République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumédiène

Faculté d'Informatique
Département d'Intelligence Artificielle

Master 2 Systèmes Informatiques intelligents

Module : Data Mining

Rapport de projet, partie 5
Clustering

Réalisé par :
BOUROUINA Rania, 181831052716
CHIBANE Ilies, 181831072041

Année universitaire : 2022 / 2023

Table des matières

Introduction Générale

Lors du chapitre précédant, nous nous étions attardés sur la classification qui, comme nous avions pu le constater, donne des résultats très convaincants, mais souffre quand même de quelques limitations. La plus notable étant la nécessité de connaitre les classes à prédire au préalable. Qui dans un cas d'application réel n'est pas toujours le cas. C'est là que le clustering rentre en jeu.
Le clustering est une technique d'apprentissage automatique permettant de regrouper des chaînes de données par distance ou par similarité. Il s'agit d'une méthode non supervisée et populaire pour une analyse des données. Il est alors possible d'appliquer des algorithmes de classification afin de gérer ces données individuelles dans chaque groupe spécifique.
Cette méthode de classification est applicable lorsqu'il est difficile de collecter les données. Pourtant, c'est un problème récurrent, car de nombreuses métriques aboutiront à différents regroupements. Elle doit ainsi être sélectionnée avec prudence selon le résultat attendu et la manipulation des données.
C'est pour cela que dans cette partie, nous allons réaliser, étudier et évaluer 2 modèles de clustering distinct et nous les comparerons entre pour voir lequel est le plus adapté à notre jeu de données.

Application d'algorithmes de clustering basée densité

Introduction Les algorithmes de clustering basée densité, comme leur nom l'indique, utilise la densité afin d'établir leurs clusters. C'est-à-dire qu'il détecte les zones où les points sont concentrés (grande densité) et celles où les points sont séparés par des zones vides ou clairsemées (basse densité). L'algorithme sur lequel nous allons nous attarder lors de cette partie est l'algorithme DBSCAN (density-based spatial clustering of applications with noise).

1.1 Définition et pseudo code

DBSCAN est un algorithme de partitionnement de données proposé en 1996 par Martin Ester, Hans-Peter Kriegel, Jörg Sander et Xiaowei Xu. Il s'agit d'un algorithme fondé sur la densité dans la mesure qui s'appuie sur la densité estimée des clusters pour effectuer le partitionnement.
Son pseudo code est le suivant :

Algorithme : DBSCAN

```
Entrées : D: Dataset, eps: distance minimum de voisinage, MinPts: nombre minimum
d'instance dans un cluster;
Sortie : D: Dataset étiqueté ;
   C = 0 ;
   Pour chaque point P non visité du dataset D Faire
      P.status = visité ; //Marquer P comme déjà visité
      PtsVoisins = epsilonVoisinage(D, P, eps) ;
      Si tailleDe(PtsVoisins) < MinPts Alors
         P.cluster = BRUIT; //Ajouter P à la liste des données bruitées
      sinon
         C++ ;//new cluster
         etendreCluster(D, P, PtsVoisins, C, eps, MinPts) ;
      FinSi;
   Fait;
Fin.
```

Fonction : etendreCluster

Entrées : ***D***: Dataset, ***eps***: distance minimum de voisinage, ***MinPts***: nombre minimum d'instance dans un cluster, ***P***: une instance de D, ***PtsVoisins***: liste des voisins de P, ***C***: numéro du cluster en cours de création ;
Sortie : *D* : Dataset étiqueté ;

```
Début
    P.cluster = C; // Ajouter P au cluster C
    Pour chaque point P' de PtsVoisins Faire
        Si P'.status <> visité Alors
            P'.status = visité ; // Marquer P' comme visité
            PtsVoisins' = epsilonVoisinage(D, P', eps) ;
            Si tailleDe(PtsVoisins') >= MinPts Alors
                PtsVoisins = PtsVoisins U PtsVoisins' ; // Inclure les voisins des voisins à la liste
            FinSi;
        FinSi;
        Si P'.cluster == 0 Alors //P' n'est membre d'aucun cluster
            P'.cluster = C ; // Ajouter P' au cluster C
        FinSi;
    Fait;
Fin.
```

Note : On appelle le "epsilon Voisinage" d'une instance P toutes les instances de D qui sont à une distance inférieure ou égale à epsilon de P.

Comme on peut le constater, cet algorithme est assez simple. Utilisant deux paramètres : la distance epsilon et le nombre minimum de points MinPts devant se trouver dans un rayon epsilon pour que ces points soient considérés comme un cluster. Les paramètres d'entrées sont donc une estimation de la densité de points des clusters. L'idée de base de l'algorithme est ensuite, pour un point donné, de récupérer son epsilon -voisinage et de vérifier qu'il contient bien MinPts points ou plus. Ce point est alors considéré comme faisant partie d'un cluster. On parcourt ensuite l'epsilon -voisinage de proche en proche afin de trouver l'ensemble des points du cluster. À noter que pour des raisons de rapidité d'exécution, nous utiliserons la distance de Manhattan.

1.2 Expérimentation des paramètres de DBSCAN sur le dataset et analyse des résultats

Maintenant que nous savons comment l'algorithme fonctionne, il est temps de l'expérimenter avec différents paramètres, c'est-à-dire en variant epsilon et le nombre minimum de voisins. Pour sélectionner les valeurs utilisées pour l'expérimentation, nous ne contenterons de prendre des valeurs variantes entre la plus courte distance recensée entre deux instances ainsi que la plus longue pour epsilon. De même, pour le nombre minimum de voisins, nous prendrons des valeurs ne dépassant pas le nombre maximum d'instances du dataset. Ce qui nous donne les résultats suivants :

eps	min_pts	clusters	len_clusters	noise
1	2	0	[]	1317
1	5	0	[]	1317
1	7	0	[]	1317
1	10	0	[]	1317
1	15	0	[]	1317
1	20	0	[]	1317
1	25	0	[]	1317
1	50	0	[]	1317
1	100	0	[]	1317
1	150	0	[]	1317
1	200	0	[]	1317
1	250	0	[]	1317
1	300	0	[]	1317
1	350	0	[]	1317
1	400	0	[]	1317
1	450	0	[]	1317
1	500	0	[]	1317
1	750	0	[]	1317
1	1000	0	[]	1317
2	2	14	[2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2]	1288
2	5	0	[]	1317
2	7	0	[]	1317
2	10	0	[]	1317
2	15	0	[]	1317
2	20	0	[]	1317
2	25	0	[]	1317
2	50	0	[]	1317
2	100	0	[]	1317
2	150	0	[]	1317
2	200	0	[]	1317
2	250	0	[]	1317
2	300	0	[]	1317
2	350	0	[]	1317
2	400	0	[]	1317
2	450	0	[]	1317
2	500	0	[]	1317
2	750	0	[]	1317
2	1000	0	[]	1317
3	2	68	[6, 3, 2, 355, 7, 3, 2, 7, 22, 3, 2, 4, 4, 2, 2, 4, 2, 2, 2, 2, 3, 2, 2, 4, 3, 2, 4, 2, 3, 3, 2, 3, 2, 3, 2, 4, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 4, 2, 4, 2, 2, 2, 2]	767
3	5	9	[256, 5, 4, 8, 3, 6, 4, 3, 6]	1042
3	7	2	[181, 5]	1138
3	10	1	[113]	1209
3	15	2	[16, 24]	1290

3	20	0	[]	1317
3	25	0	[]	1317
3	50	0	[]	1317
3	100	0	[]	1317
3	150	0	[]	1317
3	200	0	[]	1317
3	250	0	[]	1317
3	300	0	[]	1317
3	350	0	[]	1317
3	400	0	[]	1317
3	450	0	[]	1317
3	500	0	[]	1317
3	750	0	[]	1317
3	1000	0	[]	1317
4	2	13	[1151, 3, 6, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]	137
4	5	2	[1131, 4]	182
4	7	2	[1087, 5]	226
4	10	1	[1039]	279
4	15	1	[982]	336
4	20	1	[922]	398
4	25	1	[872]	450
4	50	1	[587]	735
4	100	1	[218]	1109
4	150	0	[]	1317
4	200	0	[]	1317
4	250	0	[]	1317
4	300	0	[]	1317
4	350	0	[]	1317
4	400	0	[]	1317
4	450	0	[]	1317
4	500	0	[]	1317
4	750	0	[]	1317
4	1000	0	[]	1317
5	2	1	[1307]	10
5	5	1	[1306]	11
5	7	1	[1303]	14
5	10	1	[1302]	15
5	15	1	[1297]	20
5	20	1	[1291]	27
5	25	1	[1288]	30
5	50	1	[1257]	62
5	100	1	[1198]	121
5	150	1	[1130]	196
5	200	1	[1027]	304
5	250	1	[840]	493
5	300	1	[598]	739
5	350	0	[]	1317
5	400	0	[]	1317

5	450	0	[]	1317
5	500	0	[]	1317
5	750	0	[]	1317
5	1000	0	[]	1317
6	2	1	[1317]	0
6	5	1	[1317]	0
6	7	1	[1317]	0
6	10	1	[1317]	0
6	15	1	[1317]	0
6	20	1	[1317]	0
6	25	1	[1317]	0
6	50	1	[1315]	2
6	100	1	[1313]	5
6	150	1	[1312]	7
6	200	1	[1307]	12
6	250	1	[1303]	16
6	300	1	[1297]	22
6	350	1	[1285]	34
6	400	1	[1272]	54
6	450	1	[1258]	68
6	500	1	[1231]	104
6	750	0	[]	1317
6	1000	0	[]	1317
7	2	1	[1317]	0
7	5	1	[1317]	0
7	7	1	[1317]	0
7	10	1	[1317]	0
7	15	1	[1317]	0
7	20	1	[1317]	0
7	25	1	[1317]	0
7	50	1	[1317]	0
7	100	1	[1317]	0
7	150	1	[1317]	0
7	200	1	[1317]	0
7	250	1	[1317]	0
7	300	1	[1317]	0
7	350	1	[1317]	1
7	400	1	[1317]	1
7	450	1	[1317]	2
7	500	1	[1317]	2
7	750	1	[1311]	15
7	1000	1	[1047]	999
8	2	1	[1317]	0
8	5	1	[1317]	0
8	7	1	[1317]	0
8	10	1	[1317]	0
8	15	1	[1317]	0
8	20	1	[1317]	0

8	25	1	[1317]	0
8	50	1	[1317]	0
8	100	1	[1317]	0
8	150	1	[1317]	0
8	200	1	[1317]	0
8	250	1	[1317]	0
8	300	1	[1317]	0
8	350	1	[1317]	0
8	400	1	[1317]	0
8	450	1	[1317]	0
8	500	1	[1317]	0
8	750	1	[1317]	1
8	1000	1	[1317]	3
9	2	1	[1317]	0
9	5	1	[1317]	0
9	7	1	[1317]	0
9	10	1	[1317]	0
9	15	1	[1317]	0
9	20	1	[1317]	0
9	25	1	[1317]	0
9	50	1	[1317]	0
9	100	1	[1317]	0
9	150	1	[1317]	0
9	200	1	[1317]	0
9	250	1	[1317]	0
9	300	1	[1317]	0
9	350	1	[1317]	0
9	400	1	[1317]	0
9	450	1	[1317]	0
9	500	1	[1317]	0
9	750	1	[1317]	0
9	1000	1	[1317]	0
10	2	1	[1317]	0
10	5	1	[1317]	0
10	7	1	[1317]	0
10	10	1	[1317]	0
10	15	1	[1317]	0
10	20	1	[1317]	0
10	25	1	[1317]	0
10	50	1	[1317]	0
10	100	1	[1317]	0
10	150	1	[1317]	0
10	200	1	[1317]	0
10	250	1	[1317]	0
10	300	1	[1317]	0
10	350	1	[1317]	0
10	400	1	[1317]	0
10	450	1	[1317]	0

10	500	1	[1317]	0
10	750	1	[1317]	0
10	1000	1	[1317]	0
11	2	1	[1317]	0
11	5	1	[1317]	0
11	7	1	[1317]	0
11	10	1	[1317]	0
11	15	1	[1317]	0
11	20	1	[1317]	0
11	25	1	[1317]	0
11	50	1	[1317]	0
11	100	1	[1317]	0
11	150	1	[1317]	0
11	200	1	[1317]	0
11	250	1	[1317]	0
11	300	1	[1317]	0
11	350	1	[1317]	0
11	400	1	[1317]	0
11	450	1	[1317]	0
11	500	1	[1317]	0
11	750	1	[1317]	0
11	1000	1	[1317]	0
12	2	1	[1317]	0
12	5	1	[1317]	0
12	7	1	[1317]	0
12	10	1	[1317]	0
12	15	1	[1317]	0
12	20	1	[1317]	0
12	25	1	[1317]	0
12	50	1	[1317]	0
12	100	1	[1317]	0
12	150	1	[1317]	0
12	200	1	[1317]	0
12	250	1	[1317]	0
12	300	1	[1317]	0
12	350	1	[1317]	0
12	400	1	[1317]	0
12	450	1	[1317]	0
12	500	1	[1317]	0
12	750	1	[1317]	0
12	1000	1	[1317]	0
13	2	1	[1317]	0
13	5	1	[1317]	0
13	7	1	[1317]	0
13	10	1	[1317]	0
13	15	1	[1317]	0
13	20	1	[1317]	0
13	25	1	[1317]	0

13	50	1	[1317]	0
13	100	1	[1317]	0
13	150	1	[1317]	0
13	200	1	[1317]	0
13	250	1	[1317]	0
13	300	1	[1317]	0
13	350	1	[1317]	0
13	400	1	[1317]	0
13	450	1	[1317]	0
13	500	1	[1317]	0
13	750	1	[1317]	0
13	1000	1	[1317]	0
14	2	1	[1317]	0
14	5	1	[1317]	0
14	7	1	[1317]	0
14	10	1	[1317]	0
14	15	1	[1317]	0
14	20	1	[1317]	0
14	25	1	[1317]	0
14	50	1	[1317]	0
14	100	1	[1317]	0
14	150	1	[1317]	0
14	200	1	[1317]	0
14	250	1	[1317]	0
14	300	1	[1317]	0
14	350	1	[1317]	0
14	400	1	[1317]	0
14	450	1	[1317]	0
14	500	1	[1317]	0
14	750	1	[1317]	0
14	1000	1	[1317]	0
15	2	1	[1317]	0
15	5	1	[1317]	0
15	7	1	[1317]	0
15	10	1	[1317]	0
15	15	1	[1317]	0
15	20	1	[1317]	0
15	25	1	[1317]	0
15	50	1	[1317]	0
15	100	1	[1317]	0
15	150	1	[1317]	0
15	200	1	[1317]	0
15	250	1	[1317]	0
15	300	1	[1317]	0
15	350	1	[1317]	0
15	400	1	[1317]	0
15	450	1	[1317]	0
15	500	1	[1317]	0

On peut constater une prédominance de résultats à un et zéro cluster, ce qui est logique, car la plupart des paramètres testés sont peu pertinents. On peut noter la présence de combinaisons de paramètres donnant plusieurs clusters comme montré ci-dessous :

eps	min_pts	clusters	len_clusters	noise
3	5	9	[256, 5, 4, 8, 3, 6, 4, 3, 6]	1042
4	2	13	[1151, 3, 6, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]	137
2	2	14	[2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2]	1288
3	2	68	[6, 3, 2, 355, 7, 3, 2, 7, 22, 3, 2, 4, 4, 2, 2, 4, 2, 2, 2, 2, 3, 2, 2, 4, 3, 2, 4, 2, 3, 3, 2, 3, 2, 3, 2, 4, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 4, 2, 4, 2, 2, 2, 2]	767

FIGURE 1.1 – Paramètre donnant un nombre de clusters supérieur à 2

Comme on peut le constater, les clusters obtenus ne sont pas très pertinents. Avec un schéma assez clair pour la plupart avec un cluster dominant contenant plusieurs instances et quelques miniclusters obtenus uniquement grâce au faible nombre de voisins minimum exigé. On peut aussi remarquer un nombre très important de bruit, ce qui montre que DBSCAN n'aura pas été très performant avec ces paramètres malgré un grand nombre de clusters.
Intéressons-nous maintenant au paramètre nous ayant donné deux clusters comme nous l'attendions. Nous pouvons voir via la figure ci-dessous que même si le bon nombre clusters a été atteint, leur contenu n'est pas des plus satisfaisants.

eps	min_pts	clusters	len_clusters	noise
3	7	2	[181, 5]	1138
3	15	2	[16, 24]	1290
4	5	2	[1131, 4]	182
4	7	2	[1087, 5]	226

FIGURE 1.2 – Paramètre donnant un nombre de clusters égale à 2

On peut voir à chaque fois soit la prédominance d'un cluster ou un nombre de bruits très élevé ou même les deux en même temps, ce qui montre que les clusters obtenu ne sont pas très pertinents, ce qui sera encore mieux démontré par la suite en se penchant sur les différents graphes réalisés.

1.3 Visualisation des résultats

Afin de réaliser une analyse plus constructive et plus approfondie des résultats de notre algorithme, nous effectuons de multiples visualisations dans le but de mieux appréhender le problème. L'une des visualisations les plus évidentes à effectuer et celle de nos clusters obtenus que nous effectuons sur plusieurs dimensions afin d'en avoir une vue plus concise.

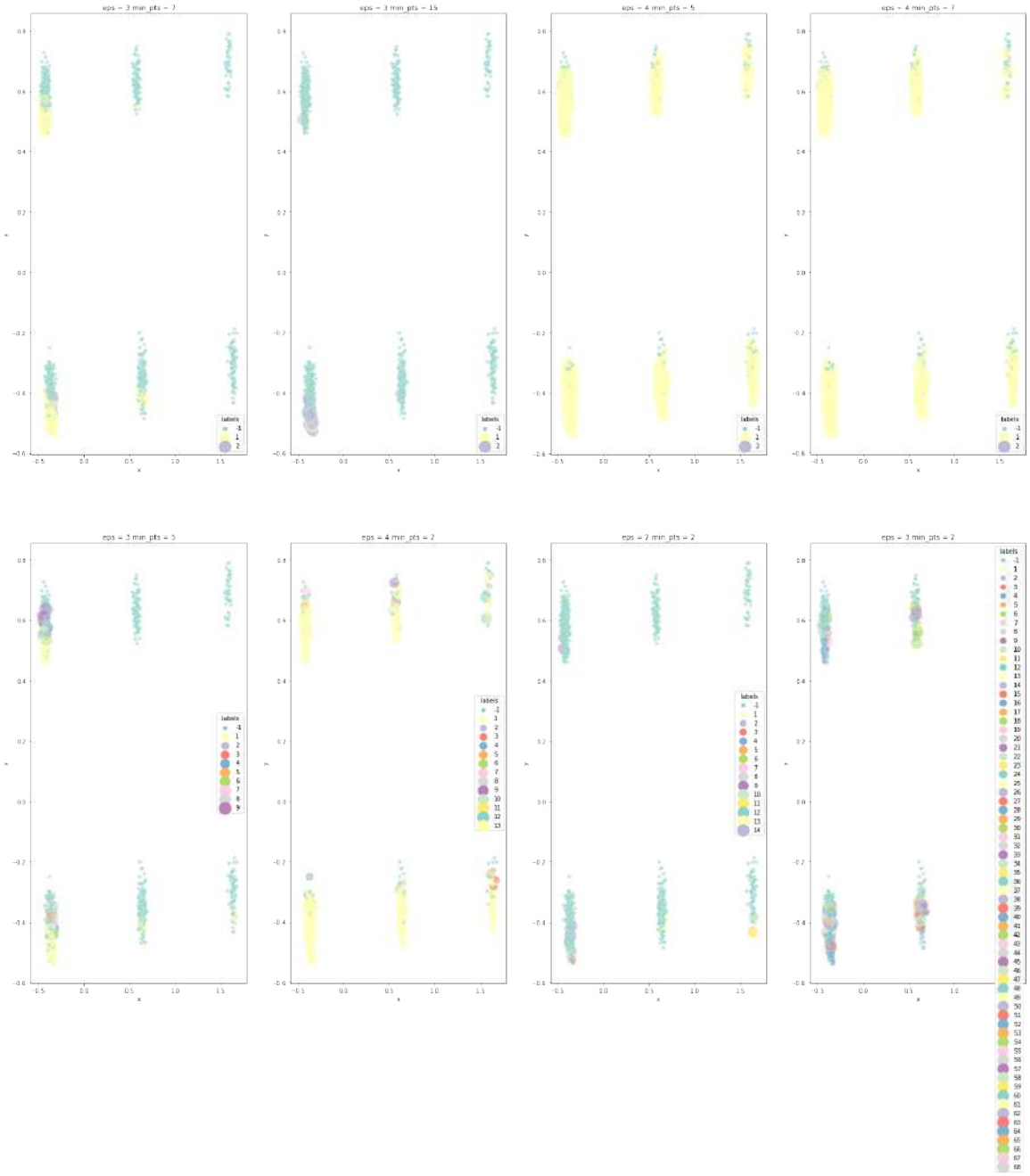

FIGURE 1.3 – Visualisant des différents clusters obtenus avec les différents paramètres résultant au moins 2 clusters en 2 dimensions

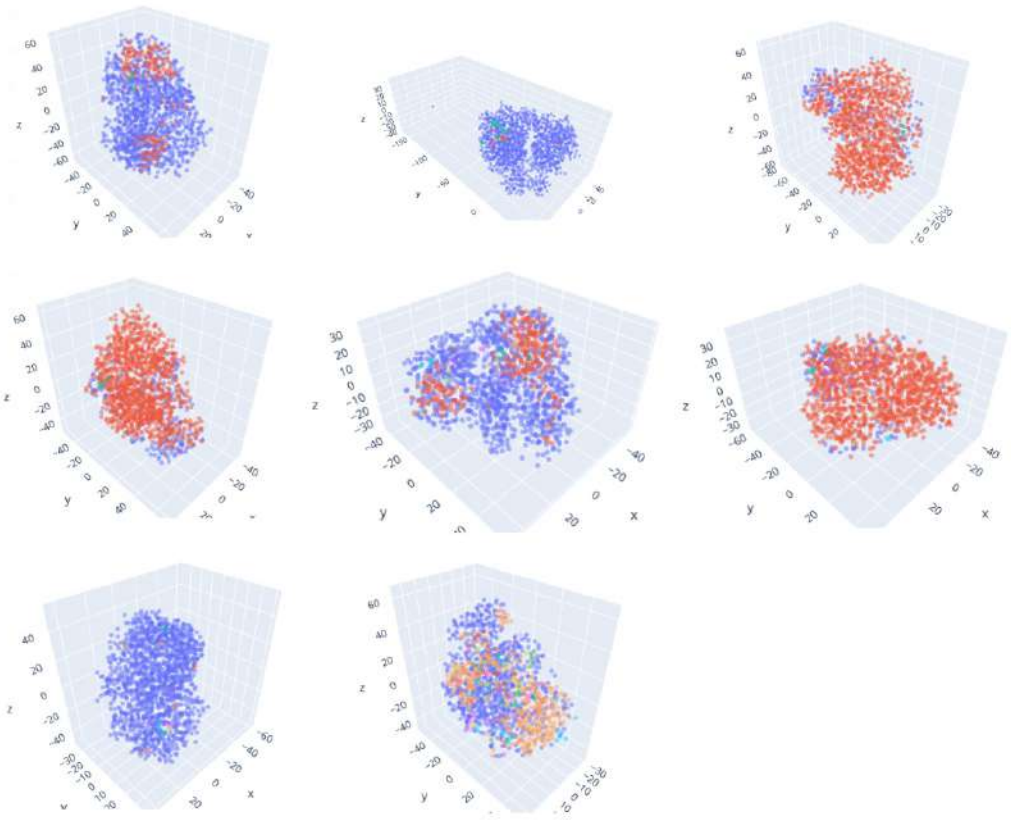

FIGURE 1.4 – Visualisant des différents clusters obtenus avec les différents paramètres résultant au moins 3 clusters en 2 dimensions

Comme on peut le constater, les clusters obtenus n'ont que très peu de sens et sont peu pertinents, la visualisation 2D n'est pas très représentative, car la réduction des données a beaucoup modifié l'aspect des données, mais même en s'attardant avec la visualisation 3D qui respecte davantage la représentation initiale des données, on peut voir que les clusters obtenus n'a que très peu de sens.

L'un des problèmes de notre visualisation est la faite que suite à l'application de réduction de données via la méthode PCA. Cela à pour effet inévitable de changer la manière dans nos données représentées sont représentées et peut induire à un changement de la forme des clusters. Pour essayer de remédier à cela, nous allons effectuer notre visualisation sur 2 attributs de notre jeu de données, ce qui donne les résultats suivants :

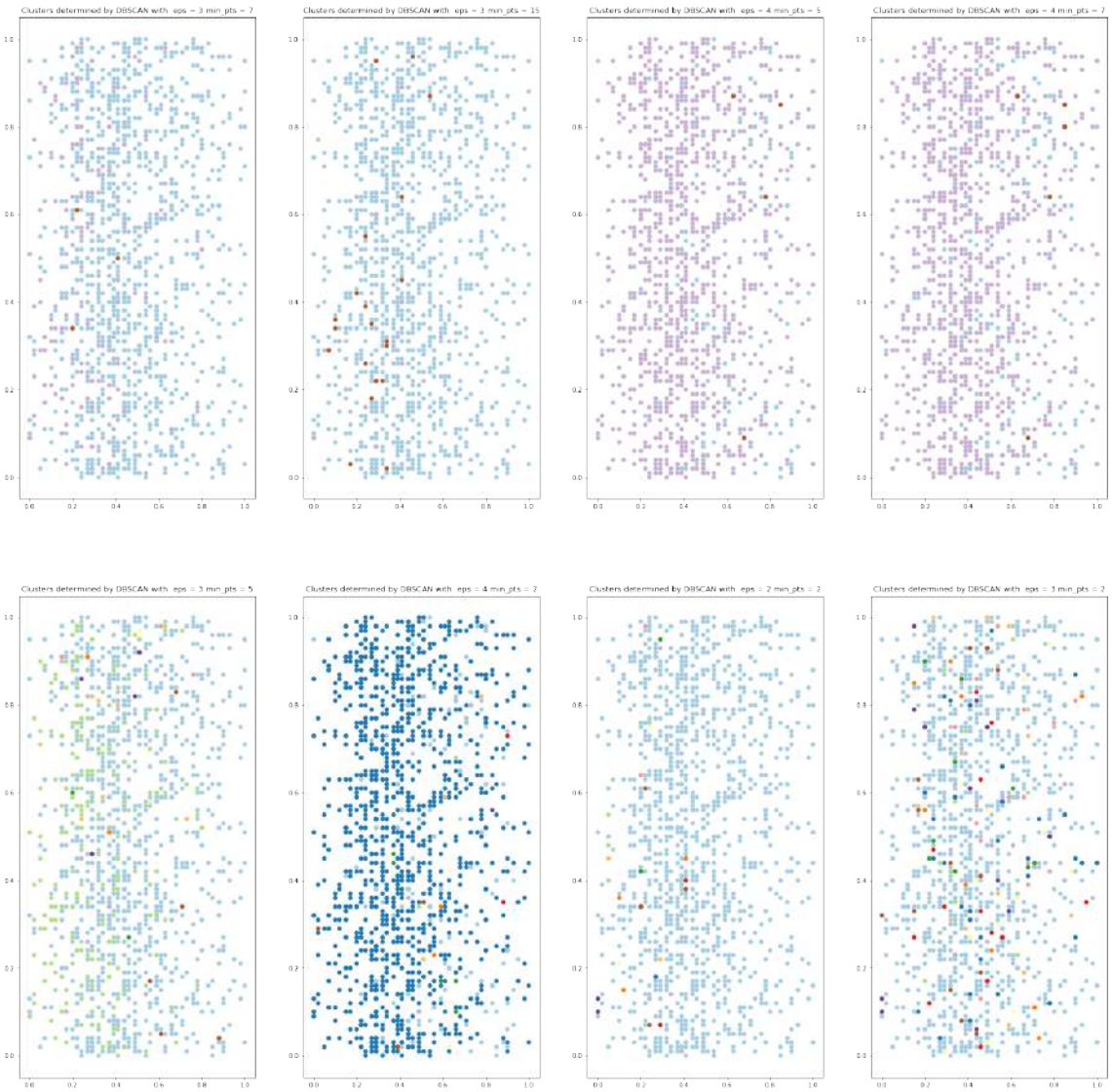

FIGURE 1.5 – Visualisant des différents clusters obtenus avec les différents paramètres résultant au moins 3 clusters avec 2 attributs uniquement

Cette visualisation met encore plus en avant l'inconsistance et l'incohérence de nos clusters. Avec les points entre chaque cluster qui sont des fois très espacées comme ce fut déjà le cas avec les visualisations 2D et 3D. Et même si cela ne reflète pas totalement la forme réelle de nos clusters qu'on ne peut malheureusement pas fidèlement représenter dû au nombre trop important de dimension nécessaire a cet effet. Une telle divergence dans les instances des clusters reste quand même assez aberrante pour le souligner.

La visualisation des clusters ayant clairement atteint sa limite, essayons de nous attarder sur les particularités de chaque cluster et cela peut se visualiser via plusieurs graphes.

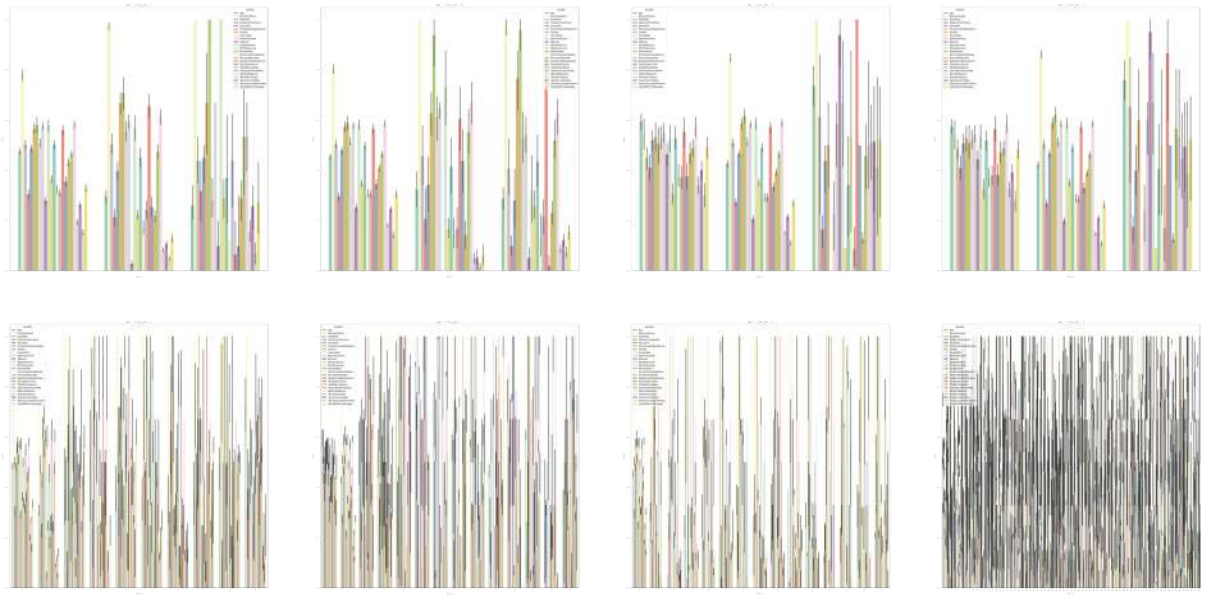

FIGURE 1.6 – Rania donne lui un titre stp

Ayant beaucoup de features cela handicape la visualisation, on va donc effectuer de la feature selection.

La technique utilisée pour la feature selection :

1- Regrouper les points de données par cluster et prendre la moyenne.

2- Calcule de l'écart type entre ces valeurs pour chaque feature.

Conclusion : Les features dont l'écart-type est élevé indiquent qu'il existe de grandes différences entre les clusters et que le feature est important.

Une fois effectuait, on obtient le graphe suivant :

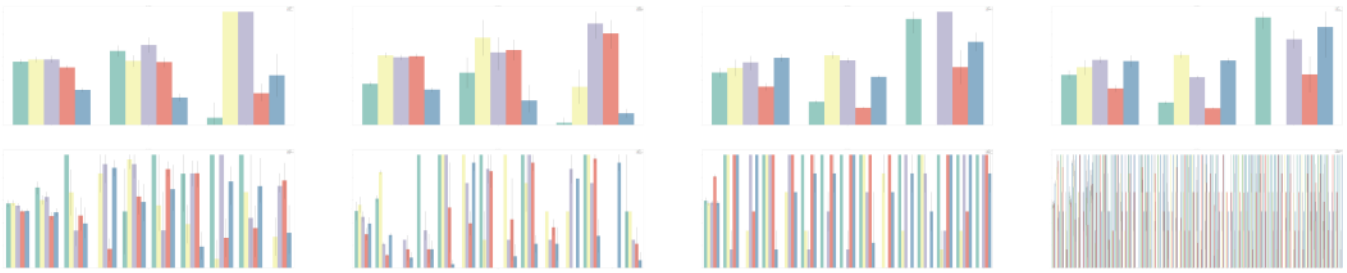

FIGURE 1.7 – Barplot des attributs de chaque cluster

Nous pouvons aussi ajouter à cela les visualisations suivantes :

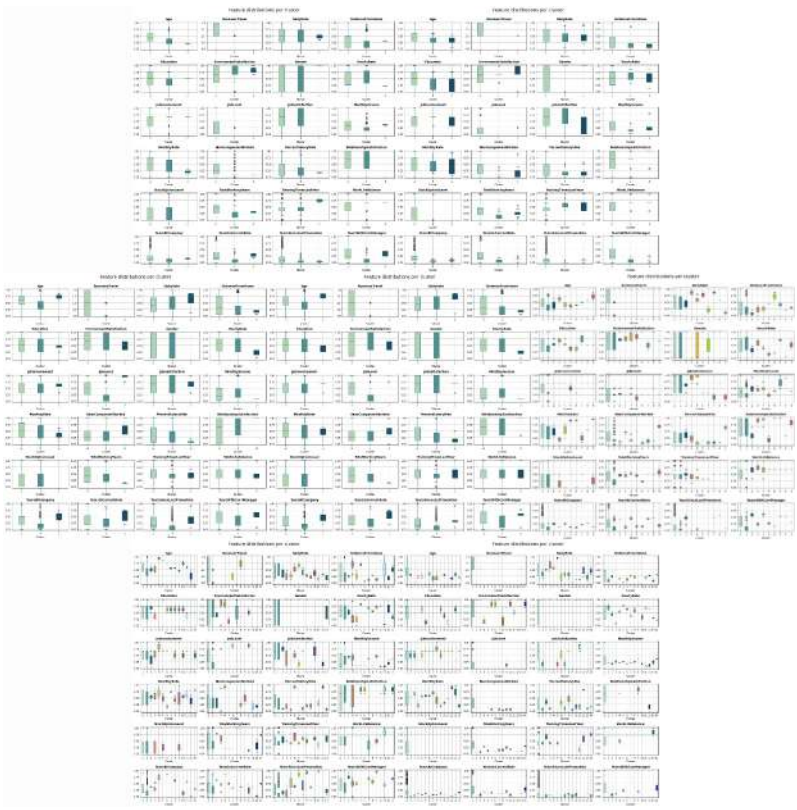

FIGURE 1.8 – Boxplots des attributs de chaque cluster

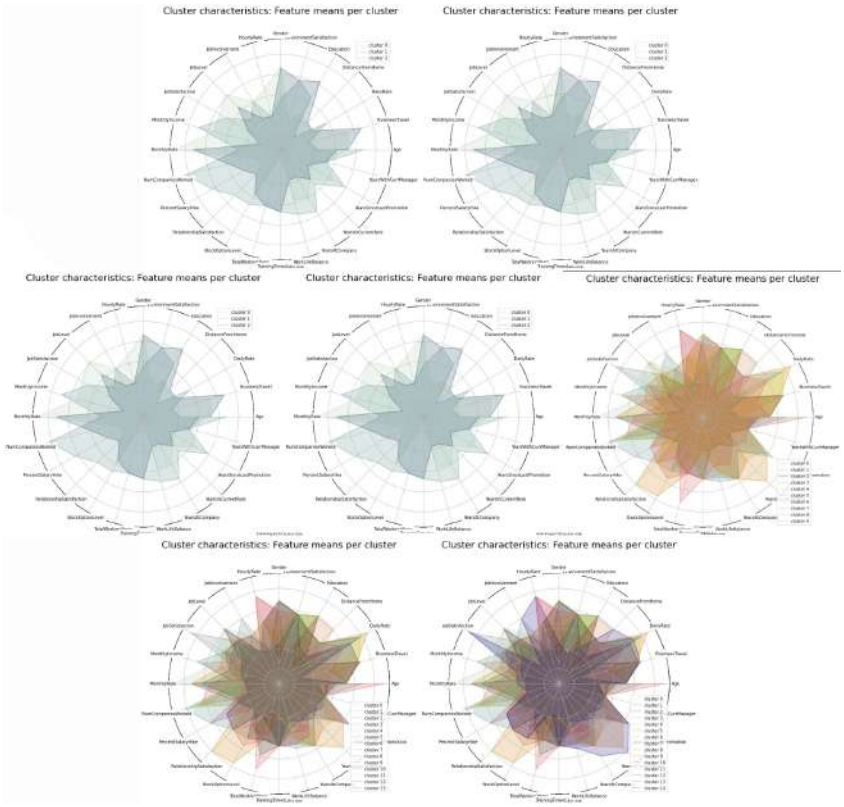

FIGURE 1.9 – radar de chaque cluster

Cela nous permet de constater que nos clusters sont très déséquilibrés du point de vue des attributs contenu, ce qui démontre encore plus la non-performance de l'algorithme DBSCAN sur notre jeu de données. Toutes ces analyses nous permettent d'émettre moult conclusions. Déjà DBSCAN n'est pas adapté pour les dataset dont les attributs sont trop nombreux, car cela a pour effet de rendre la densité plus complexe à évaluer. Il est donc préférable d'opérer un travail de feature selection ainsi que de réduction de données afin de pouvoir obtenir des résultats convenables. Il faut aussi souligner que DBSCAN est très sensible au chevauchement, ce qui fausse les clusters obtenus, il faut donc faire attention à si les instances de notre dataset reproduise cette particularité, ce qui fut le cas avec nos données comme vu lors de la visualisation. Pour terminer sur une note positive, les particularités de DBSCAN consistant à ne pas avoir besoin de spécifier le nombre de clusters et son traitement automatique du bruit reste des avantages considérables qui peuvent s'avérer extrêmement utile dans des cas plus approprié.

Application d'algorithmes de clustering basée hiérarchie

Introduction Les algorithmes de clustering basée hiérarchie permettent de partitionner un jeu de données de manière hiérarchique. Leur avantage qu'il n'est pas nécessaire de définir le nombre de clusters à l'avance (on explore toutes les possibilités). Cependant, cela ne fait que repousser cette décision. Celle-ci peut se faire sur la base d'un dendrogramme. Adapté aux échantillons contenant un faible nombre d'individus dû à sa complexité élevée, nous nous attarderons lors de ce chapitre à l'algorithme AGNES (Agglomerative Nesting).

2.1 Définition et pseudo code

AGNES ou en français classification ascendante hiérarchique consiste à partir d'une situation où tous les individus sont seuls dans une classe, puis sont rassemblés en classes de plus en plus grandes. Pour cela, on calcule la distance entre chaque cluster et on fusionne les clusters les plus proches. Il existe plusieurs manières pour calculer la distance entre 2 clusters parmi eux :
- Maximum or complete linkage : (Lien maximum ou complet) La valeur maximale de toutes les distances par paires entre les éléments du cluster C_1 et les éléments du cluster C_2.

$$dist(C_1, C_2) \ = \ Max(\, dist(e_1, e_2),\ e_1 \ \in \ C_1 \ et \ e_2 \ \in \ C_2$$

- Minimum or single linkage : (Liaison minimale ou unique) La valeur minimale de toutes les distances par paires entre les éléments du cluster C_1 et les éléments du cluster C_2.

$$dist(C_1, C_2) \ = \ Min(\, dist(e_1, e_2),\ e_1 \ \in \ C_1 \ et \ e_2 \ \in \ C_2$$

- Mean or average linkage : (Lien moyen) La distance moyenne entre les éléments du cluster C_1 et les éléments du cluster C_2.

$$dist(C_1, C_2) \ = \ \frac{1}{size(C_1) \ x \ size(C_2)} \ \sum_{e_1 \ \in \ C_1} \sum_{e_2 \ \in \ C_2} dist(e_1, e_2)$$

- Centroid linkage : (Lien centroïde) La distance entre deux clusters est définie comme la distance entre le centroïde du cluster C_1 et le centroïde du cluster C_2.

Le pseudo code est le suivant :

```
Algorithme : AGNES (AGglomerative NESting)

Entrée : D : Dataset ;
Sortie : D: Dataset étiqueté ;
Début
   Répéter
      Calculer la distance entre chaque pair de clusters avec une méthodes d'agglomération ;
      Fusionner la pair de clusters ayant la distance minimale ;
   Jusqu'à ce que les données forment un seul cluster ;
   Déterminer ou couper la hiérarchie pour obtenir les clusters voulus ;
   Retourner D ;
Fin;
```

FIGURE 2.1 – Code source de AGNES

On peut constater que cet algorithme est assez simple à implémenter, mais que le calcule répéter des distances cause une grande complexité, ce qui peut s'avérer problématique quand on doit gérer un trop grand nombre d'instances. Ce qui explique pourquoi il est préféré uniquement pour les échantillons contenant un faible nombre d'individus.

2.2 Expérimentation des paramètres de AGNES sur le dataset et analyse des résultats

Dans le cas de notre expérimentation, nous allons tester notre algorithme en utilisant plusieurs valeurs de distances voulu ainsi que les différentes méthodes de liaisons présentées précédemment (Maximum, Minimum, Moyenne et Centroïd), ce qui donne les résultats suivants :

Number of clusters	Linkage method	Clusters length
2	complete	[894, 423]
2	single	[805, 512]
2	average	[1008, 309]
2	centroid	[949, 368]
3	complete	[512, 382, 423]
3	single	[485, 512, 320]
3	average	[512, 496, 309]
3	centroid	[448, 501, 368]
4	complete	[512, 382, 256, 167]
4	single	[485, 512, 256, 64]
4	average	[512, 496, 246, 63]
4	centroid	[448, 501, 256, 112]
5	complete	[512, 254, 256, 167, 128]
5	single	[485, 256, 256, 256, 64]
5	average	[512, 256, 246, 240, 63]
5	centroid	[448, 246, 256, 255, 112]
6	complete	[256, 256, 254, 256, 167, 128]
6	single	[256, 256, 256, 256, 229, 64]
6	average	[256, 256, 256, 246, 240, 63]
6	centroid	[256, 246, 256, 255, 192, 112]
7	complete	[256, 256, 254, 256, 111, 128, 56]
7	single	[256, 256, 256, 256, 128, 101, 64]
7	average	[256, 256, 256, 246, 128, 112, 63]
7	centroid	[256, 246, 256, 255, 128, 112, 64]
8	complete	[256, 256, 254, 128, 111, 128, 128, 56]
8	single	[256, 256, 256, 128, 128, 128, 101, 64]
8	average	[256, 256, 256, 128, 128, 112, 118, 63]
8	centroid	[256, 246, 256, 127, 128, 128, 112, 64]
9	complete	[256, 256, 128, 128, 126, 111, 128, 128, 56]
9	single	[256, 256, 128, 128, 128, 128, 128, 101, 64]
9	average	[256, 256, 128, 128, 128, 128, 112, 118, 63]
9	centroid	[256, 246, 128, 128, 127, 128, 128, 112, 64]
10	complete	[256, 128, 128, 128, 128, 126, 111, 128, 128, 56]
10	single	[256, 128, 128, 128, 128, 128, 128, 128, 101, 64]
10	average	[256, 128, 128, 128, 128, 128, 128, 112, 118, 63]
10	centroid	[256, 128, 128, 128, 127, 128, 118, 128, 112, 64]

Comme on peut le constater, la répartition des instances entre les clusters est beaucoup plus homogène que sur DBSCAN qui même, on peut toujours noter un cluster un peu plus consistant que les autres, ils restent malgré tout avec une population assez élevée pour considérer comme des clusters pertinents. On peut aussi noter que la méthode liaison n'apporte que des changements mineurs, ne changeant que très peu le nombre d'instances dans chaque cluster, on peut en conclure qu'elle reste assez équivalente. En ce qui concerne es ce que ces clusters sont cohérents ou non, nous auront plus de détails à propos avec la visualisation.

2.3 Visualisation des résultats

L'une des premières visualisations qu'on se soit de faire lorsqu'on travaille avec AGNES est un dendrogramme. Un dendrogramme est un diagramme fréquemment utilisé pour illustrer l'arrangement de groupes générés par un regroupement hiérarchique ou hiérarchisant. Dans notre cas, on obtient le dendrogramme suivant :
(À noter que pour cette figure et toute les autres, nous visualiserons les résultats obtenus avec la méthode complete linkage)

FIGURE 2.2 – Dendrogramme

Cela nous permet de voir la formation de nos différents clusters avec beaucoup de discernement, mais ne nous donne pas plus d'informations sur la qualité de nos clusters, pour cela d'autres graphes seront nécessaires.

C'est pour cela que comme pour DBSCAN, nous allons effectuer à nouveau les meme visualisations vu précédemment, cela nous permettra aussi d'effectuer une comparaison plus pertinente entre les deux algorithmes.
Le premier graphe réalisé est un scatter plot sur 2 dimensions obtenu en utilisant la méthode ACP comme expliquer dans le chapitre précédent.

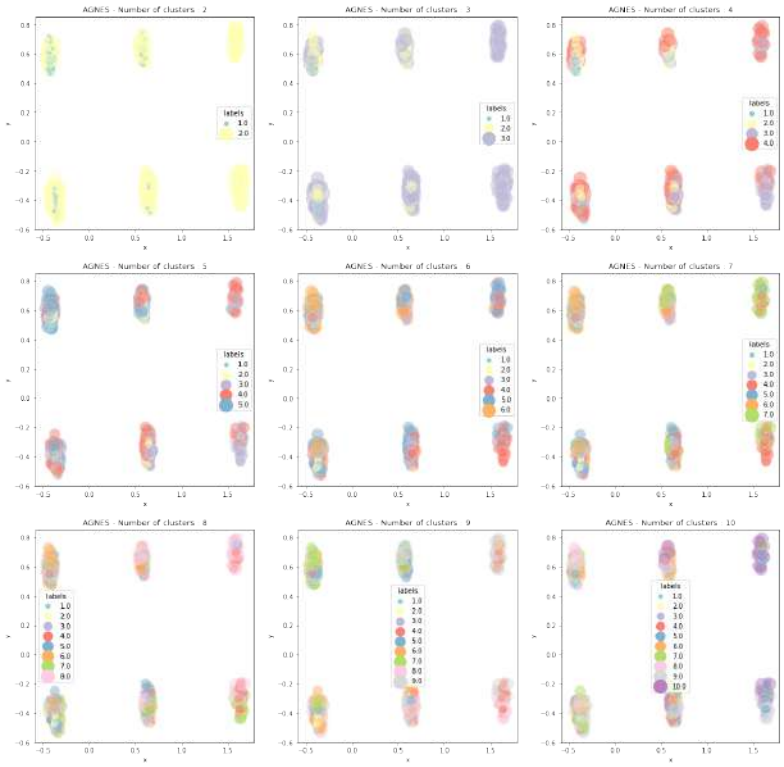

FIGURE 2.3 – Visualisation des différents clusters obtenus avec les différents paramètres résultant au moins 2 clusters en 2 dimensions

Comme la dernière fois, la réduction a deux dimensions ne nous permet pas d'avoir une bonne idée de la répartition de nos données entre les clusters. Même si on peut quand noter une amélioration comparée à DBSCAN notamment avec deux clusters.
Nous passons donc à une visualisation 3D pour essayer d'avoir une meilleure vue d'ensemble, et éventuellement confirmer nos intuitions au vu de notre analyse des plots 2D, ce qui donne les résultats suivants :

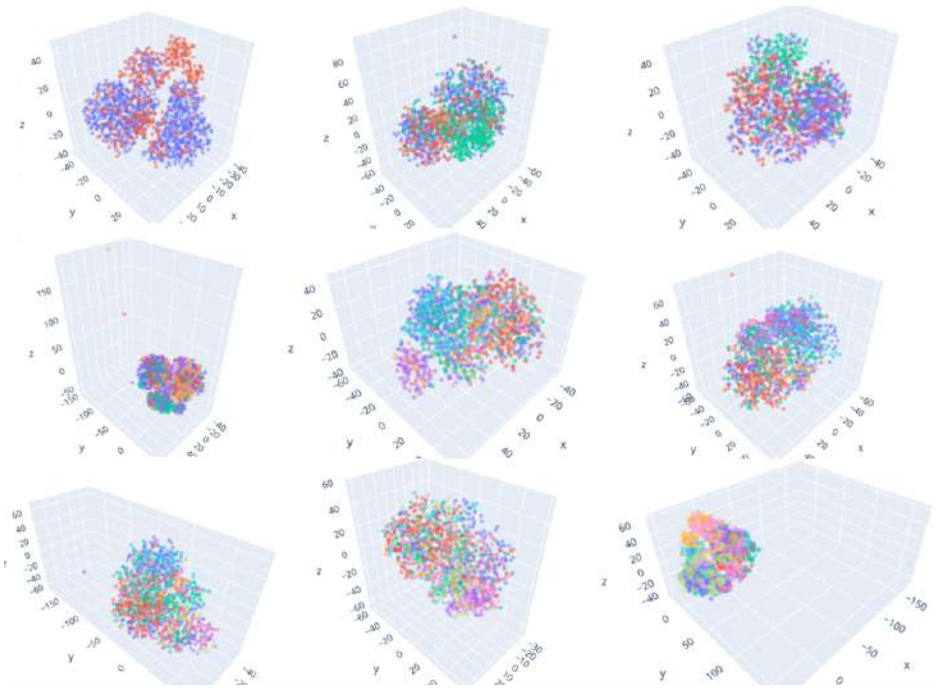

FIGURE 2.4 – Visualisation des différents clusters obtenus avec les différents paramètres résultant au moins 2 clusters en 3 dimensions

Comme on peut le voir nos données sont de mieux en mieux répartie, plus nos clusters sont limités. Les résultats obtenus avec AGNES sur deux clusters donnent une visualisation assez claire nous permettant de distinguer de manière assez distinguée nos clusters. Et plus le nombre de clusters augmentent de plus la répartition devient de moins en moins structuré. Certaines imperfections restent bien entendu, mais cela reste une visualisation suite à une réduction conséquente des données, il est donc normal d'avoir des approximations, mais on peut quand meme noté des résultats assez satisfaisants.
Nous pouvons aussi essayer de faire une visualisation sur 2 dimensions en prenant deux attributs de notre dataset comme fait précédemment avec dbscan comme illustrer ci-dessous :

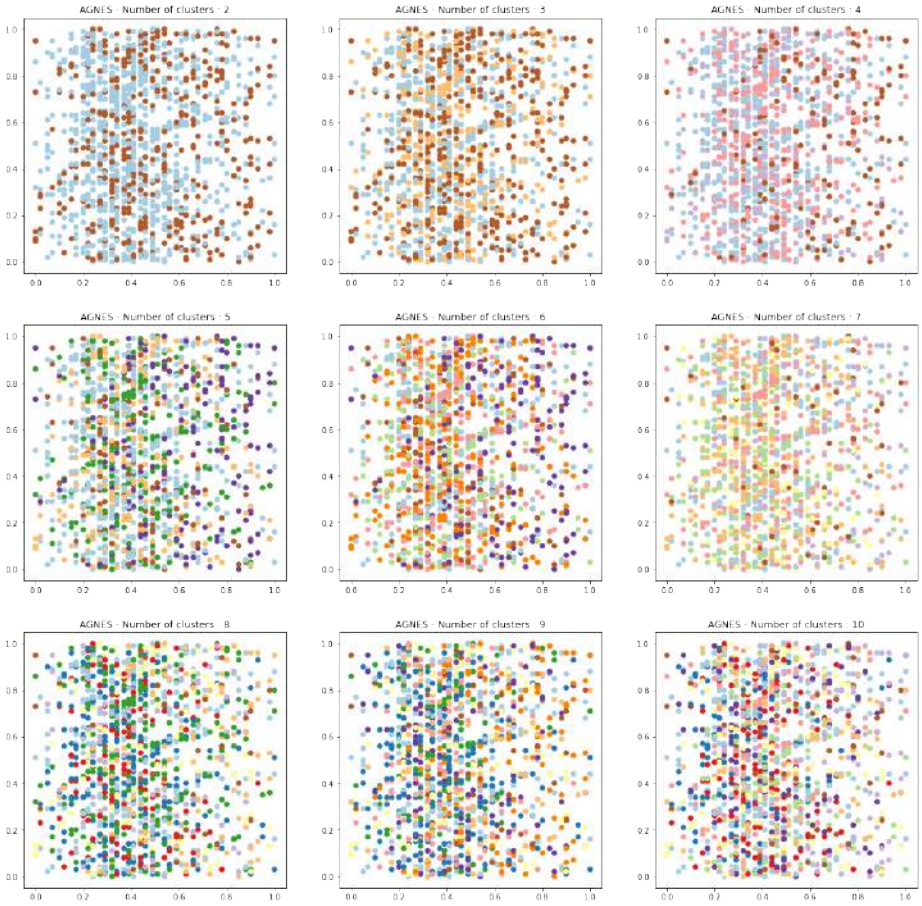

FIGURE 2.5 – Visualisation des différents clusters obtenus avec les différents paramètres résultant au moins 2 clusters en 2 dimensions avec des attributs prédéfinis

Encore fois, cette méthode de visualisation n'est pas des plus performantes, ne nous permettant pas de discerner nos clusters.
Notre visualisation 3D nous permettant d'avoir une bonne de nos clusters, on peut s'en contenter. Il est temps maintenant de pousser notre analyse plus loin et nous intéresser sur les caractéristiques internes de nos clusters en utilisant les meme graphiques que lors de notre visualisation de DBSCAN nous donnant les résultats suivants :

FIGURE 2.6 – Barplot des attributs de chaque cluster

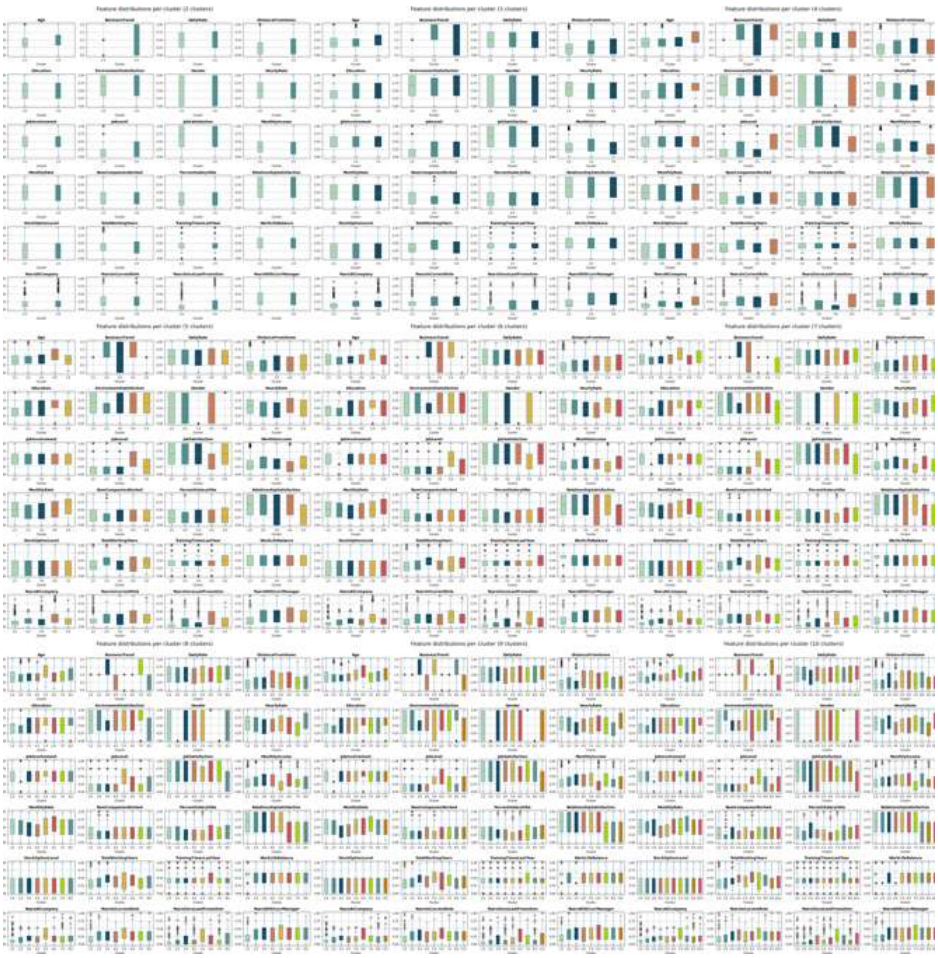

FIGURE 2.7 – Boxplots des attributs de chaque cluster

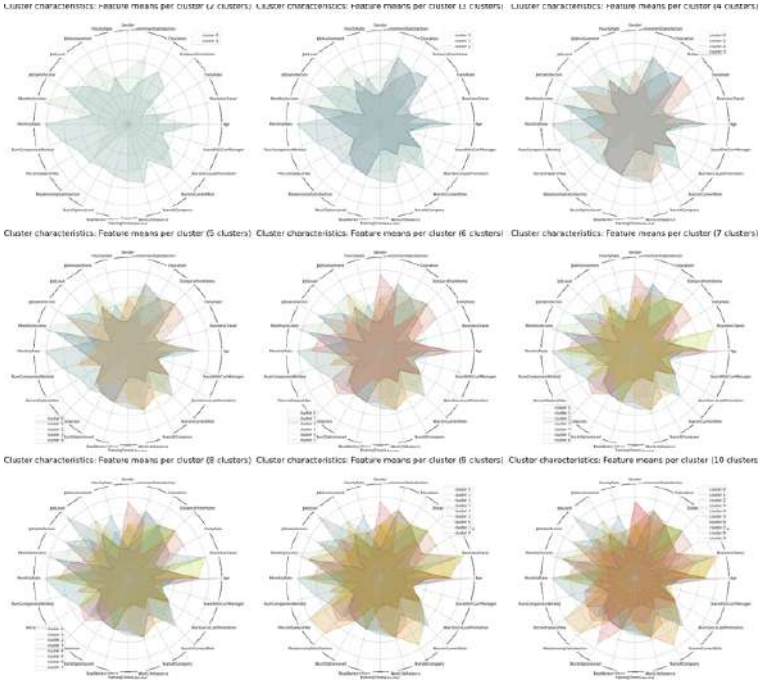

FIGURE 2.8 – Radar de chaque cluster

Cela nous permet de constater que nous clusters sont légèrement déséquilibrés sur certains points, mais cela reste plus homogène que DBSCAN dans l'ensemble.

Pour conclure, on peut noter que les résultats obtenus sont plus convaincants que ceux obtenus avec DBSCAN avec des clusters beaucoup plus logiques. Cependant, ils sont loin d'être parfait et leur qualité ceux dégradent plus on exige un nombre important de clusters. Sans oublier le temps d'exécution de l'algorithme qui est assez conséquent. Ces performances restent quand meme respectable dans l'ensemble et peuvent surement être améliorés avec un meilleur data preprocessing.

Comparaison des algorithmes DBSCAN et AGNES

Lors de cette partie du projet, nous nous sommes attardés sur type distinct de clustering et mettant de côté les résultats obtenu, il est facile de déceler de nombreuses différences entre les deux.
La plus notable reste le contrôle sur le nombre de clusters obtenu, AGNES permettant de spécifier le nombre de clusters voulant être obtenu contrairement à DBSCAN ou le seul moyen d'avoir un minimum de contrôle sur le nombre de clusters consiste à jouer avec la valeur d'epsilon et du nombre minimum de voisins.
De plus, on peut aussi constater que le calcule de distance est assez différent, cela notamment dû au fait que DBSCAN calcule la distance entre toutes les instances contrairement à AGNES qui lui calcule la distance entre les clusters, ce qui implique de nombreuses de différences au niveau des méthodes de calcule de distances.
Ce qui a un impact direct sur l'une des autres grandes différences entre ces 2 algorithmes qui est leur complexité avec DBSCAN ayant une complexité de $O(n^2)$ et AGNES de $O(kn^2)$ avec k représentant le nombre de clusters. Montrant pourquoi DBSCAN reste davantage approprié pour de grand dataset vu sa complexité réduite.
Pour finir, il est assez évident que l'une des plus grosses différences entre les 2 réside dans le type de données pour lesquelles ils sont le plus approprié. DBSCAN lui est plus approprié quand lorsque notre dataset contient plusieurs outliers et que nos données donnent des clusters de formes irrégulières. Le fait qu'il ne nécessite pas de spécifier le nombre de clusters et qu'il soit relativement rapide en font aussi un assez bon choix de manière général. Cependant, il devient très moins performant quand la densité entre les clusters n'est pas significative et que le nombre d'attributs est trop important. AGNES lui s'avère plus intéressant lorsque l'on a besoin d'identifier les observations les plus similaires à un point de données et que nous voulons voir des clusters à différents niveaux de granularité. Il peut aussi être adapté pour incorporer des variables catégoriques et qu'il soit moins stricte sur la forme des clusters en font un choix solide pour de nombreux cas d'utilisation. Malheureusement, sa lenteur, son besoin de définir le nombre de clusters et le prétraitement fastidieux qu'il peut demander pour obtenir de bons résultats font qu'il n'est pas adapté à tout type

de données.
Pour conclure, on peut affirmer que les 2 algorithmes sont très intéressants et peuvent s'avérer très performant selon les besoin de l'utilisateur ainsi que du type de données en notre disposition et du prétraitement effectué dessus.

Conclusion Générale

Dans cette partie du projet, nous nous sommes familiarisés avec le concept de clustering, son importance dans le domaine du datamining, et le fonctionnement de certain modèle. Tel que DBSCAN et AGNES.
Nous avons aussi pu constater encore plus lors de cette partie l'importance du prétraitement des données et du fait qu'il se doit d'être adapté à la problématique. Car meme si le traitement effectué lors de la partie 2, c'est avérer efficace pour la classification comme nous avions pu le voir lors de la partie 4. Elle s'est avérée bien inadapté au clustering.
Nous avons aussi pu déduire que le clustering malgré son intérêt indéniable reste difficile à implémenter et demande un grand travail afin de trouver la méthode la plus adaptée à nos données. De manière plus générale, ce projet nous aura introduit aux différents composant du datamining et de les maitriser et de savoir comment les utiliser à bon escient.