使用 Gaussian Discriminant Analysis（GDA）做二元分類

(discussion with ChatGPT)

(a) **實作細節**

1. **數值穩定性**
   - 計算 Σ^{-1}時可能遇到奇異矩陣，故加小量對角正則化 reg（例如 1e-6），並使用 np.linalg.pinv 作備援。
2. **MLE 實作**
   - ϕ直接用 y.mean()。
   - μ_k用類內平均。
   - Σ用每個樣本相對其類別均值的 outer product 累加再除以總樣本數 n。
3. **輸出**
   - 模型包含 phi, mu0, mu1, sigma, sigma_inv, w, b。
   - 預測使用 sigmoid(wˆT x + b)，閾值預設 0.5。
4. **評估**
   - 切分資料（預設 70/30），以測試集報告 accuracy、混淆矩陣、precision/recall/F1。

(b) **方法與理論(By ChatGPT provide)**

1. **模型假設**

   假設資料生成過程為條件常態分布：對於類別 $y = k$（$k \in \{0,1\}$），特徵向量 x服從多變量常態分布

$$p(x \mid y = k) = \mathcal{N}(x; \mu_k, \Sigma).$$

此處假定兩類共享同一共變異數矩陣 Σ。

2. **參數與估計（MLE）**

- $\phi = P(y = 1)$由訓練資料中 $y = 1$的頻率估計。
- μ_0, μ_1分別為類別 0 與 1 的樣本均值。
- 共享協方差矩陣 Σ由所有樣本關於其類別均值的散佈合併估計（MLE）。

3. **分類規則推導**

以貝氏定理比較後驗比 $p(y = 1 \mid x)/p(y = 0 \mid x)$，代入高斯條件機率並取對數後，如果 Σ相同，二次項會抵消，剩下線性形式：

$$w^\top x + b(\text{logit})$$

其中

$$w = \Sigma^{-1}(\mu_1 - \mu_0), b = -\frac{1}{2}\mu_1^\top \Sigma^{-1}\mu_1 + \frac{1}{2}\mu_0^\top \Sigma^{-1}\mu_0 + \log\frac{\phi}{1-\phi}.$$

以 logistic sigmoid 將 logit 轉機率，閾值 0.5 做分類。因為推導是解析解，GDA 可直接得出線性決策邊界。

4. **為何適合本資料集**

- 本分類任務以經緯度為特徵，目標是判定格點是否有有效溫度值。若有效/無效格點在空間上呈現分區（例如某些區域缺數值），則條件分布近似常態或類內變異相似時，GDA 的線性邊界易於解釋且表現良好。若類內變異明顯不同或分布高度非高斯，GDA 仍可提供基線並直觀顯示決策邊界。

(c) **訓練與評估流程**

1. **資料**：載入 classification_dataset.csv。特徵採 longitude 與 latitude，標籤為 is_valid。
2. **切分**：70% 訓練，30% 測試，stratify=y 保持類別比例。
3. **訓練**：在訓練集上計算 $\phi, \mu_0, \mu_1, \Sigma$，得到 w, b。
4. **測試**：在測試集計算分類結果，報告：
   - 測試準確度（accuracy）。
   - 混淆矩陣（rows = 真實，cols = 預測）。

- 分類報表（precision, recall, F1）。

5. **結果輸出檔案**：程式會將決策邊界圖存為 gda_decision_boundary.png，並把混淆矩陣與分類報告存為 gda_results.txt。把這兩個檔案附在報告附錄。

注意：若需報告中呈現 cross-validation，將 train_test_split 改為 k-fold（例如 sklearn 的 KFold）並對每個 fold 計算 mean/std。此處以單次訓練/測試切分作業指派要求，並在報告中明確指出採用的方法與分割比例。

(d)