Decision Tree

Introduction to Decision Tree

- **Decision Tree** is a type of **Supervised** learning **technique** that follows a tree-like structure. It starts with the root node that expands further until it reaches the leaf node.
- It is a **graphical representation** for **getting all the possible solutions** to a **problem based on given conditions**.
- An **internal nodes** represent the **features** of a dataset, **branches** represent the **decision rules** and each **leaf node represents** the **outcome.**
- It can be **used for both classification and Regression** problems, but preferred for **Classification problems**.
- The **logic behind the decision tree can be easily understood** because it shows a tree-like structure and hence decision making tasks becomes **easy**.
- It **gives all possible solutions for a given problem** that facilitate decision making process.

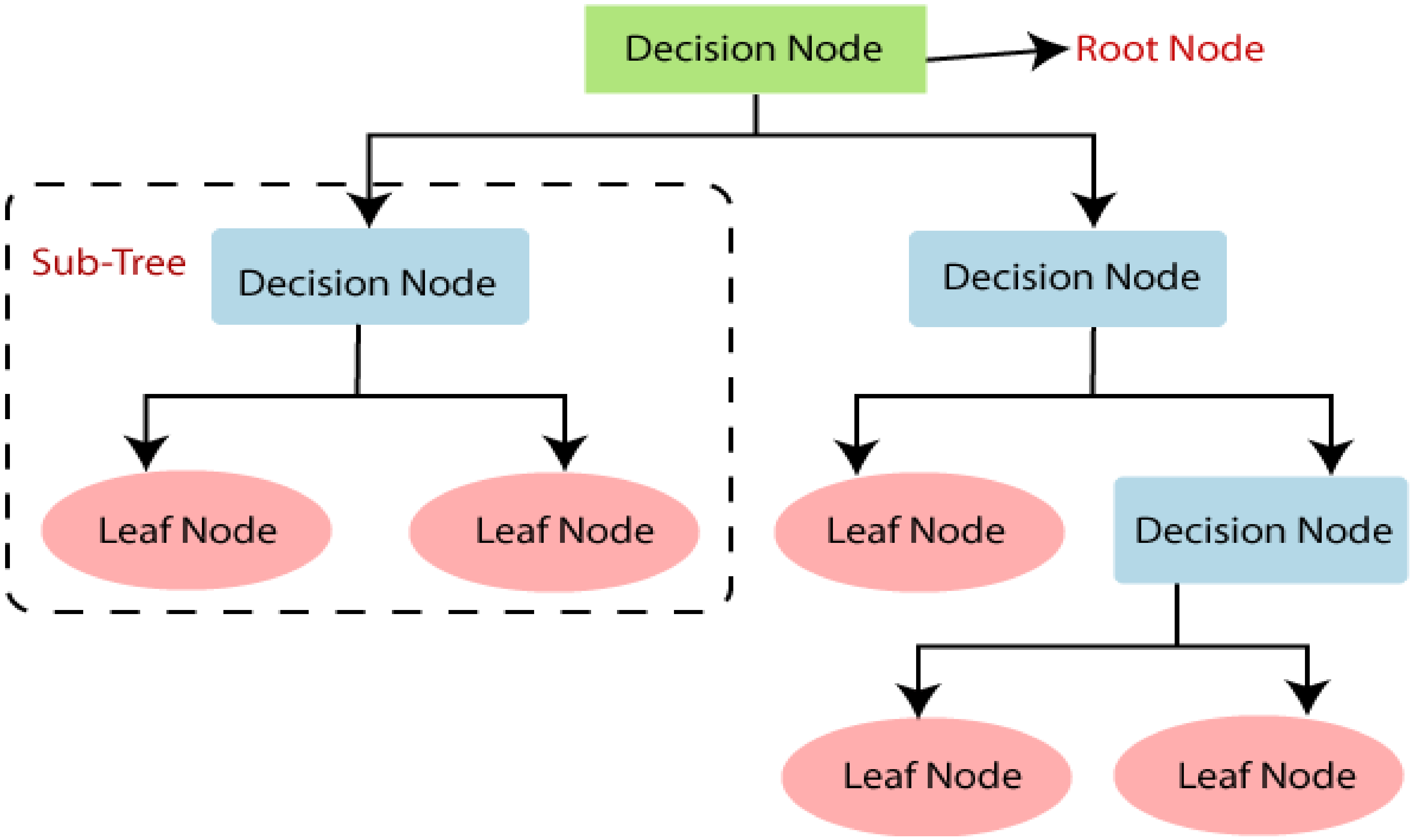
Introduction to Decision Tree

- The circle at the top of the diagram is the root node and it contains all the training data, which is used to grow the tree.
- The tree always starts from the root node and grows by splitting the data at each level into new nodes (daughter nodes).
- The root node (parent node) contains the entire data and daughter nodes (internal nodes) hold respective subsets of the data.
- All the nodes are connected by branches shown by the line segments. The nodes that are at the end of the branches are called terminal nodes or leaf nodes, shown by boxes.
- The leaf nodes in this figure are class labels.

Decision Tree Terminologies

- **Root Node:** It is the **starting node of decision tree**, which further gets divided into two or more homogeneous sets.
- **Leaf Node:** It **indicates the final output node**, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** It is the **process of dividing the decision/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 to **avoid overfitting issues**.
- **Parent/Child node:** The **root/decision node is called the parent** node, and other **nodes are called the child** nodes.

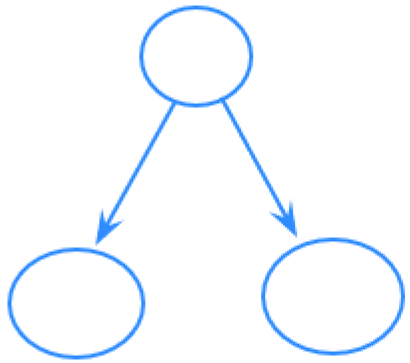
Note: A decision tree can contain categorical data (YES/NO) as well as numeric data.



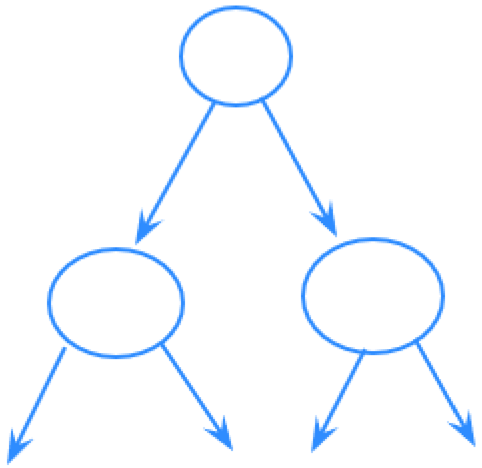
https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

Decision tree: tree like structure

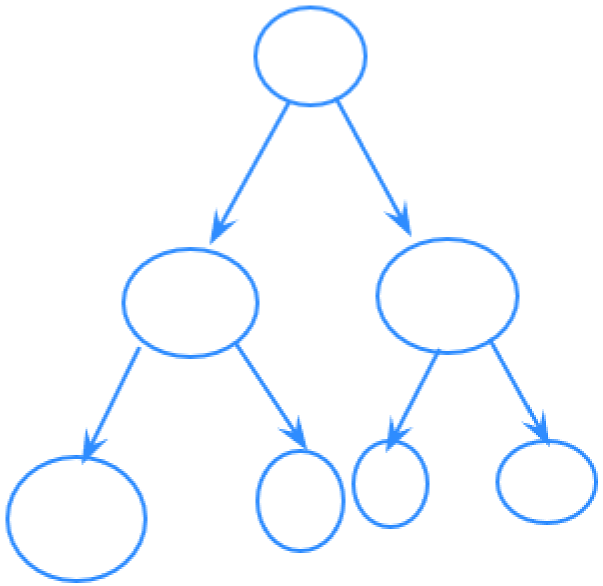
Decision tree: tree like structure



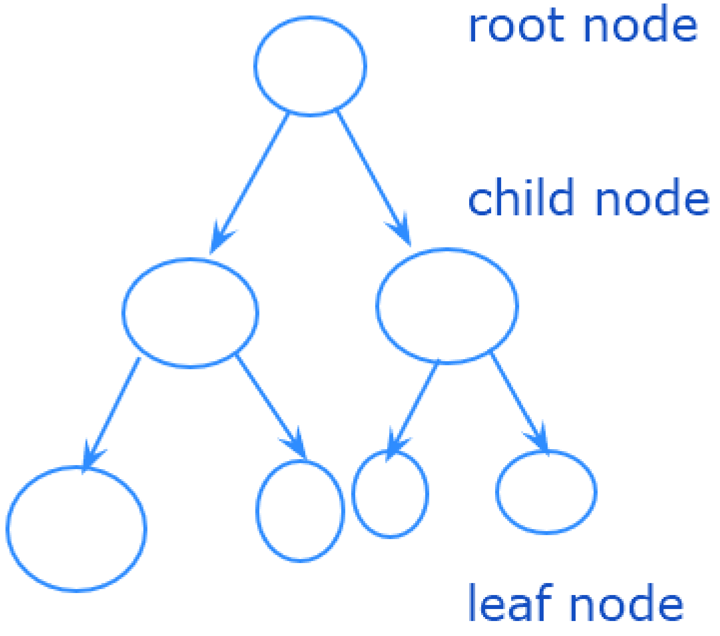
Decision tree: tree like structure



Decision tree: tree like structure

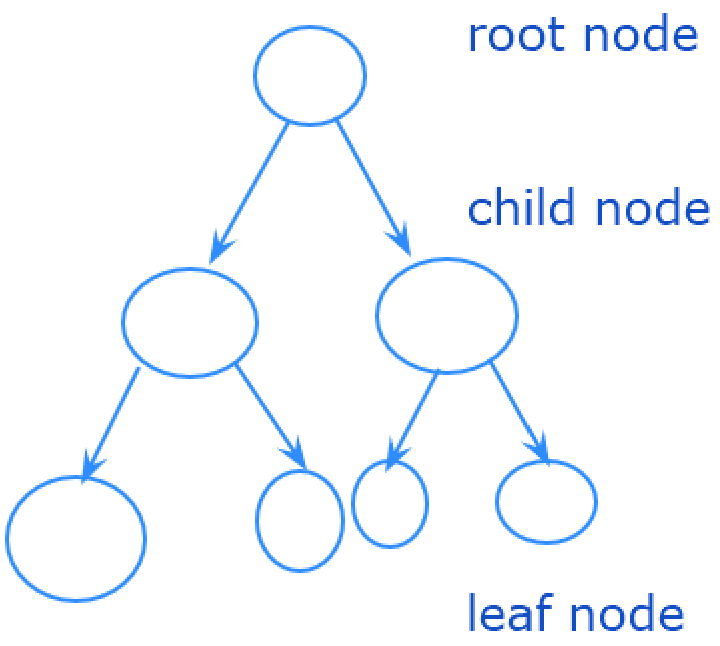


Decision tree: tree like structure



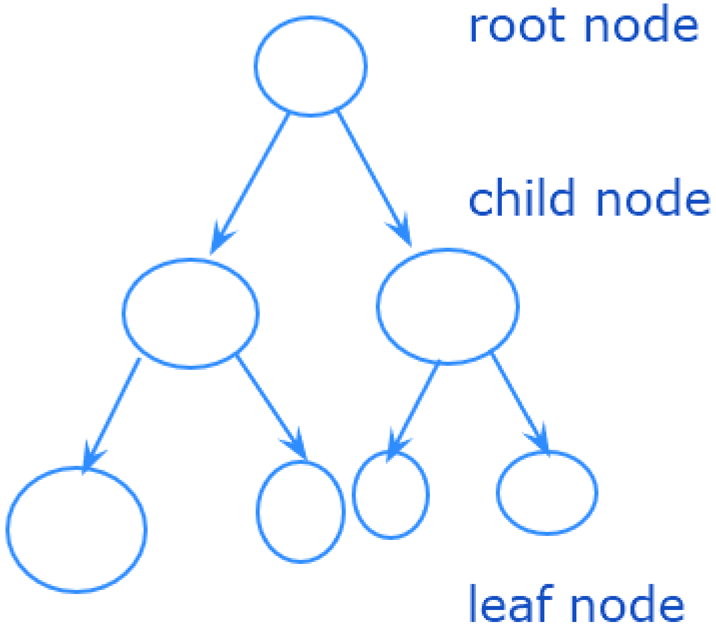
Decision tree: tree like structure

Entropy
Information gain
Gini Index= 1- summation of i=1 to n (Pi)^2



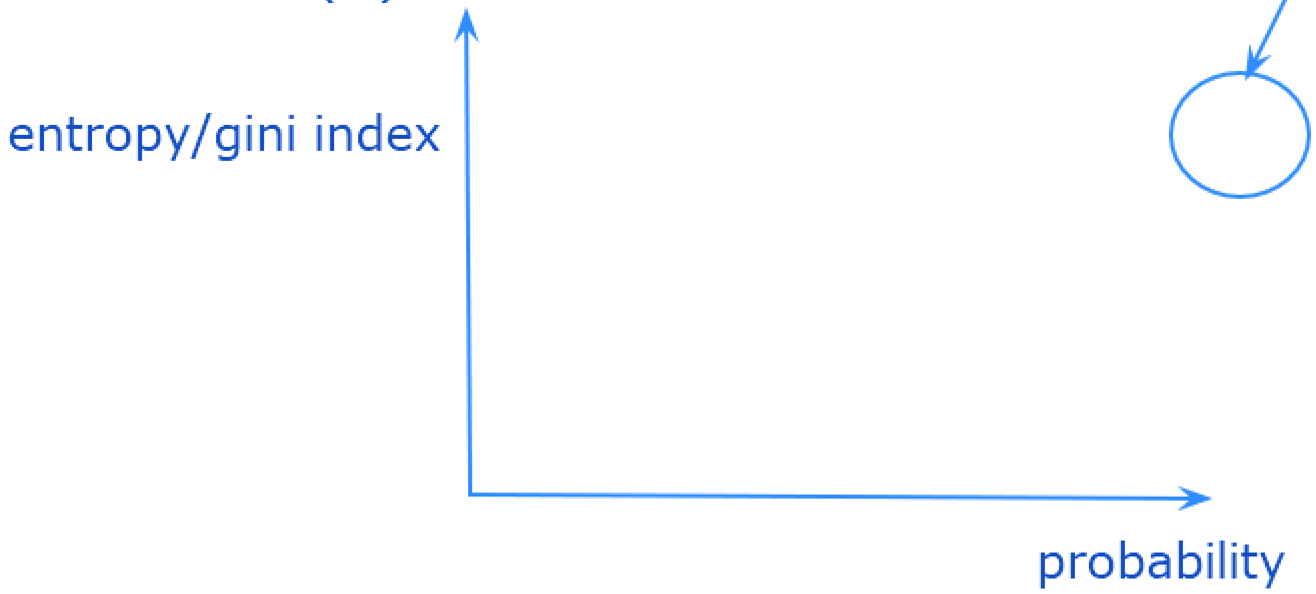
Decision tree: tree like structure

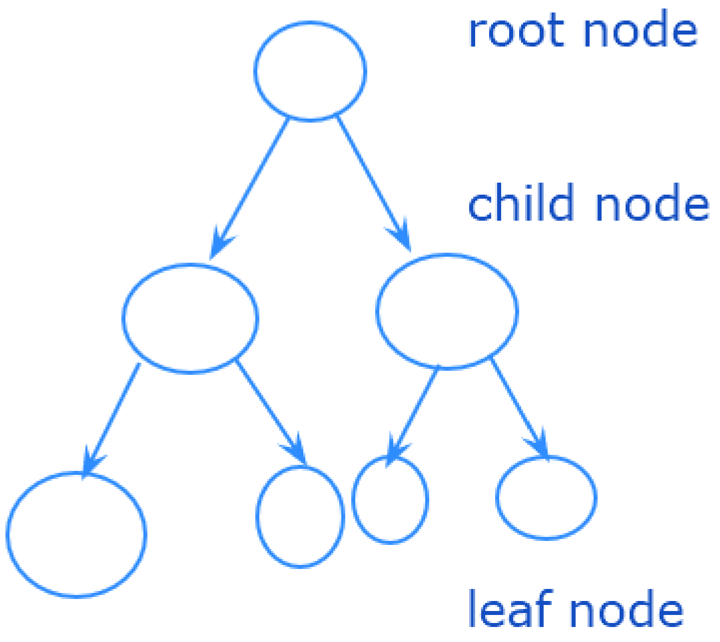
Entropy
Information gain
Gini Index= 1- summation of i=1 to n (Pi)^2



Decision tree: tree like structure

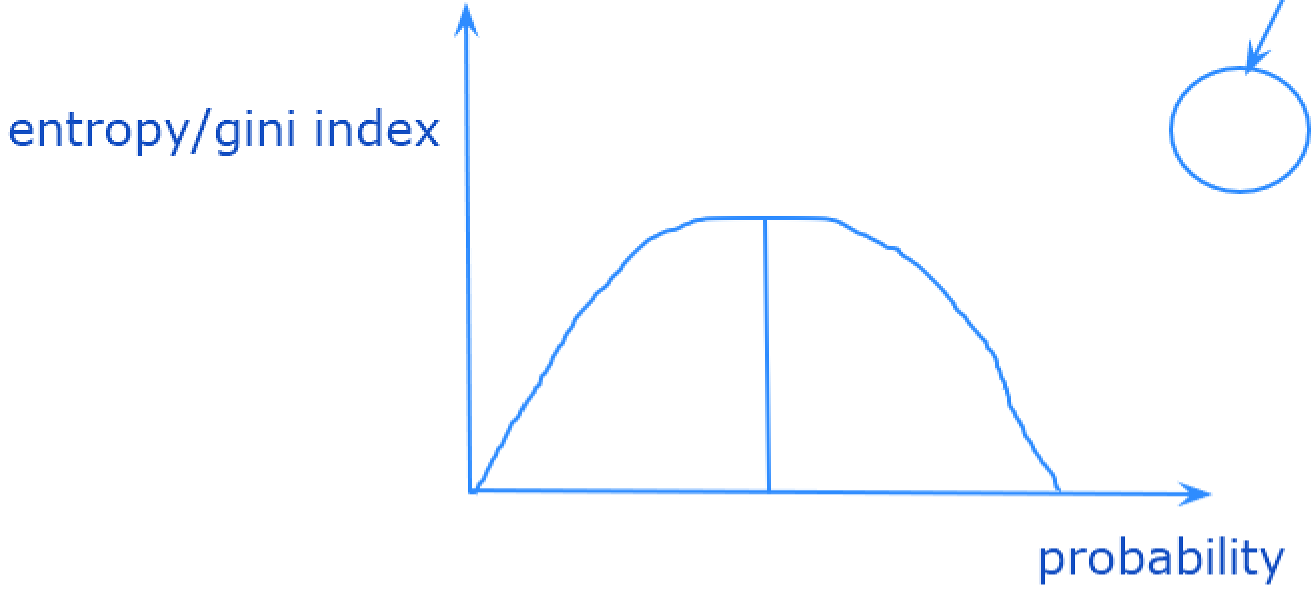
Entropy
Information gain
Gini Index= 1- summation of i=1 to n (Pi)^2

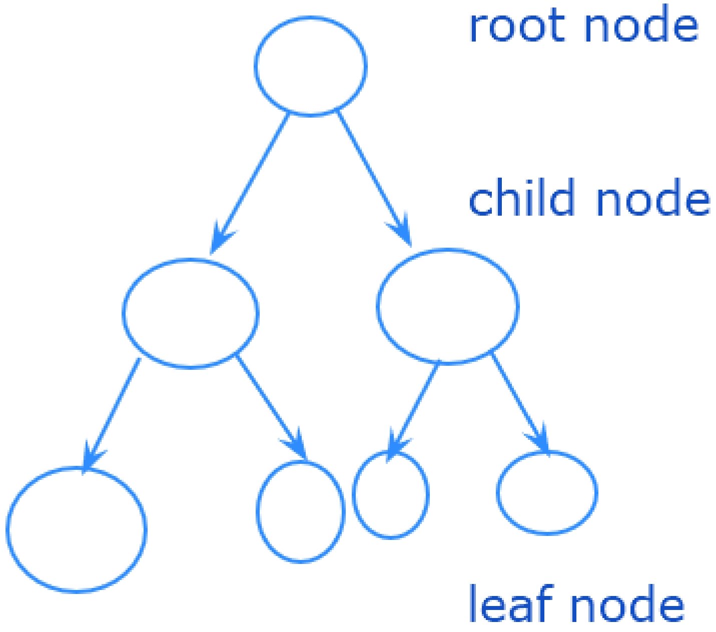




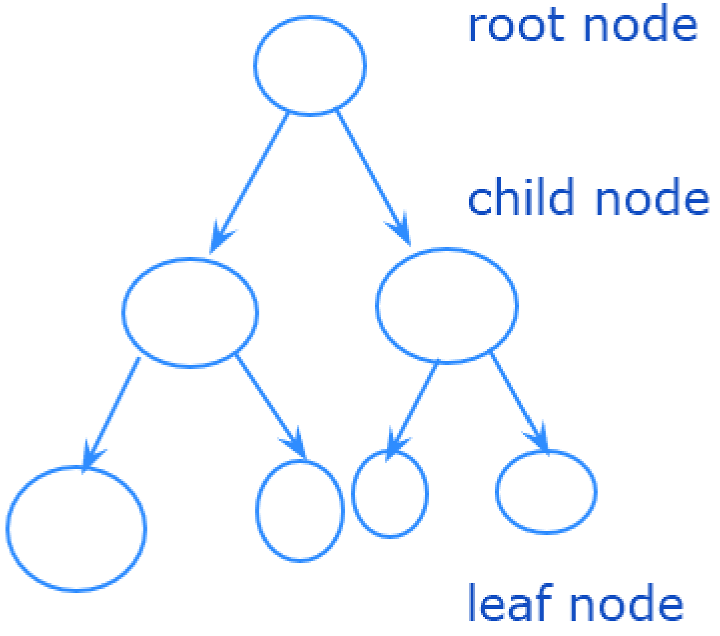
Decision tree: tree like structure

Entropy
Information gain
Gini Index= 1- summation of i=1 to n (Pi)^2

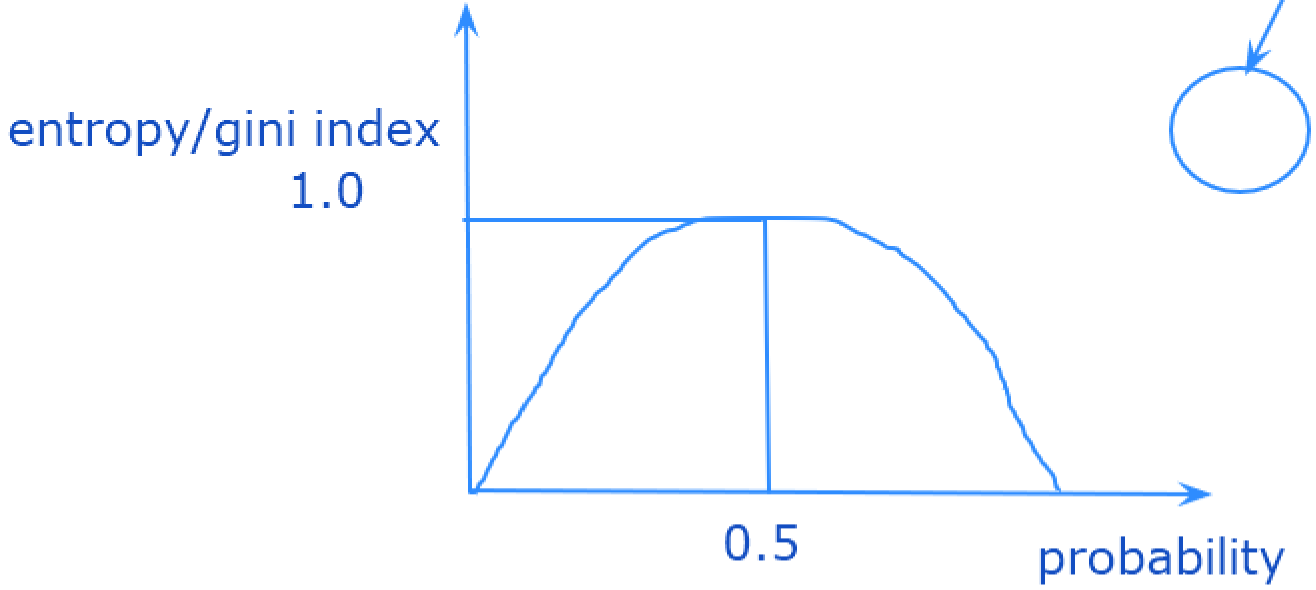




Decision tree: tree like structure

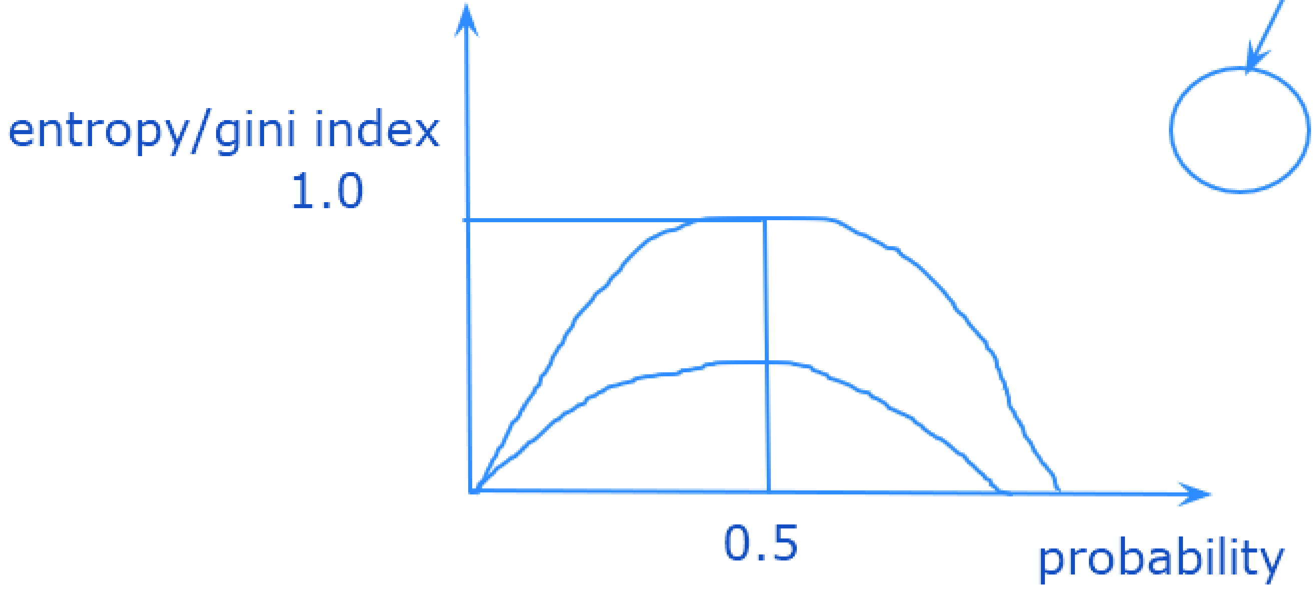


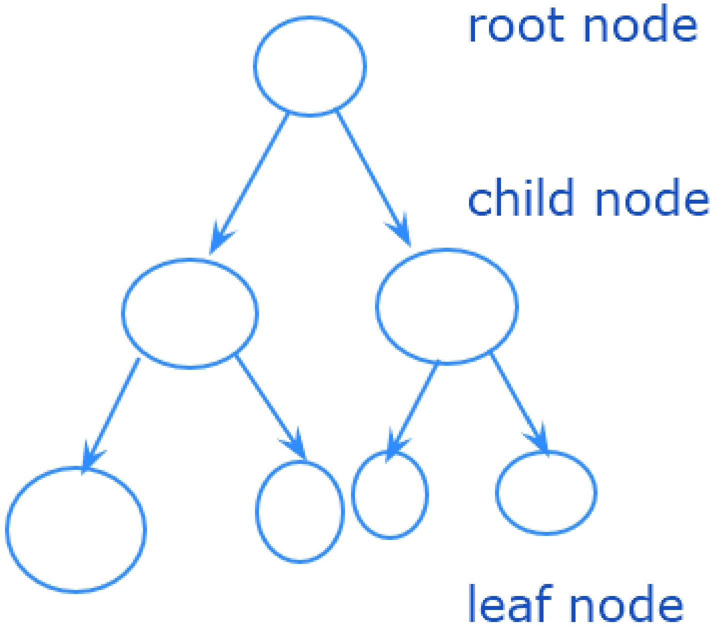
Entropy
Information gain
Gini Index= 1- summation of i=1 to n (Pi)^2



Decision tree: tree like structure

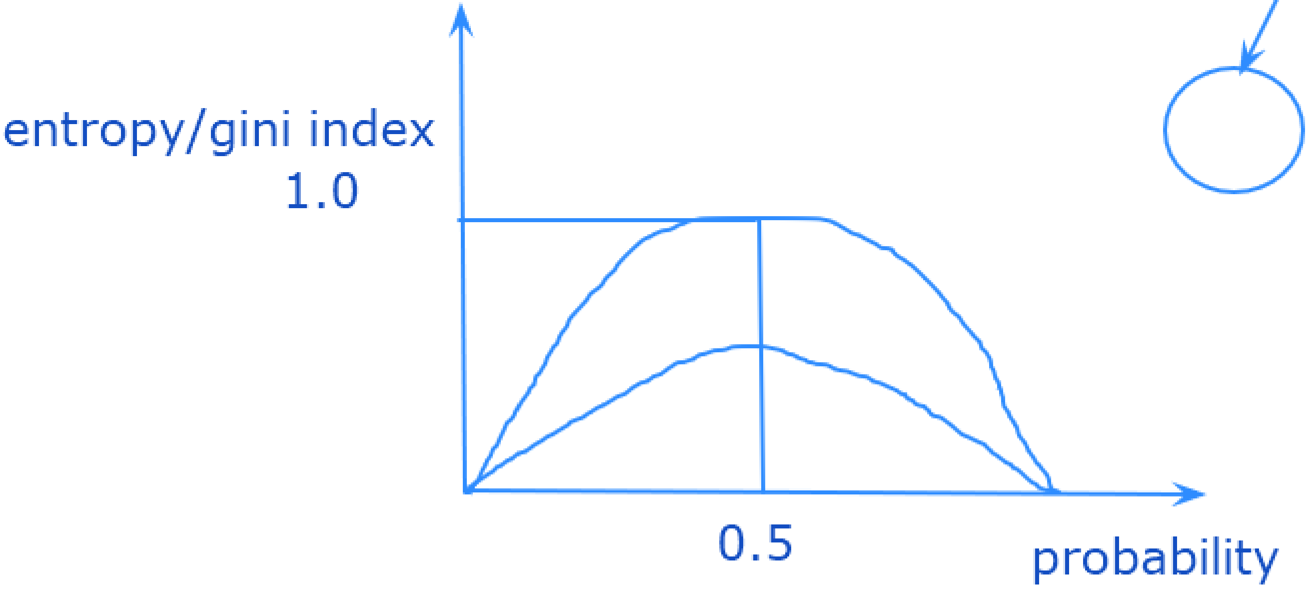
Entropy
Information gain
Gini Index= 1- summation of i=1 to n (Pi)^2

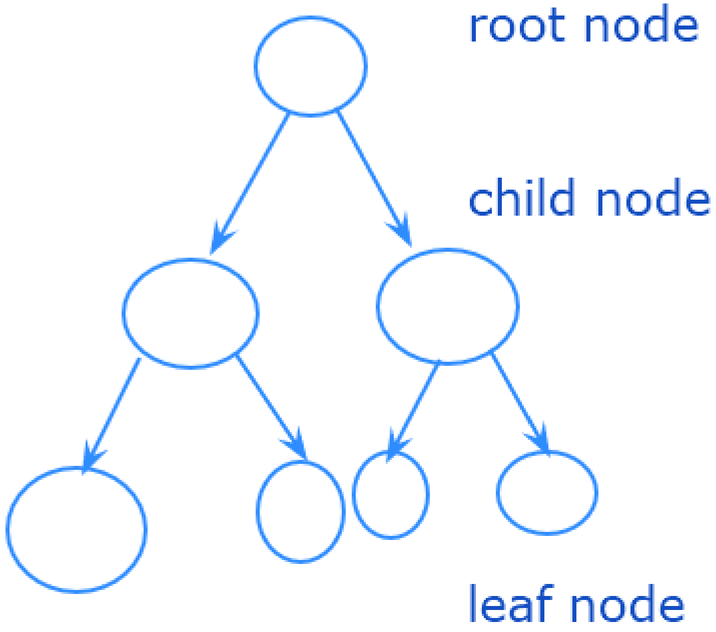




Decision tree: tree like structure

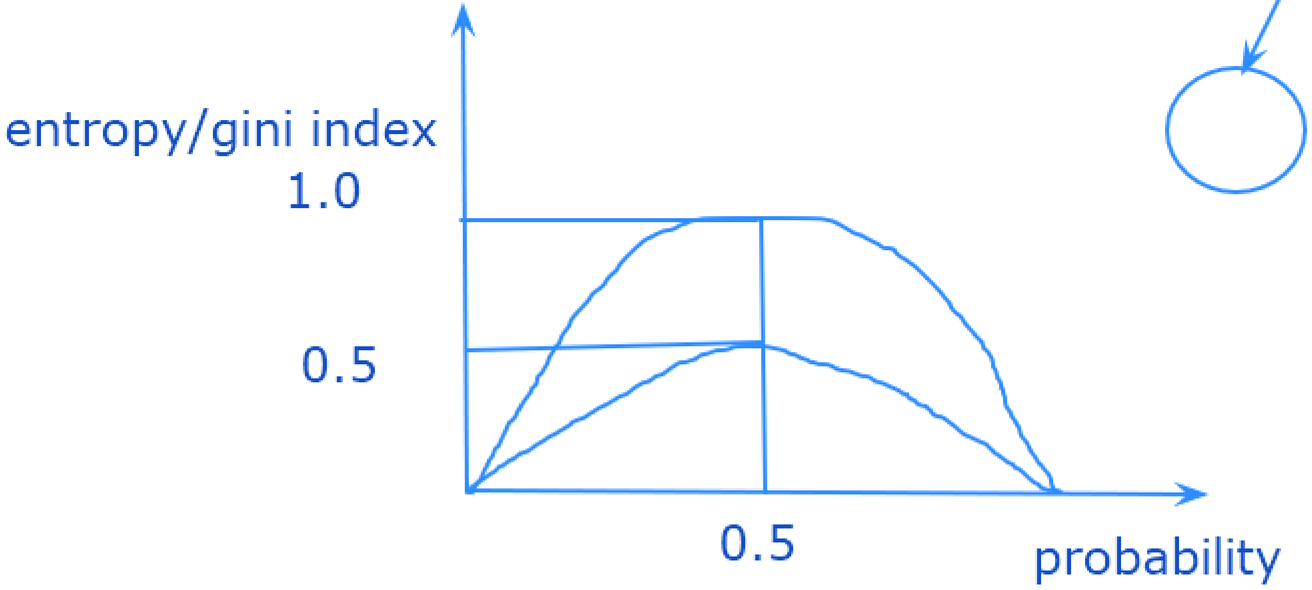
Entropy
Information gain
Gini Index= 1- summation of i=1 to n (Pi)^2

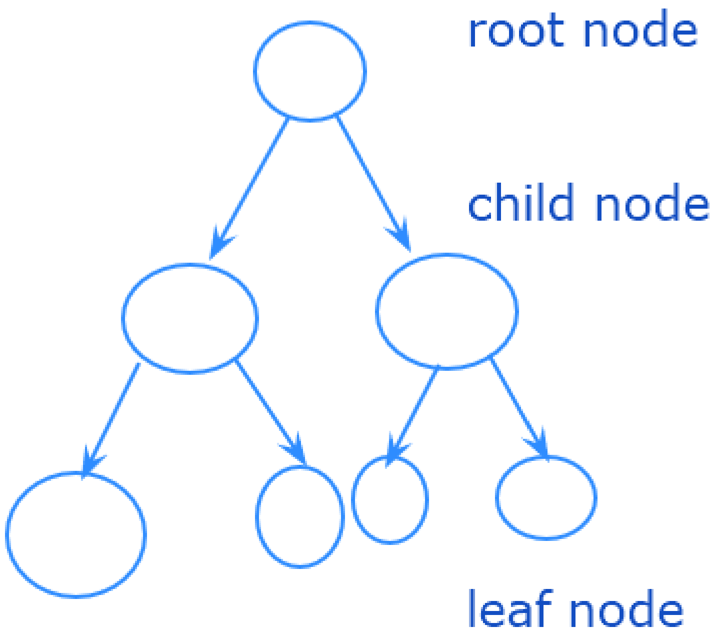




Decision tree: tree like structure

Entropy
Information gain
Gini Index= 1- summation of i=1 to n (Pi)^2





Example:

- Consider a data of 14 days with the four features that include Outlook, Temperature, Humidity, Wind and the **outcome variable is whether Golf was played on the day**. We have to build a predictive model by using the 4 parameters and predicts whether Golf will be played on the day.

Day	Outlook	Temperature	Humidity	Wind	Play Golf
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	**Overcast**	**Hot**	**High**	**Weak**	**Yes**
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	**Overcast**	**Cool**	**Normal**	**Strong**	**Yes**
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	**Overcast**	**Mild**	**High**	**Strong**	**Yes**
D13	**Overcast**	**Hot**	**Normal**	**Weak**	**Yes**
D14	Rain	Mild	High	Strong	No

Tennis player dataset

Outlook	Temp	Humidity	Wind	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	False	Yes
Rain	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rain	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rain	Mild	High	True	No

Day	Outlook	Temperature	Humidity	Wind	Play Golf
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	**Overcast**	**Hot**	**High**	**Weak**	**Yes**
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	**Overcast**	**Cool**	**Normal**	**Strong**	**Yes**
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	**Overcast**	**Mild**	**High**	**Strong**	**Yes**
D13	**Overcast**	**Hot**	**Normal**	**Weak**	**Yes**
D14	Rain	Mild	High	Strong	No

Example:

- Calculate H(S), the Entropy of the current state.
- In total there are 5 No's and 9 Yes's for total 14 outcomes.

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

Gini Index = 1- [(p+)^2 + (p-)^2]
= 1- [(9/14)2 + (5/14)2]
= 0.4592

$$\text{Entropy(S)} = -\left(\frac{9}{14}\right) \log_2 \left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) \log_2 \left(\frac{5}{14}\right) = 0.940$$

- Entropy is 0 means that all members belong to the same class, and if it 1 it indicates that half of them belong to class '0', and other half belong to class '1', which is a perfect random case.
- Here it's 0.94 means the distribution is fairly random.

- Choose the attribute that gives us highest possible Information Gain

$$IG(S, A) = H(S) - \sum_{i=1}^{n} p(x) * H(x)$$

Example:

- Lets start with "Wind".
- In total there we have 8 places where wind is weak and 6 places where wind is strong for total 14 outcomes.

P(Sweak) = Number of weak/ Total = 8/14
P(Sstrong) = Number of strong/total = 6/14

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

$$\text{Entropy(Sstrong)} = -\left(\frac{3}{6}\right) \log_2 \left(\frac{3}{6}\right) - \left(\frac{3}{6}\right) \log_2 \left(\frac{3}{6}\right) = 1$$

IG(S, Wind) = H(S) – P(Sweak)*H(Sweak) – P(Sstrong)*H(Sstrong)

$$= 0.940 - \left(\frac{8}{14}\right)(0.811) - \left(\frac{6}{14}\right)(1) = 0.048$$

- In similar way, we will calculate information gain for all other features

IG(S, Outlook) = 0.246
IG(S, Temperature) = 0.029
IG(S, Humidity) = 0.151
IG(S, Wind) = 0.048

- IG(S, Outlook) has the highest information gain of 0.246, **hence Outlook attribute is chosen as the root node**.

Example:

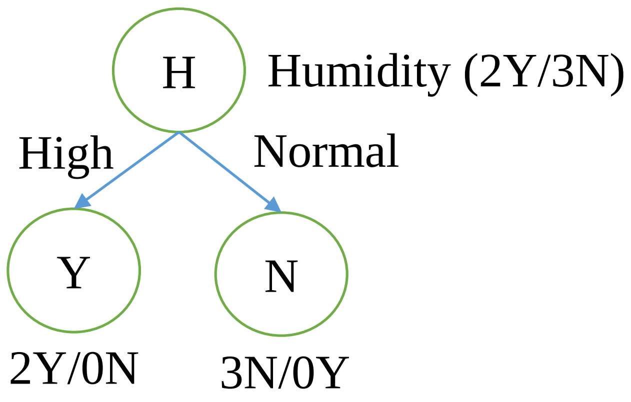
- There are three possible values of Outlook: Sunny, Overcast, and Rain.
- Overcast node already ended up having leaf node ‘Yes’, we have two subtrees to compute: Sunny and Rain.

$$H(Ssunny) = -(\frac{3}{5}) \log_2 (\frac{3}{5}) - (\frac{2}{5}) \log_2 \left(\frac{2}{5}\right) = 0.97$$

IG(Ssunny, Humidity) = 0.96
IG(Ssunny, Temperature) =0.57
IG(Ssunny, Wind) =0.019

- IG(Ssunny, Humidity) has the highest information gain of 0.96, **hence Humidity attribute is chosen.**

- Repeat the process



Detailed Calculations

Categorical values - high, normal

H(Sunny, Humidity=high) = - 0 - (3/3)*log(3/3) = 0
H(Sunny, Humidity=normal) = -(2/2)*log(2/2)-0 = 0
Average Entropy Information for Humidity –

I(Sunny, Humidity) = p(Sunny, high)*H(Sunny, Humidity=high) + p(Sunny, normal)*H(Sunny, Humidity=normal)
= (3/5)*0 + (2/5)*0 = 0 Information Gain = H(Sunny) - I(Sunny, Humidity) = 0.971 - 0 = 0.971

Categorical values - hot, mild, cool

H(Sunny, Temperature=hot) = -0-(2/2)*log(2/2) = 0
H(Sunny, Temperature=cool) = -(1)*log(1)- 0 = 0
H(Sunny, Temperature=mild) = -(1/2)*log(1/2)-(1/2)*log(1/2) = 1
Average Entropy Information for Temperature –

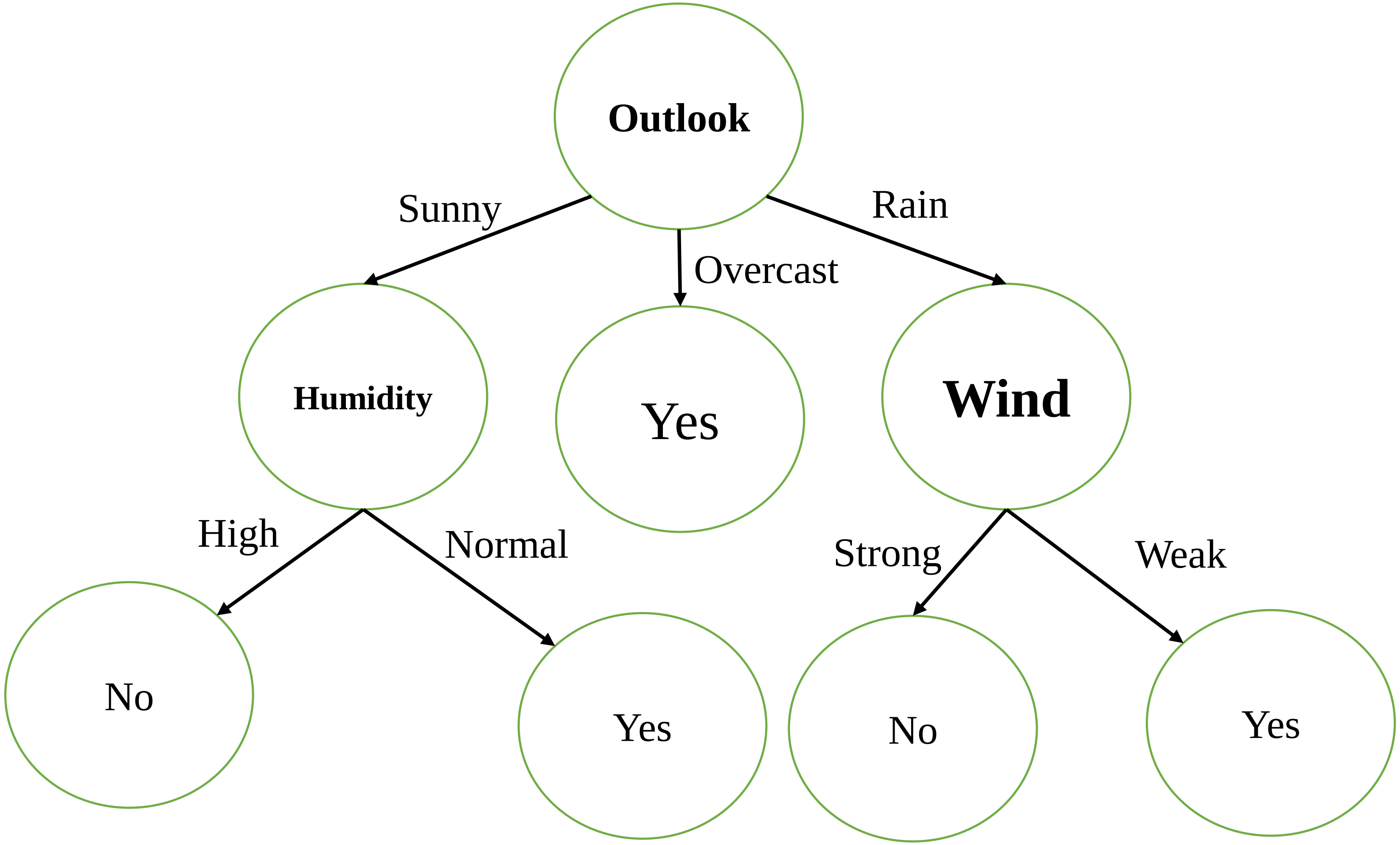
I(Sunny, Temperature) = p(Sunny, hot)*H(Sunny, Temperature=hot) + p(Sunny, mild)*H(Sunny, Temperature=mild) + p(Sunny, cool)*H(Sunny, Temperature=cool)
= (2/5)*0 + (1/5)*0 + (2/5)*1 = 0.4 Information Gain = H(Sunny) - I(Sunny, Temperature) = 0.971 - 0.4 = 0.571

Categorical values - weak, strong

H(Sunny, Wind=weak) = -(1/3)*log(1/3)-(2/3)*log(2/3) = 0.918
H(Sunny, Wind=strong) = -(1/2)*log(1/2)-(1/2)*log(1/2) = 1
Average Entropy Information for Wind –

I(Sunny, Wind) = p(Sunny, weak)*H(Sunny, Wind=weak) + p(Sunny, strong)*H(Sunny, Wind=strong)
= (3/5)*0.918 + (2/5)*1 = 0.9508 Information Gain = H(Sunny) - I(Sunny, Wind) = 0.971 - 0.9508 = 0.0202

Example:


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

Gini Index

- It is a measure of impurity that is used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.
- An attribute with the higher Gini gain should be preferred.
- It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
- Gini index can be calculated using the below formula:

$$\text{Gini Index} = 1 - \sum_j P_j^2$$

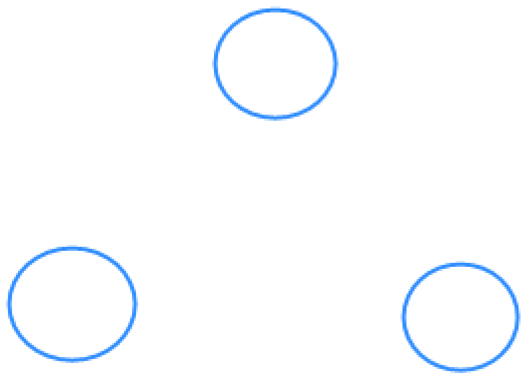
- It is used to reduce the computational time as it is free from logarithmic function so, it takes less time as compared to entropy.

X1	X2	X3	h(x)/output

X1	X2	X3	h(x)/output

Entropy: H(S)

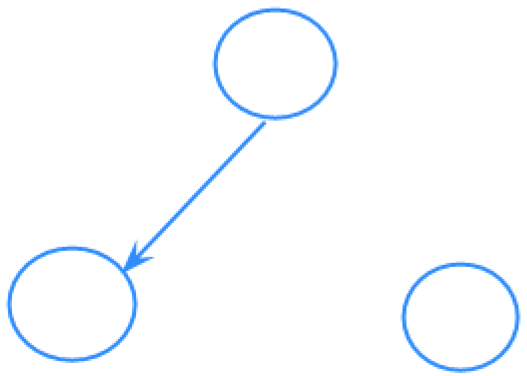
X1 X2 X3 h(x)/output



Entropy: H(S)

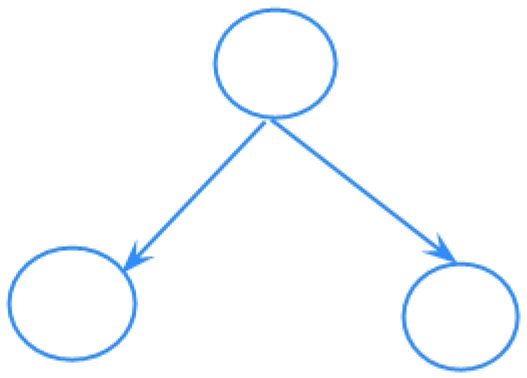
X1 X2 X3 h(x)/output

Entropy: H(S)



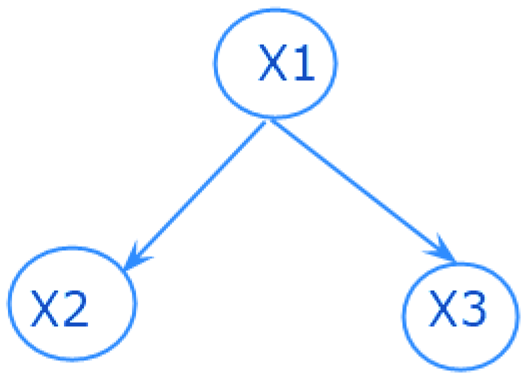
X1 X2 X3 h(x)/output

Entropy: H(S)



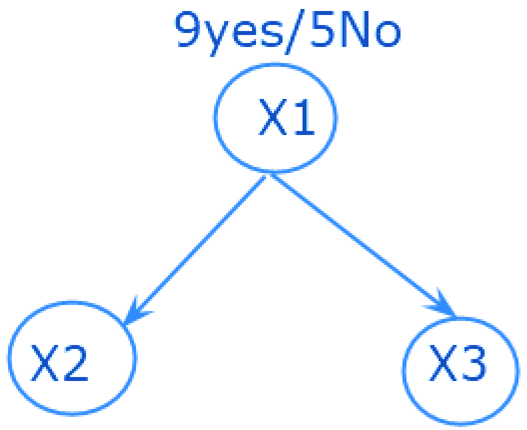
X1 X2 X3 h(x)/output

Entropy: H(S)



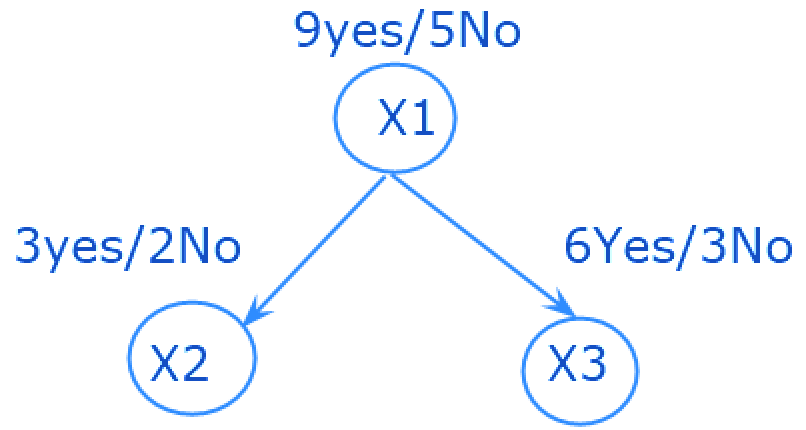
X1 X2 X3 h(x)/output

Entropy: H(S)

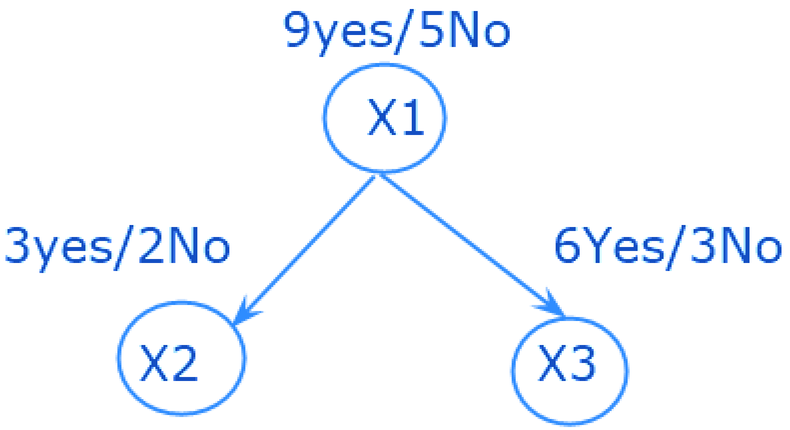


X1 X2 X3 h(x)/output

Entropy: H(S)



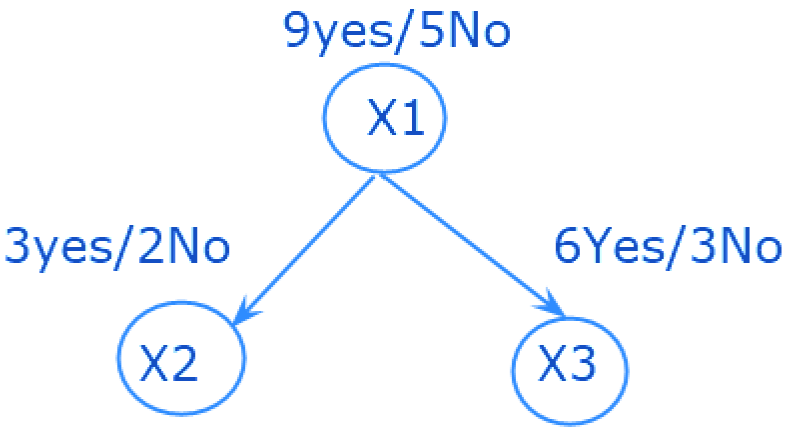
X1 X2 X3 h(x)/output



Entropy: H(S)

H(S) = -(P+) Log2 (P+) - (P-) Log2(P-)
H(X2)= -(3/5)log2(3/5) -(2/5)log2(2/5)
H(X2) = 0.97 bits
H(X3) = -(6/9)log2(6/9) -(3/9)log2(3/9)
H(X3) = 0.918 bits
H(X1) = -(9/14)log2(9/14) - (5/14)log2(5/14)
H(X1) = 0.94 bits

X1 X2 X3 h(x)/output



Entropy: H(S)

H(S) = -(P+) Log2 (P+) - (P-) Log2(P-)
H(X2)= -(3/5)log2(3/5) -(2/5)log2(2/5)
H(X2) = 0.97 bits
H(X3) = -(6/9)log2(6/9) -(3/9)log2(3/9)
H(X3) = 0.918 bits
H(X1) = -(9/14)log2(9/14) - (5/14)log2(5/14)
H(X1) = 0.94 bits

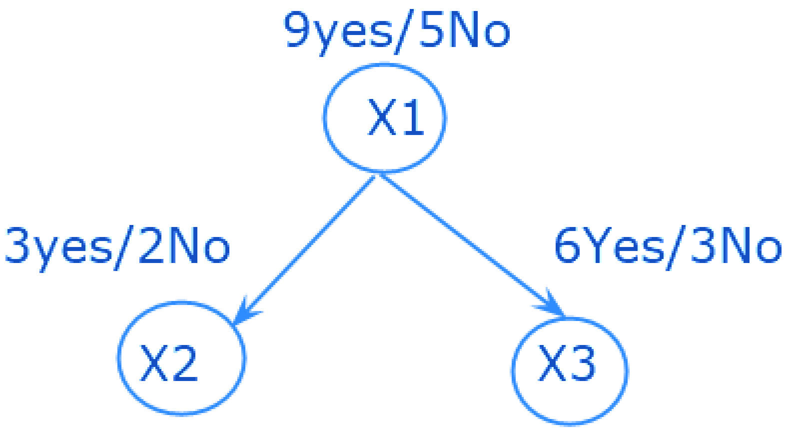
H(S)=1
H(S)=0

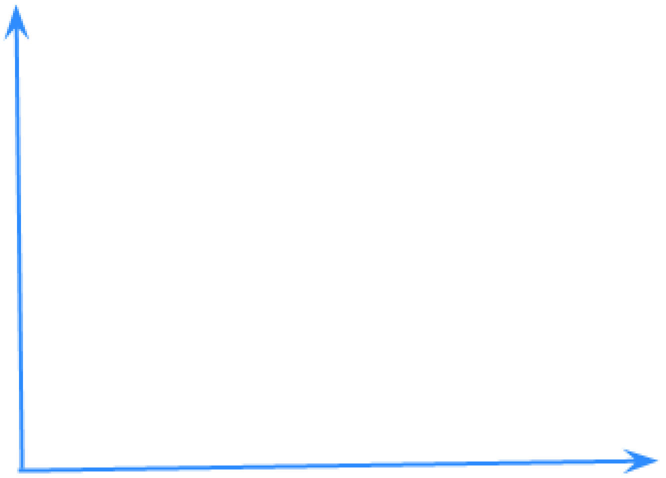
X1 X2 X3 h(x)/output

Entropy: H(S)

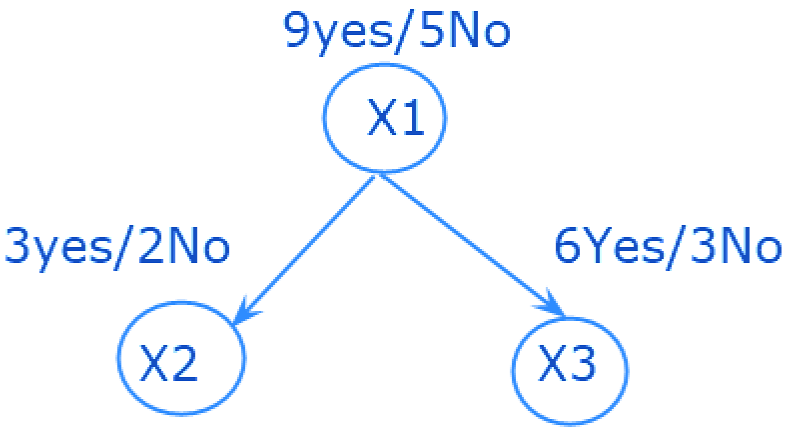
$$H(S) = -(P+)\ Log2\ (P+) - (P-)\ Log2(P-)$$
$$H(X2) = -(3/5)\log2(3/5) - (2/5)\log2(2/5)$$
$$H(X2) = 0.97 \text{ bits}$$
$$H(X3) = -(6/9)\log2(6/9) - (3/9)\log2(3/9)$$
$$H(X3) = 0.918 \text{ bits}$$
$$H(X1) = -(9/14)\log2(9/14) - (5/14)\log2(5/14)$$
$$H(X1) = 0.94 \text{ bits}$$

$H(S)=1$
$H(S)=0$



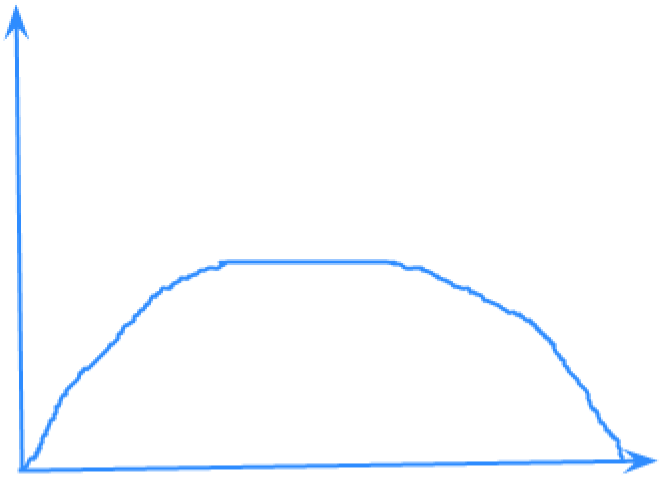


X1 X2 X3 h(x)/output



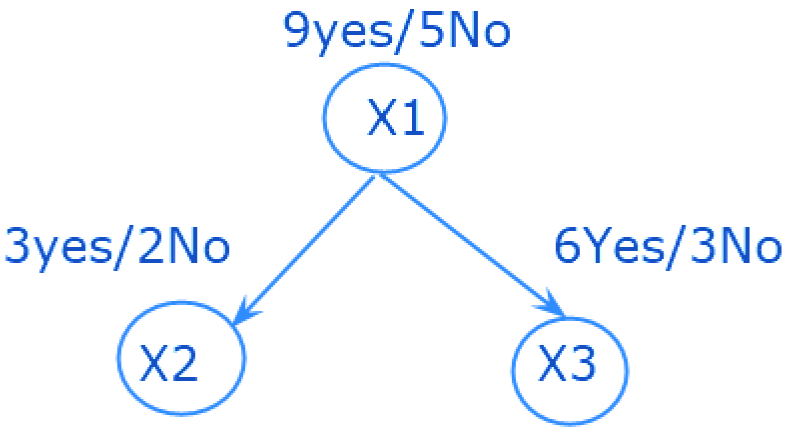
Entropy: H(S)

$$H(S) = -(P+)\ Log2\ (P+) - (P-)\ Log2(P-)$$
$$H(X2) = -(3/5)\log2(3/5) - (2/5)\log2(2/5)$$
$$H(X2) = 0.97 \text{ bits}$$
$$H(X3) = -(6/9)\log2(6/9) - (3/9)\log2(3/9)$$
$$H(X3) = 0.918 \text{ bits}$$
$$H(X1) = -(9/14)\log2(9/14) - (5/14)\log2(5/14)$$
$$H(X1) = 0.94 \text{ bits}$$



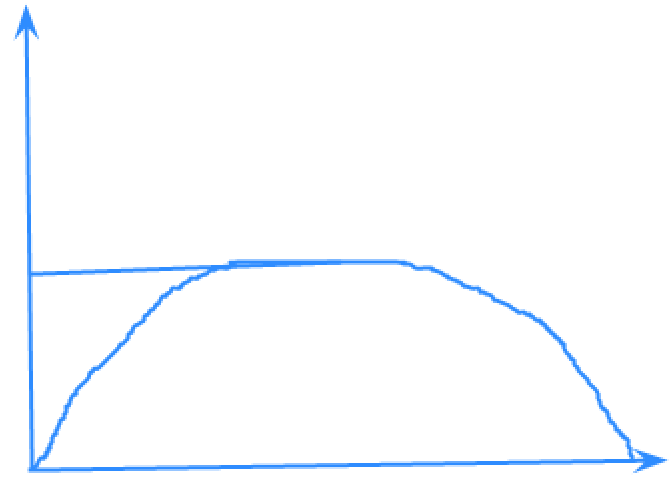
$$H(S)=1$$
$$H(S)=0$$

X1 X2 X3 h(x)/output



Entropy: H(S)

H(S) = -(P+) Log2 (P+) - (P-) Log2(P-)
H(X2)= -(3/5)log2(3/5) -(2/5)log2(2/5)
H(X2) = 0.97 bits
H(X3) = -(6/9)log2(6/9) -(3/9)log2(3/9)
H(X3) = 0.918 bits
H(X1) = -(9/14)log2(9/14) - (5/14)log2(5/14)
H(X1) = 0.94 bits



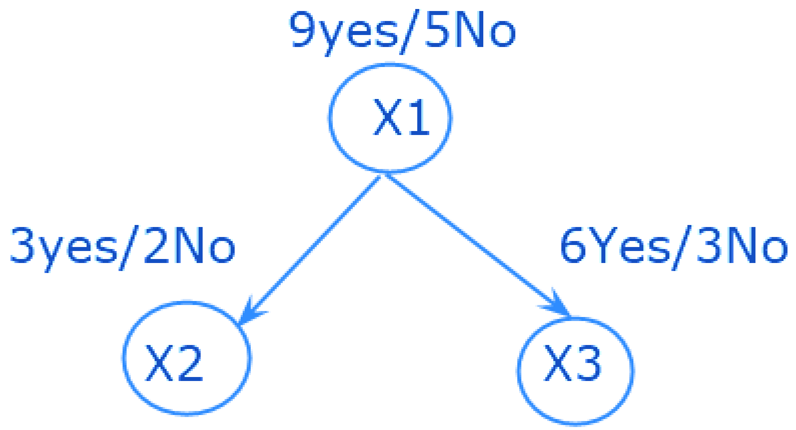
H(S)=1
H(S)=0

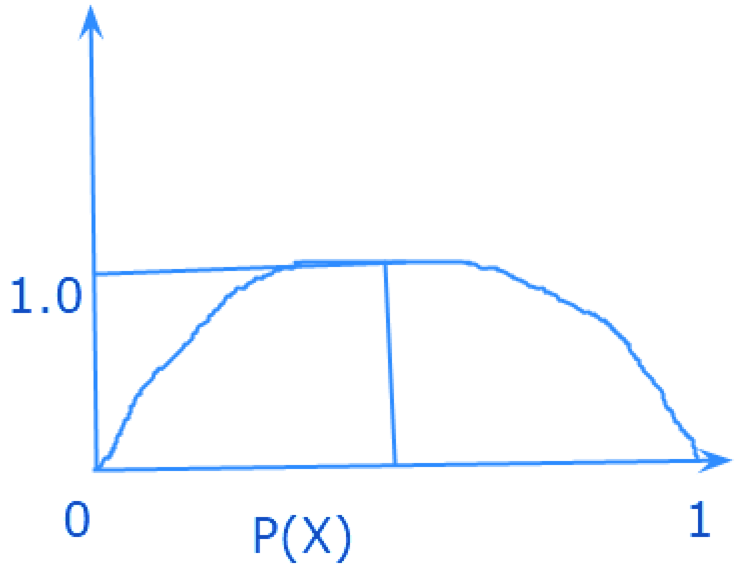
X1 X2 X3 h(x)/output

Entropy: H(S)

H(S) = -(P+) Log2 (P+) - (P-) Log2(P-)
H(X2)= -(3/5)log2(3/5) -(2/5)log2(2/5)
H(X2) = 0.97 bits
H(X3) = -(6/9)log2(6/9) -(3/9)log2(3/9)
H(X3) = 0.918 bits
H(X1) = -(9/14)log2(9/14) - (5/14)log2(5/14)
H(X1) = 0.94 bits

H(S)=1
H(S)=0





X1 X2 X3 h(x)/output

Entropy: H(S)

$H(S) = -(P+) \, Log2 \, (P+) - (P-) \, Log2(P-)$

$H(X2) = -(3/5)log2(3/5) - (2/5)log2(2/5)$

$H(X2) = 0.97$ bits

$H(X3) = -(6/9)log2(6/9) - (3/9)log2(3/9)$

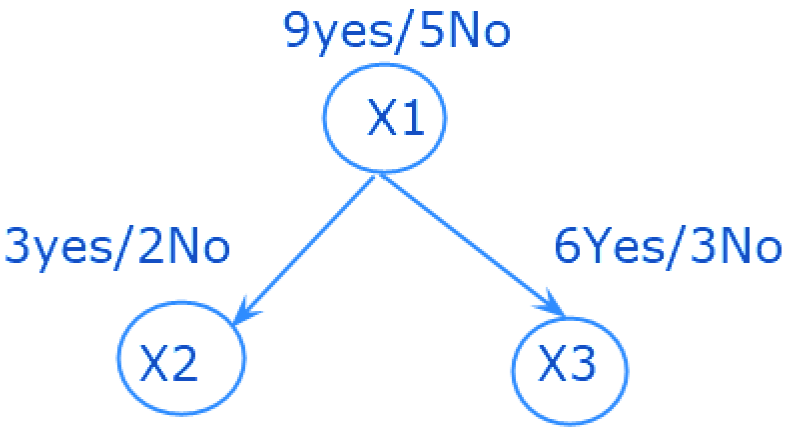
$H(X3) = 0.918$ bits

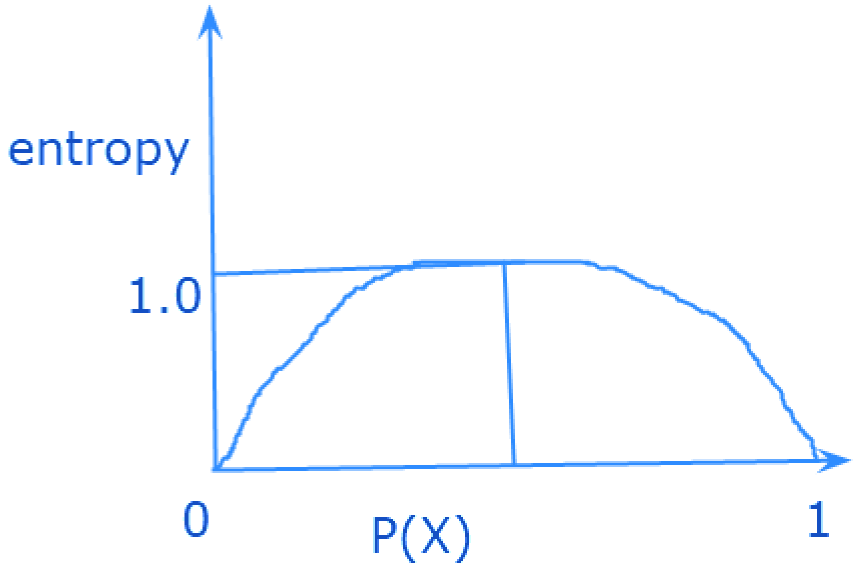
$H(X1) = -(9/14)log2(9/14) - (5/14)log2(5/14)$

$H(X1) = 0.94$ bits

$H(S)=1$

$H(S)=0$



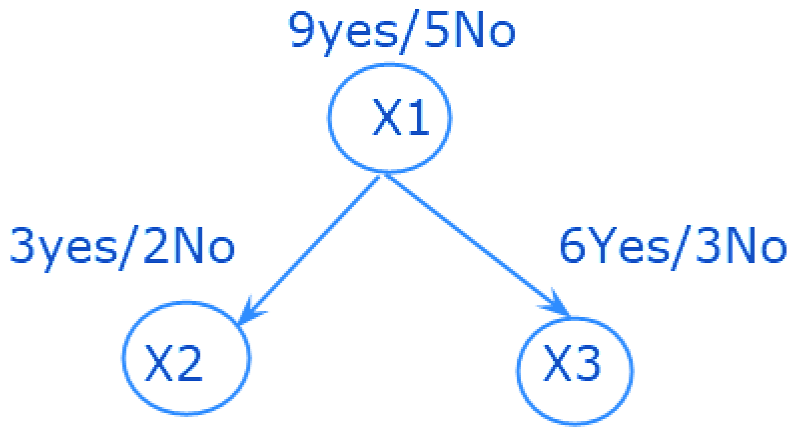


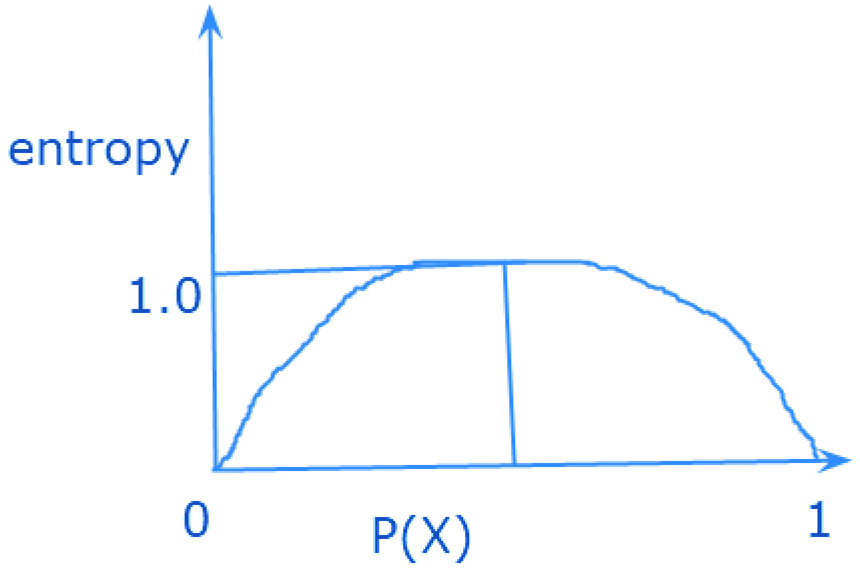
X1 X2 X3 h(x)/output

Entropy: H(S)

H(S) = -(P+) Log2 (P+) - (P-) Log2(P-)
H(X2)= -(3/5)log2(3/5) -(2/5)log2(2/5)
H(X2) = 0.97 bits
H(X3) = -(6/9)log2(6/9) -(3/9)log2(3/9)
H(X3) = 0.918 bits
H(X1) = -(9/14)log2(9/14) - (5/14)log2(5/14)
H(X1) = 0.94 bits

H(S)=1 50% chance of yes/no: Impure
H(S)=0 100% yes/no: Pure



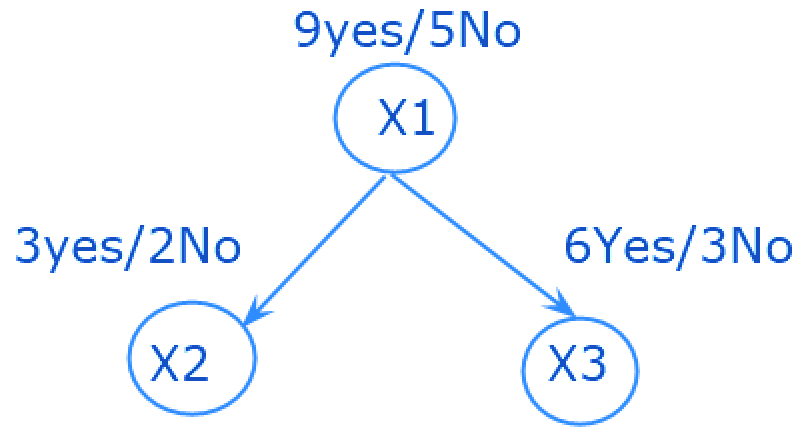


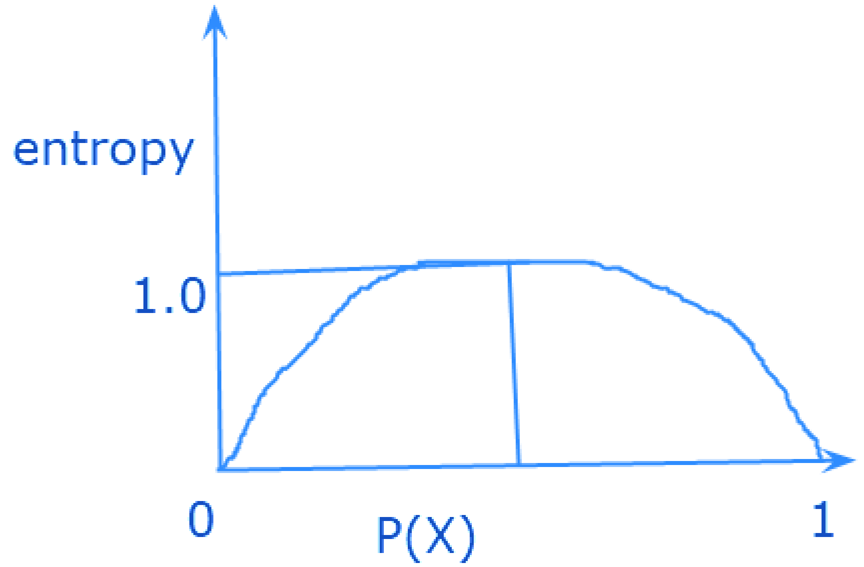
X1 X2 X3 h(x)/output

Entropy: H(S)

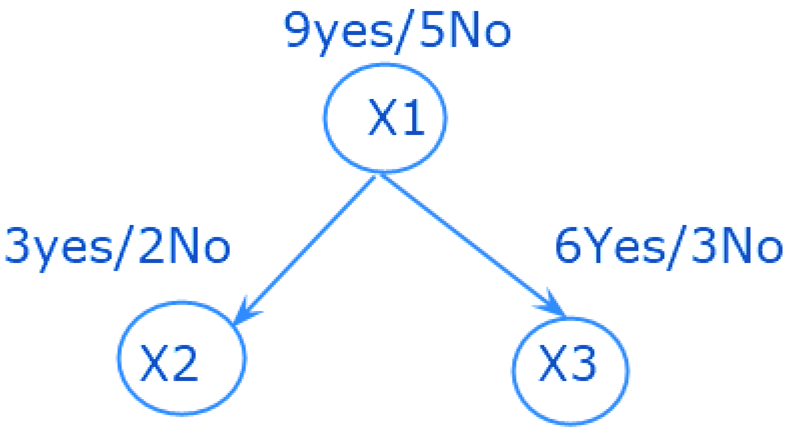
$H(S) = -(P+)\ Log2\ (P+) - (P-)\ Log2(P-)$
$H(X2) = -(3/5)log2(3/5) - (2/5)log2(2/5)$
$H(X2) = 0.97$ bits
$H(X3) = -(6/9)log2(6/9) - (3/9)log2(3/9)$
$H(X3) = 0.918$ bits
$H(X1) = -(9/14)log2(9/14) - (5/14)log2(5/14)$
$H(X1) = 0.94$ bits

H(S)=1 50% chance of yes/no: Impure
$\boxed{\text{H(S)=0 100\% yes/no: Pure}}$





X1 X2 X3 h(x)/output



Entropy: H(S)

$H(S) = -(P+)\ Log2\ (P+) - (P-)\ Log2(P-)$

$H(X2)= -(3/5)log2(3/5) -(2/5)log2(2/5)$

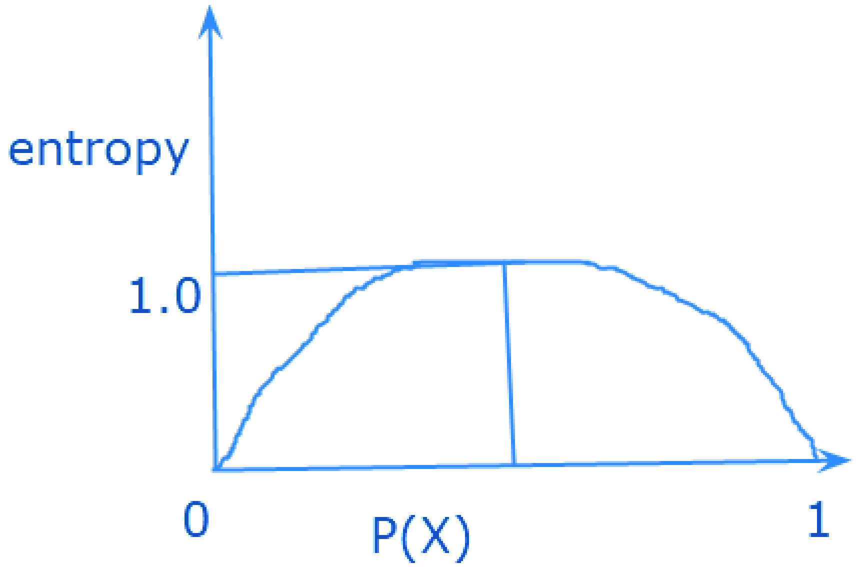
$H(X2) = 0.97$ bits

$H(X3) = -(6/9)log2(6/9) -(3/9)log2(3/9)$

$H(X3) = 0.918$ bits

$H(X1) = -(9/14)log2(9/14) - (5/14)log2(5/14)$

$H(X1) = 0.94$ bits

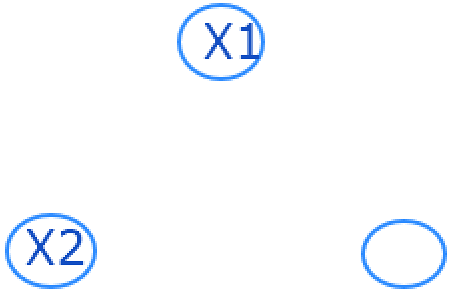


H(S)=1 50% chance of yes/no: Impure

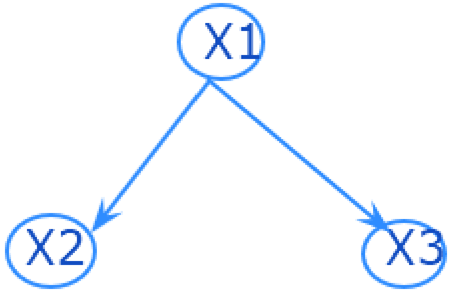
H(S)=0 100% yes/no: Pure

Information gain:

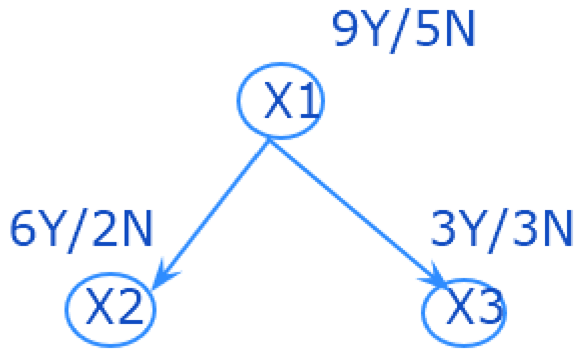
Information gain:



Information gain:

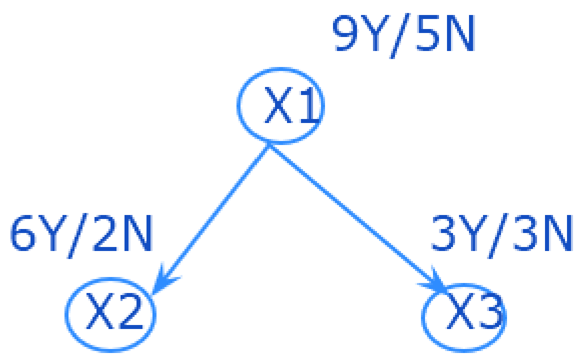


Information gain:



Information gain:

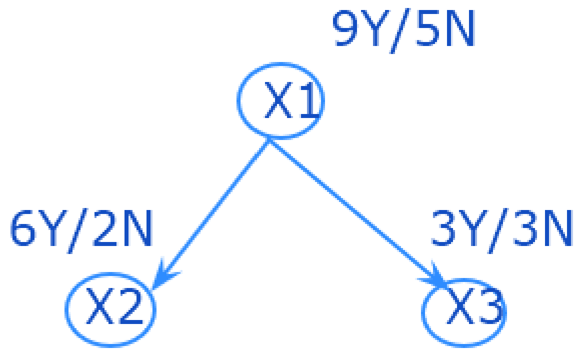
$$\begin{aligned} Gain(S,X1) &= 0.94 - (8/14)H(X2) - (6/14)(H(X3)) \\ &= 0.94 - (8/14) * 0.81 - (6/14) * 1 \\ &= 0.049 \end{aligned}$$



Information gain:

Gain(S,X1) = 0.94 - (8/14)H(X2) - (6/14)(H(X3))
= 0.94 -(8/14) * 0.81 - (6/14) * 1
= 0.049

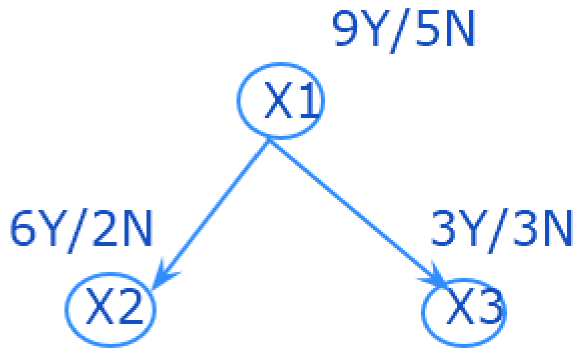
Gain(S, A)= H(S) - summation(Sv)/(S) H(Sv)

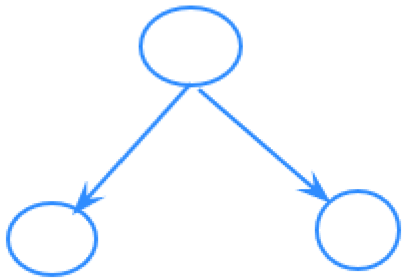


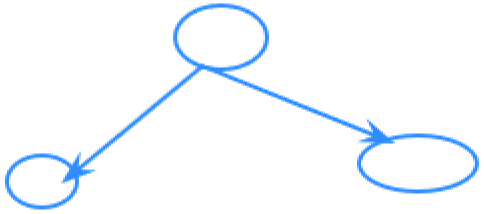
Information gain:

$$\text{Gain}(S, X1) = 0.94 - (8/14)H(X2) - (6/14)(H(X3))$$
$$= 0.94 - (8/14) * 0.81 - (6/14) * 1$$
$$= 0.049$$

$$\text{Gain}(S, A) = H(S) - \text{summation}(Sv)/(S)\ H(Sv)$$



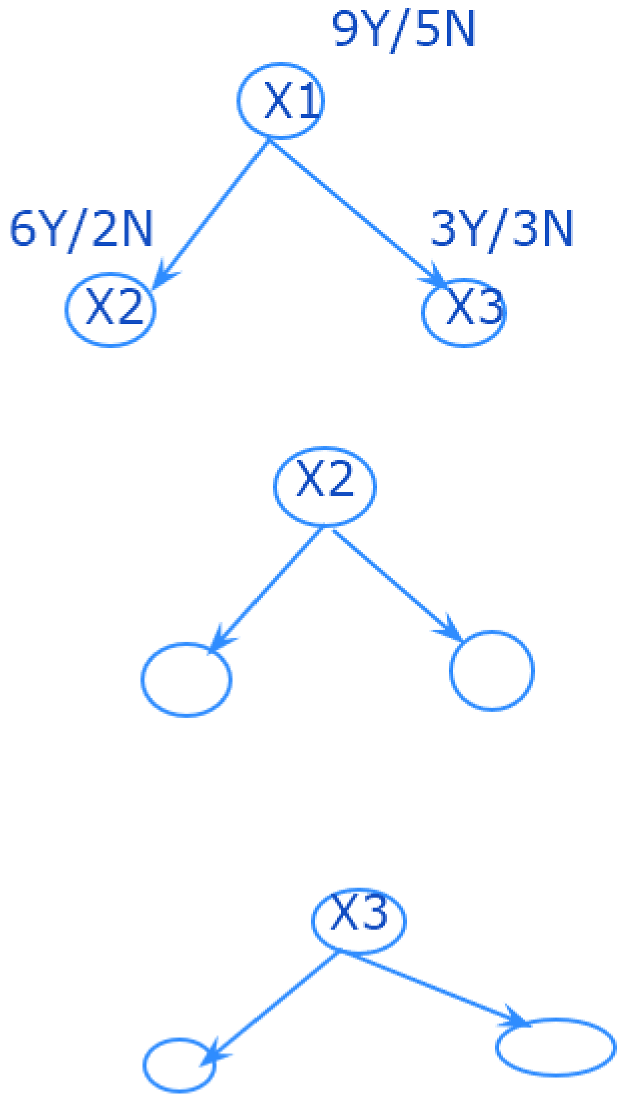




Information gain:

Gain(S,X1) = 0.94 - (8/14)H(X2) - (6/14)(H(X3))
= 0.94 -(8/14) * 0.81 - (6/14) * 1
= 0.049

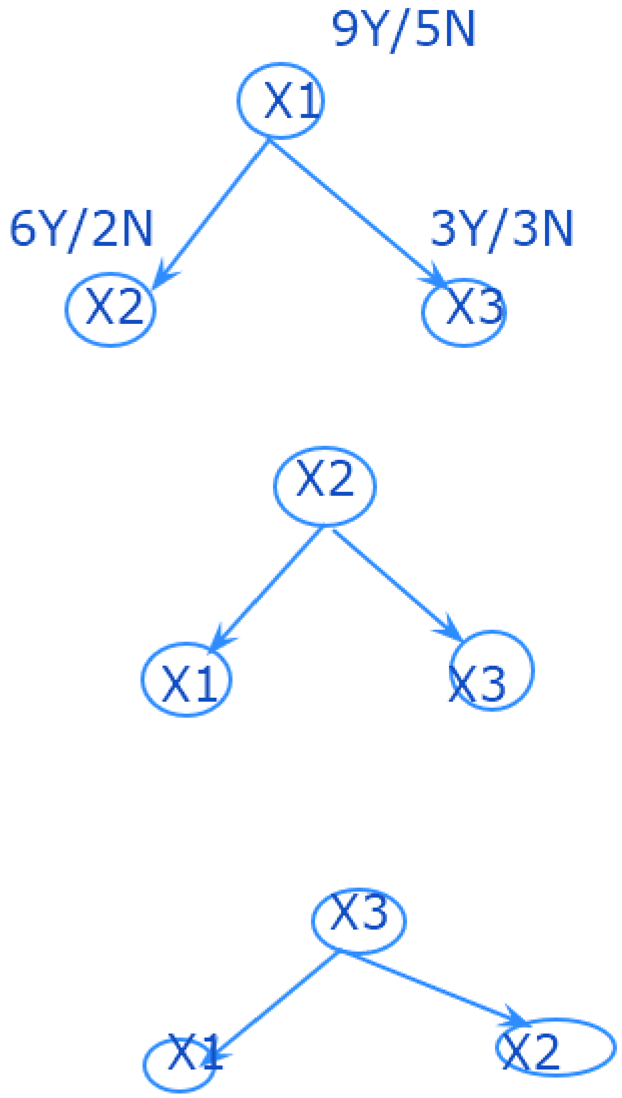
Gain(S, A)= H(S) - summation(Sv)/(S) H(Sv)



Information gain:

Gain(S,X1) = 0.94 - (8/14)H(X2) - (6/14)(H(X3))
= 0.94 -(8/14) * 0.81 - (6/14) * 1
= 0.049

Gain(S, A)= H(S) - summation(Sv)/(S) H(Sv)

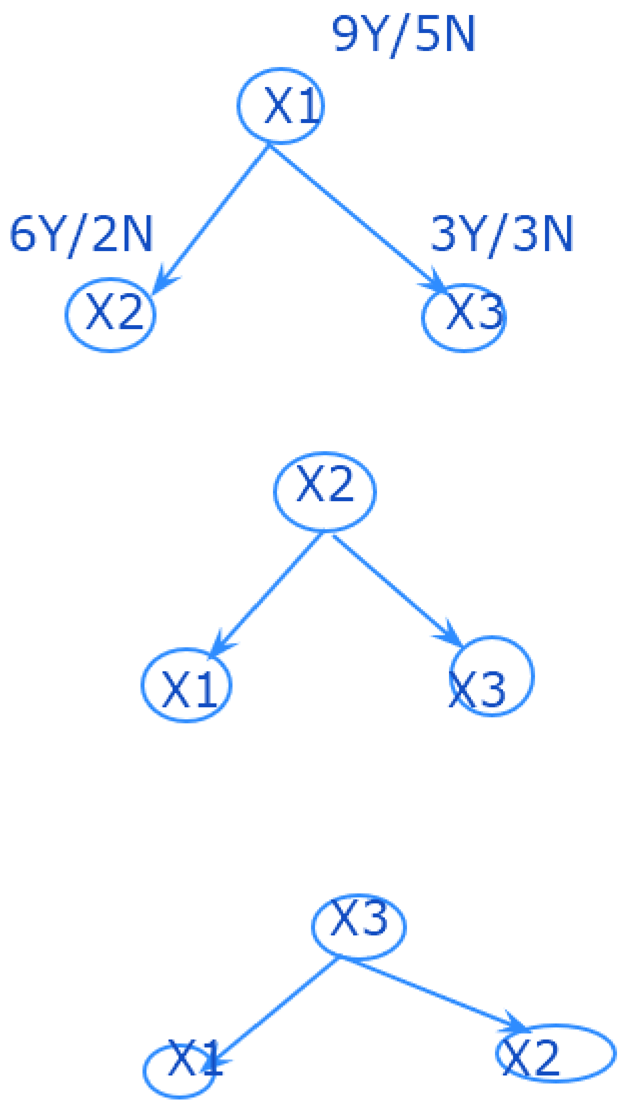


Information gain:

Gain(S,X1) = 0.94 - (8/14)H(X2) - (6/14)(H(X3))
= 0.94 -(8/14) * 0.81 - (6/14) * 1
= 0.049

Gain(S, A)= H(S) - summation(Sv)/(S) H(Sv)

Choose the one which gives maximum information gain

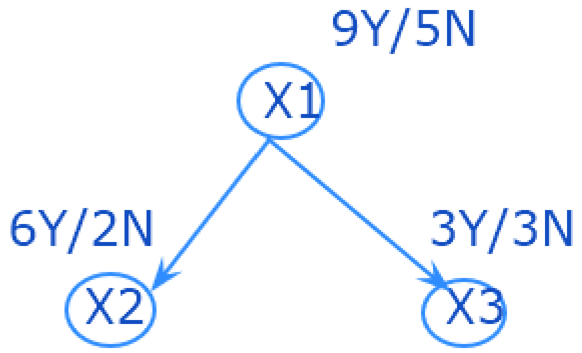


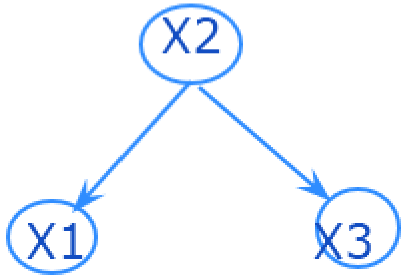
Information gain: ID3

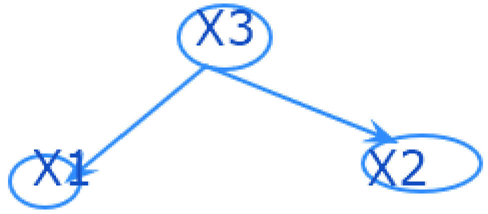
$$\text{Gain}(S, X1) = 0.94 - (8/14)H(X2) - (6/14)(H(X3))$$
$$= 0.94 - (8/14) * 0.81 - (6/14) * 1$$
$$= 0.049$$

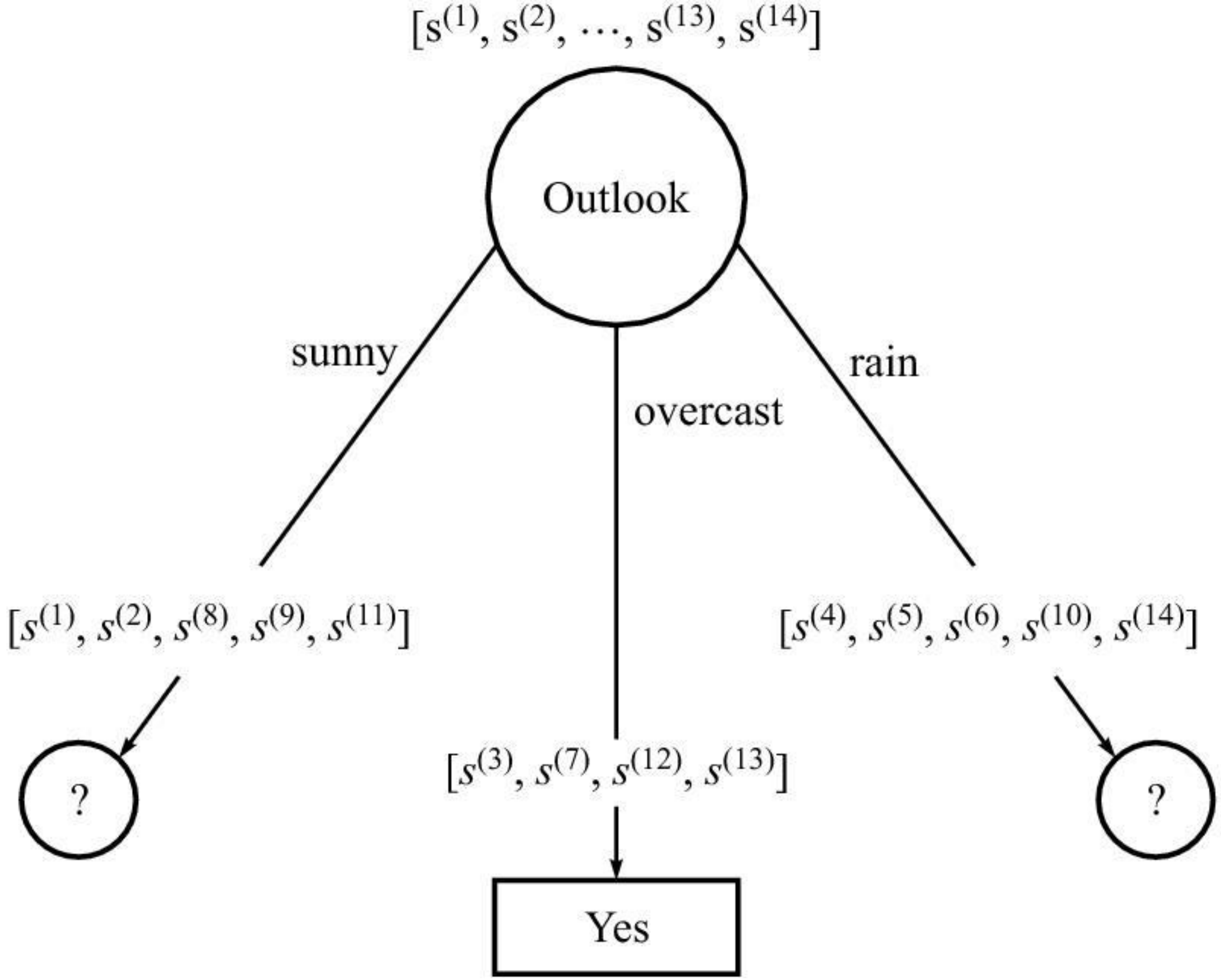
$$\text{Gain}(S, A) = H(S) - \text{summation}(Sv)/(S)\ H(Sv)$$

Choose the one which gives maximum information gain









Partially learned decision tree: the training examples are sorted to corresponding descendant nodes