MEGA Adrian 5SDBD

RABARY Anaïs

TP Apprentissage Non Supervisé

# Introduction

Ce rapport a pour but de rendre compte de nos travaux en TP d’apprentissage non supervisé. Les deux chapitres traiteront des deux parties du TP, respectivement sur DBSCAN et SNN. Le code peut être trouvé sur le dépôt suivant

<https://github.com/dx07/5SDBD-Apprentissage/tree/master/Apprentissage-Non-Supervise>

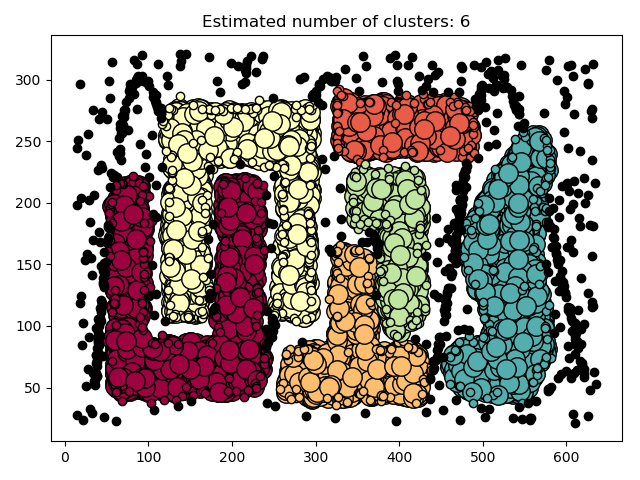
Nous parlerons dans ce rapport des résultats de nos algorithmes de clustering sur les 4 jeux de données proposés. Il sera question d’appliquer les algorithmes développés et d’étudier les paramétrages, la qualité du clustering, etc…

# I – DBSCAN de scikit-learn

DBSCAN présente l’intérêt de ne pas obliger de fixer le nombre de clusters à l’avance, tout en étant robuste au bruit et aux anomalies. De plus, il va nous permettre d’identifier des clusters concaves et d’apprécier leur qualité. En faisant varier les paramètres de la fonction DBSCAN de scikit-learn, nous allons tenter d’affiner la qualité de nos clusters.

Pour faire varier ces paramètres, nous avons tout d’abord testé plusieurs valeurs à la main par petites variations (jusqu’à manipuler des valeurs d’epsilon à décimales). Nous avons ensuite testé des valeurs en boucle pour tenter d’obtenir des clustering différents et de meilleure qualité.

## Dataset 1



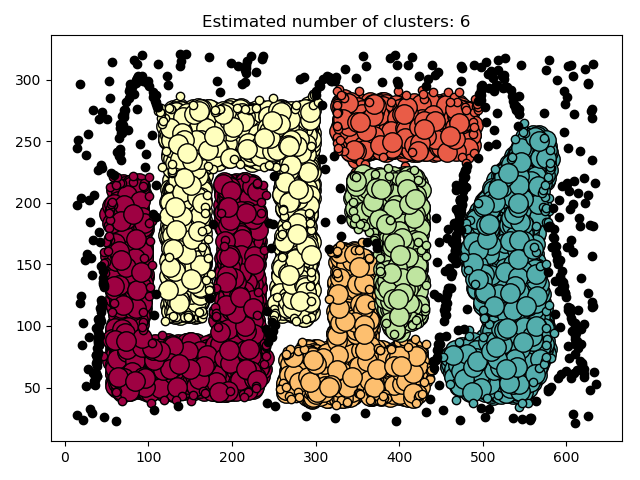
Epsilon : 10

Min Samples : 18

Estimated number of clusters : 6

Estimated number of noise points : 630

Silhouette Coefficient : 0.246



Epsilon : 12

Min Samples : 22

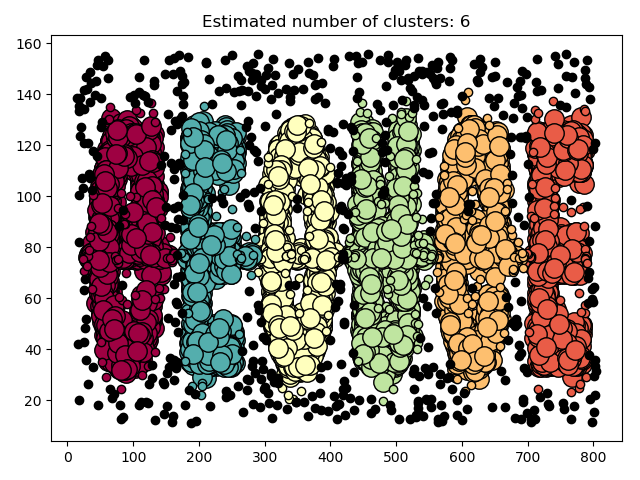
Estimated number of clusters : 6

Estimated number of noise points : 543

Silhouette Coefficient : 0.254

Pour ces deux tests, nous obtenons bien le nombre de 6 clusters, identifiables à vue d’œil. Le coefficient de silhouette reste égal à plus ou moins 0.25, cela étant la forme de nos clusters est tout à fait satisfaisante. Le bruit de sinusoïde est (légèrement) davantage ignoré sur le premier schéma (e = 10, ms = 18). On considèrera le bruit plus faible (543), cela étant il n’est pas nécessaire de chercher à le diminuer. Nos clusters doivent rester cohérents.

## Dataset 2



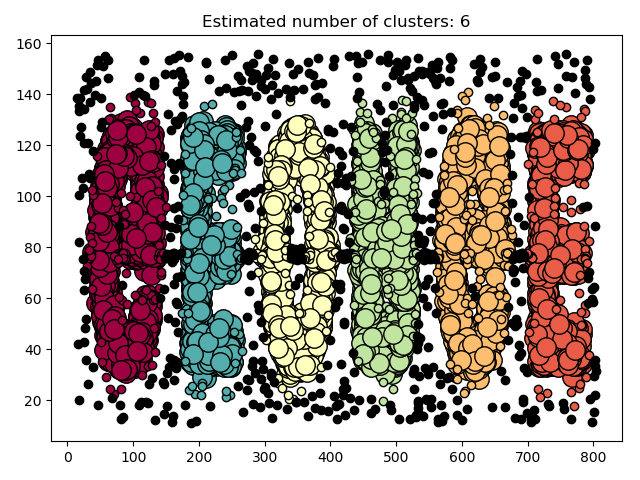
Epsilon : 9.8

Min Samples : 18

Estimated number of clusters : 6

Estimated number of noise points : 566

Silhouette Coefficient : 0.490



Epsilon : 12

Min Samples : 38

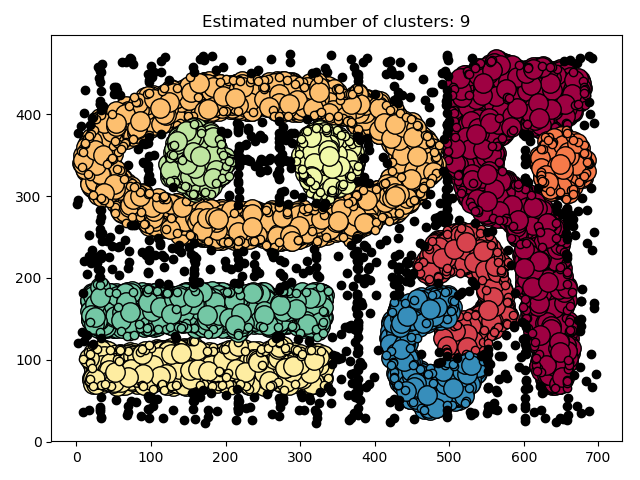
Estimated number of clusters : 6

Estimated number of noise points : 630

Silhouette Coefficient : 0.486

Sur ce second dataset, nous avons un meilleur coefficient silhouette avec des paramètres plus faibles. Sur le deuxième test, on peut constater le bruit au milieu du graphe (barre horizontale), qui est pourtant assimilé aux clusters sur le premier test. Il semble plus logique d’ignorer ce bruit donc nous considérons les seconds paramètres plus pertinents.

## Dataset 3



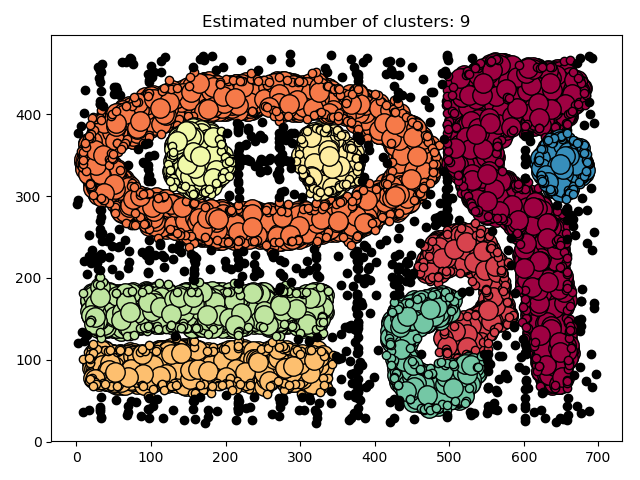
Epsilon : 10

Min Samples : 13

Estimated number of clusters : 9

Estimated number of noise points : 774

Silhouette Coefficient : -0.068



Epsilon : 14

Min Samples : 26

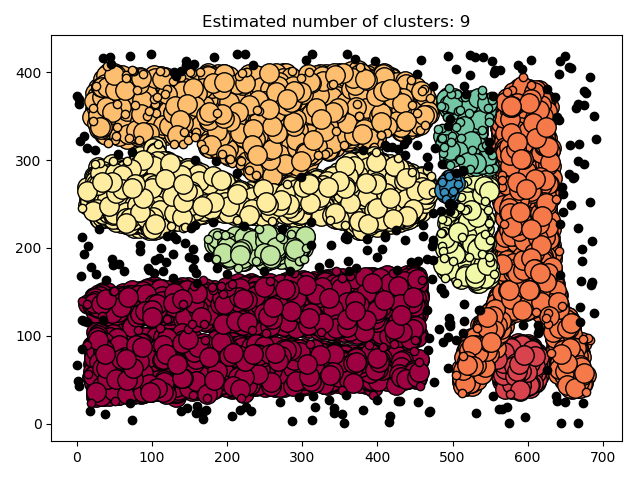
Estimated number of clusters : 9

Estimated number of noise points : 660

Silhouette Coefficient : -0.062

Très peu de différences sont visibles sur ces deux tests du troisième dataset (hormis la couleur des clusters …). Le bruit symbolisé par les lignes verticales parallèles est peu pris en compte (davantage sur le test 2). Notre clustering reste cohérent malgré une différence d’une centaine de points de bruits, qui n’ont pas l’air d’affecter grandement la qualité des clusters.

## Dataset 4



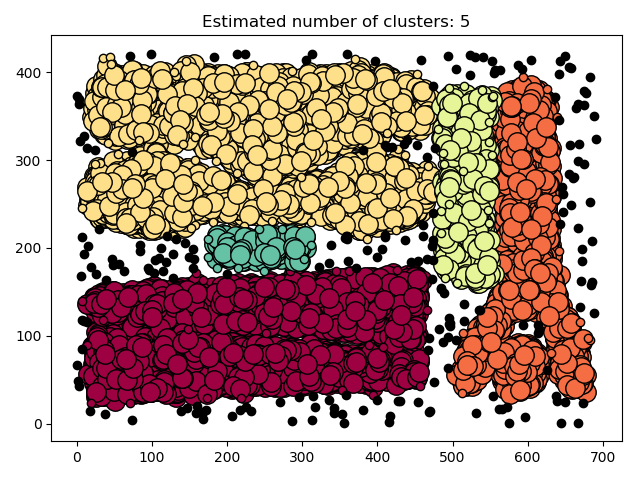
Epsilon : 10.1

Min Samples : 7

Estimated number of clusters : 9

Estimated number of noise points : 277

Silhouette Coefficient : -0.078



Epsilon : 12

Min Samples : 6

Estimated number of clusters : 5

Estimated number of noise points : 215

Silhouette Coefficient : -0.007

Il nous a été difficile d’identifier les bons paramètres pour ce dernier clustering sur le quatrième dataset. Certains clusters sont tellement proches qu’ils peuvent se retrouver groupés pour un epsilon trop grand. Nous pensons tout de même que notre premier clustering est plus pertinent car il distingue davantage de clusters que le second (plus cohérents à vue d’œil). Toutefois nous aurions aimé trouver un paramétrage pour distinguer les deux pavés mauves en bas à gauche.

# II – Algorithme SNN, les k plus proches voisins