Random Forest

Prof. Surya Prakash

IIT Indore

Introduction

- An ensemble learning method
  - It builds multiple decision trees and combines their outputs.

- Ensemble methods (like Random Forest) follow the principle of *wisdom of the crowd*.
  - Instead of relying on a single model (or "person"), they combine the knowledge of many models (like a "crowd") to get a more reliable prediction

- Used for both **classification** and **regression** tasks.

Introduction

- We'll discuss classification using Random Forest

- Dataset used in discussion: Play Tennis dataset
  - Goal:
    - Predict if a game will be played or not
    - Prediction based on weather conditions

- Play Tennis dataset
  - Weather Attributes Used to Predict 'Play'
  - Attributes/Features: Outlook, Temperature, Humidity, Windy
  - Target: Play (Yes/No)
  - 14 instances with categorical features and binary target

Play Tennis Dataset (14 instances)

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

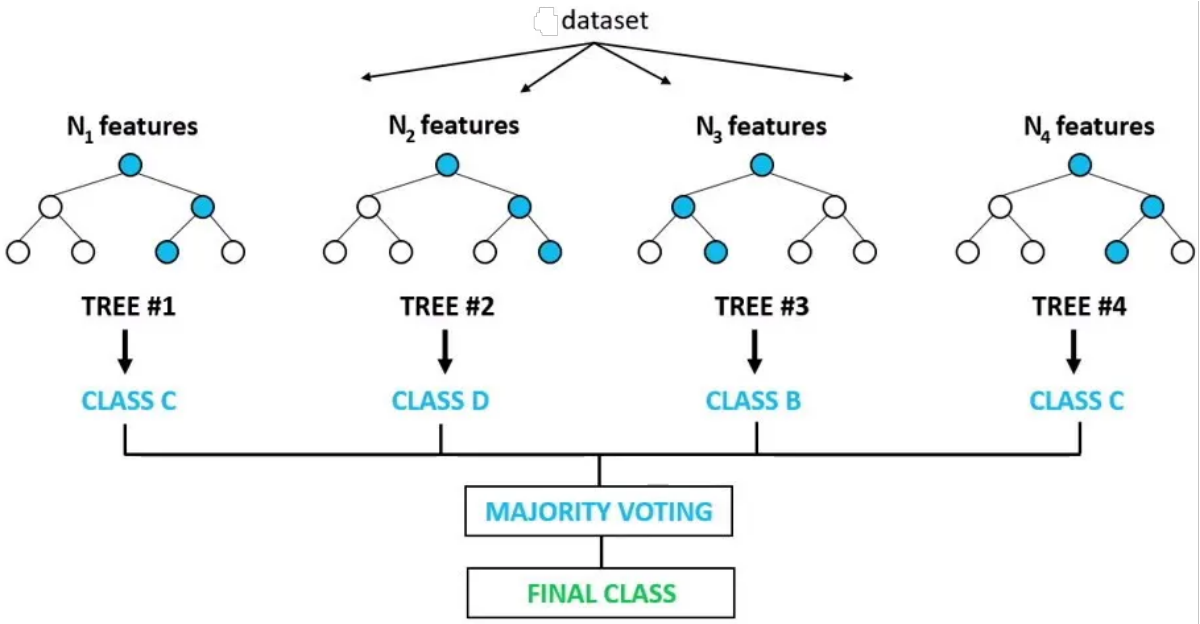
Problem Statement

- Perform classification using Random Forest
  - Use of Random forest in place of in place of Decision Tree

- Random Forest provides
  - Better accuracy
  - Better generalization
  - Reduces overfitting
  - Robust to noise

What is Random Forest?

- Ensemble method using multiple decision trees
- Makes use of bootstrapping for model training
- Combines Bagging + Decision Trees



Step 1 – Bootstrapping

- Bootstrapping is a statistical technique for sampling data **with replacement**.

- What is Sampling?
  - It is the process of selecting a subset of data (called a *sample*) from a larger set (called the *population*).

- Example:
  - Population: Students in a class = [A, B, C, D, E].
  - Sample set: [B, D, A]

- Advantage of Sampling
  - Instead of studying the **entire population**, we work with a **smaller representative group** to make predictions or conclusions.

Step 1 – Bootstrapping

- Sampling approaches
  - **Random Sampling with Replacement** – selected item goes back and can be picked again (used in bootstrapping).
  - **Sampling without Replacement** – once selected, it cannot be picked again.
- Example:
  - **Population (Original Data):** Suppose we have exam scores of 10 students: [45, 50, 55, 60, 65, 70, 75, 80, 85, 90]

Random Sampling Without Replacement (pick 5)
We randomly pick 5 scores, **no repeats:**
Example → [50, 65, 70, 85, 90]

Random Sampling With Replacement (Bootstrapping, pick 5)
We randomly pick 5 scores, allowing repeats:
Example 1 → [55, 90, 55, 70, 80]
Example 2 → [45, 65, 65, 85, 50]
Example 3 → [75, 60, 75, 90, 70]

Step 1 – Bootstrapping

- **Purpose:**
  - To create multiple datasets (called *bootstrap datasets or samples*) from the original dataset.
  - Each "bootstrapped dataset" is of the same size as the original, but some records may appear **more than once**, and some may be **left out**.
  - These multiple datasets are then used to train different models (e.g., decision trees in a Random Forest).
  - Since each model is trained on a different bootstrapped dataset (and random feature subset), the models learn differently.

Example→

Example of Bootstrapping

- Consider Dataset (6 instances):
  - D1, D2, D3, D4, D5, D6

	Outlook	Temperature	Humidity	Windy	Play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no

Now, to build Tree 1, we take a **random sample with replacement** from this dataset:

🌳 Tree 1 - Sample:
[D2, D4, D5, D2, D6, D6]

Notice:
→ D2 and D6 are repeated.
→ D1 and D3 are missing.

Now, to build Tree 2, we take a **random sample with replacement** from this dataset:

🌳 Tree 2 - Sample:
[D1, D1, D3, D4, D5, D5]

Notice:
→ D1 and D5 are repeated.
→ D2 and D6 are missing.

Now, to build Tree 3, we take a **random sample with replacement** from this dataset:

🌳 Tree 3 - Sample:
[D2, D3, D4, D5, D6, D3]

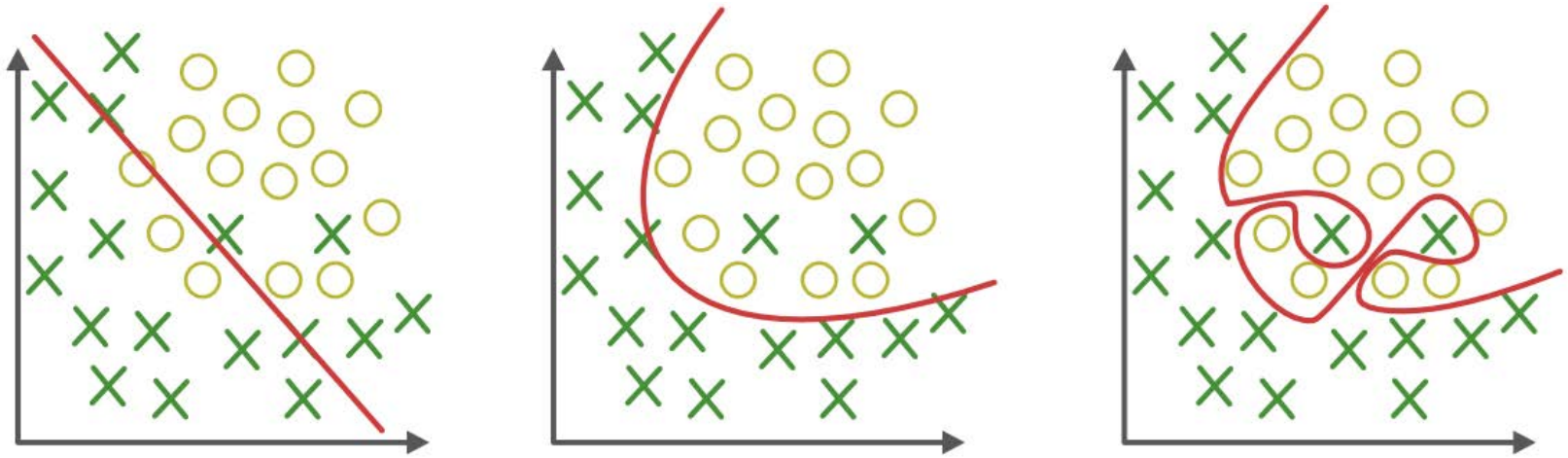
Notice:
→ D3 is repeated.
→ D1 is missing.

How does Bootstrapping help?

- Each tree **sees a slightly different view** of the data.
- This introduces **diversity** in tree structures, splits, and predictions.
- When Random Forest combines their predictions (by majority vote or averaging), this **reduces overfitting** and **improves accuracy**.

Overfitting?

- Overfitting happens when a model **learns the training data too well**, including its **noise and random fluctuations**, instead of just the general patterns.
- As a result, the model performs **very well on training data**, but **poorly on unseen/test data**.



Underfitting
(Too simple to explain the variance)

Good fitting

Overfitting
(Too good to be true)

Step 2 - Feature Subset Selection

- In **Random Forests**,
  - not only is each tree trained on a random **sample of data instances** (bootstrapping),
  - but also, at **each node/split**, the algorithm randomly selects a **subset of features** to consider for the best split — **not all features**.

- Advantages?
  - This method introduces extra randomness into the forest.
  - The randomness reduces correlation between trees (prevent them from making the same splits).
  - Improved generalization: more diverse trees = better ensemble performance.

Step 3 - Build Decision Tree

- Use selected data + features to train decision tree
- Use criteria like following to train decision trees
  - Information Gain (ID3 algorithm) or
  - Gini Index (CART - Classification and Regression Trees algorithm)
- Stop when node is pure or depth is maxed

Splitting Criteria: Information Gain

Information Gain = Entropy(Parent) - Weighted Average (Entropy Children)

- Use this to decide best feature at each node for splitting
- Repeat for each tree individually

Building a Random Forest with 5 trees

14
Instances →

	Outlook	Temperature	Humidity	Windy	Play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no
D7	overcast	cool	normal	true	yes
D8	sunny	mild	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D11	sunny	mild	normal	true	yes
D12	overcast	mild	high	true	yes
D13	overcast	hot	normal	false	yes
D14	rainy	mild	high	true	no

Step 1: Bootstrapping – Create 5 Bootstrapped Samples

- We sample **14 instances with replacement** for each tree.

- **Bootstrapped Samples (example):**
  - **Tree 1:**
    - [D2, D4, D5, D2, D11, D8, D1, D6, D9, D14, D4, D5, D10, D10]
    - Repeated: D2, D4, D5, D10
  - **Tree 2:**
    - [D1, D3, D6, D6, D13, D11, D14, D7, D2, D3, D9, D12, D12, D10]
    - Repeated: D3, D6, D12
  - **Tree 3:**
    - [D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D2, D3, D4]
    - Repeated:
  - **Tree 4:**
    - [D1, D1, D1, D2, D2, D3, D5, D5, D6, D7, D8, D9, D10, D14]
    - Repeated:
  - **Tree 5:**
    - [D4, D4, D8, D8, D13, D12, D11, D9, D5, D6, D7, D2, D3, D1]
    - Repeated:

Step 2: Random Feature Subset at Each Node

- Assume at each split, we randomly choose **2 out of 4 features**:
  - Outlook
  - Temperature
  - Humidity
  - Windy
- For each tree and each node, feature selection will differ.

- So, while building the Decision Tree-1 to Tree 5, we would consider
  - Random 2-feature splits,
  - Stopping at max depth = 3 or when nodes are pure

Step 3: Build Each Tree (simplified)

- Let us build **Tree 1** step-by-step.
- Tree 1 Bootstrap Sample
  - [D2, D4, D5, D2, D11, D8, D1, D6, D9, D14, D4, D5, D10, D10]

- **Step 1: Entropy of the Root Node**
- Play = Yes: 8
  Play = No: 6

$$Entropy(S) = -\frac{8}{14}\log_2\frac{8}{14} - \frac{6}{14}\log_2\frac{6}{14}$$

$$\approx -0.571 \cdot \log_2 0.571 - 0.429 \cdot \log_2 0.429$$

$$\approx 0.985$$

Instance	Outlook	Temp	Humidity	Windy	Play
D2	sunny	hot	high	true	no
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D11	sunny	mild	normal	true	yes
D8	sunny	mild	high	false	no
D1	sunny	hot	high	false	no
D6	rainy	cool	normal	true	no
D9	sunny	cool	normal	false	yes
D14	rainy	mild	high	true	no
D10	rainy	mild	normal	false	yes
D2	sunny	hot	high	true	no
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D10	rainy	mild	normal	false	yes

Step 3: Build Each Tree (simplified)

- Step 2: Randomly Select 2 Features (e.g., Outlook, Windy)
- ➤ Try Splitting on **Outlook**

Outlook	Count	Yes	No
Sunny	6	2	4
Rainy	8	6	2

Sunny: D2, D2, D11, D8, D1, D9 → 6 instances

- yes = 2 (D11, D9)
- no = 4 (D2, D2, D8, D1)

$$E = -\frac{2}{6}\log_2\frac{2}{6} - \frac{4}{6}\log_2\frac{4}{6} \approx 0.918$$

Rainy: D4, D5, D6, D14, D4, D5, D10, D10 → 8 instances

- yes = 6 (D4, D5, D4, D5, D10, D10)
- no = 2 (D6, D14)

$$E = -\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8} \approx 0.811$$

$$E_{\text{Outlook}} = \frac{6}{14}\cdot 0.918 + \frac{8}{14}\cdot 0.811 \approx 0.857$$

$$\text{Gain}(\text{Outlook}) = 0.985 - 0.857 = 0.128$$

Step 3: Build Each Tree (simplified)

- Step 2: Randomly Select 2 Features (e.g., Outlook, Windy)
- ➤ Try Splitting on **Windy**

Outlook	Count	Yes	No
True	5	1	4
False	9	7	2

Windy = true: D2, D2, D11, D6, D14 → 5 instances

- yes = 1 (D11)
- no = 4 (D2, D2, D6, D14)

$$E = -\frac{1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{4}{5} \approx 0.722$$

Windy = false: D4, D5, D8, D1, D9, D4, D5, D10, D10 → 9 instances

- yes = 7
- no = 2

$$E = -\frac{7}{9}\log_2\frac{7}{9} - \frac{2}{9}\log_2\frac{2}{9} \approx 0.764$$

$$E_{\text{Windy}} = \frac{5}{14}\cdot 0.722 + \frac{9}{14}\cdot 0.764 \approx 0.751$$

$$\text{Gain(Windy)} = 0.985 - 0.751 = 0.234$$

Choose Windy (higher gain)

Best Split at Root: **Windy** (Gain = 0.234)

Tree After First Split:

Step 3: Build Each Tree (simplified)

- **Tree Root Split: Windy**
- Branches: true, false
- **Left Subtree: Windy = true (5 instances)**

Instance	Outlook	Temp	Humidity	Windy	Play
D2	sunny	hot	high	true	no
D11	sunny	mild	normal	true	yes
D6	rainy	cool	normal	true	no
D14	rainy	mild	high	true	no
D2	sunny	hot	high	true	no

▶️ **Left Subtree:** `Windy = true` **(5 instances)**

| D2, D2, D11, D6, D14 |

- Play: yes = 1, no = 4
- Entropy ≈ 0.722
- Random Features: **Humidity, Windy**

(Only Windy is repeated — it has no gain here, so we use only Humidity.)

Try `Humidity`:

- **High**: D2, D2, D14 → no = 3 → *pure* (entropy = 0)
- **Normal**: D11 (yes), D6 (no) → mixed → entropy = 1

$$\text{Weighted entropy} = \frac{3}{5}(0) + \frac{2}{5}(1) = 0.4$$

Weighted entropy is still lower than parent — split accepted.

- **Humidity = high** → predict **no** (pure)
- **Humidity = normal** → mixed → **predict parent's majority = no**

✅ Final predictions:

- Humidity = high → no
- Humidity = normal → no (majority)

Step 3: Build Each Tree

- **Right Subtree: Windy = false (9 instances)**

Instance	Outlook	Temp	Humidity	Windy	Play
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D8	sunny	mild	high	false	no
D1	sunny	hot	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D10	rainy	mild	normal	false	yes

▶️ Right Subtree: `Windy = false` (9 instances)

| D4, D5, D8, D1, D9, D4, D5, D10, D10 |

- Play: yes = 7, no = 2
- Entropy ≈ 0.764
- Random Features: **Humidity, Windy**

Try `Humidity`:

- **High**: D4, D8, D1, D4 → yes = 2, no = 2 → entropy = 1
- **Normal**: D5, D9, D5, D10, D10 → yes = 5 → pure

Weighted entropy:

$$E = \frac{4}{9} \cdot 1 + \frac{5}{9} \cdot 0 = 0.444$$

$$\text{Gain} = 0.764 - 0.444 = 0.320$$

✅ Best split: **Humidity**

Step 3: Build the Tree (Tree-1)

- Now, if Humidity = normal → predict **yes**

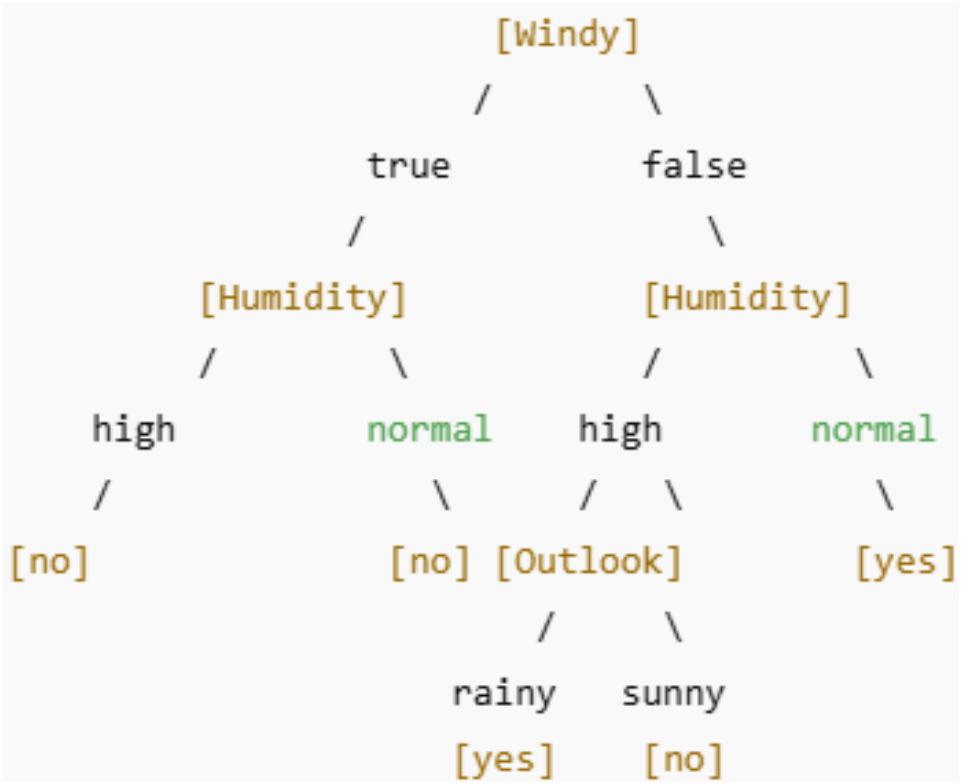
Instance	Outlook	Temp	Humidity	Play
D5	rainy	cool	normal	yes
D9	sunny	cool	normal	yes
D10	rainy	mild	normal	yes
D5	rainy	cool	normal	yes
D10	rainy	mild	normal	yes

✂️ **Subgroup:** `Humidity = normal`

| D5, D5, D9, D10, D10 |

- All have Play = **yes**
- Pure Node ✅

  → **Prediction = yes**



Step 3: Build the Tree (Tree-1)

- If Humidity = high

➡️ **Mixed Class:**

- D4 (rainy): yes, yes
- D8 & D1 (sunny): no, no

Instance	Outlook	Temp	Humidity	Play
D4	rainy	mild	high	yes
D8	sunny	mild	high	no
D1	sunny	hot	high	no
D4	rainy	mild	high	yes

- Let's now split this group further by Humidity or Outlook.
- We had decided using {Humidity, Outlook} at this level,
  - but Humidity is already used,
  - so now we consider Outlook only.

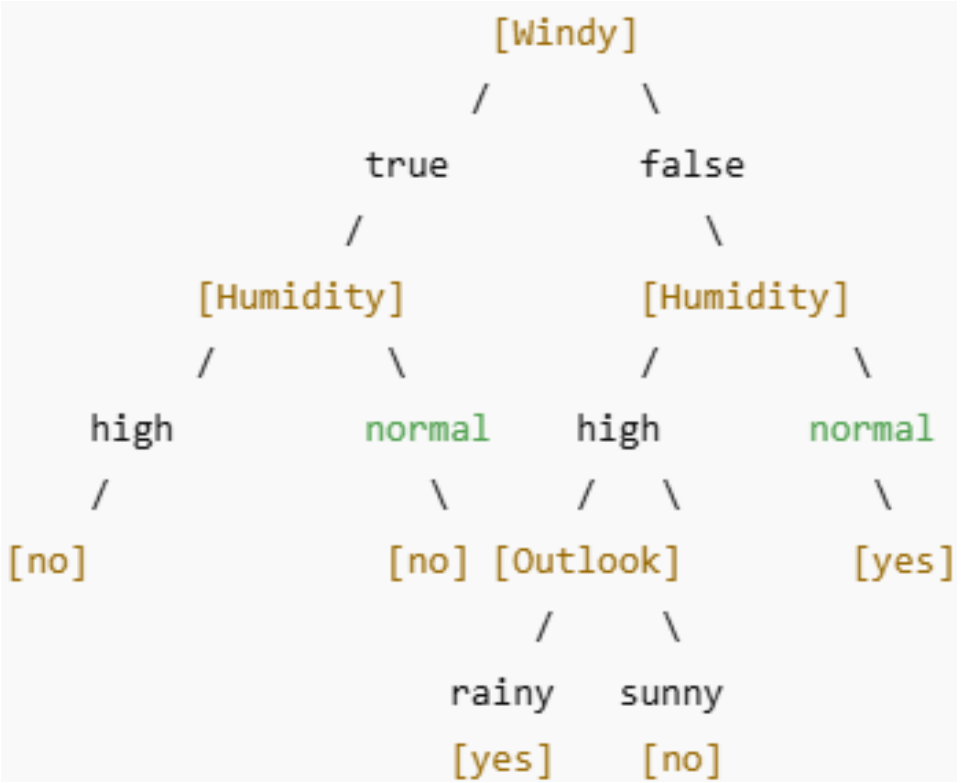
Step 3: Build the Tree (Tree-1)

- Try splitting on Outlook (for clarity):
  - Outlook = rainy: D4, D4 → Play = yes → Pure

Instance	Outlook	Temp	Humidity	Play
D4	rainy	mild	high	yes
D4	rainy	mild	high	yes

  - Outlook = sunny: D8, D1 → Play = no → Pure

Instance	Outlook	Temp	Humidity	Play
D8	sunny	mild	high	no
D1	sunny	hot	high	no



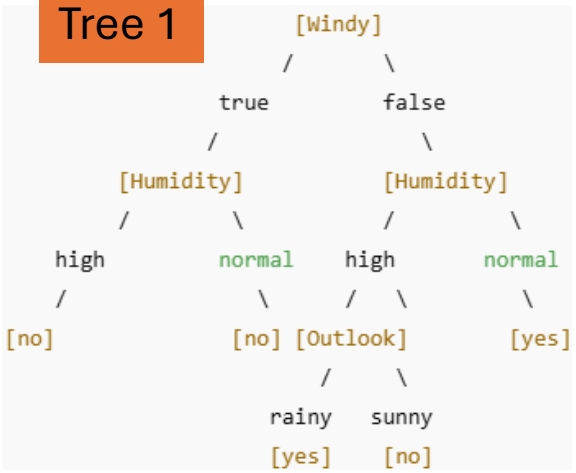
Building of other trees Each Tree

- **Tree 2:**
  - [D1, D3, D6, D6, D13, D11, D14, D7, D2, D3, D9, D12, D12, D10]
  - Repeated: D3, D6, D12
- **Tree 3:**
  - [D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D2, D3, D4]
  - Repeated:
- **Tree 4:**
  - [D1, D1, D1, D2, D2, D3, D5, D5, D6, D7, D8, D9, D10, D14]
  - Repeated:
- **Tree 5:**
  - [D4, D4, D8, D8, D13, D12, D11, D9, D5, D6, D7, D2, D3, D1]
  - Repeated:

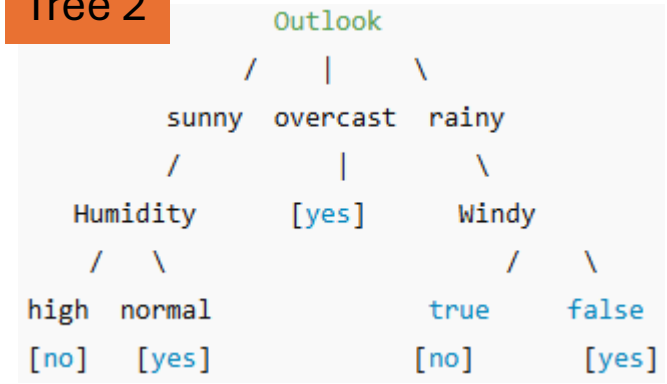
- Follow the same process as in case of Tree 1 to make Tree-2 to tree-5

All Decision Trees: Tree-1 to Tree-5

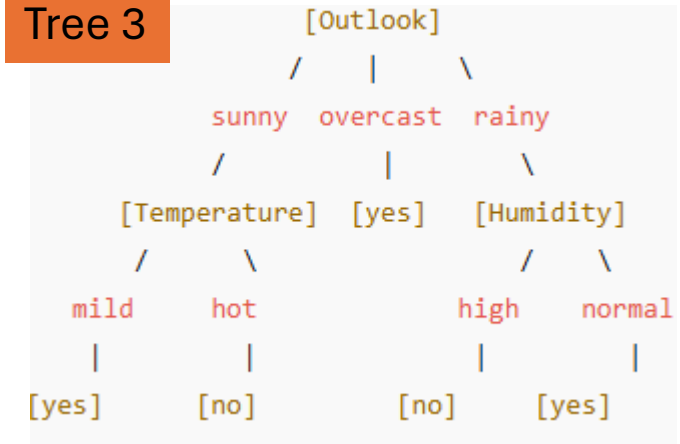
Tree 1



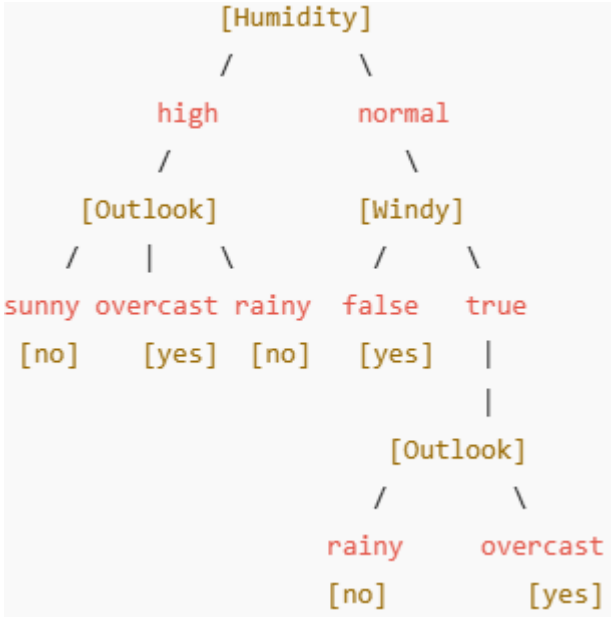
Tree 2



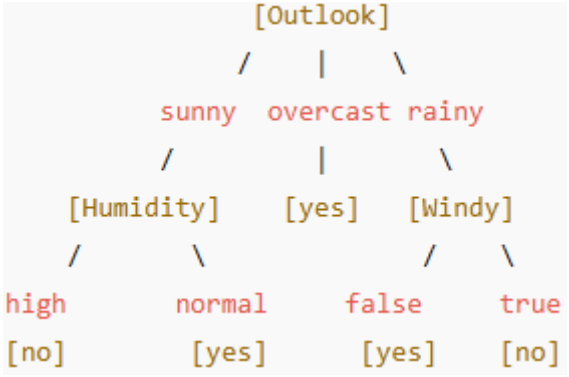
Tree 3



Tree 4



Tree 5



Classification using Random Forest

Outlook	Temperature	Humidity	Windy
sunny	mild	high	false

Tree	Prediction
1	no
2	no
3	yes
4	yes
5	no

→ Votes: yes = 2, no = 3

✅ Random Forest Final Prediction:

Play = no (by majority voting)

Instance	Outlook	Temp	Humidity	Windy	Play
D2	sunny	hot	high	true	no
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D11	sunny	mild	normal	true	yes
D8	sunny	mild	high	false	no
D1	sunny	hot	high	false	no
D6	rainy	cool	normal	true	no
D9	sunny	cool	normal	false	yes
D14	rainy	mild	high	true	no
D10	rainy	mild	normal	false	yes
D2	sunny	hot	high	true	no
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D10	rainy	mild	normal	false	yes

Step 4 - Repeat for Many Trees

- Build 10, 50, or 100+ trees
- Each learns slightly different pattern
- Increases model stability

Step 5: Bagging - Bootstrap Aggregation (Voting/Avg)

- **What it is:**
  - An ensemble learning method that uses **bootstrapping + model voting/averaging**.
- **How it works:**
  - Create **multiple bootstrap samples** from the original dataset.
  - Train a separate model (e.g., decision tree) on each sample.
  - Combine predictions:
    - For classification: majority vote.
    - For regression: average.
- Use multiple **diverse models** trained on **different bootstrap samples**, and **aggregate** their results
- This improves model stability and accuracy by preventing overfitting.

Step 5: Bagging - Bootstrap Aggregation (Voting/Avg)

- Each tree predicts: Yes/No
- Final prediction by majority voting
- For example, 7 Yes, 3 No → Final = Yes

Advantages & Limitations

- Pros:
  - High accuracy
  - Works well with missing data
  - Resistant to overfitting

- Cons:
  - Slower for large number of trees
  - Less interpretable than single tree

End