GRANDE ÉCOLE
D'ACTUARIAT
ET DE GESTION
DES RISQUES

INSTITUT DE SCIENCE FINANCIÈRE ET D'ASSURANCES

TARIFICATION AUTOMOBILE

Data Science

DIABATE Fatou
FOFANA Fadel
OKOUNLOLA-BIAOU Olaniran J.
YAO Philippe Olivier

M2 Actuariat

Enseignant : Pierrick Piette

ISFA

Table des matières

1 INTRODUCTION

La tarification est au coeur de l'activité d'assurance. Du fait de l'inversion du cycle de production (c'est à dire l'obligation de proposer aux assurés une prime sans connaître le coût des éventuels sinistres), la tarification en assurance est complexe et l'actuaire se repose sur des modèles mathématiques qui lui permettent de capter dans la limite du possible le comportement des assurés et de produire une prime appropriée. Dans cette optique, l'apprentissage statistique ou Data Science fournit à l'actuaire des outils pour la mise en oeuvre de ces modèles. L'objet de ce projet est d'utiliser les différentes techniques et la méthodologie de conduite d'un projet de data science à des fins de tarification automobile. Il s'agira plus précisément de calculer la prime pure des assurés compte tenue de leurs caractéristiques. Théoriquement, on cherche à calculer :

$$E[S|X] = E[N|X]E[Y|X]$$

où

- **S** est la variable aléatoire charge sinistre totale pour une police d'assurance dont l'espérance représente la **prime pure**
- **N** est la variable aléatoire qui modélise le nombre de sinistres annuels sur la police
- $X = (X_1, ..., X_k)$ est le vecteur aléatoire des caractéristiques de l'assuré qui bénéficie de la couverture par la police d'assurance et
- Y la variable aléatoire coût des sinistres sur la police pendant une année

Le calcul de la prime pure revient donc à estimer 2 modèles ; l'un servira à calculer la fréquence moyenne des sinistres $E[N|X]$ et l'autre le coût moyen des sinistres $E[Y|X]$
Il est important de préciser ici que dans le cadre de ce projet une hypothèse simplificatrice du calcul de la prime pure nous a été donnée : **il ne peut y avoir qu'un sinistre par assuré (par police) sur l'année**. Cette hypothèse implique un calcul de la prime pure de la manière suivante :

$$E[S|X] = P[Y > 0|X = x]E[Y|Y > 0, X = x]$$

c'est-à-dire le produit entre la probabilité qu'il y ait un sinistre compte tenu des caractéristiques de l'assuré et le coût moyen de ce sinistre. Nos travaux se partagent ainsi en 2 parties :

- une partie **régression** qui vise à modéliser $E[Y|Y > 0, X = x]$ et
- une partie **classification** qui vise à modéliser $P[Y > 0|X = x]$

2 DATA MANAGEMENT

2.1 Dictionaire des Données

Dans le cadre de ce projet de tarification automobile , il nous a été fourni deux bases de données **train** et **test** qui servent respectivement à entraîner nos modèles et à tester leur pertinence ; c'est-à-dire leur pouvoir de prédiction. Chacune des bases est constituée des variables (sauf la dernière variable qui est uniquement dans la base train) suivantes :

- **Id** : numéro d'identification du conducteur
- **Gender** : genre du conducteur/conductrice
- **carCatgory** : catégorie de la voiture
- **Occupation** : statut professionnel du conducteur
- **age** : age du conducteur
- **carGroup** : Groupe de la voiture
- **Bonus** : Bonus-Malus sur l'année dernière

— **CarValue** : valeur de la voiture
— **material** : indicatrice pour la couverture matérielle
— **region** : région géographique de la résidence
— **subRegion** : sous région géographique de la résidence
— **CityDensity** : Densité de la population (hbt / km^2) de la ville de résidence
— **claimValue** : Valeur indemnisée au titre des sinistres déclarés

Le dictionnaire des données ci - dessus est assez lisible, on remarque facilement que pour la quasi-totalité des variables, le nom est suffisamment expressif.
La variable **carGroup** déroge légèrement à cette remarque ; il pourrait être difficile d'intuiter son contenu.
La variable **claimValue** est notre variable d'intérêt ; il s'agit du montant annuel des sinistres déclarés par l'assuré.

2.2 Une Première Observation des Données

Il parait naturel de commencer par une prise de connaissance de nos bases de données. Le résumé statistique ci-dessus présente pour chacune de nos bases, les variables, leurs types respectifs, les quantités statistiques significatives (moyenne, médiane, quantile....) pour les variables quantitatives et les différentes modalités et leurs effectifs respectifs pour les variables catégorielles.

```
       id           gender      carType  carCategory          occupation       age            carGroup
 Min.   :    1   Female:10515   A:8164   Large :10415   Employed     :9388   Min.   : 18.00   Min.   : 1.00
 1st Qu.: 7501   male  : 1000   B:6612   Medium:10966   Housewife    :6114   1st Qu.: 29.00   1st Qu.: 7.00
 Median :15000   Male  :18485   C:4140   Small : 8619   Retired      :3701   Median : 40.00   Median :11.00
 Mean   :15000                  D:6009                  Self-employed:6056   Mean   : 42.54   Mean   :10.78
 3rd Qu.:22500                  E:3393                  Unemployed   :4741   3rd Qu.: 52.00   3rd Qu.:14.00
 Max.   :30000                  F:1682                                      Max.   :134.00   Max.   :20.00

     bonus           carValue         material        subRegion         region      cityDensity       claimValue
 Min.   :-50.00   Min.   :  1000   Min.   :0.0000   Q29    :  143   L      :7001   Min.   : 14.38   Min.   :    0.0
 1st Qu.:-40.00   1st Qu.:  8500   1st Qu.:0.0000   M17    :  138   Q      :6693   1st Qu.: 51.32   1st Qu.:    0.0
 Median :-20.00   Median : 15158   Median :1.0000   Q40    :  136   R      :4716   Median : 98.10   Median :    0.0
 Mean   : -4.51   Mean   : 19789   Mean   :0.5045   Q26    :  134   M      :2356   Mean   :118.83   Mean   :  514.8
 3rd Qu.: 10.00   3rd Qu.: 23471   3rd Qu.:1.0000   M19    :  129   U      :1568   3rd Qu.:180.08   3rd Qu.:    0.0
 Max.   :150.00   Max.   :149475   Max.   :1.0000   Q36    :  129   O      :1564   Max.   :297.39   Max.   :62092.7
                                                    (Other):29191   (Other):6102
>
```

FIGURE 1 – Résumé statistiques - Base Train

2.3 Data Cleaning

Cette étape consiste à "rendre propre" la base de données en faisant des traitements mineurs tels que corriger le type des variables, rechercher et corriger les possibles erreurs manuelles dans la saisie des données (à savoir valeurs manquantes, données manifestement aberrantes,...), convertir les variables aux formats exploitables (par exemple passer de *numeric* (variable quantitative) à *factor* (variable catégorielle).
Pour nous, cette étape à consister principalement à convertir en variables qualitatives les variables : **carGroup**, **material** et à corriger la modalité *male* de la variable **Gender** qui était orthographiée de deux manières différentes dans la base **train** initiale.
De manière générale, cette étape requiert beaucoup plus de traitement mais dans le cadre de ce projet il faut reconnaître que la base initiale était déjà de très bonne qualité (aucune valeur manquante, pas de valeur manifestement aberrante,...).

2.4 Feature Engennering Part 1 -

On rappelle qu'on souhaite modéliser dans la **partie régression** le montant moyen des sinistres pour un assuré étant donné son profil et **sachant qu'un sinistre a eu lieu**. De manière empirique, cette espérance se calcule en rapportant le montant total des sinistres au nombre de sinistres. Le calcul d'une telle espérance sur tout notre portefeuille risque d'être biaisé étant donné qu'un nombre significatif d'assurés n'a pas connu ou déclaré de sinistres ; en effet, 22635 assurés sur 30000 soit **75,45%** des assurés n'a déclaré aucun sinistre dans la base **train**. Il convient donc de créer pour les besoins de notre modélisation sur la partie régression une **base de sinistres non-nuls** en sélectionnant dans train uniquement les lignes pour lesquels $claimValue > 0$. On nommera cette base **train_Regression**

Dans la suite de ce rapport, nous proposons d'observer graphiquement les données et d'effectuer quelques statistiques descriptives identiques sur chacune des 3 bases suivantes :

— **train** : la base sinistre initiale
— **train_Regression** : la base des sinistres non-nuls
— **test** : la base de test

Cette approche nous permet d'étudier le comportement des variables dans chacune des bases et d'observer les rapprochements entre les trois bases.

2.5 Data Exploration

2.5.1 Analyses univariées des variables qualitatives

Il s'agit dans cette section de représenter et d'observer selon la variable qualitative un *barplot* (diagramme en bar) ou un *pie chart* (camembert).

Gender :

On représente un *pie chart* pour observer la répartition Homme Femme du portefeuille :

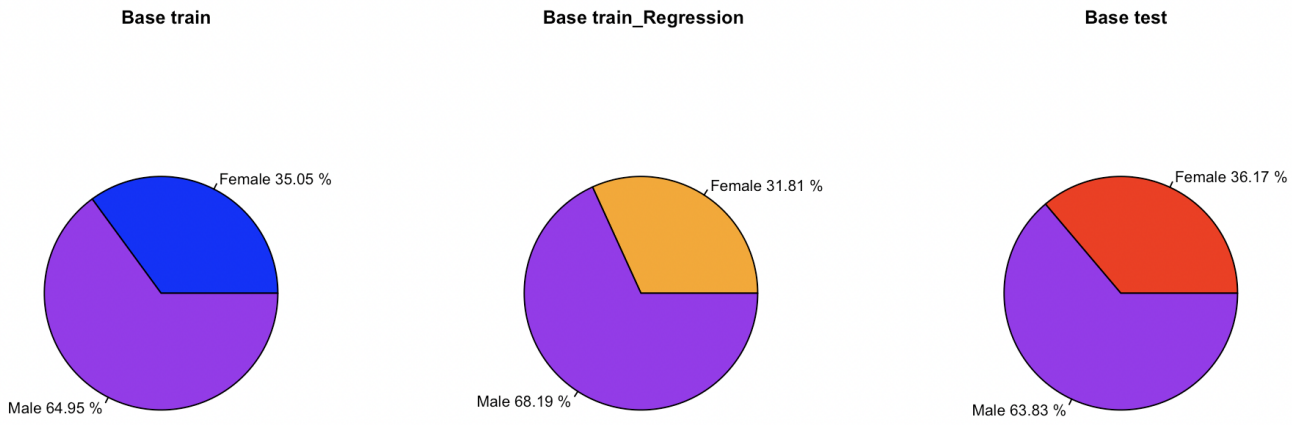

FIGURE 2 – diagramme camembert - gender

On remarque que la répartition Homme - Femme est équilibrée sur ces 3 bases avec à chaque fois plus de **60%** d'hommes que de femmes. Cette observation combinée avec l'hypothèse d'un sinistre maximum par tête nous permet de constater que les hommes enregistrent plus de sinistres que les femmes mais ce constat ne saurait être interprété car l'effectif des hommes est supérieur à celui des femmes dans la base initiale dans les mêmes proportions que le nombre des sinistres hommes dépasse le nombre des sinistres femmes. La variable gender reste toute fois utile dans notre modélisation pour capter les effets du genre du conducteur/conductrice.

carType :

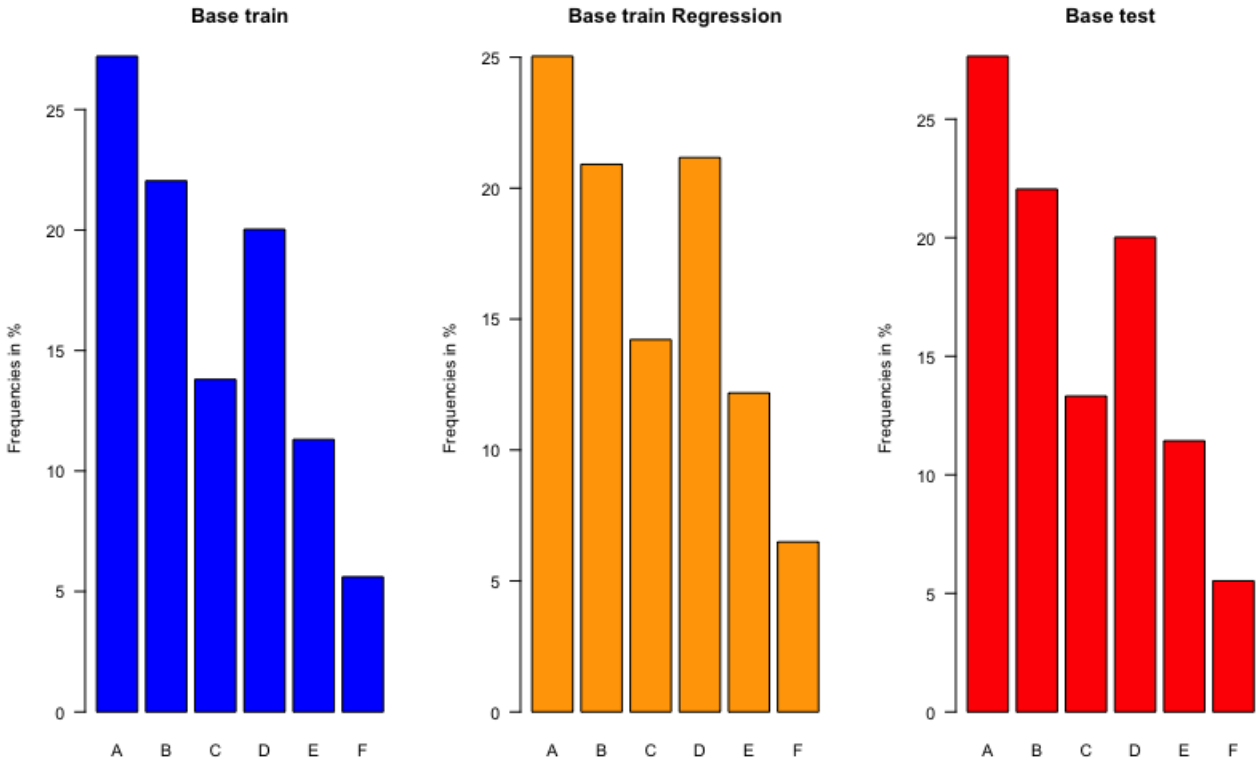

FIGURE 3 – Fréquences absolues - carType

Même constat que pour **Gender** ; on remarque que sur les 3 bases de données, les tableaux de fréquence des modalités de la variable **carType** suivent les mêmes ordres de grandeur.De plus, on fera attention aux modalités qui ont une faible fréquence ; on songera dans la partie **feature enginnering** à regrouper certaines modalités afin qu'elles forment des classes significatives d'un point de vue effectif par rapport à l'effectif total de notre base.

carCategory :

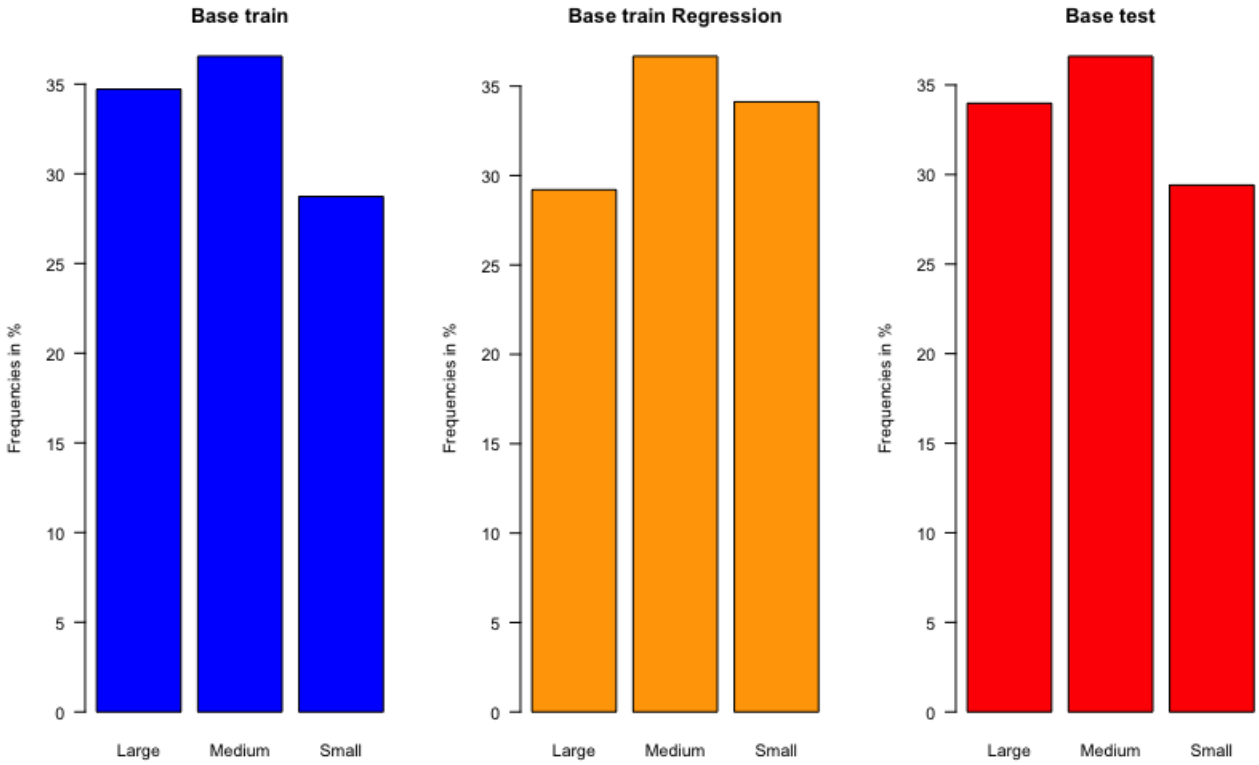

FIGURE 4 – Fréquences absolues - carCategory

On lit facilement sur les *barplots* ci - dessus que la variable **carCategory** présente le même comportement sur les bases train et test avec en moyenne 35% de véhicules dans chacune des classes *large* et *medium* et une proportion de 30% dans la catégorie *small* (petit véhicule). En ce qui concerne la base sinistre, on remarque que bien que leur effectif soit inférieur à celui des gros véhicules sur le portefeuille global (base **train**), les véhicules de petite taille connaissent plus de sinistres ; 34% de véhicules sinistrés sont de petite taille contre 29% de gros véhicules. La catégorie *medium*(véhicule taille moyenne) connaît plus de sinistres avec 36% du total de véhicules sinistrés.
On ne peut pas dire que pour les catégories de véhicules, le nombre de véhicules sinistrés est positivement corrélés à l'effectif de la catégorie dans le portefeuille globale.

Occupation :

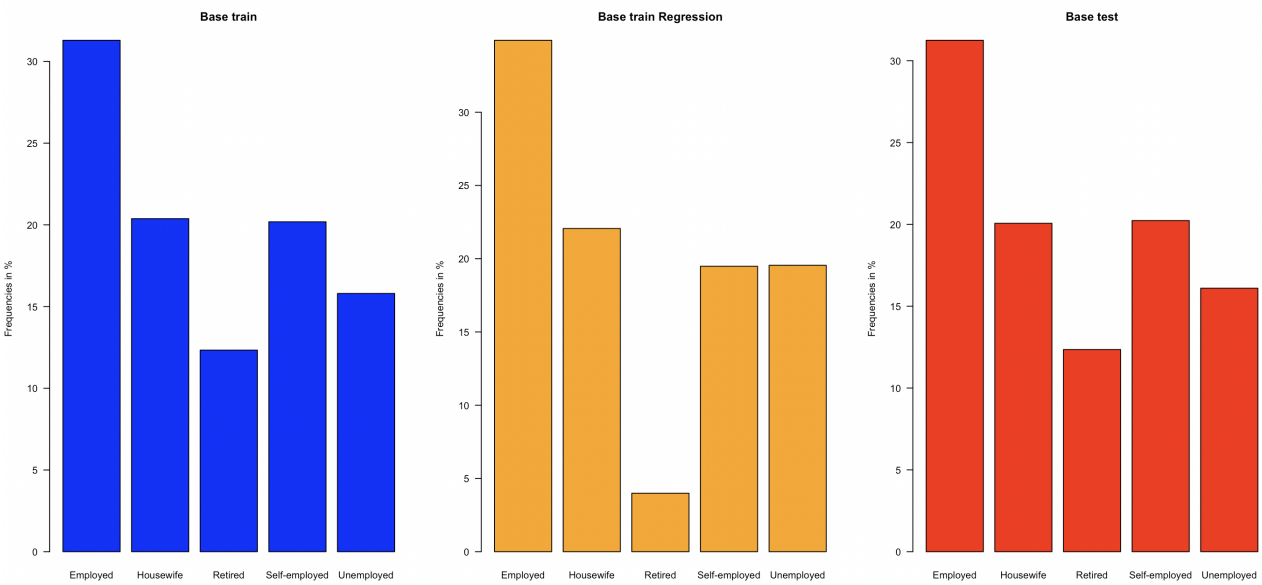

FIGURE 5 – Fréquences absolues - occupation

On observe sur ce graphique que les 3 tables sont assez homogènes en ce qui concerne la répartition des effectifs des modalités de la variable **occupation** ; à noter que seulement 3% des assurés retraités ont connu des sinistres.

carGroup :

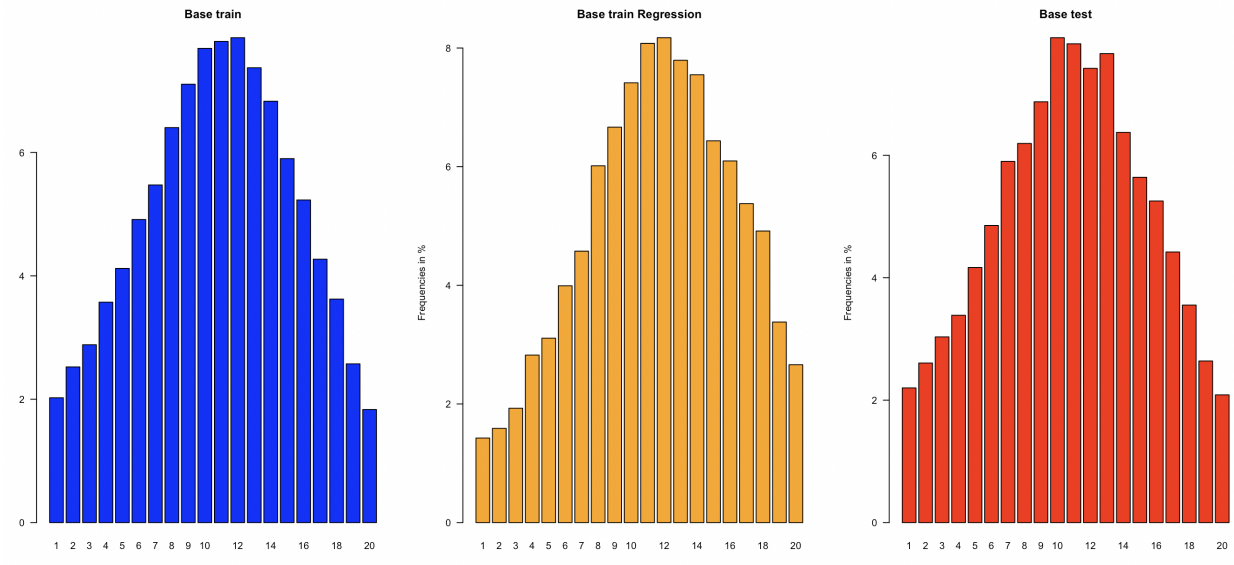

FIGURE 6 – Fréquences absolues - carGroup

La structure des effectifs des modalités semblent être la même sur les 3 tables.

La particularité de cette variable qualitative est qu'elle possède 20 modalités et chacune d'elle représente moins de 10% de l'effectif total des assurés. On peut s'attendre à faire un regroupement de modalités en classe plus significative sur cette variable.

material :

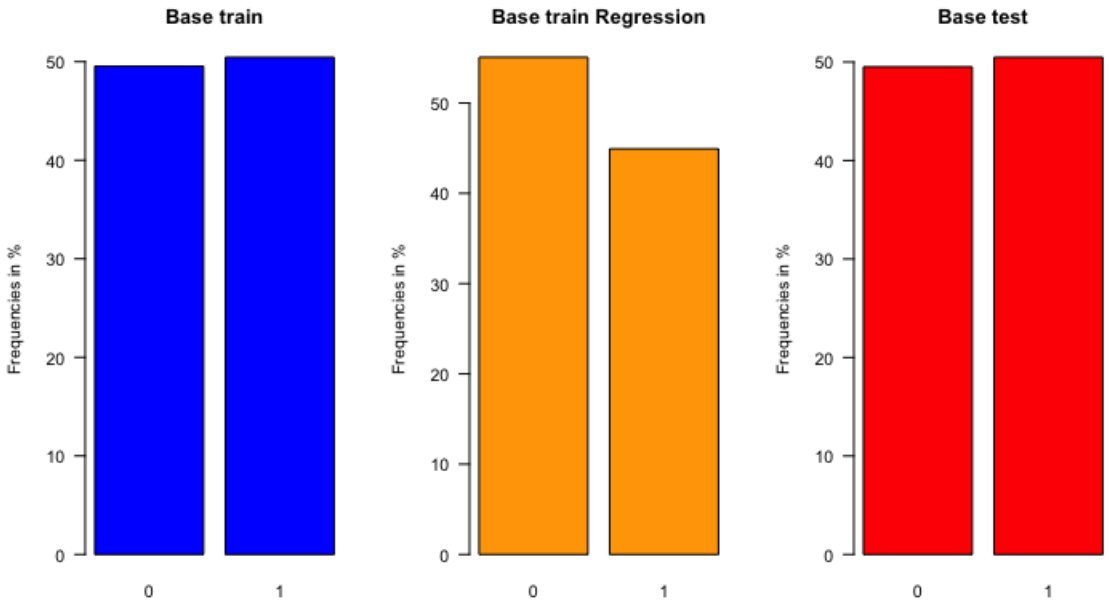

FIGURE 7 – Fréquences absolues - material

On voit grâce à ce graphique que 55% des contrats sur lesquels on a enregistré un sinistre ne possède pas la couverture matérielle supplémentaire contre 45% qui possède cette couverture alors que sur le portefeuille au globale (base **train**)il y a une répartition plus équitable (50% vs 50%) entre les contrats sans la couverture supplémentaire et les contrats avec la couverture supplémentaire.

Region :

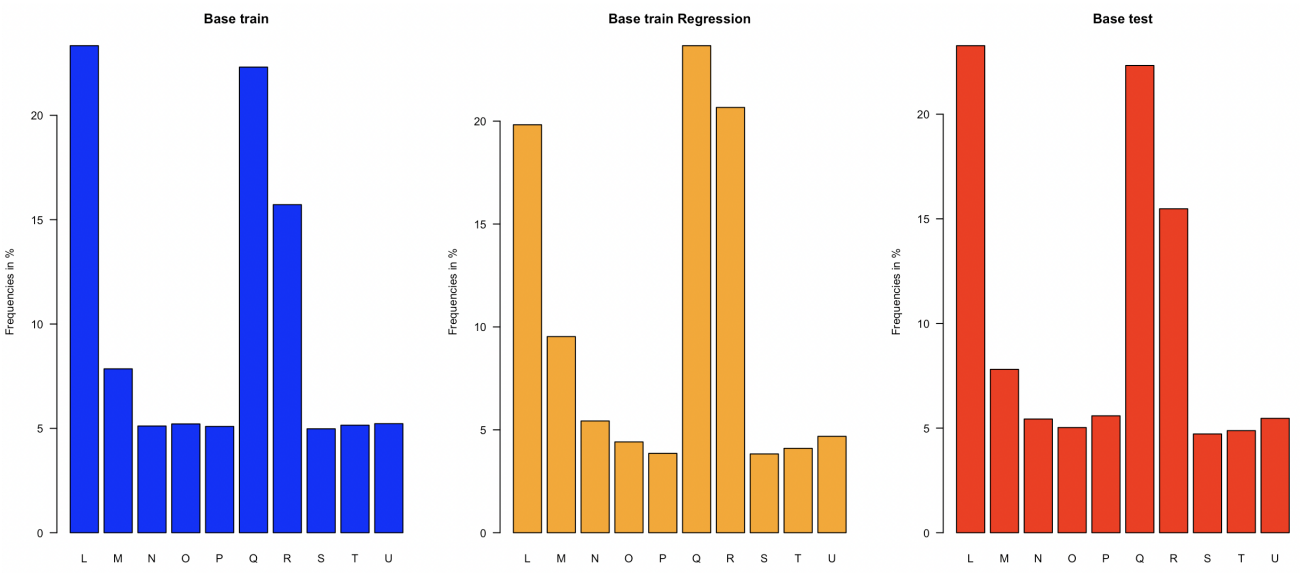

FIGURE 8 – Fréquences absolues - region

Comme pour la majeure partie des données observées ci - dessus, la variable semble avoir la même structure d'effectif sur les trois bases. On voit qu'elle a plusieurs modalités, avec quelques unes représentant moins de 10% voir moins de 5% des données. On pensera à un regroupement de ces modalités. Toutefois du fait de l'importance de la variable région, en tarification auto, il sera peut être pertinent de laisser toutes les régions telles qu'elles sont présentées et ne pas tenter un regroupement qui pourrait réduire l'impact de la région sur nos tarifs.

SubRegion :

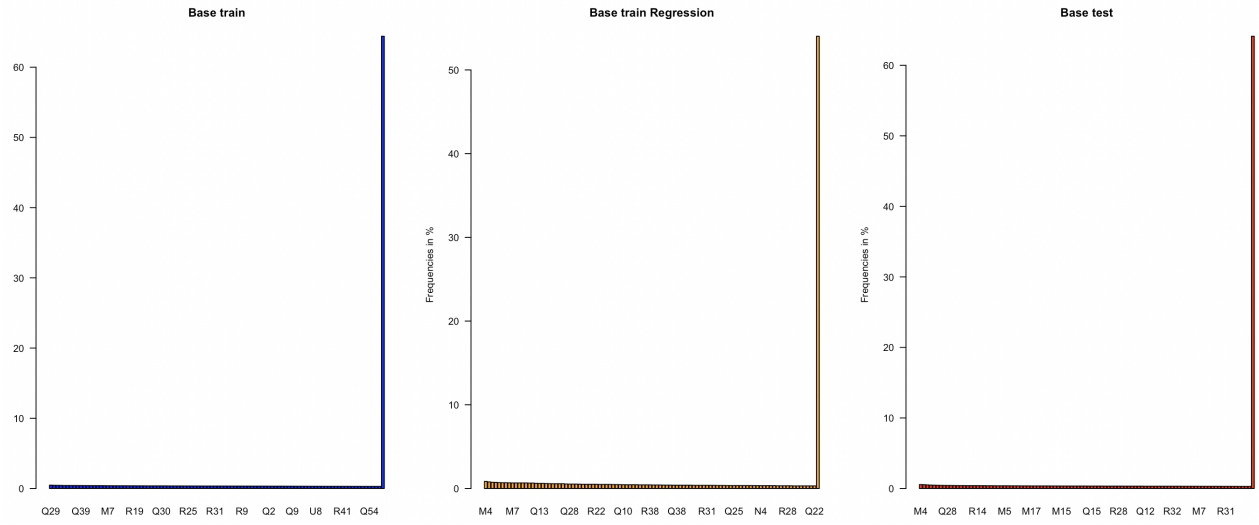

FIGURE 9 – Fréquences absolues - subregion

Sur les trois bases, elle présente un nombre extrêmement élevé de modalités qui sont pour la plupart non significative en terme d'effectif.

2.5.2 Analyses univariées des variables quantitatives

Dans cette section, on va observer les variables quantitatives sur la base **train**, la base**test** et la base de sinistres non - nuls **train_regression** au moyen de quelques graphiques ; en particulier les *plots* et *boxplots* (boites à moustaches)

Age :

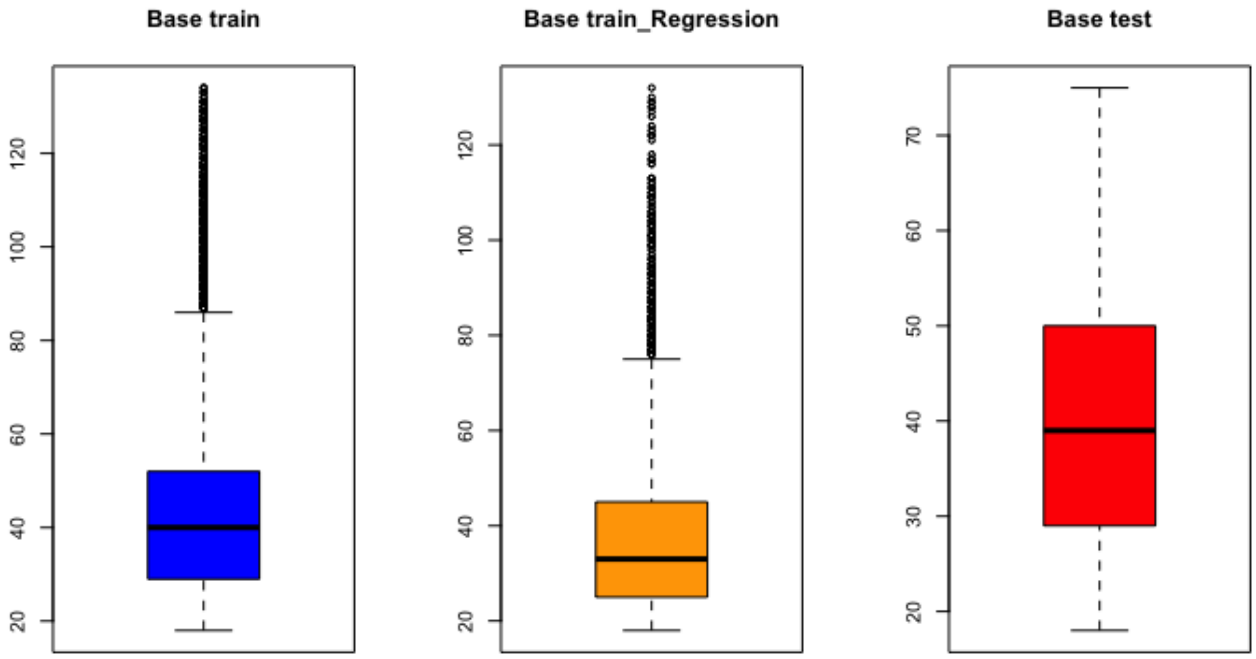

FIGURE 10 – Boite à Moustache - age

Sur les boites à moustache, on peut remarquer que la variable âge a presque le même comportement sur les 3 bases pour les observations entre 0 et 80 ans environs ; toutefois, contrairement à la base test, sur les bases trains, on remarque qu'il existe quelques assurés qui sont au dessus du seuil des 75 ans qui constitue la **moustache supérieure** de la variable âge dans la base **test**. Toutefois, ces observations représentent une faible proportion (moins de 4%) de notre base **train**.

carValue :

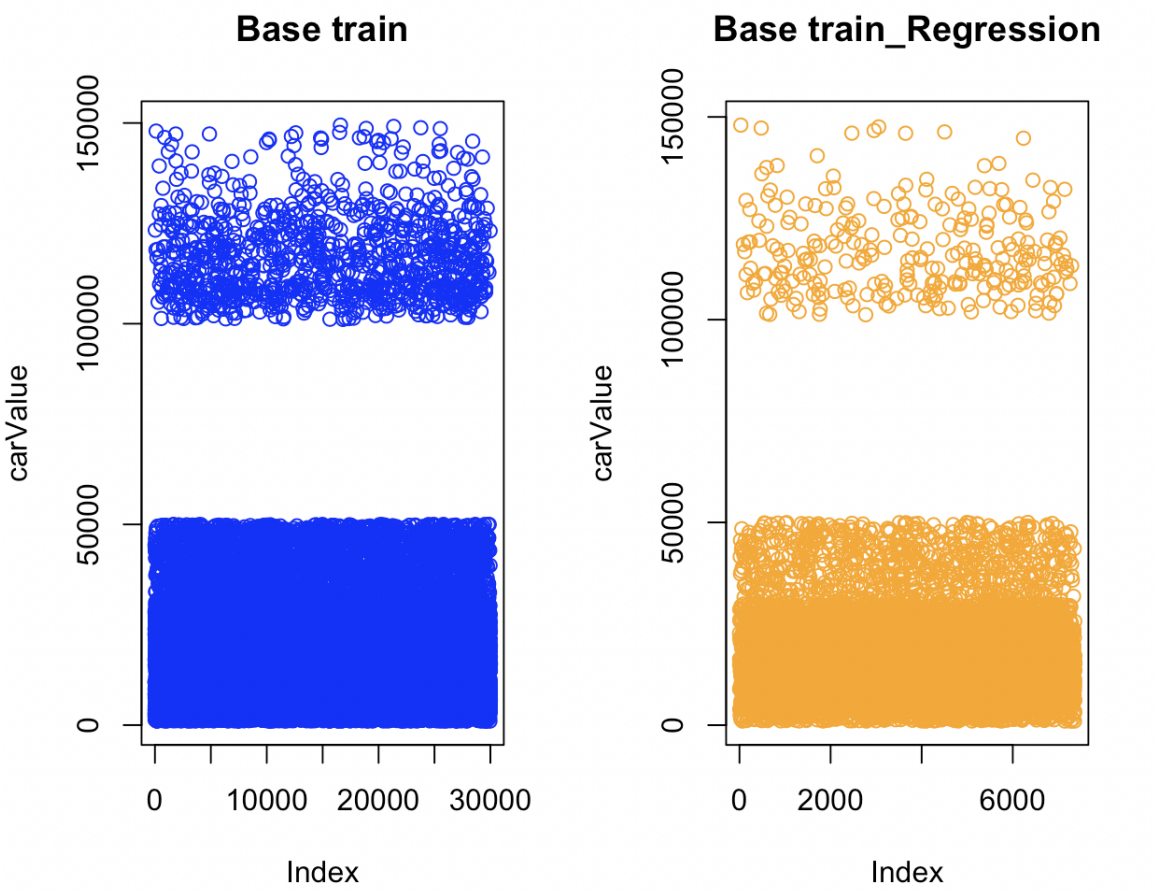

FIGURE 11 – Représentation graphique - carValue

On observe une particularité sur les *plots* de **carValue** dans les bases train : 2 sous groupes se dégagent ; les véhicules qui coûtent moins de 50000 et ceux qui coûtent plus de 100000 avec aucun véhicule entre les 2 montants ; cette remarque est à prendre en compte dans la mise en oeuvre de nos modèles (voir feature engeneering).

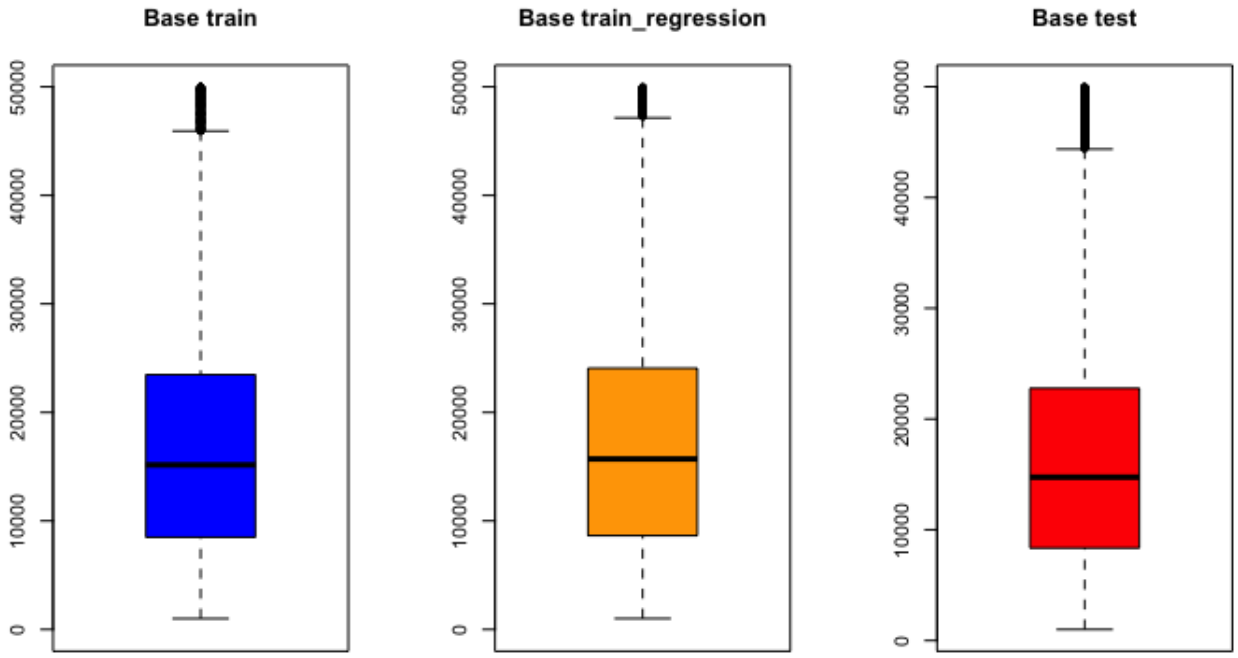

FIGURE 12 – Boite à Moustache - carValue

Sur la base test comme sur la base train, pour les observations en dessous de 50000 la variable **carValue** a une structure presque identique sur les 3 bases avec une médiane autour de 15000, une moustache supérieure autour de 45000 et quelques valeurs au dessus de la moustache supérieure.

CityDensity :

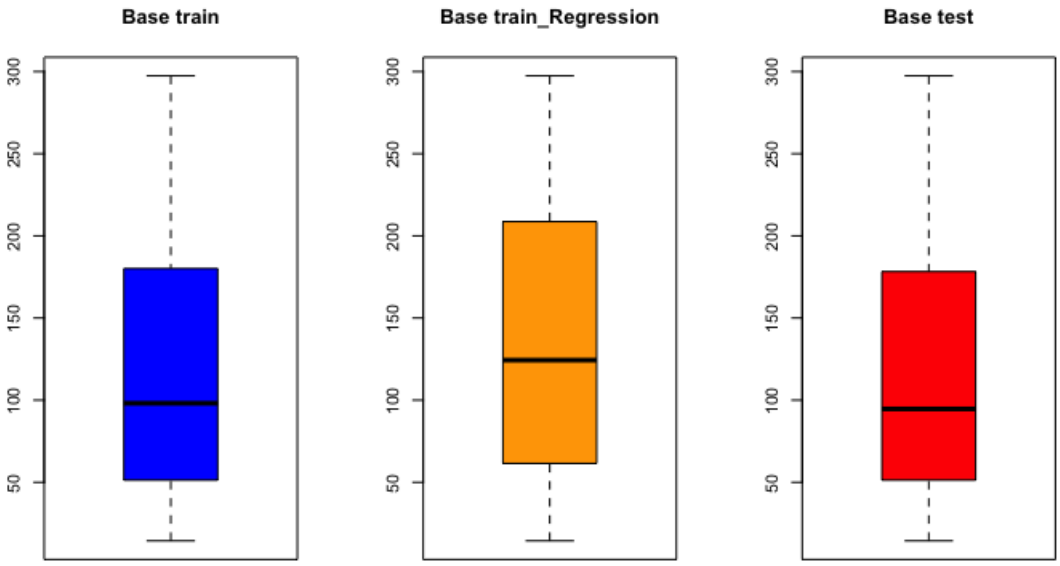

FIGURE 13 – Boite à Moustache - cityDensity

On remarque que sur la base des sinistres non - nuls la médiane et les quantiles de la variable **CityDensity** sont plus élevés que sur les bases train et test. Cette observation suggère que la fréquence de sinistre est d'autant plus élevée qu'il y a d'habitants par km^2.

ClaimValue :

ClaimValue est notre variable d'intérêt dans la partie régression ; elle n'existe pas encore dans la base test. Afin d'avoir une idée sur sa loi, nous allons observer sa densité sur la base train et la base de sinistres non - nuls.

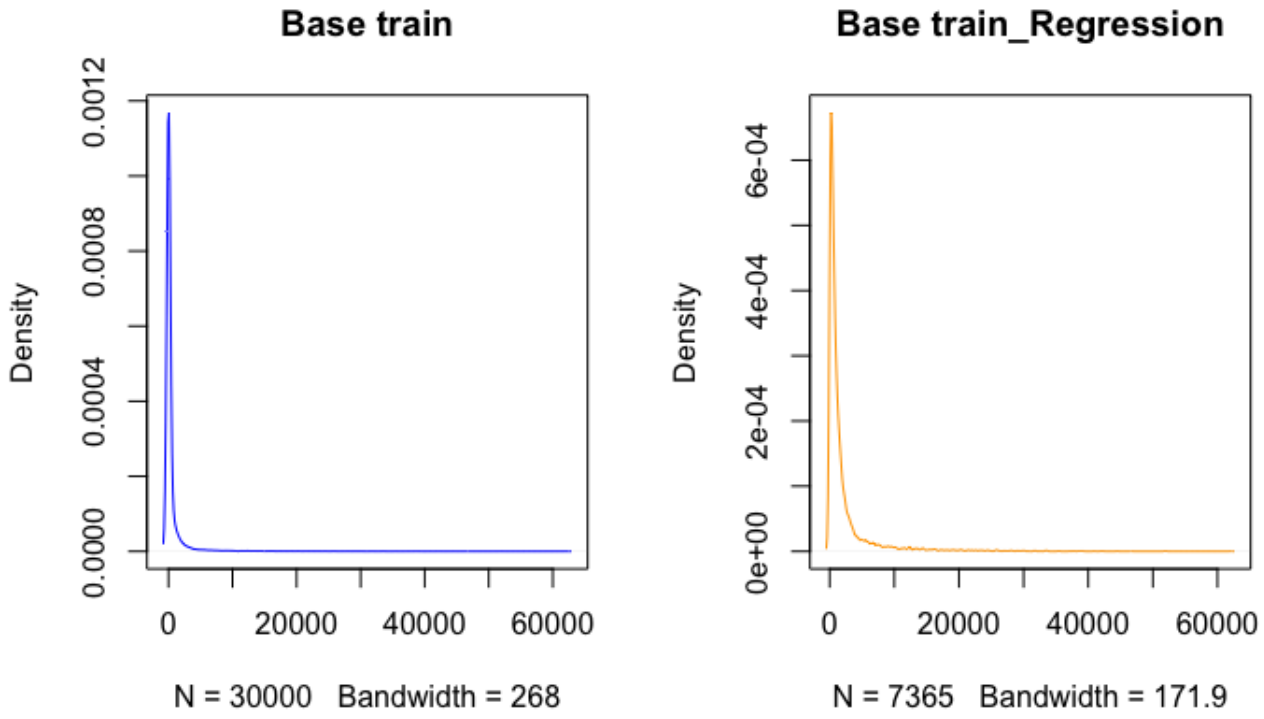

FIGURE 14 – Densité - claimValue

La densité a la même forme sur les 2 bases mais à des échelles différentes du fait de la présence de nombreuses observations pour lesquelles claimValue est nulle dans la base train ; pour cela on se servira de la base de sinistres non nuls pour modéliser claimValue ; dans la suite de cette section, on représente les graphiques uniquement à partir de **train_Regression**. La forme de la densité de **claimValue** nous permet de suggérer 2 lois usuelles connues pour approximer cette dernière. Dans un premier temps, on rapproche la densité de claimValue à la loi Gamma dont les paramètres dépendent de la moyenne et de la variance de **claimvalue** et d'un autre coté, on rapproche le logarithme de **claimValue** à la loi normale de moyenne *mean(claimValue)* et de variance *sd(claimValue)*. On trace ci-dessous les résultats du "fittings" de ces loi sur claimValue et *log(claimValue)*.

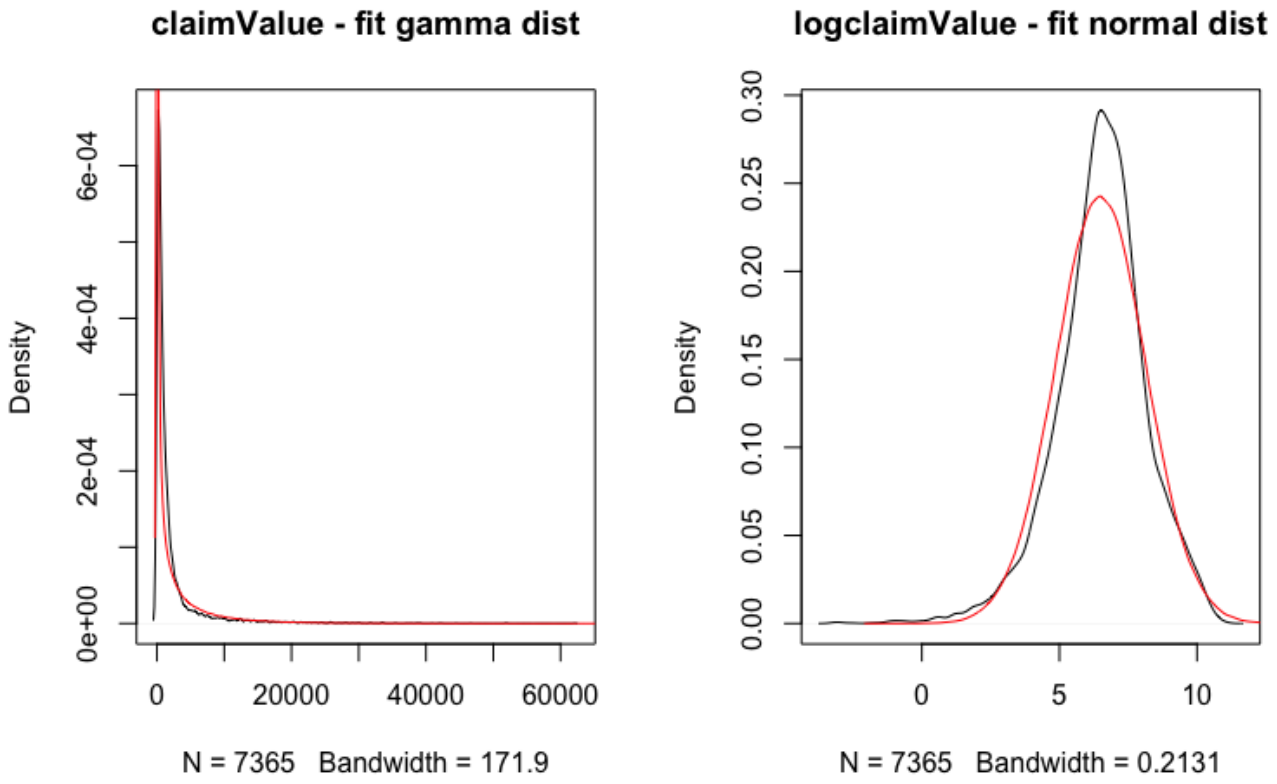

FIGURE 15 – Fitting des lois

On confirme grâce à ces graphiques qu'on pourrait approcher claimValue elle même ou sa transformation logarithmique respectivement par la loi gamma ou la loi normale.

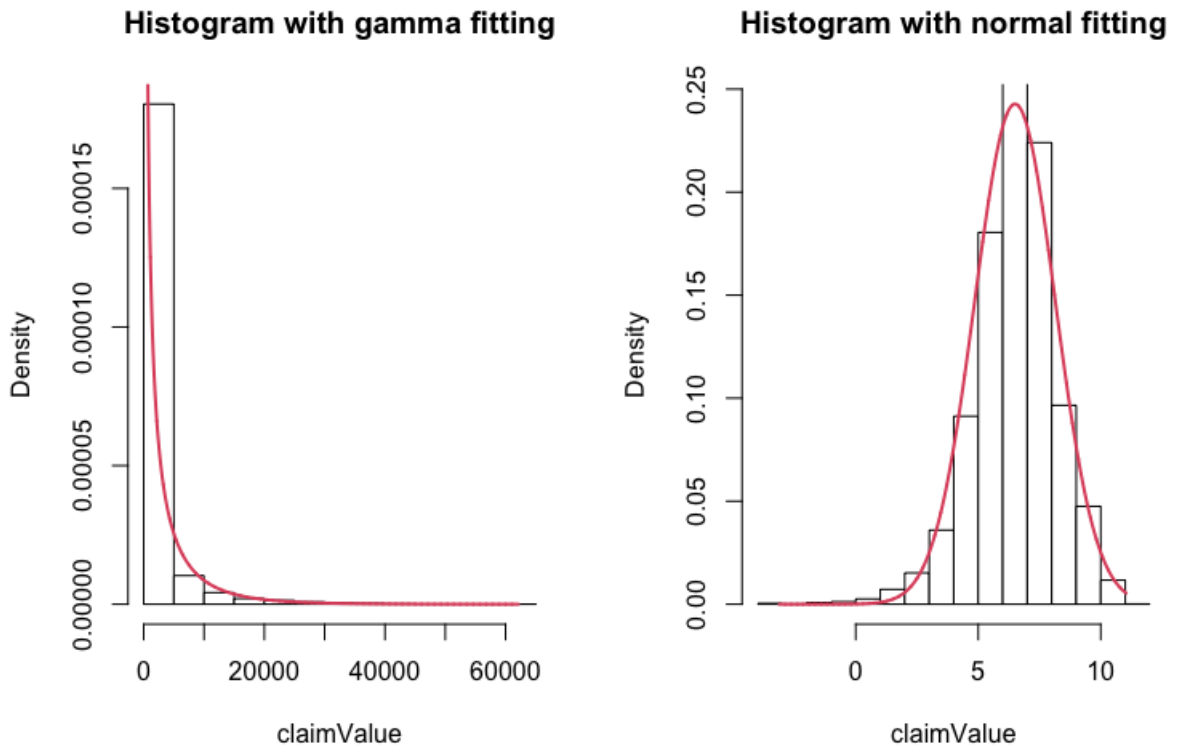

FIGURE 16 – Histogramme - claimValue

Sur les histogrammes, on remarque que la loi gamma a tendance à bien capter les queues de distribution mais à sur-estimer la fréquence dans les classes les plus abondantes de l'histogramme de **claimValue** et la loi normale a tendance à sous estimer la distribution dans les classes les plus abondantes de l'histogramme et capte moins bien les queues de distribution.

2.5.3 Analyses Multivariées

On fait dans cette section une étude des corrélations entre toutes nos variables ; On ne mènera ces études de corrélations que sur les bases **train** et **train_Regression** qui sont les bases sur lesquelles on entraînera nos modèles.
Il s'agira d'abord d'étudier les corrélations entre les différentes paires de variables quantitatives ensuite celles des variables qualitatives et enfin entre les paires constituées d'une variable quantitative et d'une variable catégorielle.

Variables Quantitatives :

Pour mesurer la corrélation entre chaque paire de variables quantitatives, on calcule le **coefficient de corrélation de Pearson** :

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

Les résultats sont présentés dans la matrice de corrélation ou corrélogramme ci - contre :

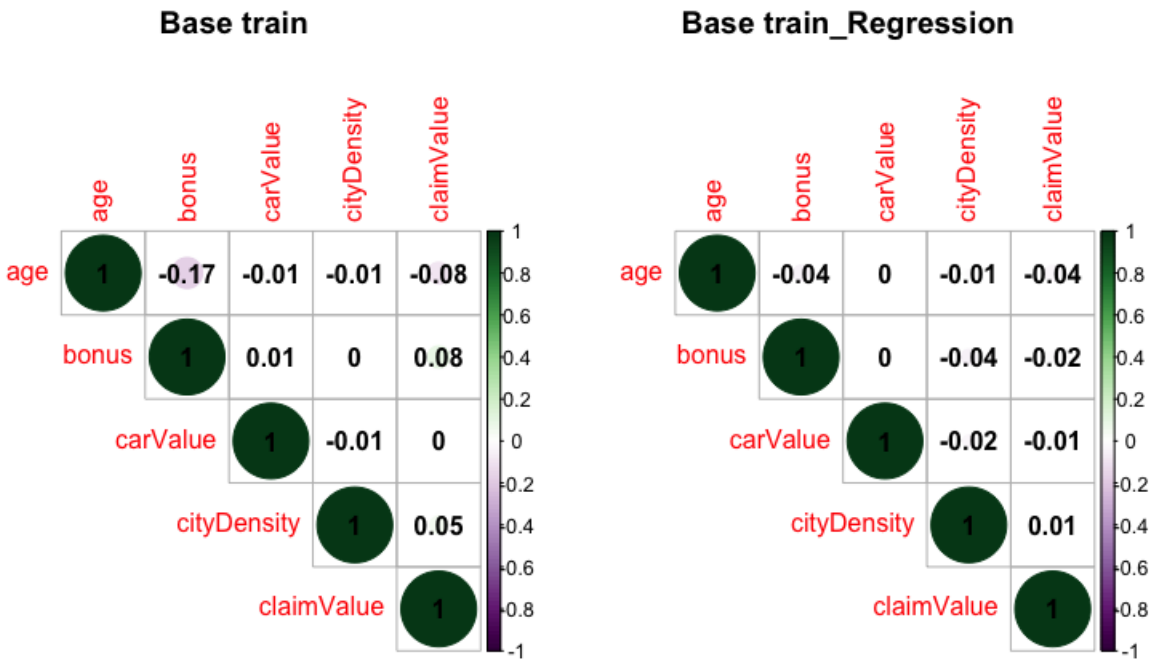

FIGURE 17 – Corrélogramme - coefficient de Pearson

Qu'il s'agisse de la base train ou celle des sinistres non - nuls on remarque qu'il n'y a pas de corrélation forte sur les variables quantitatives au sens du coefficient de corrélation de Pearson.
Pour confirmer la structure de corrélation très faible ci-dessus, on confronte les résultats du coefficient de pearson à ceux du coefficient de spearman ci - dessous affichés.

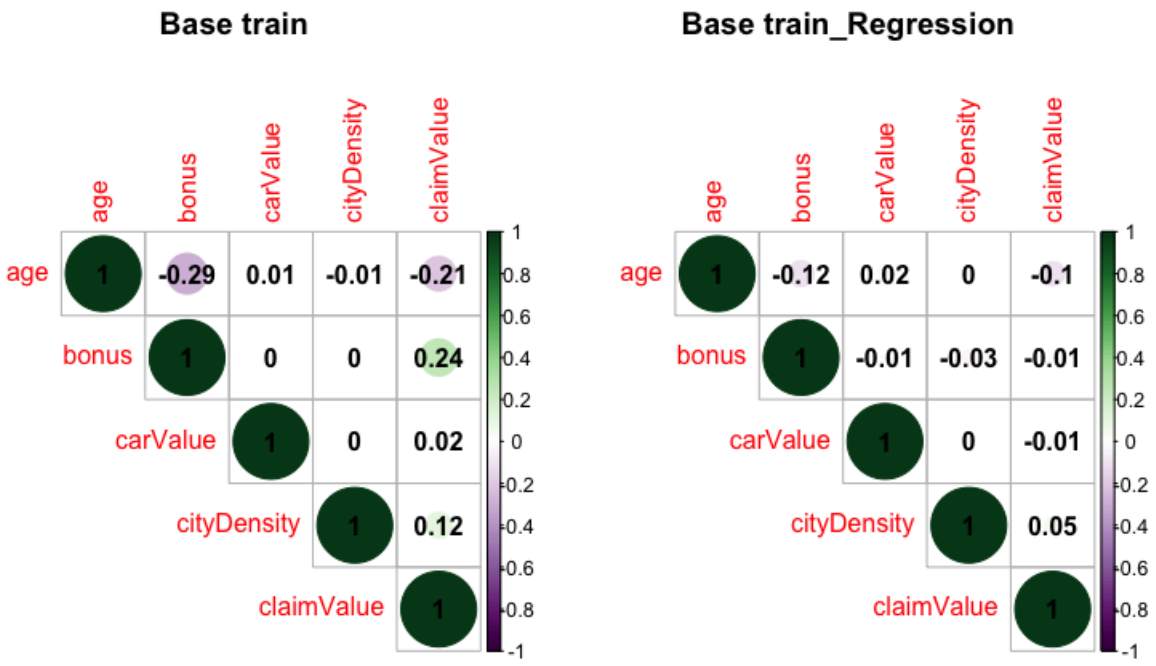

FIGURE 18 – Corrélogramme - coefficient de Spearman

Selon le coefficient de Spearman, la structure de corrélation reste tout aussi faible entre les variables quantitatives.

Variables Qualitatives :

Notre outil de mesure de la corrélation entre les variables catégorielles est un **test du Khi-2**. On teste sur les paires de variables qualitatives l'hypothèse **H0** : "les 2 variables sont indépendantes" contre **H1** : "les 2 variables sont corrélées". Dans le tableau ci - dessous, on affiche les *p-values* des tests du khi-2 à 99% pour chaque couple de variable qualitative ; la règle de décision est la suivante : **si la p-value est supérieure à 0,01 on ne pourra pas rejeter l'hypothèse H0 d'indépendance des variables.**

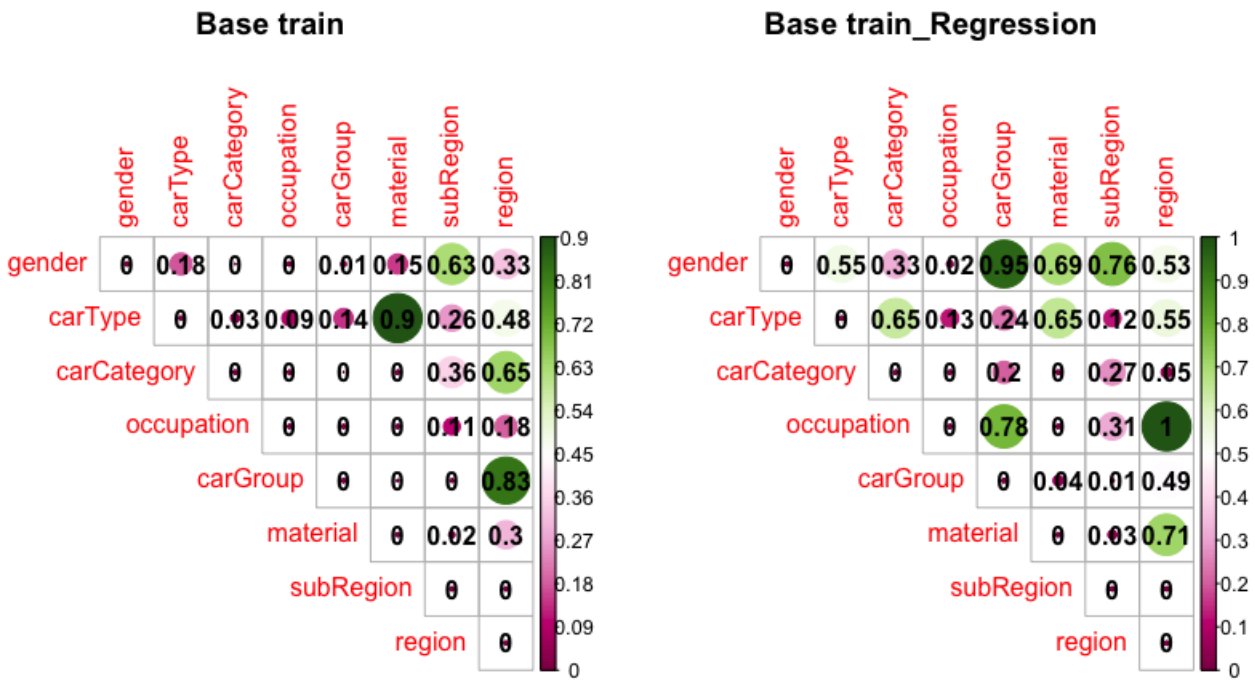

FIGURE 19 – p-Value - test du Khi-2 de corrélation

On observe plusieurs dépendances au niveau des variables catégorielles dans la base **train** ; chaque variable dans cette base à l'exception de **carType** est corrélée avec au moins une autre. Toutefois dans la base des sinistres non - nuls, seulement quelques variables sont corrélées à savoir **carCategory**, **occupation** et **material** et sont 2 à 2 dépendantes et **subregion** est naturellement corrélé à **region** mais aussi à **material**. On en déduit que la corrélation semble faible entre les profils des assurés sinistrés.

Variables Quantitatives et Qualitatives :

Dans la littérature, le **rapport de corrélation** est un indicateur statistique qui mesure l'intensité de la liaison entre une variable quantitative et une variable qualitative.

$$\eta^2 = \frac{\sum_{k=1}^{p} n_k(\bar{x}_k - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

où $\bar{x}_k$ est la moyenne du groupe k et n_k, le nombre d'individus appartenant à ce même groupe et $\bar{x}$ la moyenne globale.

— **Si le rapport est proche de 0, les 2 variables ne sont pas liées**
— **Si le rapport est proche de 1, les 2 variables sont liées**

On présente dans les matrices ci-dessous les rapports de corrélation entre variables quantitatives et qualitatives dans la base **train** et dans la base de sinistres non-nuls **train_Regression**.

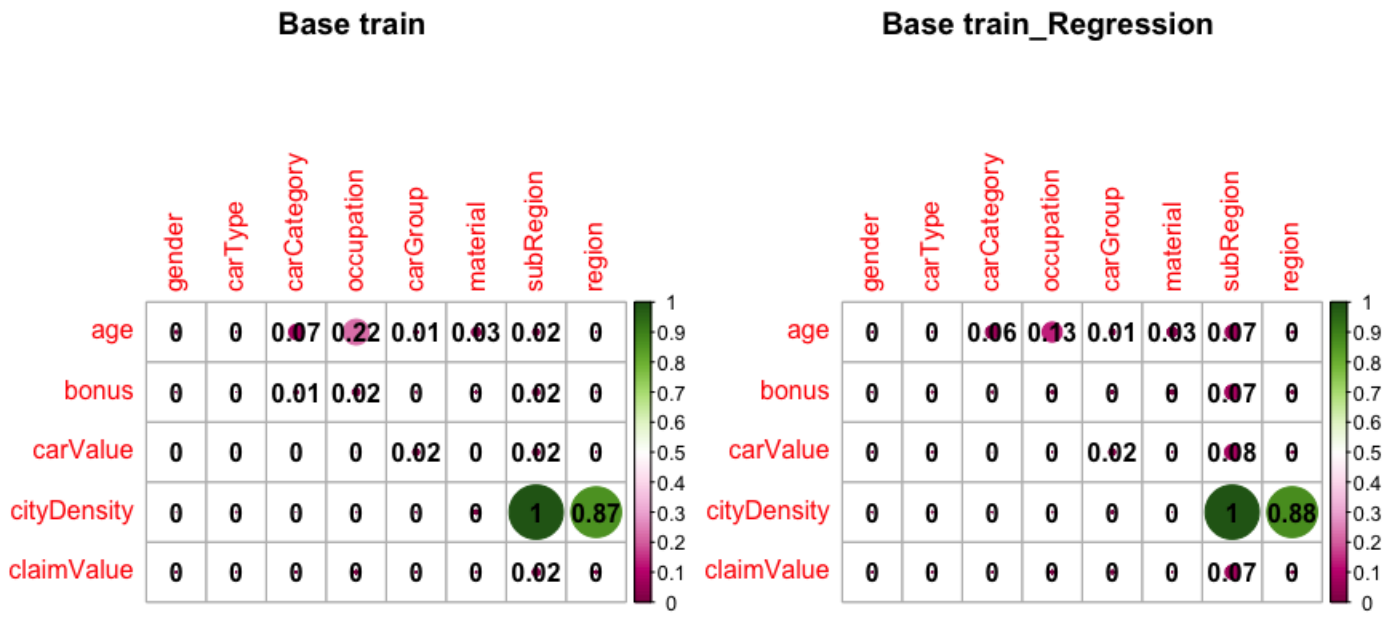

FIGURE 20 – rapport de corrélation

On remarque aisément que les résultats sont très proches sur les 2 bases et surtout qu'il n'y a aucune corrélation entre les variables quantitatives et qualitatives à l'exception de **cityDensity** (le nombre d'habitants par km^2) qui semble fortement corrélé à **region** et **subRegion** ; ce qui est plausible.

Globalement, l'absence de très forte corrélation apparaît comme un avantage pour nos modèles ; en effet, dans un modèle simple comme la régression linéaire, les corrélations entre les variables peuvent faire émerger des problèmes de multi-colinéarité et dans d'autres modèles plus complexes, elles peuvent être facteur de **surapprentissage**.

2.6 Feature Engeenering Part 2 -

Après avoir observer nos variables, on prépare nos modélisations en restructurant certaines de nos variables pour soit gagner en pertinence dans nos analyses ou soit simplifier nos modèles.

ID :

le numéro d'identification n'est pas une caractéristique intrinsèque de nos assurés donc ne fait pas partir des variables utiles pour nos modèles ; il s'en suit qu'on le supprime de la base train.

subRegion :

le retraitement de cette variable qualitative s'impose car en plus d'être presque parfaitement corrélée à **region** et **cityDensity**, elle est constituée d'une centaine de modalités très peu significatives en terme

d'effectifs ; on la supprime purement et simplement de notre base car elle serait de nature à créer de la multicolinéarité, à favoriser éventuellement le surapprentissage et à complexifier nos modèles. On est d'autant plus libre de la supprimer sans crainte de pertes d'informations car elle semble corrélée parfaitement avec **region**.

Remarque :
On retient tout de même cityDensity et region dans nos bases de modélisation car bien qu'elles semblent corrélées, elles peuvent capter différents effets importants dans les prédictions ; en effet, le nombre d'habitants par km^2 n'est pas la seule caractéristique d'une région et de la même manière le nombre d'habitants par m^2 peut caractériser plus qu'une région.

Bonus :

Dans la base de **train_Regression.**, on supprime la variable **bonus** car le **bonus-malus** n'est pas pris en compte dans le calcul de la **prime pure** qui est notre objectif sur la partie régression.

sinistre :

Dans une perspective de modéliser la probabilité d'avoir un sinistre (partie classification), nous avons crée la variable catégorielle bi-modale **sinistre** qui vaut 1 lorsque l'assuré a déclaré un sinistre et 0 le cas échéant. On montre ci-dessous un résumé statistique de la nouvelle variable :

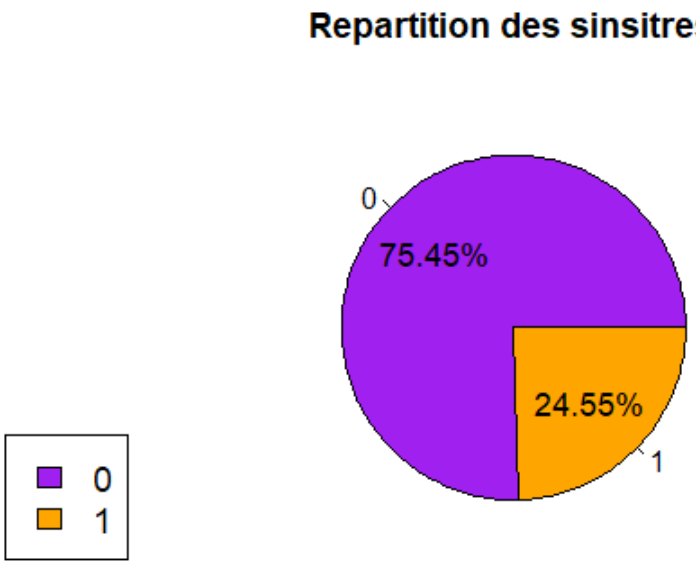

FIGURE 21 – La répartition des sinistres

Dans la partie classification, nous allons déterminer la probabilité des individus d'être dans l'une des deux classes

les pistes non exploitées :

— **Age** : Nous avons envisagé de créer des tranches d'âges pour mieux capter les effets de l'âge sur claimValue, en particulier pour les personnes âgées ; mais on remarque que le regroupement en tranche d'âge n'est pas forcément pertinent car claimValue ne semble pas trop varier d'une tranche d'âge à une autre ; la plus grande variation de claimValue s'observe lorsqu'on fait la distinction dans la base régression des moins de 75 et des plus de 75. Le box-plot ci - dessous l'illustre ; toutefois pour les besoins de notre modélisation, il n'était pas utile de ré-configurer âge en tranche d'âge.

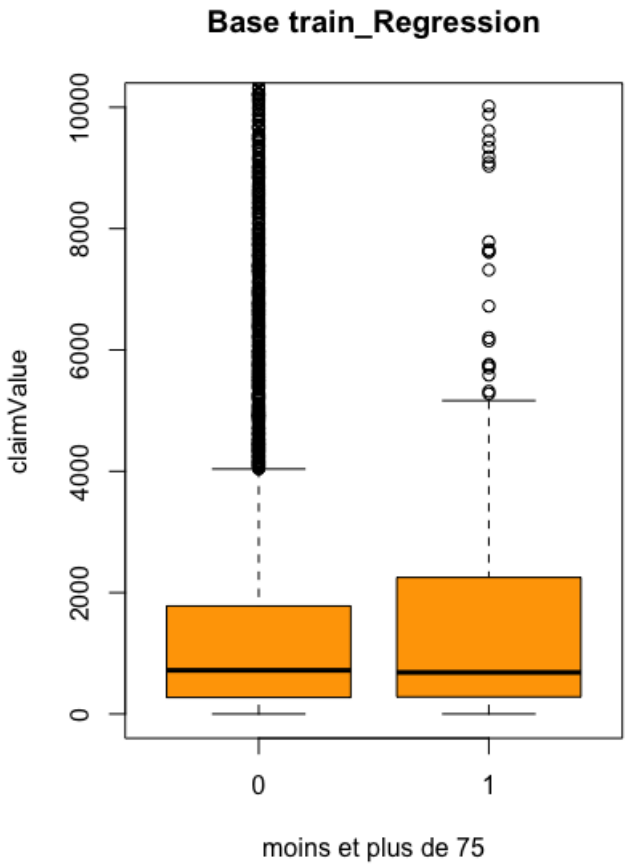

FIGURE 22 – Boite à moustache -claimValue en fonction de tranche d'âge

— **carGroup** : Nous avons eu l'idée de regrouper certaines modalités de carGroup pour en faire des classes plus significatives mais cela n'améliore pas nos résultats et projections.
— **region** : Voulant capter l'effet de toutes les régions possibles dans nos modèles, nous avons décidé de ne pas regrouper les modalités de région suivant le critère des 5%

3 ESTIMATION DU COÛT MOYEN

L'objectif de cette section est de choisir le modèle permettant d'avoir la meilleure prédiction avec une erreur minimale pour la variable "ClaimValue". Ainsi, le critère de sélection de notre modèle sera le RMSE (**Root Mean Squared Error**).

3.1 Modèles de régression

3.1.1 Régression log-normale

Les modèles linéaires évaluent une liaison linéaire entre la variable d'intérêt et les variables explicatives par une relation de la forme :

$$Y = f(X) \tag{1}$$

avec

- f définie une fonction linéaire.
- X notre vecteur de variables explicatives.

Dans le cadre de notre étude, nous avons mis en oeuvre la régression linéaire suivante :

$$log(claimValue) = f(X) \tag{2}$$

car log(claimValue) vérifie les hypothèses du modèle linéaire.

Vérification des hypothèses sous-jacentes au modèle

Homoscédasticité de l'erreur

On vérifie en observant le "graphique de dispersion des résidus".

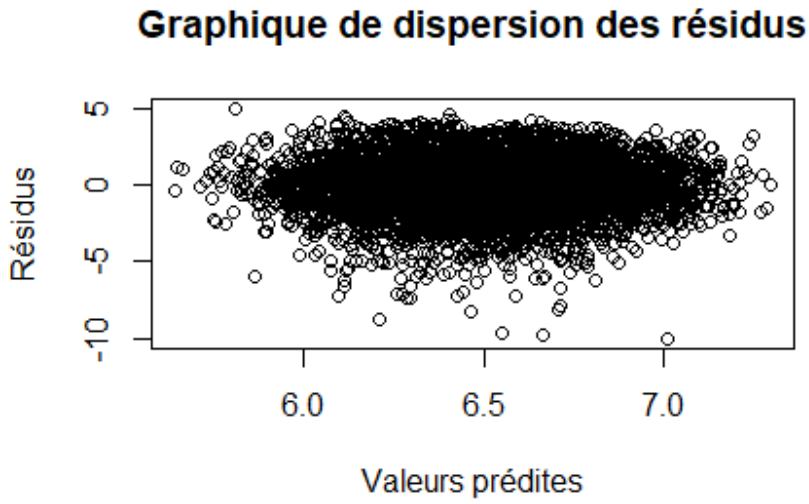

FIGURE 23 – Dispersion des résidus

Ce graphique montre que les résidus sont distribués aléatoirement sensiblement autour de 0. Ce qui permet de conclure qu'ils ont une variance constante, d'où l'homoscédasticité.

Indépendance des résidus

Le graphe des auto-corrélations entre les résidus, obtenu à l'aide de la fonction "acf" sur R, permet de trancher sur cette question.

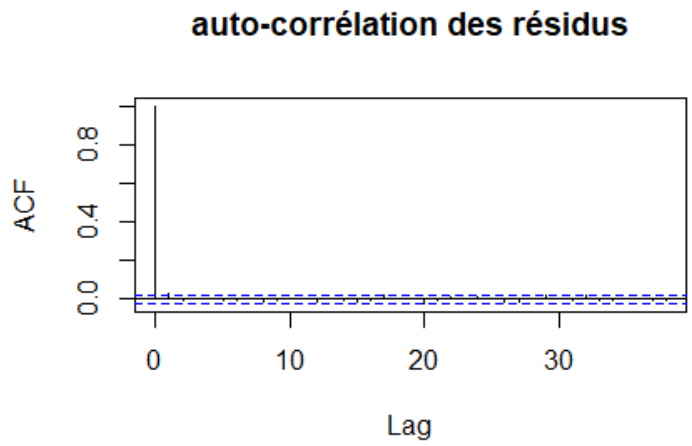

FIGURE 24 – Autocorrélation des erreurs

Ici, on voit que les résidus sont très faiblement corrélés, ce qui est favorable à notre hypothèse d'indépendance.

Normalité des résidus

Pour évaluer la normalité des résidus, on utilise comme graphique de diagnostic, en plus du "fitting" de la loi normale sur la densité (figure 15) et "fitting" de la loi normale sur l'histogramme (figure 16), le **Q-Q plot** que nous représentons ci-dessous .

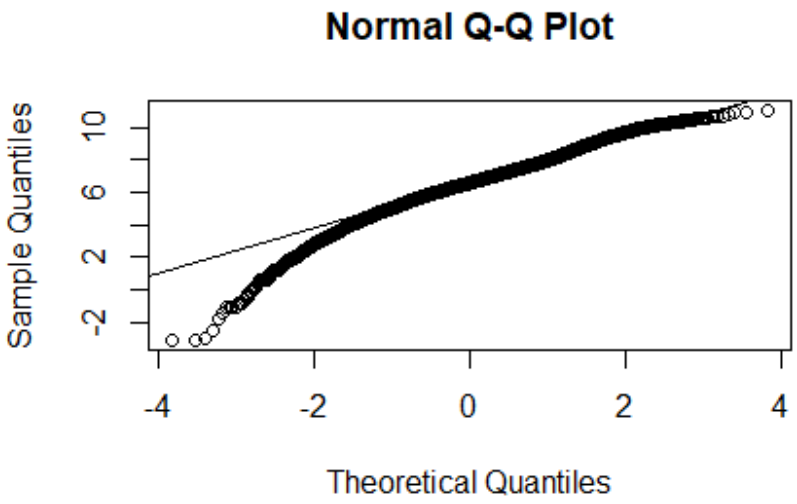

FIGURE 25 – Q-Q plot de log(claimValue)

On remarque que la "première bissectrice" est sensiblement confondue au graphique des quantiles de log(claimValue) au-delà de -1, ce qui signifie que le logarithme de la variable claimValue peut être bien estimé par une loi normale dont l'espérance m et l'écart-typeσ se calculent respectivement de la façon suivante sur R :

$$m = mean(log(claimValue))$$

$$\sigma = sd(log(claimValue))$$

Cela va nous permettre de construire un modèle de régression simple basé sur nos données $trainRegression$ à l'aide de la fonction lm de R.

Le Modèle

En pratique, on commence par construire un modèle de régression linéaire avec toutes les variables explicatives qu'on améliore en second lieu suivant le critère de l'AIC. Le critère de l'AIC ne permet pas d'améliorer le RMSE par rapport au modèle avec toutes les variables explicatives. On retient alors le modèle de régression avec toutes les variables explicatives.

Toutefois, la limite de notre modèle de régression linéaire est qu'elle ne capte pas les effets des valeurs extrêmes, allusion fait au "fitting" de la loi normale sur la densité et l'histogramme de $claimValue$ (figures 15 & 16). C'est ici qu'intervient la loi Gamma qui permet de mieux capter ces éléments à condition de sélectionner les paramètres adéquats.

3.1.2 Régression Gamma

Les modèles linéaires généralisés (MLG) se construisent à partir de lois de la famille exponentielle et s'écrivent sous la forme suivante :

$$g(E[Y]) = \beta^\top X = \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \ldots + \beta_p \cdot X_p$$

où

$$g : R \to R$$

est la **fonction lien** déterministe.

Comme nous l'avions vu plus haut sur les figures 15 et 16, la variable claimValue se rapproche beaucoup d'une loi Gamma. On estime donc un MLG **log-gamma** (où log est pour la fonction lien logarithme) sur la variable $claimValue$ à l'aide de la fonction R "glm". On va également essayer d'améliorer la qualité de notre MLG en supprimant certaines variables afin d'optimiser le critère de l'AIC. Comme pour le modèle linéaire, le critère de l'AIC ne permet pas d'ameliorer le RMSE de notre modèle MLG contenant toutes les variables explicatives.

3.1.3 Comparaison des modèles de régression

On vous présente ci - dessous les box-plots des RMSE qu'on a obtenu après cross - validation sur nos modèles de régressions

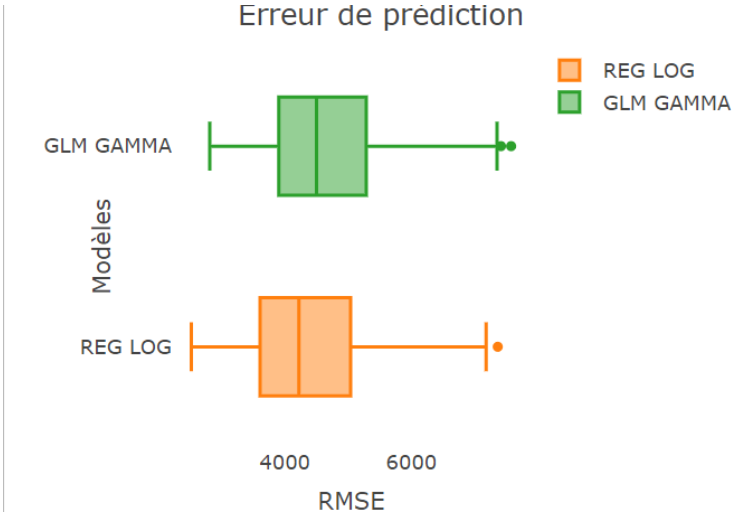

FIGURE 26 – RMSE GAMMA VS LOG-NORMALE

La régression log-normale a une erreur de prédiction plus faible que la régression Gamma (3609.671 contre 4502.279 pour la Gamma). Par la suite, nous allons donc comparer le modèle de régression log-normale aux modèles de régression pénalisés et aux modèles d'apprentissage statistique.

3.2 Modèles de régression pénalisée

On vise à travers ces modèles d'améliorer les modèles de régression en attaquant d'une part les problèmes de multicolinéarité mais aussi les problèmes de sur-apprentissage (overfitting). On implémentera dans le cas de notre étude le Ridge et l'Elastic Net.

3.2.1 Ridge

La régression Ridge vise à minimiser la fonction de coût en ajoutant une pénalité L2 (norme euclidienne) à la somme des carrés des coefficients du modèle. La pénalité Ridge conduit à des coefficients de modèle plus petits et peut stabiliser le modèle en présence de variables fortement corrélées.

Application du modèle :
On lance une première régression Ridge avec la fonction *glmnet* de R en prenant le soin d'initialiser le paramètre alpha à 0. On obtient par défaut 100 valeurs pour le paramètre lambda qu'on va chercher à "tuner"

par la suite. On procède alors à une cross-validation et on retient comme lambda optimale la valeur qui permet de minimiser notre RMSE au bout de notre cross-validation (*lambda.min* sur R). On applique à nouveau une cross - validation avec la valeur optimale et on calcule le RMSE sur chacune des sorties. On prend la moyenne des RMSE et on obtient pour notre modèle Ridge un RMSE moyen de **1417.977**.

3.2.2 Elastic Net

L'Elastic Net est une extension de la régression Ridge (L2) et de la régression Lasso (L1) qui combine les termes de pénalité des deux méthodes. Il a été développé pour tirer parti des avantages des deux approches tout en atténuant leurs limitations respectives. L'Elastic Net introduit deux paramètres de régularisation, et , pour contrôler les termes de pénalité L1 et L2, respectivement.

Application du modèle :
De manière analogue à la régression ridge, on estimera un premier modèle avec alpha pris entre 0 et 1 exclus pour éviter de faire un Ridge ou un Lasso. Ensuite, on va faire un tunning des hyperparamètres et lancer une cross validation pour avoir plusieurs RMSE et prendre la moyenne. Notre RMSE moyen obtenu est : **1417.356** ; très proche du RMSE du ridge.

3.2.3 Comparaison des modèles de régression

On vous présente ci - dessous les box-plots des RMSE qu'on a obtenu après cross - validation sur nos modèles définitifs (c'est à dire ceux pour lesquels on a déjà "tuné" et obtenu les bons paramètres)

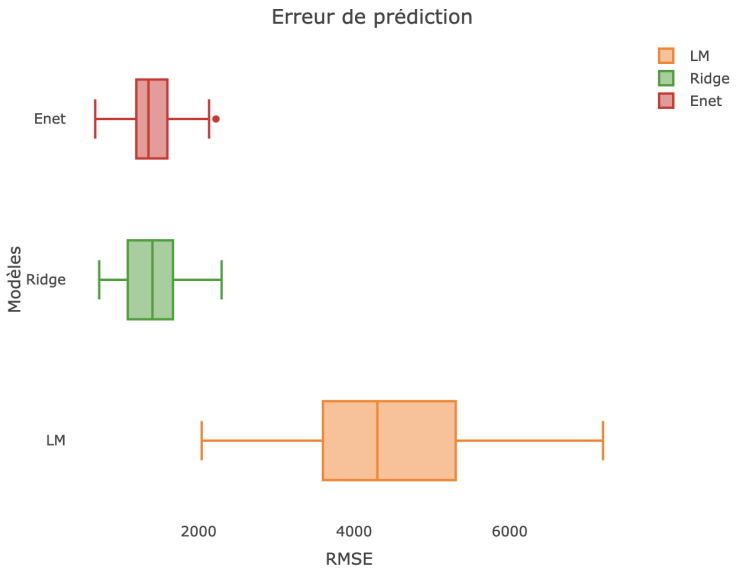

FIGURE 27 – RMSE enet vs ridge vs lm

Les modèles de régression pénalisée (Ridge et Enet) semble nettement réduire le RMSE du modèle linéaire.

3.3 Modèles d'apprentissage statistique

Les modèles d'apprentissage statistique sont considérés comme des modèles non paramétriques, ce qui signifie qu'ils ne nécessitent pas de vérification des hypothèses sur la distribution de la loi pour être utilisés, ce qui représente un avantage. Cependant, une taille d'échantillon substantielle est requise pour garantir leur robustesse.

3.3.1 CART

L'algorithme CART, abréviation pour "Classification And Regression Trees", est utilisé pour créer un arbre de décision en classant un ensemble d'enregistrements jusqu'à ce que des groupes homogènes soient obtenus. Cette méthode est appropriée à la fois pour la régression et la classification.
On observe dans un premier temps l'évolution de l'erreur de prédiction en fonction de la complexité de l'arbre à l'aide de la fonction R *plotcp*.

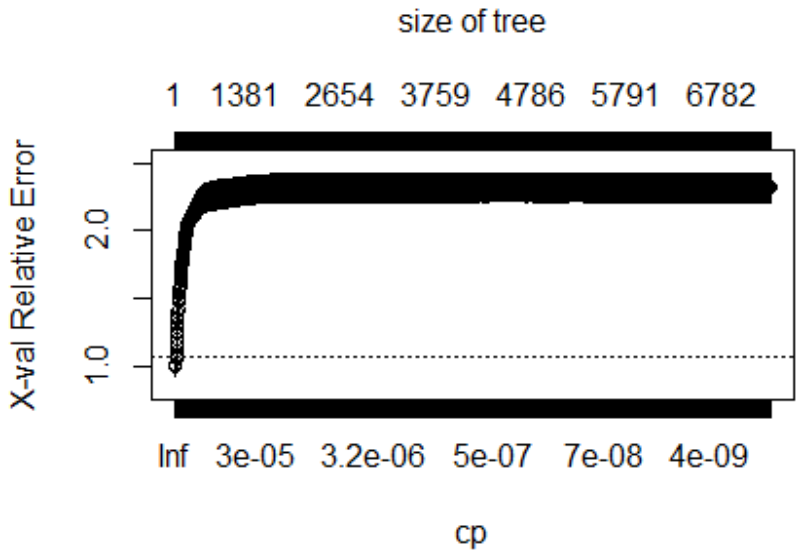

FIGURE 28 – Erreur de prédiction en fonction de la complexité du CART

Habituellement, on s'attend à observer une courbe décroissante, mais ce n'est pas le cas dans cette situation, et cela s'explique par la nature spécifique de notre jeu de données. En effet, à mesure que nous ajoutons des variables explicatives, le modèle perd en précision, un phénomène qui survient fréquemment dans les modèles de régression en apprentissage statistique. Il peut sembler paradoxal que l'erreur augmente à mesure que de nouvelles variables sont incluses, cependant, les premiers "splits" permettent d'obtenir de meilleurs résultats. En faisant un "Pruning", nous pouvons conclure que l'arbre résultant, bien qu'ayant moins de nœuds que l'arbre maximal, fournit des prédictions de meilleure qualité.

Le modèle CART est réputé pour sa sensibilité aux variations des données. Pour remédier à ce problème, nous examinerons le modèle Random Forest qui, grâce à la technique du bagging, contribue à renforcer cette stabilité.

3.3.2 Random Forest

En machine learning, les random forests, ou forêts aléatoires d'arbres de décision, représentent une technique d'apprentissage ensembliste. Cette méthode combine les principes des sous-espaces aléatoires et du bootstrap aggregating. Concrètement, l'algorithme des random forests réalise un processus d'apprentissage sur plusieurs arbres de décision, chacun entraîné sur des sous-ensembles de données légèrement différents.

A présent, on observe l'erreur de prédiction en fonction du nombre de variables dans chaque arbre.

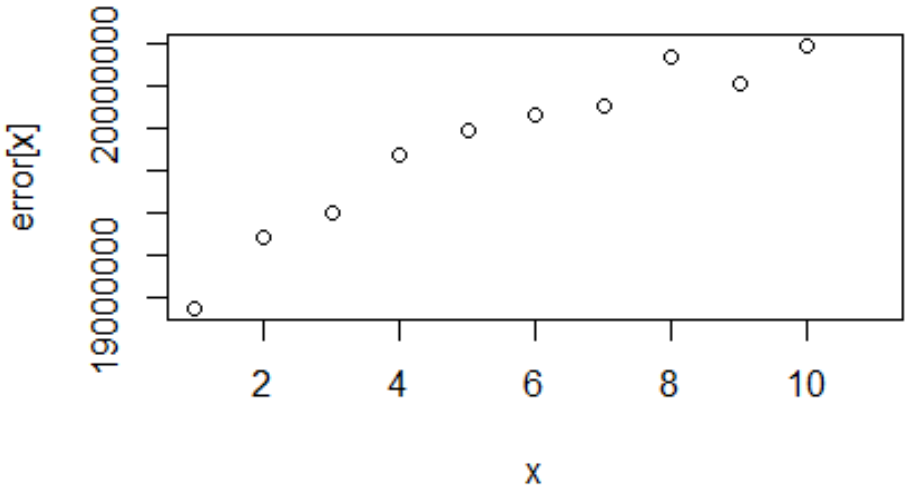

FIGURE 29 – Erreur de prédiction en fonction du nombre de variables dans chaque arbre

Chaque arbre du random forest sera associé à une seule variable explicative. Cette situation est représentée dans le graphique ci-dessus, où la valeur minimale de x=1 correspond à la minimisation de l'erreur de prédiction.
Ensuite, nous effectuons des tests avec un nombre de variables explicatives allant de 1 à 3 et un nombre de nœuds allant de 5000 à 10000.

Le graphique ci-dessous confirme que chaque arbre du random forest sera expliqué par une seule covariable.

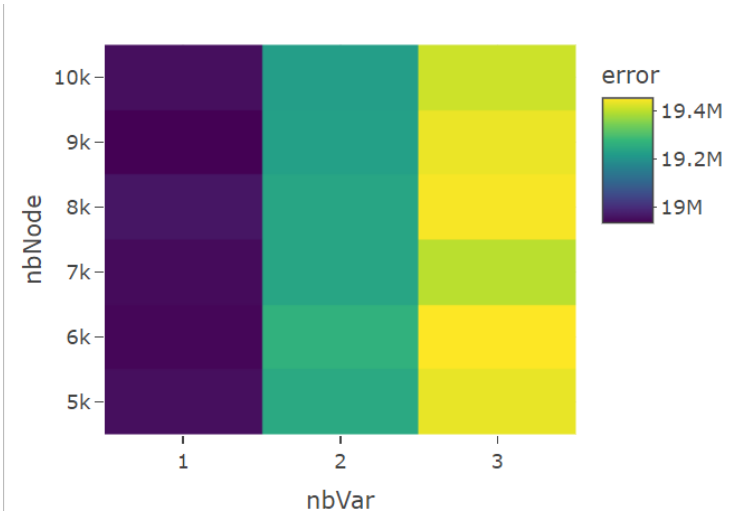

FIGURE 30 – Erreur de prédiction en fonction du nombre de variables dans chaque arbre et de noeuds

3.3.3 XgBoost

L'algorithme XgBoost est un modèle qui présente des similitudes avec l'algorithme Random Forest. La principale différence réside dans le fait que le XGBoost se concentre sur la correction séquentielle des erreurs en mettant davantage l'accent sur les observations mal prédites, alors que Random Forest ne corrige pas explicitement les erreurs dans un processus séquentiel.

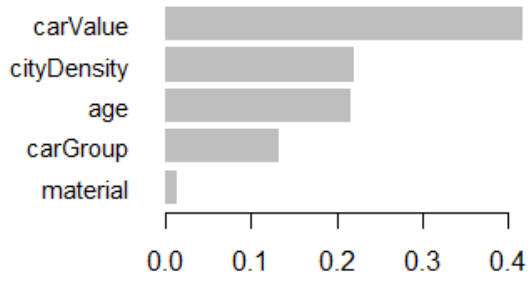

FIGURE 31 – Erreur de prédiction en fonction du nombre de variable dans chaque arbre et du nombre de noeuds Xgboost

D'après ce graphique, nous pouvons conclure les 4 variables les plus importantes sont $carValue, cityDensity, age$ et $carGroup$.

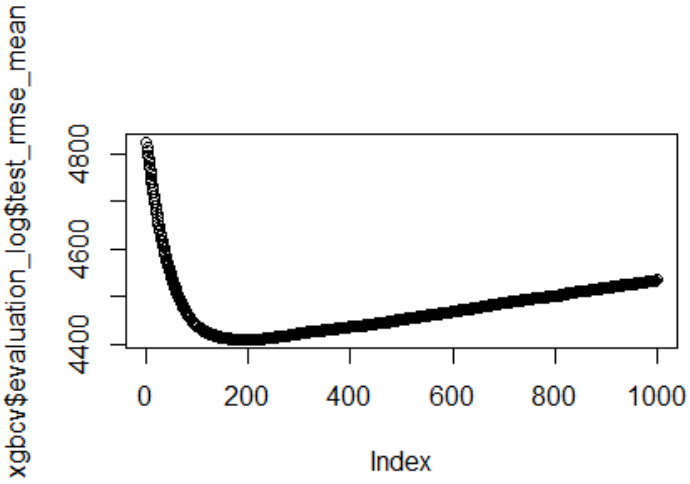

FIGURE 32 – tuning sur le nombre d'arbres

On remarque que la courbe est décroissante jusqu'à la valeur 200. Au delà de 200 arbres, l'erreur augmente : Il y a donc du sur-apprentissage. Le nombre d'arbres optimal est donc de 200.

3.4 Comparaison des modèles

Remarque : Nous procédons de la même manière qu'avec le Ridge et l'Elastic Net pour obtenir pour les autres modèles d'apprentissage statistique plusieurs RMSE par cross-validation afin de comparer les box-plots de ces derniers. On obtient ainsi pour tous les modèles :

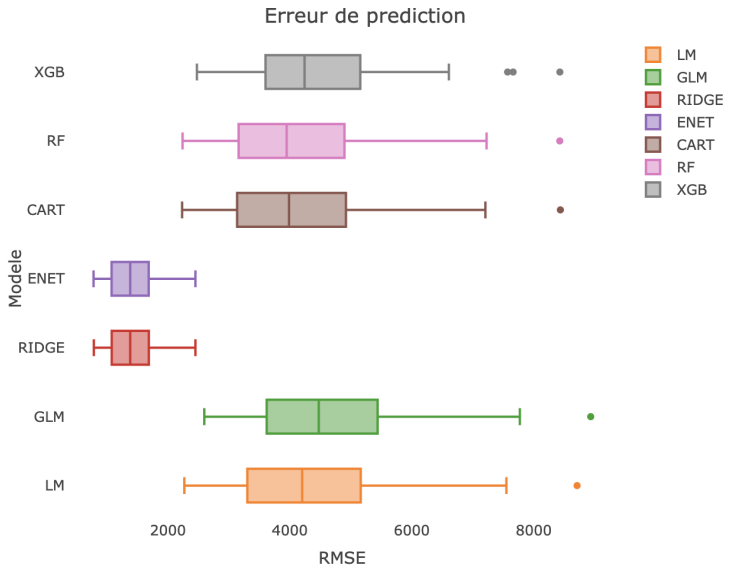

FIGURE 33 – RMSE ALL

On retient selon le critère du RMSE comme modèle de prédiction du montant moyen de sinistre le modèle de régression pénalisée Ridge. Il s'agit d'un modèle de régression qui vient en amélioration du modèle linéaire simple et il est moins complexe que les autres modèles d'apprentissage statistique.

Remarque : La limite principale de notre modèle **Ridge** qui est une amélioration de la régression linéaire est qu'elle fait l'hypothèse d'une loi normale sur la distribution de **claimValue** ; il va ainsi bien capter le montant moyen de sinistre mais aura du mal sur les valeurs extrêmes en queue de distribution. Une amélioration serait d'avoir une base des sinistres extrêmes et faire un tarif particulier pour les assurés de ce profile.

4 CLASSIFICATION

Cette partie est consacrée au calcul de la probabilité d'enregistrer un sinistre sur une police pendant une année.
On construit à partir de notre base train initiale une base de classification qui contient la variable **sinistre** qui vaut :

— 1 si la valeur de claimValue est superieure à 0 et
— 0 si la valeur de claimValue est inférieure à 0 (cf **Feature Engeneering - Part 2**.

Dans la base classification, on supprime les variables **subregion et claimValue** car elles sont sources de multicolinéarité.

4.1 Regression logistique

Le modèle s'écrit :

$$\log \frac{p}{1-p} = b_0 + b_1 * X_1 + ... + b_n * X_n$$

Où :

— $p = P(Y = 0|X_1, X_2, ..., X_n)$ Y est la variable à expliquer
— $X_1, X_2, ..., X_n$ sont les variables explicatives du modèle
— $b_0, b_1, b_2, ..., b_n$ sont les coefficients de régression

En terme de qualité du modèle, nous avons obtenu :

— Une déviance résiduelle moyenne de 23764.58
— Un AIC moyen de 23856.58
— Une précision moyenne de 76.80%

Dans la suite, nous allons essayer d'améliorer la performance du modèle en faisant une régression pénalisée.

4.2 Régression Stepwise

4.2.1 Présentation

C'est la sélection des variables explicatives optimales permettant d'avoir un modèle logistique plus performant selon le critère d'AIC.
Après son application à notre base train, nous avons remarqué que **carValue** est la seule variable explicative qui n'a pas été retenue dans le modèle final.

4.2.2 Comparaison avec la régression logistique

	Logistique Reg	Stepwise Reg
Residual Deviance moyenne	23764.58	23764.75
AIC moyenn	23856.58	23854.75
Précision moyenne	76.79%	76.79%

On a eu une précision moyenne de **76.80%** qui est finalement la même que celle du modèle logistique initial.

4.3 Régression pénalisée

4.3.1 Les modèles

Nous avons testé l'ensemble des trois modèles de régressions Ridge, LASSO et Elastic Net . Nous avons obtenu les performances suivantes après le tuning des paramètres :

	Ridge Reg	LASSO Reg	Elastic Net
Précision moyenne	76.60%	76.64%	76.70%
AUC moyen des modèles	72.24%	72.27%	72.30%

Nous avons retenu l'Elastic Net selon le critère de la précision et de l'AUC. Par la suite, nous allons comparer ce modèle à la régression logistique.

4.3.2 Comparaison avec la régression logistique

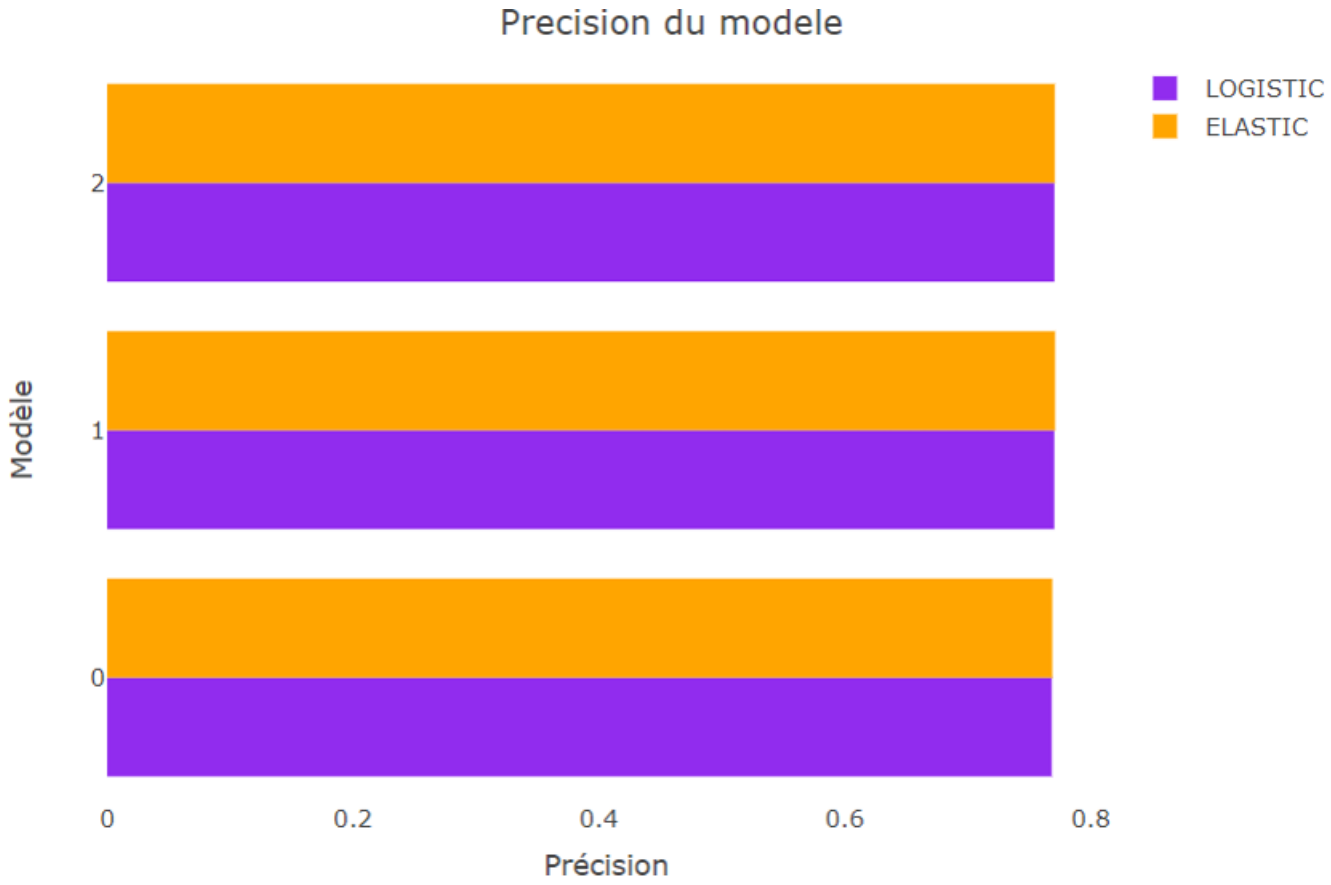

FIGURE 34 – Comparaison de Elastic Net et Logistic Reg

Suite à la cross validation on a :
— **Précision moyenne :** 76.99% pour l'Elastic Net et 77.04% pour la régression logistique
— **AUC :** 72.30% pour l'Elastic Net et 72.10% pour la régression logistique

4.4 Classificateur Naïf de Bayes

4.4.1 Principe

Ce classificateur est un modèle probabiliste basé sur le théorème de Bayes, qui permet de calculer la probabilité qu'une observation appartienne à une classe donnée en fonction des caractéristiques de cette observation. Il est qualifié de "naïf" à cause de l'hypothèse d'**indépendance** entre les caractéristiques, qui simplifie le calcul des probabilités conditionnelles.

Remarque :
— Dans notre cas, les classes sont définies par les modalités 1 et 0 de la variable sinistre ; c'est à dire 1= avoir eu un sinistre et 0 = ne pas avoir eu de sinistre

— les caractéristiques sont les différentes valeurs prises par les variables explicatives pour un individu donné.

Le principe de fonctionnement du classificateur de BAYES est le suivant :

a. On apprend sur un ensemble de données d'apprentissage, qui contient des observations dont on connaît la classe.

b. Pour chaque classe, on calcule la probabilité que chaque caractéristique soit présente à travers la formule :

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

— **c** est la classe à laquelle on veut assigner l'observation
— **x** est l'observation
— **P(c|x)** est la probabilité que l'observation appartienne à la classe c, étant donnée les caractéristiques x
— **P(x|c)** est la probabilité que les caractéristiques x soient présentes, sachant que l'observation appartient à la classe c
— **P(c)** est la probabilité que l'observation appartienne à la classe c
— **P(x)** est la probabilité que les caractéristiques x soient présentes

c. Pour une nouvelle observation, on calcule la probabilité qu'elle appartienne à chaque classe.

d. On attribue l'observation à la classe pour laquelle la probabilité d'appartenance est la plus élevée.

4.4.2 Application

Nous avons obtenu la matrice de confusion suivante :

		Reference	
		0	1
Predicted	0	4132	1045
	1	395	428

En terme de précision, on obtient $4560/6000 \times 100 = \mathbf{76\%}$.

4.5 L'arbre de décision CART

4.5.1 L'arbre maximal

Nous avons d'abord présenté l'arbre maximal avec toutes les variables de notre base. Ensuite, nous avons réalisé le graphe qui présente l'évolution de l'erreur de validation croisée en fonction du nombre du paramètre de complexité cp et de la taille de l'arbre .

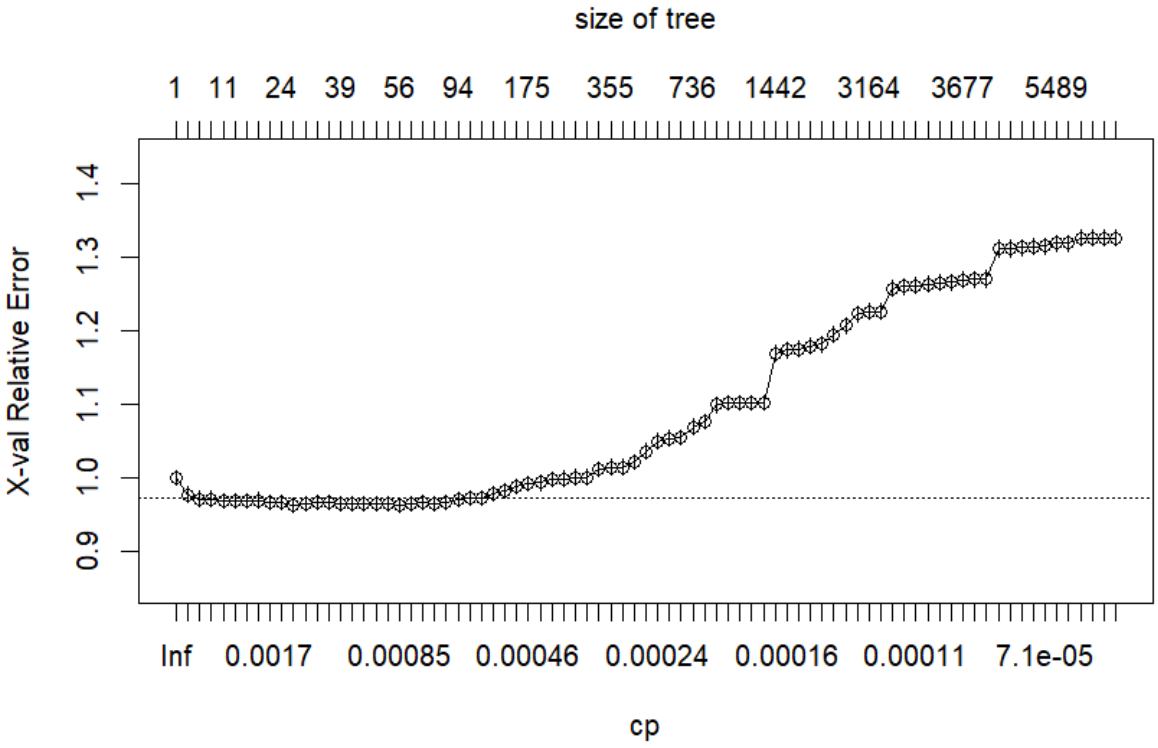

FIGURE 35 – La précision en fonction du paramètre de complexité

Le graphe ci-dessus nous a permis d'avoir cp = 0.000814664 correspondant à la plus petite erreur de validation croisée.

4.5.2 Recherche des bons hyperparamètres/ Élagage

Nous avons tuné les autres hyperparamètres en faisant varier minsplit de 1 à 15 et minbucket de 1 à 10 et calculer à chaque étape l'erreur de classification. Nous avons obtenu au final les meilleurs hyperparamètres suivants :

- cp = 0.000814664
- minsplit = 2
- minbucket = 4

Par la suite, nous avons élagué notre modèle final avec les paramètres ci-dessus

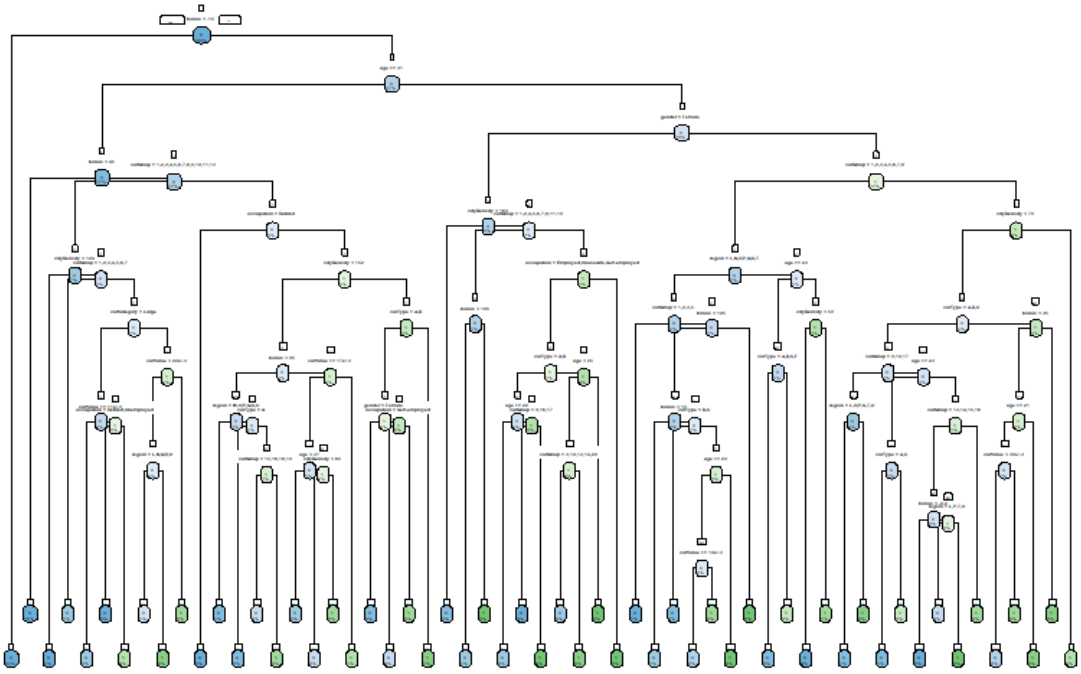

FIGURE 36 – Arbre élagué

Ce nouvel arbre a une précision moyenne de 76.51% qui est une amélioration par rapport à l'arbre maximal.

4.6 L'approche du Random Forest

4.6.1 Présentation

Il présente l'avantage de réduire l'overfitting (surapprentissage) et d'améliorer la généralisation par rapport à l'arbre CART. En utilisant un échantillonnage aléatoire des données (bagging) et des caractéristiques (feature sampling), le modèle devient plus robuste et moins sensible aux variations spécifiques aux données d'entraînement.

4.6.2 Application

Dans un premier temps, on affiche l'erreur out of bagging (oob) d'un modèle avec les paramètres par défaut en fonction du paramètre ntree (nombre d'arbres).

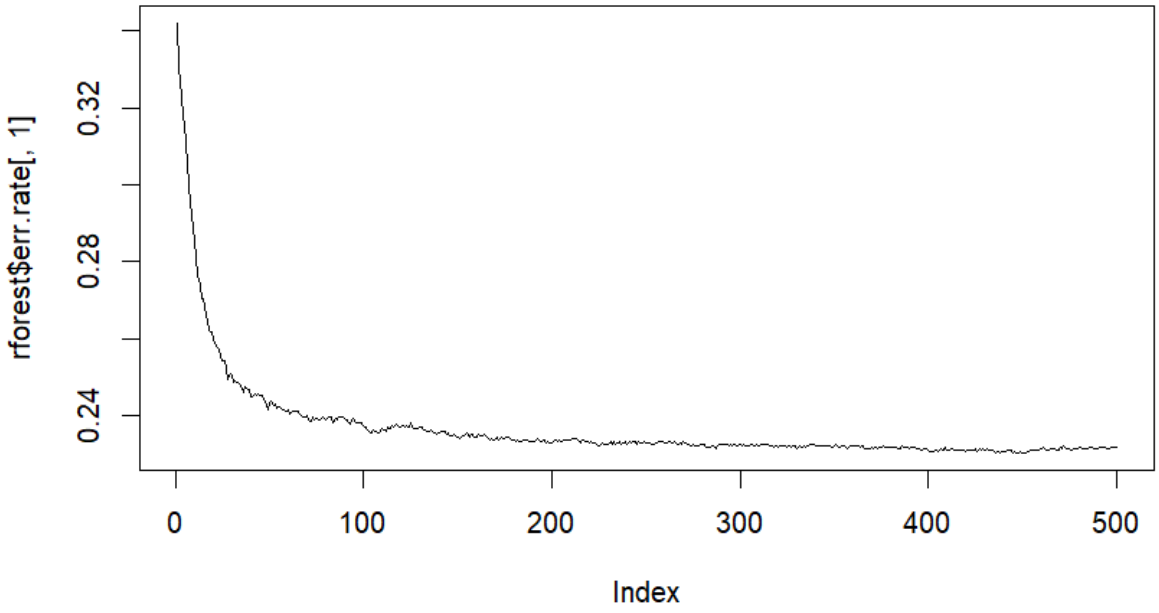

FIGURE 37 – erreur oob en fonction de ntree

Ce qui nous permet d'avoir ntree = 500 pour la suite.

Nous avons ensuite "tuner" les autres hyperparamètres en faisant varier le nombre de variables sélectionnées à chaque division d'un arbre de 1 à 3, le nombre maximum de noeud terminaux (feuilles) de 5000 à 10000 avec le nombre d'arbre étant fixé à 500. À chaque itération, nous évaluons l'erreur de classification afin de sélectionner les paramètres correspondants à la plus petite erreur de classification.

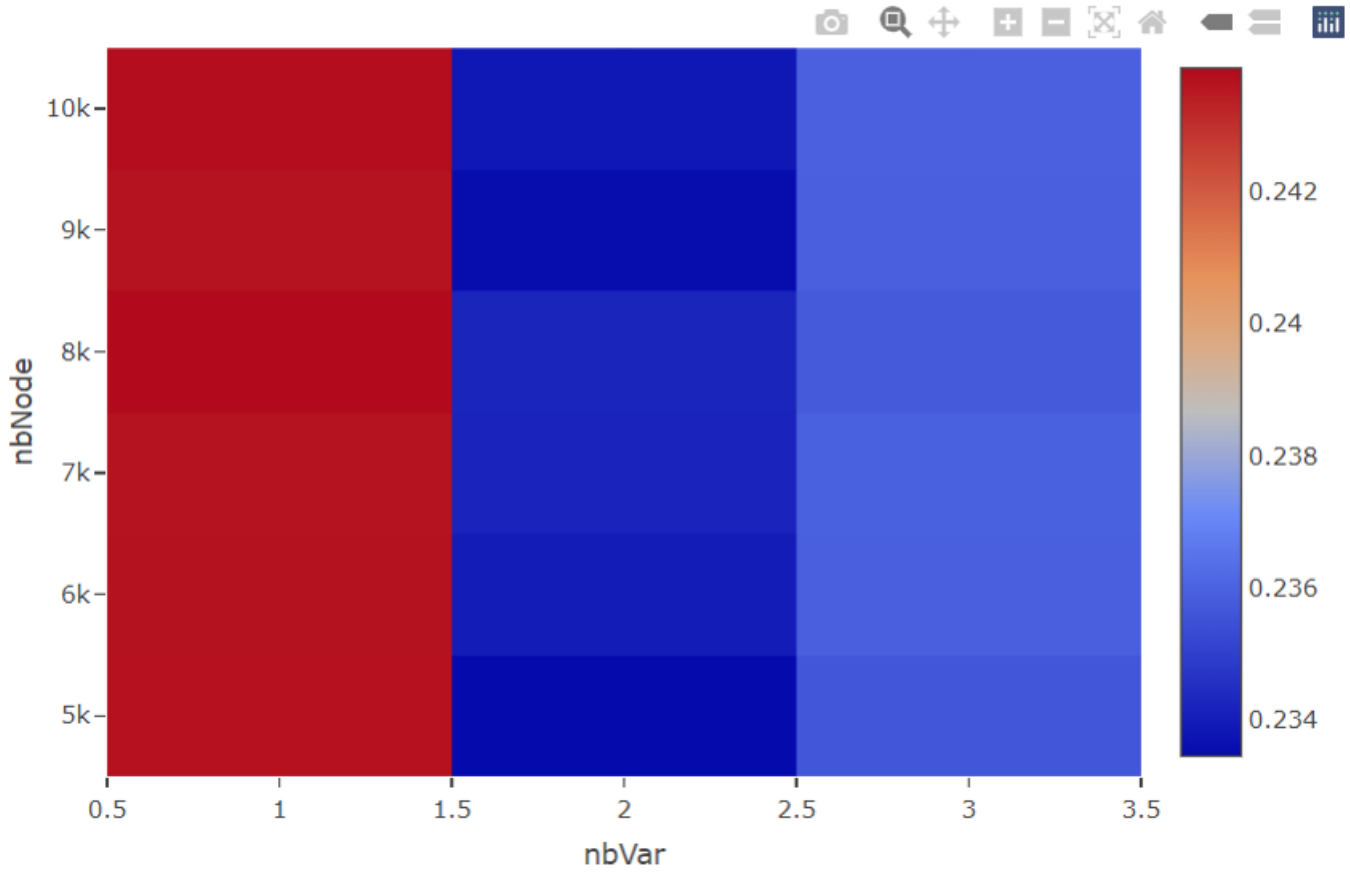

FIGURE 38 – Taux d'erreur en fonction du nombre de noeuds et du nombre de variable

Enfin, on obtient les valeurs suivantes pour nos hyperparamètres :
— mtry = 2 : le nombre de variables
— ntree = 500 : le nombre d'arbres
— maxnodes = 5000 : le nombre maximum de feuilles

Pour ces paramètres ainsi obtenus, nous avons eu une précision moyenne de **77.11%** et un AUC moyen de **72.93%** par cross-validation.

4.7 La classification avec XGBoost

Principe

C'est un modèle amélioré de l'algorithme d'amplification de gradient (Gradient Boost). Il est utilisé pour réduire le nombre d'erreurs dans l'analyse prédictive des données.

Mise en place

Nous avons procédé à une cross validation en divisant nos données en 5 classes. Nous avons "tuné" l'hyperparamètre nrounds en regardant les différentes valeurs de ce dernier en fonction des taux d'erreur (oob)

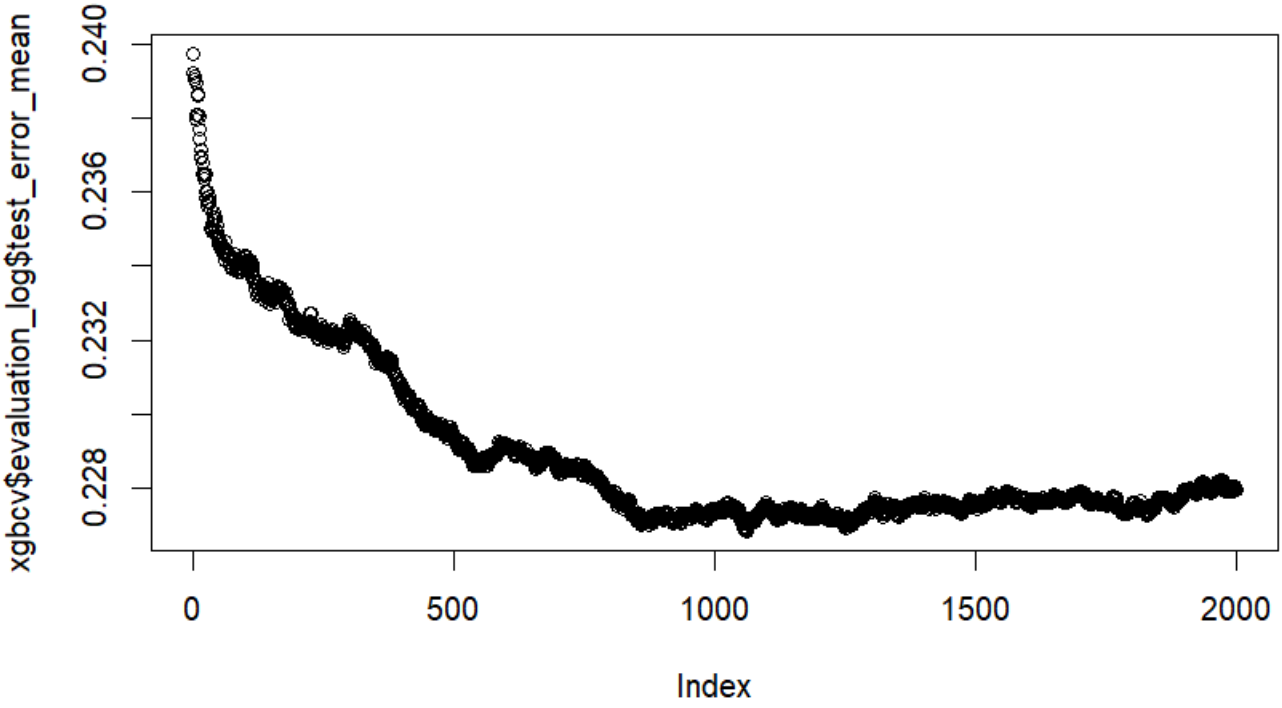

FIGURE 39 – nrounds en fonction des erreurs moyennes

Ce qui nous a permis d'avoir **nrounds = 1062**.

Tuning du paramètre eta

Nous avons fait varier eta en fonction du nombre d'itération nrounds. On choisira le "eta" le plus petit possible.

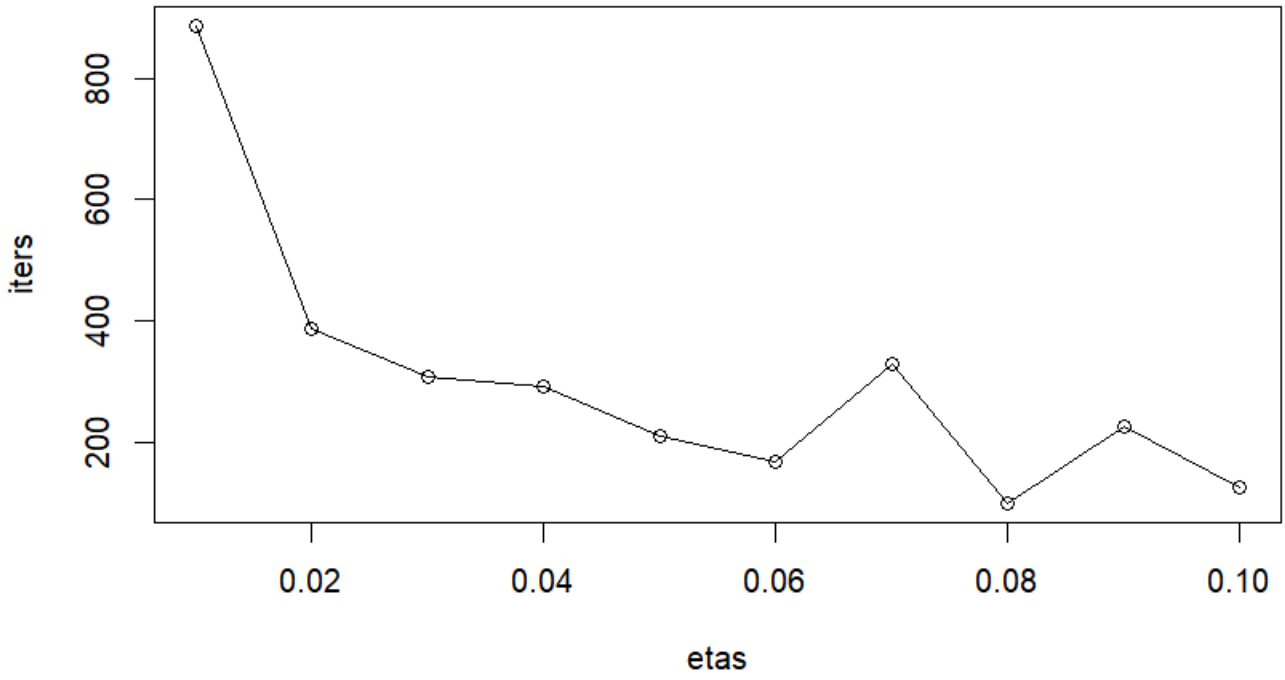

FIGURE 40 – eta en fonction de nrounds

les meilleurs hyperparamètres sont donc eta = 0.01 et nrounds = 887. Avec ces nouveaux paramètres, nous avons obtenu une précision de **77.32%** et un AUC de **73.27%**pour le XGBoost.

4.8 Le modèle final pour la classification

4.8.1 Comparaison des performances

On trace les courbes roc (Receiver Operating Characteristic) des modèles. On rappelle que la courbe roc permet de décrire la performance d'un modèle à travers deux indicateurs : sensitivity ou sensibilité qui est le taux d'individus positifs correctement prédits par le modèle et la specificity qui est le taux d'individus négatifs correctement prédits par le modèle. La courbe roc est la représentation de la sensibilité en fonction de 1-la spécificité

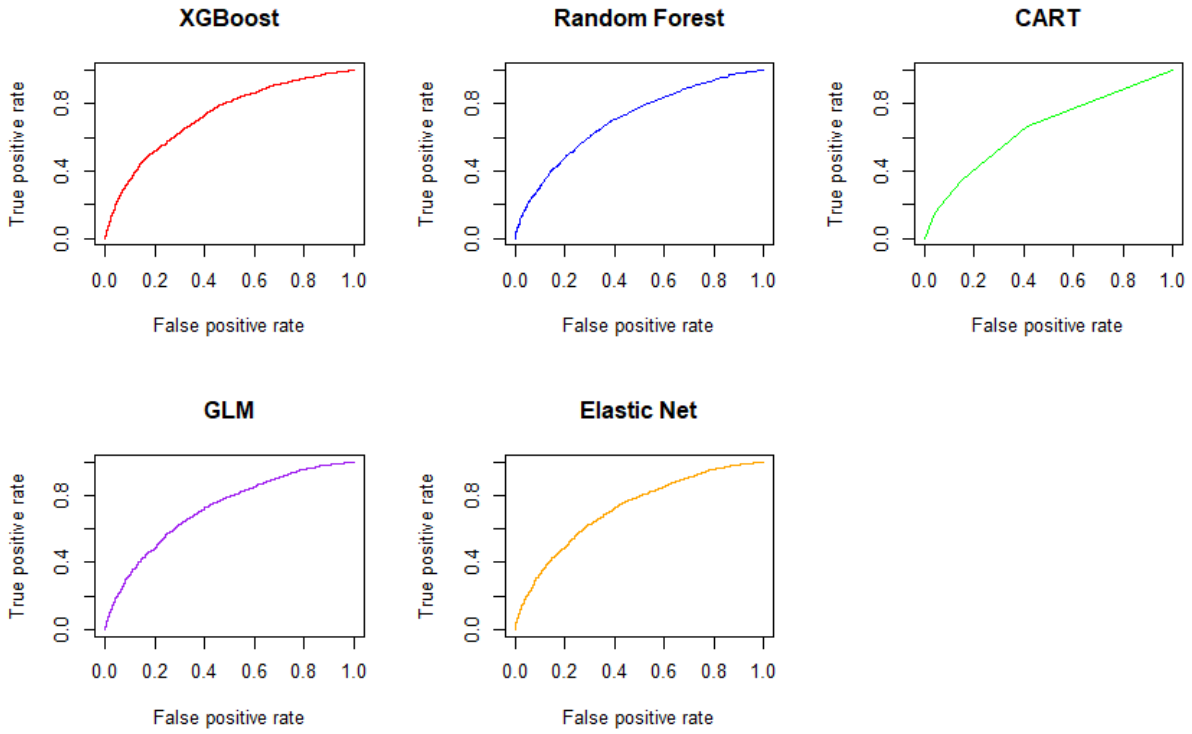

FIGURE 41 – Courbes roc des différents modèles

Nous allons dresser un récapitulatif des indicateurs de performances des différents modèles par cross-validation. On obtient :

	XGBoost	Random Forest	CART	Logistic Reg	Elastic Net
Précision	77.32%	77.11%	76.51%	76.80%	76.78%
AUC des modèles	73.27%	72.93%	67.41%	72.31%	72.30%

Globalement, tous les modèles ont une précision autour de 77% avec XGBoost qui est légèrement meilleur que les autres en terme de précision et de AUC. Il s'en suit que nous allons retenir **XGBoost** comme modèle de prédiction selon les critères de sélection.

4.8.2 Analyse du modèle retenu pour la classification

Cas d'overfitting ou surapprentissage ?

Notre but est de tester s'il y a un phénomène de surapprentissage avec notre modèle.
On procède comme suit :

- On entraîne un modèle à prédire la base d'entraînement elle-même et une base test
- On calcule l'AUC dans les deux cas.
- Ici on trouve respectivement 73.27% pour la base test et 78.43% pour la base de training : l'écart entre les deux est 5.16% $<$ 10% ce qui nous permet de conclure (de manière large) qu'il n'y a pas de surapprentissage.

Variables importance

Le graphe ci-dessous nous permet de classer les variables explicatives dans notre XGBoost selon leur pouvoir explicatif sur la probabilité d'avoir un sinistre ou non.

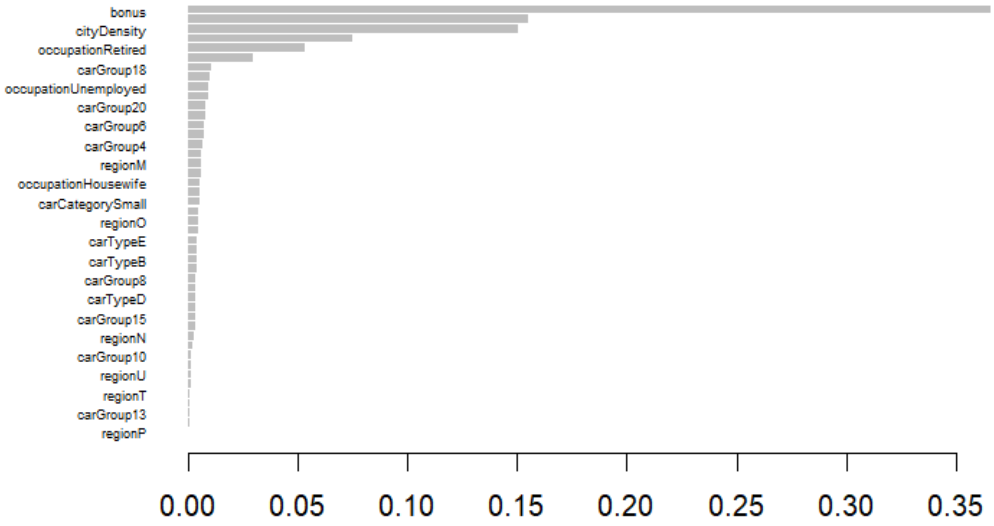

FIGURE 42 – Importance des variables

On constate que la variable **Bonus** va fortement influencer la probabilité d'avoir un sinistre ou non ; et c'est bien cela l'idée d'un bonus - malus. Il s'agit d'influencer positivement les conducteurs afin de réduire la fréquence de sinistre ; ici il est difficile en regardant ce graphique de dire dans quel sens le bonus influe sur la probabilité d'avoir un sinistre mais on s'attend à ce qu'il fasse baisser cette probabilité.

5 CONCLUSION

Comme on a pu le voir, il existe plusieurs modèles de prédiction qui présentent chacun des avantages et des limites. Pour les besoins de notre modélisation et en fonction des critères de sélection qui nous ont été fixés - à savoir le **RMSE** pour la partie régression et le **AUC** pour la partie classification - on a retenu les modèles suivant :

— le **Ridge** pour la régression
— le **XGBoost** pour la classification

La qualité de ces 2 modèles se mesure en particulier sur le calcul de la prime pure dans la base test. Notre objectif est d'avoir un S/P réaliste, c'est à dire un peu en dessous de 100% .
Avec d'autres critères de sélection ou avec des objectifs différents de la prédiction, on aurait pu retenir d'autres modèles. On se rend ainsi compte de l'importance de garder en tête l'objectif du projet à tout

instant de sa mise en oeuvre. De plus, un projet comme celui - ci n'est pas qu'une suite ordonnée d'étapes ; les étapes peuvent renvoyer l'une à l'autre dans tous les sens, en particulier pour notre projet nous avons sans cesse apporter de nouveaux éléments à notre base de données (feature engineering) pour tester des nouvelles pistes d'analyses que nous ont inspirés les résultats des modèles ou même l'exploration simple des données. Enfin, il faut retenir qu'aussi bon soit - il, un modèle reste une simplification de la réalité ; la connaissance métier, l'expérience et la considération humaine permettent d'améliorer l'exploitation des résultats que nous donnent les modèles.