

**NOM DU DOSSIER**

**Sous titre**

Date

Service

**RAPPORT DE TP**

**Oriane BERRY, Clémence LEMEILLEUR**

**Promo 56, Année 2022/2023 – 5SDBD-B1**

*« TP Apprentissage Non Supervisé »*

S1 2023

Encadrant : M.Siala

**RAPPORT DE TP**

Oriane Berry, Clémence LEMEILLEUR

Promo 56, Année 2022/2023 – 5SDBD-B1

*“TP Apprentissage Non Supervisé*”

S1 2023

Encadrant: M.SIALA

Table des matières

[Introduction 1](#_Toc116455969)

[I- Les points forts et faibles identifiés pour les différentes méthodes de Clustering 2](#_Toc116455970)

[1. Clustering k-Means et k- Medoids 2](#_Toc116455971)

[2. Clustering agglomératif 3](#_Toc116455972)

[3. Clustering DBSCAN et HDBSCAN 4](#_Toc116455973)

[4. Bilan 4](#_Toc116455974)

[II- Étude et comparaison de méthodes de clustering sur de nouvelles données fournies 4](#_Toc116455975)

[Conclusion 5](#_Toc116455976)

[Table des annexes 6](#_Toc116455977)

# *Introduction*

Dans ce TP nous avons comparé les différents algorithmes de clustering avec plusieurs méthodes qu’elles soient fournies par des outils ou externes.

En effet chaque méthode comporte des avantages, des inconvénients et fonctionne de manière plus ou moins efficace en fonction de l’aspect du jeu de données que nous voulons traiter.

Afin d’observer les effets de ces méthodes de clustering, nous utilisons des jeux de données en 2 dimensions afin d’avoir une meilleure visualisation de ces dernières. Nous finirons par les tester sur un dataset mystère afin de vérifier nos affirmations.

Nous avons travaillé sous Jupiter Notebook et héberger notre travail sur Github dont voici le lien : <https://github.com/Enario4/ApprentissageNonSupervise>

# Les points forts et faibles identifiés pour les différentes méthodes de Clustering

## Clustering k-Means et k- Medoids

**Intérêts de la méthode k-Means**

***Les métriques d’évaluation recommandées :***

* **Coefficient de silhouette :** Différence entre la distance moyenne avec les points du même groupe que lui (cohésion) et la distance moyenne avec les points des autres groupes voisins (séparation). Le coefficient de silhouette varie entre -1 (pire classification) et 1 (meilleure classification).

**🡪** Est plus adapté aux données « regroupées » de manière identifiable.

* **Indice de Davies-Bouldin :** C'est la moyenne du rapport maximal entre la distance d'un point au centre de son groupe et la distance entre deux centres de groupes. Il varie entre 0 (meilleure classification) et +∞ (pire classification).

**🡪** Est plus adapté aux données reparties de manière homogène.

* **Indice de Calinski-Harabasz :** C'est le rapport entre la variance inter-groupes et la variance intra-groupe. Il varie entre 0 (pire classification) et +∞ (meilleure classification)
* **FasterPAM :** Permet d’évaluer les pertes que l’on a en coupant autour des médoïdes. Plus l’indice est petit mieux c’est.
* **Rand\_Score** : C’est une mesure de similarité entre deux partitions d'un ensemble. Sa valeur est comprise entre 0 et 1, 0 indiquant que les deux regroupements de données ne concordent sur aucune paire de points et 1 indiquant que les regroupements de données sont exactement les mêmes.
* **Mutual\_information**: Quantité qui mesure la dépendance entre 2 variables. Une information mutuelle élevée indique une forte dépendance ; et une information mutuelle nulle entre deux variables aléatoires signifie que les variables sont indépendantes.

Dans notre cas pour cette méthode, nous avons choisi la deuxième méthode. Après avoir remarqué que l’indice de DB n’est pas adapté à ce jeu de données. Nous avons testé sur le silhouette.

Nous avons ensuite choisi nos jeux de données avec ceux qui nous semblaient avoir des données bien séparées qui permette l’identification aisée des clusters par la méthode k-Means.

Lors de l’application itérative de la méthode des k-Means sur 2 jeux de données, voici les résultats que nous avons eu au niveau des métriques d’évaluation :

**JDD « aggregation.arff»**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 | k=9 | k=10 |
| Indice de DB | **0,8964** | 0,6669 | 0,6116 | 0,7042 | 0,6400 | 0,7368 | 0,8270 | 0,7760 | 0,7252 |
| Indice silhouette | 0,4558 | 0,5234 | 0,5236 | 0,4987 | **0,6509** | 0,4804 | 0,4518 | 0,4621 | 0,4793 |
| Temps de calcul | 27,76 ms |  |  |  | 37,31 ms |  |  |  |  |
| Nb iter | 14 |  |  |  | 5 |  |  |  |  |

Nous allons faire de même avec un jeu de données du même aspect

**JDD « atom.arff»**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 | k=9 | k=10 |
| Indice de DB | **1,0942** | 0,9038 | 0,7815 | 0,6566 | 0,6398 | 0,6553 | 0,6978 | 0,0722 | 0,7653 |
| Indice de silhouette | 0,4888 | 0,5358 | 0,5945 | 0,6352 | 0,6464 | **0,6499** | 0,6440 | 0,6372 | 0,5625 |
| Temps de calcul | 23,68ms |  |  |  |  | 39,46 ms |  |  |  |
| Nb iter | 4 |  |  |  |  | 6 |  |  |  |

**Limites de la méthode k-Means**

Pour mettre en avant les limites de cette méthode nous avons pris un autre jeu de données qui est complexe à clusteriser (forme de banane).

**JDD « banana.arff»**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 | k=9 | k=10 |
| Indice de DB | 0,8930 | 0,8518 | 0,8083 | 0,6698 | 0,6567 | 0,6223 | 0,6141 | 0,5999 | 0,5931 |
| Indice de silhouette | 0,4645 | 0,4544 | 0,4623 | 0,4761 | 0,5054 | 0,5013 | 0,5218 | 0,5214 | 0,5201 |
| Temps de calcul | 34,33 | 49,57 | 48,83 | 58,49 | 52,26 | 60,69 | 88,74 | 72,34 | 71,49 |
| Nb iter | 7 | 5 | 17 | 14 | 12 | 8 | 15 | 11 | 12 |

**JDD «birch-rg1.arff»**

On remarque que l’exécution de ce code prend énormément de temps, et nous donne un résultat non satisfaisant. En effet il s’agit juste d’un amas de données coupées de manière équitables aléatoirement.

On remarque donc que la méthode k-Means comporte ses limites sur certaines formes de jeux de données, notamment celles dont les données sont très proches et il est difficiles de les séparer en plusieurs cluster.

**Méthode k-Medoids**

**JDD « aggregation.arff»**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 | k=9 | k=10 |
| FasterPAM | 6672 | **4692** | 3827 | 3289 | 2929 | 2724 | 2556 | 2361 | 2233 |
| Indice de DB | 0,9208 | 0,7233 | 0,7429 | 0,8396 | 0,7446 | 0,7883 | 0,8570 | 0,8122 | 0,9022 |
| Indice de silhouette | 0,4546 | **0,5144** | 0,4930 | 0,4564 | 0,4828 | 0,4759 | 0,4260 | 0,4457 | 0,4028 |
| Rand\_score |  | **0,9832** |  |  |  |  |  |  |  |
| Mutual information |  | **1,0102** |  |  |  |  |  |  |  |
| Temps de calcul (ms) | 17,01 | 8,07 | 8,75 | 10,62 | 10,79 | 12,27 | 8,89 | 16,66 | 13,65 |
| Nb iter | 3 | 3 | 3 | 4 | 4 | 5 | 3 | 8 | 6 |

On remarque que pour k=3, on a un des pertes FasterPam pas trop élevées et un bon indice de silhouette. Le nombre de clusters le plus pertinent à choisir semble donc être 3.

De plus, on remarque qu’on obtient un rand score proche de 1 ce qui nous permet de déduire que les 2 méthodes, **k-Means et k-Medoïds**, clusterisent quasiment de la même façon ce jeu de données.

En revanche, l’indice d’information mutuelle est assez bas donc données sont très indépendantes et l’incertitude sur le résultat est d’autant plus grande.

Pour finir nous allons tester la méthode des k-medoids en utilisant la distance de Manhattan plutôt que la distance Euclidienne pour établir les clusters.

**JDD « aggregation.arff»**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 | k=9 | k=10 |
| FasterPAM | 8305 | 5781 | 5138 | 4179 | 3817 | 3436 | 3183 | 3045 | 2770 |
| Indice de DB | 1,0240 | 0,7427 | 0,8739 | 0,7952 | 0,8630 | 0,7889 | 0,8457 | 0,8717 | 0,7663 |
| Indice de silhouette | 0,4120 | 0,4922 | 0,4116 | 0,4770 | 0,4488 | 0,4795 | 0,4415 | 0,4309 | 0,4681 |
| Temps de calcul (ms) | **5,38** | **5,26** | **5,64** | **6,46** | **5,31** | **7,52** | **9,28** | **5,18** | **6,08** |
| Nb iter | 3 | 3 | 4 | 4 | 5 | 4 | 5 | 4 | 5 |

On remarque plus de perte sur la distance de Manhattan, mais un meilleur indice de DB et surtout un temps de calcul bien plus avantageux.

## Clustering agglomératif

**Intérêts de la méthode**

**Limites de la méthode**

**Comparaison de méthodes de clustering**

## Clustering DBSCAN et HDBSCAN

**Intérêts de la méthode DBSCAN**

**Limites de la méthode DBSCAN**

**Comparaison avec la méthode HDBSCAN**

## Bilan

Après avoir étudié toutes les méthodes, nous allons les comparer sous plusieurs critères afin de mieux faire ressortir leurs avantages et inconvénients de chacun en fonction des cas d’utilisation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **k-Means** | **k-medoids** | **Clustering agglomératif** | **DBSCAN** | **HDBSCAN** |
| **Qualité des solutions obtenues** |  |  |  |  |  |
| **Performances des méthodes** |  |  |  |  |  |
| **Type de Dataset adapté** |  |  |  |  |  |

# Étude et comparaison de méthodes de clustering sur de nouvelles données fournies

Après avoir étudier les différentes méthodes, leurs points forts et points faibles respectifs, nous allons les tester sur un nouveau dataset : le dataset mystère et observer leur comportement.

Cela va nous permettre de voir si les caractéristiques mises en avant précédemment sont vérifiées ou démenties.

# *Conclusion*

Pour conclure nous pouvons dire que ce TP nous a permis de prendre connaissance et de nous familiariser avec différentes méthodes de clustering et de visualiser leurs effets sur des jeux de données en 2 dimensions.

Nous avons pu expérimenter et mieux comprendre les principes de base du clustering de manière globale dans un premier temps, puis plus approfondis via différentes méthodes en comparant leurs avantages et inconvénients.

Cela nous permet donc de nous mettre une fois de plus dans le rôle de l’ingénieur qui est d’utiliser ses connaissances et de les appliquer à des cas réels. D’autant plus que le clustering peut être appliqué à énormément de cas d’usages dans de nombreux domaines et, selon nous, est chaque jour un peu plus dans l’actualité.

# Table des annexes

1. Annexe 1 : Lien GitHubA

**Annexe 1 :** Lien GitHub : <https://github.com/Enario4/ApprentissageNonSupervise>

