Extra Tree Classifier Algorithm:

1. Input:

- Training dataset: D={(x1,y1),(x2,y2),…,(xN,yN)}D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}D={(x1,y1),(x2,y2),…,(xN,yN)} where xix_ixi is a feature vector and yiy_iyi is the corresponding label.
- Number of trees: TTT.
- Number of features to consider for each split: FFF (often M\sqrt{M}M for classification, where MMM is the total number of features).
- Other hyperparameters (tree depth, minimum samples per leaf, etc.).

---

2. Initialization:

- Create TTT empty decision trees to form the forest.

---

3. Build Each Tree:

For each tree ttt in the forest:

1. **Select a Training Subset (Optional)**:
    - Randomly select a subset of the training data DtD_tDt with or without replacement (controlled by bootstrap hyperparameter).

    - If not bootstrapped, use the entire dataset.
2. **Build the Tree**:
    - Start at the root node with all samples in DtD_tDt.
    - Repeat until the stopping criterion is met (e.g., maximum depth, minimum samples at a node):
        - **Randomly Select Features**:
            - Randomly select FFF features from the MMM features.
        - **Randomly Select Split Points**:
            - For each selected feature, choose a random split point within its range.
        - **Evaluate Split Points**:
            - Calculate the split that best separates the data based on a criterion (e.g., Gini impurity or entropy for classification).
        - **Split the Node**:
            - Divide the data at the node into left and right child nodes based on the best split.
    - Mark the node as a leaf if:
        - The stopping criterion is reached, or
        - The node cannot be split further (e.g., all samples have the same label).

---

4. Aggregate Predictions:

- For classification:

- Each tree in the forest outputs a class label prediction for the input xxx.
- Combine predictions from all trees using majority voting to determine the final output label.

---

5. Output:

- Final model consisting of TTT trees.
- For a new input sample xxx, predict its label using the ensemble of trees.

Example in Pseudocode:

Input: Training dataset D, Number of trees T, Number of features F
Output: Trained Extra Tree Classifier model

1. Initialize an empty forest: Forest = []
2. For each tree t in range(1, T):
    a. Select a random subset D_t from D (if bootstrapping is used).
    b. Build a decision tree:
        i. Start at the root node.
        ii. While stopping criteria not met:
            - Randomly select F features.

- For each selected feature, choose a random split point.
- Evaluate split points and find the best one.
- Split the node based on the best split.

iii. Mark the node as a leaf if stopping criteria are met.

c. Add the tree to the forest.

3. Return the forest as the trained model.

Prediction:

1. For an input sample x:
   a. Pass x through each tree in the forest to get predictions.
   b. Combine predictions using majority voting.
2. Return the final predicted class label.

Stopping Criteria:

- Maximum depth of the tree.
- Minimum number of samples per node to allow a split.
- Minimum number of samples per leaf node.
- Impurity threshold for a node to be split further.