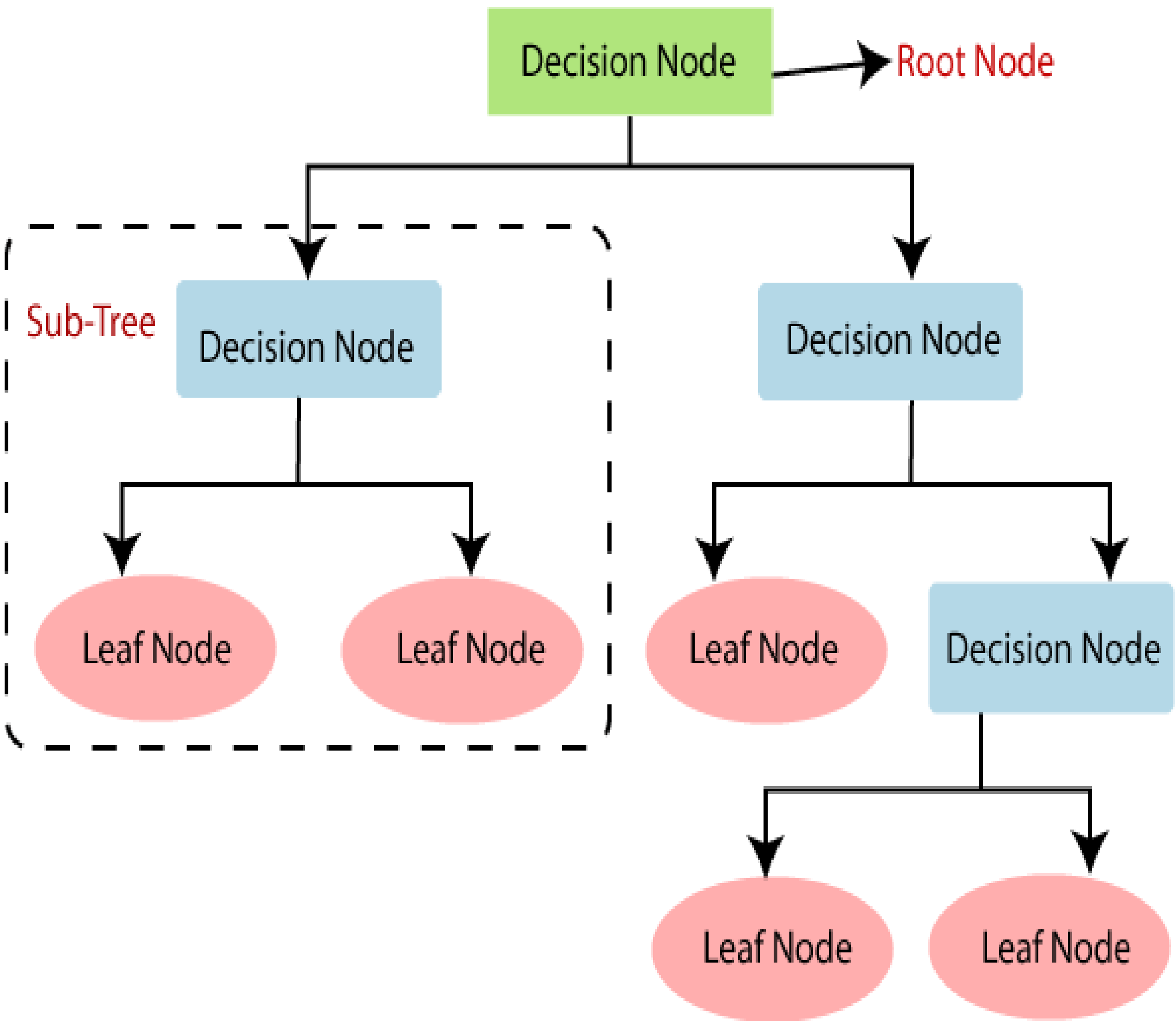
Decision Trees

Decision Tree algorithm

- Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.
- It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.**
- Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.

- ***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.



Decision Tree Terminologies

- **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets
- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- **Branch/Sub Tree:** A tree formed by splitting the tree.
- **Pruning:** Pruning is the process of removing the unwanted branches from the tree.
- **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

Algorithm

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Decision Tree

- A decision tree is a tree where each node represents a **Feature** (Attribute).
- Each link or branch represents decision (Rule) and each leaf represents an outcome
- Algorithm -. ID3 and CART algorithm
- **ID3 Algorithm:** a) Information Gain b) Entropy Function
- **CART Algorithm** a) Gini Index

Types of Decision Tree Algorithms

- The different decision tree algorithms are listed below:
- ID3(Iterative Dichotomiser 3)
- C4.5
- CART(Classification and Regression Trees)
- CHAID (Chi-Square Automatic Interaction Detection)
- MARS(Multivariate Adaptive Regression Splines)

Entropy

- **Entropy**: **measures its impurity or disorder.**
- Entropy is the measure of the degree of randomness or uncertainty in the dataset. In the case of classifications, It measures the randomness based on the distribution of class labels in the dataset.
- If a dataset is perfectly pure (all data points belong to the same class), the entropy is 0. If the classes are evenly distributed, the entropy is at its maximum.
- Mathematically, entropy is calculated using the formula:

$$H(D) = \Sigma_{i=1}^{n} p_i \, log_2(p_i)$$

Where pi represents the proportion of data points belonging to class i in the dataset D.

The base 2 logarithm is used to calculate entropy, resulting in entropy values measured in bits.

Information Gain

- Information gain measures the reduction in entropy or variance that results from splitting a dataset based on a specific property.
- It is used in decision tree algorithms to determine the usefulness of a feature by partitioning the dataset into more homogeneous subsets with respect to the class labels or target variable.
- The higher the information gain, the more valuable the feature is in predicting the target variable.
- Mathematically, information gain is calculated as follows:

$$\text{Information Gain}(\mathrm{H}, \mathrm{A}) = H - \sum \frac{|H_V|}{|H|} H_v$$

- A is the specific attribute or class label
- $|H|$ is the entropy of dataset sample S
- $|H_V|$ is the number of instances in the subset S that have the value v for attribute A

Make a decision tree that predicts whether tennis will be played on that day?

Step 1: Choose a Root Node - HOW TO CHOOSE A ROOT NODE?

The attribute that best classifies the training data, use this attribute at the root of the tree.

Outlook	Temperature	Humidity	Wind	PlayTennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rainy	Mild	High	Weak	Yes
Rainy	Cool	Normal	Weak	Yes
Rainy	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rainy	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rainy	Mild	High	Strong	No

ID3 Algorithm

- Calculate **Entropy** (Amount of uncertainity in dataset):

$$\textbf{Entropy} = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

- Calculate **Average Information:**

$$\textbf{I (Attribute)} = \sum \frac{\boldsymbol{p}_{\mathrm{i}}+\mathrm{ni}}{\boldsymbol{p}+\boldsymbol{n}}\ \textbf{Entropy (A)}$$

- Calculate **Information Gain:** (Difference in Entropy before and after splitting the dataset on attribute A)

Gain = Entropy (S) – I (Attribute)

1. Compute the **entropy** for the dataset **Entropy(S)**

2. For every Attribute/Feature:

   - Calculate Entropy for all other values **Entropy(A)**

   - Take **Average Information Entropy** for the current attribute.

   - Calculate **Gain** for the current attribute

3. Pick the **Highest Gain Attribute.**

4. **Repeat** until we get the tree we desired**.**

1.

Outlook	Temperature	Humidity	Wind	PlayTennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rainy	Mild	High	Weak	Yes
Rainy	Cool	Normal	Weak	Yes
Rainy	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rainy	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rainy	Mild	High	Strong	No

Class label is PlayTennis **PlayTennis = "Yes"** **p = 9** **Total = 14**

PlayTennis = "No" **n = 5**

- Calculate **Entropy(S):**

$$\textbf{Entropy} = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

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

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

$$= \mathbf{0.940}$$

- For each Attribute: (**Outlook**)
  - **Calculate entropy for each values. ie for “Sunny”, “Rainy” and “Overcast”**

Outlook	PlayTennis
Sunny	No
Sunny	No
Sunny	No
Sunny	Yes
Sunny	Yes

Outlook	PlayTennis
Rainy	No
Rainy	No
Rainy	Yes
Rainy	Yes
Rainy	Yes

Outlook	PlayTennis
Overcast	Yes
Overcast	Yes
Overcast	Yes
Overcast	Yes

$$-\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right)-\frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

Outlook	p	n	Entropy	
Sunny	2	3	**0.971**	**$-2/5 \log_2(2/5) – 3/5 \log_2 (3/5) = 0.4*1.322+ 0.6*0.737 = 0.53+0.44 = 0.97$ (ln(2/5)/ln(2))**
Rainy	3	2	**0.971**	**$-3/5 \log_2(3/5) – 2/5 \log_2 (2/5) = 0.6*0.737 + 0.4*1.322 = 0.44 + 0.53 = 0.97$ (ln(3/5)/ln(2))**
Overcast	4	0	**0**	**$-4/4\log_2 (1) = 0$**

- Calculate Average Information Entropy:
- I (Outlook) = $\frac{p_{Sunny} + nS_{unny}}{p+n}$ Entropy (Outlook = Sunny) +

$\frac{p_{Rainy} + nR_{ainy}}{p+n}$ Entropy (Outlook = Rainy) +

$\frac{p_{Overcast} + nO_{vercast}}{p+n}$ Entropy (Outlook = Overcast)

I (Outlook) = $\frac{2+3}{9+5}$ * 0.971 + $\frac{3+2}{9+5}$ * 0.971 + $\frac{4+0}{9+5}$ * 0 = 0.693

- Calculate **Gain** : attribute is Outlook

Gain = Entropy(S) – I (Attribute)

Entropy(S) = 0.940, I (Outlook) = 0.693

Gain (Outlook) = 0.940 – 0.693 = 0.247

- For each Attribute: (**Temperature**)
  - **Calculate entropy for each values. ie for "Hot", "Cold" and "Mild"**

Outlook	PlayTennis
Hot	No
Hot	No
Hot	Yes
Hot	Yes

Outlook	PlayTennis
Cool	No
Cool	Yes
Cool	Yes
Cool	Yes

Outlook	PlayTennis
Mild	Yes
Mild	Yes
Mild	Yes
Mild	Yes
Mild	No
Mild	No

Temperature	p	n	Entropy
Hot	2	2	**1**
Cool	3	1	**0.811**
Mild	4	2	**0.918**

- Calculate Average Information Entropy:
- I (Temperature) = $\frac{p_{Hot}+nHo_t}{p+n}$ Entropy (Temperature = Hot) +

  $\frac{p_{Cool}+nCo_{ol}}{p+n}$ Entropy (Temperature = Cool) +

  $\frac{p_{Mild}+nMi_{ld}}{p+n}$ Entropy (Temperature = Mild)

I (Temperature) = $\frac{2+2}{9+5} * 1 + \frac{3+1}{9+5} * 0.811 + \frac{4+2}{9+5} * 0.918 = 0.911$

- Calculate **Gain** : attribute is Temperature

Gain = Entropy(S) – I (Attribute)

Entropy(S) = 0.940, I (Temperature) = 0.911

Gain (Temperature) = 0.940 – 0.911 = 0.029

- For each Attribute: (**Humidity**)
  - **Calculate entropy for each values. ie for "Normal" and "High"**

Outlook	PlayTennis
Normal	No
Normal	Yes
Normal	Yes
Normal	Yes
Normal	Yes
Normal	Yes
Normal	Yes

Outlook	PlayTennis
High	Yes
High	Yes
High	Yes
High	No
High	No
High	No
High	No

Humidity	p	n	Entropy
Normal	6	1	**0.591**
High	3	4	**0.985**

- Calculate Average Information Entropy:
- **I (Humidity) =** $\frac{p_{Normal}+nNo_{rmal}}{p+n}$ **Entropy (Humidity = Normal) +**

$\frac{p_{High}+nHi_{gh}}{p+n}$ **Entropy (Humidity = High)**

I (Humidity) = $\frac{6+1}{9+5}$ *** 0.591 +** $\frac{3+4}{9+5}$ *** 0.985 = 0.788**

- Calculate **Gain :** attribute is Temperature

Gain = Entropy(S) – I (Attribute)

Entropy(S) = 0.940,   I (Humidity) = 0.788

Gain (Humidity) = 0.940 – 0.788 = 0.152

- For each Attribute: (**Wind**)
  - **Calculate entropy for each values. ie for "Strong" and "Weak"**

Outlook	**PlayTennis**
Weak	No
Weak	No
Weak	Yes
Weak	Yes
Weak	Yes
Weak	Yes
Weak	Yes
Weak	Yes

Outlook	**PlayTennis**
Strong	Yes
Strong	Yes
Strong	Yes
Strong	No
Strong	No
Strong	No

Wind	**p**	**n**	**Entropy**
Strong	3	3	**1**
Weak	6	2	**0.811**

- Calculate Average Information Entropy:
- **I (Wind) =** $\frac{p_{Strong}+n_{\ Strong}}{p+n}$ **Entropy (Wind = Strong) +**

  $\frac{p_{Weak}+nWe_{ak}}{p+n}$ **Entropy (Wind = Weak)**

I (Wind) = $\frac{3+3}{9\ +5}$ *** 1 +** $\frac{6+2}{9\ +5}$ *** 0.811 = 0.892**

- Calculate **Gain** : attribute is Temperature

Gain = Entropy(S) – I (Attribute)

Entropy(S) = 0.940, I (Wind) = 0.892

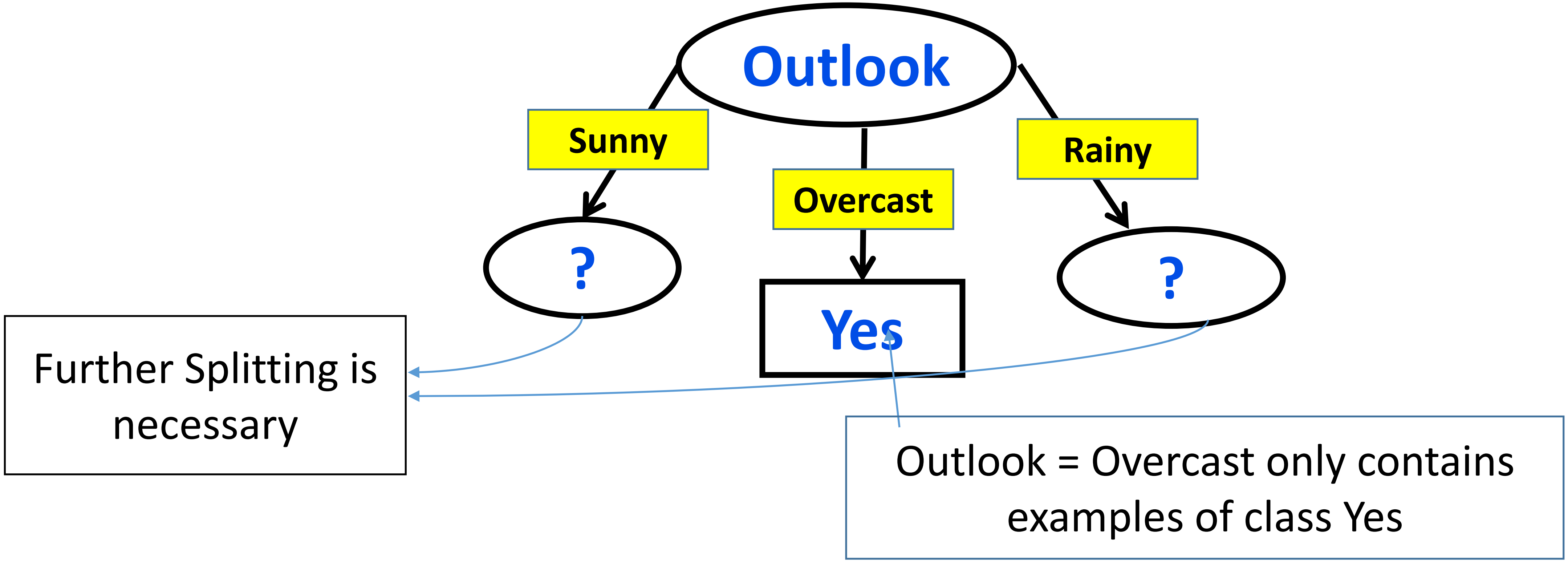
Gain (Wind) = 0.940 – 0.892 = 0.048

- Pick the **HIGHEST GAIN ATTRIBUTE**.

Attributes	**Gain**
Outlook	**0.247**
Temperature	**0.029**
Humidity	**0.152**
Wind	**0.048**

- ROOT NODE = Outlook

Outlook	Temperature	Humidity	Wind	PlayTennis
Overcast	Hot	High	Weak	Yes
Overcast	Cool	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes



- Repeat the same thing for sub-trees till we get the tree

Outlook	Temperature	Humidity	Wind	PlayTennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes

Outlook = “Sunny”

Outlook	Temperature	Humidity	Wind	PlayTennis
Rainy	Mild	High	Weak	Yes
Rainy	Cool	Normal	Weak	Yes
Rainy	Cool	Normal	Strong	No
Rainy	Mild	Normal	Weak	Yes
Rainy	Mild	High	Strong	No

Outlook = “Rainy”

Outlook	Temperature	Humidity	Wind	PlayTennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes

p = 2   n = 3

Total = 5

- **Entropy:**

$$\text{Entropy} = -\frac{p}{\mathbf{p+n}}\log_2\left(\frac{p}{\mathbf{p+n}}\right) - \frac{n}{\mathbf{p+n}}\log_2\left(\frac{n}{\mathbf{p+n}}\right)$$

$$\text{Entropy}(_{\text{Sunny}}) = -\frac{\mathbf{2}}{\mathbf{2+3}}\log_2\left(\frac{\mathbf{2}}{\mathbf{2+3}}\right) - \frac{\mathbf{3}}{\mathbf{2+3}}\log_2\left(\frac{\mathbf{3}}{\mathbf{2+3}}\right)$$

$$= 0.971$$

- For each Attribute: (**Temperature**)
  - **Calculate entropy for each Temperature. ie for "Hot", "Cool" and "Mild"**

Outlook	Temperature	PlayTennis
Sunny	Cool	Yes
Sunny	Hot	No
Sunny	Hot	No
Sunny	Mild	No
Sunny	Mild	Yes

Temperature	p	n	Entropy
Hot	0	2	**0**
Cool	1	0	**0**
Mild	1	1	**1**

- **Calculate Average Information Entropy: I (Temperature)**

$$= \frac{\mathrm{p}_{Hot}+nHo_{t}}{\mathrm{p}+n} \text{ Ent (Temp = Hot) } + \frac{\mathrm{p}_{Cool}+nCo_{ol}}{\mathrm{p}+n} \text{ Ent(Temp = Cool) } + \frac{\mathrm{p}_{Mild}+nMi_{ld}}{\mathrm{p}+n} \text{ Ent (Temp= Mild)}$$

$$= \frac{2}{5}(0) + \frac{1}{5}(0) + \frac{2}{5}(1) = 0 + 0 + 0.4 = 0.4$$

- **I (Temperature) = 0.4**

- **Calculate Gain :** **Gain (Temperature) = 0.971 – 0.4 = 0.571**

- For each Attribute: (**Humidity**)
  - **Calculate entropy for each Humidity. ie for "Normal" and "High"**

Outlook	Humidity	PlayTennis
Sunny	High	No
Sunny	High	No
Sunny	High	No
Sunny	Normal	Yes
Sunny	Normal	Yes

Humidity	p	n	Entropy
Normal	2	0	0
High	0	3	0

- **Calculate Average Information Entropy:** **I (Humidity) = 0**

- **I (Humidity) =** $\frac{p_{Normal}+nNo_{rmal}}{p+n}$ **Ent(Humi = Normal) +** $\frac{p_{High}+nHi_{gh}}{p+n}$ **Ent (Humi = High)**

$$= \frac{2}{5}(0) + \frac{3}{5}(0) = 0 + 0 = 0$$

- **Calculate Gain :**

Gain (Humidity) = 0.971 – 0 = 0.971

- For each Attribute: (**Wind**)
  - **Calculate entropy for each Humidity. ie for "Strong" and "Weak"**

Outlook	Wind	PlayTennis
Sunny	Weak	No
Sunny	Strong	No
Sunny	Weak	No
Sunny	Weak	Yes
Sunny	Strong	Yes

Wind	p	n	Entropy
Strong	1	1	1
Weak	1	2	0.918

- **Calculate Average Information Entropy:** **I (Wind) = 0.951**

$$\text{I (Wind)} = \frac{\mathrm{p}_{Strong}+n_{\ Strong}}{\mathrm{p}+n}\ \text{Ent (Wind = Strong)} + \frac{\mathrm{p}_{Weak}+nWe_{ak}}{\mathrm{p}+n}\ \text{Ent (Wind = Weak)}$$

$$= \frac{2}{5}(1) + \frac{3}{5}(0.918) = 0.4 + 0.551 = 0.951$$

- **Calculate Gain :** **Gain (Wind) = 0.971 – 0.951 = 0.020**

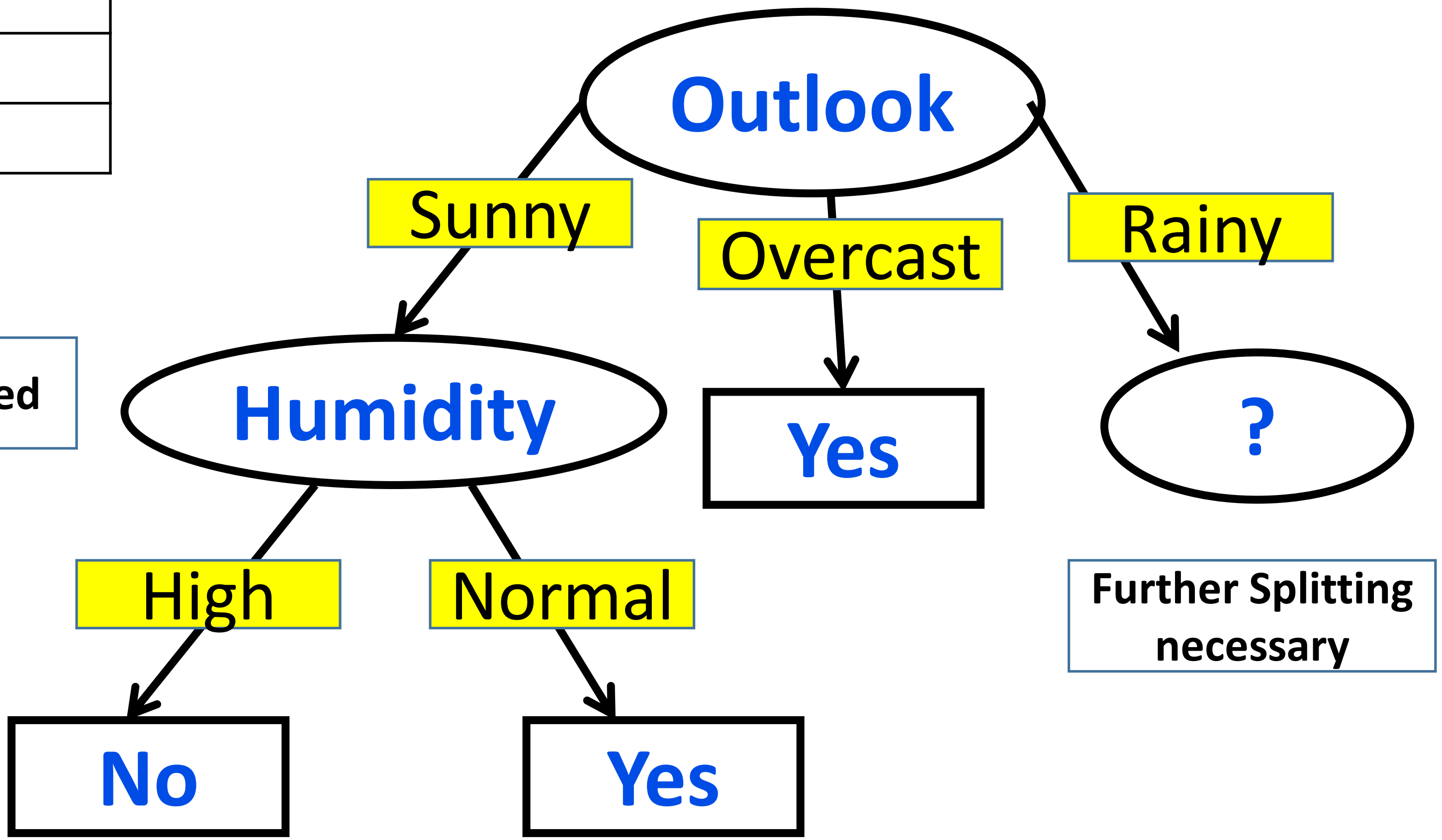
- Pick the **HIGHEST GAIN ATTRIBUTE**.

Attributes	**Gain**
Temperature	**0.571**
Humidity	**0.971**
Wind	**0.020**

- Next Node in Sunny = **Humidity**

Outlook	Humidity	PlayTennis
Sunny	High	No
Sunny	High	No
Sunny	High	No
Sunny	Normal	Yes
Sunny	Normal	Yes

Humidity is selected



Further Splitting necessary

Outlook	Temperature	Humidity	Wind	PlayTennis
Rainy	Mild	High	Weak	Yes
Rainy	Cool	Normal	Weak	Yes
Rainy	Cool	Normal	Strong	No
Rainy	Mild	Normal	Weak	Yes
Rainy	Mild	High	Strong	No

$p = 3 \quad n = 2$

Total = 5

- **Entropy:**

$$\text{Entropy} = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

$$\text{Entropy}(_{\text{Sunny}}) = -\frac{3}{2+3}\log_2\left(\frac{3}{2+3}\right) - \frac{2}{2+3}\log_2\left(\frac{2}{2+3}\right)$$

$$= 0.971$$

- For each Attribute: (**Temperature**)
  - **Calculate entropy for each Temperature. ie for "Hot", "Cool" and "Mild"**

Outlook	Temperature	PlayTennis
Rainy	Mild	Yes
Rainy	Cool	Yes
Rainy	Cool	No
Rainy	Mild	Yes
Rainy	Mild	No

Temperature	p	n	Entropy
Hot	0	0	**0**
Cool	1	1	**1**
Mild	2	1	**0.918**

- **Calculate Average Information Entropy:** **I (Temperature) = 0.951**

$$= \frac{\mathbf{p}_{Hot}+nHot}{\mathbf{p}+n}\ \textbf{Ent (Temp = Hot)} + \frac{\mathbf{p}_{Cool}+nCoo_l}{\mathbf{p}+n}\ \textbf{Ent(Temp = Cool)} + \frac{\mathbf{p}_{Mild}+nMil_d}{\mathbf{p}+n}\ \textbf{Ent (Temp= Mild)}$$

- $= \frac{0}{5}(0) + \frac{2}{5}(1) + \frac{3}{5}(0.918) = 0.4 + 0.551 = 0.951$ **I (Temperature) = 0.951**

- **Calculate Gain :** **Gain (Temperature) = 0.971 – 0.951 = 0.020**

- For each Attribute: (**Humidity**)
  - **Calculate entropy for each Humidity. ie for "Normal" and "High"**

Outlook	Humidity	PlayTennis
Rainy	High	Yes
Rainy	Normal	Yes
Rainy	Normal	No
Rainy	Normal	Yes
Rainy	High	No

Humidity	p	n	Entropy
Normal	2	1	0.918
High	1	1	1

- **Calculate Average Information Entropy:** **I (Humidity) = 0.951**
- **I (Humidity) =** $\frac{p_{Normal}+nNor_{mal}}{p+n}$ **Ent(Humi = Normal) +** $\frac{p_{High}+nHig_{h}}{p+n}$ **Ent (Humi = High)**

  = $\frac{3}{5}$ **(0.918) +** $\frac{2}{5}$ **(1) = 0.551 + 0.4 = 0.951**

- **Calculate Gain : Gain (Humidity) = 0.971 – 0.951 = 0.020**

- For each Attribute: (**Wind**)
  - **Calculate entropy for each Humidity. ie for "Strong" and "Weak"**

Outlook	Wind	PlayTennis
Rainy	Weak	Yes
Rainy	Weak	Yes
Rainy	Strong	No
Rainy	Weak	Yes
Rainy	Strong	No

Wind	p	n	Entropy
Strong	0	2	0
Weak	3	0	0

- **Calculate Average Information Entropy:** **I (Wind) = 0**

$$\textbf{I (Wind)} = \frac{p_{Strong} + n_{\;Strong}}{p + n}\,\textbf{Ent (Wind = Strong)} + \frac{p_{Weak} + nWea_k}{p + n}\,\textbf{Ent (Wind = Weak)}$$

$$= \frac{2}{5}(0) + \frac{3}{5}(0) = 0 + 0 = 0$$

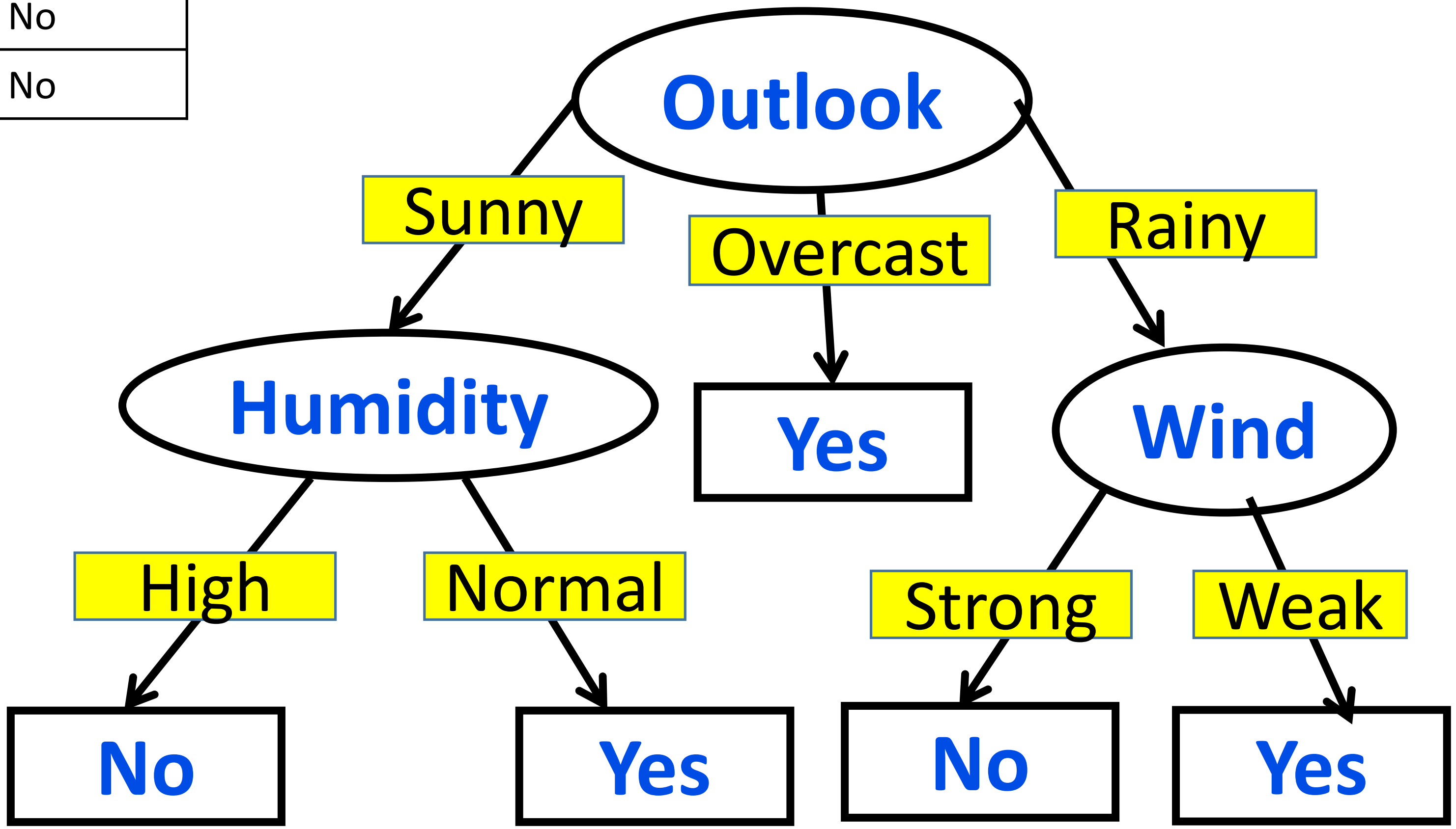
- **Calculate Gain :** **Gain (Wind) = 0.971 – 0 = 0.971**

- Pick the **HIGHEST GAIN ATTRIBUTE**.

Attributes	**Gain**
Temperature	**0.020**
Humidity	**0.020**
Wind	**0.971**

- Next Node in Rainy = **Wind**

Outlook	Wind	PlayTennis
Rainy	Weak	Yes
Rainy	Weak	Yes
Rainy	Weak	Yes
Rainy	Strong	No
Rainy	Strong	No



Rules

IF Outlook =“Overcast”

OR Outlook =“Sunny” AND Humidity = “Normal”

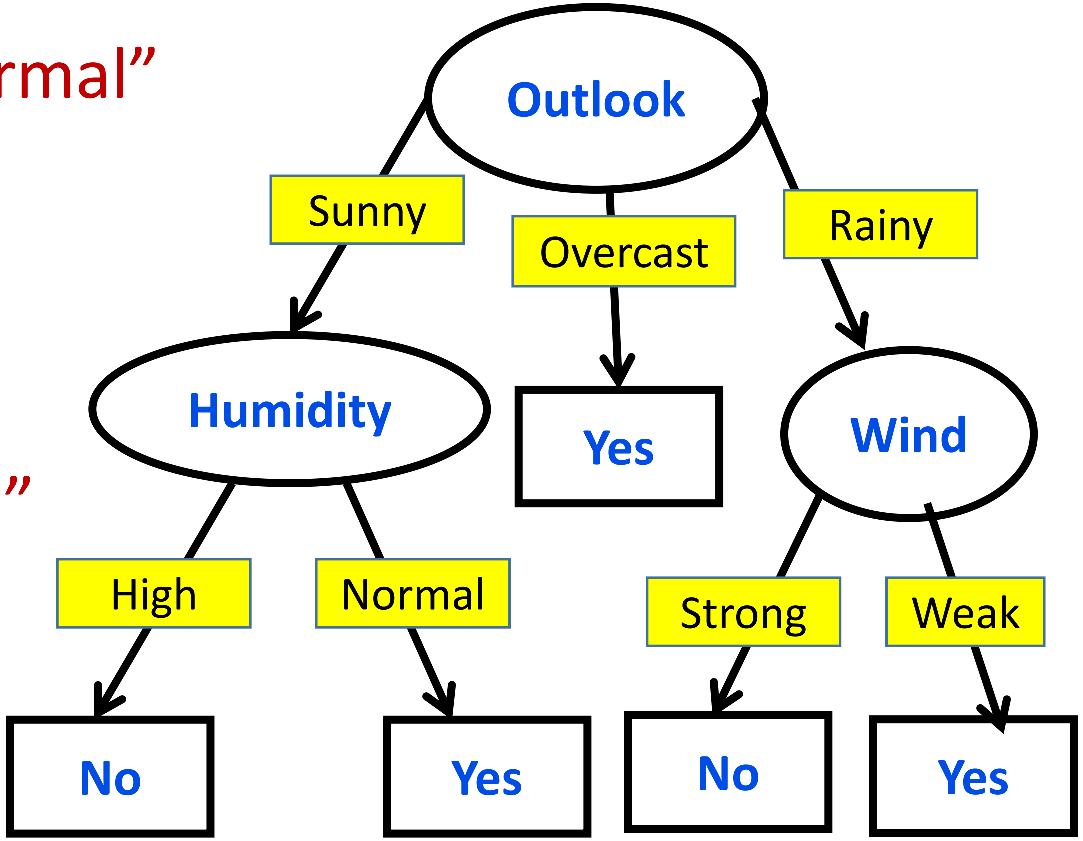
OR Outlook =“Rainy” AND Wind = “Weak”

THEN PlayTennis =“Yes”

IF Outlook =“Sunny” AND Humidity = “High”

OR Outlook =“Rainy” AND Wind = “Strong”

THEN PlayTennis =“No”



Classify the given training sample using ID3

SIZE	COLOR	SHAPE	CLASS
Small	Yellow	Round	A
Big	Yellow	Round	A
Big	Red	Round	A
Small	Red	Round	A
Small	Black	Round	B
Big	Black	Cube	B
Big	Yellow	Cube	B
Big	Black	Round	B
Small	Yellow	Cube	B

Step 1: **Entropy(s)** $= -4/9 \log_2(4/9) - 5/9 \log_2(5/9)$

$= -0.44(-1.17) - 0.555(-0.848)$

$= 0.52 + 0.471 = 0.991$

Step 2: Find Gain of all attributes

- **Attribute(size)**

$Info_{size}(D)$ or $E(size)$ = 4/9 I(2,2) + 5/9 I(2,3)

= 0.444(1) + 0.55(0.97) = 0.983

Gain(size) = Info(D) – E(size)

= 0.991 - 0.983 = **0.008**

Sixe	A	B	I(A,B)
Small	2	2	1
Big	2	3	0.97

$I(2,2) = -2/4 \log_2(2/4) - 2/4 \log_2(2/4)$

$= -0.5(-1) - 0.5(-1) = 0.5 + 0.5 = 1$

$I(2,3) = -2/5 \log_2(2/5) - 3/5 \log_2(3/5)$

$= -0.4(-1.32) - 0.6(-0.737) = 0.97$

- **Attribute(color)**

$Info_{color}(D)$ or $E(color)$

=2/9 I(2,0) + 3/9 I(0,3) + 4/9 I(2,2)

= 0 + 0 + 0.44 = 0.44 bits

Gain(size) = Info(D) – E(size)

= 0.991 – 0.44 = **0.551**

Sixe	A	B	I(A,B)
Red	2	0	0
Black	0	3	0
Yellow	2	2	1

- **Attribute(shape)**

Info_{shape} (D) or E(shape) = 6/9 I(4,2) + 3/9 I(0,3)

= 0.667(0.917) + 0 = **0.611**

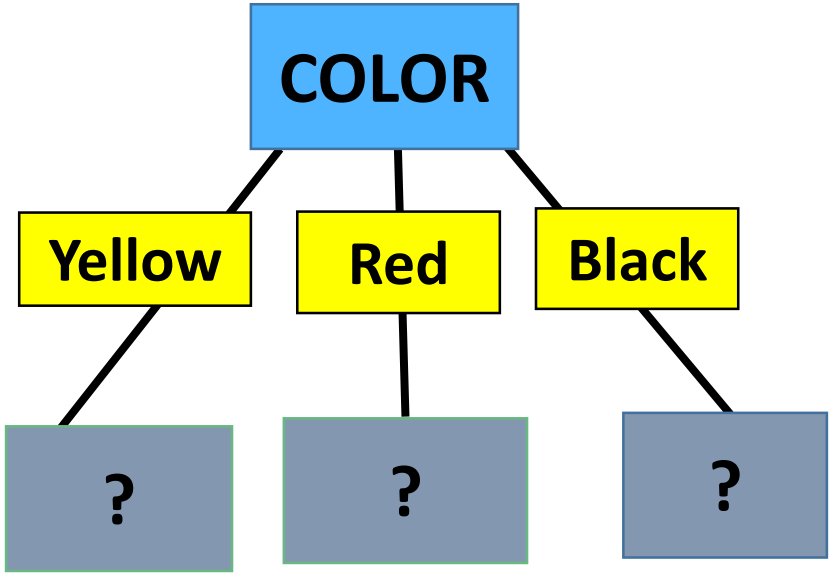
Sixe	A	B	I(A,B)
Round	4	2	0.917
Cube	0	3	0

Gain(size) = Info(D) – E(size)

= 0.991 - 0.611 = **0.379**

I(4,2) = - 4/6 log2(4/6) – 2/6 log2(2/6)
= - 0.67(- 0.584) - 0.33(- 1.585)
= 0.389+ 0.528 =0.917

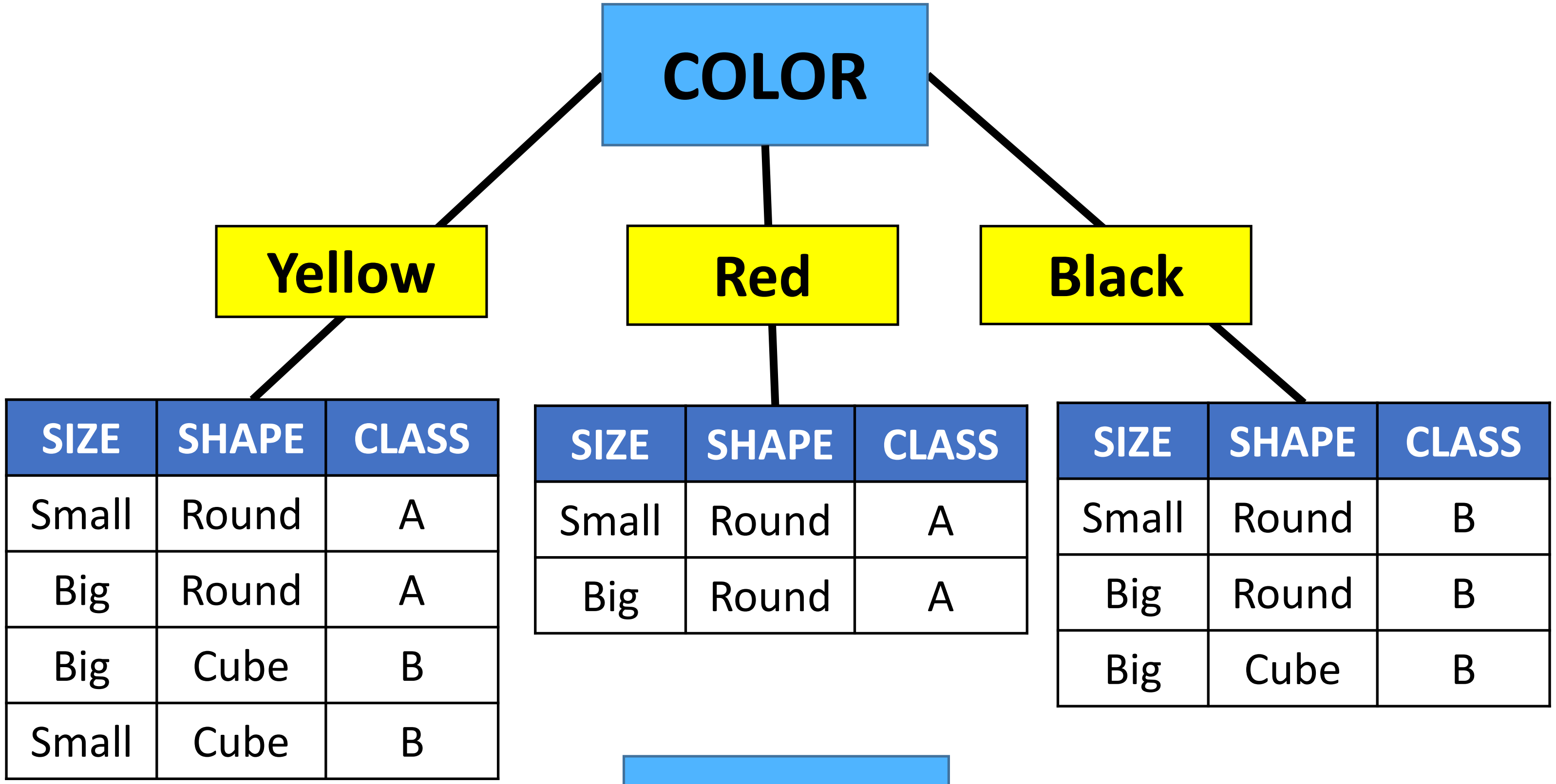
Attribute	Information Gain
Size	0.008
Color	0.551
Shape	0.379



Since Color is having the maximum information gain it is selected as the attribute to split

Step 3: Classify the tuples based on color attribute as shown

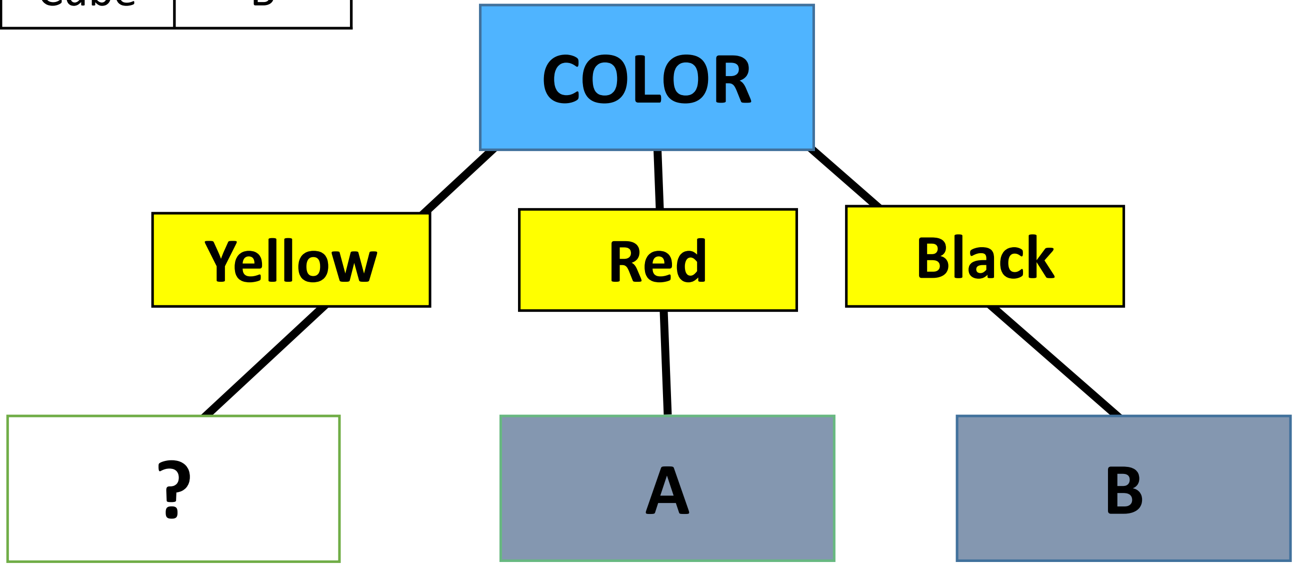
- Analyze subtree2 all tuples are of same Class A so stop the tree and put A as leaf node
- Analyze subtree3 all tuples are of same Class B so stop the tree and put B as leaf node



SIZE	SHAPE	CLASS
Small	Round	A
Big	Round	A
Big	Cube	B
Small	Cube	B

SIZE	SHAPE	CLASS
Small	Round	A
Big	Round	A

SIZE	SHAPE	CLASS
Small	Round	B
Big	Round	B
Big	Cube	B



Step 4: find the info of tuples with yellow color class A with 2 and class B with 2

I(2,2) = 1

Step 5: Find the Gain(size) and Gain(shape)

E(size) = 2/4 I(1,1) + 2/4 I(1,1)
= 0.5 (1) + 0.5 (1) = 1

Gain(size) = 1-1 = 0

E(Shape) = 2/4 I(2,0) + 2/4 I(0,2)
= 0

Gain(Shape) = 1-0 =1

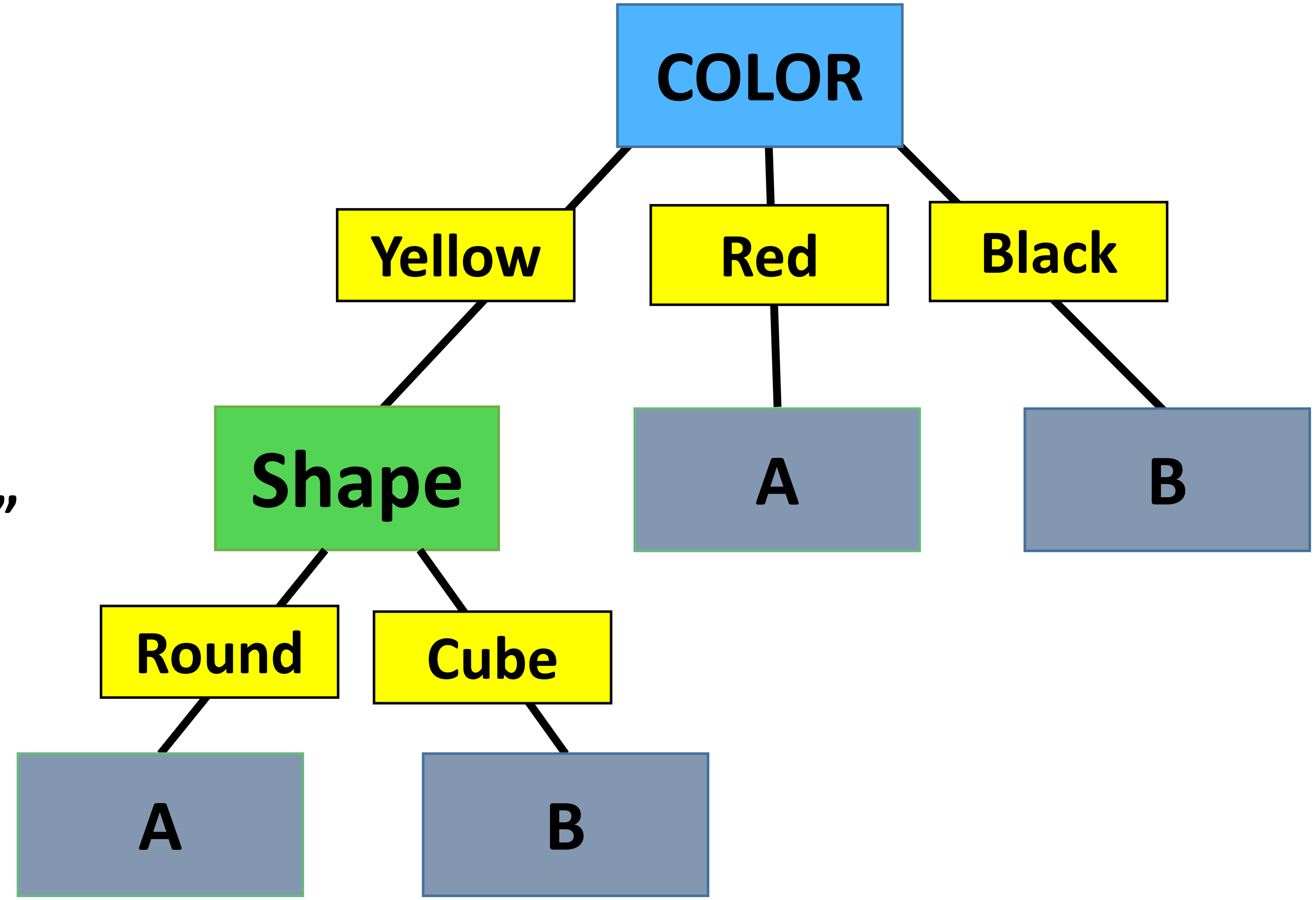
Attribute	Information Gain
Size	**0**
Shape	**1**

Since Shape is having higher information gain it is selected for splitting the sub tree1 (Yellow).

Analyze subtree2 all tuples are of same class Class A so stop the tree

Analyze subtree3 all tuples are of Class B so stop the tree

Step 5: Split the tree based on **shape**



Step 6: Association Rule:

IF color = "Red" OR
color = "Yellow" AND shape = "Round"
THEN class = "A"
ELSE
IF color = "Black" OR
color = "Yellow" AND shape = "Cube"
THEN class = "B"

Thank You