| 任務類型 | 選擇模型 | 優勢 |
| --- | --- | --- |
| 分類任務 | 隨機森林 | 天然處理類別不平衡，高維稀疏數據表現穩定 |
| 回歸任務 | 梯度提升 | 通過殘差修正逐步逼近真實值，適合連續值預測 |

本文件使用AI工具輔助生成

## 一、隨機森林分類器 (RandomForestClassifier)

### 核心原理：Bagging + 決策樹集成

`RandomForestClassifier(class_weight='balanced')`

1. **Bootstrap Aggregating (Bagging)**
   - 從原始數據集中 **有放回抽樣** 生成多個子數據集
   - 每個子集訓練一棵決策樹，公式表達：

$$\hat{f}_k(x) = \text{DecisionTree}(D_k), \quad D_k \sim \text{Bootstrap}(D)$$

     - $D$：原始數據集
     - $D_k$：第k個bootstrap抽樣子集
2. **特徵隨機性**
   - 每棵樹分裂時 **僅考慮隨機子集的特徵** (程式碼中未顯式設定，預設為$\sqrt{p}$)

$$m_{\text{try}} = \lfloor \sqrt{p} \rfloor \quad (p = \text{總特徵數})$$

3. **投票機制**
   - 最終預測為多數決：

$$\hat{y} = \text{mode}\left( \{\hat{f}_k(x)\}_{k=1}^{K} \right)$$

## 二、梯度提升回歸器 (GradientBoostingRegressor)

### 核心原理：Boosting + 加法模型

`GradientBoostingRegressor()`

1. **前向分步算法**
   - 模型為多個弱學習器（決策樹）的加權和：

$$F_m(x) = F_{m-1}(x) + \nu \cdot h_m(x)$$

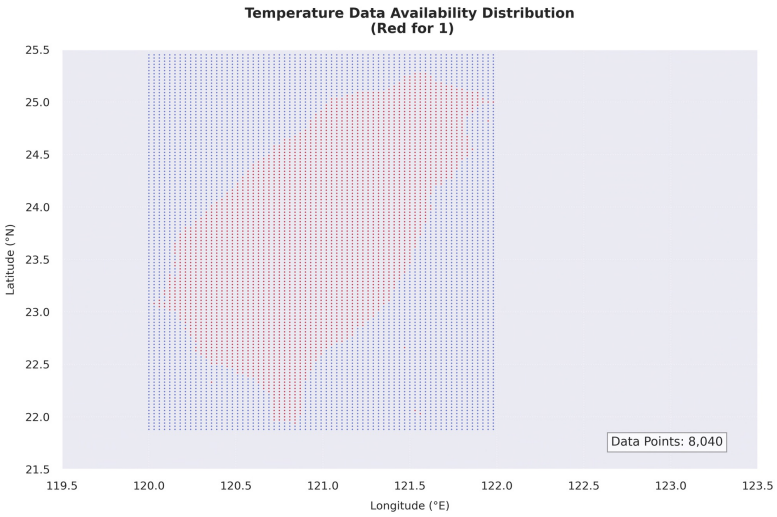
     - $\nu$：學習率 (程式碼中 `learning_rate` 參數)
     - $h_m$：第m棵樹
2. **梯度下降優化**
   - 計算當前模型的 **負梯度（偽殘差）**：

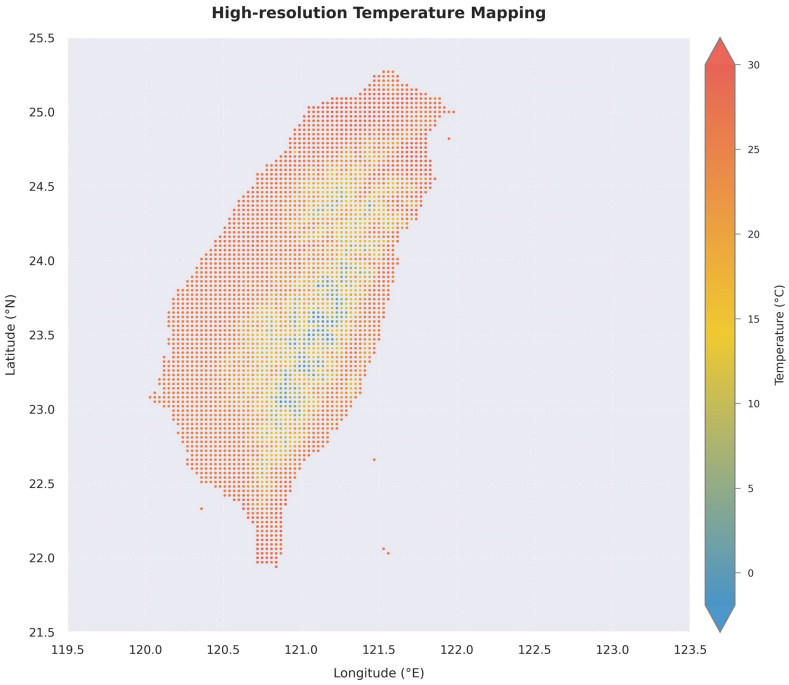
$$r_{im} = -\left[ \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F=F_{m-1}}$$

     - 對於平方誤差損失：$r_{im} = y_i - F_{m-1}(x_i)$
3. **樹擬合殘差**
   - 訓練新樹 $h_m$ 來擬合殘差：

$$h_m = \arg\min_h \sum_{i=1}^{n} (r_{im} - h(x_i))^2$$

# 真實數據可視化




# 預測結果與真實數據對照