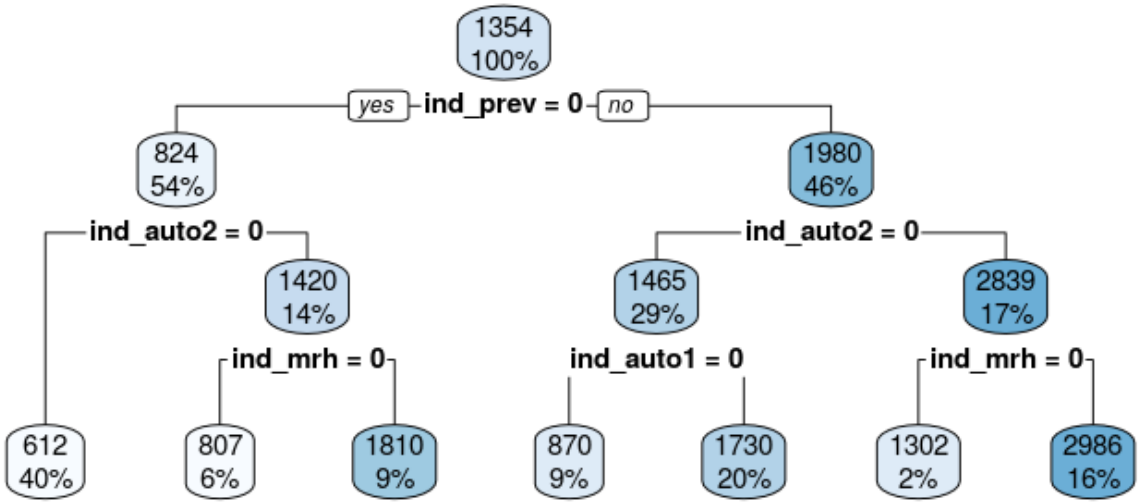


FIGURE 2.6 : Arbre CART

La Figure 2.6 ci-dessus représente notre modèle, et le tableau 2.10 résume les valeurs des métriques de comparaison du modèle GLM et du modèle CART.

	MSE	MAE
GLM	1 871 561	884
CART	1 902 705	895

TABLE 2.10 : Métriques du modèle GLM et CART

Le modèle CART apparaît comme étant moins performant que le GLM. Cependant, comme indiqué précédemment, cet arbre CART a été construit avec un paramètre de complexité $cp = 0,01$. En faisant varier ce paramètre, on change la profondeur de l'arbre, et donc son pouvoir prédictif. Une valeur de cp est calculée avant chaque coupe potentielle : si cette valeur est inférieure à la valeur de cp de seuil choisie, alors la coupe n'est pas effectuée. Autrement dit, si ajouter une nouvelle coupe n'améliore pas la performance du modèle par rapport à cp de seuil, on ne sépare pas le nœud de l'arbre. Ainsi, si le seuil de cp diminue, la profondeur de l'arbre augmente, et elle diminue lorsque le seuil de cp augmente.

2.4.2 Modèle CART avec différentes valeurs de cp

On modélise plusieurs arbres en changeant la valeur de cp, avec l'objectif de minimiser la MSE et la MAE. Le tableau 2.11 représente les valeurs de ces métriques en fonction du modèle utilisé, ainsi que le temps nécessaire à chaque modélisation. En effet, plus cp diminue, plus la profondeur de l'arbre augmente, mais cela se fait au détriment du temps de calcul. Les modèles ont été entraînés sur la base d'apprentissage, et les métriques calculées sur la base de test.

	MSE	MAE	Temps
cp = 1e-02	1 902 705	894,551	25 sec
cp = 1e-03	1 843 805	864,583	30 sec
cp = 1e-04	1 827 208	857,748	31 sec
cp = 1e-05	1 825 837	857,088	32 sec
cp = 1e-06	1 825 623	856,997	34 sec
cp = 1e-07	1 825 625	856,992	36 sec
cp = 1e-08	1 825 624	856,992	36 sec
cp = 1e-09	1 825 623	856,991	36 sec
cp = 1e-10	1 825 623	856,991	37 sec
cp = 1e-11	1 825 623	856,991	37 sec

TABLE 2.11 : Métriques selon le paramètre de complexité du modèle CART

On observe que plus le paramètre *cp* diminue, plus la MSE et la MAE diminuent. Cela semble contre-intuitif au regard du principe de sur-apprentissage : plus l'arbre est profond et plus il s'adapte à la base d'apprentissage, mais moins il sera capable de bien prédire des nouvelles données. Dans notre cas, le modèle sur-apprend la base d'apprentissage, mais prédit très bien les nouvelles données (la base de test). Cela vient du fait que notre base de donnée de départ est très fournie : elle contient plus de 2 750 000 individus. Les informations de la base de test (40% des données) sont donc en grande partie contenues dans la base d'apprentissage, et le modèle prédit donc très bien les individus de la base de test.

La Figure 2.7 représente l'arbre ayant le meilleur pouvoir prédictif sur les données de test, pour cp=1e-09. On perd l'un des principaux avantages de CART : sa lisibilité et facilité d'interprétation.

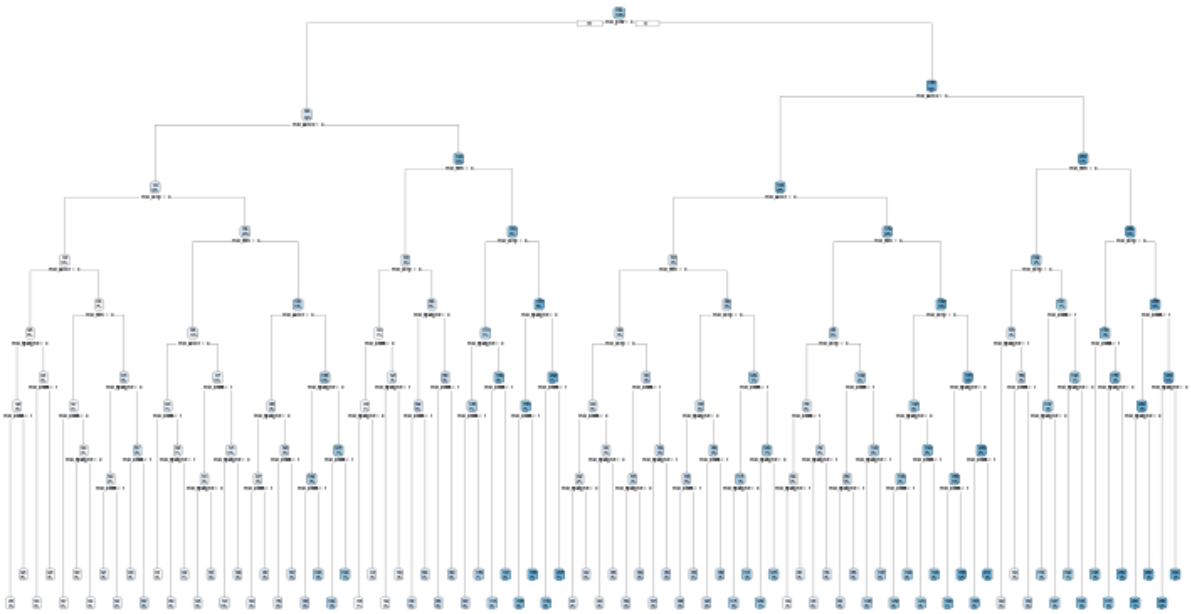


FIGURE 2.7 : Arbre CART - cp=1e-09

Le modèle CART avec cp=0,001 propose un bon compromis entre la performance et la lisibilité. Il est composé de 9 coupes et de 11 feuilles, pour une profondeur maximale de 4, ce qui le rend facilement

interprétable. Cet arbre est représenté dans la Figure 2.8 ci-dessous.

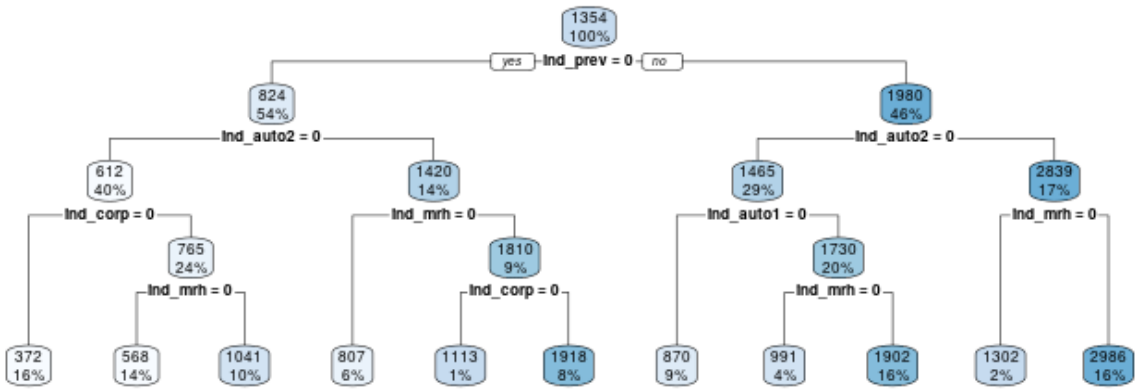


FIGURE 2.8 : Arbre CART - cp=0.001

Cependant, dans une optique de prédiction pure, le modèle CART avec $cp = 1e-09$ sera le plus adapté. La lisibilité de l'arbre ainsi que l'interprétabilité des déterminants de la marge n'étant plus importante, la meilleure performance est obtenu par l'arbre CART avec $cp = 1e-09$: cet arbre sera donc privilégié si l'assureur souhaite uniquement prédire avec précision la marge future d'un assuré. Cette étude étant tournée vers l'interprétabilité et la recherche des déterminants de la marge future des assurés, l'arbre CART présenté en Figure 2.8 ci-dessus est le plus approprié.

Chapitre 3

Modèle de partitionnement récursif

Dans la précédente section, on a modélisé la marge future à l'aide de différentes méthodes statistiques. On cherche maintenant à améliorer ces résultats à l'aide d'une nouvelle classe de modèles, les modèles MOB (*MOdel Based recursive partioning*). Il existe de nombreux modèles de machine learning ayant un pouvoir prédictif supérieur aux GLM ou aux arbres de décision (CART) vus précédemment, tels que les forêts aléatoires, les réseaux de neurones ou encore le boosting. Ces modèles ont néanmoins un gros inconvénient : ils sont peu interprétables, ce qui leur vaut le surnom de modèles "boîte noire". À l'inverse, les modèles que nous étudierons, les *GLM trees*, font partie de la classe des MOB et combinent les arbres de décision et les GLM, ce qui leur donne de meilleures performances que ces 2 méthodes utilisées séparément. Ils sont, de plus, facilement interprétables. Dans un premier temps, nous présenterons la théorie mathématique des modèles MOB, puis nous les appliquerons à notre jeu de données.

Les modèles MOB ont été introduits par Achim Zeileis, Torsten Hothorn and Kurt Hornik (Zeileis et al., 2008), et reposent sur le principe suivant : l'espace est séparé en plusieurs partitions, et on applique un modèle différent à chacune de ces partitions, créant différents sous-modèles en fonction de la partition. Ainsi, les paramètres des sous-modèles sont beaucoup plus ajustés que dans le cas d'un seul modèle appliqué à l'ensemble de l'espace. Cela permet aussi de traiter des données non-linéaires en leur appliquant des sous-modèles localement linéaires. Ces modèles sont appelés "modèles segmentés".

3.1 Théorie mathématique

Cette section se base sur les sections 2 et 3 de (Zeileis et al., 2008). La Figure 3.1 présente un modèle MOB simple, permettant d'illustrer le concept des modèles segmentés.

3.1.1 Modèles segmentés

On considère un jeu de données $(Y_i, X_i)_{i=1,\dots,n}$. $Y \in \mathbb{R}^n$ est la variable d'intérêt à expliquer, et $X \in \mathbb{R}^{n\times p}$ est la matrice contenant les p variables explicatives du modèle. On souhaite ajuster un modèle paramétrique $\mathcal{M}(Y, X, \theta)$ sur ce jeu de données, avec $\theta \in \Theta$ (vecteur de dimension p) le paramètre du modèle. Le modèle est ajusté en minimisant une fonction objectif $\Phi(Y, X, \theta)$, ce qui donne l'estimation

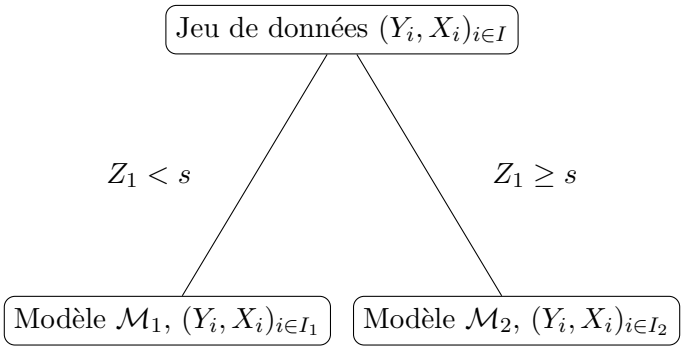


FIGURE 3.1 : Modèle MOB simplifié

du paramètre $\hat{\theta}$ suivante :

$$\hat{\theta} = \arg\min_{\hat{\theta}} \sum_{i=1}^{n} \Phi(Y_i, X_i, \theta). \tag{3.1}$$

Cet estimateur peut prendre différentes formes, dont certaines sont habituellement utilisées, telles que l'estimateur des moindres carrés ordinaire (EMCO) lorsque la fonction Φ correspond à l'erreur au carré, ou encore l'estimateur du maximum de vraisemblance (*Maximum Likelihood* en anglais) lorsque Φ est la log-vraisemblance négative.

Dans de nombreuses situations, on ne peut pas affirmer qu'un unique modèle $\mathcal{M}(Y, X, \theta)$ s'ajuste bien aux n observations. Dans ce cas, on peut séparer les observations en plusieurs groupes par rapport à certaines variables explicatives, de sorte qu'un sous-modèle puisse être ajusté localement à chaque partition. Les variables explicatives servant à ce découpage en plusieurs partitions seront appelées *variables de partitionnement*, et seront notées $Z_j, j = 1, ..., l$. Les autres variables explicatives, qui ne sont pas utilisées pour le partitionnement de l'espace, seront utilisées pour prédire Y et seront appelées *variables de régression*.

On va donc définir une partition $\{\mathcal{B}_b\}_{b=1,...,B}$ de l'espace $\mathcal{Z} = Z_1 \times ... \times Z_l$, avec B le nombre de partitions. Dans chaque partition $\{\mathcal{B}_b\}$, on ajuste un sous-modèle $\mathcal{M}(Y, X, \theta_b)$ sur les données (Y_i, X_i) de cette partition, avec θ_b le vecteur de paramètre associé à ce sous-modèle.

Afin d'estimer les paramètres $\{\theta_b\}$, on minimise la fonction objectif suivante :

$$\sum_{b=1}^{B} \sum_{i \in I_b} \Phi(Y_i, X_i, \theta_b), \tag{3.2}$$

où I_b est l'ensemble des indices de la partition $\mathcal{B}_b$.

Lorsque la partition $\{\mathcal{B}_b\}$ est connue, cette minimisation est réalisée en minimisant localement la fonction objectif sur chaque segment $\mathcal{B}_b$. Néanmoins, lorsque la partition est inconnue, il devient compliqué de minimiser la fonction objectif 3.2 sur toutes les partitions $\{\mathcal{B}_b\}$ possibles. En effet, s'il y a plus d'une variable de partitionnement (i.e. $l > 1$), le nombre de partitions potentielles devient rapidement très grand. Si, de plus, on ne connaît pas le nombre B de partitions, cette minimisation devient encore plus compliquée.

En résumé, trouver la partition optimale (par rapport à la fonction objectif Φ) est compliqué, même lorsque l'on connaît la valeur de B. Néanmoins, dans le cas particulier avec une seule variable de partitionnement, la séparation de l'espace est simple : la littérature regorge d'algorithmes permettant de segmenter un modèle par rapport à une unique variable. L'idée des MOB sera d'exploiter cette méthodologie afin de trouver une partition proche de la partition optimale pour un nombre de variables de partitionnement $l > 1$. On appliquera une méthode d'*algorithme glouton* (*greedy search* en anglais). Cet algorithme est basé sur l'idée que si l'on effectue un choix optimal localement à chaque étape, on peut tomber sur un optimal global à la fin. Les modèles MOB appliquent cet algorithme en optimisant la fonction objectif Φ localement à chaque étape.

3.1.2 Algorithme de partitionnement récursif

L'algorithme de partitionnement récursif va créer différentes partitions de l'espace, dans lesquelles un sous-modèle sera ajusté. Il fonctionne sur le même principe que les arbres de décision : toutes les observations sont présentes dans le nœud initial (la racine), puis vont être séparées en B nœuds fils, selon une variable explicative (qui peut être quantitative ou qualitative). On répète ensuite cette procédure pour les nœuds fils, si l'on considère qu'une nouvelle séparation du nœud est nécessaire.

Afin de déterminer si l'on doit séparer un nœud en B nœuds fils ou non (auquel cas le nœud devient un nœud terminal), un test d'instabilité des paramètres est effectué. Si l'instabilité est considérée significative par rapport à l'une des variables de partitionnement Z_j, le nœud est séparé en B nœuds fils, et on continue l'algorithme. À l'inverse, si aucune instabilité des paramètres n'est considérée comme significative pour tous les nœuds existants, l'algorithme récursif s'arrête et retourne l'arbre de partitionnement final. Pour chaque partition finale $\mathcal{B}_b$ (correspondant à un nœud terminal de l'arbre), un sous modèle $\mathcal{M}_b(Y_i, X_i, \theta_b)$ est ajusté.

Les étapes de l'algorithme récursif sont résumées ci-après :

1. Ajuster le modèle sur toutes les observations du nœud, en minimisant la fonction objectif Φ :

$$\hat{\theta} = \arg\min_{\hat{\theta}} \sum_{i=1}^{n} \Phi(Y_i, X_i, \theta). \tag{3.3}$$

2. Évaluer la stabilité de l'estimation du paramètre par rapport aux variables de partitionnement $Z_1, ..., Z_l$. S'il y a une instabilité significative par rapport à une ou plusieurs variables $Z_1, ..., Z_l$, sélectionner celle associée à la plus grande instabilité, notée Z_{j^*}. S'il n'y pas d'instabilité significative, s'arrêter : le nœud devient alors une feuille.

3. Calculer le ou les points de séparation, par rapport à la variable Z_{j^*}, qui optimisent localement Φ. Le nombre de séparations peut être adaptatif ou fixé.

4. Séparer le nœud en B nœud fils et recommencer pour chacun d'eux.

Les étapes 1 à 3 décrites précédemment sont détaillées ci-dessous. Pour simplifier les notations, on utilisera n pour le nombre d'observations du nœud étudié, $\hat{\theta}$ pour l'estimation du paramètre associé et B pour le nombre de nœuds fils choisis.

Estimation des paramètres

Cette étape de l'algorithme est habituelle : on trouve le paramètre θ solution de l'équation 3.3 résolvant les conditions du premier ordre :

$$\sum_{i=1}^{n} \phi(Y_i, X_i, \hat{\theta}) = 0 \tag{3.4}$$

avec

$$\phi(Y, X, \theta) = \frac{\partial \Phi(Y, X, \theta)}{\partial \theta}. \tag{3.5}$$

$\phi(Y, X, \theta)$ est appelée *fonction score*. On ne peut pas toujours trouver de formule fermée pour la valeur de $\hat{\theta}$, mais dans la majorité des cas, on dispose d'algorithme pour la calculer. La fonction $\hat{\phi}_i = \phi(Y_i, X_i, \hat{\theta})$, qui est l'évaluation de la fonction score en les paramètres estimés, est inspectée dans la section suivante pour déterminer l'instabilité des paramètres.

Test de l'instabilité des paramètres

Dans cette étape, le but de l'algorithme est de trouver si les paramètres du modèle ajusté précédemment sont stables par rapport à tout ordre des variables de partitionnement Z_j, ou si séparer l'échantillon par rapport à l'une des variables Z_j pourrait entraîner une instabilité dans les paramètres, et donc améliorer l'ajustement.

Pour évaluer l'instabilité des paramètres par rapport à une variable de partitionnement Z_j, une idée naturelle serait de vérifier si les scores $\hat{\phi}_i$ fluctuent aléatoirement autour de leur moyenne égale à 0, ou si l'on peut trouver un schéma de déviation systématique de 0 par rapport à Z_j. Ces déviations peuvent être capturées par le processus de fluctuation empirique suivant :

$$W_j(t) = \hat{J}^{-1/2} n^{-1/2} \sum_{i=1}^{\lfloor nt \rfloor} \hat{\phi}_{\sigma(Z_{ij})} \quad (0 \leq t \leq 1), \tag{3.6}$$

où $\sigma(Z_{ij})$ est la permutation d'ordonnancement qui donne l'anti-rang de l'observation Z_{ij} dans le vecteur $Z_j = (Z_{1j}, ..., Z_{nj})$. L'anti-rang d'une valeur z_i dans le vecteur $z = (z_1, ..., z_n)$ correspond à l'indice de cette valeur lorsque le vecteur z est trié par ordre décroissant.
Ainsi, $W_j(t)$ est le processus de la somme partielle des scores ordonnés par la variable Z_j, mis à l'échelle en fonction de $\hat{J}$, l'estimation de la matrice de covariance $cov(\phi(Y, X, \theta))$, et du nombre d'observations n appartenant au nœud. Pour estimer $\hat{J}$, on peut utiliser la formule suivante

$$\hat{J} = n^{-1} \sum_{i=1}^{n} \phi(Y_i, X_i, \hat{\theta}) \phi(Y_i, X_i, \hat{\theta})^T, \tag{3.7}$$

mais d'autres estimateurs robustes (comme l'estimateur HC ou l'estimateur de Newey-West) peuvent aussi être utilisés. Ils ne seront pas étudiés dans le cadre de ce mémoire.

Le processus empirique $W_j(t)$ repose sur le théorème central limite fonctionnel (aussi appelé *Théorème de Donsker*) sous l'hypothèse nulle de stabilité du paramètre : $W_j(t)$ converge vers un *pont brownien* W^0. Le théorème central limite fonctionnel établit la convergence en loi d'une marche aléatoire vers un processus stochastique gaussien. Un pont brownien est un processus stochastique $(W_t, t \geq 0)$ à temps continu suivant la loi d'un processus de Wiener sachant l'événement $B_0 = B_1 = 0$.

On effectue un test généralisé de M-fluctuation (Zeileis et Hornik, 2007). Deux versions de la statistique de test peuvent être utilisées, selon que la variable de partitionnement Z_j soit quantitative ou qualitative.

Variable quantitative : on utilise la statistique de test suivante pour capturer l'instabilité par rapport à la variable quantitative Z_j :

$$\lambda_{\sup LM}(W_j) = \max_{i=\underline{i},\ldots,\bar{i}} \left(\frac{i}{n}\cdot\frac{n-i}{n}\right)^{-1} \left\| W_j\left(\frac{i}{n}\right)\right\|_2^2. \tag{3.8}$$

Cette statistique correspond au maximum de la norme L_2 au carré du processus empirique de fluctuation, mis à l'échelle par sa variance. On obtient l'intervalle $[\underline{i}, \bar{i}]$ en fixant une taille minimale de segment $\underline{i}$, puis en prenant $\bar{i} = n - \underline{i}$. Cette statistique de test est asymptotiquement équivalente au supremum de la statistique du test de rapport de vraisemblance (*likelihood-ratio test* en anglais), mais possède l'avantage qu'on n'a besoin d'ajuster le modèle qu'une seule fois sous l'hypothèse nulle de stabilité des paramètres, et non sous toutes les hypothèses alternatives pour tous les points de changement possibles dans $[\underline{i}, \bar{i}]$. La statistique supLM a pour distribution limite le supremum d'un processus de Bessel lié au carré, donné par $\sup_t (t(1-t))^{-1} \left|W^0(t)\right|_2^2$, à partir duquel on peut calculer les p-values p_j correspondantes.

Variable qualitative : pour capturer l'instabilité par rapport à une variable quantitative Z_j possédant C modalités, on ne peut pas utiliser la même statistique que précédemment, car par définition, Z_j a des égalités et on ne peut donc ordonner les observations. La statistique la plus naturelle, qui n'est pas impactée par l'ordre des C modalités ni l'ordre des observations, est la suivante :

$$\lambda_{\chi^2}(W_j) = \sum_{c=1}^{C} \frac{|I_c|^{-1}}{n} \left\| \Delta_{I_c} W_j\left(\frac{i}{n}\right)\right\|_2^2, \tag{3.9}$$

avec $\Delta_{I_c} W_j$ l'incrément du processus de fluctuation empirique par rapport aux observations dans la modalité $c = 1, ..., C$ (avec les indices associés I_c). Cela correspond essentiellement à la somme des scores dans la catégorie c. La statistique de test est donc la somme pondérée des normes L_2 des incréments, et possède pour distribution asymptotique une loi du χ^2 avec $k \times (C-1)$ degrés de liberté. On peut alors calculer les p-value p_j correspondantes (Hjort et Koning, 2002).
L'intérêt d'utiliser cette approche basée sur les processus de fluctuation empiriques de l'équation 3.6, avec les statistiques 3.8 et 3.9, est que l'estimation des paramètres et les fonctions de score correspondantes n'ont besoin d'être calculées qu'une seule fois par nœud. Pour évaluer l'instabilité des paramètres, il suffit de réordonner et d'agréger selon la statistique de test à chaque fois.
Au final, pour tester s'il y a une instabilité significative dans le nœud, il suffit de vérifier si la p-value $\min_{j=1,\ldots,l} p_j$ est inférieure à un niveau de significativité α, fixé à l'avance. Si c'est le cas, la variable de partitionnement Z_j^* associée à la p-value minimale est choisie pour séparer l'échantillon dans l'étape suivante.

Séparation

Si une instabilité a été repérée dans l'étape précédente, l'algorithme procède à l'étape de séparation. Cela consiste à découper l'échantillon en B segments par rapport à la variable de partitionnement Z_{j^*}, avec B pouvant être fixé ou déterminé adaptativement. On utilisera l'algorithme CART dans ce mémoire pour séparer les différents nœuds, ce qui implique que B prendra la valeur 2 dans notre cas. Cette valeur étant fixée, on peut facilement comparer 2 segmentations différentes : il suffit de comparer la fonction objectif $\sum_{b=1}^{2} \sum_{i \in I_b} \Phi(Y_i, X_i, \theta_b)$. Concernant la complexité des calculs pour trouver la meilleure séparation, elle dépend du type de la variable de partitionnement

Variable quantitative : pour une variable de partitionnement quantitative, la recherche exhaustive est faisable en $O(n)$ opérations si $B = 2$, et en $O(n^{B-1})$ lorsque $B > 2$. Cependant, une partition

optimale peut être trouvée en $O(n^2)$ opérations avec un algorithme de programmation dynamique.

Variable qualitative : pour une variable qualitative à C modalités, 2 possibilités naturelles s'offrent à nous : séparer l'échantillon en C segments, chacun selon une modalité ($B = C$), ou séparer l'échantillon en 2 segments ($B = 2$). Dans le second cas, cela se fait en $O(2^{C-1})$ opérations, et est réduit à $O(C)$ opérations si la variable est ordonnée et que la séparation se fait par rapport à cet ordre.

En résumé, pour trouver B, deux stratégies sont utilisées : toujours prendre $B = 2$ et effectuer des séparations binaires (ce qui est le cas de CART), ou choisir B adaptativement si la variable de partitionnement est quantitative et prendre $B = C$ si elle est qualitative.

On procède ensuite à la séparation du nœud (étape 4), puis on recommence l'algorithme pour les B nœuds fils, jusqu'à ce qu'aucune instabilité significative ne soit plus trouvée.

3.2 Application des MOB

La modélisation des modèles MOB s'effectue à l'aide de la fonction `glmtree` du package `R partykit`. Comme présenté dans la section précédente, les MOB prennent en entrée des variables de partitionnement et des variables de régression, ainsi que différents paramètres, dont la profondeur de l'arbre. Ce paramètre est important, car c'est de lui que dépend la performance de l'algorithme : une profondeur faible impliquera un sous-apprentissage, tandis qu'une profondeur élevée segmentera beaucoup notre échantillon et engendrera un sur-apprentissage.

3.2.1 MOB en fonction de la détention des contrats

Une première méthode consiste à utiliser toutes les variables indiquant si un assuré détient ou non un type de contrat, à la fois en tant que variables de partitionnement et de régression. En effet, dans une modélisation MOB classique, l'algorithme peut utiliser une variable dans ces deux rôles : par exemple, si une variable représente la taille d'un individu, MOB peut segmenter selon que l'individu mesure plus ou moins de 176 cm, puis utiliser la taille comme variable de régression en lui affectant un coefficient. Il pourrait même segmenter de nouveau selon que les individus de moins de 176 cm mesurent plus ou moins de 160 cm.

Notre modélisation est différente, car les variables explicatives sont des indicatrices, prenant les valeurs `1` ou `0`. Si l'algorithme segmente sur la variable `ind_auto1` par exemple, représentant la détention du contrat `AUTO1`, alors toutes les observations de ce segment posséderont un contrat `AUTO1`, i.e. `ind_auto1=1`. L'algorithme ne pourra plus utiliser cette variable comme variable de partitionnement (puisque toutes les observations ont la même valeur `ind_auto1=1`), ni comme variable de régression, car cela reviendrait à ajouter le même coefficient à toutes les observations de ce segment, autrement dit créer un second intercept. Dans ce cas, la fonction `glmtree` assigne une valeur NA au coefficient de la régression, et l'intercept contient l'information de cette variable. La loi gaussienne est utilisée par défaut.

Cette méthodologie possède un inconvénient lorsque plusieurs variables indicatrices sont utilisées comme variables de partitionnement. Elles seront toutes fixées à NA dans la régression, et l'intercept sera modifié en conséquence, mais on perd l'interprétation de la variable. En effet, prenons un exemple dans lequel l'algorithme a segmenté une première fois sur la variable `ind_auto1` puis une seconde fois sur `ind_mrh`. L'intercept prend la valeur 267, mais on ne sait pas dans quelles proportions `ind_auto1`

et `ind_mrh` ont impactées cette valeur.

La Figure 3.2 représente le résultat du modèle MOB lorsque la profondeur de l'arbre est fixée à 3. On obtient donc 4 segments, sur lesquels une régression linéaire a été effectuée, donnant les coefficients affichés. Cet exemple illustre les paragraphes précédents, des valeurs NA apparaissant lorsque les variables sont utilisées comme variables de partitionnement. On observe aussi que la variable `ind_auto1` vaut NA dans les nœuds 6 et 7, alors qu'elle n'a pas été utilisée comme variable de partitionnement. Cela vient de la totale corrélation de `ind_auto1` avec `ind_auto2`. En effet, si un assuré possède le contrat `AUTO2`, alors il possède nécessairement le contrat `AUTO1` par construction du portefeuille. Les nœuds 6 et 7 ne comportant que des assurés détenant le contrat `AUTO2` (et par extension le contrat `AUTO1`), la variable `ind_auto1` vaut 1 pour tous les assurés de ces segments.

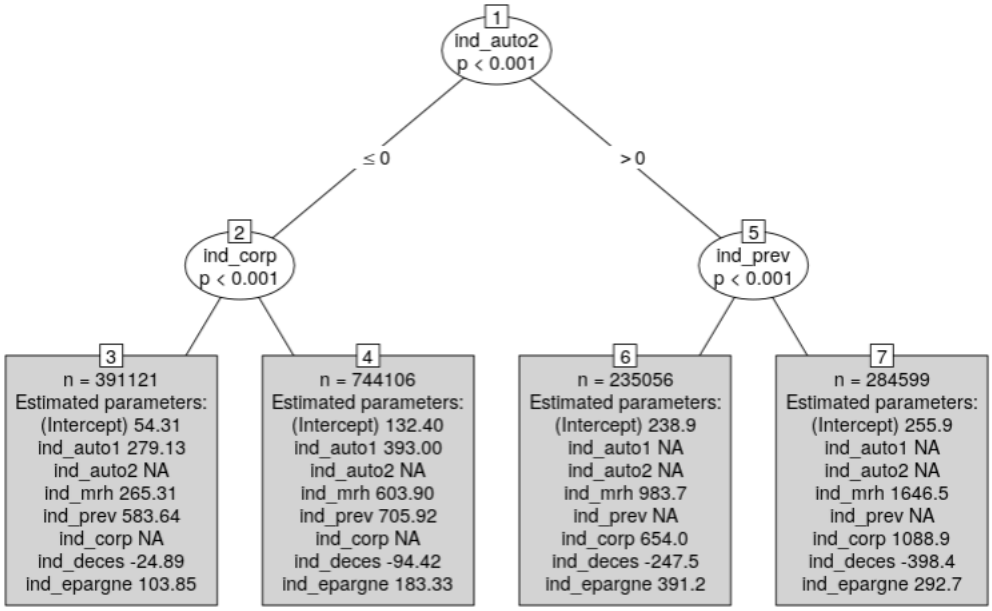


FIGURE 3.2 : Arbre MOB de profondeur 3

La lecture de l'arbre est la suivante : lorsqu'une branche a la valeur de coupe "≤ 0", la variable vaut 0 (i.e. l'assuré ne détient pas le portefeuille), et lorsque la valeur de coupe vaut "> 0", la variable vaut 1 (i.e. l'assuré détient le contrat). Afin d'évaluer la performance de ce modèle, on calcule les métriques MSE et MAE sur l'échantillon de test, et on compare avec le modèle GLM du chapitre 2. Le Tableau 3.1 résume ces valeurs.

	MSE	MAE
GLM	1 864 054	887
MOB avec profondeur 3	1 832 250	861

TABLE 3.1 : Métriques du modèle GLM et MOB avec profondeur 3

Le modèle MOB avec une profondeur de 3 performe mieux que le GLM classique, mais moins bien que d'autres modèles testés précédemment, tels que l'arbre CART avec le paramètre $cp = 0,00001$ ou le GLM avec interactions. On va donc augmenter la profondeur de l'arbre afin de rendre notre modèle plus performant. Cela se fait au détriment du temps de calcul, car augmenter la profondeur de l'arbre implique de calculer de nouvelles coupes et d'effectuer plus de régressions. Le Tableau 3.2 résume les résultats obtenus avec différentes profondeurs par l'algorithme MOB.

	MSE	MAE	Temps	Nombre de feuilles
MOB profondeur 3	1 832 250	861	67 sec	4
MOB profondeur 4	1 827 421	858	80 sec	8
MOB profondeur 5	1 825 924	857	91 sec	14
MOB profondeur 6	1 825 905	857	95 sec	16

TABLE 3.2 : Métriques selon la profondeur de l'arbre

On remarque que les métriques de MSE et MAE sont très proches pour des profondeurs de l'arbre de 4, 5 ou 6. En revanche, le temps de calcul est plus faible pour le modèle avec une profondeur de 4. On conservera donc ce modèle dans la suite de la section. Il est en outre plus simple à interpréter et expliquer, car composé de seulement 8 feuilles, contre respectivement 14 et 16 pour les arbres de profondeur 5 et 6. La Figure 3.3 représente l'arbre du modèle de profondeur 4, avec les différents coefficients des régressions linéaires pour chaque feuille.

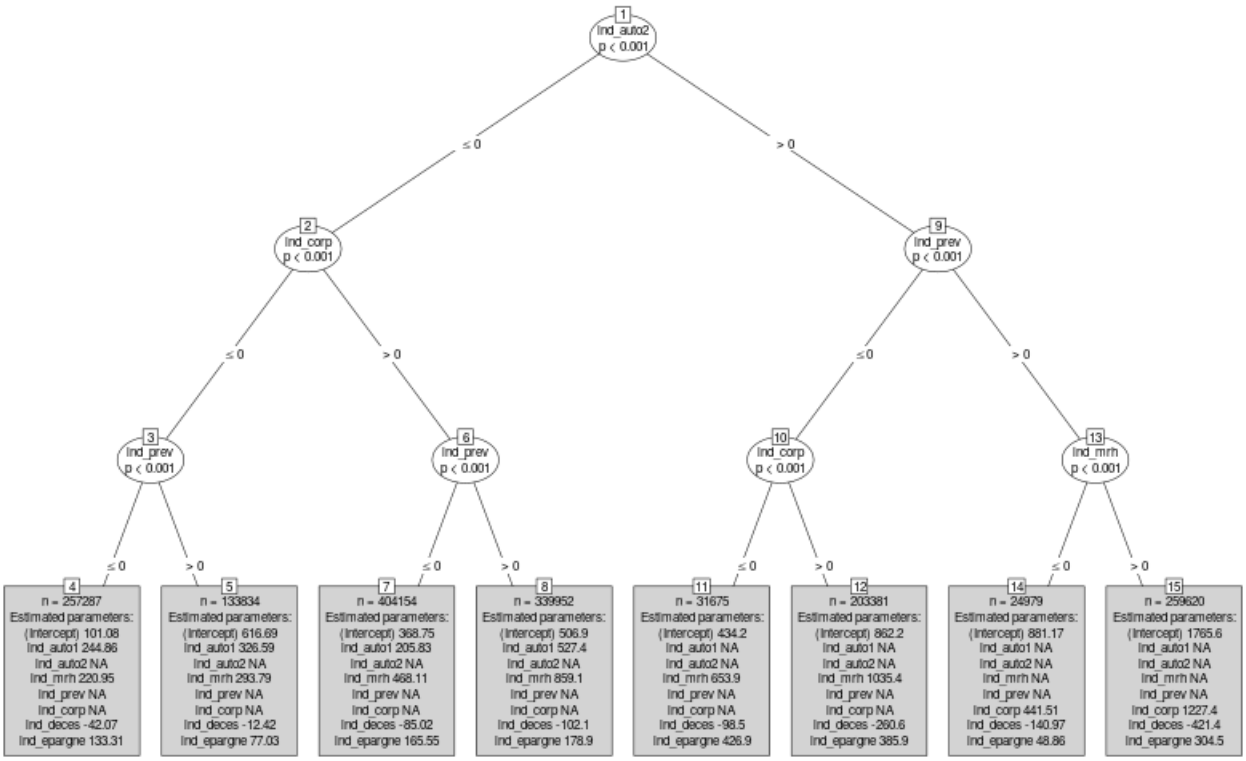


FIGURE 3.3 : Arbre MOB de profondeur 4

Comme expliqué précédemment, on observe que plus la profondeur de l'arbre augmente, autrement dit plus on utilise de variables de partitionnement, plus les modèles de régression sont réduits et sont composés de coefficients de régression nuls (`NA`).

3.2.2 Analyse de la marge par segment

Dans une perspective d'analyse de la marge future, on dispose grâce à l'algorithme MOB d'une segmentation des assurés selon la détention des différents contrats. On peut réaliser une analyse rapide de cette marge en prenant la moyenne des différentes feuilles, sans prendre en compte la partie régression,

qui est utile à une maille plus fine. La Figure 3.4 représente les moyennes des différentes feuilles du modèle MOB avec profondeur de 4.

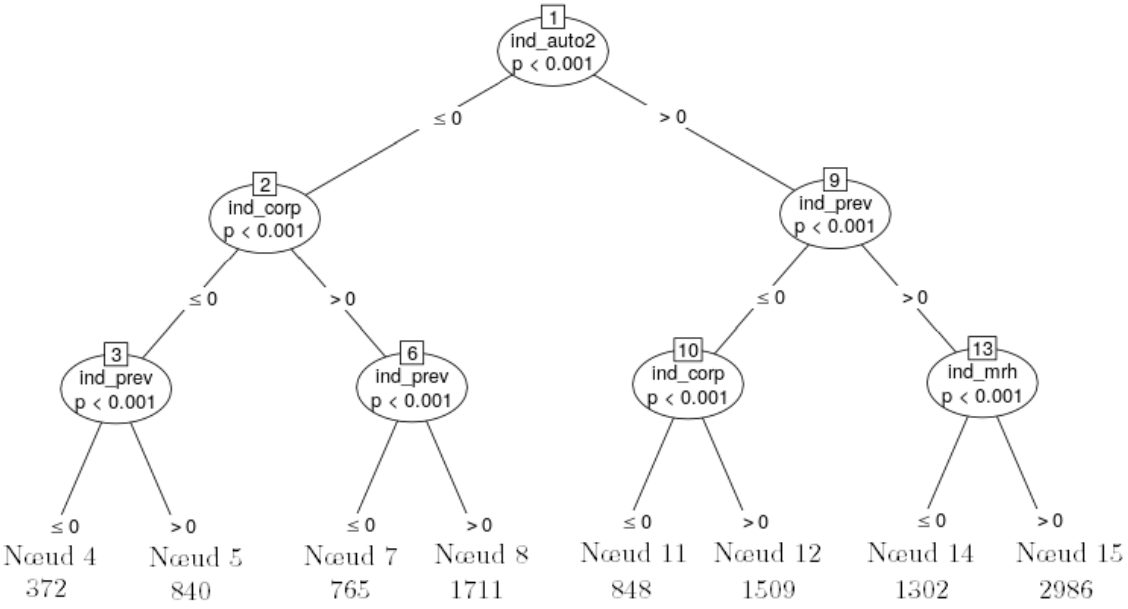


FIGURE 3.4 : Moyenne des nœuds - arbre MOB de profondeur 4

Ainsi, pour un assuré détenant un contrat `AUTO2`, un contrat `PREV` mais pas de contrat `MRH` (et sans information sur les autres contrats), on peut estimer la marge future de cet assuré à 1302. Cette analyse revient donc à considérer l'arbre MOB comme un simple CART.

Cette analyse peut aussi être conduite dans une direction différente. Elle permet de déterminer s'il serait plus profitable pour l'assureur (en termes de marge future) de vendre un type de contrat à un segment d'assurés ou un autre. Prenons l'exemple des nœuds 4 et 7. Dans les 2 cas, les assurés appartenant à ces segments ne possèdent pas de contrat `PREV`, et leur vendre ce contrat augmenterait la marge future réalisée. Mais vendre ce contrat à un assuré du nœud 7 augmenterait sa marge de 946, alors qu'elle n'augmenterait celle du nœud 4 de seulement 468. Ainsi, il sera plus profitable à l'assureur de concentrer ses efforts de vente du contrat `PREV` sur les assurés du nœud 7 plutôt que ceux du nœud 4.

3.2.3 Erreur par segment

L'erreur de prédiction n'est pas uniforme selon les segments (feuilles) du modèle. En effet, dans le modèle MOB, la partie arbre de décision permet de segmenter les observations du modèle selon les variables de partitionnement, mais c'est la partie régression qui fournit les prédictions du modèle, selon les variables de régression. On a donc différents prédicteurs, qui ont par construction des coefficients de régression différents, et donc une erreur de prédiction potentiellement très différente. Dans une optique d'amélioration de la performance du modèle, il peut être intéressant de repérer les segments pour lesquels le modèle se trompe souvent : on pourrait ainsi trouver un compromis entre temps de calcul et performance.

Au lieu d'utiliser un modèle de régression plus avancé (comme un GLM avec interactions) sur toutes les feuilles de l'arbre, améliorant la performance mais augmentant fortement le temps de calcul,

on pourrait imaginer un modèle MOB avec un mélange de modèles de régression en fonction de la performance du GLM sur les différentes feuilles. Les feuilles ayant une erreur faible (relativement à l'erreur globale du modèle) garderaient un GLM comme modèle de régression (et n'augmentent pas le temps de calcul par rapport au premier modèle composé uniquement de GLM), tandis que les feuilles ayant une erreur élevée (relativement à l'erreur globale du modèle) se verraient affecter un modèle de régression plus poussé, augmentant le temps de calcul mais réduisant l'erreur de prédiction globale.

Cette méthode offre un compromis intéressant entre performance et temps de calcul, et n'oblige pas à choisir entre l'un ou l'autre, ce qui serait le cas en appliquant un même modèle de régression à toutes les feuilles.

Le modèle que nous allons considérer dans la suite est encore le MOB avec profondeur 4, correspondant à la Figure 3.3 plus haut. La Figure 3.5 présente les valeurs de MAE selon les différentes feuilles du modèle. On observe, comme expliqué précédemment, de grandes disparités selon les segments.

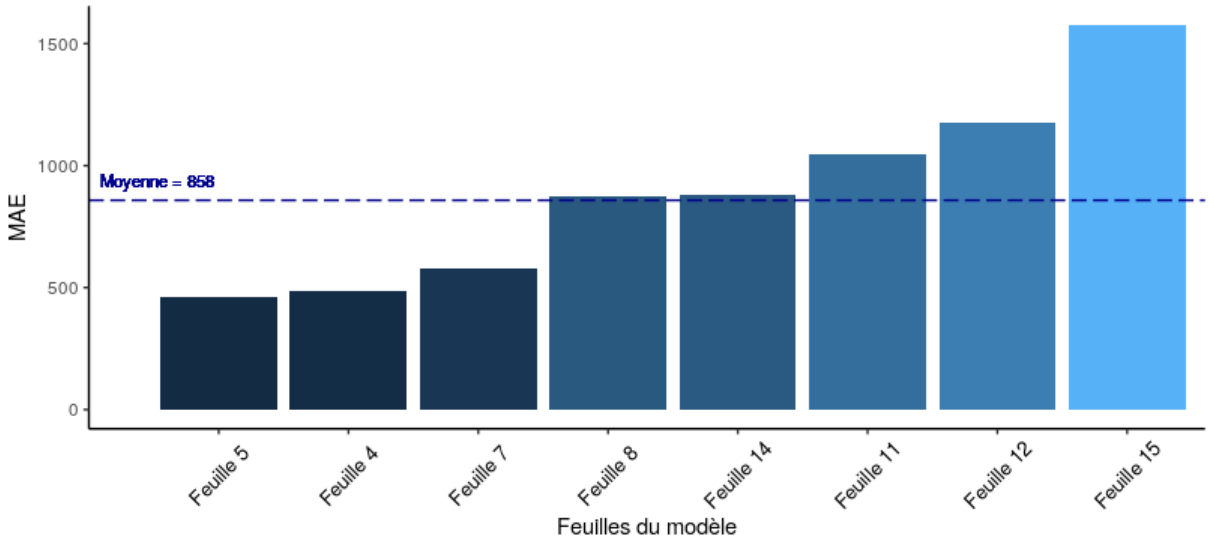


FIGURE 3.5 : MAE selon les feuilles du modèle

À titre d'exemple, pour la feuille 15, correspondant aux assurés possédant au moins 2 contrats `AUTO`, un contrat `PREV` et un contrat `MRH`, la MAE s'élève à 1576, contre 858 pour l'ensemble des observations, soit 84% de plus.

Appliquer la méthode de mélange de modèles de régression présentée précédemment impliquerait de modifier la fonction `glmtree`, ce qui n'est pas l'objet de ce mémoire. En revanche, afin d'avoir une idée du gain potentiel qu'on pourrait obtenir, on a appliqué un GLM avec interactions à la feuille 15, que l'on compare à un GLM sans interactions. La MAE diminue de moins d'un point entre ces 2 modélisations, ce qui infirme notre hypothèse sur la baisse de l'erreur de prédiction.

3.3 Amélioration de la prédiction

On a vu dans la section précédente que les algorithmes MOB ont un pouvoir prédictif plus élevé que les autres modèles "simples" étudiés, comme le résume le Tableau 3.3.

	MSE	MAE
GLM	1 871 561	884
GLM avec interactions	1 826 465	858
LASSO	1 826 618	858
CART	1 902 705	895
CART avec cp = 0.001	1 827 208	858
MOB avec profondeur 4	1 827 421	858

TABLE 3.3 : Comparaison des différents modèles

Néanmoins, pour produire ces modèles, on a seulement utilisé les variables indicatrices indiquant la détention des différents contrats. La base de données fournie par la compagnie est plus riche, car on dispose aussi des variables de caractéristiques des différents individus. On va donc incorporer ces nouvelles variables à la modélisation, en comparant non seulement les métriques utilisées précédemment (MSE, MAE), mais aussi le temps de calcul des différents modèles. Le nombre de variables augmentant fortement, on peut s'attendre à ce que les calculs prennent beaucoup de temps à être effectués. La durée de calcul devient donc un critère important dans le choix de l'algorithme le plus performant.

On modélise de nouveau les différents modèles, afin de comparer des modèles se basant sur les mêmes informations et éviter d'avoir un biais.

GLM

On modélise le GLM avec toutes les variables disponibles (les variables indicatrices ainsi que les caractéristiques des assurés), et ce modèle nous servira de point de comparaison avec les autres. Le modèle est composé de 151 coefficients, car lorsqu'une variable qualitative possède différentes modalités, chacune est considérée comme un coefficient par le GLM, à l'exception d'une modalité par variable. Le modèle est donc peu interprétable si l'on se réfère uniquement à la valeur des coefficients.

Ce modèle s'en sort beaucoup mieux que le précédent GLM, étant donné toute l'information supplémentaire dont il dispose. De la même manière que pour le premier GLM basé uniquement sur l'information de la détention des contrats, on a calculé les résidus de Pearson de ce nouveau modèle, ainsi que leur distribution. Plus de 96% de ces résidus appartiennent à l'intervalle [-2 ;2], le modèle est donc très bien adapté aux données. Ces résultats sont visibles sur les Figures 3.6 et 3.7.

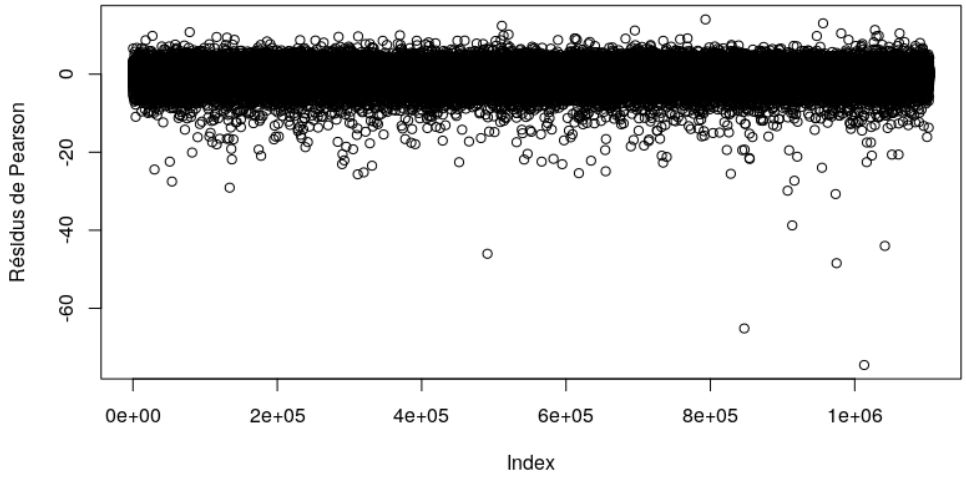


FIGURE 3.6 : Répartition des résidus de Pearson

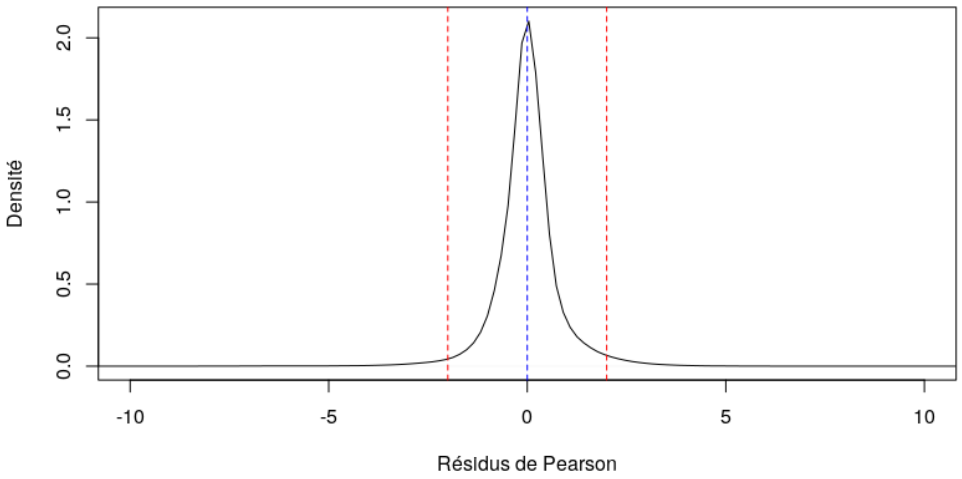


FIGURE 3.7 : Densité des résidus de Pearson

Les coefficients de ce modèle ne sont pas présentés en détail, car cela n'a pas beaucoup d'intérêt, le modèle possédant 151 coefficients. Ce modèle est donc peu interprétable compte tenu du grand nombre de coefficients. En revanche, les métriques obtenues sont présentées dans le Tableau 3.4, avec celles du premier GLM à titre de comparaison. Afin de ne pas confondre avec les modèles précédemment créés, tous ceux qui se baseront sur l'ensemble des données individuelles seront appelés "NomduModèle - Indiv".

	MSE	MAE	Temps
GLM	1 871 561	884	1.8 sec
GLM - Indiv	1 166 117	648	54 sec

TABLE 3.4 : Modèle GLM avec les données individuelles

GLM avec interactions

Le modèle GLM avec interactions entre les variables n'est pas exécuté, car il comprend un nombre de coefficients beaucoup trop importants et demanderait un long temps de calcul, en plus d'être non interprétable. En effet, comme expliqué dans la section 2.2.1 du Chapitre 2, ce modèle serait composé de $\frac{n*(n-1)}{2}+n$ covariables si l'on prend en compte les relations du second ordre, soit $\frac{151*(151-1)}{2}+151 = 11476$ coefficients.

LASSO

Le modèle LASSO est très intéressant avec ces nouvelles informations individuelles, car il fixe certains coefficients à 0, et est donc adaptés pour un grand nombre de covariables. Le modèle fixe la valeur de 71 coefficients à 0, ce qui réduit fortement la complexité du modèle, même si celui reste peu interprétable avec 80 coefficients différents de 0. Pour la même raison que le GLM - Indiv, les coefficients ne sont pas décrits. La MSE et MAE du modèle sont présentées dans la Figure 3.5, avec celles du GLM - Indiv comme point de comparaison.

	MSE	MAE	Temps
GLM - Indiv	1 166 117	648	54 sec
LASSO - Indiv	1 175 718	650	5 min

TABLE 3.5 : Modèle GLM avec les données individuelles

Le modèle LASSO est donc légèrement moins performant que le GLM, et plus long à calculer. Ce long temps de calcul vient du fait que la fonction `glmnet` effectue une cross-validation afin d'optimiser la valeur du paramètre λ utilisée dans le modèle, ce qui rend son exécution plus longue que pour un GLM simple.

Sélection des variables

Afin de réduire le nombre de covariables explicatives utilisées dans la modélisation GLM, une sélection des variables a été réalisée. La *méthode Backward* a été appliquée. Comme expliquée dans la partie 2.1.1, cette méthode de sélection de variables part du modèle complet et retire les covariables les unes après les autres, jusqu'à ce qu'un critère de performance soit minimisé. Dans notre cas, le critère utilisé est le Critère d'Information d'Akaike (AIC), présenté dans la partie 2.1.1.

Cette sélection a conservé 28 variables explicatives, à comparer aux 42 variables disponibles dans le jeu de données. Les variables conservées sont les suivantes :

AUTO1	AUTO2	MRH1	MRH2	CORP/PREV	AUTRE
auto_vehsup	auto2_puissance	mrh1_nbprop	mrh2_nbpieces	ind_prev	anc
auto1_crm	auto2_ancpermis	mrh1_tranchemob	mrh2_patmob	ind_deces	age
auto1_formule	auto2_agecond	mrh1_nbloc	mrh2_naturelri	prev1_detmrh	canal
auto1_puissance	auto2_crm		mrh2_uhsup	prev1_formule	enfant
auto1_agecond				corp_nbveh	couple
auto1_categorie					
auto1_ancpermis					

TABLE 3.6 : Variables sélectionnées par la Méthode Backward

Le modèle linéaire généralisé utilisant ces variables obtient une performance très légèrement inférieure au GLM utilisant toutes les variables (1 point de MAE en moins), ce qui est logique : toutes choses égales par ailleurs, un modèle avec des covariables supplémentaires performera mieux. Ce modèle reste néanmoins plus intéressant, car il utilise bien moins de variables, et est donc plus facilement interprétable.

CART

Comme expliqué au chapitre 2, le modèle CART permet d'obtenir une bonne interprétabilité grâce à son système de branches, compréhensible par tout le monde. Néanmoins, les arbres ne sont pas adaptés à des jeux de données possédant de nombreuses covariables. L'algorithme séparant les observations en fonction d'une seule covariable, il devra nécessairement avoir une très grande profondeur si l'on souhaite utiliser un grand nombre de covariables différentes. En effet, à l'inverse d'un modèle linéaire pour lequel ajouter une variable revient, en terme d'interprétabilité, à ajouter un coefficient, pour le modèle CART, cela revient à ajouter un étage à l'arbre final (dans le cas où la fonction objectif à optimiser est améliorée). Faire cela avec plusieurs variables devient très vite illisible sur le graphique de l'arbre final, comme on peut le voir sur la Figure A.1 en Annexe A. Cette figure représente l'arbre créé avec le paramètre $cp = 0.00001$, qui obtient de très bons résultats, mais qui est totalement non interprétable.

On effectue, selon le même principe qu'au Chapitre 2, différentes modélisations, en faisant varier le paramètre *cp*. Plus sa valeur diminue, plus la profondeur de l'arbre augmente, et donc plus sa performance augmente. En revanche, cela se fait au détriment de son interprétabilité, car l'arbre devient plus profond et peu lisible.

Dans un premier temps, le modèle a été calibré avec le paramètre $cp = 0.01$, conduisant à des métriques assez mauvaises en comparaison des autres modèles calibrés précédemment. Les résultats des métriques des différents modèles CART sont visibles sur le Tableau 3.7. Le modèle a ensuite été paramétré avec $cp = 0.001$, améliorant les métriques de performance, mais doublant le temps de calcul. La Figure 3.8 représente l'arbre final obtenu : celui-ci est déjà peu interprétable visuellement (il est composé de 46 feuilles), et ses performances sont moins bonnes que pour les modèles linéaires.

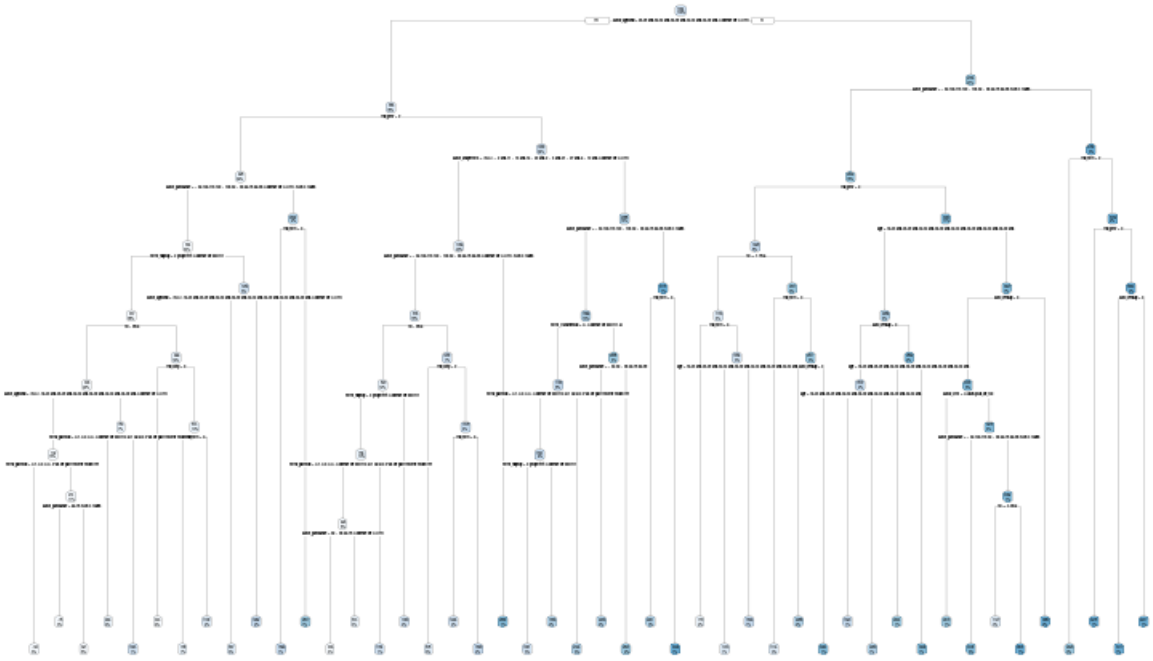


FIGURE 3.8 : Arbre CART - Indiv - cp=0.001

Le modèle a ensuite été relancé avec le paramètre $cp = 0.0001$. Les résultats sont meilleurs que pour les modèles linéaires précédents et les 2 autres modèles CART avec une cp inférieure, mais le modèle devient très complexe, et devient difficilement interprétable. Enfin, plusieurs modèles ont été modélisés afin de trouver le paramètre optimal, qui se trouve être $cp = 0.000008$. En revanche, le modèle n'est plus du tout interprétable (l'arbre correspondant à $cp = 0.00001$ est visible en Annexe A sur la Figure A.1).

	MSE	MAE	Temps
CART - Indiv - cp=0.01	1 498 353	792	2.6 min
CART - Indiv - cp=0.001	1 253 037	689	4.2 min
CART - Indiv - cp=0.0001	1 127 294	634	6 min
CART - Indiv - cp=0.00001	1 085 004	603	17.5 min
CART - Indiv - cp=0.000008	1 088 221	602	9.7 min
CART - Indiv - cp=0.000005	1 101 267	601	15.8 min
CART - Indiv - cp=0.000001	1 204 545	625	26.8 min

TABLE 3.7 : Modèles CART avec les données individuelles

L'inconvénient principal de l'utilisation des modèles CART avec ce jeu de données, composé de nombreuses covariables, est que l'on ne peut pas utiliser toutes ces covariables dans la modélisation (ou alors le modèle serait extrêmement complexe et long à calculer). La valeur prédite dans les feuilles étant la moyenne des observations appartenant à cette feuille, on perd l'information contenue dans les variables non utilisées pour la création de l'arbre. Cela justifie l'utilisation des modèles MOB, car la valeur prédite dans les feuilles de l'arbre sera le résultat d'un modèle linéaire calibré sur les variables non utilisées pour la segmentation des observations.

Modèles MOB

On rappelle que dans le cadre d'une modélisation MOB, on fournit en paramètre à l'algorithme une formule spécifiant les variables utilisables (et utilisées) pour la segmentation de l'échantillon, et celles utilisées pour les régressions au sein des feuilles. La recherche de la meilleure variable pour segmenter l'échantillon est coûteuse en temps de calcul, c'est pourquoi, dans un premier temps, on se limitera aux variables de détention des différents contrats pour les variables de segmentation. Pour les variables de régression, l'ensemble des variables (détention des contrats non utilisées pour la segmentation et variables caractéristiques individuelles) seront utilisées.

Le second paramètre important de la modélisation MOB est la profondeur de l'arbre : elle indique le nombre de coupes que devra effectuer l'algorithme pour créer les différentes feuilles sur lesquelles seront effectuées les régressions. Le temps de calcul est directement lié à ce paramètre : une profondeur de l'arbre égale à 2 impliquera une seule coupe, donc une seule recherche de meilleure variable de segmentation. Une profondeur de 3 impliquera 3 coupes (1 coupe au début, puis 2 coupes à l'étape suivante), et donc 3 recherches de la meilleure variable de segmentation, et ainsi de suite. On se limitera donc à une profondeur faible, ce qui permettra à la fois de ne pas avoir un temps de calcul trop long, et d'obtenir une forte interprétabilité.

Le premier modèle a donc été paramétré avec une profondeur égale à 2 (ce qui revient à n'effectuer qu'une seule coupe et obtenir 2 feuilles), et l'arbre obtenu est représenté dans la Figure 3.9 ci-dessous. Ce modèle n'est pas très intéressant et reste assez simpliste, il ne fait que séparer l'échantillon selon que l'assuré possède un contrat `MRH` ou non.

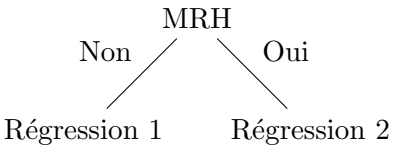


FIGURE 3.9 : Modèles MOB - Indiv avec profondeur 2

On a ensuite augmenté la profondeur d'un niveau (*i.e.* profondeur = 3) afin d'améliorer le pouvoir prédictif du modèle, tout en restant très interprétable (le modèle est visuellement compréhensible par tout le monde). La Figure 3.10 représente l'arbre ainsi obtenu. Les régressions effectuées dans chaque feuille utilisent les variables de détention non utilisées pour la segmentation de la feuille, ainsi que toutes les variables de caractéristiques individuelles cohérentes avec ladite feuille. En effet, pour la Régression 1 de la Figure 3.10 par exemple, les variables relatives aux contrats `MRH`, `AUTO1` et `AUTO2` ne sont pas utilisées, car un assuré appartenant à ce segment ne possède pas le contrat `MRH` ni le contrat `AUTO1` (et par extension le contrat `AUTO2`). Les coefficients de la Régression 1 sont présentés dans le Tableau A.1 en Annexe A.

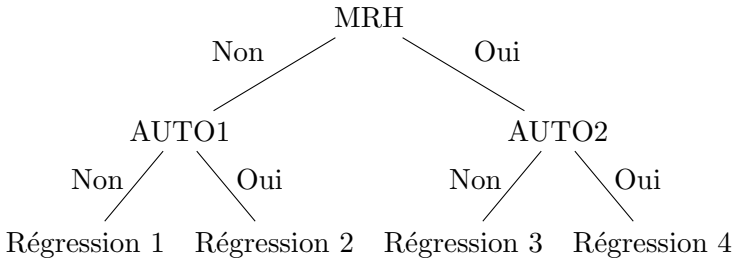


FIGURE 3.10 : Modèles MOB - Indiv avec profondeur 3

On a ensuite lancé l'algorithme avec une profondeur de 4, ce qui ajoute un étage à l'arbre de segmentation. Le modèle reste encore très interprétable, chaque feuille étant le résultat de 3 segmentations en fonction de la détention d'un type de contrat par l'assuré. L'arbre est visible plus bas sur la Figure 3.12. Pour une profondeur de 5, le modèle a de nouveau été lancé, mais les résultats montrent que la profondeur maximale autorisée n'est pas atteinte pour toutes les feuilles. L'arbre est visible en Annexe A sur la Figure A.2. Cela signifie que le modèle estime que segmenter une feuille (la feuille R1 ou R12 par exemple) n'améliore pas significativement le modèle, et qu'il est plus optimal de garder la feuille telle quelle. Cette non-segmentation d'une feuille montre que la complexité du calcul n'est pas le facteur limitant d'une modélisation MOB optimale (si cela avait été le cas, on devrait continuer à incrémenter la profondeur, conduisant à des calculs toujours plus longs), ce qui nous conforte sur la qualité du modèle.

	MSE	MAE	Temps
MOB - Indiv - 2	1 132 879	628	20 min
MOB - Indiv - 3	1 116 216	620	35 min
MOB - Indiv - 4	1 109 851	616	46 min
MOB - Indiv - 5	1 105 757	614	54 min

Le chiffre dans le nom du modèle correspond à la valeur du paramètre de profondeur maximale de l'arbre

TABLE 3.8 : Métriques de performance des modèles MOB avec les données individuelles

Les métriques de performance des différents modèles MOB - Indiv sont représentées dans le Tableau 3.8. Le pouvoir prédictif du modèle diminue bien lorsque l'on augmente sa profondeur maximale, de même que le temps de calcul. La Figure 3.11 représente les différentes améliorations entre ces 4 modèles, ce qui nous aide à faire notre choix de modèle le plus adapté.

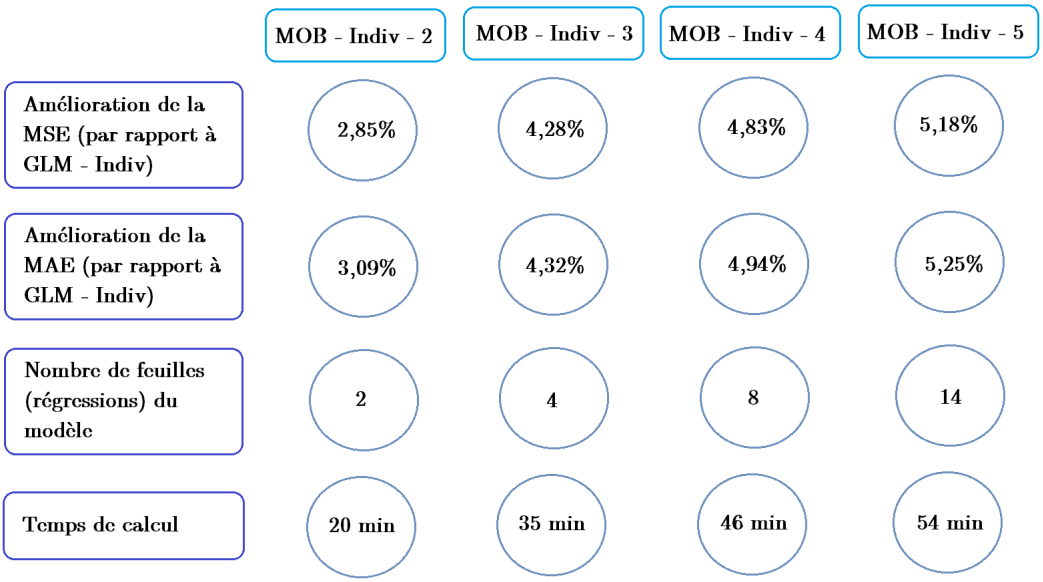


FIGURE 3.11 : Comparaison des différents modèles MOB - Indiv

À la vue de la Figure 3.11, nous choisissons comme modèle le plus adapté le modèle MOB - Indiv - 4. On observe une grande amélioration de la MAE et MSE entre le modèle 2 et le modèle 3, puis une amélioration plus légère pour les modèles 4 et 5. Il en est de même vis-à-vis du temps de calcul : celui-ci double presque entre le modèle 2 et le modèle 3, puis augmente d'environ un tiers lorsque l'on passe du modèle au 3 au 4, et d'un sixième du modèle 4 au 5. Du point de vue de l'interprétabilité, le modèle 5 possède trop de feuilles pour être facilement interprétable, et le modèle 2 en possède trop peu pour être performant. Notre choix se porte donc sur les modèles 3 et 4 : nous retiendrons le modèle 4, car il possède 8 feuilles de régression, ce qui le rend toujours interprétable, et ses performances sont meilleures que le modèle 3. La Figure 3.12 ci-dessous représente l'arbre MOB - Indiv - 4.

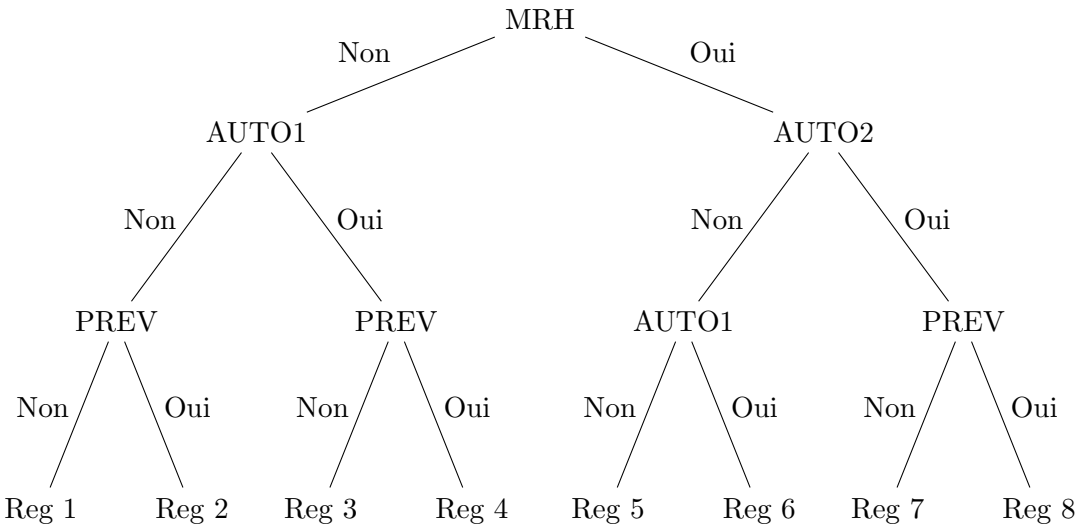


FIGURE 3.12 : Modèles MOB - Indiv avec profondeur 4

Comparaison des différents modèles

Suite à l'ajout des variables individuelles à notre modélisation, le pouvoir prédictif des modèles s'est considérablement amélioré par rapport aux modèles sans l'information sur les caractéristiques individuelles des assurés, comme on pouvait l'imaginer. Les améliorations en termes de MAE sont de 27% pour le modèle GLM, 24% pour le LASSO, 30% pour le CART (à paramètre *cp* identique) et 28% pour le MOB (avec un paramètre de profondeur égal à 5). On obtient donc en moyenne 25% d'amélioration du pouvoir prédictif de nos modèles : l'ajout des variables individuelles a donc un impact significatif sur la qualité de nos modèles.

Le Tableau 3.9 ci-dessous présente les métriques des différents modèles retenus pour chaque classe de modèle, ainsi que leur temps de calcul. Le modèle MOB est donc le plus performant, malgré un temps de calcul bien supérieur aux autres modèles.

	MSE	MAE	Temps
GLM - Indiv	1 166 117	648	54 sec
LASSO - Indiv	1 175 718	650	5 min
CART - Indiv - cp=0.0001	1 127 294	634	6 min
MOB - Indiv - 5	1 105 757	614	54 min

TABLE 3.9 : Métriques de performance des différents modèles avec les données individuelles

On rappelle que le modèle MOB a été modélisé en prenant uniquement pour variables de segmentation les variables de détention des différents contrats, et en utilisant l'ensemble des variables pour les régressions. On va à présent changer les paramètres et inclure d'autres variables disponibles pour la segmentation. Cela ne signifie pas nécessairement qu'elles seront utilisées, la profondeur de l'arbre étant finie. Les modèles peuvent donc rester totalement identiques aux précédents si l'utilisation d'une nouvelle variable pour segmenter l'échantillon n'est pas jugée pertinente par l'algorithme (à l'exception du temps de calcul qui augmentera fortement, car le modèle devrait effectuer des tests d'instabilité sur toutes les variables disponibles pour la segmentation).

Ajout de variables de segmentation

On pourrait techniquement ajouter toutes les variables individuelles à la liste des variables de segmentation. Cependant, cela n'aurait pas trop de sens pour certaines d'entre elles. En effet, notre base de données contient des variables spécifiques à certains contrats (la variable `auto2_energie` par exemple, qui vaut `NA` pour tous les assurés ne possédant pas le contrat `AUTO2`), une segmentation selon ces variables serait donc peu pertinente. En revanche, certaines variables sont communes à tous les assurés, comme l'âge ou l'ancienneté du contrat. On effectue donc de nouvelles modélisations en ajoutant les variables `age`, `anc`, `canal`, `enfant` et `couple` comme potentielles variables de segmentation.

Une première modélisation est effectuée en fixant le paramètre de profondeur maximale de l'arbre à 3. Le modèle a sélectionné la variable `canal` comme première variable de segmentation : cette variable a donc une grande influence sur la marge (la MAE passe de 620 à 614 avec l'ajout de la variable `canal`, soit une amélioration d'environ 1%). Les 2 feuilles ainsi créées (l'une composée des assurés ayant souscrit via le canal interne, l'autre via le canal externe) sont ensuite segmentées selon la variable `ind_mrh`, qui était la première variable de segmentation dans les modélisations MOB - Indiv précédentes. On en conclut que le canal de souscription a une plus grande influence sur la marge que la détention d'un contrat MRH, tandis que l'âge, l'ancienneté, la présence d'enfants ou d'un couple sont

moins impactant que la détention d'un contrat MRH. On augmente ensuite la profondeur maximale de l'arbre, dans le but de déterminer si ces nouvelles variables de segmentation ont un impact sur la marge supérieur à la détention des autres contrats.

Avec une profondeur maximale égale à 4, l'arbre obtient une meilleure performance (présentée dans le Tableau 3.10) que le modèle *MOB - Indiv - 4*. Cela vient de l'utilisation de la variable `canal` comme variable de segmentation, qui a une forte influence sur la marge. Il en est de même pour le modèle avec une profondeur maximale égale à 5 : la MAE passe de 614 à 601, soit une amélioration d'un peu plus de 2%. La variable `canal` a donc un grand impact sur la marge future.

	MSE	MAE	Temps
MOB - Indiv - 4	1 109 851	616	46 min
MOB - Indiv - 3 - canal	1 077 173	614	36 min
MOB - Indiv - 4 - canal	1 058 625	604	50 min
MOB - Indiv - 5 - canal	1 052 689	601	75 min
MOB - Indiv - 6 - canal	1 049 209	599	85 min

TABLE 3.10 : Métriques de performance des modèles MOB

Le Tableau 3.11 présente le nombre de nœuds terminaux (feuilles) des différents modèles MOB (avec ajout des variables individuelles communes). On observe que les arbres créés ne sont pas forcément symétriques, ce qui indique que l'algorithme choisi ses coupes avec parcimonie, sans complexifier l'arbre final lorsque cela n'améliore pas significativement la performance. Le nombre de feuille maximal théorique vaut 2^{n-1} pour un arbre de profondeur n.

	Feuilles	Feuilles maximales théoriques
MOB - Indiv - 3 - canal	4 feuilles	4 feuilles
MOB - Indiv - 4 - canal	8 feuilles	8 feuilles
MOB - Indiv - 5 - canal	15 feuilles	16 feuilles
MOB - Indiv - 6 - canal	23 feuilles	32 feuilles
MOB - Indiv - 7 - canal	29 feuilles	64 feuilles

TABLE 3.11 : Nombre de nœuds terminaux des modèles MOB

Ainsi, l'arbre **MOB - Indiv 5 - canal** n'est pas un arbre binomial symétrique, comme on peut le voir sur la Figure 3.13 ci-dessous.

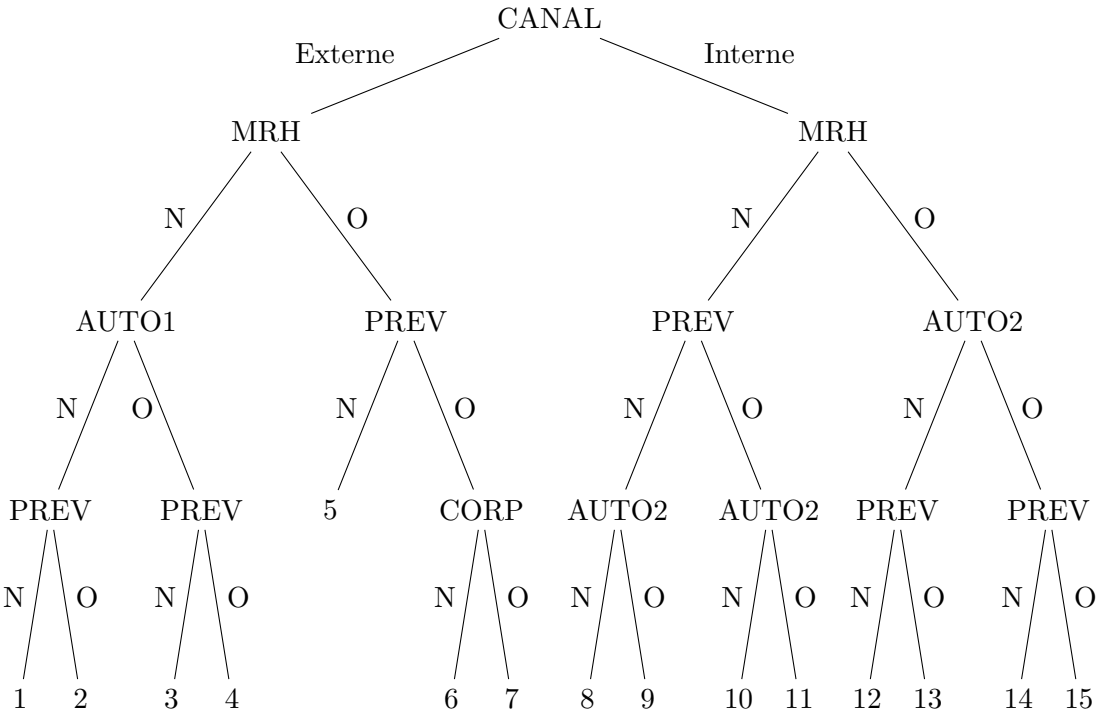


FIGURE 3.13 : Exemple d'arbre non symétrique

Sélection des variables

Afin de réduire le nombre de variables explicatives au sein de chaque feuille, la modélisation MOB a été lancée de nouveau en utilisant les variables sélectionnées par la *Méthode Backward* (présentées dans le Tableau 3.6) comme variables de régression. Les performances des modèles diminuent peu par rapport aux modèles MOB utilisant toutes les variables comme variables de régression (1 point de MAE en moyenne). Pour les variables de partitionnement, on a utilisé les indicatrices de détention des types de contrat, ainsi que les variables individuelles communes aux assurés (`age`, `anc`, `canal`, `couple` et `enfant`).

Ces modèles deviennent donc bien plus lisibles et interprétables que les précédents, car les feuilles sont composées d'un nombre réduit de coefficients. En effet, le nombre de variables de régression passe de 42 à 28. Celles-ci pouvant posséder entre 2 et 12 modalités, on réduit fortement le nombre de coefficients de régression au sein des feuilles.

De même que pour les précédentes modélisations MOB, il faut aussi prendre en compte que certains coefficients prennent la valeur `NA` en fonction des feuilles. En effet, par exemple, si une feuille se trouve sur une branche de l'arbre ayant été coupée selon que l'assuré possède ou non un contrat `MRH`, les coefficients de régression des modalités de la variable `mrh1_nbprop` prendront tous la valeur `NA`, puisque les assurés appartenant à cette feuille ne possèdent pas de contrat `MRH`. On obtiendra donc au final des feuilles composées différemment, car certains coefficients de régression ne seront pas utilisés. Ainsi, pour une baisse de performance très légère, on obtient des modèles bien moins complexes, et plus facilement interprétables.

Utilisation de la loi Gamma

La régression linéaire avec une loi Gamma nécessitant que les valeurs cibles soient de même signe, on ne peut pas appliquer la méthode utilisée au paragraphe 2.2.3 du Chapitre 2, qui consistait à créer un modèle Bernoulli-Gamma en fonction du signe de la marge.

En effet, l'arbre de partitionnement obtenu sera potentiellement différent en fonction de l'échantillon sélectionné. L'arbre étant créé en fonction des caractéristiques des individus base d'apprentissage, les variables utilisées pour les coupes successives ne sont pas nécessairement identiques, car les échantillons sont différents selon que la marge est positive ou négative.

On ne peut donc pas simplement créer de modèle Bernoulli-Gamma et effectuer les régressions dans les feuilles, puisque les structures des 2 arbres ne sont pas identiques. A titre d'exemple, les Figures 3.14 et 3.15 représentent les 2 arbres MOB créés avec une profondeur maximale de 4, le premier sur une base d'apprentissage avec marge positive, le second sur une base d'apprentissage avec marge négative.

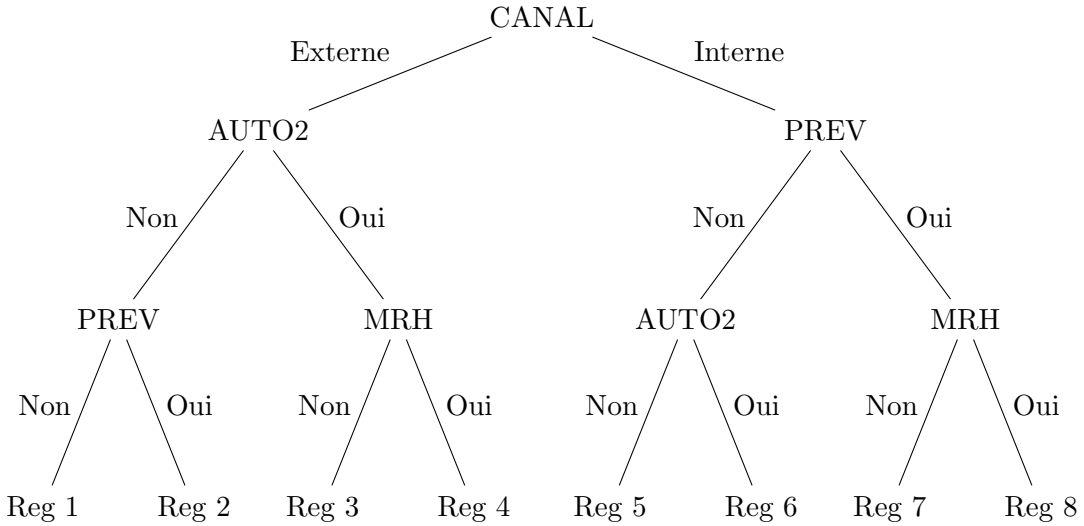


FIGURE 3.14 : Modèle MOB sur marges négatives

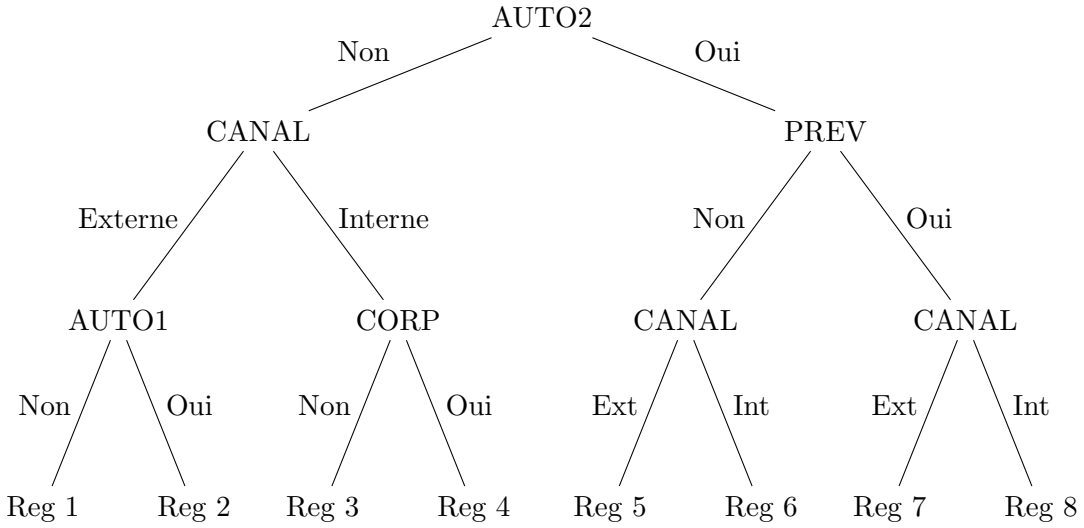


FIGURE 3.15 : Modèle MOB sur marges positives

On observe bien dans cet exemple que l'on ne peut pas utiliser un modèle Bernoulli-Gamma dans les régressions, les structures des arbres étant différentes.

En revanche, cette approche permet de faire ressortir les déterminants de la marge en fonction de son signe. En créant deux arbres de partitionnement, l'algorithme MOB segmente les échantillons selon différentes variables, ce que les modélisations précédentes ne nous permettaient pas de faire, étant effectuées sur l'ensemble du jeu de données.

On observe donc que la variable `canal` est la plus importante pour les assurés dont la marge future estimée est négative, alors qu'il s'agit de la variable `ind_auto2` pour ceux dont la marge future estimée est positive.

	MSE	MAE	Temps
GLM - Indiv	1 166 117	648	54 sec
LASSO - Indiv	1 175 718	650	5 min
CART - Indiv - cp=0.00001	1 088 004	603	17.5 min
CART - Indiv - cp=0.000005	1 101 267	601	15.8 min
MOB - Indiv - 5	1 105 757	614	54 min
MOB - Indiv - 5 - canal	1 052 689	601	75 min
MOB - Indiv - 6 - canal	1 049 209	599	85 min

TABLE 3.12 : Métriques de performance des différents modèles étudiés avec ajout des caractéristiques individuelles

Le Tableau 3.12 synthétise les métriques des différents modèles étudiés dans la partie 3.3 (le tableau synthétisant les modèles étudiés avant l'ajout des caractéristiques individuelles se trouve au début de la partie 3.3). Les modèles MOB semblent être les meilleurs modèles pour répondre à la

problématique de ce mémoire, ils obtiennent de meilleures performances que les différentes régressions linéaires étudiées, ainsi que les arbres CART. Ils sont par ailleurs assez facilement interprétable, à l'inverse des modèles "boîtes noires" évoqués au chapitre 1, très performant mais peu interprétables. Le choix de la profondeur maximale de l'arbre permet de satisfaire les différents besoins des utilisateurs : on augmentera la profondeur si une meilleure performance est recherchée, tandis qu'on la diminuera si l'on souhaite obtenir une meilleur interprétabilité des résultats.

Conclusion

Dans cette étude, on a cherché à reconstituer la marge future d'un portefeuille d'assurés, afin de la rendre plus interprétable. En se basant sur les valeurs de marge future individuelles calculées par le modèle de l'assureur sur les 31 prochaines années, on a construit différents modèles statistiques permettant de reconstituer cette marge et d'en faire ressortir les déterminants. Cette analyse apporte une meilleure interprétabilité des résultats, permettant différentes applications métier (quels segments d'assurés privilégier pour la souscription d'un nouveau contrat par exemple).

Dans un premier temps, on a modélisé la marge en prenant uniquement en compte l'information sur la détention des différents contrats (`AUTO`, `MRH`, `PREV`, `CORP`, `EPARGNE` et `DECES`) par les assurés. On a modélisé la marge future à l'aide d'un modèle linéaire généralisé (GLM), qui nous a servi de modèle étalon pour les autres modélisations. Ce modèle est très facilement interprétable, car il ne possède que 8 coefficients et permet donc une lecture facilitée des déterminants de la marge, mais son pouvoir prédictif est assez limité. On a ensuite amélioré ce modèle linéaire généralisé en ajoutant les effets d'interaction entre les variables de détention des contrats : cette information supplémentaire entraîne une hausse de performance par rapport au GLM de 2,9%. Néanmoins, ce modèle possède 29 variables et est donc moins interprétable. On a utilisé une modélisation LASSO afin de réduire le nombre de coefficients, diminuant ceux-ci à 22, tout en conservant la même performance. Enfin, on a séparé l'échantillon en fonction du signe de la marge future afin d'utiliser la loi Gamma dans un modèle composé Bernoulli-Gamma, ce qui diminue la performance et limite le périmètre d'utilisation de cette méthode dans un but prédictif, car on ne connaît pas *a priori* le signe de la marge future d'un assuré.

Ces modèles sont simples à mettre en place, et offrent une bonne compréhension du portefeuille et des déterminants de la marge. Grâce aux coefficients du GLM, facilement interprétables, on observe que la détention des contrats `AUTO2`, `MRH` et `PREV` impacte fortement à la hausse la marge future, tandis que la détention d'un contrat `DECES` impacte à la baisse la marge future. Ainsi, l'assureur peut utiliser cette information pour déterminer les contrats qui lui sont le plus profitable. La même lecture peut être faite grâce à la modélisation LASSO, donnant les combinaisons de 2 contrats les plus favorables (ou défavorables) en termes de marge future pour l'assureur.

Afin d'améliorer le pouvoir prédictif des modèles, on a ensuite modélisé la marge future à l'aide d'arbres CART, en changeant les paramètres de profondeur. Le choix de ce paramètre implique de faire le bon compromis entre performance et interprétabilité du modèle, car un arbre trop profond serait illisible, tandis qu'un arbre trop petit serait peu performant. L'arbre possédant le meilleur compromis obtient une performance (MAE) 2,3% meilleure que le GLM, et est facilement interprétable.

Les arbres CART offrent un nouvel axe de lecture à l'assureur : ceux-ci étant très visuels, on peut aisément voir l'impact de la souscription d'un nouveau contrat pour un segment d'assurés donné. Par ailleurs, ils sont facilement explicables à une personne sans formation mathématique, qui comprendra les résultats du modèle. Cette modélisation est aussi intéressante d'un point de vue métier, pour un

agent général par exemple, qui saura quel contrat proposer en priorité à ses clients afin d'augmenter la marge future.

Enfin, on a complété cette modélisation à l'aide de modèles segmentés : les modèles MOB. Ces modèles, combinant une arborescence et des modèles linéaires généralisés dans les feuilles de l'arbre, obtiennent de meilleures performances que les modèles précédemment utilisés, tout en restant très interprétable. Après avoir fait varier le paramètre de profondeur de l'arbre, qui influe sur le nombre de régressions à effectuer et donc sur l'interprétabilité du modèle, on a choisi le modèle avec une profondeur de 4, qui améliore la MAE de 3% par rapport au GLM.

Dans un second temps, on a rajouté à notre modélisation les variables individuelles des assurés, augmentant fortement l'information dont disposent nos modèles pour reconstituer la marge. Les performances sont donc naturellement améliorées (de l'ordre de 25% en moyenne). Cependant, cet ajout des caractéristiques individuelles complique l'interprétabilité des résultats. Les arbres sont trop profonds, devenant illisibles, et les modèles linéaires possèdent trop de coefficients pour être facilement compréhensibles. Afin de réduire le nombre de coefficients de régression, une sélection des variables a été effectuée, entraînant une faible diminution de performance pour un gain significatif d'interprétabilité. Néanmoins, il reste de nombreux coefficients, principalement dû au fait que les variables explicatives contiennent plusieurs modalités. Les modèles MOB montrent donc tout leur intérêt dans le cas de l'ajout des variables individuelles, en segmentant puis en effectuant des régressions sur les feuilles de l'arbre créé. Ils obtiennent de meilleurs résultats que les modèles linéaires ou les arbres, tout en restant interprétables.

Bien qu'étant le meilleur modèle dans le cadre de cette étude, le modèle MOB reste un peu plus long à paramétrer que les autres modèles étudiés. Dans le cadre du portefeuille d'assurés utilisé dans cette étude, ce problème n'est pas un facteur limitant, car la base de données est mise à jour annuellement. Il faut néanmoins garder en tête que le temps de calcul augmente aussi en fonction du nombre de variables de segmentation. Si la structure de la base de données venait à être modifiée via l'ajout de nouvelles variables individuelles communes à tous les assurés, et que l'assureur souhaitait reconstruire un modèle en intégrant ces variables, le temps de calcul serait considérablement plus élevé qu'une simple mise à jour annuelle des données.

Cette étude s'est focalisée sur la reconstitution de la marge future des assurés au travers de trois types de modèles (modèles linéaires, arbres de décision et MOB) facilement interprétables. Afin d'aller plus loin et d'améliorer les résultats, on pourrait envisager d'utiliser des modèles "boîte noire", plus performants. Il faudra néanmoins les coupler avec des outils d'interprétabilité agnostiques (les *Partial Dependence Plot* par exemple) afin de ne pas perdre la dimension interprétable des résultats au profit d'une meilleure performance. Une autre piste d'amélioration serait d'implémenter un autre type de régression linéaire dans les feuilles du modèle MOB, comme des *Modèles Additifs Généralisés (GAM)* (Hastie et Tibshirani, 1990) par exemple. Cela permettrait de garder l'aspect interprétable du modèle en segmentant la population selon certains critères, et d'améliorer le pouvoir prédictif du modèle.

Bibliographie

Boutahar, E. (2021). Application à la tarification automobile de méthodes de partitionnement récursif de modèles linéaires généralisés. *ISFA*.

Charpentier, A. (2011). La loi des grands nombres et le théorème central limite comme base de l'assurabilité. Risques 86.

Charpentier, A. (2015). Segmentation et Mutualisation, les deux faces d'une même pièce. Risques 103.

Clément, M. (2022). Utilisation d'arbres de régression pour la prédiction de coûts automobiles. *Université Paris Dauphine*.

Ehrhardt, A. (2018). scoring: Credit Scoring tools. url : https://CRAN.R-project.org/package=scoring.

FranceAssureur (2020). L'assurance française. Données clés 2020. url : https://www.franceassureurs.fr/wp-content/uploads/VF-Donnees-cles-2020.pdf (visité le 21/07/2022).

Hastie, T. et Tibshirani, R. (1990). Generalized Additive Models. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.

Hjort, N. L. et Koning, A. (2002). Tests For Constancy Of Model Parameters Over Time. *Journal of Nonparametric Statistics* 14.1-2, p. 113-132. eprint : https://doi.org/10.1080/10485250211394.

Hothorn, T. et Zeileis, A. (2015). partykit: A Modular Toolkit for Recursive Partytioning in R. *Journal of Machine Learning Research* 16, p. 3905-3909.

InstitutdesActuaires (2019). Synthèse des travaux du groupe de travail "transparence des algorithmes".

James, G., Witten, D., Hastie, T. et Tibshirani, R. (2014). An Introduction to Statistical Learning: with Applications in R. Springer Texts in Statistics. Springer New York.

Molnar, C. (2022). Interpretable Machine Learning. A Guide for Making Black Box Models Explainable. 2e éd.

R Core Team (2021). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. url : https://www.R-project.org/.

RStudio Team (2020). RStudio: Integrated Development Environment for R. RStudio, PBC. Boston, MA. url : http://www.rstudio.com/.

Sorochynskyi, O. (2020). Analyse des déterminants de la marge dans des contrats IARD. *UFR de Mathématique et Informatique de Strasbourg*.

Statista (2018). Montant des cotisations de l'assurance non vie dans le monde en 2018, selon le pays. url : https://fr.statista.com/statistiques/515477/montant-cotisations-assurance-non-vie-selon-pays-monde/ (visité le 04/07/2022).

Wright, M. N. et Ziegler, A. (2017). ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. *Journal of Statistical Software* 77.1, p. 1-17.

Zeileis, A. et Hornik, K. (2007). Generalized M-fluctuation tests for parameter instability. *Statistica Neerlandica* 61.4, p. 488-508. eprint : https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9574.2007.00371.x.

Zeileis, A., Hothorn, T. et Hornik, K. (2008). Model-Based Recursive Partitioning. *Journal of Computational and Graphical Statistics* 17.2, p. 492-514.

Annexe A

Compléments visuels sur les modèles peu intepétables

La Figure A.1 représente un arbre trop profond, qui n'est pas interprétable bien que plus performant :

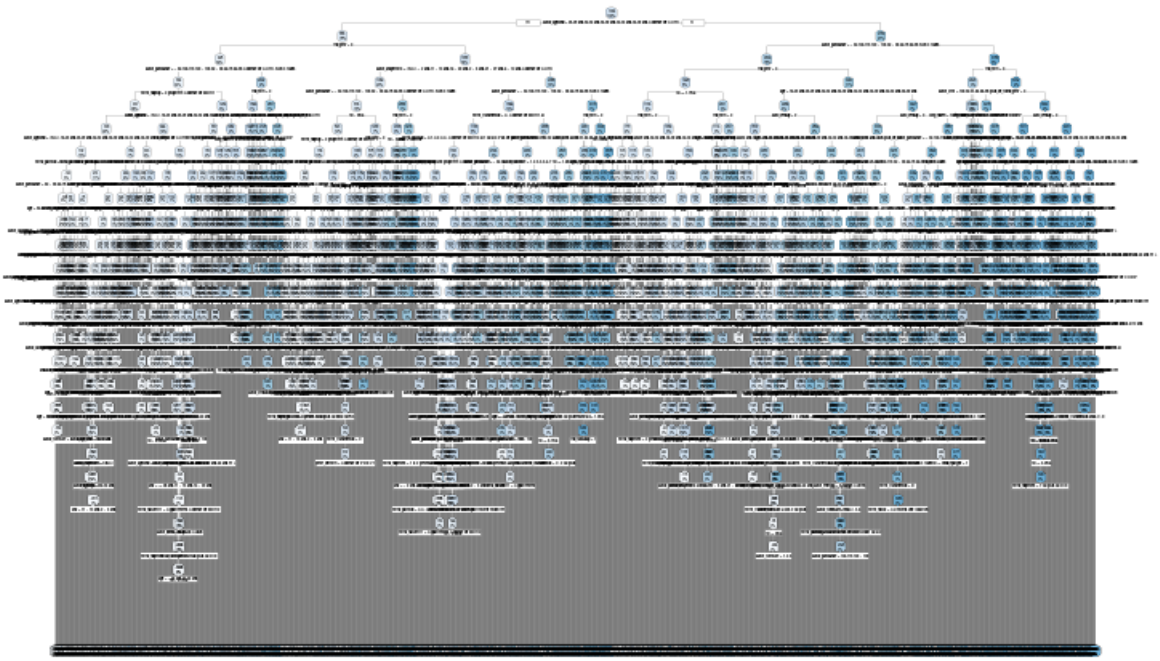


Figure A.1 : Arbre CART illisible

Les valeurs des coefficients de la Régression 1 de la Figure 3.10 sont présentés dans le Tableau A.1 ci-dessous :

	Estimate
(Intercept)	141.853
ind_corp	344.938
ind_prev	441.461
ind_epargne	−3.193
ind_deces	10.865
anc ≥ 28 ans	93.303
anc 11-15 ans	42.109
anc 16-20 ans	63.371
anc 2-5 ans	25.695
anc 21-27 ans	81.553
anc 6-10 ans	26.158
age 18-24 ans	−9.462
age 25-29 ans	54.531
age 30-34 ans	71.703
age 35-39 ans	125.074
age 40-44 ans	193.257

	Estimate
age 45-49 ans	251.518
age 50-54 ans	279.370
age 55-59 ans	316.114
age 60-64 ans	319.781
canalinterne	−140.454
corp_facteuraggr 2 roues > 125	41.044
corp_facteuraggr Pas de 2 roues	−45.739
corp_nbveh1 véhicule	−58.406
corp_nbveh2 véhicules	21.104
corp_nbveh3 véhicules ou plus	78.056
prev1_detmrh MRH1	−5.347
prev1_detmrh MRH2 option	−1.066
prev1_detmrh Non MRH1	−9.224
enfantTRUE	−73.062
coupleTRUE	−121.370
prev1_formule2	62.344

TABLE A.1 : Coefficients de Régression 1

L'arbre A.2 ci-dessous représente le modèle MOB - Indiv avec profondeur 5 :

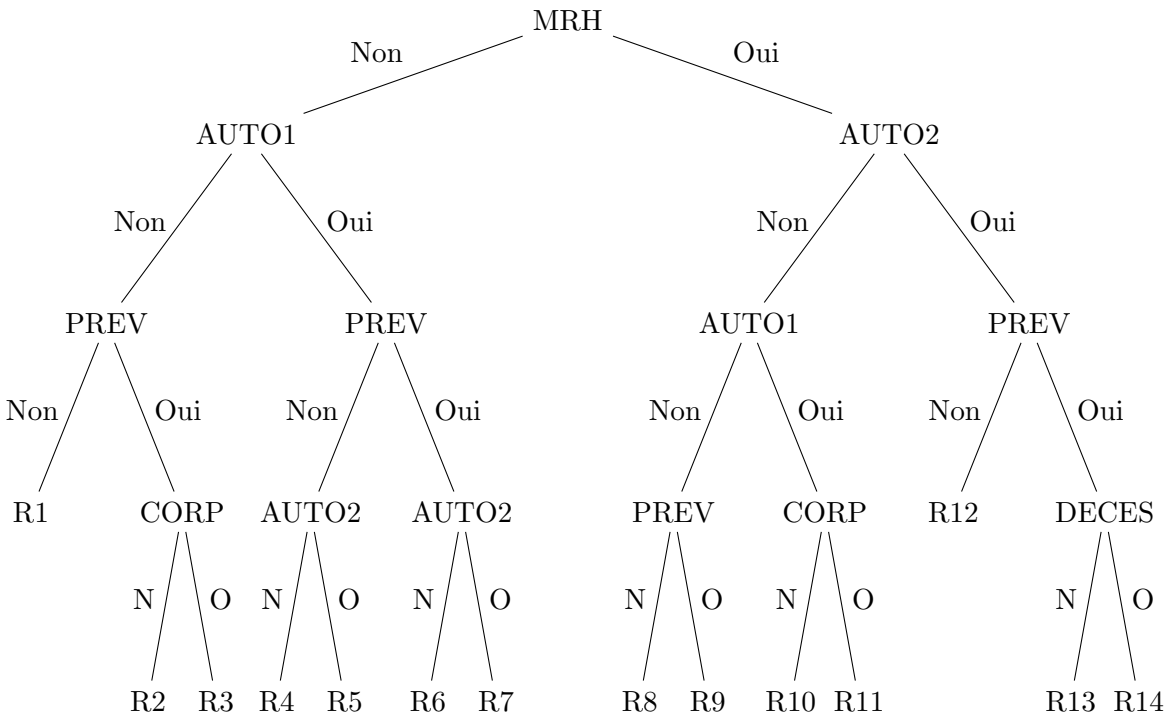


FIGURE A.2 : Modèles MOB - Indiv avec profondeur 5