Lecture 2.6

- Decision Tree

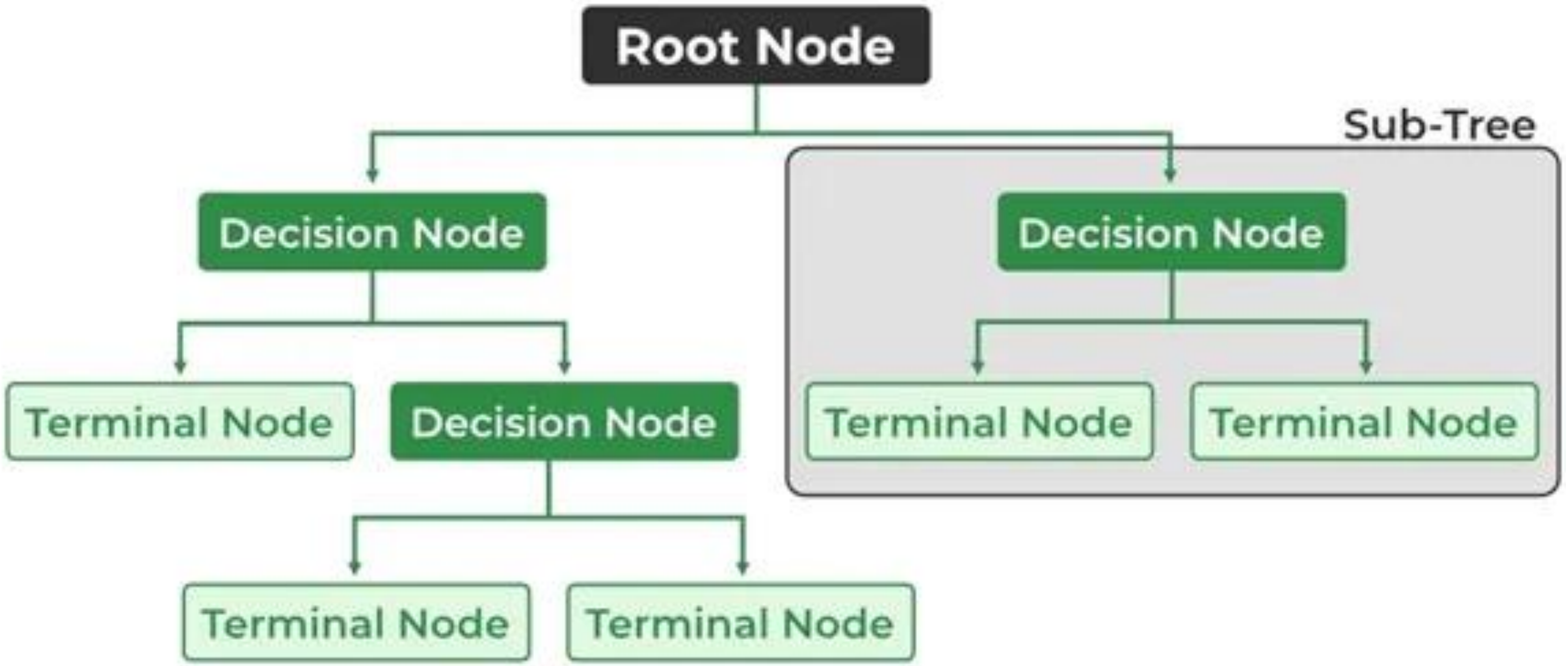
Decision Tree Introduction

- **Decision Tree** is a supervised Machine learning algorithms used for both regression and classification problem statement
- It uses the tree representation to solve a problem in which
  - each **node** represents an **attribute**
  - each **link** represents a **decision rule**
  - each **leaf** represents an **outcome**(categorical or continuous value)

Decision Tree Terminologies

- **Root Node-** It is the topmost node in the tree, which represent the complete dataset
- **Decision/Internal Node-** Decision nodes are nothing but the result in the splitting of data into multiple data segments and main goal is to have the children nodes with maximum homogeneity or purity
- **Leaf/Terminal Node-** This node represent the data section having highest homogeneity

Decision Tree Image



Decision Tree Algorithm: ID3

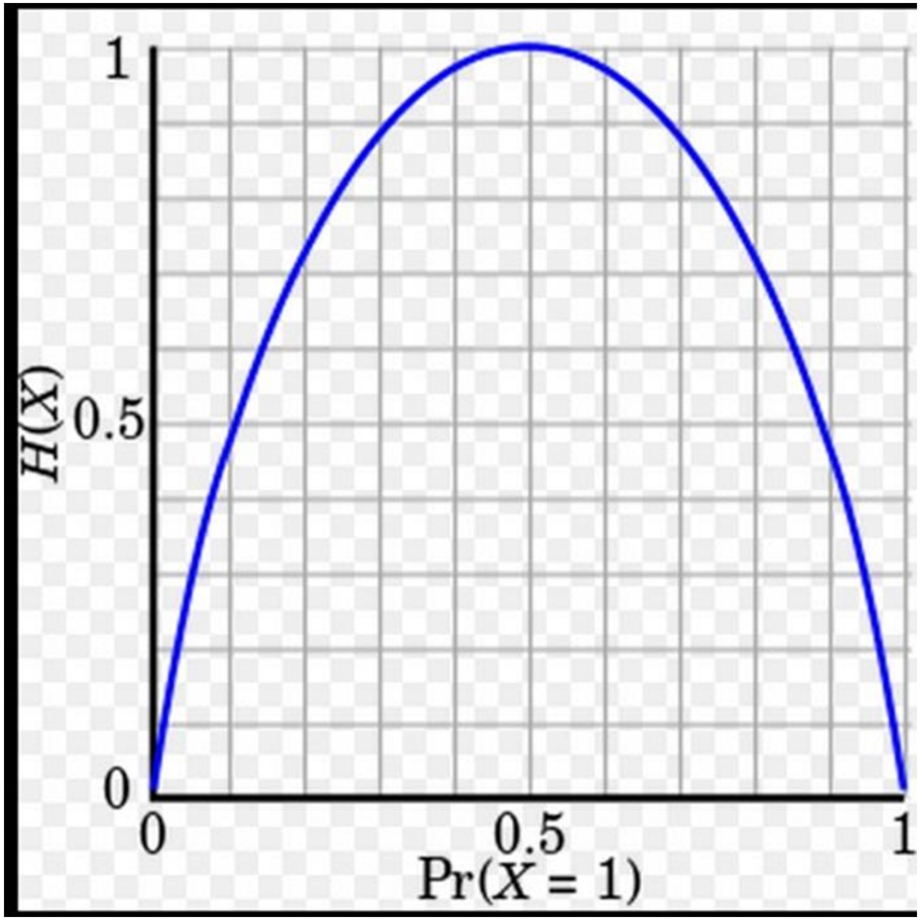
- **The ID3 algorithm (Iterative Dichotomiser 3)** is used to create decision trees by employing the following steps:
  - It calculates **information gain** for each feature and **chooses the one with the highest information gain** as the root
  - **Recursively partitions** the data based on the selected feature
  - **Stops** when all instances in a subset belong to a **single class** or **other stopping criteria are met**

Entropy

- When the number of either yes OR no is zero (that is the node is pure) the information is zero.
- When the number of yes and no is equal, the information reaches its maximum because we are very uncertain about the outcome.
- Complex scenarios: the measure should be applicable to a multiclass situation, where a multi-staged decision must be made

$$E = -\sum p(x) \log p(x)$$

Entropy



How to compute Information Gain:

- **Information gain** is denoted by **IG(S,A)** for a set **S** is the effective change in **entropy** after deciding on a particular attribute **A**
- It measures the relative change in entropy with respect to the independent variables

$$IG(S,A) = E(S) - E(S,A)$$

Or

$$IG(S,A) = E(S) - \sum p(A)H(A)$$

Example 1

- Forecast whether the match will be played or not according to the weather condition.

Day	Outlook	Temp.	Humidity	Wind	Decision
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	Cool	Normal	Strong	No
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

Solution 1 Step 1

- The initial step is to calculate E(S), the Entropy of the current state
- In the above example, we can see in total there are 5 No's and 9 Yes's

$$Entropy(S) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}$$

$$Entropy(S) = -\left(\frac{9}{14}\right)\log_2\left(\frac{9}{14}\right) - \left(\frac{5}{14}\right)\log_2\left(\frac{5}{14}\right)$$

$$= 0.940$$

Solution 1 Step 2

- Now, the next step is to choose the attribute that gives us highest possible **Information Gain**
- Here, attribute 'Wind' takes two possible values in the sample data, hence x = {Weak, Strong} We'll have to calculate

1. $H(S_{weak})$
2. $H(S_{strong})$
3. $P(S_{weak})$
4. $P(S_{strong})$
5. $H(S) = 0.94$ which we had already calculated in the previous example

Solution 1 Step 2: Wind1

- Amongst all the 14 examples we have **8 places where the wind is weak** and 6 where the wind is Strong

$$P(S_{weak}) = \frac{Number\ of\ Weak}{Total}$$

$$= \frac{8}{14}$$

$$P(S_{strong}) = \frac{Number\ of\ Strong}{Total}$$

$$= \frac{6}{14}$$

Solution 1 Step 2: Wind2

- Now, out of the 8 Weak examples, 6 of them were 'Yes' for Play Golf and 2 of them were 'No' for 'Play Golf'

$$Entropy(S_{weak}) = -\left(\frac{6}{8}\right)\log_2\left(\frac{6}{8}\right) - \left(\frac{2}{8}\right)\log_2\left(\frac{2}{8}\right)$$
$$= 0.811$$

- Similarly, out of 6 Strong examples, we have 3 examples where the outcome was 'Yes' for Play Golf and 3 where we had 'No' for Play Golf.

$$Entropy(S_{strong}) = -\left(\frac{3}{6}\right)\log_2\left(\frac{3}{6}\right) - \left(\frac{3}{6}\right)\log_2\left(\frac{3}{6}\right)$$
$$= 1.000$$

Solution 1 Step 2: Wind3

$$IG(S, Wind) = H(S) - \sum_{i=0}^{n} P(x) * H(x)$$

$$IG(S, Wind) = H(S) - P(S_{weak}) * H(S_{weak}) - P(S_{strong}) * H(S_{strong})$$

$$= 0.940 - \left(\frac{8}{14}\right)(0.811) - \left(\frac{6}{14}\right)(1.00)$$

$$= 0.048$$

Solution 1 Step 3

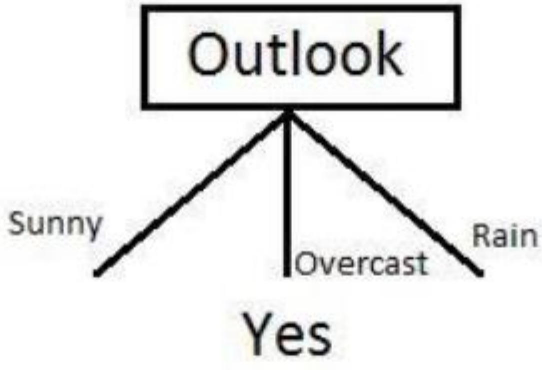
$IG(S, Outlook) = 0.246$

$IG(S, Temperature) = 0.029$

$IG(S, Humidity) = 0.151$

$IG(S, Wind) = 0.048$ (Previous example)

We can clearly see that IG(S, Outlook) has the highest information gain of 0.246, hence we chose Outlook attribute as the root node.



Solution 1 Step 4

- Now that we’ve used Outlook, we’ve got three of them remaining Humidity, Temperature, and Wind
- And, we had three possible values of Outlook: Sunny, Overcast, Rain
- Where the Overcast node already ended up having leaf node ‘Yes’, so we’re left with two subtrees to compute: Sunny and Rain

Solution 1 Step 4: Overcast

Overcast outlook on decision

Basically, decision will always be yes if outlook were overcast.

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

Solution 1 Step 4: Sunny

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

$$H(S_{sunny}) = \left(\frac{3}{5}\right)\log_2\left(\frac{3}{5}\right) - \left(\frac{2}{5}\right)\log_2\left(\frac{2}{5}\right) = 0.96$$

$$IG(S_{sunny}, Humidity) = 0.96$$

$$IG(S_{sunny}, Temperature) = 0.57$$

$$IG(S_{sunny}, Wind) = 0.019$$

Solution 1 Step 4: Rain

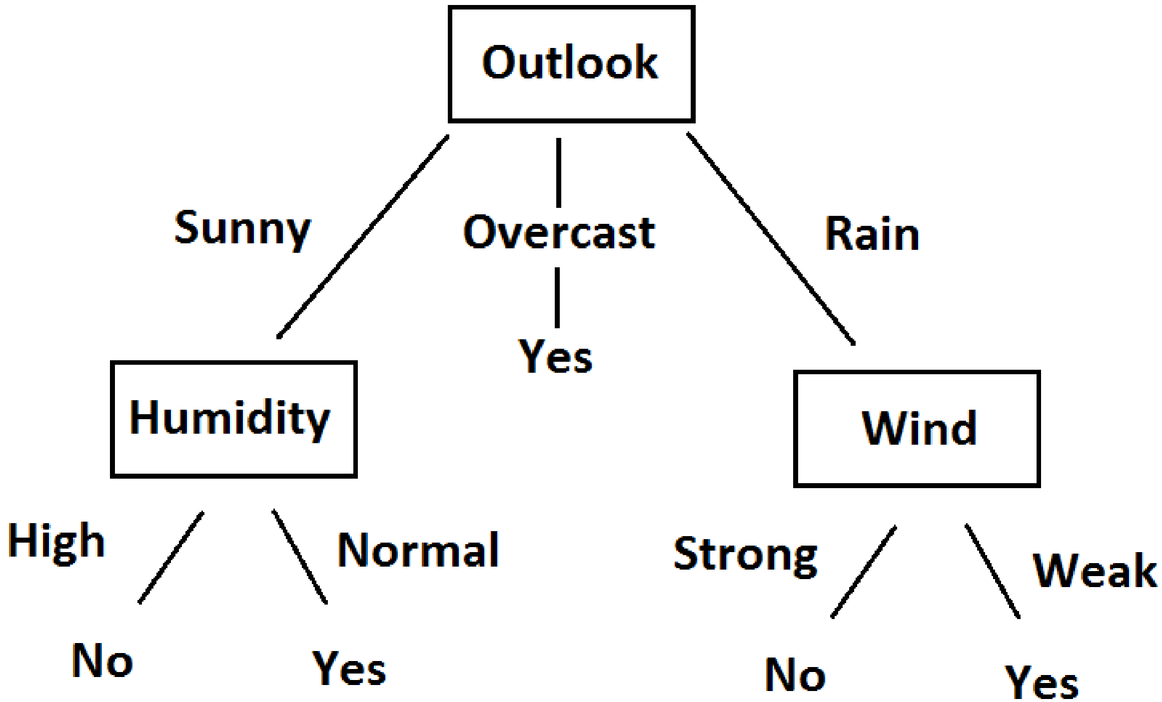
Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes

1- Gain(Outlook=Rain | Temperature) = 0.01997309402197489

2- Gain(Outlook=Rain | Humidity) = 0.01997309402197489

3- Gain(Outlook=Rain | Wind) = 0.9709505944546686

Final


Outlook
Sunny
Overcast
Rain
Yes
Humidity
Wind
High
Normal
Strong
Weak
No
Yes
No
Yes

Problem of Overfitting in Decision Trees

- Overfitting occurs when a decision tree learns patterns that are specific to the training data but do not generalize well to unseen data
- This results in high accuracy on the training set but poor performance on validation or test sets

Causes of Overfitting

- **Too Deep Trees**: The tree grows to fit every detail in the training data, including noise
- **Small Subsets**: When the training data is partitioned into very small subsets, splits may capture irrelevant patterns
- **Noisy Data:** Errors or outliers in the dataset can lead to over-complex models.

Solutions to Overfitting

- **Pre-pruning (Early Stopping)**: Pre-pruning halts the tree-building process early, avoiding over-complex trees
  - Set Maximum Depth: Restrict the depth of the tree
  - Minimum Samples per Split: Require a minimum number of samples for a split to occur
  - Minimum Information Gain: Stop splitting if the information gain is below a threshold
- **Post-pruning (Pruning After Training)**: Post-pruning involves growing the full tree and then trimming branches that do not improve generalization
  - Reduced Error Pruning: Evaluate the effect of removing a branch on validation accuracy, Retain the branch only if it improves accuracy
  - Cost-Complexity Pruning: Minimize a tradeoff between tree complexity and classification accuracy

Gini Index

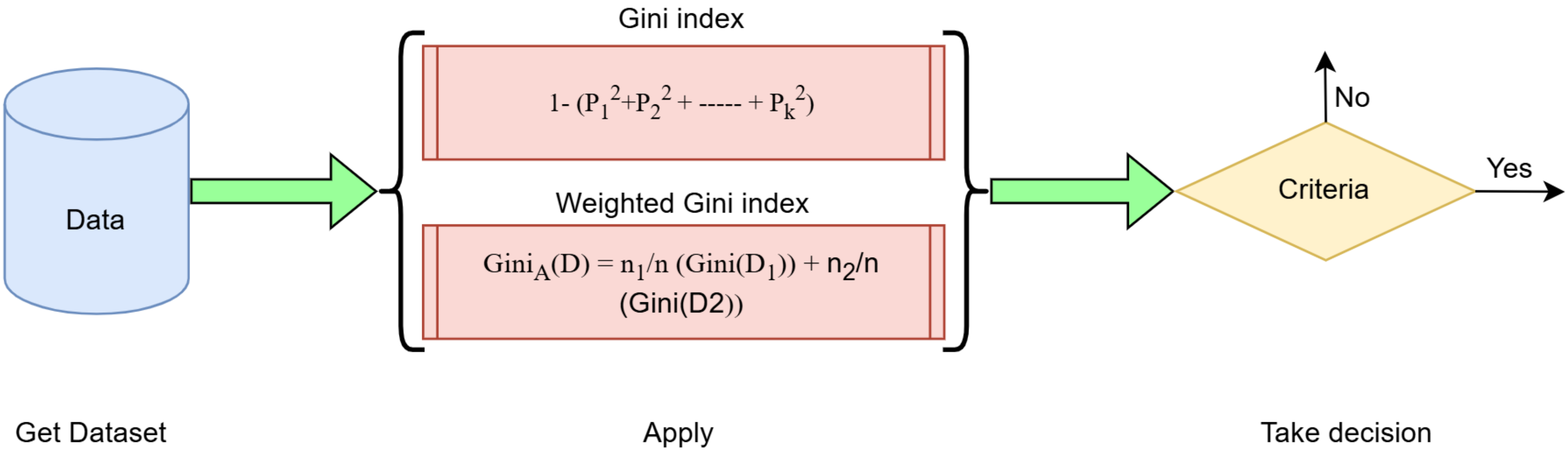
- The **Gini index** measures impurity or inequality frequently used in decision tree algorithms
- It quantifies the probability of misclassifying a randomly chosen element if it were randomly labeled according to the distribution of labels in a particular node

$$Gini\ Index = 1 - (p_1^2 + p_2^2 + \cdots + p_n^2)$$

Where, $p_1, p_2 \ldots, p_n$ are the probabilities of each class in the node

- The Gini impurity ranges between 0 and 1, where 0 represents a pure dataset and 1 represents a completely impure dataset

Construction of Decision Tree



- The $Gini\ (D)$represents the weighted Gini index for the entire dataset D
- It's a measure of impurity or inequality in the dataset, considering the weighted average of the impurities of two subsets $D_1$and D_2

Example 2

- Forecast whether the match will be played or not according to the weather condition using Decision Tree (Gini-index)

Day	Outlook	Temp.	Humidity	Wind	Decision
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	Cool	Normal	Strong	No
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

Solution: Step 1: Analyze the given data and calculate the Gini index for each attribute at the first step

Outlook

Sunny
Overcast
Rainy

Temperature

Hot
Mild
Cool

Humidity

High
Normal

Windy

No
Yes

Play

No
Yes

Step 2

Calculate Gini index for **Outlook**

For Sunny:

- Play=No count: 3
- Play=Yes count: 2
- Gini index for Sunny:
  - $= 1 - (2/5)^2 - (3/5)^2$
  - $= 1 - 4/25 - 9/25$
  - $= 1 - 13/25$
  - $= 12/25$

For Overcast:

- Play=No count: 0
- Play=Yes count: 4
- Gini index for Overcast:
  - $= 1 - (0/4)^2 - (4/4)^2$
  - $= 1 - 0/16 - 16/16$
  - $= 0$

For Rainy:

- Play=No count: 3
- Play=Yes count: 2
- Gini index for Rainy:
  - $= 1 - (3/5)^2 - (2/5)^2$
  - $= 1 - 9/25 - 4/25$
  - $= 12/25$

Calculate weighted Gini index for **Outlook**

$$(5/14) * (12/25) + (4/14) * 0 + (5/14) * (12/25) = 0.342$$

Calculate Gini index for **Windy**

For No:

- Play=No count: 2
- Play=Yes count: 6
  - $= 1 - (6/8)^2 - (2/8)^2 = 0.375$

For Yes:

- Play=No count: 3
- Play=Yes count: 3
  - $= 1 - (3/6)^2 - (3/6)^2 = 0.5$

Calculate weighted Gini index for **Windy**

$$(8/14) * (3/8) + (6/14) * (1/2) = 0.428$$

Calculate Gini index for **Temperature**

For Hot:

- Play=No count: 2
- Play=Yes count: 2
- Gini index for Hot:
  - $= 1 - (2/4)^2 - (2/4)^2 = 0.5$

For Mild:

- Play=No count: 2
- Play=Yes count: 4
- Gini index for Mild:
  - $= 1 - (2/6)^2 - (4/6)^2 = 4$

For Cool:

- Play=No count: 1
- Play=Yes count: 3
- Gini index for Cool:
  - $= 1 - (1/4)^2 - (3/4)^2 = 0.375$

Calculate weighted Gini index for **Temperature**

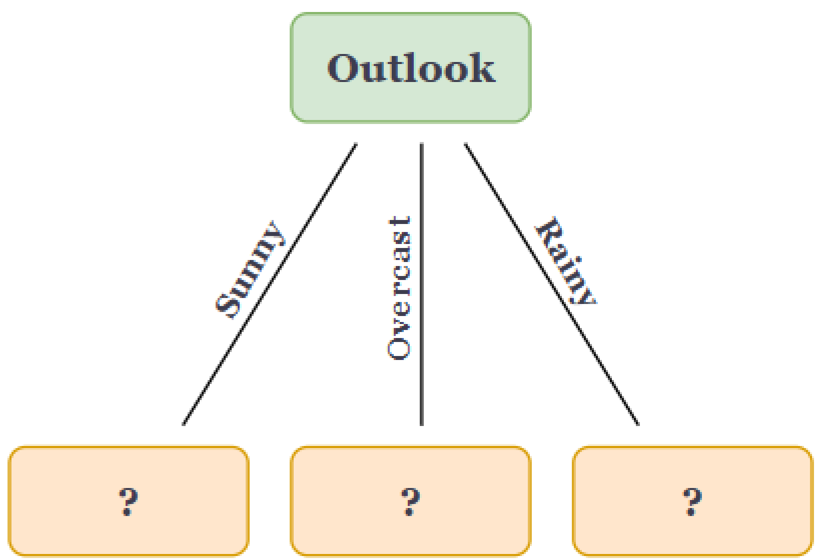
$$(4/14) * 0.5 + (6/14) * (4/9) + (4/14) * (0.375) = 0.4404$$

Take a decision base on the calculated result for the root node

Now we have the Gini index calculations for each attribute at the first step:

- Outlook: 0.3429
- Temperature: 0.4404
- Humidity: 0.4898
- Windy: 0.4286

The attribute with the lowest Gini index is Outlook, so it would be selected as the root of the decision tree in the next step.



Step 3

Extract the dataset under the selected root node for each subtree.

- Outlook -> Sunny

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	No	No
Sunny	Hot	High	Yes	No
Sunny	Mild	High	No	No
Sunny	Cool	Normal	No	Yes
Sunny	Mild	Normal	Yes	Yes

- Outlook -> Overcast

Outlook	Temperature	Humidity	Windy	Play
Overcast	Hot	High	No	Yes
Overcast	Cool	Normal	Yes	Yes
Overcast	Mild	High	Yes	Yes
Overcast	Hot	Normal	No	Yes

- Outlook -> Rainy

Outlook	Temperature	Humidity	Windy	Play
Rainy	Mild	High	No	Yes
Rainy	Cool	Normal	No	Yes
Rainy	Cool	Normal	Yes	No
Rainy	Mild	Normal	No	Yes
Rainy	Mild	High	Yes	No

Repeat **Step1**, **Step2** and **Step3** for each subtree until we reach the leaf node

Here we have three sub branches:

- Sunny
- Overcast
- Rainy

After repeating step1, step2 and step3, we will find these calculated results for leaf node

Outlook -> Sunny

- Temperature: 0.44
- Humidity: 0
- Windy: 0.44

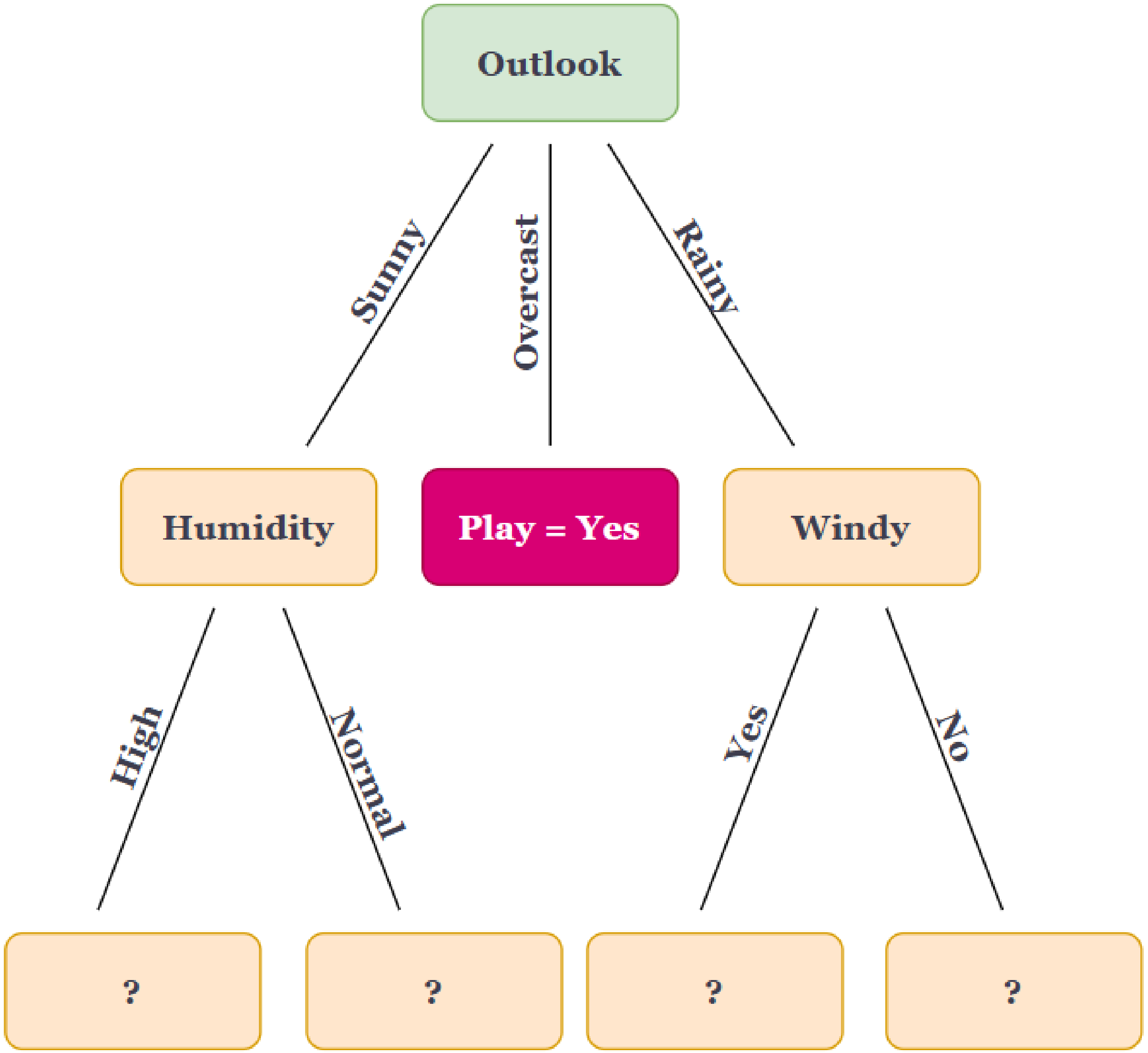
Outlook -> Overcast

- Temperature: 0
- Humidity: 0
- Windy: 0

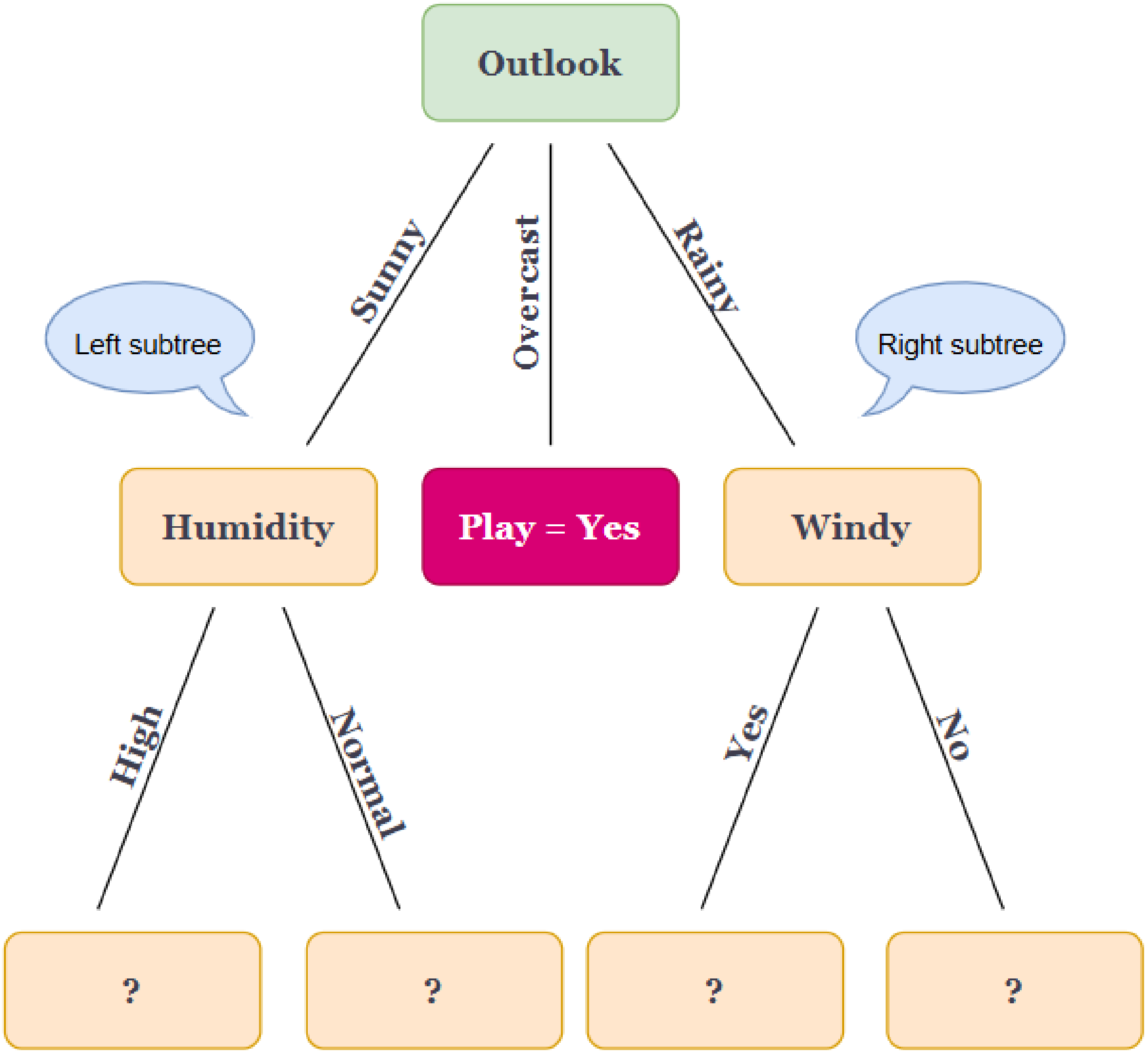
Outlook -> Rainy

- Temperature: 0.464
- Humidity: 0.464
- Windy: 0

Tree at this moment



Repeat the same steps for the subtrees



Extract the dataset under the selected root node for each attribute.

- Humidity -> High

Humidity	Temperature	Windy	Play
High	Hot	No	No
High	Hot	Yes	No
High	Mild	No	No

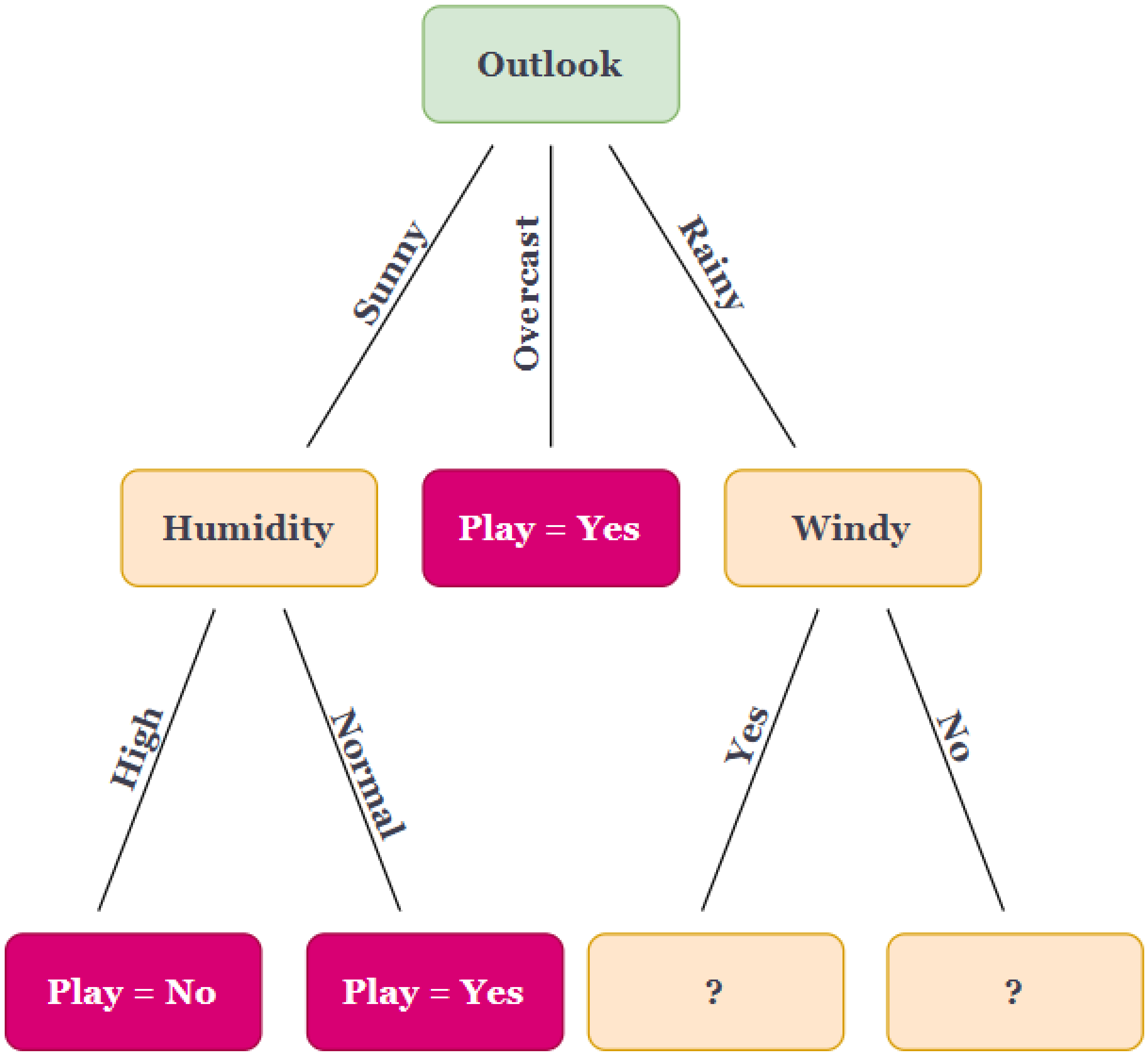
- Humidity -> Normal

Humidity	Temperature	Windy	Play
Normal	Cool	No	Yes
Normal	Mild	Yes	Yes

We can repeat step1 , step2 and step3 for above dataset of we can observe that for every case under

- Humidity -> High
  - Play= No
- Humidity -> Normal
  - Play = Yes

Tree at this moment



Extract the dataset under the selected root node for each attribute.

- Windy -> Yes

Windy	Temperature	Humidity	Play
Yes	Mild	High	No
Yes	Cool	Normal	No
Yes	Mild	Normal	No

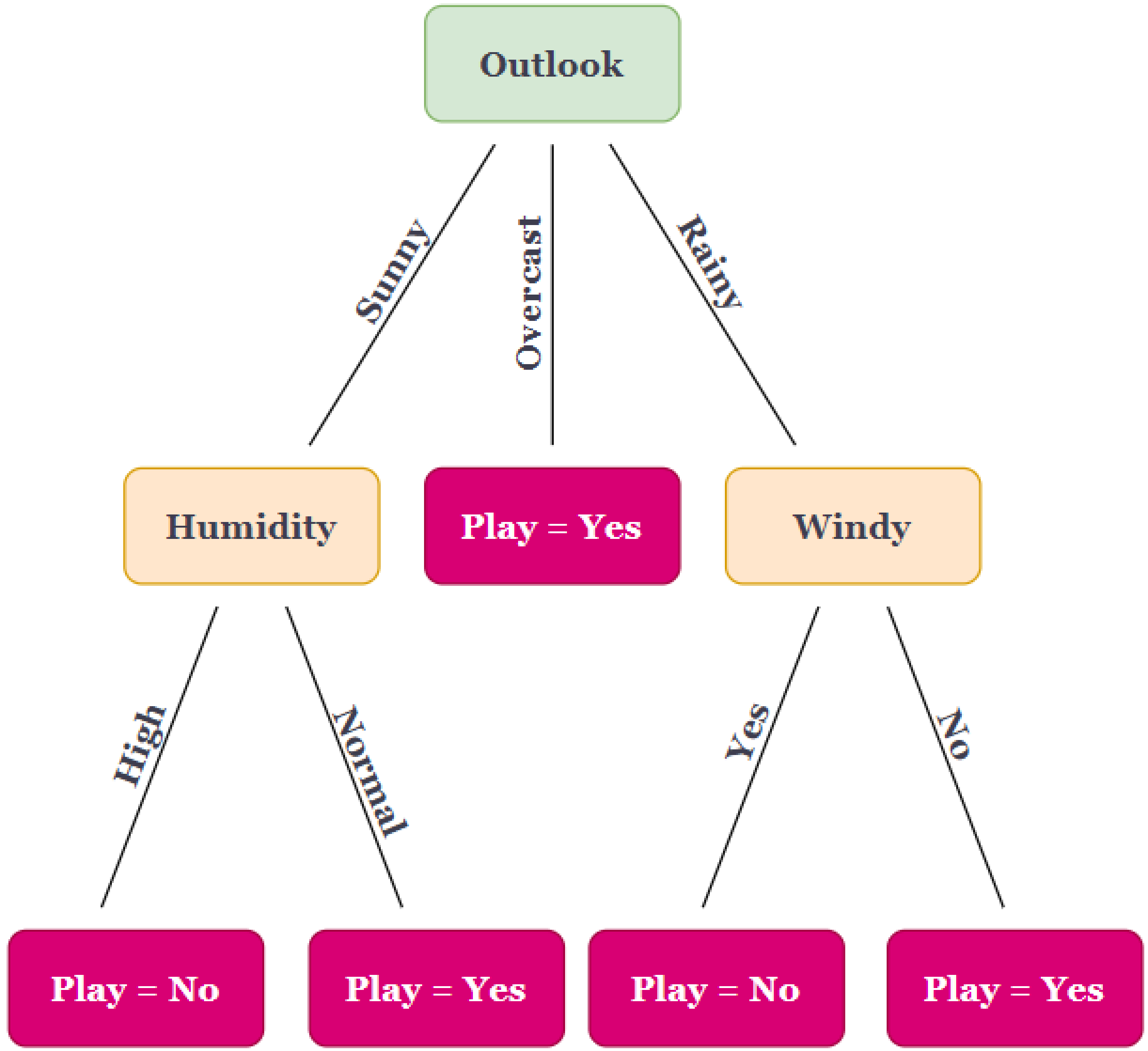
- Windy -> No

Windy	Temperature	Humidity	Play
No	Mild	High	Yes
No	Cool	Normal	Yes
No	Mild	Normal	Yes

We can repeat step1 , step2 and step3 for above dataset of we can observe that for every case under

- Windy -> Yes
  - Play= No
- Windy -> No
  - Play = Yes

Final Tree



Outlook
Sunny
Overcast
Rainy
Humidity
Play = Yes
Windy
High
Normal
Yes
No
Play = No
Play = Yes
Play = No
Play = Yes