DBSCAN

01 May 2022 21:03

Density Based Spatial Clustering of Applications with Noise(DBSCAN)

First, let's clear up the role of clustering.

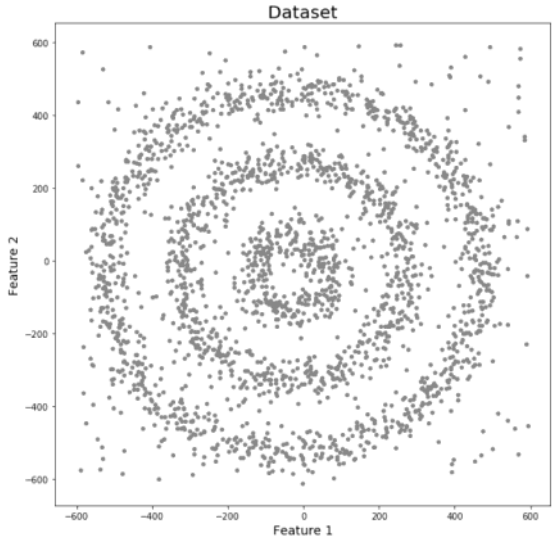
Clustering is an **unsupervised learning** technique where we try to group the data points based on specific characteristics. There are various clustering algorithms with **K-Means** and **Hierarchical** being the most used ones.
Some of the use cases of clustering algorithms include:

- Document Clustering
- Recommendation Engine
- Image Segmentation
- Market Segmentation
- Search Result Grouping
- and Anomaly Detection.

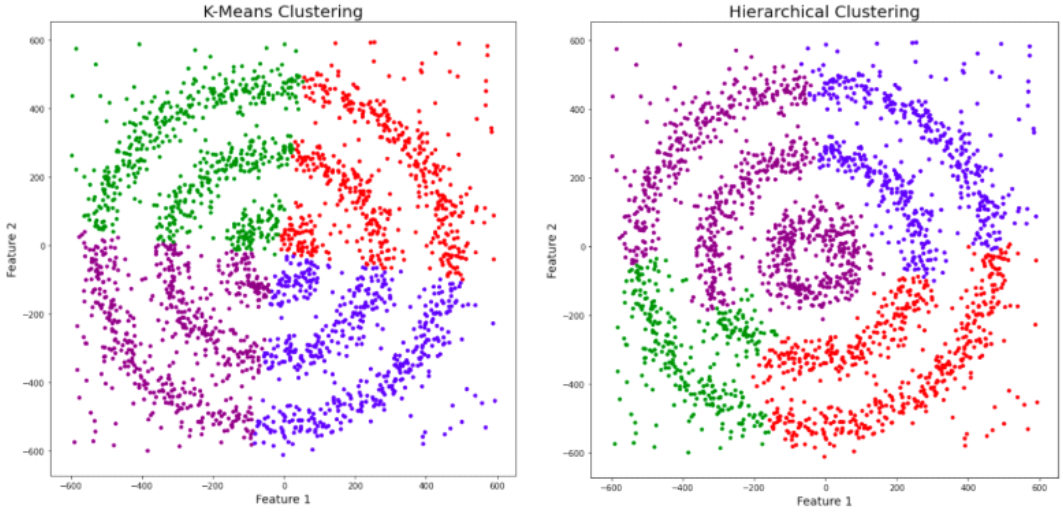
All these problems use the concept of clustering to reach their end goal. Therefore, it is crucial to understand the concept of clustering. But here's the issue with these two clustering algorithms.

K-Means and Hierarchical Clustering both fail in creating clusters of arbitrary shapes. They are not able to form clusters based on varying densities. That's why we need DBSCAN clustering.

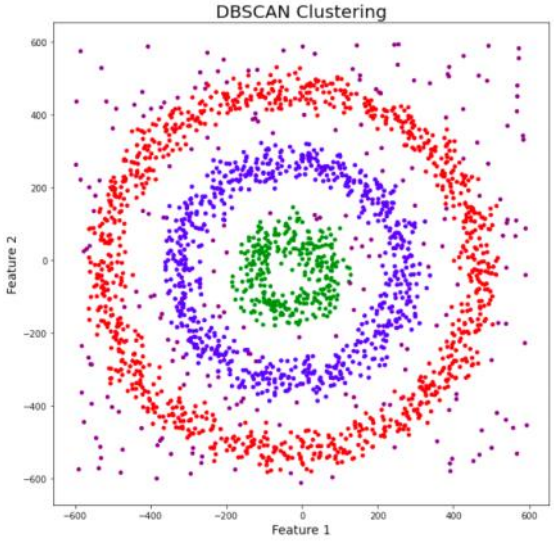
Let's try to understand it with an example. Here we have data points densely present in the form of concentric circles:



We can see three different dense clusters in the form of concentric circles with some noise here. Now, let's run K-Means and Hierarchical clustering algorithms and see how they cluster these data points.



You might be wondering why there are four colors in the graph? As I said earlier, this data contains noise too, therefore, I have taken noise as a different cluster which is represented by the purple color. Sadly, both of them failed to cluster the data points. Also, they were not able to properly detect the noise present in the dataset. Now, let's take a look at the results from DBSCAN clustering.



Awesome! DBSCAN is not just able to cluster the data points correctly, but it also perfectly detects noise in the dataset.

- DBSCAN is a density-based clustering algorithm that works on the assumption that clusters are dense regions in space separated by regions of lower density.
- It groups 'densely grouped' data points into a single cluster. It can identify clusters in large spatial datasets by looking at the local density of the data points.

- The most exciting feature of DBSCAN clustering is that it is robust to outliers. It also does not require the number of clusters to be told beforehand, unlike K-Means, where we have to specify the number of centroids.

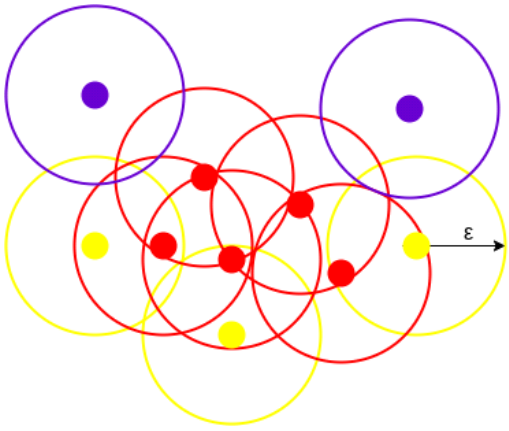
- DBSCAN requires only two parameters: **epsilon and min Points**. **Epsilon** is the radius of the circle to be created around each data point to check the density and **min Points** is the minimum number of data points required inside that circle for that data point to be classified as a Core point.

- Here, we have some data points represented by grey color. Let's see how DBSCAN clusters these data points.

- **DBSCAN** creates a circle of epsilon radius around every data point and classifies them into Core point, Border point, and Noise.

- A data point is a **Core point** if the circle around it contains at least 'min Points' number of points.

- If the number of points is less than min Points, then it is classified as **Border Point**, and if there are no other data points around any data point within epsilon radius, then it treated as **Noise**.



- The above figure shows us a cluster created by DBCAN with min Points = 3. Here, we draw a circle of equal radius epsilon around every data point. These two parameters help in creating spatial clusters.

- All the data points with at least 3 points in the circle including itself are considered as **Core points** represented by red color

- All the data points with less than 3 but greater than 1 point in the circle including itself are considered as **Border points**. They are represented by yellow color.

- Finally, data points with no point other than itself present inside the circle are considered as **Noise** represented by the purple color.

Pros and Cons of DBSCAN

Pros:

- Does not require to specify number of clusters beforehand.
- Performs well with arbitrary shapes clusters.
- DBSCAN is robust to outliers and able to detect the outliers.

Cons:

- In some cases, determining an appropriate distance of neighbour hood (eps) is not easy and it requires domain knowledge.
- If clusters are very different in terms of in-cluster densities, DBSCAN is not well suited to define clusters. The characteristics of clusters are defined by the combination of eps-min Pts parameters. Since we pass in one eps-min Pts combination to the algorithm, it cannot generalize well to clusters with much different densities.