INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Analyse et traitement de données

Rapport de TP
Clustering

Florien LEON

Walid KHALED

TP 1 Clustering k-Means

Dans le TP1, nous appliquerons la méthode K-means sur un jeu de données appelé xclara.arff, Nous testerons les différents paramètres de la méthode ;

En fin, nous évaluerons le résultat avec d'autres métriques de scikit-learn.

TP 2 Clustering Agglomératif

Nous allons faire la même chose avec le jeu de données mais en appliquant cette fois ci la méthode de clustering Agglomératif

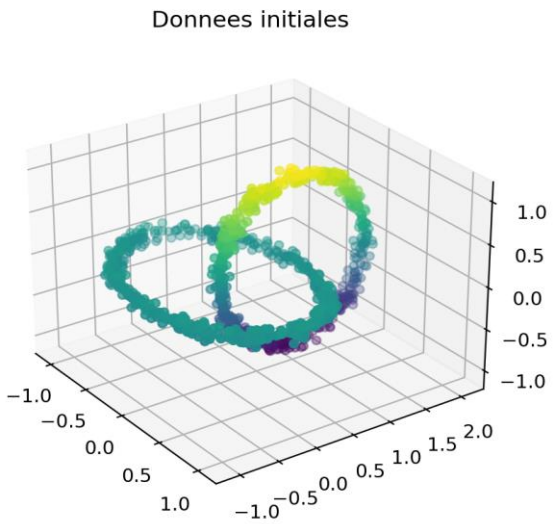
TP 3 Clustering DBSCAN

Nous voulons ici évaluer la méthode de clustering DBSCAN qui détermine automatiquement le bon nombre de clusters.

Table des matières

K-Means

Afficher les données en 3D (sans prétraitement) :



Méthode	Cas d'utilisation
K-Means	Pour les géométries plates, pas trop de clusters
Affinity propagation	Pour les géométries non plates avec beaucoup de clusters
Mean-shift	Pour les géométries non plates avec beaucoup de clusters
Spectral clustering	Pour les géométries non plates avec peu de clusters
Ward hierarchical clustering	Plusieurs clusters
Agglomerative clustering	Plusieurs clusters, distances non euclidiennes
DBSCAN	Pour les géométries non plates avec des tailles de clusters inégales avec élimination des valeurs aberrantes
OPTICS	Pour les géométries non plates avec des tailles de clusters inégales, densités variables avec élimination des valeurs aberrantes
Gaussian mixtures	Pour les géométries plates
BIRCH	Pour les grands jeux de données avec suppression des valeurs aberrantes

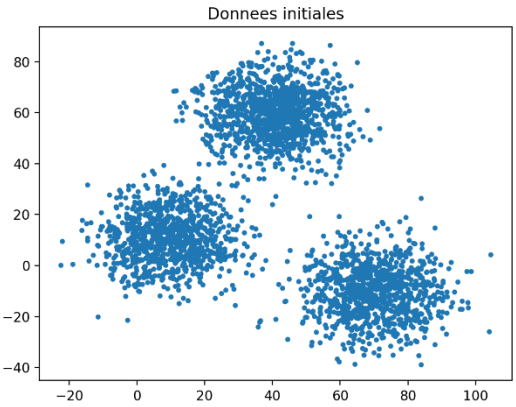
Source : https://scikit-learn.org/stable/modules/clustering.html#clustering

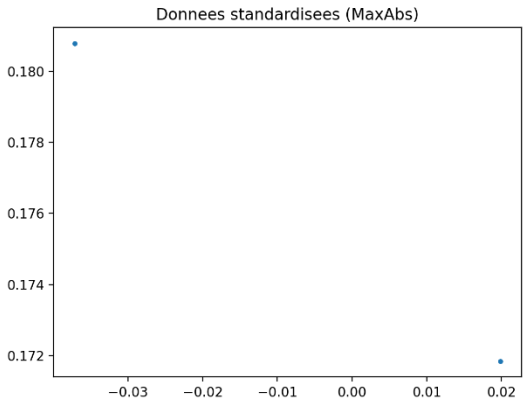
K_Means se base sur la variance pour définir ses clusters, donc un cluster regroupe les points à variance égale. Elle assume que les clusters sont de forme Convexe.

La méthode nécessite de connaitre le nombre de clusters K à l'avance et correspond à plusieurs domaines d'utilisation.

Appliquer le prétraitement sur le jeu de données « xclara.arff »

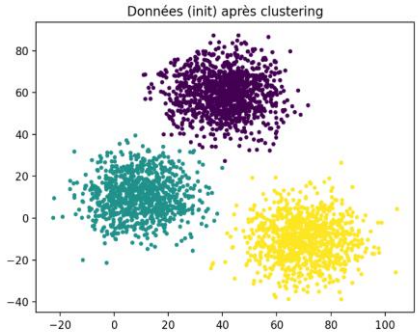
Nous avons appliqué plusieurs méthodes de standardisations qui ont donné plus ou moins le meme résultat, nous les avons donc pas mis ici pour ne pas encombrer le document.





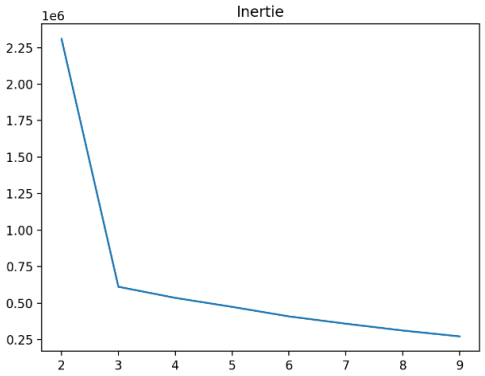
Evaluation de la méthode

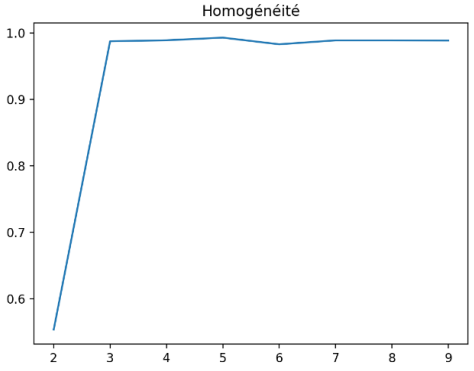
Application de K_Means avec le code qui calcule le meilleure K (nombre de clusters)

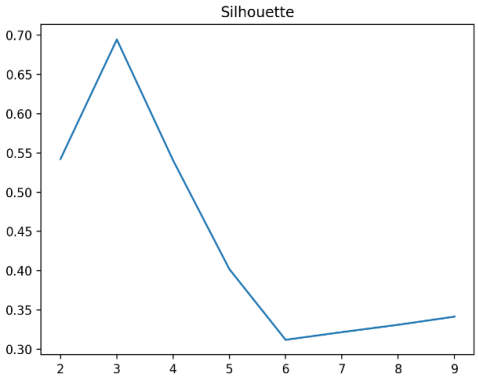


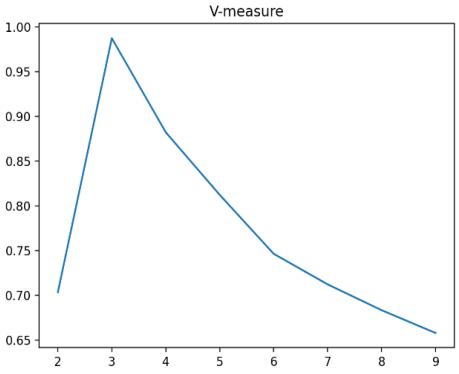
Nous remarquons que K-Means a bien fonctionné sur ce jeu de données, ce qui était prévu. Notre code retourne Best K = 3.

Les différentes métriques utilisées :







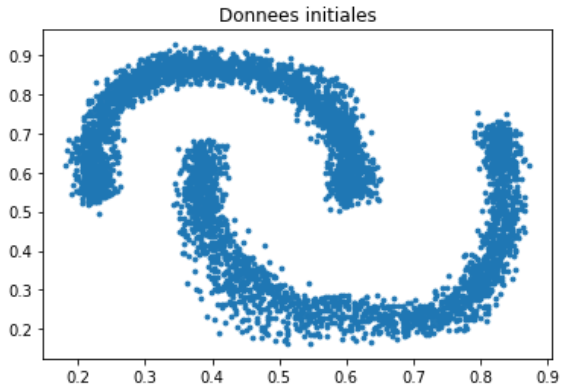


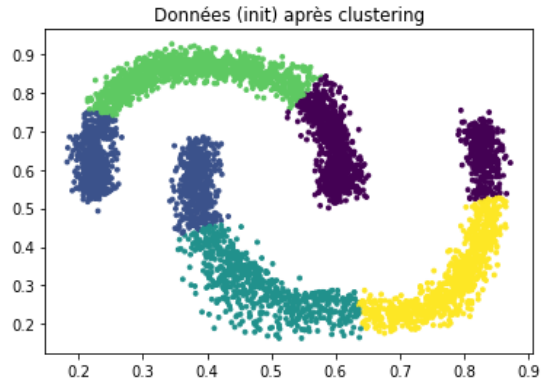
Toutes les métriques montrent que le meilleur K est 3.

```
------------------------------------------------------------
Appel KMeans pour une valeur de k fixée (données init)
nb clusters = 3 , nb iter = 3 , runtime =  45.56 ms
Inertie :  611605.8806933893
Coefficient de silhouette :  0.6945587736089913
Coefficient d'homogénéité :  0.9875007954296411
Coefficient de v-mesure :  0.9872347608204503
```

Jeux de données où K-Means ne fonctionne pas

1) Banana.arff



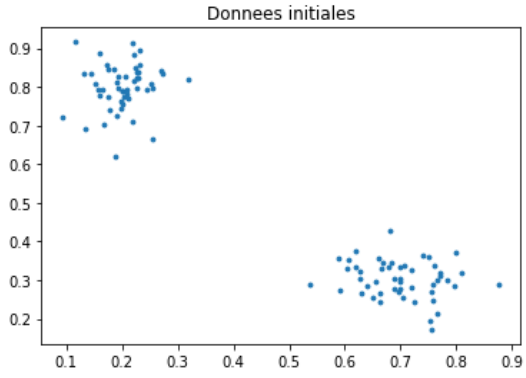


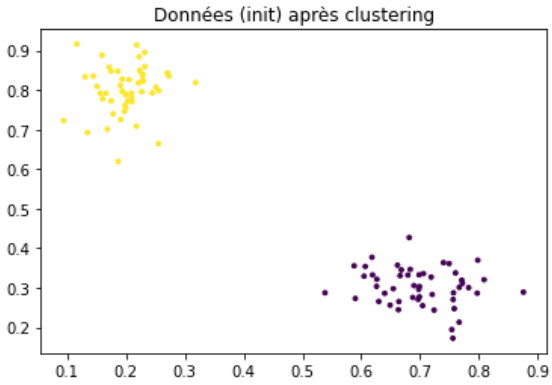
```
-------------------------------------------------------
Appel KMeans pour une valeur de k fixée (données init)
best k is =  5
Best inert 3
Best silh 8
Best v_meas 8
Best homs 2
-------------------------------------------------------
nb clusters = 5 , nb iter = 10 , runtime =  6459.81 ms
```

On remarque que K-means n'a pas pu faire un bon clustering a cause de la forme du jeu de donnée qui ne correspond pas à cette méthode.

Jeux de données où K-Means fonctionne

1) Gaussians1.arff

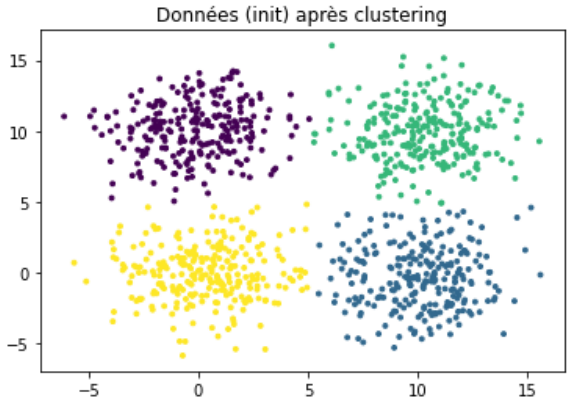


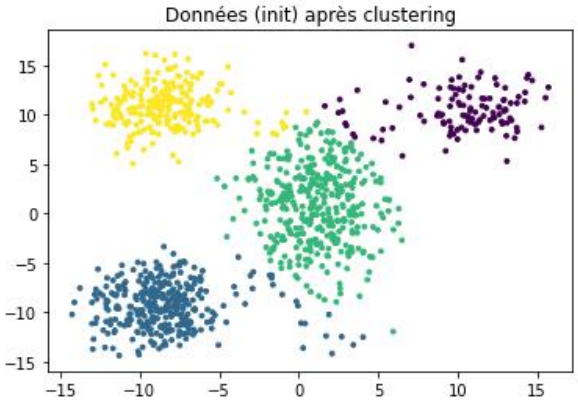



```
-------------------------------------------------------
Appel KMeans pour une valeur de k fixée (données init)
best k is =  2
Best inert 3
Best silh 2
Best v_meas 2
Best homs 3
-------------------------------------------------------
nb clusters = 2 , nb iter = 2 , runtime =  469.86 ms
```

La méthode K-means a bien fonctionné avec ce jeu de donnée avec peu de clusters et une forme plate.

Elle marche aussi pour le jeu : square1.arff & triangle2.arff





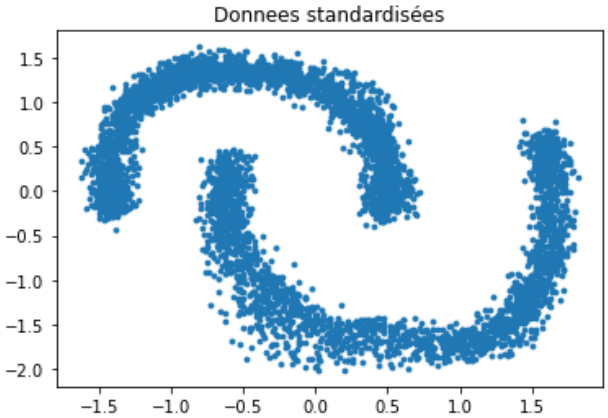
Avantages et inconvénients de la méthode K-Means

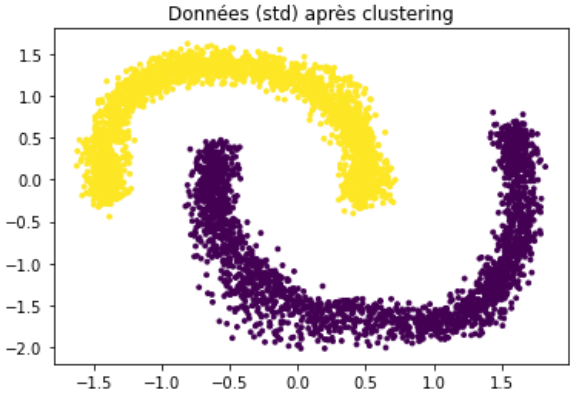
Avantages	Inconvénients
Simplicité : facile à implémenter	Choix manuel de K
Flexibilité : adaptation à plusieurs jeux de données	Effet uniforme : cluster de taille uniforme même si les données ont des tailles différentes
Facile à interpréter	Fonctionne qu'avec des données numériques
Faible coût de calcul	

Agglomératif

Cette méthode de clustering forme ces clusters en imbriquant ou en divisant les différents clusters pour former un arbre avec comme racine un unique cluster qui comportes les feuilles (clusters avec un seul échantillon).

Application de la méthode sur le jeu : banana :

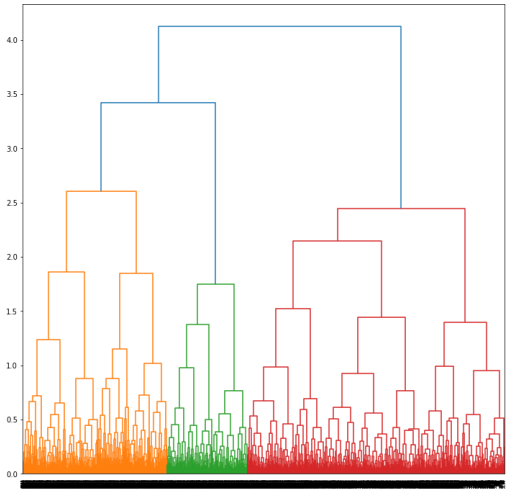




```
----------------------------------------
Best complexity :  3
Best homogeneity :  3
Best silhouette :  2
Best v-measure :  2
Best Davies-Bouldin :  2

best k is =  2
----------------------------------------------------------
Appel Aglo Clustering 'complete' pour une valeur de 2
determinée automatiquement
nb clusters = 2 , runtime =  6548.49 ms
```

Graphe des différentes métriques (voir le Git pour des raisons de surcharge du document)



On remarque sur le dendrogramme que nous pouvons faire un découpage à 2 clusters ce qui est claire, mais nous pouvons aussi faire un découpage un peu plus bas pour avoir 3 clusters, ce qui correspond avec les résultats des metrics.

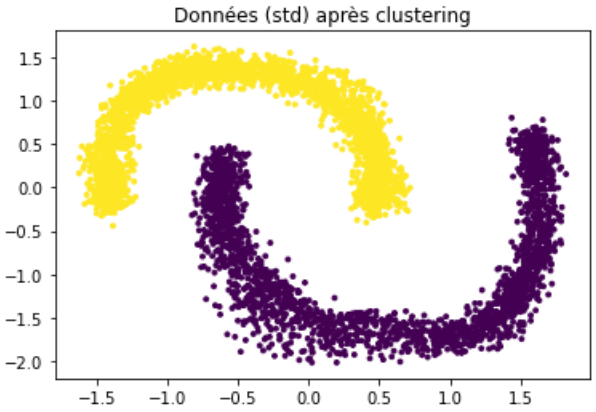
Variation des paramètres de la méthode et observation des résultats

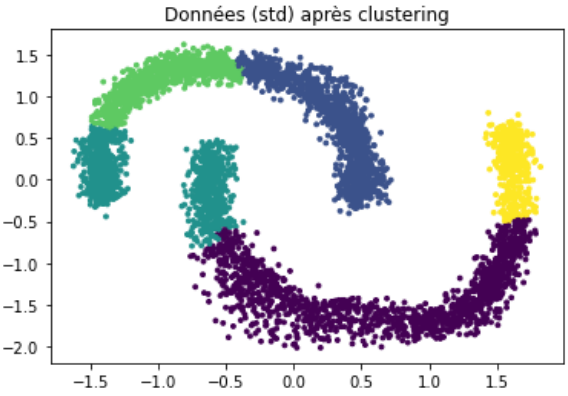
Les paramètres de « Linkage » de la méthode Agglomération sont :

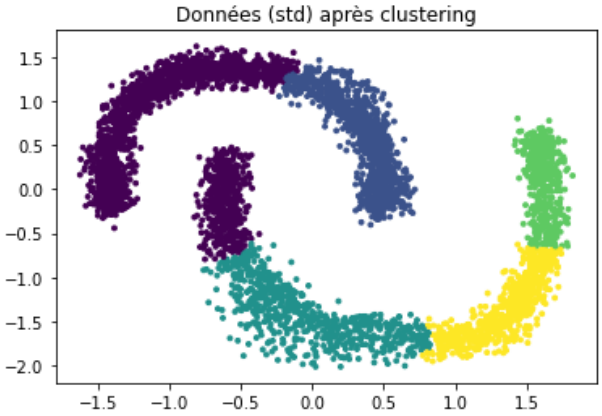
- **Single (1) :** la distance entre deux clusters représente celle entre les deux points les plus proches
- **Ward (2) :** minimise la variance entre deux clusters pour calculer la distance qui les sépare
- **Average (3) :** considère que la distance entre deux clusters est la distance moyenne entre les points des clusters
- **Complete (4) :** contrairement à single la distance entre deux clusters représente celle entre les deux points les plus éloignés

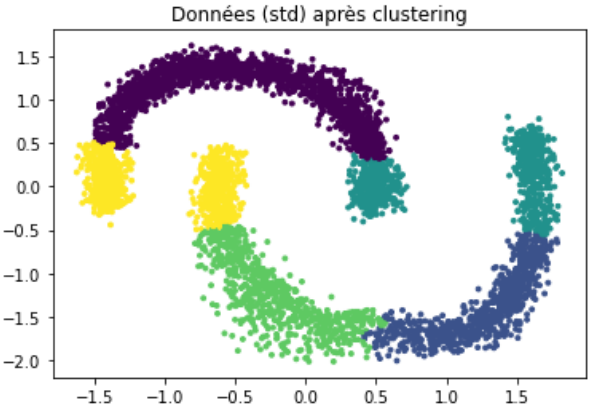
Nous varions les paramètres pour les deux jeux de données « xcalara » et « banana » :

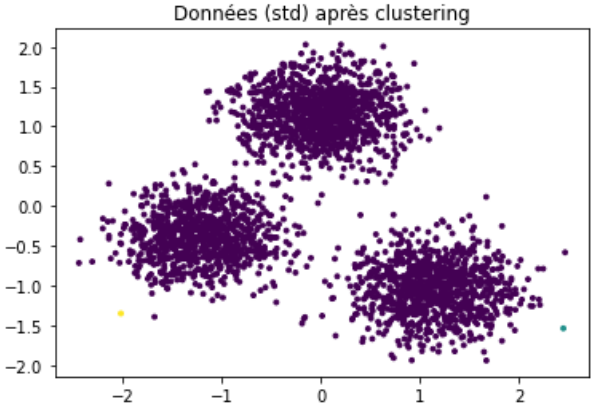
NB : Les linkage sont numéroté en partant de la gauche haut vers le bas à droite

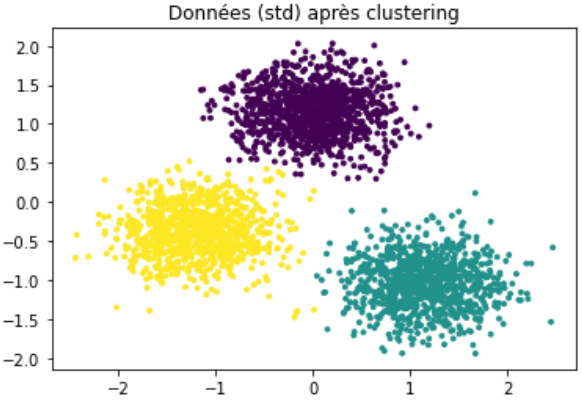

Données (std) après clustering

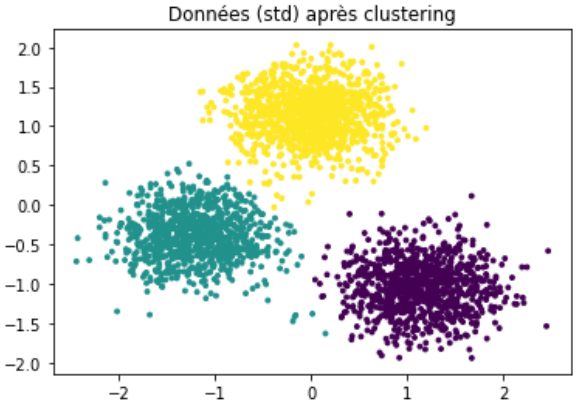

Données (std) après clustering

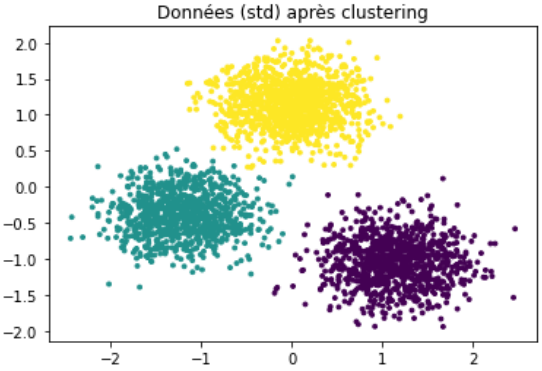

Données (std) après clustering


Données (std) après clustering


Données (std) après clustering


Données (std) après clustering


Données (std) après clustering


Données (std) après clustering

Nous remarquons que pour le jeu **Banana**, le linkage **Single** est le seul qui a marché. Par contre, sur le jeu **Xclara**, c'est le seul à ne pas avoir marché. Ce qui explique que les linkages peuvent être efficaces sur une forme de données et inefficaces pour d'autres.

Avantages et inconvénients de la méthode Agglomération

Avantages	Inconvénients
Il n'est pas nécessaire de définir K l'avance	Complexité algorithmiques
	Temps de traitement important
	Découpage facile du dendrogramme pour un petit jeu de données mais complexe pour des jeux plus grands

DBSCAN

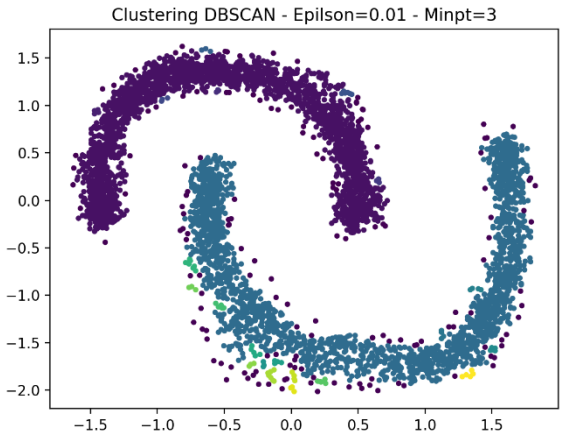
DBSCAN se base sur la densité des clusters pour faire son clustering, c'est-à-dire que les clusters plus denses sont séparés par les clusters moins denses ;

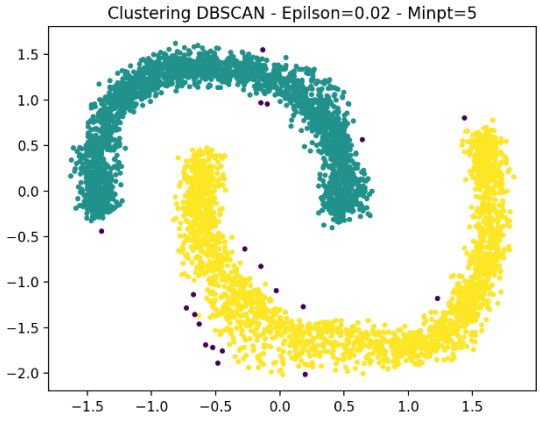
Les deux paramètres qui définissent le terme de densité sont :

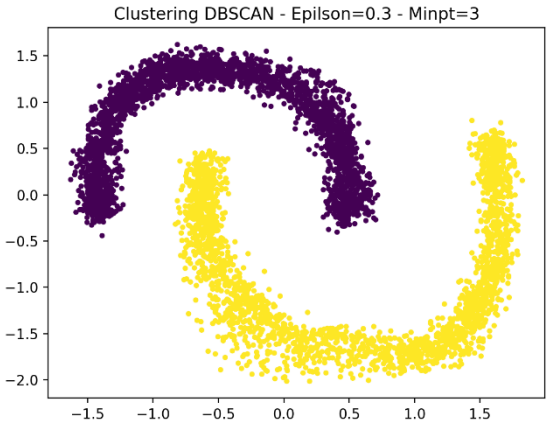
- Epsilon : distance entre deux points
- Min_sample : nombre de points dans un seul cluster

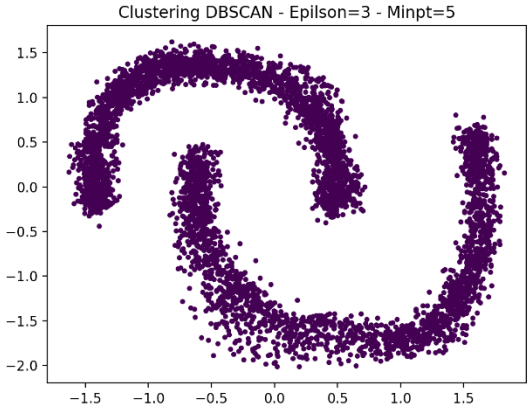
Application de plusieurs valeurs de eps et min_sample

On prend ici le jeu de données « Banana » :









On remarque que les deux paramètres affectent le résultat du clustering, il faut donc trouver les deux meilleures valeurs de deux paramètres en fixant l'un et variant l'autre.

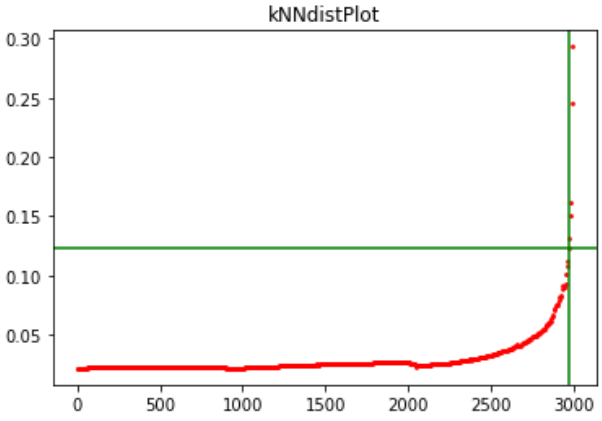
Si nous prenons un Epsilon trop petit la plupart des données ne sont pas regroupées comme sur la figure n°1 !

Si nous le prenons trop grand plusieurs clusters sont fusionnés comme sur la figure n°4 (eps =3)

La méthode qui nous permet de choisir un epsilon optimal est le tracé des distances du voisin le plus proche en prenant le genou pour déterminer Epsilon.

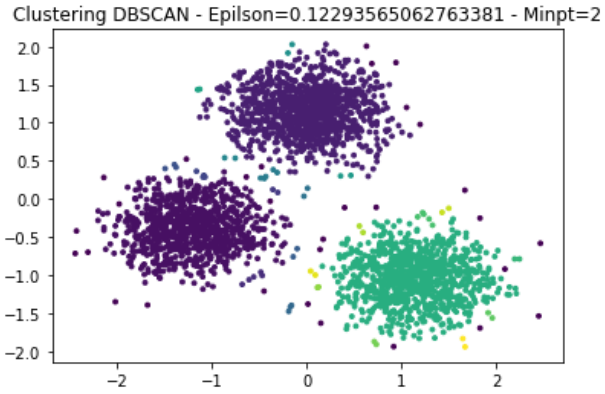
Application du code qui calcule Epsilon optimale sur xclara :

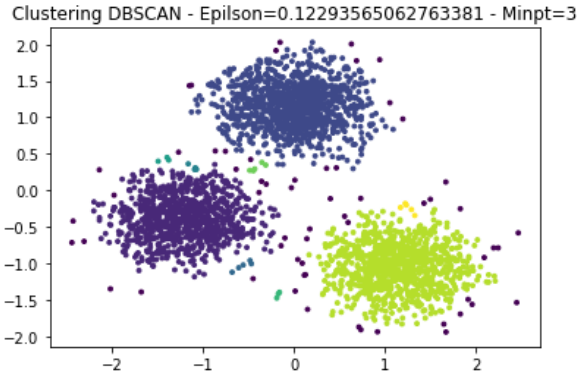
Le point d'intersection vert nous donne l'Epsilon optimale (y) :

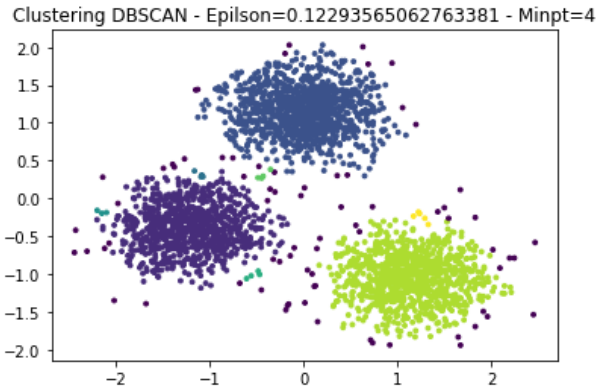


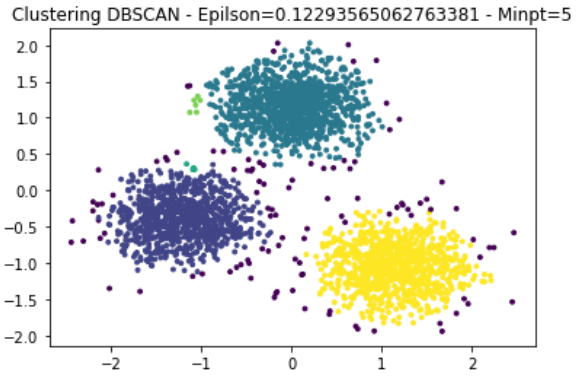

```
Affichage données standardisées
Knee value: x=2970 , y=0.12293565062763381
Estimated number of clusters: 24
Estimated number of noise points: 30
Estimated number of clusters: 9
Estimated number of noise points: 60
Estimated number of clusters: 8
Estimated number of noise points: 74
Estimated number of clusters: 5
Estimated number of noise points: 106
Estimated number of clusters: 3
Estimated number of noise points: 129
Estimated number of clusters: 4
Estimated number of noise points: 152
Estimated number of clusters: 4
Estimated number of noise points: 179
Estimated number of clusters: 3
Estimated number of noise points: 205
```

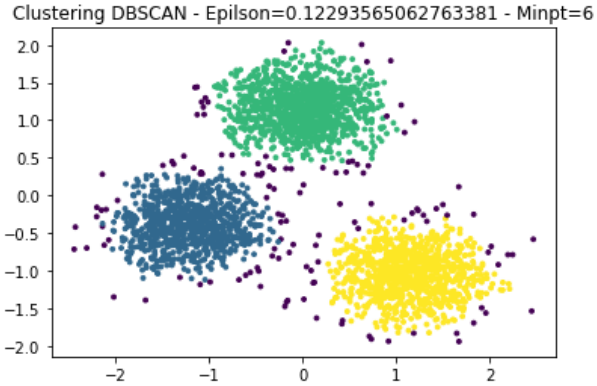
Notre code applique maintenant pour la valeur optimale d'Epsilon différentes valeur de Min_sample :

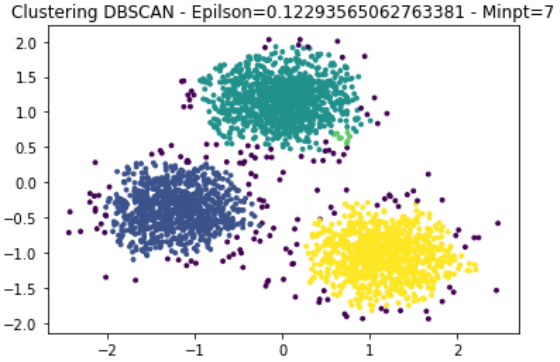


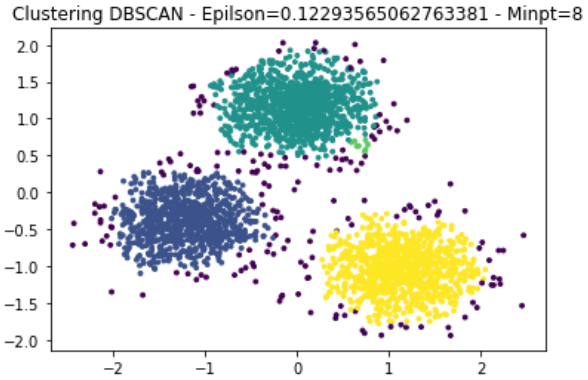


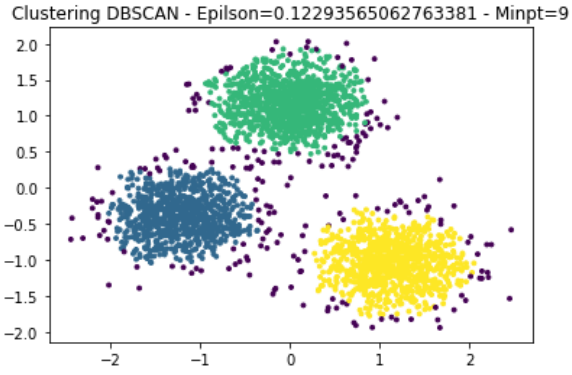












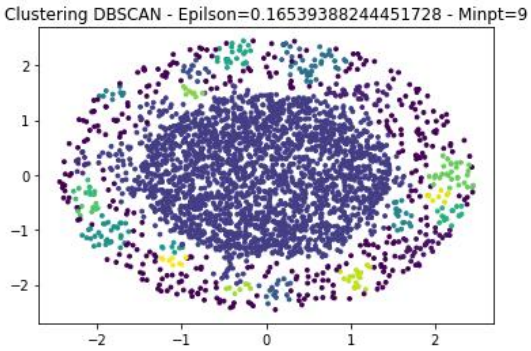
Les deux meilleurs résultats sont ceux avec min_sample = 6 et 9 avec 3 clusters et du bruit détecté.

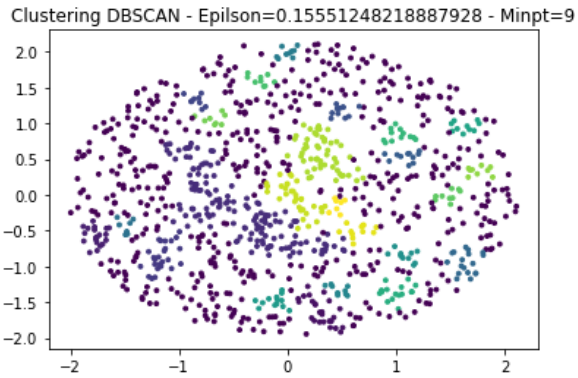
Jeux de données avec lesquels DBSCAN est inefficace

1- Dense-disk-3000

2- disk_1000n

Les deux jeux de données ont un nombre d'échantillons très important ce qui doit montrer les limites de DBSCAN ;

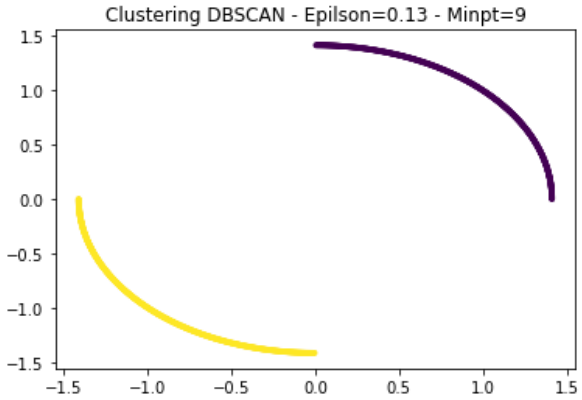


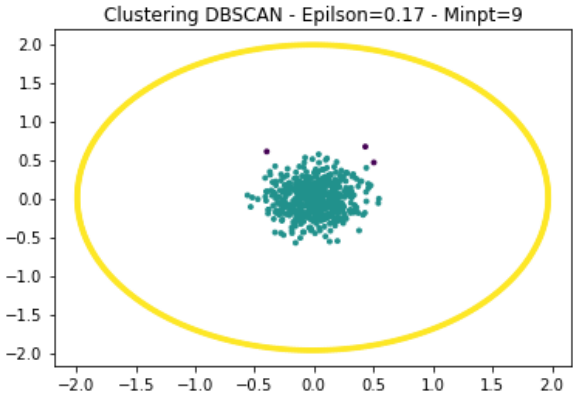


Jeux de données avec lesquels DBSCAN est efficace

1- Curves1

2- Donut1





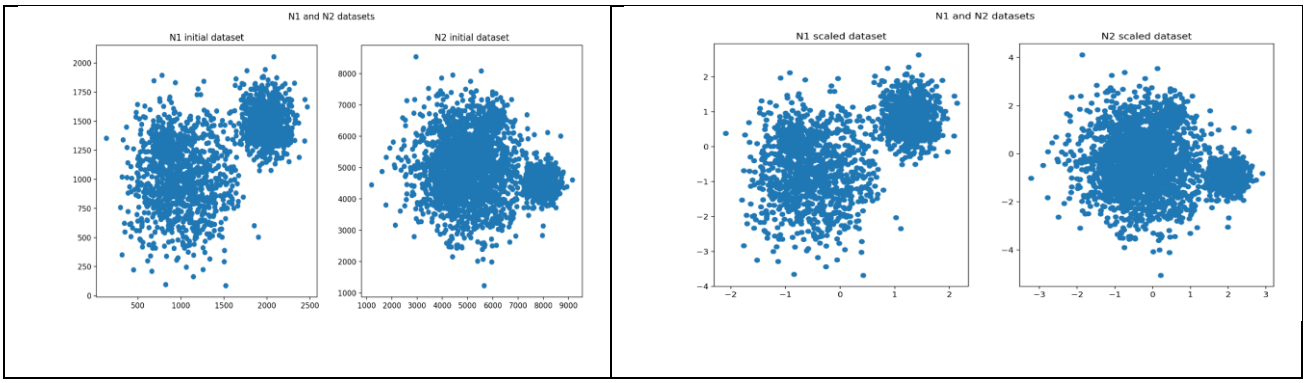
Avantages et inconvénients de DBSCAN

Avantages	Inconvénients
Pas besoin de renseigner K à l'avance	Elle est Incapable de gérer les clusters à densités différentes
Elle gère les anomalies et les données aberrantes	Pour des données de grande dimension Epsilon et min_sample deviennent très difficile à estimer
Temps de calcul	
Prend en charge les formes aléatoires	

Partie TP3 / Deux types de données à traiter

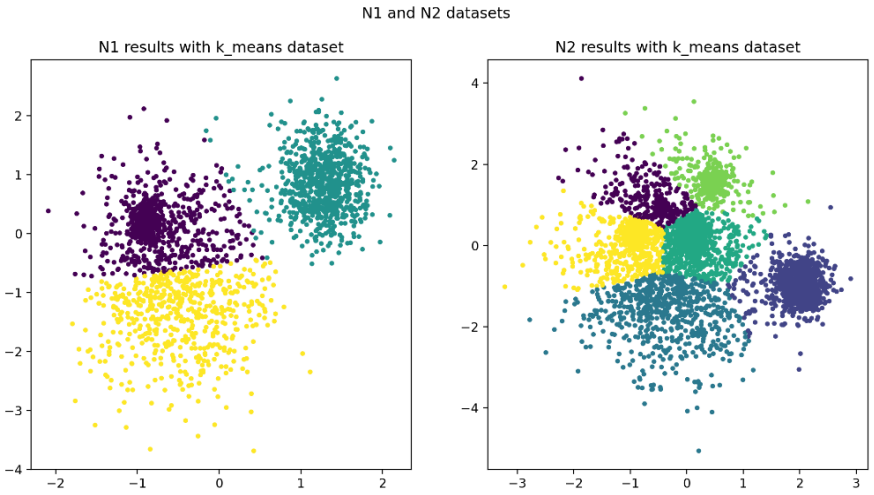
Affichage des deux data sets

Les figures suivantes représentent les deux data sets avant et après la standardisation :

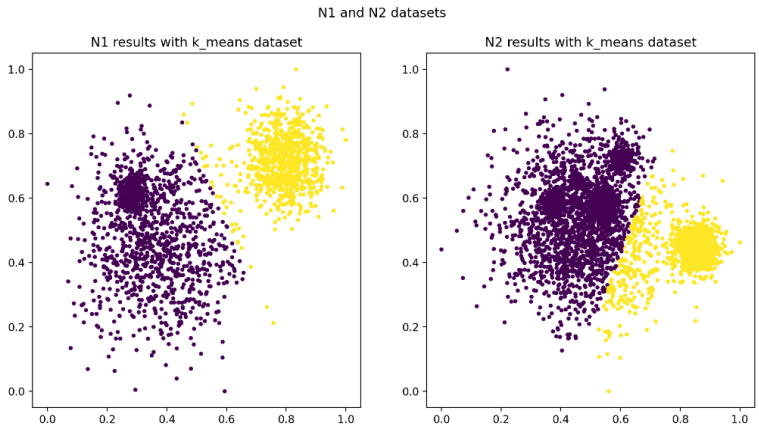


K-Means

Nous appliquons ici K-means sur les jeux de données :



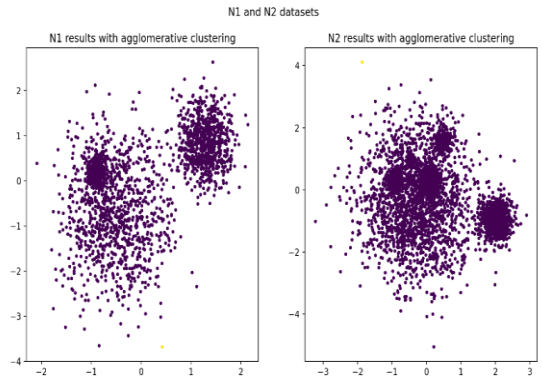
La détermination automatique du k étant mauvaise, on passe sur un preprocessing en MinMax dont les résultats sont les suivants :

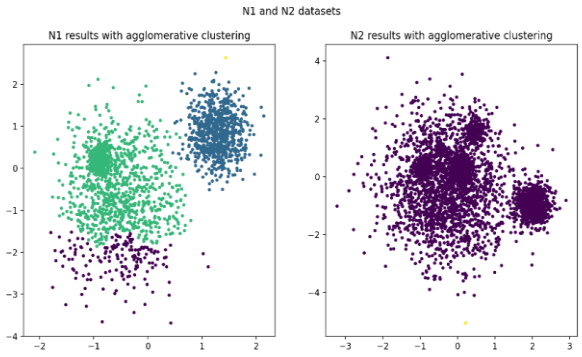


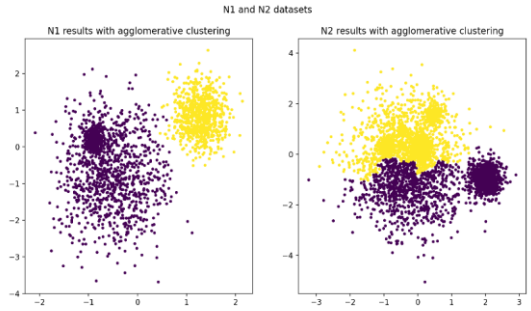
Agglomération

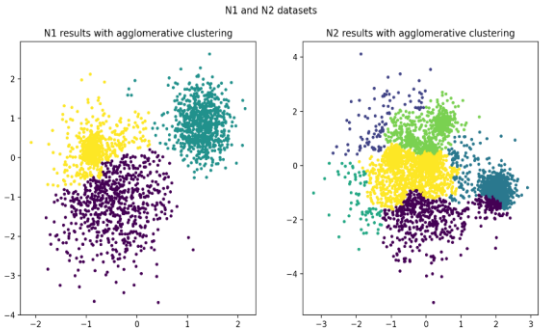
Nous appliquons ici la méthode d'agglomération en jouant sur les paramètres de la méthode :

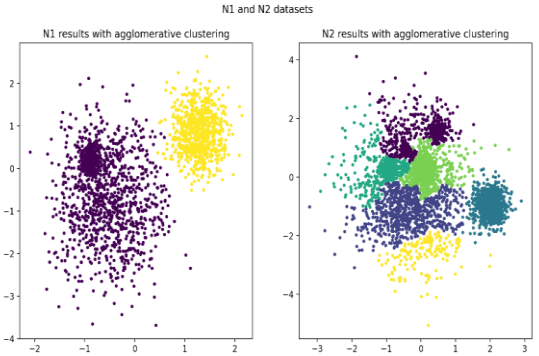
Linkage, Average, Ward fixe, Complete, Automatique k Ward Sans preprocessing, Avec preprocessing



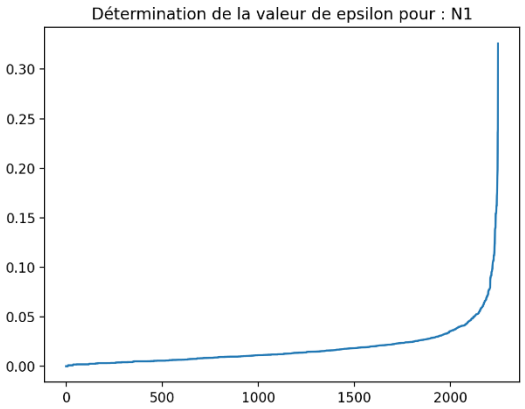


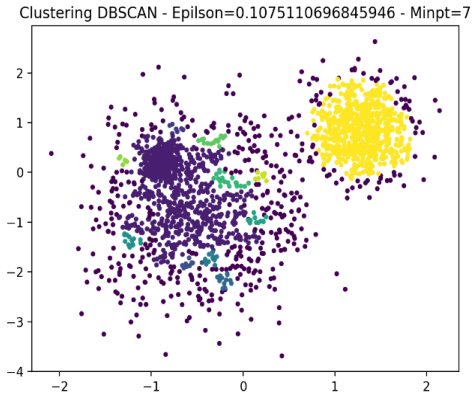






DBSCAN

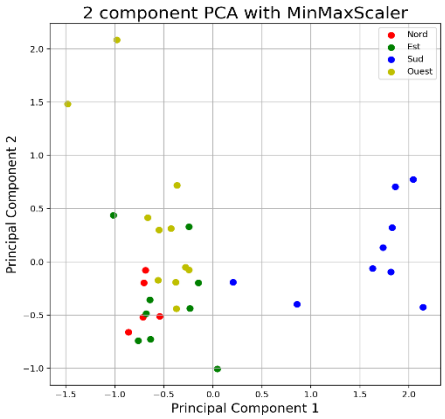




Application du PCA :

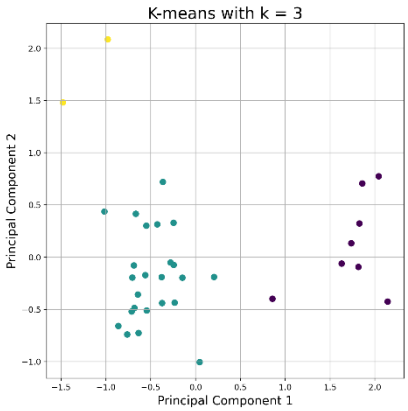
Nous appliquons le PCA après avoir standardisé les données avec MinMax scaler :

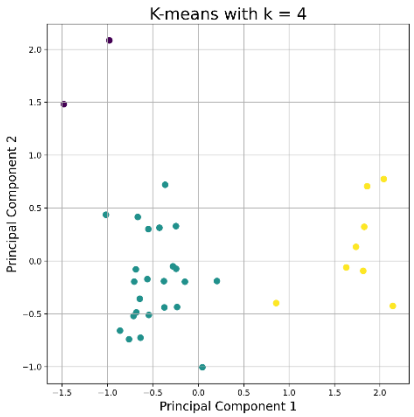
PCA est utilisé pour réduire la dimension du jeu de donnée afin d'en garder que les échantillons les plus importants et ainsi avoir un clustering plus stable.



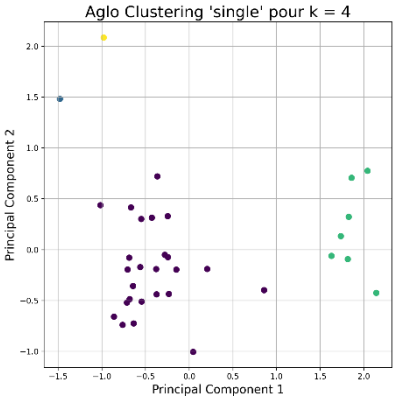
Application de K-means :

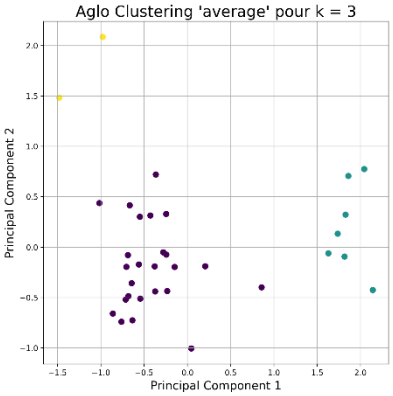
Nous appliquons la méthode sur les données avec k auto 3 et k forcé 4 et nous obtenons ceci :

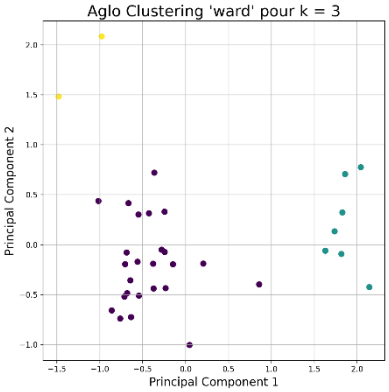


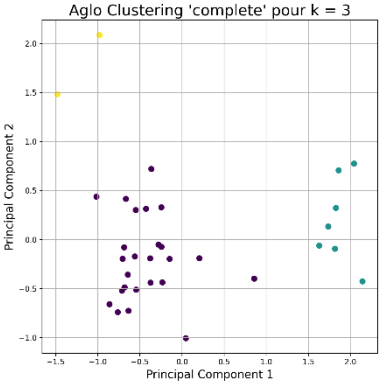


Application d'agglomération avec variation des paramètres









Application de DBSCAN