RAPPORT DE TP DE CLUSTERING DE TEO GENEAU

ANALYSE ET TRAITEMENT DE DONNÉES – BIG DATA

30 juin 2022

Informations relatives au document

INFORMATIONS GÉNÉRALES

Auteur(s)	Teo Geneau
Lieu	INSA Toulouse
Version	V3.0
Référence	EA1

HISTORIQUE DES MODIFICATIONS

Version	Date	Rédigé par	Modifications
V3.0	30/06/2022	Teo Geneau	Version Finale : ajouts de chapitres et corrections

DESTINATAIRES

Nom	Fonction / Entité
Marie-José Huguet	Professeur / INSA

TABLE DES MATIERES

TABLE DES MATIERES

INTRODUCTION

Lors de ce projet d'analyse et traitement de données, nous nous sommes familiarisés avec différentes méthodes de clustering :

- Le clustering k-Means
- Le clustering Agglomératif
- Le clustering DBSCAN

L'objectif principal est de tester ces différentes méthodes de clustering sur des jeux de données différents afin d'étudier l'impact de leurs paramètres respectifs sur les résultats, il s'agira aussi de déterminer sur quel type de données ils peuvent s'appliquer, et cela, afin de dégager les points forts et les points faibles de chaque méthode. Pour cela, la librairie en analyse de données et apprentissage automatique scikit-learn a été utilisée.

Une analyse comparative de ces différentes méthodes sera réalisée.

Méthodes de clustering

Clustering k-Means

Le clustering est une méthode d'apprentissage non supervisée, de plus en plus utilisée notamment pour le traitement de l'image, le traitement des données du signal, etc.
Dans ce chapitre, nous nous intéresserons à la méthode K-Means, aussi appelée méthode des centres mobiles, elle est apparue dans les années 50/60, il s'agit d'une méthode très populaire due à sa simplicité et sa rapidité de calcul. L'algorithme est simple et applique une stratégie gloutonne se reposant sur un nombre k de cluster à fixer, puis on affecte à chaque donné le centre le plus proche, on recalcule des nouveaux centres et on réitère jusqu'à atteindre une certaine stabilité ou bien lorsqu'on atteint un nombre défini d'itérations. Ainsi, on peut discriminer des groupes dans un jeu de données.

Afin d'étudier cette méthode, on utilisera des jeux de données fournis : x1, x2, z1_20 et z3_50. Par évaluation visuelle, on peut compter 15 clusters pour x1, 15 pour x2, 20 pour z1_20 et 50 pour z3_50, que l'on utilisera comme valeur de référence pour la partie analytique.

On applique alors la méthode K-Means pour un nombre fixé de clusters sur ces différents jeux de données que l'on représentera en 2D.

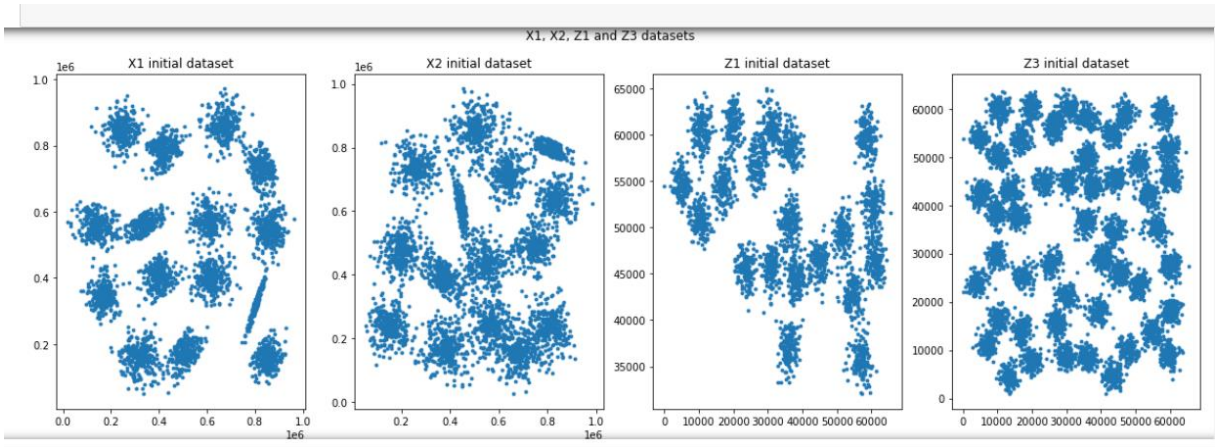


Figure 1 : représentation en 2D des différents jeux de données

Afin d'évaluer le nombre de cluster il est nécessaire d'utiliser différentes métriques telles que : le coefficient de silhouette, l'indicateur d'inertie, l'indictateur d'homogénéité, le V-measure, l'indice de Davies-Bouldin, etc.

Ici, nous n'utiliserons comme indicateurs que le coefficient de silhouette et l'indice de Davies-Bouldin, permettant d'évaluer la densité et la séparation des clusters, et ainsi évaluer la qualité de chaque clustering.

Nous allons tester l'algorithme sur les 4 jeux de données précédents en faisant varier le paramètre k, correspondant au nombre de cluster, pour cela, nous utiliserons une boucle for, fixant le nombre de k de 2 à 55, afin de tester 54 valeurs de k, en évaluant la qualité de chaque clustering. On utilisera l'initialisation kmeans++ (centres éloignés les uns des autres).

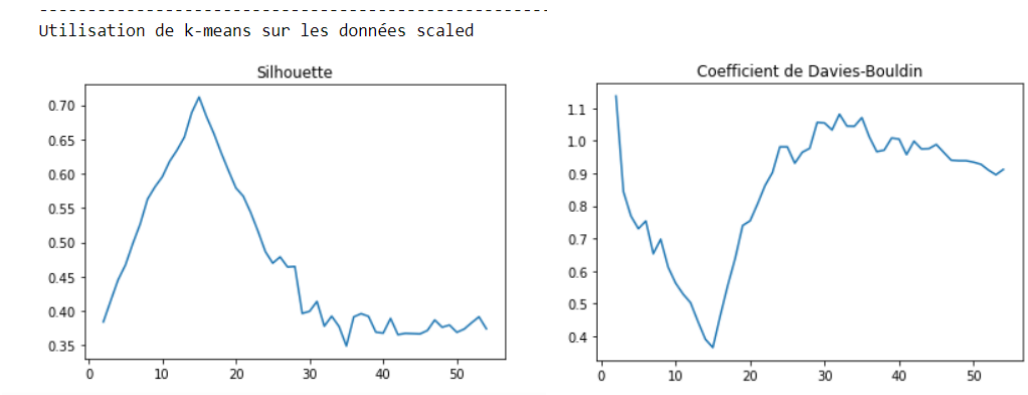


Figure 2 : graphiques des différentes métriques utilisées

Chaque valeur des métriques est stockée, selon la valeur de k, dans des listes que l'on utilisera pour déterminer la meilleure qualité de clustering. Plus le score de silhouette est proche de 1, plus les clusters sont denses et bien séparés les uns des autres, la fonction max sera alors utilisée afin de déterminer la valeur maximum du coefficient et la valeur de k correspondante. Inversement, plus l'indice de Davies-Bouldin est faible meilleure sera la qualité du clustering, c'est pourquoi la fonction min sera utilisée dans ce cas pour déterminer la valeur de l'indice.

Il se peut que la valeur de k diffère selon la métrique utilisée, cette dernière sera donc moyennée afin de sélectionner la meilleure valeur de k.

Les résultats suivants sont obtenus :

	X1	X2	Z1	Z3
Coeff de Silhouette max :	**15**	**15**	**20**	**50**
Indice de Davies-Bouldin min :	**15**	**15**	**19**	**50**
K idéal	**15**	**15**	**20**	**50**
Temps de calcul	**51.1s**	**64s**	**37.1s**	**154.4**

On peut remarquer que les temps de calcul est relativement élevé, cela peut s'expliquer par le fait que l'ordinateur utilisée pour réaliser les calculs dispose seulement de 4Go de RAM et un processeur peu performant, qui plus est, on applique l'algorithme à chaque tour de la boucle for, soit 54 fois.

Par ailleurs, plus le nombre de points est élevé, plus le temps de calcul est grand, x1 et x2 sont des tableaux de 5000 lignes et 2 colonnes, z1 3000 lignes et 2 colonnes alors que z3 fait 7500 lignes et 2 colonnes.

Le nombre de clusters obtenu pour les 4 jeux de données correspondant au nombre attendu, les clusters sont plutôt bien séparés même si l'on peut noter des différenrces.

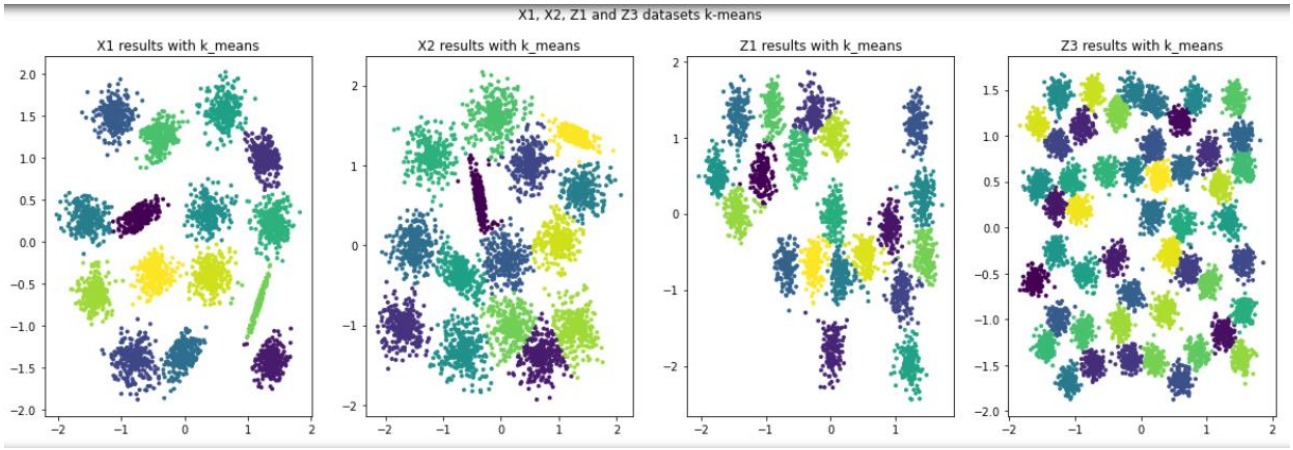


Figure 3 : clustering obtenu sur les 4 jeux de données avec la méthode k-means

Analyse des points faibles

Comme on peut le voir sur les jeux de données, certaines données sont attribuées aux mauvais clusters, cela est notamment due à la sensibilité de la méthode k-Means à la densité des points, à leur forme et à leur taille, ce qui signifie qu'une donnée peut être attribué au mauvais cluster s'il considère que la densité d'un autre cluster est plus forte par exemple.

C'est une méthode basée sur la distance entre les points, or, la distance en y et en x n'est pas assez élevée pour dissocier correctement les clusters et l'algorithme semble parfois bloqué dans un optimum local (que l'on peut identifier entre le cluster vert et violet en bas à gauche du graphique d'X1). C'est bien ce qui est observable notamment en comparant les graphiques pour X1 et X2, le nombre de données est similaire entre les deux jeux, néanmoins la densité des clusters est plus élevée pour X1 et les distances entre clusters sont plus élevées, c'est ce qui explique le nombre d'erreur plus élevé pour X2.

D'autres points faibles existent, comme la nécessité de fixer le nombre de cluster au départ, or cela a un fort impact sur le résultat final car la phase d'initialisation est importante.

Analyse des points forts

Pour ces jeux de données, l'algorithme reste tout de même performant, notamment car il fonctionne bien pour les géométrie plate (en deux dimensions, ce qui est le cas ici) et les clusters possédant en majorité la même taille.
En outre, cette méthode offre d'autres avantages : elle est plutôt simple à utiliser et il n'y a qu'un seul paramètre à gérer (k).

Clustering agglomératif

Dans cette partie, nous allons évaluer une méthode hiérarchique ascendante, appelé Clustering Agglomératif. La méthode consiste à considérer chaque point au départ comme un cluster et puis, peu à peu, fusionner les observations les plus proches par similarité, on itère jusqu'à obtenir 1 seul cluster.
Les résultats sont représentés avec un dendrogramme.
Le nombre de cluster initial est nécessaire pour initialiser l'algorithme ainsi que le choix des paramètres d'agglomération pour déterminer les similarités, il en existe quatre différents :

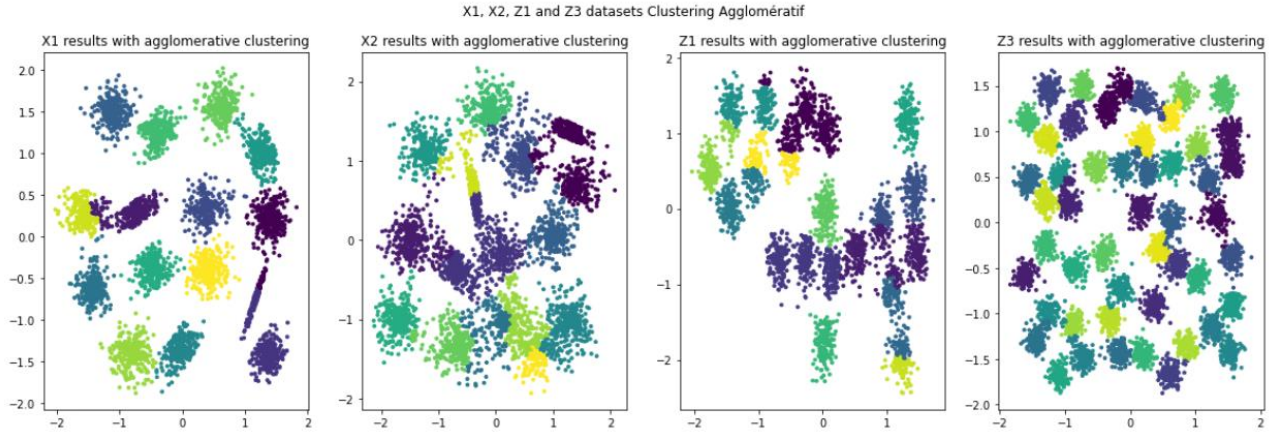
- Single linkage : consistant à minimiser la distance à celle des deux points les plus proches deux à deux.
- Complete linkage : correspondant à la distance maximale entre deux paires de clusters/
- Ward : l'augmentation de la variance intra-cluster est minimale.
- Average linkage : la distance moyenne entre deux points correspond à la moyenne des distances de tous les points.

Les résultats suivants sont obtenus :

Paramètre Complete

	X1	X2	Z1	Z3
Coeff de Silhouette max :	**14**	**16**	**9**	**50**
Indice de Davies-Bouldin min :	**14**	**11**	**17**	**47**
K idéal	**14**	**14**	**13**	**48**
Temps de calcul	**65.8s**	**64s**	**21.6s**	**150.9s**

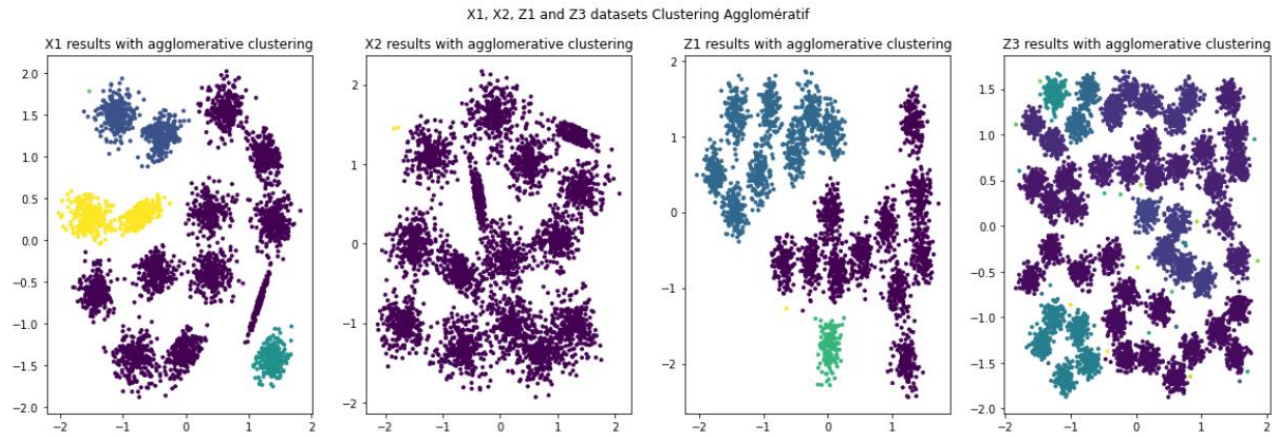
```
Appel Aglo Clustering 'complete' pour une valeur de {k} determinée automatiquement
```



Paramètre Single

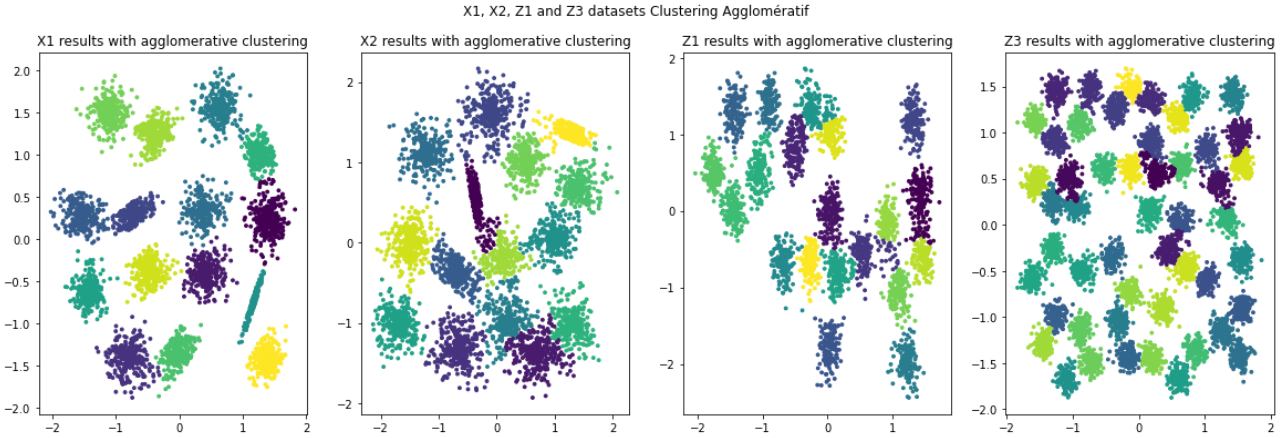
	X1	X2	Z1	Z3
Coeff de Silhouette max :	**8**	**2**	**2**	**18**
Indice de Davies-Bouldin min :	**2**	**3**	**5**	**50**
K idéal	**5**	**2**	**4**	**34**
Temps de calcul	**41.2s**	**40.4s**	**16s**	**101.4s**

```
Appel Aglo Clustering 'single' pour une valeur de {k} determinée automatiquement
```



Paramètre Ward

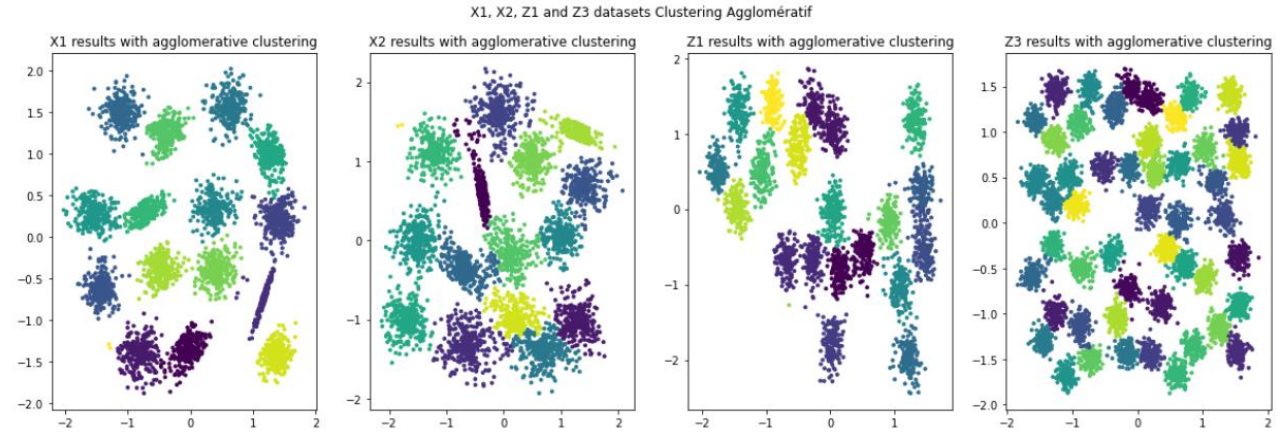
	X1	X2	Z1	Z3
Coeff de Silhouette max :	**15**	**15**	**20**	**50**
Indice de Davies-Bouldin min :	**15**	**15**	**19**	**50**
K idéal	**15**	**15**	**20**	**50**
Temps de calcul	**74.6s**	**69.4s**	**23.3s**	**154.7**



Paramètre Average

	X1	X2	Z1	Z3
Coeff de Silhouette max :	**15**	**15**	**21**	**50**
Indice de Davies-Bouldin min :	**17**	**16**	**16**	**48**
K idéal	**16**	**16**	**18**	**49**
Temps de calcul	**61.7s**	**59s**	**21.5s**	**143.9s**

```
Appel Aglo Clustering 'average' pour une valeur de {k} determinée automatiquement
```



Plusieurs observations sont possibles, premièrement, selon le paramètre d'agglomération les résultats varient en fonction des jeux de données.

Le nombre de cluster exact a été trouvé seulement avec le paramètre ward, il est plus précis que la méthode k-means pour le jeu X1 mais il est moins précis pour les jeux de données avec des distances moins élevées. Le paramètre average offre aussi un bon résultat à un ou deux clusters près tandis que le paramètre complete a des résultats plutôt moyens, notamment avec le jeu de données Z3, qui présente des distances en x et y plus faibles par rapport à X1 et X2. Et finalement, le paramètre single est le moins adapté des quatre, il n'a pas déterminé le bon nombre de cluster quelque soit le jeu de données.

Les temps de calculs ne varient pas énormément par rapport à la méthode k-mean alors que les calculs sont plus complexes.

Analyse des points faibles

La complexité du calcul augmente sensiblement, on passe ici à une échelle plus difficile. L'algorithme est sensible aux différentes anomalies car elle est toujours basée sur un calcul de distance. La lecture du dendrogramme peut être compliquée pour des jeux avec beaucoup de données et de clusters.

Analyse des points forts

Utilisable avec un nombre de clusters plus élevé que la méthode K-Means. Par ailleurs, il n'a pas besoin d'avoir un nombre fixe de clusters, ce qui en fait une méthode plutôt flexible, ce dernier est à établir en fonction du dendrogramme.

Clustering DBSCAN

L'objectif de cette méthode est de pouvoir obtenir des clusters avec des formes non convexes. Il fonctionne alors par voisinage (le nombre de point compris dans un rayon donné) : tous les points atteignables dans un rayon sont considérés comme appartenant au même cluster. Elle se repose donc sur la distance entre les deux points les plus proches. Deux paramètres sont utilisés pour optimiser l'algorithme : epsilon (la distance entre deux points) et min-samples (le nombre de points dans un même cluster).

Pour déterminer la valeur d'epsilon, on utilise une fonction de calcul des plus proches voisins qui maximise le k-nearest neighbors (KNN) score afin de fixer epsilon pour que quasi tous les points du jeu de données possèdent un voisin à une distance égale à epsilon.

Le paramètre min-samples sera déterminé grâce à une boucle for de 2 à 20, la valeur offrant un nombre de cluster le plus proche de la valeur de référence et le bruit le plus faible possible.

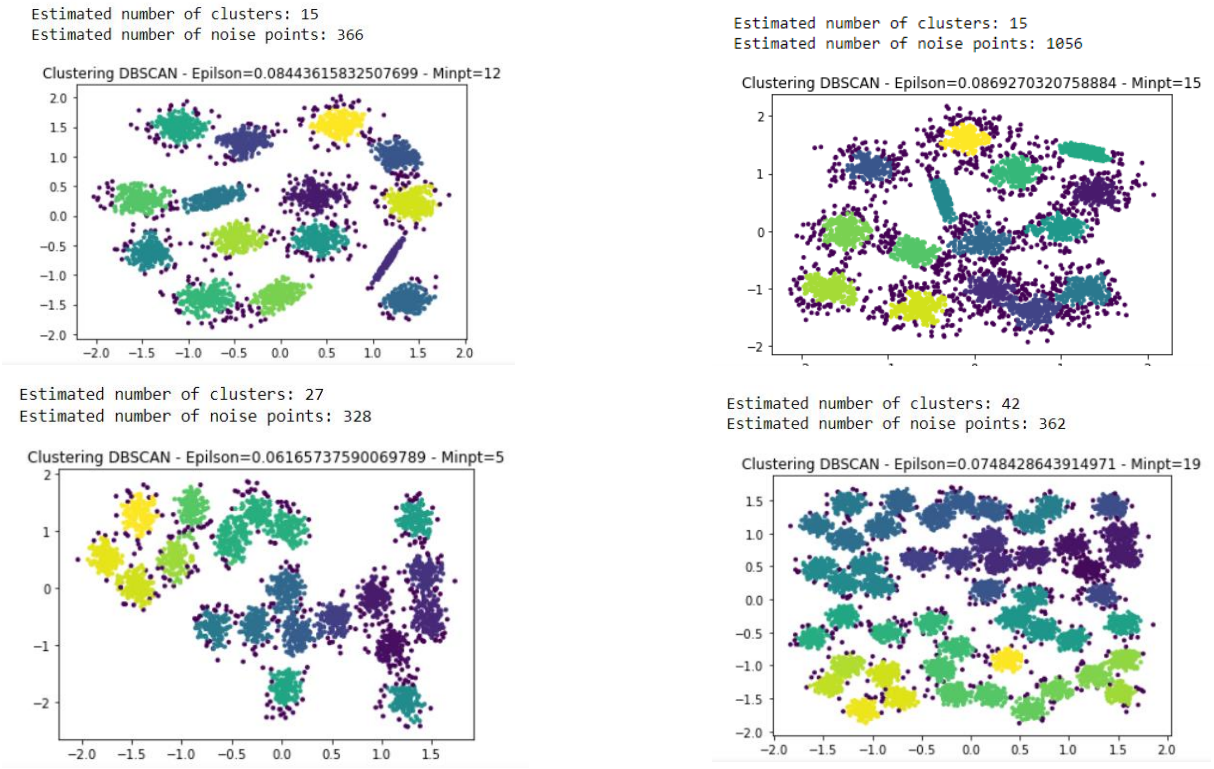


Figure 4 : clustering obtenu sur les 4 jeux de données avec la méthode DBSCAN (X1, X2, Z1, Z3)

Afin de trouver le meilleur réglage, après avoir fixé epsilon par le calcul, min-samples est alors paramétré entre 2 et 20 afin d'optimiser le clustering. L'algorithme a les meilleurs résultats pour le jeu X1 avec le nombre de clusters exact et un faible bruit, pour X2, le nombre de clusters est aussi exact cependant le bruit est beaucoup

plus élevé. Cependant l'algorithme est moins adapté pour les jeux Z1 et Z3, le nombre de cluster est assez loin du nombre exact.

Analyse des points faibles

Les paramètres ne sont pas évidents à déterminer et le fait de devoir trouver le bon nombre de voisins ainsi que la taille du voisinage complexifie l'implémentation de l'algorithme. En effet, si on utilise une valeur d'epsilon trop faible, on aura beaucoup d'anomalies, alors que si on met une valeur trop grande on aura, au contraire, une discrimination de clusters trop faible et certains clusters ne seront pas dissociés.
De plus, l'algorithme est sensible aux variations de densité des données et plus les dimensions sont importantes plus le réglage des paramètres est complexe.

Analyse des points forts

L'un des principaux intérêts par rapport aux autres méthodes est qu'il n'y a pas besoin de fixer le nombre de cluster k. De plus, cette méthode offre l'avantage de pouvoir déterminer des clusters non convexes. Il permet aussi d'éliminer le bruit, ce qui le rend plus robuste aux anomalies que les autres méthodes précédentes.

De plus, avec un temps de calcul de 5.2s pour X1, 4.6s pour X2, 4.7s pour Z1 et 9.5s pour Z3, c'est la méthode de clustering la plus rapide parmi les 3 testés.

CONCLUSION

Au cours de ce projet, j'ai pu explorer différentes méthodes de clustering, appréhender leurs avantages et désavantages afin d'être en mesure de déterminer quelles méthodes seraient la plus approprié selon le contexte du jeu de données. Il n'existe pas de méthode universelle permettant de traiter parfaitement n'importe quel jeu de données, en effet, certaines méthodes sont plus appropriées dans certains contextes que d'autres. Pour les jeux de données de ce projet, c'est la méthode k-means qui a permis de traiter au mieux tous les jeux, même si le clustering agglomératif avec le réglage ward était aussi pertinent, tandis que la méthode DBSCAN est la moins appropriée des trois.

Maîtrisant déjà les parties de collecte, transport, stockage et visualisation des données, ce projet m'a donné envie d'aller plus loin afin d'approfondir la partie analyse et traitement des données, je compte ainsi exploiter les différentes méthodes vues sur d'ancien et futurs projets, mais aussi tester de nouvelles méthodes qu'offre la librairie scikit-learn.