We choose a stump that minimizes weighted error w.r.t. the current weights.

Checking candidate thresholds, the stump that works best is:

$M_2(x) = +1$ if $x \leqslant 5.5$, else -1.

Predictions of M_2:

- x=1 → +1 (correct)
- x=2 → +1 (correct)
- x=3 → +1 (**incorrect**, true −1)
- x=4 → +1 (**incorrect**, true −1)
- x=5 → +1 (correct)
- x=6 → −1 (correct)

Misclassified: tuples 3 and 4.

Compute error of M_2:

$$\text{error}_2 = w_3^{(2)} + w_4^{(2)} = 0.1 + 0.1 = 0.2.$$

Classifier weight

$$\alpha_2 = \ln\left(\frac{1 - \text{error}_2}{\text{error}_2}\right) = \ln\left(\frac{0.8}{0.2}\right) = \ln(4) \approx 1.3862943611.$$

Update weights (correctly classified multiplied by $\frac{\text{error}_2}{1 - \text{error}_2}$):

Factor $= \frac{0.2}{0.8} = \frac{1}{4} = 0.25.$

Unnormalized after applying factor:

- Correct (i=1,2,5,6): multiply by 0.25:
  - i=1: $0.1 \times 0.25 = 0.025$
  - i=2: 0.025
  - i=5: $0.5 \times 0.25 = 0.125$
  - i=6: 0.025
- Misclassified (i=3,4): remain 0.1 each.

Sum unnormalized = $0.025 + 0.025 + 0.1 + 0.1 + 0.125 + 0.025 = 0.4.$

Normalize by dividing by 0.4 (multiply by 2.5):

Final weights after round 2:

- $w_1^{(3)} = 0.025/0.4 = 0.0625 = \frac{1}{16}$.
- $w_2^{(3)} = 0.0625 = \frac{1}{16}$.
- $w_3^{(3)} = 0.1/0.4 = 0.25 = \frac{1}{4}$.
- $w_4^{(3)} = 0.25 = \frac{1}{4}$.
- $w_5^{(3)} = 0.125/0.4 = 0.3125 = \frac{5}{16}$.
- $w_6^{(3)} = 0.0625 = \frac{1}{16}$.

So as fractions:

$$w^{(3)} = [\frac{1}{16}, \frac{1}{16}, \frac{1}{4}, \frac{1}{4}, \frac{5}{16}, \frac{1}{16}].$$

Round 3 (i = 3)

Find best stump under current weights. The best stump we choose is:

$M_3(x) = -1$ if $x \leqslant 4.5$, else $+1$.
(This is the reversed orientation at threshold 4.5.)

Predictions of M_3:

- x=1 → −1 (incorrect; true +1)
- x=2 → −1 (incorrect; true +1)
- x=3 → −1 (correct)
- x=4 → −1 (correct)
- x=5 → +1 (correct)
- x=6 → +1 (incorrect; true −1)

Misclassified: tuples 1, 2, 6.

Compute error of M_3:

$$\text{error}_3 = w_1^{(3)} + w_2^{(3)} + w_6^{(3)} = \frac{1}{16} + \frac{1}{16} + \frac{1}{16} = \frac{3}{16} = 0.1875.$$

Classifier weight

$$\alpha_3 = \ln(\frac{1 - \text{error}_3}{\text{error}_3}) = \ln(\frac{13/16}{3/16}) = \ln(\frac{13}{3}) \approx 1.4663370688.$$

Update weights (correctly classified multiplied by $\frac{\text{error}_3}{1 - \text{error}_3}$):
Factor = $\frac{3/16}{13/16} = \frac{3}{13} \approx 0.2307692308.$

Unnormalized after applying factor:

- Correct (i=3,4,5): multiply their weights by $3/13$:
  - w3: $\frac{1}{4} \times \frac{3}{13} = \frac{3}{52}$
  - w4: $\frac{1}{4} \times \frac{3}{13} = \frac{3}{52}$
  - w5: $\frac{5}{16} \times \frac{3}{13} = \frac{15}{208}$
- Misclassified (i=1,2,6): stay $\frac{1}{16}$ each.

Compute numeric unnormalized values:

- $\frac{3}{52} \approx 0.05769230769$
- $\frac{15}{208} \approx 0.07211538462$
- misclassified each: $\frac{1}{16} = 0.0625$

Sum unnormalized:

$$3 \text{ misclass}: \; 3 \times \frac{1}{16} = \frac{3}{16} = 0.1875$$

$$\text{plus } \frac{3}{52} + \frac{3}{52} + \frac{15}{208} = \frac{3}{52} + \frac{3}{52} + \frac{15}{208} = \frac{12}{208} + \frac{12}{208} + \frac{15}{208} = \frac{39}{208} \approx 0.1875$$

So total unnormalized sum = $0.1875 + 0.1875 = 0.3750.$

Normalize (divide each by 0.375 = 3/8) to get final example weights $w^{(4)}$. The normalized weights simplify to rational fractions; final weights are:

$$w^{(4)} = [\tfrac{13}{78}, \; \tfrac{13}{78}, \; \tfrac{12}{78}, \; \tfrac{12}{78}, \; \tfrac{15}{78}, \; \tfrac{13}{78}],$$

which in decimals is approximately

$$[0.1666667, \; 0.1666667, \; 0.1538462, \; 0.1538462, \; 0.1923077, \; 0.1666667].$$

(We won’t need these for classification — they show how weights evolve.)

Build the final ensemble classifier

We have three weak learners M_1, M_2, M_3 with weights (votes)

$$\alpha_1 = \ln(5) \approx 1.6094379124, \quad \alpha_2 = \ln(4) \approx 1.3862943611, \quad \alpha_3 = \ln(\frac{13}{3}) \approx 1.46$$

To classify any x we compute the weighted vote:

$$V(x) = \sum_{t=1} \alpha_t M_t(x),$$

then output $\hat{y} = \text{sign}(V(x))$ (if $V = 0$ you can break ties arbitrarily).

Ensemble predictions on the training set (showing arithmetic)

Compute $V(x_i)$ for each training tuple:

i = 1 (x=1)

- $M_1(1) = +1 \to \alpha_1 \cdot (+1) = 1.6094379124$
- $M_2(1) = +1 \to \alpha_2 \cdot (+1) = 1.3862943611$
- $M_3(1) = -1 \to \alpha_3 \cdot (-1) = -1.4663370688$

  Sum: $1.6094379124 + 1.3862943611 - 1.4663370688 = 1.5293952048.$

  Sign = +1 → predicted +1 (correct).

i = 2 (x=2) — same pattern as i=1, so sum = 1.5293952048 → predicted +1 (correct).

i = 3 (x=3)

- $M_1(3) = -1 \to -1.6094379124$
- $M_2(3) = +1 \to +1.3862943611$
- $M_3(3) = -1 \to -1.4663370688$

  Sum: $-1.6094379124 + 1.3862943611 - 1.4663370688 = -1.6894806201$

  Sign = −1 → predicted −1 (correct).

i = 4 (x=4) — same sum as i=3 → −1 (correct).

i = 5 (x=5)

- $M_1(5) = -1 \to -1.6094379124$
- $M_2(5) = +1 \to +1.3862943611$
- $M_3(5) = +1 \to +1.4663370688$

  Sum: $-1.6094379124 + 1.3862943611 + 1.4663370688 = 1.2431935175$

  Sign = +1 → predicted +1 (correct).

i = 6 (x=6)

- $M_1(6) = -1 \rightarrow -1.6094379124$
- $M_2(6) = -1 \rightarrow -1.3862943611$
- $M_3(6) = +1 \rightarrow +1.4663370688$

  Sum: $-1.6094379124 - 1.3862943611 + 1.4663370688 = -1.5293952048$

  Sign = −1 → predicted −1 (correct).

Training error: All 6 training points are classified correctly by the ensemble → **training error = 0 / 6 = 0.0**.

Summary

1. We started with equal weights $1/6$. After Round 1 the incorrectly classified point (index 5) received larger weight (0.5), focusing the next weak learner on that point.
2. Each weak learner M_t gets a vote weight $\alpha_t = \ln((1 - \text{error}_t)/\text{error}_t)$; smaller error → larger α_t.
   - $\alpha_1 = \ln(5) \approx 1.6094$
   - $\alpha_2 = \ln(4) \approx 1.3863$
   - $\alpha_3 = \ln(13/3) \approx 1.4663$
3. The final ensemble combines weak learners by weighted voting. In this example 3 weak stumps suffice to reach zero training error.
4. The arithmetic used here follows exactly the algorithm you posted (resampling/weights and the update factor $\frac{\text{error}}{1-\text{error}}$ for correctly classified tuples), and you can verify each numeric operation step-by-step from the fractions shown.