

Phân Cụm Dựa Trên Mật Độ: Thuật Toán DBSCAN

Khám phá cấu trúc dữ liệu theo hình dạng tự nhiên, không cần xác định trước số cụm.

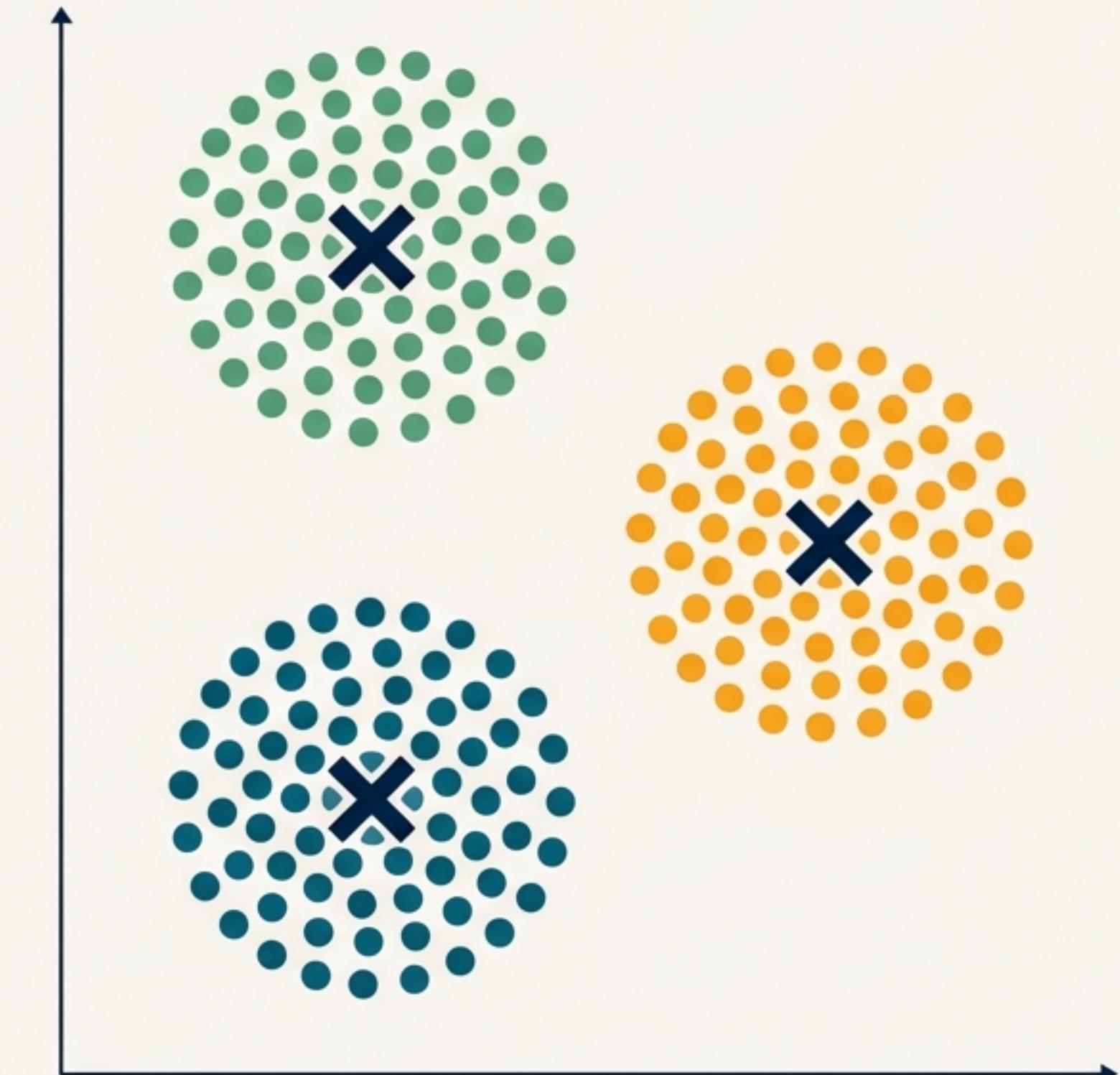


Điểm Khởi Đầu Quen Thuộc: Thuật Toán K-Means

Tóm tắt nhanh về K-Means: Là một thuật toán học không giám sát (Unsupervised Learning) phổ biến nhất để phân cụm dữ liệu.

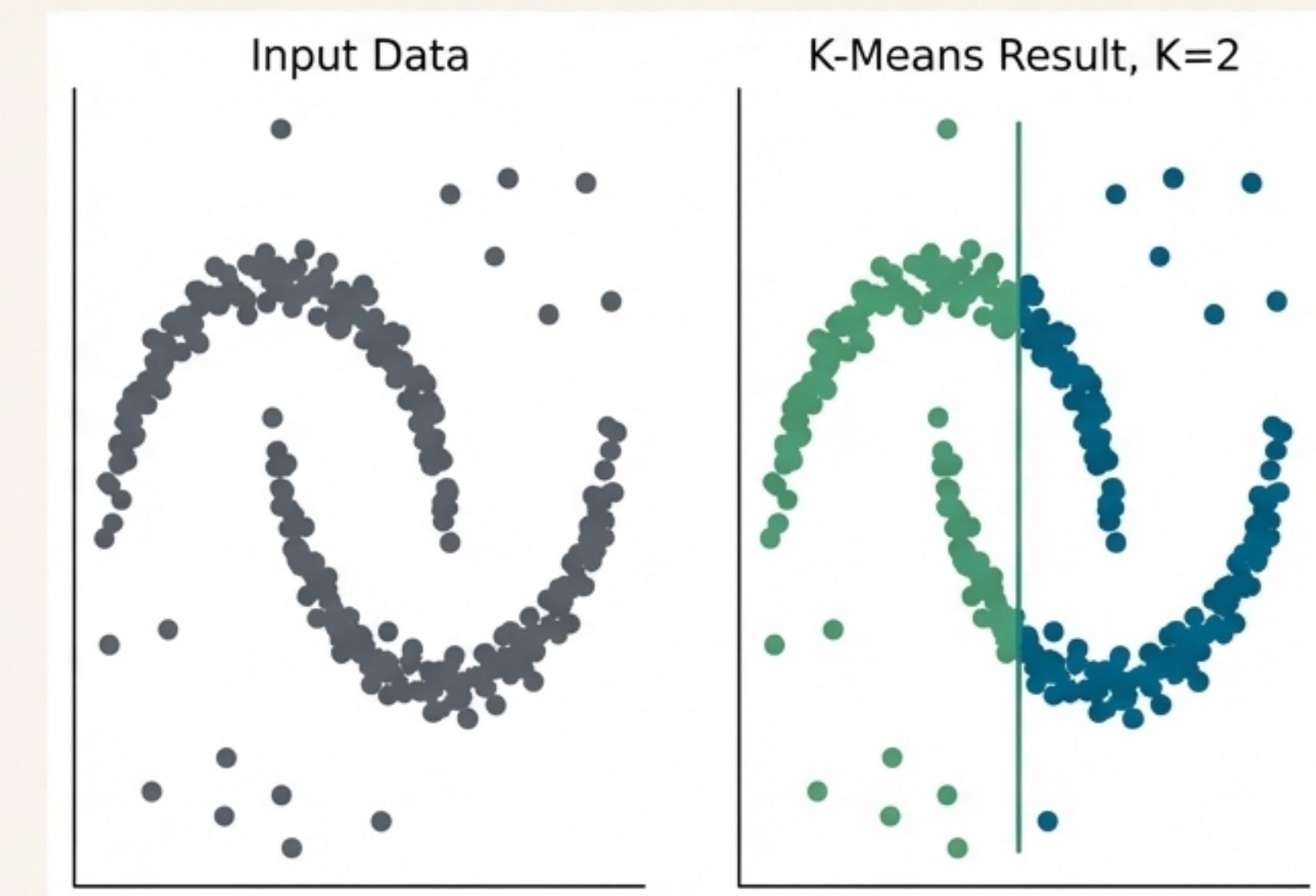
Cách hoạt động cốt lõi:

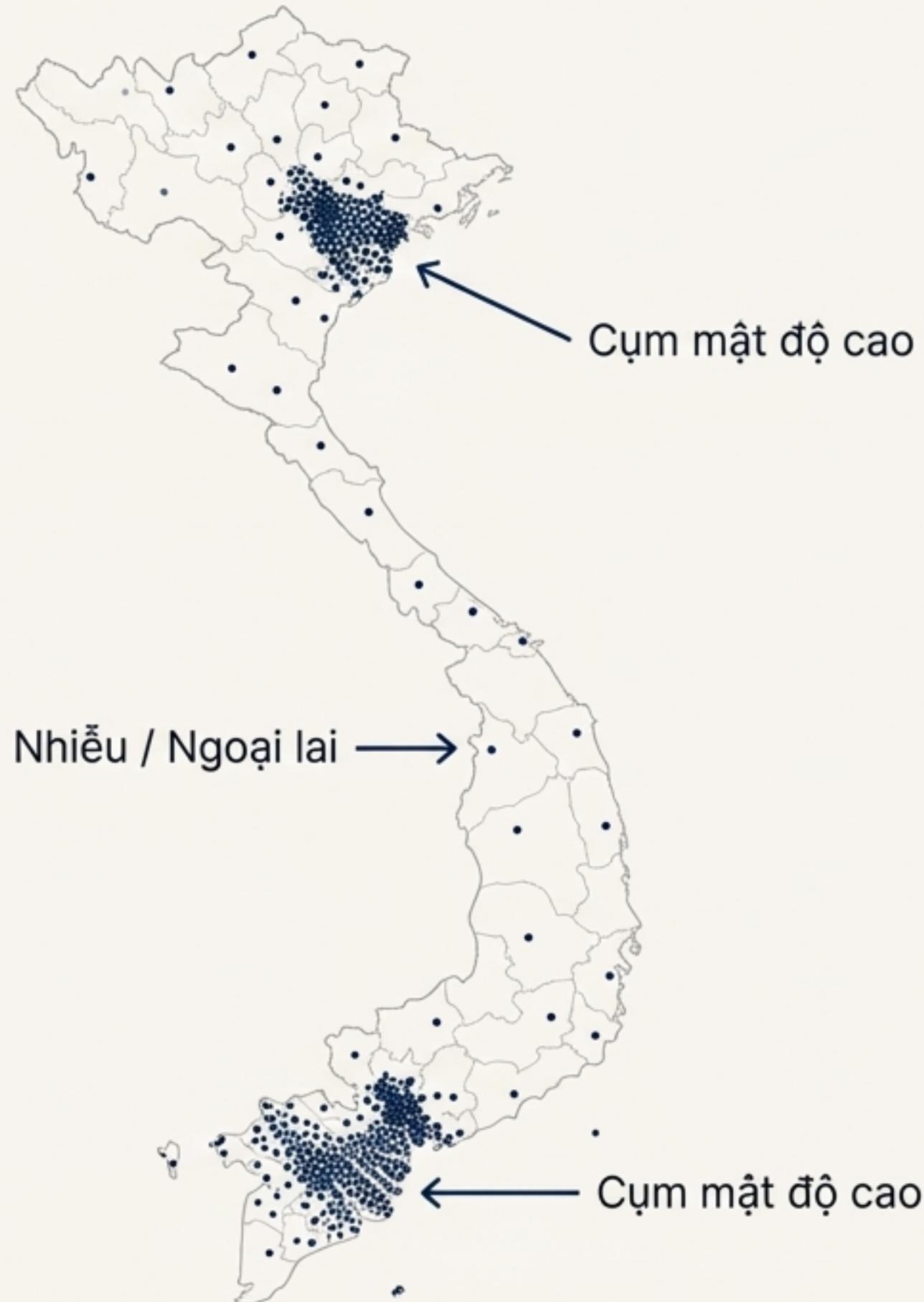
1. ***Xác định số cụm (K):** Người dùng phải chọn trước số lượng cụm cần tìm.
2. ***Gán điểm vào cụm*:** Mỗi điểm dữ liệu được gán vào cụm có tâm (centroid) gần nhất.
3. ***Cập nhật tâm cụm*:** Vị trí tâm cụm được tính toán lại dựa trên các điểm trong cụm.
4. Quá trình lặp lại cho đến khi tâm cụm không còn thay đổi.



Những Thử Thách K-Means Khó Lòng Vượt Qua

- Vấn đề 1: Nhạy cảm với điểm ngoại lai (Outliers):** Các điểm outlier có thể kéo tâm cụm đi sai hướng, làm sai lệch hình dạng và kết quả phân cụm.
- Vấn đề 2: Phải xác định trước số K:** Trong nhiều bài toán thực tế (ví dụ: phân khách hàng), việc biết trước số cụm là bất khả thi.
- Vấn đề 3: Chỉ hoạt động tốt với cụm hình cầu (Spherical):** K-Means giả định các cụm có dạng hình cầu và tách biệt tuyến tính, dẫn đến kết quả sai khi dữ liệu có cấu trúc phức tạp.





Một Tư Duy Mới: 'Cụm' Là Nơi Dữ Liệu 'Tập Trung Đông Đúc'

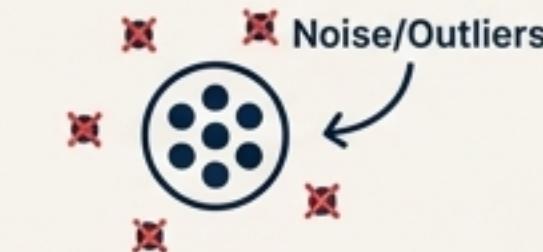
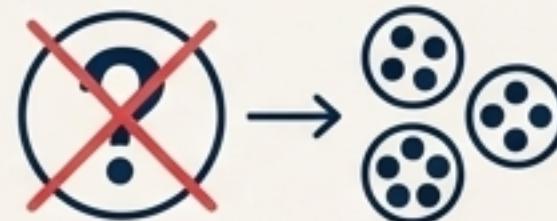
Thay vì tìm một "tâm" (center), DBSCAN tìm các khu vực có "mật độ" (density) điểm dữ liệu cao.

Tương tự như mật độ dân số: Các cụm dữ liệu giống như các khu vực đô thị hoặc đồng bằng đông dân (ví dụ: đồng bằng sông Cửu Long, đồng bằng sông Hồng). Các điểm dữ liệu riêng lẻ, thưa thớt ở vùng sâu vùng xa được xem là "nhiễu" (noise) hoặc "ngoại lai" (outliers).

Cách tiếp cận này cho phép chúng ta tìm ra các cụm với hình dạng bất kỳ, vì chúng ta chỉ đơn giản là đi theo nơi dữ liệu dày đặc.

Lời Giải Đáp: DBSCAN

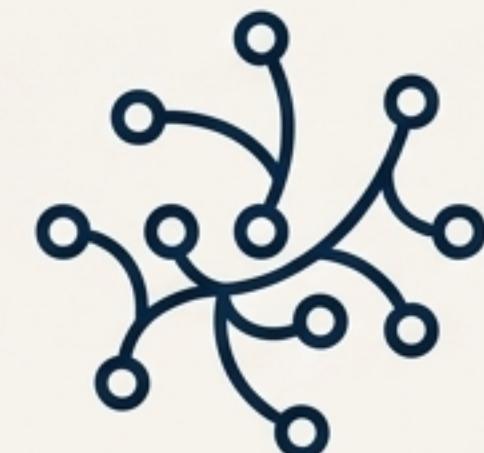
Density-Based Spatial Clustering of Applications with Noise



Không cần định nghĩa trước số cụm (K): Thuật toán tự động tìm ra số lượng cụm dựa trên mật độ.

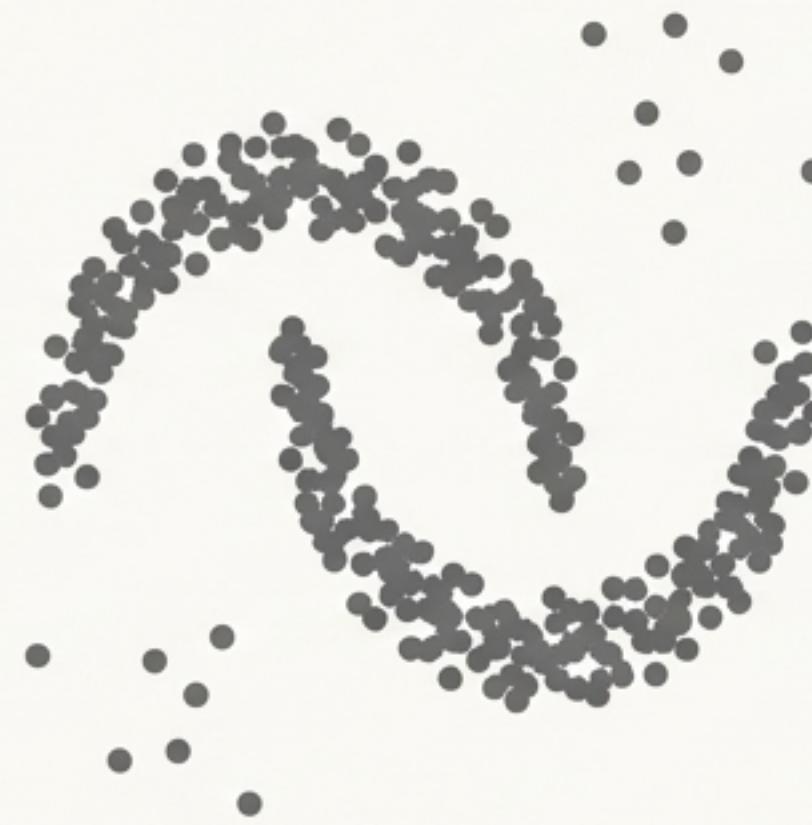
Phát hiện các cụm có hình dạng tùy ý: Có thể nhận diện các cấu trúc phức tạp, không phải hình cầu. Đường phân chia giữa các cụm không nhất thiết phải là đường thẳng (non-linear).

Khả năng chống nhiễu (Resistance to noise): Các điểm ngoại lai được xác định và tách riêng, không làm ảnh hưởng đến các cụm hiện có.

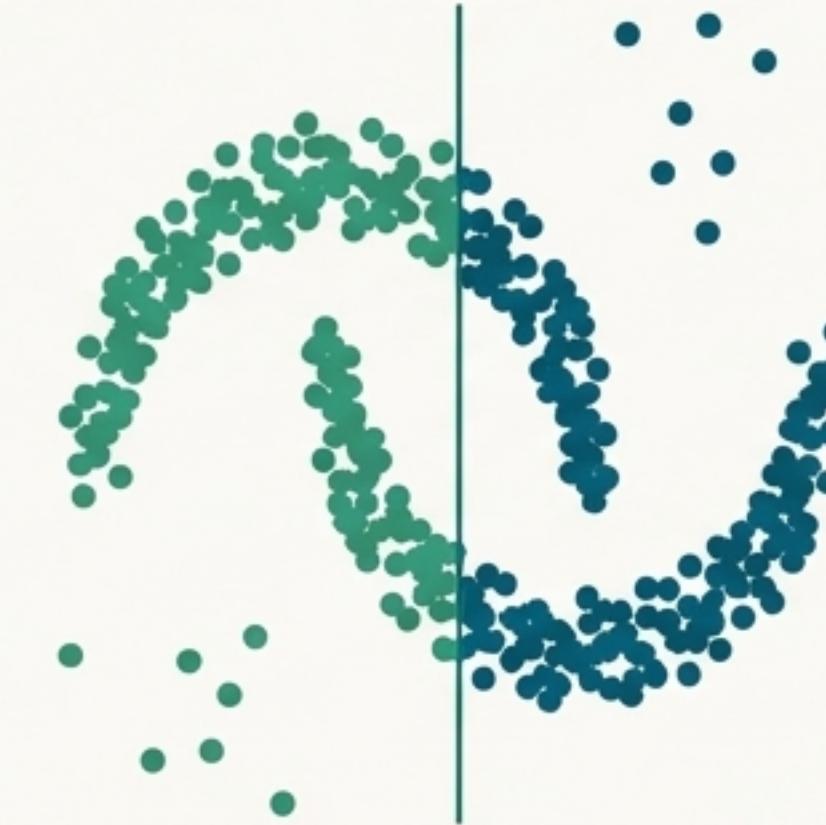


Cuộc Đổi Đầu Trực Diện: K-Means vs. DBSCAN

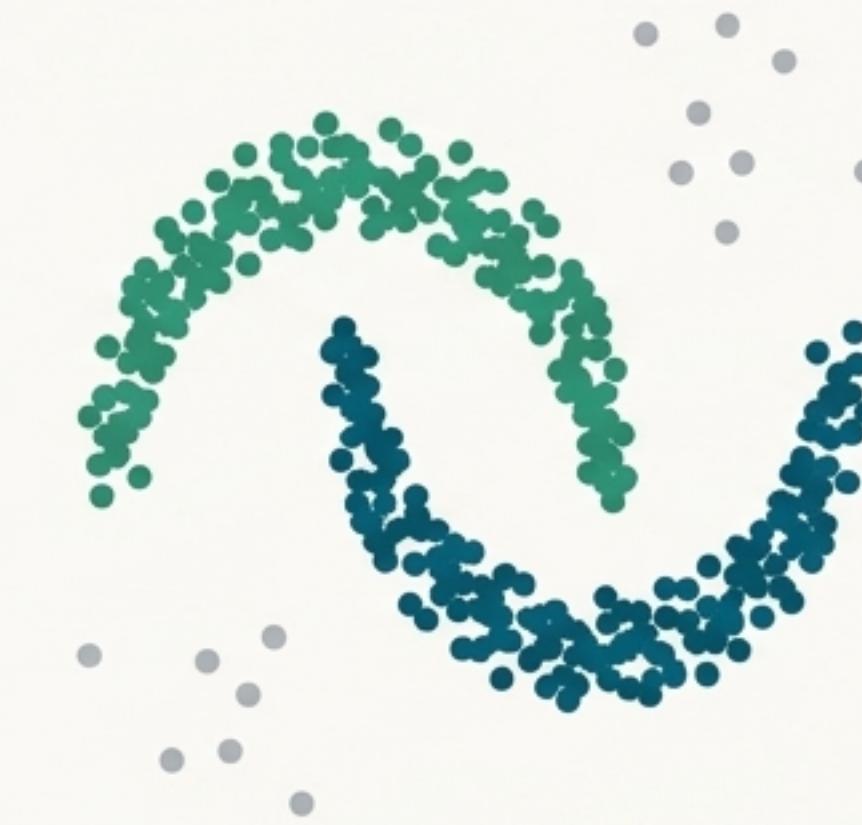
Cùng xem lại bộ dữ liệu đã làm khó K-Means, nhưng lần này với sức mạnh của DBSCAN.



Dữ Liệu Gốc



K-Means: Thất bại trong việc
nhận diện hình dạng

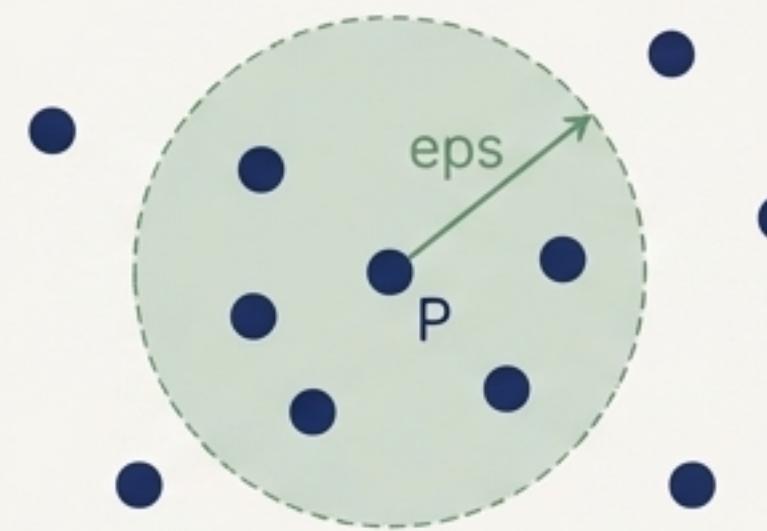


DBSCAN: Nhận diện chính xác
hình dạng và nhiễu

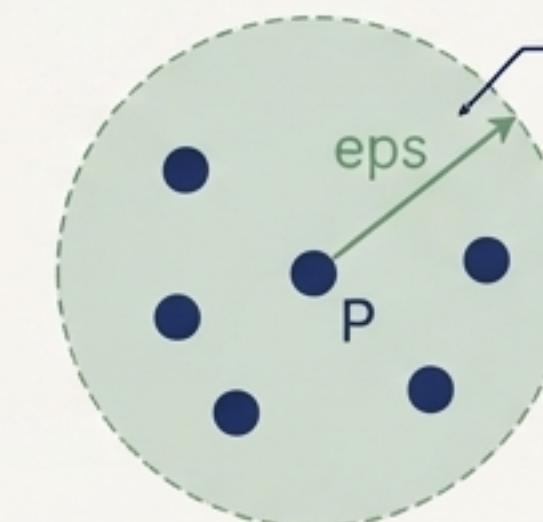
Ngôn Ngữ Của DBSCAN: Hai Tham Số Quyết Định

Thay vì chọn K, chúng ta định nghĩa "**mật độ**" thông qua hai tham số:

1. **Epsilon (eps)**: Bán kính của một "vùng lân cận". Đây là khoảng cách tối đa giữa hai điểm để chúng được coi là "hàng xóm" của nhau. Khoảng cách này thường được tính bằng khoảng cách Euclid.



2. **Minimum Points (MinPts)**: Số lượng điểm hàng xóm tối thiểu (bao gồm cả chính nó) mà một điểm phải có trong vùng lân cận `eps` để được coi là một "điểm lõi".

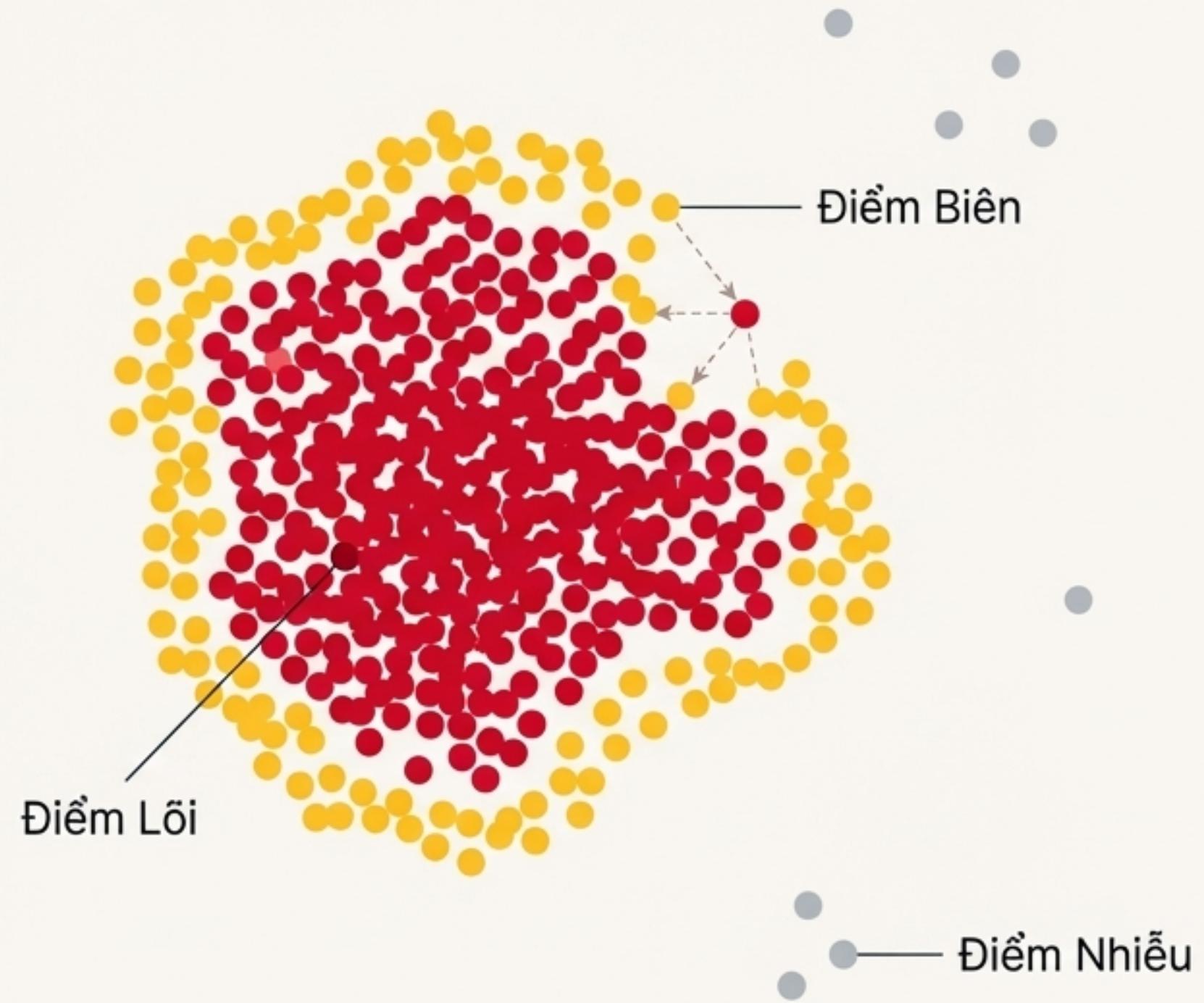


Nếu $\text{MinPts} = 4$, điểm 'P' này là một điểm lõi

Phân Loại Điểm: Lõi, Biên, và Nhiễu

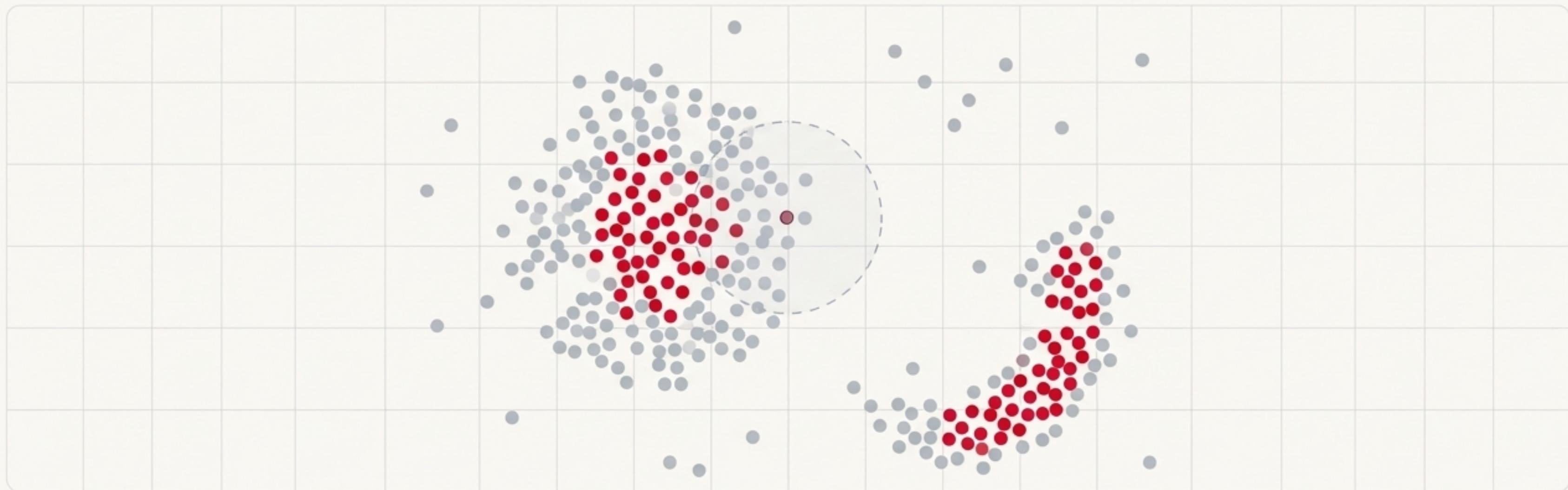
Dựa trên `eps` và `MinPts`, DBSCAN phân loại mọi điểm trong dữ liệu thành một trong ba loại:

- **Điểm Lõi (Core Point)**: Một điểm có ít nhất `MinPts` hàng xóm (bao gồm cả chính nó) trong bán kính `eps`. Đây là những điểm nằm sâu bên trong một cụm.
- **Điểm Biên (Border Point)**: Một điểm có ít hàng xóm hơn `MinPts`, nhưng lại là hàng xóm của một Điểm Lõi. Đây là những điểm nằm ở rìa của một cụm.
- **Điểm Nhiễu (Noise Point / Outlier)**: Một điểm không phải là Điểm Lõi và cũng không phải là hàng xóm của bất kỳ Điểm Lõi nào.



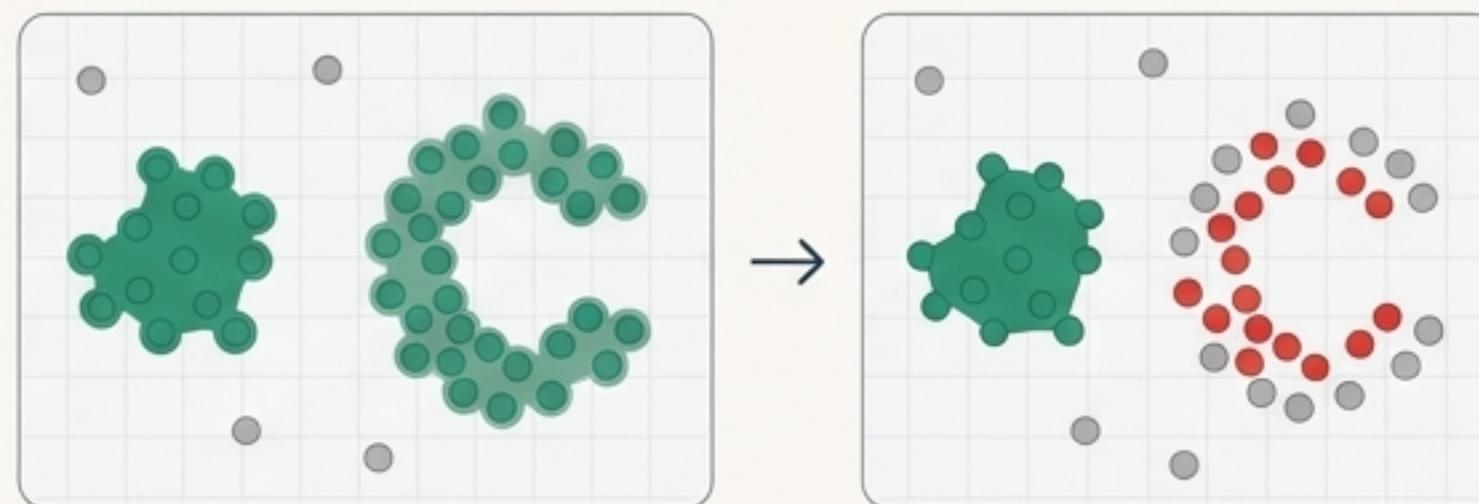
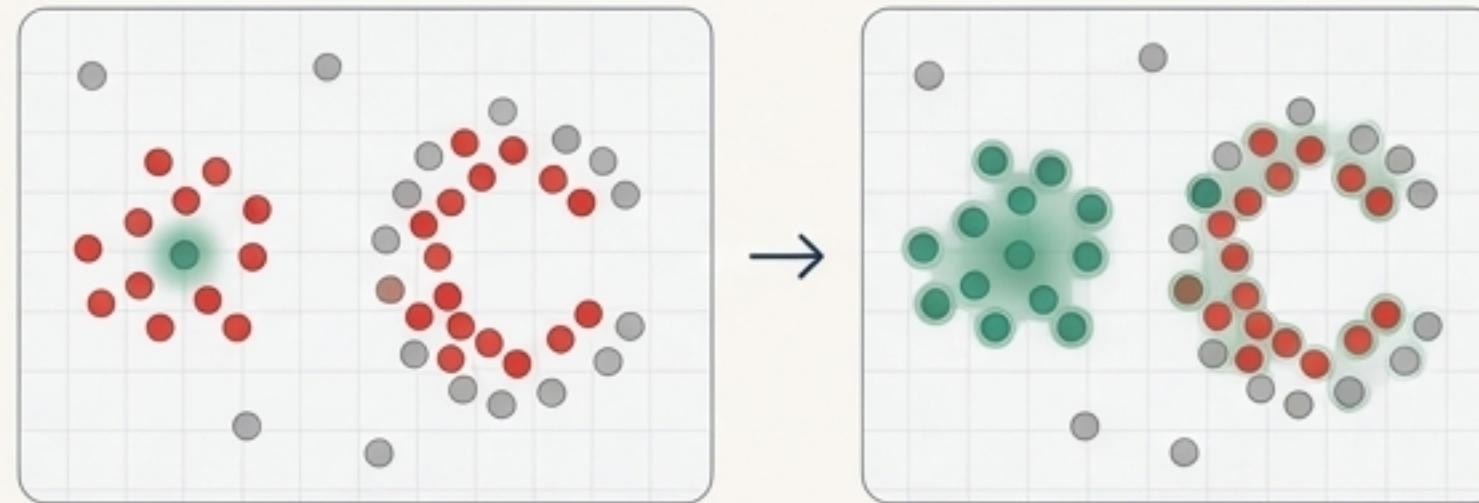
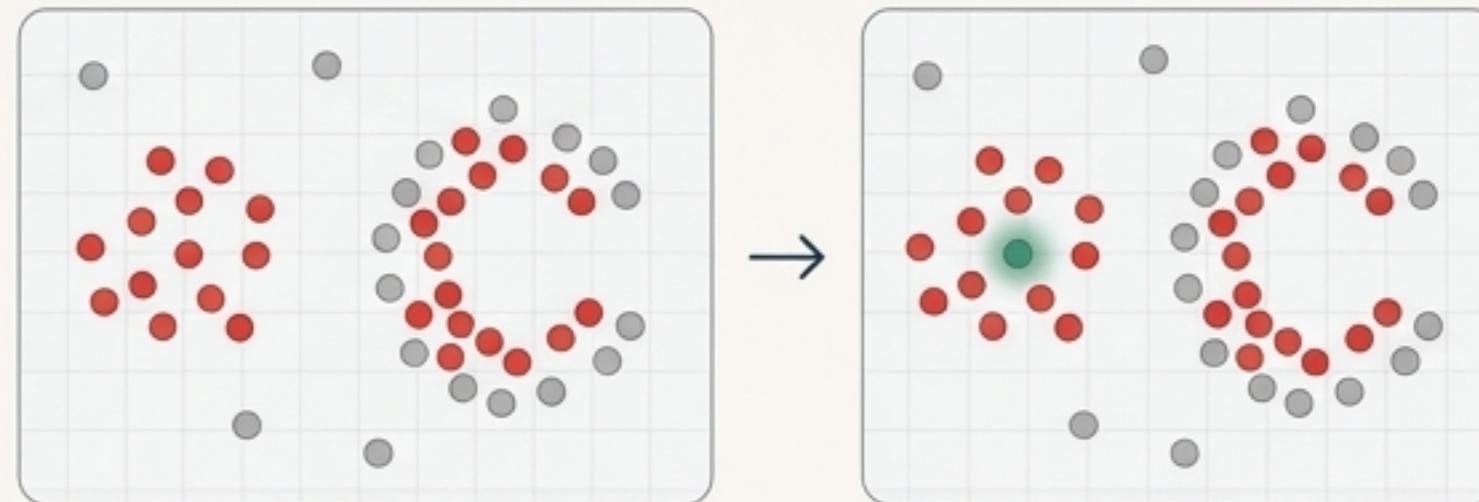
Thuật Toán Hoạt Động - Bước 1: Xác Định Tất Cả Các Điểm Lõi

Thuật toán bắt đầu bằng cách duyệt qua tất cả các điểm trong bộ dữ liệu. Với mỗi điểm, nó đếm số lượng “hàng xóm” trong bán kính ‘`eps`’. Nếu số lượng hàng xóm $\geq \text{'MinPts'}$, điểm đó được đánh dấu là một **Điểm Lõi**.



Thuật Toán Hoạt Động - Bước 2: Mở Rộng Cụm Từ Điểm Lõi

1. Chọn một Điểm Lõi (màu đỏ) bất kỳ chưa được gán vào cụm nào và tạo một cụm mới.
2. Thêm tất cả các hàng xóm (cả Lõi và Biên) của điểm đó vào cụm.
3. Lặp lại quá trình này với các Điểm Lõi mới được thêm vào, liên tục mở rộng cụm cho đến khi không thể thêm điểm nào nữa.
4. Quá trình này được gọi là tìm kiếm các điểm "có thể kết nối về mặt mật độ" (density-reachable).



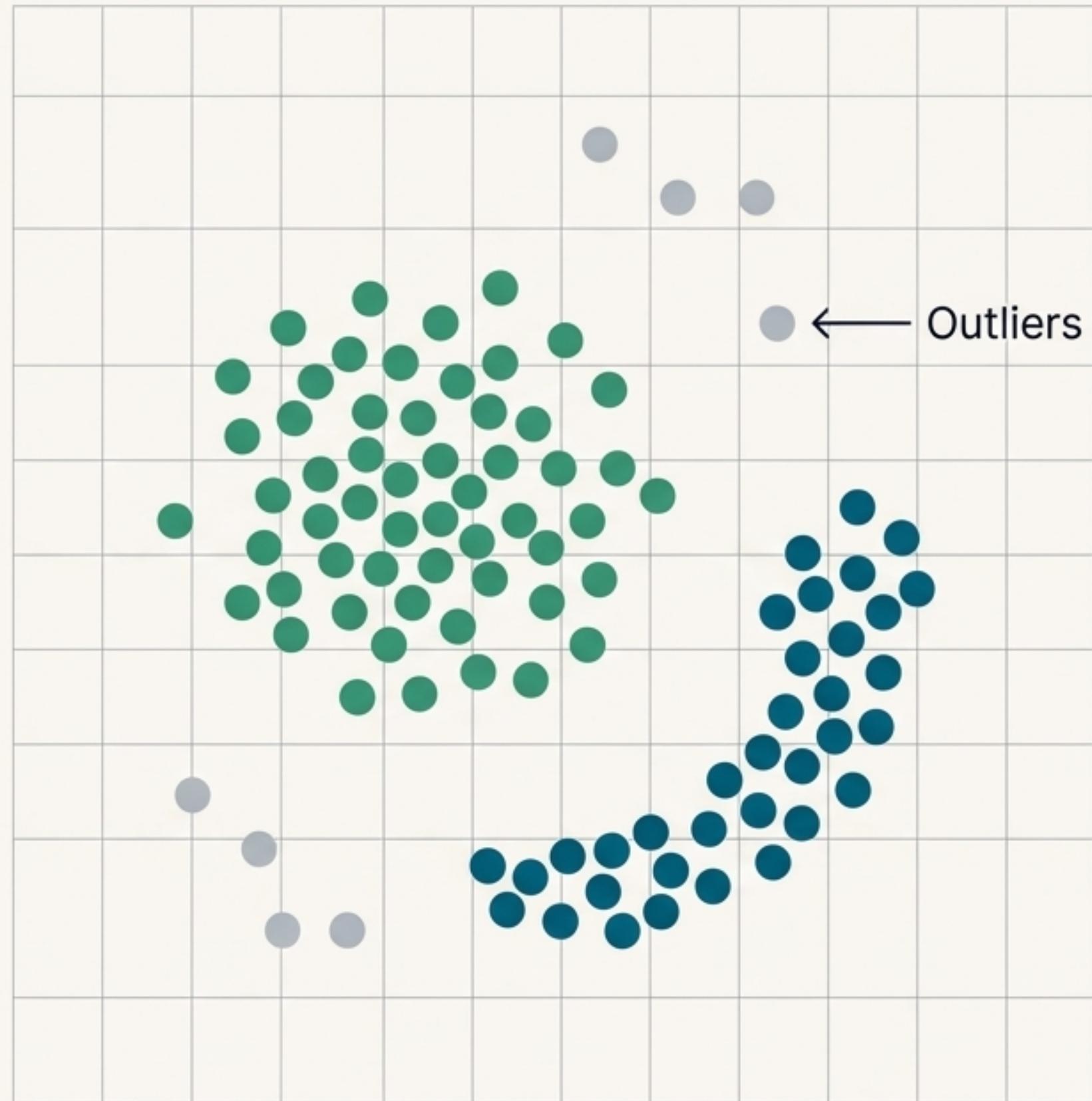
Thuật Toán Hoạt Động

- Bước 3: Hoàn Tất và Xác Định Nhiễu

Sau khi cụm đầu tiên được hình thành, thuật toán tìm một Điểm Lỗi khác chưa được gán và lặp lại quá trình tạo cụm mới (ví dụ: cụm màu xanh dương).

Quá trình tiếp tục cho đến khi tất cả các Điểm Lỗi đều thuộc về một cụm nào đó.

Bất kỳ điểm nào còn lại (những điểm không phải là Lỗi và không thuộc vùng lân cận của bất kỳ Điểm Lỗi nào) sẽ được phân loại là **Nhiễu (Noise)**.



Đánh Giá Toàn Diện: Ưu và Nhược Điểm Của DBSCAN

Ưu Điểm

- ✓ Không yêu cầu xác định trước số cụm.
- ✓ Hỗ trợ các cụm có hình thù đa dạng, phức tạp.
- ✓ Có khả năng phát hiện và cô lập các điểm ngoại lai (outliers).

Nhược Điểm

- ✗ Khó khăn khi các cụm có mật độ rất khác nhau (một cụm rất dày đặc, một cụm rất thưa).
- ✗ Việc chọn tham số `eps` và `MinPts` không phải lúc nào cũng dễ dàng và đòi hỏi sự hiểu biết về dữ liệu (domain knowledge).
- ✗ Hiệu năng có thể giảm với dữ liệu có số chiều rất cao (high-dimensional data).

Từ Lý Thuyết Đến Thực Hành: Triển Khai Với Scikit-learn

Việc áp dụng DBSCAN trong thực tế vô cùng đơn giản với thư viện `scikit-learn` của Python. Bạn không cần phải viết lại thuật toán từ đầu.

```
from sklearn.cluster import DBSCAN # Keywords: muted purple (#8338ec)
from sklearn.preprocessing import StandardScaler
import numpy as np

# Giả sử X là dữ liệu của bạn
# Scale dữ liệu là một bước quan trọng
X = StandardScaler().fit_transform(X) # Code Syntax - Comments: neutral grey (#ADB5BD)

# Khởi tạo và huấn luyện mô hình DBSCAN
db = DBSCAN(eps=0.4, min_samples=20).fit(X) # Strings/Values: muted green (#1A936F)

# Lấy nhãn của các cụm
# Nhãn -1 thể hiện các điểm outlier
cluster_labels = db.labels_ # Code Syntax - Comments: neutral grey (#ADB5BD)
```

eps: Tương ứng với tham số Epsilon, định nghĩa bán kính lân cận.

min_samples: Tương ứng với tham số MinPts.

Kết Quả Trực Quan: Phân Cụm Chính Xác và Loại Bỏ Nhiễu

Áp dụng đoạn code trên vào một bộ dữ liệu mẫu, DBSCAN đã thành công trong việc xác định ba cụm riêng biệt và phân loại 18 điểm là ngoại lai, giữ lại hình dạng nguyên bản của dữ liệu.



Phân loại gốc



Kết quả phân cụm của DBSCAN