

# Unit 05: DBSCAN

## 分群演算法

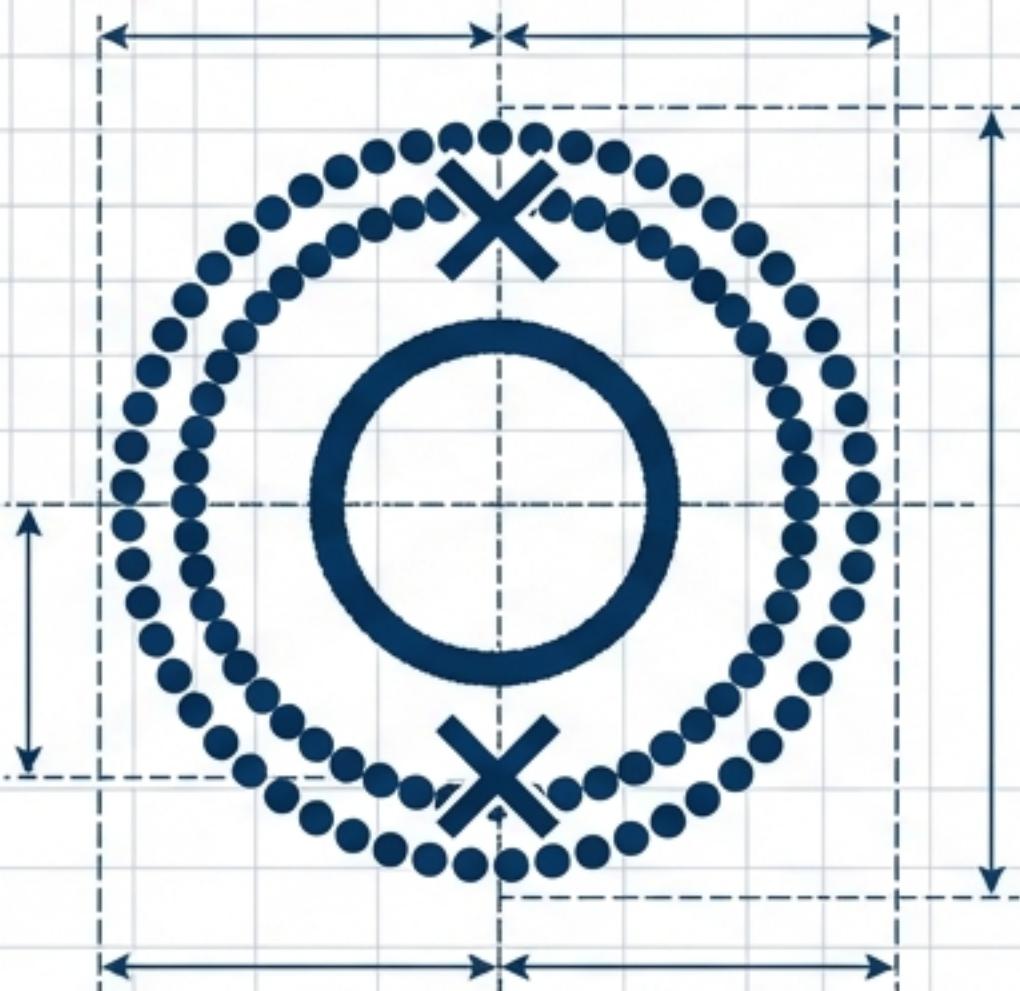
基於密度的空間分群應用：處理化工  
數據中的噪音與不規則形狀

### Metadata Block

Course: AI 在化工上之應用  
Module: 非監督式學習 / 密度分析  
Instructor: 莊曜禎 助理教授

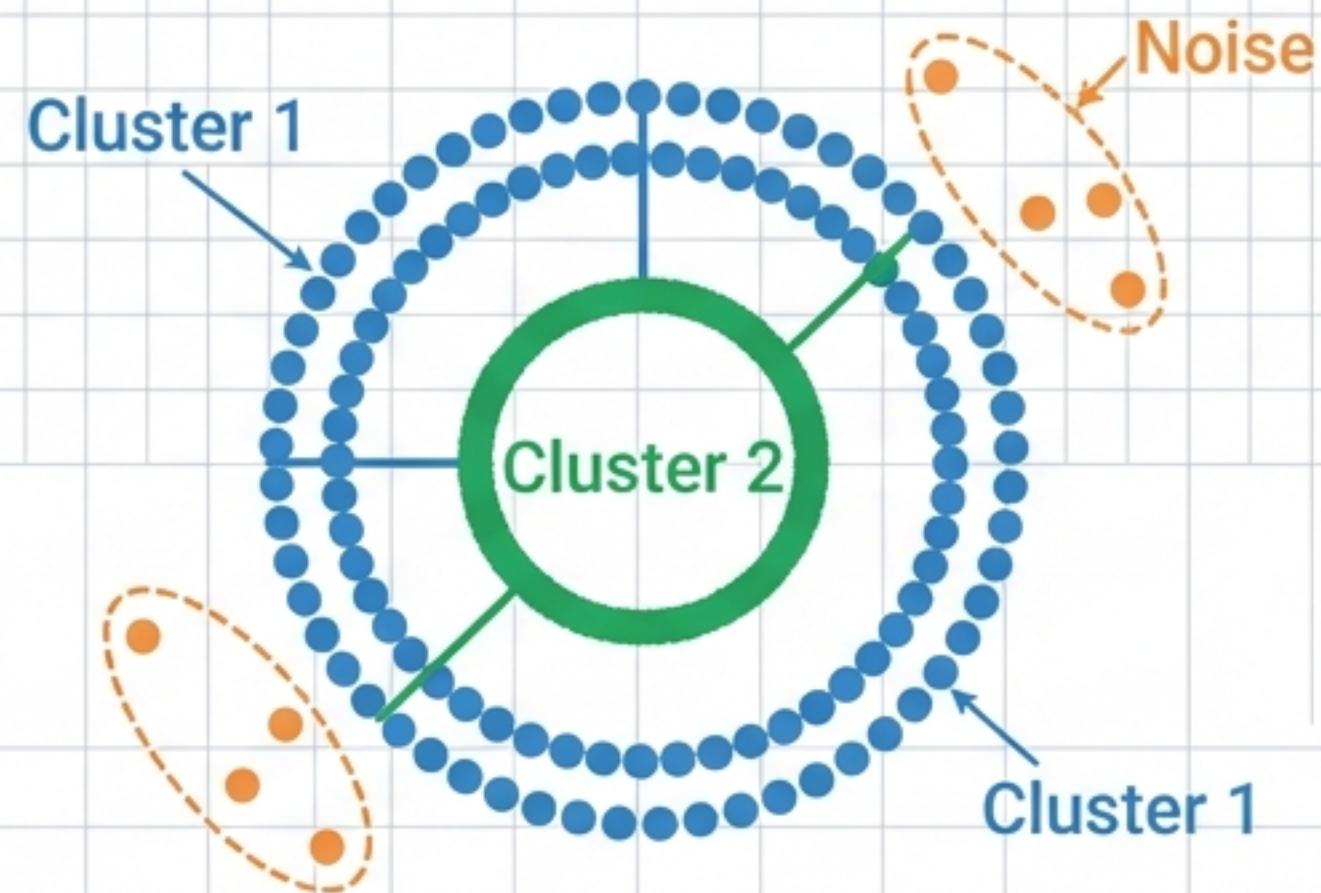
# 工程挑戰：當數據不再完美

## K-Means 限制



假設群集為球形且密度均勻

## DBSCAN 優勢



適應任意形狀 + 自動濾除噪音

✓ 異常操作探索：自動識別不屬於任何正常模式的數據點

✓ 複雜狀態識別：反應器操作區間往往是不規則的非線性形狀

# 核心參數規格：定義密度

**Parameter 1:  $\varepsilon$  (eps)**

鄰域半徑 (Neighborhood Radius)

數學定義：

$$N_{\varepsilon}(p) = \{q \in D \mid \text{dist}(p, q) \leq \varepsilon\}$$

化工類比：

操作條件的容許誤差範圍

$\varepsilon$  (eps)

P

**Parameter 2: MinPts (min\_samples)**

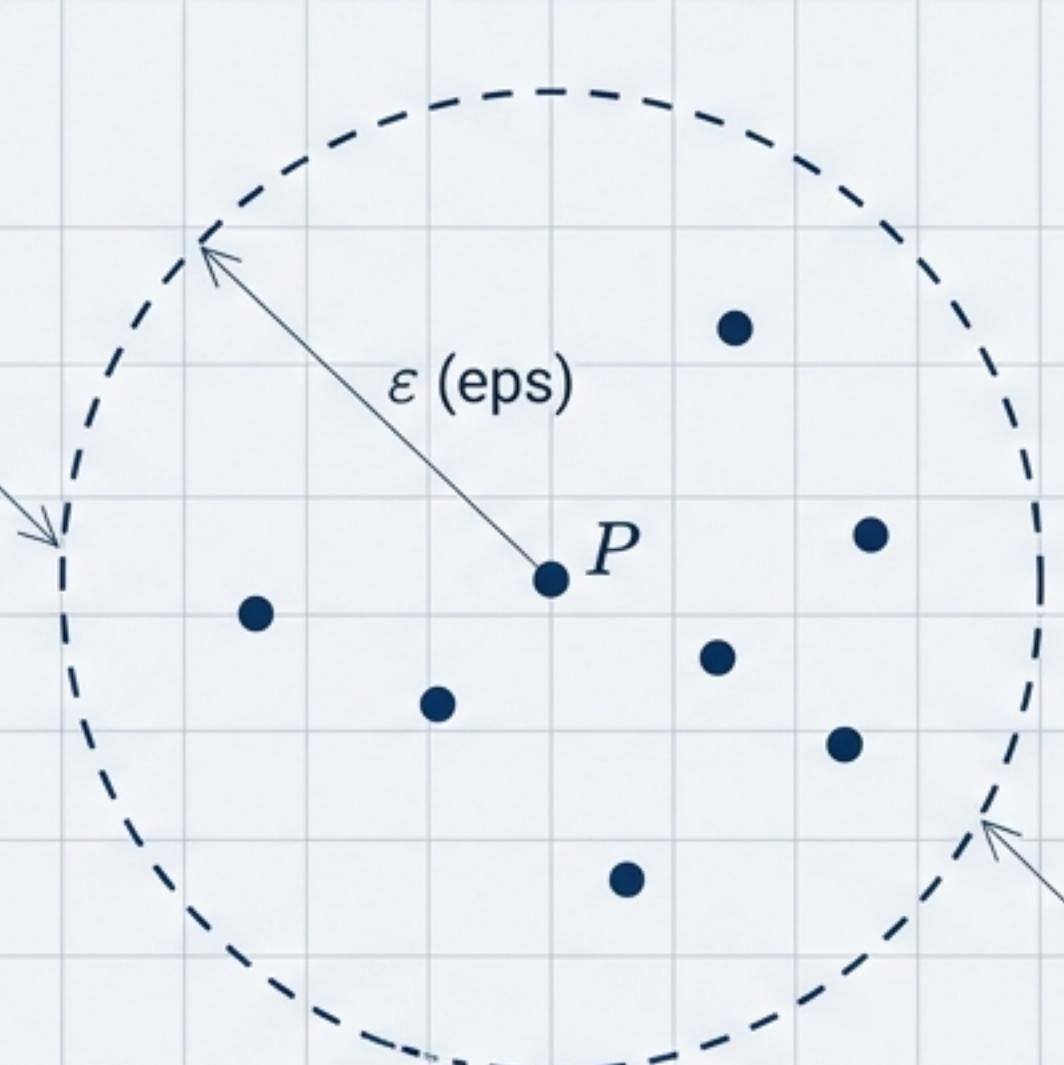
密度閾值 (Density Threshold)

規則：

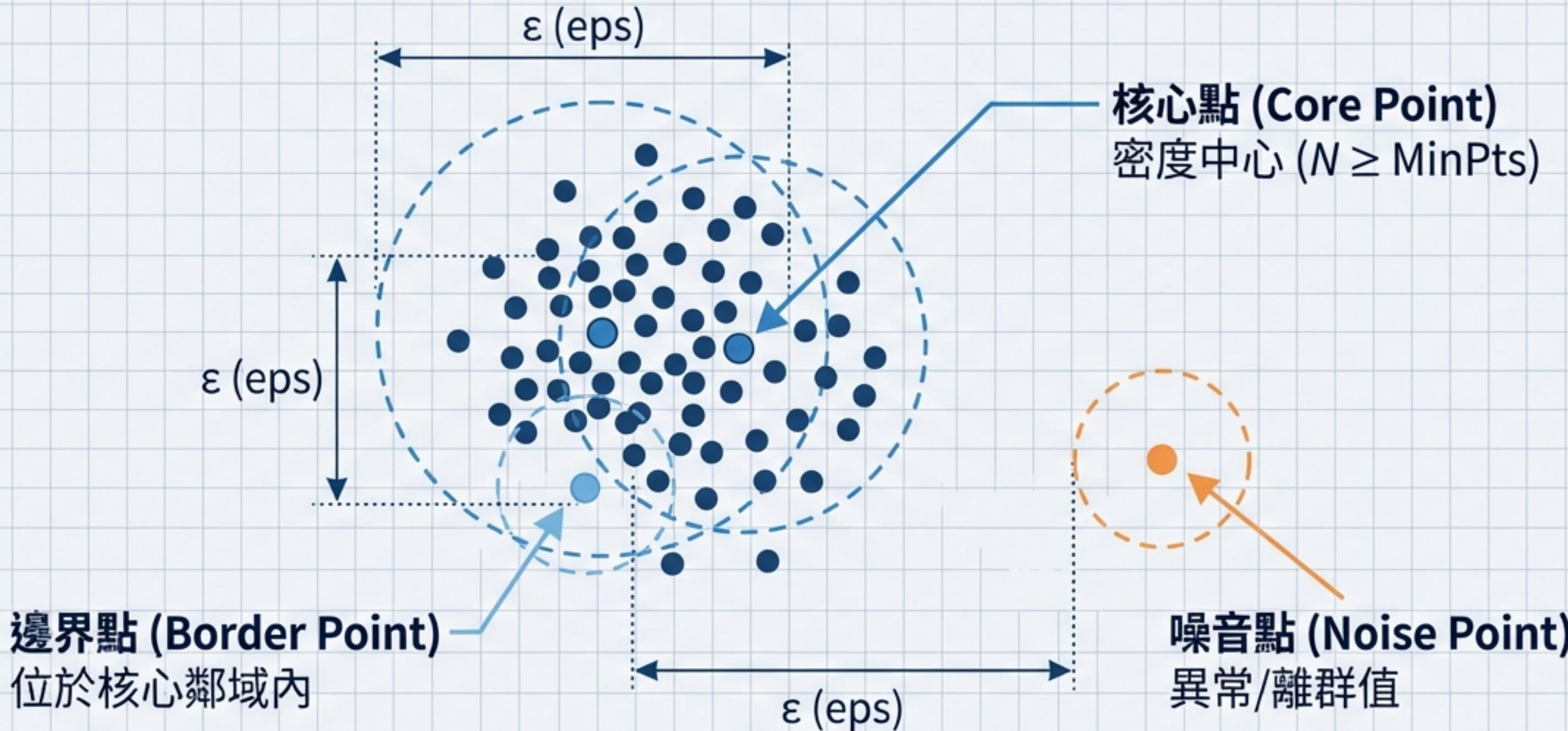
形成核心點所需的最小鄰居數 (含自己)

化工類比：

確認穩定操作所需的最小樣本數

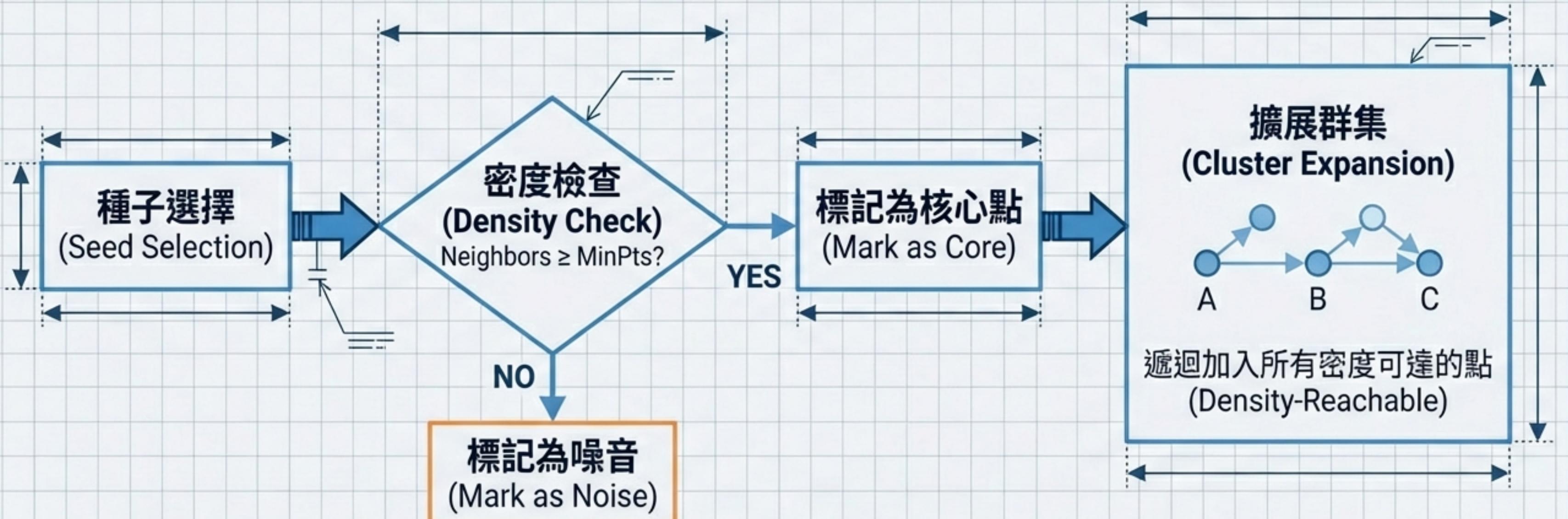


# 數據點分類：核心、邊界與噪音



Insight: DBSCAN 的強大之處在於它允許『噪音』的存在，而不是強迫將所有點歸類。

# 演算法機制：密度連接與擴展



直接密度可達： $P \rightarrow Q$  (一步)  $\rightarrow$

密度連接： $P \leftrightarrow Q$  (經由核心點鏈結)

# Scikit-learn 實作：建立模型



```
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler

# 1. 數據標準化（關鍵步驟！）
X_scaled = StandardScaler().fit_transform(X_raw)

# 2. 初始化模型
# eps: 鄰域半徑，min_samples: 最小鄰居數
dbscan = DBSCAN(eps=0.5, min_samples=5)

# 3. 摊合與預測
clusters = dbscan.fit_predict(X_scaled)

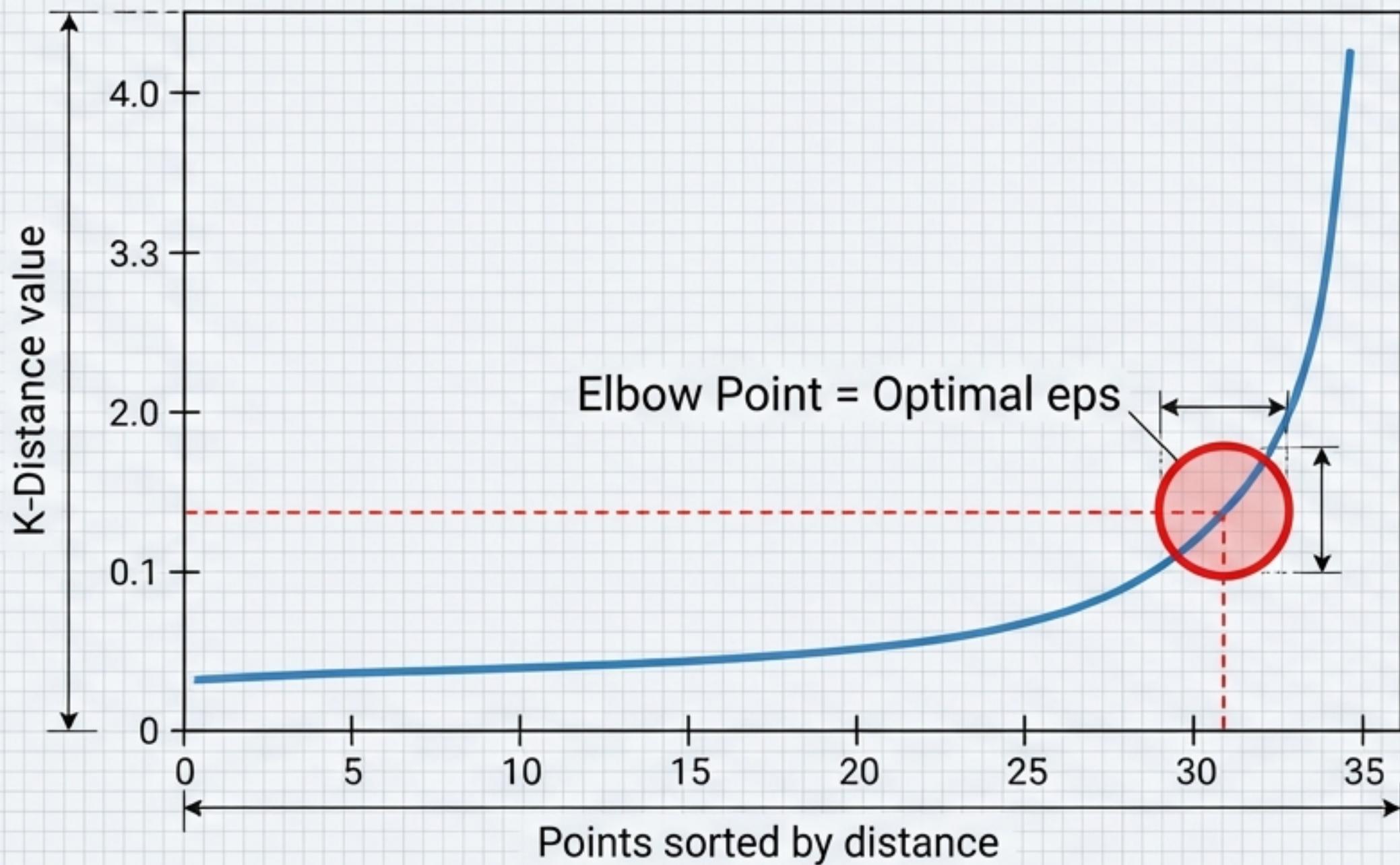
# labels_: -1 代表噪音點
print(dbscan.labels_)
```



重要提示：DBSCAN 基於距離計算，務必先進行數據標準化 (Standardization) !

# 參數調優 (1)：選擇 $\text{eps} (\varepsilon)$

## K-Distance Graph (K-距離圖法)



### 操作步驟

- 1. 計算每個點到第  $k$  個最近鄰居的距離 ( $k = \text{MinPts}$ )
- 2. 排序並繪製圖表
- 3. 尋找『肘部』(Elbow) 轉折點

化工經驗法則：亦可依據製程容許誤差（例如  $\pm 5^\circ\text{C}$ ）設定  $\text{eps}$ 。

## 參數調優 (2)：選擇 min\_samples

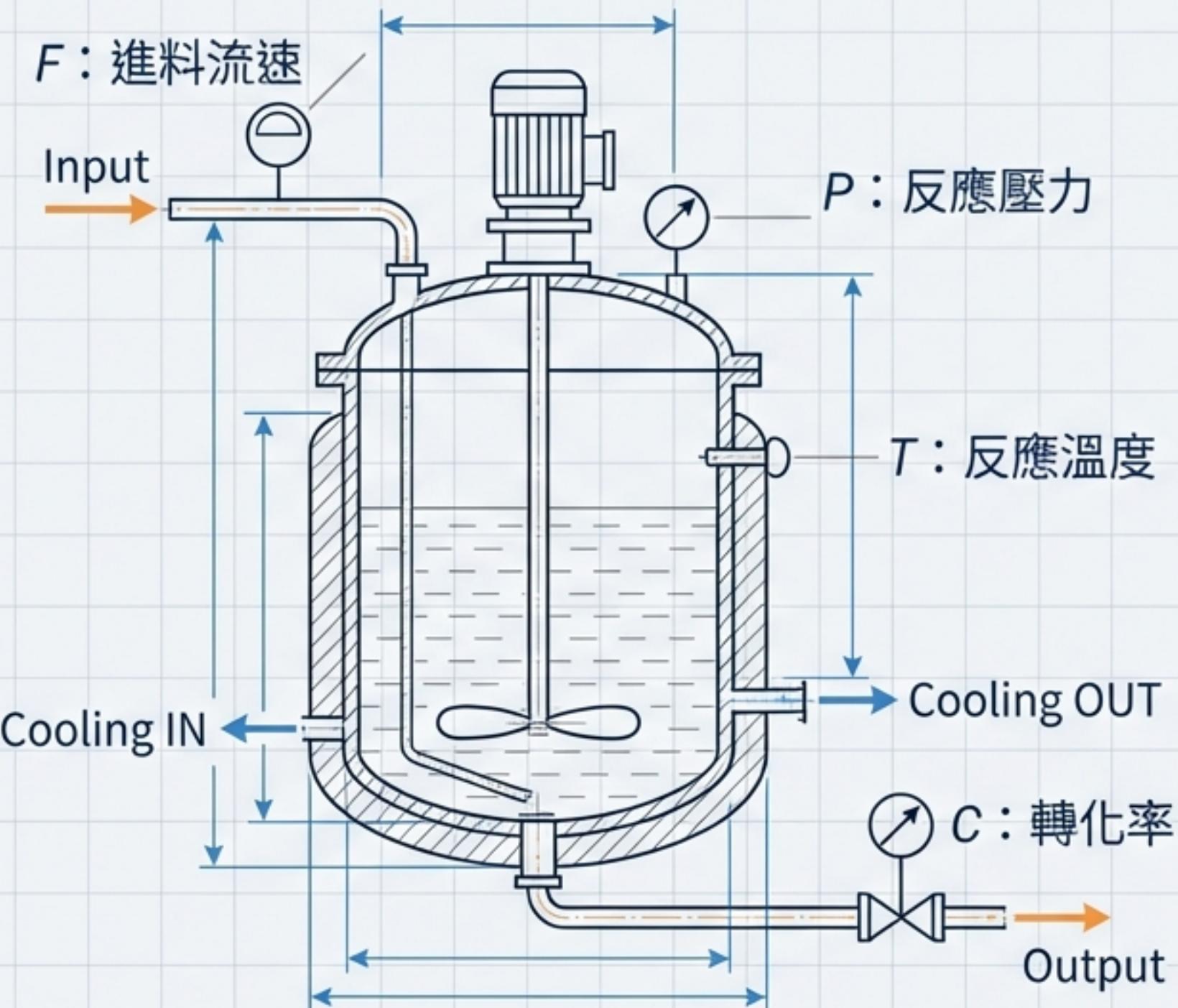
# 基本法則 (General Rule)

若特徵維度  $D > 2$ ，建議  $MinPts \geq 2 \times D$



The diagram features a large, dark blue vertical double-headed arrow centered on a grid background. The arrow points upwards on the left and downwards on the right, symbolizing a trade-off. To the left of the arrow, the text "High MinPts = 更少群集，更多噪音" is written vertically. To the right of the arrow, the text "Low MinPts = 更多群集，更少噪音" is written vertically.

# 應用案例：CSTR 反應器異常檢測



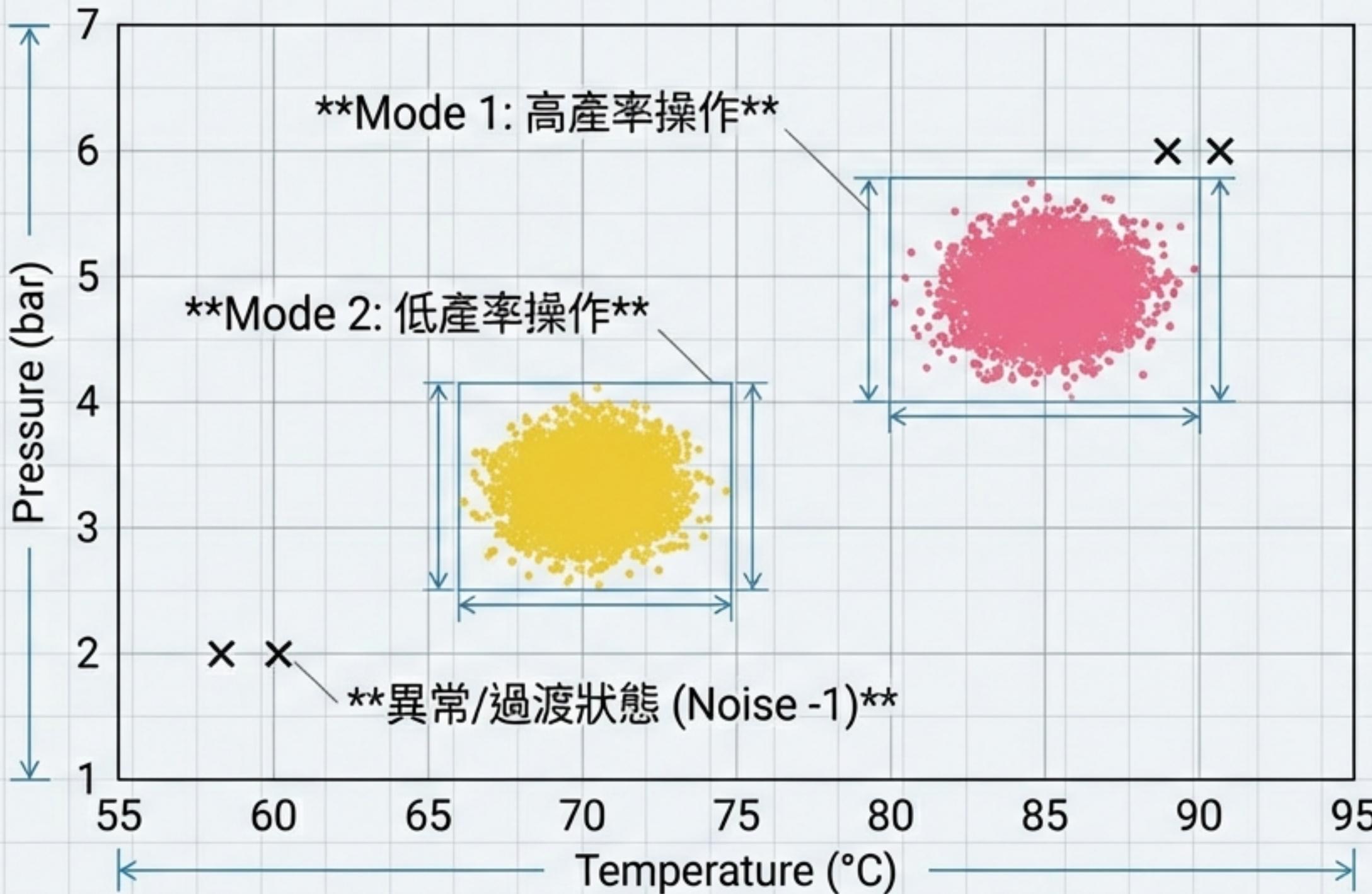
## 問題陳述 (Problem Statement)

- 目標：在無標籤數據的情況下，自動區分正常操作模式與異常偏移。
- 挑戰：無法預知異常發生的形式與頻率。

## 輸入特徵

Temperature ( $T$ ), Pressure ( $P$ ),  
Flow Rate ( $F$ ), Conversion ( $C$ )

# 分群結果視覺化：操作視窗



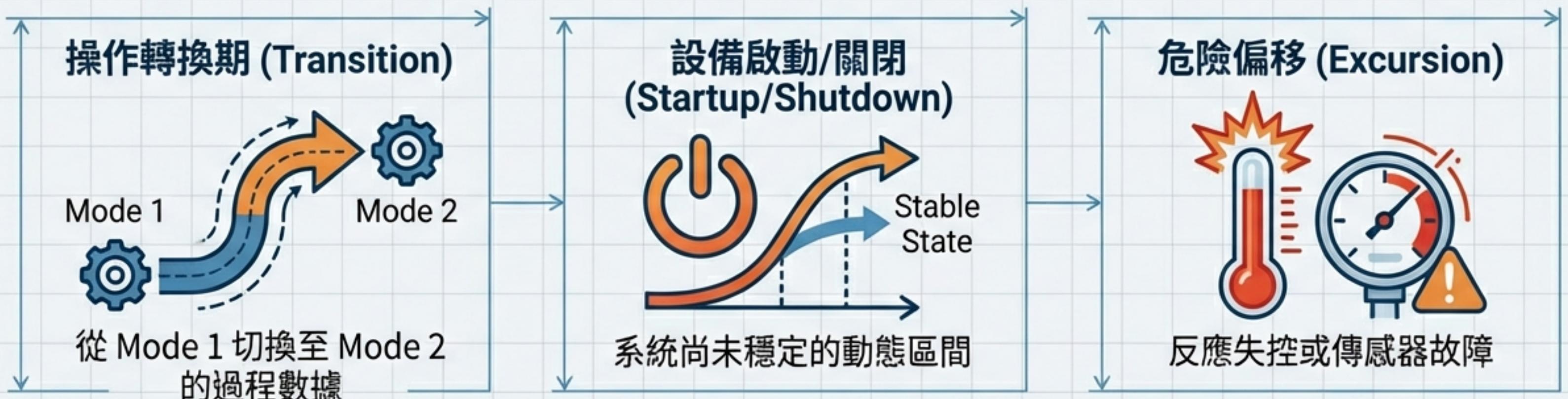
## 模型成效

- Accuracy: > 96%
- Noise Ratio: ~13%
- 結論：DBSCAN 成功分離出兩個主要操作區間，並隔離了異常點。

# 噪音分析：解讀異常訊號

## Noise ≠ Trash

在化工領域，噪音往往是最具價值的訊號。



Action Item: 將噪音點導入『**早期預警系統**』，即時通知操作員。

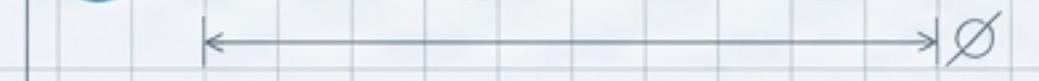
# 技術對比：DBSCAN vs. K-Means

特性 (Feature)	DBSCAN (本單元)	K-Means (傳統方法)
群集形狀	任意形狀 (Arbitrary)	球形 (Spherical)
噪音容忍	<b>強 (Robust) - 自動排除</b>	<b>弱 (Sensitive) - 受離群值影響</b>
參數需求	$\epsilon$ , MinPts	K (群集數量)
運作原理	密度連接 (Density-based)	距離中心 (Centroid-based)
最佳用途	異常檢測、非線性結構	數據壓縮、一般分群

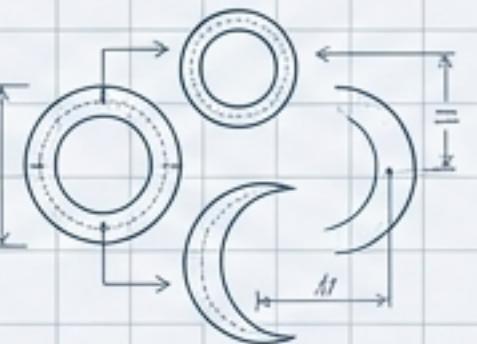
# 優勢與局限

## 優勢 (Pros)

⊕ 無需預先猜測  $K$  值



⊕ 能處理複雜幾何形狀  
(如環狀、新月形)

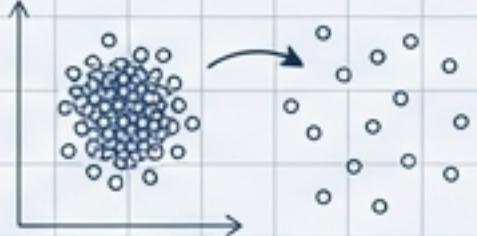


⊕ 優秀的異常值過濾能力

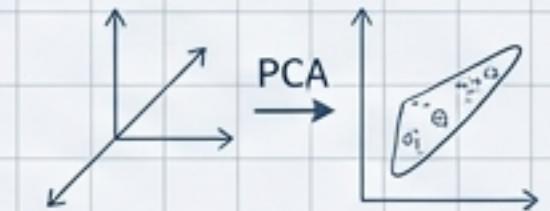


## 局限 (Cons)

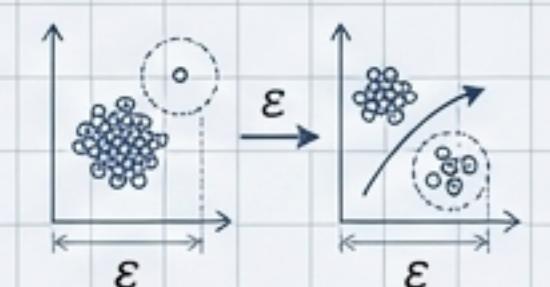
⊖ 密度不均：難以處理密度差異巨大的不同群集



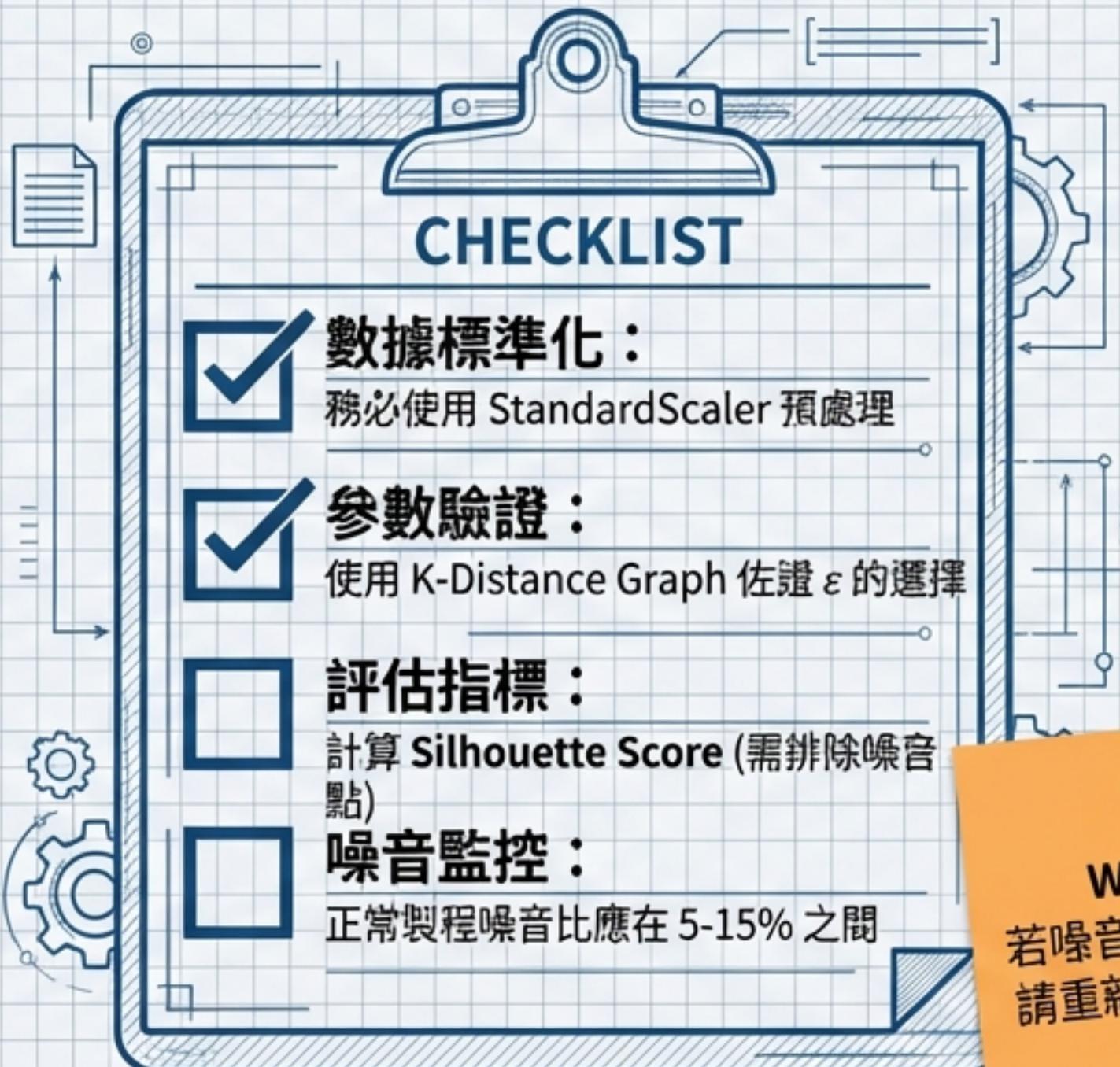
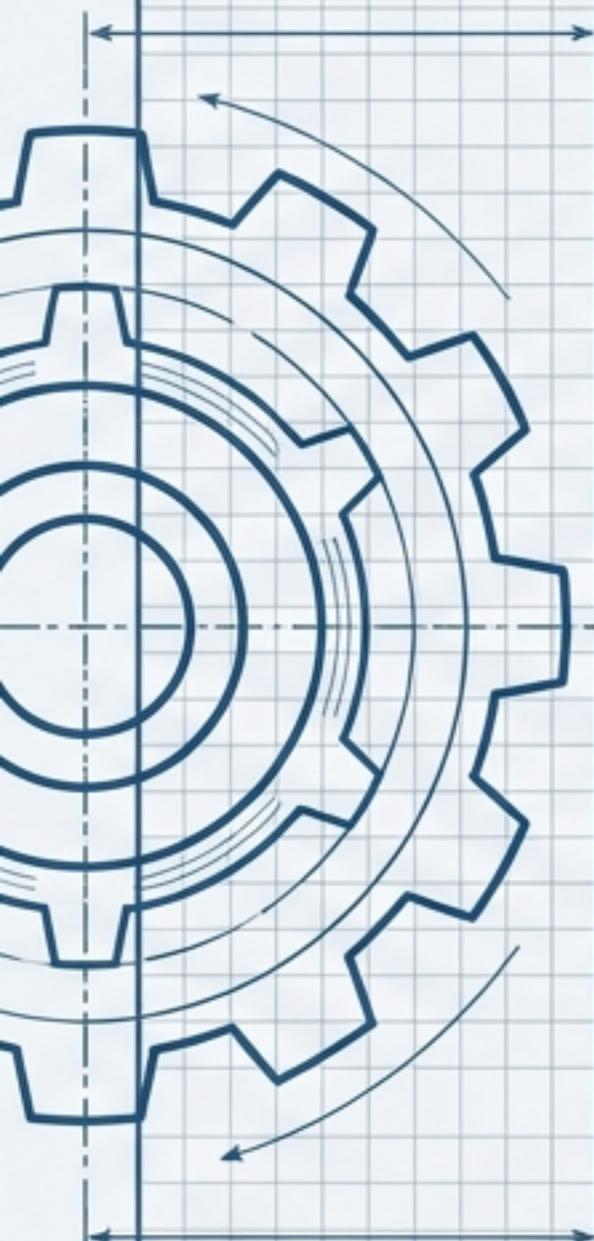
⊖ 維度災難：高維度數據需先降維 (PCA)



⊖ 參數敏感： $\epsilon$  的微小變動可能導致結果劇變



# 工程最佳實踐清單



Warning:  
若噪音比例 > 30%，  
請重新調整參數！

*"Code is read much more often than it is written."*

# 總結與下一步

