Week 4 – 溫度格點資料：資料轉換、分類與回歸

> 檔案： `hw4_programming_assignment.ipynb`
> 資料來源：CWA `O-A0038-003.xml` ，格點大小 **120×67**（緯向×經向），解析度 **0.03°**，左下角 `(120.00E, 21.88N)` 。

1. 問題設定與資料轉換

1.1 任務

- **分類 (classification)**：給定 `(經度, 緯度)` ，預測該格點溫度是否為**有效值**（label=1）或**無效值**（label=0）。規則：溫度 `-999.0` 視為無效，其餘為有效。
- **回歸 (regression)**：在**有效格點**上，預測攝氏溫度（連續值）。

1.2 讀檔與資料轉換

- 僅解析 XML 的 `<Content>...</Content>` ，以正則表達式擷取所有浮點數（含科學記號，如 `-999.0E+00` ），重塑為 **(H=120, W=67)** 的矩陣 `grid` 。
- 由題目提供的左下角座標與解析度，建立每個格點對應的 `(lon, lat)` 。
- 依規則產生兩個資料集：
  - **分類**： `(longitude, latitude, label)` ， `label = 1` 若 `grid != -999.0` ，否則 `0` 。
  - **回歸**： `(longitude, latitude, value)` ，僅保留 `grid != -999.0` 的筆數。

資料量：

- 分類：**8040** 筆（有效 **3495**、無效 **4545**）
- 回歸：**3495** 筆
- 皆匯出 `dataset_classification.csv` 、 `dataset_regression.csv` 。

2. 模型與訓練流程

2.1 模型

- **特徵**： `(lon, lat)` 兩個數值特徵。

- **分類模型 (MLP)**
  結構：
  - Linear **2 → 8**
  - ReLU
  - Linear **8 → 16**
  - ReLU
  - Linear **16 → 8**
  - ReLU
  - Linear **8 → 4**
  - ReLU
  - Linear **4 → 1**

  Loss： `BCEWithLogitsLoss` （推論用門檻 0.5）
  Optimizer：Adam(lr=1e-3)、Batch size：64、Epoch：80
- **回歸模型 (MLP)**
  結構：
  - Linear **2 → 8**
  - ReLU
  - Linear **8 → 16**
  - ReLU
  - Linear **16 → 8**
  - ReLU
  - Linear **8 → 4**
  - ReLU
  - Linear **4 → 1**

    Loss： `MSELoss`
    Optimizer：Adam(lr=1e-3, weight_decay=0.01)、Batch size：64、Epoch：300

3. 訓練過程與結果

3.1 分類結果

- `label = 1` 的比例 `3495/8040 ≈ 0.435` ； `label = 0` 之比例 ≈ **0.565**。
- 訓練過程（節選）：

```
[Classification] epoch 40  loss=0.7810  accuracy=0.6327
[Classification] epoch 60  loss=0.7782  accuracy=0.7129
[Classification] epoch 75  loss=0.7740  accuracy=0.7184
[Classification] epoch 100 loss=0.7624 accuracy=0.7197
[Classification] epoch 105 loss=0.7592 accuracy=0.7133
[Classification] epoch 110 loss=0.7558 accuracy=0.7119
[Classification] epoch 115 loss=0.7516 accuracy=0.7139
[Classification] epoch 120 loss=0.7474 accuracy=0.7205
[Classification] epoch 125 loss=0.7426 accuracy=0.7090
[Classification] epoch 130 loss=0.7375 accuracy=0.7214
[Classification] epoch 135 loss=0.7326 accuracy=0.7113
[Classification] epoch 140 loss=0.7278 accuracy=0.7163
[Classification] epoch 145 loss=0.7207 accuracy=0.7208
[Classification] epoch 150 loss=0.7153 accuracy=0.7235
```

- **解讀**：明顯高於基線，說明模型已學到空間位置與有效/無效的關聯。未切驗證的情況下，真實泛化表現可能略低。

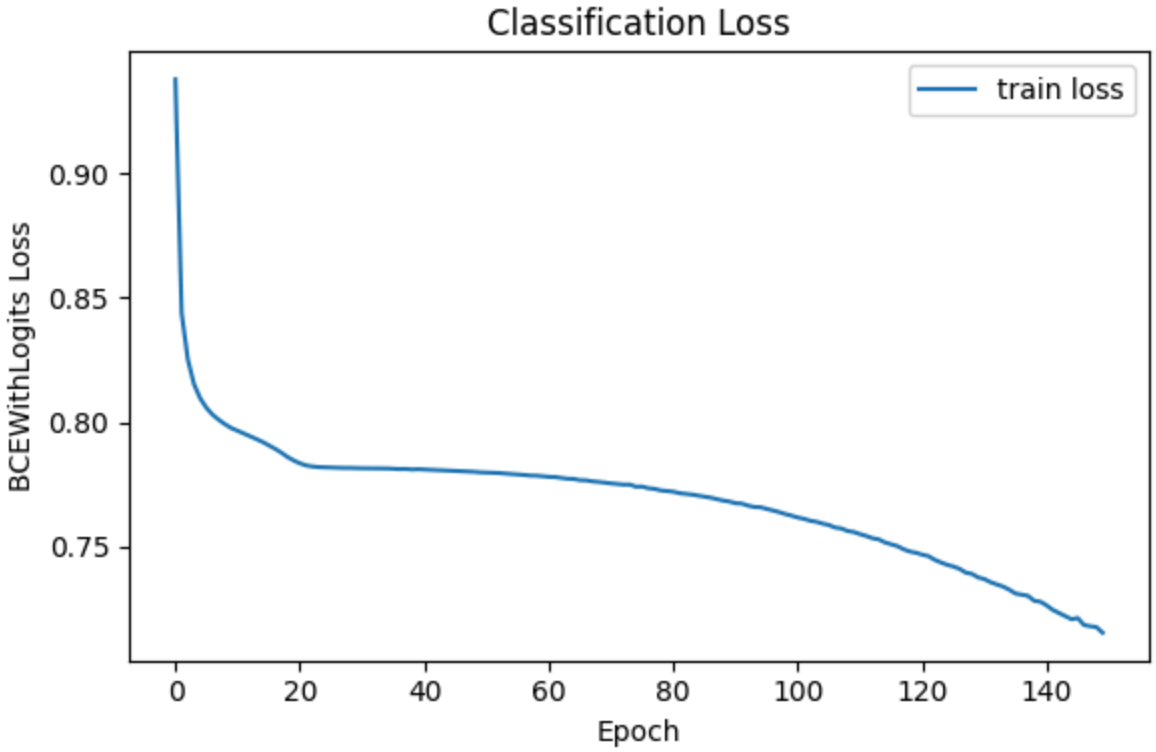
3.2 回歸結果

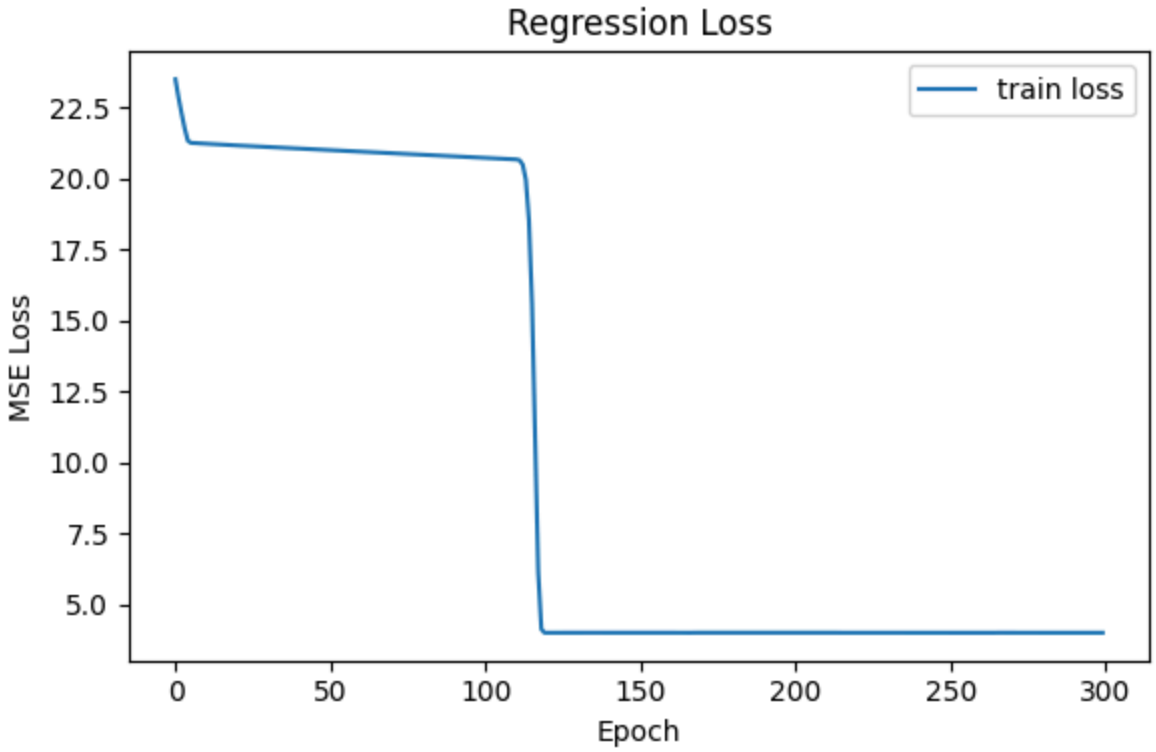
- 訓練過程（節選）：

```
[Regression] epoch 300  loss=4.0164  RMSE=6.6934
```

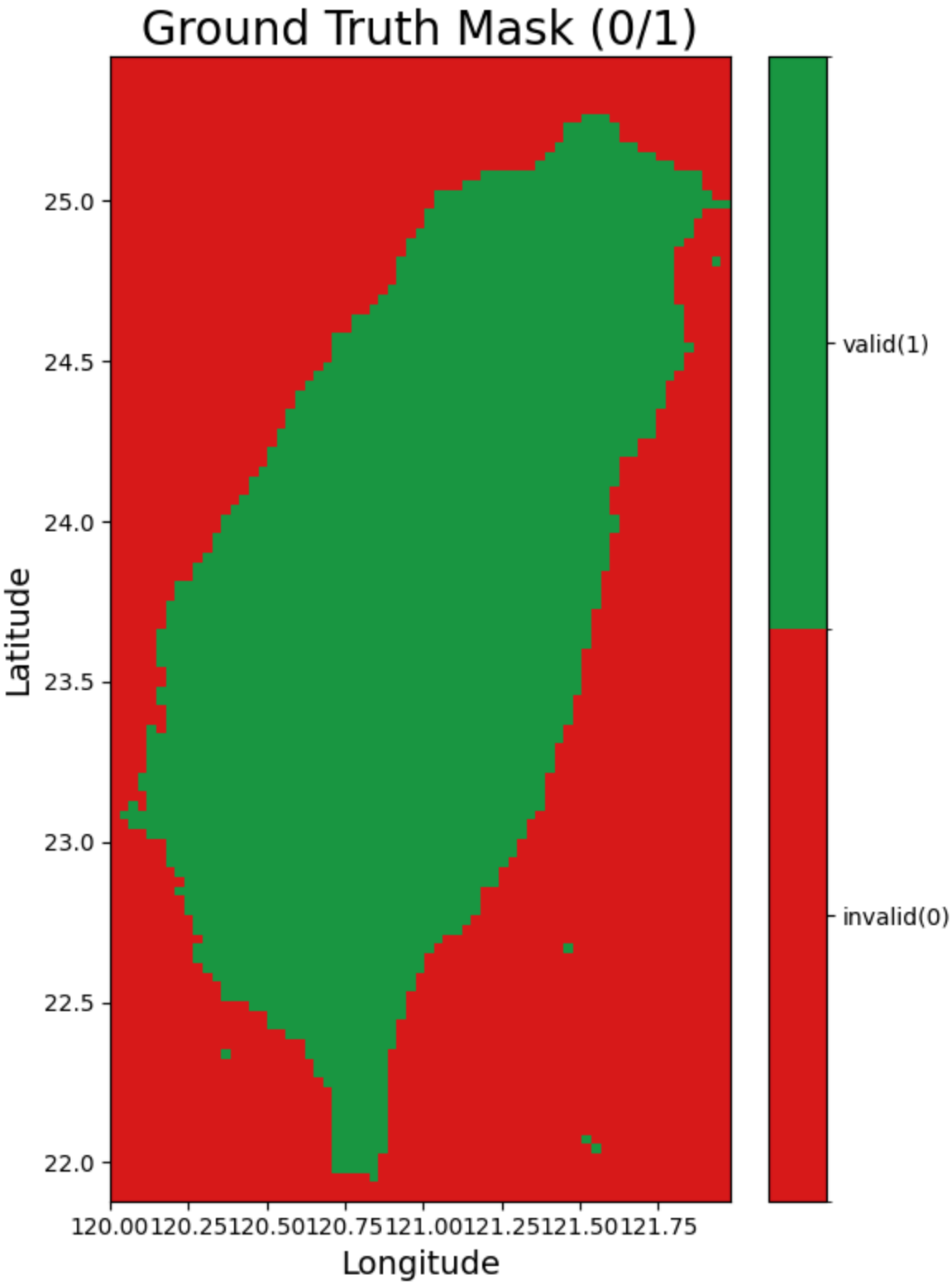
- **基線**：有效格點真實溫度的標準差約 **6.14°C**。以「全域平均值預測器」為基線，RMSE ≈ 此標準差。
- **解讀**：目前模型 **RMSE ≈ 6.69°C**，略差於基線，推論圖呈現接近常數，尚未學到足夠的空間變化。

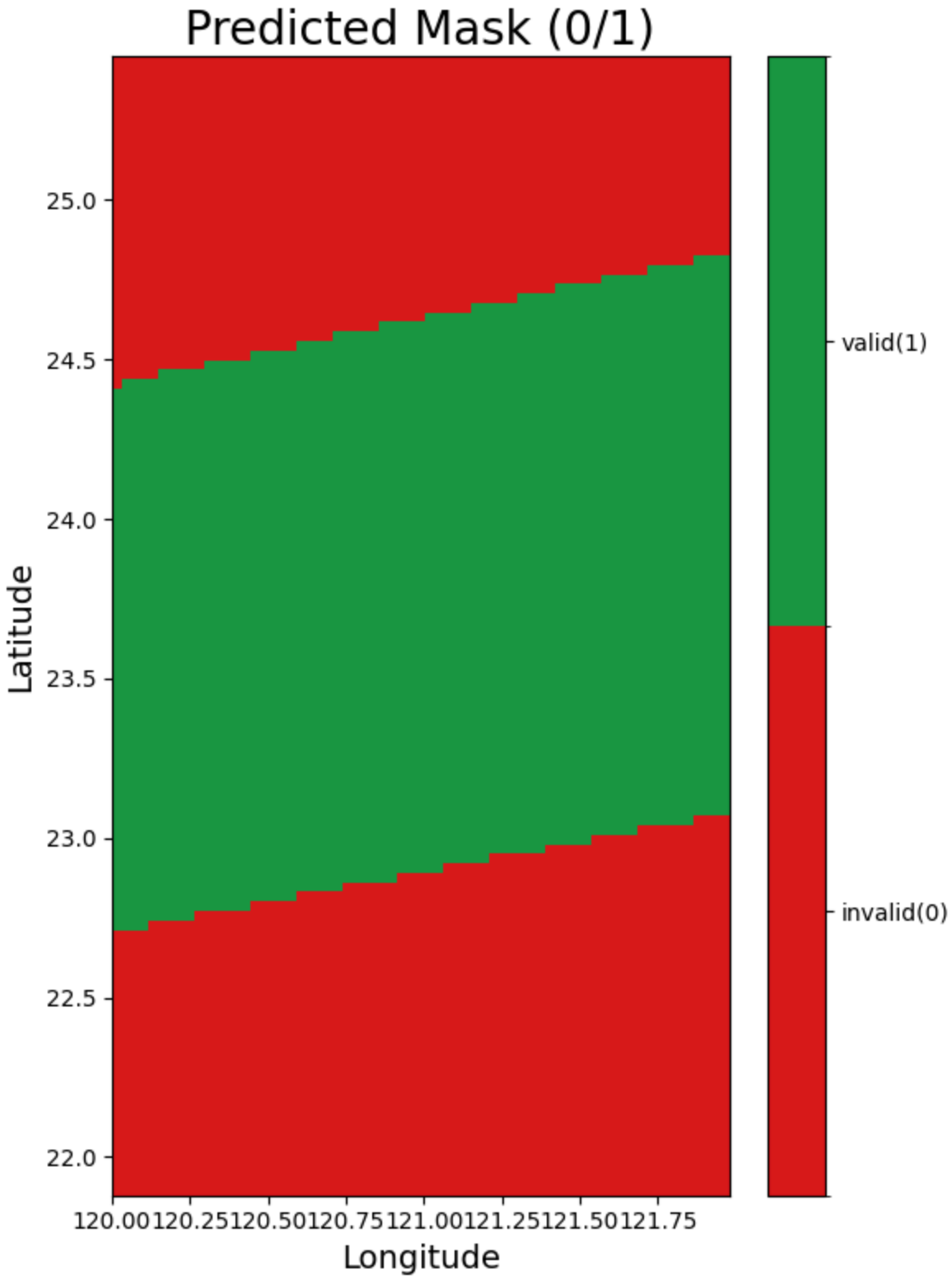
3.3 視覺化



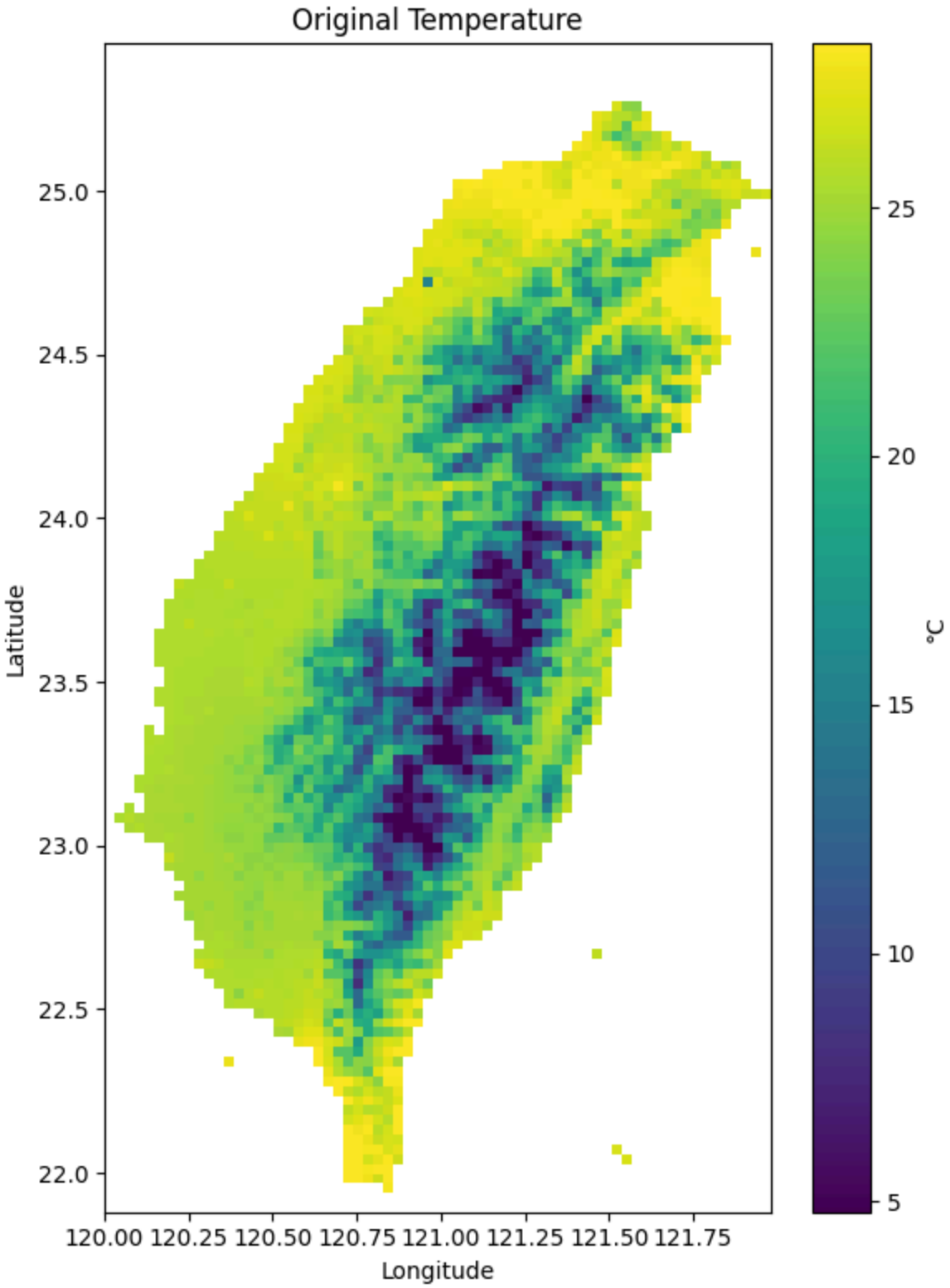


- **Classification Loss Plot and Regression Loss Plot**

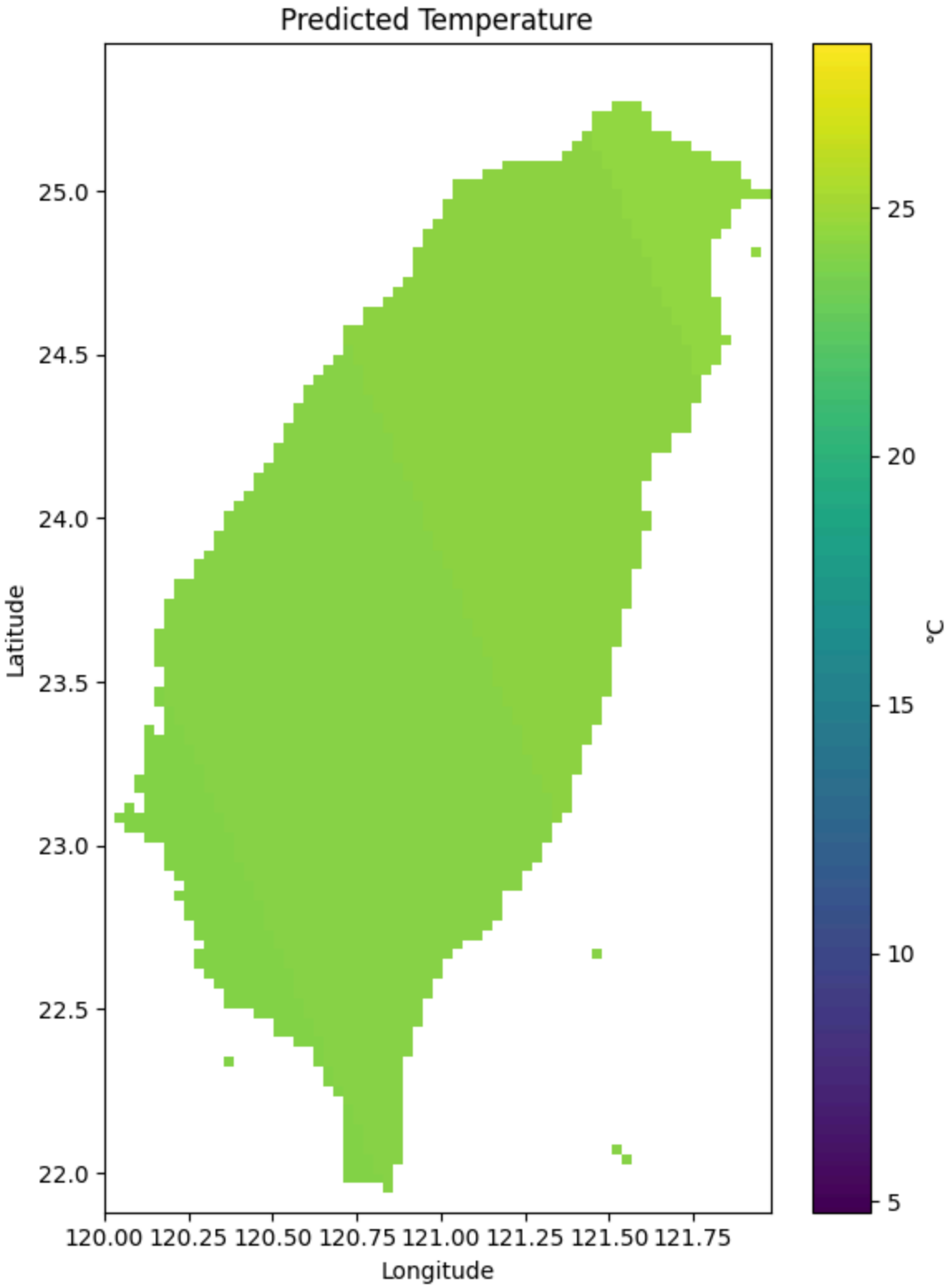

Ground Truth Mask (0/1)
Latitude
25.0
24.5
24.0
23.5
23.0
22.5
22.0
120.00
120.25
120.50
120.75
121.00
121.25
121.50
121.75
Longitude
valid(1)
invalid(0)



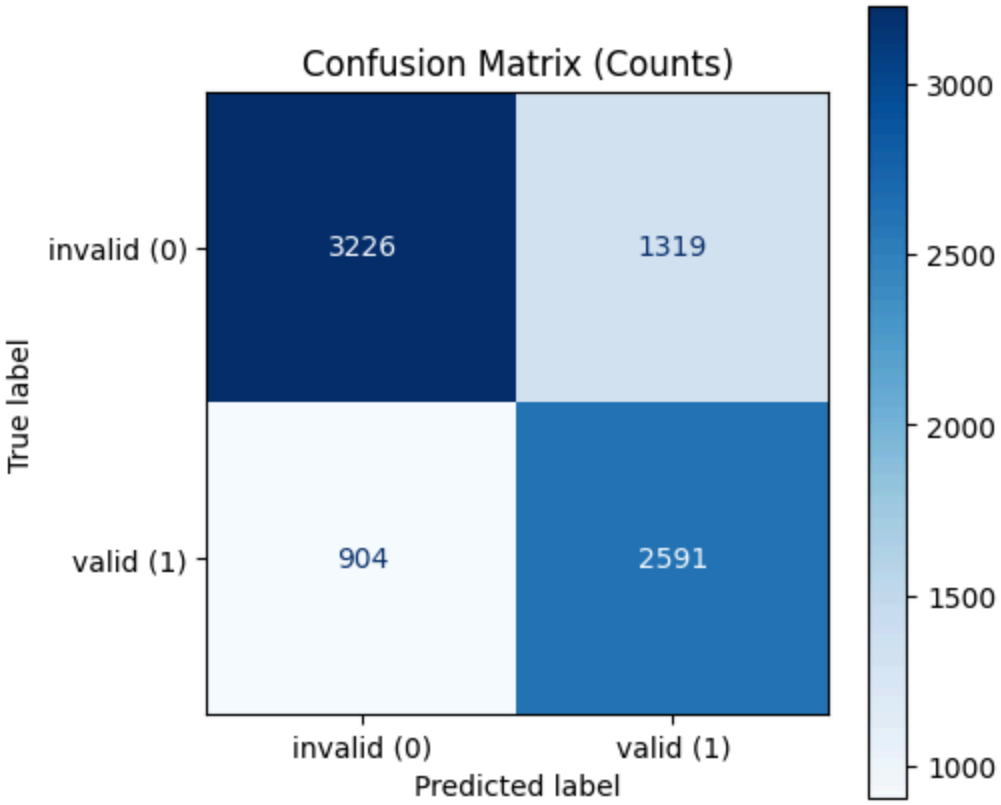
- **Classification (0/1 全圖)**：清楚區分陸域（多為 1）與海面（多為 0），下面為預測出來的結果。

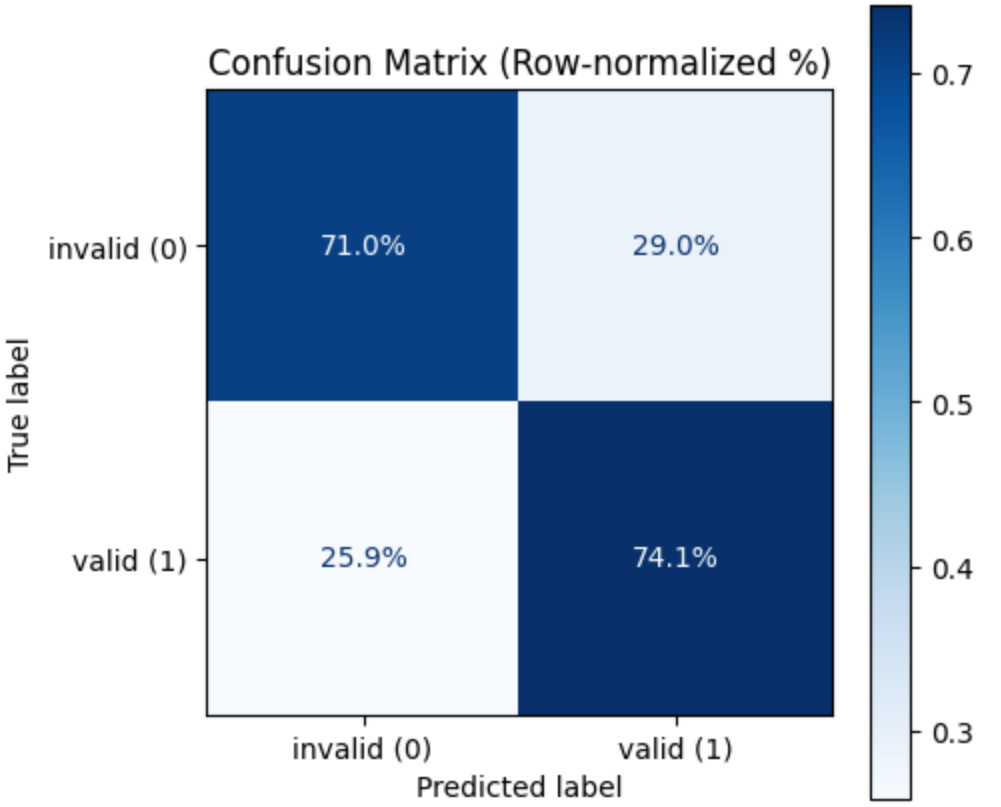


- **Original Temperature（valid only）**：呈現山區偏低、平原偏高的合理結構。



- **Predicted Temperature（valid only）**：近似單色，與真值差異大。





- **Confusion Matrix**：使用 `sklearn.metrics.ConfusionMatrixDisplay` 繪製 counts 與 row-normalized 兩張。

4. 討論與改進

4.1 分類

- **做法合理**： `BCEWithLogitsLoss` ；資料不嚴重失衡，無需特殊加權。
- **建議**：
  i. 切獨立 validation，回報 `Accuracy/Precision/Recall/F1` 與混淆矩陣數字；
  ii. 在驗證集掃描決策閾值（非 0.5）可提升 F1；
  iii. 加入鄰格統計（3×3 均值/方差）或地形高度等特徵，通常能再提升 3–8 個百分點。
  iv. 使用多日相同時間的溫度資料，一部份作為訓練集，另一部份作為驗證集，最後用來當作測試集。

4.2 回歸

- **小改動、效果大**：
  - 換 **Huber** 損失（ `SmoothL1Loss(beta=1.0)` ）以提升對極端值的韌性；
  - 調整學習率至 `2e-3 ~ 3e-3` ，Epoch ≥ 150；
  - **Fourier features**：在 `(lon, lat)` 上加 `sin/cos(γ·x)` （如 m=8, scale≈3）強化空間表達力，預測圖會顯著擺脫單色。

5. 結論

- 已完成**資料轉換、分類與回歸**兩個任務的實作與圖表。
- **分類**優於多數類基線（~0.72 vs 0.565），能有效辨識有效/無效格點。
- **回歸**仍低於基線（RMSE 6.69°C > ~6.14°C），需要依建議調整（Huber、標準化、Fourier features與更長訓練）。
- 後續若依建議微調，預期 RMSE 可顯著下降，視覺上的細節也會更貼近真值。