INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Clustering

Machine Learning

Département Génie Electrique et Informatique
INSA Toulouse
11 Janvier, 2023

Samantha Gorrin
Minh Hieu Bui
Alexis Pierre Dit Lambert

Contents

1 Introduction

Dans le cadre de l'apprentissage non-supervisé, le clustering consiste à regrouper des données brutes non labelisées, sur la base de leurs similarités et de leurs différences. Il existe différentes méthodes de clustering telles que la méthode K-means, le clustering agglomératif ou encore le clustering DBSCAN. Dans ce rapport, nous allons mettre en oeuvre ces différentes méthodes sur des jeux de données en 2 dimensions seulement. Nous comparerons ces différentes méthodes afin d'identifier leurs avantages et leurs limites respectifs.

Le code python est disponible sur le dépôt git suivant : https://github.com/Alexix24/Machine-Learning

2 Comparaison des différentes méthodes de clustering

2.1 Étude de la méthode k-Means

2.1.1 Intérêts de la méthode k-Means

Afin d'étudier les avantages de la méthodes k-means, nous avons choisi trois datasets bien adaptés à cette méthode, c'est-à-dire avec des clusters facilement identifiables (les points sont plus proches de leur propre centre de gravité que des autres). Nous avons donc choisi le dataset twodiamonds.arff pour observer les intérêts de la méthode.

Nous avons cherché à identifier automatiquement le nombre de clusters, en utilisant le coefficient de silhouette.

```
For n_clusters = 2 The average silhouette_score is : 0.6305972114000131  , runtime =  30.31 ms
For n_clusters = 3 The average silhouette_score is : 0.47158111103878236  , runtime =  33.92 ms
For n_clusters = 4 The average silhouette_score is : 0.36064934833689605  , runtime =  44.28 ms
For n_clusters = 6 The average silhouette_score is : 0.38943688932529885  , runtime =  37.07 ms
For n_clusters = 8 The average silhouette_score is : 0.4046449242765549  , runtime =  40.3 ms
```

Figure 1: Coefficients de silhouette et temps d'exécution pour le dataset "twodiamonds"

Le coefficient de silhouette le plus grand correspond à deux clusters (voir figure 1). En effet, nous pouvons facilement identifier visuellement deux clusters. Les résultats de l'analyse pour 2 clusters et 3 clusters sont disponibles en figure 2 et 3 respectivement.

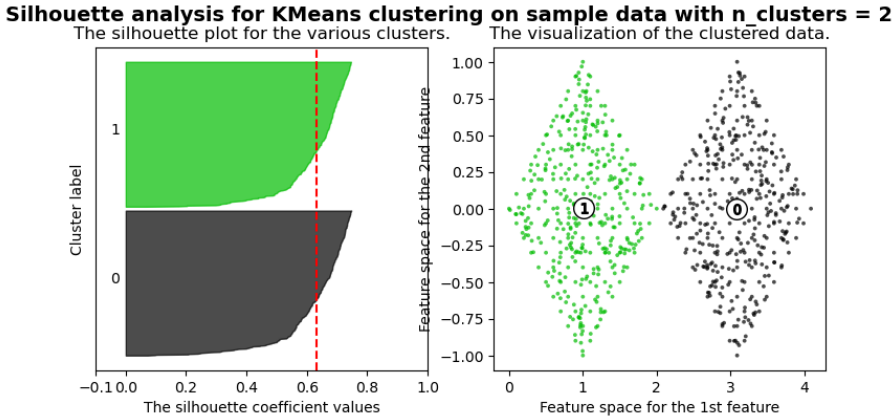

Figure 2: Analyse K-Means pour le dataset "twodiamonds", 2 clusters

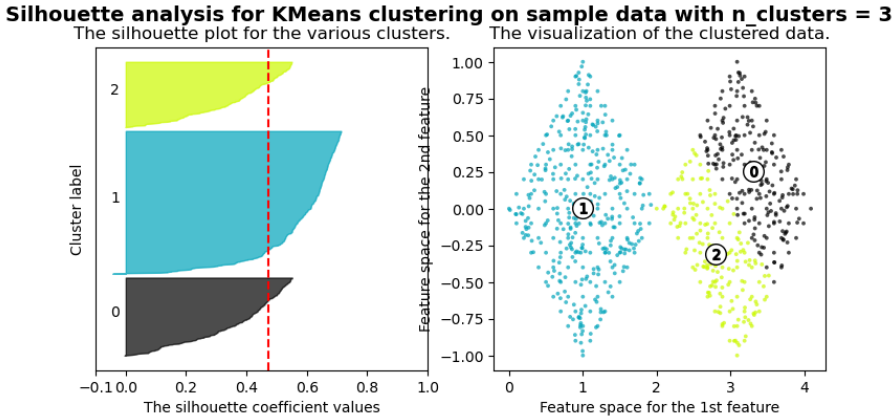

Figure 3: Analyse K-Means pour le dataset "twodiamonds", 3 clusters

La méthode K-Means avec l'analyse du coefficient de silhouette fonctionne très bien pour ce dataset. En effet, cette méthode est adaptée pour des clusters isotropic et de forme convexe.

2.1.2 Limites de la méthode k-Means

Pour observer les limites de la méthode k-means, nous avons choisi le dataset "xor". Quelque soit le nombre de cluster choisi, le coefficient de silhouette reste très faible et ne permet pas d'identifier un nombre de cluster adéquat (voir figure 4)

```
For n_clusters = 2 The average silhouette_score is : 0.3921542107435631  , runtime =  44.64 ms
For n_clusters = 8 The average silhouette_score is : 0.5665265467956482  , runtime =  60.14 ms
For n_clusters = 10 The average silhouette_score is : 0.5991501544091691  , runtime =  45.98 ms
For n_clusters = 12 The average silhouette_score is : 0.6407993194520872  , runtime =  57.06 ms
For n_clusters = 14 The average silhouette_score is : 0.6898844415289176  , runtime =  108.18 ms
For n_clusters = 15 The average silhouette_score is : 0.711278614093076  , runtime =  78.15 ms
For n_clusters = 18 The average silhouette_score is : 0.6280609954603872  , runtime =  66.66 ms
```

Figure 4: Coefficients de silhouette et temps d'exécution pour le dataset "xor"

Nous pouvons effectivement voir sur la figure 5 que la méthode k-means ne permet pas d'identifier les clusters pour ce dataset.

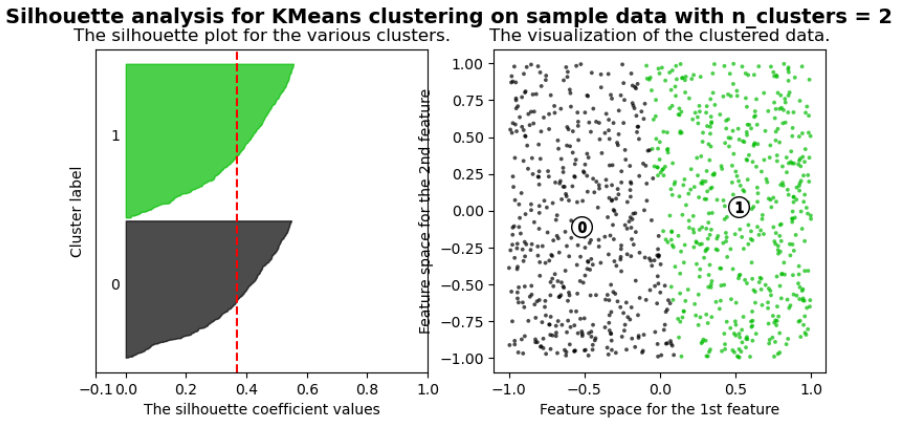

Figure 5: Analyse K-Means pour le dataset "xor", 2 clusters

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

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

Figure 6: Analyse agglomérative pour le dataset "x_clara", distance_threshold = 100

Figure 7: Analyse agglomérative pour le dataset "x_clara", distance_threshold = 5

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

On remarque l'incidence de ce paramètre dans les figures 6 et 7 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_threshlod à none.

Figure 8: Analyse agglomérative pour le dataset "x_clara", n_cluster = 3

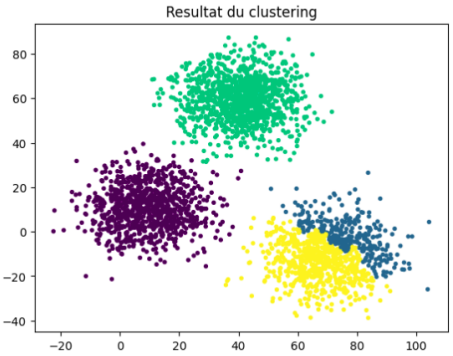

Figure 9: Analyse agglomérative pour le dataset "x_clara", n_cluster = 4

On remarque l'incidence de ce paramètre dans les figures 8 et 9. Avec un nombre de clusters de 3, le résultat est correct comme attendu.

2.2.2 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 adapatée. Il est difficile d'extraire un score de silhouette car quelque soit les paramètres/réglages utilisés, les scores restent proches de 0.36 (voir figures 10 et 11).

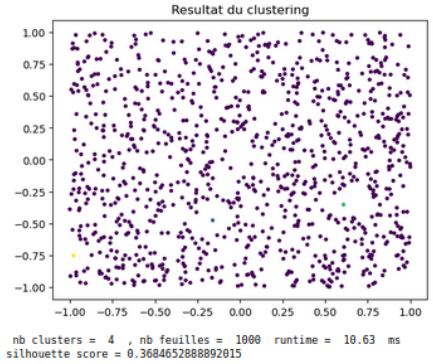

Figure 10: Analyse du dataset "xor" avec la méthode agglomérative, n_clusters = 4, single

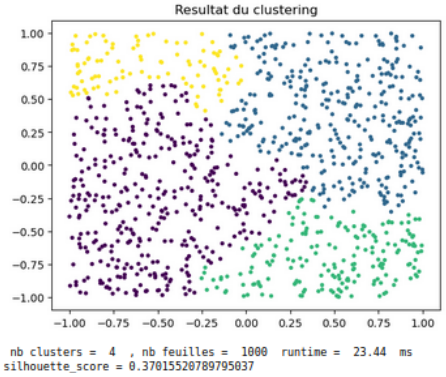

Figure 11: Analyse du dataset "xor" avec la méthode agglomérative, n_clusters = 4, complete

2.3 Étude de la méthode DBSCAN

La méthode DBSCAN est adaptée pour des clusters de toutes formes (convexes, non convexes ...). Elle est aussi robuste aux outliers, c'est-à-dire aux points représentants du "bruit". Cependant, cette méthode est très sensible lorsque le dataset est composé de clusters de densité différentes et le choix des paramètres peut s'avérer difficile.

2.3.1 Intérêts de la méthode DBSCAN

Selon nous, le dataset "smiley" est adapté à cette méthode car les clusters sont de même densité. On obtient donc 4 clusters, avec un coefficient de silhouette de 0.38 (voir figure 12).

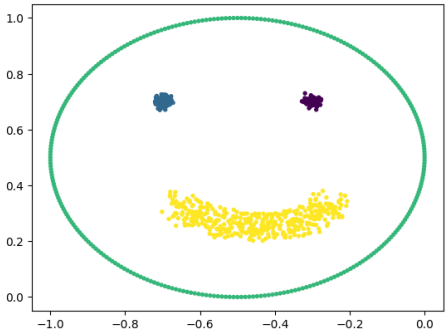

Figure 12: Analyse DBSCAN pour le dataset "smiley", eps = 0.074, min_samples = 1

2.3.2 Limites de la méthode DBSCAN

Pour observer les limites de DBSCAN, nous choisissons le dataset "x_clara" car il comporte des densités variables.

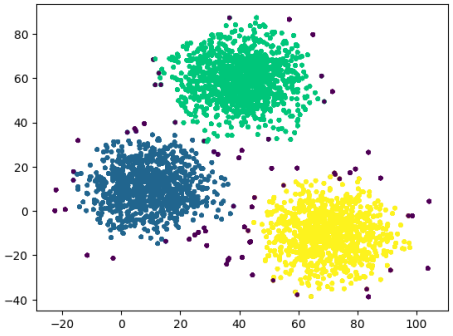

Figure 13: Analyse du dataset "x_clara" avec la méthode DBSCAN

cas1: eps = 5.215539166134209	min_samples = 9	▼ The average silhouette_score is : 0.6514758717956664	runtime = 187.65 ms
cas1: eps = 5.669760966211139	min_samples = 7	The average silhouette_score is : 0.6625391680695722	runtime = 266.35 ms
cas1: eps = 5.855520936884376	min_samples = 10	The average silhouette_score is : 0.6591334218747488	runtime = 266.26 ms
cas1: eps = 5.84764820770622	min_samples = 10	The average silhouette_score is : 0.6570588622914377	runtime = 295.81 ms
cas1: eps = 5.611369278982272	min_samples = 8	The average silhouette_score is : 0.6553807968608947	runtime = 260.21 ms

Figure 14: Analyse du dataset "x_clara" avec la méthode DBSCAN

Comme attendu (voir figures 13 et 14), le coefficient de silhouette reste inchangé quelques soit les paramètres eps et min_samples. Nous ne parvenons donc pas à identifier les clusters sur ce dataset avec la méthode DBSCAN. Plus précisément, les zones de faible densité sont mal gérées par rapport aux zones très denses.

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

3.1 Étude du dataset "x1"

Pour analyser ce dataset, nous utilisons la méthode K-means. En effet chaque point est plus proche de son propre centre de gravité que des autres.

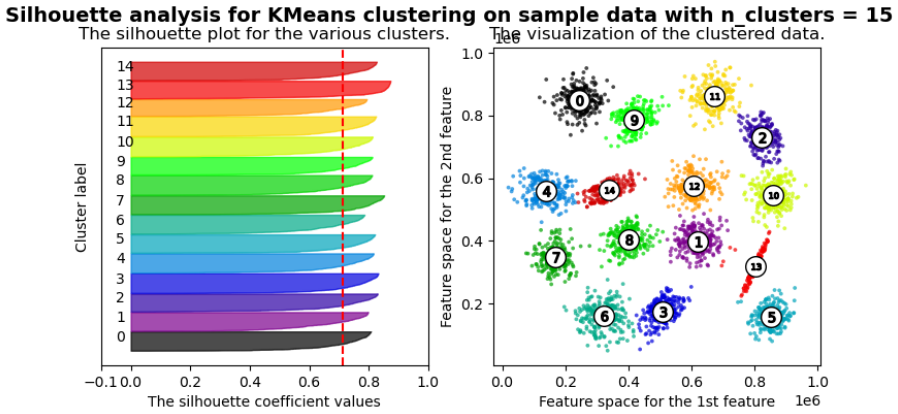

Figure 15: Méthode k-means pour le dataset "x1"

Nous parvenons donc à identifier 15 clusters pour ce dataset en constatant que le silhouette score pour un n_clusters de 15 est le meilleur (0.71).

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).

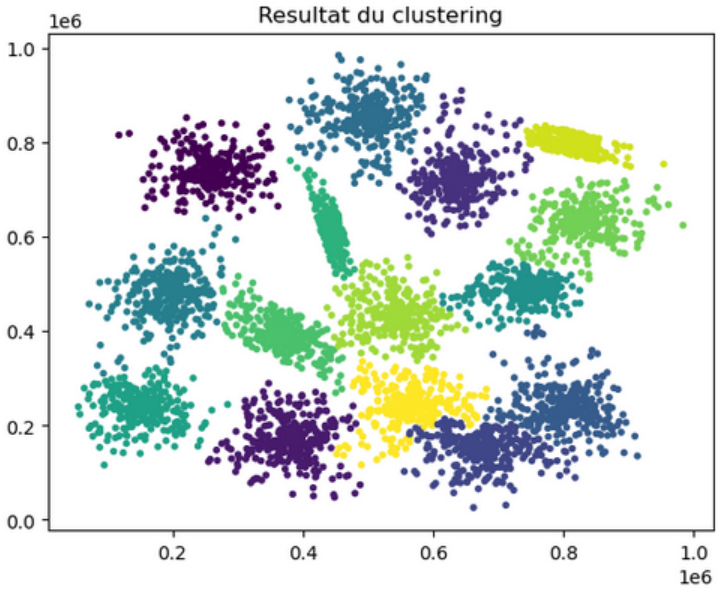

Figure 16: Méthode agglomérative pour le dataset "x2"

3.3 Étude du dataset "x3"

Nous observons que ce dataset est très dense donc pas adapté à la méthode agglomérative. Le réglage de la méthode DBSCAN pour ce dataset est particulièrement sensible et ne permet pas d'atteindre un score de silhouette intéressant. Nous effectuons donc l'analyse avec l'algorithme k-means. Le meilleur résultat que nous obtenons correspond à 15 clusters avec un score de 0.48 (voir figure 17).

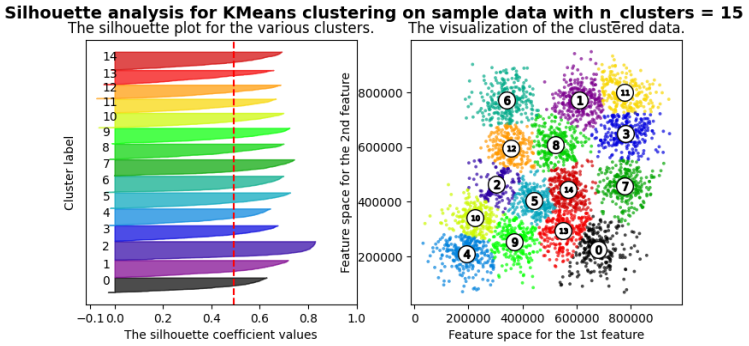

Figure 17: Méthode k-means pour le dataset "x3"

3.4 Étude du dataset "x4"

Pour les mêmes raisons que le cas précédent, nous effectuons l'analyse avec l'algorithme k-means. Le meilleur résultat que nous obtenons correspond à 15 clusters avec un score de 0.48 (voir figure 18).

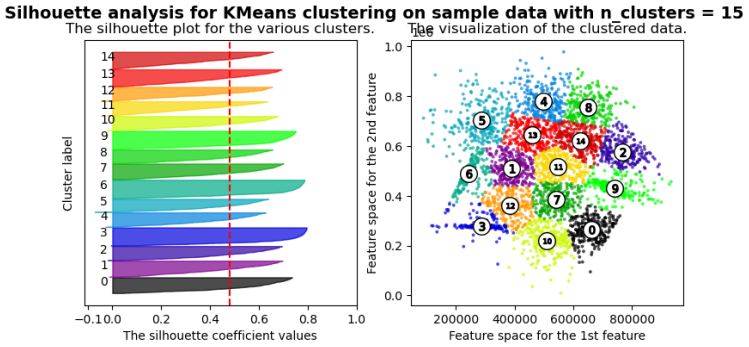

Figure 18: Méthode k-means pour le dataset "x4"

3.5 Étude du dataset "y1"

Ce dataset comporte énormément de données. Il est donc difficile de le traiter avec les méthodes k-means ou agglomérative car cela nécessite un temps de calcul trop important. Le réglage des paramètres DBSCAN est également très difficile pour ce cas. La figure 19 illustre le résultat obtenu avec k-means pour 2 clusters. Le temps de calcul étant déjà très long (10000ms).

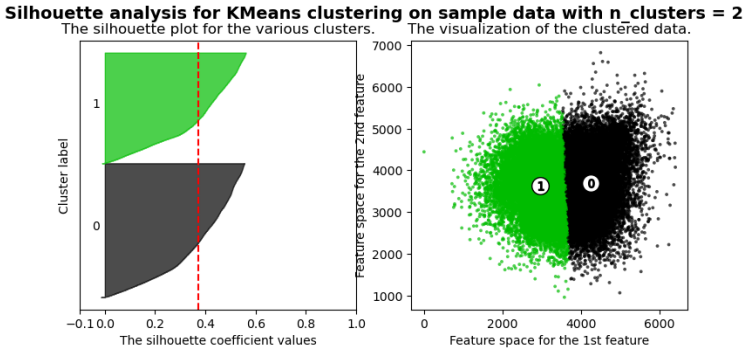

Figure 19: Méthode k-means pour le dataset "y1"

3.6 Étude du dataset "zz1"

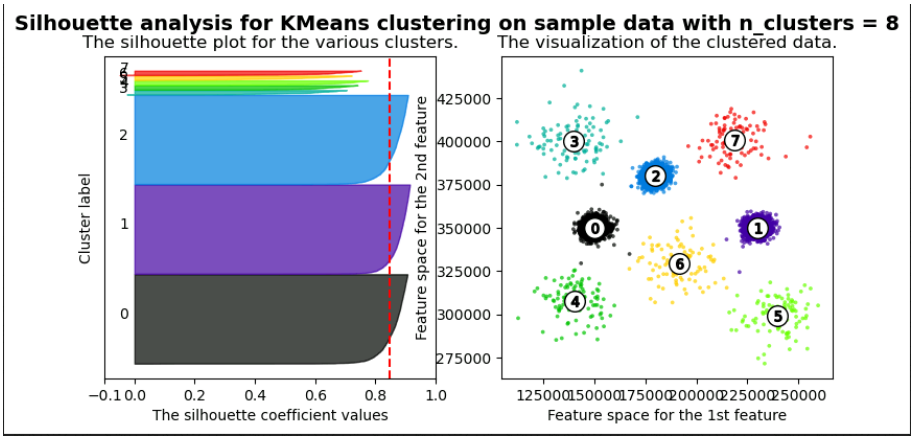

Figure 20: Méthode k-means pour le dataset "zz1"

La méthode K-means ne semble pas adaptée a ce dataset car certains points sont plus proches du centre d'un autre cluster que de leur centre de gravité. Cela est visible sur la figure 20 dans laquelle on peut voir que certains points appartenant au cluster 1 sont attribués au cluster 3.

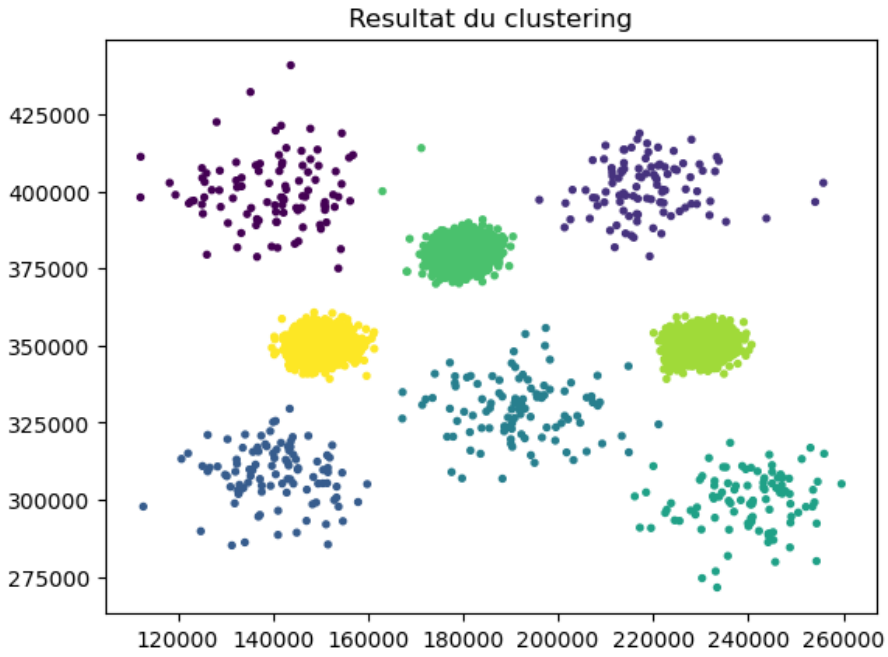

Figure 21: Méthode agglomérative pour le dataset "zz1", average

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 sont assez isolés et le dataset n'est pas trop bruité. Nous utilisons la méthode k-means pour l'analyse.

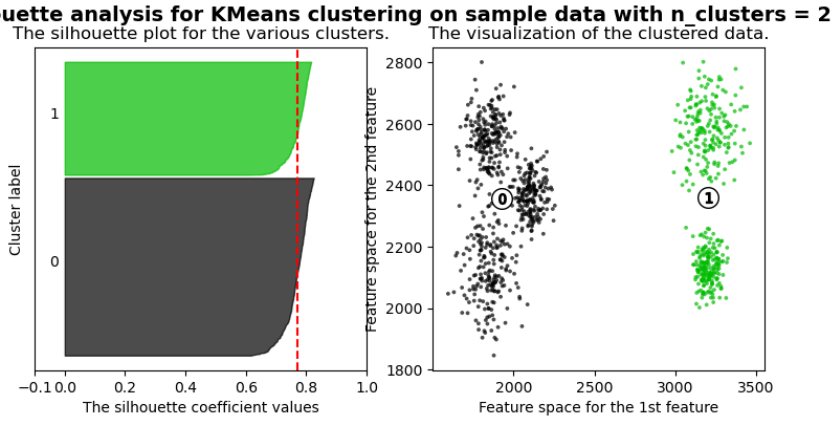

Figure 22: Méthode k-means pour le dataset "zz2"

Le meilleur coefficient de silhouette (0.77) correspond à deux clusters (voir figure 22).

4 Conclusion

Au cours de ce TP, nous avons utilisé et étudié différentes méthodes de clustering : k-means, agglomérative et DBSCAN. Comme nous l'avons vu, chaque méthode a ses avantages, ses inconvéniants et ses limites.

La méthode k-mean est assez facile à utiliser, ce qui explique sa popularité. Cependant, elle est particulièrement sensible au bruit et aux outliers. Il faut également que les clusters soit de forme isotropique et de forme convexe, et que chaque point soit plus proche de son centre gravité que des autres.

Concernant la méthode agglomérative, elle ne nécessite pas de paramétrer à l'avance un nombre de clusters (comme c'est le cas pour k-means). Cependant, elle est sensible au bruit et n'est pas adaptée pour des datasets de grande dimensions (temps de calcul trop long).

Pour finir, la méthode DBSCAN est particulièrement efficace et permet d'analyser des clusters de différentes formes (convexes ou non ...). Elle est aussi robuste pour des datasets bruités, même si elle rencontre des difficultés lorsque les densités ne sont pas uniformes. En revanche, cette méthode est plus difficile à paramétrer, ce qui est son principal inconvénient.