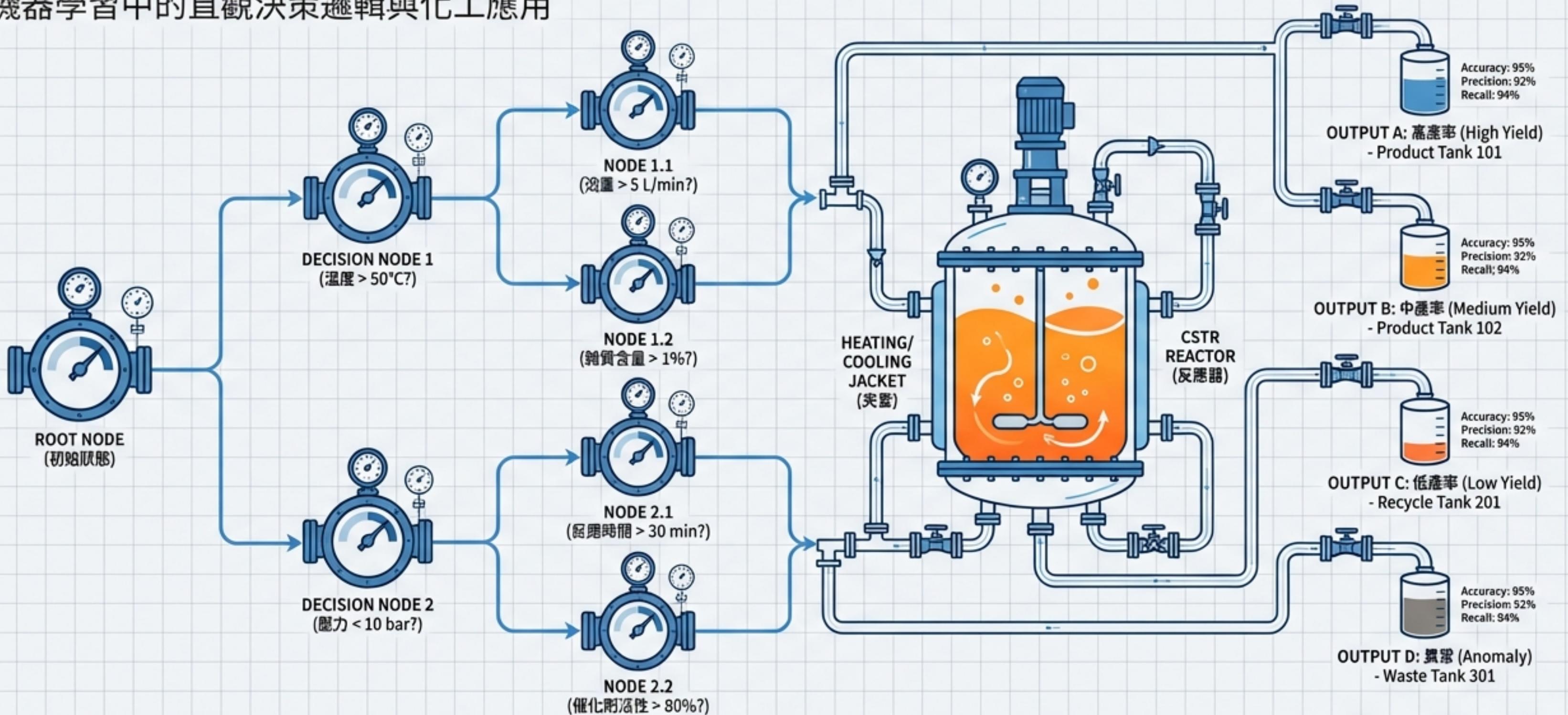


# Unit 12 決策樹分類 | Decision Tree Classifier

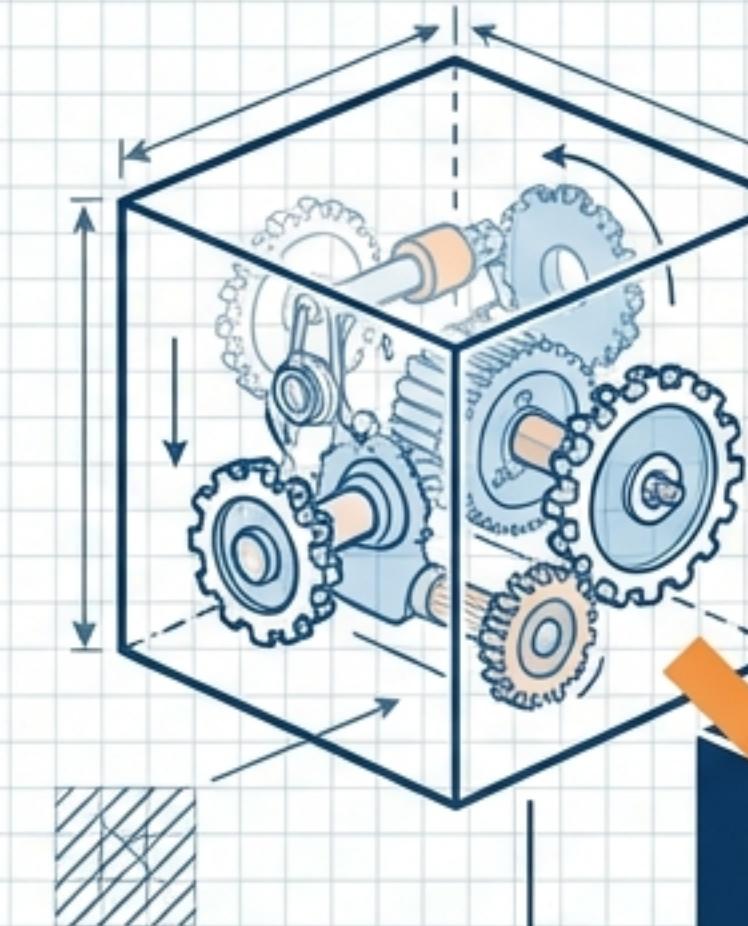
機器學習中的直觀決策邏輯與化工應用



# 本單元學習目標 (Learning Objectives)

## 本單元學習目標

- 核心原理：理解 Gini 不純度與熵 (Entropy) 的物理意義。
- 模型控制：掌握防止過擬合 (Overfitting) 的剪枝技術 (Pruning)。
- 工具應用：Scikit-learn DecisionTreeClassifier 參數實戰。
- 化工實例：化工反應成功率預測與特徵重要性分析。



**WHITE BOX MODEL  
(INTERPRETABILITY)**



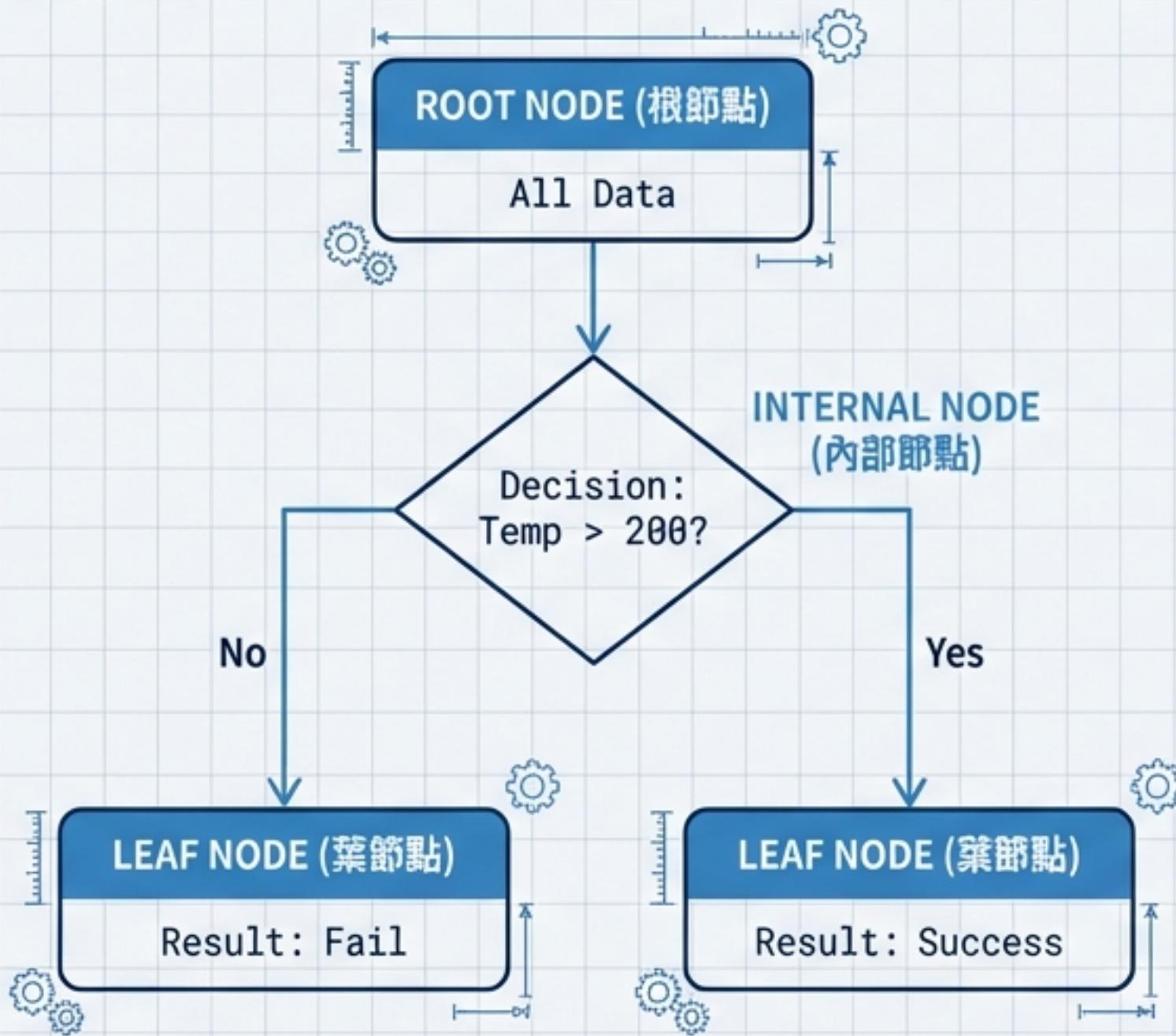
**BLACK BOX MODEL  
(OPAQUE)**

透明度與可解釋性  
(Transparency & Explainability)

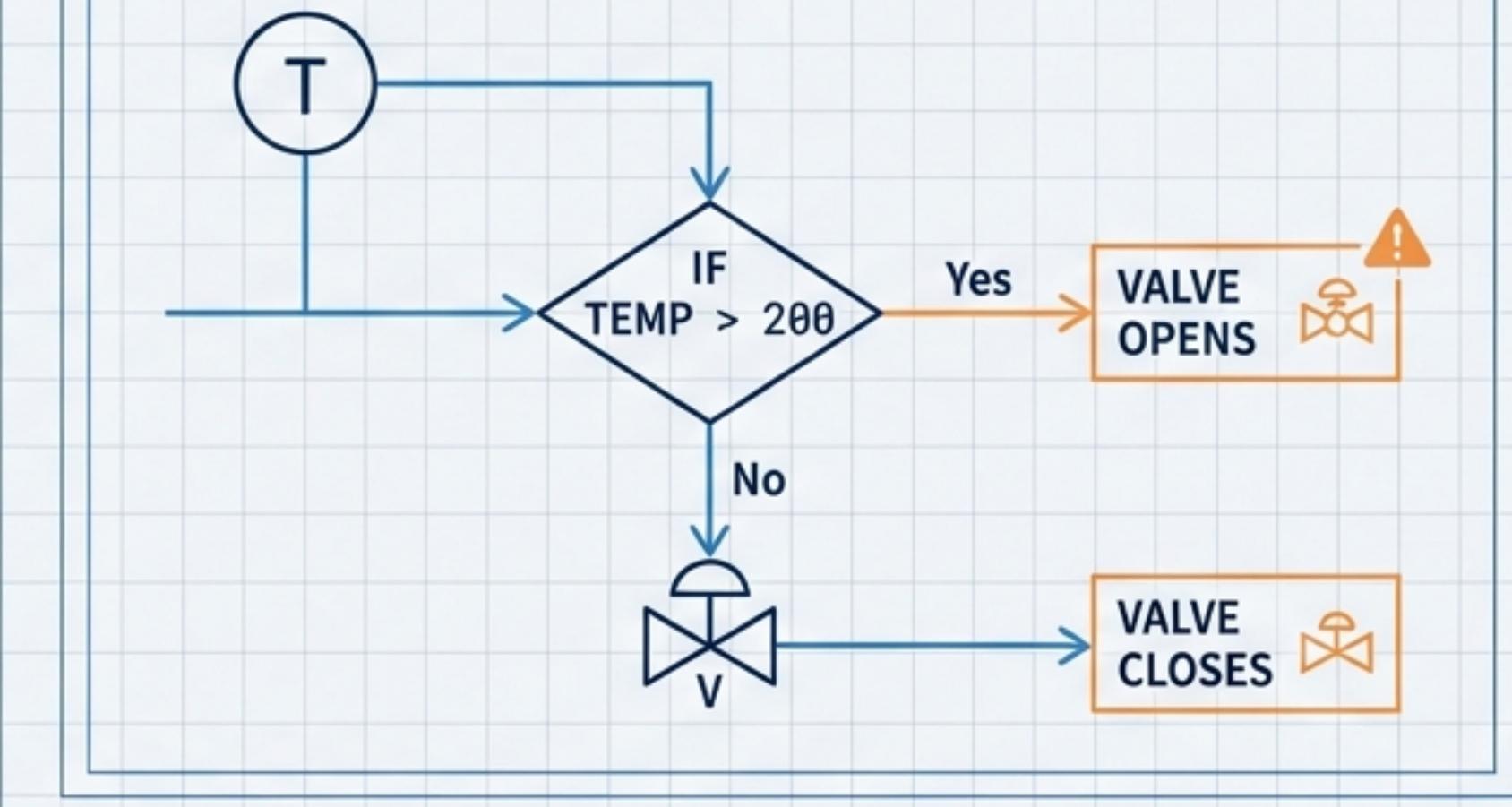
不透明與難以解釋  
(Opaque & Hard to Explain)

# 決策樹：模仿真人思維的分類器

## 決策樹結構



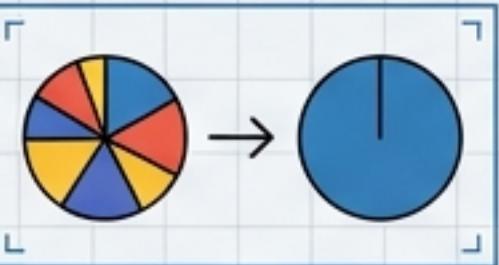
## 工業類比：安全連鎖系統



### 關鍵概念 (KEY CONCEPTS)

- **遞歸分割 (Recursive Splitting)**：不斷將數據切分為更純淨的子集。
- **貪婪算法 (Greedy Algorithm)**：每一步都尋找當前最佳的分裂點。
- **白箱模型 (White Box)**：邏輯透明，工程師可完全追溯決策過程。

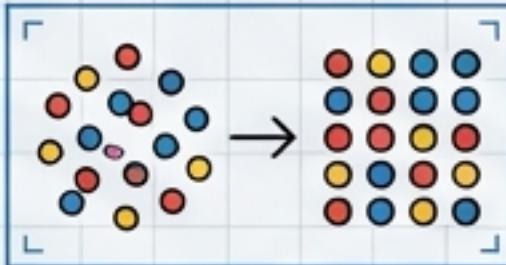
# 分裂標準：如何選擇最佳路徑？



Gini 不純度  
(Gini Impurity)

$$Gini = 1 - \sum_k (p_k)^2$$

- 定義：Sklearn 預設標準。目標是讓節點純度最大化 ( $Gini = 0$ )。
- 優勢：計算速度快（無需對數運算）。



熵 (Entropy)

$$Entropy = - \sum p_k \log_2(p_k)$$

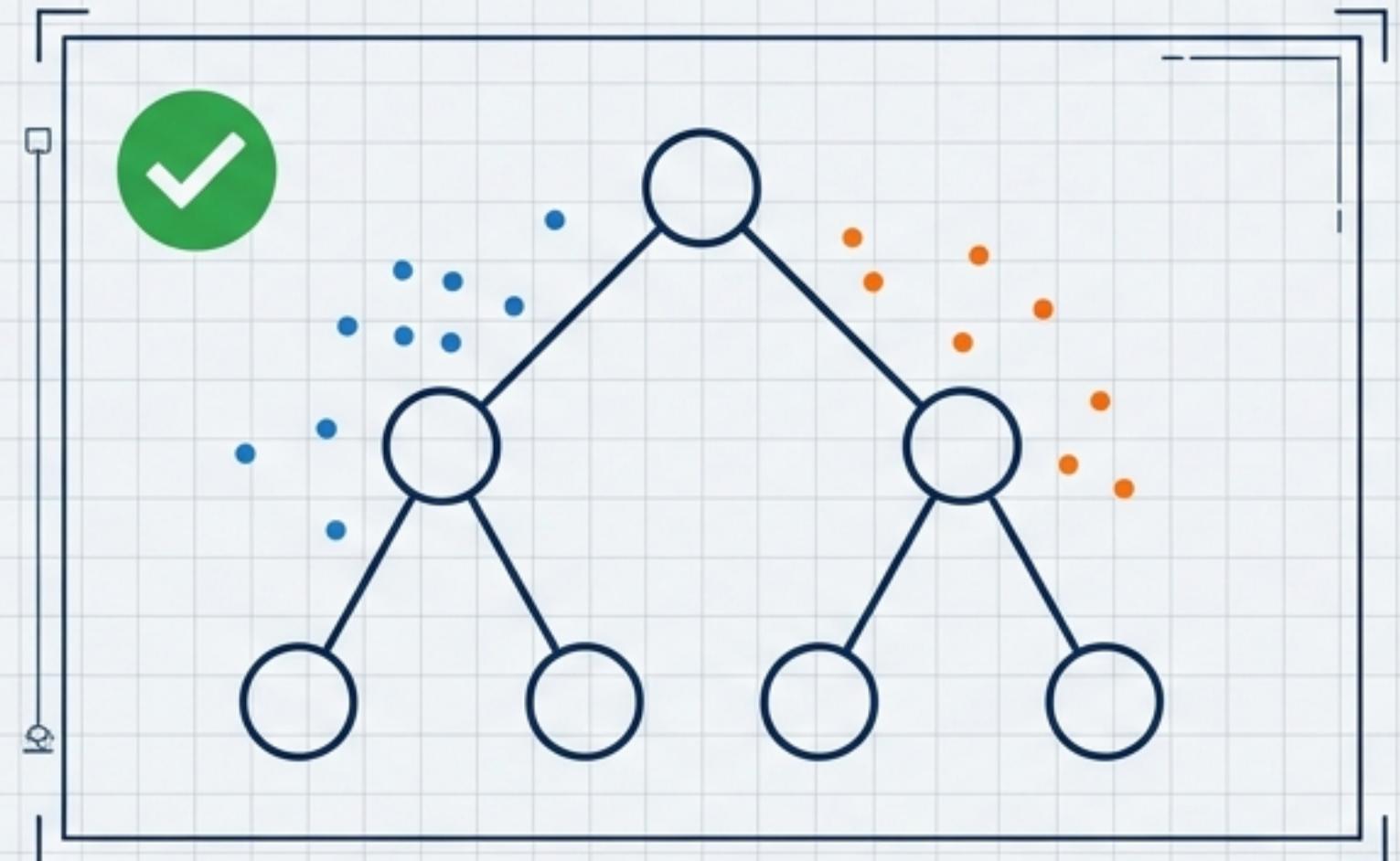
- 定義：來自信息論。目標是信息增益 (Information Gain) 最大化。
- 優勢：理論基礎嚴謹，傾向產生平衡的分裂。

## ENGINEERING NOTE

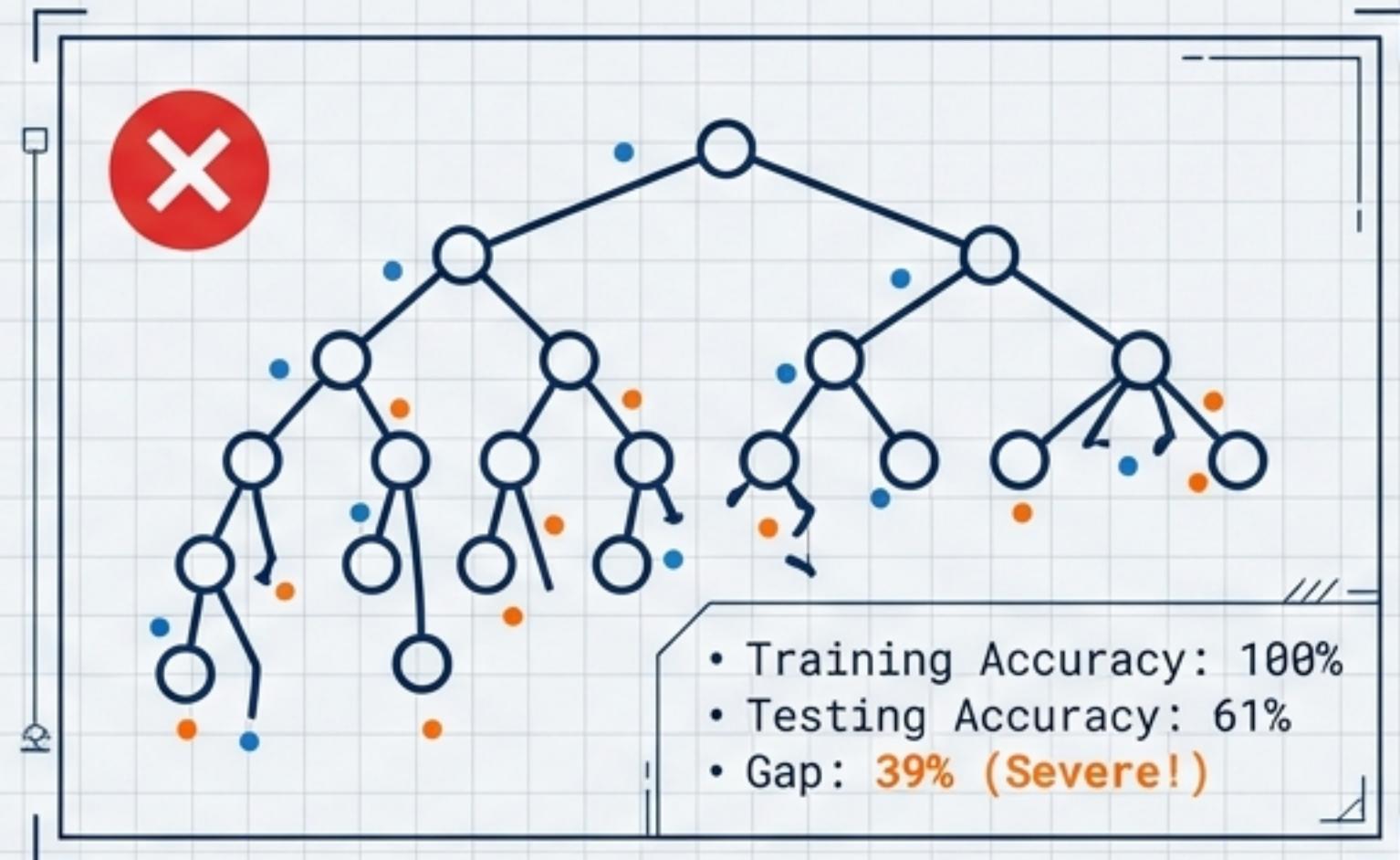
工程觀點：實務上兩者產生的樹結構差異甚微 (< 2%)。優先使用 Gini 以提升運算效率。



# 過擬合 (Overfitting)：記住了噪音，而非規律



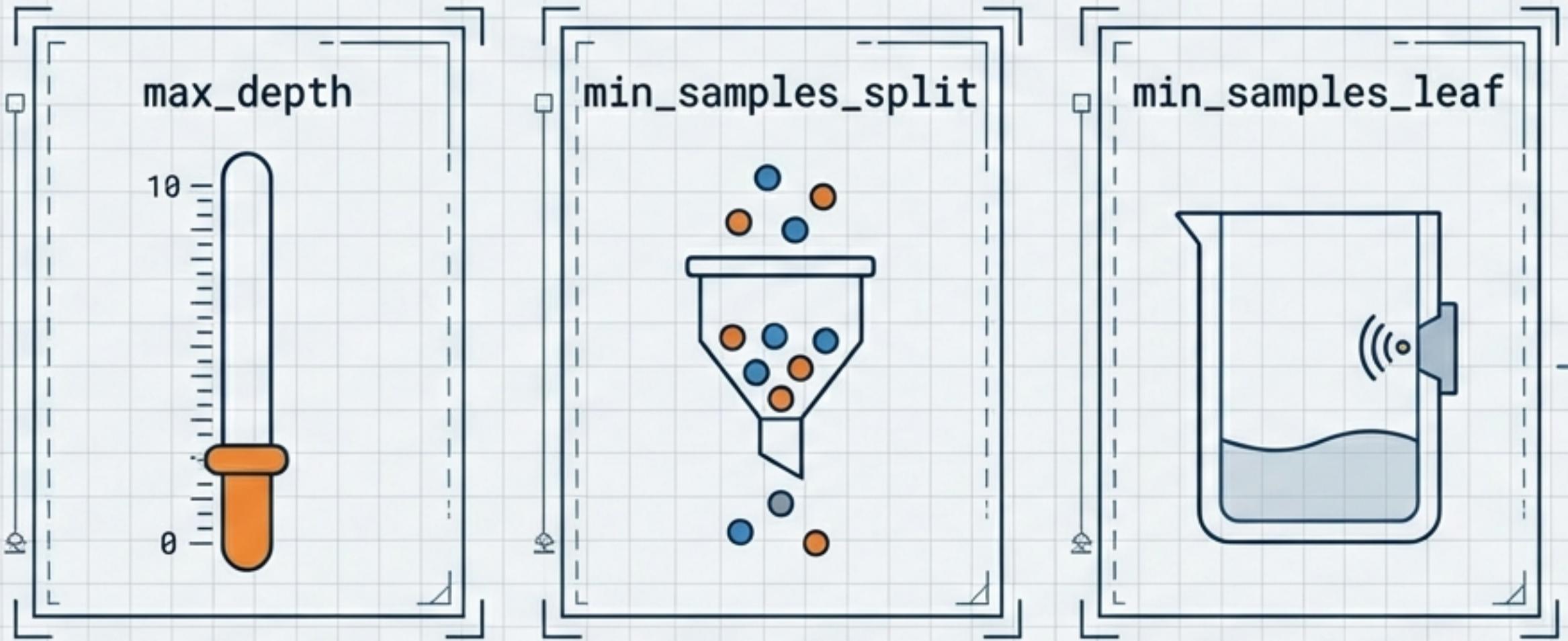
泛化模型 (Generalization)



過擬合模型 (Overfitting)

類比：就像為了配合一個錯誤的實驗數據點 (Outlier)，重新設計了整個工廠管線配置。

# 解決方案 I：預剪枝 (Pre-Pruning) — 設定生長限制



限制樹的最大高度  
(推薦範圍：3-10)。

節點分裂所需的  
最小樣本數。

葉節點必須保有的  
最小樣本數。

反應器最大壓力限制 (Max Pressure Limit)。

**Simple Tree  
(Depth=3)**  
測試準確率：**71.75%**  
(vs. **61% Unpruned**)

# 解決方案 II：後剪枝 (Post-Pruning) – 修剪冗餘枝葉

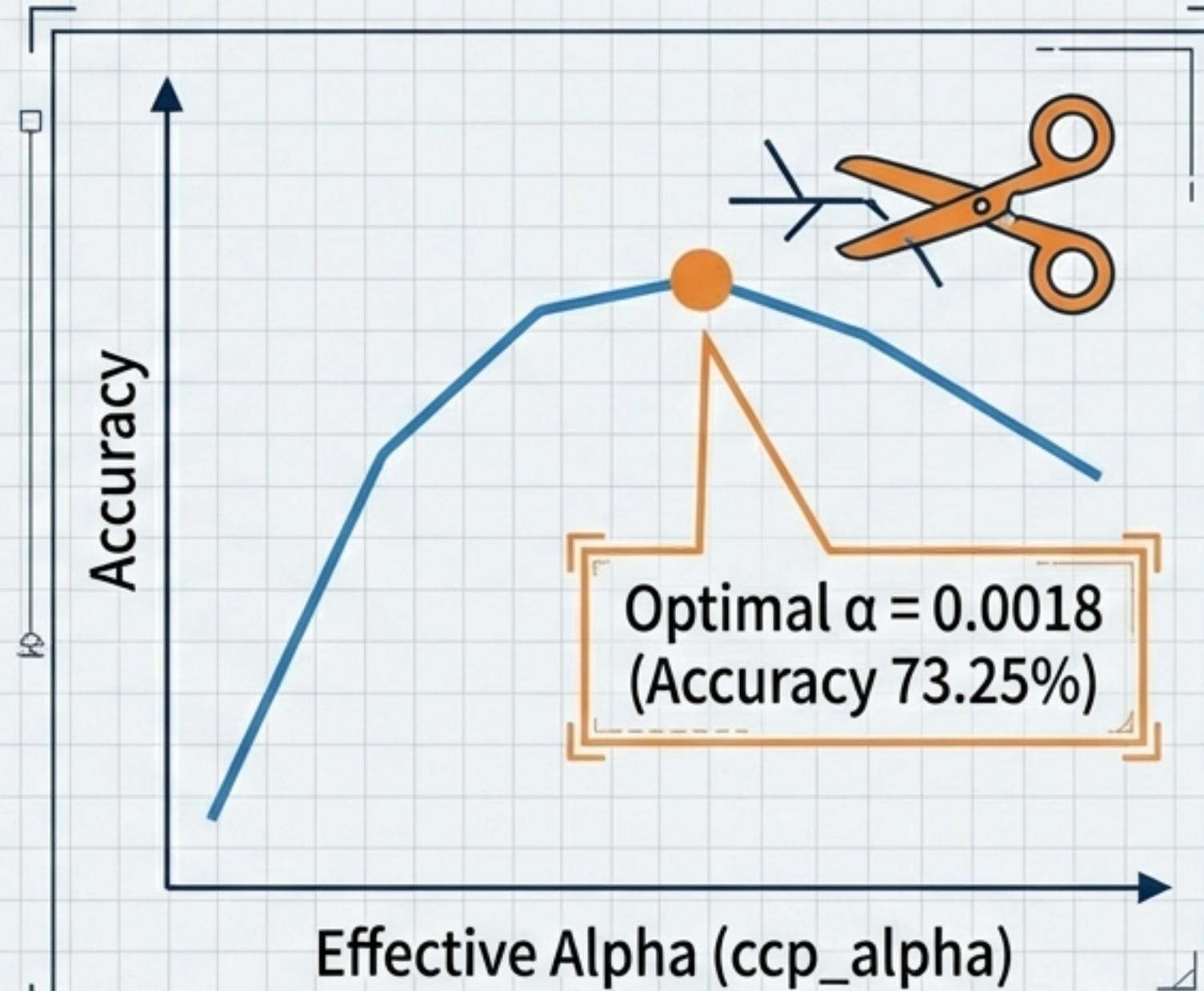
概念：先讓樹完全生長，再使用  
**成本複雜度剪枝 (CCP)** 移除對  
性能貢獻低的分支。

$$\text{Cost Function: } R_{-\alpha}(T) = R(T) + \alpha|T|$$

$R(T)$ : 誤差 (**Error**)

$|T|$ : 複雜度 (**Complexity**)

$\alpha$  : 懲罰係數 (**Penalty**)



# 程式實作：sklearn DecisionTreeClassifier

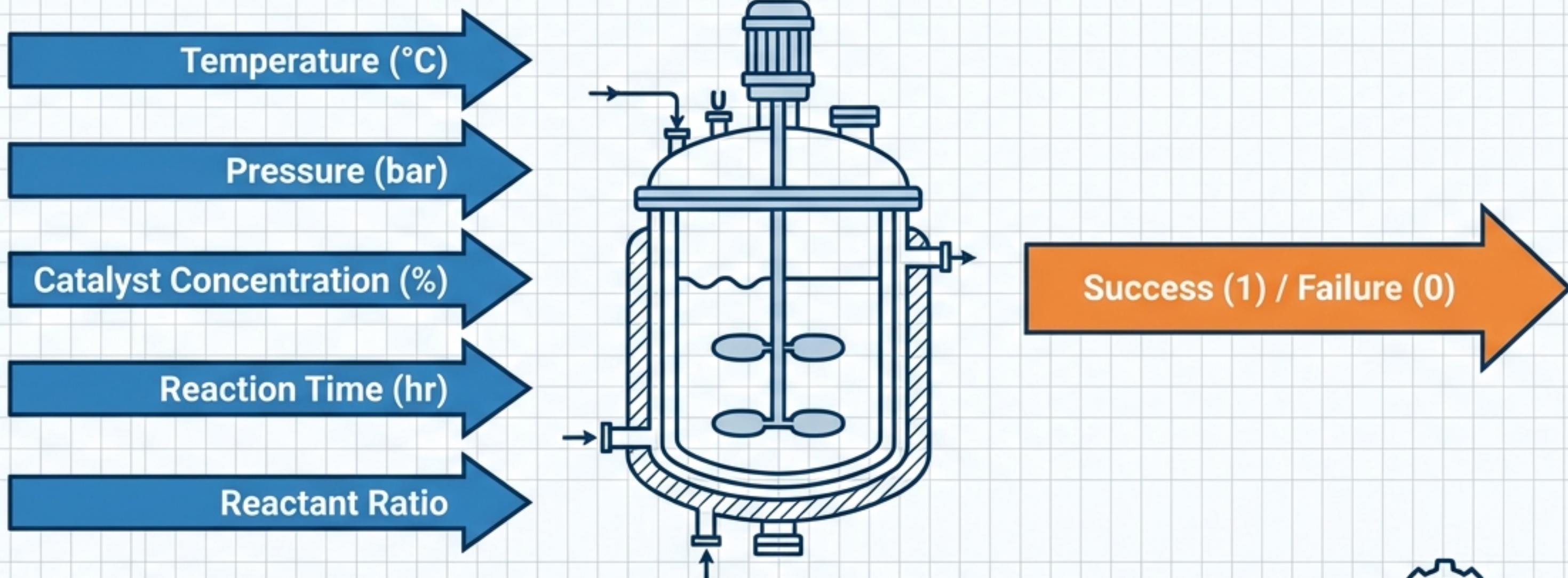
```
from sklearn.tree import DecisionTreeClassifier  
  
# 初始化模型（包含預剪枝與後剪枝參數）  
model = DecisionTreeClassifier(  
    criterion='gini', # 分裂標準  
    max_depth=3, # 預剪枝：限制深度  
    ccp_alpha=0.0018, # 後剪枝：複雜度懲罰  
    class_weight='balanced', # 處理類別不平衡  
    random_state=42  
)  
model.fit(X_train, y_train)
```

Design Limit  
(預剪枝)

Optimization Factor  
(後剪枝)

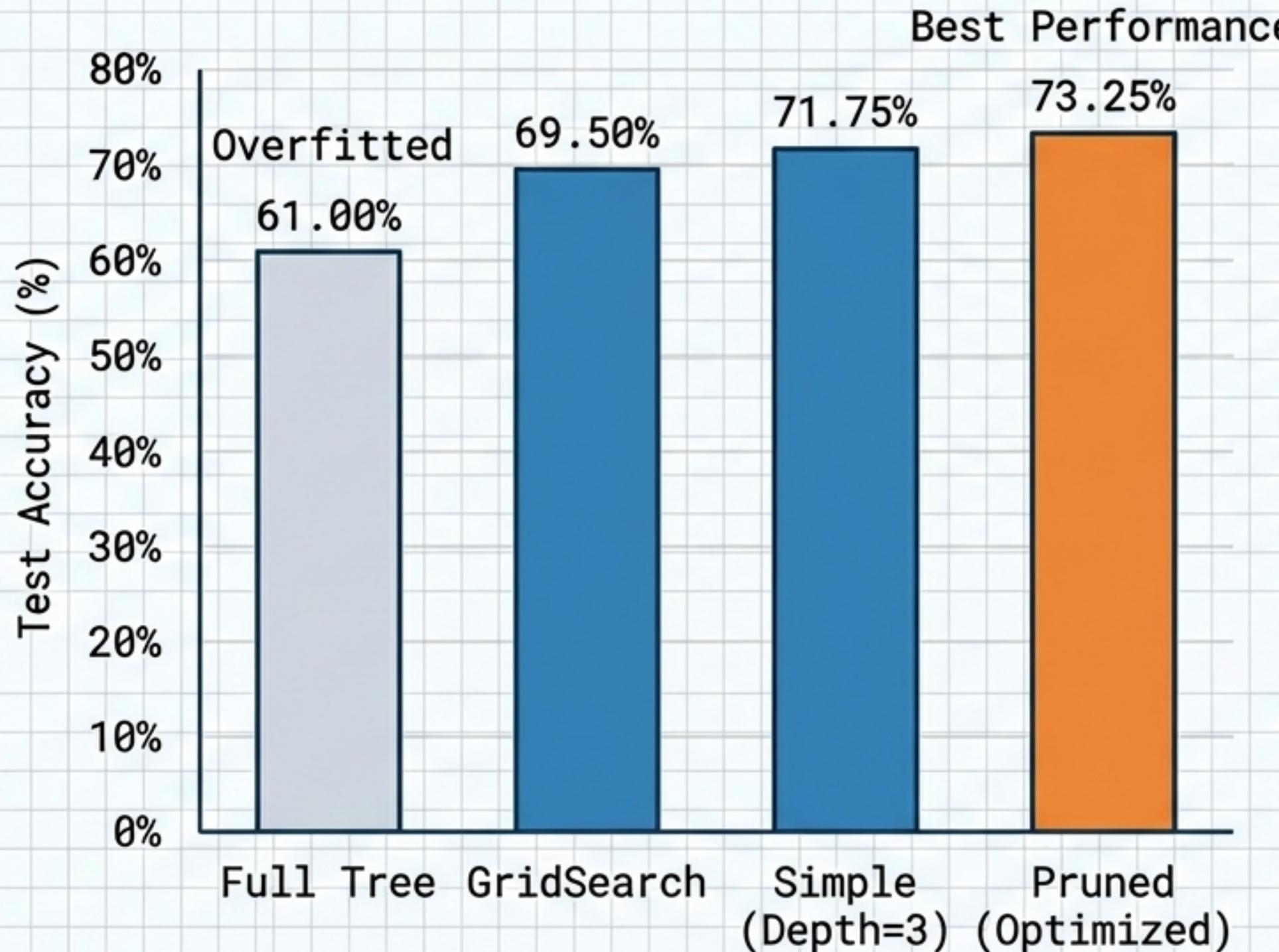
Data Balance  
(數據平衡)

# 實戰案例：化工反應成功率預測



- 樣本總數 (Dataset): 2000 Samples
- 分佈 (Distribution): 56% Success | 44% Failure
- 目標 (Goal): 找出保證反應成功的操作區間 (Operational Window)。

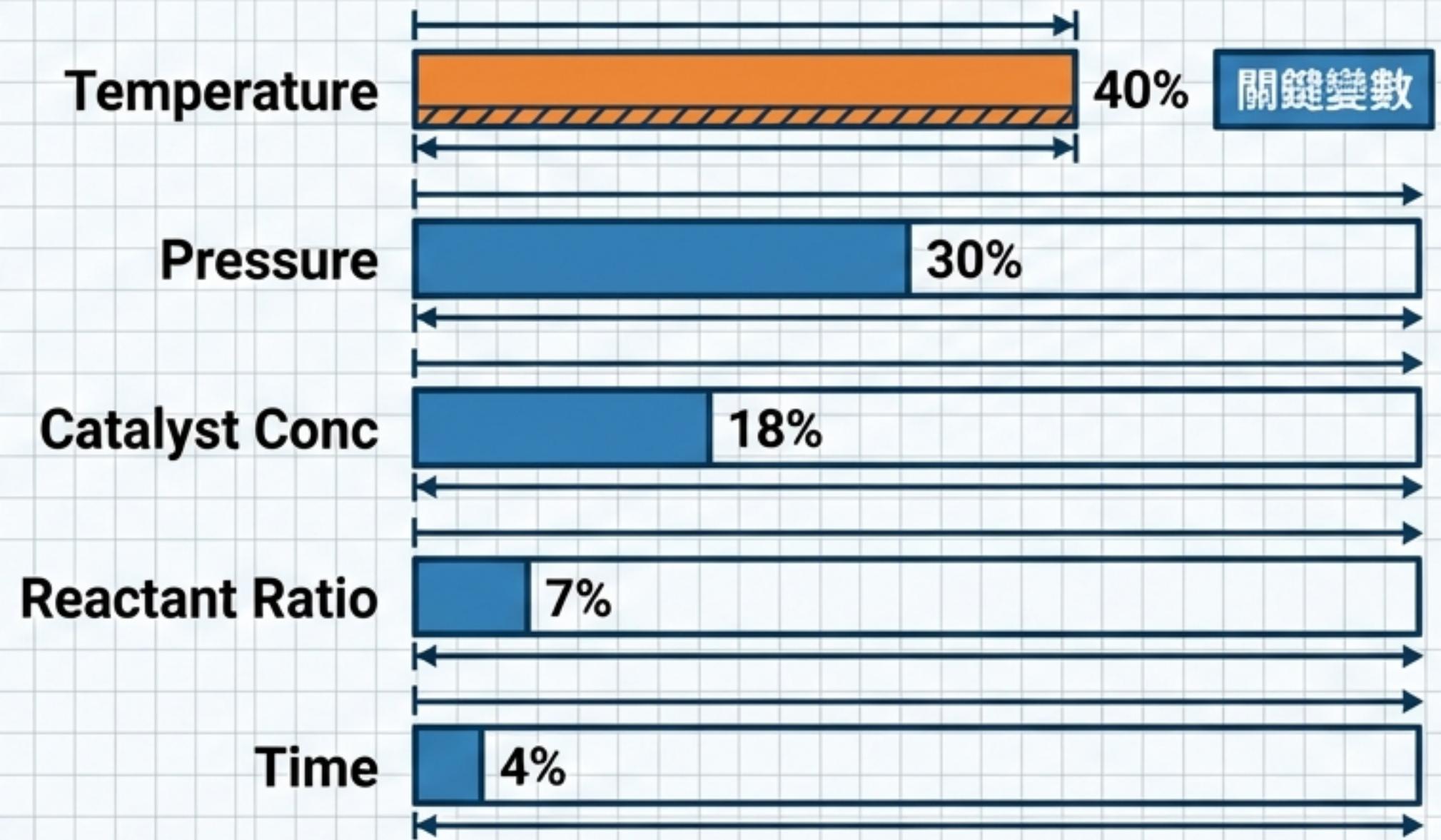
# 模型性能比較：尋找最佳平衡



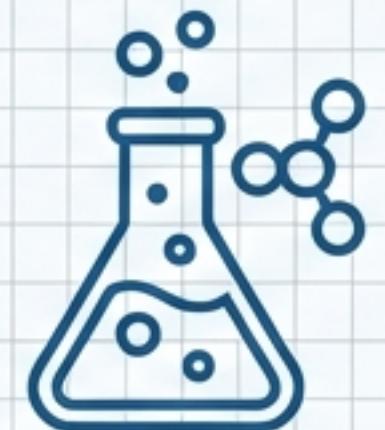
- 奧卡姆剃刀 (Occam's Razor)：簡單模型往往比複雜模型更穩健。
- 後剪枝 (Post-Pruning) 效果優於預剪枝。

# 特徵重要性：什麼控制了反應？

## Feature Importance

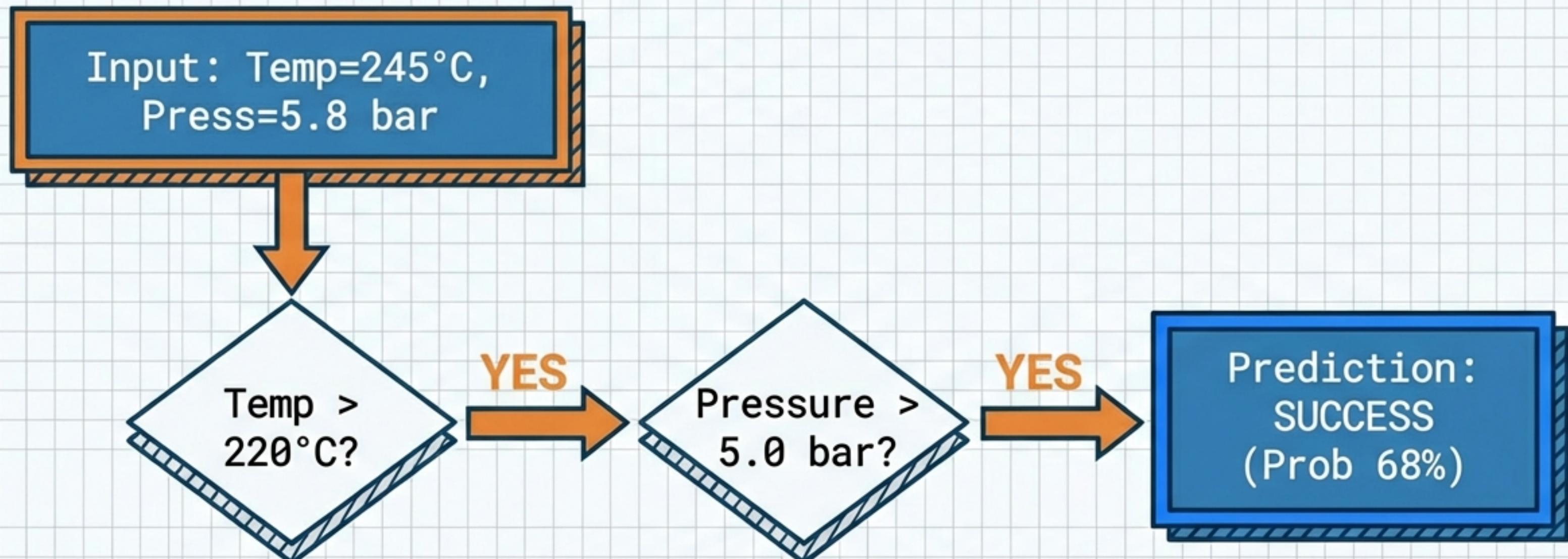


關鍵變數



化工原理驗證：  
結果符合 Arrhenius  
方程與動力學原理 —  
溫度是指數級影響反  
應速率的最關鍵變數，  
其次是壓力。

# 決策路徑追蹤：透視模型思維



就像追蹤管線中的流體流向，我們可以清楚向操作員解釋：為什麼這個批次被預測為 'Success'。

# 決策樹的工程評估：優勢與限制



## 優勢

- 可解釋性強 (Interpretable)：化工領域首選，容易與現場操作員溝通。
- 無需前處理：不需要特徵縮放 (Scaling)，可直接使用物理量數值。
- 非線性處理：自動捕捉  $\text{Temp} \times \text{Pressure}$  的交互作用。



## 限制

- 容易過擬合 (Overfitting)：必須嚴格進行剪枝。
- 不穩定性 (Instability)：數據微小變動可能導致樹結構大變。
- 性能天花板：單一樹準確率通常有上限 (本案約 73-75%)。

# 實務建議：工程師的決策樹指南 (SOP)



## SOP CHECKLIST

### 1. Start Simple (從簡開始)

先試  $\text{max\_depth}=3$  的簡單樹，建立基準線 (Baseline)。

### 2. Use Pruning (善用剪枝)

成本複雜度剪枝 (CCP) 通常優於手動 GridSearch 調參。

### 3. Feature Engineering (特徵工程)

若單一樹效果不佳，考慮加入交互特徵 (如 Temp  $\times$  Pressure)。

### 4. Know Limits (了解極限)

若需準確率  $> 80\%$ ，請升級至 Random Forest (集成學習)。



# 總結與下一步

決策樹是理解數據邏輯的最佳工具，雖有性能天花板，但其‘白箱’(White Box)特性對製程安全與優化至關重要。我們利用剪枝技術將準確率從 61% 提升至 73.25%。

## Next Unit: Unit 13 Random Forest Classifier

從一棵樹到一片森林：利用集成學習突破性能極限。



Unit12\_Decision\_Tree\_Classifier.ipynb

QR Code