

**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

[I- Le Clustering 1](#_Toc115792236)

[1. Clustering k-Means et k- Medoids 1](#_Toc115792237)

[Table des annexes 2](#_Toc115792238)

***Introduction :***

# Dans ce TP nous avons comparé les différents algorithmes de clustering avec plusieurs méthodes qu’elles soient fournies par des outils ou externes. Nous utilisons pour cela des jeux de données en 2 dimensions afin d’avoir une meilleure visualisation de ces dernières.

# Le Clustering

## Clustering k-Means et k- Medoids

Intérêts de la méthode k-Means

Les métriques d’évaluation recommandés :

* 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)

Nous avons choisi la deuxième méthode.

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 :

**Paramètre : 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 |  |  |  |  |  |  |  |  |
| Nb iter | 14 |  |  |  |  |  |  |  |  |

Après avoir remarqué que l’indice de DB n’est pas adapté à ce jeu de données. Nous avons tester sur le silhouette.

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

**Paramètre : 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 |  |  |  |  |  | 6 |  |  |  |

Limites de la méthode k-Means

**Paramètre : 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 (ms) | 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 |

**Paramètre: 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 forme de jeux de données.

Méthode k-Medoids

**Paramètre : JDD « banana.arff»**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 | k=9 | k=10 |
| FasterPAM | 937,0 | 726,8 | 583,0 | 490,7 | 417,2 | 370,9 | 320,5 | 292,3 | 270,8 |
| Indice de DB | 0,9027 | 0,9859 | 09090 | 0,8088 | 0,7166 | 0,6597 | 0,6321 | 0,6378 | 0,6162 |
| Indice de silhouette | 0,4326 | 0,4116 | 0,4398 | 0,4413 | 0,4902 | 0,4803 | 0,5231 | 0,5144 | 0,5162 |
| Temps de calcul (ms) | 160,49 | 205,24 | 198,31 | 194,74 | 214,26 | 183,49 | 288,33 | 202,52 | 181,59 |
| Nb iter | 2 | 2 | 2 | 2 | 2 | 3 | 4 | 2 | 3 |

## Clustering agglomératif

Intérêts de la méthode

Limites de la méthode

Comparaison de méthodes de clustering

***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 GitHub A

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

