Projet OMA

Hamza Rami
Ouassim Hammouch

March 2020

1 Introduction

2 Le jeu de données

2.1 Présentation

2.2 Premier traitement

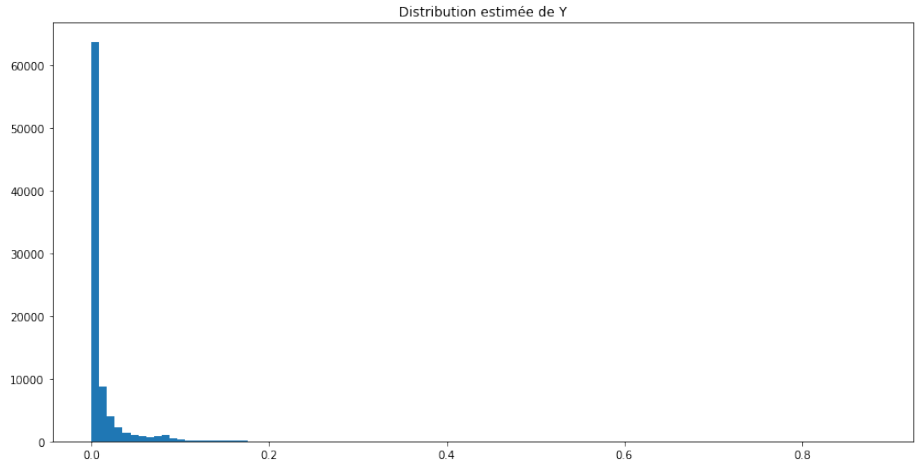
2.3 Approximation de la fréquence de sinistres

2.3.1 Méthode

La variable d'intérêt dans notre cas est la fréquence de sinistre annuelle. Elle peut être calculée à partir de données disponible dans le jeu de données initial :

$$\text{Frequence de sinitre} = \frac{\text{nombre de sinistre}}{\text{durée du contrat}}$$

Celle-ci sera notée et appelée Y dans la suite. Une rapide observation de cette variable montre une distribution avec plus de 95% de zéros. La décroissance semble ensuite être exponentielle.



En pratique, la fréquence sinistre moyenne est petite par rapport à 1. Chaque y_i représente une estimation de cette moyenne, effectuée sur une durée de l'ordre de l'année, parfois moins. Ne disposant que de techniquement un seul point par estimation, la moyenne empirique ne peut être considérée comme un bon estimateur. Une majorité y_i sont des sur-estimations ou sous-estimations grossières de la fréquence de sinistre réelle.

Afin de palier à ce problème, la variable Y a été corrigée en se basant sur l'hypothèse que des profils similaires devrait avoir des fréquences de sinistres moyens relativement proches. Cela impose des contraintes sur l'ajustement :

- La fréquence d'un profile doit être "tirée" vers celle de ses voisins
- L'ajustement apporté par un voisin doit être d'autant plus important que le voisin est proche

En notant x_i les coordonnées d'un profil dans l'espace des variables explicatives d'un profil i, et y_i sa fréquence de sinistre observée, l'ajustement suivant a été retenu :

$$\tilde{y_i} = max(y_i + \sum_j d_\lambda(x_i, x_j)(y_j - y_i), 0)$$

où d est une distance paramétré par λ. Cela permet de définir une nouvelle variable $\tilde{Y}$.
Plusieurs distances ont été considérées. La distance doit permettre de localement homogénéiser les valeurs de Y, tout en gardant une fidélité aux valeurs initiales. Nous avons retenu les noyaux gaussien et de Laplace, car ils sont invariant par rotation, le degré d'ajustement peut être contrôlé par une unique variable λ, et une décroissance exponentielle de l'impact d'un voisin en son éloignement nous a paru plus réaliste que linéaire ou quadratique.
Finalement, les distances considérées sont :

- $d_\lambda(x_i, x_j) = e^{-||x_i - x_j||/\lambda}$
- $d_\lambda(x_i, x_j) = e^{-||x_i - x_j||^2/\lambda}$

La correction est donc une somme pondérée des différences des fréquences du profil considérés et de ses voisins.

2.3.2 Optimisation et résultats

L'impact de cet ajustement est contrôlé de deux manières :

- La proportion de 0 dans les observations
- L'écart quadratique moyen relatif : $\frac{||Y - \tilde{Y}||}{||Y||}$

Afin d'éviter un calcul trop long, seul les 1000 voisins les plus proches situés ont été considérés. Ces voisins peuvent être rapidement trouvé avec l'algorithme **K-D Tree**. La distribution de $\tilde{Y}$ présente désormais moins de 0, mais beaucoup

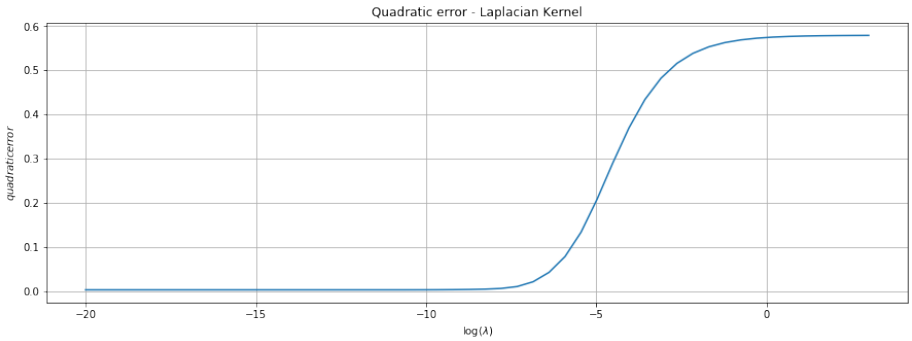
de valeurs très faible, qui peuvent raisonnablement être considérées comme des sous estimation de la fréquence moyenne réelle. Pour la suite et dans le décompte des fréquences nulles, nous considérerons toute valeur inférieure à 10^{-6} nulle.

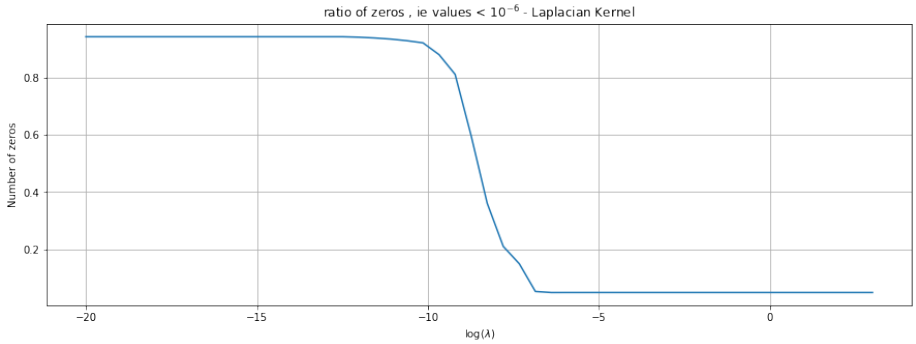
Il déjà possible prévoir certains résultats sur l'allure des deux métriques de contrôle.

- L'erreur quadratique sera globalement croissante en λ. En effet, plus ce dernier est élevé, plus les distances sont élevées, et plus la correction est importante.
- Lorsque λ tend vers 0, les poids tendent vers 0, et l'écart quadratique tend vers 0.
- si λ tend vers l'infini, les poids tendent à devenir identiques et égaux à 1, et les fréquences moyennent se déplacent vers la moyenne des fréquences de tous les profiles, et peuvent la dépasser : $\tilde{y}_i = max(y_i + n(\bar{Y} - y_i), 0)$. Avec l'approximation, grossière mais valable pour 95% des points, $Y \sim 0$, on obtient $\tilde{Y} = n\bar{Y}$, ce qui correspond à une erreur quadratique asymptotique proche de 0.5.
- A l'inverse, la courbe du nombre de zéros devrait quand à elle être décroissante.
- Pour λ proche de 0, le nombre de zéros tend vers l'initial, représentant près de 95% du total.
- Pour des grandes valeurs de λ, si $y_i \geq \frac{n-1}{n}\bar{Y}$ alors $\tilde{y}_i = 0$ et inversement. Pour $n = 1000$, le premier cas concerne près de 5% des points, d'où une proportion asymptotique de 0.05.

Pour chaque métrique, un intervalle a été choisi afin d'observer l'impact de l'ajustement pour différentes valeurs du paramètre.

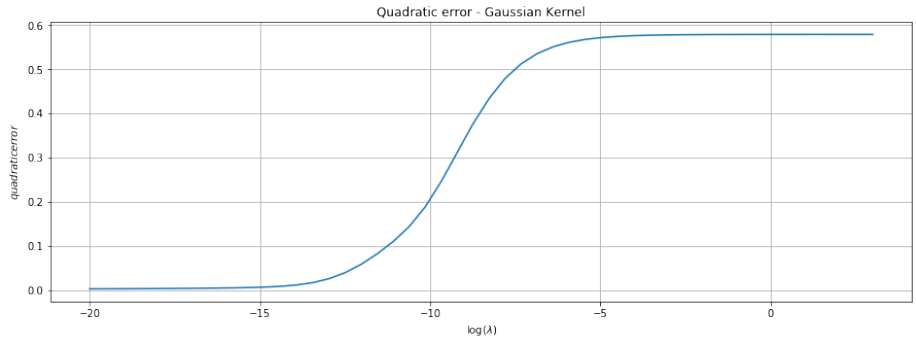
Noyau Laplacien Les figures suivantes présentent les résultats pour le noyau Laplacien.

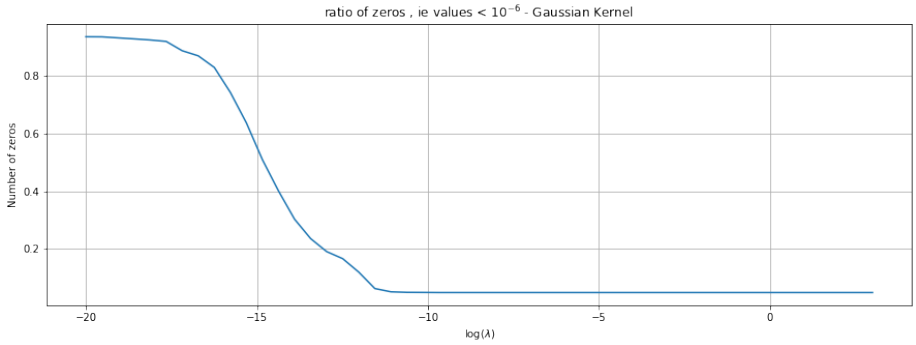




On constate que les courbes ont les allures attendues.

Noyau Gaussien Les figures suivantes présentent les résultats pour le noyau Laplacien.





On constate ici aussi que les courbes ont les allures attendues.

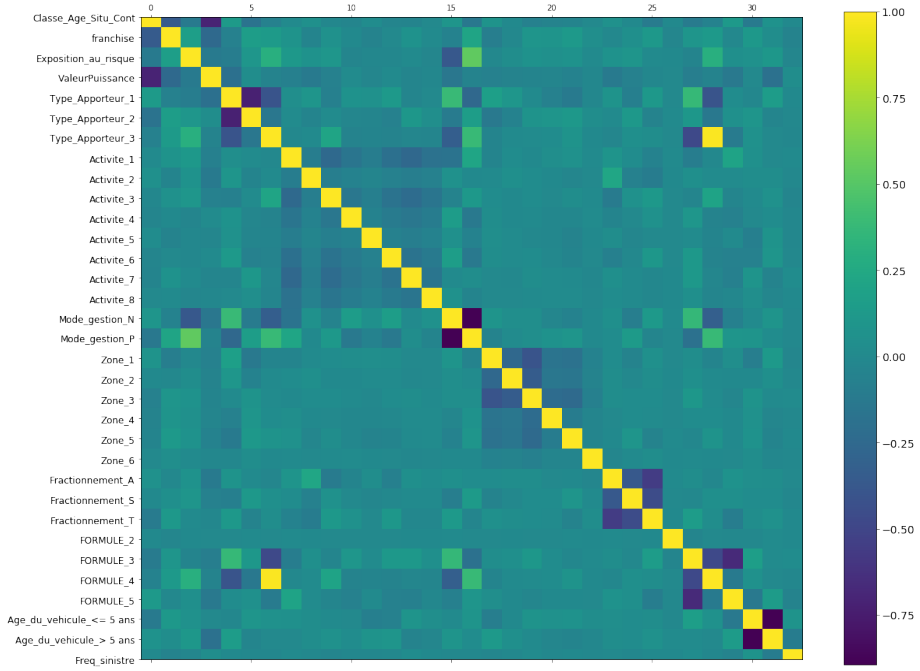
Finalement, c'est la distance de Laplace qui a été retenue, avec le paramètre $\log \lambda = -8$ car elle permet d'obtenir un meilleur compromis sur les deux métriques.. Cela permettait d'avoir 6.3% de 0, pour une erreur quadratique relative de 1.7%.

Pour la suite, nous ne considérerons que la variable ajustée, qui sera désormais appelée et notée Y par souci de simplification.

2.4 Réduction de dimension

2.4.1 Analyse univariée

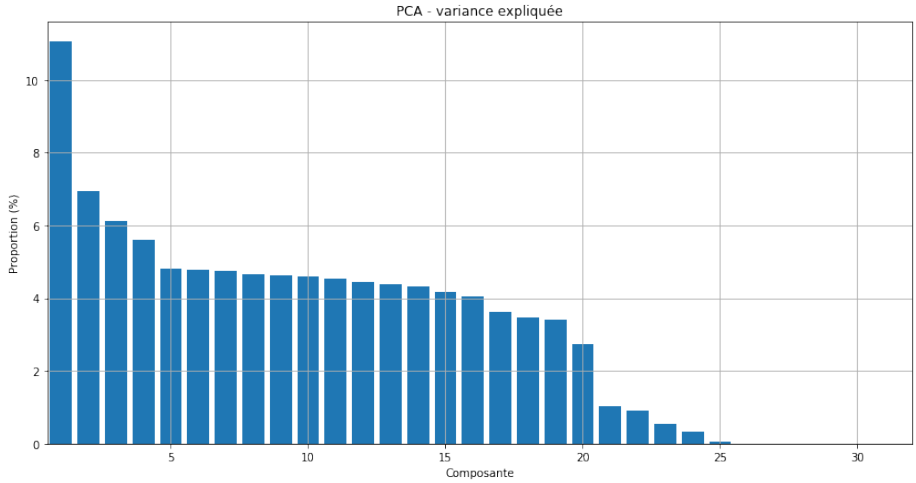
Afin de déceler d'éventuelles dépendances au sein des variables explicatives, leurs corrélation deux à deux ont été calculées, et sont affichées dans la figure suivante :



On peut tout d'abord observer une corrélation négative et généralement significative entre les variables dummies correspondant à une même variable catégorielle initiale. Cela est normal car la somme de ces variables est toujours égale à 1. Hormis certaines exceptions, les corrélations entre les variables sont très faibles. On ne peut donc pas ignorer de variables pour l'instant. Enfin, on peut remarquer l'absence de corrélation entre Y et les variables explicatives.

2.4.2 ACP

L'analyse en composantes principales est une méthode qui consiste à trouver les combinaisons des variables qui maximisent la variance. C'est un technique très utilisée pour réduire la dimension du jeu de données. La figure suivante présente la variance expliquée de chaque composante obtenue lors de la décomposition.



On constate que les composantes principales ont une variance expliquée faible, et qu'un grand nombre de variables ont un taux très proche (5%).

2.4.3 Kernel ACP

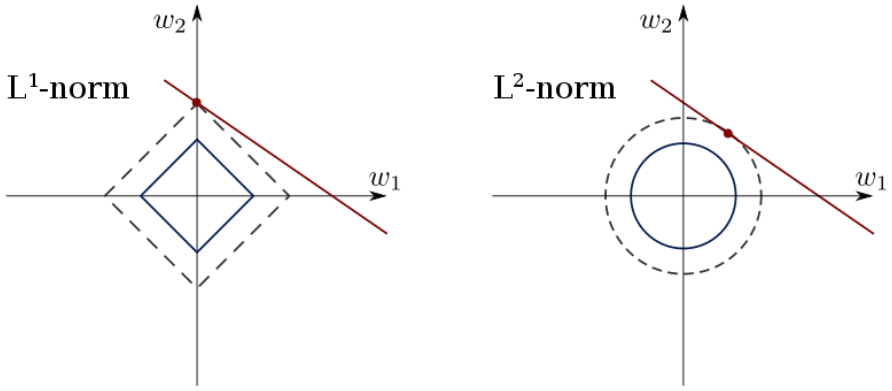
Une ACP a également été appliquée dans un espace transformée afin de comparer les résultats. Les noyeaux testés sont : polynomial, rbf, simgoid et cosine. Malheureusement, aucune de nos machine n'avait assez de RAM (16 n'est pas suffisant) pour manipuler la matrice du noyau, qui est dans tous les cas une des première étapes de l'algorithme. Des tests effectuées sur des sous ensembles de 5000 profils sont très similaires à ceux obtenus avec uen ACP classique.

2.4.4 Lasso regression

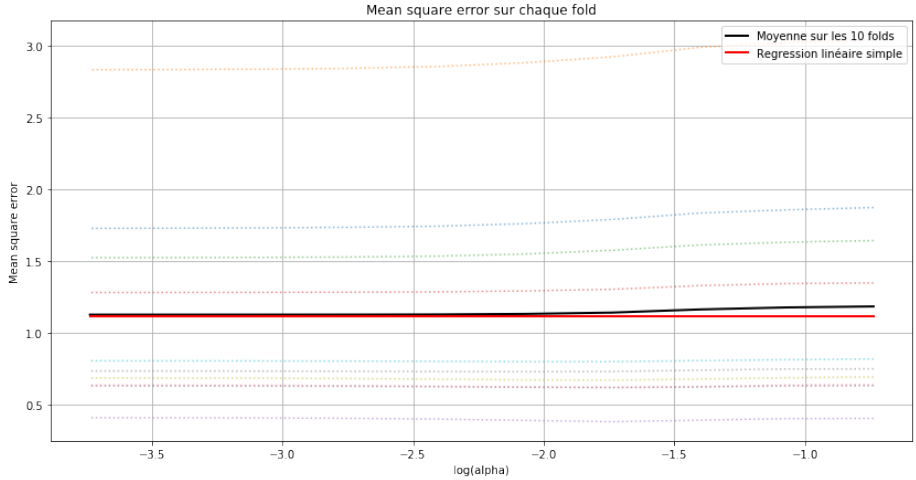
Dans le cas d'une régression linéaire entre les variables explicatives et expliquée, une autre méthode de sélection de variables consiste à entraîner un modèle de régression linéaire avec une pénalisation Lasso. En effet, dans ce cas la fonction de coût s'écrit :

$$L(\beta) = \frac{1}{2}\|Y - X\beta\|_2^2 + \alpha\|\beta\|_1$$

La forme de la pénalité va permettre à certains coefficients de valoir exactement 0. En effet, cela revient à minimiser le premier terme sous une contrainte de la forme $\beta \leq b$. Comme on peut le voir sur la figure suivante (gauche), un objet tangent à une frontière linéaire est très susceptible de rencontrer un des coins de l'hypercube, pour lequel certaines coordonnées de β sont nulles. Ce n'est par exemple pas le cas dans avec une pénalisation ridge.



Afin de fixer le paramètre α, plusieurs valeurs différentes ont été testées. Chaque modèle évalué par cross-validation avec 10 folds. La figure suivante présente l'erreur quadratique moyenne pour chaque fold et en moyenne sur les 10 folds. A titre de comparaison, le score d'une régression linéaire simple a été ajouté en rouge, lui même calculé avec une validation-croisée 10 folds.



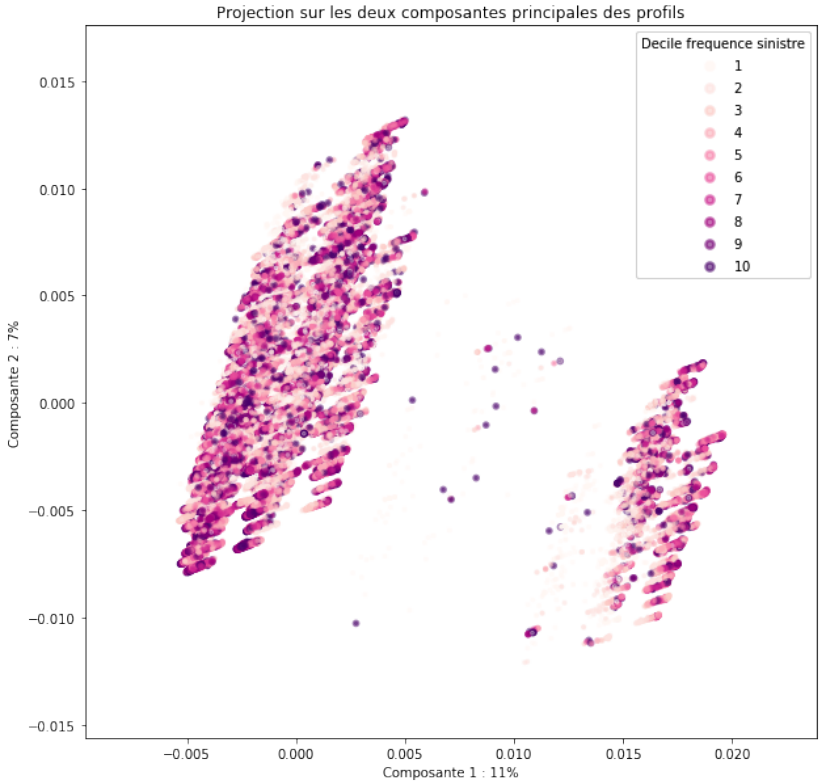
On observe tout d'abord qu'ajouter une pénalisation lasso n'a pas amélioré les capacités prédicatives de notre modèle. Pour des valeurs faibles de α, le modèle est équivalent à une régression linéaire simple. Lorsque α augmente, les poids de la regression tendent vers 0, jusqu'à avoir un modèle qui predit systématiquement $\bar{Y}$. La faible capacité prédictive de ces modèles, avec un R^2 de 5% nous a conduit à rejeter cette méthode.

2.5 Visualisation

Différentes méthodes de visualisation ont été testées, afin de nous donner plus d'informations sur le jeu de données.

2.5.1 PCA

Le but est d'étudier la projection des points dans l'espace engendré par les deux composantes principales. Dans notre cas, la PCA peut aussi être utilisée pour observer la distribution de Y dans l'espace réduit, et permet de mesurer le pouvoir discriminant des deux composantes principales. En effet, à chaque point est associé le décile auquel appartient sa fréquence de sinistre moyenne. Ces valeurs vont donc de 1 à 10. Sur la figure, elles valeurs sont représentées par un gradient de couleur. Si les points de même couleur sont proches, et distants de points d'autre couleurs, alors les variables ont un pouvoir discriminant relatif à Y qui est important. Les résultats sont présentée dans la figure suivante :



On constate une discretisation partielle des coordonnées des points. En effet, dans l'espace de base, certaines variable issus de variables initialement catégorielles ne peuvent prendre que deux valeurs possibles. Chaque composante principale étant une combinaison linéaire de ces variables, les coordonnées de chaque point peuvent donc être vues comme la somme de pondération de valeurs

discrètes (le centre de chaque petite tache rectangulaire par exemple) et de valeurs continues, ce qui explique la largeur des taches que l'on peut observer.

Il est néanmoins difficile d'observer des tendances sur la figure. Le pouvoir discriminant des deux composantes principales est réduit.

2.5.2 t-SNE

L'algorithme t-SNE est une méthode de réduction de dimension non linéaire. Elle permet de construire un espace réduit de deux ou trois dimensions, ce qui se porte bien à la visualisation de données. L'algorithme se base sur une interprétation probabiliste des proximités en transformant les similarités entre les points en probabilité jointe, selon l'expression :

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}$$

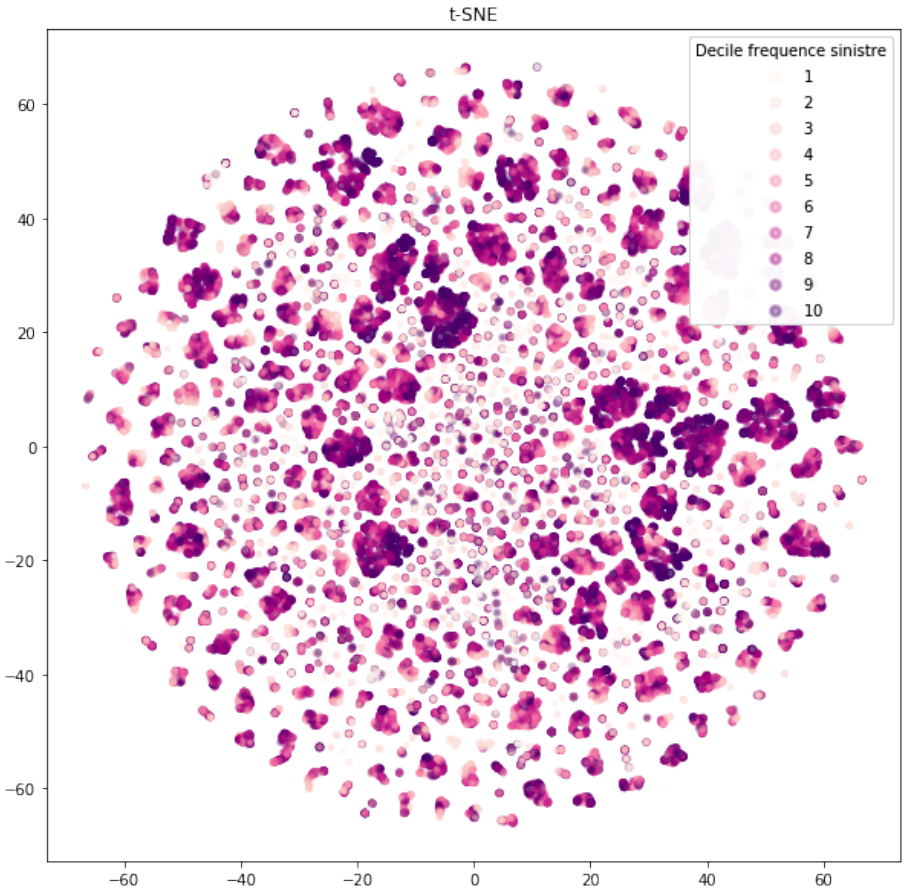
Cette probabilité est également calculée dans l'espace réduit, dans lequel les points sont notées z :

$$q_{j|i} = \frac{\exp(-\|\mathbf{z}_i - \mathbf{z}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{z}_i - \mathbf{z}_k\|^2/2\sigma_i^2)}$$

L'algorithme essaie alors de minimiser la divergence de Kullback–Leibler entre les deux distributions trouvant le meilleur embedding pour les profiles.

Comme avec la PCA, une dimension supplémentaire a été ajoutée, représentée par la couleur de chaque point, et représentant le décile auquel correspond sa fréquence de sinistre moyen, par rapport à celle des autres points.c La motivation est toujours d'observer des tendances liées à Y sur la figure.

La figure suivante présente le résultat obtenu :



On observe que l'algorithme a du mal à séparer les points en groupes homogènes. La figure présente à l'inverse une densité grossièrement constantes, pour les différents déciles.

La méthode n'a pas été utilisée en tant que réduction de dimension dans ce projet car elle perd beaucoup d'information, relative à la densité de points dans l'espace de départ par exemple, et peut donner lieu à des interprétations biaisées. Dans cette section, elle n'a donc pour seul but que de visualiser le jeu de données.

3 Clustering

3.1 Méthodes

3.1.1 K-means

Description Kmeans : K-means est un algorithme non supervisé de clustering non hiérarchique. L'algorithme se base essentielement sur une similarité

(distance euclidienne dans notre cas) pour regrouper des clusters contenant les élements les plus proches aux centres des ces clusters.

Algorithm 1 K-means algorithm

On a utilisé la bibliothèque sklearn pour apliquer un estimateur k-means sur notre jeu de données. L'apprentissage de cinq clusters a pris trois secondes.

3.1.2 Hierarchical Clustering

Contrairement au k-means, le Hierarchical clustering est un algorithm de clustering hiérarchique. L'algorithme commence d'abord par attribuer un cluster à chaque élement de la base de données. Ensuite, on repète les deux étapes suivantes:

- Identifier les deux clusters les plus proches au sens d'une métrique (similarité) bien définie.
- Combiner ces deux clusters dans un seul cluster.

jusquà avoir un cluster qui regroupe toute la base de données.

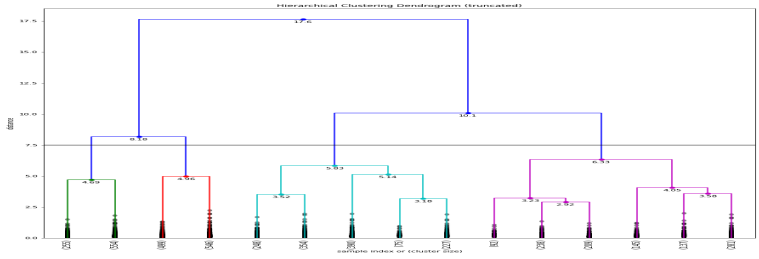


Figure 1: Hierarchical Clustering Dendrogram (truncated)

Sur la figure 1 on voit l'évolution de l'algorithme à différents steps de l'apprentissage. Le nombre de cluster souhaité est par la suite établi en fixant une distance maximale tolérée entre les centres des clusters. Pour 4 clusters dans notre base de données, on a dû fixer cette distance à 7.5.

3.1.3 Gaussian Mixture of Models

Idée : modéliser le comportement statistique à partir de plusieurs populations, groupes ou classes.
Notations :

- n réalisations des variables aléatoires indépendantes et identiquement distribuées (i.i.d), qu'on va noter $(X_1, X_2, ..., X_n)$.

- K different clusters contenant n_k observations.
- p_k La probabilité d'être dans la kième classe et f_k la distribution associée à cette probabilité.

On peut alors suivant les notations précédentes définir la densité de distribution suivante:

$$f(x) = \sum_{k=1}^{K} p_k * f_k(x)$$

Dans notre cas où on a considéré les f_k des gaussiennes on trouve :

$$f(x) = \sum_{k=1}^{K} p_k * \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp(-\frac{(x-\mu_k)^2}{2\sigma_k^2})$$

On se ramène alors à un problème d'estimation des variables inconnues suivantes $\theta = (p_k, \mu_k, \sigma_k)_{k=1,...,K}$ qui être résolu en utilisant l'algorithme Expectation-Maximization (EM).

Une implémentation complète est disponible sur le github du projet. A noter aussi, qu'une deuxième version où on considère des gaussiennes dont la matrice de covariance est diagonale, a été implémenté et est disponible sur le github aussi.

3.2 Evaluation d'un clustering

Il y a plusieurs types de manière d'évaluer un clustering. Certaines necessitent la connaissance des vrais labels, ce qui est dans notre cas impossible comme l'Asjuted Rand Index (ARI) et le Mutual Information Score (MI). Néanmoins, ils peuvent être utilisés pour avoir une indication de la stabilité d'une méthode de la manière suivante : Une clusterisation est opérée sur 75% de la base de donnée, donnant lieu à des labels $c_1^{(1)}, c_2^{(1)}, ..., c_p^{(1)}$, et une autre sur toute la base de donnée, donnant lieu aux labels $c_1^{(2)}, c_2^{(2)}, ..., c_p^{(2)}, ..., c_n^{(2)}$, alors le score entre les séries $(c_i^{(1)})_{1,...,p}$ et $(c_i^{(2)})_{1,...,p}$ est calculé. Ce score permet de mesurer la stabilité du modèle face à l'introduction d'un nombre conséquent de nouveaux points. Dans le cas d'usage, la base de données va être continuellement agrandies, et une bonne méthode de clusterisation ne doit pas voir ses groupes profondément changer.

D'autres ne nécessitent pas de vrais labels, mais font des hypothèses sur la structure de la data. Dans certains cas, le Davies-Bouldin Index a été calculé.

3.2.1 Adjusted Rand Index

L'adjusted rand index ou ARI, est une métrique prenant ses valeurs entre -1 et 1, permettant de mesurer la similarité de deux séries de labels. Elle se base sur le Rand Index, méthode consiste à calculer la proportion de couples de points qui sont soit dans le même groupe pour chaque clustering, A ou B, ou dans des

groupes différents pour chaque clustering, A ou B. Si note ces nombres a et b, alors le random index a pour expression :

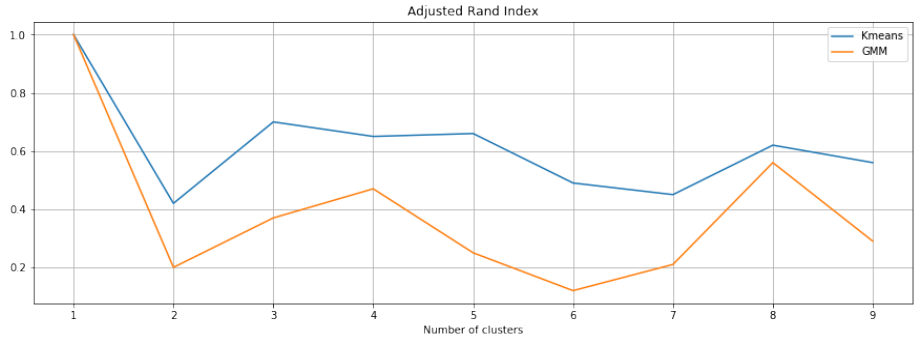
$$Ri = \frac{a+b}{C_2^n}$$

Pour rendre cette métrique indépendante du nombre de cluster, et garantir un score proche de 0 pour une distribution aléatoire des labels, on définit le Adjusted Rand Index :

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]}$$

où $E[RI]$ représente le rand index moyenne pour un labelling aléatoire.

Pour une méthode donnée, si on fixe le nombre de cluster à 1, alors tous les points ont le même label, et l'ARI vaut 1. A l'inverse, si les labels sont



Nous avons observé beaucoup de variance dans l'ARI selont le seed aléatoire utilisé. A terminer (D'autres métriques sont disponibles ici : https://scikit-learn.org/stable/modules/clustering.html)

3.2.2 Davies-Bouldin Index

L'indice de Davies-Bouldin dournit une mesure de la séparabilité des clusters. Soit (C_i) un ensemble de clusters. Notons, pour tout i, s_i la distance moyenne entre chaque point du cluster i et le centre de ce cluster, d_ij la distance entre les centres des clusrters i et j. on définit une mesure de similarité : $R_ij = \frac{s_i+s_j}{d_ij}$
Alors l'index s'écrit :

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j} R_{ij}$$

3.3 Résultats

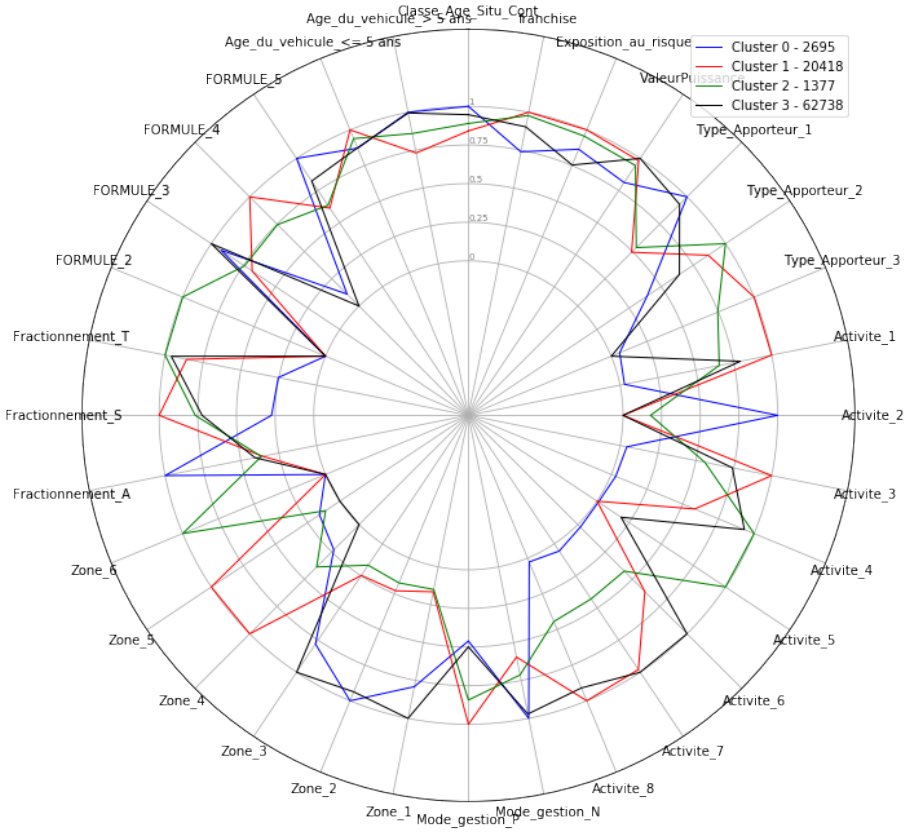
3.3.1 Adjusted Rand Index

Cet index est calculé pour chaque méthode de clustering, et pour un nombre de cluster allant de 1 à 10. A chaque couple de point peut être associée une variable booléenne

3.4 Profil moyen

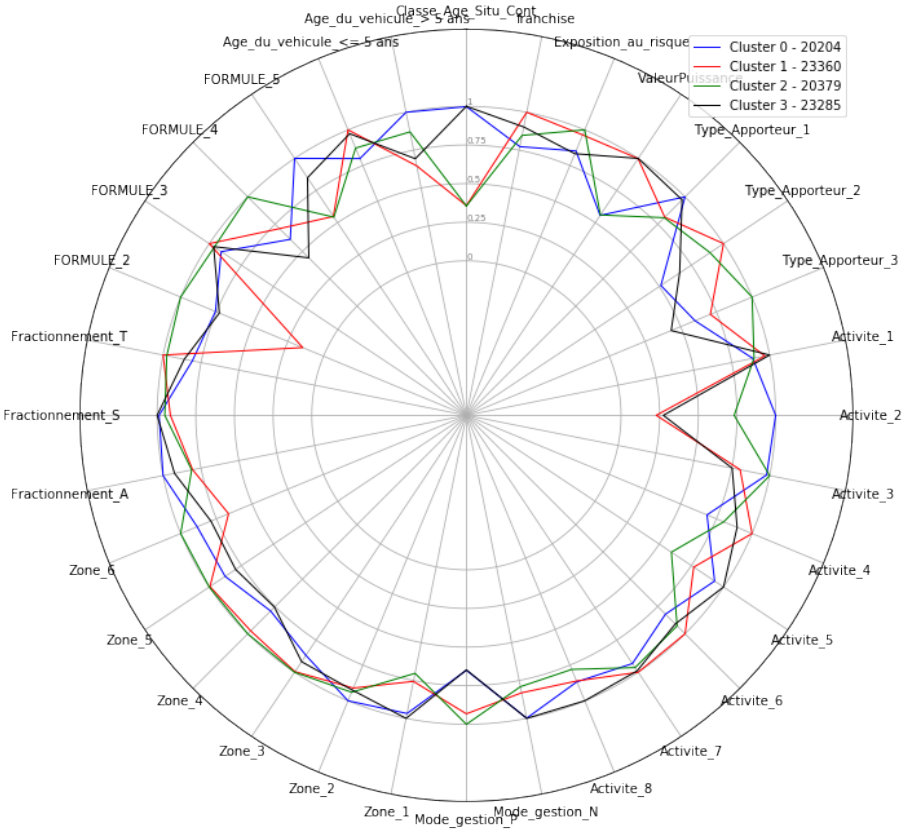
Dans cette section sont étudiées les profils moyens de chaque cluster pour différentes méthodes. Ils permettent de voir quels variables ont été discriminantes lors de la séparation des profils en groupes.

3.4.1 GMM



On remarque que les clusters on des tailles très différentes (cf légende) mais que les plus petits clusters contiennent tout de même plus de 1000 profils. On observe également que les variables catégorielles sont très significatives. Plusieurs d'entre elles sont en moyenne (donc pour tous les profils) nulles pour certains clusters. Ainsi, dans le cluster 1 (en rouge), on ne retrouve que les zones 1, 5 et 6. Dans le cluster 0 (en bleu), on ne trouve que des profiles exerçant l'activité 3. A l'inverse, l'exposition au risque, et la puissance du véhicule sont en moyenne quasi-similaire dans tous les clusters.

3.4.2 Kmeans



On observe tout d'abord que les clusters sont de tailles très homogènes. On remarque que dans tous les clusters, la moyenne de chaque variables est presque la même (hormis pour activité 2 et FORMULE_2). Les variables étant soit continues, soit binaires, cela montre que dans chaque cluster, la présence d'une catégorie d'une variable (initiale) est la même, et les variables continues prennent des valeurs moyennes très proches. Cela reflète une mauvaise séparation des points en sous groupes.

4 Livrable

Github