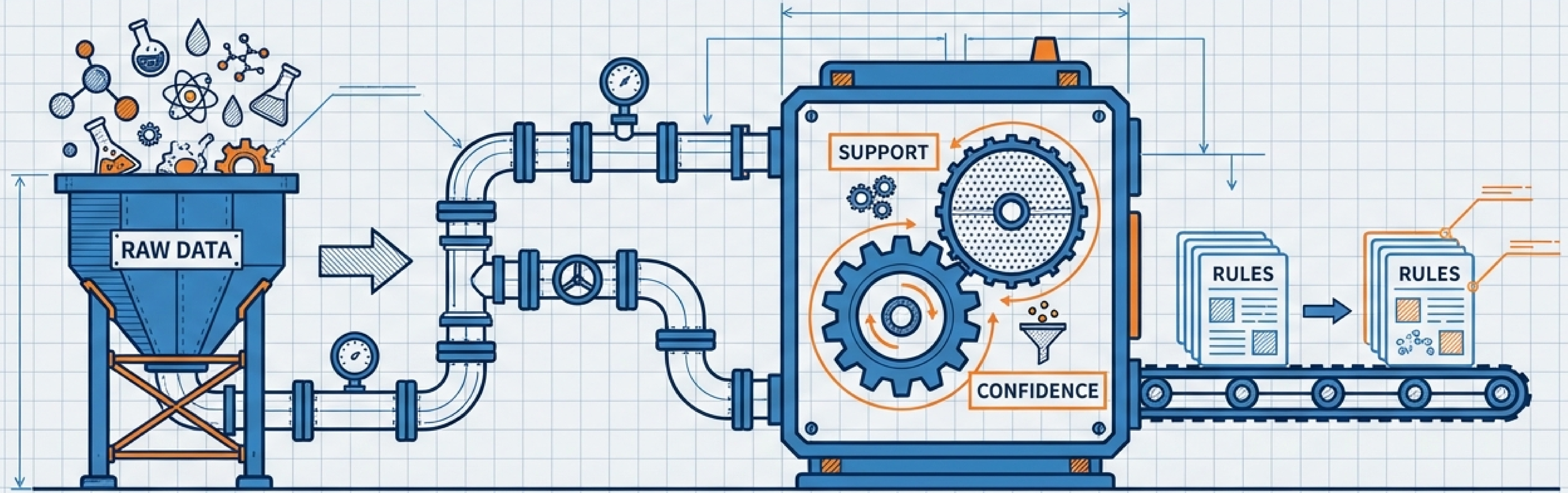
Unit 08: Apriori Algorithm (Apriori 演算法)

關聯規則學習：從化工配方數據中提煉黃金法則



課程：AI在化工上之應用 | 講師：莊曜禎 助理教授

配方開發的挑戰：如何在無限組合中找到完美配方？

化工配方變數 (Variables)

成分 (Ingredients):
- 單體 (Monomer)
- 引發劑 (Initiator)
- 溶劑 (Solvent)

條件 (Conditions):
- 溫度 (Temperature)
- 壓力 (Pressure)
- 時間 (Time)

目標 (Goal):
- 尋找產生「高品質」產品的特定組合

Recipe Matrix

	Roboto Mono Monomer A	Roboto Mono Solvent X	Roboto Mono High Temp	Roboto Mono High Yield
Exp_01	1	0	1	0
Exp_02	0	1	1	1
Exp_03	1	1	0	0
Exp_04	0	1	1	1

我們能否**自動挖掘**出隱藏在實驗記錄中的**「成功模式」**？

Apriori 演算法：利用「先驗」知識的規則挖掘引擎

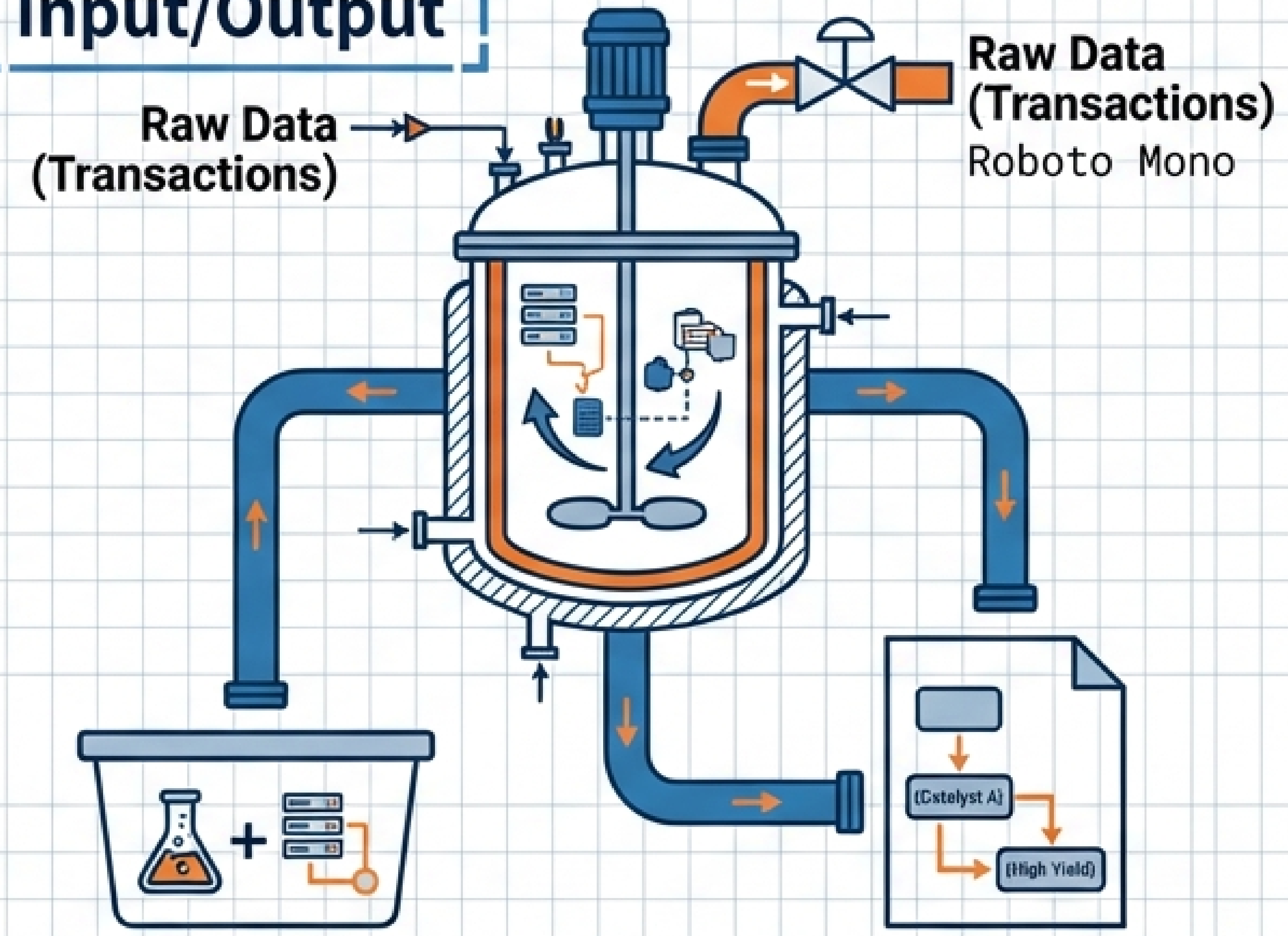
Definition (定義)

- 起源：1994, Agrawal & Srikant
  Roboto Mono
- 語源：Latin “a priori” (from the prior)
  Roboto Mono

核心概念：利用先前發現的頻繁項目集，來產生下一步的候選項目集。

Noto Sans TC

Input/Output

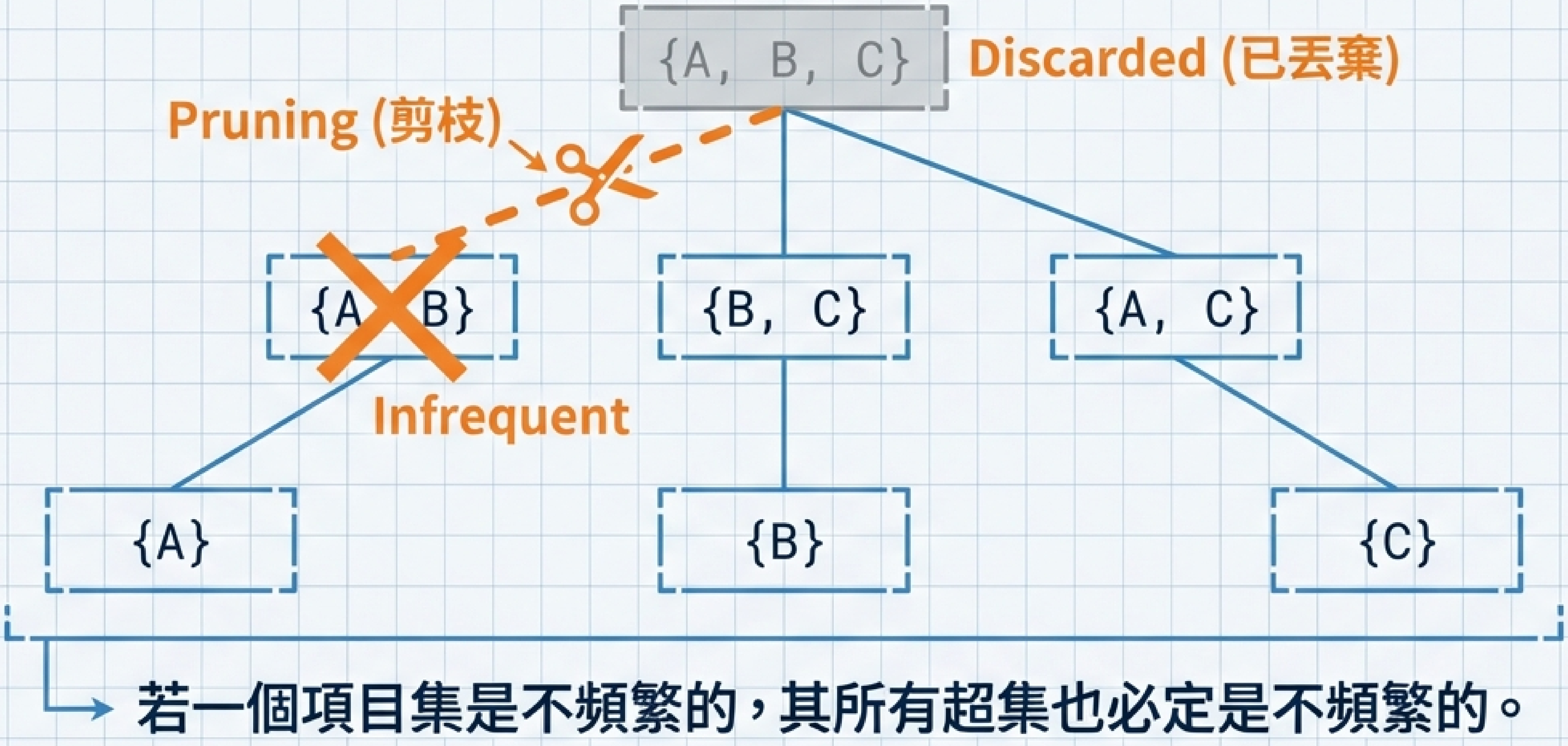


1. 頻繁項目集
(Frequent Itemsets)
經常一起出現的成分組合
(e.g., {Catalyst A, Solvent X})

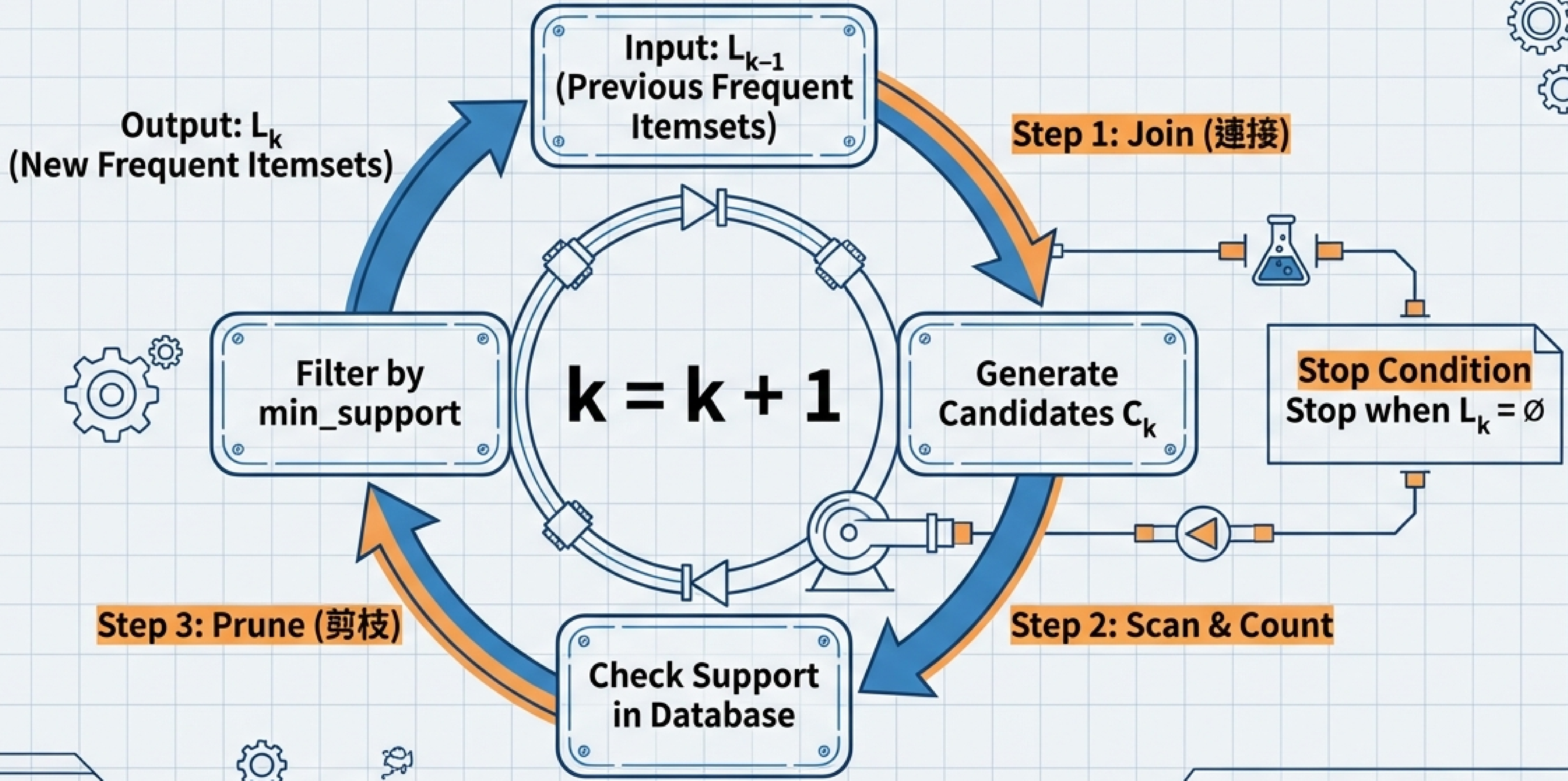
2. 關聯規則
(Association Rules)
因果或強相關的推論
(e.g., {Catalyst A} ⇒ {High Yield})

核心機制：Apriori 性質與剪枝 (Pruning)

向下封閉性 (Downward Closure Property) 確保運算效率



運作流程：逐層搜索的迭代循環 (Level-wise Search)



執行實例：從原始數據到頻繁項目集

Transactions
T1: {A, X, High}
T2: {A, Y, High}
T3: {B, X, Low}
T4: {A, X, High}
T5: {A, X}

Stage 1: L1
(1-Itemsets)

{A} ✓
{B} ✗
{X}
{Y} ✗
{High} ✓

Stage 2: C2
(Candidates)

{A, X}
{A, High}
{X, High}

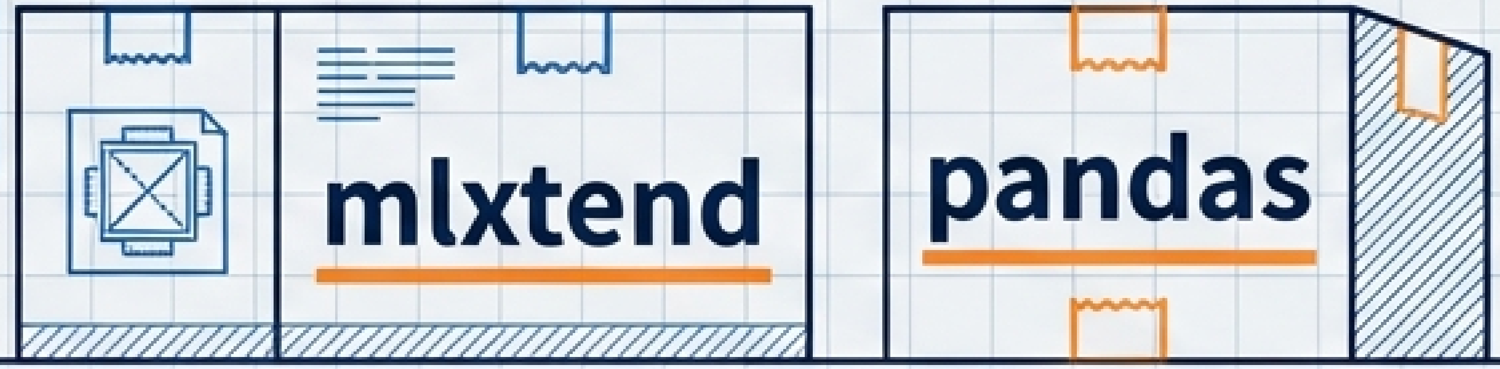
Stage 3: L2
(Frequent 2-Itemsets)

{A, X}: 3 ✓
{A, High}: 3 ✓
{X, High}: 2 ✓

Stage 4: L3
(Frequent 3-Itemsets)

{A, X, High}
Count: 2

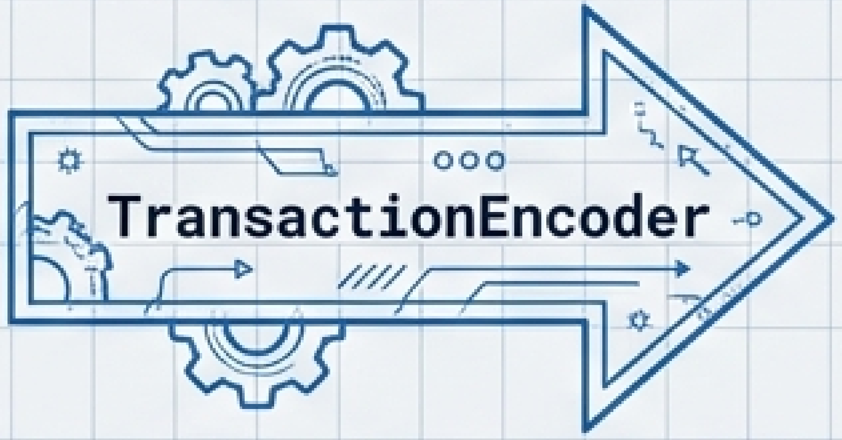
實作工具：Python 與 mlxtend 套件



Transaction List

```
[['A', 'B'],
 ['B', 'C']]
```

Data Preparation



One-Hot Encoding (DataFrame)

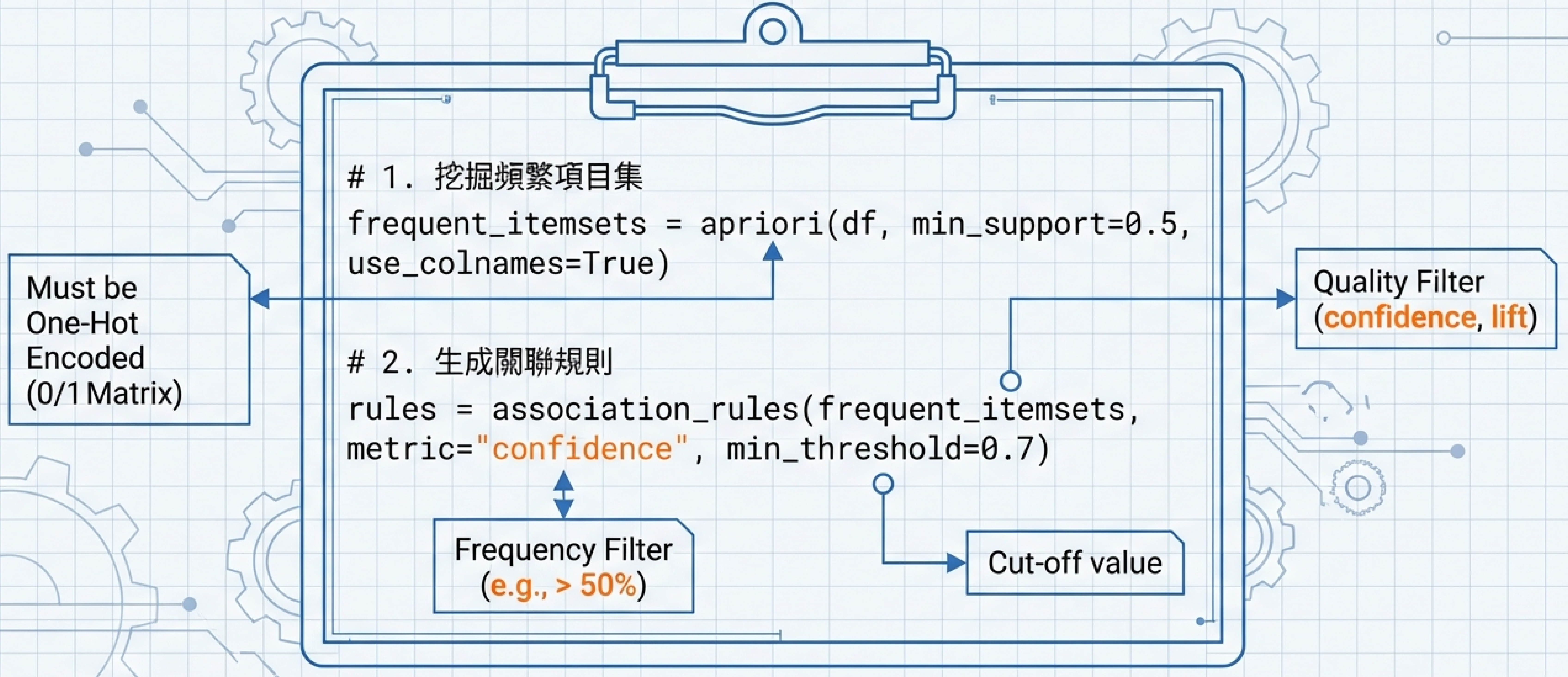
	A	B	C
1	1	1	0
2	0	1	1

```
from mlxtend.frequent_patterns import apriori
from mlxtend.preprocessing import TransactionEncoder
```

程式語法：操作指令與參數設定

```
# 1. 挖掘頻繁項目集
frequent_itemsets = apriori(df, min_support=0.5,
use_colnames=True)

# 2. 生成關聯規則
rules = association_rules(frequent_itemsets,
metric="confidence", min_threshold=0.7)
```



過程控制指標：支持度、置信度與提升度

$$\text{Lift} = \frac{\text{Confidence}}{\text{Support(Consequent)}}$$

Lift > 1 : Effective Rule (Signal)

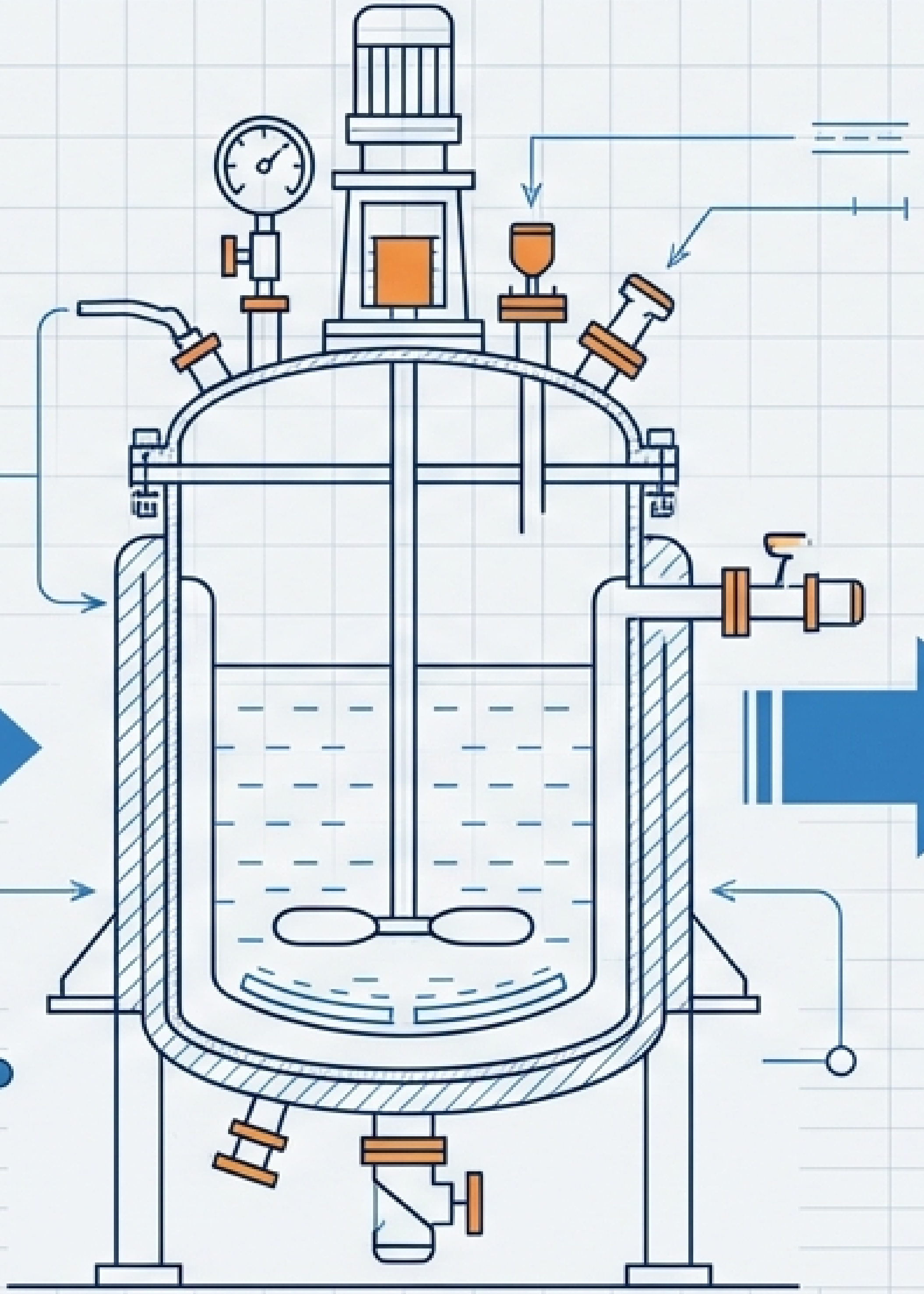
Lift = 1 : Independent (Noise)

應用案例：聚合物配方優化 (Polymer Optimization)

Dataset: 200
Historical Batches

Input Variables (29 Items)

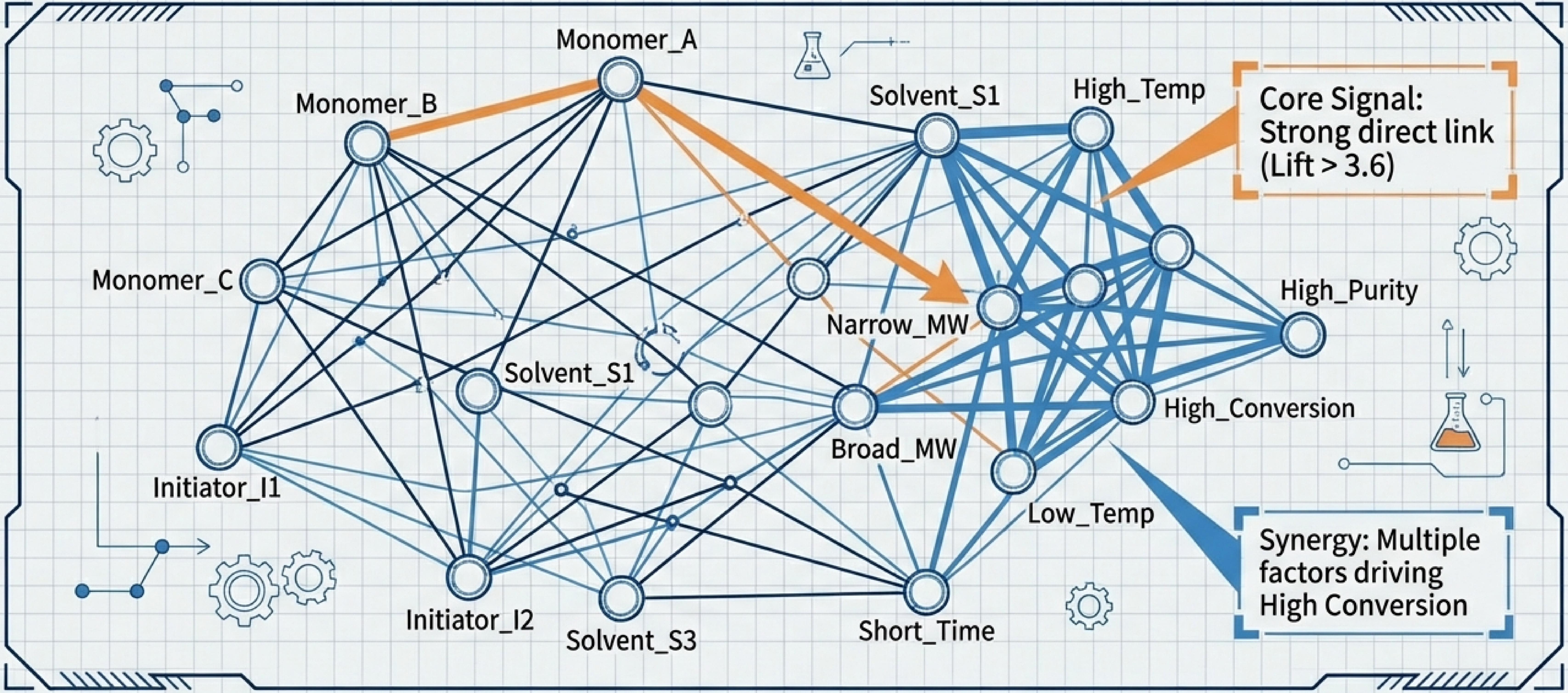
- Monomer (A–E)
- Initiator (I1–I3)
- Solvent (S1–S3)
- Conditions: Temp, Time



Target Outputs

- Narrow MW Distribution (窄分子量分布)
- High Conversion (高轉化率)
- High Purity (高純度)

視覺化洞察：配方關聯網絡圖



網絡密集區域代表高交互作用的配方空間 (High Interaction Space)

關鍵發現：從數據中提取的黃金法則

Narrow MW Distribution

{Monomer_A + Initiator_I1} ⇒ {Narrow MW}

Lift: 3.66 (Strong Synergy)

High Conversion

{Solvent_S1 + High Temp} ⇒ {High Conversion}

Lift: 3.75

High Purity

{Chain Transfer T2 + Med Temp} ⇒ {High Purity}

Lift: 3.25

業務行動建議：從數據洞察到製程決策

STEP 1

1. 配方標準化

Create standard templates based on High Lift rules.

STEP 2

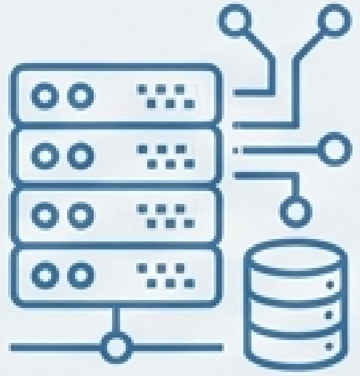
2. 實驗驗證

Lab tests for "Solvent S1 + High Temp" to confirm safety.

STEP 3

3. 成本效益

Compare Solvent S1 vs S2 cost/yield ratio.



STEP 4

4. 知識庫建立

Store 368 high-quality rules for R&D retrieval.

演算法選擇：Apriori vs. FP-Growth

Apriori Algorithm

Pros

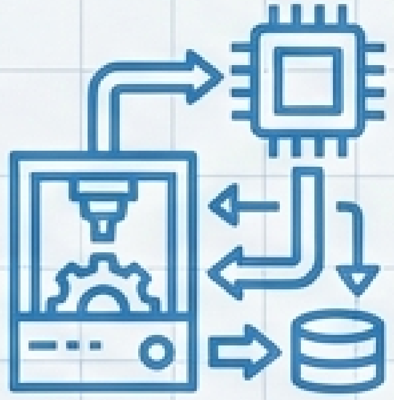
- Simple logic
- Interpretable
- Easy implementation

Cons

- Slow on Big Data
- High memory usage

Verdict
Use for Teaching & Small Recipe Analysis

FP-Growth Algorithm



Pros

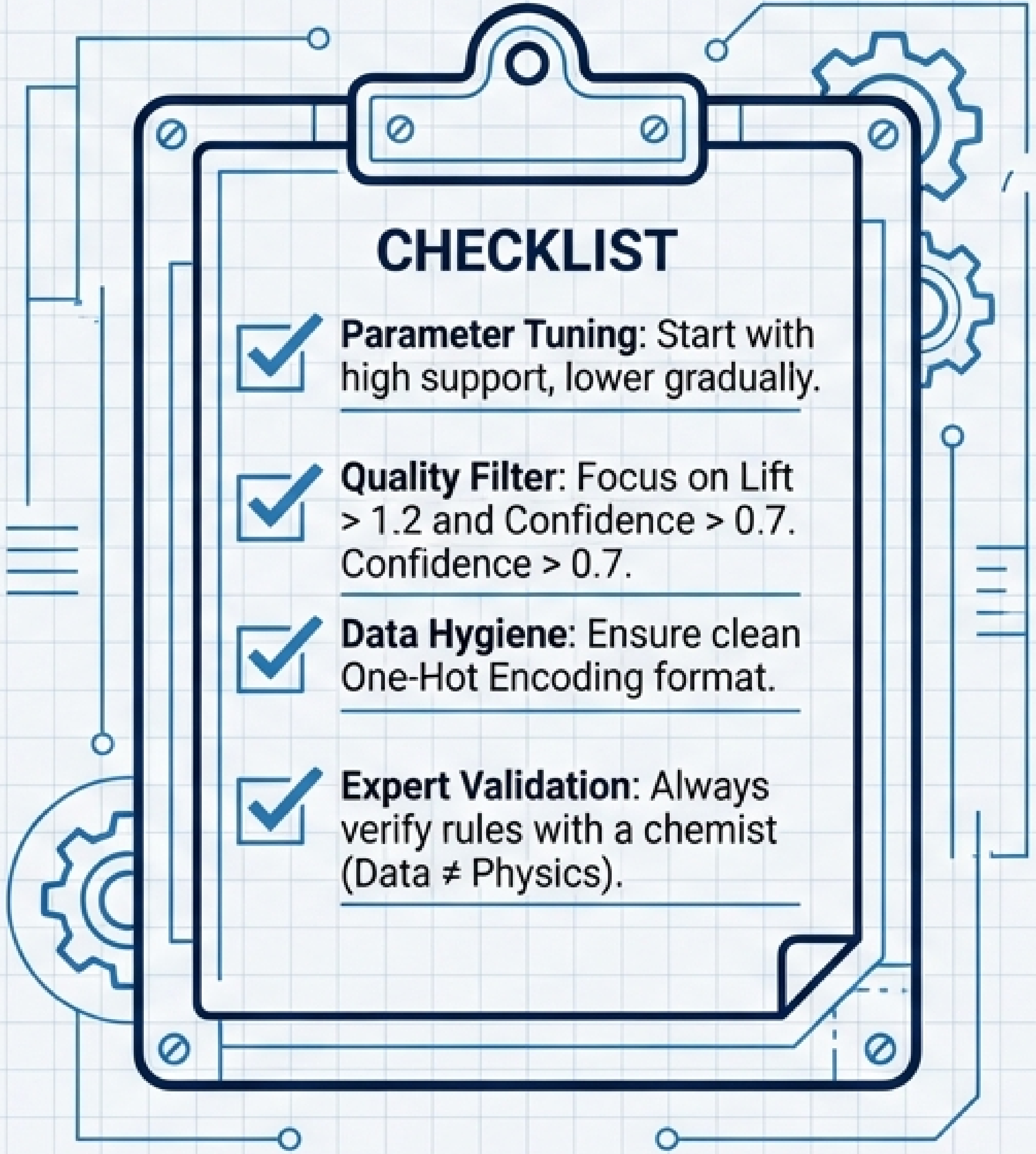
- Fast (Tree-based)
- Efficient (2 scans)

Cons

- Complex structure (FP-Tree)
- Harder to code

Verdict
Use for Massive Process Logs

最佳實踐與總結 (Summary & Best Practices)



Next Step: Unit 09
FP-Growth Algorithm
更高效的大數據解決方案