CS246 Exam 2025 — Question 9: Decision Trees (14 points)

Part 1 (2 points) — Reducing Variance in Decision Trees

Question: Which methods reduce variance and improve generalization? Circle all that apply.

Answer: (b), (c)

- **(a) Dropout — No.** Dropout is a neural network technique; decision trees don't have neurons to drop.
- **(b) Restricting tree growth — Yes.** This is **early stopping** — requiring a minimum number of examples per leaf prevents memorizing noise.
- **(c) Ensemble learning — Yes.** Random Forests and Boosting combine multiple trees; averaging cancels out individual errors.
- **(d) Random feature subsets — No.** A single tree built with a random feature subset doesn't reliably reduce variance on its own. Feature bagging only helps as part of an ensemble (Random Forests), which is already covered by (c).

Part 2 (3 points) — Decision Boundary Identification

(a) Plot (A): i. Decision Tree

Simple, axis-aligned rectangular regions are the hallmark of a single decision tree. Ensemble methods *could* produce simple boundaries, but the clean rectangular partition is most characteristic of a single tree.

(b) Plot (B): ii. Random Forest, iii. AdaBoost

A more complex but still axis-aligned boundary, smoother than (A). Ensembles produce smoother boundaries by aggregating many axis-aligned splits. Random Forests average many trees; AdaBoost combines many stumps. A single Decision Tree would produce sharper, more jagged rectangular regions rather than this refined boundary.

(c) Plot (C): iv. None of the Above

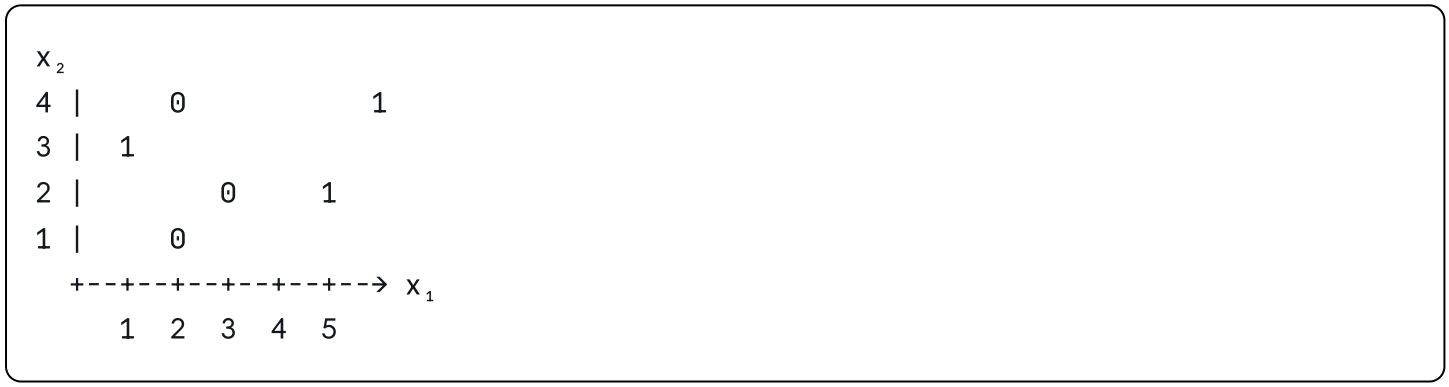
Plot (C) has a smooth, **non-axis-aligned** (diagonal/curved) decision boundary. All tree-based methods — Decision Trees, Random Forests, and AdaBoost — produce axis-aligned splits ($x_1 > a$ or $x_2 > a$). While ensembles can approximate curves with many tiny rectangles, the smooth diagonal boundary in (C) is characteristic of a non-tree-based method (e.g., SVM, logistic regression, neural network).

Part 3 (6 points) — Best Split at Root Node

Dataset:

x_1	x_2	y
1	3	1
2	1	0
2	4	0
3	2	0
4	4	1
5	2	1

Visualizing the data (following the hint):



Positive points: (1,3), (4,4), (5,2). Negative points: (2,1), (2,4), (3,2). The two rightmost positives can be cleanly separated by a vertical line at $x_1 = 3.5$.

Parent entropy

3 positive, 3 negative: **H(Y) = 1 bit**

Testing all splits

Thresholds lie between consecutive distinct sorted values. We compute IG = H(Y) − weighted child entropies for each:

Split	Left	Right	IG (approx.)
$x_1 > 3.5$	{(1,3),(2,1),(2,4),(3,2)}: 1pos, 3neg	{(4,4),(5,2)}: 2pos, 0neg	**≈ 0.459**
$x_1 > 1.5$	{(1,3)}: 1pos, 0neg	rest: 2pos, 3neg	≈ 0.191
$x_1 > 4.5$	4 points: 2pos, 3neg	{(5,2)}: 1pos, 0neg	≈ 0.191
$x_2 > 1.5$	{(2,1)}: 0pos, 1neg	rest: 3pos, 2neg	≈ 0.191
$x_1 > 2.5$	3 points: 1pos, 2neg	3 points: 2pos, 1neg	≈ 0.082
$x_2 > 2.5$	3 points: 1pos, 2neg	3 points: 2pos, 1neg	≈ 0.082
$x_2 > 3.5$	4 points: 2pos, 2neg	2 points: 1pos, 1neg	0

The best split is **$x_1 > 3.5$** — the right side is pure positive (2 out of 2).

Part 3a — Answer

$$\boxed{i = 1, \quad \square =>, \quad a = 3.5}$$

(Any a in the range [3, 4) is equally correct — i.e., a ⩾ 3 and a < 4.)

Part 3b — Information Gain

The right child ($x_1 > 3.5$) has H = 0 (pure). The left child has 1 positive, 3 negatives:

$$IG = 1 - \frac{4}{6}\left[-\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4}\right] - \frac{2}{6}\cdot 0$$

$$\boxed{IG = 1 - \frac{2}{3}\left[-\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4}\right] \approx 0.459}$$

Part 4 (3 points) — AdaBoost vs. GBDT Facts

Answer: (a), (b), (c)

- **(a) True.** AdaBoost maintains sample weights; GBDT computes gradients of the loss function and fits trees to those gradients.
- **(b) True.** Core AdaBoost mechanism — misclassified examples get weights multiplied by $\exp(\alpha_t) > 1$, forcing the next stump to focus on harder examples.

- **(c) True.** GBDT (e.g., XGBoost) includes regularization in its objective: $\gamma \cdot T$ (penalizing number of leaves) + $\frac{1}{2}\lambda \cdot \Sigma w^2$ (L2 on leaf weights), plus a learning rate shrinkage parameter.
- **(d) False.** Confuses the two: **AdaBoost** uses decision stumps (1-level trees), while **GBDT** uses multi-level decision trees.