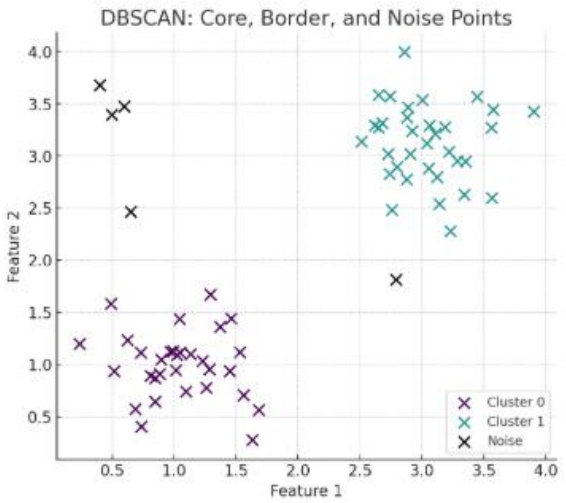
# ML - Clustering Algorithms

*Santhosh Gs*

# DBSCAN (Density-Based Spatial Clustering of Applications with Noise)




Here's a **visual of DBSCAN** :

- Colored groups = **clusters** (formed from dense regions).
- Black dots = **noise points (outliers)**.

**DBSCAN, or Density-Based Spatial Clustering of Applications with Noise defines clusters as continuous regions of high-density points**

**DBSCAN groups together points that are close and dense, and labels far-away points as noise (outliers).**

**Key Concepts in DBSCAN**

There are two parameters we must know:

1. ε (eps) → The radius (neighborhood size).

Example - Think of it like "how far I look around each point to check neighbors."

2. minPts → Minimum number of points inside that radius.

Example -"How many friends do I need nearby to feel like I'm in a group?"

**Types of Points in DBSCAN**

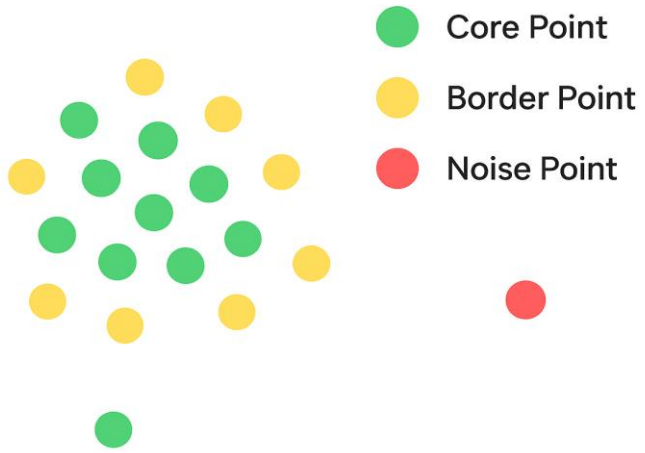
- ***Core Point*** → A scatter plot of data points shows several densely packed regions. In one region, a point is designated as a "core point" .Has at least minPts neighbors inside radius ε.
- ***Border Point*** → A point is shown that is not a core point itself (its ε-neighborhood contains fewer than minPts points) but is located within the ε-neighborhood of a core point. This point is labeled as a "border point.
- ***Noise Point (Outlier)*** → isolated neighbors.

A cluster is then formed by connecting all core points that are "density-reachable" from one another and including all associated border points.

# DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

***How DBSCAN Works***

- Pick a random point.
- Look within radius ε to find neighbors.
- If neighbors ≥ minPts → mark it a core point and form a cluster.
- If not, mark it as noise (for now).
- Expand the cluster
- Add all neighbors of that core point.
- If any neighbor is also a core → expand again (snowball effect).
- Repeat until all points are visited.
- **Advantages**
  - No need to specify number of clusters (unlike K-Means).
  - Handles **outliers(*Noise Point)*** naturally.
- **Disadvantages**
  - A significant drawback of DBSCAN is its high sensitivity to its two parameters, ε and minPts. Even a small change in these values can drastically alter the clustering outcome, potentially causing clusters to fragment or merge



# Affinity Propagation(AP)

- Affinity Propagation creates clusters by sending messages between pairs of samples until convergence.
- In simple , It's a clustering algorithm where **data points talk to each other** to decide:
  - Who will be the **leaders** (called **exemplars**).
  - Which points will **belong to which leader**
- Affinity Propagation works based on similarities between data points
  - Imagine all your data points in a group chat:
  - Each point sends messages to others about who should be their leader.
  - After many rounds of messaging, a few points get chosen as exemplars (leaders).
  - Other points join the leader they like most. This naturally forms clusters.

  **Two Kinds of Messages**

  1. Responsibility (R)

  How much point i thinks point j is a good leader for it.

  2. Availability (A)

  How much point j is willing to accept point i as its follower.

  These messages are updated back and forth until things stabilize.

# Affinity Propagation(AP)

**Sequence for Affinity Propagation**

Step 1: Raw data, no structure.

Step 2: Points exchange “messages” and candidates for leaders appear

(red circles).

Step 3: Exemplars (leaders) are chosen (blue stars).

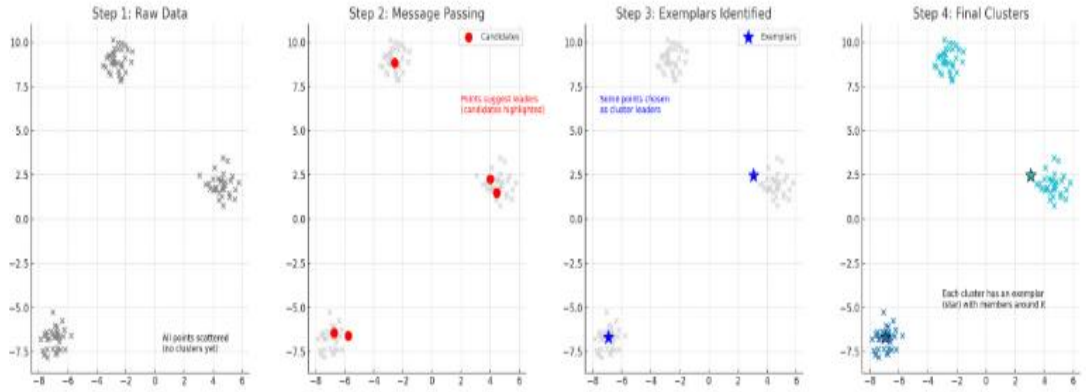
Step 4: Final clusters form around exemplars, with each group colored**.**

- **Advantages**

Has better performance and lower clustering error

- **Disadvantages**

It is quite slow and memory-heavy, making it difficult to scale to larger datasets.



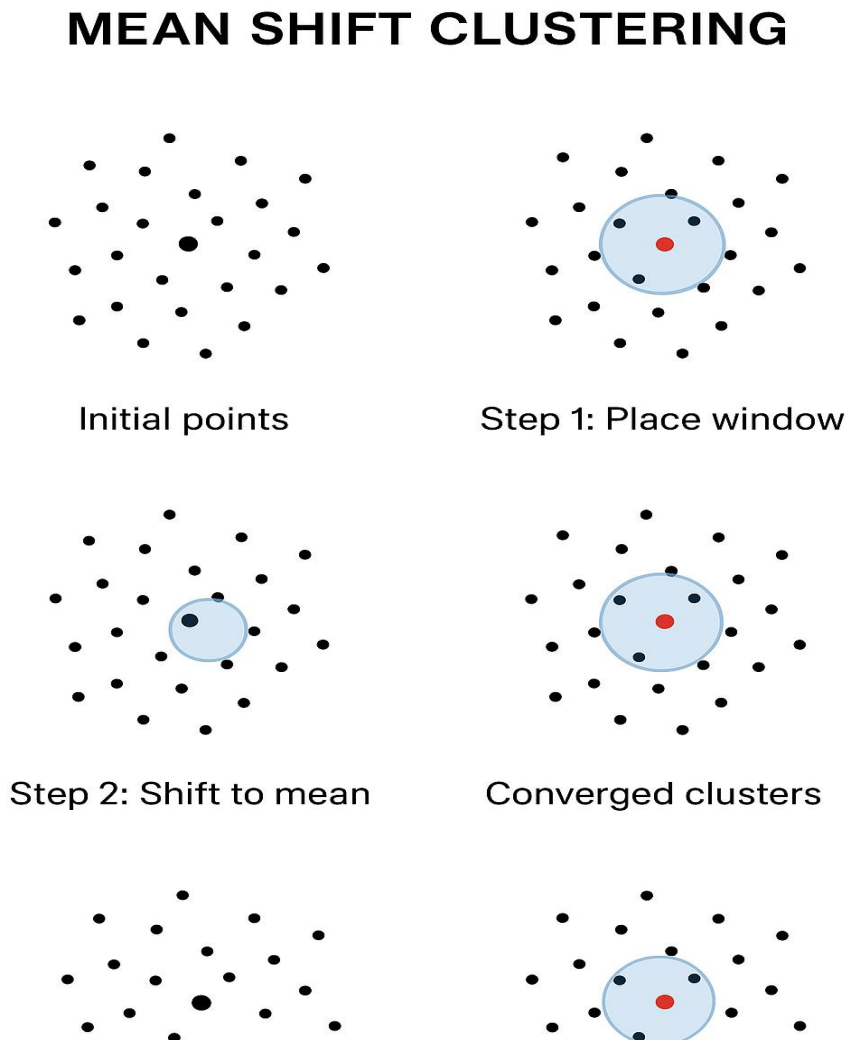
# Mean-Shift Clustering

Mean-Shift is an iterative algorithm that works by moving data points to the mean of the points within a specified radius or "kernel."

The process is as follows:

- Start: Pick a random point in the dataset.
- Define a Window: Place a circular or spherical "window" (defined by a radius, h) around this point.
- Calculate the Mean: Compute the mean of all data points that fall within this window. This mean represents the new center of the window.
- Shift the Window: Shift the window to this new mean.
- Repeat: Continue steps 3 and 4 until the window stops moving, which means it has converged to a region of high density—a cluster center. All points that converge to the same cluster center are assigned to that cluster.

The size of the window (the bandwidth h) is the most critical parameter. A smaller bandwidth can lead to more, smaller clusters, while a larger bandwidth can lead to fewer, larger clusters.



# Mean-Shift Clustering

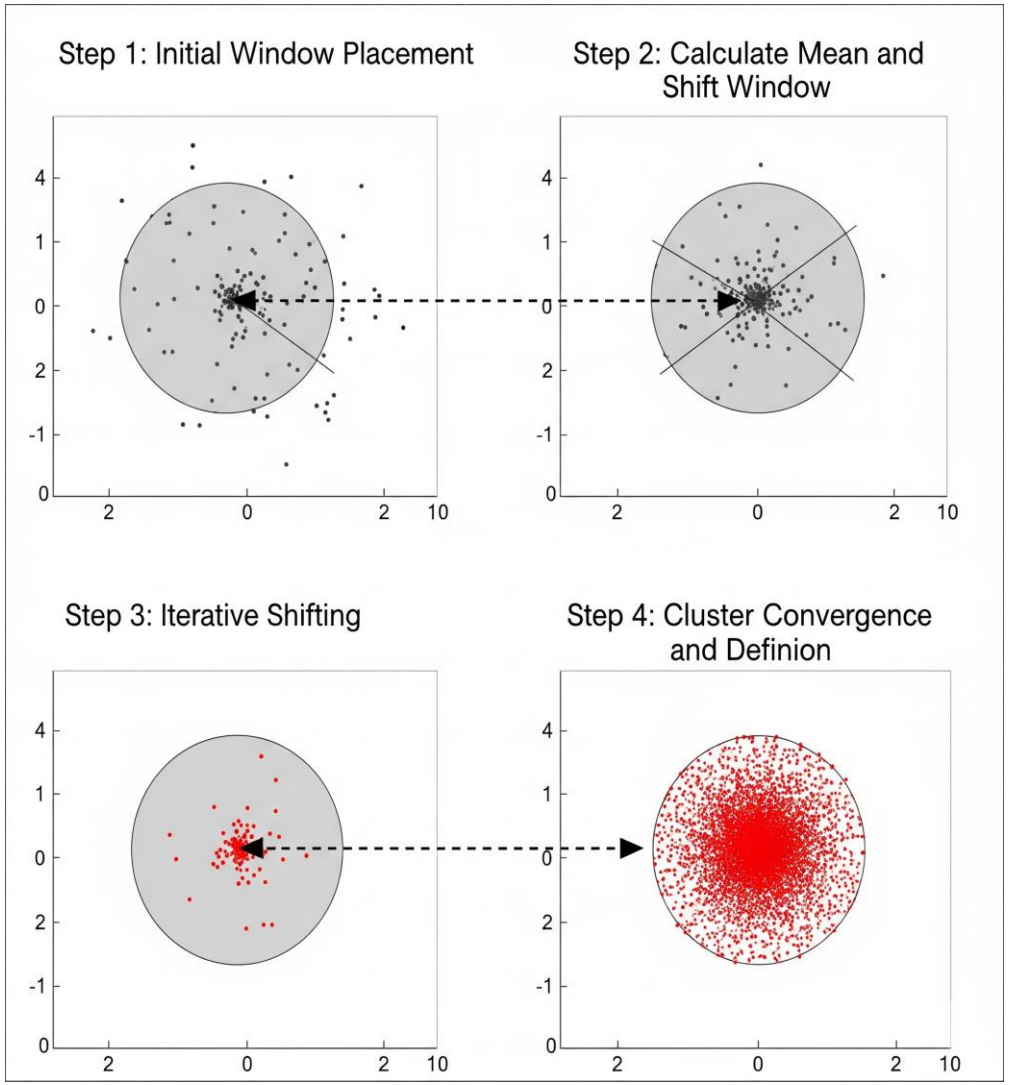
The image above shows a four-panel diagram explaining the Mean-Shift clustering process:

Panel 1: Initial Window Placement. The algorithm starts by placing a circular "window" (or a kernel) at a random data point.

Panel 2: Calculate Mean and Shift Window. The algorithm calculates the mean of all the points contained within the window. An arrow shows that the window then shifts to this new mean.

Panel 3: Iterative Shifting. The process repeats: the algorithm finds the new mean of the points within the shifted window and moves the window to that location. The window is getting closer to the densest group of points.

Panel 4: Cluster Convergence and Definition. This iterative process continues until the window stops moving, meaning it has converged on a region of high data density. All points that end up within this final, stable window are assigned to the same cluster, as indicated by the red points.




**Advantages**

Does not require the number of clusters: Mean-Shift automatically determines the number of clusters. You don't need to specify the parameter k like you do with K-Means.

**Disadvantages**

Mean-shift algorithm does not work well in case of high dimension, where number of clusters changes abruptly.