MD. RAFIQUL ISLAM
CSE, CoU

Training Set

0	1	0	0	85	85	0	No
1	1	0	0	80	90	1	No
2	0	1	0	83	78	0	Yes
3	0	0	1	70	96	0	Yes
4	0	0	1	68	80	0	Yes
5	0	0	1	65	70	1	No
6	0	1	0	64	65	1	Yes
7	1	0	0	72	95	0	No
8	1	0	0	69	70	0	Yes
9	0	0	1	75	10	0	Yes
10	1	0	0	75	70	1	Yes
11	0	1	0	72	90	1	Yes
12	0	1	0	81	75	0	Yes
13	0	0	1	71	80	1	No

Test Set

14	1	0	0	81	88	0	No
15	0	0	1	74	92	1	Yes
16	0	1	0	76	85	0	Yes
17	1	0	0	78	75	0	No
18	1	0	0	82	92	0	No
19	0	1	0	67	90	1	No
20	0	0	1	85	85	1	Yes
21	0	1	0	73	88	0	Yes
22	1	0	0	88	65	0	Yes
23	0	0	1	77	70	0	Yes
24	1	0	0	79	60	1	Yes
25	0	0	1	80	95	1	Yes
26	0	1	0	66	70	0	No
27	0	0	1	84	78	1	Yes

Here,
Columns are:

'Outlook (Sunny, Overcast, Rainy are one-hot-encoded into 3 columns)',

'Temperature' (in Fahrenheit),

'Humidity' (in %),

'Windy' (Yes/No)

'Play' (Yes/No, target feature)

Step-1

At first, we create depth-1 decision trees/ decision trees stumps as our weak learners. Each stump makes just one split, and we'll train 50 of them sequentially (Default).

Then, start by giving each training example equal weight:

- Each sample gets weight = 1/N (N is total number of samples)
- All weights together sum to 1

Training Set

								Weight W_1
0	1	0	0	85	85	0	No	0.0714
1	1	0	0	80	90	1	No	0.0714
2	0	1	0	83	78	0	Yes	0.0714
3	0	0	1	70	96	0	Yes	0.0714
4	0	0	1	68	80	0	Yes	0.0714
5	0	0	1	65	70	1	No	0.0714
6	0	1	0	64	65	1	Yes	0.0714
7	1	0	0	72	95	0	No	0.0714
8	1	0	0	69	70	0	Yes	0.0714
9	0	0	1	75	10	0	Yes	0.0714
10	1	0	0	75	70	1	Yes	0.0714
11	0	1	0	72	90	1	Yes	0.0714
12	0	1	0	81	75	0	Yes	0.0714
13	0	0	1	71	80	1	No	0.0714
								1.000

For the First Tree

Step-2 Build a decision tree stump while considering sample weights.

Training Set

SORTED	W_0
0 NO	0.0714
1 NO	0.0714
5 NO	0.0714
7 NO	0.0714
13 NO	0.0714
2 YES	0.0714
3 YES	0.0714
4 YES	0.0714
6 YES	0.0714
8 YES	0.0714
9 YES	0.0714
10 YES	0.0714
11 YES	0.0714
12 YES	0.0714
	1

NO: 0.3571
YES: 0.6426

a. Calculate initial weighted Gini impurity for the root node.

FORMULA

$$1-\left(\frac{YES}{\text{Total Weight}}\right)^2-\left(\frac{NO}{\text{Total Weight}}\right)^2$$

Gini Impurity (with weight)

$$1 - (0.3571)^2 - (0.6426)^2 = 0.4596$$

b. For each feature:

- Sort data by feature values (exactly like in Decision Tree classifier)

SORTED (Sunny)

5	0
13	0
2	0
3	0
4	0
6	0
9	0
11	0
12	0
0.5	
0	1
1	1
7	1
8	1
10	1

1 Split Point

SORTED (Overcast)

0	0
1	0
5	0
7	0
13	0
3	0
4	0
8	0
9	0
10	0
0.5	
2	1
6	1
11	1
12	1

1 Split Point

SORTED (Rainy)

0	0
1	0
7	0
2	0
6	0
8	0
10	0
11	0
12	0
0.5	
5	1
13	1
3	1
4	1
9	1

1 Split Point

SORTED (Temperature)

6	64
64.5	
5	65
66.5	
4	68
68.5	
8	69
69.5	
3	70
70.5	
13	71
71.5	
7	72
11	72
73.0	
9	75
10	75
77.5	
1	80
80.5	
12	81
82.0	
2	83
84.0	
0	85

11 Split Points

SORTED (Humidity)

6	65
67.5	
5	70
8	70
10	70
72.5	
12	75
76.5	
2	78
79.0	
13	80
4	80
9	80
82.5	
0	85
87.5	
1	90
11	90
92.5	
7	95
95.5	
3	96

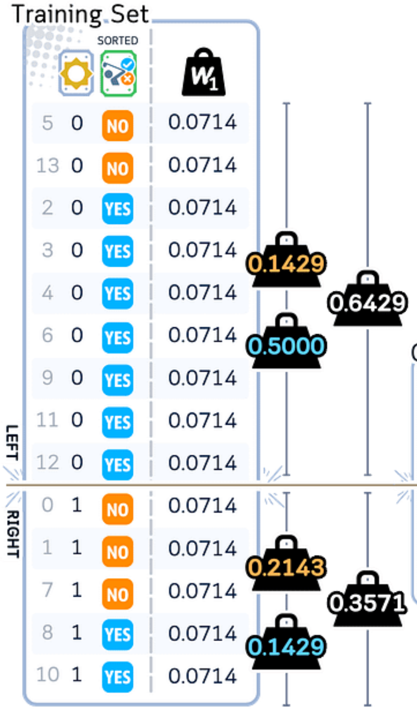
8 Split Points

SORTED (Windy)

5	0
13	0
2	0
3	0
4	0
6	0
9	0
11	0
0.5	
12	1
0	1
1	1
7	1
8	1
10	1

1 Split Point

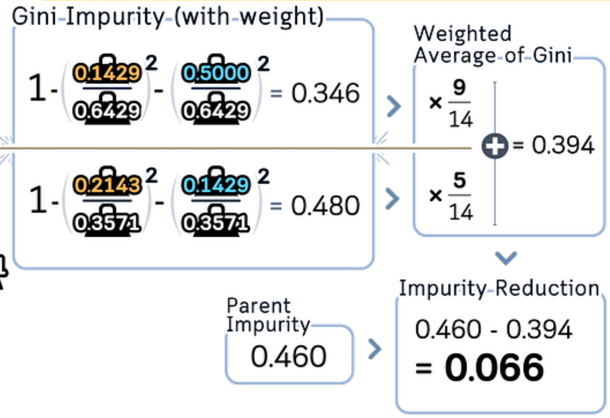
MD. RAFIQUL ISLAM
CSE, CoU



For each possible split point:

- Split samples into left and right groups
- Calculate weighted Gini impurity for both groups
- Calculate weighted Gini impurity reduction for this split

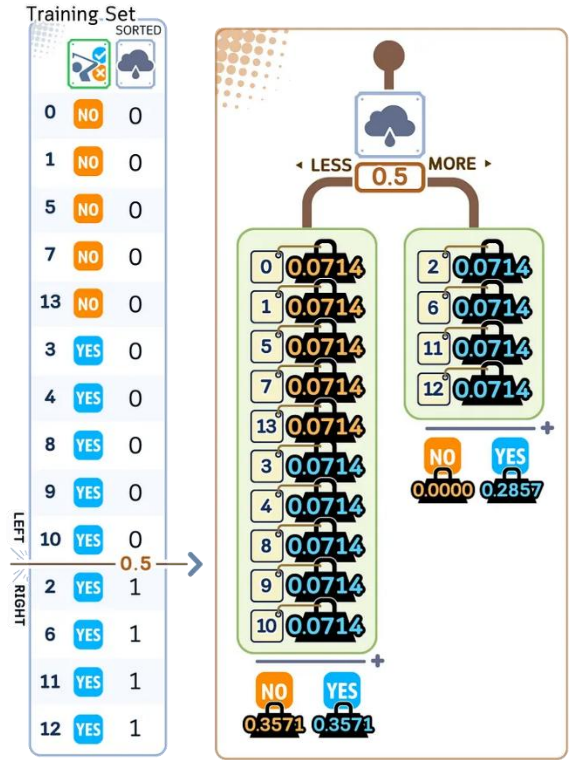
For feature "sunny"



c. Pick the split that gives the largest Gini impurity reduction.

Split Points	Impurity Reduction	Split Points	Impurity Reduction	Split Points	Impurity Reduction	Split Points	Impurity Reduction
0.5	0.066	69.5	0.009	82.0	0.007	82.5	0.066
0.5	HIGHEST 0.102	70.5	0.027	84.0	0.064	87.5	0.016
0.5	0.002	71.5	0.001	67.5	0.020	92.5	0.007
64.5	0.020	73.0	0.001	72.5	0.009	95.5	0.020
66.5	0.007	77.5	0.016	76.5	0.027	0.5	0.031
68.5	0.000	80.5	0.000	79.0	0.054		

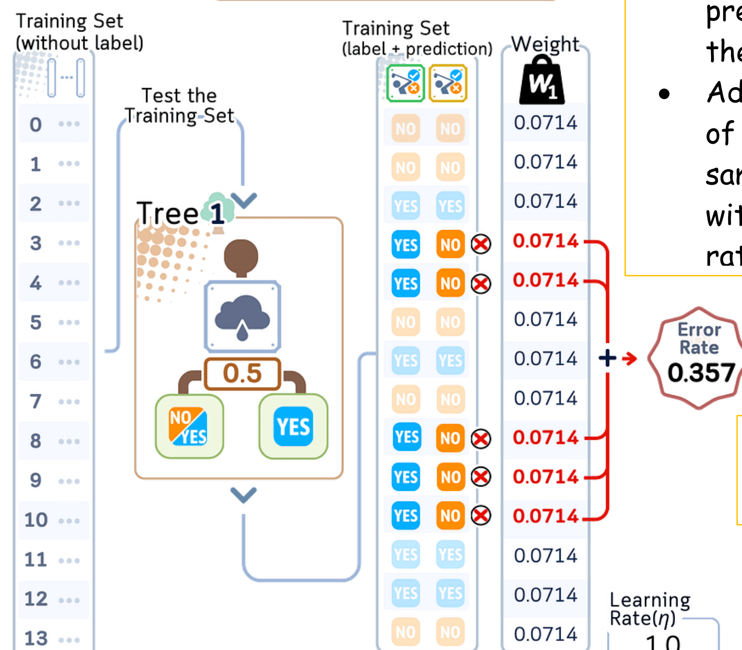
After checking all possible splits across features, the column 'overcast' (with split point 0.5) gives the highest impurity reduction of 0.102. This means it's the most effective way to separate the classes, making it the best choice for the first split.



d. Create a simple one-split tree using this decision.

e. Evaluate how good this tree is

- Use the tree to predict the label of the training set.
- Add up the weights of all misclassified samples (marked with X) to get error rate.



f. Calculate tree importance (α)

FORMULA

$$\alpha = \eta \cdot \log\left(\frac{1-\text{Error Rate}}{\text{Error Rate}}\right)$$

Learning Rate(η)
1.0

Error Rate
0.357

α_1
0.5878

MD. RAFIQUL ISLAM
CSE, CoU

WEIGHT UPDATE FORMULA

$W_{new} = W_{old} \cdot \exp(\alpha)$ if ✗

Training Set (label + prediction)			Weight W_1	New Weight W_2	New Weight (Normalized) W_2
0	NO	NO	0.0714	0.0714	0.0556
1	NO	NO	0.0714	0.0714	0.0556
2	YES	YES	0.0714	0.0714	0.0556
3	YES	NO ✗	0.0714	0.1286	0.1000
4	YES	NO ✗	0.0714	0.1286	0.1000
5	NO	NO	0.0714	0.0714	0.0556
6	YES	YES	0.0714	0.0714	0.0556
7	NO	NO	0.0714	0.0714	0.0556
8	YES	NO ✗	0.0714	0.1286	0.1000
9	YES	NO ✗	0.0714	0.1286	0.1000
10	YES	NO ✗	0.0714	0.1286	0.1000
11	YES	YES	0.0714	0.0714	0.0556
12	YES	YES	0.0714	0.0714	0.0556
13	NO	NO	0.0714	0.0714	0.0556

α_1 0.5878

NORMALIZE

1

g. Update sample weights

- Keep the original weights for correctly classified samples
- Multiply the weights of misclassified samples by e^(α).
- Divide each weight by the sum of all weights. This normalization ensures all weights still sum to 1 while maintaining their relative proportions.

Cases where the tree made mistakes (marked with X) get higher weights for the next round.

For the Second Tree

Training Set (SORTED)		W_2
0	NO	0.0556
1	NO	0.0556
5	NO	0.0556
7	NO	0.0556
13	NO	0.0556
2	YES	0.0556
3	YES	0.1000
4	YES	0.1000
6	YES	0.0556
8	YES	0.1000
9	YES	0.1000
10	YES	0.1000
11	YES	0.0556
12	YES	0.0556

NO: 0.2778
YES: 0.7222
1

a. Build a new stump, but now using the updated weights. Calculate new weighted Gini impurity for root node.

b. For each feature:
Same process as before, but the weights have changed.

Gini Impurity (with weight)

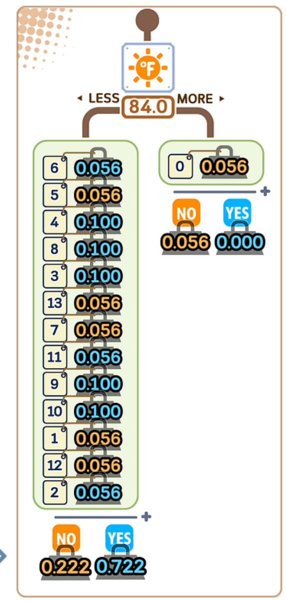
$$1 - 0.2778^2 - 0.7222^2 = 0.4012$$

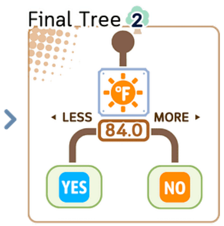
c. Pick the split with best weighted Gini impurity reduction.
Notice that "overcast" is no longer the best split — the algorithm now finds temperature (84.0) gives the highest impurity reduction,

Split Points	Impurity Reduction	Split Points	Impurity Reduction	Split Points	Impurity Reduction	Split Points	Impurity Reduction
0.5	0.036	69.5	0.009	82.0	0.012	82.5	0.055
0.5	0.044 (Not highest anymore)	70.5	0.028	84.0	0.061 (HIGHEST)	87.5	0.014
0.5	0.0	71.5	0.003	67.5	0.009	92.5	0.002
64.5	0.009	73.0	0.0	72.5	0.009	95.5	0.017
66.5	0.012	77.5	0.028	76.5	0.018	0.5	0.032
68.5	0.0	80.5	0.001	79.0	0.031		

Training Set		SORTED
6	YES	64
5	NO	65
4	YES	68
8	YES	69
3	YES	70
13	NO	71
7	NO	72
11	YES	72
9	YES	75
10	YES	75
1	NO	80
12	NO	81
2	YES	83
0	YES	85

LEFT 84 RIGHT



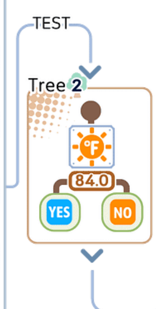


d. Create the second stump.

After **e**, **f**, **g**

2nd tree achieves a lower error rate (0.222) and higher importance score (α = 1.253) than the first tree.

Like before, misclassified examples get higher weights for the next round.



Training Set (without label)	Training Set (label + prediction)		Weight W_2	Updated Weight W_3	Updated Weight (Normalized) W_3
0	NO	NO	0.0556	0.0556	0.0357
1	NO	YES ✗	0.0556	0.1944	0.1250
2	YES	YES	0.0556	0.0556	0.0357
3	YES	YES	0.1000	0.1000	0.0643
4	YES	YES	0.1000	0.1000	0.0643
5	NO	YES ✗	0.0556	0.1944	0.1250
6	YES	YES	0.0556	0.0556	0.0357
7	NO	YES ✗	0.0556	0.1944	0.1250
8	YES	YES	0.1000	0.1000	0.0643
9	YES	YES	0.1000	0.1000	0.0643
10	YES	YES	0.1000	0.1000	0.0643
11	YES	YES	0.0556	0.0556	0.0357
12	YES	YES	0.0556	0.0556	0.0357
13	NO	YES ✗	0.0556	0.1944	0.1250

Error Rate 0.222

α_2 1.253

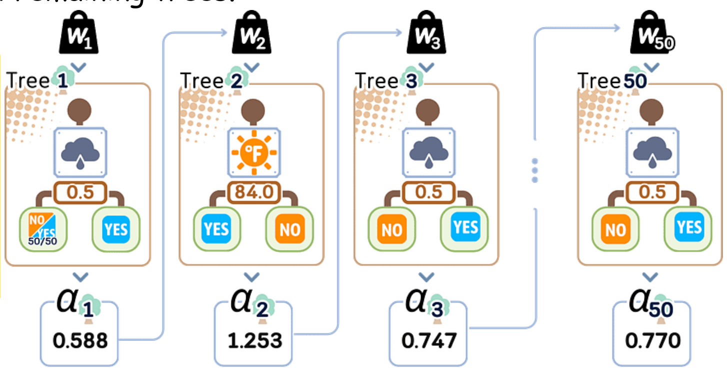
1

MD. RAFIQUL ISLAM
CSE, CoU

For the Third Tree onwards

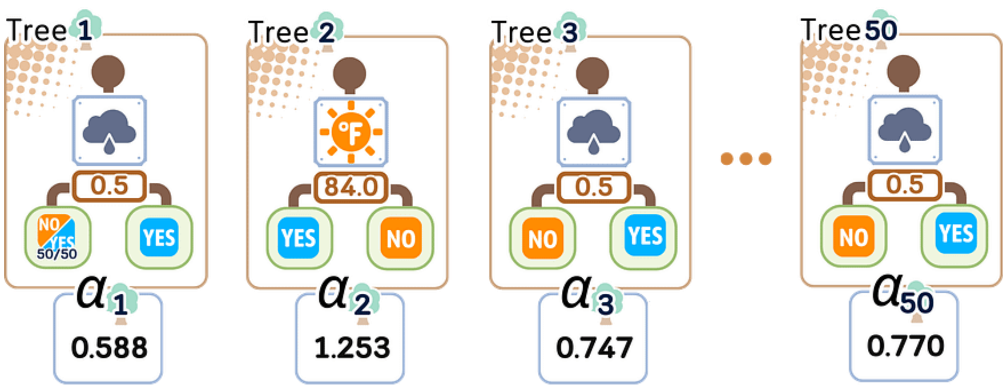
Repeat Step-2 for all remaining trees.

The algorithm builds 50 simple decision trees sequentially, each with its own importance score (α).

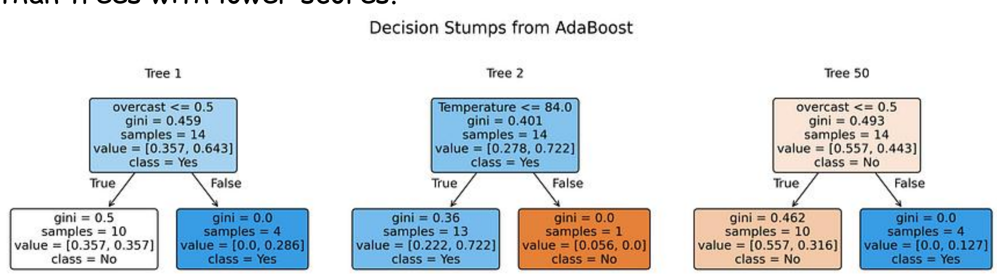


Step 3: Final Ensemble

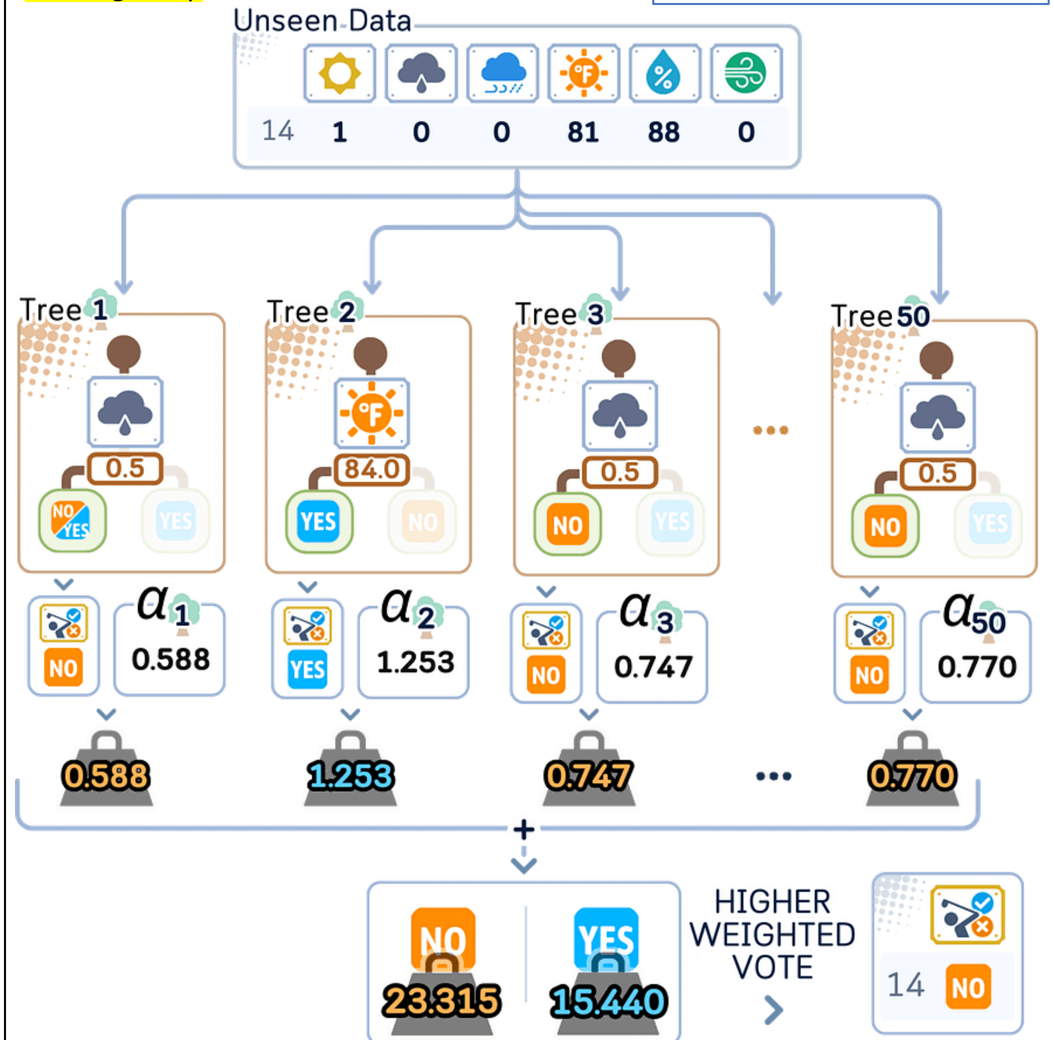
Keep all trees and their importance scores



The 50 simple decision trees work together as a team, each with its own importance score (α). When making predictions, trees with higher α values (like Tree 2 with 1.253) have more influence on the final decision than trees with lower scores.



Testing Step



- When predicting for new data, each tree makes its prediction and multiplies it by its importance score (α).
- The final decision comes from adding up all weighted votes — here, the NO class gets a higher total score (23.315 vs 15.440),
- So, the model predicts NO for this unseen example.