Random Forests

A Random forest is a machine learning algorithm that builds many decision trees and then combines their results to make a final decision.

Advantages :-
1. High Accuracy : usually better than single trees
2. Handles missing data well
3. Reduces overfitting
4. Works well for both regression & classification

Disadvantages :-
1. Slower → many trees → more computation
2. Less interpretable → hard to visualise whole forest
3. Takes more memory

Usage :-

1. Finance	fraud detection, loan approval
2. Healthcare	disease prediction from symptom
3. E-commerce	sentiment review
4. Agriculture	crop yield prediction
5. Education	student performance prediction.

Main Goal :- Reduce overfitting & improving accuracy

Algorithm with example :-

ID	Age	Income	Buys - Computer
1	<=30	H	N
2	<=30	H	N
3	31-40	H	Y
4	>40	M	Y
5	>40	L	Y
6	>40	L	N
7	31-40	L	Y
8	<=30	M	N
9	<=30	L	Y
10	>40	M	Y

Features :- Age, income

Target :- Buys - computer

Step 1

Create Bootstrap Samples (Random)

Sample 1 ⇒ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 (full dataset)

Sample 2 ⇒ 2, 4, 5, 7, 8, 9, 9, 1, 6, 3

Sample 3 ⇒ 3, 3, 5, 6, 7, 8, 9, 10, 10, 4

Step 2

Build Tree 1 (using sample 1), Calculate root Gini for full data

Total Yes ⇒ IDs : 3, 4, 5, 7, 9, 10 ⇒ 6

Total No ⇒ IDs : 1, 2, 6, 8 ⇒ 4

$Gini = 1 - \left(\left(\frac{6}{10}\right)^2 + \left(\frac{4}{10}\right)^2 \right) \Rightarrow .48$ $\left\{ Gini = 1 - \left(P_{yes}^2 + P_{No}^2 \right) \right.$

step 2a :- Try splitting on Age, incomes to decide root node

~~Calculate root Gini for 31 (Age <=30)~~

~~Age <=30 ⇒ ID~~

We'll try splitting on age,

Age divisions ⇒ 1. <=30, 2. 31-40, 3. >40 → summarise in 2 categories $\boxed{\text{1. } <=30 \quad \text{2. } >30}$

now, Split 1. Age <=30 vs Age >30

Left (<=30): IDs = 1, 2, 8, 9   Y=1, N=3

Right (>30) : IDs = 3, 4, 5, 6, 7, 10   Y=5, N=1

$$Gini_{left} = 1 - \{(P_{left\ yes})^2 + (P_{left\ no})^2\} = .375$$

$$Gini_{Right} = .278$$

$$Gini_{Age\ (combined)} \Rightarrow .375 \times \frac{4}{10} + .278 \times \frac{6}{10} \Rightarrow .316$$

↳ weighted Gini

now, split 2. a) Income = High vs Income ≠ High

weighted Gini = .419

b) low vs others

weighted Gini = .45

c) med vs others

Weighted Gini = .475

Now, what we have done so far is, we calculated root Gini, i.e. Gini for full data, i.e. .48

& then we have done several splits, based on age & income.

So, the split with the lowest Gini or the one which has the maxᵐ decrease from the root Gini is chosen as the Root Node

So, best split so far was Age (Gini = ·316)

Root = "Age <= 30 ?"

ep 3 :-

Left Node (Age <=30) : IDS ⇒ 1, 2, 8, 9

Y=1, N=3, Gini = ·375

Try splitting on income again for left Node

- High (with Age <= 30) ⇒ IDs 1, 2 → N & N → Gini = 0

Best split ⇒ Income High vs others

Left leafs

High → No

Medium / Low → further split possible

ow, Med vs Low,

Med → (ID 8) → No → Gini = 0

Low → (ID 9) → Yes → Gini = 0

ight Node (Age > 30): IDS ⇒ 3, 4, 5, 6, 7, 10

Y = 5 N = 1 Gini = ·278

Income: high vs others ⇒ G = 0 ✓

med & low can be split again

med → ID (4, 10) → Y & Y, G = 0

low → ID (5, 6, 7) → 2Y & 1N, $G \Rightarrow 1 - \left(\left(\frac{2}{3}\right)^2 + \left(\frac{1}{3}\right)^2\right)$

$\Rightarrow 1 - \left(\frac{4}{9} + \frac{1}{9}\right) \Rightarrow \cdot 44$

Tree 1

Root Gini ⇒ .48   Root Node ⇒ (Age <= 30)? , lowest Gini

[Root : Age <= 30 ?]

- Yes → Left Node (Age <= 30)
  - [Income = High?] best split
    - Yes → Income = High → NO
    - NO → Income ≠ High, i.e. (med or low)
      - [Income = Low?]
        - Yes → Income = low → Yes
        - NO → Income ≠ low i.e. med → NO
- NO → Right Node (Age > 30)
  - [Income = High?]
    - Yes → High → Yes
    - NO → Med/L
      - [Income = Me
        - Yes → med → Y
        - N → lo → Y(majo

The same procedure goes on with other two trees as well & finally, build a table as

ID	Tree 1	Tree 2	Tree 3	Majority
1	N	Y	N	N
2	N	Y	N	N
3	Y	Y	Y	Y
.				
.				
.				