IRM | institut du risk management BY Institut.

Mémoire présenté pour la validation de la Formation
« Certificat d'Expertise Actuarielle »
de l'Institut du Risk Management
et l'admission à l'Institut des actuaires
le 28 Mars 2022

Par : Guillaume BOULLET

Titre : Création d'une valeur au niveau foyer en assurance

Confidentialité : ☐ NON ☒ OUI (Durée : ☐ 1an ☒ 2 ans)
Les signataires s'engagent à respecter la confidentialité indiquée ci-dessus

Membres présents du jury de l'Institut des actuaires :

_____

_____

_____

Membres présents du jury de l'Institut du Risk Management :

_____

_____

_____

_____

_____

_____

_____

_____

Entreprise :

Nom : THELEM ASSURANCES

Signature et Cachet :

thélem assurances
Entreprise régie par le Code des Assurances
SOCIÉTÉ D'ASSURANCE MUTUELLE
à cotisations variables
Siège Social : LE [illegible]
B.P. 63130 - 45431 CHE[illegible]
DIRECTEUR

Directeur de mémoire en entreprise :

Nom : PICHARD Céline

Signature :

Invité :

Nom : _____

Signature :

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

Signature du responsable entreprise

Signature(s) du candidat(s)

IRM | institut du risk management BY Institut.

28/03/2022

Création d'une valeur au niveau foyer en assurance

BOULLET Guillaume
THELEM ASSURANCES

Table des matières

Introduction

« L'assurance repose fondamentalement sur l'idée que la mutualisation des risques entre des assurés est possible. Cette mutualisation, qui peut être vue comme une relecture actuarielle de la loi des grands nombres, n'a de sens qu'au sein d'une population de risques homogènes » [Charpentier, 2011].

Cela conditionne les assureurs à créer des classes d'assurés avec des profils et des risques différents entre les groupes et homogènes dans les groupes d'où la segmentation utilisée chez tous les assureurs.

Cependant, les nouvelles technologies comme le Big Data et les objets connectés qui donnent accès à un volume de données impressionnant permettent à certains assureurs d'évoquer l'idée d'un tarif individuel, remettant en cause l'idée fondamentale de mutualisation des risques.

De plus, dans le contexte concurrentiel d'aujourd'hui renforcé par la loi Hamon et une meilleure connaissance des produits grâce notamment aux comparateurs, les assurés sont plus ouverts à fournir des données en échange d'une amélioration de leurs couvertures, de leurs tarifs, et/ou de services proposés.

Les enjeux de la tarification prennent tout leur sens en essayant de trouver une segmentation maximale tout en imposant une solidarité minimale et les travaux de Charpentier, Denuit et Elie illustrent parfaitement ces enjeux en montrant (assez simplement) que l'assureur ne segmentant pas va se trouver en réelle difficulté car il n'aura récupéré que les « mauvais » risques en les sous tarifant mais que l'assureur qui a segmenté parfaitement sera certes à l'équilibre financier mais sur une niche de population beaucoup plus petite et s'exposera à un autre type de risque à savoir la volatilité de son portefeuille.

Ceci est le postulat de base de l'assurance et est également la première chose que l'on apprend dans les tarifs des assurances quand on évoque la segmentation. Un exemple basique est la tarification automobile selon l'âge. Prenons les éléments suivants comme acquis :

- Coût du risque :
    - Moins de 30 ans : 125
    - Plus de 30 ans : 75
- Tarif Thelem (pas de différenciation en fonction du sexe) :
    - Moins de 30 ans : 100
    - Plus de 30 ans : 100
- Tarif Concurrence :
    - Moins de 30 ans : 115
    - Plus de 30 ans : 85

Dans cet exemple simple, on imagine les clients « jeunes » être attiré par Thelem et les clients plus âgés attirés par la concurrence ce qui aurait pour conséquence une perte de marge pour Thelem qui n'aurait en portefeuille que les mauvais risques. D'où l'importance d'avoir une « bonne » segmentation.

Les assureurs se livrent donc une « guerre » afin d'avoir les meilleurs critères de segmentation pour leur tarification.

Toutes ces questions se sont multipliées et ont subi une accélération avec l'apparition du Big Data dans le monde de l'assurance qui est passé par les 3V, les 4V jusqu'aux 5V que nous connaissons aujourd'hui et qui sont les principes et les caractéristiques clés du Big Data essentiels à maîtriser :

- Le Volume qui fait référence aux énormes quantités de data générées à chaque instant. Ces volumes sont devenus tellement massif que les quantifications se font maintenant en Zettaoctets.
- La Variété désignant la multiplicité des types de données disponibles. Auparavant, les data étaient essentiellement des données structurées. Aujourd'hui, de nombreuses data non structurées comme les images ou les données textuelles sont générées à chaque seconde.
- La Vitesse (ou Vélocité) qui correspond à la rapidité à laquelle les data sont générées et circulent.
- La Véracité qui désigne la fiabilité de la data est un élément indispensable et essentiel pour pouvoir en tirer profit.
- La Valeur représente le fait que chaque donnée doit apporter une valeur ajoutée à l'entreprise.
- Un 6ème V émerge également, la Vertu faisant référence aux réglementations en matière de confidentialité et de conformité des datas telles que le RGPD.

Le big data ouvre donc la porte à une quasi-infinité de données. Reste à savoir comment la capter, lui donner sens et l'utiliser. Les données externes ne seront pas utilisées dans ce mémoire mais cela pourrait être envisagé pour ajouter de l'information au foyer notamment les données géographiques à la maille Insee ou à la maille IRIS qui complèteraient la vision sur le cadre de vie des foyers.

Attention tout de même aux données utilisées car comment parler de cette quantité phénoménale de données sans parler du règlement général sur la protection des données (RGPD). En effet, mis en place en 2018, cela oblige les acteurs de l'assurance à être en conformité avec ce règlement notamment sur le fait que les données traitées dans le cadre des contrats d'assurance doivent être pertinentes et nécessaires pour répondre à un objectif précis.

Les données liées à la tarification sont spécifiques à chaque contrat. Les données liées au véhicule pour les contrats automobiles ou les données liées au logement pour les contrats

habitation par exemple. Certains critères peuvent ensuite être communs à différents produits comme le code postal pour le lieu de garage du véhicule et le lieu de l'habitation mais cela reste lié au bien.

Le client est ensuite inséré dans l'équation en observant son équipement ou encore son statut d'occupation et son logement quand on tarifie un contrat automobile.

L'objectif de ce mémoire est de créer un nouveau critère qu'on nommera « Valeur Actuarielle » et qui aura pour but de prendre en compte des données à un niveau supérieur qui sera au niveau du foyer. Ce critère prendra en compte entre autres la rentabilité du foyer, son ancienneté, son nombre d'affaires nouvelles ou encore l'évolution temporelle de certains de ses statuts.

Par ailleurs, la volatilité des contrats ne cesse de croitre depuis la mise en place de la loi Chatel et plus récemment avec la loi Hamon. De plus, les comparateurs sur Internet facilite grandement la tâche des assurés à la recherche de tarifs attractifs au dépend, parfois, d'un niveau de couverture optimale pour leurs risques. Une question qui peut se poser ici est de savoir si l'on préfère engranger des profils fortement bénéficiaires sur des courtes durées car avec une sensibilité au tarif élevée ou est-il plus intéressant de conserver des profils moins volatiles plus longtemps avec une rentabilité plus faible. Les profils que l'on conserverait plus longtemps pourrait à long terme s'équiper plus largement et conseiller Thelem à leur entourage.

L'objectif est donc clair : « Comment construire une valeur actuarielle au niveau du foyer et quelles utilités lui donner ? »

Nous étudierons dans un premier temps comment a été construite la valeur actuarielle et comment l'interpréter afin d'être en mesure de communiquer dessus. Dans un second temps, nous essayerons d'intégrer cette variable au modèle de tarification automobile pour quantifier son impact.

Par la suite, une prédiction sera effectuée pour essayer d'anticiper cette valeur actuarielle au moment de la souscription du contrat.

I. Valeur actuarielle

1. Construction de la base de données

i. Le périmètre et les données

Le but de cette partie est d'expliciter la construction de la base de données, et le périmètre observé.

Au vu de l'objectif premier de la valeur actuarielle qui est de l'utiliser dans le tarif automobile, nous avons fait le choix de sélectionner uniquement les clients « particulier » avec un contrat automobile et/ou habitation actif au 31/12/2016.

A partir de cela, nous observons le foyer du client à l'instant T. Nous collectons tous les critères tarifaires du contrat automobile tel que (liste non exhaustive) :

- Age du conducteur / Ancienneté du permis
- Date de mise en circulation du véhicule
- Groupe SRA du véhicule représentant sa puissance
- Classe SRA du véhicule reflétant sa tranche de prix en valeur à neuf
- Usage du véhicule

De la même manière, nous récupérons également les données liées au lieu d'habitation. Celles si seront plus complètes si le foyer possède un contrat habitation chez Thélem, sachant que sur le dernier produit auto, le type de logement et la qualité d'occupation sont des questions posées au client et que nous aurons forcément cette information.

Le fait d'aller récupérer le contrat habitation du foyer complètera cette information avec différents critères tarifaires tel que :

- Type de logement
- Nombre de pièces
- Surface de dépendances
- Capital mobilier assuré

Le périmètre choisi est de 36 mois à l'image de la sinistralité antérieure utilisée dans les critères tarifaires de la majorité des produits d'assurance. On récupère donc les mêmes données que précédemment au 31/12/2019. Cela nous permettra, si les données sont présentes c'est-à-dire si le foyer à des contrats automobile et habitation d'actifs, d'observer l'évolution du foyer. Cela permettra de voir si un changement de véhicule (véhicule plus récent, plus cher…) ou un changement de logement (passage de locataire à propriétaire, d'appartement à maison, évolution du nombre de pièces) est survenu entre les deux périodes observées.

Une fois les informations récupérées à ces deux dates d'états (31/12/2016 et 31/12/2019), nous allons nous intéresser à la vie du foyer durant les années 2017,2018 et 2019.

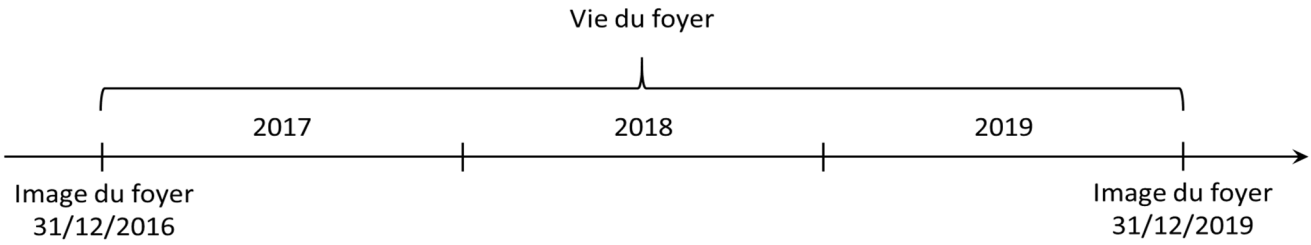
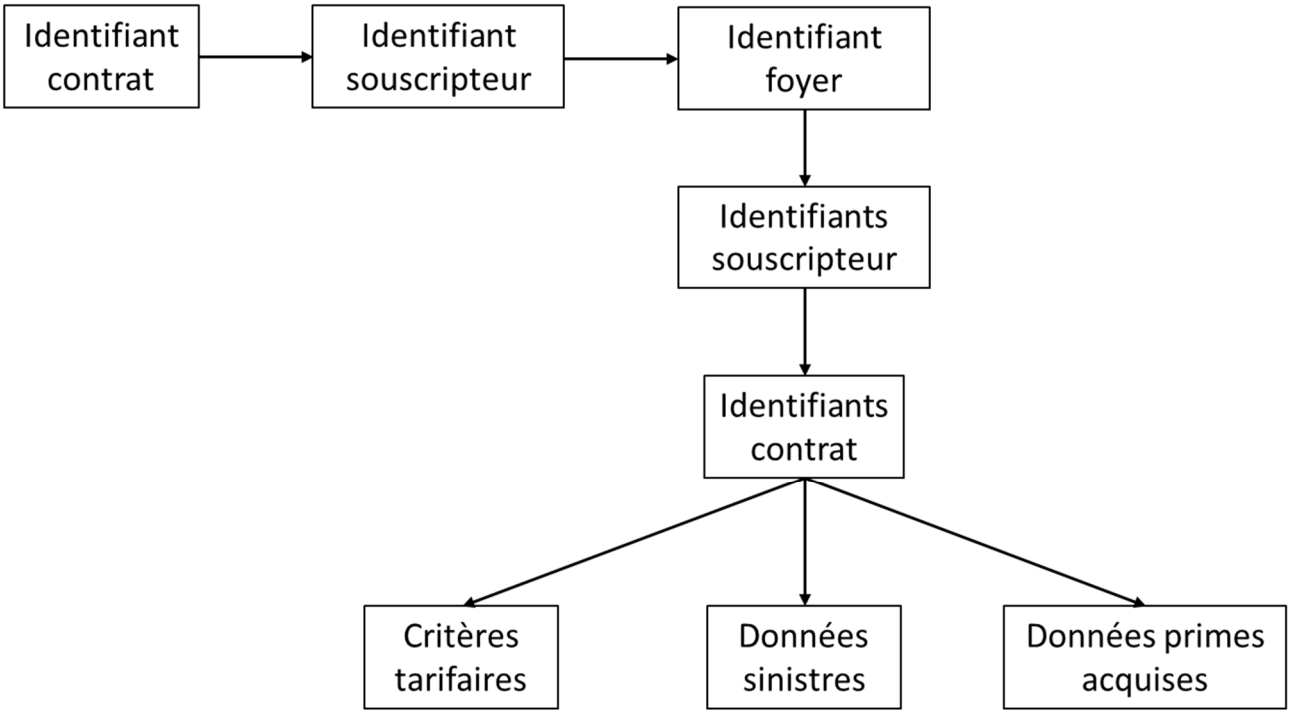


Figure 1 : Périmètre d'observation du foyer

Comme bon nombre de *DataWare* existants, les données ne sont pas regroupées en une seule table et la construction des bases de données, que ce soit pour cette étude, les études ponctuelles ou les modélisations tarifaires peut s'avérer fastidieuse et chronophage.

Pour cette étude, voici schématiquement ce qu'il a fallu réaliser.



En partant de la base contrat qui nous sert à établir notre périmètre, nous obtenons l'identifiant contrat. Cet identifiant nous sera utile pour récupérer l'identifiant souscripteur qui lui-même nous servira à obtenir l'identifiant foyer.

De cet identifiant, nous obtenons les identifiants souscripteurs rattachés. En découlent les identifiants contrats qui nous permettent d'obtenir, en fonction des différentes dates d'états souhaitées les informations nécessaires à la construction de notre base.

Concernant la vie du foyer, nous observerons uniquement les contrats « particuliers » (exclusion des contrats professionnels comme la garantie décennale, la multi risque professionnel …).

Durant cette période de 3 ans, nous prendrons en compte les données suivantes :

- Nombre d'affaires nouvelles au global
- Nombre d'affaires nouvelles par année civile
- Nombre de résiliations au global
- Nombre de résiliations par année civile
- Nombre d'avenants administratifs et techniques
- Primes acquises et charge sinistre de tous les contrats actifs durant la période observée avec actualisation

Toutes ces données nous seront utiles au calcul de la rentabilité expliqué dans le paragraphe suivant et à la création de variables potentiellement utilisables dans le processus de calcul de la valeur actuarielle.

ii. Retraitement des données brutes

Une fois la base de données construite, il est nécessaire de vérifier la qualité des données, d'épurer la base des données ayant servi à en construire d'autres mais aussi de retravailler certaines variables en effectuant des regroupements par exemple.

La partie précédente étant principalement une étape d'extraction, elle a été réalisée intégralement sous SAS Enterprise Guide. La suite va se dérouler sur Python qui fût pour moi une première expérience avec ce langage de programmation.

Dans les différents retraitements, on peut retrouver une harmonisation des différentes générations de produits. En effet avec la création de nouveaux produits, les formules et la gamme du produit évoluent avec les nouveaux besoins des clients mais aussi avec le contexte qui peux évoluer. Nous avons donc des générations de produits qui cohabitent en portefeuille et une harmonisation est nécessaire pour comparer les bons éléments entre eux. A titre d'exemple, on pense bien évidemment aux formules ou encore aux franchises qui, au lieu d'être exprimées en montant (qui peut dépendre de la classe SRA en auto) seront exprimées en niveau. Le tableau suivant répertorie pour l'exemple, l'harmonisation des formules entre les différentes générations de produits. Cette harmonisation des formules est une préconisation établie au moment de la sortie de la nouvelle offre pour faciliter le réseau de distribution à effectuer des migrations et à prendre ses marques entre les différentes générations de produits.

Produit MRH	Produit Néologis 2
Formule	Formule
Budget	Tonic
	Déclic si appart < 3pp
Harmonie 1	Dynamic
Harmonie 1 + Rà9	Confort
Harmonie 2	Tonic
Harmonie 3	Dynamic
Harmonie 3 + Rà9	Confort
Leader	Confort ou Zen
Logement étudiant	Déclic étudiant
Mobil Home	Mobil-Home
PNO	PNO

Produit MHA (Néologis)		Produit Néologis 2
Formule	Options	Formule
F1		Tonic
F1	DE	Dynamic
F1	OV	
F1	DE+OV	
F1 appt < 3 pp		Déclic
F2		Confort
F2	Honoraires expert	Zen
F2	Plein air + Canalisations	Zen
F3		Déclic étudiant
F5		Mobil-Home
F4		PNO

Tableau 1 : Exemple d'harmonisation sur les formules habitation entre les différentes générations de produits

Des variables catégorielles sont retraitées afin de créer des catégories personnalisées et plus facilement exploitables dans les modélisations ou dans l'interprétabilité des résultats de classifications. Prenons exemple sur le code qualité. Entre différentes générations de produits, la modalité de la variable désignant un propriétaire occupant a pu être modifiée (propriétaire occupant total sur Néologis, propriétaire occupant sur Néologis2).Dans la même génération de produits, la modalité peut être fine pour correspondre parfaitement au risque souscrit quand techniquement la distinction n'est pas nécessaire (propriétaire occupant total et propriétaire occupant partiel).

Cela nous amène avec un code qualité contenant 20 catégories. Un nombre de modalités trop importantes quand on sait que certaines modalités peuvent être regroupées sans aucun perte d'information. Cet exemple est complet car en plus de grouper les modalités semblables, les qualités peuvent se regrouper pour nous donner une information sur le statut d'occupation (Occupant / Non Occupant) et sur le type d'occupation (Propriétaire / Locataire).

Catégories brutes	Catégories retraitées	Type d'Occupation	Statut d'Occupation
Propriétaire occupant total	Propriétaire Occupant	Propriétaire	Occupant
Propriétaire occupant partiel			
Copropriétaire occupant partiel			
Copropriétaire occupant unique			
Propriétaire Occupant			
Copropriétaire Occupant			
Compte commun usufruitier / nu pptaire Occupant			
Usufruitier	Locataire Occupant	Locataire	
Colocataire occupant partiel			
Colocataire occupant unique			
Locataire Occupant			
Usufruitier Occupant			
Locataire occupant partiel			
Locataire occupant unique			
Propriétaire non occupant	Propriétaire Non Occupant	Propriétaire	Non Occupant
Copropriétaire non occupant			
Nu Propriétaire Non Occupant			
Nu-propriétaire			
Locataire non occupant	Locataire Non Occupant	Locataire	
Usufruitier Non Occupant			

Tableau 2 : Exemple de regroupement pour la variable "Qulaité d'occupation"

Pour terminer, certaines variables continues sont transformées en variable catégorielle afin de segmenter au mieux le portefeuille et de potentiellement répondre aux problèmes de volumétrie. Les regroupements seront faits en suivant la répartition de la variable, notamment sur le nombre de pièces élevé qui peuvent être regroupé à partir de 10 pièces ou la surface des dépendances.

iii. Statistiques sur la base de données

La base comporte 310905 lignes soit autant de foyers et 96 variables.

Dans un premier temps des statistiques univariées ont été réalisés sur toutes les variables pour décrire la composition de la base de données. Cela permet de faire des premiers constats sur les foyers étudiés.

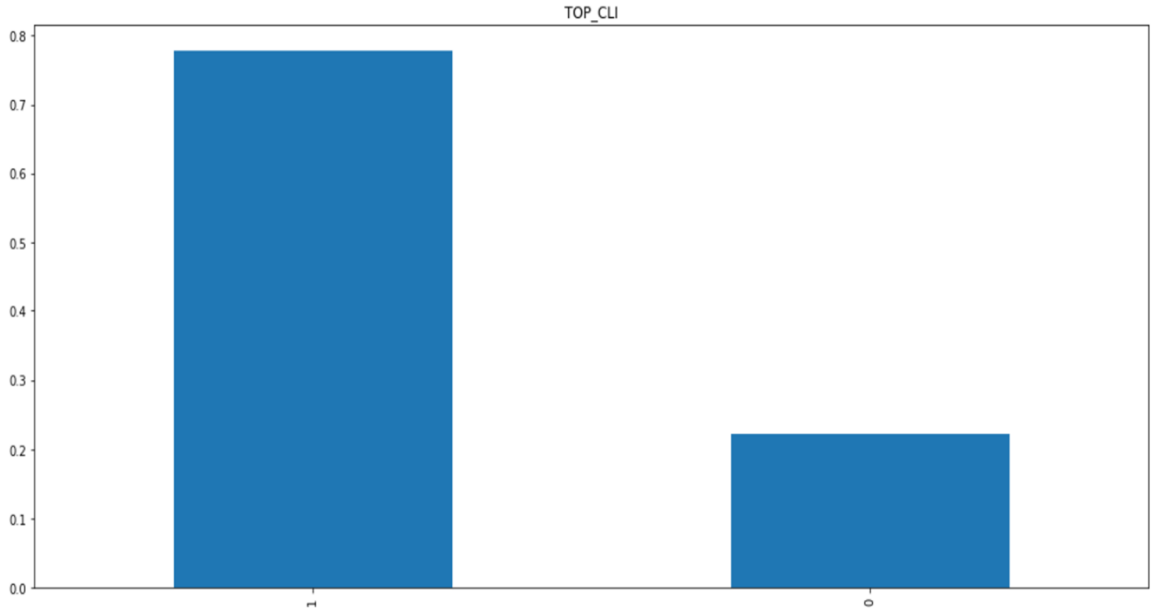


Figure 2 : Foyer en portefeuille à la fin de la période d'observation

Sur la totalité des foyers observés, un peu plus de 20% ne sont plus clients Thélem à la fin de la période d'observations sachant que les nouveaux foyers apparus durant cette période d'observation n'ont pas été intégré.

De même, les graphiques ci-dessous représentant le nombre d'affaires nouvelles et le nombre de résiliations montre que durant la période de 3 ans observée, 40% des foyers ont souscrits entre 1 et 3 contrats et 58% des foyers ont résiliés entre 1 et 3 contrats. Les valeurs extrêmes comme une vingtaine de résiliations ou une vingtaine d'affaires nouvelles correspondent à des flottes automobiles.

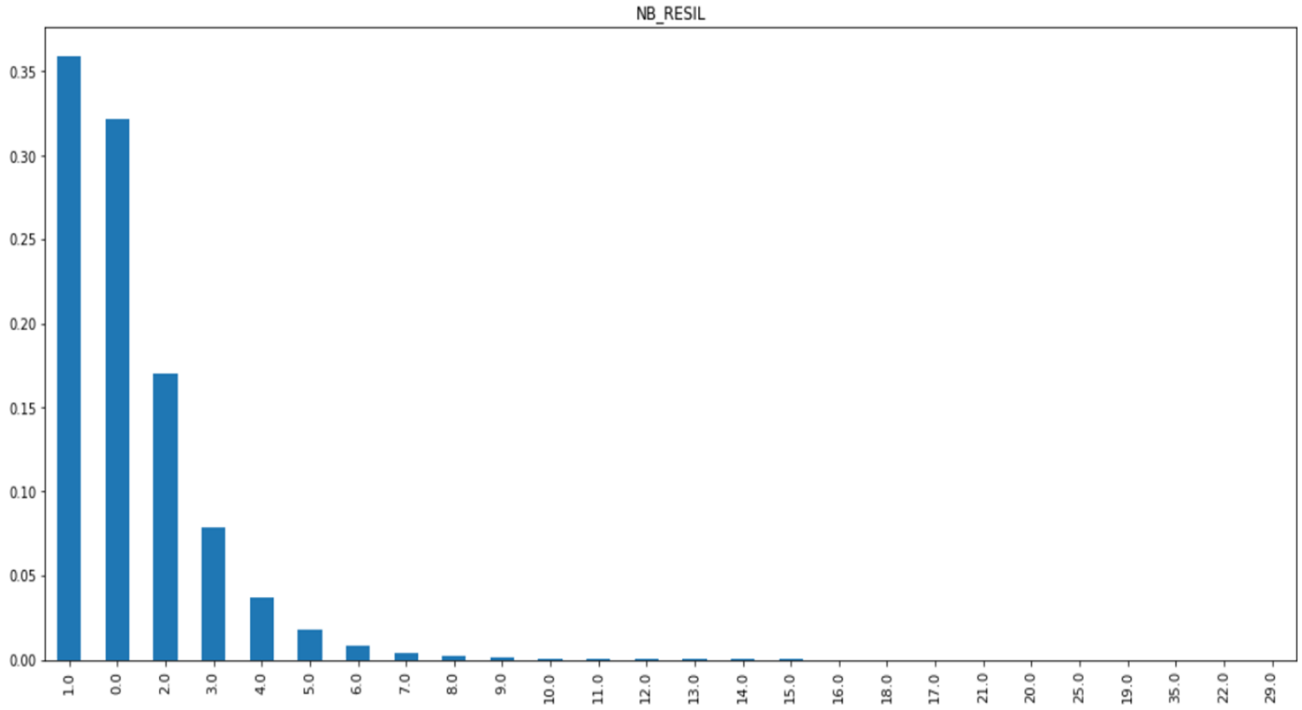


Figure 3 : Nombre de résiliations par foyer sur la période d'observation

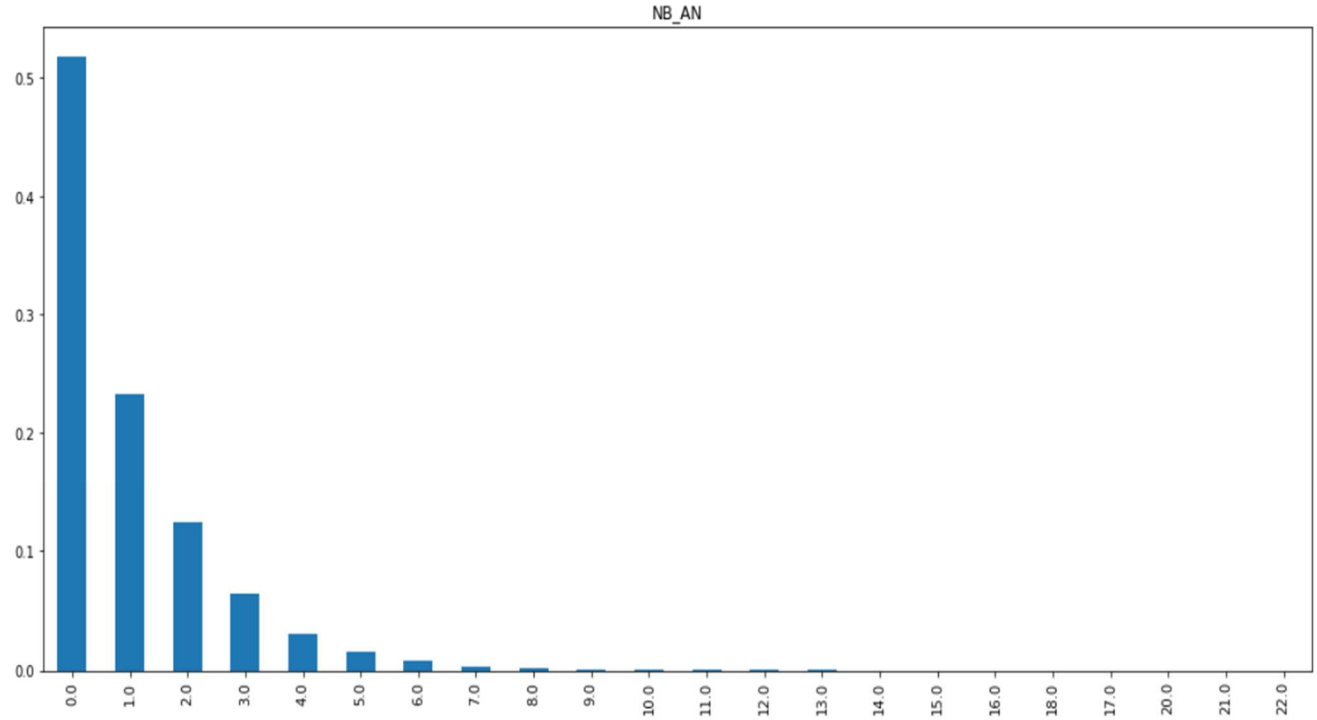


Figure 4 : Nombre d'affaires nouvelles par foyer sur la période d'observation

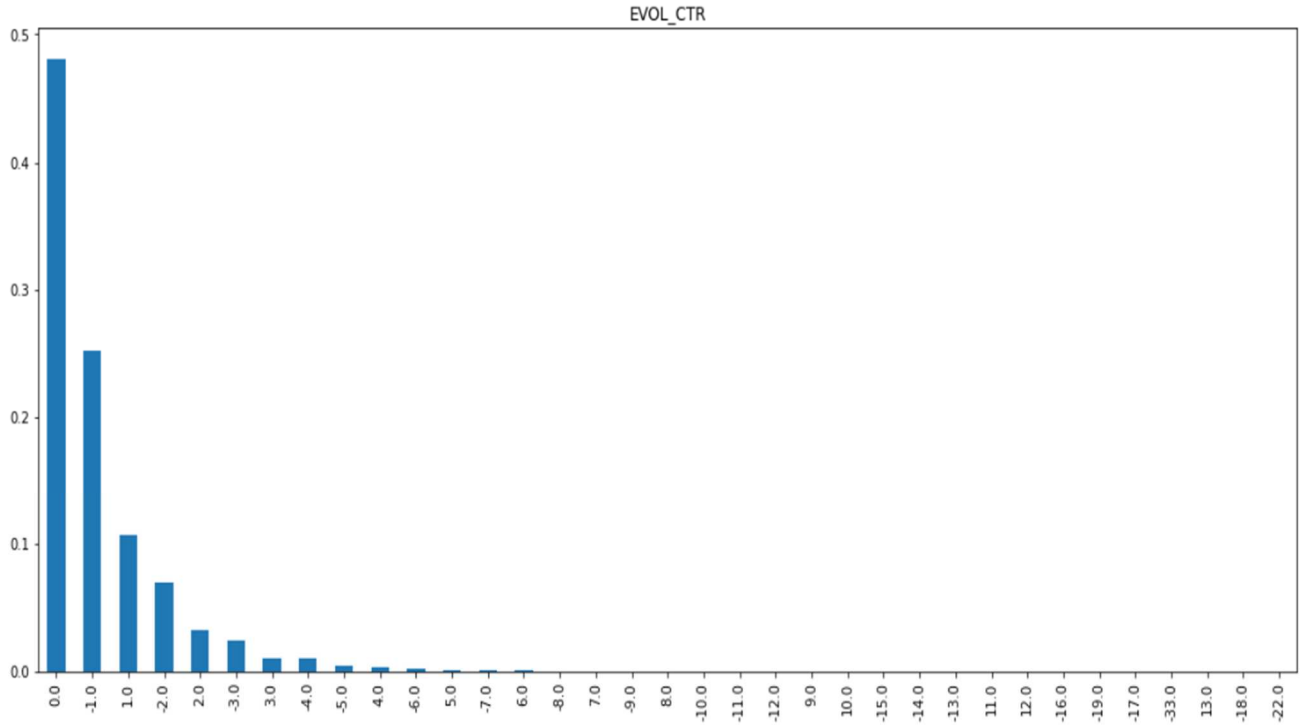


Figure 5 : Evolution du nombre de contrats par foyer

Pour finir, en observant l'évolution des contrats au sein du foyer, 25% des foyers ont « perdu » un contrat. Parmi ces 25%, environ 14% étaient des foyers monocontrats (un seul contrat chez Thélem) et font partie des foyers qui ne sont donc plus clients. La valeur « 0 » dans ce graphique signifie une compensation entre les AN et les résiliations sur la période ou une absence de mouvement.

➢ Etude de la corrélation entre les variables

Le graphique ci-dessous représente la corrélation entre les variables à l'aide du Rho de Spearman.

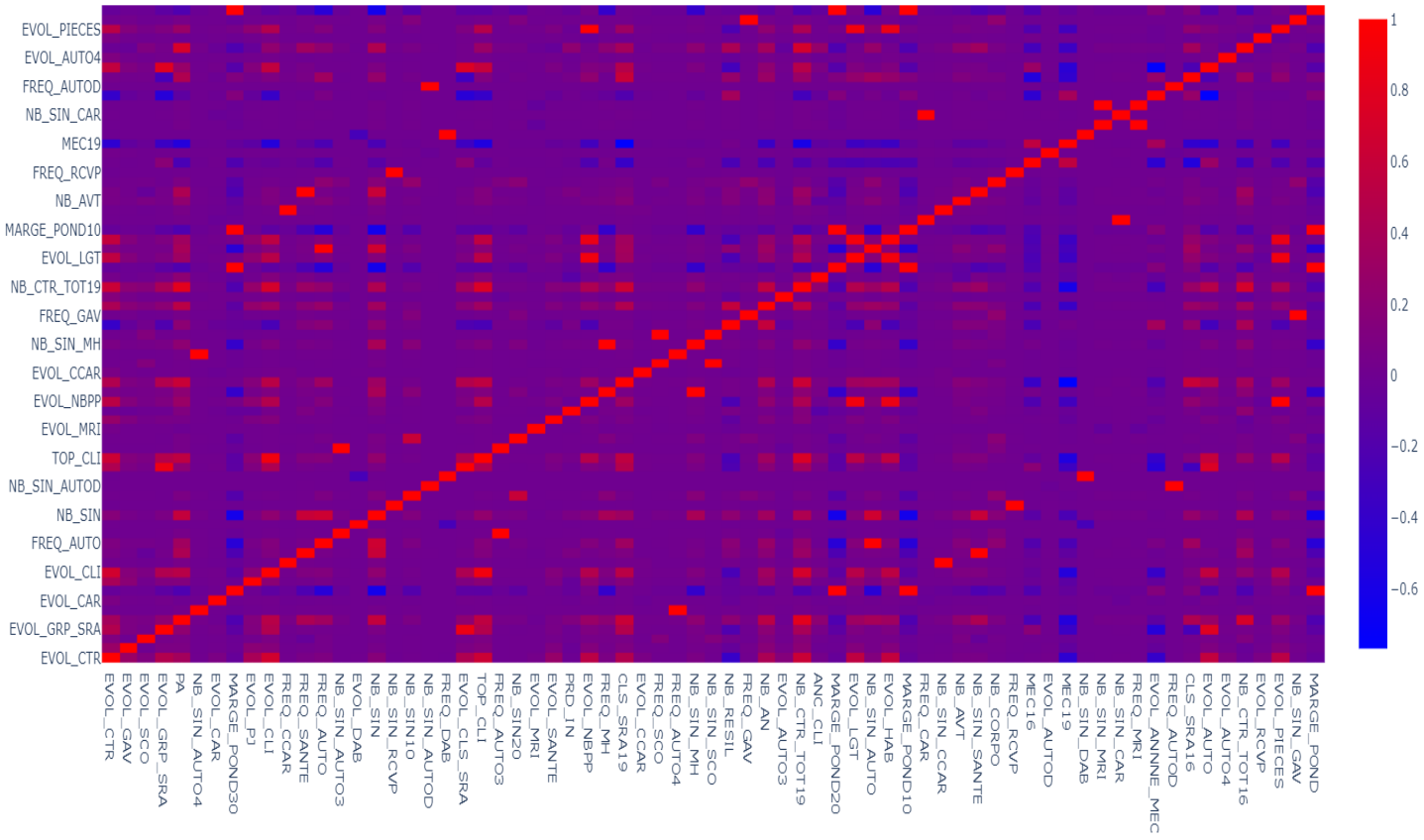


Figure 6 : Matrice de corrélation de Spearman

Comme attendu et pressenti, certaines variables se retrouvent fortement corrélés mais l'inverse aurait été étonnant et aurait plutôt apporté une preuve d'une mauvaise construction des variables. Par exemple, on retrouve une corrélation entre les différentes marges pondérées selon le seuil de grave choisi. De même, on retrouve une forte corrélation entre le nombre de sinistres et la fréquence des sinistres.

La matrice de corrélation nous permet de faire un premier choix sur les variables à conserver. En effet, entre deux variables fortement corrélées, que ce soit positivement ou négativement, on orientera notre choix sur la variable la plus pertinente, ou celle avec la meilleure interprétabilité. Par exemple, nous observons une corrélation entre le nombre de sinistres et la fréquence. Notre choix de variable à conserver se portera sur la fréquence. Raisonner avec la fréquence est plus équitable entre les foyers car cette donnée tient compte de la période d'exposition au risque.

Une corrélation est également observée entre le nombre de contrats détenus par le foyer vu à fin 2016 et vu à fin 2019. Selon l'évolution des foyers, cette variable peut être identique aux

deux différentes dates de vues, même si des mouvements ont eu lieu entre ces périodes et ceci s'expliquent par un solde nul sur la période d'observation.

Logiquement, nous avons une corrélation négative entre le nombre de sinistres et la marge pondérée. Un nombre de sinistre plus important, selon le coût, va diminuer la marge voir la rendre négative.

Pour finir, la corrélation entre la présence ou non du client à la fin de la période d'observation avec les variables d'évolution des critères automobile ou habitation. Cette corrélation s'explique par le fait que si le foyer ne fait plus partie des clients de Thelem, on perd l'information d'une potentielle évolution de gamme dans la vie du contrat. Un foyer toujours présent à la fin de la période d'observation apporte donc une plus grande probabilité sur l'évolution des caractéristiques du contrat automobile et/ou habitation.

- Stats sur la marge pondérée et traitements des valeurs aberrantes

Voici un premier aperçu brut de la marge pondérée avec une boîte à moustache

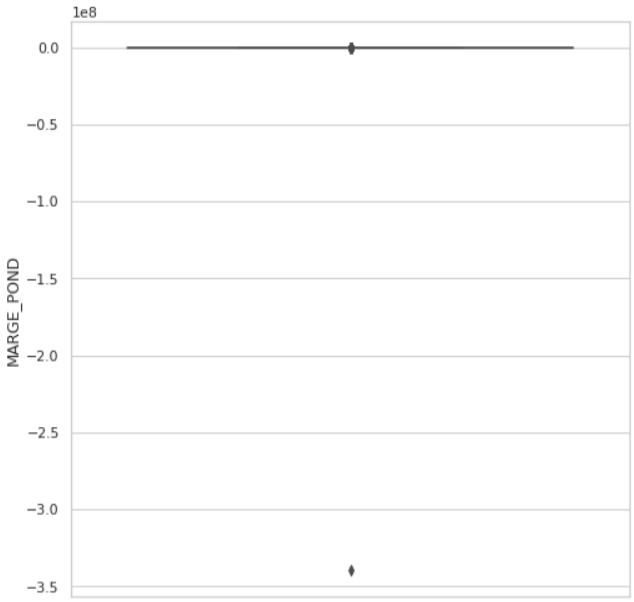


Figure 7 : Boîte à moustache brute de la variable "marge pondérée"

A première vue, des valeurs extrêmes « polluent » la base de données. Ces données sont plus communément appelées *outliers* ou valeurs aberrantes. Elles sont anormalement différentes de la distribution de la variable observée. Il est important de détecter les valeurs aberrantes puisque plusieurs algorithmes de Machine Learning sont sensibles aux données d'entrainements ainsi qu'à leur distribution.

Pour les détecter, nous utiliserons la boite à moustache qui se basent sur les valeurs de la médiane, ainsi que des quartiles inférieur et supérieur.

Avec cette méthode, on considère que toutes les valeurs inférieures ou supérieures à 1.5 fois l'écart interquartile sont des valeurs aberrantes.

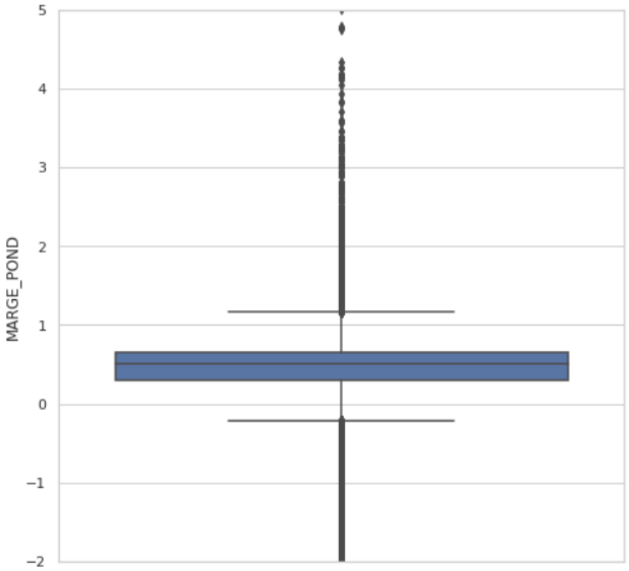


Figure 8 : Boite à moustache brute avec zoom entre -2 et 5

Une fois les valeurs aberrantes détectées, la question du traitement de ces valeurs se pose, à savoir les conserve-t-on ou non. Les valeurs extrêmes représentent ici les marges extrêmes. Une marge extrême négative reflète la présence d'un sinistre grave dans le foyer. A l'inverse, une marge extrême positive correspond à un foyer avec un nombre de contrats très élevé et n'ayant eu que très peu voire aucun sinistre.

On choisira ici de les supprimer de la base de données. Cela entraine une suppression de 5% de la base en conservant tous les foyers avec une marge comprise entre -23% et 118%.

La représentation de la marge après suppression des *outliers* nous donne la boîte à moustache suivante :

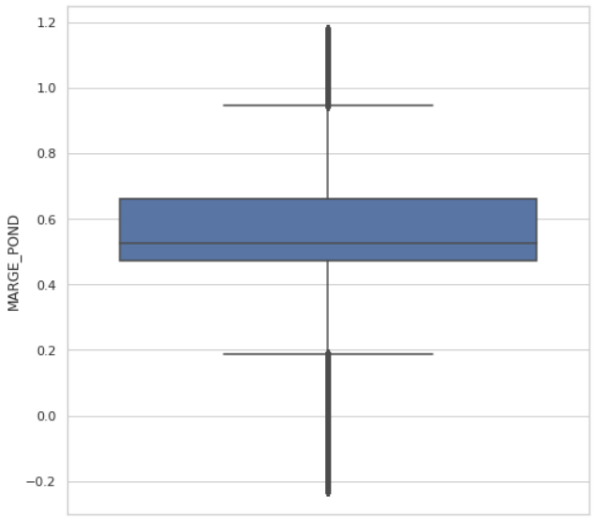


Figure 9 : Représentation de la marge pondérée après suppression des outliers

Le calcul détaillé de la marge pondérée sera explicité par la suite et sera l'une des variables principales à utiliser pour construire notre « valeur actuarielle ».

2. La rentabilité

Chez Thélem Assurances, les ratios S/C cibles fixent les objectifs de rentabilité des produits commercialisés. Ils servent de référence pour l'atteinte des objectifs de marge de chaque exercice. Les structures des coûts de l'entreprise ont été distinguées par produit ou par groupe de produits comme suit :

- **Auto 1ère catégorie :** Autonéo2 + Autonéo + Autocat1
- **Autres Auto :** Autocat2 + Autocat3 + Caravane + Camping-Car
- **Autocat4**
- **Garauto**
- **Santé :** Santalis + 890 + Novasanté
- **Accident de la vie :** GAV + Daylis + AccVie + ACCVP
- **Prévoyance :** MutuaPrev + Génériques
- **Dommages aux biens du particulier :** Néologis2 + Néologis + MRH + RG_multi + CatNat
- **Multirisques Immeuble**
- **Dommages aux biens professionnels :** MRP + MAC + DAB + RG_multi + CatNat
- **Dommages aux biens agricoles :** Terrenis + RG_NC pack Agricole + CatNat
- **Protection Juridique :** PJ Pro + PJ Part
- **Rc Construction**
- **Responsabilité Civile Générale**

Le compte de résultat technique de chacun des produits est construit tel que :

Cotisations acquises	+ C
Primes de réassurances	- PREASS
Produits financiers alloués	+ PFIN
Charge sinistres nette de réassurance	- S_n
Frais de gestion	- FG
Résultat net de réassurance	= R (marge)

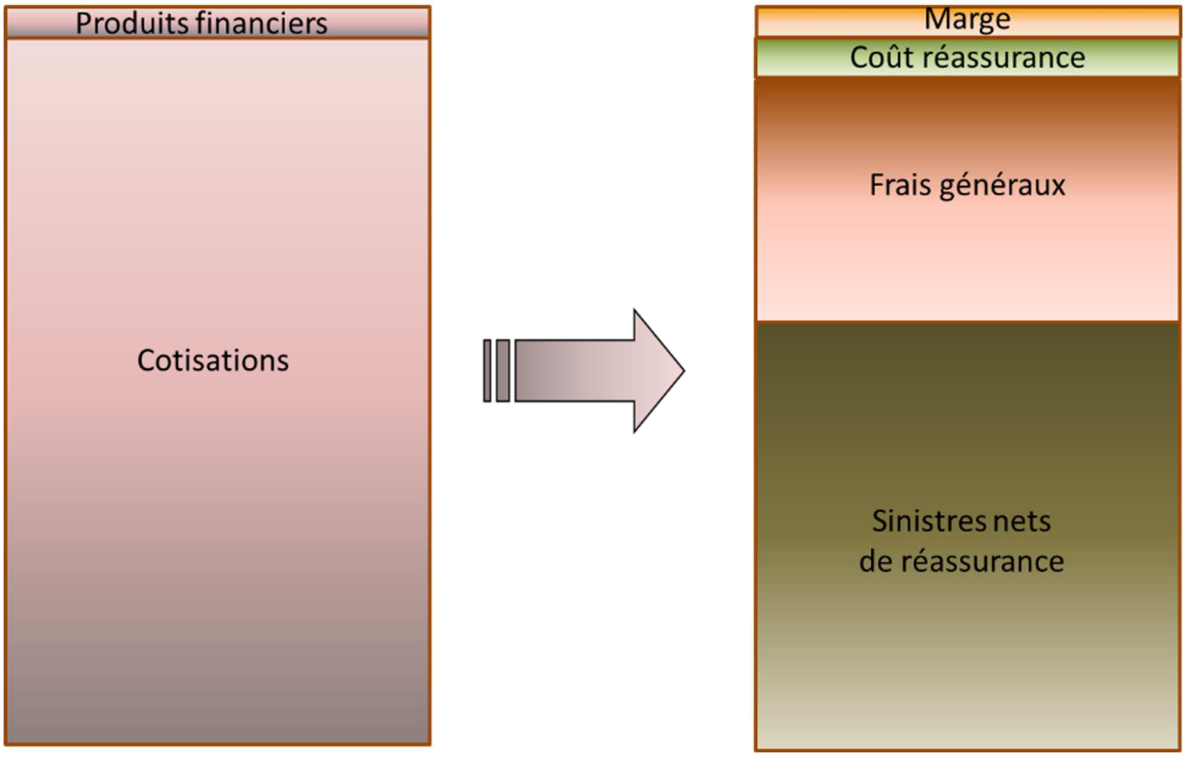


Figure 10 : Economie d'un produit d'assurance

Par la suite nous appellerons Chargements la somme suivante : PFI – FG – REASS

La rentabilité est calculée et pilotée par produit tout au long de l'année pour atteindre un objectif global au niveau société.

	Branche Auto
Cotisations	100,0%
Chargements	22,0%
S/C cible	75,0%
Résultat (marge)	3,0%

Tableau 3 : Exemple de résultats techniques sur la branche Automobile

Ce tableau, comprenant des données fictives, est un exemple de résultat technique des différents produits de la branche automobile. Pour avoir une marge à 3%, le S/C cible, au vu des chargements est de 75%. Un S/C supérieur sera signe d'une rentabilité inférieure à la cible fixée et un S/C inférieur signe d'une rentabilité supérieure à la cible fixée.

3. Conceptualisation de la valeur

La valeur actuarielle sera une métrique prospective. L'objectif est de catégoriser les contrats en fonction des valeurs du foyer suivantes durant la période observée :

- Rentabilité

En considérant l'intégralité des produits « particuliers », on calcule la marge du foyer pondérée par les primes acquises :

$$Marge\ pondérée = \frac{\sum_{produit}((1 - SC - Chargements)\ \times Primes)}{\sum Primes\ acquises}\ .$$

Prenons l'exemple d'un foyer ayant uniquement un contrat auto et n'ayant pas eu de sinistres sur la période observée.

Les éléments servant à calculer sa marge pondérée sont les suivants :

- S/C Auto : 0%
- Chargements : 30%
- Primes acquises Auto : 1175€

$$Marge\ pondérée = \frac{\left((1 - 0 - 30\%)\ \times 1174{,}98\right)}{1174{,}98} = 70\%\ .$$

Avec ces éléments, la marge pondérée du foyer sera donc de 70,7%.

Voyons maintenant l'exemple d'un foyer équipé d'un contrat Auto et Habitation et ayant eu au moins un sinistre. En plus des éléments précédents restant inchangés (hormis pour le S/C Auto et les primes acquises Auto propres à chaque foyer), les éléments servant à calculer sa marge pondérée sont les suivants :

- S/C Auto : 11,1%
- S/C Habitation : 100,7%
- Chargements : 50%
- Primes acquises Auto : 824€
- Primes acquises Habitation : 872€

$$Marge\ pondérée = \frac{\left((1 - 11{,}1\% - 30\%)\ \times 824\right) + \left((1 - 100{,}7\% - 50\%)\ \times 872\right)}{(824 + 872)} = 2{,}5\%\ .$$

Comme attendu, plus la charge sinistre sera élevée, plus vite la marge sera négative. De plus en fonction des produits détenus par le foyer, la marge pondérée (même si le foyer n'a aucun sinistre) fluctuera à cause des chargements propres à chaque produit.

- Ancienneté du foyer

En prenant la date d'effet du contrat le plus ancien, on calcule l'ancienneté en mois. Le contrat concerné n'est pas forcément encore actif.

- Nombre d'affaires nouvelle et de résiliations

On comptabilise ses deux indicateurs durant la période observé. Elle nous renseigne sur l'évolution de l'équipement du foyer et potentiellement sur une évolution du foyer.

- Evolution du logement

Quand le foyer possède son contrat habitation chez Thélem, cet indicateur pourra nous renseigner sur l'évolution du type de logement et/ou du nombre de pièces

- Evolution du véhicule

Idem concernant le véhicule en observant cette fois l'évolution de l'année de mise en circulation et la classe de prix SRA.

Avec ces critères au niveau foyer, différentes méthodes de classification non supervisée seront testées pour attribuer une valeur à chaque foyer.

Cette valeur pourra avoir différentes utilisations potentielles :

- Tarif
- Souscription
- Surveillance du portefeuille

Ce qui nous intéressera dans le cadre de ce mémoire comme première utilisation est l'impact sur le tarif automobile comme une surcouche sur le tarif.

II. Construction de la valeur

Comme dit précédemment, la valeur actuarielle est une métrique prospective et nous n'avons pas d'exemple de ce que nous voulons car ne nous l'avons même pas défini. L'objectif ici est donc que cette valeur se crée sans cible à atteindre, en opérant des regroupements de clients ayant des caractéristiques similaires afin de les intégrer dans le modèle de tarification. Elle ne doit pas répondre à une variable à prédire comme il est habituel de voir dans l'apprentissage supervisé.

L'apprentissage non supervisé répond à cet objectif puisqu'il consiste à entrainer des modèles sur la seule base des échantillons d'apprentissage afin de trouver des patterns ou une structuration naturels dans les données.

Pour effectuer cet apprentissage, nous utiliserons les techniques de clustering cherchant à décomposer la base d'apprentissage en plusieurs sous-ensembles les plus homogènes possibles.

Dans un premier temps, nous avons discrétisé les variables quantitatives de la base puisque les techniques de clustering ne fonctionnent qu'avec des variables qualitatives. La discrétisation est un processus qui permet de recoder une variable quantitative en qualitative ordinale.

Le tableau ci-dessous montre un exemple avec l'évolution de la classe SRA :

EVOL_CLS_SRA	EVOL_CLS_SRA_A-A	...	EVOL_CLS_SRA_M-M	...	EVOL_CLS_SRA_M-P	...	EVOL_CLS_SRA_R-I	...	EVOL_CLS_SRA_S_HC	...
M-P	0		0		1		0		0	
M-M	0		1		0		0		0	
S-HC	0		0		0		0		1	
R-I	0		0		0		1		0	
...										

Tableau 4 : Discrétisation de la variable "Evolution de la classe SRA"

Le problème en effectuant cette discrétisation est le nombre de variables que cela induit. Sur des variables d'évolution comme l'exemple de l'évolution de la classe SRA ci-dessus, cela transforme une variable en potentiellement 900 variables si toutes les combinaisons sont réalisées. Cela pose donc un problème de dimensions que nous allons essayer de résoudre avec une analyse en composantes principales (ACP). Une première solution avant de recourir à l'ACP est de transformer les variables d'évolution en variables numériques. Pour rester sur l'exemple de l'évolution de la classe SRA, un foyer ne changeant pas de véhicule et conservant donc la même classe aura une valeur de 0 quand un passage d'une classe S à N, représentant une baisse de gamme au niveau du véhicule aura une valeur de -5.

1. L'Analyse en Composantes Principales (ACP)

Les grands ensembles de données sont de plus en plus courants et souvent difficiles à interpréter et l'ACP est une technique permettant de réduire la dimensionnalité de ces ensembles de données. L'idée est simple, à savoir réduire la dimensionnalité tout en préservant autant de variabilité que possible.

La préservation de la variabilité consiste à construire de nouvelles variables qui sont des fonctions linéaires de celles de l'ensemble de données d'origine, qui maximisent successivement la variance et qui ne sont pas corrélées les unes aux autres.

De plus l'ACP est utilisé ici non pas pour une visualisation plus simple des données mais pour la mise en œuvre d'apprentissages automatiques que sont les méthodes de classification que nous allons utiliser par la suite.

L'ACP est une méthode d'analyse de données. Elle cherche à synthétiser l'information contenue dans un tableau croisant des individus et des variables quantitatives, en réduisant le nombre de variables explicatives.

Soit $X_1, X_2, \dots, X_p$ les variables initiales quantitatives et centrées. Le but de cette technique est de créer des combinaisons linéaires de ces variables avec une perte minimum de l'information.

Une nouvelle variable Y_1 est déterminée comme étant une combinaison linéaire des X tel que:

$$Y_1 = c_1X_1 + c_2X_2 + \dots + c_pX_p$$

Ici, $c_1, c_2 \dots$ et c_p sont des constantes à déterminer de telle manière que Y_1 ait une variance maximum. En effet, la combinaison linéaire ayant la variance la plus importante est celle qui détruit le moins d'information. La nouvelle variable Y_1 est appelée première composante principale.

En général, Y_1 ne capte pas toute la variance des variables d'origine, c'est pourquoi il est nécessaire de construire une seconde variable Y_2, non corrélée avec Y_1, de variance maximum, combinaison linéaire des X.

Soit V_1 et V_2 la variance respective de Y_1 et Y_2. Alors par construction, $V_1 \geq V_2$.

L'opération est répétée jusqu'à la construction de p variables $Y_3, Y_4, \dots, Y_p$. Chacune de ces variables étant non corrélées avec les précédentes et de variance maximum.

Finalement, il en découlera que $V_1 \geq V_2 \geq \cdots \geq V_p$

Les composantes principales dépendent des variances des variables initiales. Une variable présentant une variance très grande risque de « tirer » à elle tout l'effet de l'ACP. Pour éviter ce phénomène, il convient de centrer et réduire les variables, c'est-à-dire de leur soustraire leur moyenne et de les diviser ensuite par leur variance. On dira alors que nous sommes en ACP normée.

En ce qui concerne le choix *p* du nombre de variables, plusieurs critères peuvent être utilisés en reprenant les valeurs propres de la matrice de corrélation :

- Critère du coude : Sur l'histogramme des valeurs propres, on observe un décrochement (coude) suivi d'une décroissance régulière. On sélectionne les axes avant le décrochement.
- Critère de Kaiser : En ACP normée, ce critère conduit à ne retenir que les valeurs propres supérieures à 1.
- Critère lié à l'information : On choisit le nombre d'axes en fonction de la restitution minimale d'information que l'on souhaite.

Afin de l'implémenter sous Python, nous normaliserons dans un premier temps les données avec la librairie *sklearn* et la fonction *StandardScaler*. Très simplement cette fonction normalise et centre les données.

Par la suite la mise en place de l'ACP s'effectue avec la librairie *sklearn* et la fonction *PCA* qui comme expliciter précédemment réduit la dimensionnalité linéaire. Le paramètre restant à déterminer pour cette fonction est le nombre de composante à conserver pour conserver. Pour effectuer ceci, on utilisera la règle du coude de Cattell. Sur le graphique des valeurs propres, on observe un décrochement suivi d'une croissance régulière, et on sélectionne le nombre d'axe avant jusqu'au coude. Le but étant d'obtenir le maximum d'inertie conservée avec le minimum de facteurs.

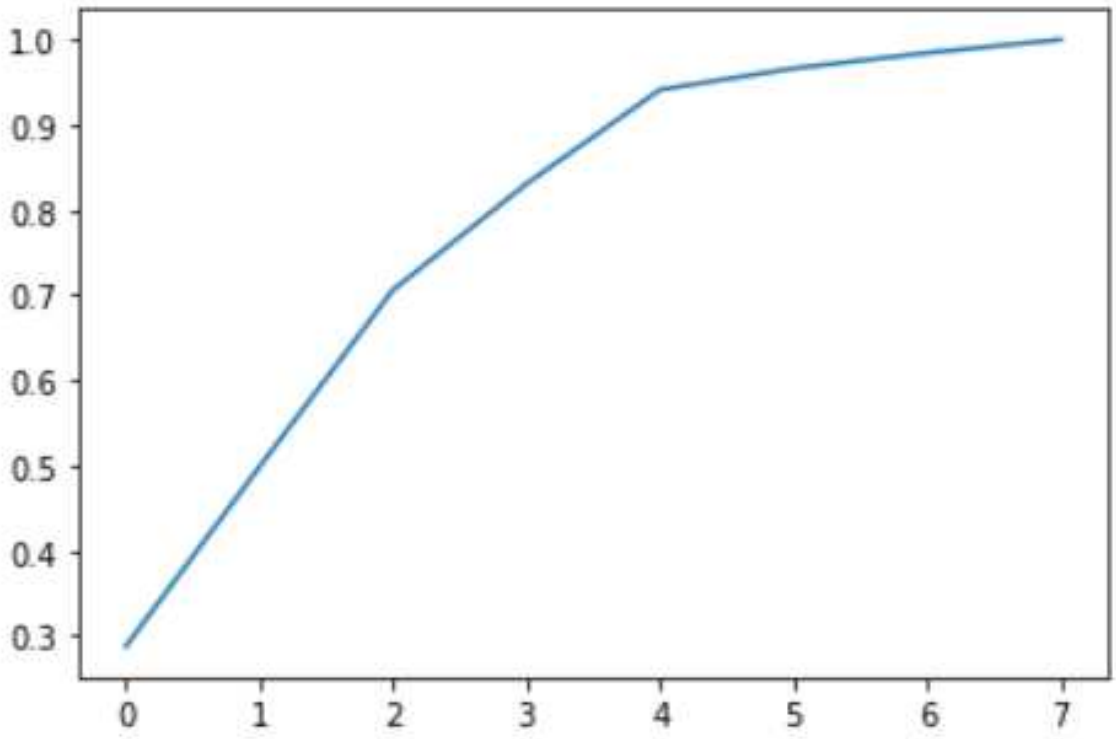


Figure 11 : Représentation graphique de l'inertie par nombre de composantes

Avec la méthode du coude, on peut conserver 4 composantes. Ce choix de paramètre conserve 95% de l'inertie et de l'information des variables.

A titre purement informatif, nous effectuons également l'ACP avec 2 composantes pour avoir une visualisation des données en deux dimensions avec un nuage de points.

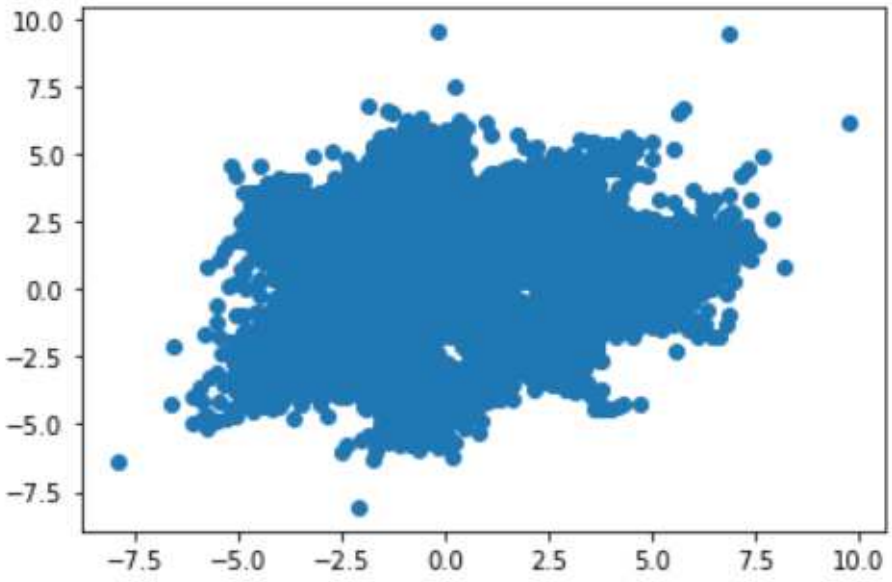


Figure 12 : Représentation en 2 dimensions de la base de données

Cette représentation nous montre que tous les foyers observés sont globalement centrés mais il faut prendre cette représentation avec prudence puisque l'inertie conservée avec 2 composantes n'est que de 70%.

Pour les différentes méthodes de classification à venir, nous utiliserons l'ACP en conservant 4 composantes principales.

2. Apprentissage non supervisée et clustering

Depuis plusieurs dizaines d'années, une question revient sans cesse : pourra-t-on concevoir un jour une intelligence artificielle de niveau humain ? Aujourd'hui nous sommes encore loin d'y arriver. Et cela ne sera pas possible en utilisant les méthodes classiques d'apprentissages supervisé. Selon Yann LeCun (un des plus grands chercheurs en intelligence artificielle, considéré comme un des inventeurs de l'apprentissage profond), « Nous n'obtiendrons pas une intelligence aussi générale que celle des humains avec la supervision ou l'apprentissage multi-tâche. Il va nous falloir autre chose. » Et pourquoi pas l'apprentissage non supervisé.

L'apprentissage non supervisé consiste à apprendre à un algorithme d'intelligence artificielle des informations qui ne sont ni étiquetées, ni classées pour permettre de réagir à ces informations sans intervention humaine, c'est-à-dire sans superviseur. De plus, l'algorithme traite les données sans aucun entrainement préalable, il « s'entraine lui-même » avec les données qu'il reçoit.

Néanmoins, ce n'est pas parce que l'on parle d'apprentissage non supervisé que l'on doit omettre la notion de catégories pour les algorithmes de classifications. Un algorithme d'apprentissage non supervisé utilise des catégories associées aux données qu'on lui soumet, mais il doit les faire émerger lui-même, afin, par exemple, de reconnaître qu'un chien est un chien, ou qu'un article d'Arthur Charpentier est un article d'Arthur Charpentier.

Contrairement à l'apprentissage supervisé, aucun label (ou variable réponse) n'est fourni. L'algorithme doit donc être en mesure de créer lui-même des catégories, et même s'il ne sait pas ce qu'elles représentent, il remarquera leurs similarités. Par exemple si on fournit une base d'images d'animaux, l'algorithme regroupera toutes les images de chat ensemble car elles auront toutes un certain nombre de points communs : taille, quatre pattes, forme du visage etc…

Concrètement, les algorithmes non supervisés repèrent des similarités dans les données pour pouvoir ensuite les structurer. Ce partitionnement des individus est appelé *clustering* ou classification. Avant que l'intelligence artificielle ne devienne capable de détecter des similarités entre individus, ce sont bien des intelligences humaines qui ont implémentés les algorithmes de clustering.

Pour chaque méthode, il est nécessaire de choisir comment mesurer la similarité entre deux individus que l'on peut imaginer comme deux points de l'espace des réels en dimension p. Il nous faut donc une distance, comme la distance euclidienne. Les n individus sont des « points » de l'espace de variables $\mathbb{R}$ en dimension p.

Dans cet apprentissage, les données sont représentées comme suit :

$$\begin{pmatrix} x_{(1,1)} & x_{(1,2)} & x_{(1,\dots)} & x_{(1,p)} \\ x_{(2,1)} & x_{(2,2)} & x_{(2,\dots)} & x_{(2,p)} \\ \dots & \dots & \dots & \dots \\ x_{(n,1)} & x_{(n,2)} & x_{(n,\dots)} & x_{(n,p)} \end{pmatrix}$$

Chaque ligne représente un foyer. A l'issu de l'application du clustering, on retrouvera ces données regroupées par ressemblance. Le clustering va regrouper en plusieurs familles les foyers en fonction de leurs caractéristiques. Ainsi, les individus se trouvant dans une même classe sont similaires et les données se trouvant dans un autre cluster ne le sont pas.

Les grandes catégories de clustering sont les méthodes suivantes :

- Hiérarchiques tel que la classification ascendante hiérarchique
- Centroïdes tel que la méthode des k-moyennes et MeanShift
- À densité tel que DBSCAN et propagation d'affinité

Les différentes méthodes cités comme exemples sont celles qui vont être testées pour notre problématique.

i. K means

La première méthode de classification non supervisée utilisée, qui est également la plus connue, est le K-means, ou algorithme des centres mobiles.

Il faut commencer par déterminer combien de groupes on souhaite trouver et on appellera ce nombre *K*. L'objectif de la méthode est de partitionner en différentes classes des individus et elle a besoin pour cela d'un moyen de comparer le degré de similarité entre les différentes observations. Ainsi, deux données qui se ressemblent auront une distance de similarité réduite, alors que deux objets différents auront une distance de séparation plus grande.

Généralement, on utilise la distance euclidienne. Entre deux observations x_1 et x_2, elle se calcule comme suit :

$$d(x_1, x_2) = \sqrt{\sum_{j=1}^{n} (x_{1n} - x_{2n})^2}\,.$$

L'algorithme se déroule ensuite par étape jusqu'à sa convergence comme suit :

- ***Etape 0 :*** Pour initialiser l'algorithme, on tire au hasard k individus appartenant à la population, $C_1^0, C_{21}^0, \dots, C_k^0$. Ce sont les k centres initiaux. L'indice numérote les différents centres et l'exposant indique qu'il s'agit des k centres.
- ***Etape 1 :*** **Constitution de classes** : On répartit l'ensemble des individus en k classes $\Gamma_1^0, \Gamma_2^0, \dots, \Gamma_k^0$ en regroupant autour de chaque centre C_i^0 pour $i = 1, \dots, n$ l'ensemble des individus qui sont plus proches du centre C_i^0 que des autres centres C_j^0 pour $j \neq i$.
- ***Etape 2 :*** **Calcul des nouveaux centres :** On détermine les centres de gravité $G_1, G_2, \dots, G_k$ des k classes ainsi obtenues et on désigne ces points comme les nouveaux centres $C_1^1 = G_1, \dots, C_k^1 = G_k$.
- ***Répétition des étapes 1 et 2 :*** On répète ces deux étapes jusqu'à la stabilisation de l'algorithme, c'est-à-dire jusqu'à ce que le découpage en classes obtenu ne soit plus modifié par une itération supplémentaire.

On peut illustrer la méthode par le schéma ci-dessous :

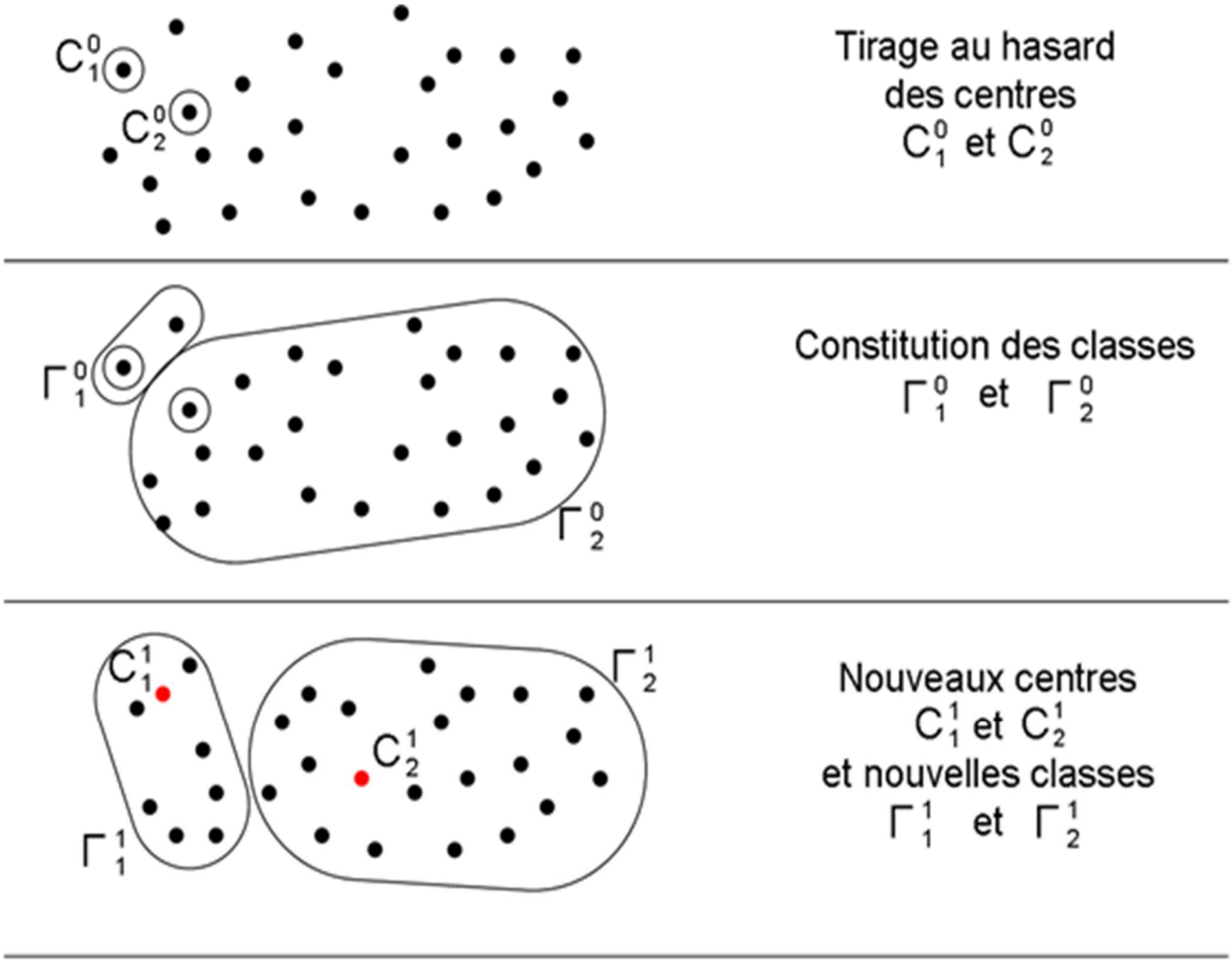


Figure 13 : Algorithme des centres mobiles

Reste à choisir le nombre de classes K. Choisir ce paramètre n'est pas forcément intuitif. Trop grand, le partitionnement des données sera trop fragmenté ce qui empêchera de découvrir des *patterns* intéressants. A l'inverse, un nombre de classes trop petit, conduira à avoir, potentiellement, des classes trop généralistes contenant beaucoup de données. Pour un même jeu de données, il n'existe pas une unique classification possible. La difficulté résidera donc à choisir le juste K. Malheureusement, il n'existe pas de procédé automatisé pour trouver le bon nombre de clusters.

La méthode la plus usuelle pour choisir le nombre de classes est de lancer l'algorithme avec différentes valeurs de K et de calculer la variance des différentes classes. La variance est la somme des distances entre chaque centroïde d'une classe et des différentes observations inclues dans cette même classe. On cherche donc à trouver K de telle sorte que les classes retenues minimisent la distance entre leurs centroïdes. On parle de minimisation de la distance intra-classe.

La variance des classes se calcule comme suit :

$$V = \sum_j \sum_{x_i \to c_j} D(c_j, x_i)^2 .$$

Avec :

- c_j : le centre de la classe (le centroïde)
- x_i : la ième observation dans la classe ayant pour centroïde c_j
- $D(c_j, x_i)$: la distance euclidienne entre le centre de la classe et le point x_i

En mettant dans un graphique la variance en fonction du nombre de classes, nous obtenons ceci :

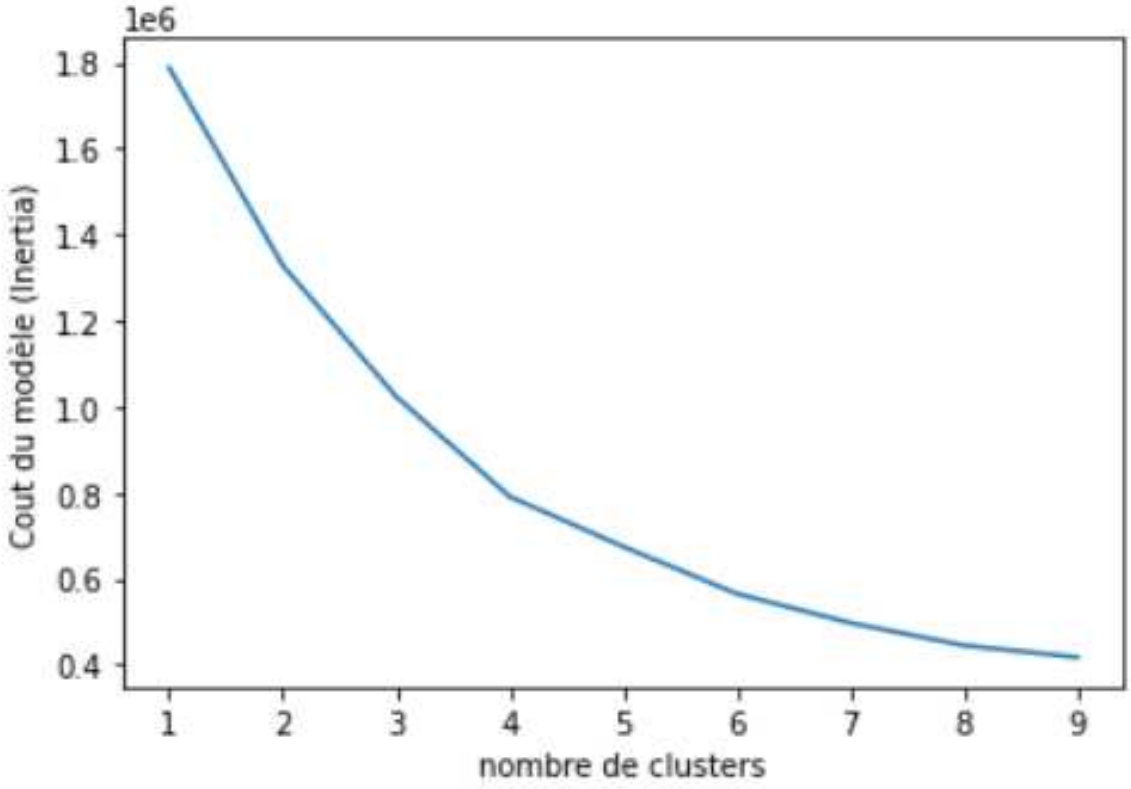


Figure 14 : Méthode du coude pour choix optimal du critère K

On peut imaginer sur ce graphique la forme d'un bras où le point le plus haut représente l'épaule et l'autre extrémité représente la main. Le nombre optimal est le point représentant le coude. Cette méthode se nomme la méthode Elbow (la méthode du coude en français). Généralement, le point du coude est celui du nombre de classes à partir duquel la variance ne se réduit pas significativement. C'est le nombre optimal de classes. Ici, le coude n'est pas franc mais on pourrait se diriger vers $K = 4$ ou $K = 6$.

On peut compléter ou mettre en compétition cette méthode avec une approche plus précise mais qui demande plus de temps de calcul à savoir le coefficient de silhouette. Pour un point x données, le coefficient de silhouette $s(x)$ permet d'évaluer si ce point appartient au bon cluster : est-il proche des points de la classe auquel il appartient ? Est-il loin des autres points ?

Le coefficient de silhouette se définit comme suit :

$$s = \frac{b - a}{\max(a, b)}$$

Avec a la moyenne intra classe et b la distance moyenne au cluster le plus proche.

Ce coefficient peut varier entre -1 et +1. Plus il est proche de 1, plus l'assignation du foyer à sa classe est satisfaisante. A l'inverse, un coefficient proche de 1 signifiera que le foyer est associé à la mauvaise classe et proche de 0 qu'elle se situe près d'une frontière.

A l'image de la méthode du coude, il est judicieux d'afficher l'évolution de ce coefficient sur un graphique en fonction du nombre de classes pour déterminer le K optimal.

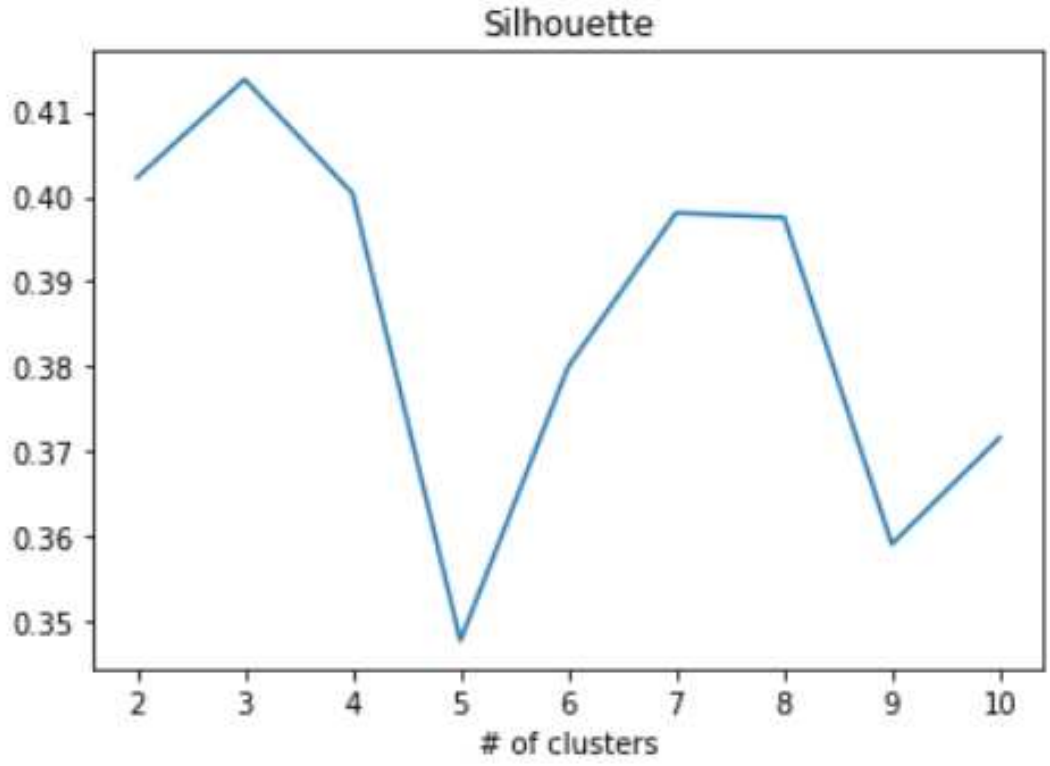


Figure 15 : Coefficient de silhouette moyen en fonction du nombre de classes

A première vue, le nombre de classes optimales pour la classification est 3. Cependant, les coefficients de silhouette pour le nombre de classes 3, 4, 7 et 8 varient de 0,02. Nous pouvons donc orienter notre choix en fonction du nombre de classes que nous souhaiterions obtenir pour les besoins de l'étude. Dans cet optique, nous opterons pour $K = 7$.

ii. MeanShift

Le *MeanShift* est un algorithme itératif qui a pour objectif de faire converger un point vers le maximum local le plus proche. Son objectif est de découvrir des masses dans une densité d'échantillons lisses. Il est basé sur une fenêtre coulissante qui tente de trouver des zones denses de points de données. On commence par une fenêtre circulaire centrée en un point C et dont le rayon est le noyau.

L'algorithme consiste à déplacer ce noyau de façon itérative vers une région à densité plus élevée à chaque étape jusqu'à la convergence c'est-à-dire jusqu'à ce qu'il n'y ait plus de direction à laquelle un déplacement pourra contenir plus de points.

Ce processus se fait avec de multiples fenêtre coulissantes du moment que tous les points ne se trouvent pas à l'intérieur d'une des fenêtres. En cas de chevauchement, la fenêtre contenant le plus grand nombre de points sera conservée. Les données seront ensuite groupées en fonction de la zone dans laquelle ils se trouvent.

Pour cet algorithme, il n'est pas nécessaire de connaitre le nombre de classes à créer mais le paramètre de la taille du rayon doit être imposé et est non négligeable sur le nombre de clusters que le processus va engendrer.

Rayon	Nombre de clusters	% foyers dans la classe la plus représentée	Nb de classes où 95% des foyers sont représentés
None	12	69%	4
1	247	54%	9
2	28	64%	5
2,2	18	66%	4
2,5	12	69%	3
2,6	10	76%	3
2,7	9	87%	2
2,75	8	87%	2
2,8	8	87%	2
2,85	8	85%	2
3	6	99%	1
4	2	100%	1
5	1	100%	1
10	1	100%	1
20	1	100%	1

Tableau 5 : Rayons et nombres de classes de la procédure MeanShift

Quand la valeur du rayon est définie sur « None », le rayon est estimé par une autre fonction Python mais il est recommandé pour les bases de données volumineuses, de définir le rayon sur une petite valeur.

Comme le montre le tableau précédent, la taille du rayon choisi comme paramètre est très influente sur le nombre de classes créées allant de 1 pour un rayon supérieur à 4, qui ne nous apporterait donc rien, à 247 pour un rayon de 1, qui est beaucoup trop élevé pour l'utilisation que nous envisageons de cette valeur. On remarque également que quel que soit le nombre de classes construit par l'algorithme, nous retrouvons une sur-représentation d'une ou deux classes.

Classe	Proportion
0	86,7%
1	13,1%
2	0,1%
3	0,0%
4	0,0%
5	0,0%
6	0,0%
7	0,0%
8	0,1%

Tableau 6 : Proportion de la prédiction de l'algorithme MeanShift avec un rayon de 2.7

Par exemple, avec un rayon de 2.7, l'algorithme crée 9 classes. Cependant, 87% des foyers sont regroupés dans une seule classe et 95% des foyers sont représentés par 2 classes. Il est donc difficile d'envisager l'utilisation de cet algorithme si cela nous amène à avoir une surreprésentation d'une classe. La segmentation recherché par cette valeur ne serait pas assez « forte » pour qu'elle soit utilisable. Au vu de ces éléments, cet algorithme ne sera pas retenu pour la suite.

iii. La propagation d'affinité

Une autre méthode de classification est la propagation d'affinité qui est un algorithme récent de partitionnement de données.

Afin de classifier par propagation d'affinité, il est nécessaire de calculer une matrice de similarité qu'on notera S. Les éléments $S(x_i, x_k)$ de la matrice représentent la similitude entre tous les couples d'individus (x_i, x_k) pour $x_i \neq x_k$. La distance euclidienne négative est régulièrement utilisée comme mesure de similarité :

$$S(x_i, x_k) = -\|a_i - a_k\|_2 \,, \forall\, x_i \neq x_k$$

Avec a_i le vecteur d'attributs représentant l'individu x_i.

Les éléments diagonaux eux reflètent la pertinence à priori du choix de l'individu k pour servir de représentant et sont nommés les préférences. Ces éléments ne sont pas calculés de la même manière que les éléments $S(x_i, x_k)$ pour $x_i \neq x_k$ puisqu'ils sont initialisés à la valeur médiane des éléments de S pour $x_i \neq x_k$.

$$S(x_k, x_k) = p \,, \forall\, x_k \,.$$

L'algorithme de propagation d'affinité crée des clusters en envoyant des messages entre les points de données jusqu'à la convergence de celui-ci. A chaque itération, la question suivante est posée : quel individu sera considéré comme représentant ou exemplaire de tous les autres, et quel représentant sera choisi pour chaque individu ?

Deux procédures de transmission de messages sont utilisées pour échanger les messages entre l'individu x_i et un représentant candidat x_k. Ils sont appelés responsabilité et disponibilité et sont illustrés dans un exemple sur la figure suivante.

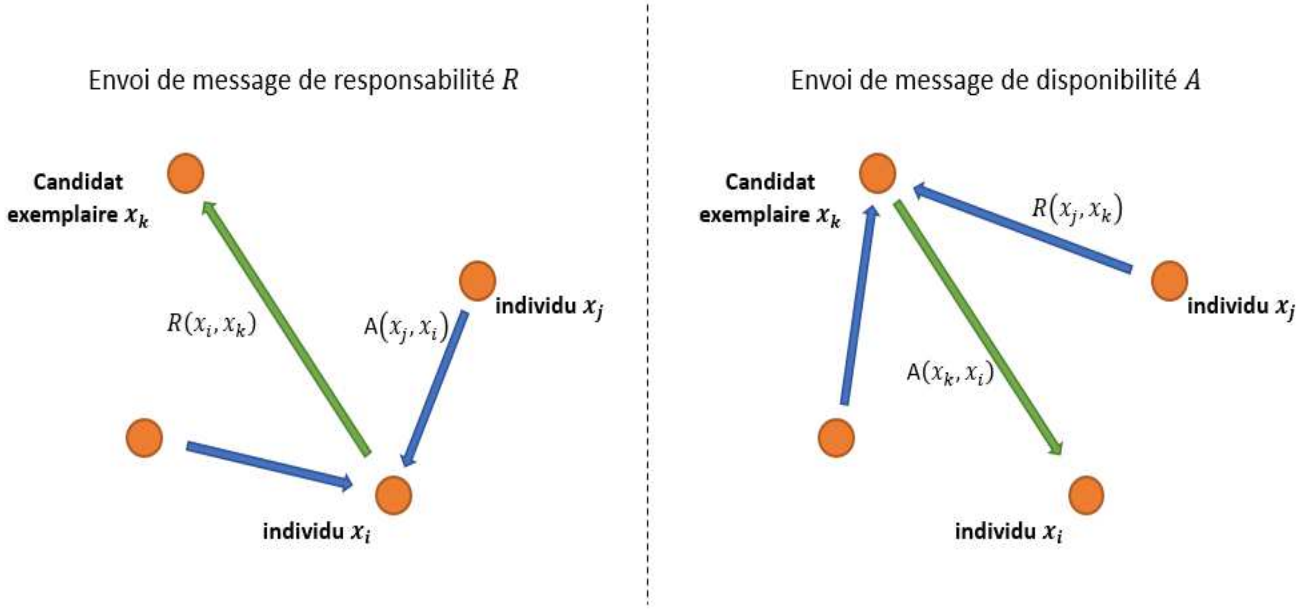


Figure 16 : Procédures d'échanges de message de la propagation d'affinité

La responsabilité $R(x_i, x_k)$ est le message envoyé à partir de x_i au candidat représentant x_k, indiquant combien il serait un bon représentant de x_i. Alternativement, la disponibilité $A(x_i, x_k)$ est le message envoyé du candidat représentant x_k à l'individu x_i, indiquant à quel point il serait approprié pour lui de choisir x_k comme son représentant. Cette procédure cherche donc pour chaque individu le représentant qui maximise la somme des disponibilités et des responsabilités.

Les principales étapes de l'algorithme sont les suivants :

- **Initialisation**

  Une fois la matrice de similarité calculée, les matrices de responsabilité R et de disponibilité A sont initialisées à zéros. L'algorithme effectue ensuite les mises à jour suivantes de manière itérative.

- **Mise à jour de la responsabilité**

$$R(x_i, x_k) = S(x_i, x_k) - max_{j, j \neq k} \left(A(x_j, x_i) + S(x_i, x_j) \right) \quad \forall\, x_i \neq x_k$$

- **Mise à jour de la disponibilité**

$$A(x_k, x_i) = min\left\{0\,, R(x_k, x_k) + \sum_{(j, j\neq k)} max\{0\,, R(x_j, x_k)\}\right\} \;\; \forall\, x_i \neq x_k$$

$$A(x_k, x_k) = \sum_{(j, j\neq k)} max\{0\,, R(x_k, x_k)\} \;\; \forall\, x_k$$

- **Affectation des représentants**

$$E^*(x_i) = \arg\max_k \{R(x_i, x_k) + A(x_k, x_i)\}$$

Avec $E^*(x_i)$ le représentant attribué à l'individu x_i.

Les itérations sont effectuées jusqu'à ce que les conditions d'arrêts soient remplis à savoir que les exemplaires ne doivent pas changer pour les n dernières itérations de la boucle principale, où n est une valeur prédéfinie par l'utilisateur. Le tableau suivant indique le nombre de classes crées en fonction de la préférence choisi.

Preference	Nombre de clusters
None	diverge
-50	diverge
-10	diverge
-5	diverge
0	diverge
5	26907
10	26907
50	26907
100	26907
1000	26907

Tableau 7: Nombre de classes créées selon le paramètre de préférence

En variant le paramètre de préférence, que ce soit la valeur par défaut « None », les valeurs négatives nulles, positives, plus ou moins importantes, nous observons que les résultats ne correspondent pas à l'attendu. D'un côté, les préférences négatives ainsi que la valeur par défaut amène un algorithme qui ne converge pas et ne joue donc pas son rôle de classifieur. D'un autre côté, avec des préférences positives, le nombre de classes crée par l'algorithme est beaucoup trop élevé et inexploitable pour l'utilité attendue de la valeur.

iv. DBSCAN

Cette méthode de classification est un algorithme basé sur la densité dans la mesure qui s'appuie sur la densité estimée des classes pour effectuer le partitionnement.

DBSCAN utilise deux paramètres : la distance maximale ε qui peut définir deux individus comme voisins, et N le nombre minimum de points nécessaires dans un rayon ε pour que ces points soient considérés comme une classe.

Ensuite l'algorithme peut se diviser en quatre étapes :

- On regarde le nombre de points à au plus une distance ε de chaque observation. On appelle cette zone le ε-voisinage de l'observation
- Si une observation compte au moins un certain nombre de points y compris elle-même, elle est considérée comme une observation cœur. On a alors décelé une observation à haute densité.
- Toutes les observations au voisinage d'une observation cœur appartiennent au même cluster. Il peut y avoir des observations cœur proche les unes des autres. Par conséquent de proche en proche on obtient une longue séquence d'observations cœur qui constitue un unique cluster.
- Toute observation qui n'est pas une observation cœur et qui ne comporte pas d'observation cœur dans son voisinage est considérée comme un bruit.

Les informations ε et N sont renseignées par l'utilisateur et contrairement au *K-Means* il n'y a pas besoin de définir en amont le nombre de classes.

Le choix de la distance ε va faire varier le nombre de classes et le nombre de points considéré comme des bruits. Si ε est trop petit, le ε-voisinage est trop faible et toutes les observations seront considérées comme des anomalies. Au contraire, si ε est trop grand, chaque observation contiendra dans son ε-voisinage toutes les autres observations de la base de données. Cela impliquerait l'obtention d'une unique classe. Il est donc important de tester plusieurs ε et N pour obtenir un partitionnement de qualité.

Les tests effectués sur notre ensemble de données donnent les résultats suivants :

Paramètres		Résultats obtenus	
Distance epsilon ε	mminimum de points N	Nombre de classes	Nombre de bruits
0,2	10	285	42 634
0,3	10	74	21 323
0,4	10	12	10 658
0,45	10	12	7 453
0,5	5	30	3 555
0,5	10	7	5 327
0,5	15	6	6 965
0,5	20	7	8 187
0,6	10	5	3 013
0,7	10	7	1 748
0,8	10	3	1 061

Tableau 8 : Résultats DBSCAN en fonction des paramètres choisis

Les résultats obtenus concordent avec la théorie à savoir que plus ε est petit, plus le nombre de bruits, considérés comme des anomalies et des points sans classes, est important. Il en est de même pour l'augmentation du nombre de points minimal pour constituer une classe. Au vu des résultats, une distance $\varepsilon = 0{,}7$ et $N = 10$ semble être la meilleure combinaison pourra voir un partitionnement de qualité.

Cependant, en observant les résultats de la classification, on observe une sur-représentation d'une classe quel que soit les paramètres choisis.

Classe	Proportion
-1	2,0%
0	93,6%
1	4,1%
2	0,1%
3	0,1%
4	0,0%
5	0,0%

Tableau 9 : Extraction résultat classification DBSCAN ε=0,5 et N=10

Classe	Proportion
-1	0,6%
0	99,3%
1	0,0%
2	0,0%
3	0,0%
4	0,0%
5	0,0%

Tableau 10 : Extraction résultat classification DBSCAN ε=0,7 et N=10

Classe	Proportion
-1	4,0%
0	92,2%
1	3,7%
2	0,0%
3	0,1%
4	0,0%
5	0,0%
6	0,0%
7	0,0%
8	0,0%
9	0,0%
10	0,0%

Tableau 11 : Extraction résultat classification DBSCAN ε=0,45 et N=10

Comme évoqué précédemment, les extractions des résultats des classifications avec 3 paramètres ε différents nous montrent un déséquilibre sur une classe avec une sur représentation dépassant les 90% de la base observée. On émet donc l'hypothèse que cette méthode ne correspond pas à nos données et qu'elle nous donne des résultats incohérents par rapport à ce que l'on recherche. Même si nous savons qu'une répartition parfaite ne sera jamais envisageable, la sur représentation d'une classe à ce niveau reviendrait à n'avoir qu'une classe unique et n'aurait donc aucun sens pour notre étude.

Les résultats de cet algorithme ne seront donc pas conservés ni étudiés par la suite.

v. Classification ascendante hiérarchique (CAH)

Il y a deux approches possibles concernant la classification hiérarchique. L'approche ascendante, aussi appelé clustering agglomératif et l'approche descendante, aussi appelé clustering divisif. Concernant l'approche descendante, on part d'une grande classe contenant tous les foyers, puis on le divise successivement jusqu'à obtenir autan de classes que d'individus

Concernant l'approche ascendante qui sera utilisée ici, l'algorithme peux se décomposer comme suit :

- ***Etape 0 :*** A l'étape initiale, les n individus constituent des classes à eux seuls.
- ***Etape 1 :*** On calcule les distances deux à deux entre individus et les deux individus les plus proches sont regroupés en une classe.
- ***Etape 2 :*** La distance entre cette nouvelle classe et les n-2 individus restants est ensuite calculée, et à nouveau les deux éléments (classes ou individus) les plus proches sont réunis.
- ***Répétition de l'étape 2 :*** On répète cette étape jusqu'à ce qu'il ne reste plus qu'une seule classe constituée de tous les individus.

On obtient donc une arborescence qui a un cluster à son sommet et qui se divise petit à petit jusqu'à avoir autant de classes que d'individus. On appelle cette arborescence un dendrogramme. Quel que soit l'approche, on a besoin de mesurer la distance entre 2 classes. Pour cela, on utilise des méthodes de liens qui permettent de lier les clusters lorsque l'on construit pas à pas l'arborescence. Nous avons :

- Le lien simple (simple linkage) : on considère que la distance entre 2 classes est la distance entre leurs 2 points les plus proches.
- Le lien complet (complete linkage) : on considère que la distance entre 2 classes est la distance entre leurs 2 points les plus éloignés.
- Le lien moyen : on considère que la distance entre 2 classes est la moyenne de toutes les distances entre les points d'une classe et les points de l'autre classe. Pour la calculer, on énumère toutes les paires de points possibles d'une classe à l'autre, puis on calcule la distance de chaque paire, puis on calcule la moyenne.
- Le lien centroïdal : on considère que la distance entre 2 classes est la distance entre les centroïdes de ceux-ci.

Les méthodes de lien permettent de garantir que les classes sont bien séparés mais elles ne garantissent pas que les classes sont resserrés sur elles-mêmes. C'est là qu'intervient la notion d'inertie intraclasse. Il s'agit simplement de la somme des distances euclidiennes entre chaque point associé à la classe et le centre de gravité nouvellement calculé. Le fait de regrouper les classes au fur et à mesure va augmenter l'inertie intraclasse. Le but est de faire en sorte de minimiser cette augmentation. Pour résoudre cela, il existe la méthode de Ward qui, à chaque itération (à chaque fois que deux classes vont être regroupés en une), cherche à minimiser l'augmentation d'inertie intraclasse. Cette méthode est très souvent utilisée par défaut et elle le sera dans le cadre de notre classification.

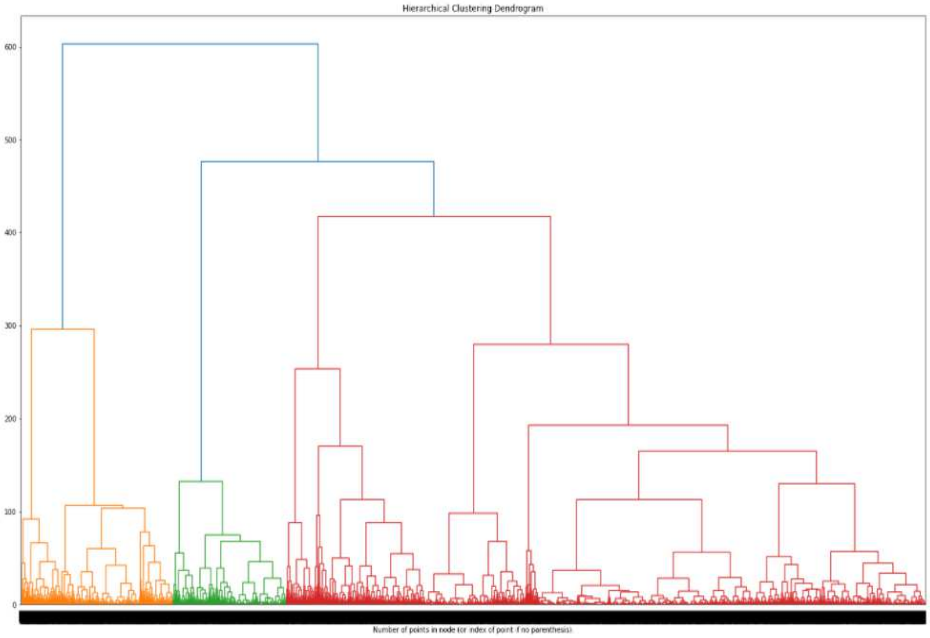


Figure 17 : CAH complète

Avec un nombre important d'individus comme c'est le cas dans notre étude, le bas du dendrogramme est illisible, c'est pourquoi on peut ne représenter que le haut de l'arborescence en tronquant les niveaux les plus fins. Cela améliorera grandement la lisibilité.

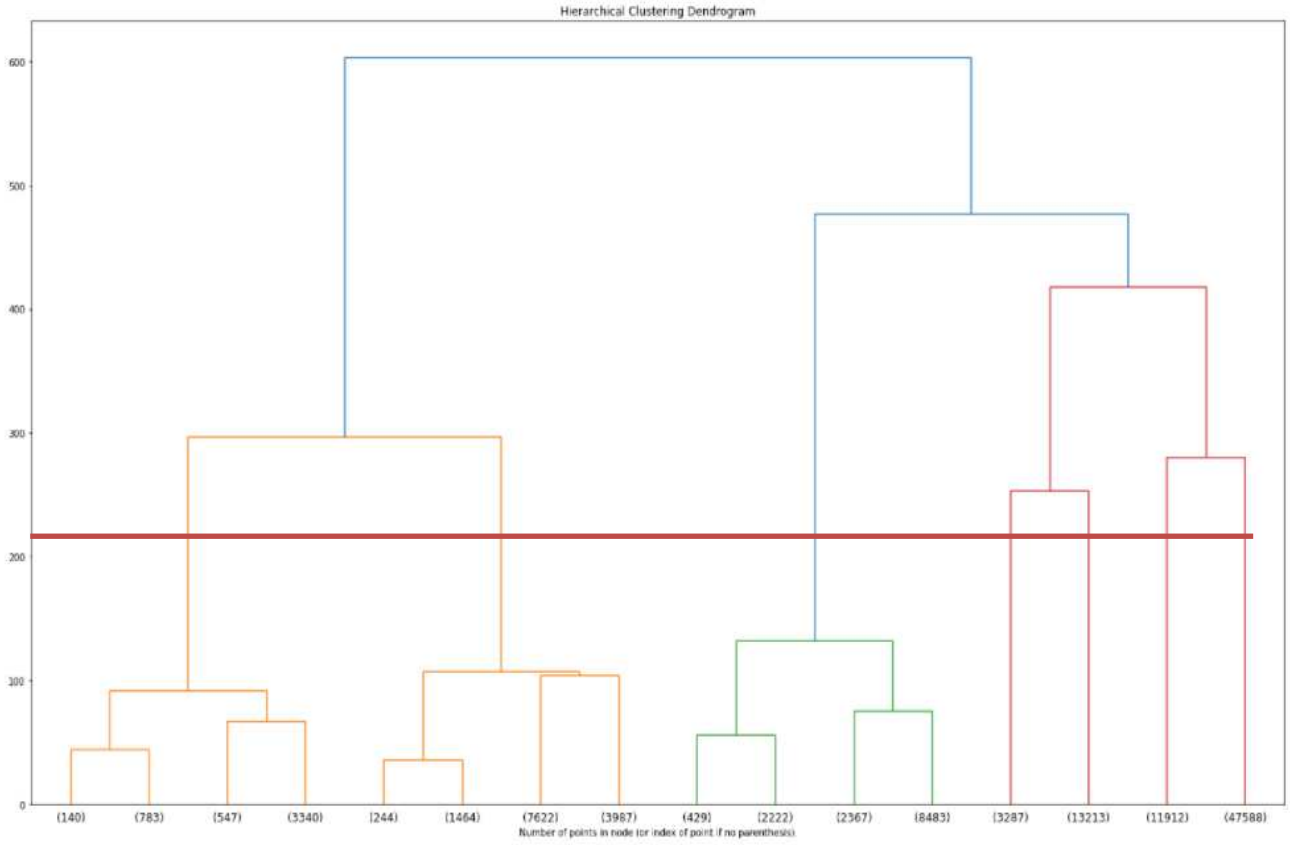


Figure 18 : CAH tronqué

En ayant tronqué les niveaux les plus fins, on a une meilleure visibilité des résultats et du nombre d'individus par branche. Ce dendrogramme va nous permettre de déterminer le nombre de cluster optimal. Le partitionnement correspondant au plus grand saut entre deux clusters consécutifs est souvent le choix le plus pertinent (représenté par l'axe vertical rouge sur le dendrogramme ci-dessus).

Notre choix se portera donc sur 7 classes avec cette méthode.

Pour cette méthode, nous avons rencontré des problèmes de mémoire vive pour effectuer les calculs de l'algorithme et il nous a été impossible d'entrainer la classification sur l'ensemble des données. Nous avons dû réduire la taille de la base à 40% de sa taille initiale pour que la procédure python soit capable de tourner.

Pour extraire une partie de la base de données initiale, nous utiliserons sous python la procédure *sample* qui renvoie un échantillon aléatoire. Dans cette procédure, nous utiliserons l'option *random_state* qui permet de conserver le même échantillon aléatoire même si le code est relancé plusieurs fois.

Afin d'attribuer les points non entrainés aux classes déterminés par l'algorithme, nous utiliserons la méthode des K_plus proches voisins.

vi. K plus proches voisins (KNN : *K Nearest Neighbors*)

Comme expliquer précédemment, la méthode *KNN* va palier le fait de travailler et de n'avoir la classification que sur une partie de la base. Le principe de ce modèle consiste à choisis les *K* données les plus proches du point étudié afin de le rattacher afin d'en prédire sa valeur. Dans notre cas, à rattacher le foyer non entrainé à la bonne classe.

Les étapes de cet algorithme sont assez simples :

- ***Etape 1 :*** Sélectionner le nombre *K* de voisins
- ***Etape 2 :*** Calculer la distance euclidienne du point non classifié aux autres points
- ***Etape 3 :*** Prendre les *K* voisins les plus proches selon la distance calculée
- ***Etape 4 :*** Parmi ces *K* voisins, compter le nombre de points appartenant à chaque catégorie
- ***Etape 5 :*** Attribuer le nouveau point à la catégorie la plus présente parmi ces *K* voisins

Reste maintenant à déterminer la valeur de *K*. Pour cela, nous exécutons plusieurs fois l'algorithme *KNN* avec différentes valeurs de *K* afin de voir celui qui à le pourcentage d'erreur le plus faible.

Cependant, *K* ne doit être ni trop petit ni trop grand. Prenons l'exemple ci-dessous.

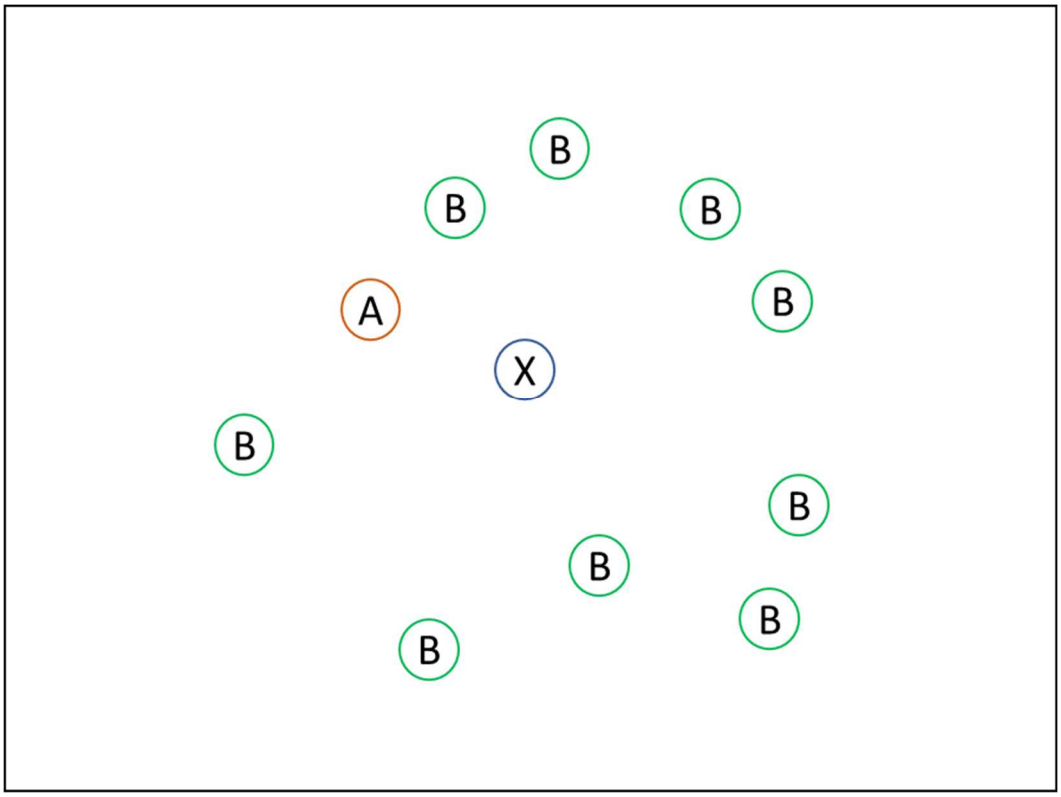


Figure 19 : Exemple de cas extrême de KNN : affectation de X

Dans cet exemple, nous aurions tendance à affecter logiquement notre individu X à la classe B qui est la plus représenté ici mais si on choisit $K = 1$, le point le plus proche est la classe A et l'individu X serait donc affecté à la classe A. Avec cette valeur de K, l'algorithme prédit donc de manière incorrect l'affectation de X.

Inversement, plus nous augmentons la valeur de K, plus nous courons le risque d'observer le phénomène de sur apprentissage qui se produit lorsque les données d'apprentissage expliquent très voire « trop » bien les données mais ne parviennent pas à faire des prédictions utiles et fiables sur de nouvelles données.

Appliquons donc cet algorithme à notre classification ascendante hiérarchique.

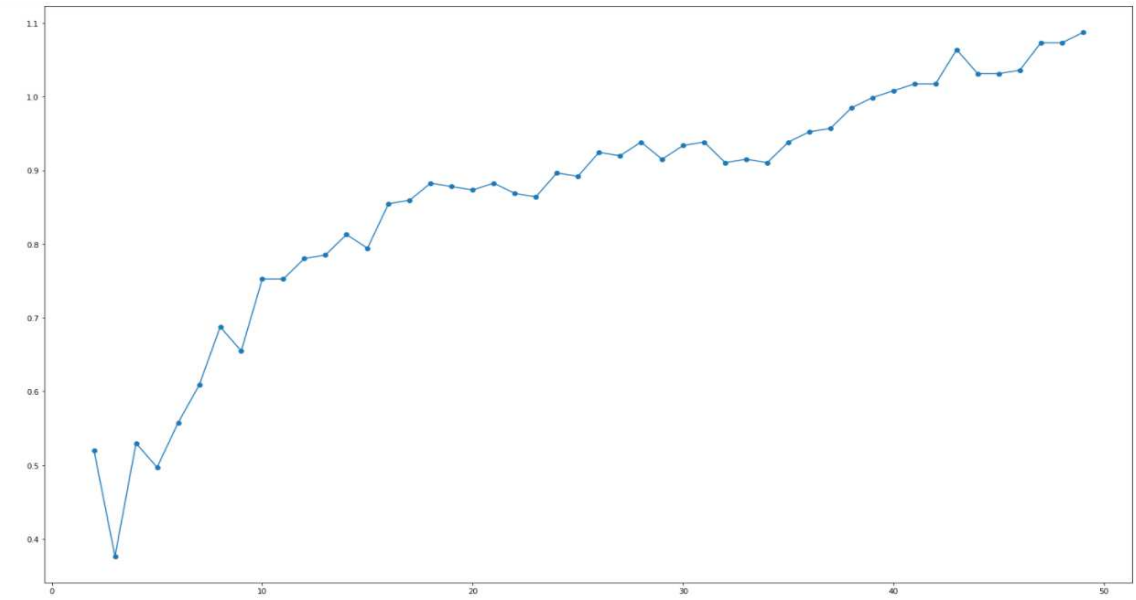


Figure 20 : Pourcentage d'erreur en fonction du k choisi

Le graphique ci-dessous nous montre le pourcentage d'erreur en fonction du k choisi. Comme évoqué précédemment, il faut effectuer le choix d'un K ni trop faible ni trop élevé. Au vu des résultats nous décidons de partir avec $K = 10$ pour attribuer les points non entrainés de la classification ascendante hiérarchique entrainé sur une partition de la base complète.

3. Interprétation des classes crées

Avec les différentes méthodes de classification présentés, nous arrivons à construire trois variables différentes.

Nous supprimons tout de suite celle crée par le DBSCAN car comme vu précédemment, le nombre de classes crées nous convenait mais la répartition n'était pas du tout au rendez-vous avec une sur représentation d'une classe quel que soit les paramètres utilisés dans l'algorithme.

Nous nous intéressons donc aux classes créées avec l'algorithme K-Means et la classification ascendante hiérarchique que nous nommerons par la suite respectivement *cluster_km* et *cluster_agg*.

L'objectif de cette partie est donc d'effectuer des statistiques sur les deux variables crées afin de voir leur explicabilité et de leur donner, si possible, une interprétation et du sens.

i. Le cluster_km

Le cluster_km résulte de la classification du K-means. Cette méthode nous a donc créer une variable à 7 classes que nous allons essayer de décrire dans cette partie. Nous allons pour cela utiliser des diagrammes en violon qui sont constitués de deux graphiques de densité en miroir. La partie interne de chaque violon est une boite à moustache des données, où le bas et le haut de la boîte intérieure se trouvent au premier et au troisième quartile.

Outre qu'il représente de manière concise la nature de la distribution d'une variable numérique, le diagramme en violon est excellent pour visualiser la relation entre une variable numérique et une variable catégorielle en créant un diagramme en violon distinct pour chaque valeur de la variable catégorielle.

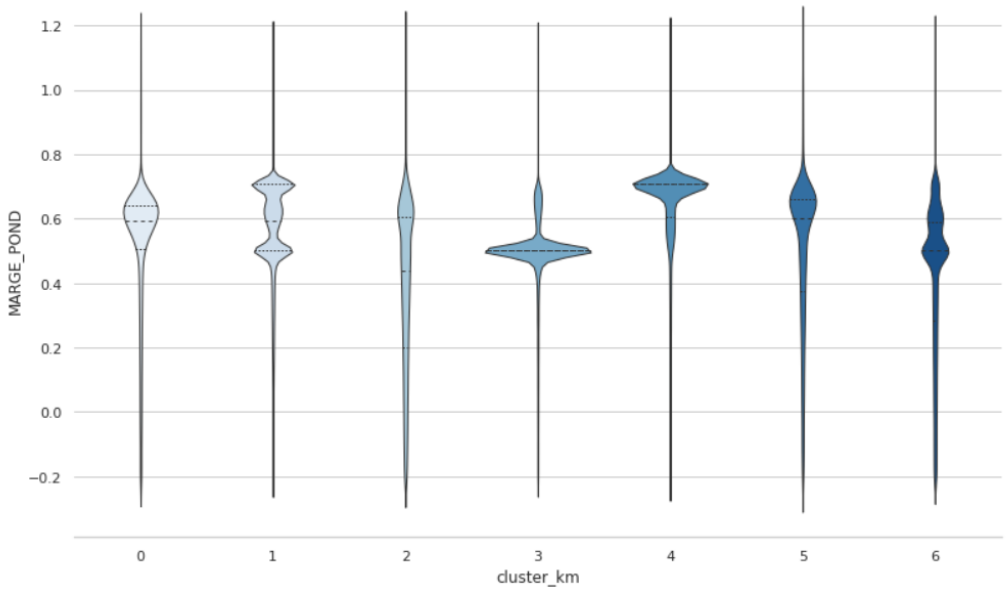


Figure 21 : Marge pondérée en fonction du cluster_km

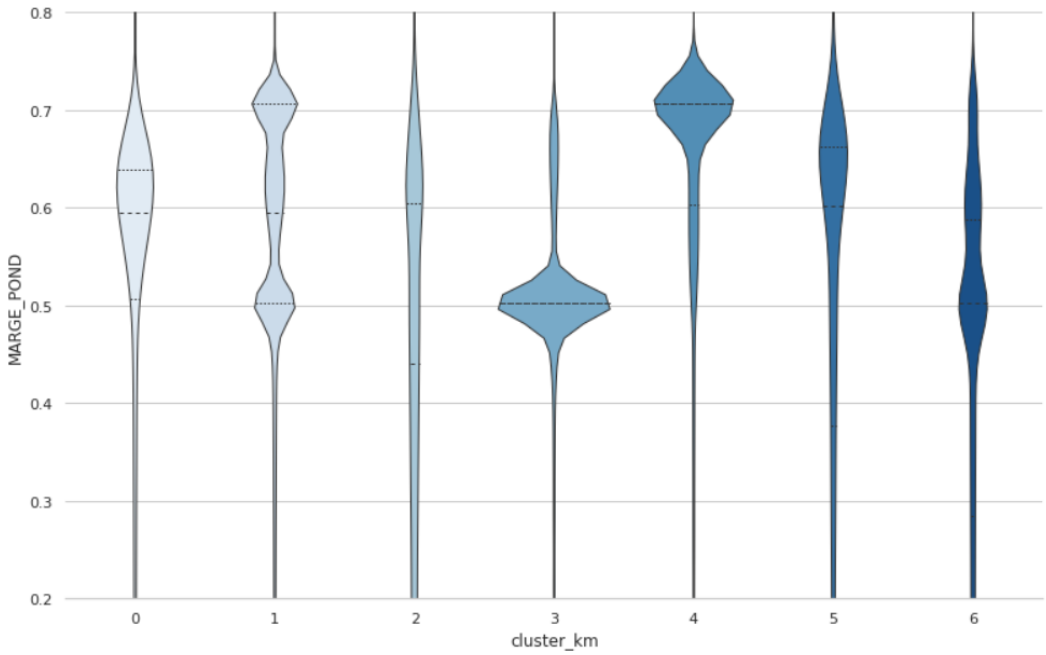


Figure 22 : Zoom de la marge pondérée en fonction du cluster_km

Ces graphiques représentent la distribution de la marge pondérée sur les différentes classes. On observe que la classe 4 est celle ayant les meilleures marges pondérées et est donc la classe la plus rentable. A l'inverse, la classe 3 représente, à première vue, les clients les moins rentables de notre base de données. La marge semble mieux répartie sur les individus des autres classes. A noter que dans toutes les classes, on retrouve des foyers avec des valeurs « extrêmes » positives donc très rentable ou négatives donc en perte.

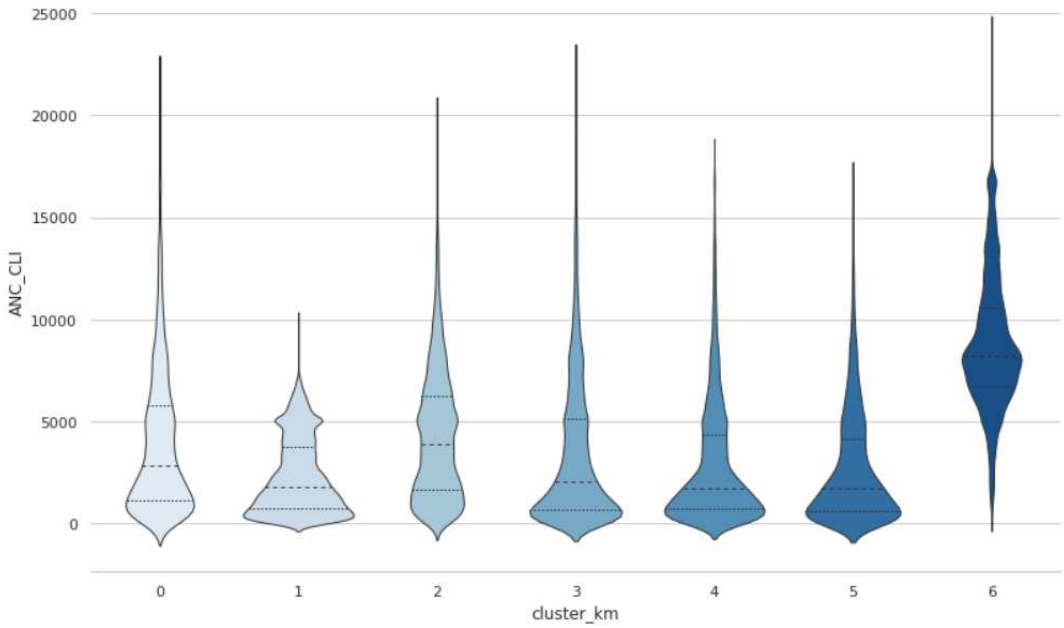


Figure 23 : Ancienneté du client en fonction du cluster_km

Le graphique précédent représente la distribution de l'ancienneté du client en fonction des clusters. Seul la classe 6 est fondamentalement différente avec des clients relativement plus anciens que dans les autres classes

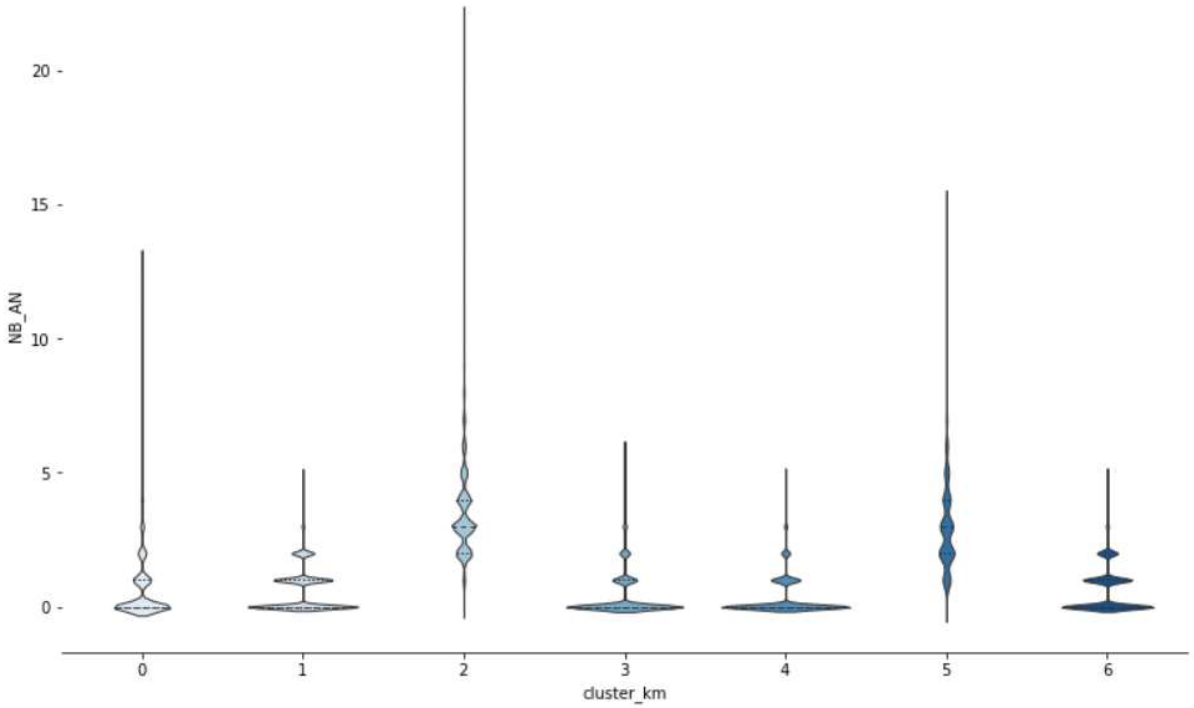


Figure 24 : Nombre d'affaires nouvelles en fonction du cluster_km

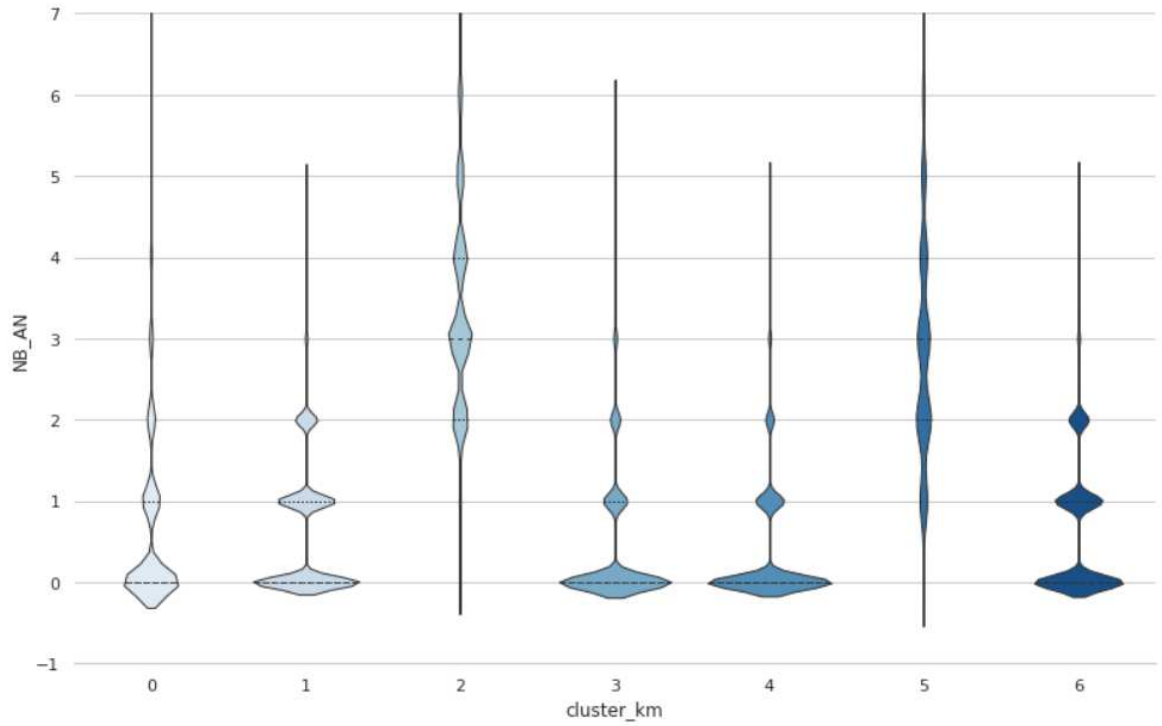


Figure 25 : Zoom sur le nombre d'affaires nouvelles en fonction du cluster_km

Les distributions des affaires nouvelles sont représentées sur les graphiques précédents. Les classes 3 et 4 sont celles ayant fait le moins d'AN. Les classes 0, 2 et 5 ont une répartition plus équilibrée des affaires nouvelles et ce sont également celles qui compte le plus d'affaires nouvelles sur la période observée.

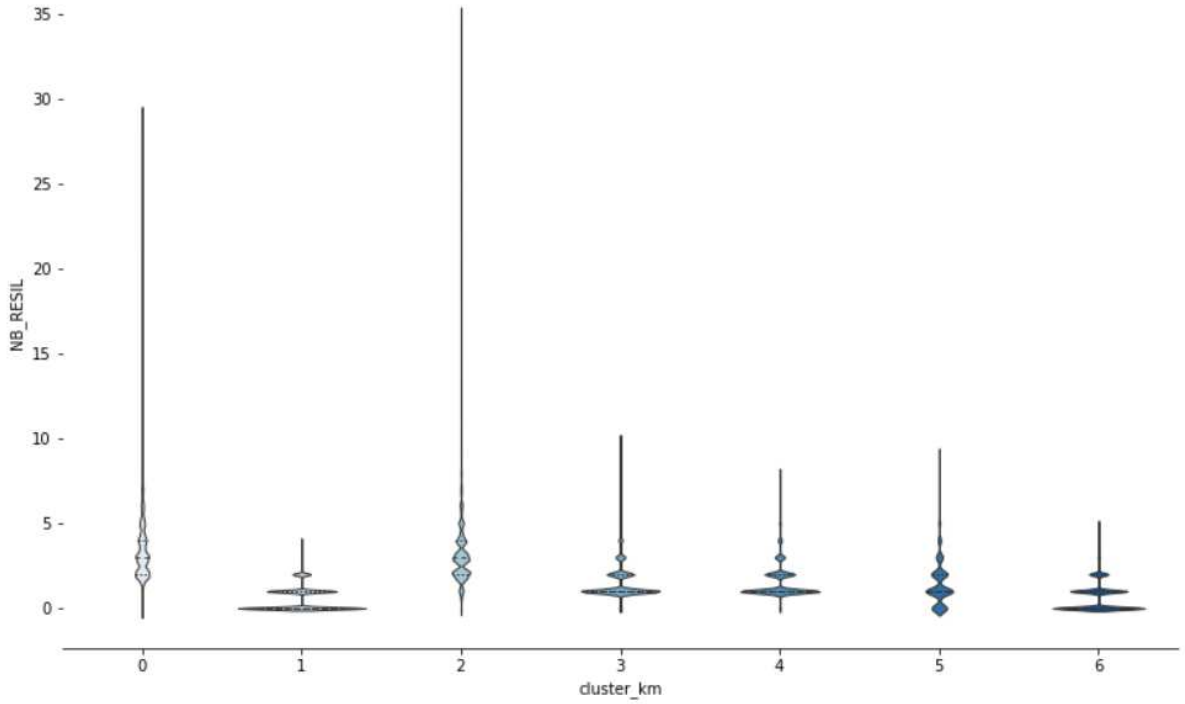


Figure 26 : Nombre de résiliations en fonction du cluster_km

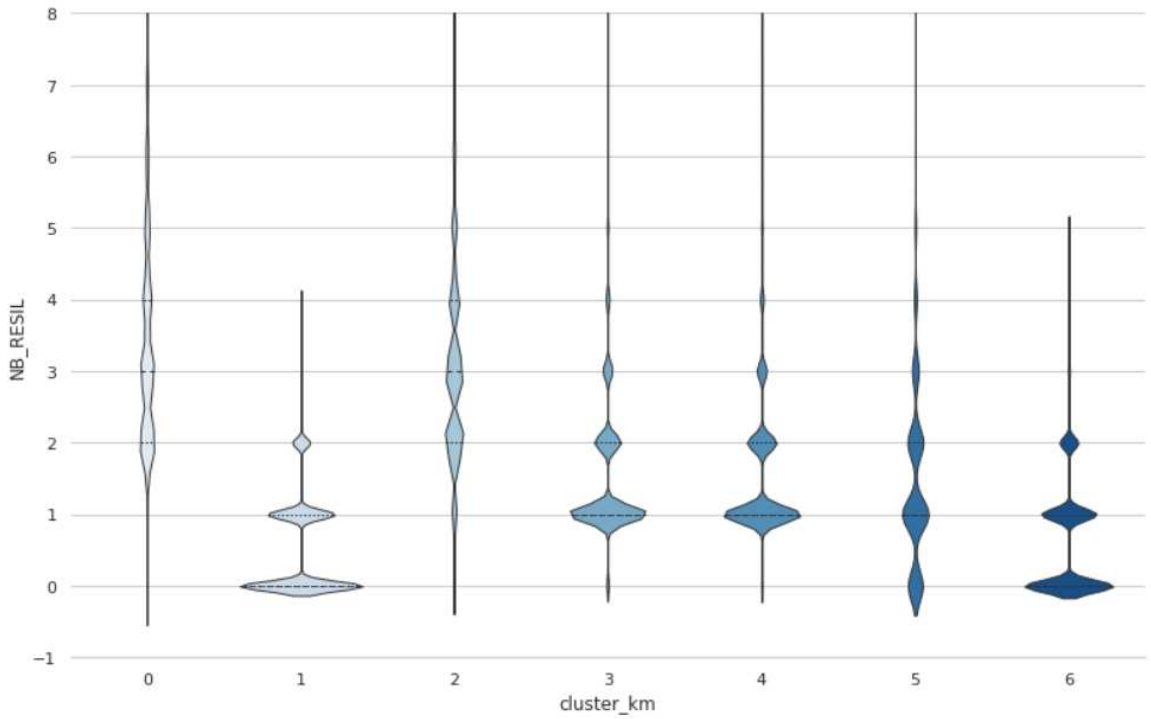


Figure 27 : Zoom sur le nombre de résiliations en fonction du cluster_km

Au tour des résiliations, la classe 1 est celle ayant le moins de résiliations. Les classes 0 et 2 sont celles ayant le plus de résiliations. Pour terminer les classes 3 à 6 ont des résiliations mieux réparties.

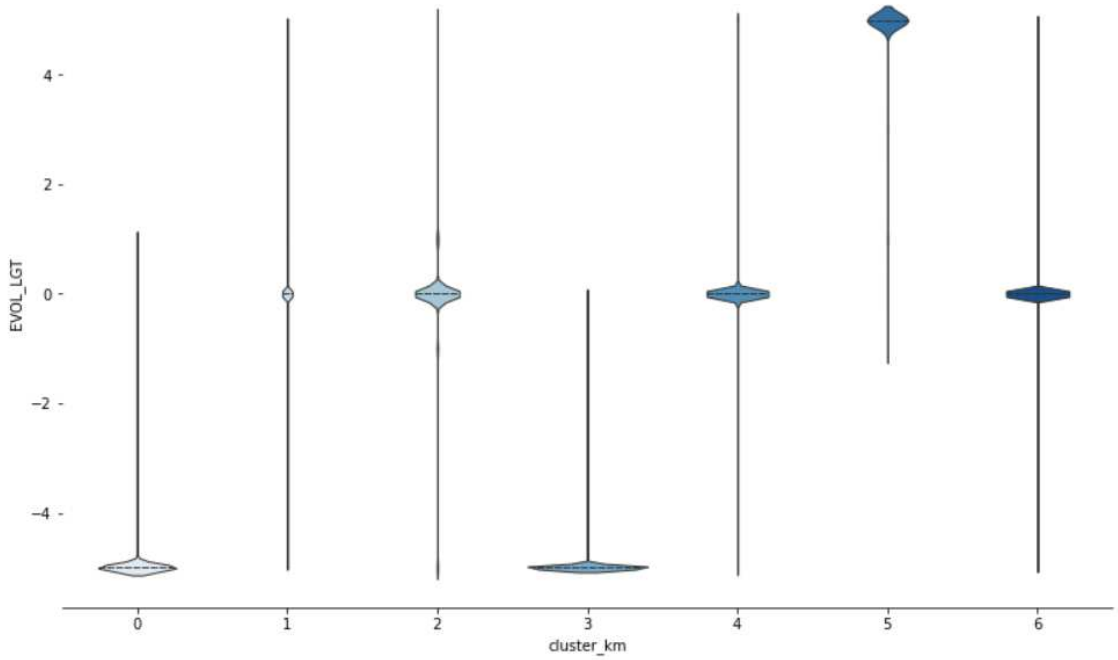


Figure 28 : Evolution du logement en fonction du cluster_km

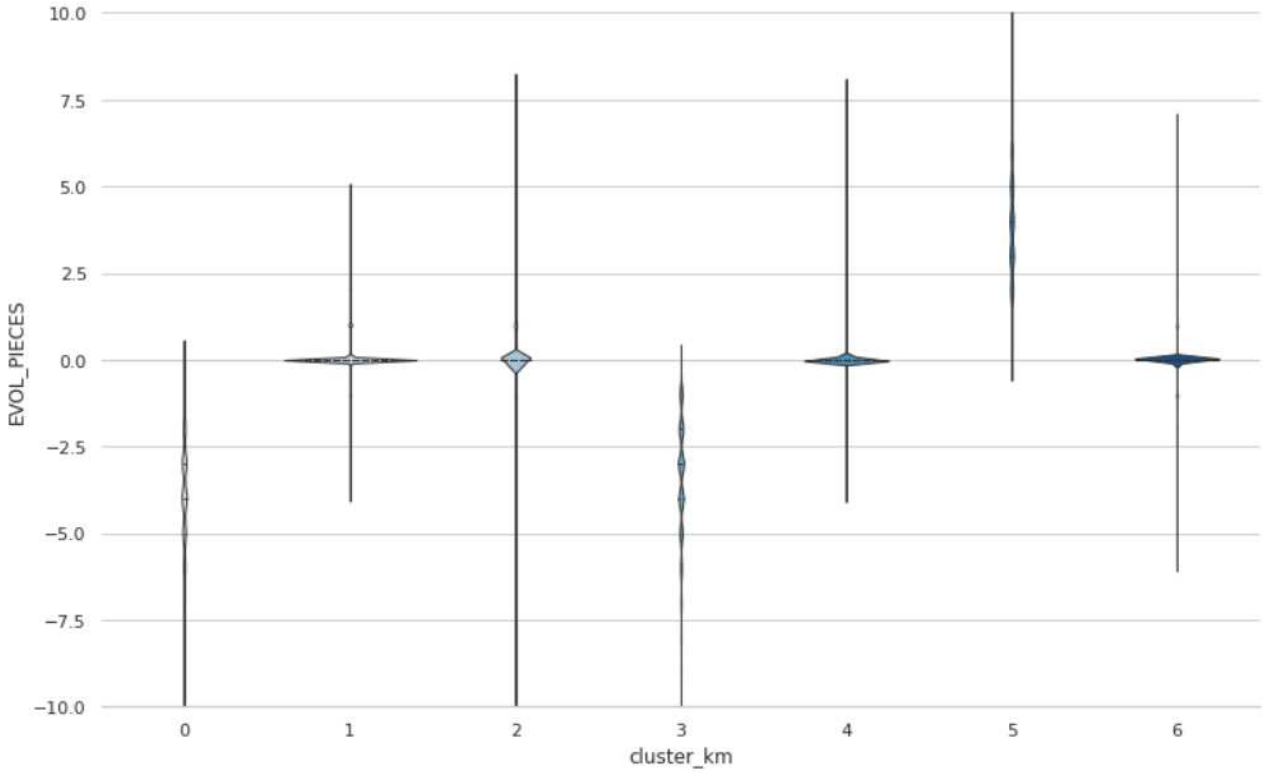


Figure 29 : Evolution du nombre de pièces en fonction du cluster_km

Les deux graphiques qui précèdent représentent l'évolution des critères habitation que sont le logement en résidence principale et le nombre de pièces principales. Certaines dispersions sont liées comme sur la classe 0, la majorité des foyers n'ont plus de contrat habitation à la fin de la période d'observation. L'évolution du logement est donc au plus bas et représente la perte d'information. L'évolution du nombre de pièces est donc négative sur tous les foyers de cette même classe. Même constats sur la classe 3. A l'inverse, la majorité des foyers de la classe 5 n'avaient pas de données habitation au début de la période d'observation et ont souscrit à un contrat habitation. On retrouve donc une évolution positive du nombre de pièces et une montée en gamme du type de logement. Pour les classes 2, 4 et 6, la majorité des foyers ne change pas de logements et le nombre de pièces n'évolue pas non plus.

Pour finir, les deux derniers graphiques qui suivent, représentent la distribution de l'évolution de critères automobiles que sont la classe SRA et l'année de mise en circulation. La classe 0 est majoritairement représenté par une baisse de gamme (qu'elle soit réelle avec une vraie diminution de la classe du véhicule possédé ou artificielle avec la disparition du risque). Les classes 1, 3 et 6 sont des classes où la majorité des foyers n'ont pas de changement de véhicule durant la période observée.

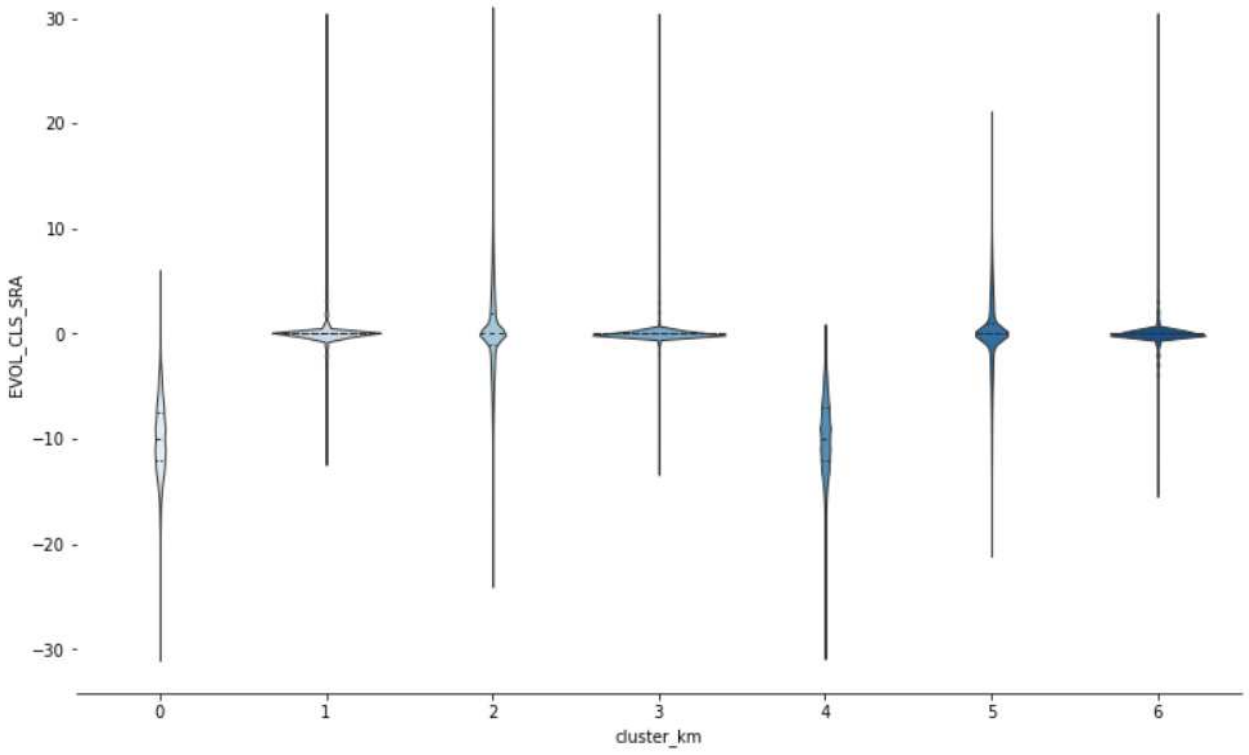


Figure 30 : Evolution de la classe SRA en fonction du cluster_km

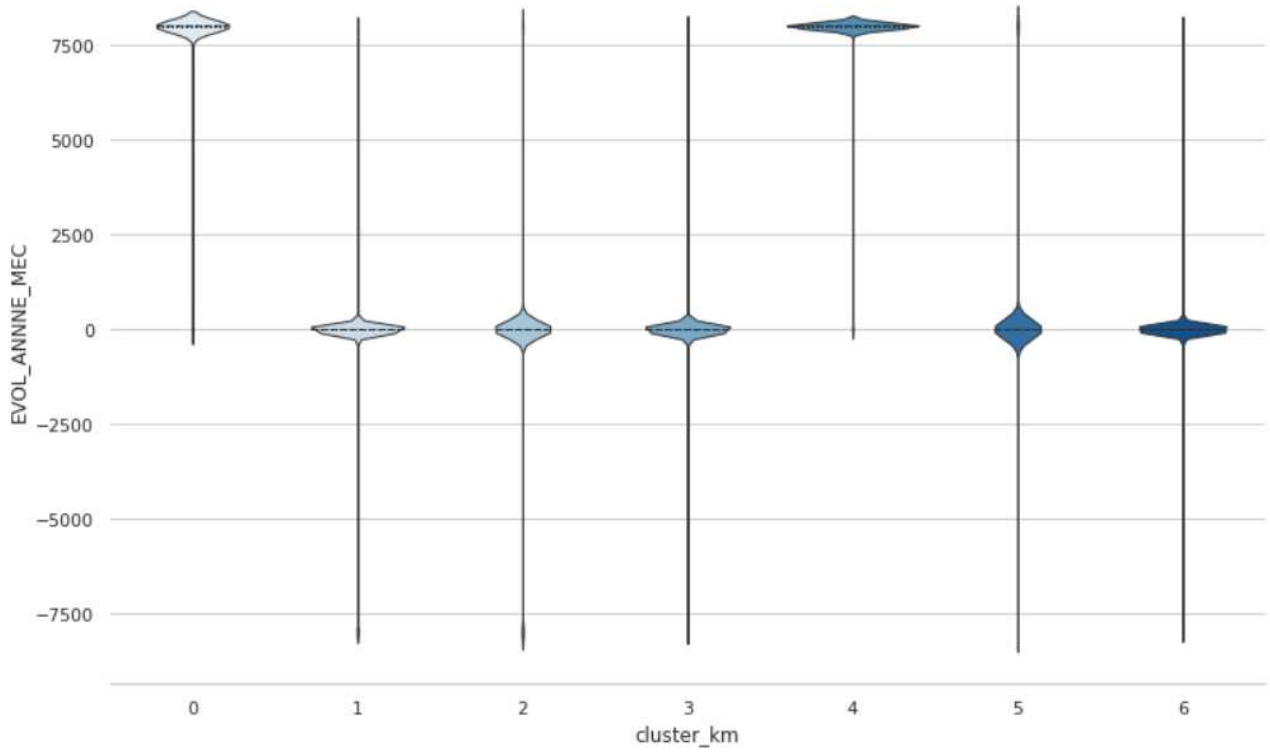


Figure 31 : Evolution de l'année de mise en circulation en fonction du cluster_km

En synthèse, la classe 0 a une rentabilité supérieure à la moyenne globale malgré le fait qu'elle dénombre le plus de résiliations. Ces résiliations font que les foyers de cette classe perdent leur contrat auto et/ou habitation ou du moins baisse de gamme que ce soit au niveau de leur véhicule ou de leur logement.

La classe 1 représente les clients les plus récents avec peu de mouvements dans leurs foyers. Peu ou pas d'AN et de résiliations et peu de changements sur leurs types de logements ou leurs véhicules.

La classe 2 sont plus anciens que la moyenne et ce sont qui présentent le plus de turn over avec un nombre d'AN et de résiliations importants. Peu de changement sur les contrats habitations et auto en leurs possessions.

La classe 3 est la moins rentable. Elle présente peu de mouvements que ce soit en AN ou en résiliations. On retrouve très peu de changements sur le véhicule assurée et principalement une baisse de gamme ou une perte du contrat habitation.

La classe 4 est celle présentant la meilleure rentabilité. Quelques mouvements sont présents sur le foyer et principalement l'apparition d'un contrat automobile.

La classe 5 représente les foyers ayant une évolution à la hausse de leurs nombres de contrats. On observe l'apparition d'un contrat habitation ou du moins une montée en gamme du type de logement et du nombre de pièces.

La classe 6 contient les clients les plus anciens et les plus stables avec peu de résiliations et peu de changement des caractéristiques de leurs contrats auto et habitation quand ils en possèdent.

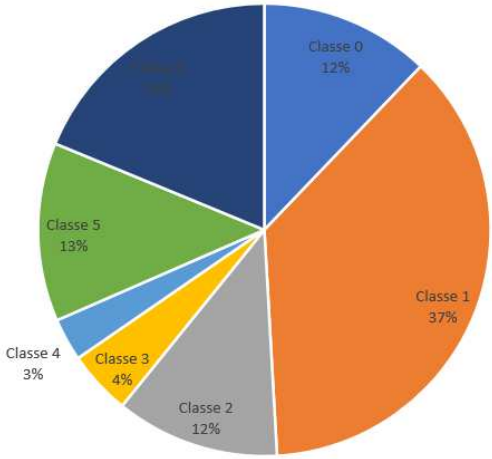


Figure 32 : Proportion des classes pour l'algorithme Kmeans

L'observation de la proportion des foyers dans les différentes classes montre que le portefeuille de Thelem est bien diversifié. Les classes 3 et 4, respectivement celle présentant la moins bonne rentabilité et la meilleure rentabilité, sont les moins représentés. Les autres classes représentent les rentabilités moyennes qui se différencient par les autres critères observés.

ii. Le cluster_agg

De la même manière, le cluster_agg est le résultat de la classification ascendante hiérarchique et une description des classes va être effectuée dans cette partie.

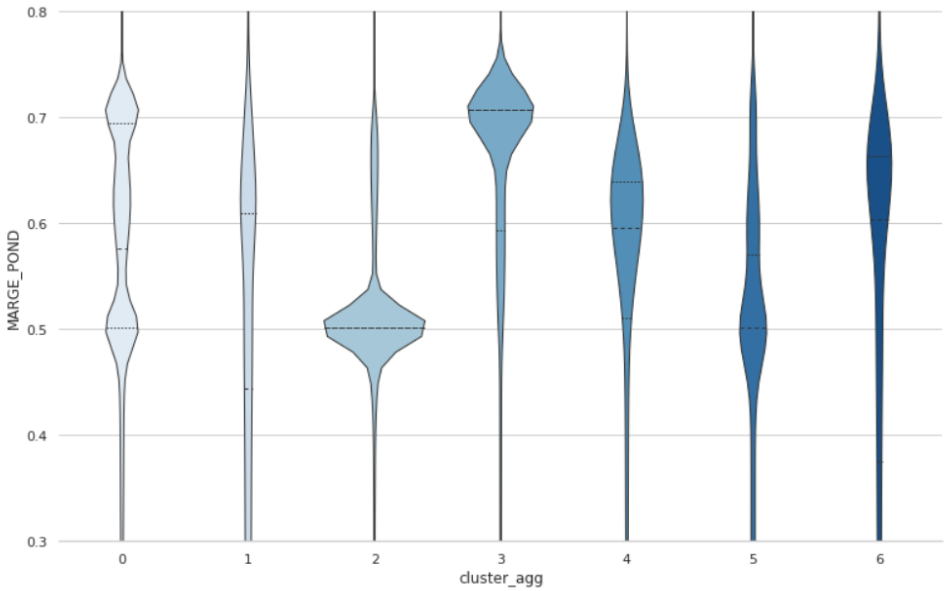


Figure 33 : Zoom de la marge pondérée par cluster_agg

La classe 3 est celle présentant la meilleure rentabilité. A l'inverse, la classe 2 a la rentabilité la plus faible. Les autres classes ont une rentabilité mieux réparti et se différencient par d'autres critères.

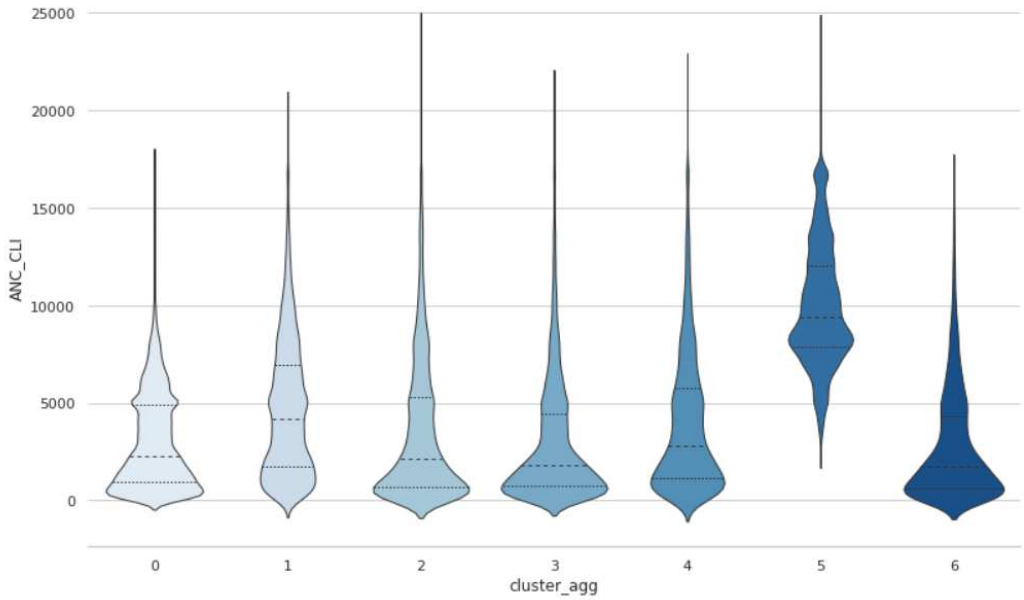


Figure 34 : Ancienneté du client par cluster_agg

La classe 5 regroupe les foyers les plus anciens de la base de données quand les classes 0, 2, 3 et 6 ont une majorité de foyers récents.

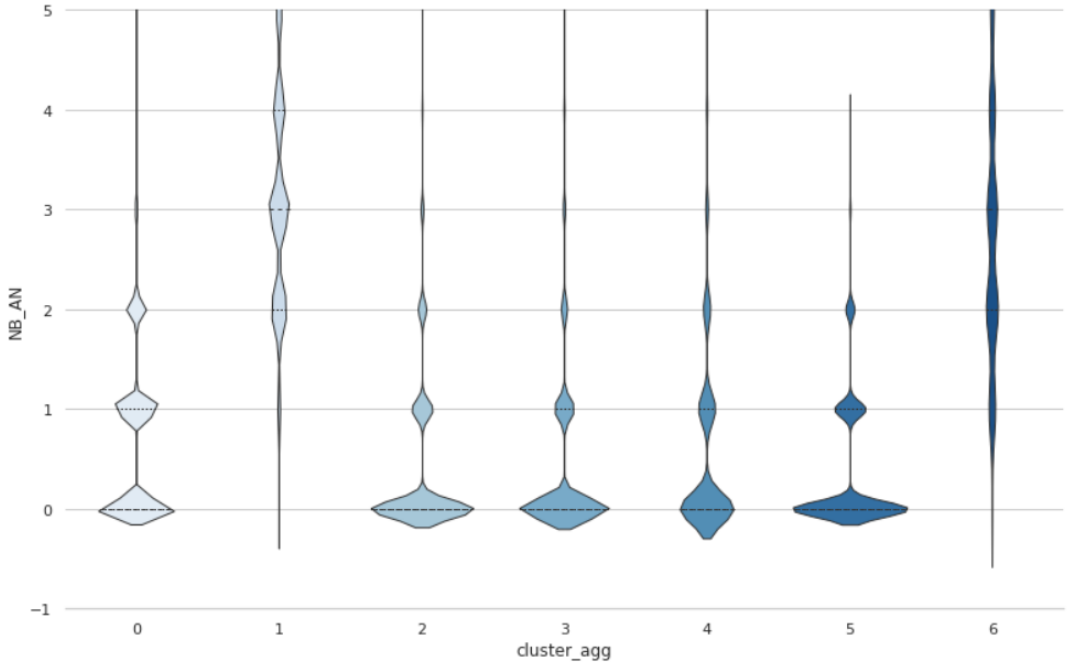


Figure 35 : Nombre d'affaires nouvelles par cluster_agg

Les classes 1 et 6 sont semblables sur les affaires nouvelles au cours de la période d'observation à savoir que ces deux classes ont quasiment au minimum une affaire nouvelle. Les autres classes regroupent les foyers ayant pour la plupart effectué moins de 2 affaires nouvelles sur la période.

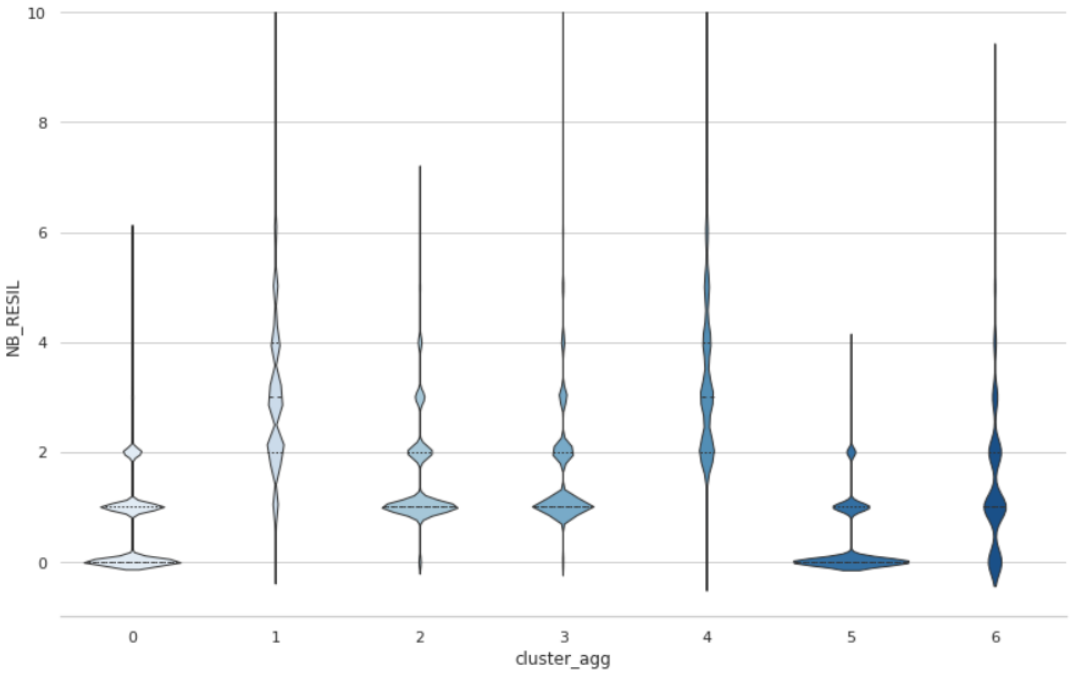


Figure 36 : Nombre de résiliations par cluster_agg

De même pour les résiliations, les classes 0 et 5 sont les classes ou la majorité des foyers ne résilie pas. Contrairement aux autres classes où il y a une grande majorité de foyers qui a au moins une résiliation.

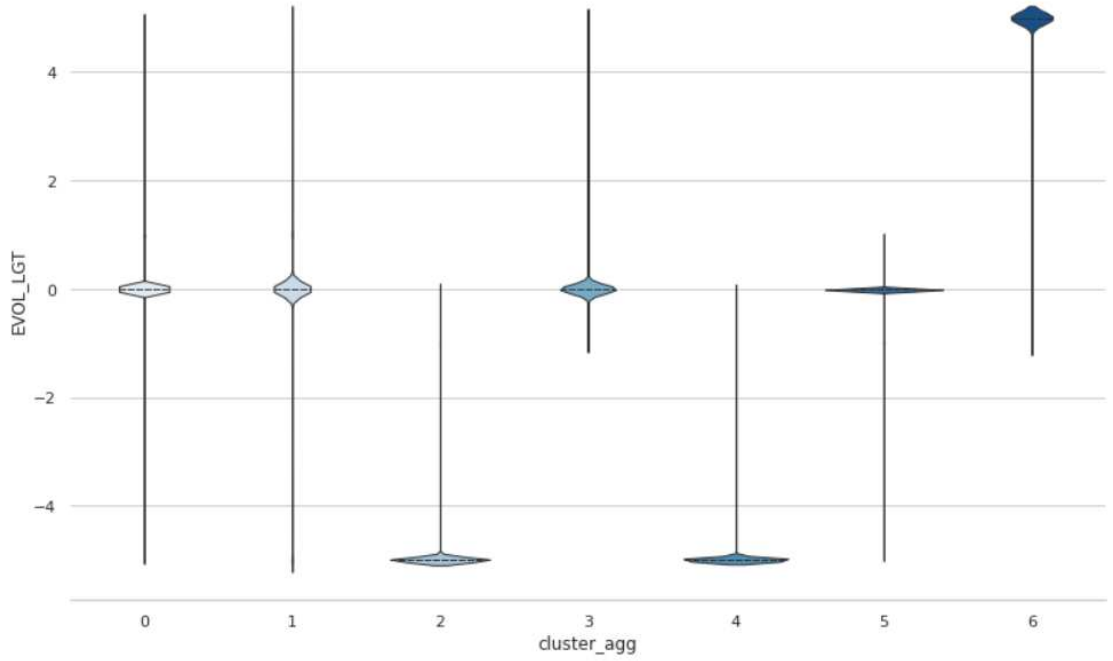


Figure 37 : Evolution du logement par cluster_agg

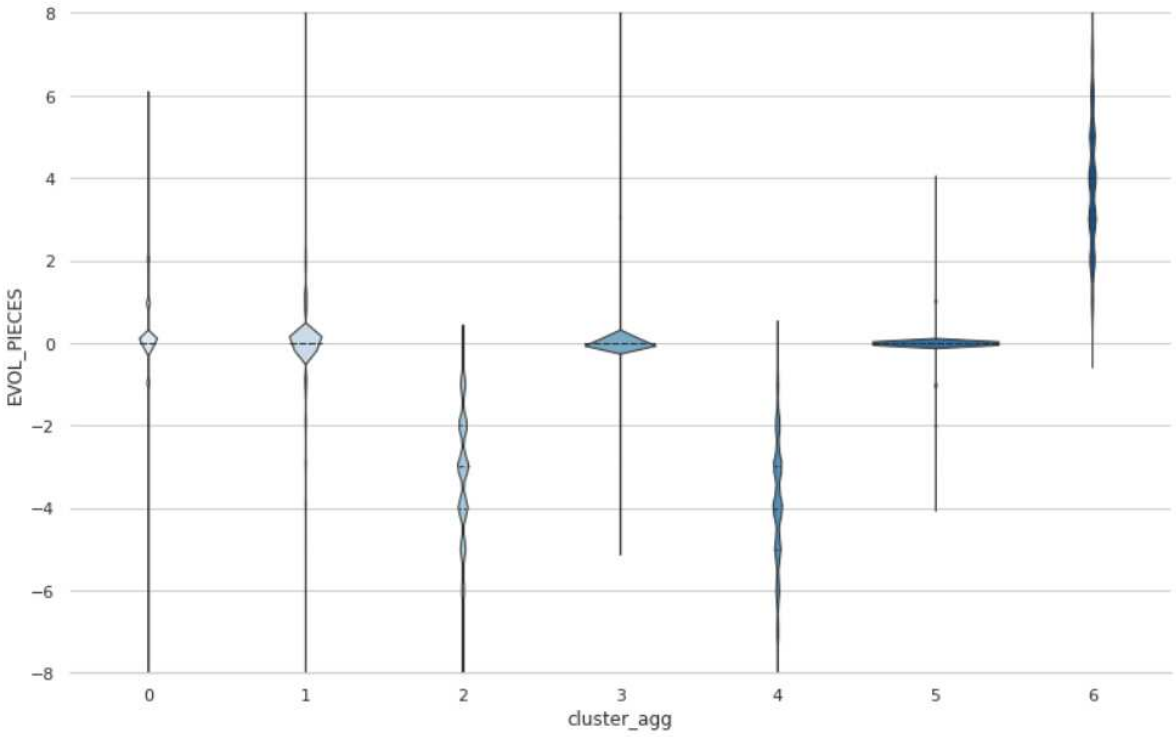


Figure 38 : Evolution du nombre de pièces par cluster_agg

Concernant l'évolution du logement, les classes 2 et 4 représentent majoritairement les classes ayant perdu leur contrat habitation durant la période d'observation, les classes 0, 1, 3 et 5 regroupent les classes n'ayant pas d'évolution de leur habitation principale. Pour finir la classe 6 est composée de foyers ayant souscrit à un contrat habitation.

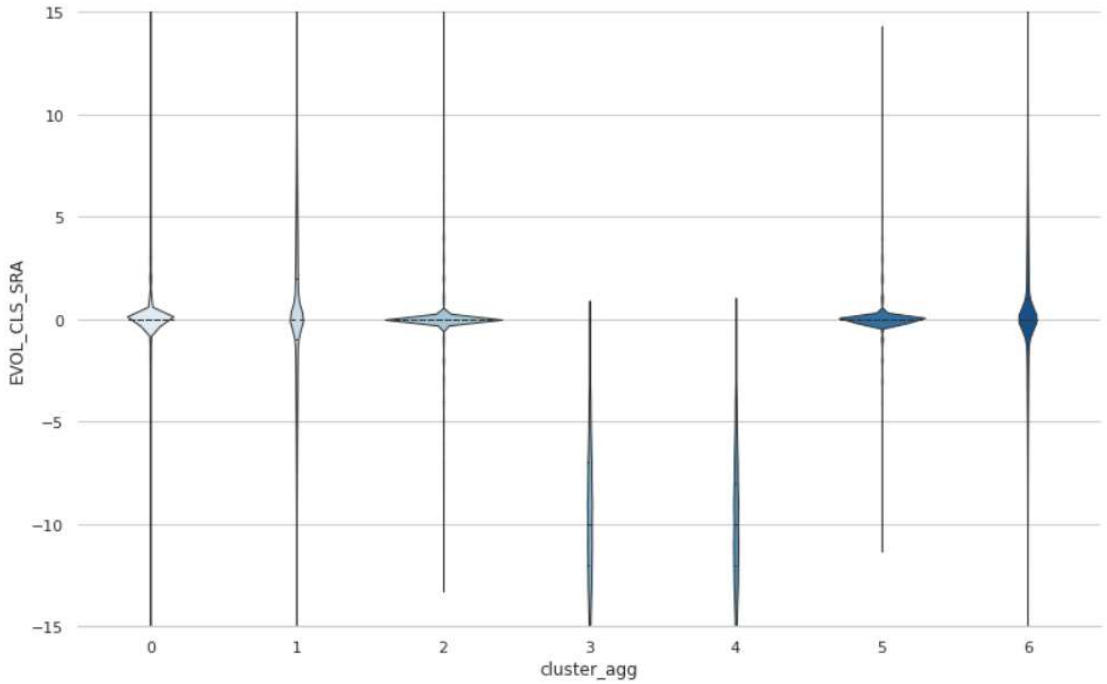


Figure 39 : Evolution de la classe SRA par cluster_agg

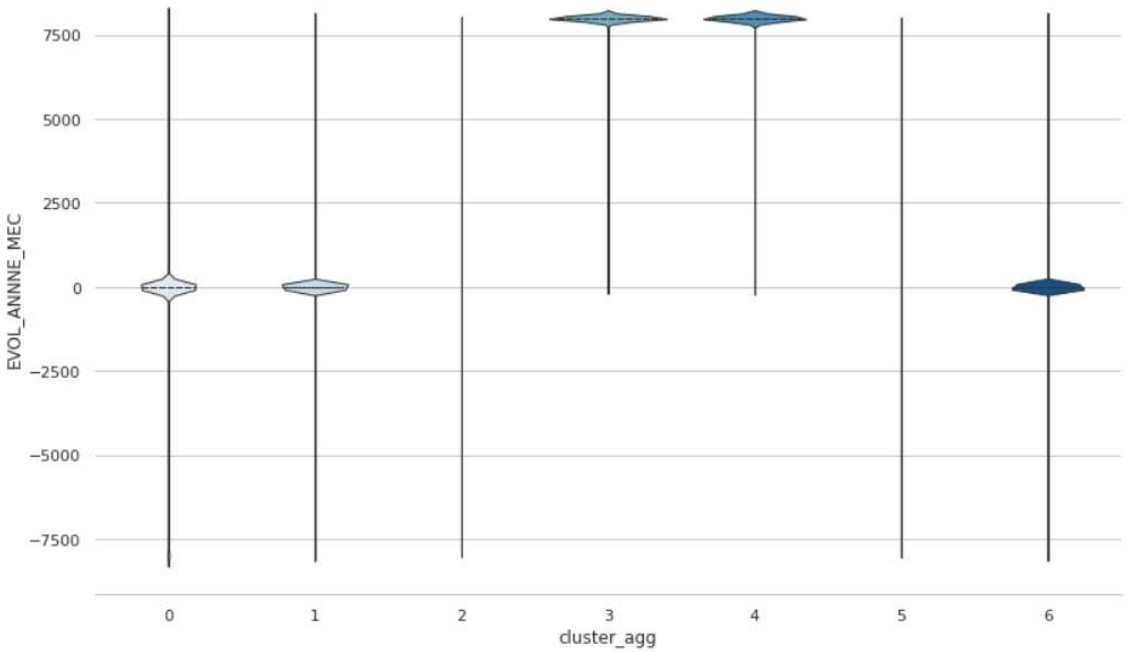


Figure 40 : Evolution de l'année de mise en circulation par cluster_agg

Pour l'évolution du véhicule, les classes 3 et 4 sont majoritairement des foyers ayant un nouveau contrat automobile. Les classes 2 et 5 sont principalement des foyers n'ayant pas de mouvements sur leurs véhicules. Pour finir les classes 0, 1 et 6 représentent les foyers ayant eu un changement de contrat automobile qui fait évoluer les caractéristiques du véhicule.

En synthèse, la classe 0 peut être considéré comme la classe moyenne. Tous ses indicateurs sont moyens, elle présente quelques AN et quelques résiliations et quelques changements sur les contrats auto et habitation. Aucune caractéristique ne prédomine cette classe.

La classe 1 a une rentabilité basse et présente également des foyers en perte. Les foyers sont plus anciens que la moyenne. Ils ont effectué au moins une affaire nouvelle et ont plus de résiliations que la moyenne.

La classe 2 est la moins rentable avec très peu d'AN et au moins une résiliation qui pour la plupart concernera le contrat habitation avec une perte d'information entre les deux dates observées. A l'inverse, on observe peu de changement sur le contrat auto.

La classe 3 est la plus rentable. On constate la souscription d'un contrat auto avec l'apparition de l'information lié au véhicule.

La classe 4 est caractérisée par la souscription d'un contrat auto et la résiliation ou la baisse de gamme d'un contrat habitation.

La classe 5 représente les foyers les plus anciens. Ils ont la caractéristique de rester stable. Ils ont très peu d'an et très peu de résiliations. Au niveau du logement et du véhicule, très peu de changements à constater également.

La classe 6 est caractérisé par un turn over plus important que la moyenne avec la souscription d'un contrat habitation.

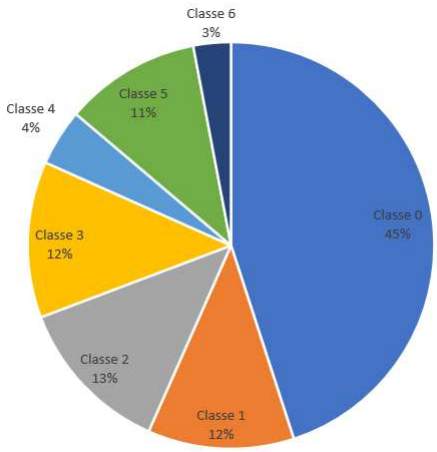


Figure 41 : Proportion des classes pour l'algorithme CAH

De la même manière que pour le cluster_km, l'observation de la répartition des foyers par classe dans l'algorithme de classification ascendante hiérarchique indique que la classe 0 jugé comme la classe moyenne est majoritairement présente. Les classes 4 et 6 avec un turn over plus marqué sont peu représentés.

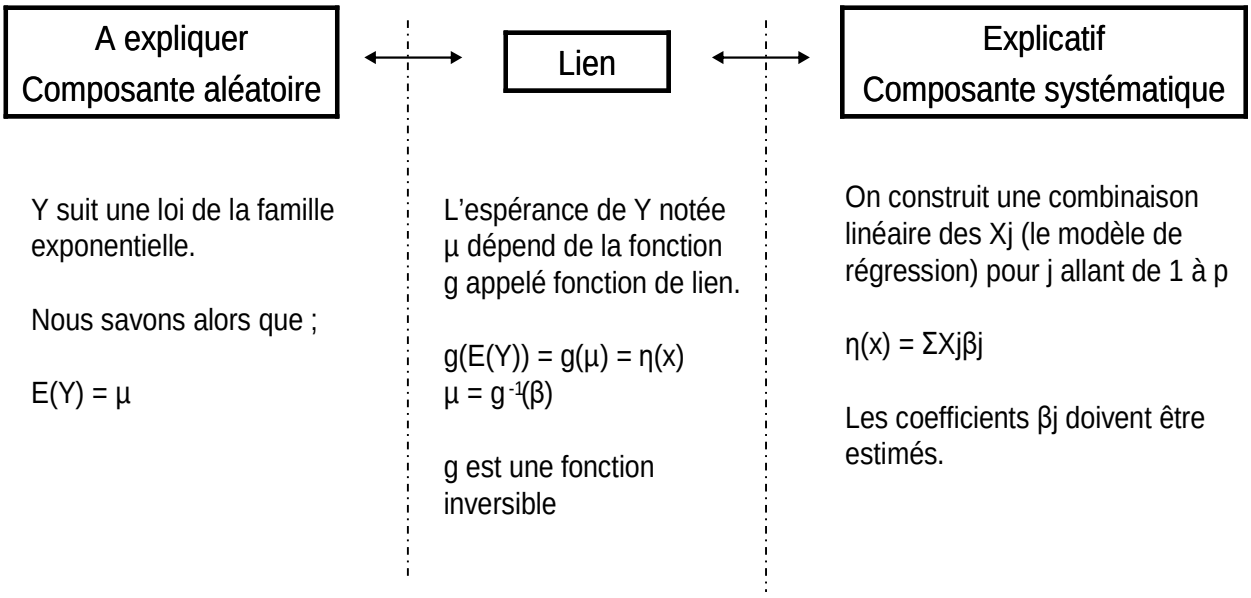
III. Un usage envisagé : test sur le tarif auto

Après avoir donné une description des différentes classes créées par les deux méthodes conservées, la question est de savoir quelles seront leurs utilités. Le premier usage envisagé est l'intégration de cette valeur comme un critère de tarif. On l'utilisera dans un premier temps comme une surcouche du tarif existant.

Pour tester les *clusters* créés précédemment dans le tarif, nous allons utiliser le logiciel de tarification « Addactis Pricing » qui se base sur les modèles linéaires généralisés (GLM).

1. Quelques éléments théoriques

Un modèle linéaire généralisé est un modèle qui tente de relier des variables explicatives X_1, X_2, ..., X_p à une variable à expliquer Y. C'est un modèle défini par une combinaison linéaire de variables, une fonction de lien et une loi de distribution de la variable à régresser.



Finalement, un GLM est défini comme suit : $f(Y) = \alpha + X_1.\beta_1 + X_1.\beta_1 + \ldots + X_1.\beta_1 + \varepsilon$

Pour construire ce modèle, il faut choisir la loi de Y dans la famille exponentielle et choisir une fonction de lien inversible. Et pour l'utiliser, il faut estimer les paramètres $\beta 1, \beta 2, \ldots, \beta p$.

Habituellement, on décompose la variable aléatoire qu'est la charge sinistre en deux phénomènes distincts indépendants à savoir la fréquence des sinistres et le coût des sinistres. Dans ce cas de figures, les modèles utilisés seront le modèle de Poisson pour l'estimation de fréquence et le modèle Gamma pour l'estimation du coût moyen.

Dans notre cas, la variable cluster sera utilisé comme une surcouche du tarif existant dans un premier temps, c'est pourquoi on décide de modéliser directement la prime pure avec le modèle de Tweedie qui possède une distribution autorisant les valeurs nulles.

Tweedie (1984) a suggéré la famille suivante :

$$f(y;\mu,\emptyset) = A(y,\emptyset)e^{\left\{\frac{1}{\emptyset}[y\theta(\mu)-k(\theta(\mu))]\right\}};$$

avec :

$$\theta(\mu) = \begin{cases} \dfrac{\mu^{1-\gamma}}{1-\gamma}, \gamma \neq 1 \\ \log(\mu) \end{cases} \quad et \quad k(\theta(\mu)) = \begin{cases} \dfrac{\mu^{2-\gamma}}{2-\gamma}, \gamma \neq 2 \\ \log(\mu) \end{cases}$$

Avec la famille Tweedie, la loi de Y est une loi de Poisson composée avec des sauts de Gamma. Cette loi présente une mesure de Dirac en zéro et une fonction de densité continue pour les valeurs supérieures à zéro.

$$Y \sim PC\left(\mu^{2-\gamma}\emptyset(2-\gamma), G\left(-\frac{2-\gamma}{\phi(1-\gamma)}, \phi(2-\gamma)\mu^{\gamma-1}\right)\right) \; avec \; \gamma \in [1,2]$$

et,

$$E(y) = \mu$$

$$Var(y) = \phi\mu^{\gamma}$$

Avec la forme de la fonction de variance obtenu, on retrouve le modèle de Poisson quand $\gamma \to 1$ et une loi Gamma quand $\gamma \to 2$.

Pour obtenir la significativité des variables, nous utiliserons le test de « type III » qui sert à tester la significativité des variables en comparant le modèle contenant toutes les variables sauf celles testées avec le modèle contenant toutes les variables. Une valeur inférieure à 5% confirmera que la variable testée est significative et qu'elle a donc un impact sur le modèle.

Une fois cette significativité confirmée, on observera la significativité des modalités de la variable. En effet, si les modalités de la variable sont non significatives, la variable n'aura donc aucun intérêt tarifaire.

L'impact tarifaire sera testé sur les principales variables du tarif Auto à savoir :

- Responsabilité Civile Matérielle
- Bris de Glaces
- Vol
- Dommages Tout Accidents

En plus de la significativité de la variable, nous observerons l'AIC, le BIC et l'indice de Gini pour déterminer ce qu'apporte l'ajout de cette variable. Améliore-t-elle ou dégrade-t-elle la qualité du modèle ?

Le critère d'information d'Akaike (en anglais *Akaike information criterion* ou **AIC**) est une mesure de la qualité d'un modèle statistique. Il est défini par :

$$AIC = 2k - 2\log \tilde{L}$$

où $\tilde{L}$ est la vraisemblance maximisée et k le nombre de paramètres dans le modèle. Avec ce critère, la déviance du modèle $-2\ log(L)$ est pénalisée par 2 fois le nombre de paramètres.

Le critère d'information bayésien (en anglais *Bayesian Information Criterion* ou **BIC**) est un critère d'information dérivé de l'AIC. Il est défini par :

$$BIC = k\log(n) - 2\log \tilde{L}$$

Il est plus parcimonieux que le critère AIC puisqu'il pénalise plus le nombre de variables présentent dans le modèle.

Ripley (2003) souligne que l'AIC a été introduit pour retenir des variables pertinentes lors de prévisions, et que le critère BIC vise la sélection de variables statistiquement significative dans le modèle.

L'indice de Gini est un indicateur synthétique permettant de rendre compte du niveau d'inégalité pour une variable et sur une population donnée. Il varie entre 0 (égalité parfaite) et 1 (inégalité extrême). Entre 0 et 1, l'inégalité est d'autant plus forte que l'indice de Gini est élevé. Il est égal à 0 dans une situation d'inégalité parfaite où la variable prend une valeur identique sur l'ensemble de la population. A l'autre extrême, il est égal à 1 dans la situation la plus inégalitaire possible, où la variable vaut 0 sur toute la population à l'exception d'un seul individu.

Pour tester la variable *cluster* comme une surcouche du tarif, on imposera en contrainte toutes les valeurs actuelles des coefficients de prime pure de toutes les modalités des variables utilisées pour le tarif technique automobile actuellement en vigueur. Cela revient à imposer les coefficients tarifaires existants prenant en compte les différentes aménagements tarifaires depuis le lancement du produit.

Le test sera effectué sur le *cluster* crée avec le k-means (cluster_km) et celui crée avec la classification ascendante hiérarchique avec l'implémentation python *agglomérative clustering* (cluster_agg).

2. Modélisation tarif des garanties

Quel que soit la garantie traitée, la méthode sera la même pour déterminer l'apport de la variable. Nous utiliserons les critères mentionnés précédemment (AIC, BIC et indice de Gini) pour comparer la qualité du modèle référent qui sera le modèle de tarification aujourd'hui en production avec les modèles ou on ajoute en critère libre le *cluster_km* et le *cluster_agg*. Une fois cette amélioration de la qualité du modèle constaté, on s'intéressera à l'impact tarifaire des classes en elle-même.

i. Garantie Responsabilité Civile Matérielle (RCM)

La première garantie observée est la RCM. Tous les modèles testés convergents et les variables de classification sont significatifs.

RCM	AIC	BIC	Coefficient de Gini
Modèle référent	2 149 197,10	2 149 122,47	0,0964
Modèle cluster km	2 148 317,78	2 148 419,27	0,1430
Modèle cluster agg	2 148 178,40	2 148 279,89	0,1392
Cluster km / Référent	-0,04%	-0,03%	48,34%
Cluster agg / Référent	-0,05%	-0,04%	44,40%
Cluster agg / Cluster km	-0,01%	-0,01%	-2,66%

Tableau 12 : Statistiques des modèles tarifaires RCM

Au vu des statistiques des modèles, les deux classifications créées améliorent la qualité du modèle référent que ce soit le *cluster_km* ou le *cluster_agg* avec une nette amélioration du coefficient de Gini.

Lorsqu'on compare les deux nouveaux modèles, on constate un coefficient de Gini supérieur pour le *cluster_km* et des AIC et des BIC relativement proches.

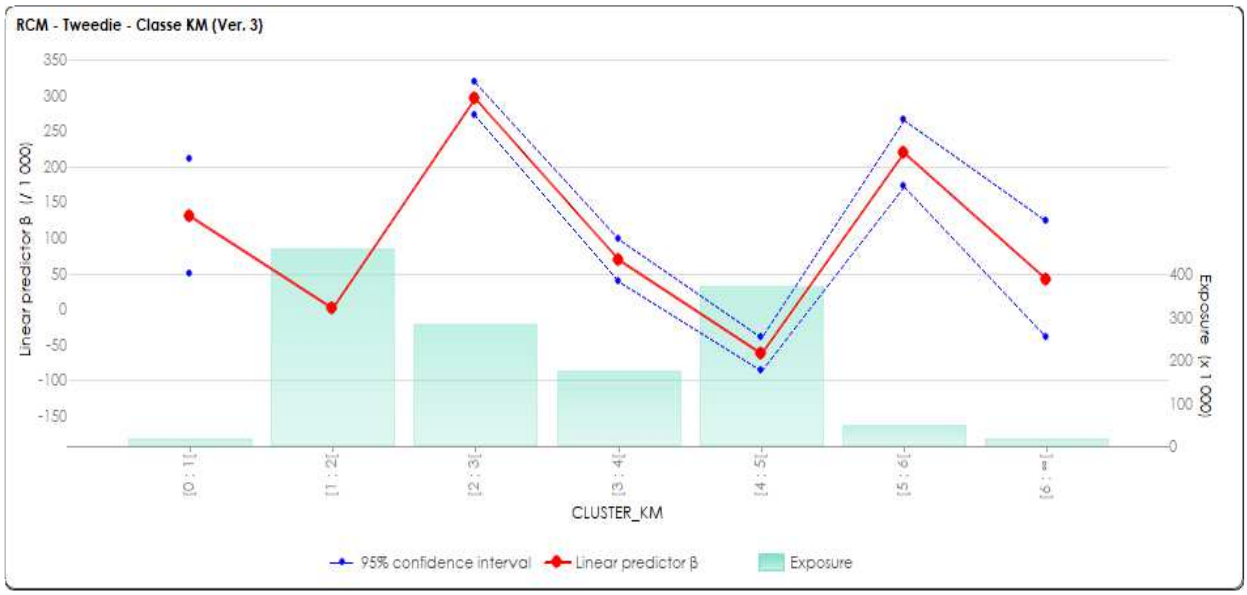


Figure 42 : impact tarifaire RCM par cluster_km

Concernant la garantie RCM, le cluster_km 4 qui présente la meilleure rentabilité aurait le tarif le moins élevé quand la classe 2 représentant les clients les plus anciens avec du turn over serait la plus majorée.

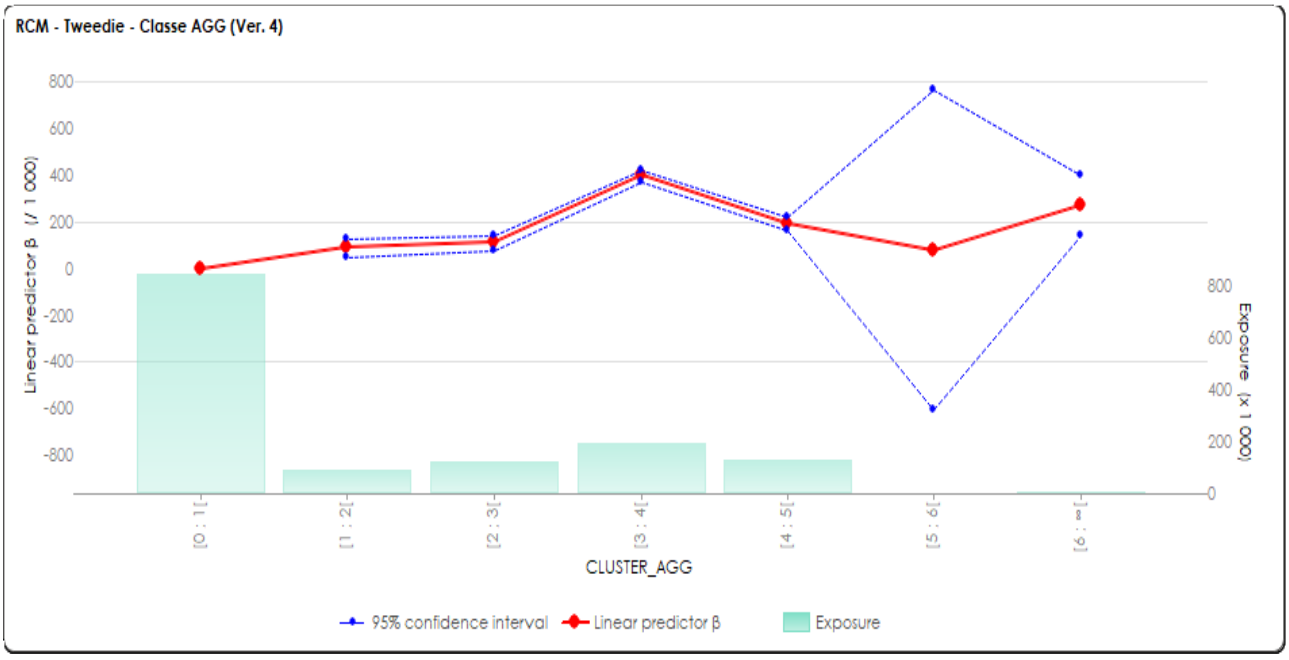


Figure 43 : Impact tarifaire RCM par cluster_agg

Les résultats bruts montrent une variable significative mais les modalités ne sont pas toutes significatives. Après regroupements, on obtient les impacts suivants :

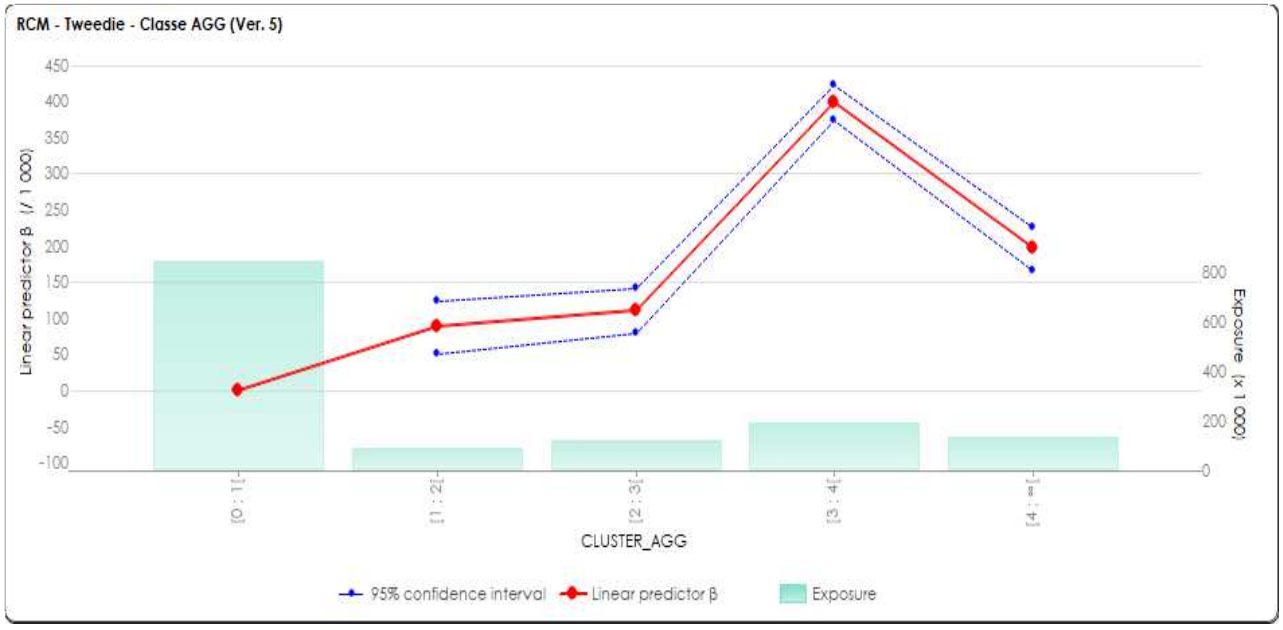


Figure 44 : Impact tarifaire RCM par cluster_agg après regroupement

Le cluster_agg 3, la classe la plus rentable, est la classe présentant le plus de risque sur la garantie RCM.

ii. Garantie Bris de Glaces (BDG)

Poursuivons avec la garantie BDG

BDG	AIC	BIC	Coefficient de Gini
Modèle référent	1 774 583,24	1 774 608,46	0,2708
Modèle cluster km	1 773 324,48	1 773 425,35	0,3034
Modèle cluster agg	1 773 315,23	1 773 416,11	0,3029
Cluster km / Référent	-0,07%	-0,07%	12,04%
Cluster agg / Référent	-0,07%	-0,07%	11,85%
Cluster agg / Cluster km	0,00%	0,00%	-0,16%

Tableau 13 : Statistiques des modèles de tarification BDG

Au vu des indicateurs, les variables de classifications crées n'améliore que de peu la qualité du modèle référent. Si on les compare entre eux, les 3 indicateurs montrent des qualités équivalentes entre le cluster créé avec le K-means et celui créé avec l'agglomérative clustering.

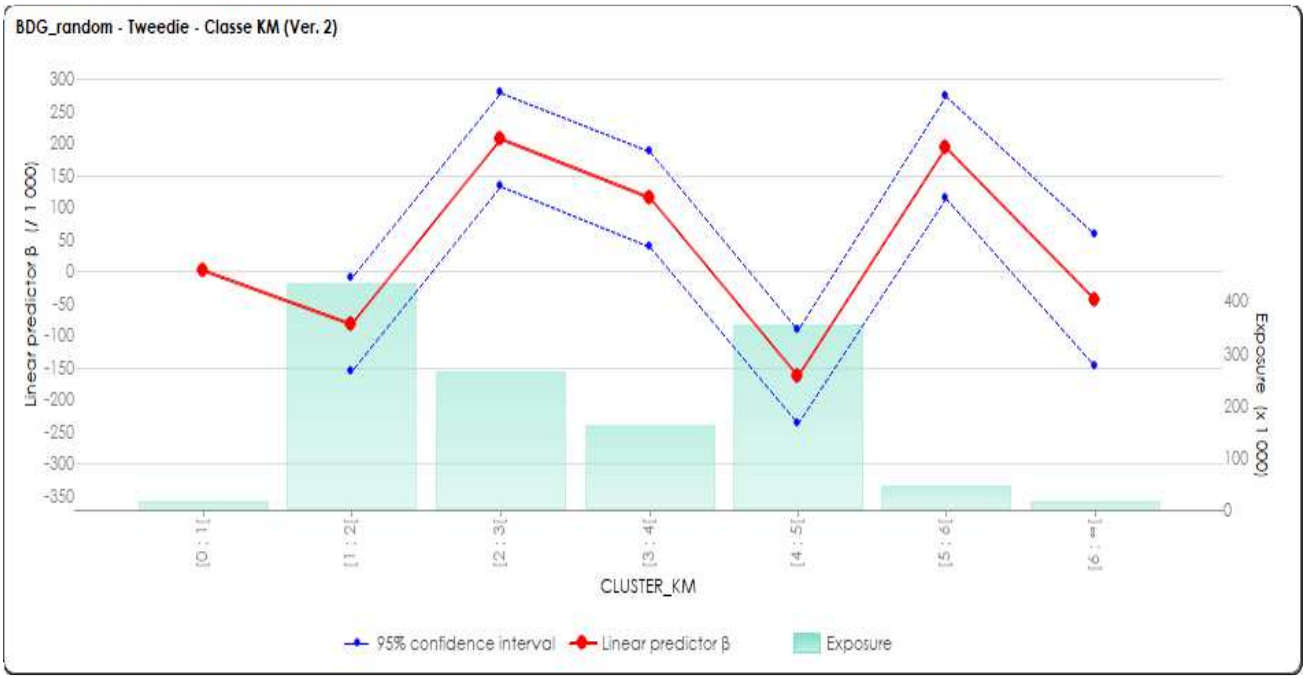


Figure 45 : Impact tarifaire BDG par cluster_km

Concernant le cluster_km le constat est sensiblement le même que sur la RCM, à savoir que le cluster_km 4 qui présente la meilleure rentabilité aurait le tarif le moins élevé. A l'inverse, les foyers de la classe 5 considérés comme les plus anciens sont les plus risqués.

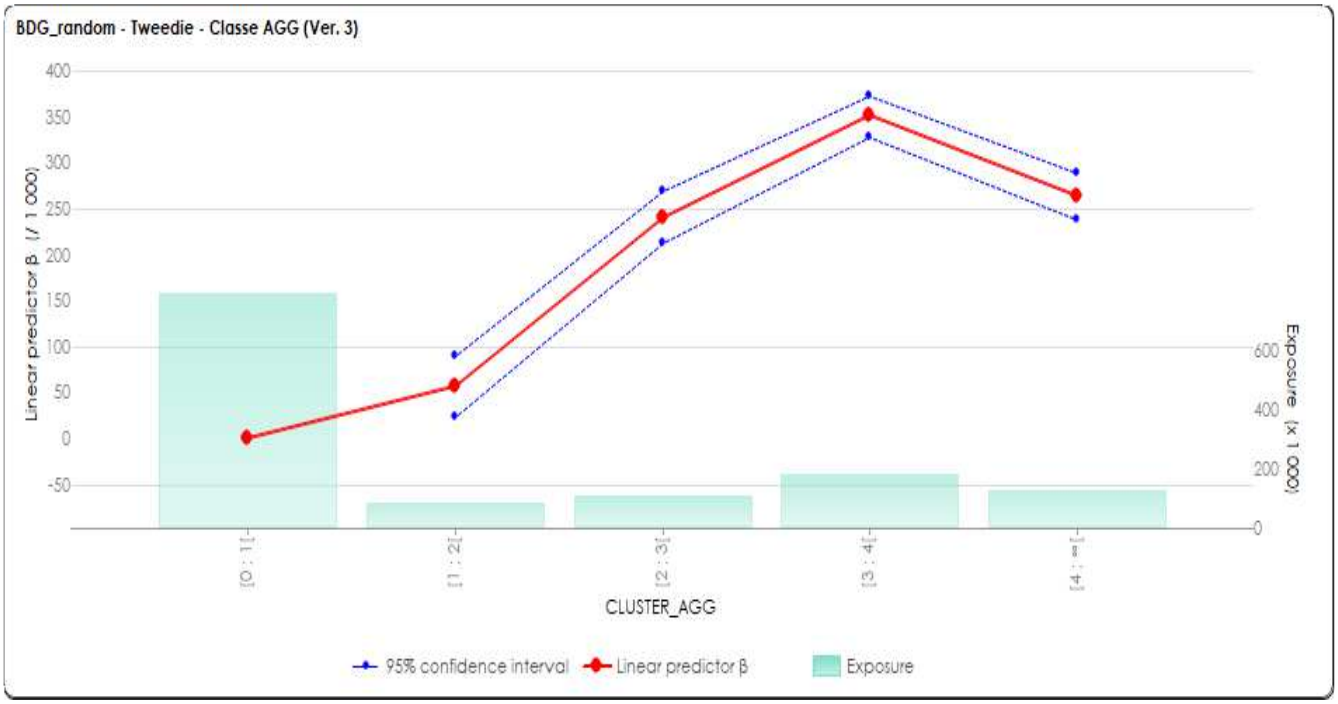


Figure 46 : Impact tarifaire BDG par cluster_agg

Idem concernant le cluster_agg qui présente la même typologie de risque que la garantie RCM avec la classe la plus rentable qui aurait le plus de risque.

iii. Garantie Vol

La garantie Vol est la suivante à être modelisée.

VOL	AIC	BIC	Coefficient de Gini
Modèle référent	103 513,61	103 538,71	0,3402
Modèle cluster km	103 408,27	103 508,69	0,4001
Modèle cluster agg	103 420,91	103 483,67	0,3835
Cluster km / Référent	-0,10%	-0,03%	17,61%
Cluster agg / Référent	-0,09%	-0,05%	12,73%
Cluster agg / Cluster km	0,01%	-0,02%	-4,15%

Tableau 14 : Statistiques des modèles de tarif Vol

Au vu des indicateurs, les variables de classification créées améliorent la qualité du modèle. On observe surtout l'amélioration du modèle avec l'évolution du coefficient de Gini

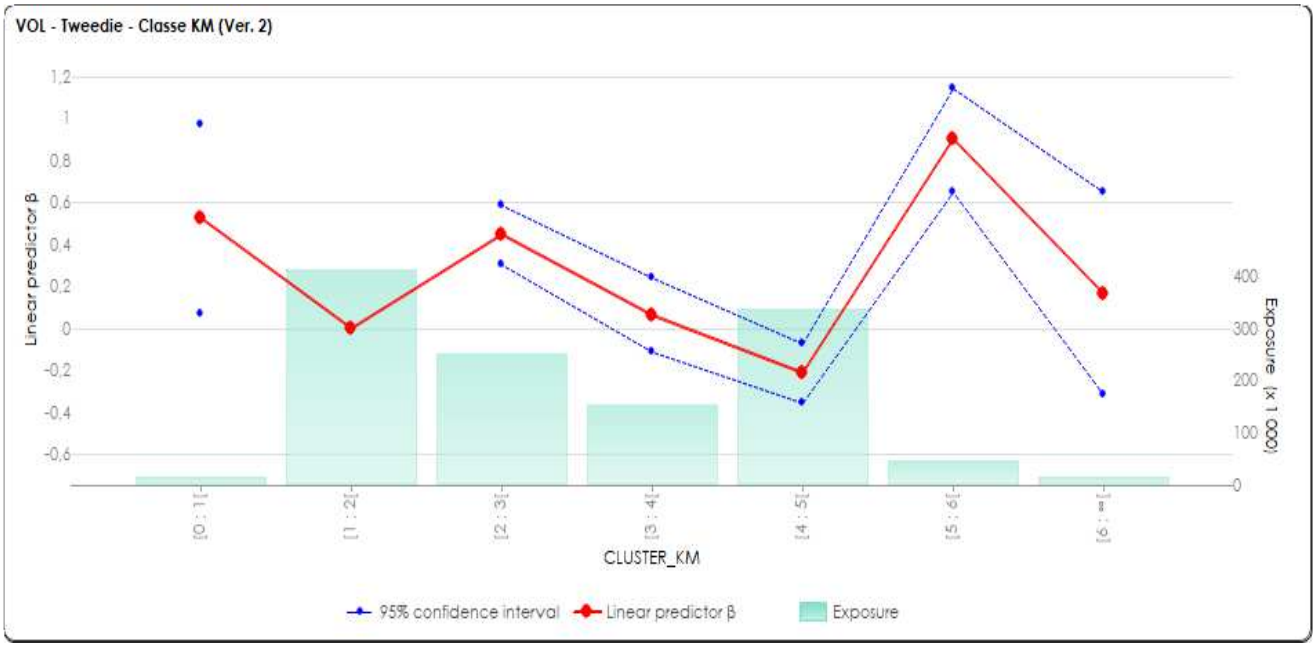


Figure 47 : Impact tarifaire Vol par cluster_km

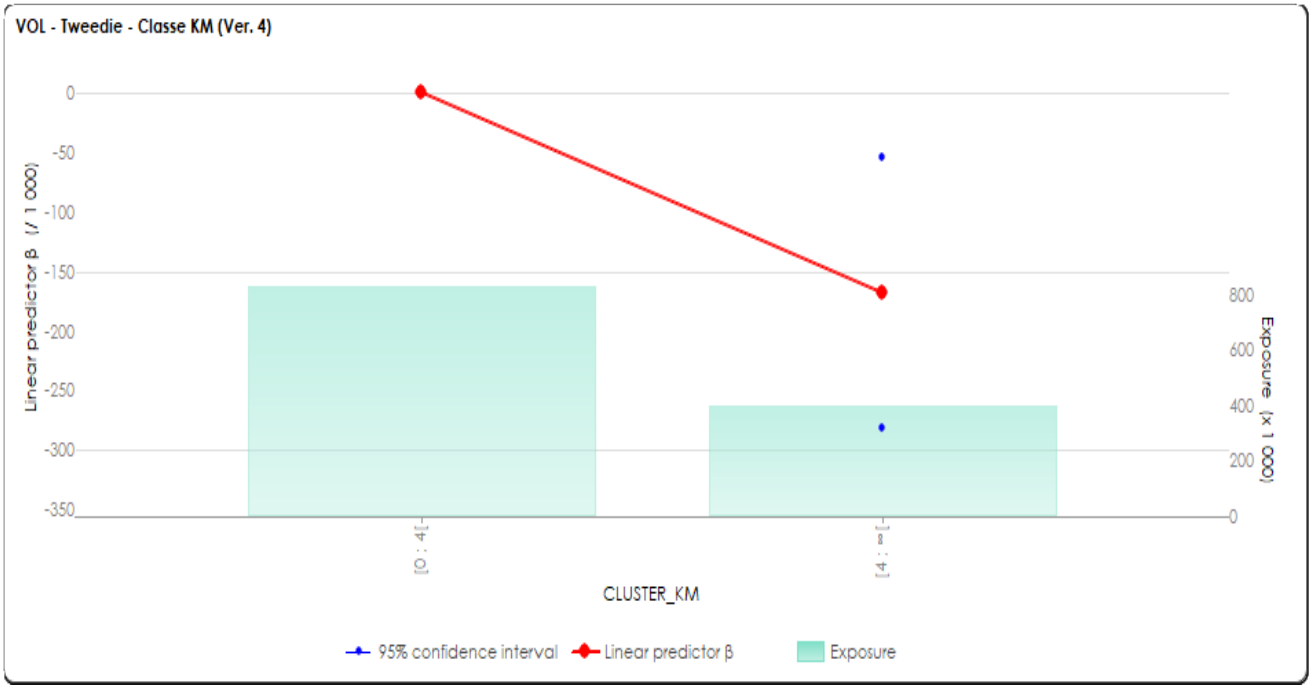


Figure 48 : Impact tarifaire Vol par cluster_km après regroupement

Pour la garantie VOL, les classes 4, 5 et 6 ayant pour point commun la souscription de nouveaux contrats sont moins risqués.

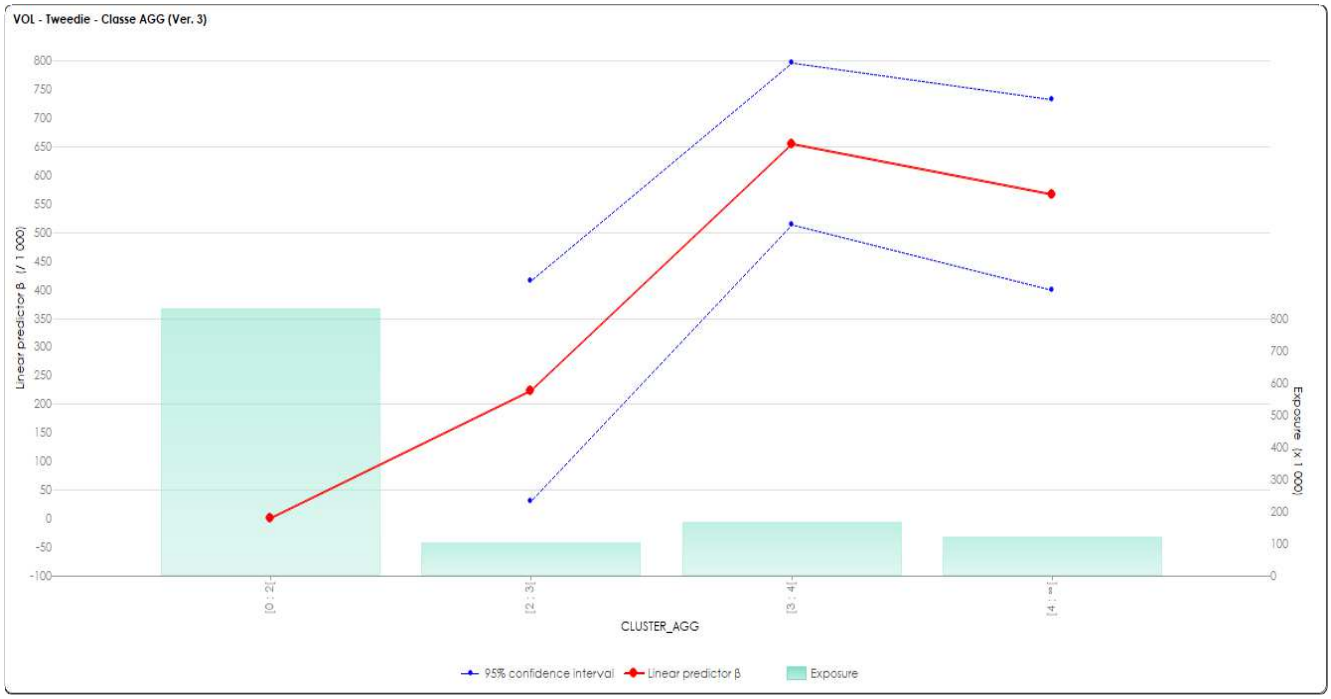


Figure 49 : Impact tarifaire Vol par cluster_agg après regroupement

La classe 3 présentant la meilleure rentabilité est la classe la plus risquée.

iv. Garantie Dommages Tous Accidents (DTA)

La garantie DTA sera la dernière testée.

DTA	AIC	BIC	Coefficient de Gini
Modèle référent	1 023 392,99	1 023 417,51	0,0655
Modèle cluster km	1 022 799,35	1 022 897,41	0,1355
Modèle cluster agg	1 022 821,96	1 022 920,02	0,1331
Cluster km / Référent	-0,06%	-0,05%	106,87%
Cluster agg / Référent	-0,06%	-0,05%	103,21%
Cluster agg / Cluster km	0,00%	0,00%	-1,77%

Tableau 15 : Statistiques des modèles de tarif DTA

Au vu des indicateurs, le constat effectué sur les garanties précédentes peut également être fait sur la garantie DTA à savoir une amélioration faible de la qualité du modèle au vu de l'AIC et du BIC mais on observe un coefficient de Gini qui double ce qui est excellent.

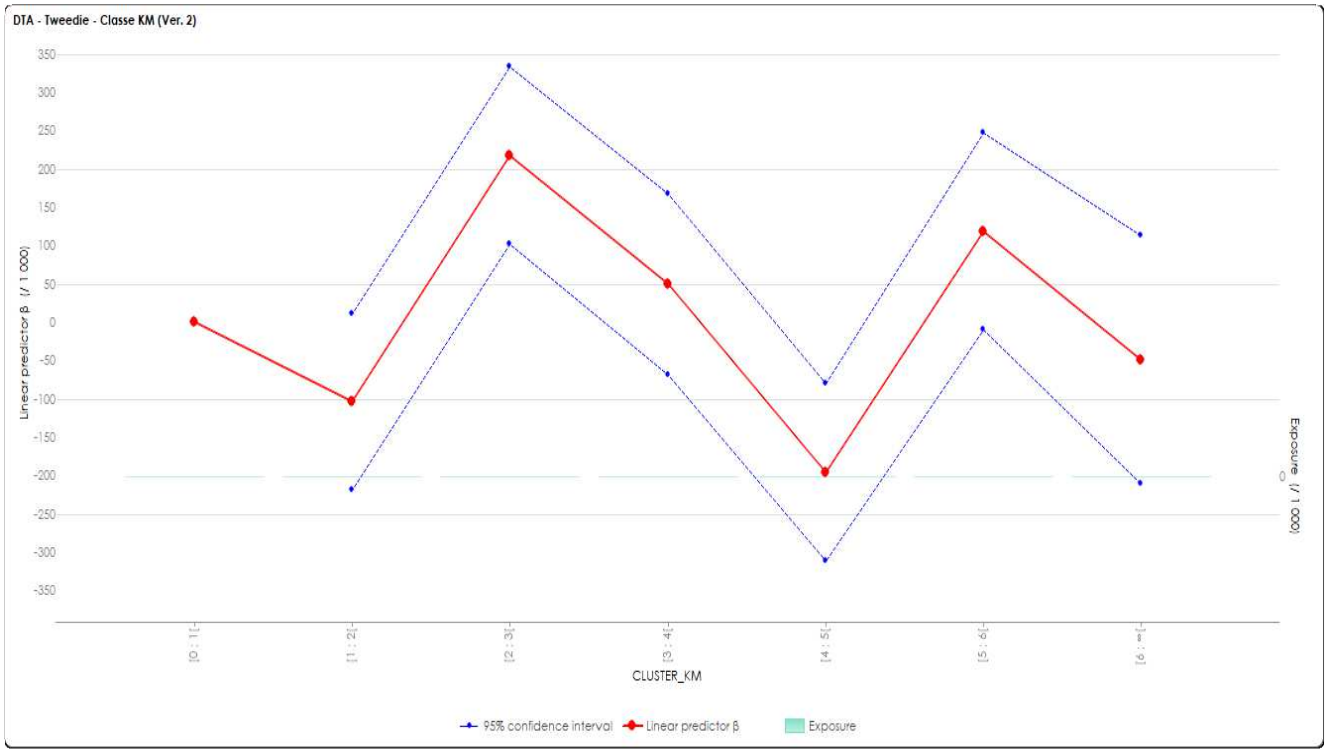


Figure 50 : Impact tarifaire DTA par cluster_km

On observe approximativement le même comportement que sur les garanties précédentes notamment sur la classe 4 qui est une nouvelle fois la classe la moins risquée.

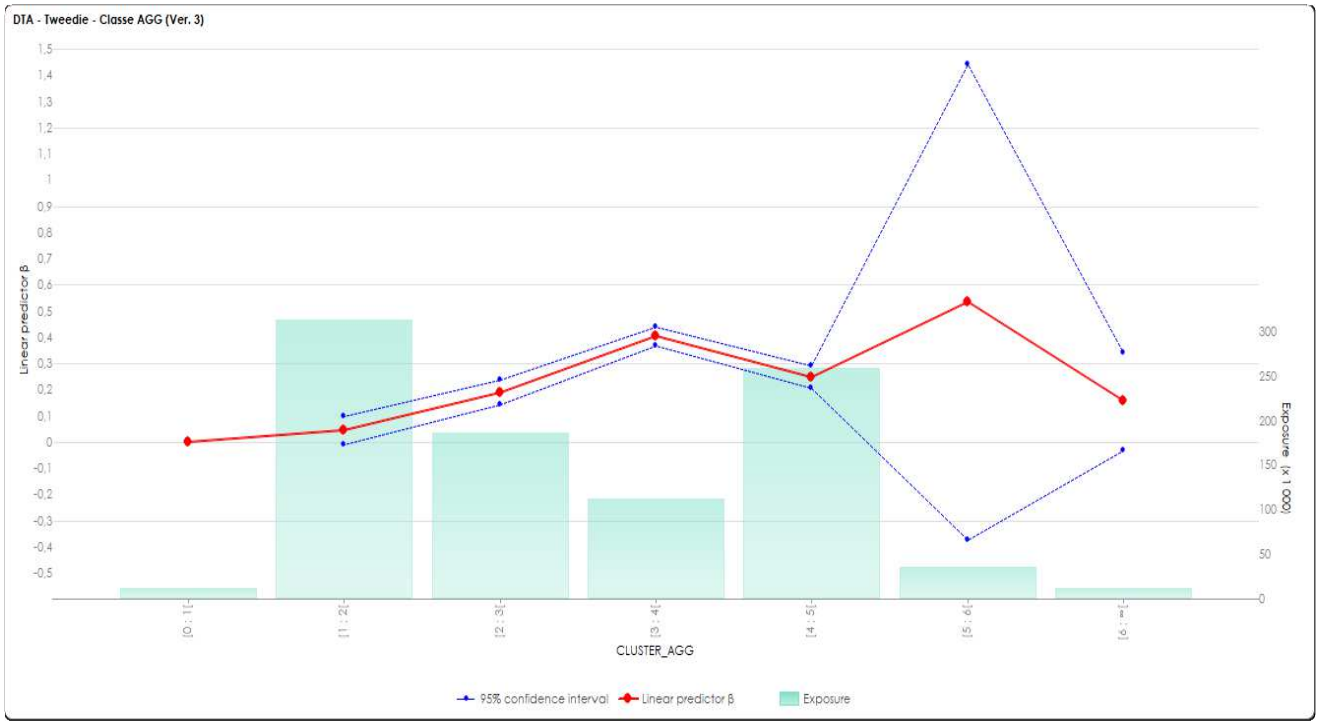


Figure 51 : Impact tarifaire DTA par cluster_agg

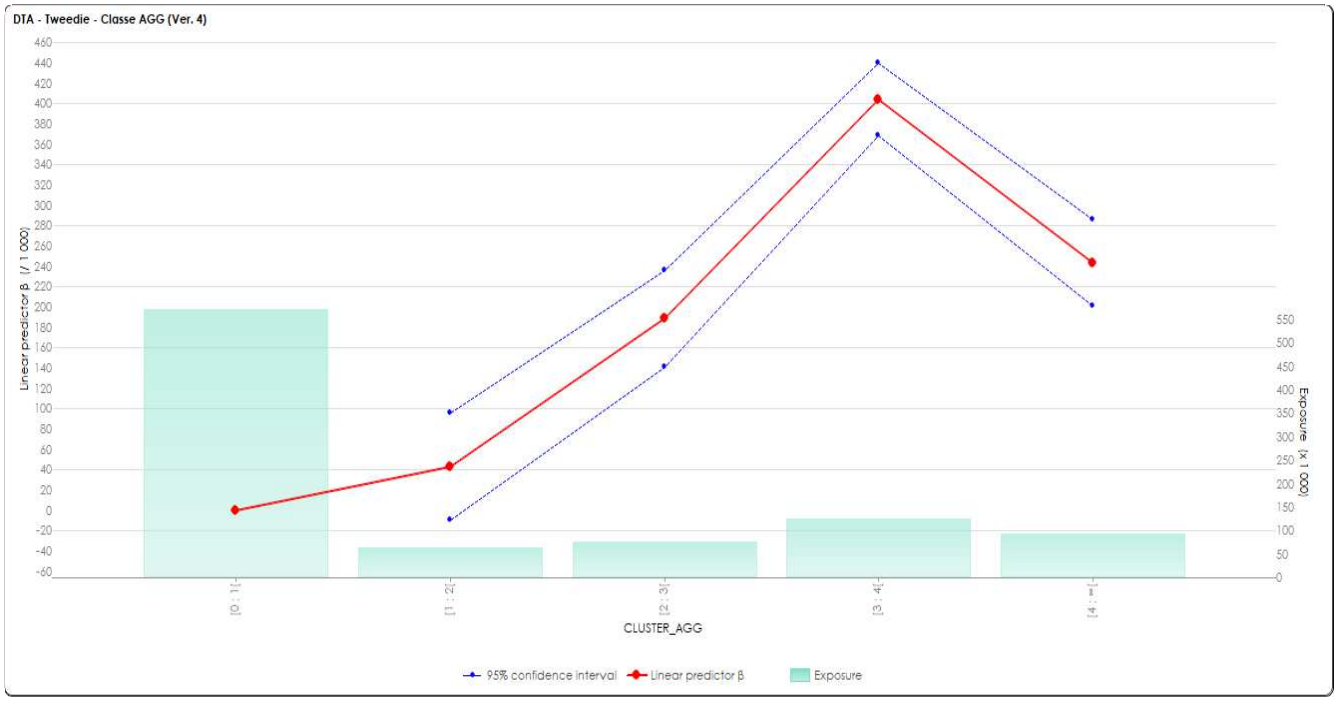


Figure 52 : Impact tarifaire DTA par cluster_agg après regroupement

Même constat pour le cluster_agg avec la classe 3, qui est la plus rentable, ressortant comme la plus risquée.

Quel que soit la garantie modélisée et observée, les constats sont les mêmes sur les résultats liées aux différentes classes modélisés. Le premier étant que l'intégration de ces classifications dans le model tarifaire automobile existant est significatif et a un réel impact technique sur la probabilité du risque.

Ensuite, concernant le cluster_km, la classe la plus rentable est la moins risquée suivi de la classe la plus récente présentant peu de mouvements dans son foyer.

C'est l'effet inverse qui est observé quand on s'intéresse aux résultats du cluster_agg avec la classe 3, représentant également la classe la plus rentable comme celle ayant la plus grande probabilité d'avoir des sinistres.

Intuitivement, cela nous amène à vouloir conserver les résultats du cluster_km. En effet, même si cela se vérifie techniquement, il est difficilement explicable commercialement de dire que la classe la plus rentable serait la plus risquée et également celle qui subirait une majoration de tarif.

On peut émettre l'hypothèse que la modélisation du tarif en utilisant cette classification capte un autre effet non identifié de la classe 3 que la rentabilité que nous mettons en avant ou les autres critères que nous observons pour décrire les classes. Elle pourrait par exemple capter un effet de sensibilité au prix faible. Cela voudrait dire que malgré leur rentabilité plus élevée, l'augmentation de tarif que ces foyers seraient susceptibles de subir n'impacteraient pas leurs décisions de partir à la concurrence s'ils le souhaitaient. Une autre explication serait que les profils du foyer regroupant la classe, malgré leur bonne rentabilité, seraient plus risquées concernant le risque automobile.

Dans le cas où une des deux classifications serait retenue pour être intégrée au tarif, le travail ne s'arrêterait pas là. Il faudrait par la suite recalibrer les primes de base des différentes garanties pour avoir un impact nul sur le chiffre d'affaires estimé.

Ce test comme une surcouche du tarif nous montre donc bien que la classification créée (quel que soit la méthode) à un impact technique. Il serait donc intéressant, lors d'une refonte complète du produit, de tester ce critère comme un critère libre c'est-à-dire sans contraindre les coefficients des autres critères tarifaires envisagés afin de capter les réels impacts.

IV. Modèle prédictif

Après avoir vu l'impact que pouvait avoir les classifications comme un critère tarifaire, intéressons-nous ici à la prédiction de cette classification. Pour la suite du mémoire, nous partirons sur le fait que nous conservons uniquement le cluster_km.

La contrainte qui se pose en ayant un tel critère dans notre modélisation tarifaire est la période d'observation nécessaire à la création de cette variable. Que faire en attendant d'avoir le recul nécessaire ? Essayons donc de prédire cette valeur avec les critères d'entrée du produit automobile. Nous pourrons alors directement utiliser la valeur dans le modèle tarifaire en attendant d'avoir l'historique complet qui nous permettra d'avoir les données nécessaires pour calculer la valeur actuarielle « réelle ».

Pour essayer de prédire cette valeur, différents modèles seront testés à savoir la régression logistique, le random forest, les arbres de décision et le XGBoost.

Un modèle de prédiction à l'inverse de nos modèles de classification étudiés précédemment est un modèle d'apprentissage supervisé. Cela consiste en l'entrainement d'un algorithme en utilisant des données labellisées c'est à dire des données qui ont déjà été étiquetées avec la bonne classe nous concernant.

Avant d'expliquer brièvement les modèles utilisés et de découvrir leurs résultats, nous utiliserons pour tous les modèles la matrice de confusion afin de juger de la qualité du modèle. Elle se présente comme suit :

		Réalité		
		Classe 0	Classe 1	Classe 2
Prédiction	Classe 0	Correct	Incorrect	Incorrect
	Classe 1	Incorrect	Correct	Incorrect
	Classe 2	Incorrect	Incorrect	Correct

Tableau 16 : Matrice de confusion

La matrice de confusion nous permettra de calculer le taux de succès (et d'en déterminer le taux d'erreur). Un autre indicateur de nous observerons est le f-score qui est une métrique pour évaluer la performance des modèles. Il permet de résumer les valeurs de la précision et du rappel en une seule métrique. Plus il est élevé, plus le modèle sera performant.

Ensuite concernant l'apprentissage, deux méthodes sont possibles. La méthode classique, à savoir scinder la base en deux échantillons (un échantillon d'apprentissage et un échantillon de test), entrainer le modèle sur la base d'apprentissage et le tester sur la base de test. La seconde méthode est la validation croisée qui consiste à découper le jeu de données en *k* parties et, tour à tour, chacune des *k* parties sera utilisé comme jeu de test. Le reste (l'union des *k-1* autres parties) est utilisé pour l'entrainement. Avec cette méthode peut se poser le problème de la répartition des données (absence d'une classe dans un jeu de test). Ce problème sera résolu en utilisant la stratification qui permettra de conserver la répartition de la base d'origine dans les différents jeux de test.

1. La régression logistique

Le premier modèle est le plus simple et l'un des plus connus. C'est un modèle statistique permettant d'étudier les relations entre un ensemble de variables qualitatives (nos critères de tarif automobiles) et une variable réponse (notre classification). Il s'agit d'un modèle linéaire généralisé utilisant une fonction logistique comme fonction de lien.

On utilisera la fonction python *LogisticRegression()*

Le résultat de cette fonction nous donne la matrice de confusion suivante :

	Classe 0	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6
Classe 0	0	0	0	0	0	0	0
Classe 1	1 867	11 927	5 047	936	5 000	1 433	4 811
Classe 2	3	10	12	1	8	1	3
Classe 3	0	0	0	0	0	0	0
Classe 4	129	489	206	39	495	70	75
Classe 5	0	0	0	0	0	0	0
Classe 6	241	902	236	44	426	53	1 201

Tableau 17 : Matrice de confusion de la régression logistique

Pour ce premier exemple, nous détaillerons la lecture de la matrice de confusion. Pour les modèles suivants, seuls les conclusions ou indicateurs seront directement exploitées.

Les éléments diagonaux représentent ceux qui ont bien été prédits par le modèle. La somme de ces éléments sur le nombre de cas testés représente le taux de succès. Il est ici de 38%.

Le fait d'avoir des lignes vides sur les classe 0,3 et 5 signifie qu'aucun foyer n'a été prédit dans ces classes avec le modèle ce qui n'est pas envisageable pour utiliser ce modèle de prédiction. Au contraire, la plupart des prédictions ont été classés dans la classe 1 et seul 38% de ces prédictions étaient corrects.

Classe 0	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	**Global**
0,00	0,54	0,01	0,00	0,16	0,00	0,26	**0,27**

Tableau 18 : F-Score du modèle de régression logistique

Le F-Score de chaque classe ou même du global confirme la non-performance du modèle.

2. Arbre de décision

Le principe de fonctionnement est simple. Un arbre de décision permet d'expliquer une variable cible à partir d'autres variables explicatives. L'algorithme va chercher à partitionner les individus en groupes d'individus les plus similaires possibles en fonction de la variable à prédire ici notre *cluster_km*. Le résultat produit alors un arbre qui révèle des relations hiérarchiques entre les variables. On peut alors déterminer et comprendre rapidement les règles expliquant la variable cible.

L'arbre de décision est un algorithme itératif qui, à chaque itération, va séparer les foyers en *k* groupes pour expliquer la cible. Le premier split est obtenu en prenant la variable explicative qui permet la meilleure séparation des foyers. Cette division donne des sous-groupes correspondant au premier nœud de l'arbre.

Le processus est ensuite répété pour chaque nœud précédemment calculé jusqu'à ce que le processus de split s'arrête.

La construction d'un arbre de décision ressemble donc à ceci :

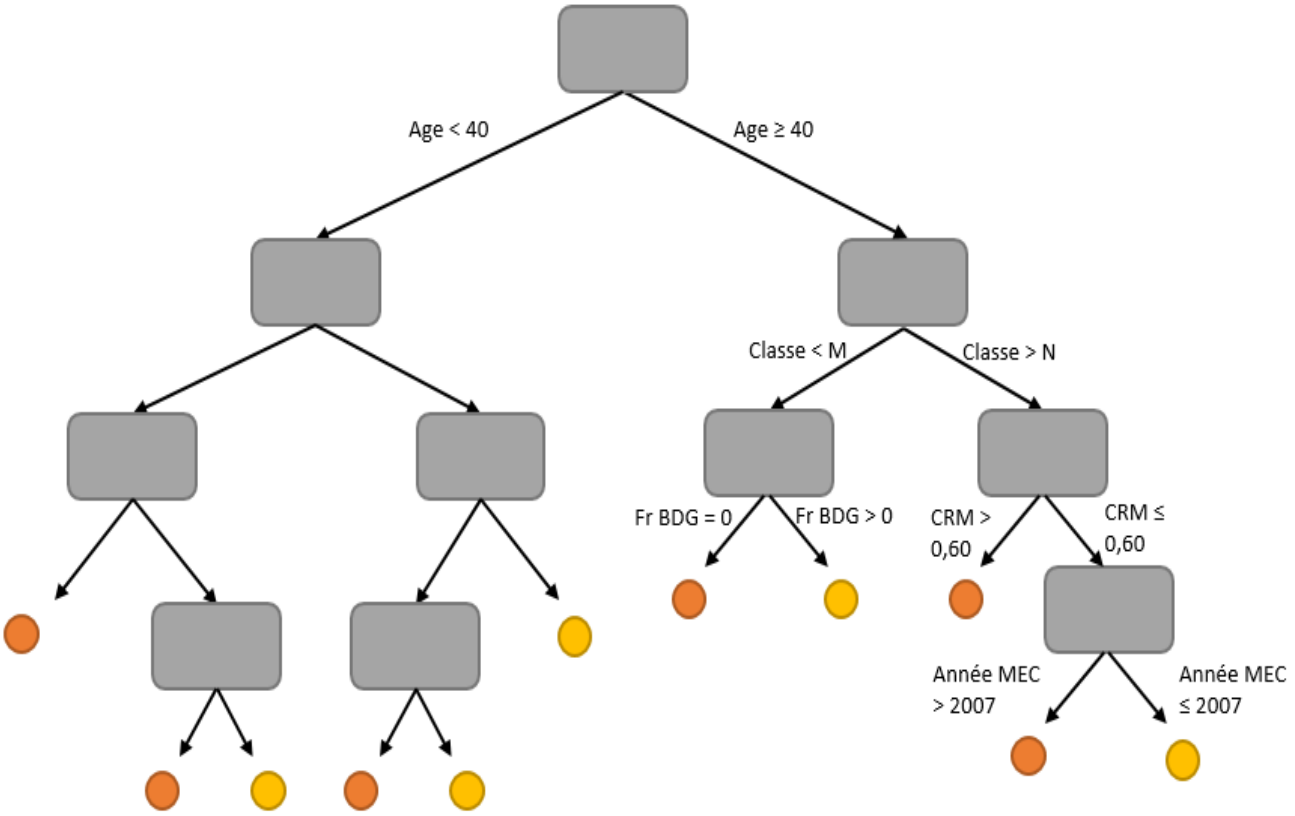


Figure 53 : Exemple d'arbre de décision

Sous python, la fonction utilisée pour cet algorithme se nomme *DecisionTreeClassifier()*

Cette fonction nous donne les résultats suivants :

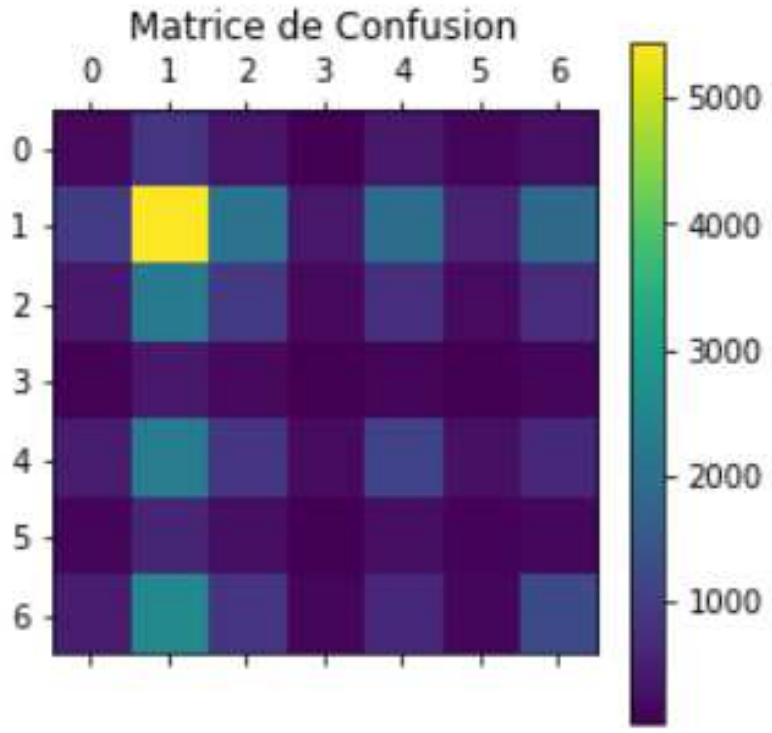


Figure 54 : Matrice de confusion de l'arbre de décision

Le taux de succès de la modélisation n'est que de 25%. La classe 1 est la mieux prédite mais la performance du modèle même pour la prédiction de cette classe reste faible comme le montre le F_score ci-dessous avec un score de 0,39.

Classe 0	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	**Global**
0,07	0,39	0,17	0,04	0,20	0,06	0,23	**0,25**

Tableau 19 : F_Score de l'arbre de décision

Comme avec le modèle de régression logistique, il est difficilement envisageable d'utiliser un modèle avec un taux de succès si faible.

3. Random Forest

« Random forest » signifie « forêt aléatoire » et c'est un algorithme qui se base sur l'assemblage d'arbres de décision, modèle précédemment testé.

Cet algorithme peut se décomposer en deux parties à savoir le tree bagging et le feature sampling. La première étape est un processus de tirage aléatoire sur les foyers qui se déroule en 3 étapes :

- Découpage de la base de données en n sous-ensemble aléatoirement constitués d'échantillons pour construire n arbres de décisions.
- Entrainement de chaque arbre de décision précédemment construit.
- Combiner tous les résultats des modèles. Pour une prévision sur de nouvelles données, il faut appliquer chacun des n arbres et prendre la majorité.

La seconde étape consiste à effectuer un tirage aléatoire sur les variables. Par défaut on utilise $\sqrt{j}$ variables pour une base contenant j variables explicatives.

Cela revient à entrainer les arbres avec un accès limité aux informations automobile à notre disposition. Par exemple un arbre utilisera uniquement les infos de l'ancienneté du permis, du groupe SRA et du niveau de franchise BDG quand un autre utilisera l'âge, la formule souscrite et l'année de mise en circulation du véhicule. Ce processus permet de réduire la corrélation entre les arbres qui pourrait perturber la qualité des résultats.

Schématiquement, l'algorithme peut se représenter comme suit :

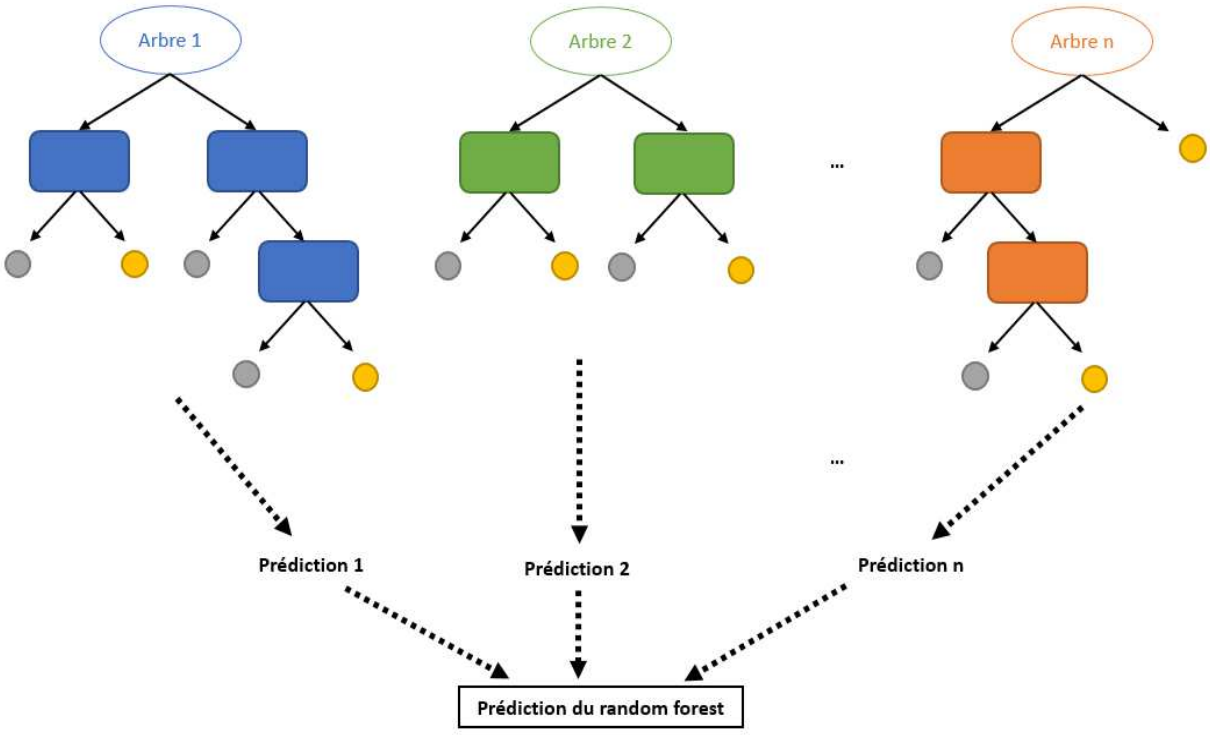


Figure 55 : Principe du Random Forest

En pratique voici ce que nous donne la matrice de confusion de cet algorithme.

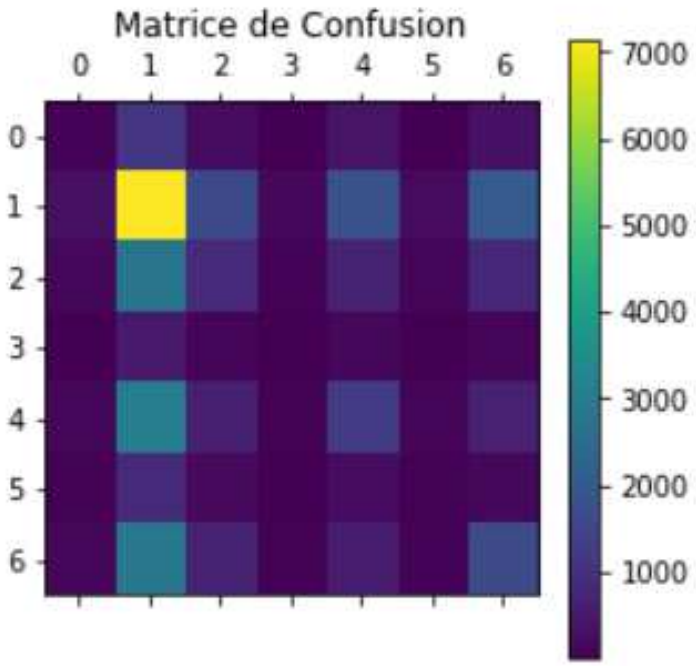


Figure 56 : Matrice de confusion Random Forest

Le taux de succès est de 30.6% ce qui est assez faible. Au vu de la matrice de confusion, la classe 1 est la mieux prédite mais comme pour les modèles testés précédemment, les performances ne sont pas assez élevées pour être susceptible d'être utilisé. Le F_Score ci-dessous nous le confirme avec un score de 0,29 au global.

Classe 0	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	**Global**
0,04	0,45	0,18	0,01	0,23	0,04	0,28	**0,29**

Tableau 20 : F_Score du modèle Random Forest

4. XGBoost

XGBoost signifiant eXtreme Gradient Boosting est le dernier algorithme de prédiction testé pour notre problème.

Le boosting va produire des modèles qui sont très dépendants les uns des autres. La première étape consiste à créer un premier modèle de base à partir d'un algorithme choisi. Au début, on attribue des poids égaux à toutes les observations et à partir des résultats obtenus sur le premier modèle, le poids d'une observation mal classée augmentera. Ensuite, un second modèle est construit pour essayer de corriger les erreurs présentes dans le premier modèle. Il est entrainé à l'aide des données pondérées obtenue lors de la première étape. Cette procédure se répète et des modèles sont ajoutés jusqu'à ce que l'ensemble complet des données d'apprentissage soit prédit correctement ou que le nombre maximal de modèles soit ajouté.

Les prédictions du dernier modèle ajouté seront les prédictions globales pondérées fournies par les anciens modèles d'arbres.

Le gradient boosting est un cas particulier de boosting où les erreurs sont minimisées par l'algorithme de descente de gradient, c'est-à-dire que chaque nouveau modèle va compenser les erreurs commises précédemment sans détériorer les prédictions déjà justes. Pour y arriver, la base d'apprentissage va changer à chaque itération. Au lieu de prédire la valeur d'origine, le p-ème modèle va prédire les résidus du (p-1)-ème modèle.

C'est également comme cela que fonctionne le modèle XGBoost qui va partir d'un classifieur faible et qui, à chaque itération va construire un classifieur plus optimisé que le précédent. Au bout d'un certain nombre de fois *k*, le classifieur final est une combinaison linéaire de tous les classifieurs intermédiaires.

Observons les résultats fournis par la méthode XGBoost

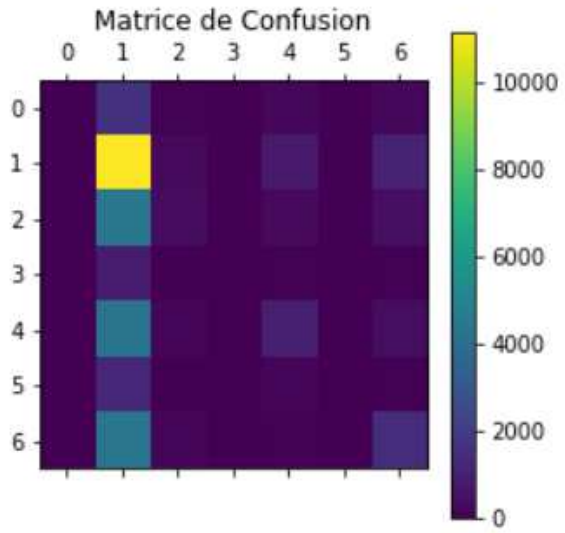


Figure 57 : Matrice de confusion XGBoost

Le taux de succès de ce modèle est de 39%. C'est le « meilleur » taux obtenu après avoir testé différentes méthodes mais cela reste insuffisant pour envisager de l'utiliser.

Classe 0	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	**Global**
0,07	0,39	0,18	0,03	0,21	0,05	0,26	**0,26**

Tableau 21 : F_Score du modèle XGBoost

Le F_Score nous conforte dans notre idée de ne pas utiliser le modèle à terme avec des indicateurs de performance assez faible. Même la prédiction de la classe 1 qui a le score le plus élevé ne peut pas être considéré comme un modèle performant

5. Support vector machine (SVM)

Le principe des SVM est simple, ils ont pour but de séparer les données à l'aide d'une frontière aussi simple que possible en maximisant la distance entre ces deux classes.

Schématiquement, on peut le représenter comme suit :

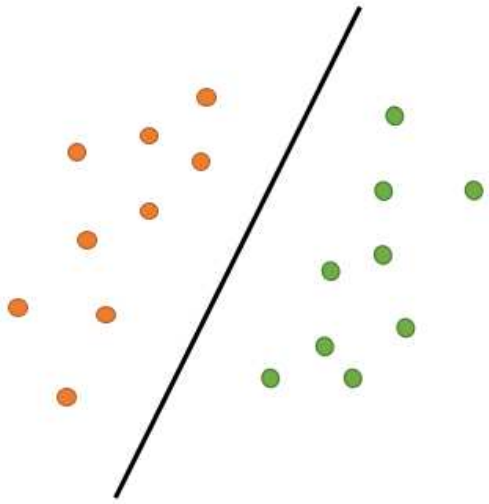


Figure 58 : Frontière entre deux espaces de points

Les points les plus proches, utilisés pour déterminer la frontière, sont appelés « vecteurs de support » et la distance entre ces points et la frontière est appelée la marge. Comme on peut l'imaginer, il existe une multitude de frontières valides et il faut donc chercher la plus optimale ce qui revient à chercher celle qui maximise la marge.

L'exemple ci-dessus montre un cas simple avec des données linéairement séparables ce qui est rarement le cas. Pour y remédier, les SVMs utilise des noyaux. Les fonctions noyaux vont permettre de séparer les données en les projetant dans un espace de plus grande dimension de façon que les données deviennent linéairement séparables comme le montre la figure ci-dessous.

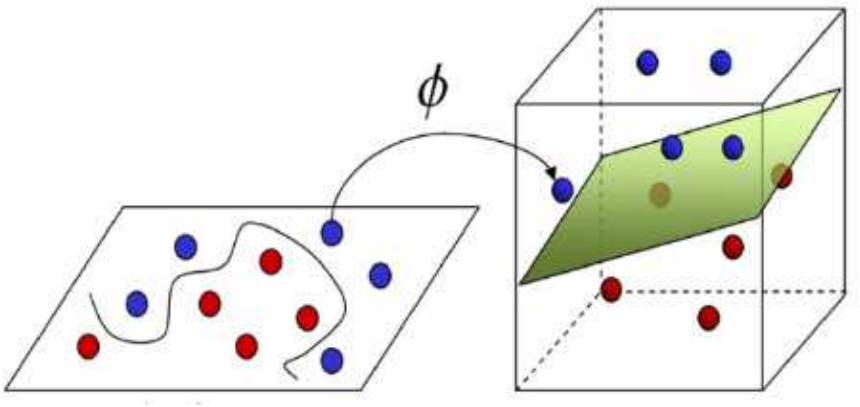


Figure 59 : Représentation d'un cas non linéairement séparables dans un espace de plus grande dimension

A l'origine, les SVM ont été conçus pour les problèmes binaires. Cependant, nous pouvons passer du cas 2 classes au multi-classes de deux façons différentes :

- **Un contre tous :** L'approche la plus naturelle est d'utiliser la discrimination binaire et consiste à créer un classifieur f_k pour chacune des classes, qui sépare les points de cette classe de tous les autres points. Cela nécessite d'apprendre K classifieurs. Un exemple de cette approche est la suivante.

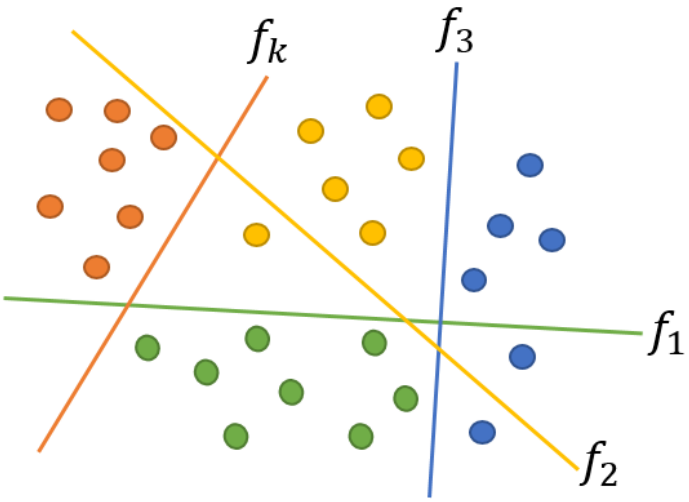


Figure 60 : Une frontière par classe séparant les points de ladite classe à tous les autres

Pour prédire l'affectation d'un nouveau point x, $f_k(x)$ indiquant la distance entre x et la frontière qui sépare la classe k des autres sera utilisée. Plus cette valeur est élevée, plus x a de probabilité d'appartenir à k. Autrement dit, x appartiendra à la classe i tel que :

$$i = \arg\max_k f_k(x) \ .$$

- **Un contre un :** Contrairement à l'approche précédente, on calcule ici des classifieurs séparant une classe d'une autre en faisant abstraction des autres classes. Sur l'exemple précédent, cela donnerait :

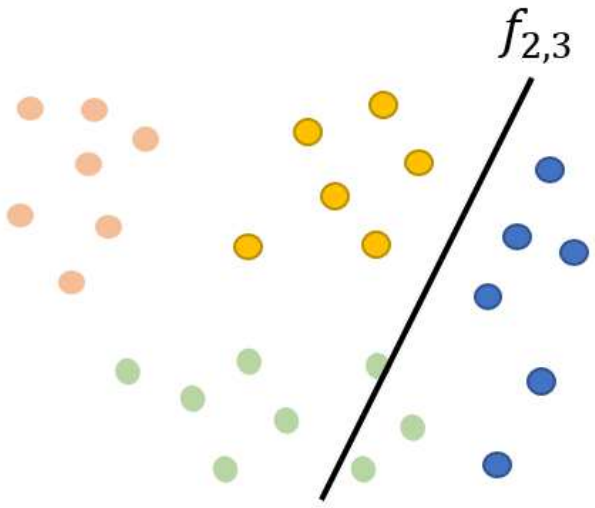


Figure 61 : Frontière séparant la classe 2 de la classe 3 sans prendre en compte les autres classes

Cela revient à calculer $K(K-1)$ classifieurs et pour affecter un nouveau point, nous utiliserons un vote de la majorité, c'est-à-dire que la classe prédite sera celle retournée par le plus grand nombre de classifieurs.

Appliquons maintenant cet algorithme à notre problématique.

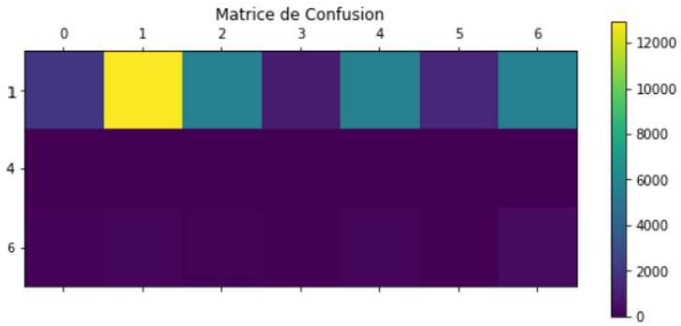


Figure 62 : Matrice de confusion SVM

Le taux de succès de ce modèle est de 38% grâce à la bonne prédiction de la classe 1 mais les classes 0,2,3 et 5 ne sont pas prédites du tout ce qui se ressent dans la qualité du modèle avec les F_scores ci-dessous. Même sans observer les scores, l'utilisation de ce modèle semble impossible si plusieurs classes ne sont pas du tout prédites.

Classe 0	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	**Global**
0	0,54	0	0	0,006	0	0,11	**0,22**

Tableau 22 : F_Score du modèle SVM

Le F_Score confirme les constatation effectuées avec la matrice de confusion et les performances sont trop faibles pour qu'il soit envisageable de mettre le modèle en production.

Aucun modèle de prédiction n'est assez satisfaisant pour être mis en production dans le cadre de notre problématique. Rappelons que le but premier de cette prédiction était d'anticiper la classe du foyer après 3 années d'observation afin d'activer l'impact tarifaire que nous avons déterminé dès l'entrée du foyer chez Thelem assurances. Dans l'état actuel des choses, l'activation de ce critère n'interviendrait qu'après plusieurs années de vie du foyer chez Thelem.

On retient donc qu'avec notre approche de prédire la classe uniquement avec les critères tarifaires d'entrée du produit automobile, cela n'est pas concluant.

Différentes pistes peuvent être envisagées par la suite :

- Tester une prédiction prenant en compte les critères d'entrée du produit habitation seule.
- Tester une prédiction prenant en compte les critères des produits automobiles et habitation.
- Tester une prédiction avec les critères communs du foyer quelque soit le produit d'entré chez Thelem Assurances.
- Attendre le passage d'un terme, potentiellement synonyme de sinistre et d'équipement du foyer, afin d'avoir les données nécessaires à un « pré-calcul » de la valeur actuarielle sur une période d'observation plus courte.

Conclusion

Ce mémoire a pour but de créer une « valeur actuarielle » en classifiant les foyers en fonctions des caractéristiques de leurs contrats principaux, à savoir automobile et habitation, de leur comportement et de leur rentabilité. Pour créer cette valeur, nous avons opté pour de l'apprentissage non supervisé. En effet, l'objectif était de créer une classification à partir des données voulues mais sans variable réponse.

Plusieurs méthodes ont été testées à savoir le K-Means, le MeanShift, la propagation d'affinité, le DBSCAN et la classification ascendante hiérarchique.

Les résultats du MeanShift ont montré, dès la répartition des foyers dans les différentes classes, un déséquilibre avec une sur représentation d'une classe à plus de 90%. Ces résultats ne conviennent pas, dans l'état, à l'utilisation que nous souhaitons de cette classification. Cependant, certains paramètres de l'algorithme pourraient être revus et étudiés plus en détail notamment la bande passante qui pourrait relever d'une étude en elle-même. Il ressort de la littérature que cette méthode crée des clusters avec des tailles inégales et les résultats obtenus sont trop dispersés pour notre besoin.

La propagation d'affinité fut également un échec mais la méthode est complexe avec un grand nombre d'échantillons à cause de la complexité quadratique de l'algorithme notamment avec la construction de la matrice de similarité de taille NxN.

L'algorithme DBSCAN nous a également posé problème dans le sens ou epsilon est difficile à estimer. Malgré en avoir testé un bon nombre, nous avons retrouvé une inégalité dans la taille des classes. A l'image du MeanShift, on peut se poser la question d'un problème de densité sur notre jeu de données potentiellement dû à la réduction de la dimensionnalité.

De tels résultats sont inconcevables à mettre en production puisque cela n'apporterait aucune plus-value.

Les résultats fournis par le K-means furent satisfaisant. La question restant en suspens est le choix du nombre de classes qui reste à la main de l'utilisateur. Même si le choix est pris en utilisant des méthodes mathématiques, il reste arbitraire et tester d'autres nombre de classes pour voir l'impact sur les caractéristiques des variables et leur interprétabilité est envisageable.

La classification ascendante hiérarchique donne des résultats satisfaisants également. Face à cet algorithme, un nouveau problème est apparu à savoir un problème de mémoire vive puisque nous avons dû réduire de 60% la taille de la base pour que l'algorithme converge. Il serait intéressant d'être en mesure d'apprendre le modèle sur l'intégralité de la base pour observer le biais que cela amènerait dans la constitution des classes.

La suite à donner à cette classification était d'interpréter au maximum les classes créées afin de pouvoir les décrire et d'être en mesure de communiquer sur ces classes.

Chaque classe était donc plus ou moins marquée ou sur représentée par des caractéristiques différentes et à ce stade du mémoire, aucune classe ne se détachait pour effectuer notre choix.

La valeur actuarielle pourrait avoir plusieurs utilités notamment en surveillance du portefeuille en repérant plus facilement qu'aujourd'hui les foyers peu rentables dès leur premier sinistre sans attendre un phénomène de surfréquence sur un produit. Autre exemple en souscription, si un foyer ayant une valeur correcte souhaite souscrire un nouveau contrat, on pourrait imaginer une plus grande souplesse sur la proposition des ajustements tarifaires.

Une autre utilisation envisagée serait évidemment de l'intégrer comme un critère dans la tarification. Les deux variables créées ont donc été testées comme une surcouche tarifaire sur le produit automobile existant. Quatre garanties ont été modélisées et le même constat est effectué à savoir que l'ajout de la variable améliore la qualité du modèle ainsi que la qualité de la segmentation. De plus la variable, et ses différentes modalités, ressortent toujours significatives avec un impact tarifaire marqué.

C'est en modélisant les deux valeurs dans le tarif automobile que notre choix s'est porté sur la classification créée avec la méthode K-means. En effet, la variable créée avec la classification ascendante hiérarchique aurait un impact tarifaire contraire à l'idée qu'on en attend et contraire à la logique, à savoir que les foyers ayant la plus forte rentabilité auraient une augmentation de tarif. L'hypothèse effectuée est que la classe la plus rentable « cache » un comportement plus à risque en Auto.

La dernière partie consistait à essayer d'anticiper cette valeur actuarielle au moment de l'entrée dans l'entreprise du foyer avec un contrat automobile. En d'autres termes, prédire la valeur grâce aux critères de tarification du produit automobile.

Différents modèles d'apprentissage supervisés ont été testés pour prédire la valeur que sont la régression logistique, les arbres de décisions, le XGBoost et le support vector machine.

Malgré l'utilisation de la validation croisée stratifiée assurant de conserver la répartition de la base d'origine dans les bases d'apprentissages, certaines classes ne sont pas du tout prédites par la régression logistique. La précision étant à 0 sur plusieurs classes, la qualité du modèle est discutable.

Les résultats sont sensiblement les mêmes pour les différentes méthodes à savoir que les modèles créés ne sont pas assez performants et que nous ne donnerons pas suite à cette anticipation de la valeur actuarielle. Dans l'état, cela signifie donc que les critères automobiles à eux seuls ne suffisent pas à prédire la valeur créée.

Plusieurs pistes peuvent être envisagées pour essayer d'avoir de meilleures performances sur les modèles testées. On pourrait, par exemple, collecter plus de données en intégrant, quand les données du foyer le permettent, des éléments supplémentaires. Une autre piste serait de tester une prédiction au bout d'1 an, ce qui permettrait de prendre des mesures à la 1ère échéance principale du contrat automobile.

Au niveau des données, la limitation du nombre de variables dans les modèles avec des techniques de sélection de variable ou en analysant leur importance permettrait de nettoyer la base et d'améliorer la qualité globale des données utilisées pour la prédiction. De même, retraiter les variables existantes, soit en les décomposant, soit en les catégorisant comme l'âge du véhicule qu'on pourrait regrouper en « neuf / récent / ancien » représenterait potentiellement des informations plus intéressantes sur les données à prédire.

Pour finir, les réglages liés aux modèles seraient une piste à explorer comme optimiser les paramètres de chaque modèle ou bien effectuer un diagnostic de chaque algorithme en examinant la courbe d'apprentissage pour détecter, le cas échéant, un sur ou sous apprentissage.

D'une manière générale, arrivé à ce stade, d'autres pistes plus générales peuvent être imaginées ou envisagées et d'autres questions peuvent se poser.

Avec une fréquence de 15% en automobile et de 8% en habitation, une question sur le périmètre peut se poser. Une période d'observation plus longue pourrait apporter de la consistance à la variable de la marge pondérée avec une période d'exposition au risque plus longue ainsi qu'une liquidation des sinistres plus importantes.

Le test fut donc concluant sur l'impact de la valeur actuarielle comme un critère de tarif en surcouche sur le tarif automobile existant, il faudrait prendre le temps de le tester sur différents produits mais aussi lors de différentes refontes en l'intégrant comme un critère classique et pas seulement comme une surcouche sur un tarif existant.

Avec l'open data, pourquoi ne pas envisager d'inclure des variables externes tels que des variables géographiques au niveau INSEE ou IRIS qui traduirait ou nous donnerait des indications sur le comportement du foyer ou sur son environnement proche.

Pour finir et comme précisé en introduction, l'utilité première que nous voyons de cette valeur est dans le tarif mais d'autres applications sont envisageables comme une utilisation dans la surveillance du portefeuille ou une aide à la prise de décision pour des questions de souscription sur des foyers existants.

Bibliographie

[1] Ricco RAKOTOMALALA , Laboratoire Eric, Université Lumière Lyon 2 :

Ouvrages de statistiques, d'analyse de données, de data mining et de data science - Ricco Rakotomalala (univ-lyon2.fr)

[2] Statistiques & Machine Learning : www.wikistat.fr

[3] Tutoriels Python : www.delftstack.com

[4] Documentation cours débutant : www.python.doctor

[5] Bibliothèque libre Python destinée à l'apprentissage automatique : https://scikit-learn.org

[6] Organisme de formation en Data Science : www.datascientest.com

[7] Site d'hébergement de vidéos : www.youtube.com On peut tout trouver sur cette plateforme y compris des tutoriels python et des cours de data science.

[8] Aide à l'apprentissage de Python et aide-mémoire en ligne : www.python-simple.com

[9] Partages de publications, idées et codes data science : www.towardsdatascience.com

[10] Revue française spécialisée en intelligence artificielle : www.larevueia.fr

[11] Formation en ligne et cours en accès libre : www.openclassrooms.com

[12] Apprendre le machine learning de A à Z www.mrmint.fr

[13] www.datascience.eu

[14] Commission nationale de l'informatique et des libertés : www.cnil.fr

[15] ministère de l'Économie, des finances et de la relance : www.economie.gouv.fr