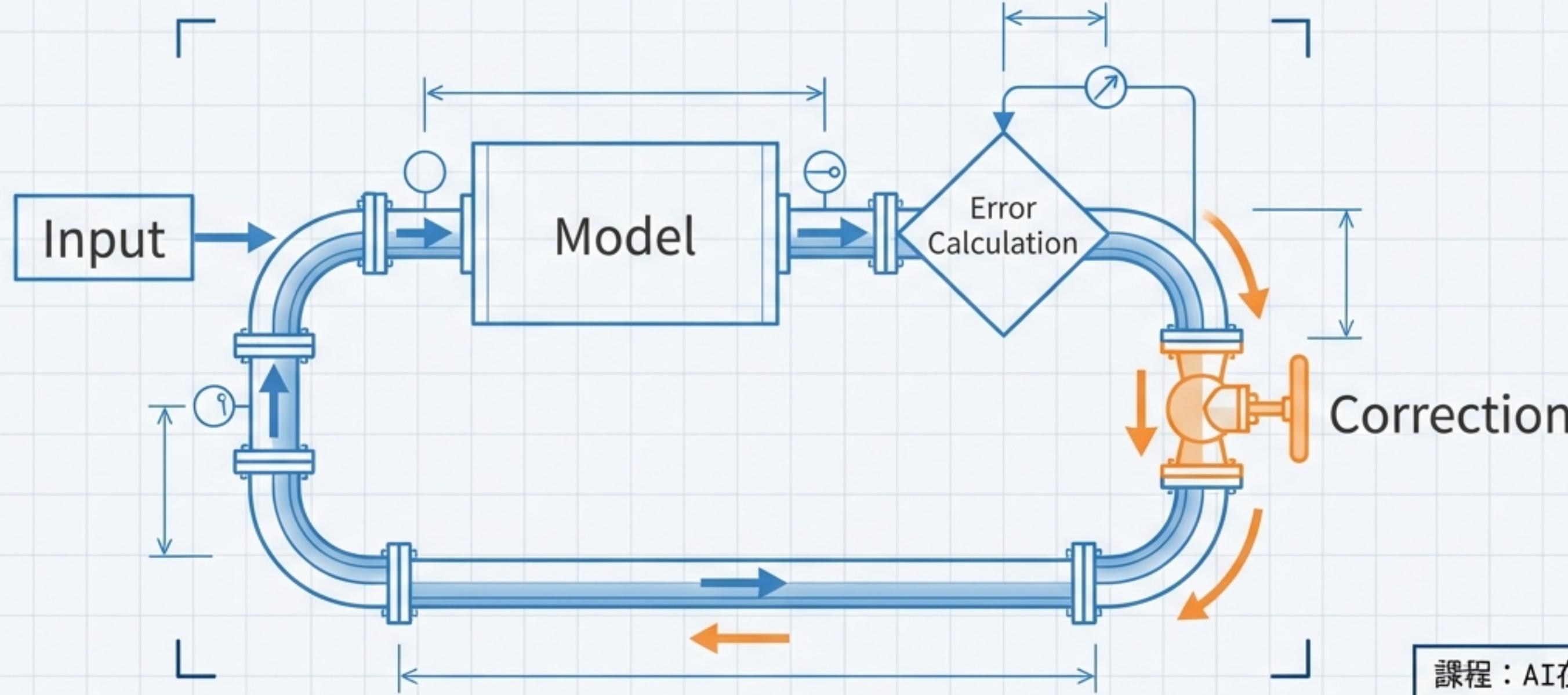


Unit 12 梯度提升分類器：化工AI的高精度修正引擎

從並行投票到序列優化：打造具備自我修正能力的預測系統



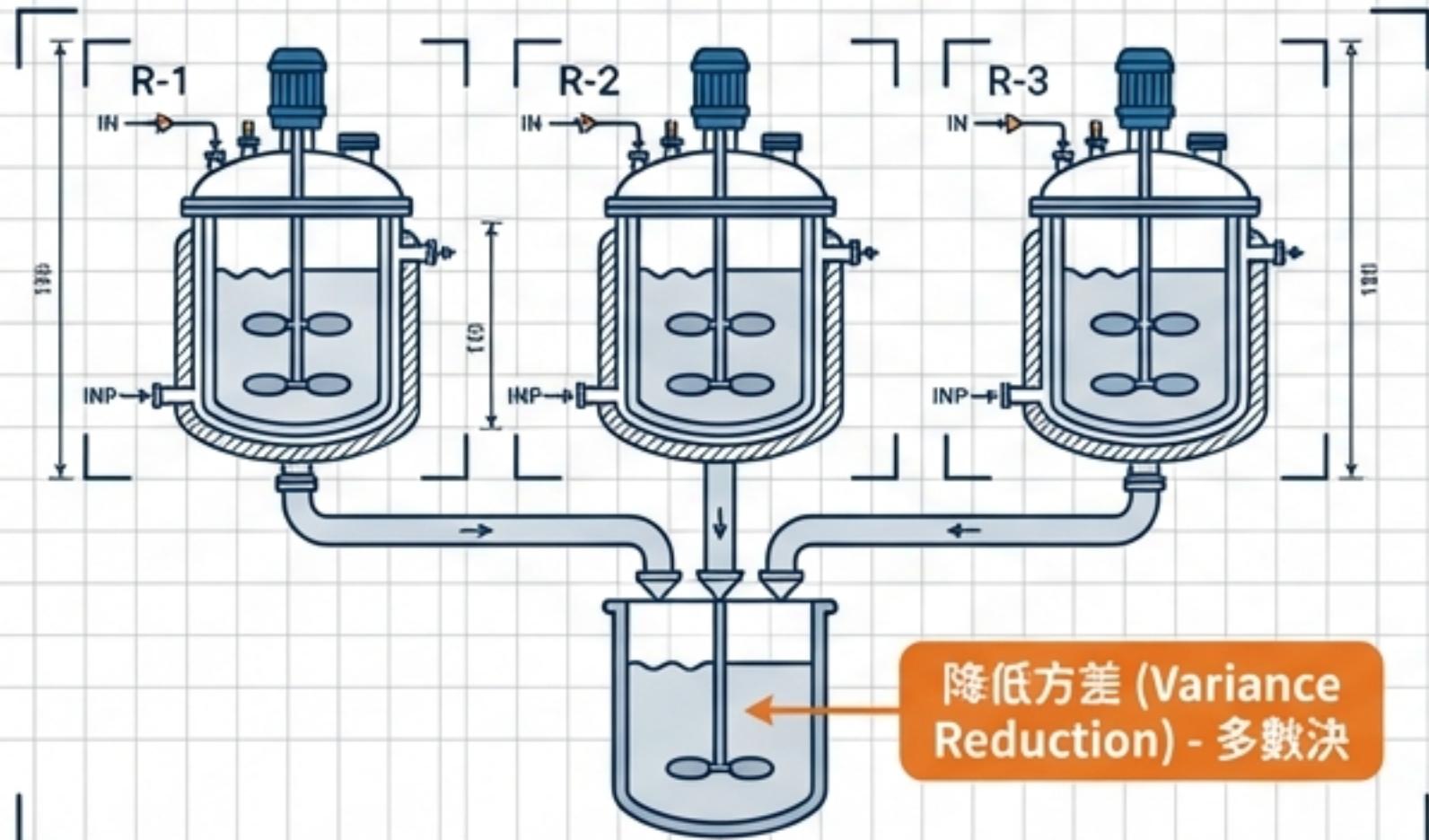
課程：AI在化工上之應用

最後更新：2026-01-28

授權：CC BY-NC-SA 4.0

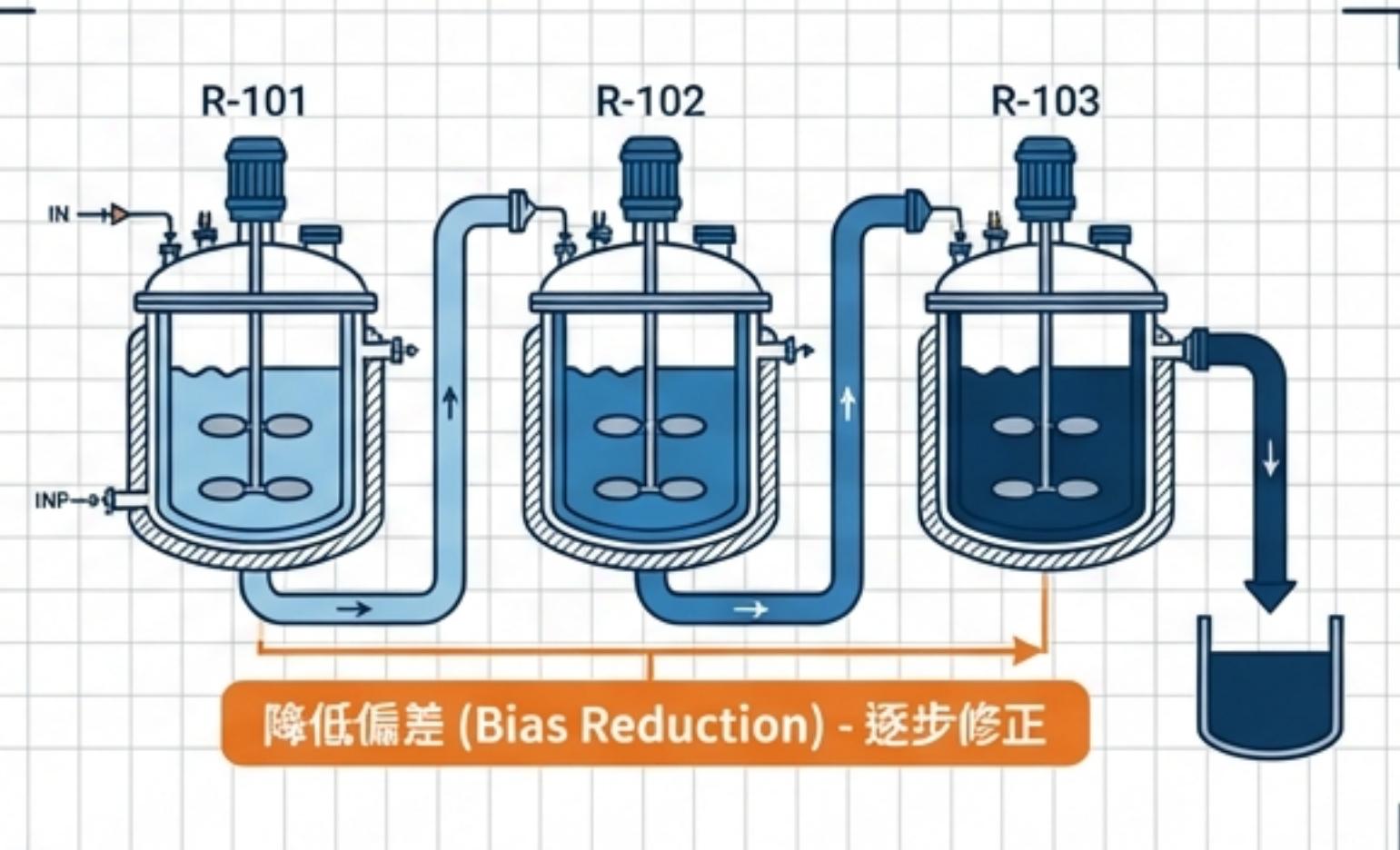
架構對比：並聯反應器 (Bagging) vs. 串聯精製 (Boosting)

Random Forest (Bagging)



- 獨立訓練 (Independent)
- 並行運算 (Parallel)
- 深樹 (Deep Trees)
- 防止過擬合 (Low Risk)

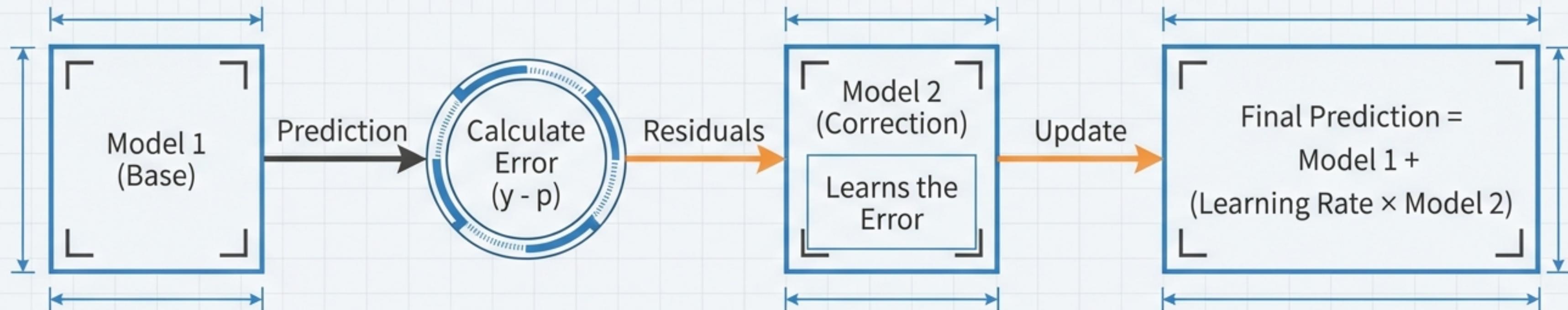
Gradient Boosting



- 依賴關係 (Dependent)
- 序列訓練 (Sequential)
- 淺樹 (Shallow Trees)
- 需精細調參 (High Maintenance)

Key Insight: Boosting 的核心不在於「投票」，而在於「專注修正前一個模型的錯誤」。

運作原理：基於殘差 (Residuals) 的序列優化



加法模型 (Additive Model)

$F(x) = \sum f_m(x)$ 。最終模型是所有弱學習器的總和。

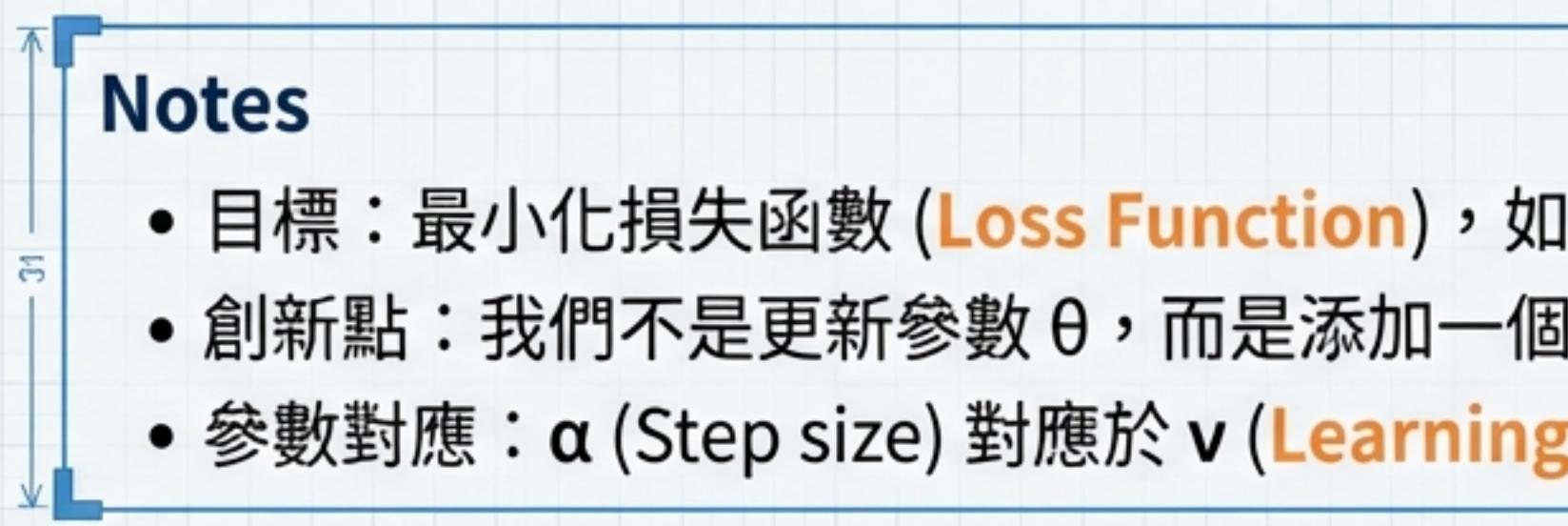
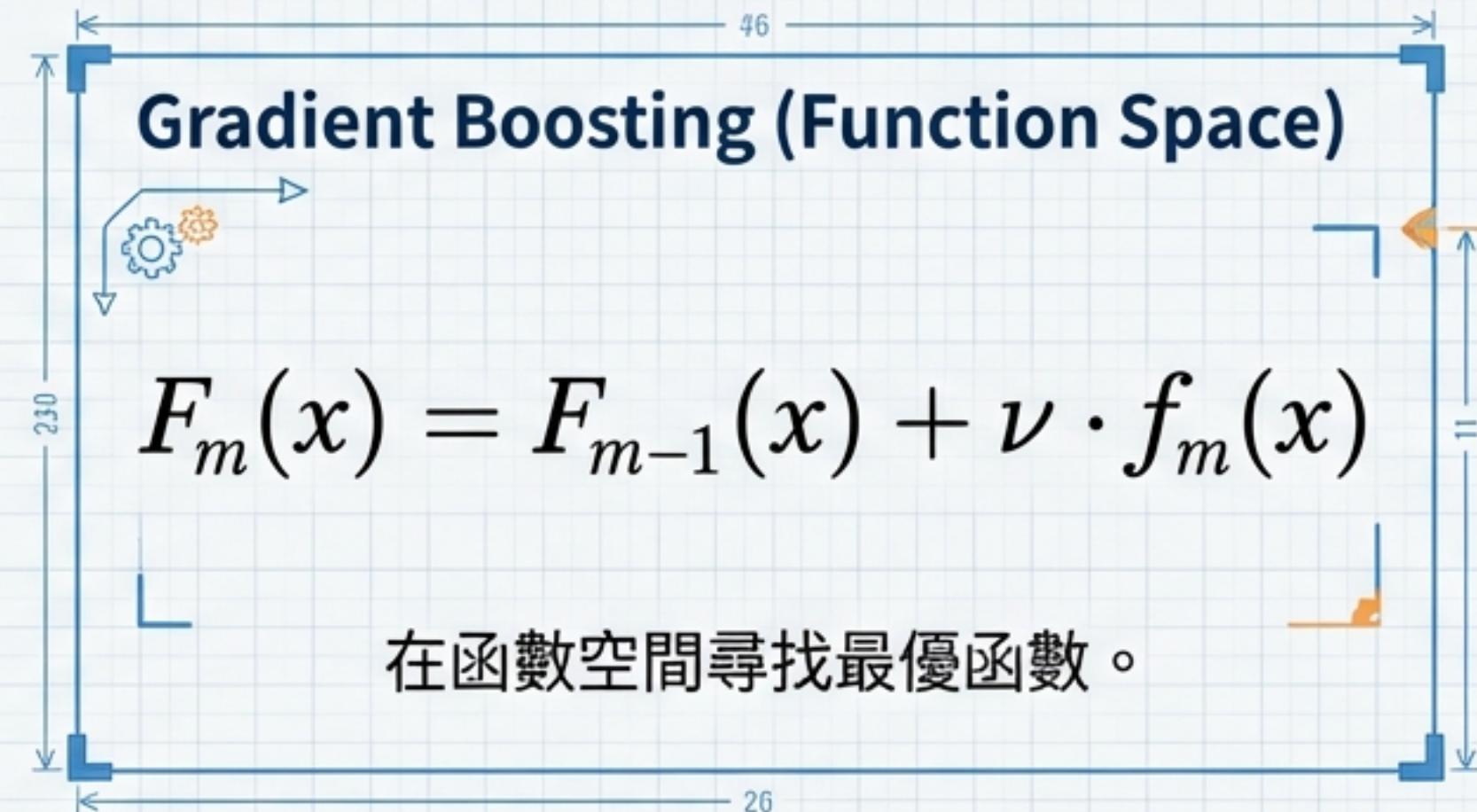
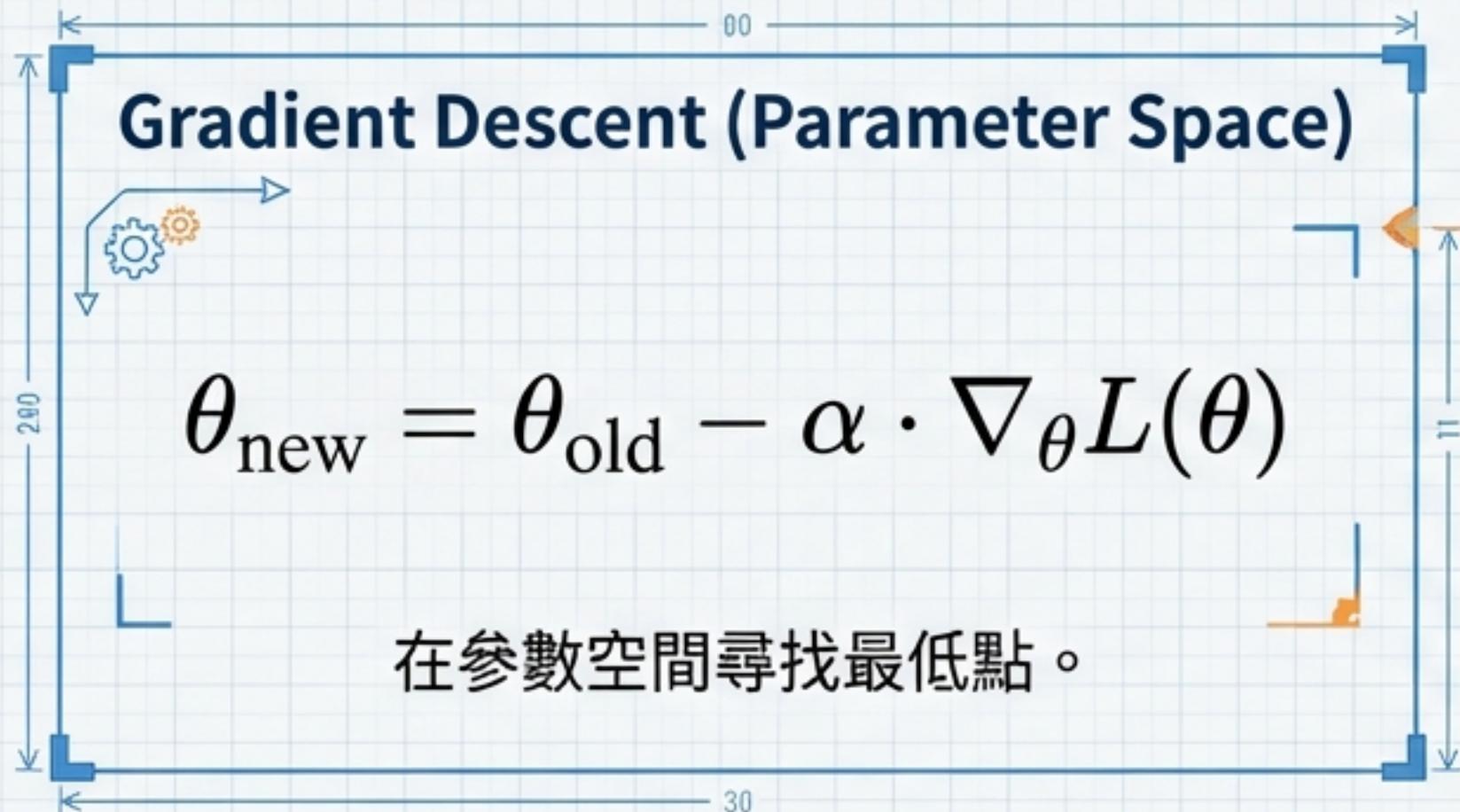
殘差 (Residuals)

負梯度 (Negative Gradient) 在數學上等同於殘差。每棵新樹都在擬合當前模型的預測誤差。

弱學習器 (Weak Learner)

通常使用深度僅 3-5 層的決策樹 (Decision Stumps)。

數學框架：函數空間中的梯度下降



實作介面：Sklearn GradientBoostingClassifier

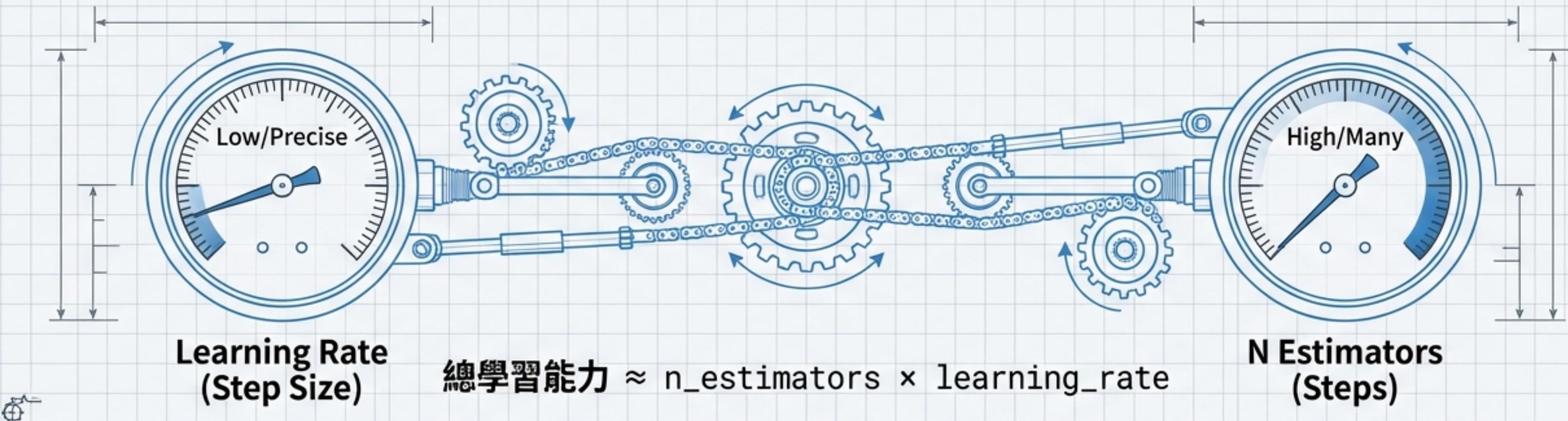
```
from sklearn.ensemble import GradientBoostingClassifier

# 初始化模型：化工反應預測
gb_clf = GradientBoostingClassifier(
    n_estimators=100,          # 樹的數量
    learning_rate=0.1,         # 學習率（修正步長）
    max_depth=3,              # 樹的深度（淺樹！）
    subsample=0.8,             # 隨機採樣比例
    random_state=42
)

# 訓練與預測
gb_clf.fit(X_train, y_train)
y_pred = gb_clf.predict(X_test)
```

關鍵控制閥
(Critical Control
Valves)

動力參數調校：Learning Rate 與 Estimators 的權衡



策略

策略 A：快速實驗

策略 B：最佳性能

參數配置

LR=0.1, Trees=100

LR=0.01, Trees=500+

性能特徵

快，但較粗糙

慢，但精度高，泛化好



注意：學習率過高容易導致振盪或過擬合；學習率過低則需要大量的樹，增加計算成本。

2 mm

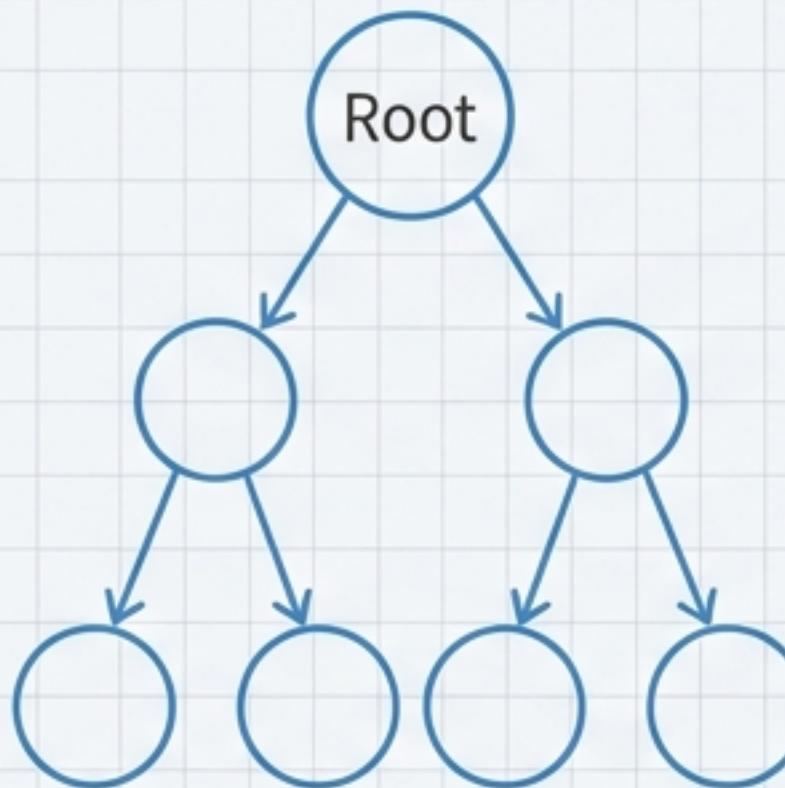
20 mm

5 mm

3 mm

穩定性控制：限制複雜度與隨機性

1. Max Depth (反應器規格)

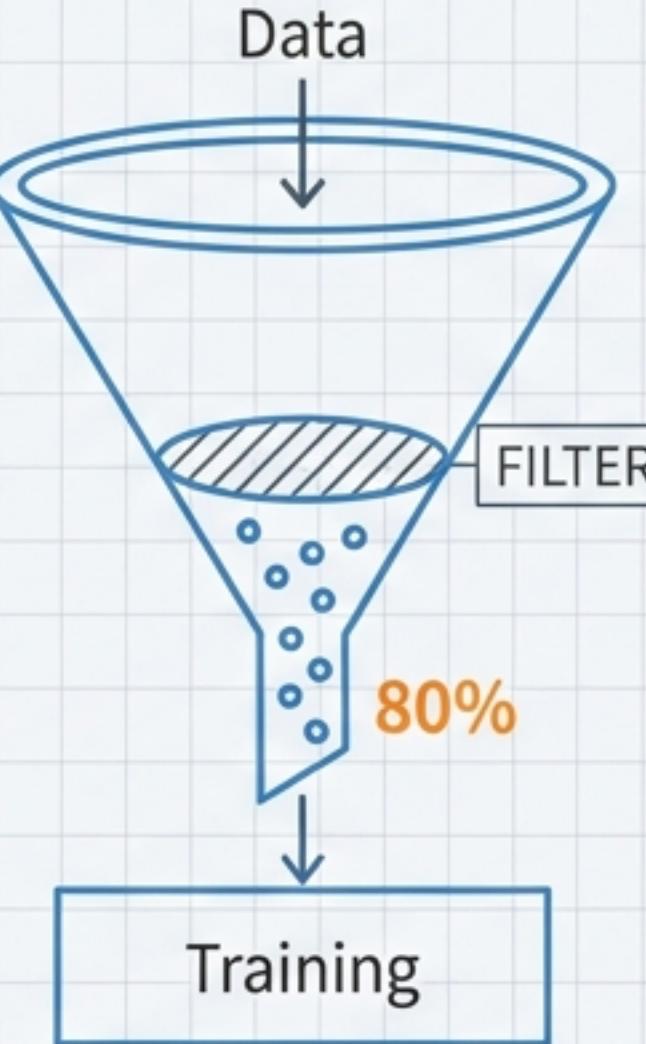


Rule: Gradient Boosting
必須使用淺樹 (Weak Learners)。

Recommendation:
`max_depth = 3 - 5`

Reasoning: GB 透過累加降低偏差，單棵樹不需太強，否則極易過擬合。

2. Subsample (隨機進料)



Concept: Stochastic Gradient Boosting

Setting:
`subsample = 0.8`
(每次迭代只使用 80% 數據)

Benefit: 引入隨機性以降低方差，同時加速訓練。

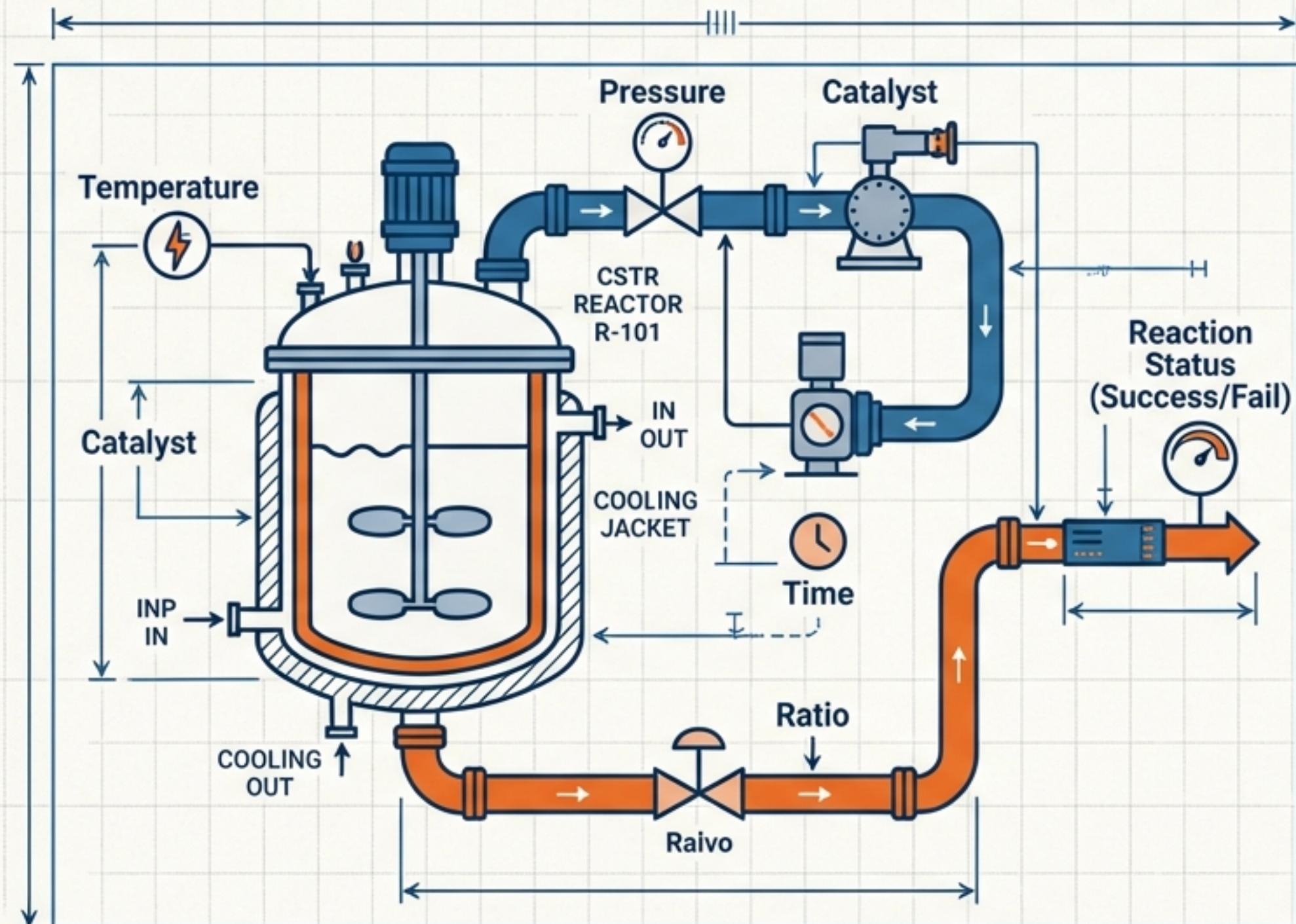
規格對比表：Random Forest vs. Gradient Boosting

規格指標 (Spec)	Random Forest	Gradient Boosting
主要功能	降低方差 (Variance)	降低偏差 (Bias)
單元組件	深樹 (Deep Trees)	淺樹 (Shallow Trees)
並行處理	✓ 支援 (快速)	✗ 不支援 (序列慢速)
容錯率	高 (不易過擬合)	低 (需監控 Learning Rate)
精度上限	中高	極高 (競賽首選)
調校難度	簡單 (隨插即用)	困難 (參數連動強)



結論：RF 是穩健的「通用工作馬」，GB 是需要精細操作的「F1 賽車」。

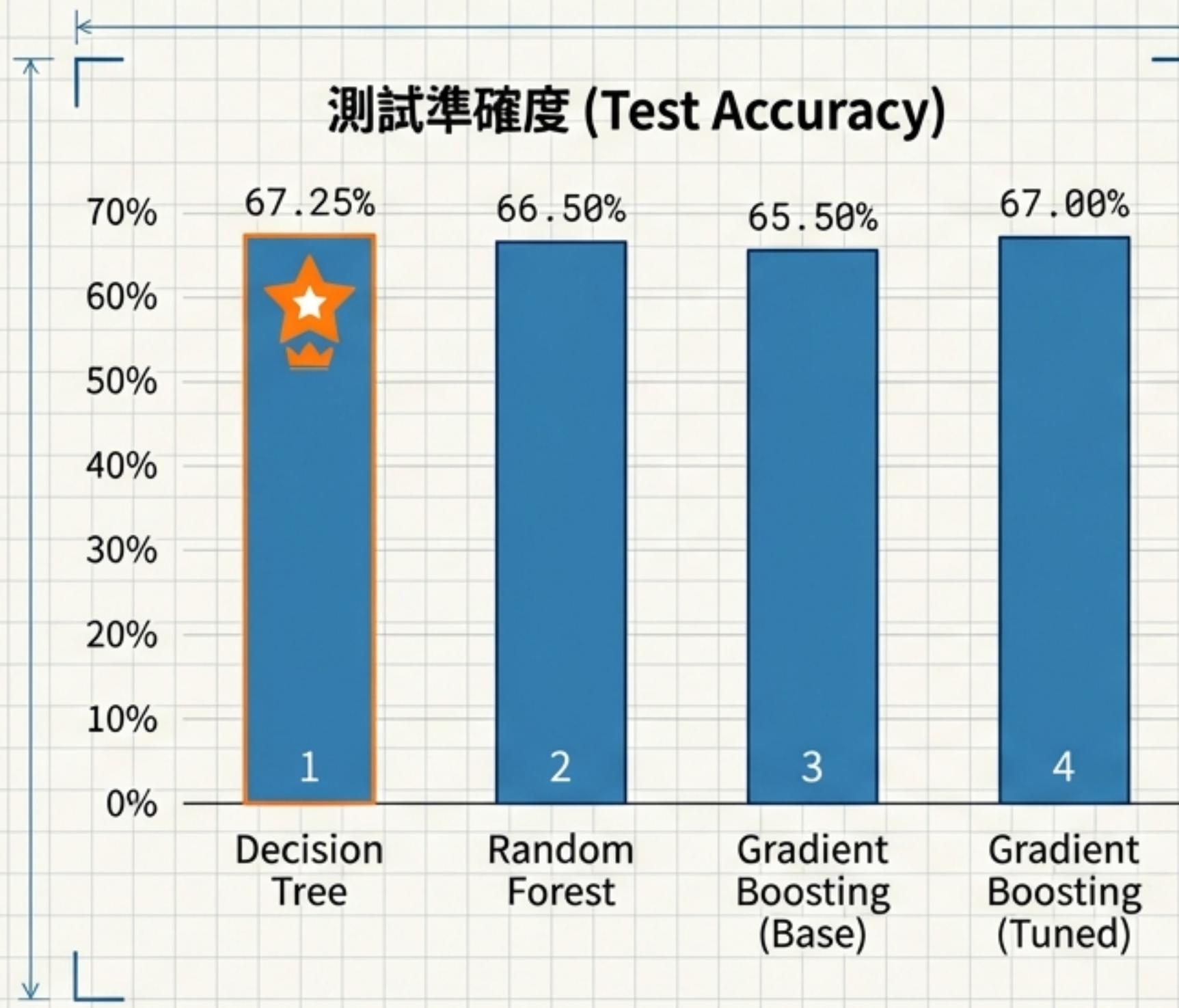
案例研究：化工反應器產率預測



數據規格 (Data Specs) :

- 樣本數 : 2000 (生成的模擬數據 v4.0)
- 特徵 (Features) : 5 個化學製程變數
- 目標 (Target) : 預測反應是否成功 (Binary Classification)
- 類別分布 : 失敗 56.2% / 成功 43.8%

性能測試：當高階工具遇上簡單工況



分析與結論 (Analysis & Conclusion)



現象：最複雜的模型表現反而略遜於單棵決策樹。

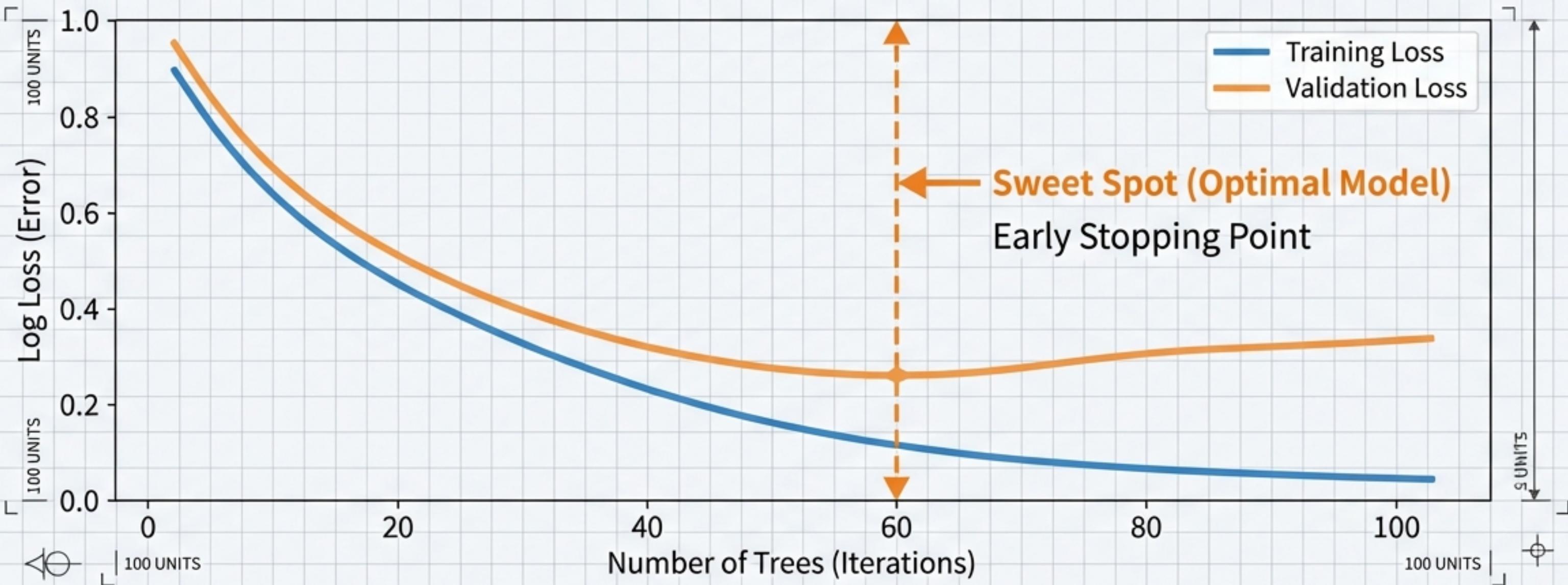


原因：模擬數據的決策邊界是簡單的「階梯式」函數 (Step Function)。



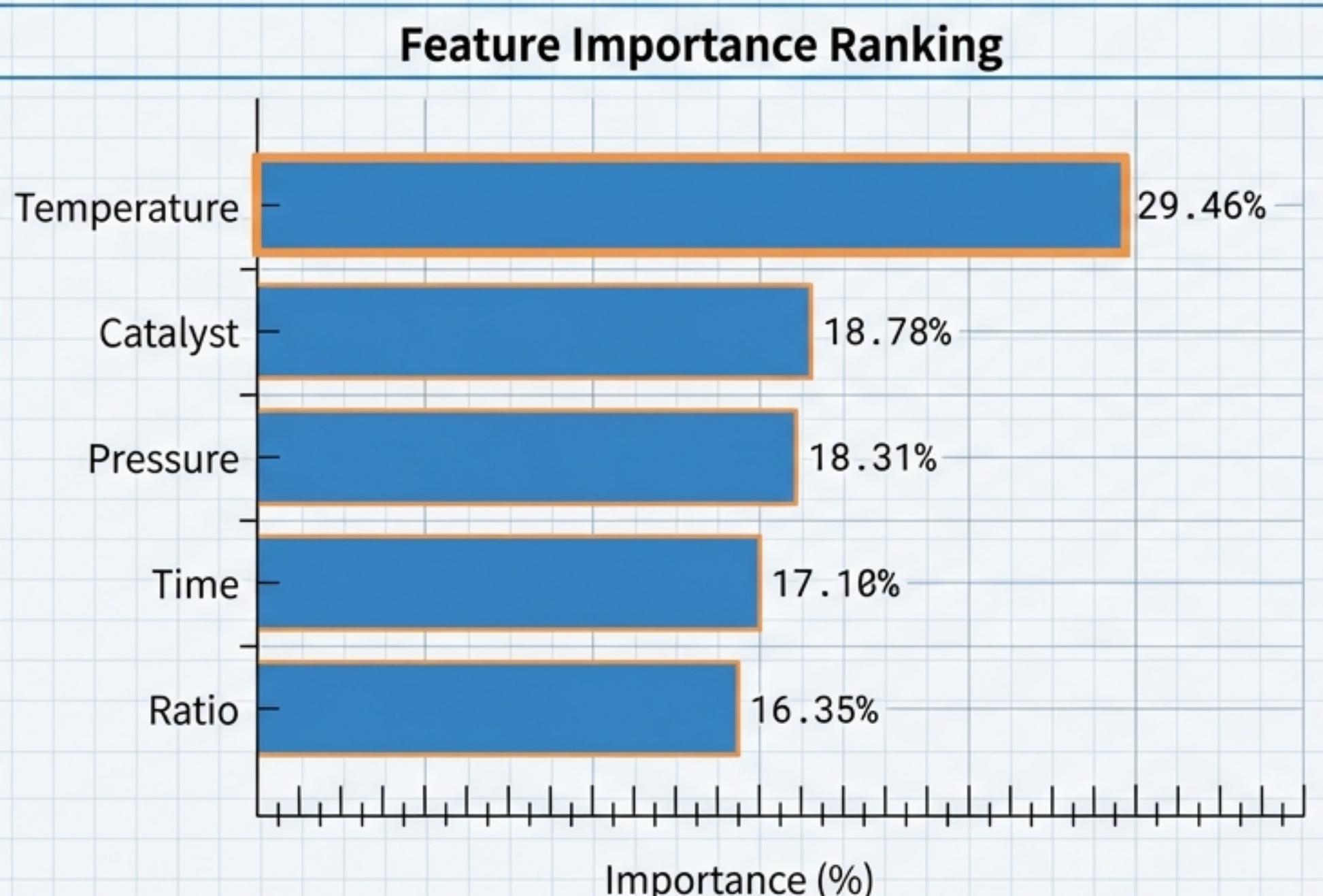
結論：殺雞焉用牛刀。GB 的優勢在於處理複雜非線性與噪聲，在簡單規則下優勢無法體現。

過程監控：學習曲線與 Early Stopping



- Staged Predict: 監控每增加一棵樹後的性能變化。
- Early Stopping: 當驗證分數在 `n_iter_no_change` 次迭代後不再提升時，自動停止訓練。防止過擬合的最有效手段。

關鍵變數分析：Feature Importance



洞察 (Insights) :

- 結果與 Random Forest 一致，確認溫度是主導反應的關鍵物理變量。
- GB 的特徵重要性計算考慮了每棵樹的權重，更能反映變數對「修正錯誤」的貢獻度。



決策指南：何時選用 Gradient Boosting ?

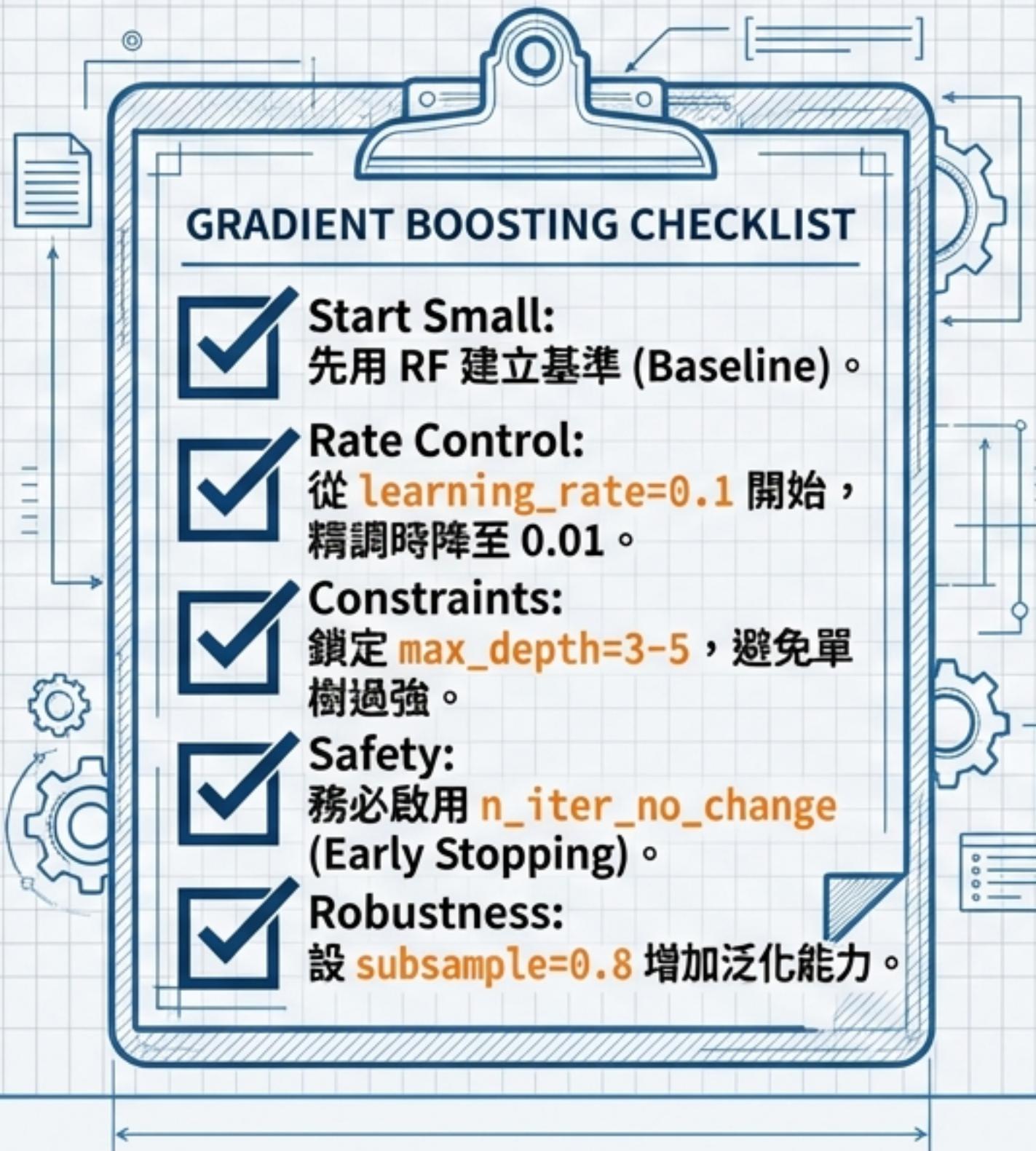
✓ 推薦使用 (Recommended)

- 表格數據 (Tabular Data)
- 特徵數量中等 (10-1000)
- 追求極致準確率 (High Accuracy)
- 數據乾淨但關係複雜 (Complex Non-linearities)

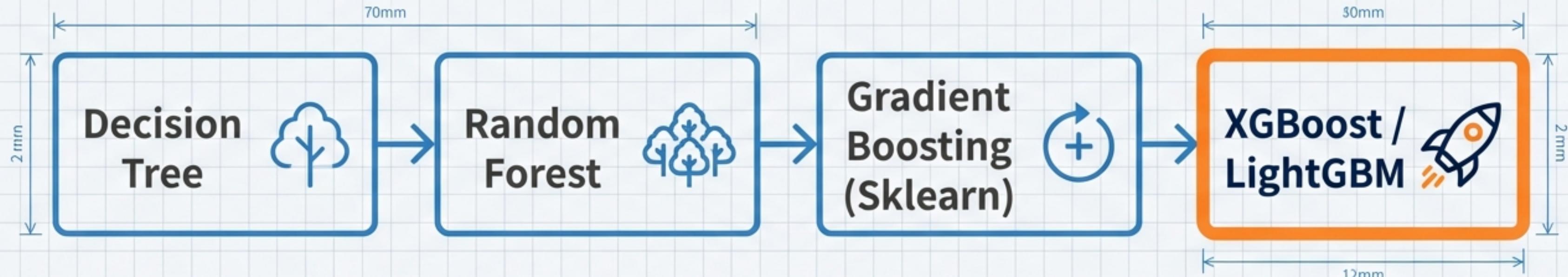
✗ 不推薦 (Not Recommended)

- 影像或文本數據 (Use Deep Learning)
- 數據量極大 (>1M 樣本，Use XGBoost)
- 噪聲極高 (Random Forest 更好)
- 需要即時/毫秒級預測 (RF 或線性模型)

工程實務檢查表 (Best Practices)



總結與展望：通往 XGBoost 之路



- 1. GB 是基於「錯誤修正」的加法模型。
- 2. 以時間換取精度：訓練慢，但理論上限高。
- 3. 本案例雖未超越 RF，但在真實複雜化工數據中通常有 3-8% 優勢。

Next Step: 下一階段將介紹 XGBoost —— 工業界的梯度提升標準，具備正則化項與硬體加速能力的終極進化版。