Random Forest

Definition

→ Ensemble learning algorithm (non-parametric)

→ Ensemble learning : multiple models try to solve the same problem by combining predictio

→ part of the Bootstrap Aggregating (Bagging) fam.

Workflow (Basic algorithm)

→ create n decision tree models from the dataset.

→ conduct majority voting / averaging for final pred.
classification regression

Detailed algorithm

1. Randomly sample rows with replacement (Bootstrapping)
2. Randomly pick a subset of features.
3. Calculate measure of randomness (e.g., entropy, gini impurity) of the selected features, and place the best feature in the root node.
4. Split recursively until stopping condition is met.

5. Stopping condition will be met, and we will get a tree.

6. Repeat step 1 to step 5 to build n more trees.

Prediction making

7. After making enough trees, send a test instance to the Random Forest model.

8. The 'n' trees will make individual predictio The final prediction is the majority voting of the models (for classification). For regression, we take average.

Visualization

Assume a dataset,

age	smoking	fatigue	cough	cancer
—	—	—	—	—
—	—	—	—	—
—	—	—	—	—

RF does the following for a test set:

Tree 1 makes prediction with age, smoking, fatigue

Tree 2 " " with smoking, cough, age

Tree 3 " " with age, fatigue

⋮

Tree n " " with smoking, cough

Let, n = 8 (8 trees)

Tree	Prediction
1	Yes
2	No
3	Yes
4	Yes
5	No
6	Yes
7	Yes
8	No

Yes → 5 times

No → 3 times

So, the patient will be diagnosed as cancer.

Exercise: Create a Random Forest classifier using the following dataset to predict whether a girl should play tennis or not.

Day	Outlook	Temp	Hum	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cold	Normal	Strong	Yes
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Step 01: Bootstrapped Sampling #1

Day	Outlook	Temp	Hum	Wind	PlayTennis
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No
2	Sunny	Hot	High	Strong	No

Step 02: Randomly take subset of features and calculate gini impurity.

feature_set = {Outlook, Temp, Hum, Wind}

chosen_subset = {Temp, Hum}

GINI (Temperature)

$$= \frac{4}{6} \times \left[1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2\right] + \frac{2}{6} \times \left[1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2\right]$$

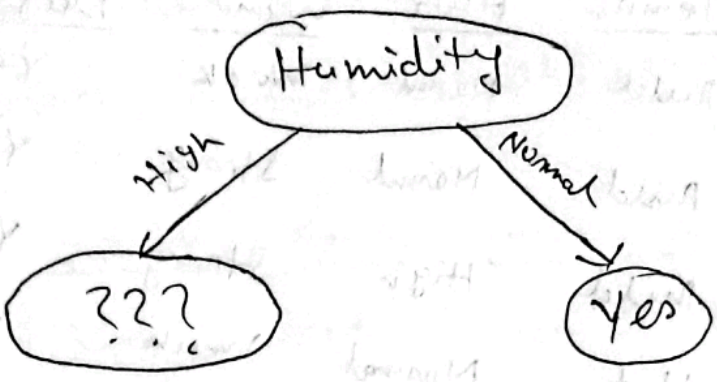
$$= 0.417$$

GINI (Humidity)

$$= \frac{3}{6} \times \left[1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2\right] + \frac{3}{6} \times \left[1 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2\right]$$

$$= 0.2223$$

Since, GINI (Temperature) > GINI (Humidity)



Randomly pick subset of features
we pick, [Temp, outlook]

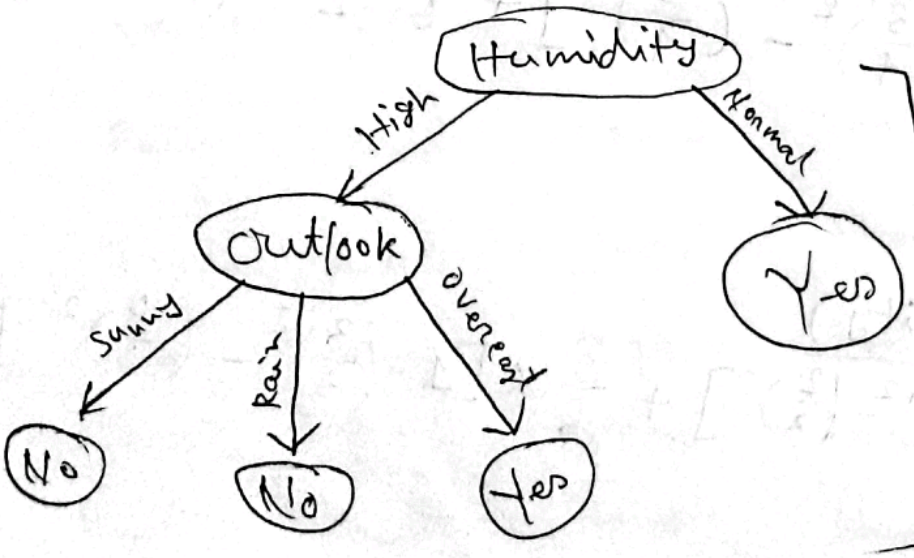
Gini (Temperature)

$$\frac{2}{3} \times \left[1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2\right] + \frac{1}{3} \times \left[1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2\right]$$

$$= 0.333$$

Gini (outlook) = 0

Since, Gini (outlook) < Gini (temperature)



We built our first tree. Thus, we can create more trees to perform majority voting.

Pebo and bormula ob Gini impurity

Calculating gini impurity ob a category under a variable:

Example; Temperature → hot, cool, mild
variable categories

⊛⊗ there are 6 Yes, 2 No in cool category.
Find ~~Gini (coo~~ Gini(Temperature = cool)

Soln: Gini(temp = cool)

$= 1 - (\frac{6}{8})^2 - (\frac{2}{8})^2 = 0.375$ (Ans)

⊛ Gini $\in [0, 1 - \frac{1}{c}]$, where, c = num of classes

⊛ Definition: Gini impurity is the probability that a randomly chosen item from this node would be incorrectly classified if we assign the class label randomly.

⊛ Gini (variable), or Gini(X) =

$$= \sum_{i=1}^{k} \frac{|Ci|}{B} \cdot Gini(Ci)$$

B = num_sample
|Ci| = num category in the samples
Ci = category name

How many rows to include in bootstrap?

→ All rows. But in exams, take 6-8 rows.

How many features to select?

→ for classification: $\sqrt{d}$, where d = # features

for regression: $d/3$

(These are the defaults in scikit-learn)

Pros and cons of Random Forest

Pros

1. Handles both cf and rg.
2. Robust to overfitting.
3. Handles high-dimensional data.
4. Captures non-linear relationship.

Cons

1. Slower inference due to many trees.
2. Doesn't extrapolate well for regression.
3. Can still overfit with many deep trees.

Shallow tree → low depth, typicall 2 to 4

Deep tree → high depth, maybe 10+

Why do RF is less prone to overbitting than DT?

→ Trees even capture subtle patterns and noise in the data due to randomness measurement metrics, that leads to overbitting. RF, creates multiple trees with different feature subsets, leading to a forest of different trees designed for the same task.