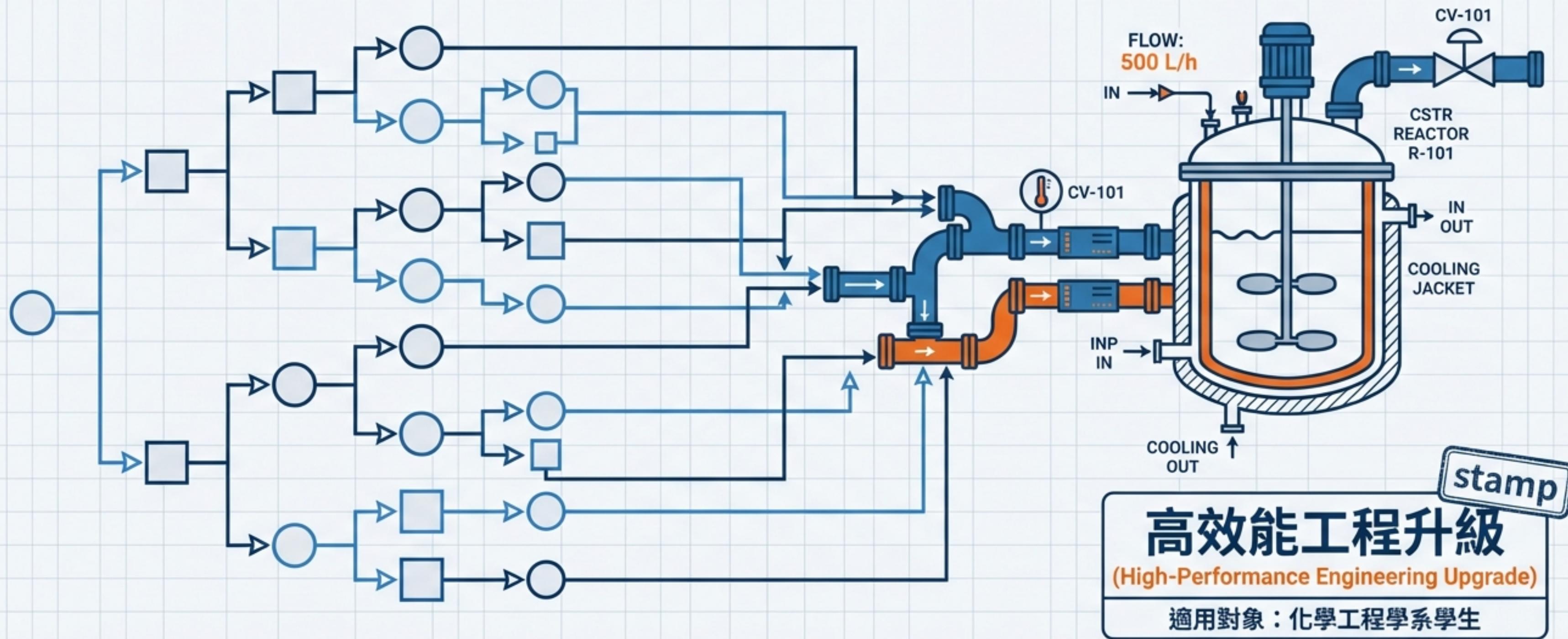


Unit 13: XGBoost 模型

Extreme Gradient Boosting | 課程代碼：CHE-AI-114

Instructor Line | AI 在化工上之應用 | 授課教師：莊曜楨 助理教授



您的數位工具箱升級 (YOUR TOOLKIT UPGRADE)



核心原則 (Core Principles)

-  **極致優化的 GBDT**: 在傳統梯度提升樹基礎上，加入二階導數資訊與正則化。
-  **工業標準**: Kaggle 競賽冠軍與工業界 (如製程優化、故障預測) 的首選工具。
-  **化工應用**: 專為結構化表格資料設計，精準捕捉非線性關係 (如反應動力學)。

本單元學習目標 (Learning Objectives)

- 理解二階泰勒展開與正則化機制。
- 掌握關鍵超參數 (`max_depth`, `learning_rate`)。
- 實作化工設備故障診斷與反應器產率預測。

系統架構對比：XGBoost vs 傳統 GBDT

特性 (Feature)	sklearn GBDT (Legacy)	XGBoost (Turbo)
訓練速度	慢 (序列訓練)	快 (並行化/Cache優化)
準確度	高 (一階導數)	非常高 (二階導數 + 正則化)
記憶體使用	高	低 (優化 Column Block)
缺失值處理	需人工預處理	內建自動學習方向
硬體加速	無 CPU only	完整 GPU 支援



結論：XGBoost 提供 10-60 倍的訓練速度提升，
是處理大數據 (>10k 筆) 的唯一選擇。

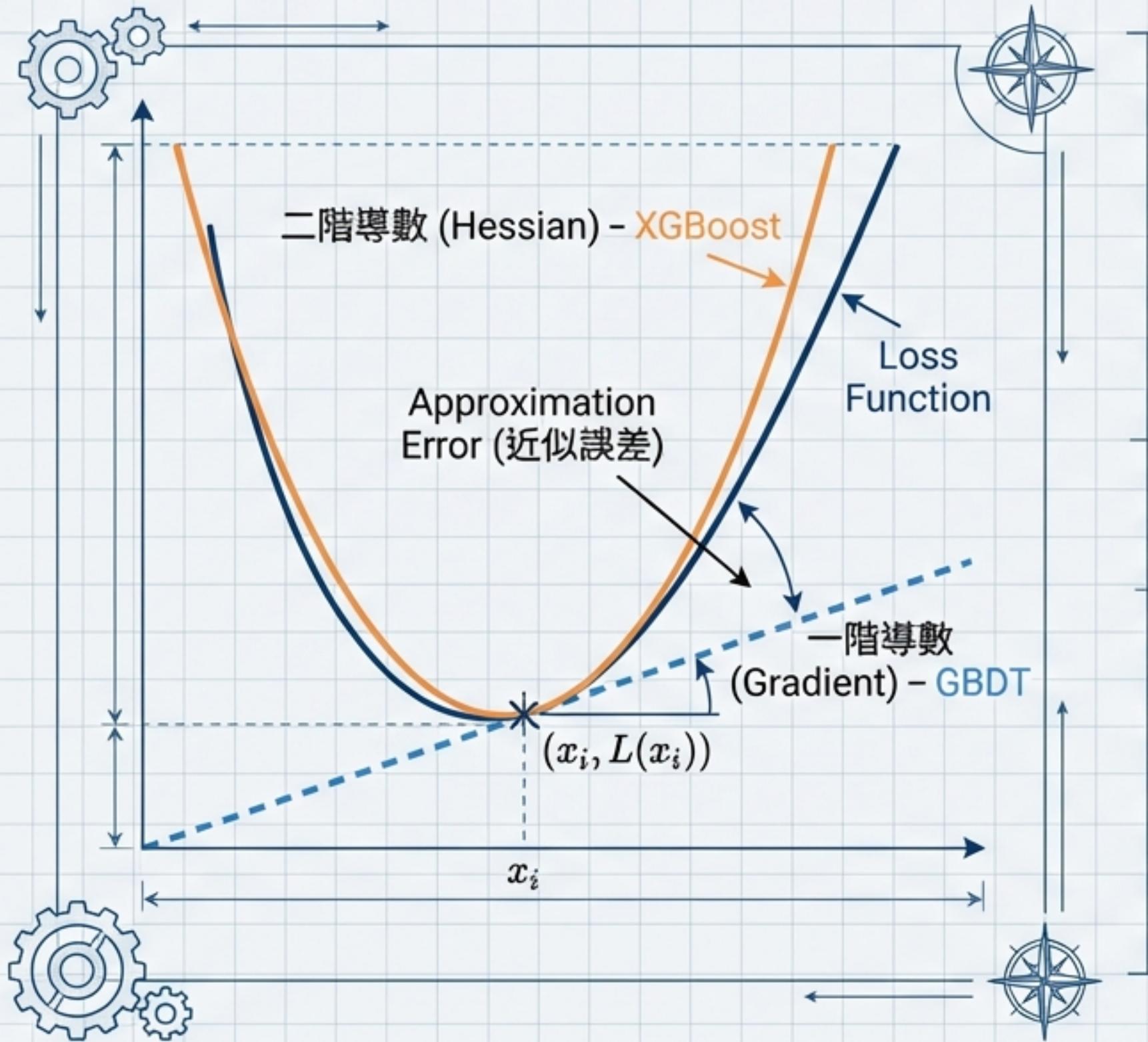
ENGINEERING BLUEPRINT V1

- SYSTEM ARCHITECTURE CONTRAST

PROJECT: XGBOOST IMPLEMENTATION

REV 1.0

核心機制 (1)：二階泰勒展開 (The Mathematical Engine)



為何更準確？

- 傳統 GBDT：僅使用一階導數（梯度 g_i ），只知道下降的方向。
- XGBoost：加入二階導數（Hessian h_i ），同時知道方向與「曲率」，收斂更快。

$$L \approx l(y, \hat{y}) + g_i f_t(x) + \frac{1}{2} h_i f_t^2(x)$$

g_i : Gradient (斜率)

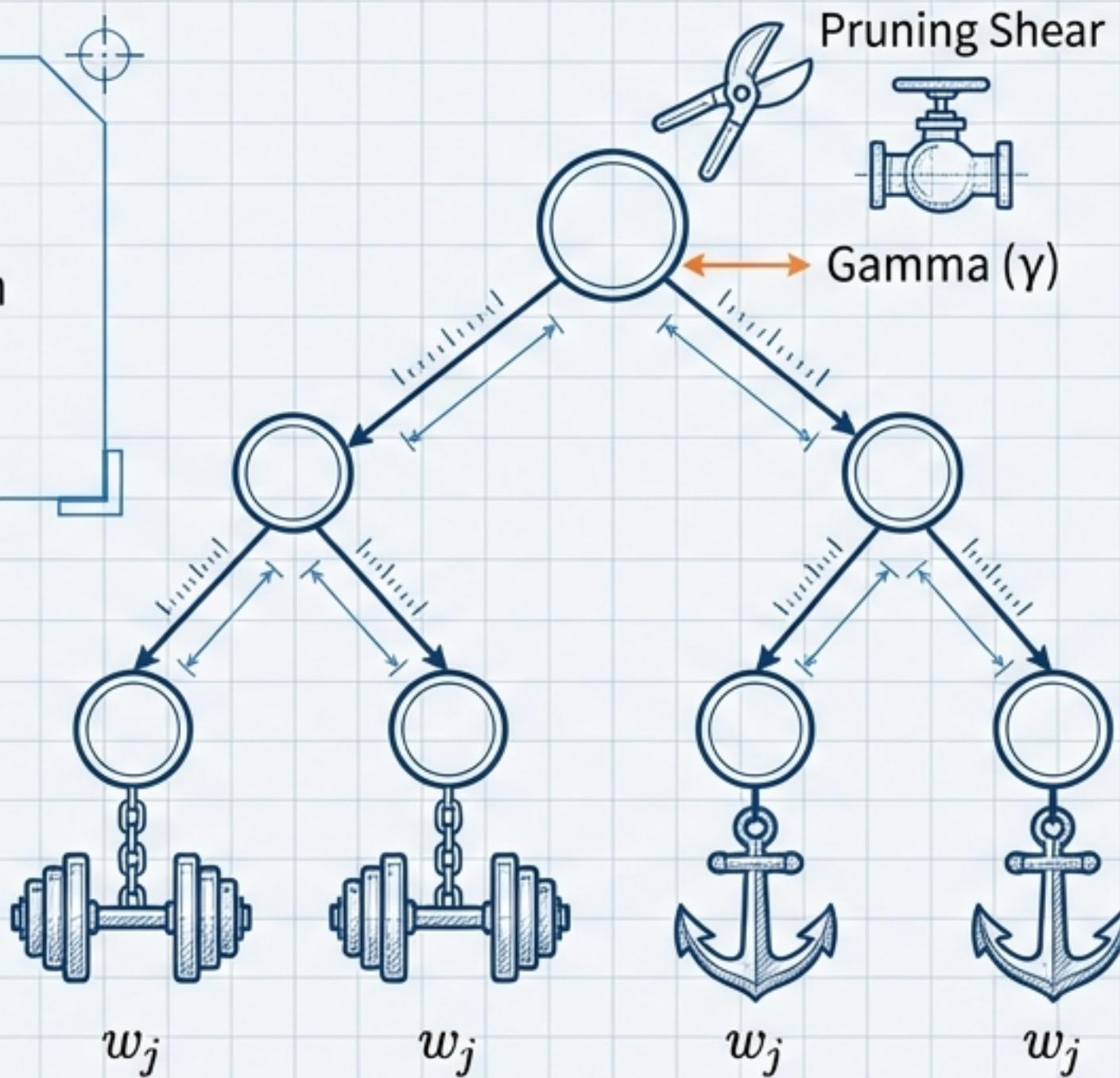
h_i : Hessian (曲率)

核心機制 (2)：正則化與樹結構 (Safety Systems)

目標函數 (Objective Function):

$$Obj = \text{Loss} + \text{Regularization}$$

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_j w^2$$



化工類比：正則化就像反應器的安全閥，防止模型為了擬合雜訊而「過熱」。

防止系統過熱 (Overfitting Prevention)

- 1. **Gamma (γ)**: 葉節點懲罰係數。類似「最小分裂損失」，如果 Gain < γ ，則修剪該分支 (Pruning)。
- 2. **Lambda (λ)**: L2 正則化係數。控制葉節點權重 w ，防止預測值過於極端 (Smoothing)。

控制面板：關鍵超參數 (The Control Panel)

樹結構 (Tree Structure)

`max_depth`

建議 3-10。控制樹深，過深易過擬合。

`min_child_weight`

建議 1-10。葉子節點最小權重和，越大越保守。

`gamma`

建議 0-5。分裂所需最小損失減少量。

提升策略 (Boosting)

`learning_rate`
(eta)

建議 0.01-0.3。學習率，越小越穩但需更多樹。

`n_estimators`
(=50-1000)

建議 50-1000。樹的總數量。

隨機採樣 (Sampling)

`subsample`

On

樣本採樣比例
(Row Subsampling)。

`colsample_bytree`

On

特徵採樣比例
(Column Subsampling)。

1	•	•
2	•	•
3	•	•
4	•	•

1	•	•
2	•	•
3	•	•
4	•	•

安裝與實作 (System Installation)

安裝指令：pip install xgboost

```
import xgboost as xgb
from xgboost import XGBRegressor

# 1. 初始化模型（回歸任務）
model = XGBRegressor(
    n_estimators=1000,          # 樹的數量
    learning_rate=0.05,          # 學習率
    max_depth=6,                # 樹深
    tree_method='gpu_hist',      # GPU 加速（關鍵！） ←
    subsample=0.8
)

# 2. 訓練與預測（Scikit-learn 風格介面）
model.fit(
    X_train, y_train,
    eval_set=[(X_val, y_val)],
    early_stopping_rounds=50,    # 早停機制：若50輪沒進步則停止 ←
    verbose=False
)
```

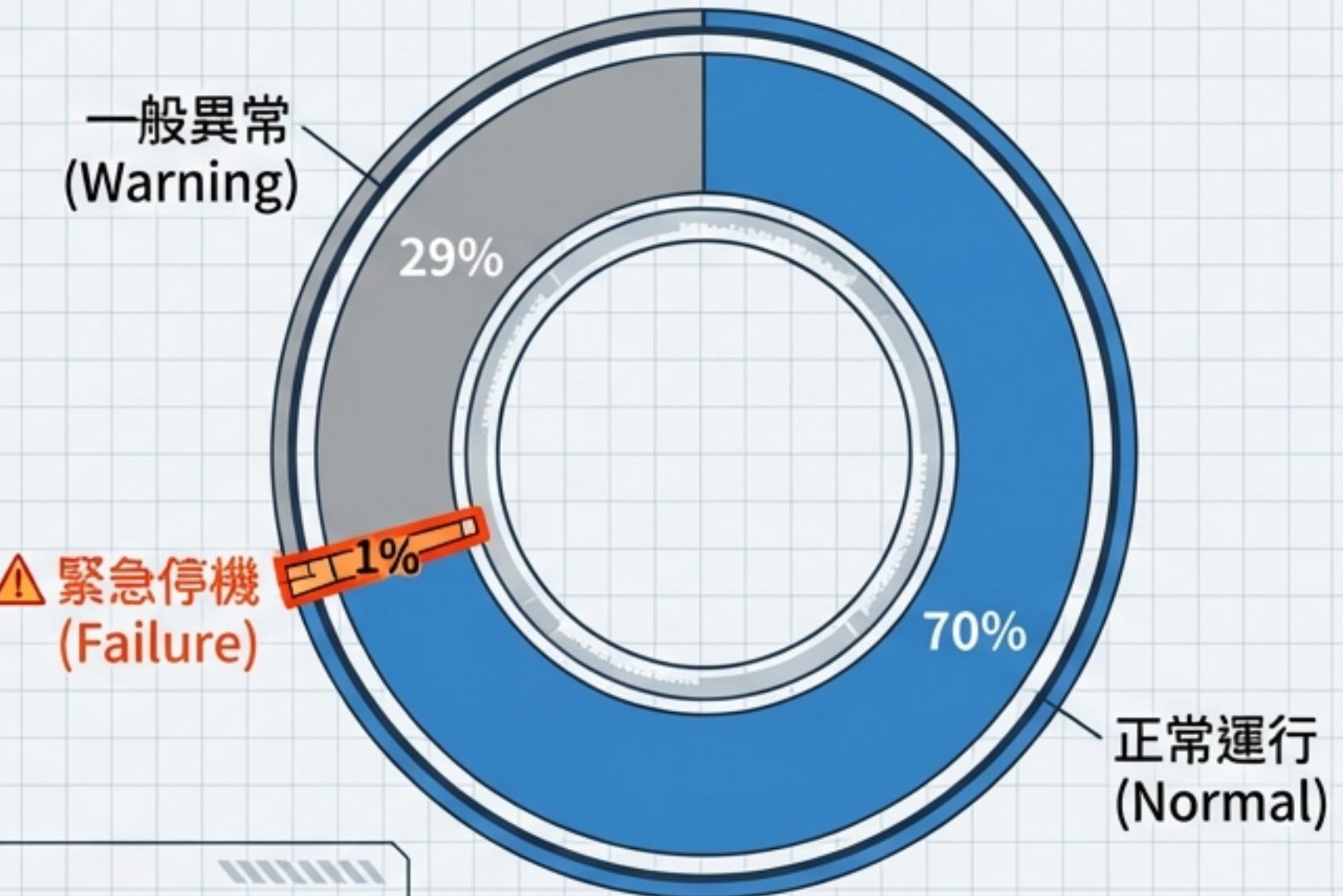
開啟硬體加速

避免過度訓練

實戰案例一：化工設備故障診斷 (Classification)

案例背景 (Context)

- **目標:** 預測設備未來 7 天內是否發生故障。
- **數據:** 150,000 筆時間序列數據，30 個特徵 (傳感器 + 衍生特徵)。
- **核心挑戰:** 極度不平衡 (Imbalanced Data)。

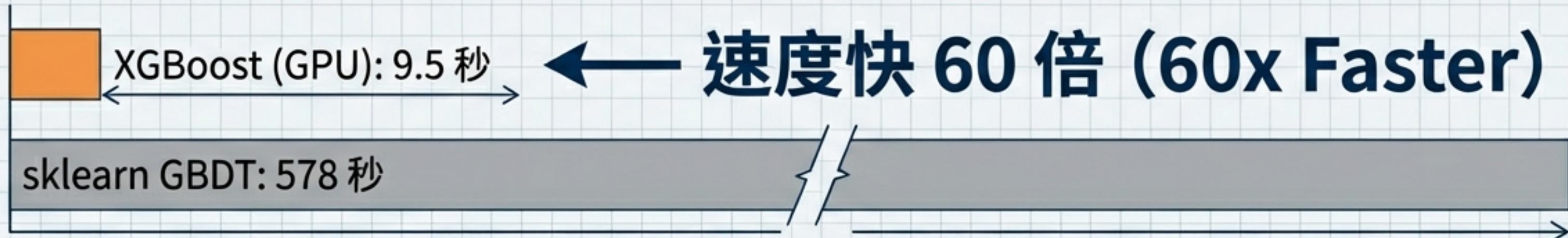


解決策略

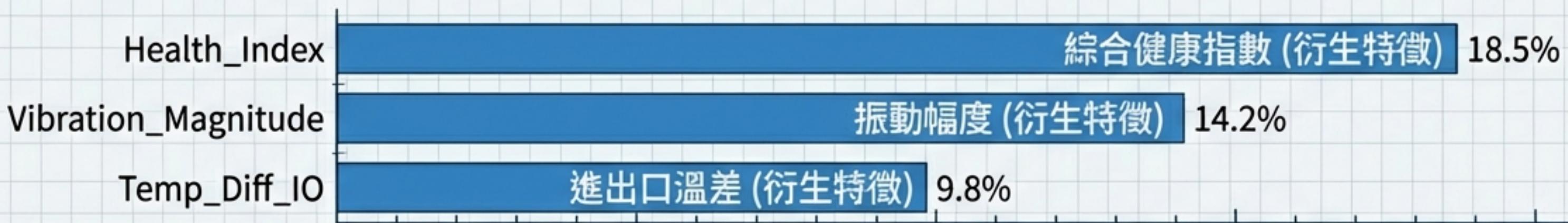
- 使用 `sample_weight` 平衡類別影響。
- 評估指標：F1-Score (Weighted) 而非單純 Accuracy。

診斷結果分析：速度與關鍵指標

效能競賽 (Performance Matrix)



特徵重要性 (Feature Importance)



關鍵洞察：原始傳感器數據 (Sensor Data) 不如經工程處理的衍生特徵 (Health Index) 重要。特徵工程是成功的關鍵。

實戰案例二：反應器產率預測 (Regression)

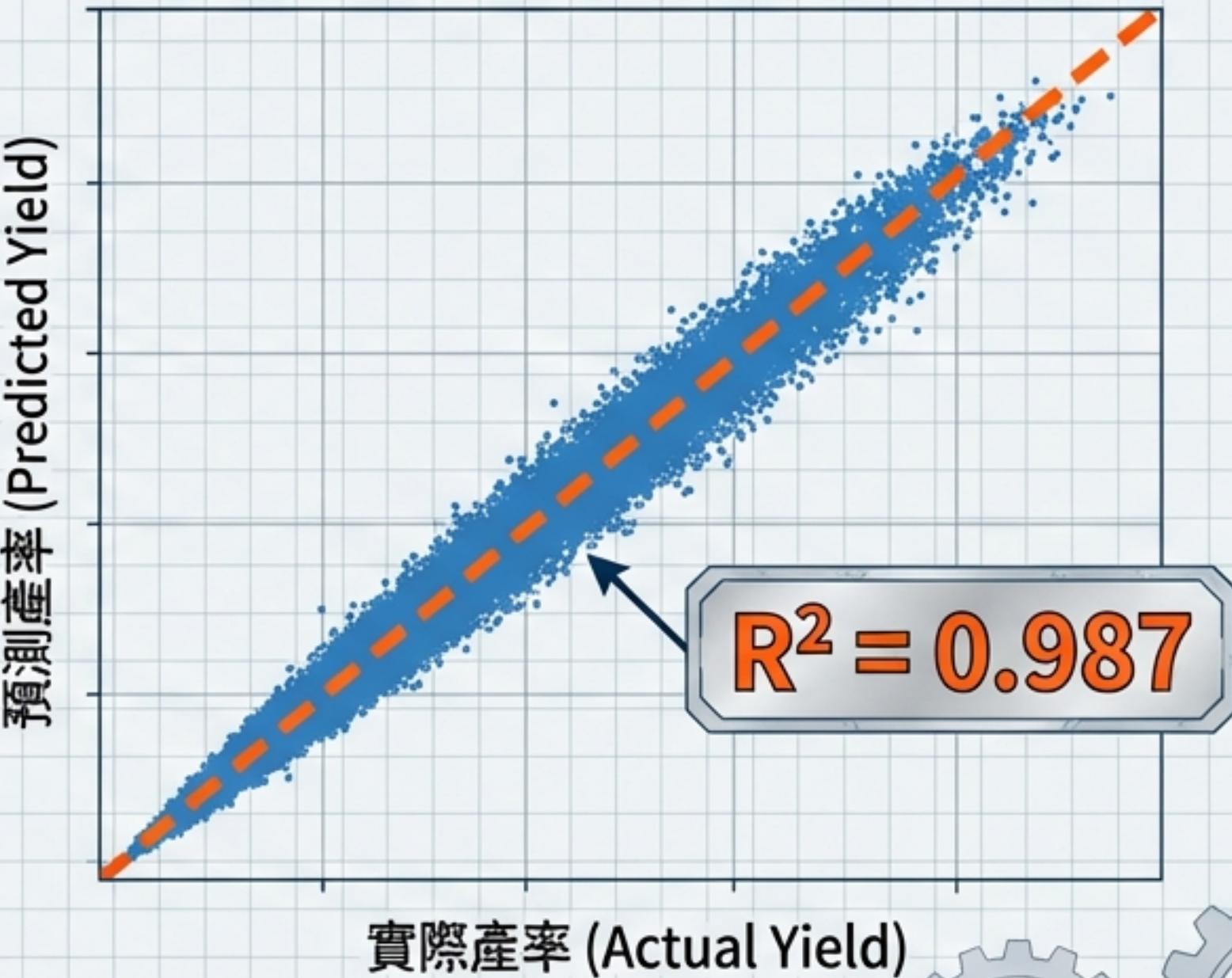
目標：優化生產產率 (Yield %)，
數據量 100,000 筆。

物理挑戰：流量 (Flow) 與產率存在
「三次方 (Cubic)」非線性關係。

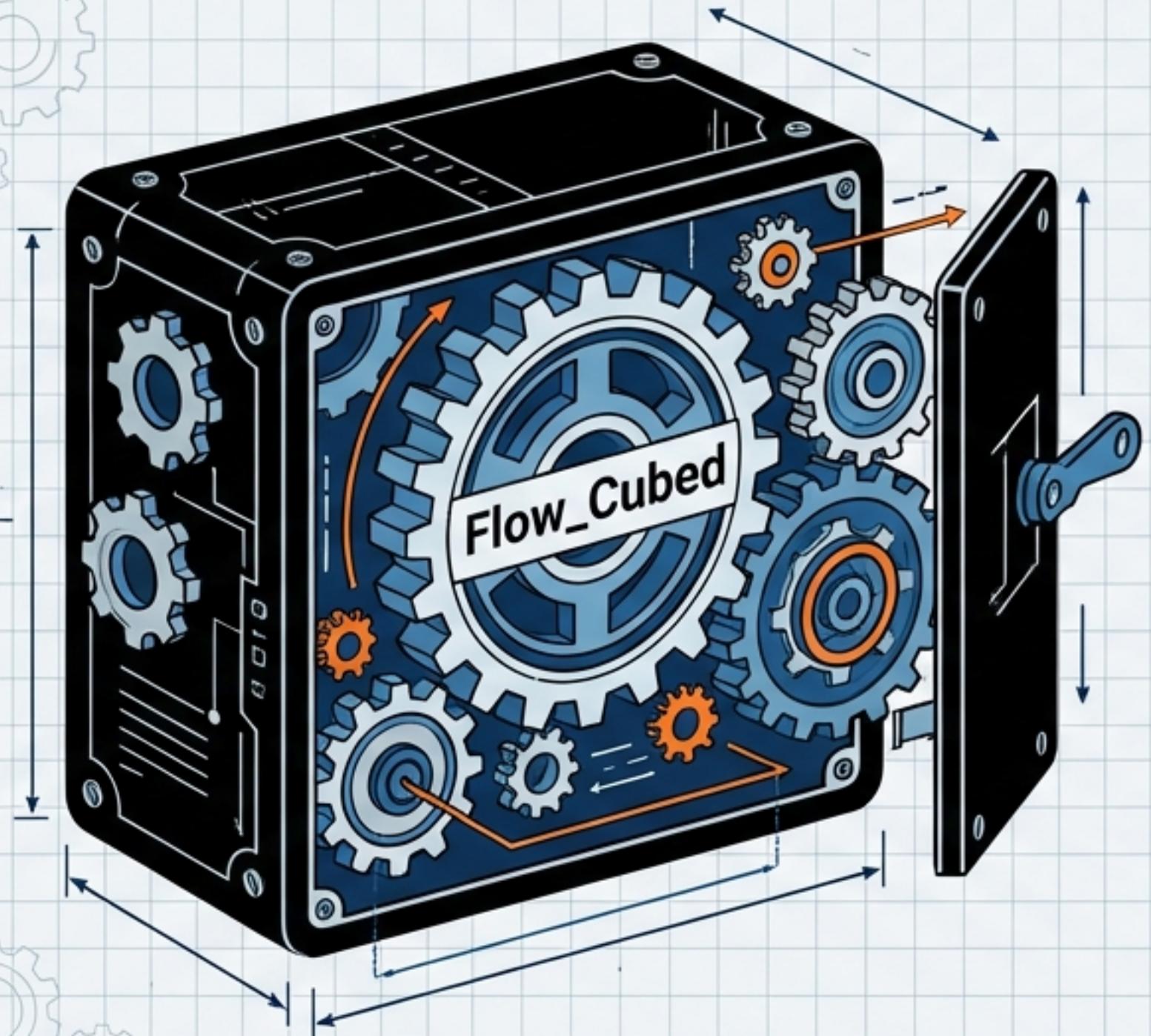
模型表現：

- Linear Regression: RMSE 11.44
- Random Forest: RMSE 16.46 (無法捕捉複雜關係)
- **XGBoost: RMSE 9.70 ($R^2=0.987$)**

Parity Plot (Predicted vs. Actual)



模型可解釋性：打開黑盒子 (White Box Engineering)



為何 XGBoost 表現最好？

- **Flow_Cubed 重要性: 60.19%**
 - XGBoost 成功「學會」了化工原理：
產率與流量的三次方成正比。
- **Operating_Hours 重要性: 11.58%**
 - 反映了設備老化對產率的影響。

相較於 Linear Regression 無法處理非線性，
以及 Random Forest 在高維度下的稀疏，
XGBoost 精準捕捉了物理定律。

進階工具：可使用 SHAP Values 進一步解釋單一樣本預測。

效能優化與故障排除 (Optimization & Troubleshooting)



Best Practice: 始終使用 `early_stopping_rounds=50` 搭配驗證集。



決策矩陣：模型選擇指南

XGBoost

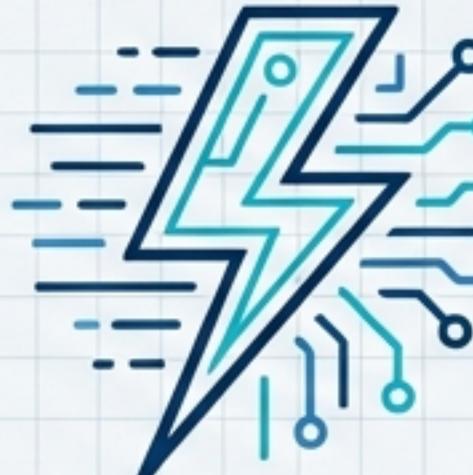


通用首選 (The Standard)。穩定、社群支援最強、精度極高。適合中大型表格數據。



工業界標配

LightGBM



極致速度。適合超大規模數據 (>10M rows)，訓練更快但較易過擬合。

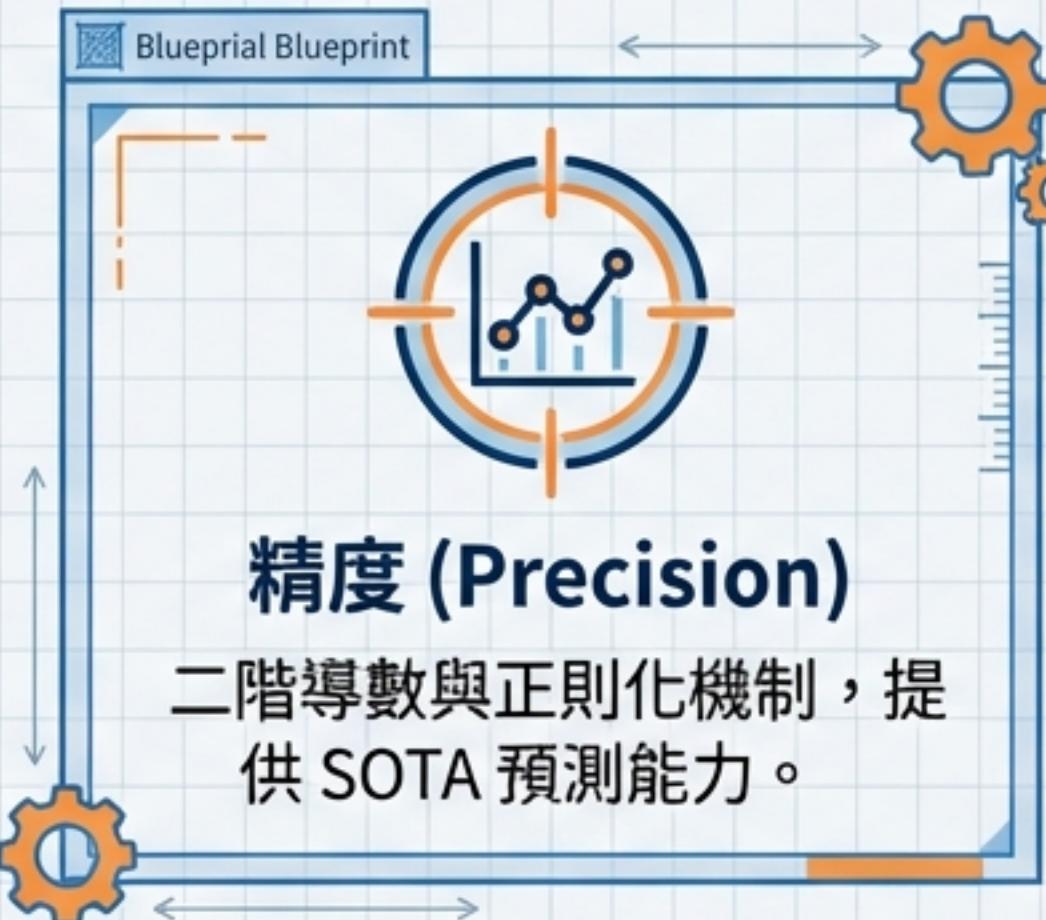
CatBoost



類別專家。對 Categorical Features 支援最好，無需預先編碼 (One-hot)。

XGBoost 是目前化工領域最平衡、最可靠的選擇。

總結：準備部署 (Ready for Deployment)



下一步 (Next Step):

- 開啟 Unit13_XGBoost_Regression.ipynb (產率預測)
- 開啟 Unit13_XGBoost_Classification.ipynb (故障診斷)

→ ↗ 現在就開始您的實作練習!