Machine Learning

Assignment 4

CHANG Yung-Hsuan (張永瑨)
111652004
eiken.sc11@nycu.edu.tw

September 30, 2025

1. There are unanswered questions from the lecture, and there are likely more questions we haven't covered. Take a moment to think about these questions. Write down the ones you find important, confusing, or interesting.

   Answer. The MNIST dataset is mentioned during the lecture. It is know that there is CNN to solve this kind of problem of recognizing images (or classifying images). My idea is naive, but I wonder if we consider those images points in a space (e.g., in $\mathbb{R}^{28\times 28}$) and cluster them, does this clustering do the same with classification?

2. Using the dataset from CODiS, transform the original data to two datasets for supervised learning:

   a. Classification: `(longitude, latitude, label)`, where `label` $= 1$ if the value is valid or $= 0$ if invalid (it would be -999).

   b. Regression: `(longitude, latitude, value)`, where `value` is valid.

   In addition, use both of your datasets to train a machine learning model separately:

   a. Classification Model: Use `(longitude, latitude)` to predict whether the data point is valid (denoted with `1`) or invalid (denoted with `0`).

   b. Regression Model: Use `(longitude, latitude)` to predict corresponding observed temperature.

   Solution. I choose to report decision tree for classification model and neural network for regression.

   I. **Data and Maps**

- Grid size: 67 × 120 with 0.03° resolution (`lon` increments within row, `lat` by rows).
- Invalid value: `-999`.
- Two datasets:
  - Classification: `(longitude, latitude, label)`, where `label` $\in \{0, 1\}$.
  - Regression: `(longitude, latitude, value)`, where invalids are removed.

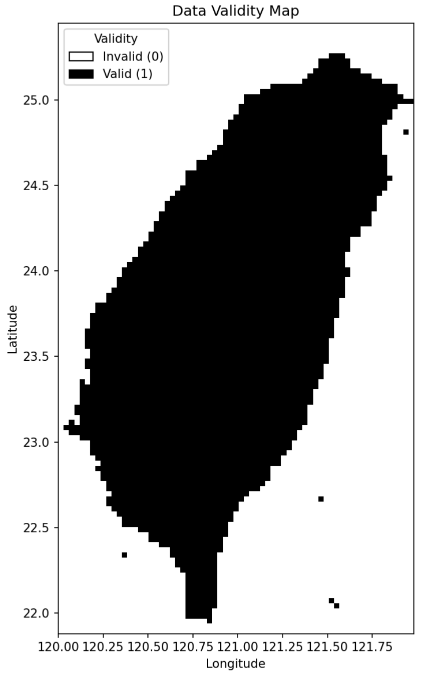


Figure 1. Data Validity Map (1 = valid, 0 = invalid)

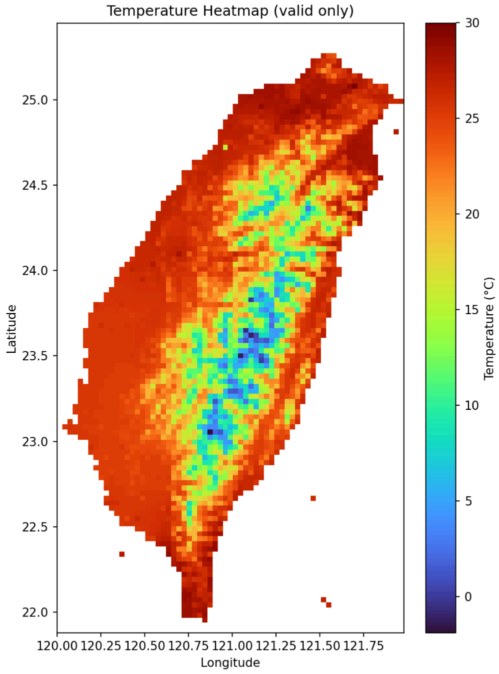


Figure 2. Temperature Heatmap (valid cells only)

II. **Models**

Classification. I use decision tree on features `(lon, lat)`.

Rationale: trees capture non-linear coastline/terrain boundaries without feature engineering. Training split: 70/15/15 (train/val/test) with class balancing; thresholding is not required for the tree.

Table 1. Test Metrics on Classification (Decision Tree)

Metric	Accuracy	F1	Precision	Recall
Value	0.9511	0.9457	0.9130	0.9809

High recall (0.9809) indicates the model rarely misses valid cells; precision (0.9130) reflects a small amount of over-flagging, which is acceptable when the cost of discarding a valid cell is high.

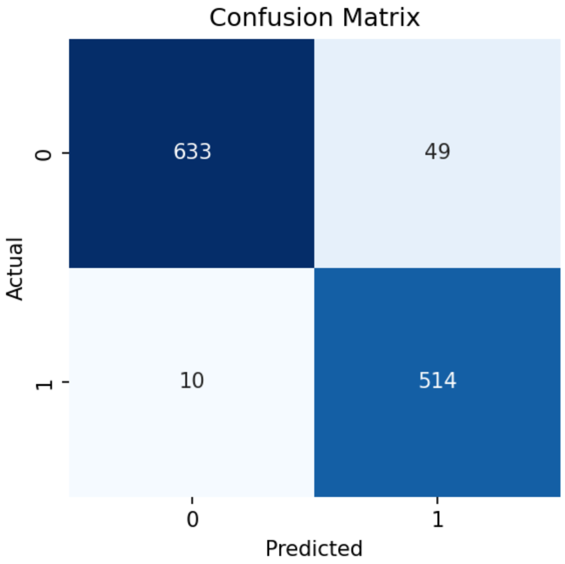
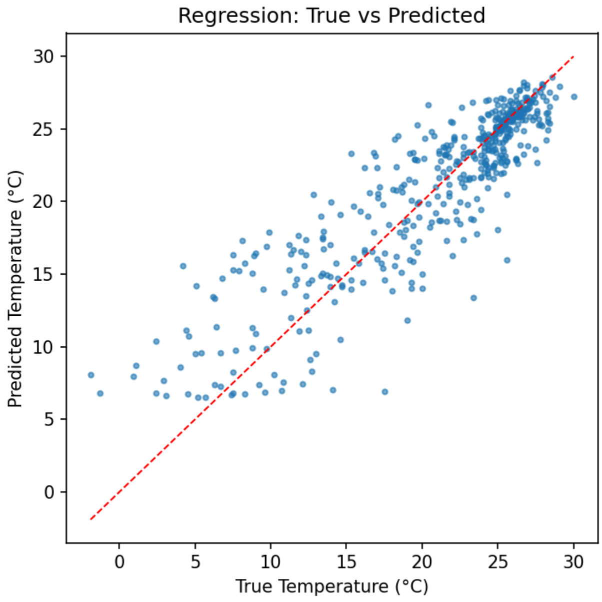


Figure 3. Confusion Matrix (Decision Tree Testing Set)

Regression. I use a two-hidden-layer neural network (NN) with sigmoid activations on standardized `(lon, lat)`, trained with optimizer Adam and early stopping (validation loss).

Table 2. Test Metrics on Regression (Neural Network)

MAE (°C)	RMSE (°C)	R^2
1.977	2.918	0.7928



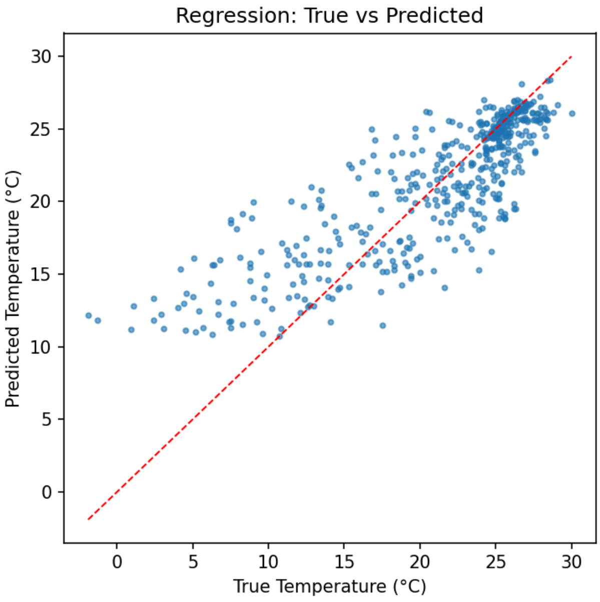


Figure 4. Regression (NN): True vs. Predicted

Figure 5. Regression (GMM): True vs. Predicted

The cloud is tight along $x = y$ at mid-to-high temperatures (see Figure 4); residual spread is larger at low temperatures, consistent with sparse valid points in cooler/high-elevation areas.

I also tried a conditional the Gaussian mixture model (GMM) on `(lon, lat, value)` with $\mathbb{E}(y \mid x)$ prediction. It works but is not ideal here: the spatial field is smoother than a small mixture can capture without overfitting. Compare Figure 4 and Figure 5.

Regression errors grow in areas with fewer valid neighbors (data sparsity) and at distribution tails. Decision trees are a strong baseline for binary validity, giving high recall with minimal tuning. A compact neural network on standardized coordinates captures smooth spatial temperature gradients effectively.

III. **Summary**

For validity, a decision tree model with basic class balancing is sufficient and interpretable. For temperature: a small neural network offers the best bias–variance trade-off among tried methods on this grid.

Please see my GitHub repository `temperature.py` for full code.