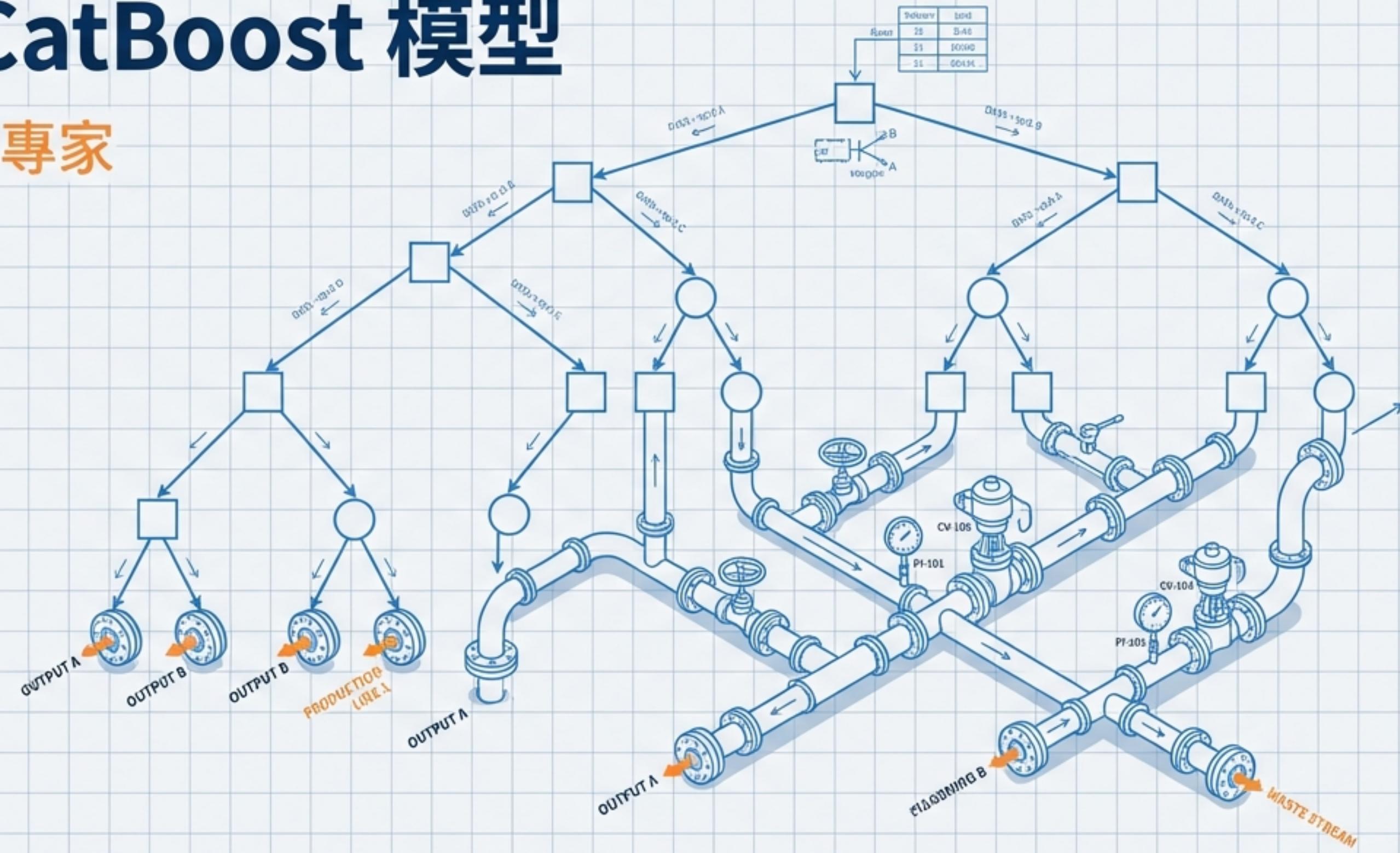
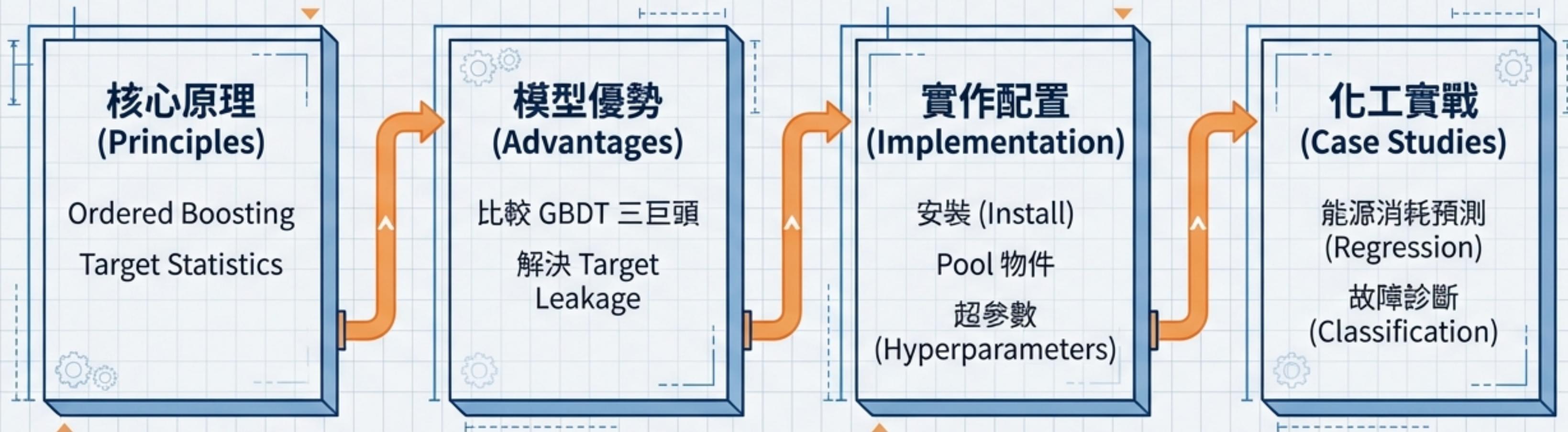


# Unit 13: CatBoost 模型

處理類別特徵的專家



# 單元學習路徑圖 (Learning Objectives)



# 為什麼我們需要 CatBoost？工程數據中的痛點

## 傳統方法 (The Old Way)

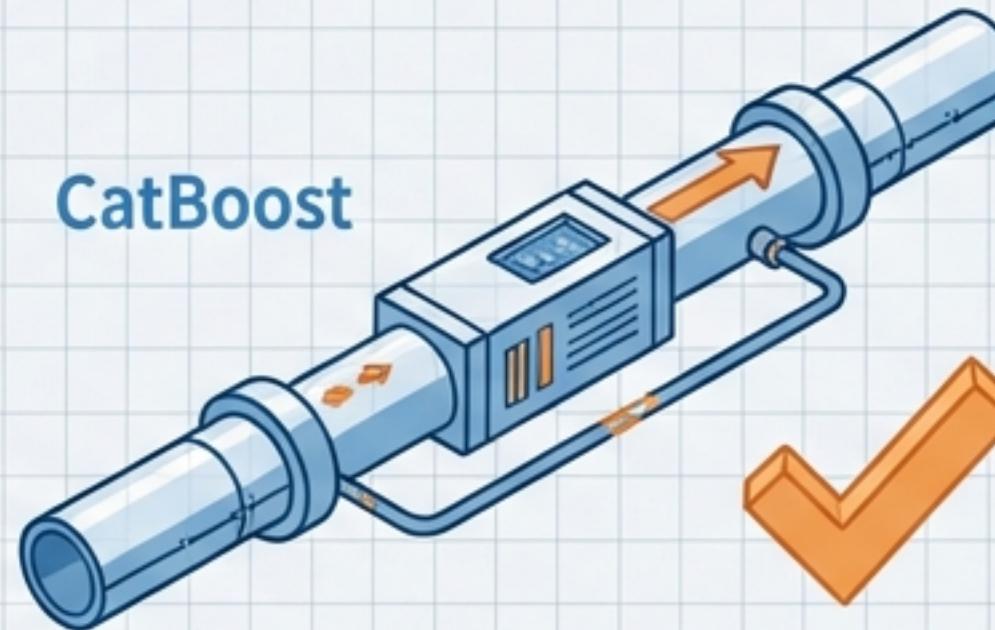
One-Hot  
Encoding



- ⚠ 特徵爆炸 (Feature Explosion): 1000 個類別 → 1000 個稀疏特徵
- ⚠ Target Leakage: 容易導致過擬合
- ⚠ 化工場景: 原料供應商, 反應器編號, 催化劑類型

## CatBoost (The Solution)

CatBoost

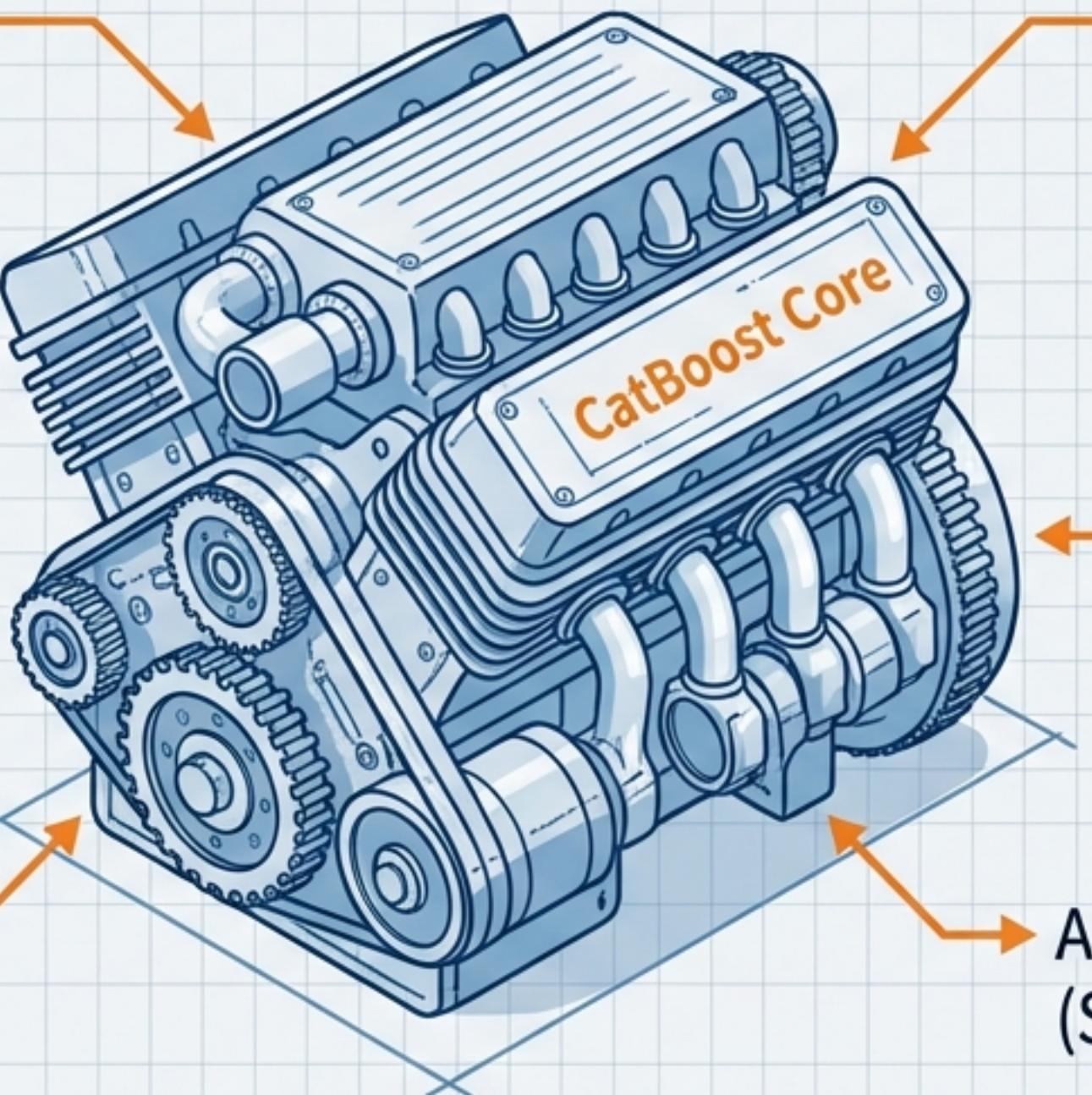


- ✓ 原生支援: 自動最佳化編碼
- ✓ 高準確度: 解決預測偏移

傳統模型在處理高基數 (High Cardinality) 類別特徵時，往往面臨維度災難或信息洩漏的兩難。

# CatBoost (Categorical Boosting) 簡介

Origin: Yandex (2017),  
Open Source



Core Strength:  
**類別特徵處理專家**  
(無需預處理)

Hardware: **支援 GPU 加速**  
(Task\_type='GPU')

Performance: **魯棒性強**，  
預設參數即有優異表現

Architecture: **對稱樹結構**  
(Symmetric Trees) → 快速推理

論文來源：'CatBoost: unbiased boosting with categorical features' (NeurIPS 2018)

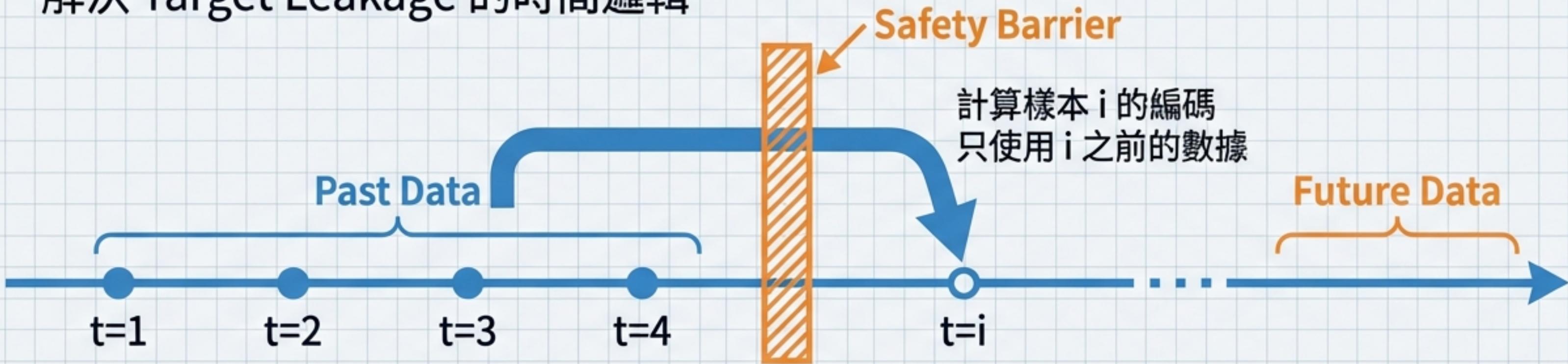
# GBDT 三巨頭規格對決

Feature	XGBoost	LightGBM	CatBoost
準確度	非常高	非常高	<b>最高 (Highest)</b>
類別特徵	需手動編碼	基本支援	<b>原生最佳支援 (Native)</b>
高基數類別	弱	中	<b>強 (High Cardinality)</b>
Target Leakage	風險	風險	<b>已解決 (Solved)</b>
魯棒性	中	低	<b>高 (High)</b>
訓練速度	快	非常快	



# 核心技術 I : Ordered Target Statistics

解決 Target Leakage 的時間邏輯



**Problem:** 傳統 Target Encoding 使用包含自己的數據計算平均值 → 洩漏答案。

**Solution:** 計算樣本  $i$  的編碼時，只使用排在  $i$  之前的數據。

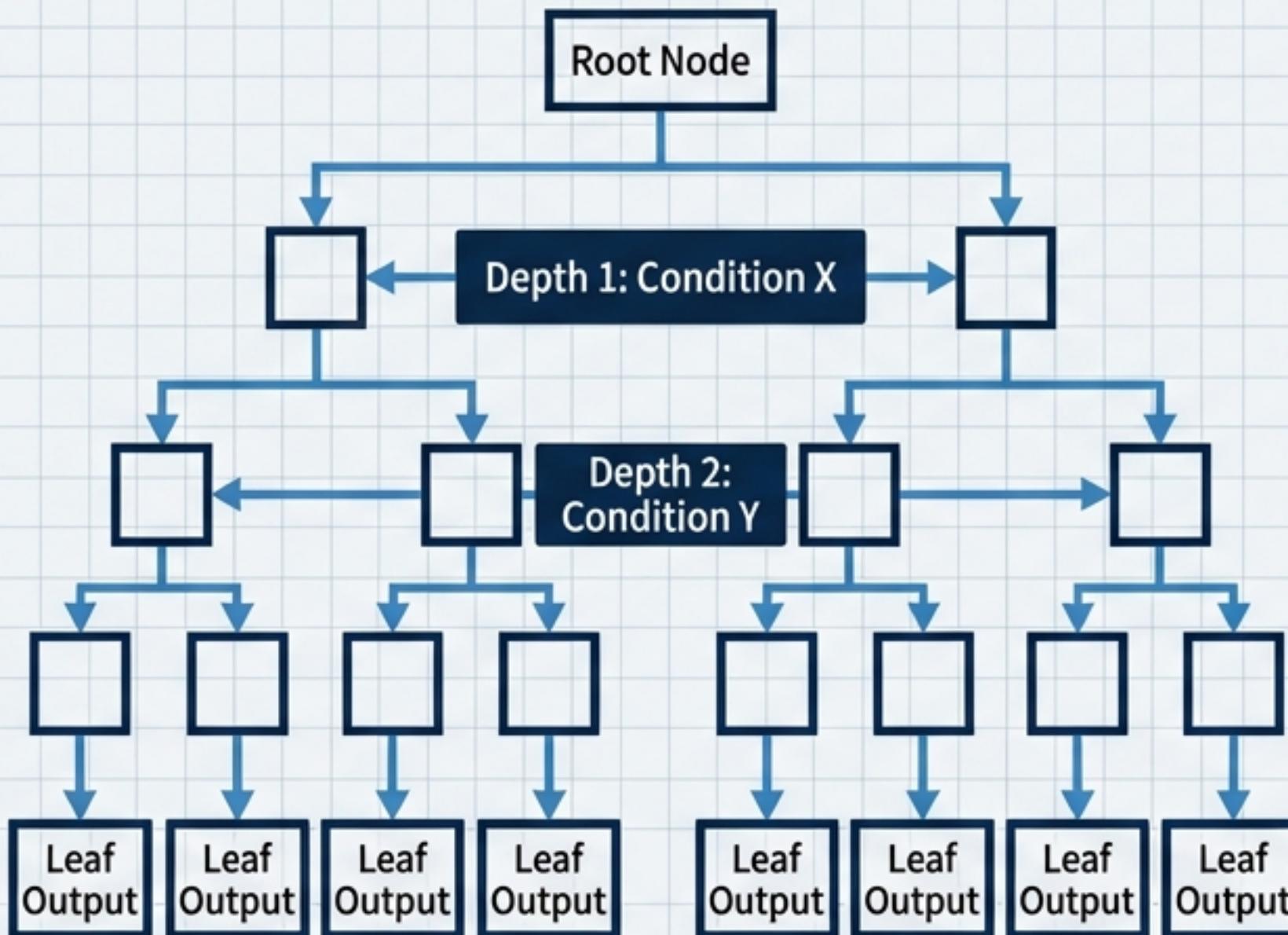
**Formula:**

$$\text{TargetStat}_i(c) = \frac{\sum(y_{-j} \text{ where } j < i) + \alpha * P}{\sum(1 \text{ where } j < i) + \alpha}$$

$y_{-j}$ : 目標值     $P$ : 全局平均     $\alpha$ : 平滑參數

模擬時間序列的處理方式，確保模型不會'偷看'未來的答案。

# 核心技術 II：對稱樹與 Ordered Boosting



對稱樹 (Oblivious Trees)：  
同一層節點使用相同的分裂條件。

## Ordered Boosting

⚠️ **Prediction Shift:** 傳統 Boosting 在預測訓練集時過於“樂觀”。

➡️ **Solution:** 訓練第  $t$  棵樹時，只利用之前的樣本模型計算殘差 (Residuals)。

$$\text{Residuals}_i^{(t)} = y_i - \text{Model}_{t-1}(\text{samples}_j : j < i)$$

## Engineering Benefit:

結構限制降低過擬合，極適合 GPU 平行運算，  
並提升模型在新生產數據上的穩定性 (Stability)。

# 安裝與數據封裝：Pool 類別

```
pip install catboost # 終端機指令
from catboost import CatBoostRegressor, Pool

# 定義類別特徵索引（無需 One-Hot）
cat_features_indices = [0, 1, 5]

# 使用 Pool 類別封裝數據（高效記憶體管理）
train_pool = Pool(data=X_train, label=y_train,
                   cat_features=cat_features_indices)
test_pool = Pool(data=X_test, label=y_test,
                  cat_features=cat_features_indices)
```

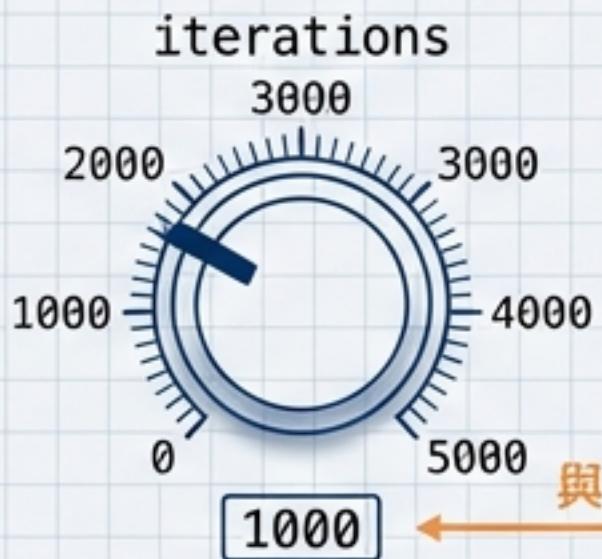
Best Practice 

## Why use Pool?

- 高效記憶體使用 (Memory Efficient)
- 自動追蹤類別特徵
- 自動處理缺失值 (NaN)

# 關鍵超參數控制面板 (Control Panel)

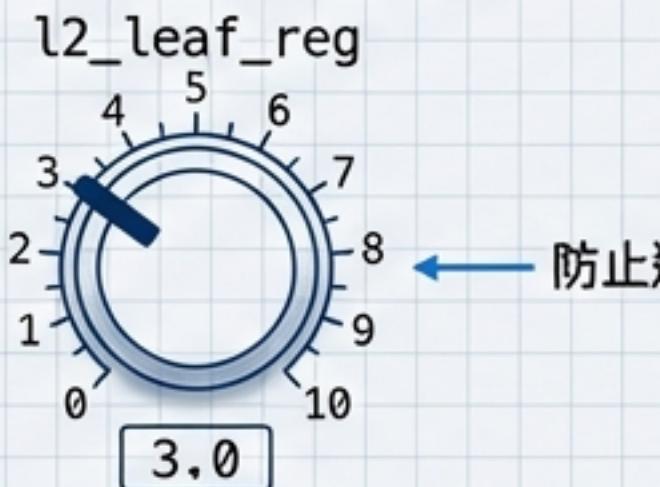
## Priority High (主控)



樹的數量 (Default: 1000)

學習率 (Default: Auto)

## Regulation (穩健性)



L2 正則化 (Default: 3.0)

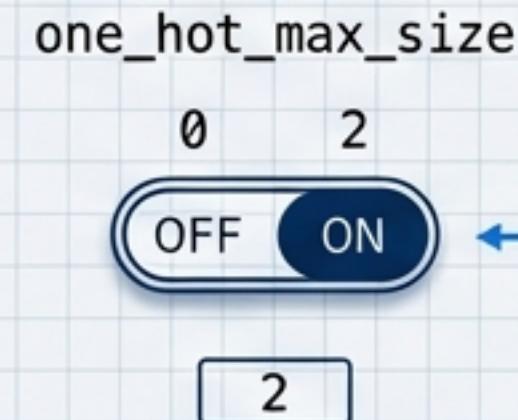
## Structure (結構)



樹深度 (Default: 6)

← 對稱樹通常不需要太深 (4-10)

## Categorical (類別)



← 低基數類別使用 One-Hot

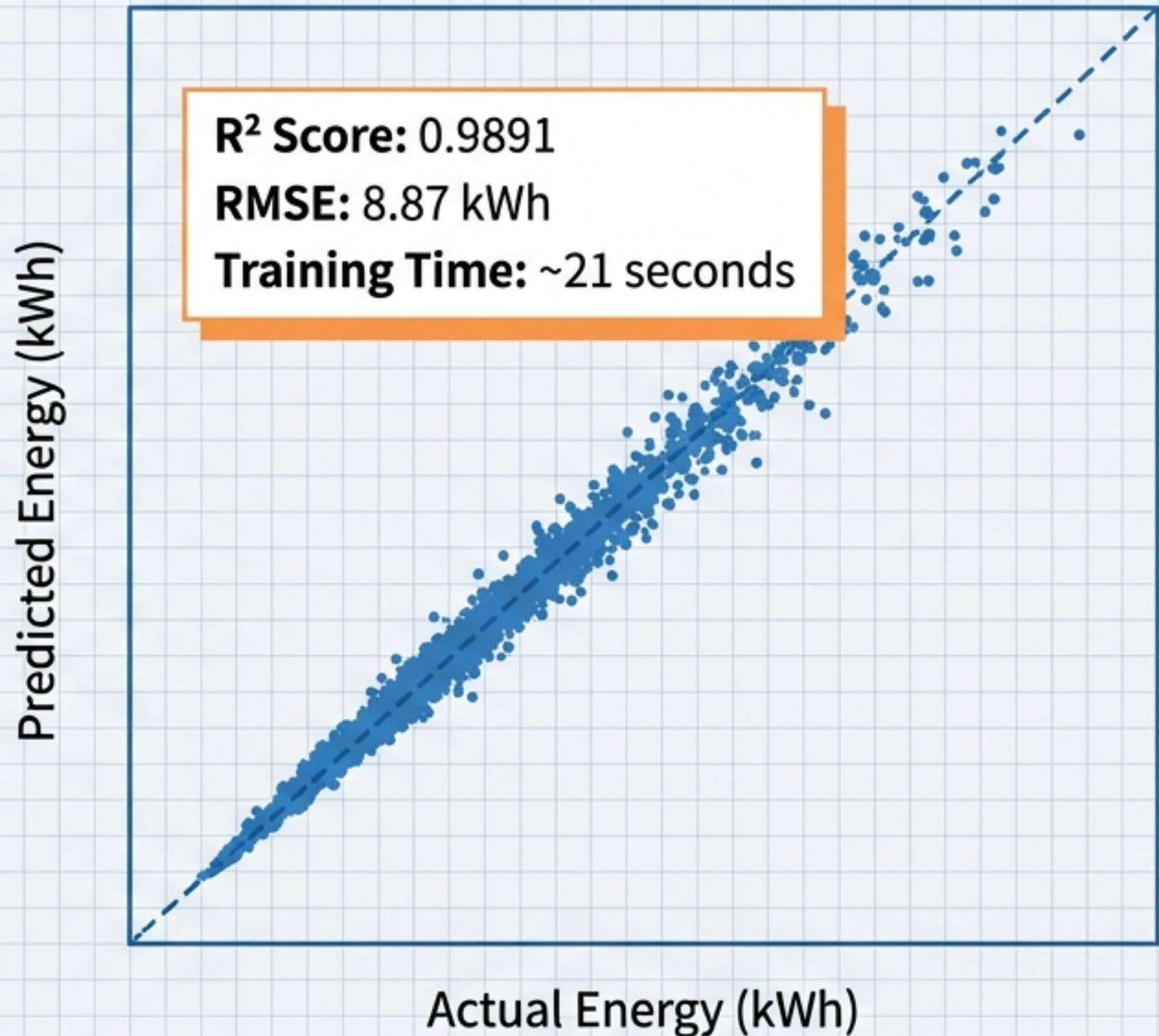
閾值 (Default: 2)

Pro Tip: 調參首選：iterations 與 learning\_rate。

# 實戰案例 I：能源消耗預測 (Regression)

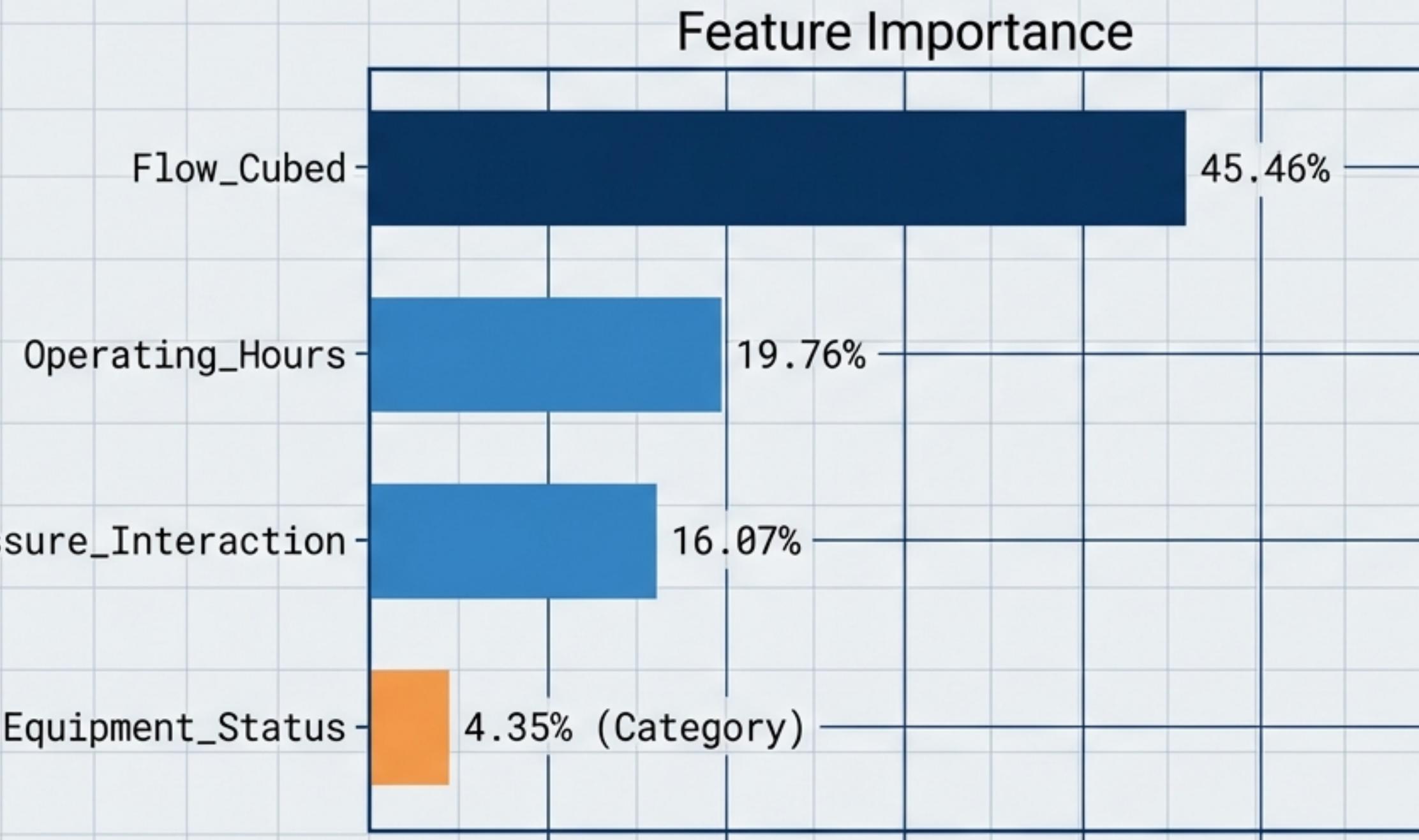
**Context (專案背景)**

- **Goal:** Predict Energy Consumption (kWh)
- **Data:** 100,000 samples
- **Numerical Inputs:** Flow, Temp, Pressure
- **Categorical Inputs:** Equipment Status, Operation Mode



CatBoost 自動處理 Equipment\_Status 與 Operation\_Mode，無需手動編碼。

# 案例 I 分析：特徵重要性與物理意義

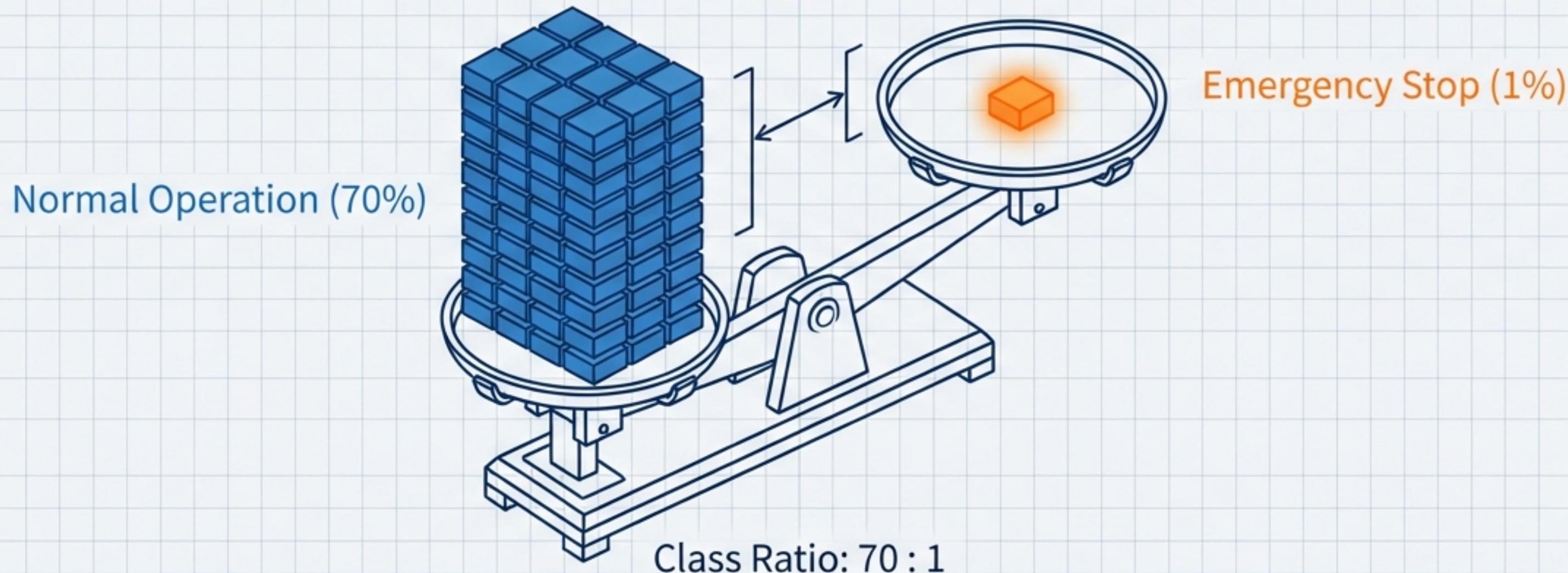


- **Physics Verified:** 流體力學中，功率/壓降與流量的三次方成正比 ( $\text{Power} \propto Q^3$ )。

- **Categorical Impact:** 類別特徵 (Status & Mode) 合計貢獻約 7%，對精準度至關重要。

# 實戰案例 II：設備故障診斷 (Classification)

挑戰極度不平衡資料 (Extreme Imbalance)



## The Solution

```
auto_class_weights='Balanced'
```

CatBoost 自動賦予稀有類別較高的權重，無需人工過採樣 (Oversampling)。

# 案例 II 分析：安全與準確率的權衡

Heatmap / Confusion Matrix

Row	Normal	Leak Warning	Temp Abnormal	Emergency Stop
Normal				
Leak Warning		Recall 96%		
Temp Abnormal			Recall 98%	
Emergency Stop				Conservative Error

## Performance Metrics

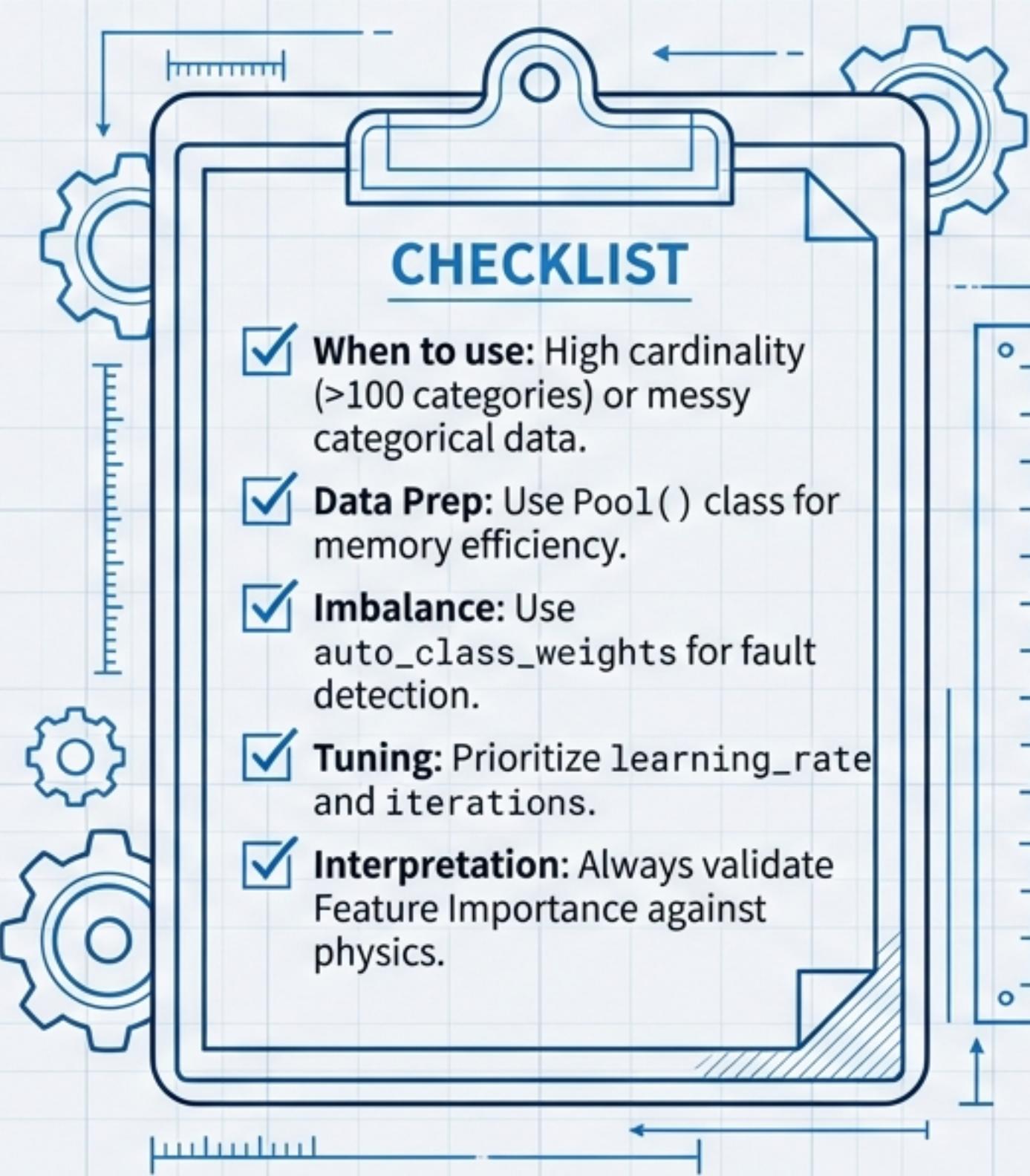
Overall Accuracy: 81.95%  
Macro F1: 74.55%

## Safety Insight (安全洞察)

- Conservative Strategy (保守策略)：  
模型傾向於發出假警報(False Alarm)，  
也不願漏掉任何一次危險故障。
- 高風險故障（洩漏/溫控）具有極高的  
召回率。

Pro Tip: 根據業務需求調整決策閾值 (Decision Threshold) 以優化安全策略。

# 總結與最佳實踐 (Engineer's Checklist)



Next Step: Unit 14 - Deep Learning Applications