INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

TP Apprentissage non-supervisé

Machine Learning

5SIEC

Département Génie Electronique et Informatique, INSA Toulouse
29 Janvier 2024

Moad El Haddad Louzari
Liam Chrisment

1 Introduction

Dans le cadre de ce TP, notre attention se porte sur l'apprentissage non supervisé. Plus précisément, nous explorons différentes méthodes, dont le clustering, une approche visant à regrouper des données brutes non étiquetées en fonction de leurs similitudes et disparités. Parmi les méthodes de clustering disponibles, nous nous intéressons particulièrement au k-Means et au clustering agglomératif. L'objectif de ce travail est de mettre en œuvre ces techniques sur des ensembles de données bidimensionnels afin de comparer leurs performances et d'identifier les avantages et les limites spécifiques à chacune.

Le code python est disponible sur le dépôt git suivant: https://github.com/Moadvincent/Machine_Learning.git

2 Clustering k-Means

2.1 Étude de la méthode k-Means

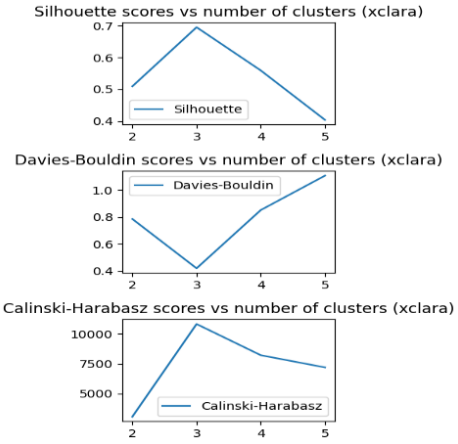
2.1.1 Intérêts de la méthode k-Means

Pour examiner les avantages de la méthode k-Means, nous avons opté pour deux ensembles de données bien adaptés à cette approche, caractérisés par des clusters facilement discernables (les points étant plus proches de leur propre centre de gravité que des autres). Ainsi, nous avons sélectionné les ensembles de données xclara et square4 pour explorer les bénéfices de cette méthode. Nous avons cherché à déterminer automatiquement le nombre de clusters en utilisant le coefficient de silhouette, l'indice de Davies-Bouldin, ainsi que l'indice de Calinski-Harabasz, offrant ainsi plusieurs moyens d'évaluation.

```
Dataset 1: xclara
nb clusters = 2 , runtime = 151.35
Silhouette score = 0.5090670981434706
Davies-Bouldin score = 0.7846968819787055
Calinski-Harabasz score = 3057.312941061147
nb clusters = 3 , runtime = 165.87
Silhouette score = 0.6946138271011265
Davies-Bouldin score = 0.4205054893046378
Calinski-Harabasz score = 10831.591395142126
nb clusters = 4 , runtime = 128.76
Silhouette score = 0.5588966518177478
Davies-Bouldin score = 0.8504553227553966
Calinski-Harabasz score = 8216.229130123924
nb clusters = 5 , runtime = 155.09
Silhouette score = 0.4036577173491296
Davies-Bouldin score = 1.1045502903260789
Calinski-Harabasz score = 7192.466669787075
```

Figure 1: Coefficient de silhouette , Indices de Davies-Bouldin et Calinski-Harabasz et temps de calcul pour le dataset xclara

Nous remarquons sur la figure 1 que le nombre optimal de clusters semble être de 3. En effet sur les itérations qui suivent, nous notons que le coefficient de Silhouette et l'indice de Calinski-Harabasz diminuent tandis que l'indice de Davies-Bouldin augmente. La figure 2 vient confirmer cela.




```
Optimal number of clusters (Silhouette) = 3
Optimal number of clusters (Davies-Bouldin) = 3
Optimal number of clusters (Calinski-Harabasz) = 3
```

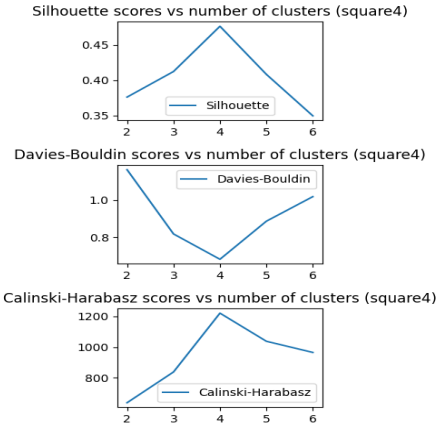
Figure 2 : Nombre optimal de clusters (n = 3)

Nous avons appliqué la même approche au dataset square4 pour confirmer l'efficacité de la méthode k-Means. En analysant le coefficient de silhouette ainsi que les indices de Davies-Bouldin et Calinski-Harabasz, nous avons pu déterminer un nombre optimal de clusters, à savoir 4.

```
nb clusters = 4 , runtime = 30.7
Silhouette score = 0.4762987953658658
Davies-Bouldin score = 0.679796453895008
Calinski-Harabasz score = 1222.0277229514272
nb clusters = 5 , runtime = 39.08
Silhouette score = 0.4083621933939093
Davies-Bouldin score = 0.8860965512234301
Calinski-Harabasz score = 1037.7853792910753
nb clusters = 6 , runtime = 27.65
Silhouette score = 0.3498012849704048
Davies-Bouldin score = 1.0190070763844685
Calinski-Harabasz score = 964.7974512447857
```

Figure 3: Coefficient de silhouette , indices de Davies-Bouldin et Calinski-Harabasz, et temps de calcul pour le dataset square4

Pour ce dataset, dès la 4ème itération, le coefficient de silhouette et l'indice de Calinski-Harabasz diminuent tandis que l'indice de Davies-Bouldin augmente. La figure 4 vient appuyer cela.




```
Optimal number of clusters (Silhouette) = 4
Optimal number of clusters (Davies-Bouldin) = 4
Optimal number of clusters (Calinski-Harabasz) = 4
```

Figure 4 : Nombre optimal de clusters (n = 4)

La méthode k-Means semble être adapté pour des clusters isotropic et de forme convexe.

2.1.2 Limites de la méthode k-Means

Pour évaluer les limitations de la méthode k-Means, nous avons opté pour l'utilisation des datasets xor et zelnik1. Peu importe le nombre de clusters choisi, le coefficient de silhouette demeure remarquablement bas. De manière similaire, l'indice de Davies-Bouldin reste étroitement proche de 1. Par ailleurs, l'indice de Calinski-Harabasz n'atteint pas des valeurs significativement élevées, indiquant ainsi que la méthode ne parvient pas à déterminer de manière adéquate le nombre de clusters approprié (voir Figure 3).

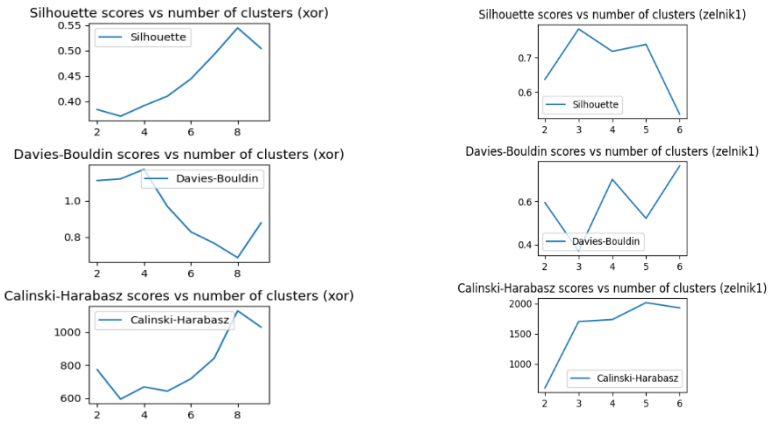


Figure 5: Coefficients de silhouette et indices de Davies- Bouldin et Calinski-Harabasz pour le datasets xor et zelnik1

Nous pouvons effectivement voir sur la figure 6 que la méthode k-means ne permet pas d'identifier correctement les clusters pour ces datasets.

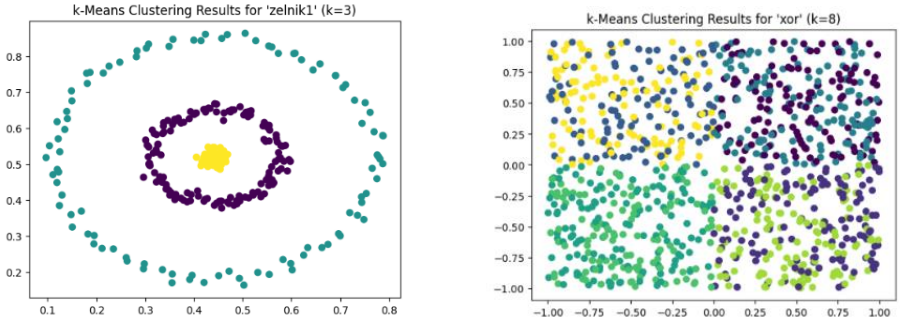


Figure 6: Analyse k-Means pour les datasets zelnik1 (3 clusters) et xor (8 clusters)

2.2 Étude de la méthode de clustering agglomératif

La méthode de clustering agglomératif est plus flexible et généralement plus facile à interpréter que la méthode k-Means. Cependant, elle est sensible aux outliers et peut être particulièrement lente pour des datasets de trop grande dimension.

2.2.1 Intérêts de la méthode de clustering agglomératif

Pour illustrer la méthode du clustering agglomératif, nous avons sélectionné trois datasets (xclara, diamond9 et fourty).

Un intérêt majeur de la méthode agglomérative s'agit de la manière dont elle cherche les clusters. N'ayant aucune connaissance du nombre de clusters au préalable, l'algorithme regroupe tous les points dans un seul cluster. Ce cluster est divisé en sous-clusters qui seront également divisés par la suite. Le regroupement en clusters se fait sur la base de paramètres tel que la manière de calculer la distance entre deux clusters.

Appliquons l'algorithme de clustering agglomératif sur les trois datasets. Pour analyser le résultat, il est possible d'observer ce que l'on appelle un dendrogramme.

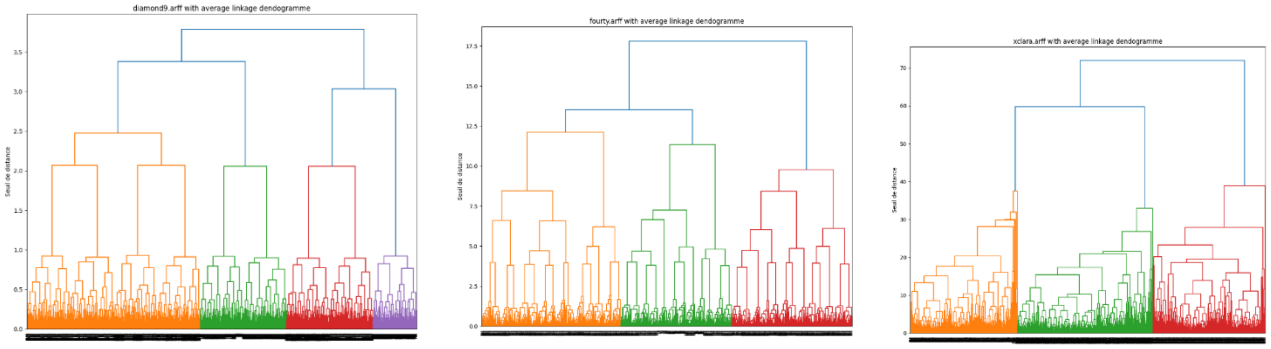


Figure 7 : Dendogrammes pour les datasets diamond9, fourty et xclara

Les dendrogrammes exposent le nombre de clusters le plus probable, lorsque l'on regarde à vue d'œil les clusters identifiés, on peut voir qu'ils ont bien été identifiés. Il est à noter que le clustering reste plutôt subjectif dans quelques cas.

Certains critères ont été développés pour évaluer de manière objective la qualité des clusters. Le critère de Davies-Bouldin, qui, plus il donne un score faible, mieux sont les clusters. C'est à dire que la distance intra clusters est faible mais la distance inter clusters est élevée.

Il existe d'autres métriques tels que le score de silhouette et le score de Calinski-Harabasz. Dans le cas de ces métriques, plus le score est proche de 1 ou plus il est élevé, respectivement, mieux sont les clusters.

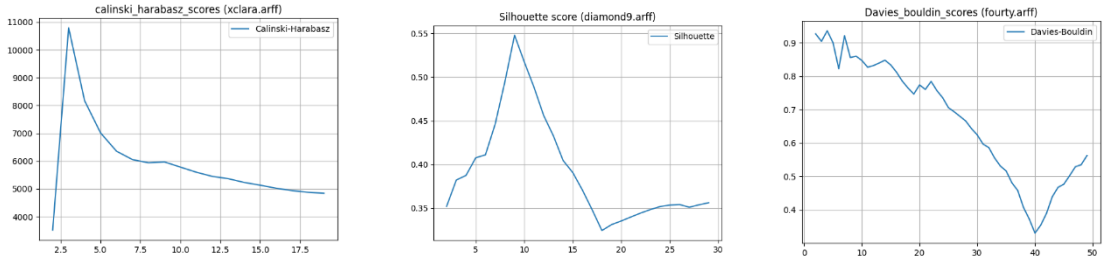


Figure 8 : Evolution des métriques au cours des itérations pour les datasets choisis

Ces scores ne sont pas toujours de parfaits descripteurs des résultats du clustering.

Quant aux différents paramètres réglables de l'algorithme, l'un des plus significatifs est la technique de regroupement des points d'un dataset. Il en existe quatre principales, ward, average, single et complete.

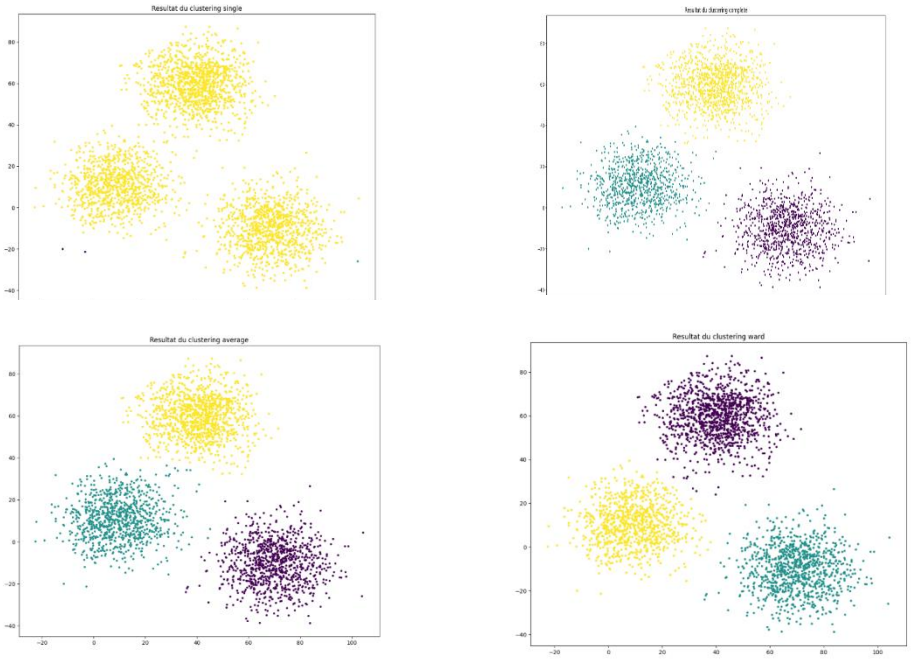


Figure 9 : Résultats des clustering single, complete, average et ward pour le dataset xclara

Ces techniques sont chacune efficaces sur différents types de lots de données. Par exemple ici on peut voir que la méthode single n'est pas adaptée à xclara. Cependant, elle possède des avantages tels qu'un temps de calcul inférieur à toutes les autres méthodes :

```
Seuil de Distance : 1Option de linkage : average
nb clusters = 64 , nb_feuilles =  3000 runtime =  119.16 ms
nb clusters = 1 , nb_feuilles =  3000 runtime =  98.75 ms
nb clusters = 5 , nb_feuilles =  1000 runtime =  14.3 ms
Seuil de Distance : 1Option de linkage : ward
nb clusters = 263 , nb_feuilles =  3000 runtime =  151.63 ms
nb clusters = 9 , nb_feuilles =  3000 runtime =  123.43 ms
nb clusters = 40 , nb_feuilles =  1000 runtime =  14.33 ms
Seuil de Distance : 1Option de linkage : single
nb clusters = 2 , nb_feuilles =  3000 runtime =  33.92 ms
nb clusters = 1 , nb_feuilles =  3000 runtime =  30.04 ms
nb clusters = 1 , nb_feuilles =  1000 runtime =  4.83 ms
```

On peut également voir que la méthode ward semble plus gourmande en temps d'exécution comparée aux autres méthodes. De plus, la méthode single présente des avantages lorsque les clusters ne sont pas convexes. Le clustering à gauche résulte de la méthode single, le clustering à droite résulte de la méthode ward. On peut clairement voir que la méthode single est plus adaptée à ce dataset.

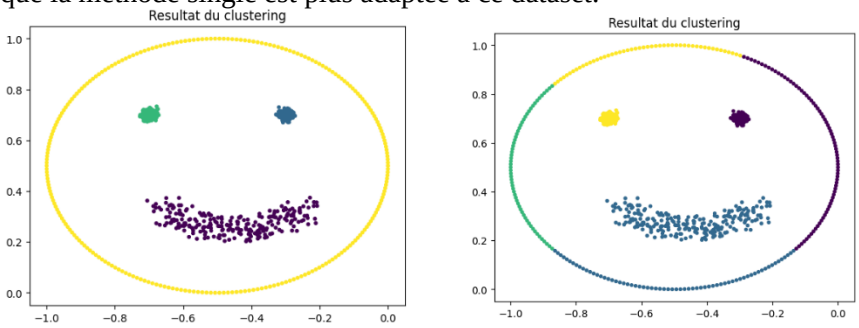


Figure 10 : Analyse agglomérative (single vs ward) pour le dataset smile

Selon nous, le dataset xclara est adapté à cette méthode. Dans un premier temps, nous faisons varier le "distance_threshold" en laissant le paramètre "n_clusters" à none.

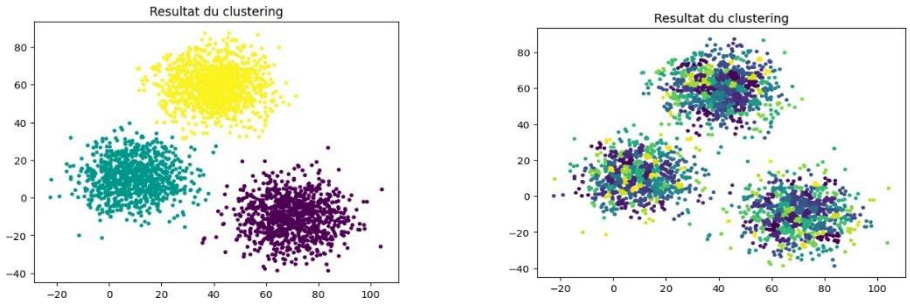


Figure 11 : Analyse agglomérative pour le dataset x_clara, distance_threshold = 100 puis 5 (respectivement)

distance_threshold	nb_clusters	silhouette	runtime
100	3	0.69	151ms
5	375	0.33	150ms

On remarque l'incidence de ce paramètre dans la figure 11 ainsi que dans le tableau ci-dessus. Avec une valeur de distance à 100, la méthode nous permet d'identifier le bon nombre de clusters (à savoir 3 clusters) avec un coefficient de silhouette de 0.69. Avec une distance de 5, le coefficient est plus petit (0.33) et le nombre de clusters incohérent. On peut aussi remarquer que dans les deux cas, la durée de calcul est équivalente.

Nous faisons maintenant varier le paramètre n_clusters en laissant distance_threshold à none.

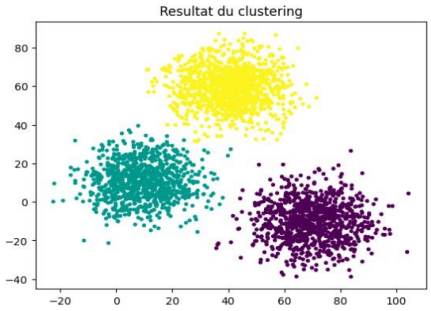


Figure 12: Analyse agglomérative pour le dataset x_clara, n_cluster = 4

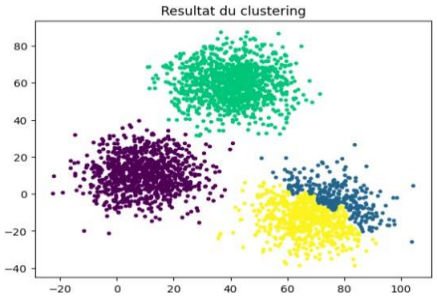


Figure 13 : Analyse agglomérative pour le dataset x_clara, n_cluster = 4

On remarque l'incidence de ce paramètre dans les figures 12 et 13. Avec un nombre de clusters de 3, le résultat est correct comme attendu.

1.1.1 Limites de la méthode de clustering agglomératif

Par exemple pour le dataset xor, qui est dense et bruité, la méthode de clustering agglomératif n'est pas adaptée. Il est difficile d'extraire un score de silhouette car quel que soit les paramètres/réglages utilisés, les scores restent proches de 0.36.

3 Etude et Analyse comparative de méthodes de clustering sur de nouvelles données

3.1 Étude du dataset "x1"

Pour examiner ce dataset, nous employons la méthode k-Means. En effet chaque point se trouve plus près de son propre centre de gravité que de celui des autres points.

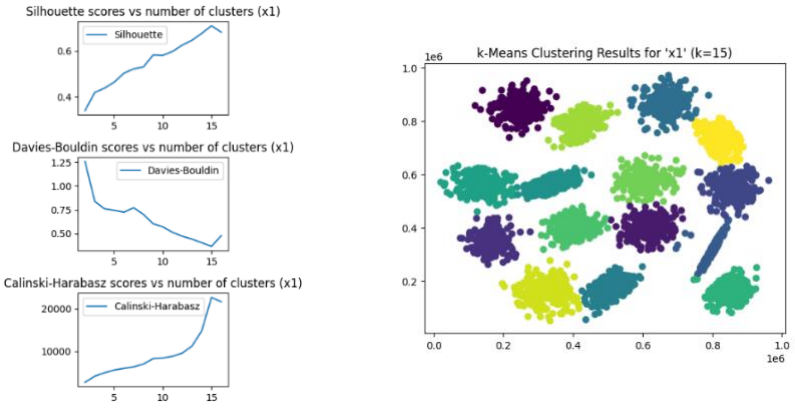


Figure 14: Méthode k-Means pour le dataset x1

Nous identifions donc 15 clusters pour ce dataset. Les courbes des scores appuyent ce résultat.

La méthode agglomérative permet d'identifier les 15 clusters, comme on peut le voir sur le dendrogramme ainsi que visuellement avec les résultats du clustering.

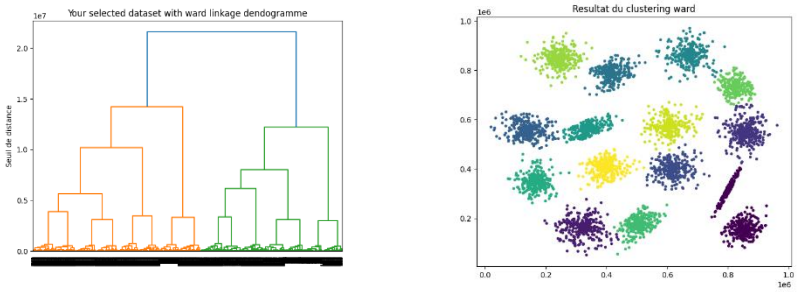


Figure 15 : Résultats méthode agglomerative sur le dataset x1

3.2 Étude du dataset "x2"

Ce dataset se rapproche du dataset x1 mais certains points sont plus éloignés de leur centre de gravité. Nous utilisons donc la méthode agglomérative. Le résultat comporte 15 clusters (voir figure 16).

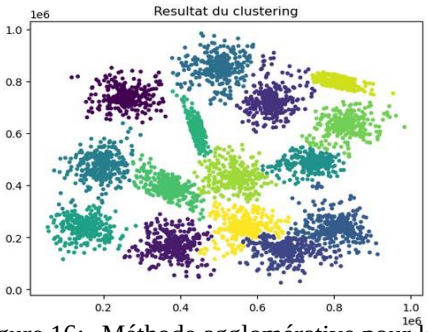


Figure 16: Méthode agglomérative pour le dataset x2

3.3 Étude du dataset "x3"

Nous constatons que ce dataset présente une densité élevée, ce qui le rend peu propice à l'utilisation de la méthode agglomérative. Nous optons donc pour k-Means.

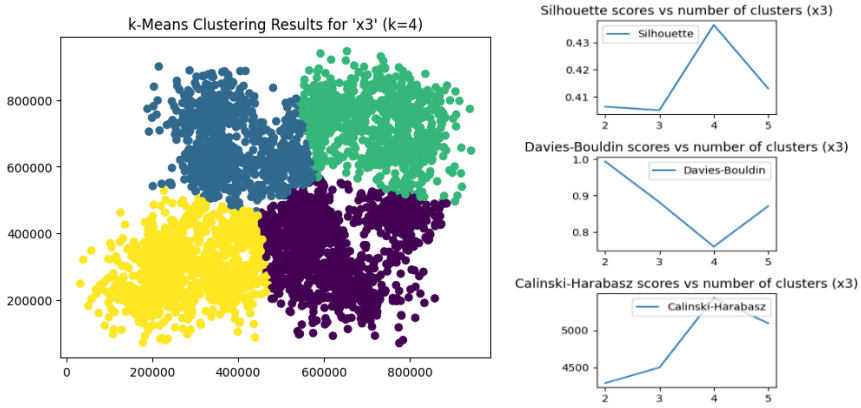


Figure 17: Méthode k-Means pour le dataset x3

3.4 Étude du dataset "x4"

Pour les raisons évoquées précédemment, nous optons également pour l'analyse avec k-Means. Le résultat optimal que nous obtenons est associé à 5 clusters, comme illustré dans la figure 18.

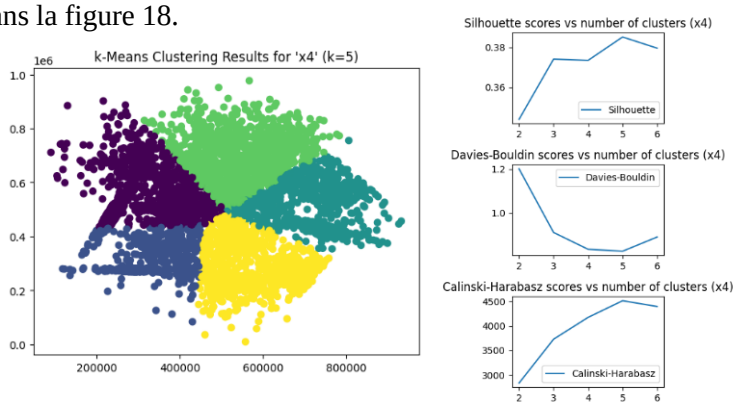


Figure 18: Méthode k-Means pour le dataset x4

3.5 Étude du dataset "y1"

Ce dataset est extrêmement volumineux, ce qui rend complexe son traitement avec les méthodes k-Means ou agglomérative en raison des temps de calcul considérables. Par conséquent, nous procédons à une comparaison entre les deux approches. En utilisant k-Means, nous parvenons à identifier 4 clusters, mais le temps de calcul s'avère significativement élevé (environ 10 minutes).

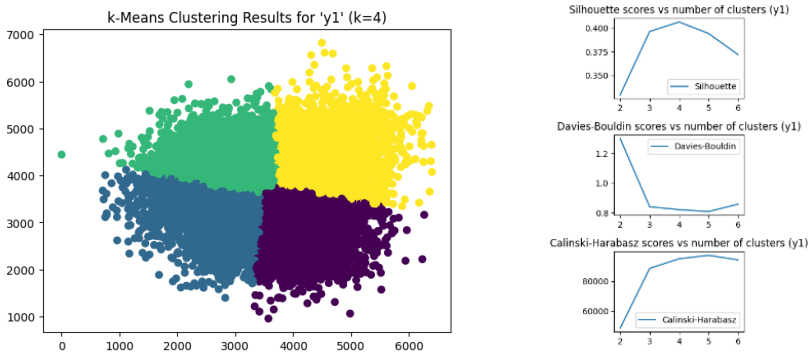


Figure 19: Méthode k-Means pour le dataset y1

La méthode agglomérative est très inadaptée à des datasets de cette taille. Lorsque l'on tente d'effectuer le clustering, l'erreur suivante s'affiche :

```
numpy.core._exceptions._ArrayMemoryError: Unable to allocate 41.5 GiB for an array with shape (5575627200,) and data type float64
```

3.6 Étude du dataset "zz1"

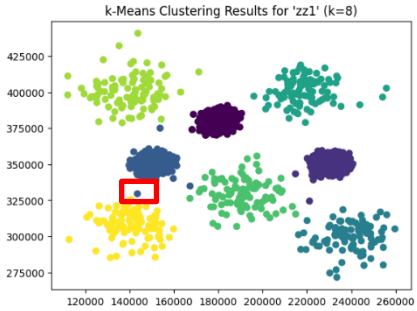


Figure 20: Méthode k-Means pour le dataset zz1

La méthode k-Means semble inappropriée pour ce dataset car certains points se trouvent plus près du centre d'un autre cluster que de leur propre centre de gravité. Cette observation est visible sur la figure 20, où l'on remarque que certains points appartenant à un cluster donné sont assignés à un autre cluster.

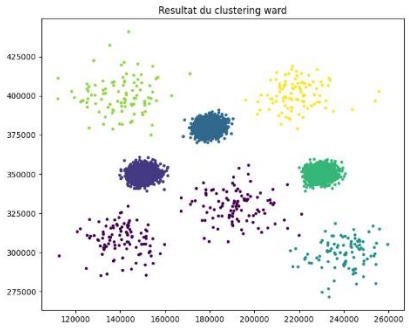


Figure 21: Méthode agglomérative pour le dataset zz1 (average)

Puisque la densité des clusters est variable, nous utilisons finalement la méthode agglomérative pour identifier le nombre de clusters.

La figure 21 montre donc que cette méthode nous permet d'obtenir 8 clusters.

3.7 Étude du dataset "zz2"

Les clusters semblent être isolés, et le dataset ne présente pas un niveau de bruit trop élevé. Nous optons pour l'utilisation de la méthode k-Means pour l'analyse.

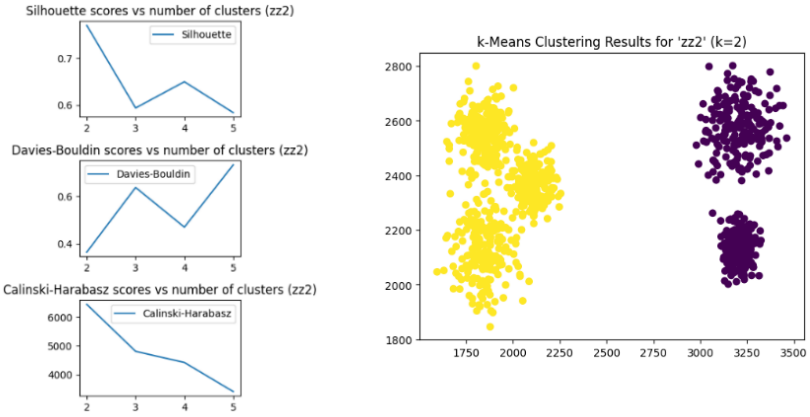


Figure 22: Méthode k-Means pour le dataset zz2

La méthode agglomérative aboutit au même clustering lorsque l'on analyse le score de silhouette. Cependant, il est clair que cinq clusters sont identifiables également, ce qui témoigne de la subjectivité du clustering.

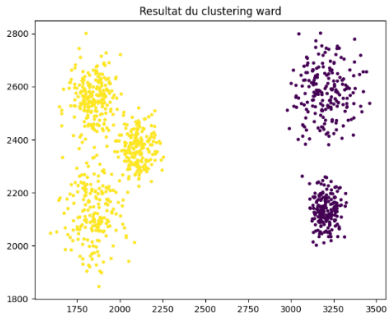


Figure 23 : Résultat avec la méthode agglomérative (ward)

4 Conclusion

Pendant cette séance de TP, nous avons exploré et examiné diverses approches de clustering, notamment k-Means et agglomératif. Comme il a été observé, chaque méthode présente ses propres avantages, inconvénients et limitations.

La méthode k-Means est relativement facile à mettre en œuvre, expliquant ainsi sa popularité. Cependant, elle se montre particulièrement réactive aux perturbations et aux valeurs aberrantes. De plus, elle nécessite que les clusters aient une forme isotropique et convexe, et que chaque point soit plus proche de son centre de gravité que des autres.

En ce qui concerne la méthode agglomérative, elle ne requiert pas la pré-spécification d'un nombre de clusters (contrairement à k-Means). Cependant, elle est également sensible au bruit et n'est pas appropriée pour des ensembles de données de grande dimension en raison de son temps de calcul trop long.