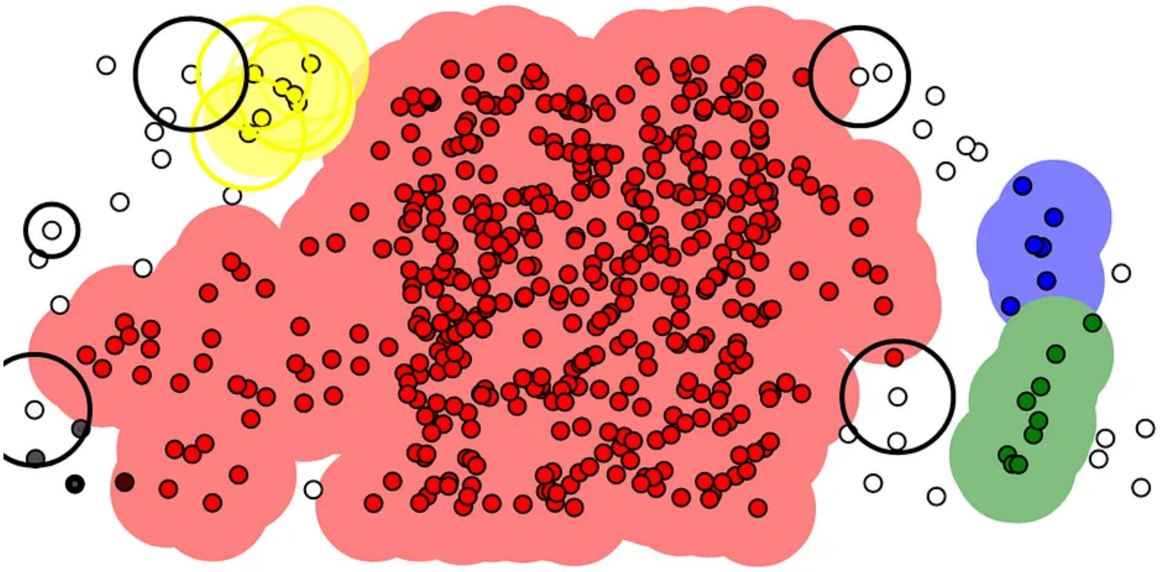
Clustering Like a Pro: A Beginner’s Guide to DBSCAN

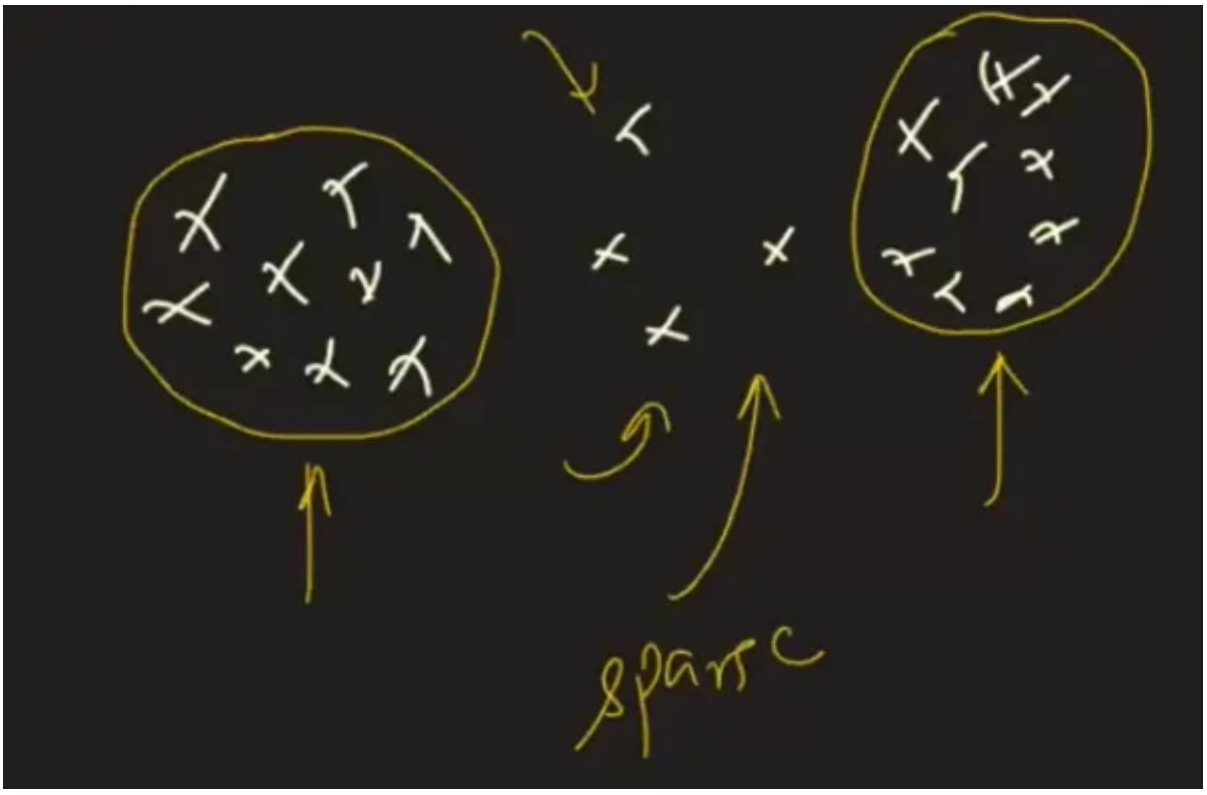
Sachinsoni Follow 7 min read · Dec 26, 2023

Data clustering is a fundamental task in machine learning and data analysis. One powerful technique that has gained prominence is Density-Based Spatial Clustering of Applications with Noise (DBSCAN). In this blog, we delve into the world of DBSCAN, exploring its principles and applications in uncovering hidden structures within datasets. Join us on a journey to understand how DBSCAN goes beyond traditional clustering methods, offering a unique approach to identifying clusters based on the density of data points. Let’s unravel the intricacies of DBSCAN and unlock its potential for unraveling patterns in your data.



Idea behind density based clustering :

Density based clustering algorithms divides your entire dataset into dense regions separated by sparse regions.



There are two popular algorithms that is based on the above idea which are :

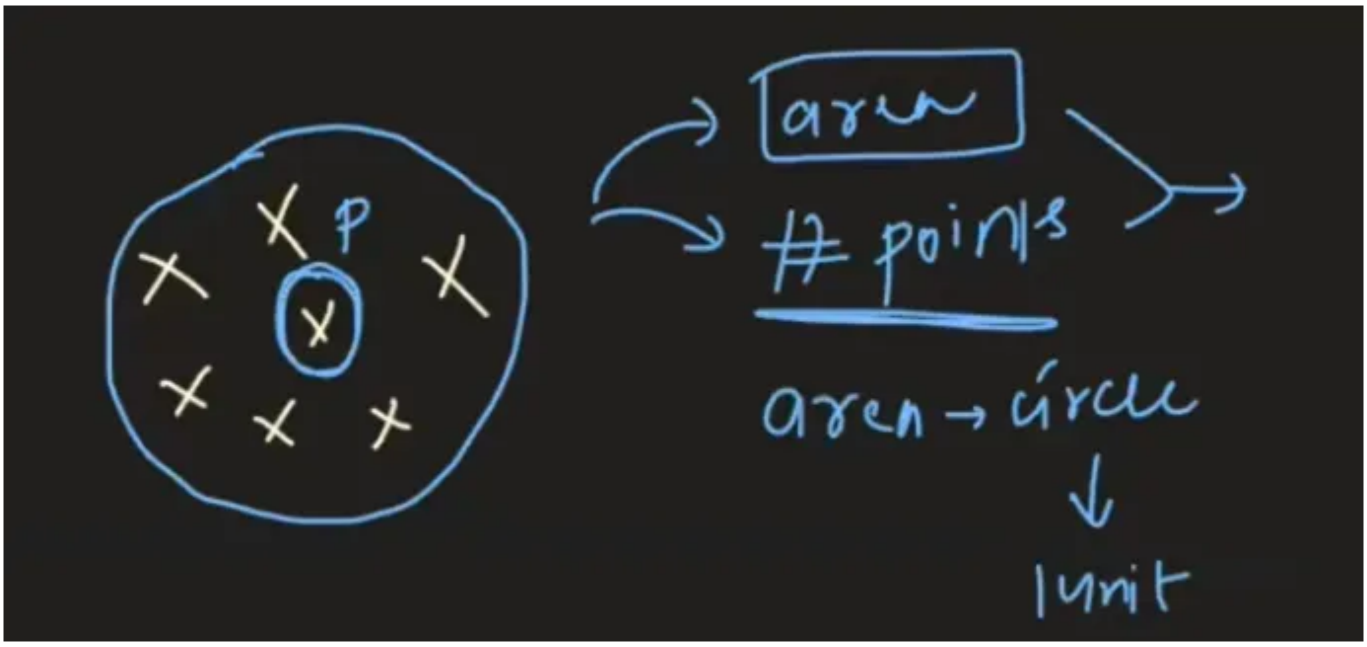
a. DBSCAN
b. OPTICS

In this blog, we will discuss about DBSCAN in brief and will try to understand why this algorithm works better than KMeans clustering algorithm. Before starting the DBSCAN, let's understand some basic terms which are used in DBSCAN.

MinPts and Epsilon :

> *How to measure density around a point ?*

Measuring density around a point is straightforward — **we define a region around the point and assess the number of points within that designated area.** This approach serves as a practical method for gauging the density surrounding a specific point.



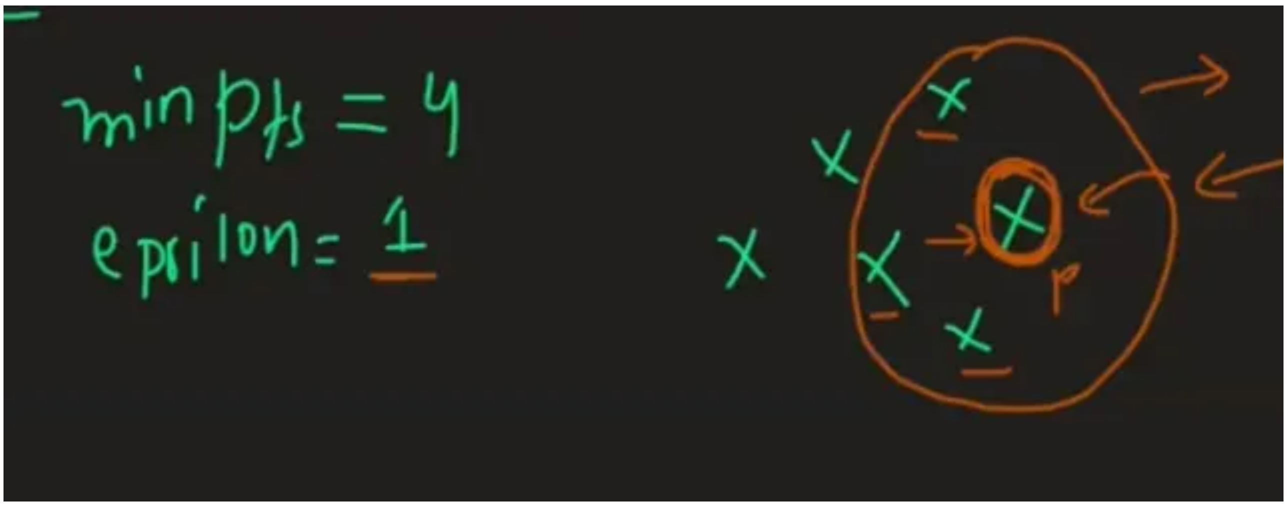
To determine density around a point, we employ circles in 2-D, spheres in 3-D, and hyper-spheres in n-dimensional spaces. Suppose we draw unit radius circle around a point P as shown in above figure and here we establish a criterion: a region is considered sparse if it contains fewer than 3 points and dense if it contains 3 or more points.

MinPts stands for "Minimum Points", is a parameter that specifies the minimum number of points required to form a dense region, which is consider a cluster.
While Epsilon is a key parameter that defines the radius of the neighborhood around a given data point. Specifically, epsilon is the maximum distance between two points for them to be considered as part of the same neighborhood.

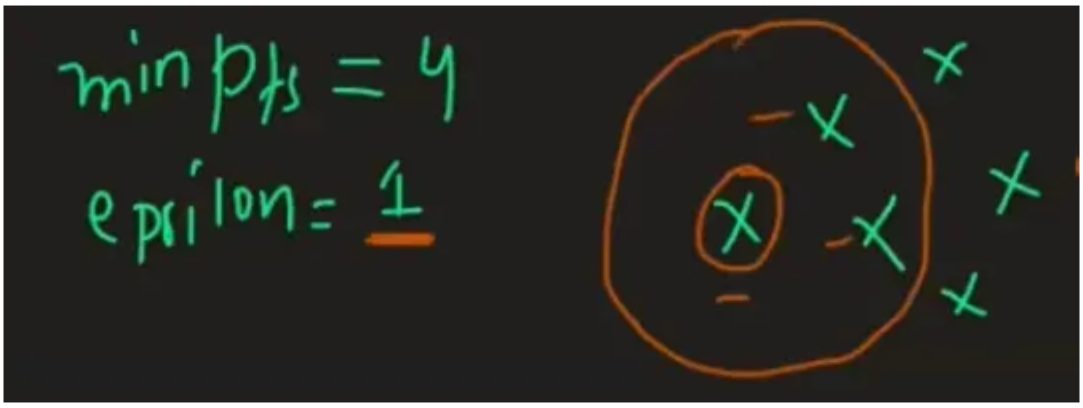
In the above example, the chosen radius value of 1 unit corresponds to epsilon, while the minimum point threshold of 3, determining sparse or dense regions, is representative of MinPts. Both MinPts and Epsilon are hyperparameters that necessitate fine-tuning to achieve optimal results.

Core Points, Border Points and Noise Points :

A point is considered a core point if it has a minimum number of other points(specified by MinPts) within a given radius ε of itself.



In the depicted diagram, with ε set to 1 and MinPts to 4, let's focus on a specific point, P. To determine if P qualifies as a core point, we create a circle with a radius of 1 unit around P. Observing the diagram, it's evident that point P, along with three additional points within the circle, satisfies the MinPts condition. Hence, we can confidently classify point P as a core point.

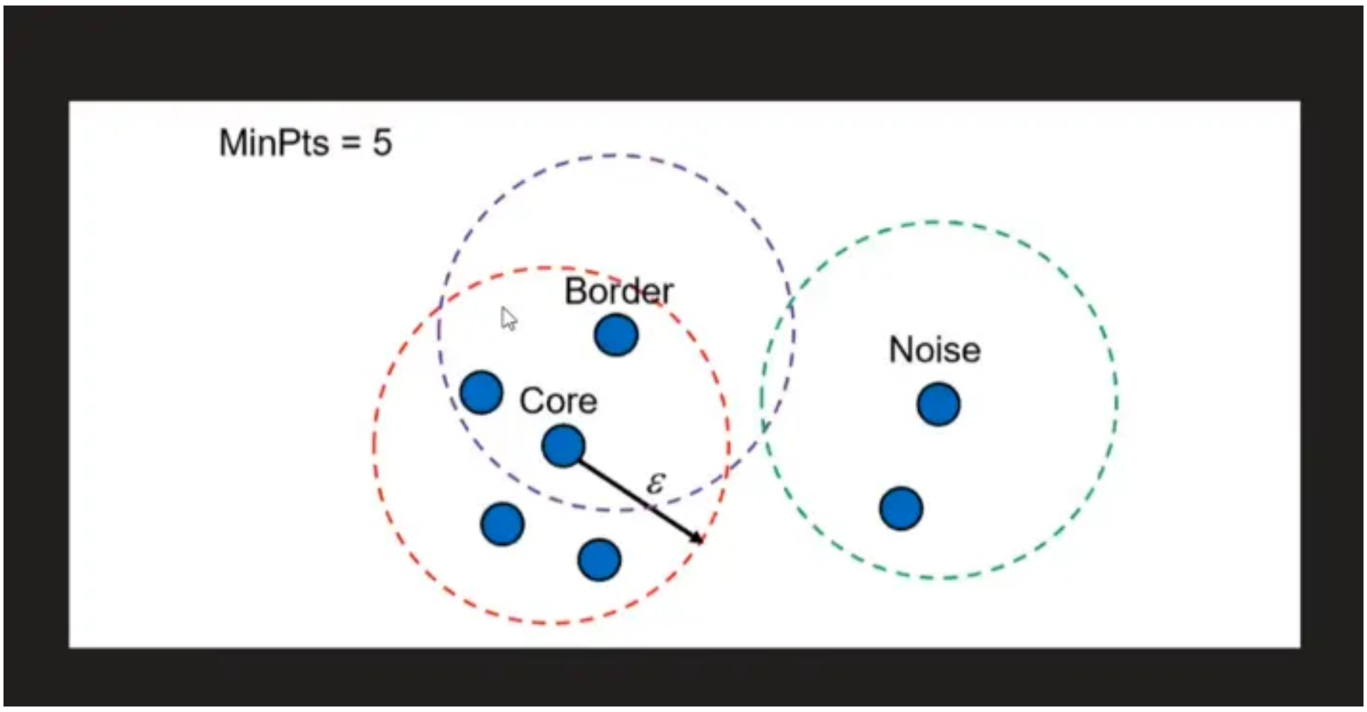


Examining the diagram, it's evident that within the circle surrounding a specific point, there are only two points in addition to the point itself, totaling three points. This doesn't meet the MinPts requirement of 4, leading us to conclude that it is not a core point.

Border Point :

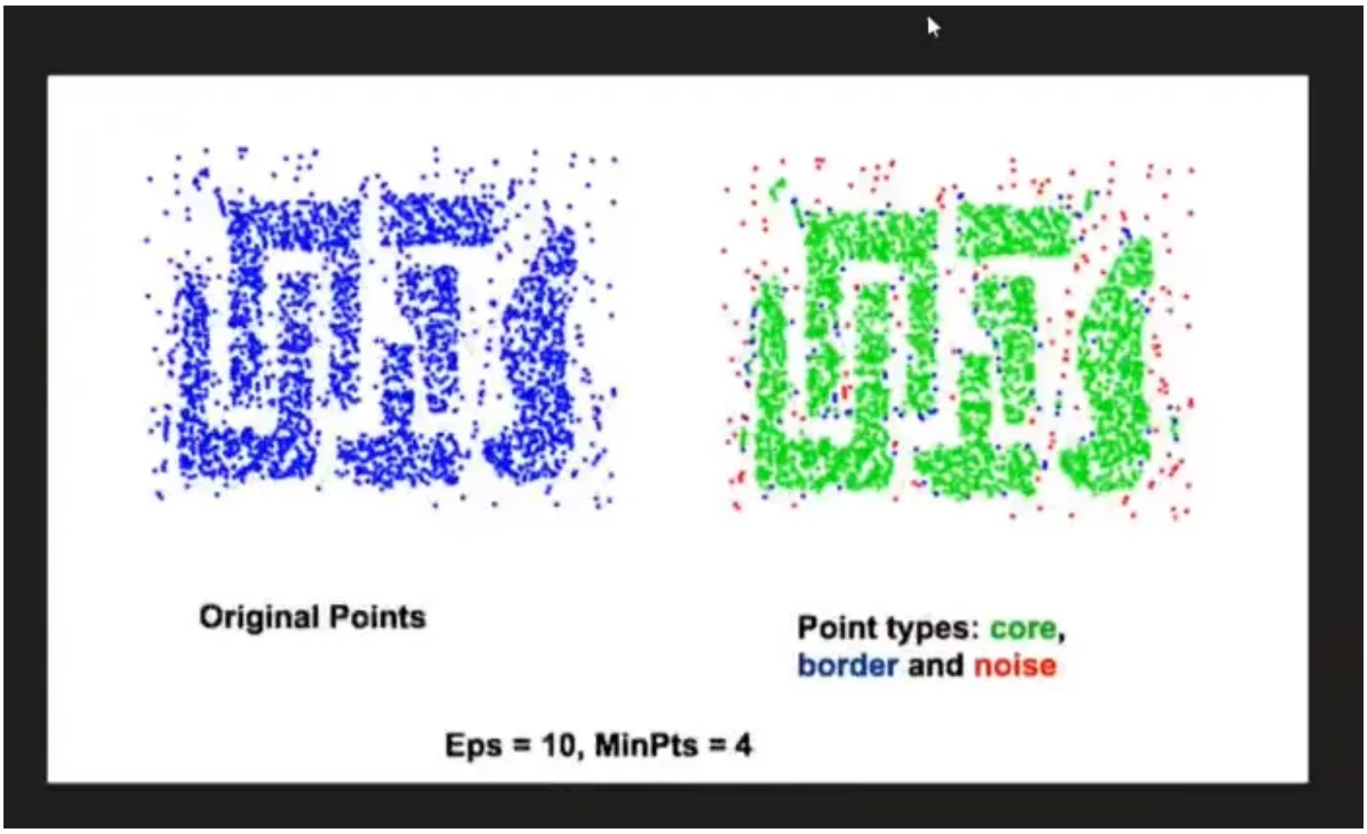
A border point is defined as follows:

- **Not a Core Point:** A border point does not meet the criteria to be a core point. It has fewer than MinPts within its ε-neighbourhood.
- **Neighbor of a Core Point: A border point is within the ε distance of one or more core points.** In other words, it lies on the edge of a cluster, within the radius ε of at least one core point.



Noise Point :

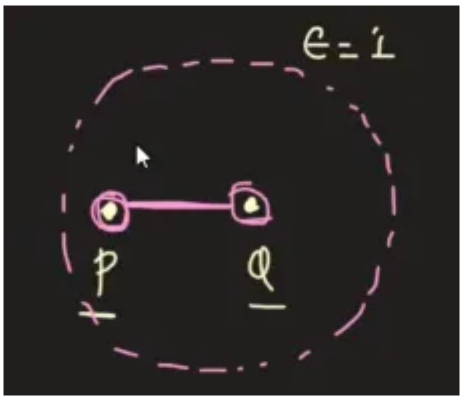
A noise point is a data point which can neither a core point nor a border point.



Visualization of Core points, Border points and Noise Points (Photo by CampusX)

Directly Density Reachable :

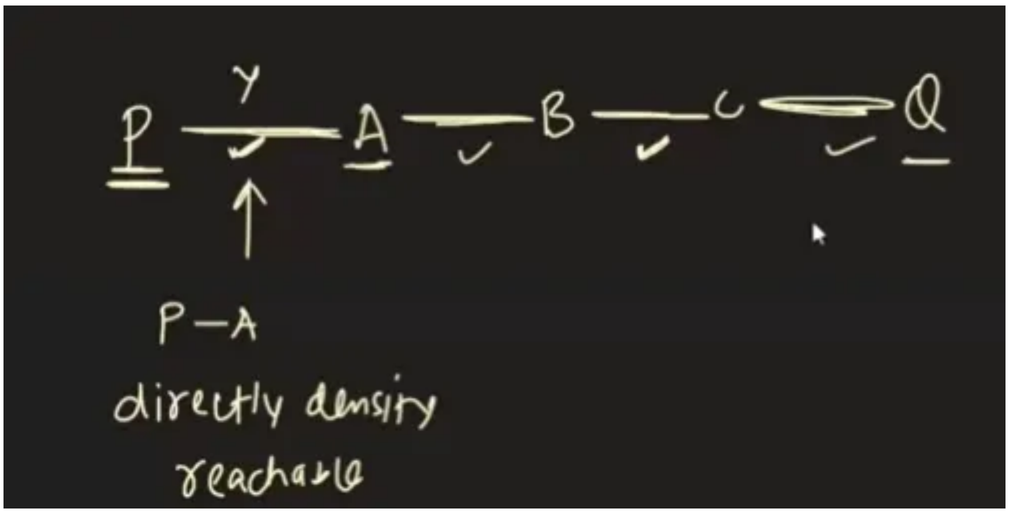
A point P is directly density-reachable from a point Q given Eps, MinPts if:
1. P is in the Eps-neighborhood of Q
2. Both P and Q are core points



Both P and Q are Core points and P lies in the Eps-neighborhood of Q

Density Connected Points

A point P is density connected to Q given Eps, MinPts if there is a chain of points P1, P2, P3 …… Pn, P1 = P and Pn = Q such that Pi+1 is directly density reachable from Pi .



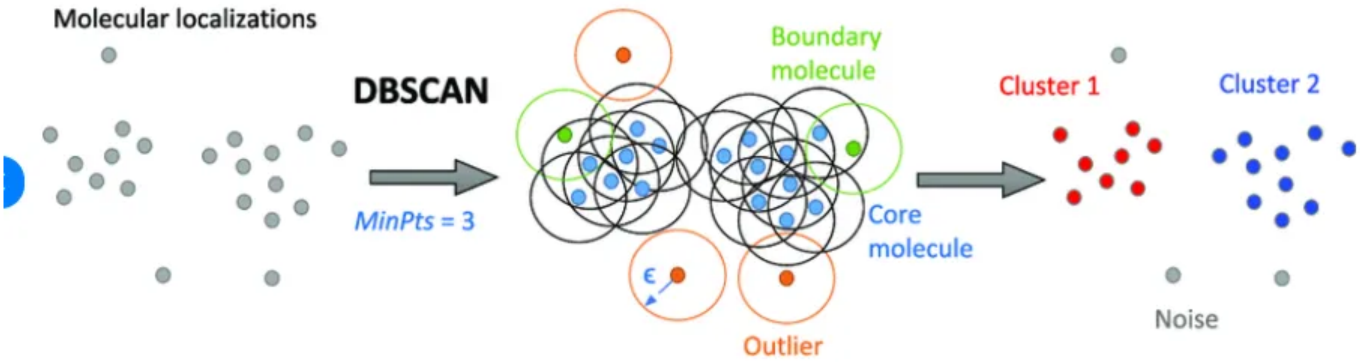
Simplified DBSCAN Algorithm

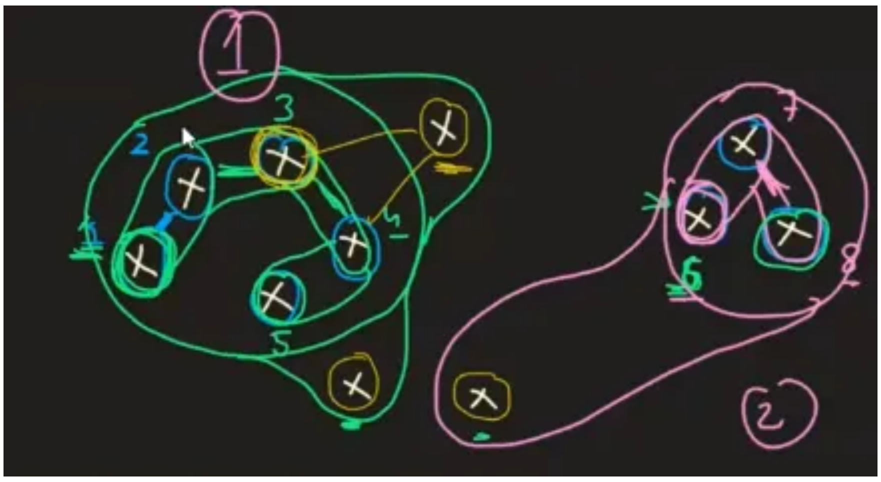
Step 1 — Identify all points as either core point, border point or noise point.

Step 2 — For all of the unclustered core points.

Step 2a — Create a new cluster.

Step 2b — add all the points that are unclustered and density connected to the current point into this cluster.





Step 3 — For each unclustered border point assign it to the cluster of nearest core point.

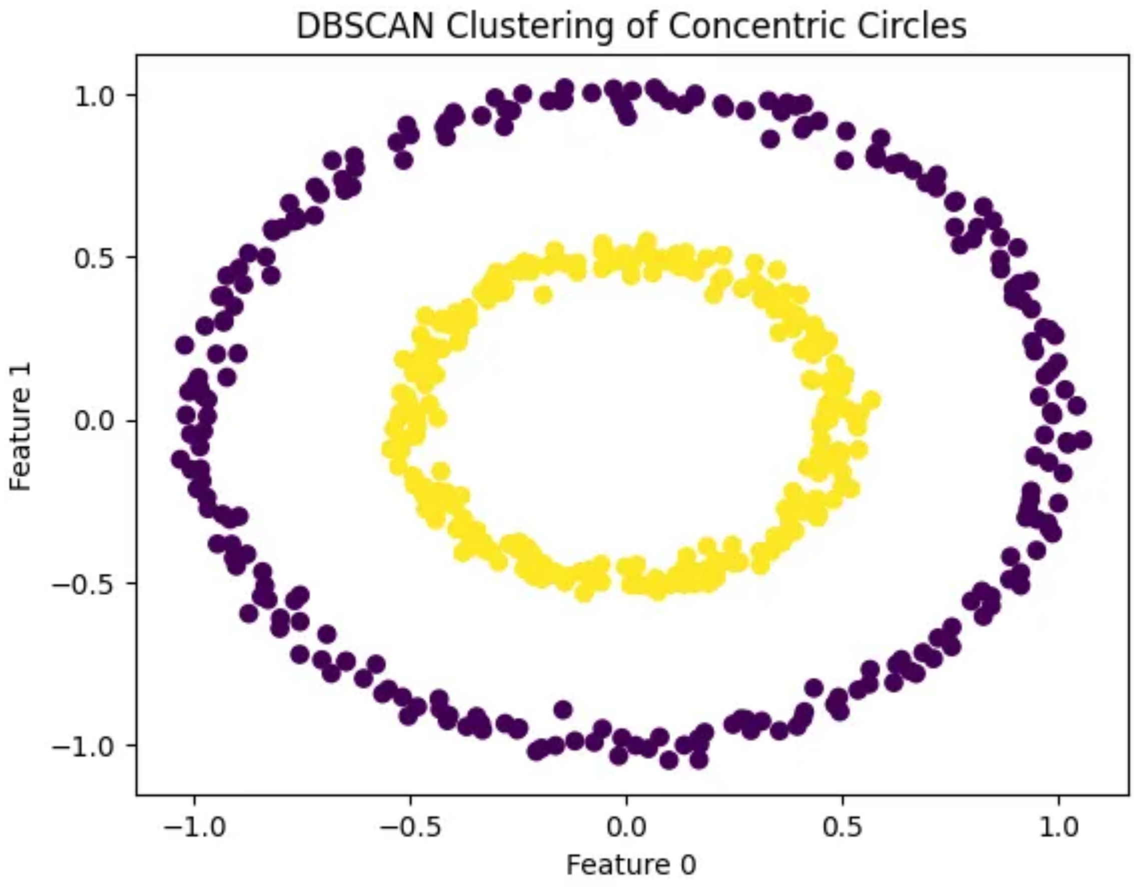
Step 4 — Leave all the noise points as it is.

Video Animation of working of DBSCAN :

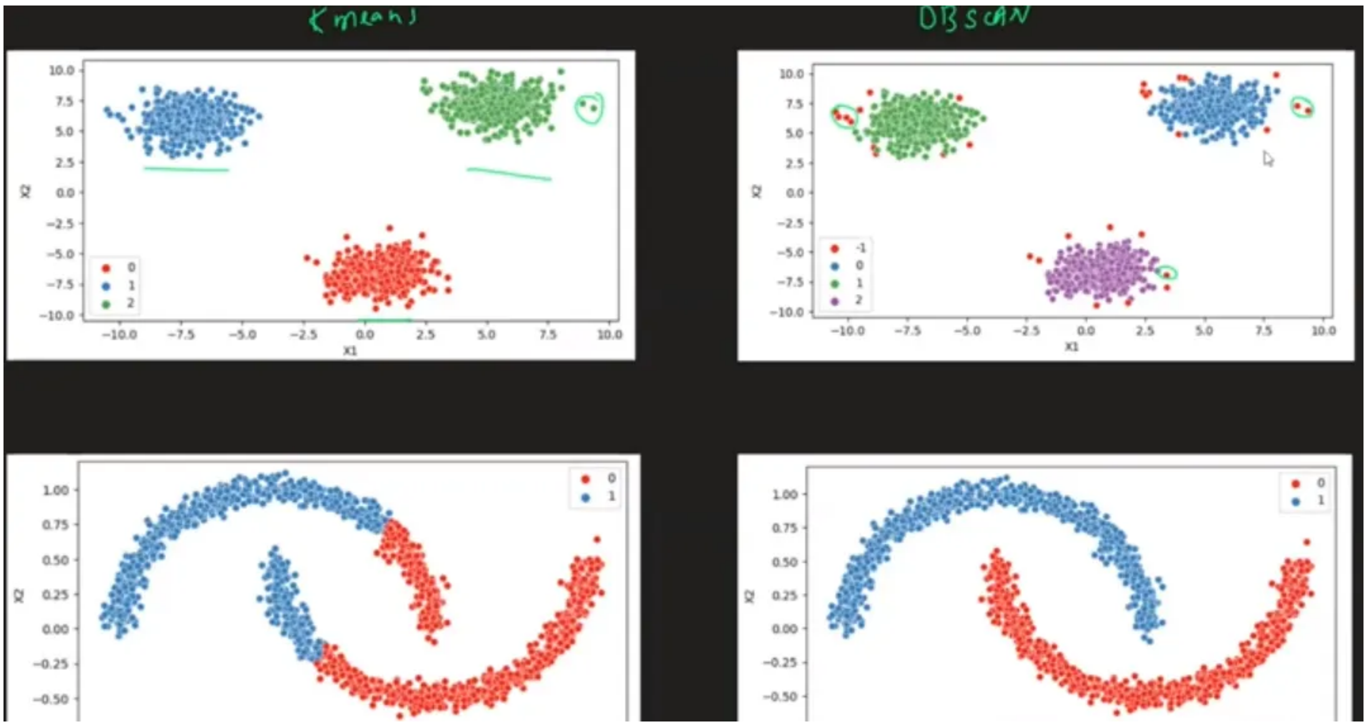
Code implementation of DBSCAN :

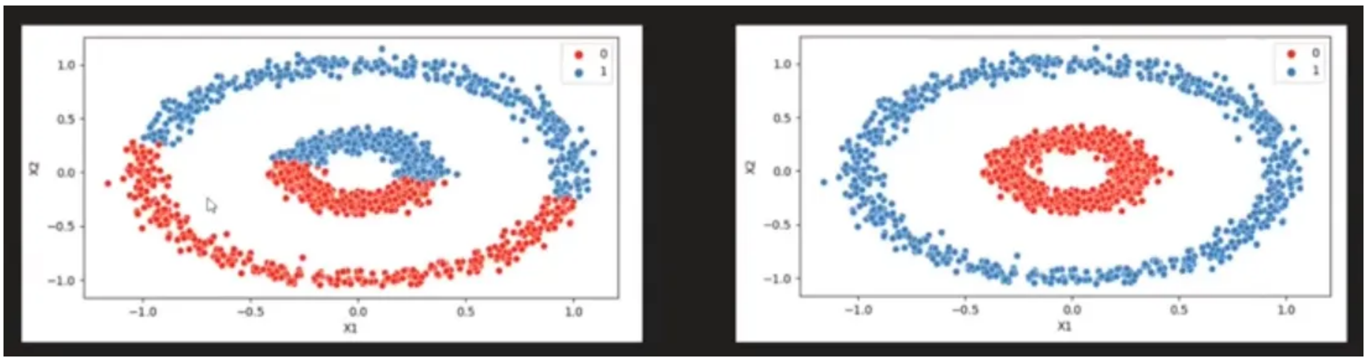
```
import matplotlib.pyplot as plt
from sklearn.datasets import make_circles
from sklearn.cluster import DBSCAN
import numpy as np

# Create a concentric circle dataset
X, _ = make_circles(n_samples=500, factor=.5, noise=.03, random_state=4)

# Apply DBSCAN to the dataset
dbscan = DBSCAN(eps=0.1, min_samples=5)
clusters = dbscan.fit_predict(X)

# Plotting
plt.scatter(X[:, 0], X[:, 1], c=clusters, cmap='viridis', marker='o')
plt.title("DBSCAN Clustering of Concentric Circles")
plt.xlabel("Feature 0")
plt.ylabel("Feature 1")
plt.show()
```

and we got the following result,



KMeans vs DBSCAN comparisons on different datasets :





Advantages of DBSCAN :

1. **Robust to outliers :** It is robust to outliers as it defines clusters based on dense regions of data, and isolated points are treated as noise.

2. **No need to specify clusters :** Unlike some clustering algorithms, DBSCAN does not require the user to specify the number of clusters beforehand, making it more flexible and applicable to a variety of datasets.

3. **Can find arbitrary shaped clusters :** DBSCAN can identify clusters with complex shapes and is not constrained by assumptions of cluster shapes, making it suitable for data with irregular structures.

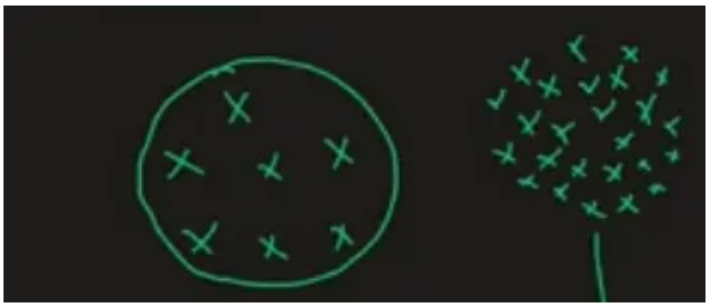
4. **Only 2 hyperparameters to tune :** DBSCAN has only two primary hyperparameters to tune: “eps” (distance threshold for defining neighborhood) and “min_samples” (minimum number of points required to form a dense region). This simplicity can make parameter tuning more straightforward.

Disadvantages of DBSCAN :

1. **Sensitivity to hyperparameters :** The performance of DBSCAN can be

sensitive to the choice of its hyperparameters, especially the distance threshold (eps) and the minimum number of points (min_samples). Suboptimal parameter selection may lead to under-segmentation or over-segmentation.

2. **Difficulty with varying density clusters :** DBSCAN struggles with clusters of varying densities. It may fail to connect regions with lower point density to the rest of the cluster, leading to suboptimal cluster assignments in datasets with regions of varying densities.



3. Does not predict : Unlike some clustering algorithms, DBSCAN does not predict the cluster membership of new, unseen data points. Once the model is trained, it is applied to the existing dataset without the ability to generalize to new observations outside the training set.

Application Areas of DBSCAN :

1. **Spatial Data Analysis:** DBSCAN is particularly well-suited for spatial data clustering due to its ability to find clusters of arbitrary shapes, which is common in geographic data. It's used in applications like identifying regions of similar land use in satellite images or grouping locations with similar activities in **GIS (Geographic Information Systems)**.

2. **Anomaly Detection:** The algorithm's effectiveness in distinguishing noise or outliers from core clusters makes it useful in anomaly detection tasks, such as detecting fraudulent activities in banking transactions or identifying unusual patterns in network traffic.

3. **Customer Segmentation:** In marketing and business analytics, DBSCAN can be used for customer segmentation by identifying clusters of customers with similar buying behaviors or preferences.

4. **Environmental Studies:** DBSCAN can be used in environmental monitoring, for example, to cluster areas based on pollution levels or to identify regions with similar environmental characteristics.

5. **Traffic Analysis:** In traffic and transportation studies, DBSCAN is useful for identifying hotspots of traffic congestion or for clustering routes with

similar traffic patterns.

6. **Machine Learning and Data Mining:** More broadly, in the fields of machine learning and data mining, DBSCAN is employed for exploratory data analysis, helping to uncover natural structures or patterns in data that might not be apparent otherwise.

I hope this blog has enhanced your comprehension of the DBSCAN concept. If you've gained value from this content, consider following me for more insightful posts. Appreciate your time in reading this article. Thank you!

Written by Sachinsoni

782 followers · 21 following

Follow

Responses (3)

Jagannath Rao

What are your thoughts?

Ranvir Allhabadiya
Aug 30

good explain

Reply

Justin
Aug 17

Good explanation, thank you!

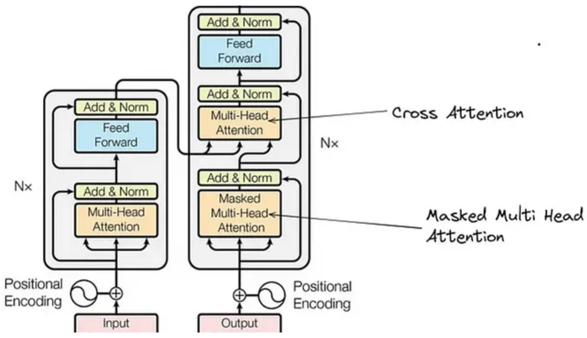
Reply

Lawal Oladipupo
Dec 16, 2024

This is a good read. Thank you.

Reply

More from Sachinsoni

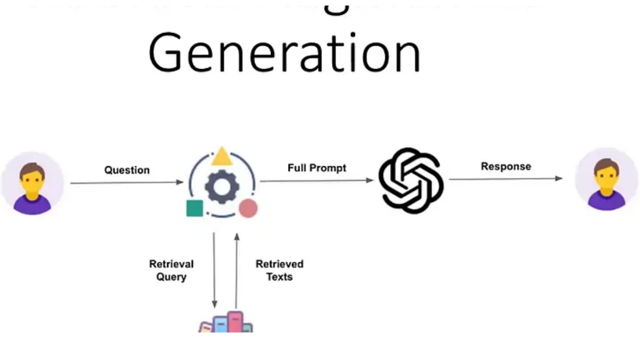


Sachinsoni

Cross Attention in Transformer

Cross attention is a key component in transformers, where a sequence can attend t…

Sep 6, 2024 155 3

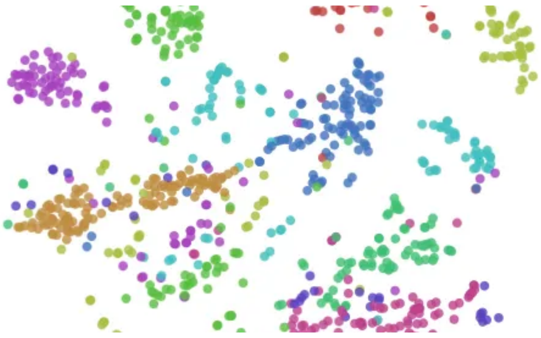


Sachinsoni

Introduction to RAG (Retrieval Augmented Generation) and…

Retrieval-Augmented Generation (RAG) is a powerful technique that enhances the…

Sep 14, 2024 75 3

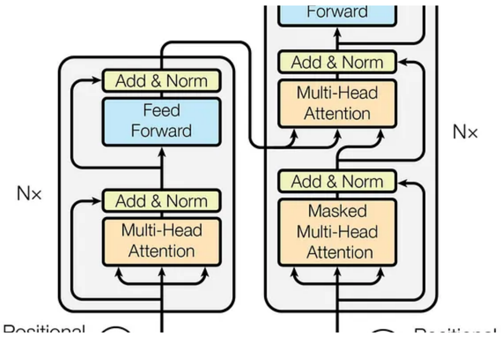


Sachinsoni

Mastering t-SNE(t-distributed stochastic neighbor embedding)

A better dimensionality reduction technique as compared to PCA (Principal Component…

Feb 11, 2024 224 5



In Towards AI by Sachinsoni

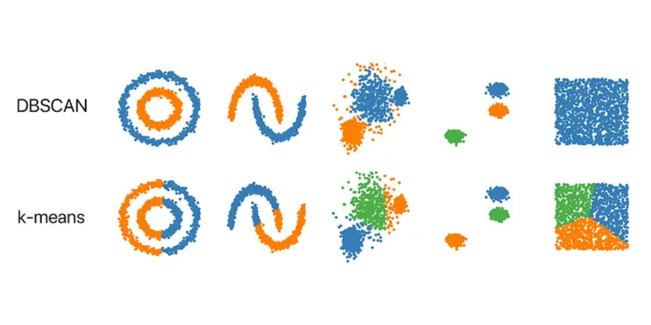
Transformer Architecture Part -1

In recent years, transformers have revolutionized the world of deep learning,…

Sep 5, 2024 148

See all from Sachinsoni

Recommended from Medium



Abhay singh

DBSCAN Explained: Unleashing the Power of Density-Based Clustering

Mastering unsupervised learning opens up many avenues for a data scientist. There is s...

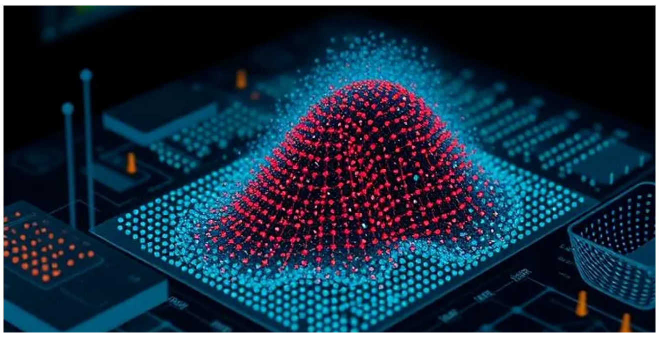
Jul 18 26

Lakhan Bukkawar

PCA vs t-SNE vs UMAP: Visualizing the Invisible in Your Data

See how each technique reveals hidden structure — with real-world cases, code, and...

May 5 9

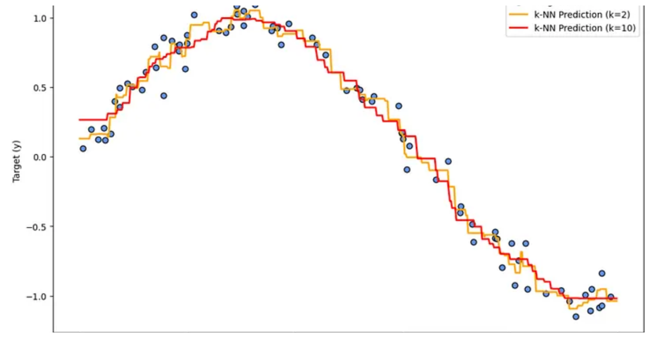


Priyanthan Govindaraj

Clustering in High-Dimensional Space: Building and Evaluating a...

A principled implementation combining K-Means++ and Spherical K-Means for...

May 15 2

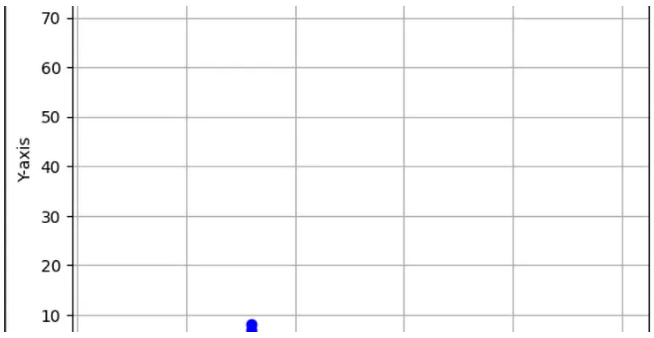


Dilip Kumar

k-Nearest Neighbors (k-NN) Regressor

1. The Core Concept: The Neighborhood Average

Jul 9



Abhishek Jain

UML Part 5 — HDBSCAN Clustering

The biggest problem of DBSCAN was: It was not able to cluster the data where the densit...

Jul 15 4

KishoreS

K-Means clustering

Its a popular unsupervised machine learning algorithm that is used to create clusters /...

Oct 12

See more recommendations