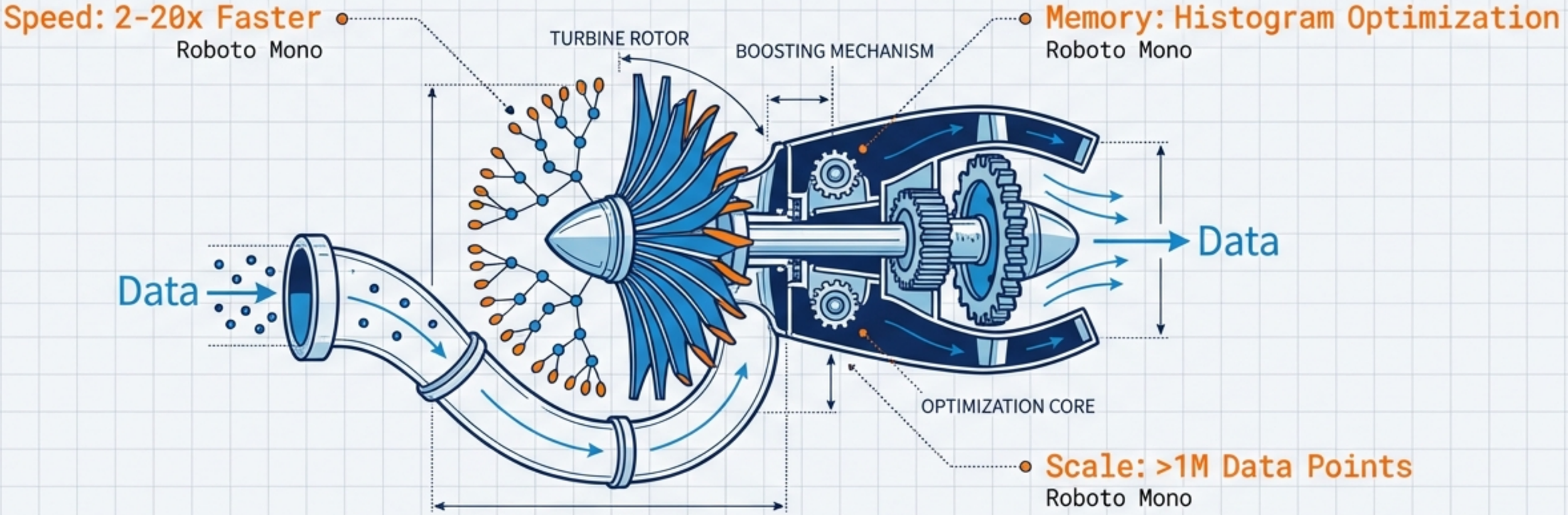
Unit 13: LightGBM 模型應用實務

專為大規模化工數據設計的極速梯度提升框架



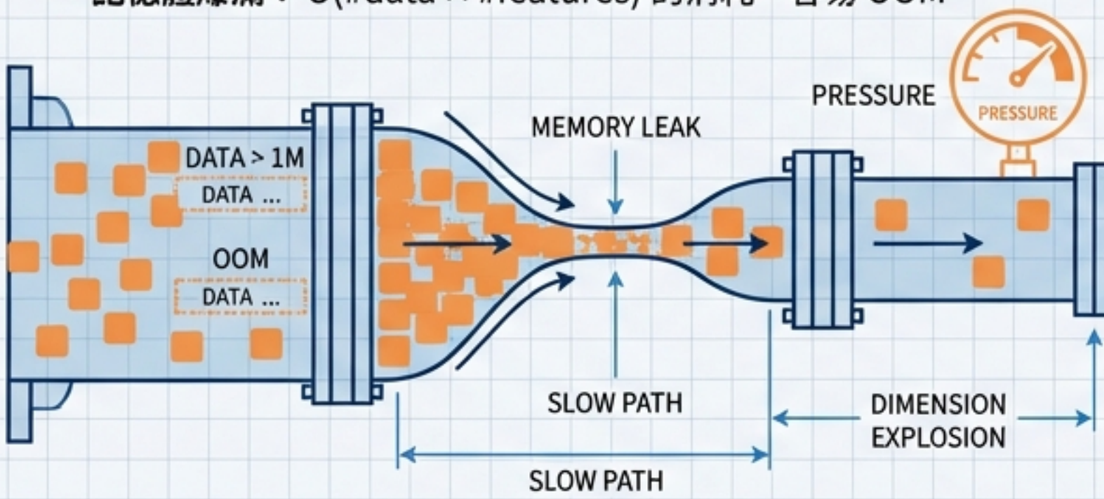
為什麼需要升級？大數據下的效能瓶頸

The Bottleneck (XGBoost)

ALERT: Performance Issues!

傳統 XGBoost 在處理百萬級數據 (>1M) 時面臨挑戰：

- **速度慢：**Pre-sorted 演算法效率低，訓練耗時長。
- **記憶體爆滿：**O(#data × #features) 的消耗，容易 OOM。



傳統 XGBoost 在處理百萬級數據 (>1M) 時面臨挑戰：

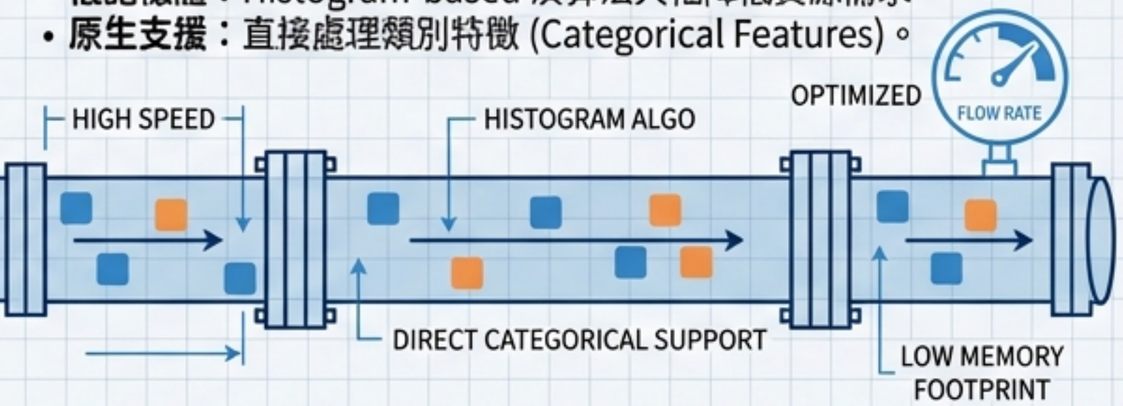
- 速度慢：Pre-sorted 演算法效率低，訓練耗時長。
- 記憶體爆滿：O(#data × #features) 的消耗，容易 OOM。
- 特徵處理：類別特徵需手動 One-Hot Encoding，導致維度爆炸。

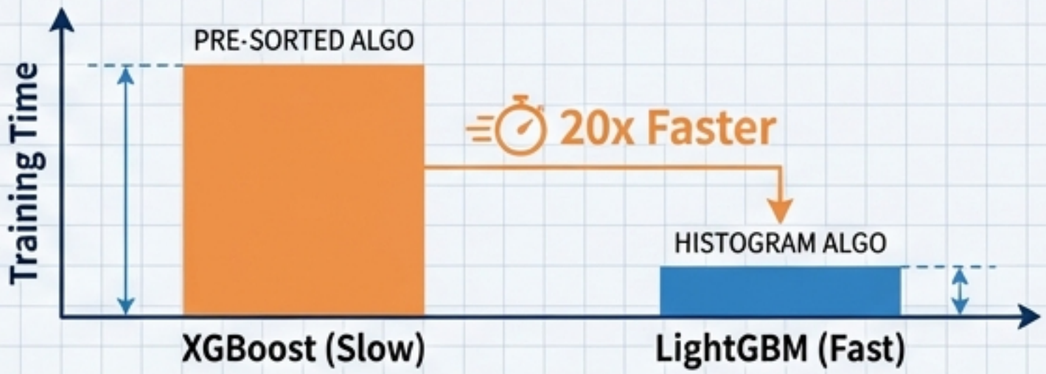
The Solution (LightGBM)

SOLUTION: Optimized Flow!

LightGBM 的突破性優勢：

- **極速訓練：**比 XGBoost 快 2-20 倍。
- **低記憶體：**Histogram-based 演算法大幅降低資源需求。
- **原生支援：**直接處理類別特徵 (Categorical Features)。



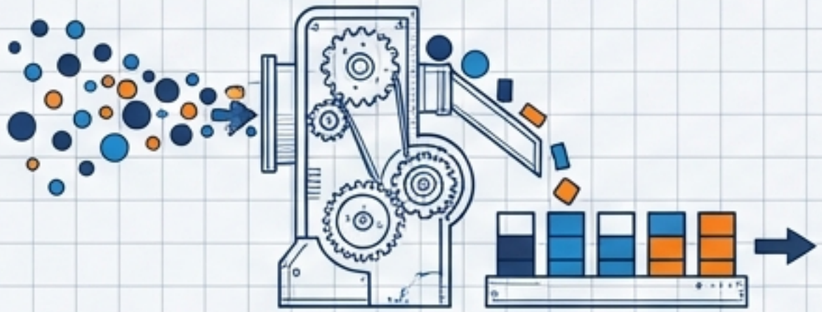


Technical specs and performance metrics are approximate and may vary based on hardware and dataset characteristics.

核心架構：LightGBM 的動力來源

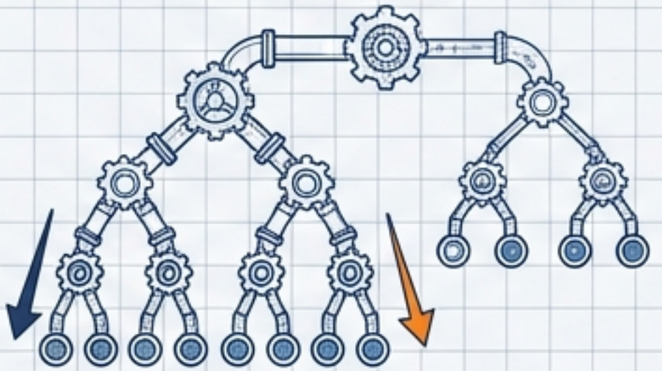
LightGBM Engine

A Histogram-based Algorithm



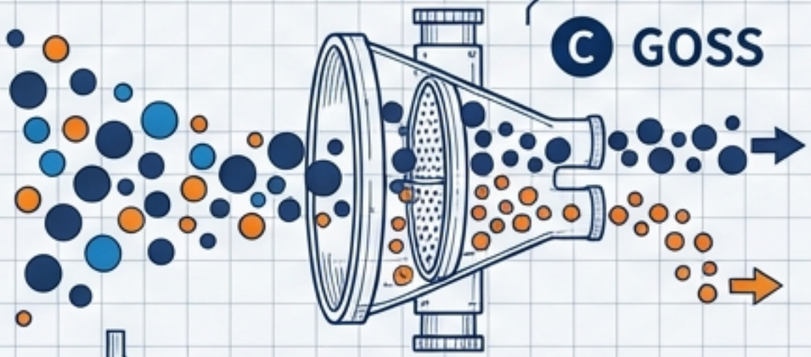
降低記憶體與計算複雜度
O(data) → O(k)

B Leaf-wise Growth

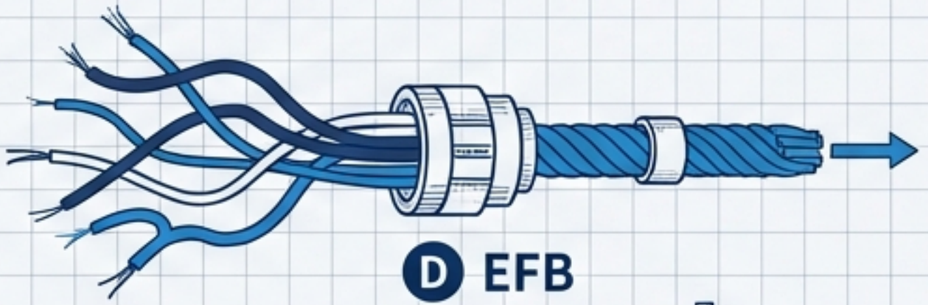


追求最大誤差降低
(Max Gain)，收斂更快

C GOSS

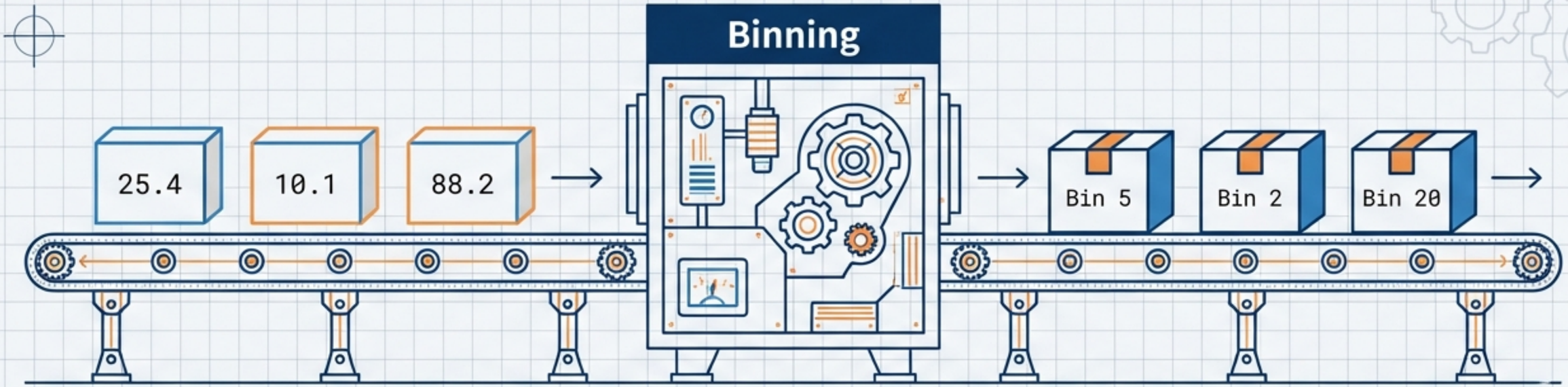


D EFB



針對數據採樣與特徵維度的雙重加速技術

Histogram-based 演算法：離散化與差分加速

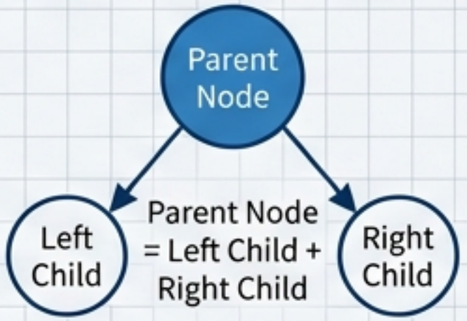


Binning (分箱)

將連續特徵值離散化為 k 個區間 (default k=255)。記憶體消耗從 32-bit float 降為 8-bit integer (1/8 記憶體占用)。

```
feature_value → bin_index ∈ {0, 1, ..., k-1}
```

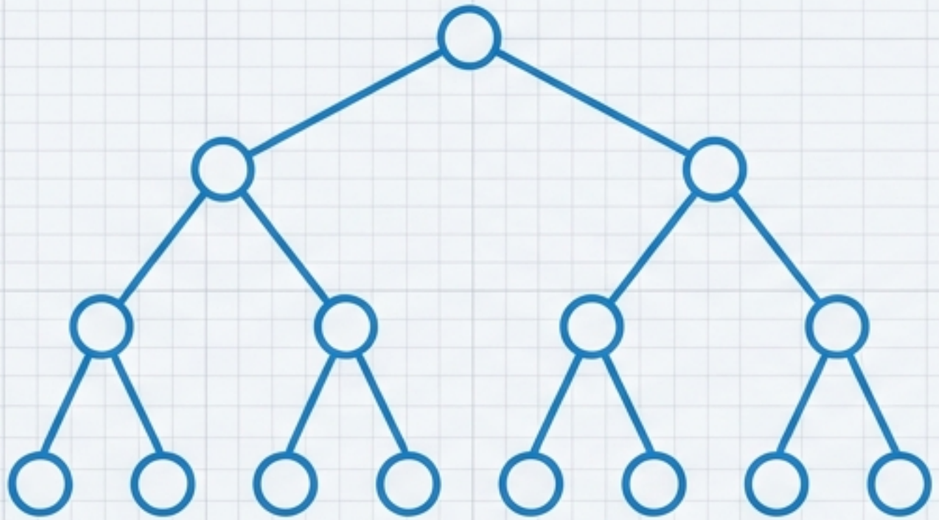
Histogram Subtraction (差分加速)



父節點的 Histogram = 左子樹 + 右子樹。只需計算較小子樹的 Histogram，另一邊可透過「減法」得出計算量減少 50%。

樹的生長策略：Leaf-wise vs. Level-wise

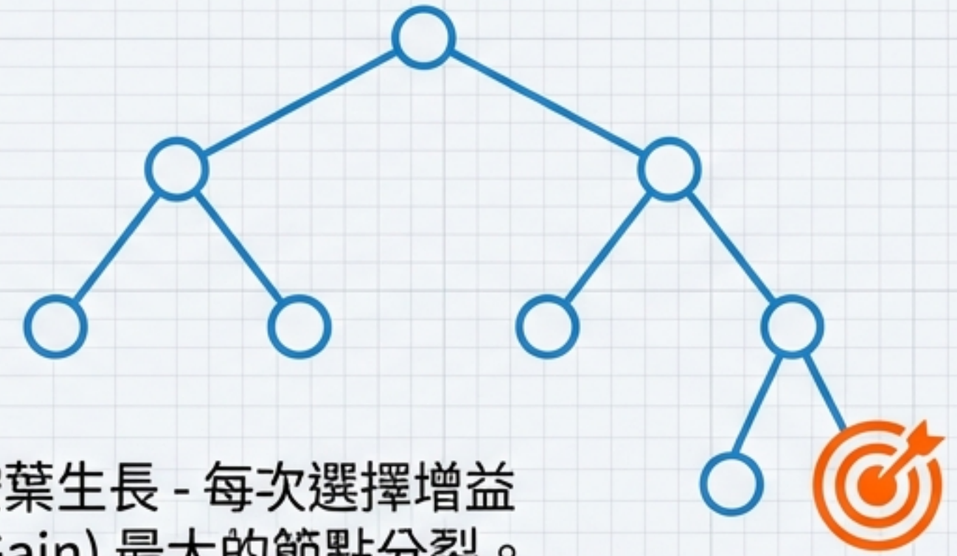
Level-wise (XGBoost)



按層生長 - 同一層所有葉子同時分裂。

結構平衡，不易過擬合，但分裂低增益節點浪費資源。

Leaf-wise (LightGBM)



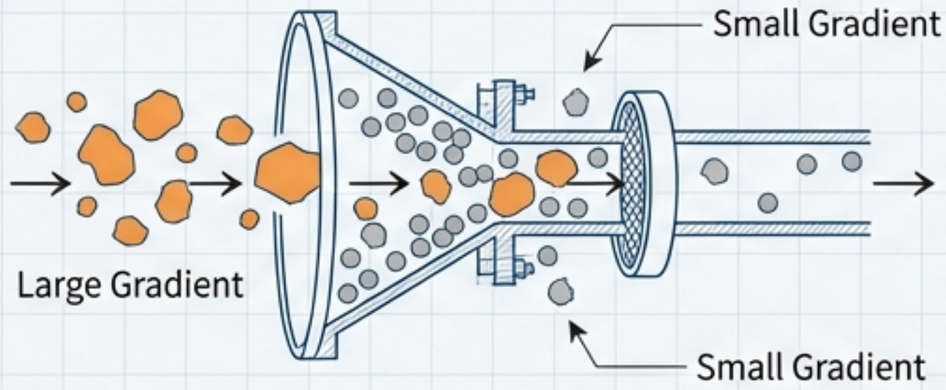
按葉生長 - 每次選擇增益 (Gain) 最大的節點分裂。

$$L^* = \text{argmax } Gain(L)$$

容易生長過深導致過擬合 (Overfitting)，必須限制 max_depth。

加速雙引擎：GOSS 與 EFB 技術

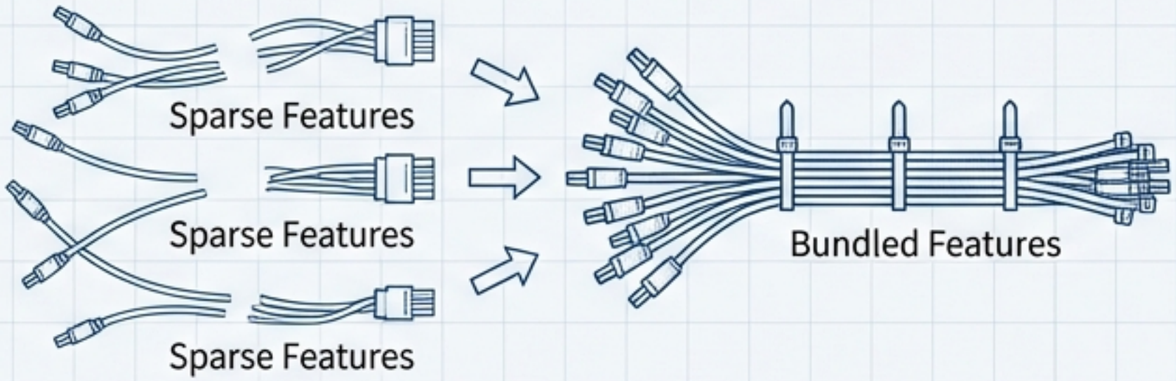
GOSS (Gradient-based One-Side Sampling)



只關注學不好的樣本：保留大誤差樣本，隨機採樣小誤差樣本。訓練數據量大幅減少，但保留了最重要的梯度資訊。

```
  Data Sample:
   | Gradient | Action
>  |   0.85   | Keep          |
   |   0.12   | Sample (10%)  |
   |   0.92   | Keep          |
```

EFB (Exclusive Feature Bundling)

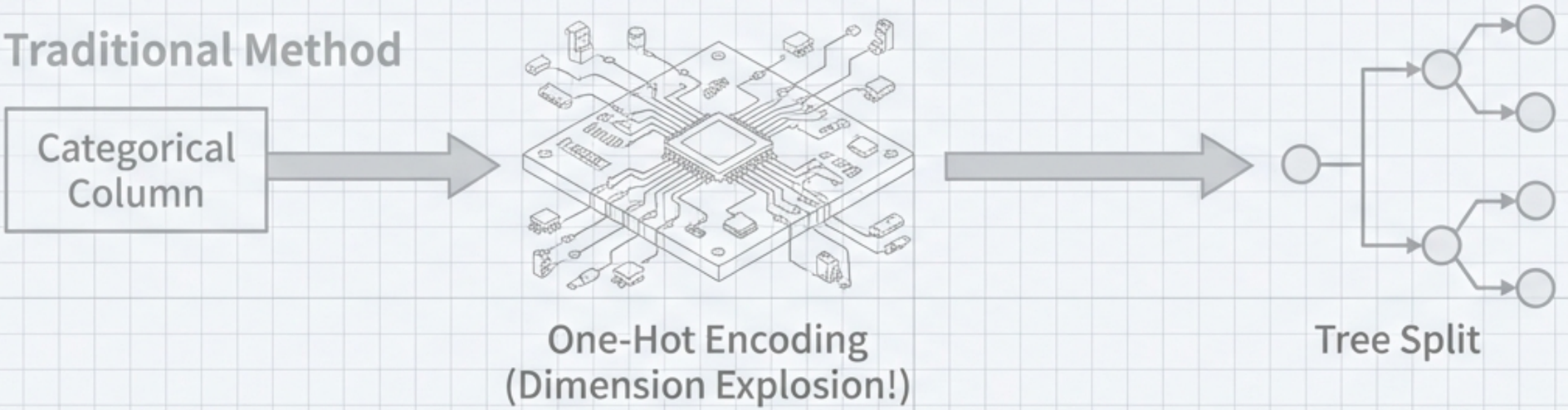


互斥特徵的捆綁：高維稀疏特徵 (Sparse Features) 降維，將互斥特徵合併，不損失資訊。

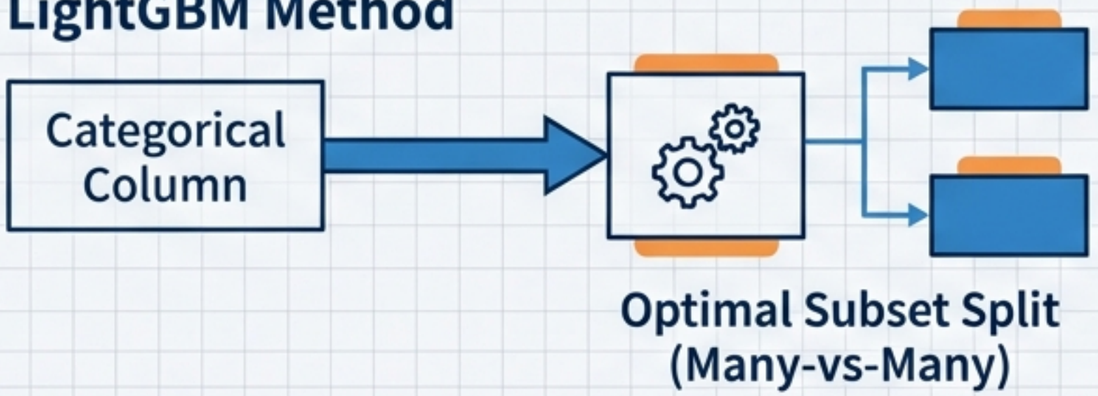
```
 Feature Mapping:
  | Feature A | Feature B | Bundled |
>|     1     |     0     |    1    |
  |     0     |     2     |    2    |
  |     0     |     0     |    0    |
```

類別特徵優化：告別 One-Hot Encoding

Traditional Method



LightGBM Method



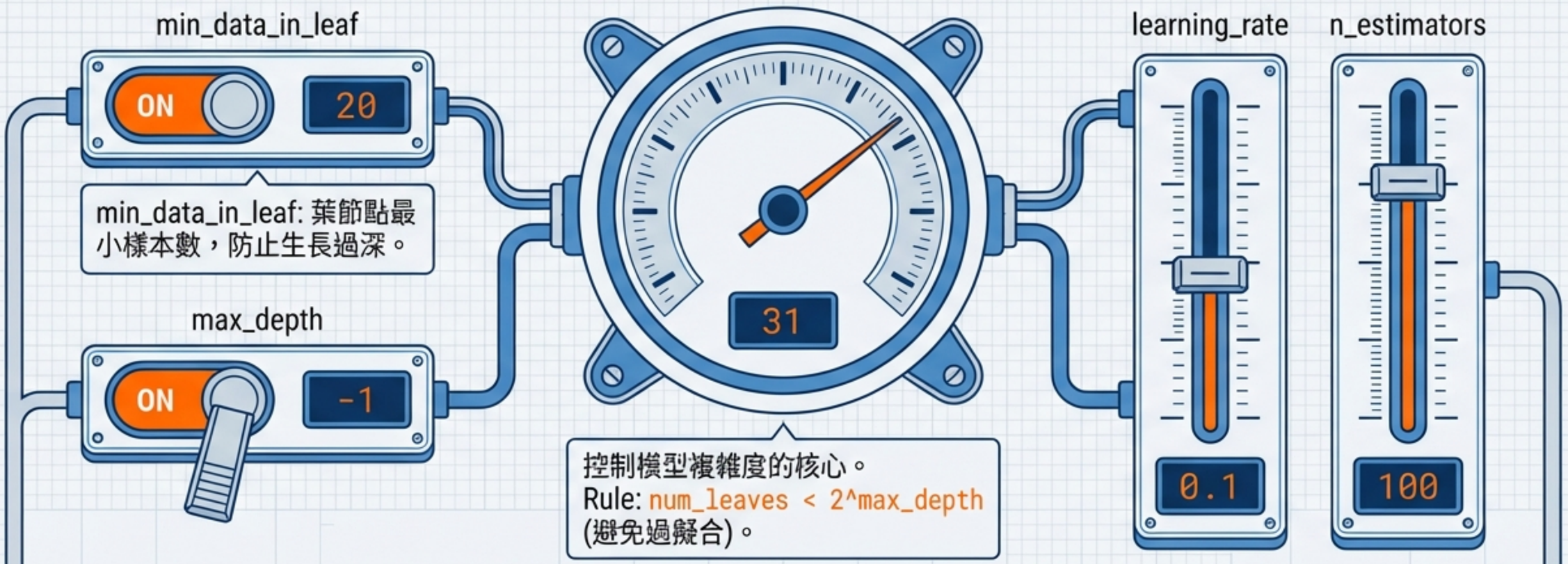
LightGBM 原生支援類別特徵，無需預處理。

採用 Many-vs-Many 分割策略，尋找最佳類別子集 S。

Formula: `S* = argmax Gain(S)`

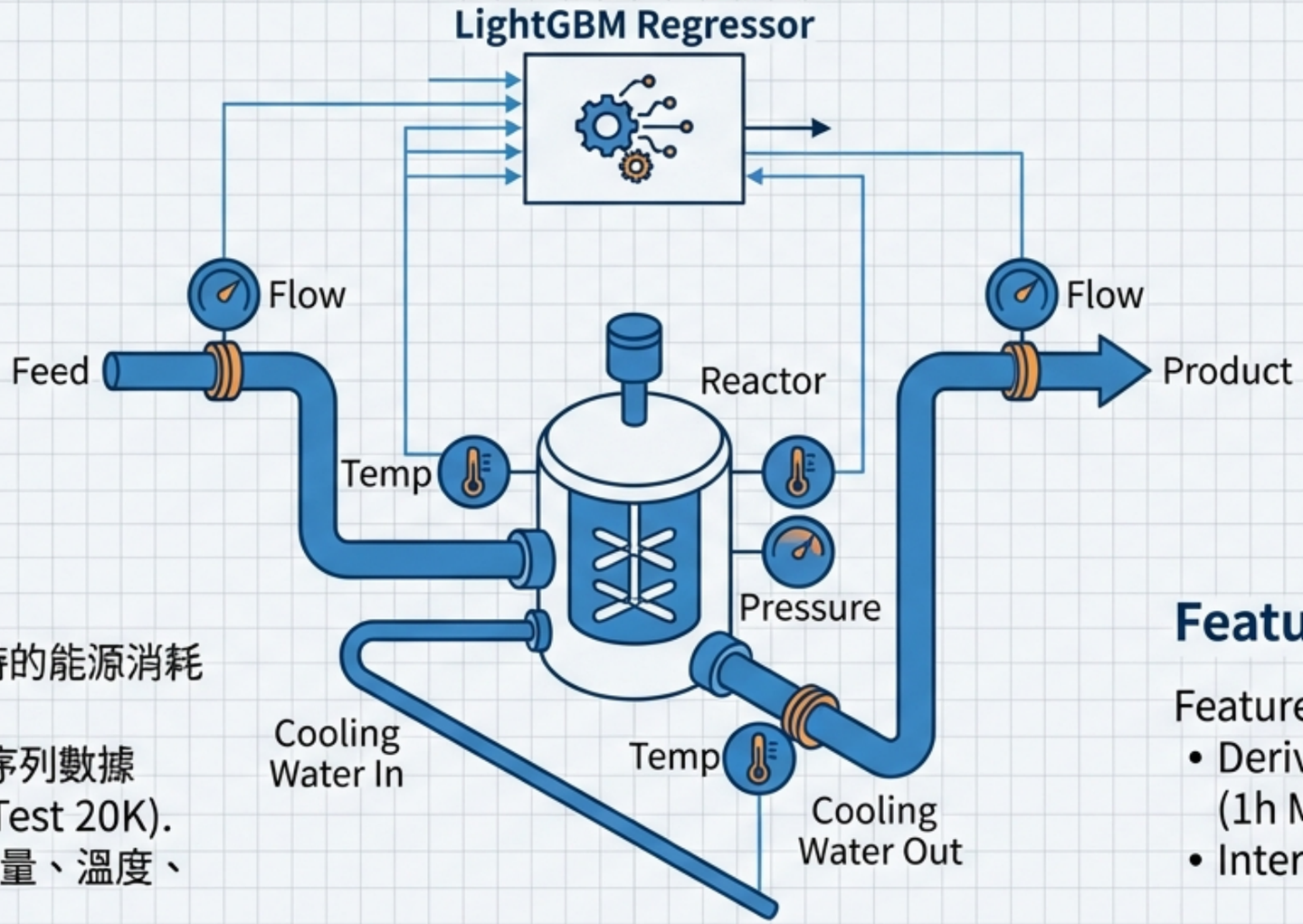
Performance: 在類別特徵多的數據集上，速度提升可達 8 倍。

參數控制面板 (Hyperparameters Dashboard)



Hyperparameter Mapping		
LightGBM	num_leaves	≈ XGBoost max_depth (conceptually)
LightGBM	min_data_in_leaf	≈ XGBoost min_child_weight

實戰案例 I：化工廠能源消耗預測 (Regression)



Context

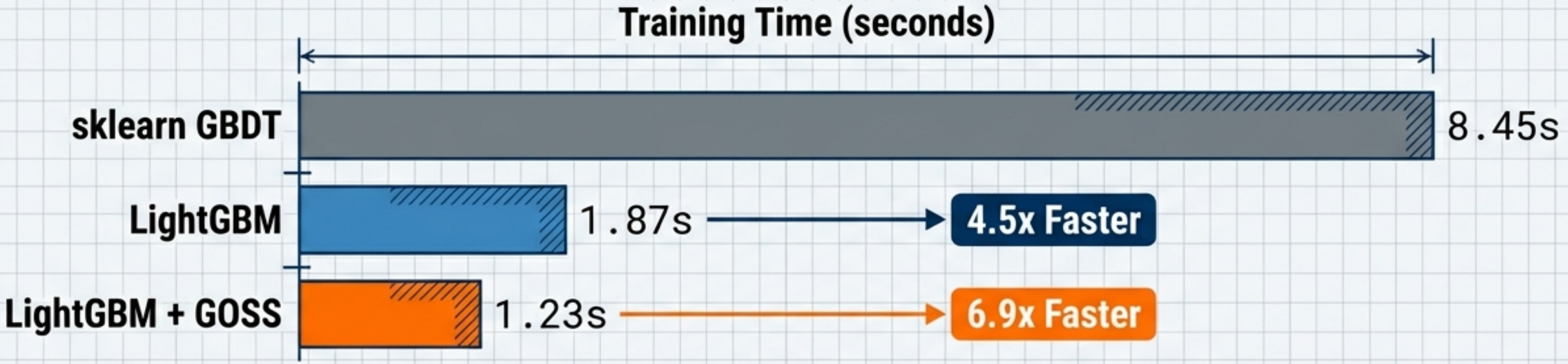
目標：預測化工廠每小時的能源消耗量 (kWh)。
Scale: 100,000 筆時間序列數據 (Train 60K / Val 20K / Test 20K).
Features: 27 個 (包含流量、溫度、壓力、時間特徵)。

Feature Engineering

Feature Engineering:

- Derived: Rolling Statistics (1h Mean/Std)
- Interaction: Temp × Flow

案例 I 結果：GOSS 的極速體驗



Performance Metrics Table

Metric:	RMSE	R^2	
LightGBM:	85.68	0.9838 (Best Accuracy)	✓
LightGBM + GOSS:	87.12	0.9832 (Minimal loss)	✓

Top Factor: Flow_Cubed (流量三次方)。**Insight:** 實時監控場景首選 GOSS 模式。

實戰案例 II：設備故障診斷 (Classification)

Context

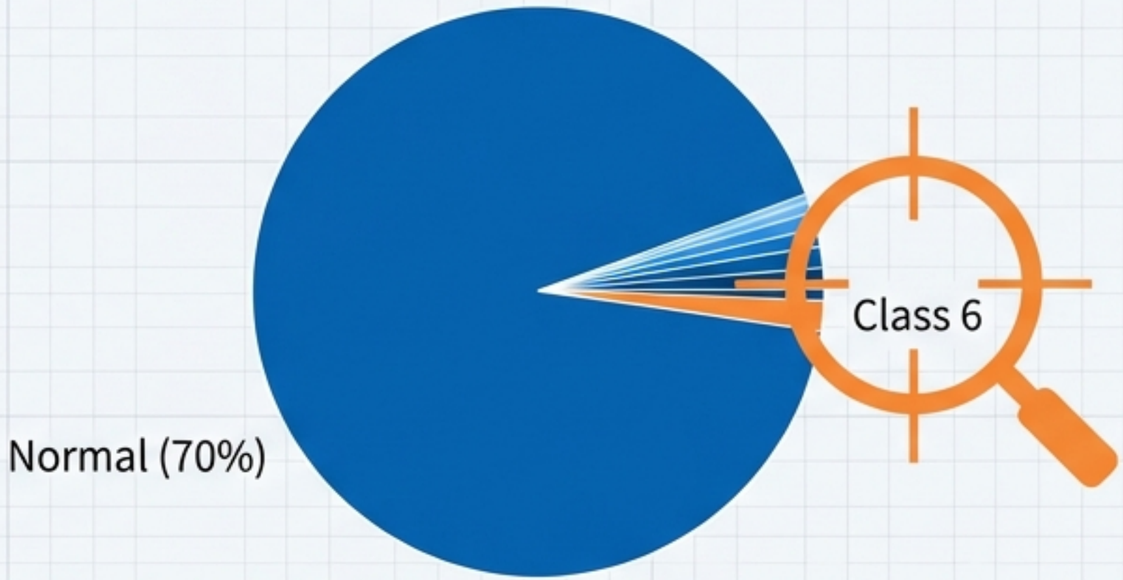
Context: 7 類設備狀態分類。
Class 0: 正常運行 (Normal).

Class 6: 緊急停機 (Emergency Shutdown) - Critical!

Features
30 Dimensions

The Challenge

Class Distribution



Ratio: Normal : Fault = 70 : 1
Risk: 模型容易忽略少數類別 (Class 6)，導致安全隱患。

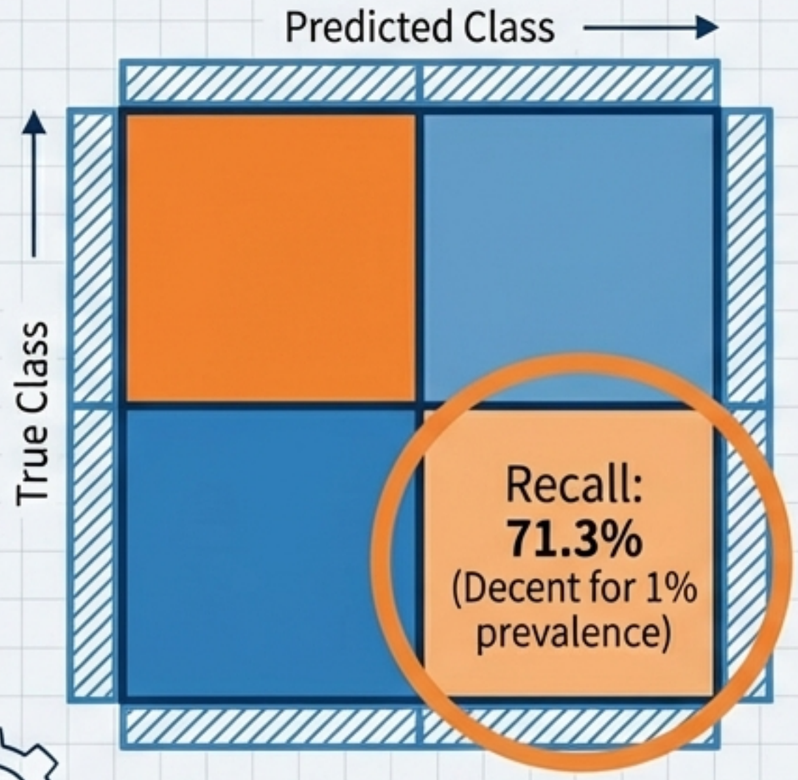
解決數據不平衡：權重策略與性能

使用 sample_weight 或 class_weight 增加少數類別的懲罰權重。

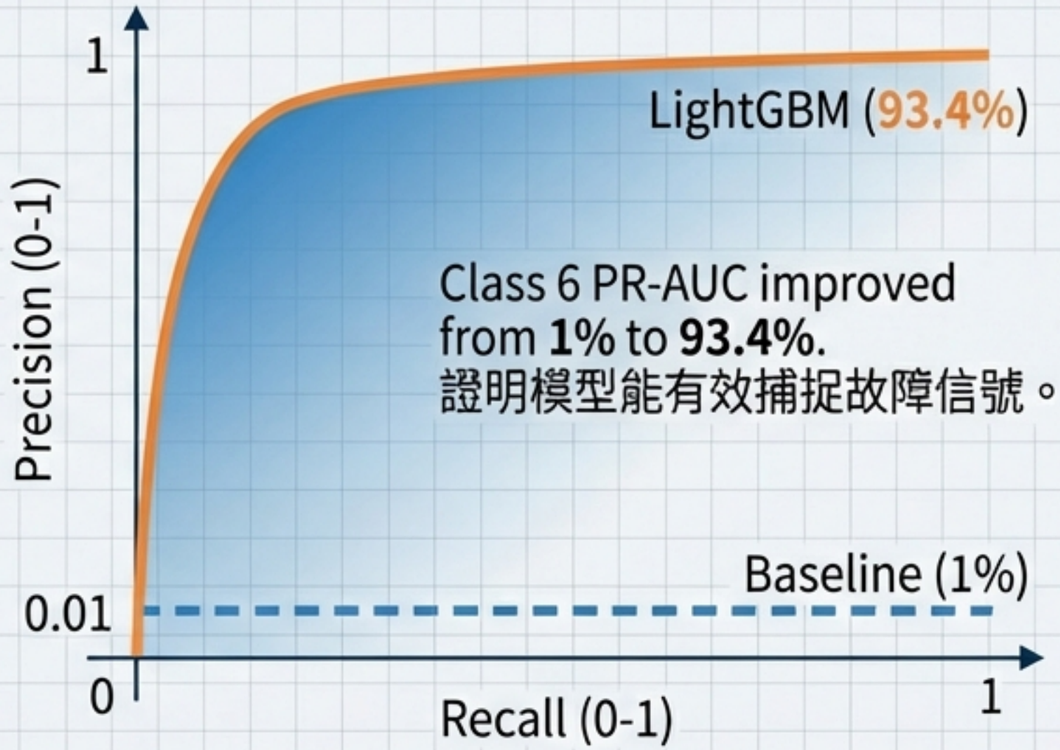
	Model	Time (s)	F1 (Weighted)
1	SVM	45.23s (Roboto Mono)	0.6890 (Roboto Mono)
2	sklearn GBDT	52.89s (Roboto Mono)	0.8056 (Roboto Mono)
3	LightGBM	5.67s (Roboto Mono)	0.8289 (Best) (Roboto Mono)

關鍵指標分析：混淆矩陣與 PR 曲線

Confusion Matrix (Class 6 Focus)



PR-Curve vs Baseline



Top Feature: Health_Index (Derived feature) dominates.

決策矩陣：LightGBM vs. XGBoost

Use XGBoost if:

- ☐ 數據量小 (< 10K 樣本)
- ☐ 追求絕對的模型穩定性
- ☐ 剛入門，依賴豐富社群資源

Use LightGBM if:

- ☑ 數據量大 (> 100K ~ 10M)
- ☑ 記憶體資源受限
- ☑ 擁有高維度的類別特徵 (High Cardinality)
- ☑ 需要快速迭代實驗 (Fast Prototyping)

總結：化工大數據的工業標準

極速 (Speed)

Histogram & GOSS 帶來 2-20 倍加速。

精準 (Accuracy)

Leaf-wise 生長策略在大數據下表現優異。

實用 (Practicality)

原生支援類別特徵與缺失值。

Next Step (Actionable):

- 開啟 `Unit13_LightGBM_Regression.ipynb`
- 練習調整 `num_leaves` 與 `min_data_in_leaf`
- 嘗試使用 GOSS 加速訓練

Industrial Blueprint_v1

Course Materials Available on LMS.