Week 3 Worksheet — Grid Search & Error Surface

Health Mini-Project (Linear Model with 4 Features; Train Focus)

Name: ______________________ **Date:** ____________________

Story Setup — If We Were the First Mathematicians to Invent ML...
In Weeks 1–2, we chose rules by hand. Today we let the computer try many combinations and *find a rule* that makes the **error** as small as possible. This is our first concrete step toward "learning."

Goals of This Week

- Understand and compute an **error function** using **accuracy**.
- Use **for-loops** to perform a **grid search** over weights and a threshold.
- Draw a **2D error surface** (heat/contour plot) for selected parameters.

Dataset (Train: IDs 1–10; Test Preview: 11–12)
We use the same four inputs as Week 1. The hidden ground-truth label is provided as `true_label` for training.

ID	Height (in)	Weight (lb)	Waist (in)	Favorite Color	true_label
1	70	159	32	Blue	Healthy
2	65	187	37	Red	Unhealthy
3	72	198	39	Green	Unhealthy
4	63	121	28	Yellow	Healthy
5	68	150	31	Black	Healthy
6	67	209	41	Blue	Unhealthy
7	71	172	33	Pink	Healthy
8	62	181	36	Purple	Unhealthy
9	69	146	31	Orange	Healthy
10	73	231	43	White	Unhealthy
11	64	128	28	Green	Healthy
12	70	154	33	Red	Healthy

Note: IDs 11–12 are held out for testing (next week). This week we focus on training only.

Task A — Normalize Inputs (fit on Train only)

1. Convert **Favorite Color** to a numeric code (any fixed mapping; document your mapping).
2. For the **training set (IDs 1–10)** compute min–max normalization for each feature:

$$x' = \frac{x - \min(x_{\text{train}})}{\max(x_{\text{train}}) - \min(x_{\text{train}})} \in [0, 1].$$

3. Apply the same train-fitted min–max to IDs 11–12 (do *not* use them for training today).

Task B — Linear Model (same as Week 1)

$$\text{Score} = w_1 H + w_2 W + w_3 X + w_4 C, \qquad w_i \geq 0, \;\; w_1 + w_2 + w_3 + w_4 = 1,$$

where H, W, X, C are the *normalized* Height, Weight, Waist, and (encoded) Color.
Classification rule with threshold T:

$$\text{If Score} < T \Rightarrow \text{Healthy}, \quad \text{else Unhealthy.}$$

Task C — Error Function (Accuracy)

$$\text{Accuracy} = \frac{\#\{\text{correct on train}\}}{10}, \qquad \text{Error} = 1 - \text{Accuracy}.$$

We will maximize accuracy (equivalently, minimize error) on IDs 1–10.

Task D — Grid Search with For-Loops (Train Only)

1. Search w_1, w_2, w_3, w_4 on a grid (e.g., step 0.1) with $w_1 + w_2 + w_3 + w_4 = 1$.
2. Search T on a grid (e.g., 0.30 to 0.70 by 0.05).
3. For each $(w_1, \ldots, w_4, T)$ on IDs 1–10:
   (a) Compute Score (using normalized inputs).
   (b) Predict labels with the rule above.
   (c) Compute **Accuracy**; keep the best combination (highest accuracy).

Task E — Draw a 2D Error Surface

- Fix three parameters and vary two (e.g., plot Error as a function of (w_1, T)).
- Draw a *heatmap* or *contour plot*. Mark the best point you found.

Reflection

- On your 2D error plot, does there appear to be **one clear minimum** or **several comparable regions**?

- How does the error surface change when you switch which two parameters you visualize (e.g., (w_1, T) vs. (w_2, T))?
- Why do we **fit normalization on train only**? What could go wrong if we used test data to fit normalization?

Deep Questions & Next Week Preview

1. If the grid step is 0.01, roughly how many weight combinations exist for (w_1, w_2, w_3, w_4) with $w_1 + \cdots + w_4 = 1$? (Order-of-magnitude: think $\sim 100^4$.)
2. Do error surfaces always have a **single global minimum**, or can there be **multiple local minima**?
3. Is there a **smarter way** than checking every grid point to move toward low error? *(Next week: Gradient Descent — "slide downhill" on the error surface.)*

Turn-in Checklist

Your normalization procedure and color encoding (brief).

Best (w_1, w_2, w_3, w_4, T) on **train (IDs 1–10)** and the accuracy value.

A 2D error plot (heatmap/contour) with the best point marked.

Short answers to Reflection & Deep Questions.