ICT Academy of Kerala

Building the Nation's Future

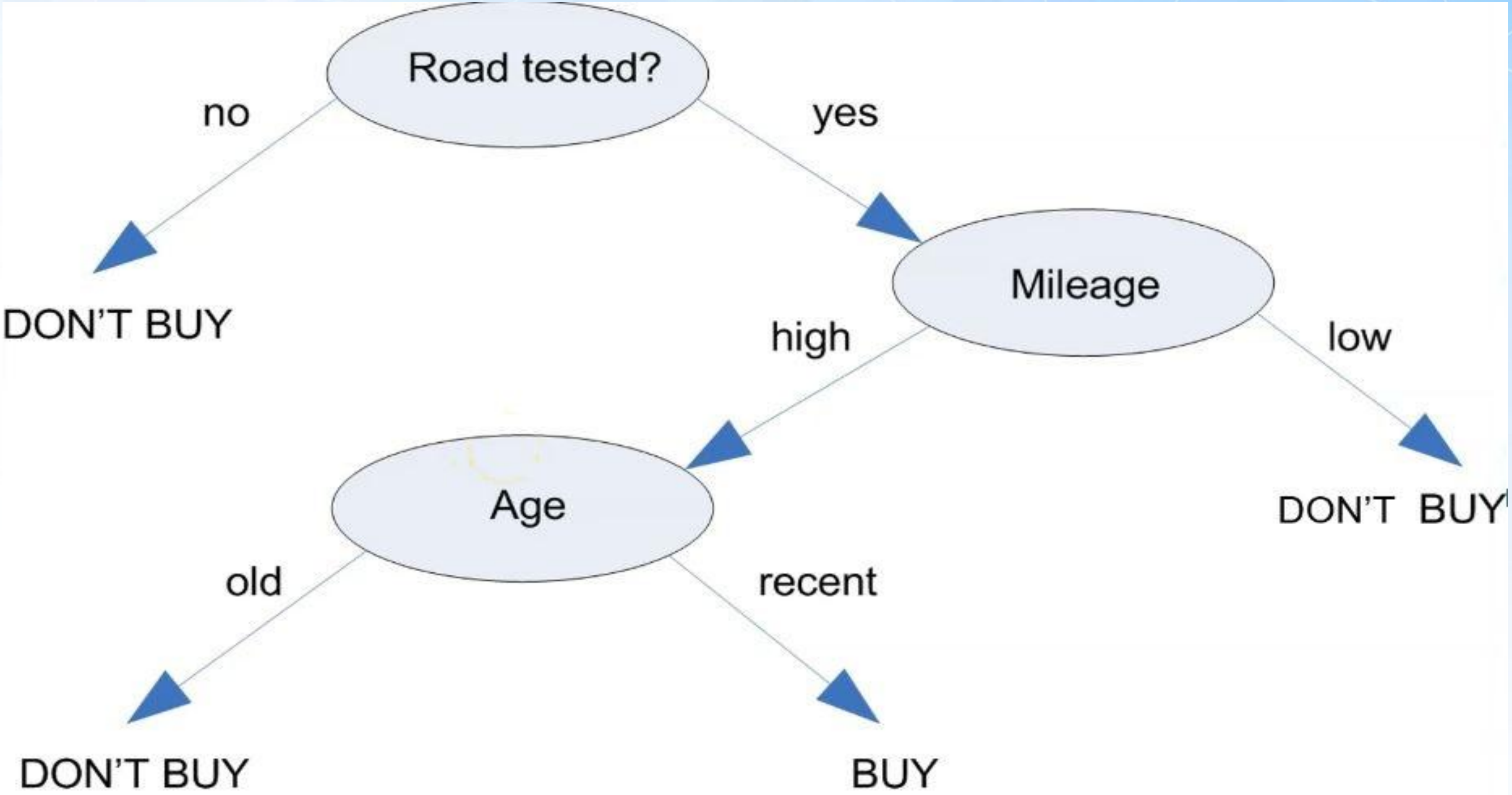
SVM, Decision trees & Random forest

A **GOVT. OF INDIA** SUPPORTED, **GOVT. OF KERALA** PARTNERED SOCIAL ENTERPRISE.

What is Decision Tree?

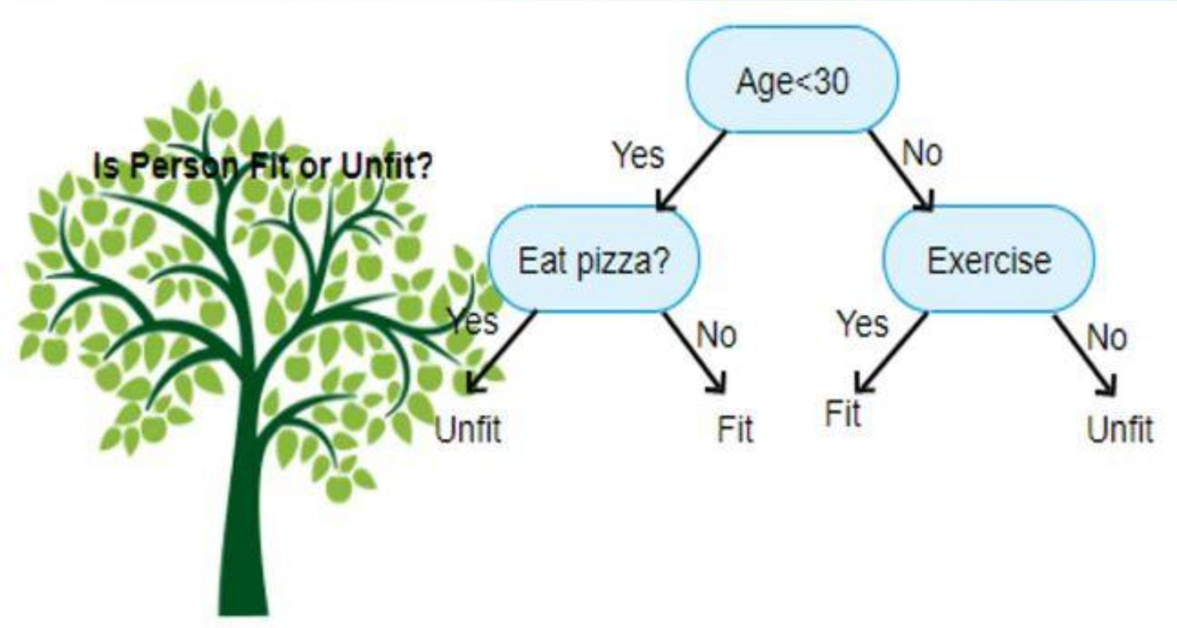
- Supervised learning algorithm
- Used to solve both regression and classification problems
- Also known as CART(**Classification And Regression Trees)**
- Tries to solve the problem by using tree representation

What is Decision Tree?

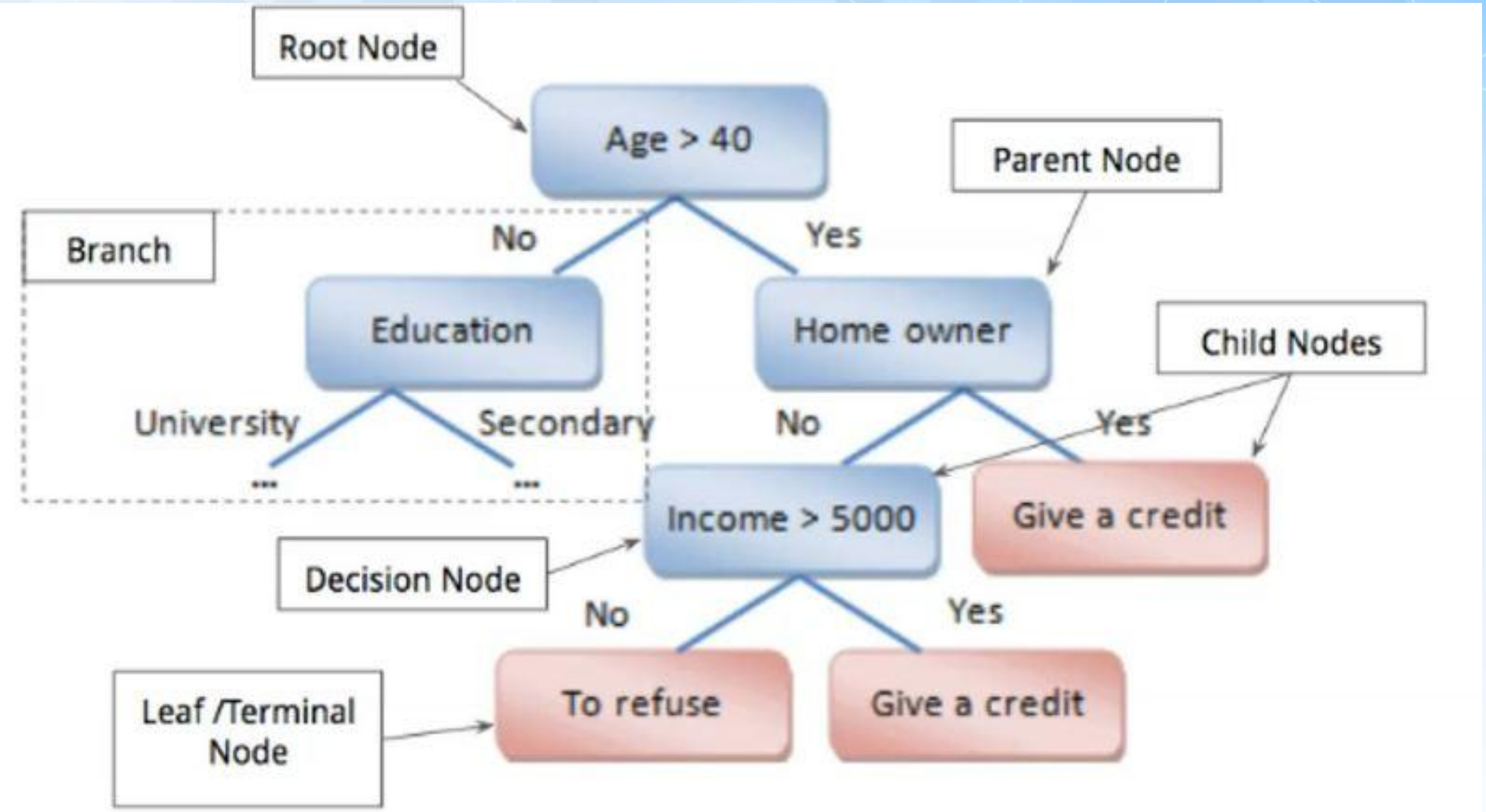


Role of A Decision Tree

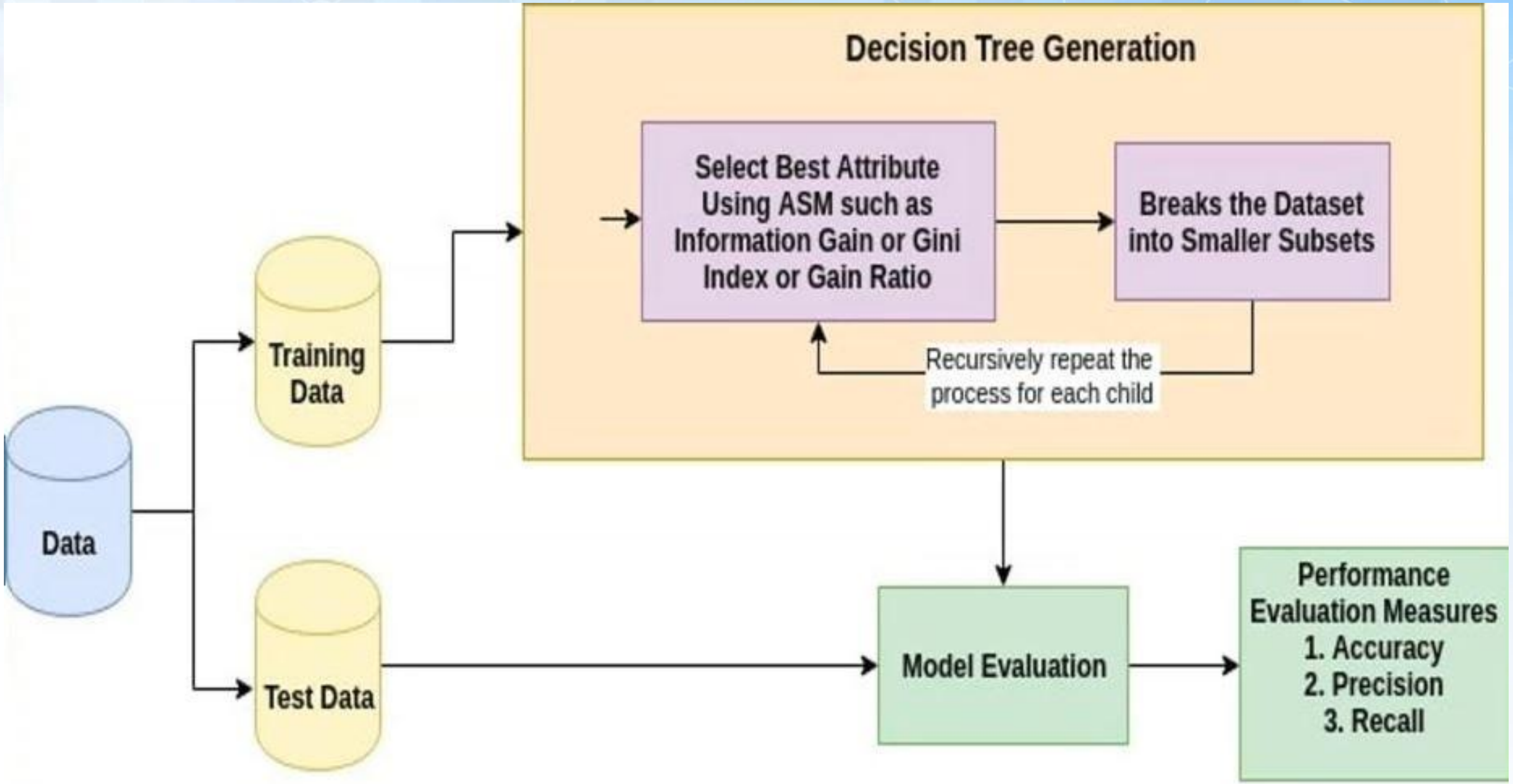
To make a series of decisions to come to a final prediction based on data provided



Terminologies in Decision Tree



Working of a Decision Tree



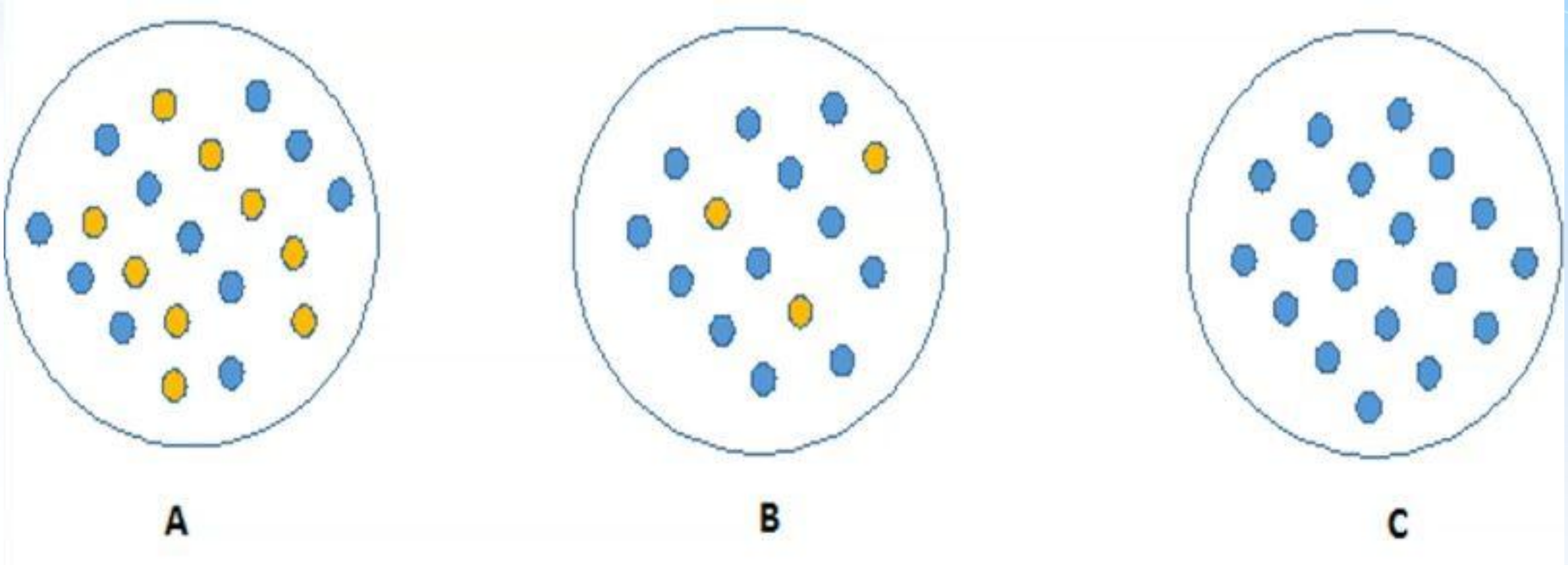
Attribute Selection Measure (ASM)

- Heuristic for selecting the splitting criterion
- Also known as splitting rules
- Provides a value to each feature by explaining the given dataset
- High attribute will be selected as a splitting attribute

Information Gain

- A statistical measure
- How well a given attribute separate the training examples

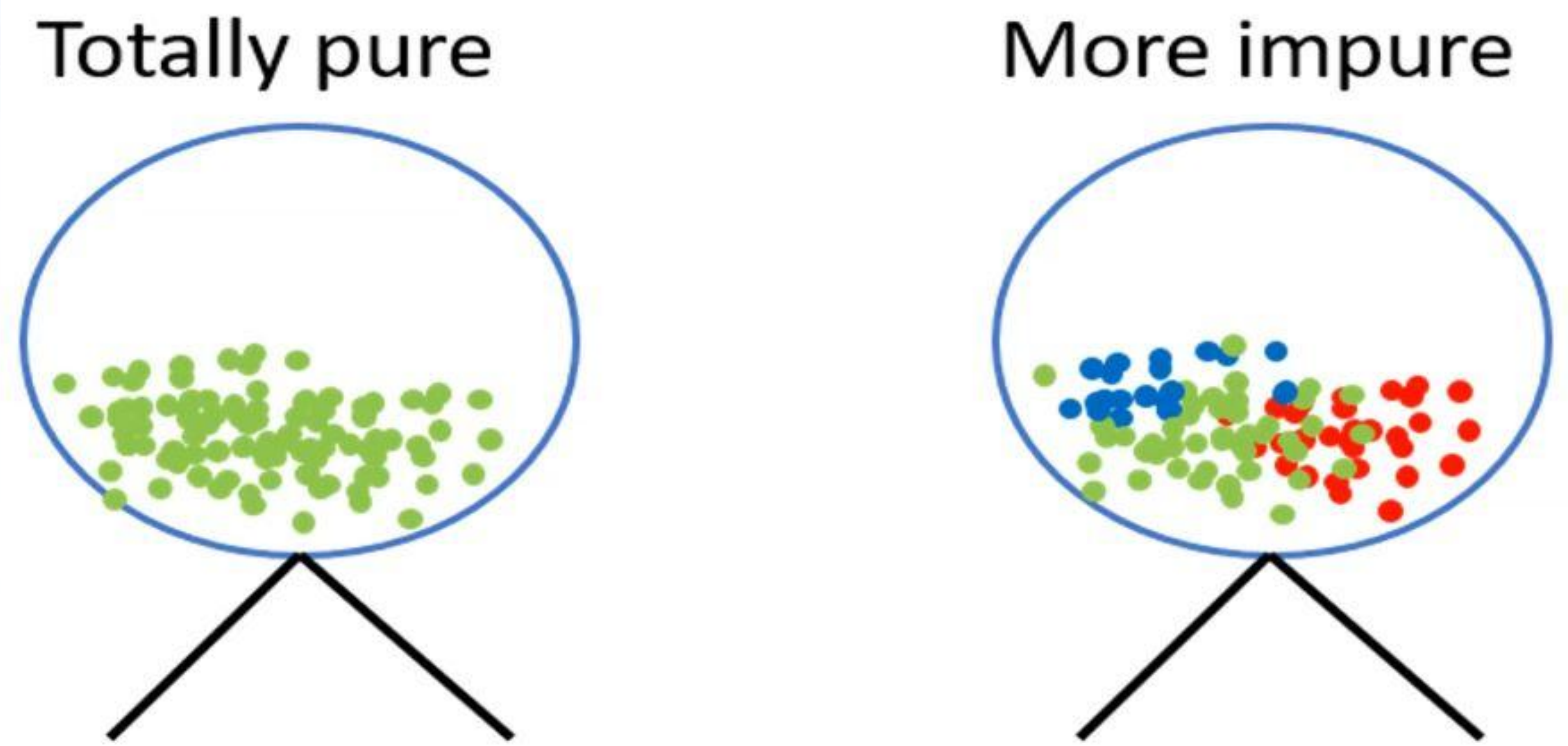
Information Gain



Entropy

- Measures the impurity of the input set
- IG is a decrease of Entropy

Entropy



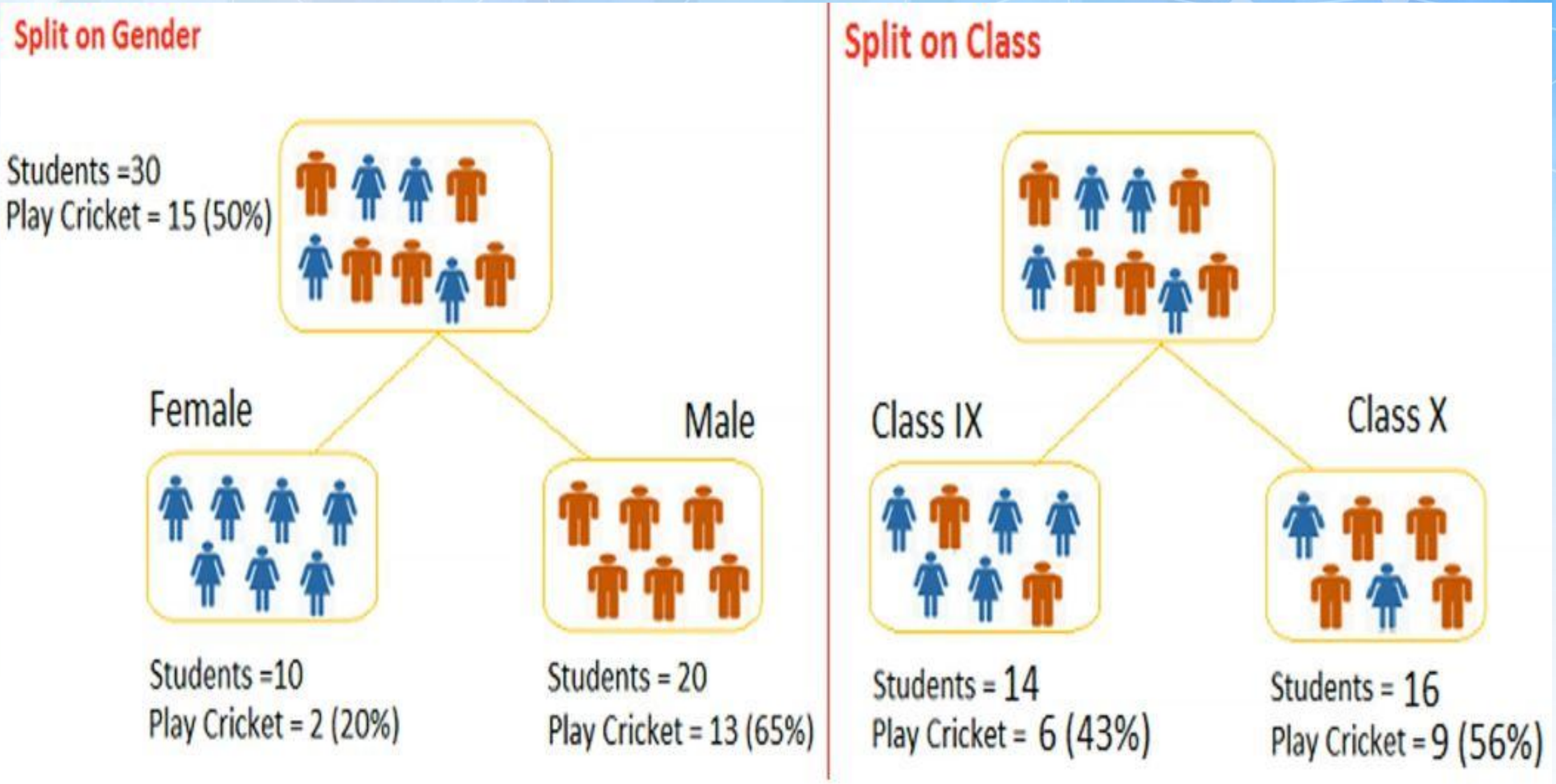
Entropy

The equation is

$$\text{Entropy} = -p \log_2 p - q \log_2 q$$

IG vs Entropy

- IG = Entropy (Parent Node) – [Average Entropy(Children)]


Split on Gender
Students =30
Play Cricket = 15 (50%)
Female
Male
Students =10
Play Cricket = 2 (20%)
Students = 20
Play Cricket = 13 (65%)
Split on Class
Class IX
Class X
Students = 14
Play Cricket = 6 (43%)
Students = 16
Play Cricket = 9 (56%)

Entropy for parent node = -(15/30) log2 (15/30) — (15/30) log2 (15/30) = **1**

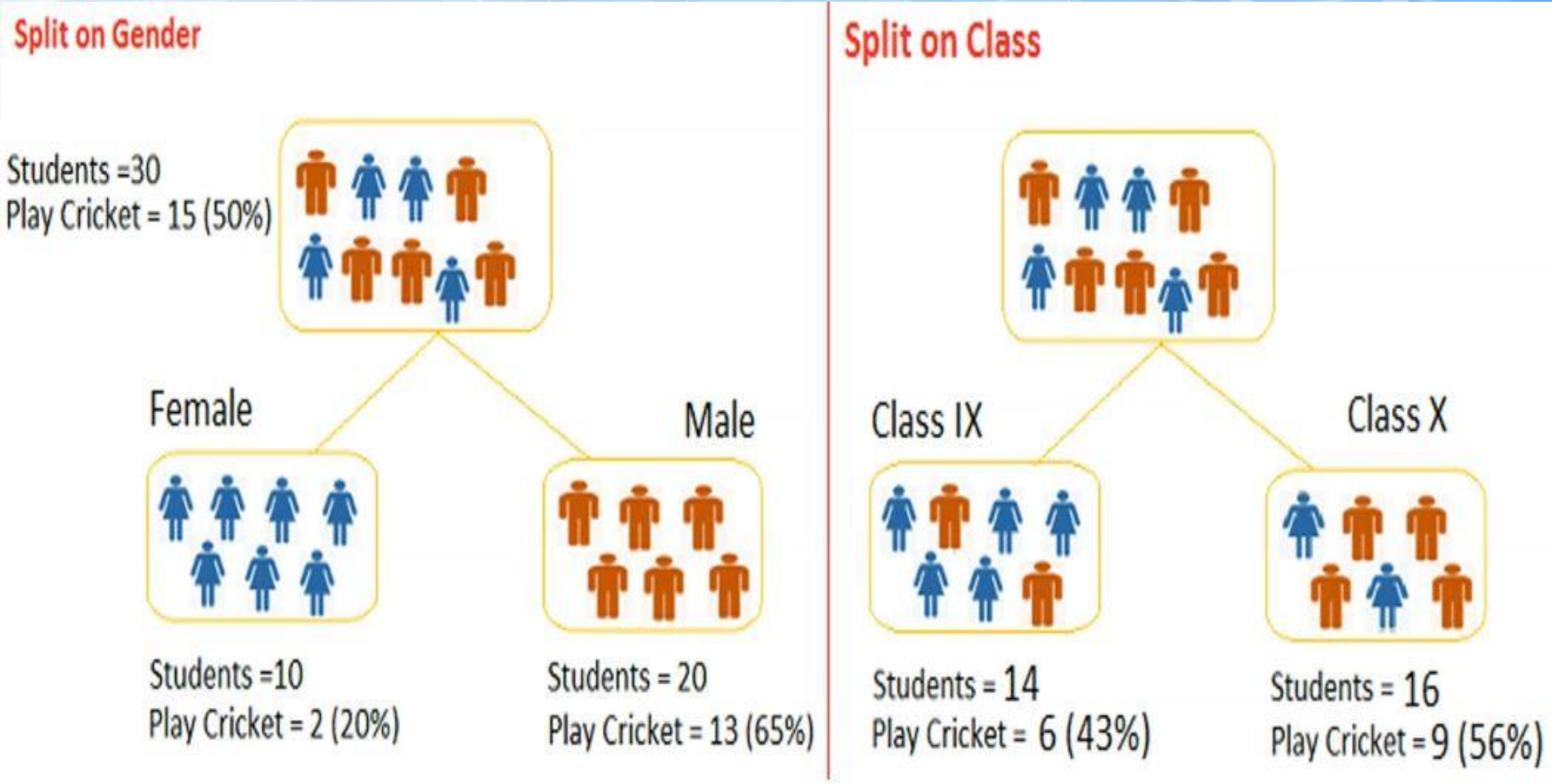
For Split on gender:

Entropy for Female node = -(2/10) log2 (2/10) — (8/10) log2 (8/10) = 0.72

Entropy for Male node = -(13/20) log2 (13/20) — (7/20) log2 (7/20) = 0.93

Entropy for split Gender = (10/30)*0.72 + (20/30)*0.93 = **0.86**

Information Gain for split on gender = 1–0.86 = ***0.14***


Split on Gender
Students =30
Play Cricket = 15 (50%)
Female
Male
Students =10
Play Cricket = 2 (20%)
Students = 20
Play Cricket = 13 (65%)
Split on Class
Class IX
Class X
Students = 14
Play Cricket = 6 (43%)
Students = 16
Play Cricket = 9 (56%)

Entropy for parent node = -(15/30) log2 (15/30) — (15/30) log2 (15/30) = **1**

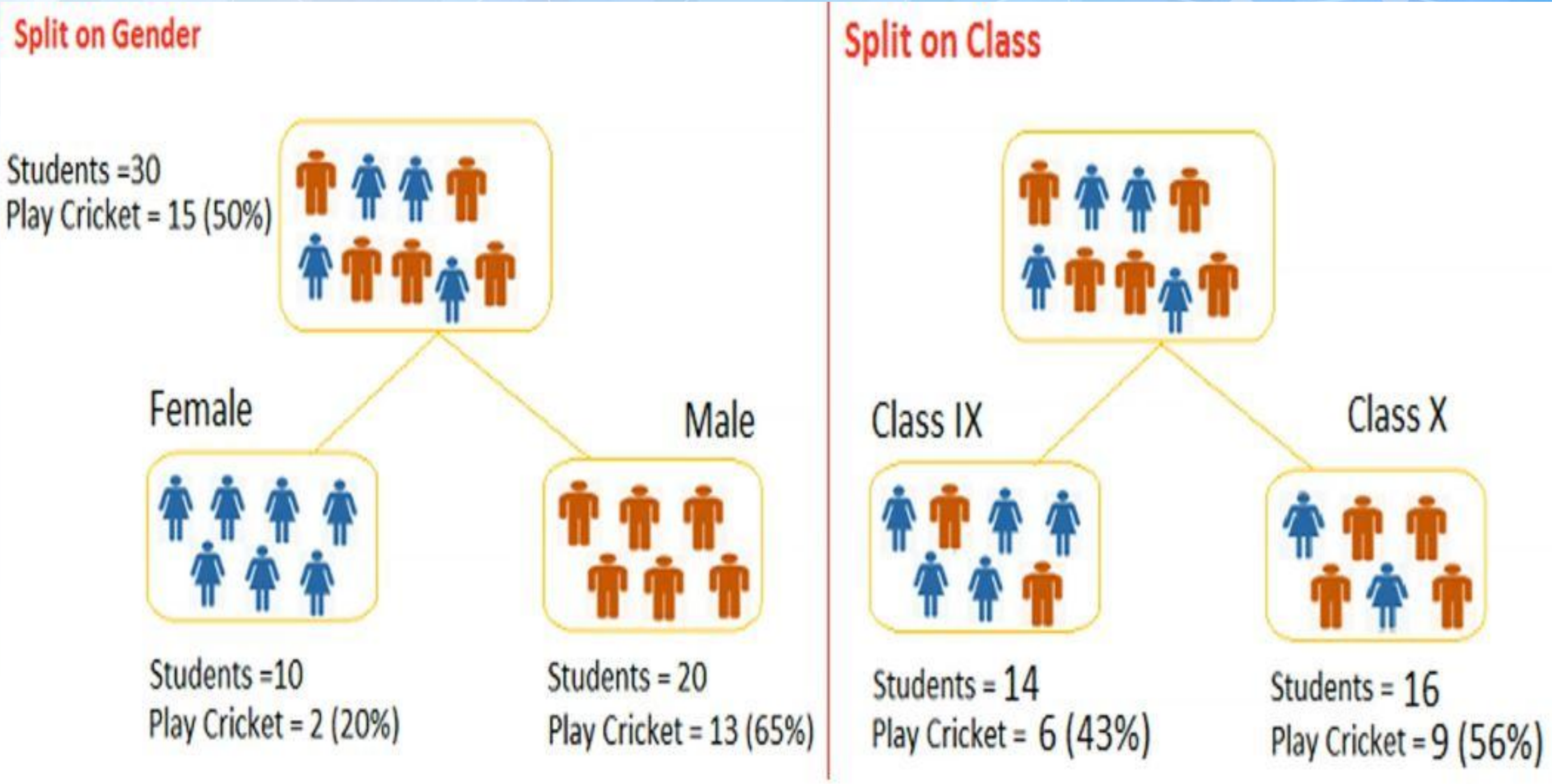
For Split on gender:

Entropy for Female node = -(2/10) log2 (2/10) — (8/10) log2 (8/10) = 0.72

Entropy for Male node = -(13/20) log2 (13/20) — (7/20) log2 (7/20) = 0.93

Entropy for split Gender = (10/30)*0.72 + (20/30)*0.93 = **0.86**

Information Gain for split on gender = 1–0.86 = ***0.14***


Split on Gender
Students =30
Play Cricket = 15 (50%)
Female
Male
Students =10
Play Cricket = 2 (20%)
Students = 20
Play Cricket = 13 (65%)
Split on Class
Class IX
Class X
Students = 14
Play Cricket = 6 (43%)
Students = 16
Play Cricket = 9 (56%)

For Split on Class:

Entropy for Class IX node = $-(6/14) \log_2 (6/14) - (8/14) \log_2 (8/14) = 0.99$

Entropy for Class X node = $-(9/16) \log_2 (9/16) - (7/16) \log_2 (7/16) = 0.99$

Entropy for split Class = $(14/30)*0.99 + (16/30)*0.99 = \mathbf{0.99}$

Information Gain for split on Class = $1 - 0.99 = \mathbf{0.01}$

Entropy for parent node = -(15/30) log2 (15/30) — (15/30) log2 (15/30) = **1**

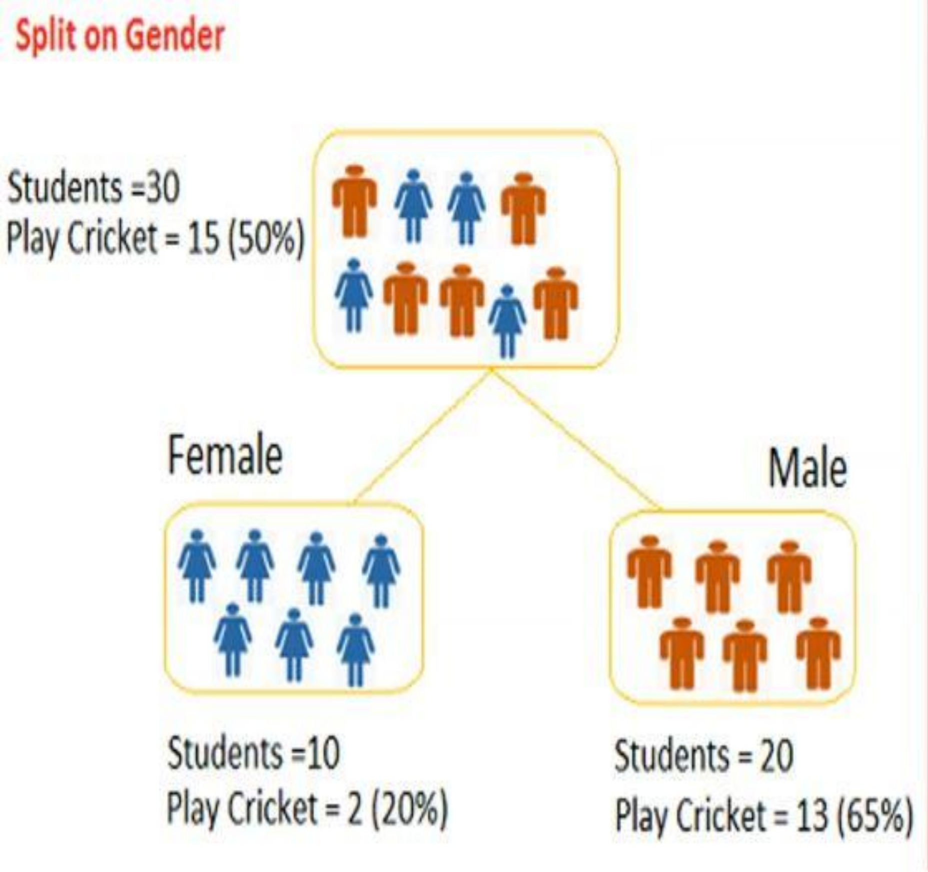
For Split on gender:

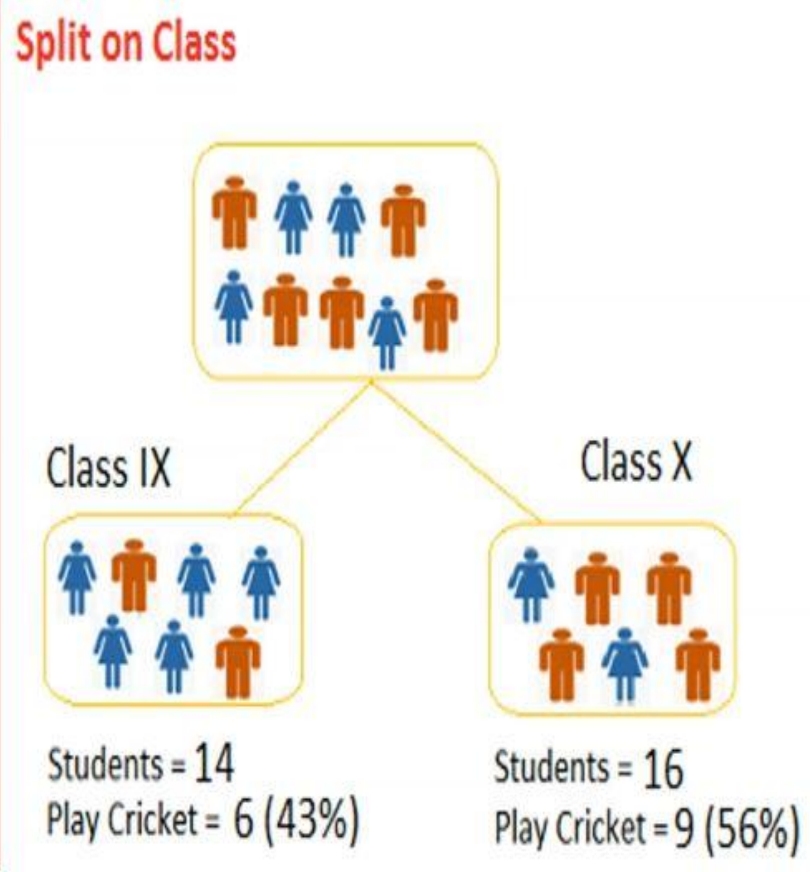
Entropy for Female node = -(2/10) log2 (2/10) — (8/10) log2 (8/10) = 0.72

Entropy for Male node = -(13/20) log2 (13/20) — (7/20) log2 (7/20) = 0.93

Entropy for split Gender = (10/30)*0.72 + (20/30)*0.93 = **0.86**

Information Gain for split on gender = 1–0.86 = ***0.14***


Split on Gender
Students =30
Play Cricket = 15 (50%)
Female
Male
Students =10
Play Cricket = 2 (20%)
Students = 20
Play Cricket = 13 (65%)


Split on Class
Class IX
Class X
Students = 14
Play Cricket = 6 (43%)
Students = 16
Play Cricket = 9 (56%)

Decision Trees

Take the entire dataset as input

↓

Calculate entropy of target variable as well as predictor attributes

↓

Calculate information gain of all attributes

↓

Choose the attribute with highest information gain as the root node

↓

Repeat the same process on every branch till the decision node of each branch is finalized

Decision Trees

Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Predictors: Outlook, Temp., Humidity, Windy

Target: Play Golf

Decision Trees

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)

= Entropy (0.36, 0.64)

= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)

= 0.94

Decision Trees

Entropy(PlayGolf) = Entropy (5,9)

= Entropy (0.36, 0.64)

$= - (0.36 \log_2 0.36) - (0.64 \log_2 0.64)$

= 0.94

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

$$Gain(T,X) = Entropy(T) - Entropy(T,X)$$

G(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)

= 0.940 – 0.693 = 0.247

Decision Trees

Decide to go for play or not.

outlook	temp.	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

Decision Trees

<u>Calculate the Entropy of the data set.</u>

Decision column consists of 14 instances and includes two labels: yes and No

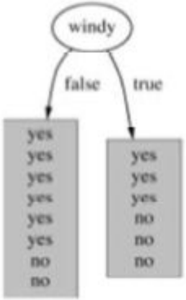
There are 9 Decision label with Yes and 5 Decision labels with No

$$
\begin{aligned}
Entropy(Decision) &= -p(yes)*\log_2 p(yes) - p(no)*\log_2 p(no) \\
&= -(9/14)*\log_2(9/14) - (5/14)*\log_2(5/14) \\
&= 0.940
\end{aligned}
$$

Decision Trees

<u>Wind Factor on Decision</u>

$$\text{Entropy(Decision|wind=false)} = -p(no)*\log_2 p(no) - p(yes)*\log_2 p(yes)$$
$$= -(2/8)*\log_2(2/8) - (6/8)*\log_2(6/8)$$
$$= 0.811$$



$$\text{Entropy(Decision|wind=True)} = -p(no)*\log_2 p(no) - p(yes)*\log_2 p(yes)$$
$$= -(3/6)*\log_2(3/6) - (3/6)*\log_2(3/6)$$
$$= 1$$

$$\text{Gain(Decision|wind)} = \text{Entropy(Decision)} - [p(\text{Decision|wind=false})* \text{Entropy(Decision|wind=false)} - [p(\text{Decision|wind=True})* \text{Entropy(Decision|wind=True)}$$
$$= 0.940-[(8/14)*0.811]-[(6/14)*1]$$
$$= 0.048$$

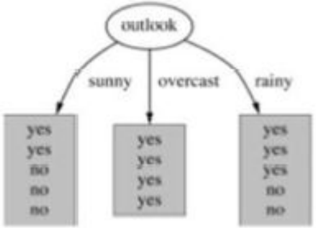
Decision Trees

<u>Outlook factor on Decision</u>

Entropy(Decision|Outlook=sunny)=-p(no)*$\log_2$p(no)-
p(yes)*$\log_2$p(yes)
=-(3/5)*$\log_2$(3/5)-
(2/5)*$\log_2$(2/5)
= 0.9708

Entropy(Decision|Outlook=Overcast)=-p(no)*$\log_2$p(no)-
p(yes)*$\log_2$p(yes)
=-(0/4)*$\log_2$(0/4)-
(4/4)*$\log_2$(4/4)
= 0

Entropy(Decision|Outlook=Rain)=-p(no)*$\log_2$p(no)-
p(yes)*$\log_2$p(yes)
=-(2/5)*$\log_2$(2/5)-
(3/5)*$\log_2$(3/5)
= 0.971



Gain(Decision|Outlook)=Entropy(Decision)-[p(Decision|Outlook=sunny)*
Entropy(Decision|Outlook=sunny)- [p(Decision|outlook=overcast)*
Entropy(Decision|Outlook=overcast) - [p(Decision|outlook=Rain)*
Entropy(Decision|Outlook=Rain)
= 0.940-[5/14)*0.9708]-[(4/14)*0]-[(5/14)*0.971]
= 0.2465

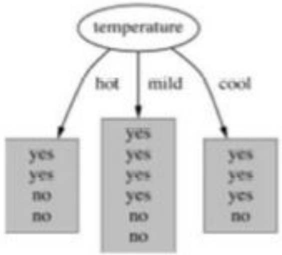
Decision Trees

Temperature factor on Decision

Entropy(Decision|Temp=Hot)=-p(no)*$\log_2$p(no)-
p(yes)*$\log_2$p(yes)
=-(2/4)*$\log_2$(2/4)-
(2/4)*$\log_2$(2/4)
= 1

Entropy(Decision|Temp=mild)=-p(no)*$\log_2$p(no)-
p(yes)*$\log_2$p(yes)
=-(2/6)*$\log_2$(2/6)-
(4/6)*$\log_2$(4/6)
= 0.9148

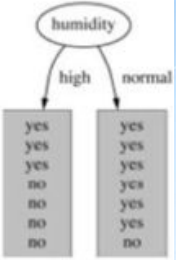
Entropy(Decision|Temp=cool)=-p(no)*$\log_2$p(no)-
p(yes)*$\log_2$p(yes)
=-(1/4)*$\log_2$(1/4)-
(3/4)*$\log_2$(3/4)
= 0.8112



Gain(Decision|Outlook)=Entropy(Decision)-[p(Decision|Temp=Hot)* Entropy(Decision|Temp=Hot)- [p(Decision|Temp=mild)* Entropy(Decision|Temp=mild) - [p(Decision|temp=cool)* Entropy(Decision|temp=cool)
= 0.940-[4/14)*1]-[(6/14)*0.9148]-[(5/14)*0.971]-[(4/14)*0.8112]
= 0.030

Decision Trees

<u>Humidity Factor on Decision</u>



$$\text{Entropy(Decision|Humidity=high)} = -p(no) * \log_2 p(no) - p(yes) * \log_2 p(yes)$$
$$= -(4/7) * \log_2(4/7) - (3/7) * \log_2(3/7)$$
$$= 0.9851$$

$$\text{Entropy(Decision|Humidity=Normal)} = -p(no) * \log_2 p(no) - p(yes) * \log_2 p(yes)$$
$$= -(1/7) * \log_2(1/7) - (6/7) * \log_2(6/7)$$
$$= 0.5913$$

Gain(Decision|Humidity)=Entropy(Decision)-
[p(Decision|Humidity=high)* Entropy(Decision|Humidity=high)-
[p(Decision|Humidity=normal)*
Entropy(Decision|Humidity=normal)

$$= 0.940 - [(7/14) * 0.9851] - [(7/14) * 0.5913]$$
$$= 0.1519$$

Decision Trees

Therefore

Gain(Decision,wind)=0.048

Gain(Decision,Humidity)=0.1519

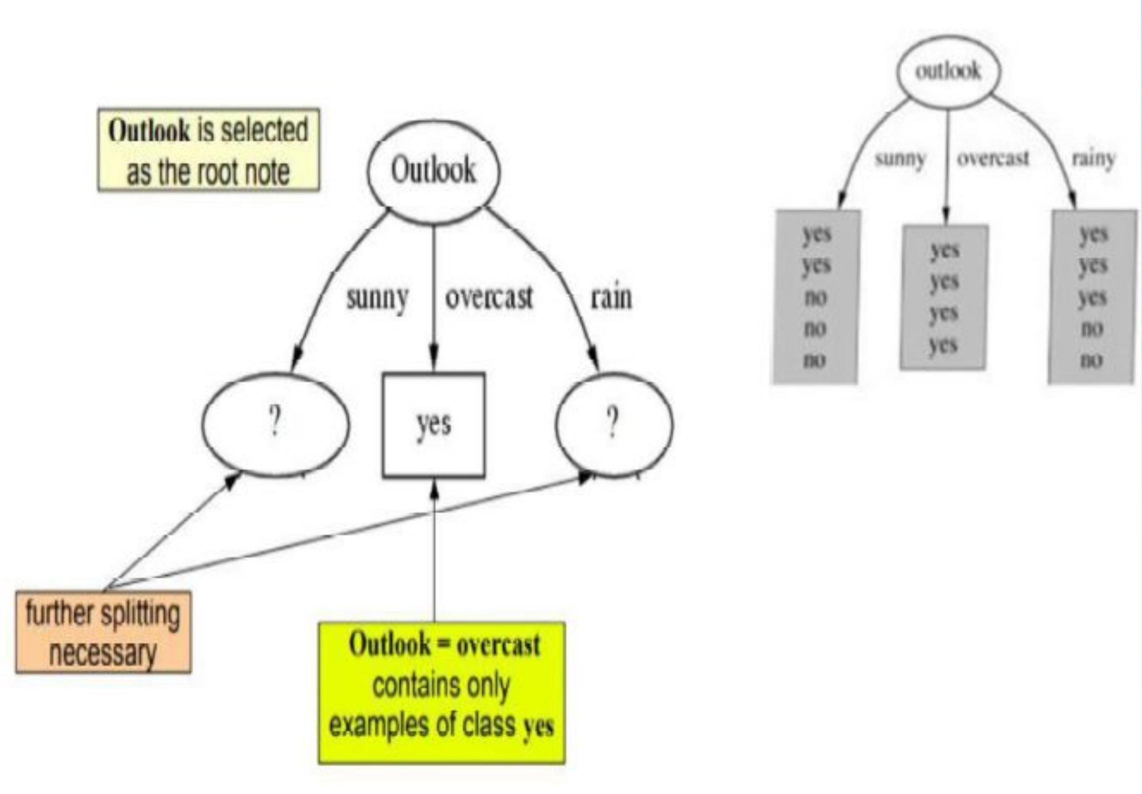
Gain(Decision,Temp)=0.030

Gain(Decision,Outlook)=0.2465 (Max Gain)

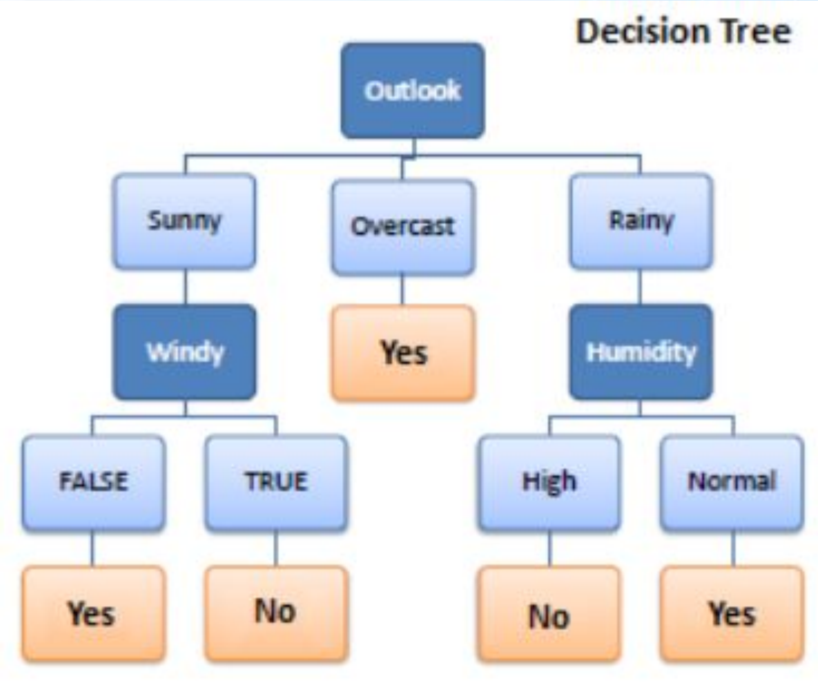
Outlook Factor on decision produces highest score.

So Outlook decision appears on the root node of the tree

Decision Trees



Decision Trees



Gini Index

Steps to calculate Gini for a split

- Calculate Gini for sub-nodes, using formula sum of the squares of probability for success and failure (p^2+q^2)
- Calculate Gini for split using weighted Gini score of each node of that split

Split on Gender:

1. Calculate, Gini for sub-node Female =

   $(0.2)*(0.2)+(0.8)*(0.8)=0.68$

2. Gini for sub-node Male = $(0.65)*(0.65)+(0.35)*(0.35)=0.55$

3. Calculate weighted Gini for Split Gender =

   $(10/30)*0.68+(20/30)*0.55 = \mathbf{0.59}$

Similar for Split on Class:

1. Gini for sub-node Class IX = (0.43)*(0.43)+(0.57)*(0.57)=0.51

2. Gini for sub-node Class X = (0.56)*(0.56)+(0.44)*(0.44)=0.51

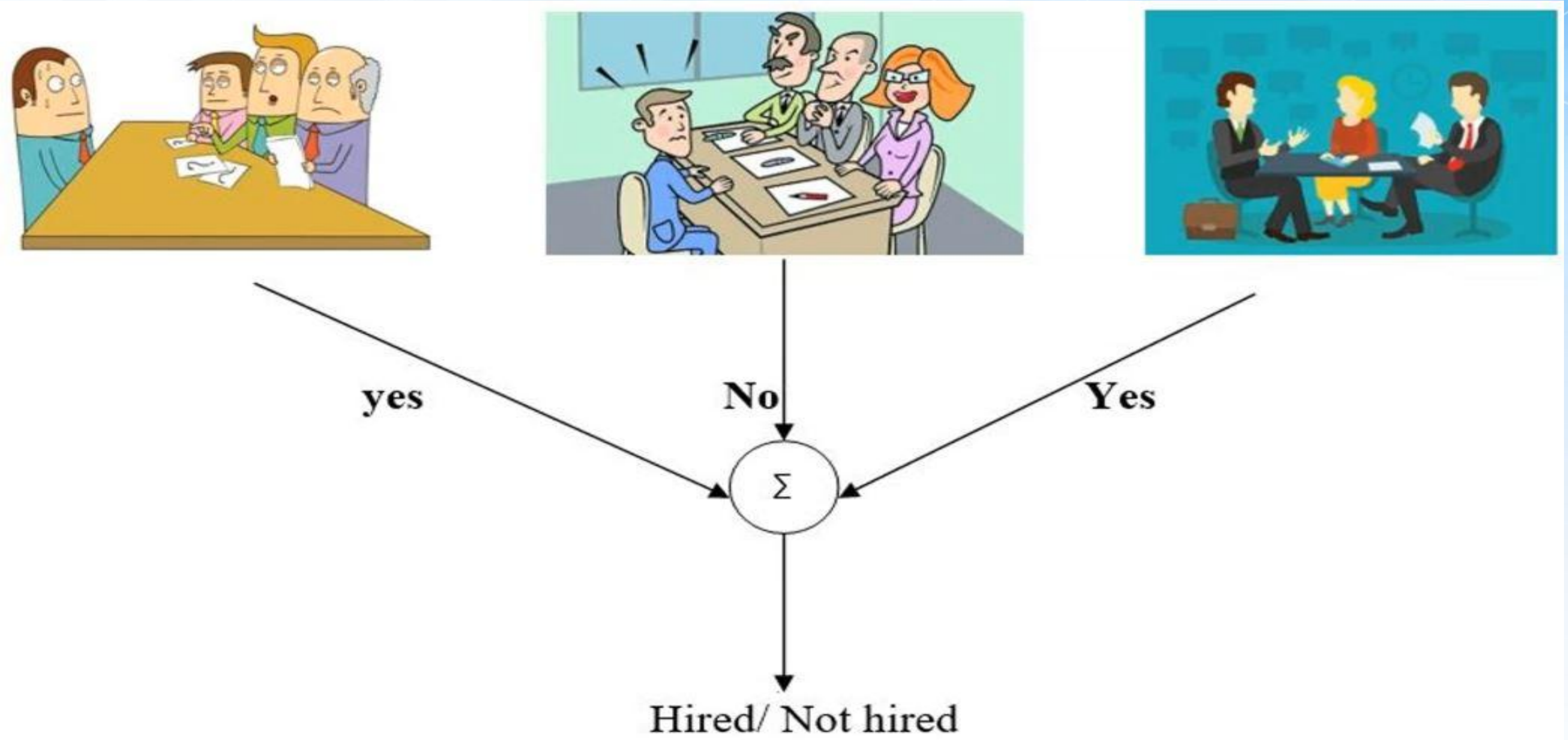
3. Calculate weighted Gini for Split Class =

   (14/30)*0.51+(16/30)*0.51 = **0.51**

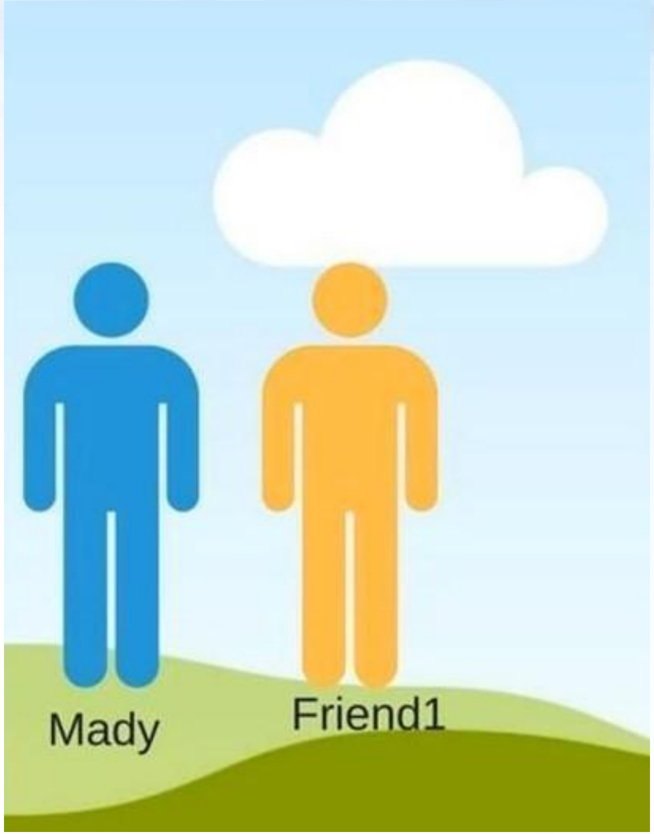
Random Forest Algorithm

- Supervised learning algorithm
- Ensemble of decision trees
- Bagging method in which the result of different multiple models are combined to bring a better result
- Used for both classification and regression problems

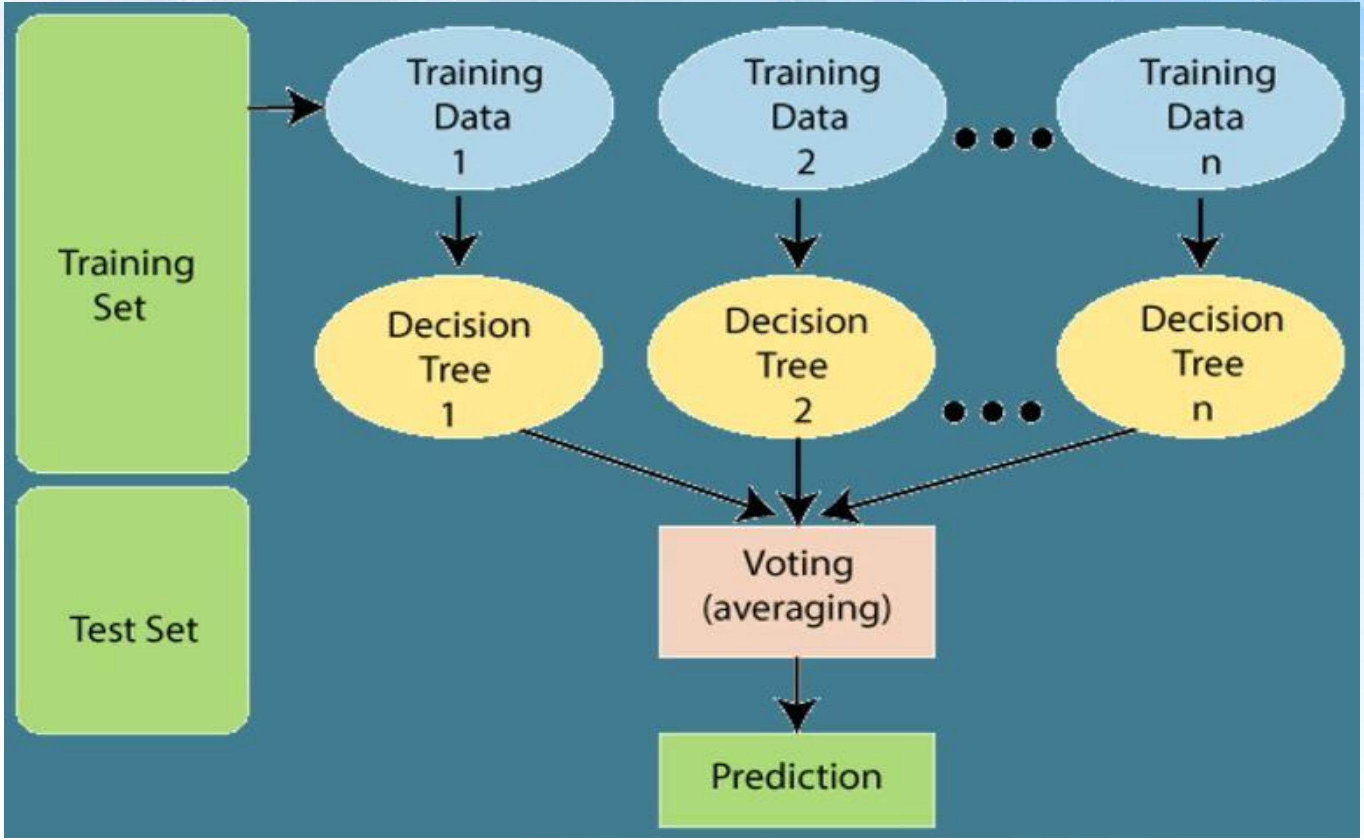
Real time Analogy



Real time Analogy

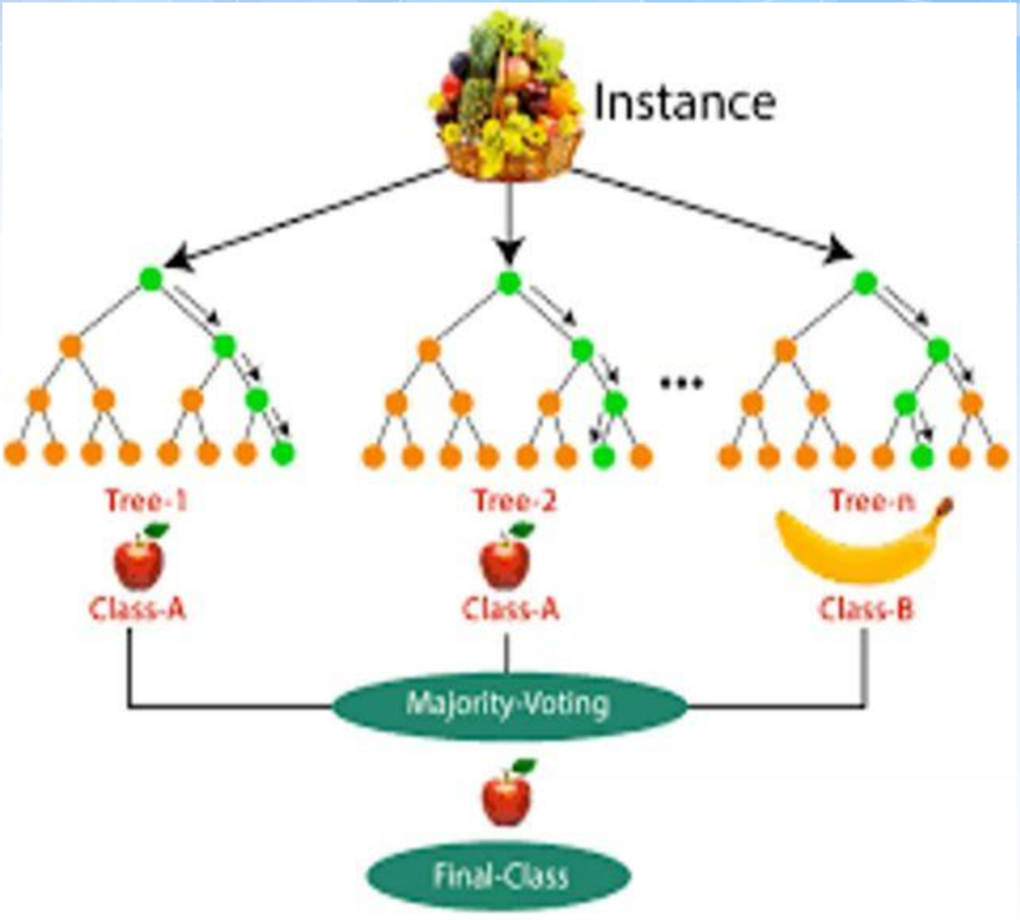


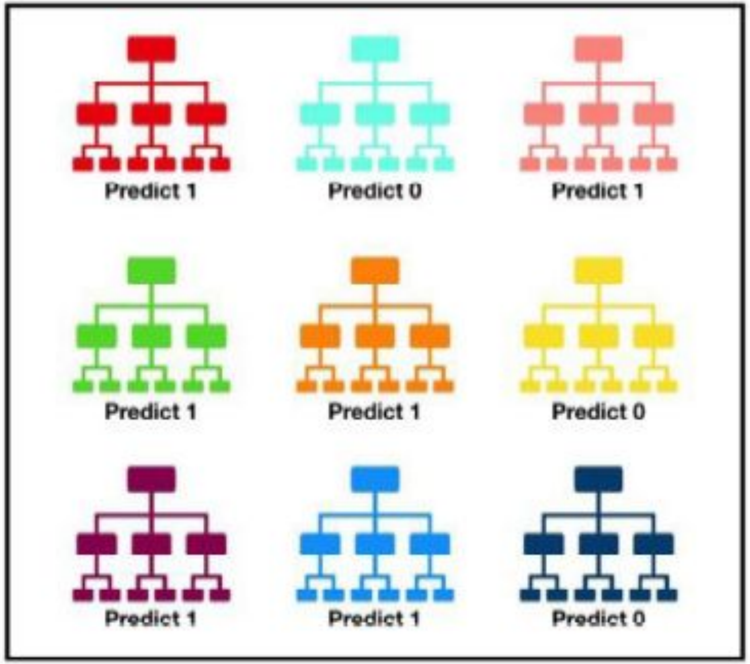
Working of Random Forest



Algorithm

- Step 1: Select random k data points from the training set
- Step 2: Build the decision trees associated with the selected data points
- Step 3: Choose the number N for decision trees that you want to build
- Step 4: Repeat step 1 & 2
- Step 5: For new data points, find the predictions of each decision tree and assign the new data points to the category that wins the majority votes


Instance
Tree-1
Class-A
Tree-2
Class-A
Tree-n
Class-B
Majority-Voting
Final-Class



Tally: Six 1s and Three 0s
Prediction: 1

Applications