1 point

1. Recall that the **classification error for unweighted data** is defined as follows:

$$\text{classification error} = \frac{\#\text{ mistakes}}{\#\text{ all data points}}$$

Meanwhile, the **weight of mistakes for weighted data** is given by

$$\mathrm{WM}(\alpha, \hat{\mathbf{y}}) = \sum_{i=1}^{n} \alpha_i \times 1[y_i \neq \hat{y}_i].$$

If we set the weights **α=1** for all data points, how is the weight of mistakes **WM(α,ŷ)** related to the classification error?

- ( ) **WM(α,ŷ)** = [classification error]
- ( ) **WM(α,ŷ)** = [classification error] * [weight of correctly classified data points]
- (●) **WM(α,ŷ)** = N * [classification error]
- ( ) **WM(α,ŷ)** = 1 - [classification error]

---

1 point

2. Refer to section **Example: Training a weighted decision tree**.

Will you get the same model as **small_data_decision_tree_subset_20** if you trained a decision tree with only 20 data points from the set of points in **subset_20**?

- (●) Yes
- ( ) No

---

1 point

3. Refer to the 10-component ensemble of tree stumps trained with Adaboost.

As each component is trained sequentially, are the component weights monotonically decreasing, monotonically increasing, or neither?

- ( ) Monotonically decreasing
- ( ) Monotonically increasing
- (●) Neither

---

1 point

4. Which of the following best describes a **general trend in accuracy** as we add more and more components? Answer based on the 30 components learned so far.

- ( ) Training error goes down monotonically, i.e. the training error reduces with each iteration but never increases.
- (●) Training error goes down in general, with some ups and downs in the middle.
- ( ) Training error goes up in general, with some ups and downs in the middle.
- ( ) Training error goes down in the beginning, achieves the best error, and then goes up sharply.
- ( ) None of the above

---

1 point

5. From this plot (with 30 trees), is there massive overfitting as the # of iterations increases?

- ( ) Yes
- (●) No