Extend your decision tree implementation to include a stochastic pruning strategy during tree growth. In this approach, a node may be **forcibly turned into a leaf node** based on a probability that increases with its depth in the tree. This is combined with feature bagging to maintain tree diversity. Modify your decision tree construction so that:

- At each node, compute a probability of stopping (i.e., not splitting further) that depends on the current node's depth d. Use the formula:

$$P_{\text{stop}}(d) = \min(1, \alpha \cdot d)$$

  where α is a tunable parameter (e.g., 0.05) passed as a command-line argument.

- With probability $P_{\text{stop}}(d)$, stop growing the tree at this node and make it a leaf, labeling it with the majority class of the current data subset.

- Otherwise, select $\lfloor \sqrt{D} \rfloor$ features uniformly at random (feature bagging) and perform the usual split using a splitting criterion.

Clarification

- The stopping probability increases as the tree goes deeper. For example, if $\alpha = 0.05$, then:

$$P_{\text{stop}}(1) = 0.05$$
$$P_{\text{stop}}(4) = 0.20$$
$$P_{\text{stop}}(10) = 0.50$$
$$P_{\text{stop}}(20) = 1.00$$

  This encourages early stopping without explicitly specifying a maximum depth.

- The feature bag at each node is still selected via uniform sampling of k features.

Output Requirements

- Accept α as a command-line parameter.

- Report:

  - Number of nodes forced into being leaf nodes via this method.
  - Accuracy of the resulting tree