RAPPORT DE TP : TRAITEMENT DES DONNEES / BIG DATA

MESLOUH Mohamed Arselan - BERTON Thomas - BOUKEZZATA Aymen

31 Décembre 2021

Table des matières

Table des figures

Pour ce TP, nous avons fait le choix d'intégrer les courbes et figures, que nous jugeons éssentiels à la bonne présentation et à la compréhension du document, au coeur du document.
Nous avons , bien évidemment, veiller à respecter la limites des 15 pages maximum en terme de texte pure.
Les figures auxilliaires et peu parlante ont été insérées en annexe.

Tous nos codes sont disponibles au format notebook sur le dépot GIT suivant : https://github.com/MeslouhArselan96/TP$_BIG-DATA_MESLOUH.git$

Introduction

Le but de ce TP est de presenter comparer trois (03) méthodes de clusterisation repandues. Chacune des méthodes à une algorithmique différente et des propriétés variées qui permettent de la rendre plus ou moins adéquate à utiliser en fonction du dataset que l'on a. Nous allons d'abord appliquer ces trois algorithmes à des datasets artificiels dont on connaît la clusterisation adéquate au préalable. Nous allons faire en sorte de choisir pour chacune des méthodes, une panoplie de datasets avec des spécificités différentes (granularité, concave/convex, nombre de clusters, etc.), pour ensuite faire une analyse (en nous aidons des datasheets des algorithmes en question) pour determiner les atouts et les limites de chacune des méthodes.

Par la suite, nous allons appliquer ces trois méthodes à une dataset réelle qui contient un nombre supérieur de "features". Pour être en mesure d'avoir une visualisation quelque peu raisonnable, nous allons appliquer un algorithme de réduction de features, en l'occurrence : PCA.

Nous n'avons pas de métrique spécifique qui permet de dire lequel des algorithmes est le plus adapté pour la dataset, mais nous allons néanmoins tenter d'analyser les résultats obtenus et en tirer des remarques concernant le comportement de chacun des outils.
Les méthodes que nous avons choisies sont : - K-Means ; - Clustering Agglommeratif ; - DBSCAN.

Le choix des datasets peu ne pas être idéal, et c'est pourquoi nous allons faire appel à la litératture concernant ces algorithmes pour confirmer nos appréciations.

Nous considérons que le but n'est pas de tester un maximum de datasets, ni de pouvoir aboutir à des clustering réussis mais plutot de pouvoir faire face à des cas d'atude pertinents qui vont nous pousser à faire appel à une observation critique fine pour essayer de cerner les atouts et les limites de ces algorithmes, mais aussi de developper un sens de la nuance lorsque l'on aborde le domaine des sciences de données qui manipule des outils (métrique d'avluations, algorithmes) pour la plupart non détérministes.

1 Clustering K-Means

1.1 Clustering

La premiere partie du Lab consistait en le fait de manipuler K-Means avec les datasets les plus différents qui soient. Nous avons choisis quatres (04) datasets differents pour cet effet, en l'occurence :

1. **2d-10c** : Qui est un dataset en 2D relativement simple, très peu bruité, avec un nombre de clusters elevé mais distincts, une densité de cluster relativement élévé mais une granularité générale basse.
2. **cluto-t8-8k** : Qui est un dataset 2D assez granuleux, avec un nombre elevé de clusters et nottamment des formes non convexes et un niveau de bruit relativement considérable.
3. **3-Spiral** : C'est un dataset en 2D avec seuelement 3 clusters mais de formes non convexes
4. **hepta** : C'est une dataset en 3D relativement semblable a 2d-10c, le but est de tester la robustesse de l'algorithme avec des datasets de dimensions supèrieure à 2.

Pour chacun des sets de données nous avons fait des barplot des histogrammes pour vérrifier si la distribution des échantillons etait "normale" (gauss), et ce pour éviter des éviter des altérations de calculs dues à la présence d'outliers.

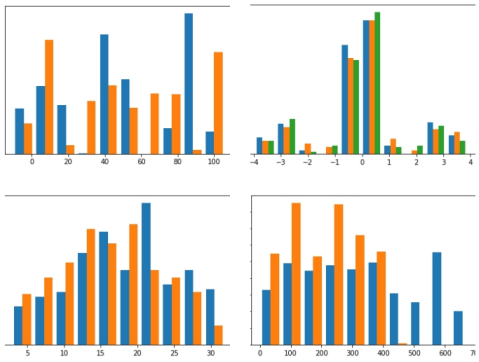
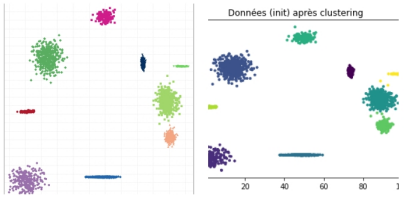


FIGURE 1 – Histogrammes des datasets K-Means

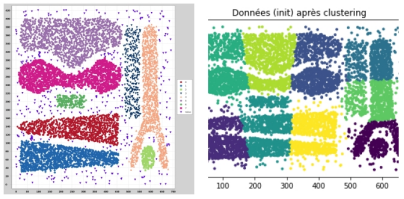
Nous remarquons clairement que les quatres datasets sont normalement distrbués.
Nous appliquons un K-Means aux trois premières datasets :



Résultat attendu / Clustering K-Means

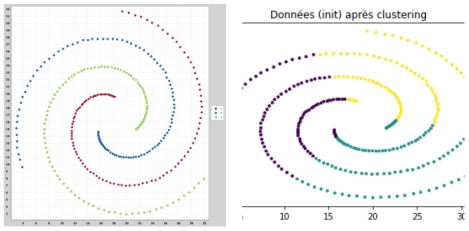
FIGURE 2 – Clustering K-Means - 2d-10c

Nous remarquons que K-Means repond trés bien à ce genre de dataset. Nous avons effectué un premier clustering (Figure 2) avec K=9 qui à l'air de trés bien fonctionner.



Résultat attendu / Clustering K-Means

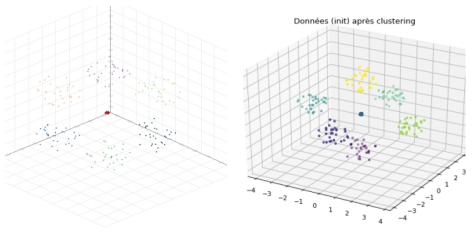
FIGURE 3 – Clustering K-Means - cluto-t8-8k



Résultat attendu / Clustering K-Means

FIGURE 4 – Clustering K-Means - 3-Spiral

Nous avons voulu, à titre experimental, tester la robustesse de l'algorithme K-Means face à des datasets de dimension supériere à 2.
pour cela nous avons utiliser la dataset "hepta", qui reste une dataset qui correspond visuellement à des datasets avec lesquels K-Means ne rencontre pas tellement de difficultés (High density, low noise, low granularity).
Nous avons eu les résultats suivants :



Résultat attendu / Clustering K-Means

FIGURE 5 – Clustering K-Means - hepta (3D)

1.2 Evaluation

Pour évaluer notre clustering nous avons utilisé deux (02) métriques spécifiques, à savoir : L'inertie et le coéfficient de silhouette.
Comme nous connaissons dèja le nombre de clusters attendu pour chaque dataset, nous verrifions si les métriques d'évaluations confirment ce que nous pensons.
L'inertie est une métrique qui sert à determiner une plage adequate au nombre de clusters qu'il faut choisir pour les algorithmes qui prennent le nombre de clusters en entrée. Nous appliquons la méthode du coude pour determiner cette plage.
Le coéfficient de silhouette lui, atteint son maximum pour le nombre de clusters convenable, dans les deux cas (2D et 3D).

Dans notre première dataset, le coéfficient de silhouette fonctionne parfaitement en affichant un maximum à k=9 clusters tandis que l'inertie, si nous appliquons la méthode du coude suggére elle, quatres (04) clusters. Nous avons essayé un clustering avec k=4 (résultat consultable sur notebook du TP1), le résultats était assez satisfaisant avec un regroupement de clusters visuellement cohérent, mais la métrique dans ce cas precis montre ces limites, ce qui démontre bien que les métriques elles mêmes ne sont passi fiables que cela, et que le choix de la métrique est sensible à la nature de la dataset.

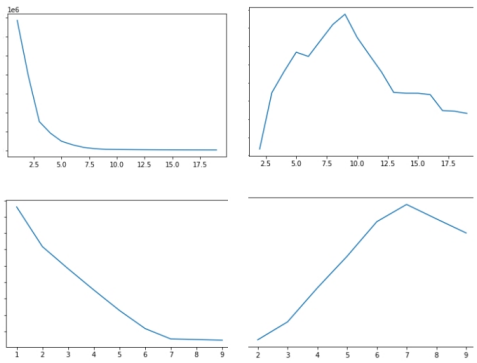


FIGURE 6 – Inertie et Silhouette pour 2d-10c et hepta

2 Clustering Agglomeratif

2.1 Clustering

La deuxième partie du TP consiste à manipuler le clustering agglomeratif avec différentes datasets pour que, comme avec K-Means, on arrive à cerner les atouts et limites de cet algorithme.
Pour ce faire, nous avons choisis 2 datasets, en l'occurence : "2d-4c-no4" et "banana". Il s'agit de 2 datasets qui, ensemble, couvrent un spectre assez large de differentes caractéristiques (l'un est concave et l'autre convexe, le nombre de clusters est differents, la difference du niveau de bruit est consistente).
Le clustering agglomeratif utilise dans ses calculs, 4 stratégies différentes pour merge les échantillons en clusters, à savoir : Single, Ward, Complete, Average, chacune étant plus propice à un type de datasets.
Nous avons appliqué un clustering agglomératif avec les 04 stratégies pour chacun de nos dataset, dans le but de determiner la stratégie appropriée pour chaque dataset. Nous avons obtenu les résultats suivants :

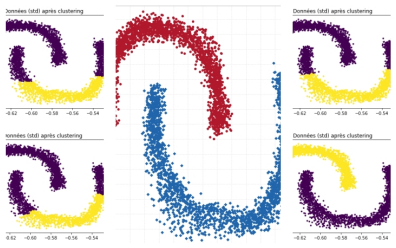
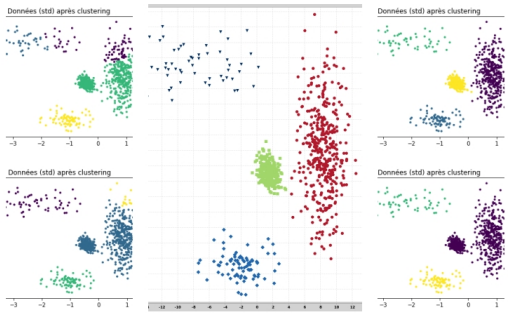


FIGURE 7 – Clustering Agglomératif - banana



Clustering agglomératif (Complete, Average, Ward, Single)

FIGURE 8 – Clustering Agglomératif - 2d-4c-no4

Pour chacune des figures ci-dessus, nous retrouvons le résultat attendu au centre, tandis que les clusterings effectuées sont sur les bords dans l'ordre suivants :

- Complete - en Haut à Gauche
- Average - en Bas à Gauche
- Ward - en Haut à Droite
- Single - en Bas à Droite

Nous remarquons que pour le premier dataset (banana), c'est le linkage = Single qui a le mieux fonctionnée, tandis que pour le second dataset, c'est le linkage = Ward qui à permis d'aboutir au résultat attendu.

2.2 Evaluation

Pour évaluer ces clustering, nous avons voulu comparer les coéfficients de sihlouette en fonction du linkage pour chacun des datasets.
Par la suite, nous avons jugé intéréssant de pouvoir observer l'évolution du coéfficient de silhouette en fonction du nombre de clusters, et ce, pour chaque linkage.
Nous avons abouti à des courbes comme suit :

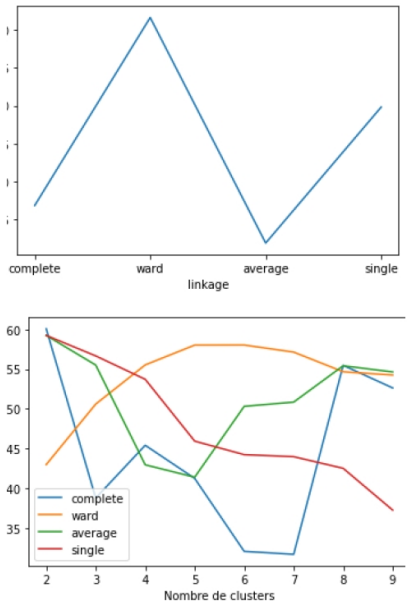


FIGURE 9 – Evaluation - agglomeratif - banana

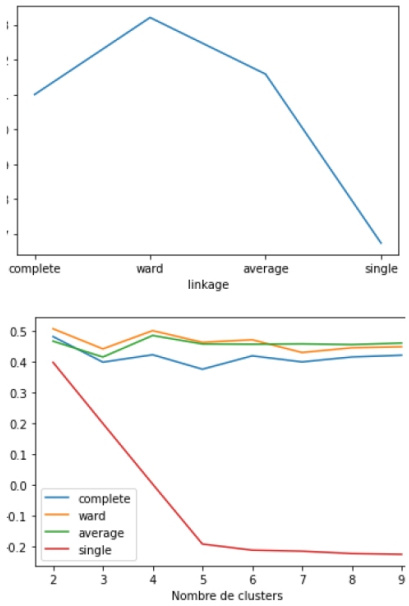


FIGURE 10 – Evaluation - agglomeratif - 2d-4c-no4

Nous remarquons sur le graphe "hybride" que "single" se distingue clairement comme étant le meilleur linkage pour le dataset banana lorsque le nombre de clusters = 2. Cependant, lorsque l'on observe le premier graphe single est inférieur à ward.
Aussi, il est clair que le nombre de clusters idéal est 2 car le coeff de silhouette atteint son maximum.

Pour ce qui est de la 2ème dataset, ward se distingue trés clairement, et ce sur toute la ligne. Nous remarquons aussi que la courbe jaune atteint son maximum autours des 04 clusters.

Pour conclure, nous pouvons dire que la métrique "silhouette" est assez fiable pour ce qui est de l'évaluation du clustering agglomératif.

3 Clustering DBSCAN

Dans cette partie, nous allons aborder le clustering DBSCAN. Cette méthode repose sur deux paramètres primordiaux, à savoir le rayon du clusters (epsilon) et le nombre minimum de points à l'intérieur du rayon nécéssaire pour définir l'agglomérat de point comme étant un cluster (min-pts).

Ce mode de fontionnement trés spécifique, nous pousse à modifier quelque peu notre démarche. En effet, nous allons commencer par l'évaluation et nous passerons au clustering par la suite.

3.1 Evaluation

Le but ici, est de de trouver les meilleurs valeurs pour epsilon et min-pts.
Pour trouver epsilon, nous allons utiliser un algorithme appelée KNN qui va renvoyer pour chaque point, la distance avec les K plus proches voisins en mesurant la distance (dans notre cas : euclidienne). Nous allons choisir un valeur de K et faire en sorte pour chaque point d'obtenir une moyenne de distance des K plus proches voisins.
Par la suite, nous allons ordonner (par ordre croissant) ces distances et les tracer de facon à avoir une courbe croissante. La théorie veux que la bonne valeur de epsilon se trouve au voisinnage du coude de la courbe.

Nous allons considerer ensuite 03 valeurs de epsilon dans le voisinnage de la première courbe et tracer pour chacune de ses valeurs : le coéfficient de silhouette en fonction de min-pts.
Le point Maximum du tracé va être considéré comme étant le point critique pour aboutir à un bon clustering.

Pour ce faire, nous avons choisis 3 datasets assez différentes qui vont nous permettre de cerner les points faibles et forts de l'algorithme.
Nous avons opter pour :

- banana : qui est un dataset peu bruité, avec un petit nombre de clusters et des formes non convexes ;
- cluto-t7-10k : qui est un dataset trés granuleux, assez bruité et des formes à la fois convexes et non convexes ;

— cure-t2-4k : qui est un dataset assez buité et des formes à la fois convexes et non convexes ;

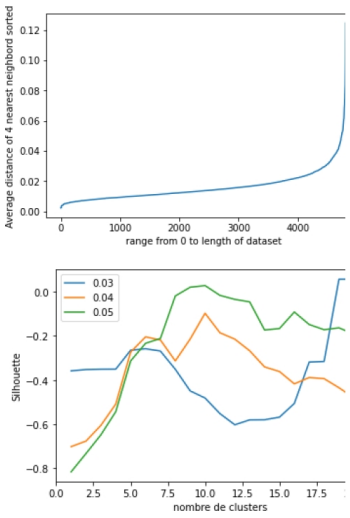


FIGURE 11 – epsilon / min-pts - banana

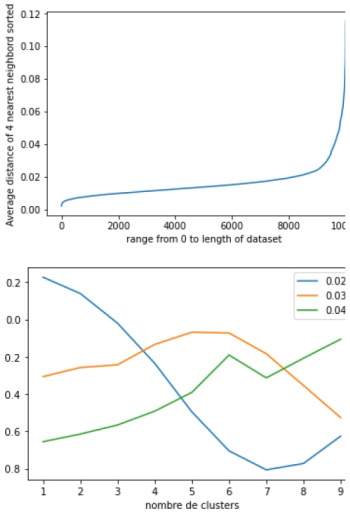


FIGURE 12 – epsilon / min-pts - cluto-t7-10k

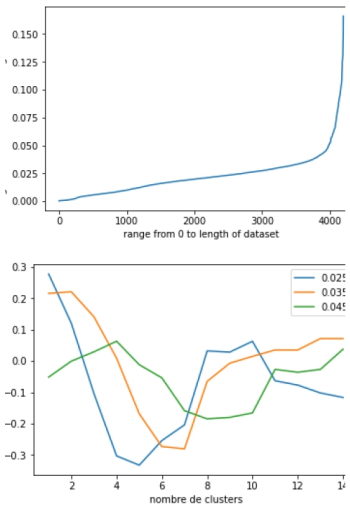
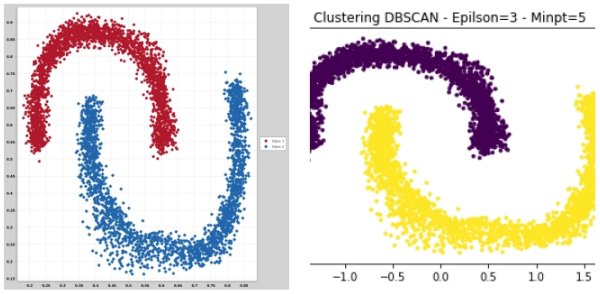


FIGURE 13 – epsilon / min-pts - cure-t2-4k

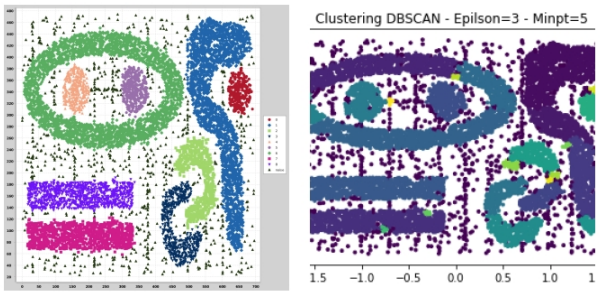
Ces trois graphes nous permettent de déduire les paramètres nécassaires à nos clustering.

3.2 Clustering



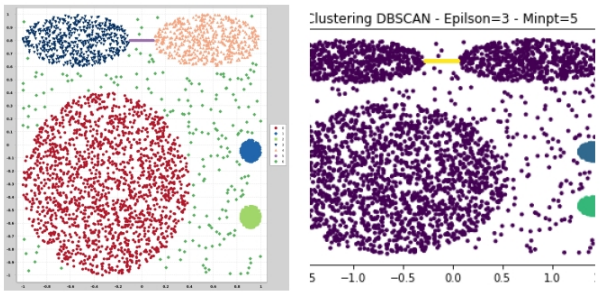
Résultat attendue/ Clusters DBSCAN (0.3 - 10)

FIGURE 14 – Clustering DBSCAN - banana



Résultat attendue/ Clusters DBSCAN (0.05 - 10)

FIGURE 15 – Clustering DBSCAN - cluto-t7-10k



Résultat attendue/ Clusters DBSCAN (0.05 - 10)

FIGURE 16 – Clustering DBSCAN - cure-t2-4k

Nous remarquons, à partir d'une analyse visuelle des clustering que :

— Le clustering DBSCAN n'est pas approprié pour les formes non-convexes : sur la 2eme et 3eme datasets, seuls des portions convexes des clusters ont été reconnus comme étant des clusters.

— Le clustering DBSCAN est sensible au bruit, d'apparence en tous cas, car dans les 2 datasets bruitées il reconnait le bruit comme cluster.

— La métrique d'évaluation 'silhouette' n'est pas adaptée pour évaluer cette algorithme. En effet, nous remarquons que le coéfficients de silhouette passe parfois en dessous de 0 ce qui signifie une erreur et que la dataset ne permet pas d'être évaluée avec cette métrique. Il aurait été possible de tester d'autres métriques mieux adaptée comme "**Davies-Bouldin score**", mais le but ici était pour nous, justement, de pointer du doigt ces limites avant d'appliquer nos algorithmes à la dataset réelle.

4 Dataset réelle

Pour cette partie, nous avons décider d'appliquer nos algorithmes directement sur la dataset réelle et de concentrer nos éfforts sur l'interpretation et l'analyse des résultats de clustering sur la dataset, mais aussi sur quelques manières de tranformer notre dataset afin de faciliter la visualisation et la bonne application des algorithmes.

Pour ce qui est du jeu de données, il s'agit d'un dataframe regroupant des données de pluviométries et autres indicateurs météos et géographiques, et ce sur une année de temps pour 33 villes francaises.

4.1 Data pre-processing

Tout d'abord nous appliquons quelques bons usages de pré-traitement de données afin d'optimiser les résultats de nos traitements.
Nous vérrifions qu'il n'y a pas de données manquantes. Ensuite, nous affichons un histogramme de chacune des features. Le but ici est de vérrifier la distribution des données et faire en sorte de d'avoir un jeu de données homogène qui ne comporte pas d'outliers.
Les figures des histogrammes ont été insérés en annexe pour consultation de l'avant aprés traitement. Les outliers sont des exemples isolés et extremes qui ne refletent pas le jeu de donnée et qui risquent de biaiser le comportement de nos algorithmes. Nous appliquons donc un "**outlier treatment**" pour les features qui nous semblent en contenir.
Nos process est simple, nous voulons détérminer le 1er et 3eme quartille de notre distribution, et réduire la distribution globale à une distribution inter-quartille qui s'apparente à une cloche 'normale' trés adoptés dans les sciences de données.

Par la suite, nous avons "drop" la colonne contenant le nom des villes, qui ne vas pas être interprétée par l'algorithme et qui de ce fait represente une surcharge.

Nous avons aussi, fait un mapping de la colonne géographie en termes numériques pour que'elle puisse être interprétée.

Nous avons fini avec un scaling des données à l'aide d'un standard-scaler disponible dans la librairie sklearn. Le but du scaling est de mettre à l'échelle standard (de 0 à 1) les données pour homogéinisé leur impact sur la prise de décision et le calcul algorithmique.

4.2 Réduction des composants - PCA

Dans cette phase, nous avons appliquée un algorithme PCA de réduction de dimension, le nombre de collonne étant trop large. Le but est de créer des nouvelles features que nous ne savons pas encore interpreter, qui representent la grande majorité des informations utiles de notre jeu de donnée et de permettre une visualisation plus claire.

Nous avons décidé de réduire notre datasets à 6 features. Nous avons par la suite tracer un eboulis de la somme des pourcentages d'inertie représentée par chacun des components.

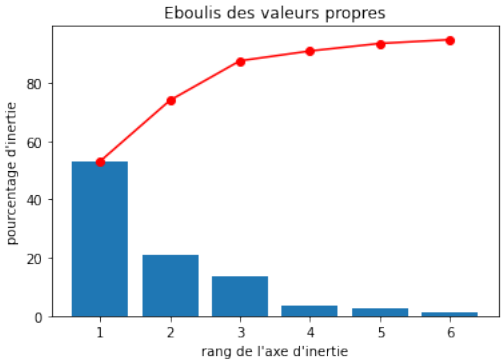


FIGURE 17 – Eboulis des inérties - PCA

Nous remarquons que les 04 premieres features representent plus de 90 pourcents de notre dataset. Il s'agit d'un bon compromis entre une datasets réduite en terme de features et de représenation des données utiles.
Nous aurions pu opter pour 03, le but est juste de trouver un nombre assez petit qui puisse representer le maximum d'informations. Nous avons opter pour 04.

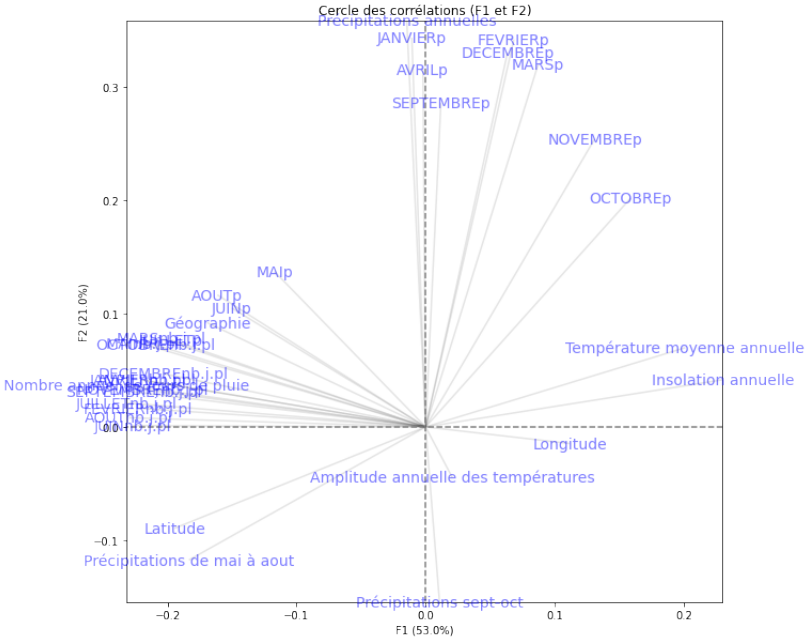


FIGURE 18 – Cercle de corrélations - PCA 1 et PCA 2

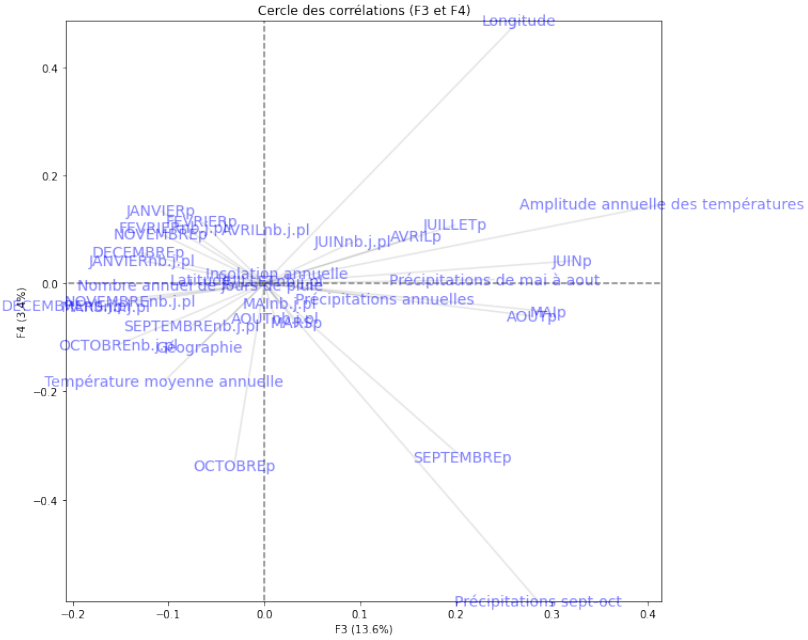


FIGURE 19 – Cercle de corrélations - PCA 3 et PCA 4

Dans le but d'illustrer ce que peuvent representer nos nouvelles features, nous

établissons un cercle de corrélation entre les anciennes et les nouvelles features.

A partir du premier cercle, qui represente PCA 1 et PCA 2, nous arrivons a observer qu'il existe une grande corrélation entre les features qui representent la pluviometrie et PCA 1. De la même manière, nous remarquons une corrélation forte entre les features qui representent les nombres de jours de pluie et PCA 2. Nous arrivons a déduire, donc, que PCA1 et PCA2 representent respectivement des variations dans les données de pluviometrie et de nombre de jours de pluie. Nous pouvons à partir de cette interpretation aboutir à d'autres déductions en ce qui concerne nos clusters et l'influence de chaque parametre sur la clusterisation.

Pour ce qui est de PCA 3 et PCA 4, il n'est pas très claire de pouvoir dire ce que ces nouvelles features regroupent des données des anciennes features, car il n'existe pas de corrélation forte particulierement evidente à observer.

4.3 Clustering et Evaluation

Nous avons dans cette dèrniere étape, appliquer les algorithmes de clustering étudiés auparavant, et nous avons opté pour une represenation des clusters en "pairplot", à savoir, en regroupant les données en affichant un plot 2D pour chaque 2 features, 2 par 2. Le résultat est symétrique pour les paiplot affichés, il suffit d'observer les 06 plots supérieurs de droite pour avoir toutes les combinaisons possibles.

4.3.1 Clustering Agglomératif

Nous appliquons un clustering agglomeratif sur notre dataset. En observant la variation du coéfficient de silhouette en fonction du nombre de clusters et du linkage, nous remarquons qu'un bon clustering se situerait autours des 3 OU 4 clusters pour un linkage complete ou ward. Le but pour nous est d'opter pour un nombre de clusters relativement petit pour lequel ont obtiens une bonne évaluation, de facon à pouvoir ensuite interpréter le resultat.

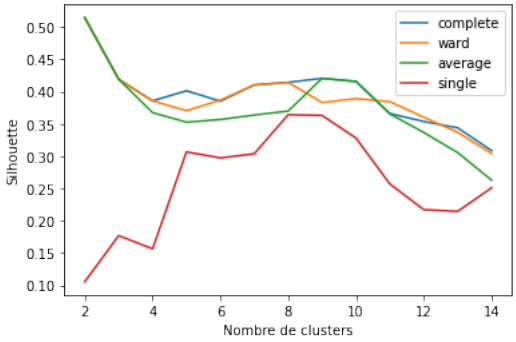


FIGURE 20 – Evaluation/Sihlouette - Agglomeratif

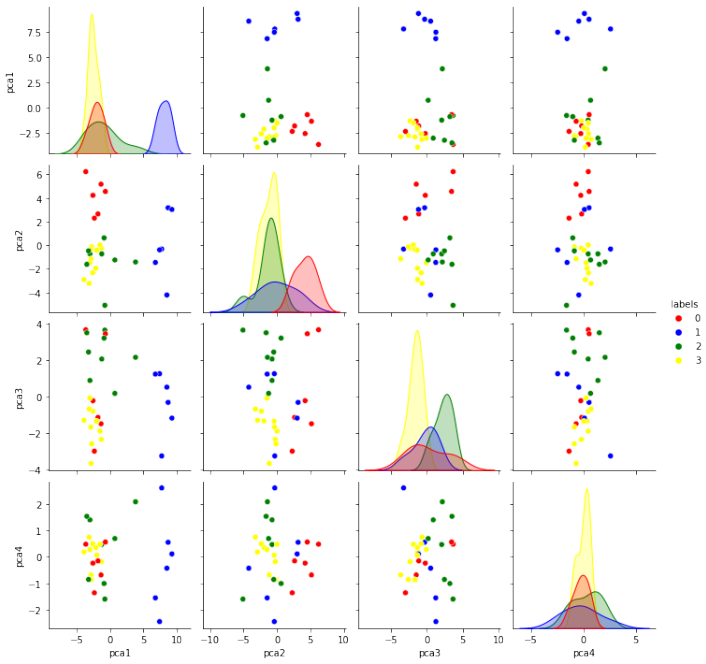


FIGURE 21 – Clustering Agglomeratif

4.3.2 Clustering K-Means

Nous appliquons cette fois-ci, un clustering K-Means, avec un nombre de clusters égal à 3 comme le suggere la méthode du coude appliquée en mesurant le coéfficient d'inertie.

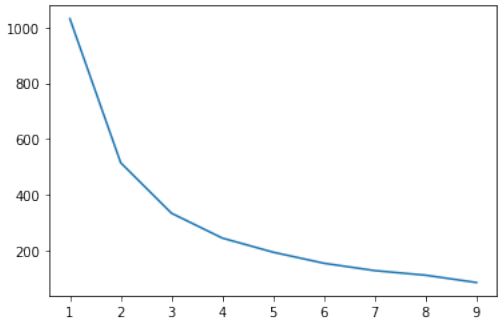


FIGURE 22 – Evaluation/Intertie - K-Means

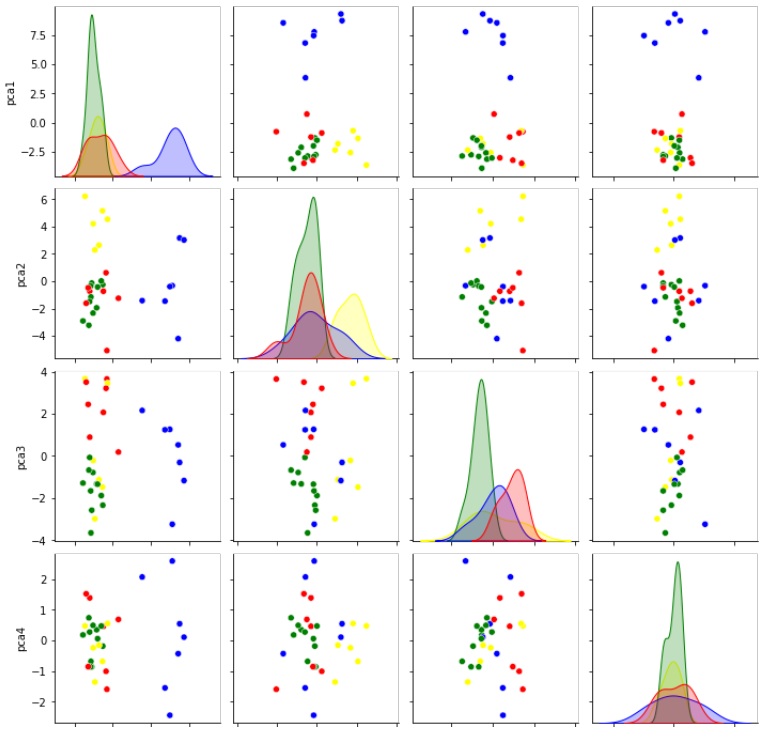


FIGURE 23 – Clustering K-Means

4.3.3 Clustering DBSCAN

Nous appliquons finalement un clustering DBSCAN, en appliquons la méthode utilisée auparant pour determiner epsilon et min-pts. Nous aboutissons à une valeur de min-pts=3 pour un epsilon de 1.5 qui donne un coéfficient de silhouette convenable pour un nombre entier naturel de clusters supérieure à 1.

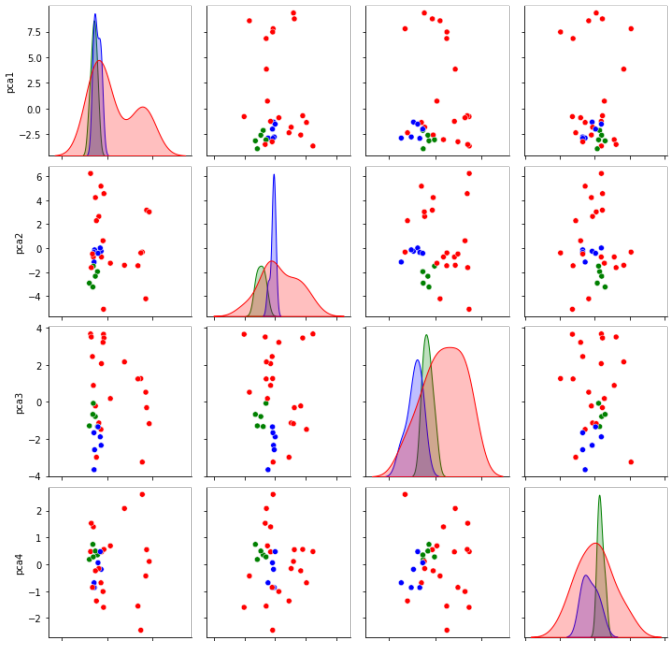


FIGURE 24 – Clustering DBSCAN

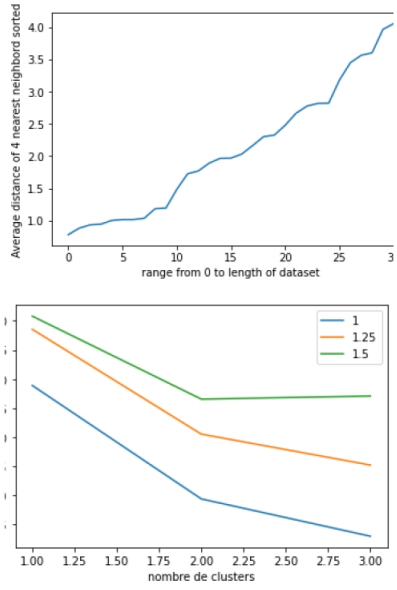


FIGURE 25 – Evaluation/Silhouette-KNN - DBSCAN

Conclusion

Une démarche qui consiste à comparer les résultats obtenus à partir des 03 clusering n'est pas envisageable, en raison du manque de métriques d'évaluation qui permettent de determiner la qualité d'un clustering. Néanmoins, il est possible d'analyser les Clustering obtenus à partir des 03 méthodes.

Comme nous arrivons à proposer des significations raisonnables pour la significations des deux premières features PCA, nous allons nous concentrer sur les clusters issus des plot (PCA1, PCA2).
Nous remarquons, que les 03 algorithmes distingues toujours 3 clusters simmilaires.

En supposant que PCA1 et PC2 representent respectivement la pluviométrie et le nombre de jours de pluie, nous arrivons a reconnaitre que tous nos points de données se situent dans une sorte de gamme milieu en terme de pluviométrie. Néanmoins, une différence appreciable se distignue lorsqu'il s'agit du nombre de jours de pluie.Les clusters sont distribués selons cette coordonnée (Y) et nous distinguons 3 clusters évidents pour chacun des algorithmes.

Lorsque l'on s'atarde sur le cercle de corrélation de PCA3 nous nous rendons compte que les features liées aux mois de l'année ou l'hiver est present (OCTOBRE, NOVEMBRE, DECEMRBE, JANVIER, FEVRIER) sont considérablement corrélés avec l'axe PCA3. Nous pouvons donc imaginer que PCA3 represente une modulation des caractéristiques pluviométriques de la saison hivernale. Il existe une concordence entre cette analyse éventuellement crédible et le fait que les memes clusters qui aparaissent sur (PCA1, PCA2) soient présents sur (PCA1, PCA3).

Une facon de voir les choses serait de se dire que les clusters representent les villes chaudes et froides. En réalité, et pour avoir un comportement algorithmique pértinent, il faudrait augmenter le nombre de points de données qui ici est insuffisant pour admettre le dataset comme exploitable sur le plan analytique.

Annexes

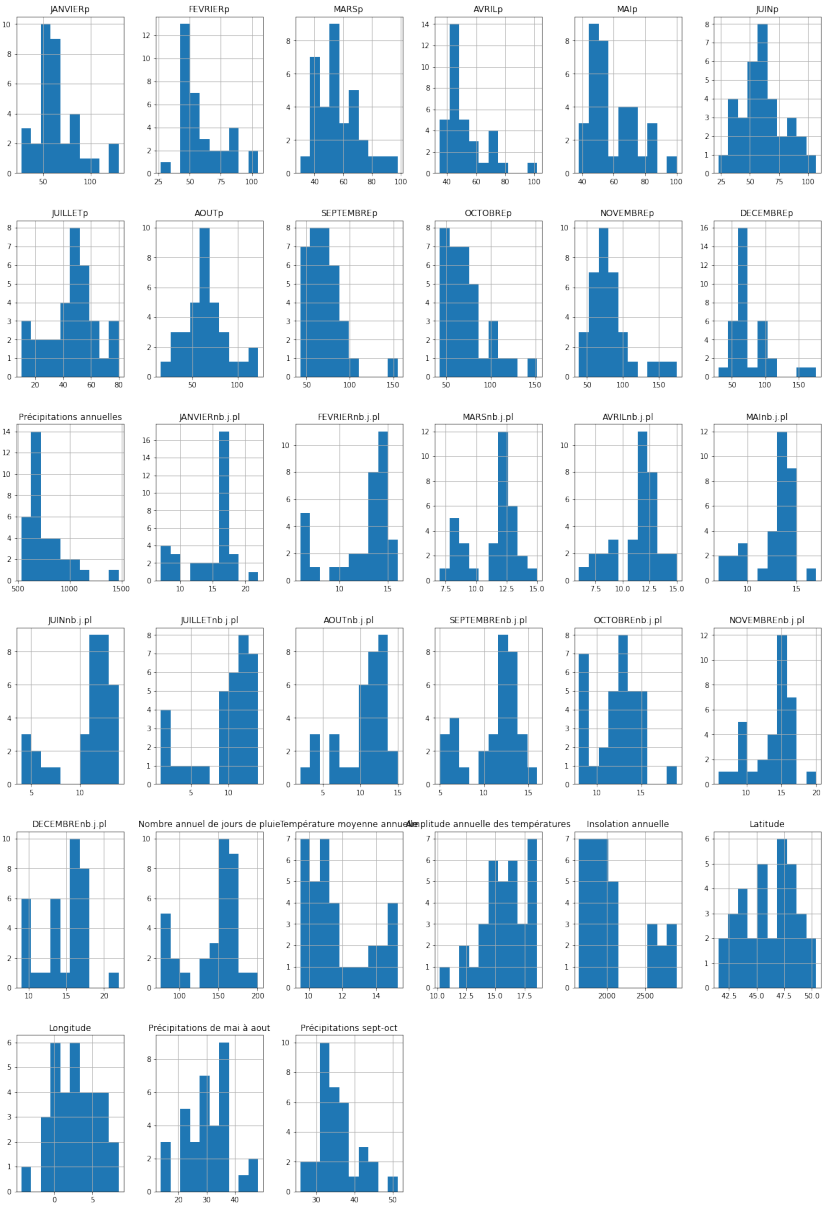


FIGURE 26 – Histogrammes avant cleaning

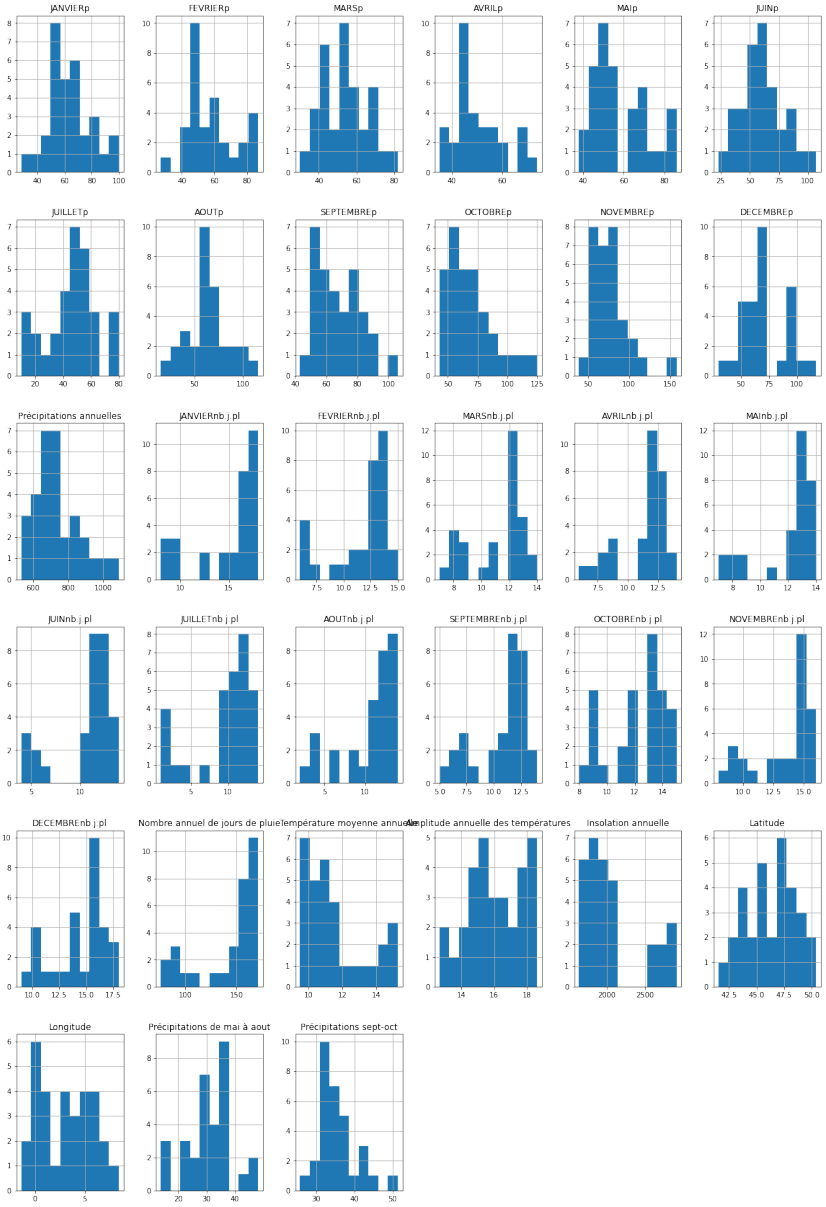


FIGURE 27 – Histogrammes après cleaning

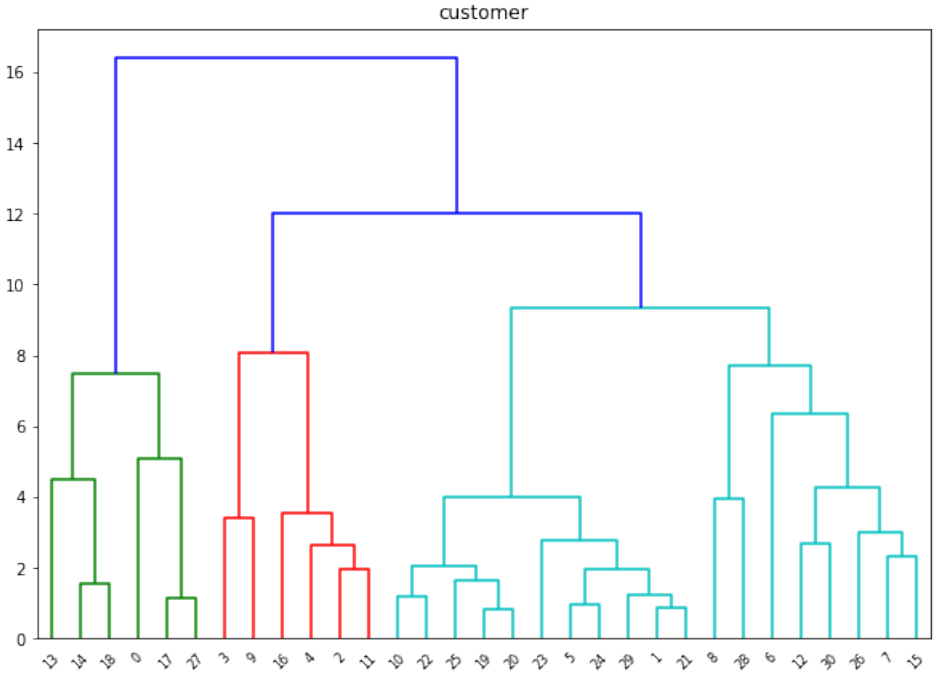


FIGURE 28 – Dendrogramme - Pluie