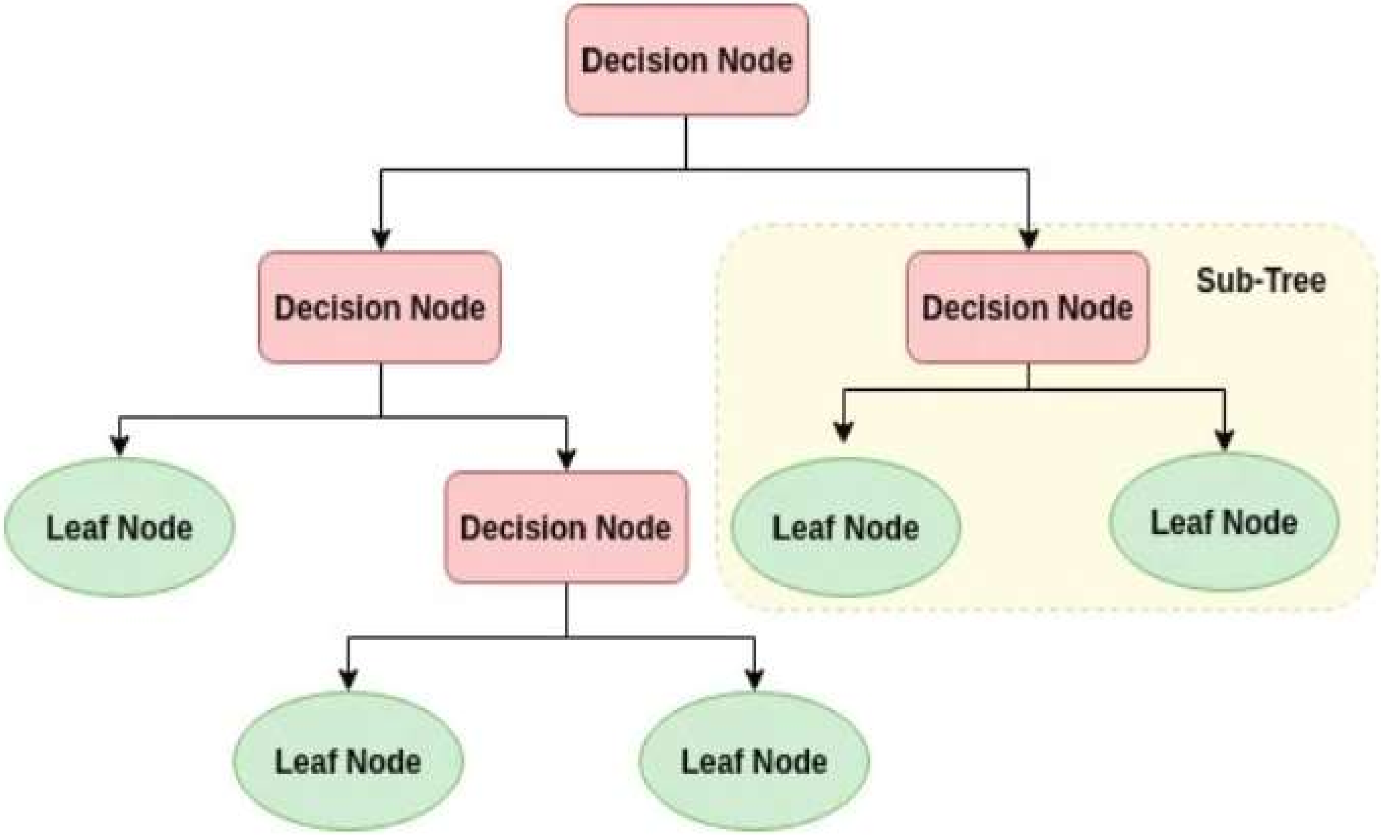
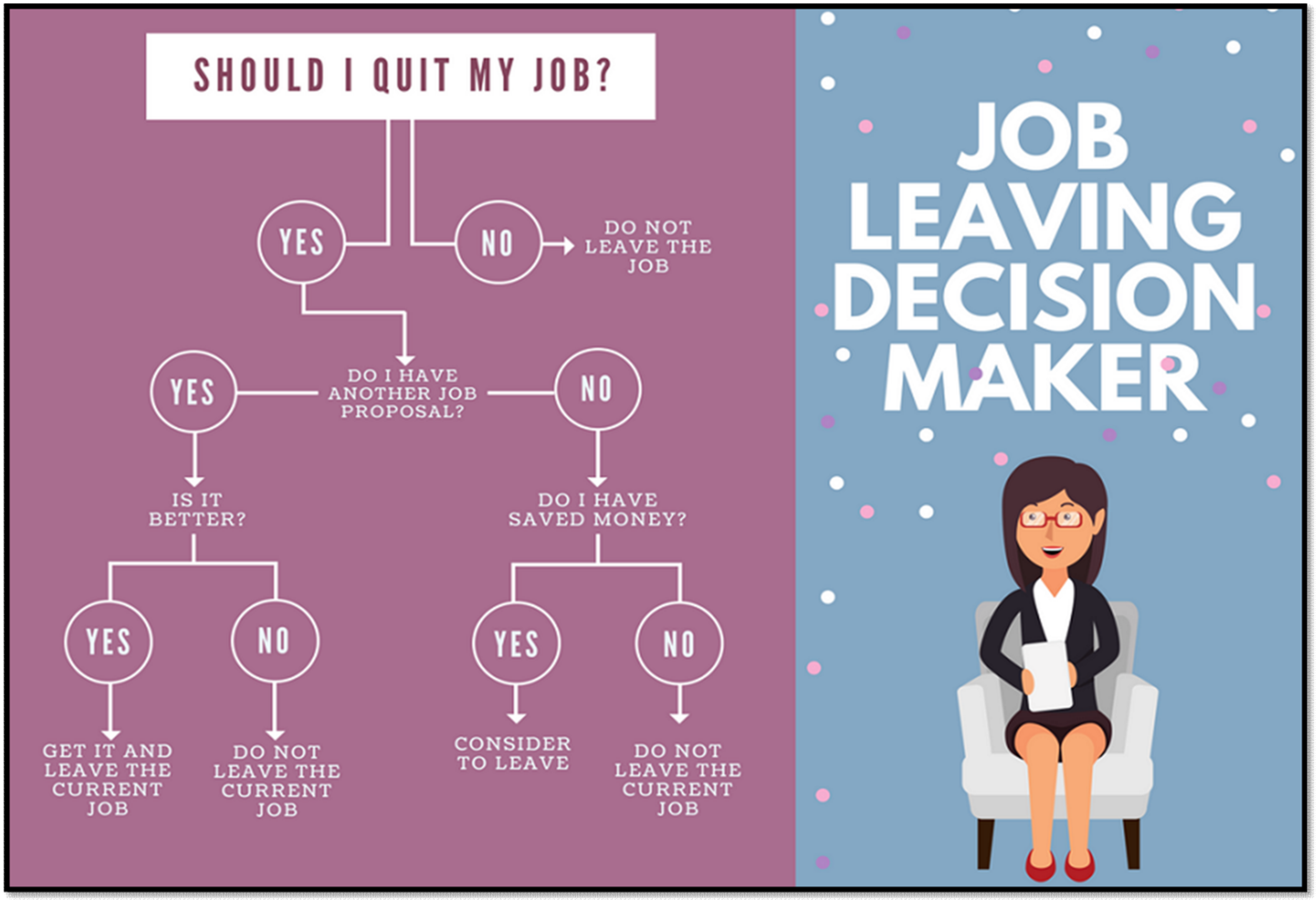
Decision Tree Learning

- 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.

- 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.


Decision Node
Decision Node
Decision Node
Sub-Tree
Leaf Node
Decision Node
Leaf Node
Leaf Node
Leaf Node
Leaf Node



SHOULD I QUIT MY JOB?

YES

NO

DO NOT LEAVE THE JOB

DO I HAVE ANOTHER JOB PROPOSAL?

YES

IS IT BETTER?

YES

GET IT AND LEAVE THE CURRENT JOB

NO

DO NOT LEAVE THE CURRENT JOB

NO

DO I HAVE SAVED MONEY?

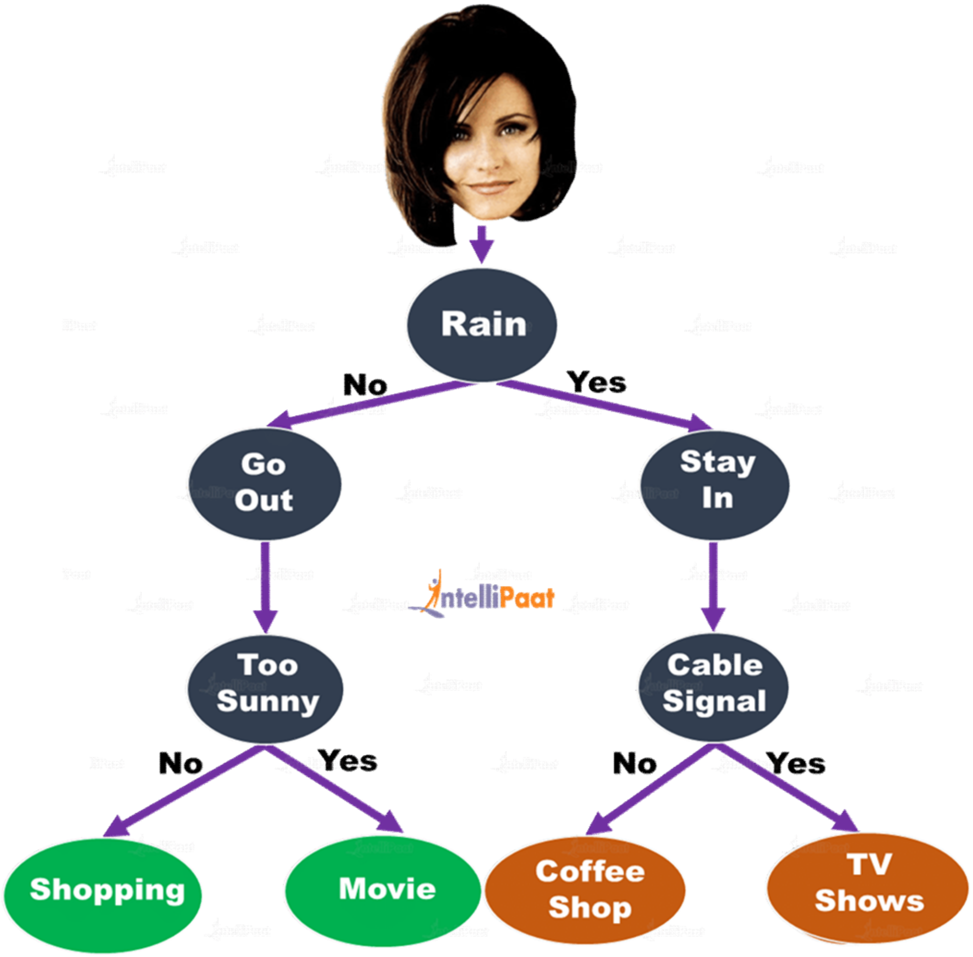
YES

CONSIDER TO LEAVE

NO

DO NOT LEAVE THE CURRENT JOB

JOB LEAVING DECISION MAKER



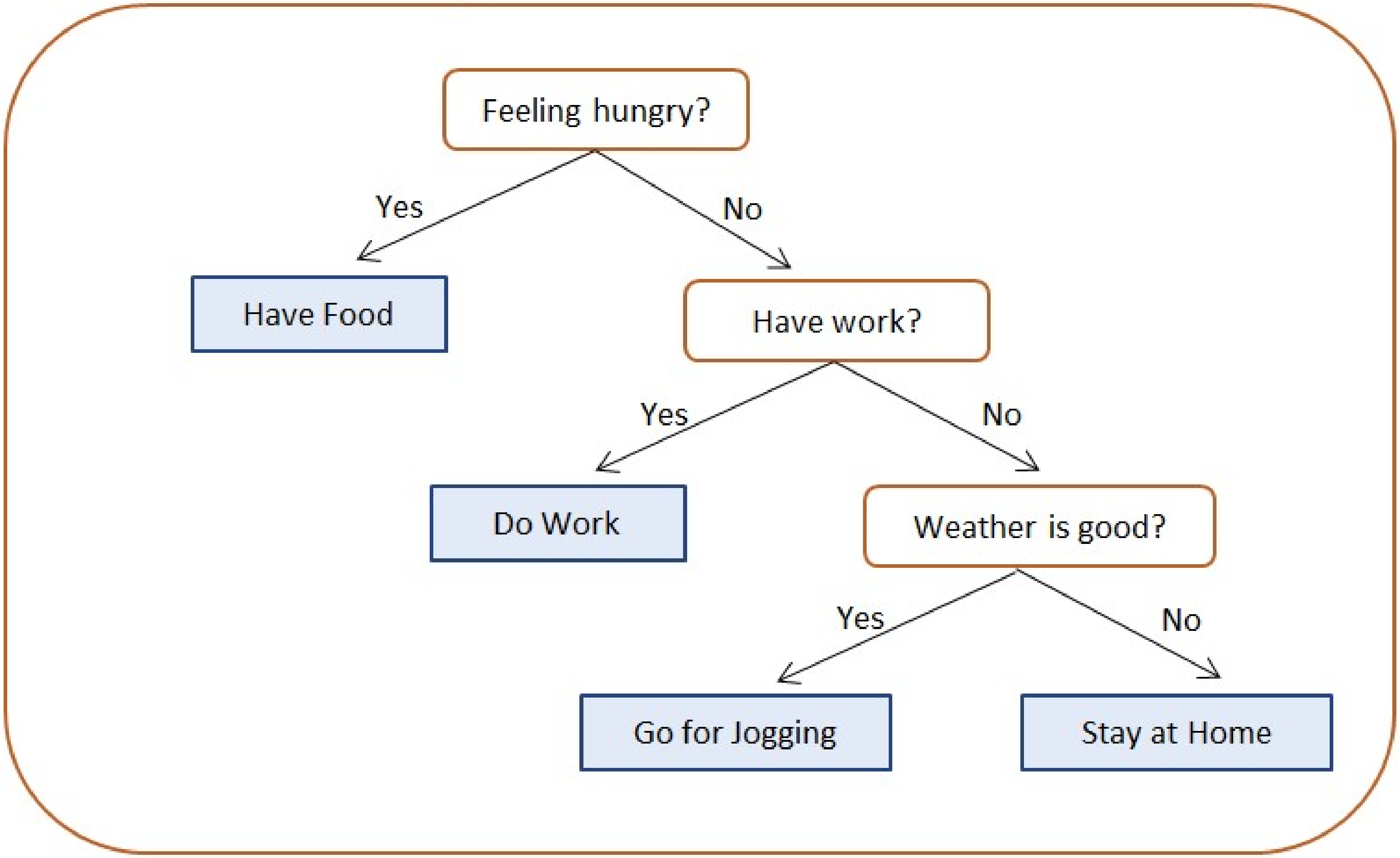
Monica’s Weekend Plan

Root Node

Interior Nodes

Leaf Nodes

Sample Decision Tree Choices



Feeling hungry?

Yes

No

Have Food

Have work?

Yes

No

Do Work

Weather is good?

Yes

No

Go for Jogging

Stay at Home

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.

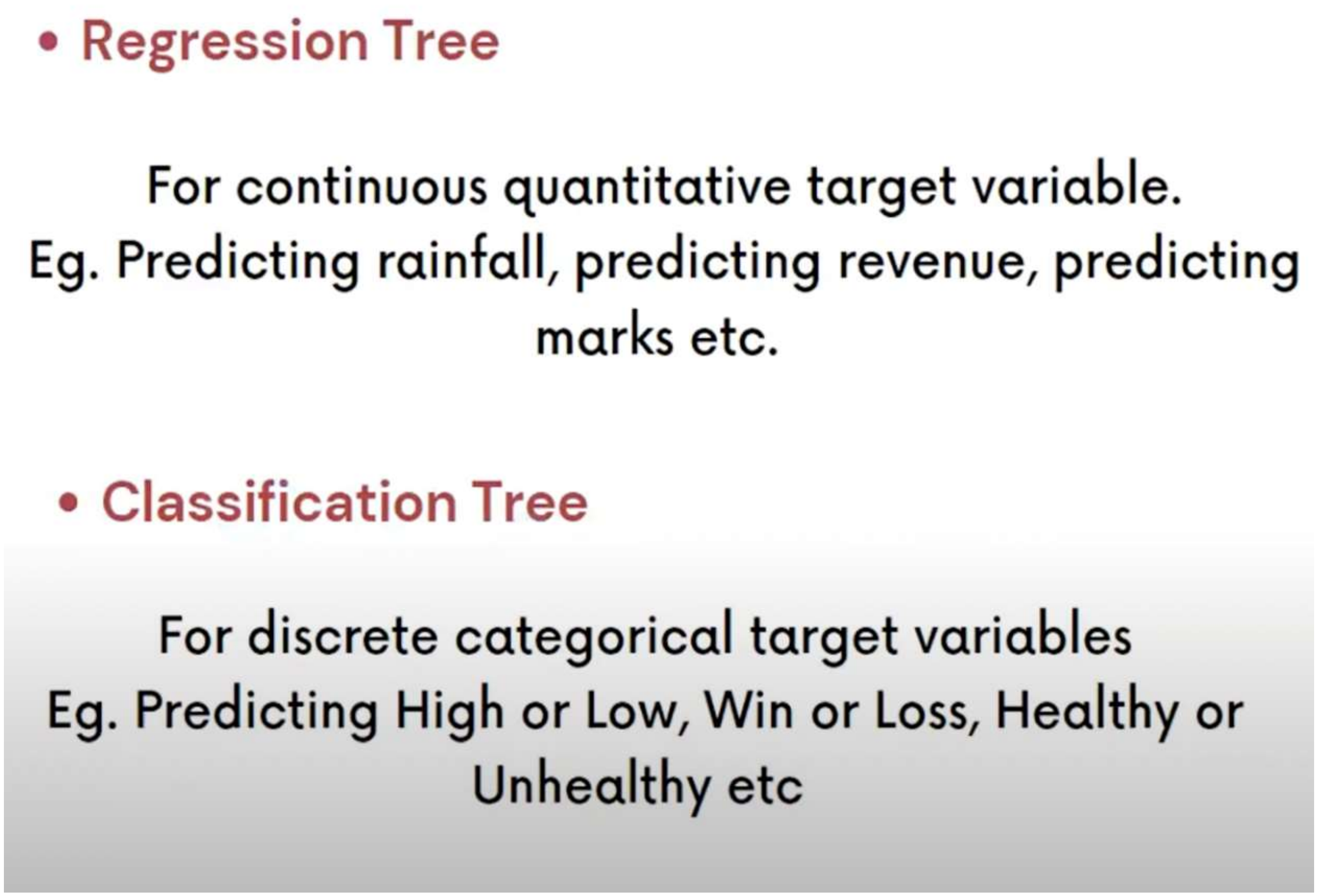
Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

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.



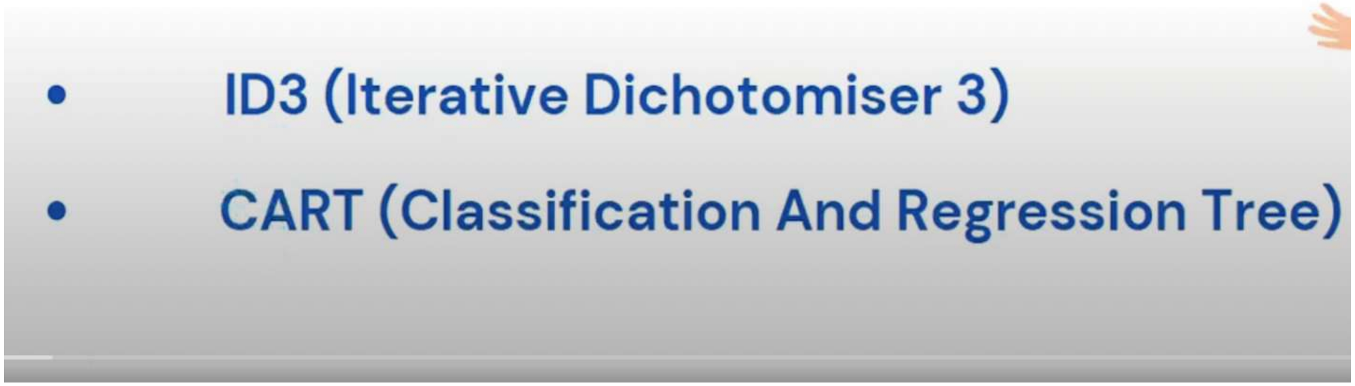
- **Regression Tree**

For continuous quantitative target variable.
Eg. Predicting rainfall, predicting revenue, predicting marks etc.

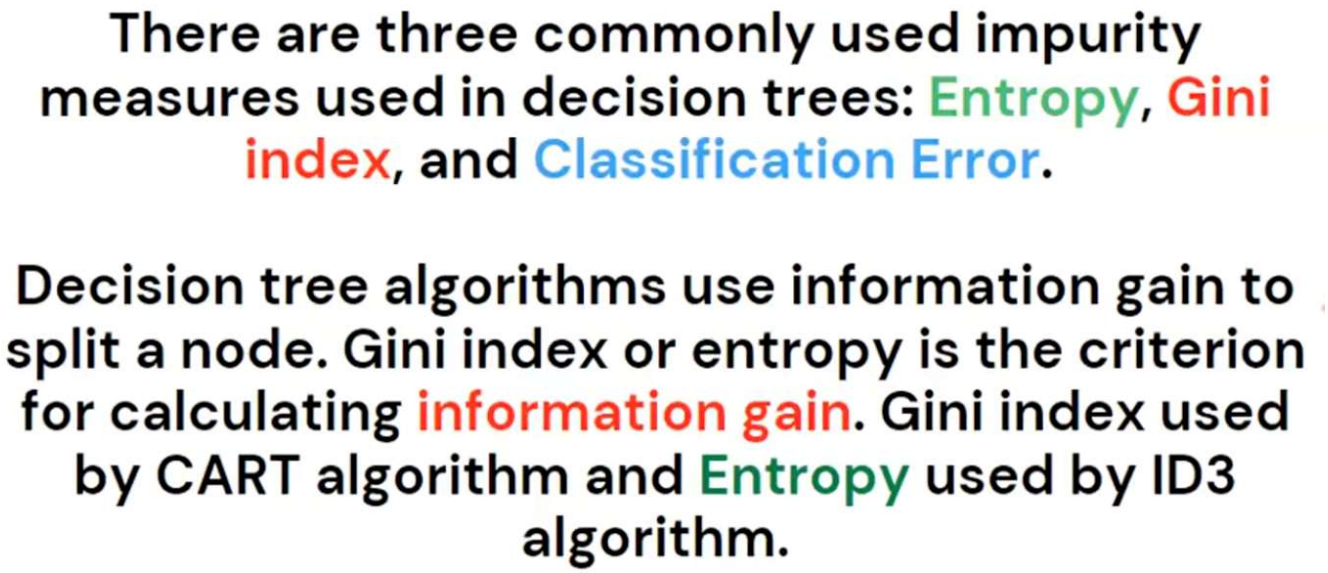
- **Classification Tree**

For discrete categorical target variables
Eg. Predicting High or Low, Win or Loss, Healthy or Unhealthy etc

How to build Classification Trees ?



- ID3 (Iterative Dichotomiser 3)
- CART (Classification And Regression Tree)



There are three commonly used impurity measures used in decision trees: Entropy, Gini index, and Classification Error.

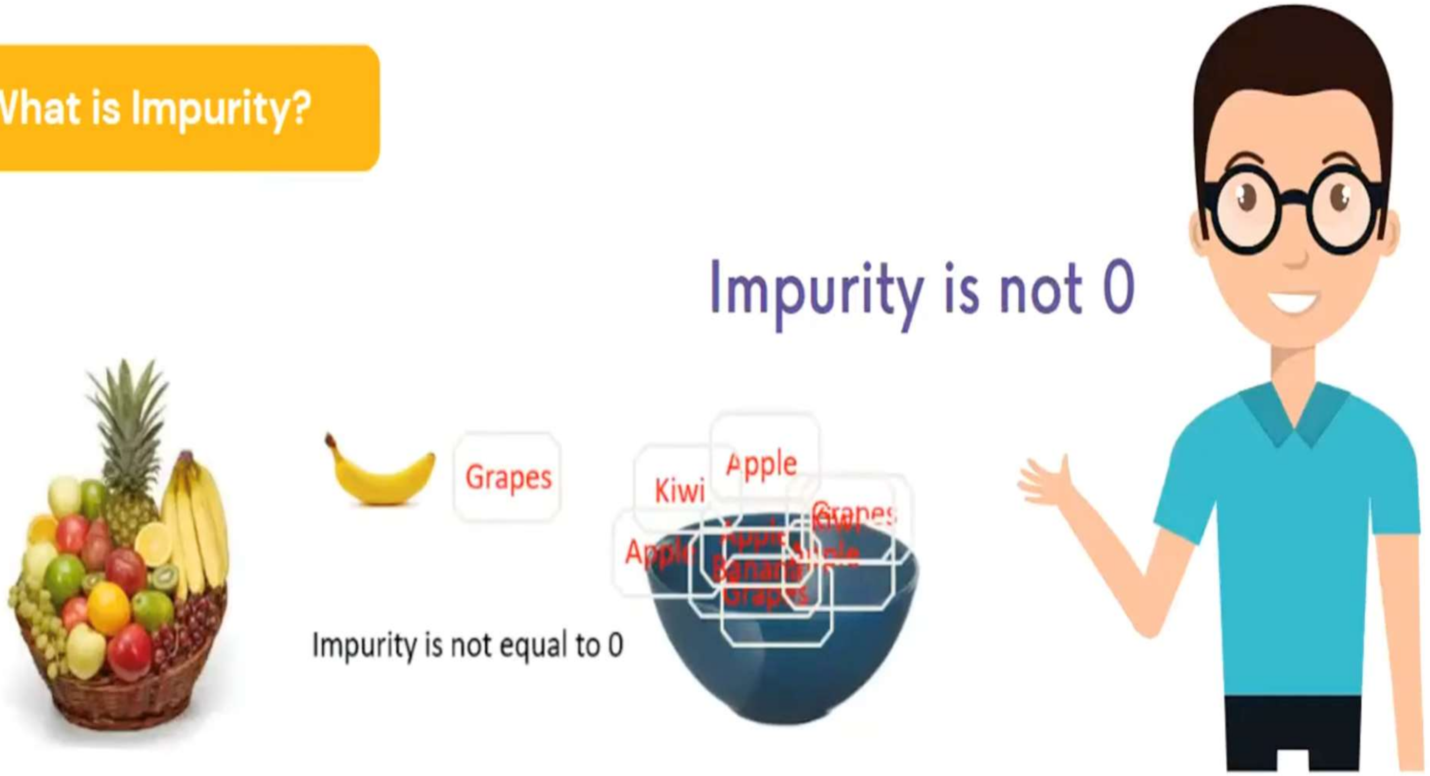
Decision tree algorithms use information gain to split a node. Gini index or entropy is the criterion for calculating information gain. Gini index used by CART algorithm and Entropy used by ID3 algorithm.

What is Impurity?

Impurity is 0

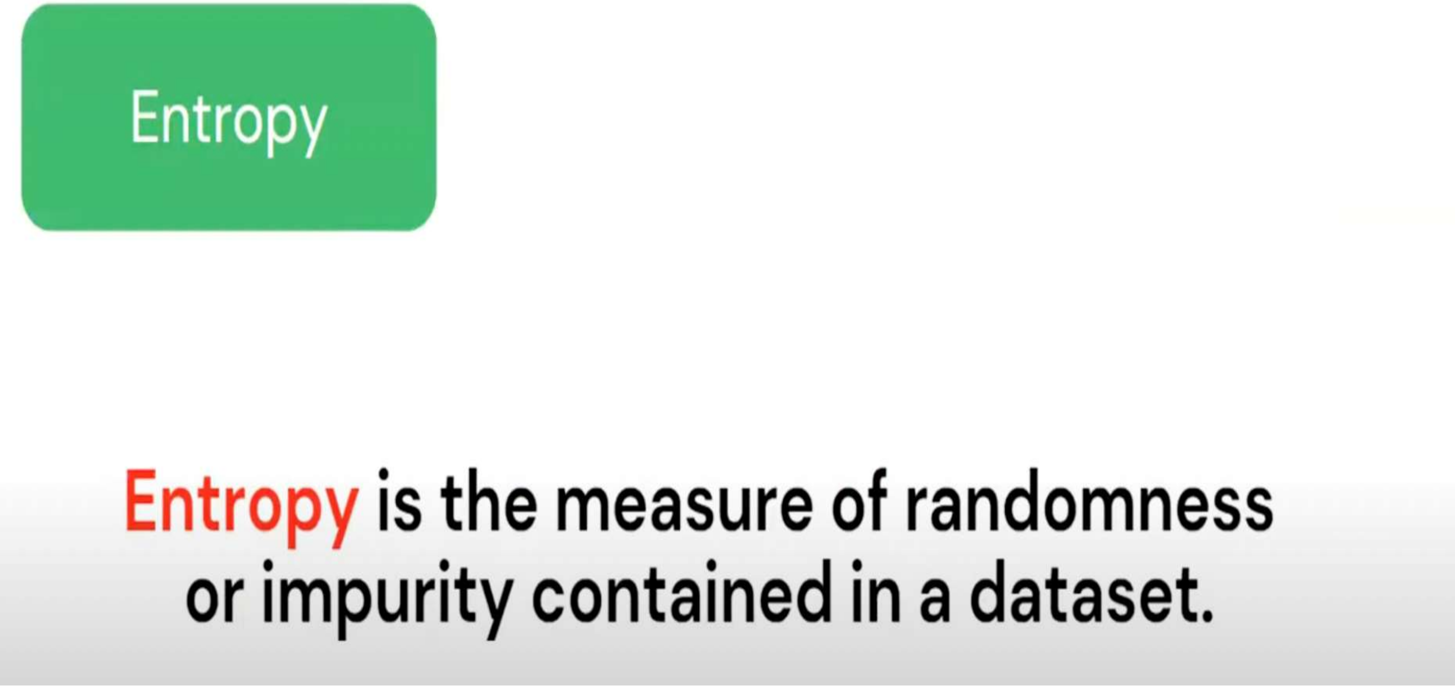
Apple
Apple
Apple
Apple
Apple
Impurity = 0

What is Impurity?



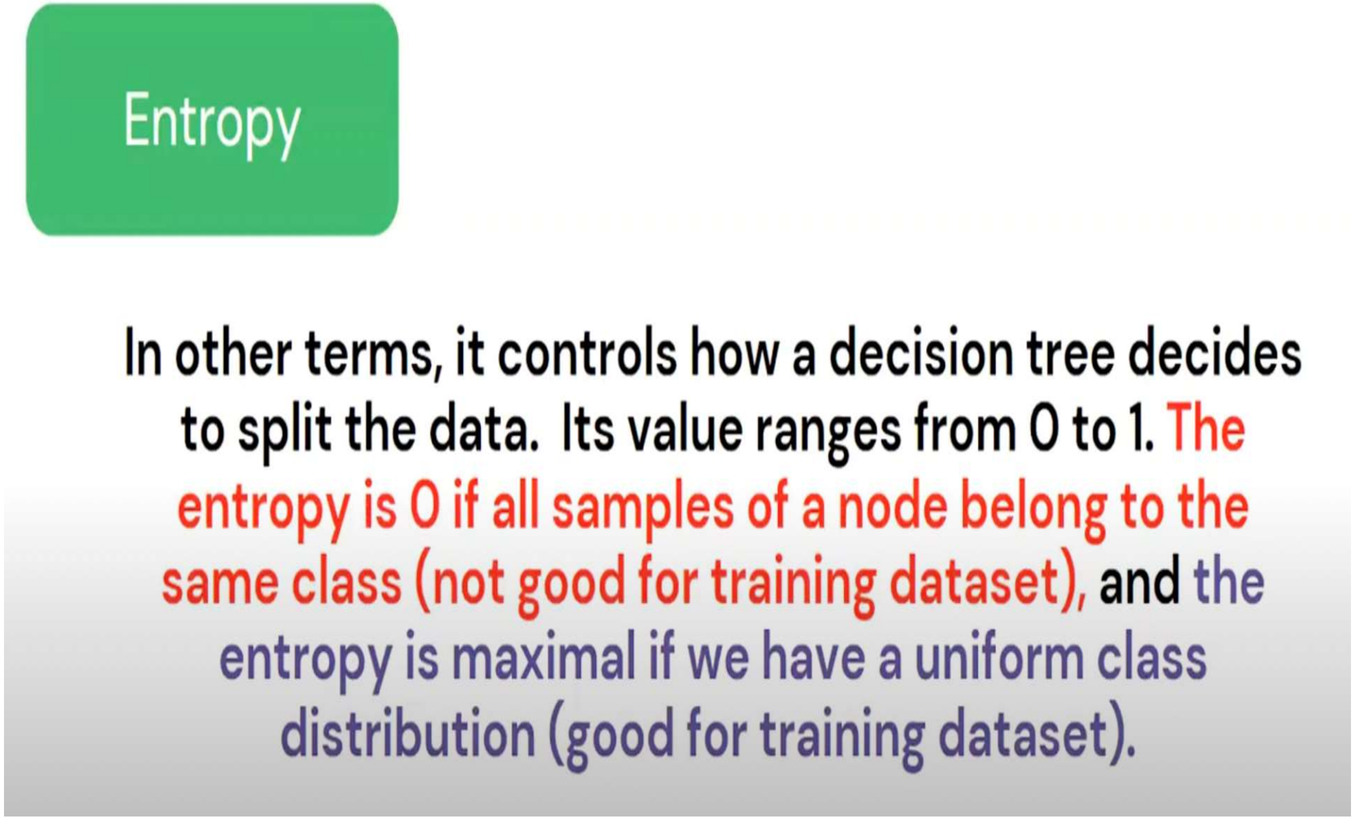
Impurity is not 0

Impurity is not equal to 0



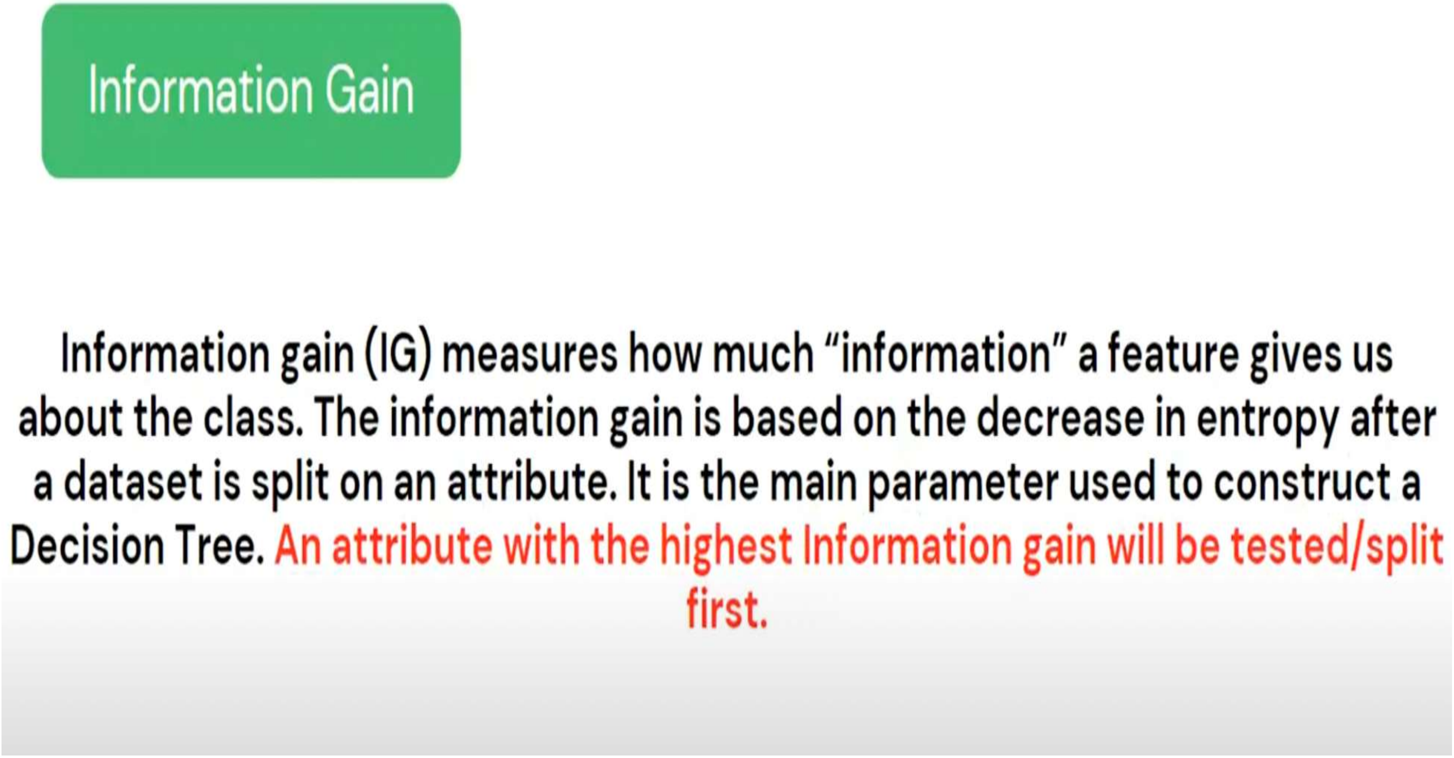
Entropy

Entropy is the measure of randomness or impurity contained in a dataset.



Entropy

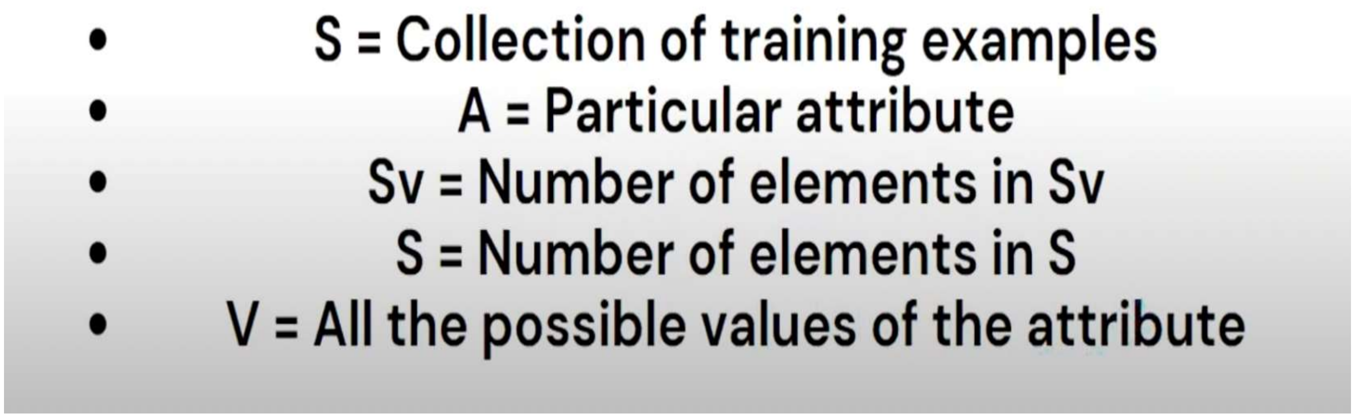
In other terms, it controls how a decision tree decides to split the data. Its value ranges from 0 to 1. The entropy is 0 if all samples of a node belong to the same class (not good for training dataset), and the entropy is maximal if we have a uniform class distribution (good for training dataset).



Information Gain

Information gain (IG) measures how much "information" a feature gives us about the class. The information gain is based on the decrease in entropy after a dataset is split on an attribute. It is the main parameter used to construct a Decision Tree. An attribute with the highest Information gain will be tested/split first.

$$Gain(S,A) = Entropy(S) - \sum_{v} \frac{|S_v|}{|S|} Entropy(S_v)$$



- S = Collection of training examples
- A = Particular attribute
- Sv = Number of elements in Sv
- S = Number of elements in S
- V = All the possible values of the attribute

HOW DOES IT WORKS?

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.**

Attribute Selection Measure:

It is a heuristic for **selecting the splitting criterion** that **“best” separates a given data partition, D**, of a **class-labeled training tuples into individual classes.**

The **Three Important attribute Selection measures** are

Information gain	**ID3/C4.5**
Gain Ratio	**C4.5**
Gini Index	**CART**

(ID3 Algorithm)

- **Entropy** is the main concept of this algorithm, which helps determine a **feature or attribute** that gives **maximum information about a class** is called **Information gain or ID3 algorithm.**
- By using this method, we can **reduce the level of entropy** from the **root node to the leaf node.**

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

- where **p_i** is the probability of randomly picking an element of class **i** (i.e. the proportion of the dataset made up of class i).

Steps in ID3 algorithm:

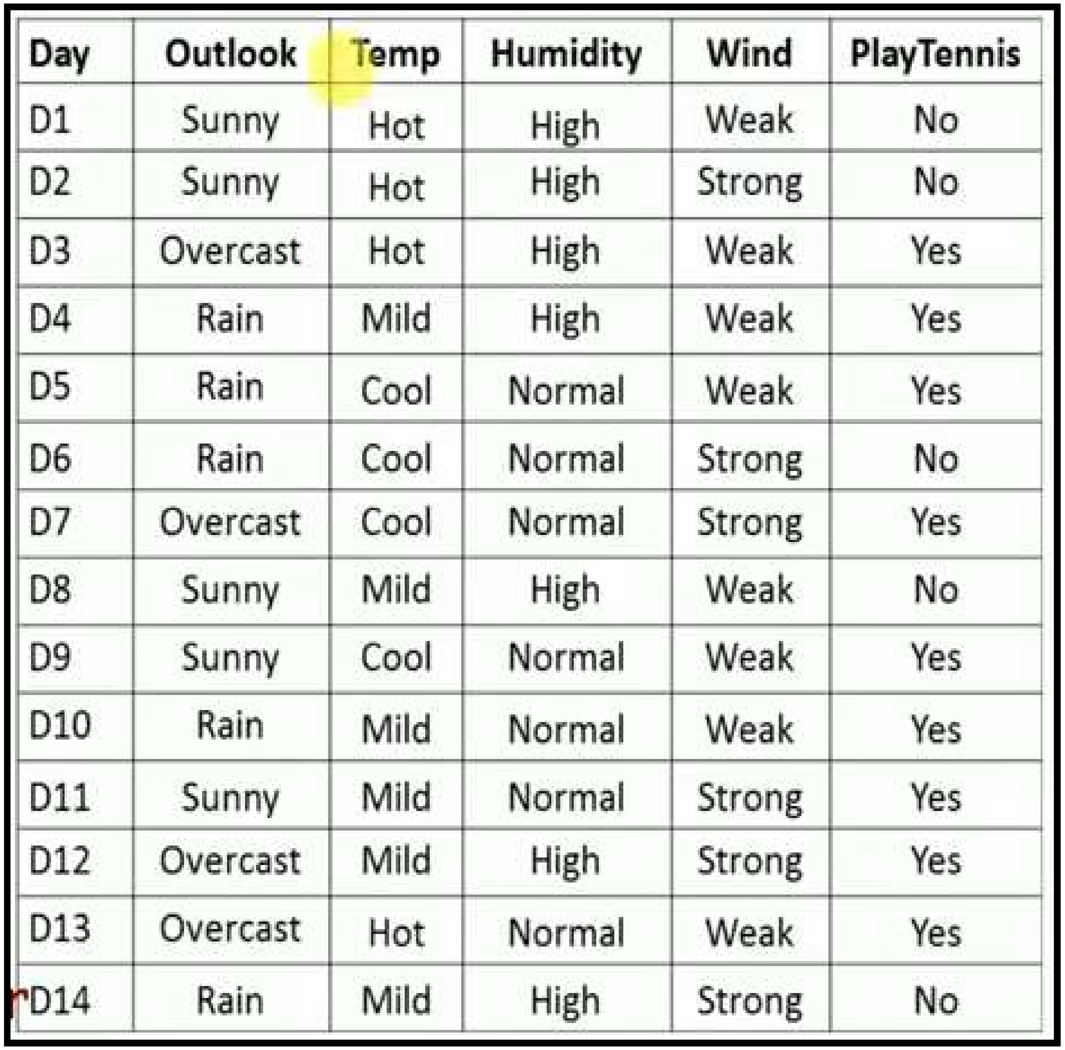
1. It **begins with the original set S as the root node.**

2. On **each iteration of the algorithm**, it iterates through the very unused attribute of the set S and **calculates Entropy(H) and Information gain(IG)** of this attribute.

3. It then selects the attribute which has the **smallest Entropy or Largest Information gain.**

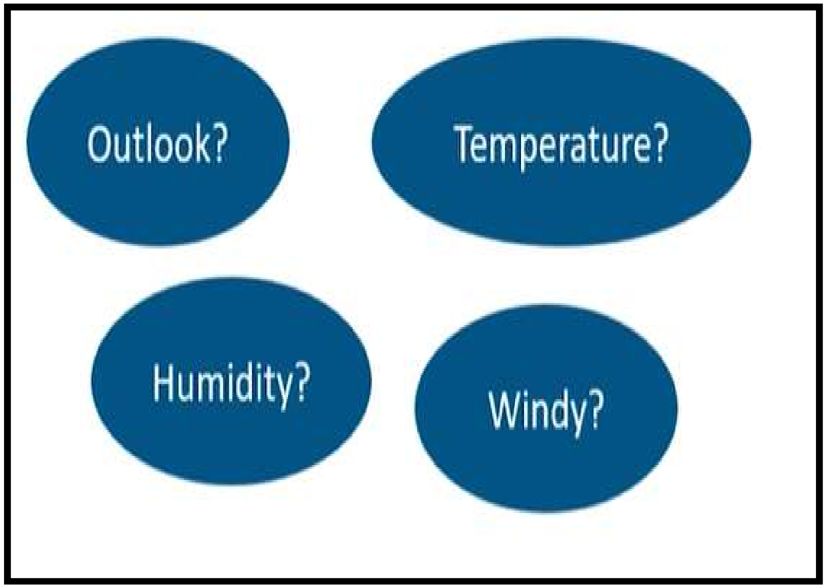
4. The **set S is then split by the selected attribute** to produce a subset of the data.

Example



Day	Outlook	Temp	Humidity	Wind	PlayTennis
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

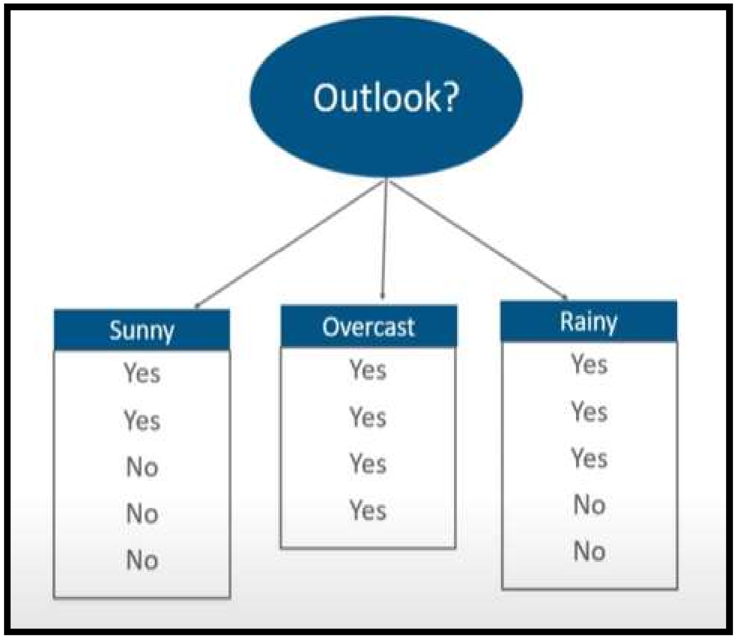
- In this Data Set contains 14 instances and 4 Attributes
- The 4 Attributes are



- If we want to draw ID3 Algorithm, first we must identify which attribute is having Maximum Information out of the available attributes.
- In this case we have 4 attributes, out of 4 attributes we need to calculate the Information Gain of every attribute
- Here , the attribute having the maximum information Gain will be the <u>Root Node.</u>
- After that we will start the Tree here.
- So, here First consider the First Attribute : the Attribute is <u>Outlook.</u>

- And the possible values of the outlook is

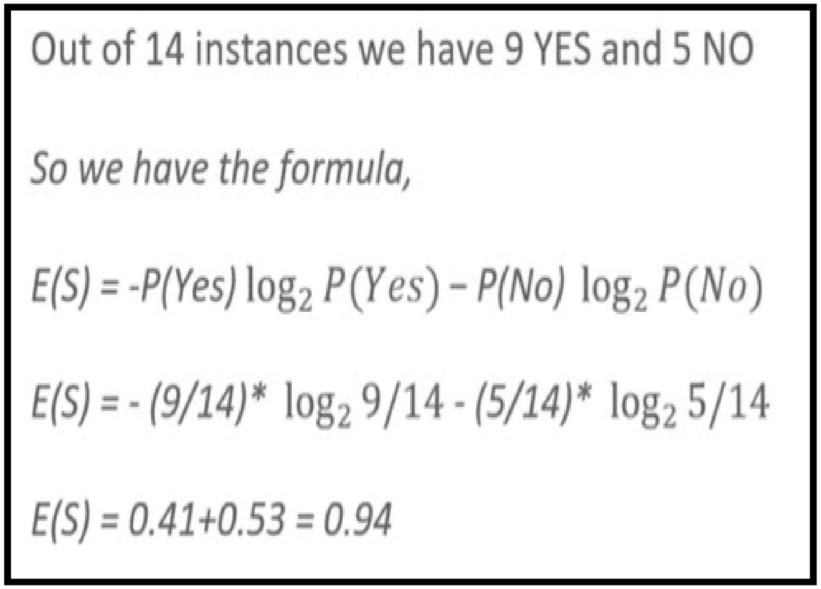
{Sunny, Overcast, Rain}



- If we want to calculate the Information Gain of Every attribute , first we will calculate the Entropy of the Whole Data set and the Entropy of each and every attribute.

Step-1: Compute the Entropy of Entire Dataset.
(We calculate for output attribute)

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



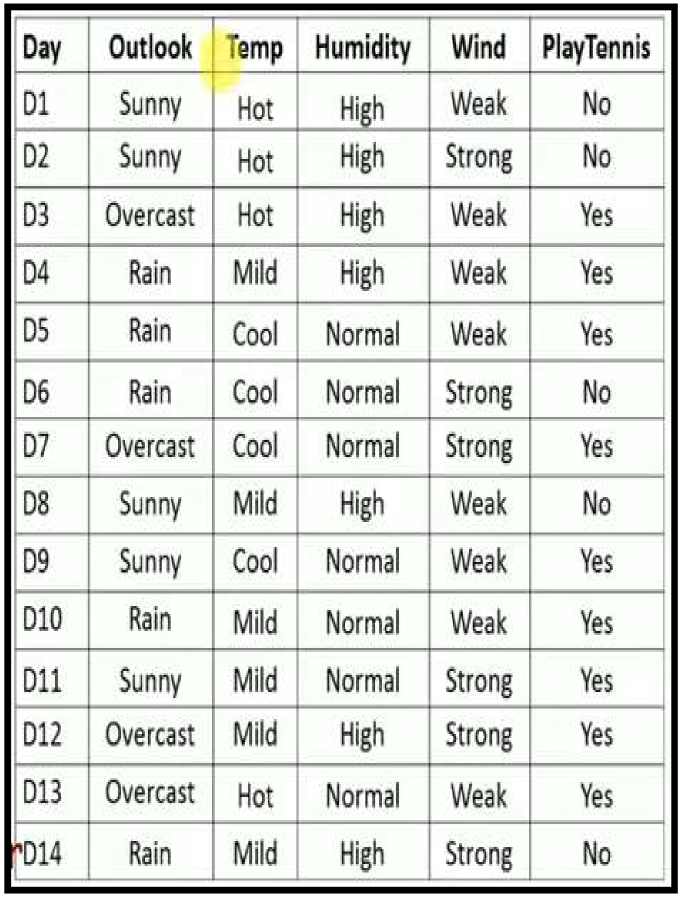
Out of 14 instances we have 9 YES and 5 NO

So we have the formula,

$E(S) = -P(Yes) \log_2 P(Yes) - P(No) \log_2 P(No)$

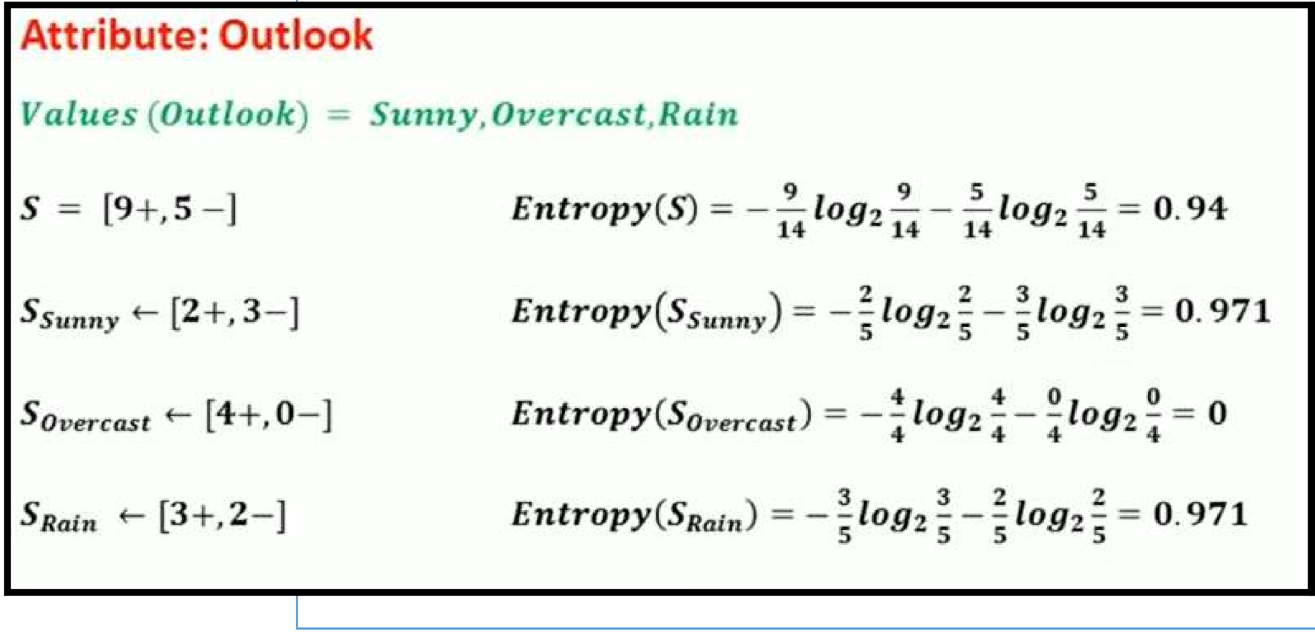
$E(S) = - (9/14)* \log_2 9/14 - (5/14)* \log_2 5/14$

$E(S) = 0.41+0.53 = 0.94$



Day	Outlook	Temp	Humidity	Wind	PlayTennis
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

- After that we need to calculate the Entropy of each and every attribute . So, the First Attribute is <u>Outlook</u>



Attribute: Outlook

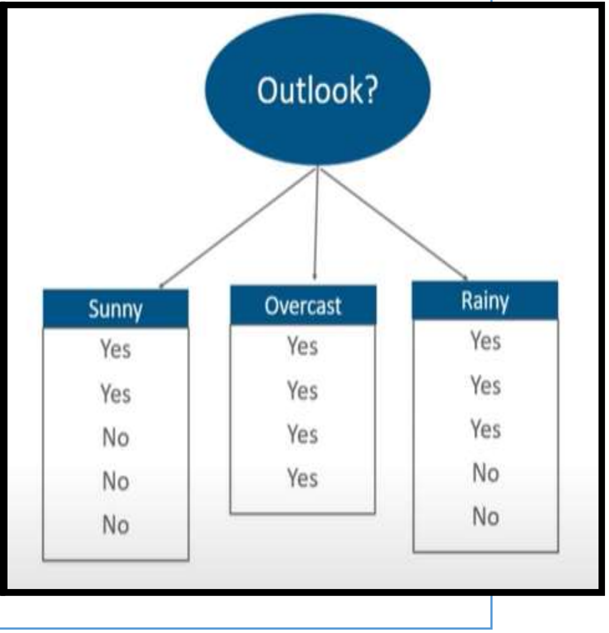
$Values\ (Outlook)\ =\ Sunny, Overcast, Rain$

$S\ =\ [9+, 5\ -]$ $\quad Entropy(S) = -\frac{9}{14}log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.94$

$S_{Sunny} \leftarrow [2+, 3-]$ $\quad Entropy(S_{Sunny}) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.971$

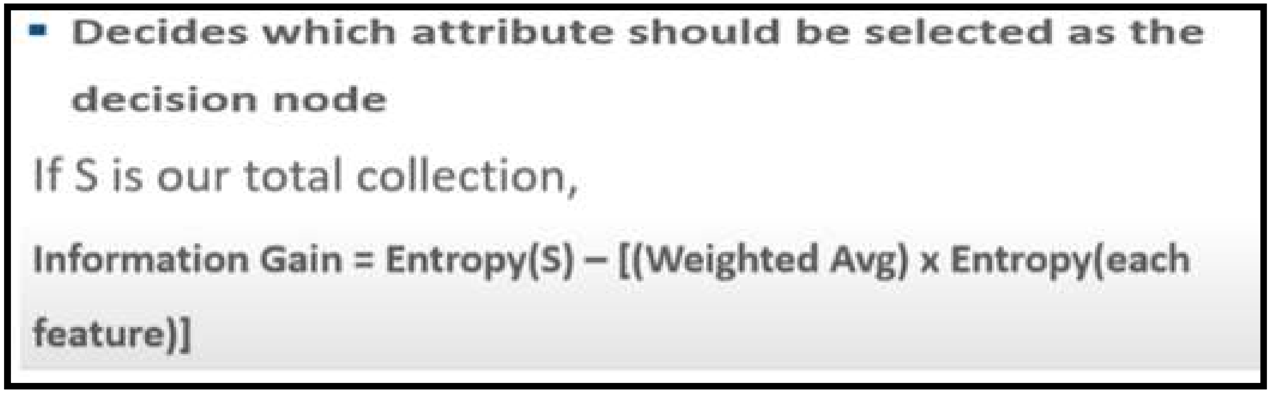
$S_{Overcast} \leftarrow [4+, 0-]$ $\quad Entropy(S_{Overcast}) = -\frac{4}{4}log_2\frac{4}{4} - \frac{0}{4}log_2\frac{0}{4} = 0$

$S_{Rain}\ \leftarrow [3+, 2-]$ $\quad Entropy(S_{Rain}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.971$



Sunny	Overcast	Rainy
Yes	Yes	Yes
Yes	Yes	Yes
No	Yes	Yes
No	Yes	No
No		No

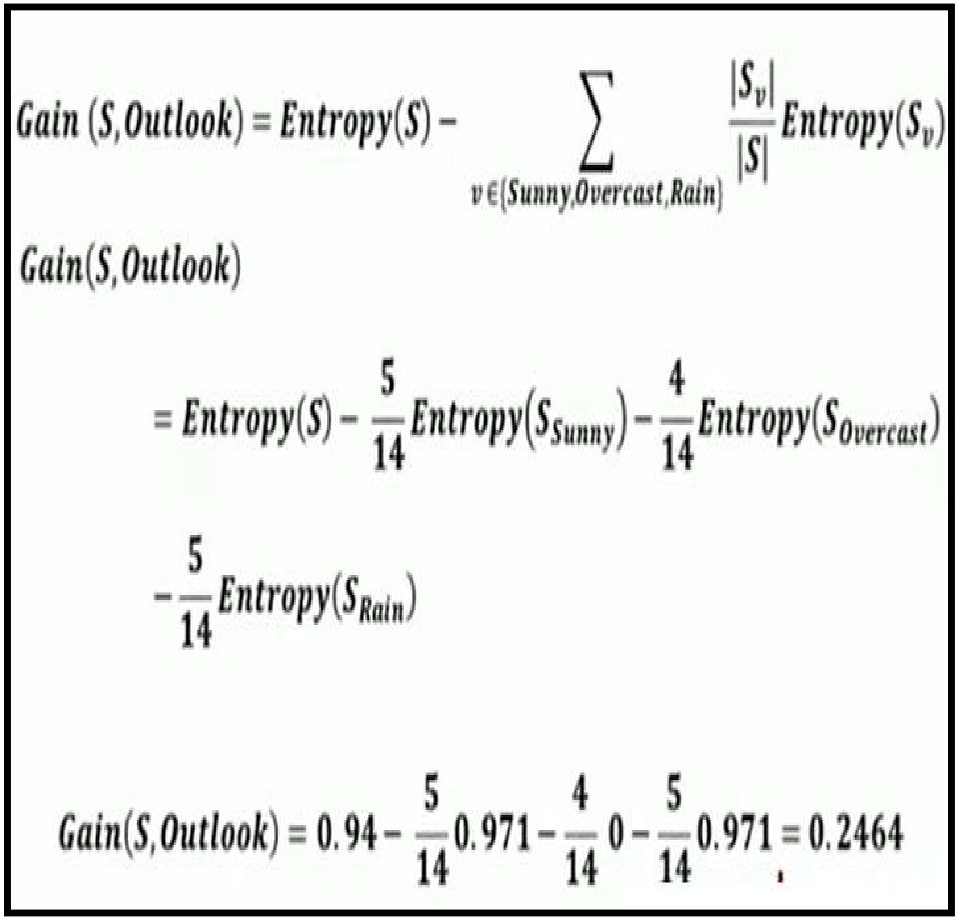
- Next, we need to calculate the Information Gain of each and every attribute.
- **Information Gain :**



- Decides which attribute should be selected as the decision node

If S is our total collection,

Information Gain = Entropy(S) – [(Weighted Avg) x Entropy(each feature)]



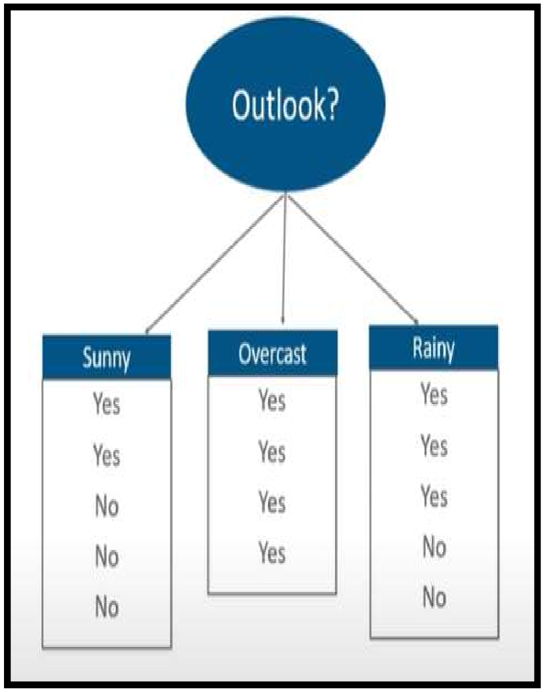
$$Gain\,(S,Outlook) = Entropy(S) - \sum_{v \in \{Sunny,Overcast,Rain\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S,Outlook)$$

$$= Entropy(S) - \frac{5}{14} Entropy(S_{Sunny}) - \frac{4}{14} Entropy(S_{Overcast})$$

$$- \frac{5}{14} Entropy(S_{Rain})$$

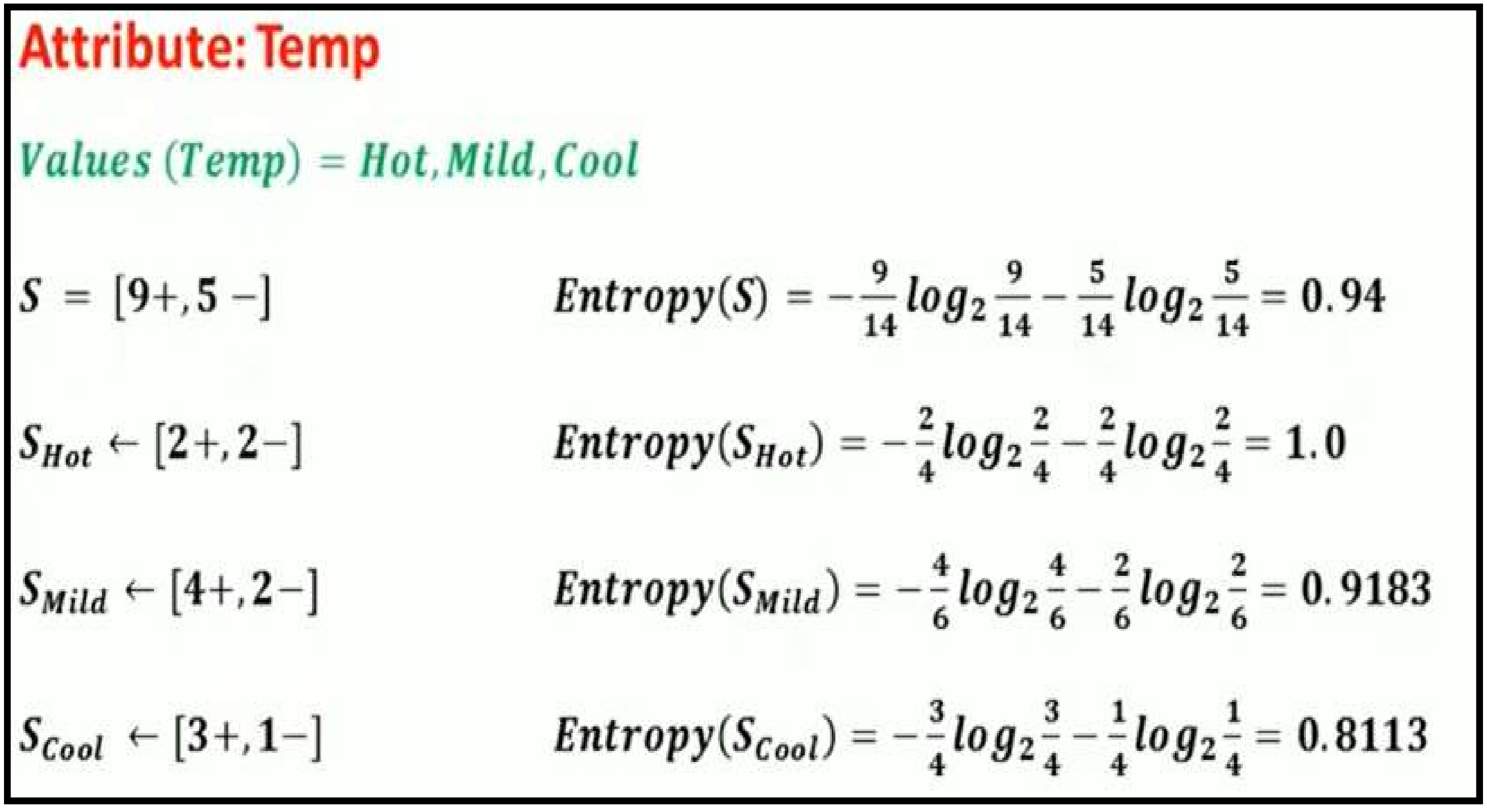
$$Gain(S,Outlook) = 0.94 - \frac{5}{14}0.971 - \frac{4}{14}0 - \frac{5}{14}0.971 = 0.2464$$



Outlook?

Sunny	Overcast	Rainy
Yes	Yes	Yes
Yes	Yes	Yes
No	Yes	Yes
No	Yes	No
No		No

- Similarly we need to calculate the remaining 3 attributes.
- Next attribute is **{Temperature}**



Attribute: Temp

$Values\ (Temp) = Hot, Mild, Cool$

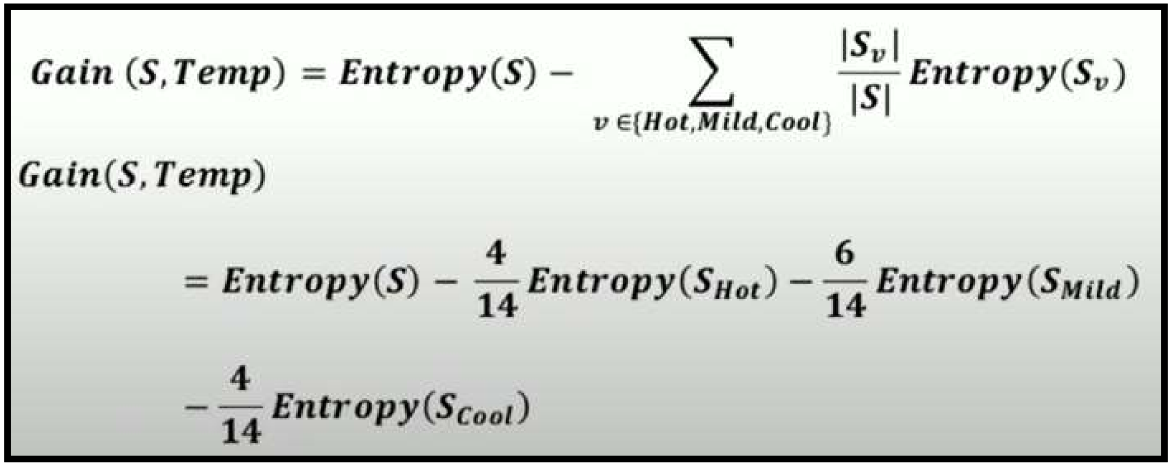
$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14}log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.94$

$S_{Hot} \leftarrow [2+, 2-]$ $\quad Entropy(S_{Hot}) = -\frac{2}{4}log_2\frac{2}{4} - \frac{2}{4}log_2\frac{2}{4} = 1.0$

$S_{Mild} \leftarrow [4+, 2-]$ $\quad Entropy(S_{Mild}) = -\frac{4}{6}log_2\frac{4}{6} - \frac{2}{6}log_2\frac{2}{6} = 0.9183$

$S_{Cool} \leftarrow [3+, 1-]$ $\quad Entropy(S_{Cool}) = -\frac{3}{4}log_2\frac{3}{4} - \frac{1}{4}log_2\frac{1}{4} = 0.8113$

- Next we calculate the **Information Gain** of **Temperature.**



$$Gain\ (S, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Temp)$$

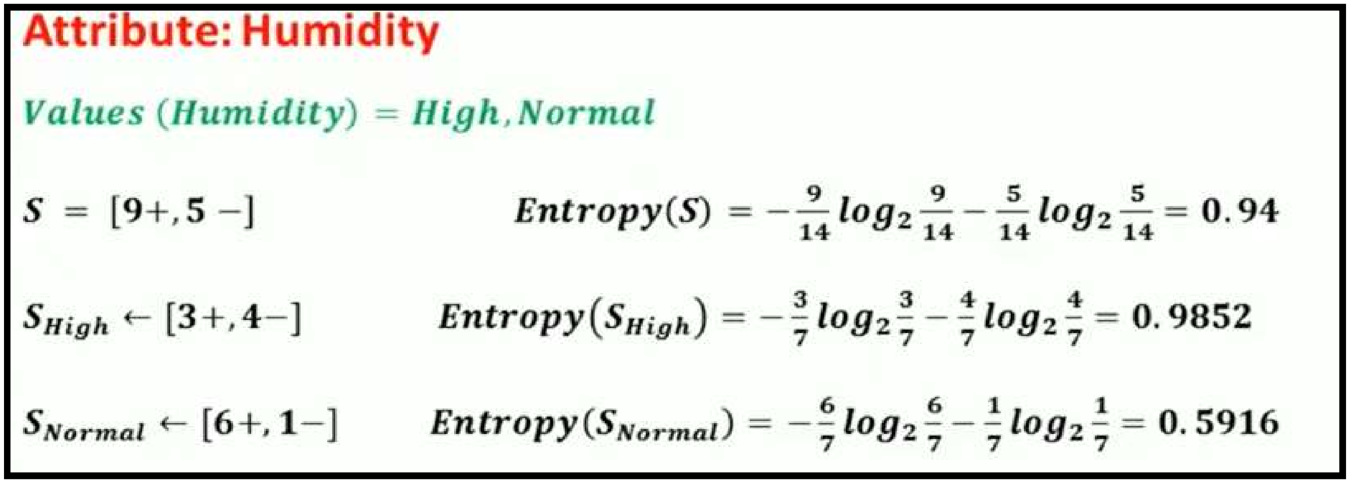
$$= Entropy(S) - \frac{4}{14} Entropy(S_{Hot}) - \frac{6}{14} Entropy(S_{Mild})$$

$$- \frac{4}{14} Entropy(S_{Cool})$$

Gain(S, Temp) = 0.94 – 4/14(1.0) – 6/14(0.9183) – 4/14(0.8113)

= 0.0289

- Next attribute is **{Humidity}**



Attribute: Humidity

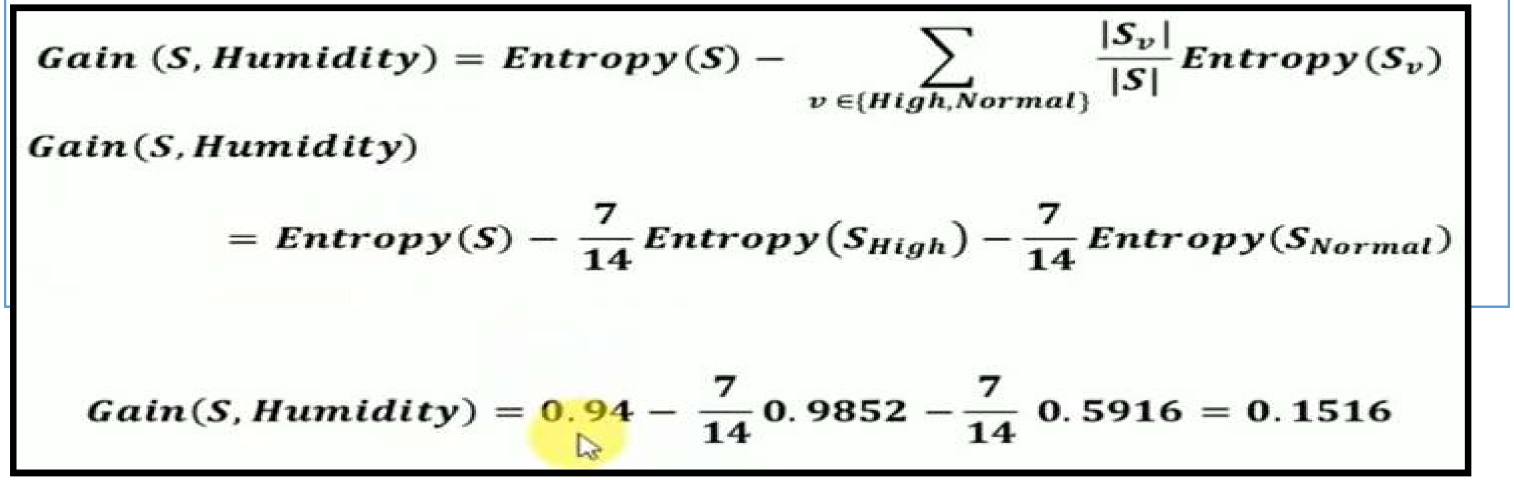
$Values\ (Humidity) = High, Normal$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$

$S_{High} \leftarrow [3+, 4-]$ $\quad Entropy(S_{High}) = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7} = 0.9852$

$S_{Normal} \leftarrow [6+, 1-]$ $\quad Entropy(S_{Normal}) = -\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7} = 0.5916$

Next, Calculate the Information Gain of Humidity



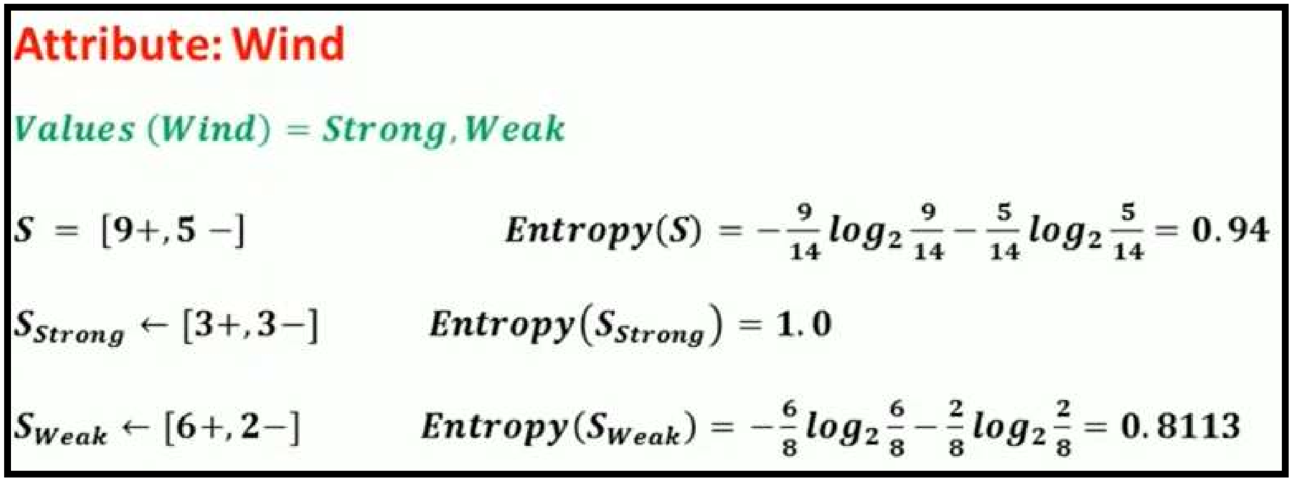
$$Gain\ (S, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Humidity)$$

$$= Entropy(S) - \frac{7}{14} Entropy(S_{High}) - \frac{7}{14} Entropy(S_{Normal})$$

$$Gain(S, Humidity) = 0.94 - \frac{7}{14} 0.9852 - \frac{7}{14} 0.5916 = 0.1516$$

- Next attribute is **{Wind}**



Attribute: Wind

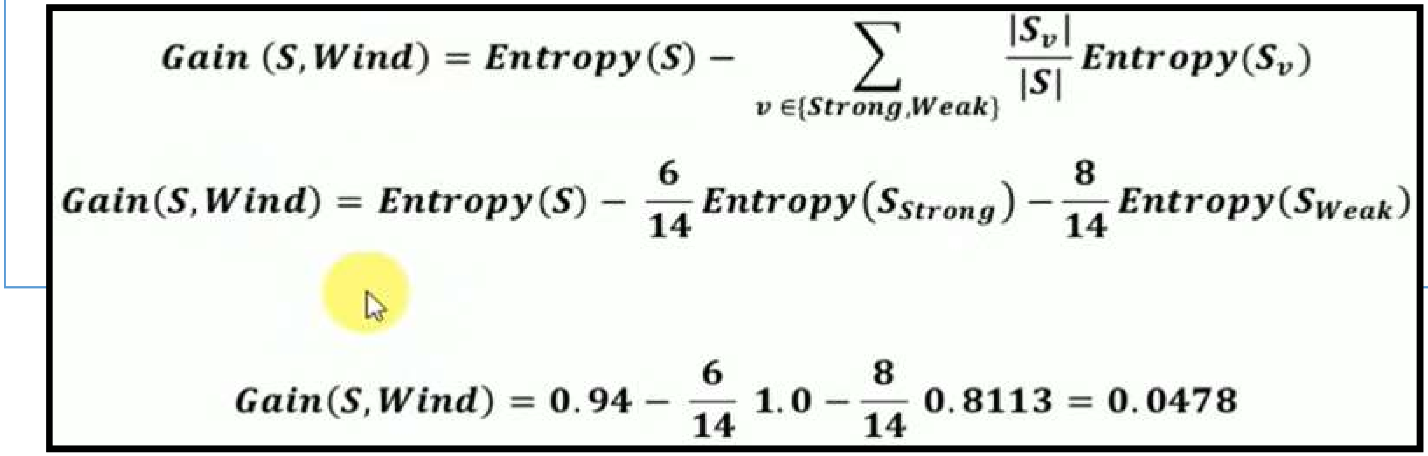
$Values\ (Wind) = Strong, Weak$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14}log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.94$

$S_{Strong} \leftarrow [3+, 3-]$ $\quad Entropy(S_{Strong}) = 1.0$

$S_{Weak} \leftarrow [6+, 2-]$ $\quad Entropy(S_{Weak}) = -\frac{6}{8}log_2\frac{6}{8} - \frac{2}{8}log_2\frac{2}{8} = 0.8113$

Next, Calculate the Information Gain of Wind

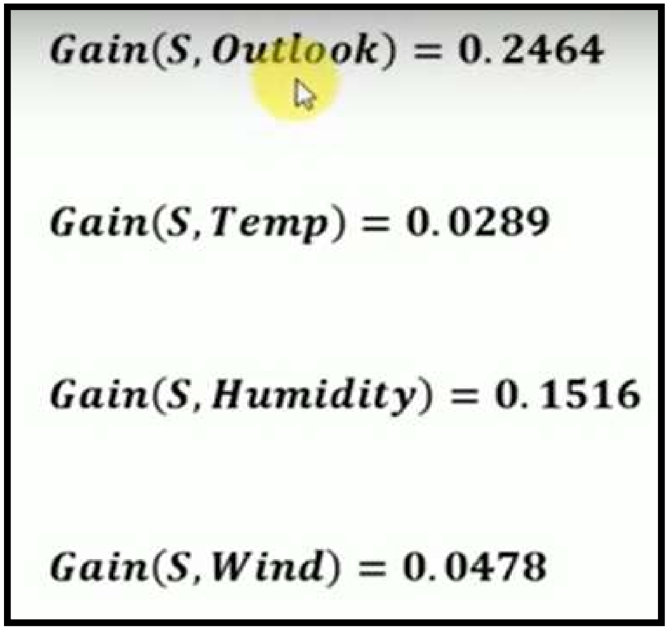


$$Gain\ (S, Wind) = Entropy(S) - \sum_{v\, \in\{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Wind) = Entropy(S) - \frac{6}{14} Entropy(S_{Strong}) - \frac{8}{14} Entropy(S_{Weak})$$

$$Gain(S, Wind) = 0.94 - \frac{6}{14}\, 1.0 - \frac{8}{14}\, 0.8113 = 0.0478$$

- So, we calculated all the Attributes. Next we need to check the **Maximum Information Gain** of **these attributes.**



$Gain(S, Outlook) = 0.2464$

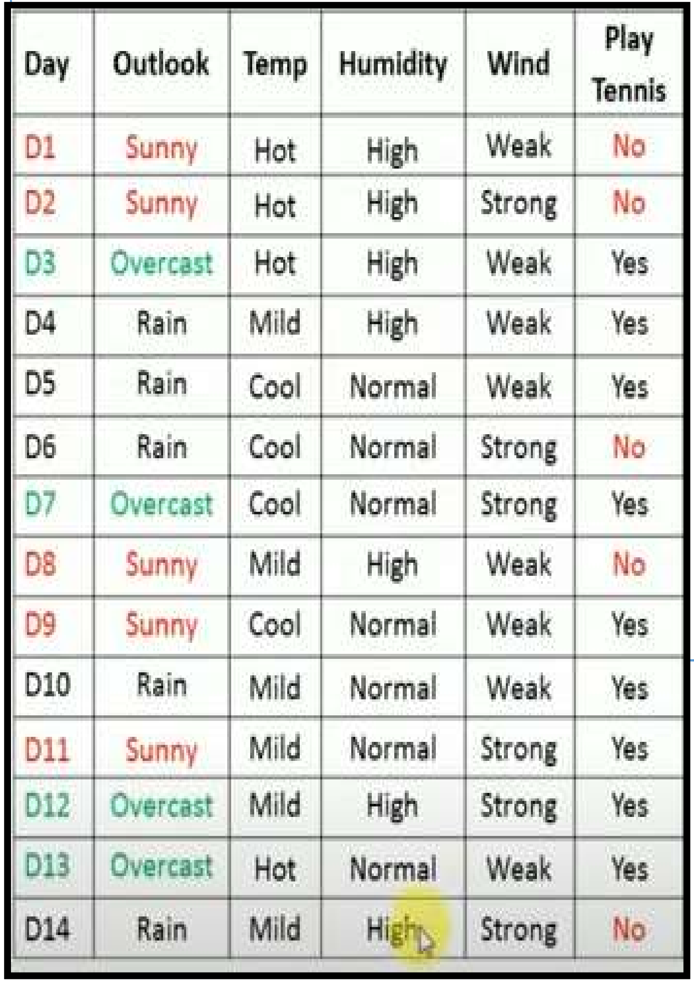
$Gain(S, Temp) = 0.0289$

$Gain(S, Humidity) = 0.1516$

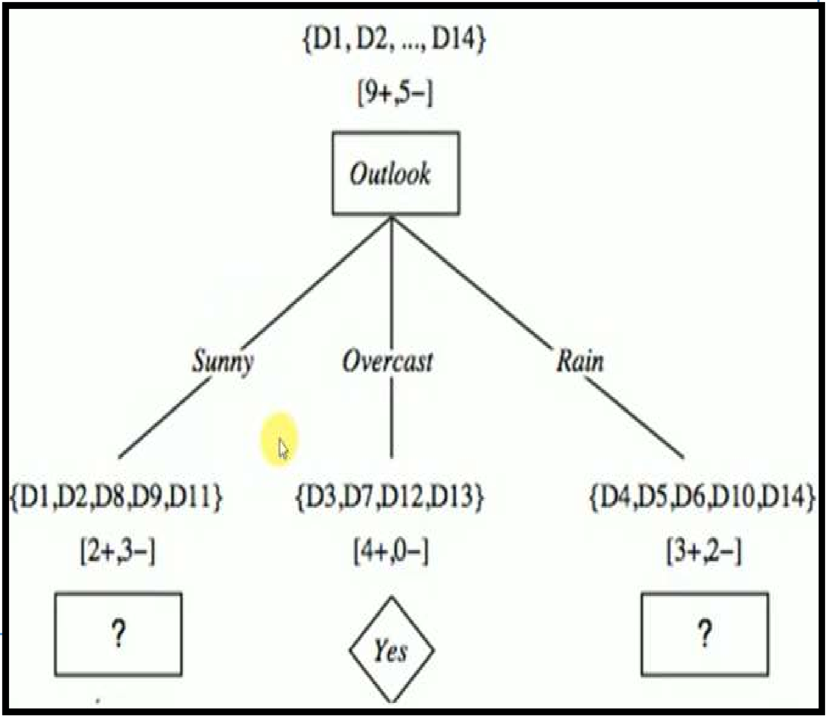
$Gain(S, Wind) = 0.0478$

- **SO, here OUTLOOK is having Maximum Information Gain . So that OUTLOOK is the <u>Root Node.</u>**

➢ So, that the Root Node is **OUTLOOK.**

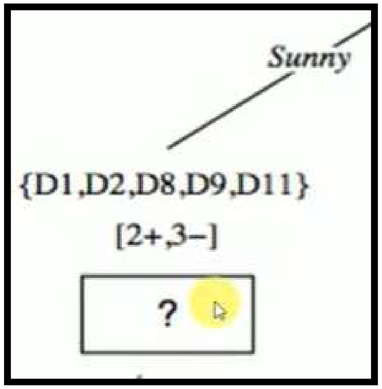


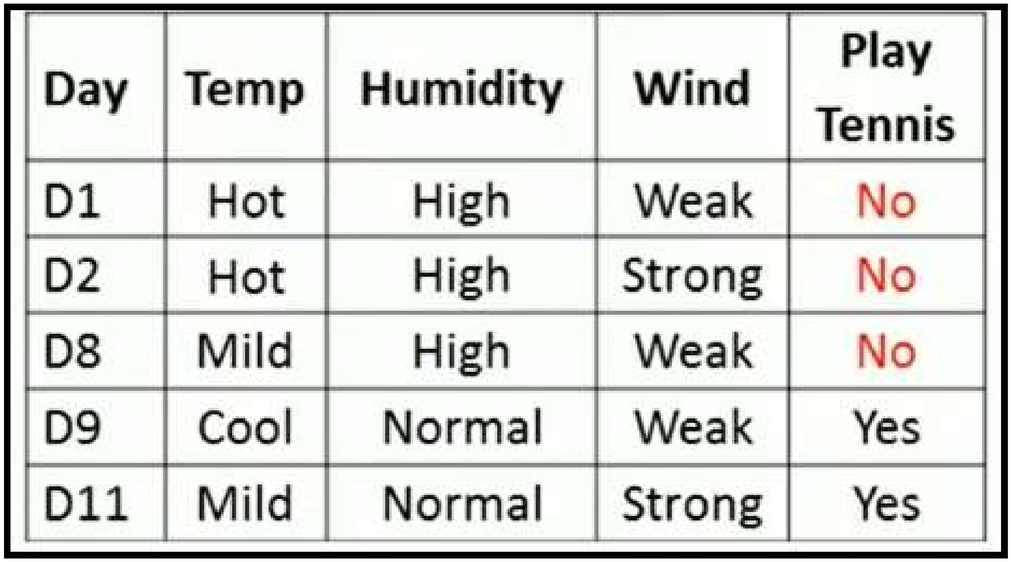
Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	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



- Next, we need to take the Left Hand Side " **SUNNY"** attributes to calculate the **Internal Nodes**.
- **So, here we can take only**

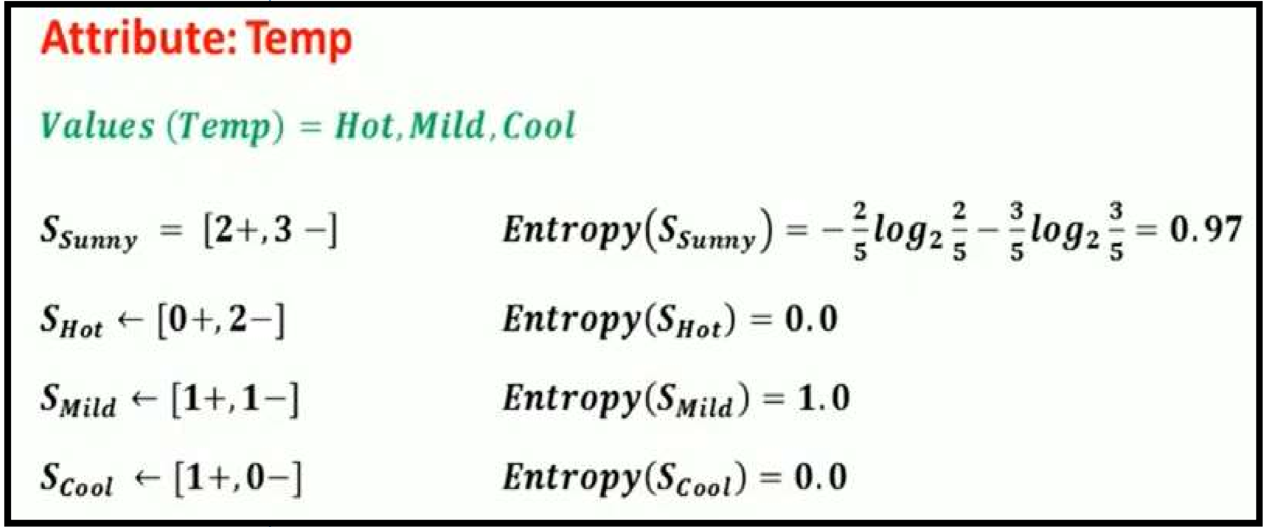
  { D1, D2, D8, D9, D11}





Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

- Again the **same procedure**, First We can calculate the **Entropy of each attribute**. So, here the **First Attribute** is **“ TEMP”**



Attribute: Temp

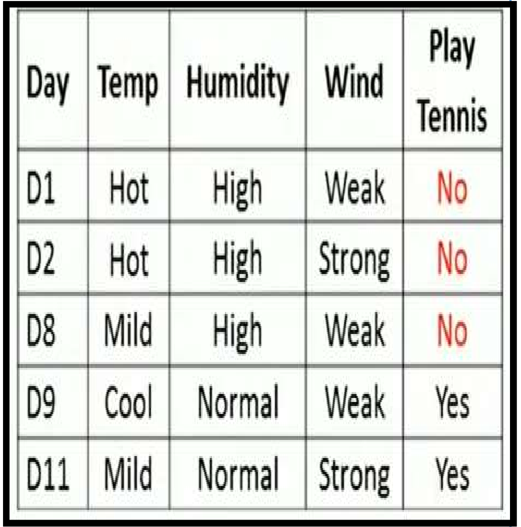
$Values\ (Temp) = Hot, Mild, Cool$

$S_{Sunny} = [2+, 3-]$ $Entropy(S_{Sunny}) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$

$S_{Hot} \leftarrow [0+, 2-]$ $Entropy(S_{Hot}) = 0.0$

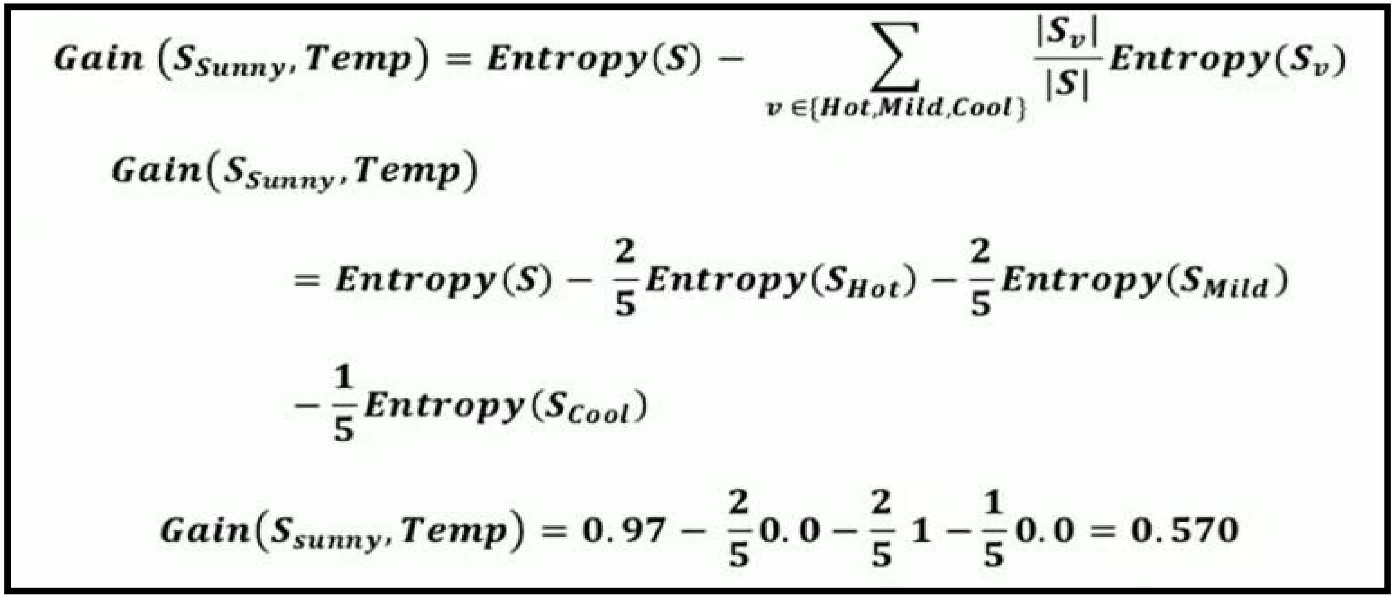
$S_{Mild} \leftarrow [1+, 1-]$ $Entropy(S_{Mild}) = 1.0$

$S_{Cool} \leftarrow [1+, 0-]$ $Entropy(S_{Cool}) = 0.0$



Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

- Next we can calculate the **Information Gain** of **Temp**



$$Gain\ (S_{Sunny}, Temp) = Entropy(S) - \sum_{v\ \in\{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

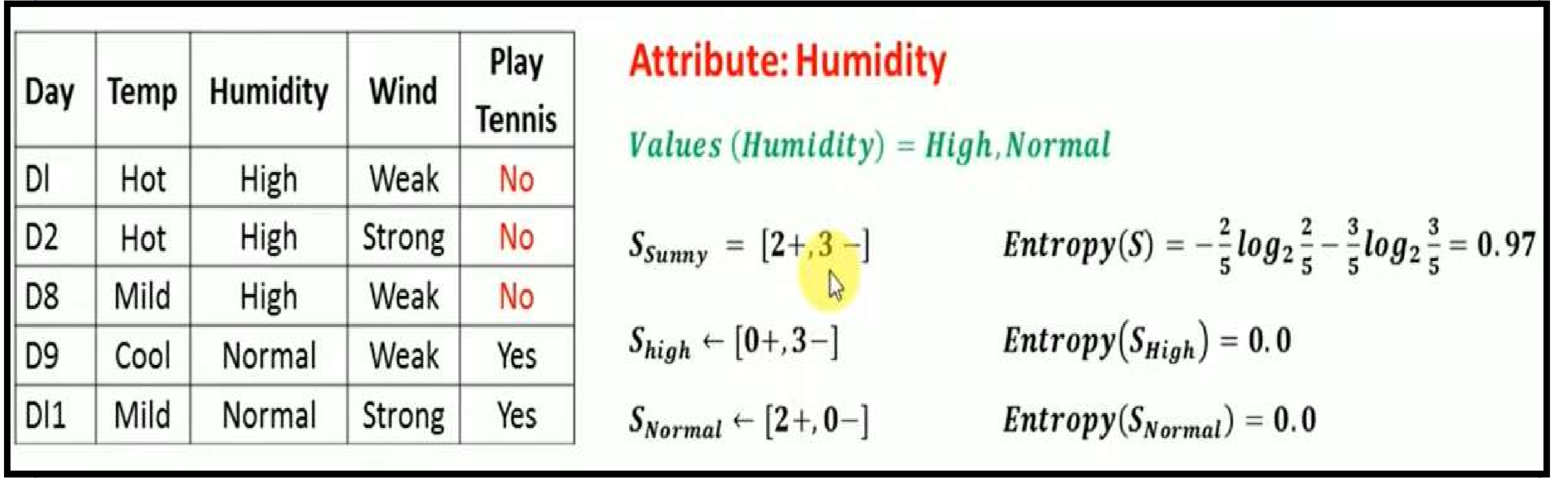
$$Gain(S_{Sunny}, Temp)$$

$$= Entropy(S) - \frac{2}{5} Entropy(S_{Hot}) - \frac{2}{5} Entropy(S_{Mild})$$

$$- \frac{1}{5} Entropy(S_{Cool})$$

$$Gain(S_{sunny}, Temp) = 0.97 - \frac{2}{5} 0.0 - \frac{2}{5}\ 1 - \frac{1}{5} 0.0 = 0.570$$

- Next attribute is **{Humidity}**



Day	Temp	Humidity	Wind	Play Tennis
DI	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
DI1	Mild	Normal	Strong	Yes

Attribute: Humidity

$Values\ (Humidity) = High, Normal$

$S_{Sunny} = [2+, 3-]$

$Entropy(S) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$

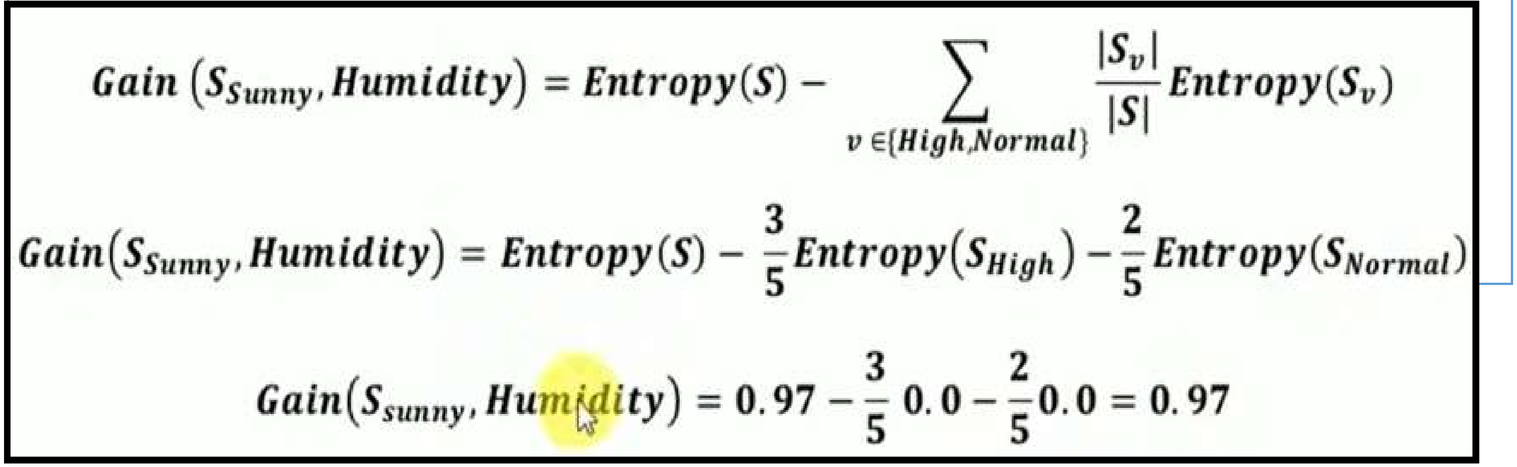
$S_{high} \leftarrow [0+, 3-]$

$Entropy(S_{High}) = 0.0$

$S_{Normal} \leftarrow [2+, 0-]$

$Entropy(S_{Normal}) = 0.0$

Next, Calculate the Information Gain of Humidity

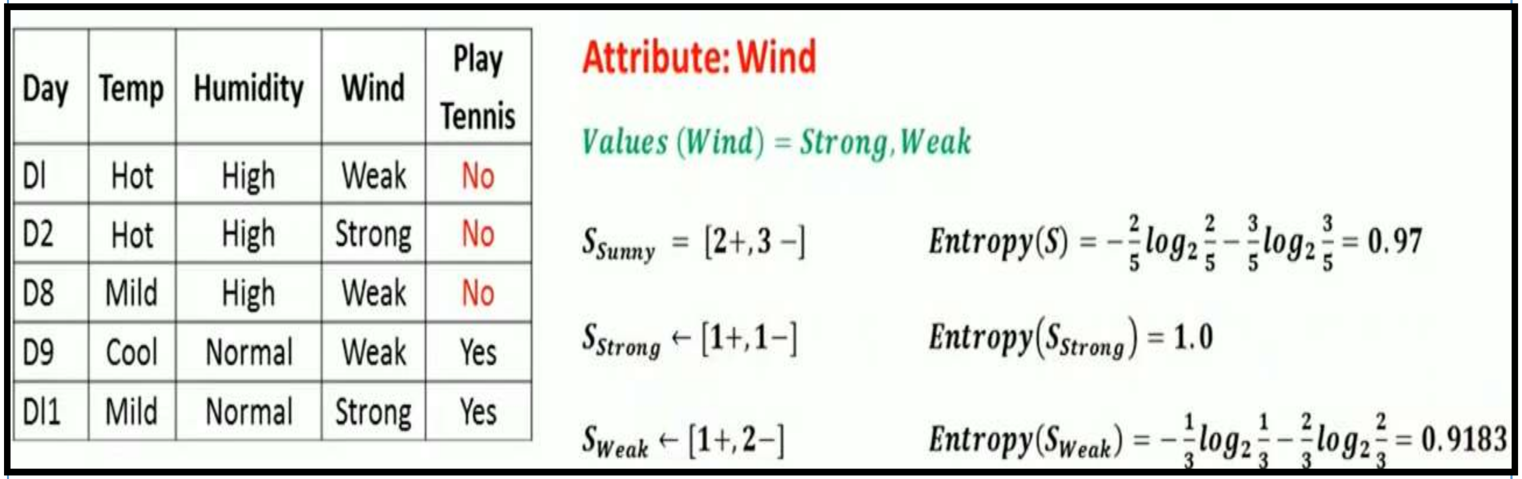


$$Gain\ (S_{Sunny}, Humidity) = Entropy(S) - \sum_{v\ \in\{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Humidity) = Entropy(S) - \frac{3}{5}Entropy(S_{High}) - \frac{2}{5}Entropy(S_{Normal})$$

$$Gain(S_{sunny}, Humidity) = 0.97 - \frac{3}{5}\,0.0 - \frac{2}{5}0.0 = 0.97$$

- Next attribute is **{Wind}**



Day	Temp	Humidity	Wind	Play Tennis
DI	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
DI1	Mild	Normal	Strong	Yes

Attribute: Wind

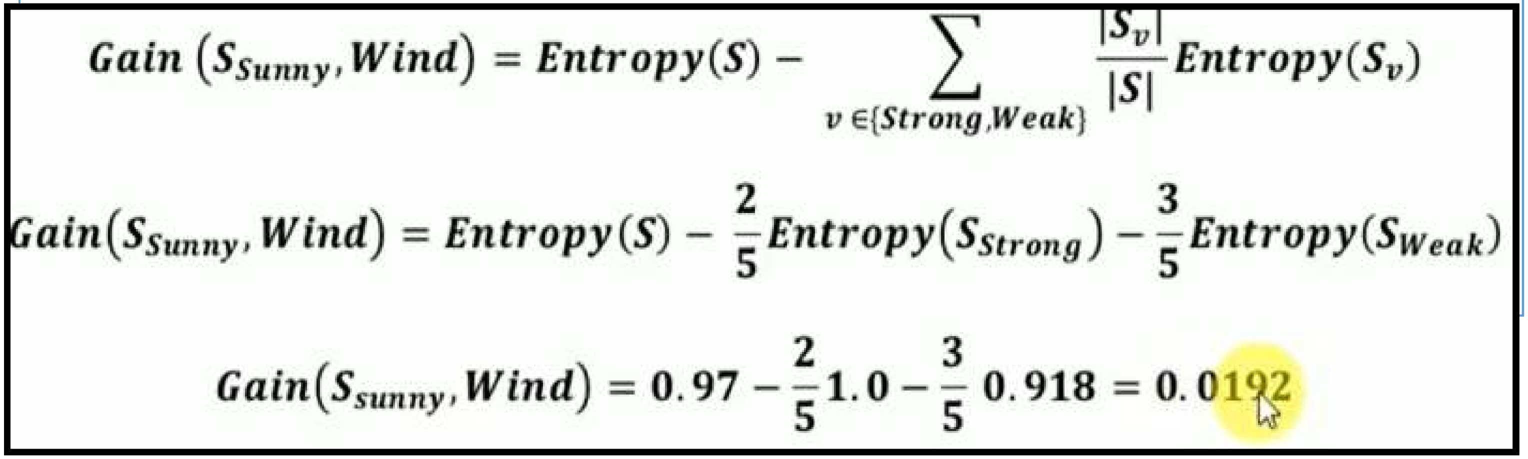
$Values\ (Wind) = Strong, Weak$

$S_{Sunny} = [2+, 3-]$ $\quad Entropy(S) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$

$S_{Strong} \leftarrow [1+, 1-]$ $\quad Entropy(S_{Strong}) = 1.0$

$S_{Weak} \leftarrow [1+, 2-]$ $\quad Entropy(S_{Weak}) = -\frac{1}{3}log_2\frac{1}{3} - \frac{2}{3}log_2\frac{2}{3} = 0.9183$

Next, Calculate the Information Gain of Wind

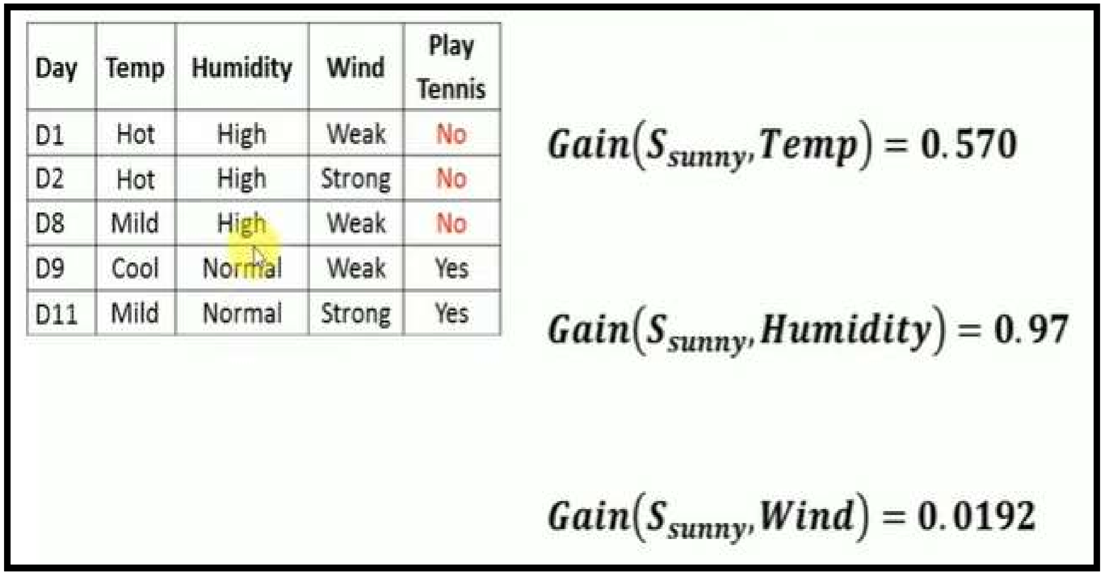


$$Gain\ (S_{Sunny}, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Wind) = Entropy(S) - \frac{2}{5}Entropy(S_{Strong}) - \frac{3}{5}Entropy(S_{Weak})$$

$$Gain(S_{sunny}, Wind) = 0.97 - \frac{2}{5}1.0 - \frac{3}{5}\ 0.918 = 0.0192$$

- So, we calculated all the Attributes. Next we need to check the **Maximum Information Gain** of **these attributes.**



Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

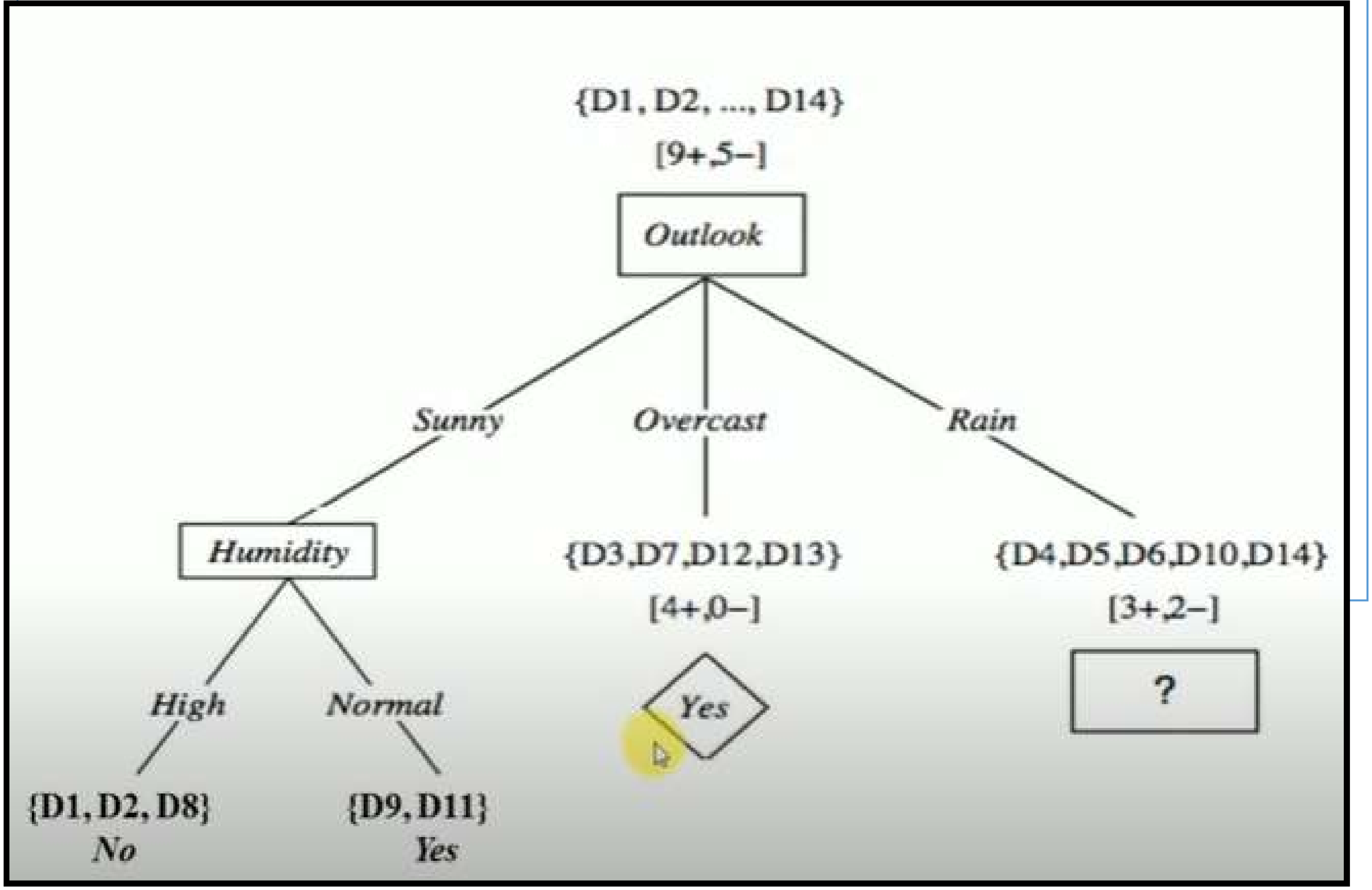
$$Gain(S_{sunny}, Temp) = 0.570$$

$$Gain(S_{sunny}, Humidity) = 0.97$$

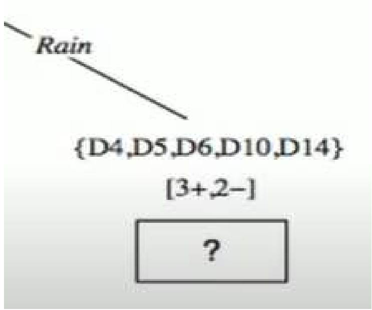
$$Gain(S_{sunny}, Wind) = 0.0192$$

- **SO, here HUMIDITY is having Maximum Information Gain . So that HUMIDITY is the <u>Internal Node.</u>**

- So that the **updated Tree is**



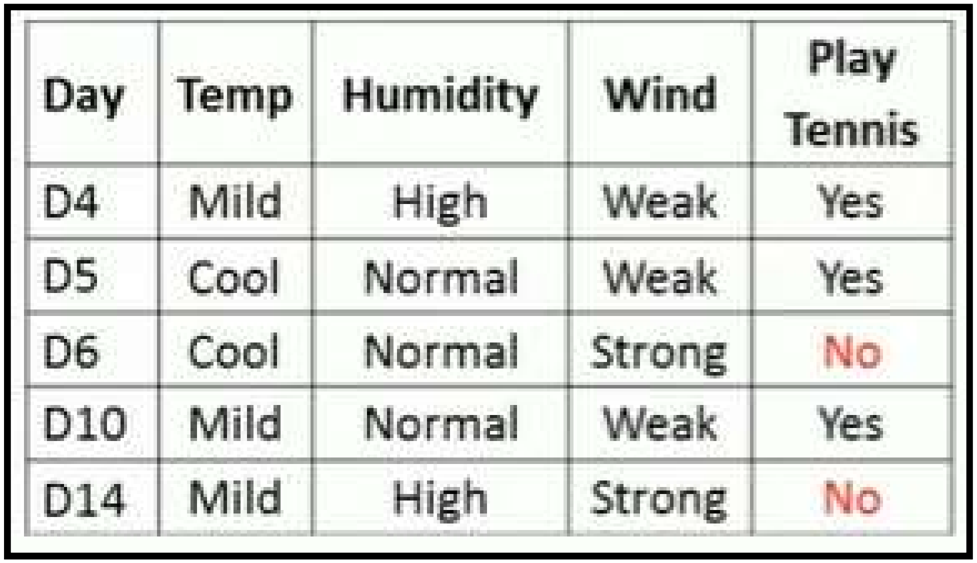
- Next we need to calculate the Right hand Side **“ Rain”** attributes to calculate the **Internal Nodes**.



- **So, here we can take only**

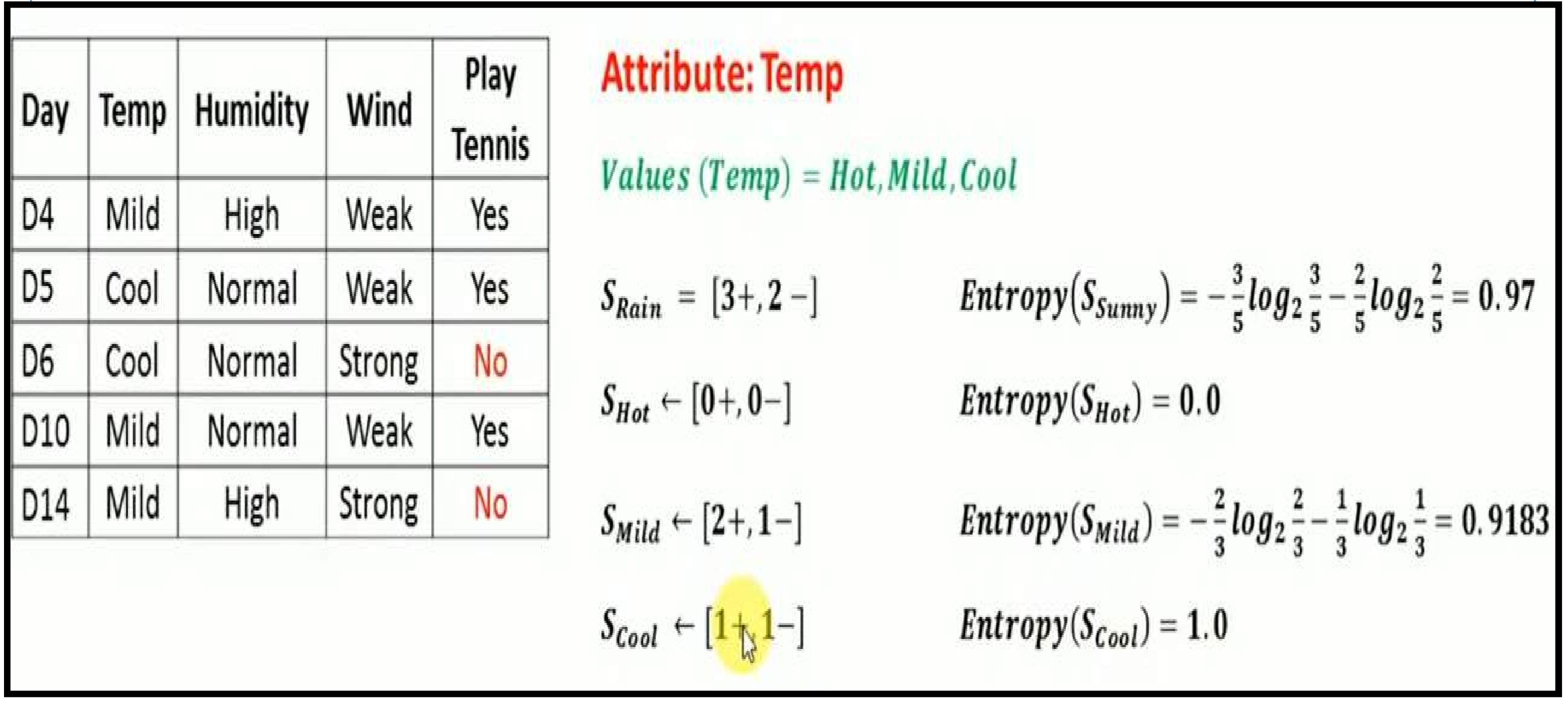
  { D4, D5, D6, D10, D14}

-



Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

- Again the **same procedure**, First We can calculate the **Entropy of each attribute**. So, here the **First Attribute** is **" TEMP"**



Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

Attribute: Temp

$Values\,(Temp) = Hot, Mild, Cool$

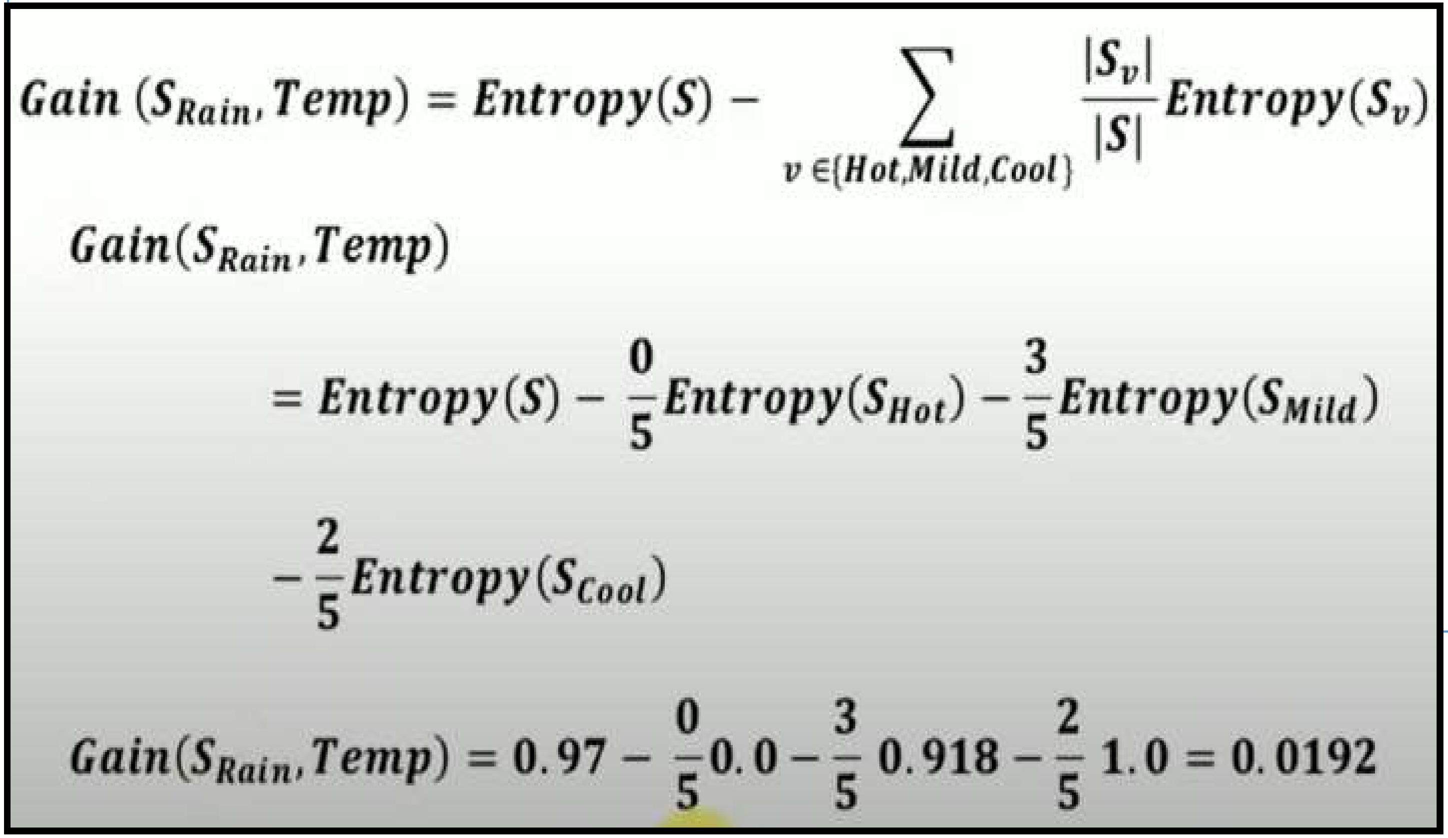
$S_{Rain} = [3+, 2-]$ $\qquad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$

$S_{Hot} \leftarrow [0+, 0-]$ $\qquad Entropy(S_{Hot}) = 0.0$

$S_{Mild} \leftarrow [2+, 1-]$ $\qquad Entropy(S_{Mild}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$

$S_{Cool} \leftarrow [1+, 1-]$ $\qquad Entropy(S_{Cool}) = 1.0$

- Next we can calculate the **Information Gain** of **Temp**



$$Gain\ (S_{Rain}, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

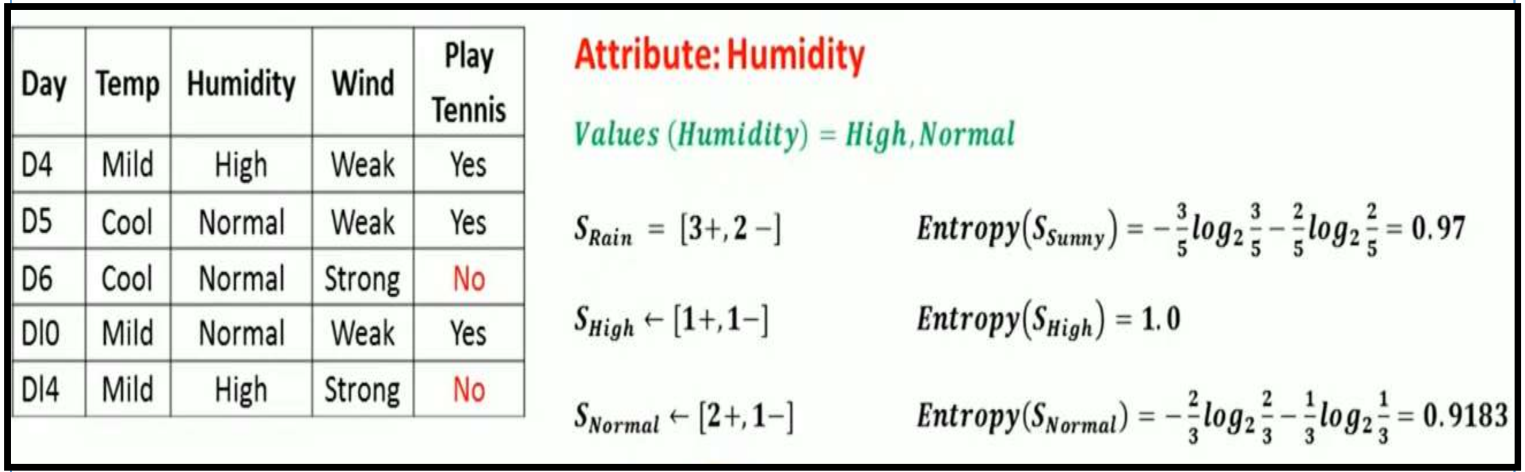
$$Gain(S_{Rain}, Temp)$$

$$= Entropy(S) - \frac{0}{5} Entropy(S_{Hot}) - \frac{3}{5} Entropy(S_{Mild})$$

$$- \frac{2}{5} Entropy(S_{Cool})$$

$$Gain(S_{Rain}, Temp) = 0.97 - \frac{0}{5} 0.0 - \frac{3}{5}\ 0.918 - \frac{2}{5}\ 1.0 = 0.0192$$

- Next attribute is **{Humidity}**



Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
DI0	Mild	Normal	Weak	Yes
DI4	Mild	High	Strong	No

Attribute: Humidity

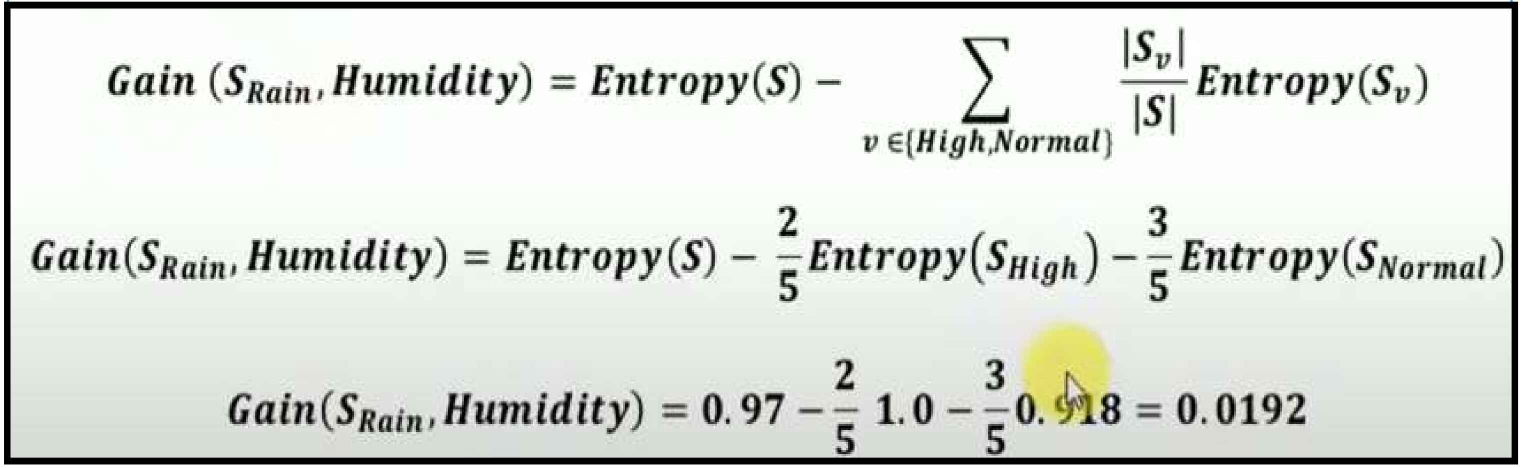
Values (Humidity) = High, Normal

$S_{Rain} = [3+, 2-]$ $\quad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$

$S_{High} \leftarrow [1+, 1-]$ $\quad Entropy(S_{High}) = 1.0$

$S_{Normal} \leftarrow [2+, 1-]$ $\quad Entropy(S_{Normal}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$

Next, Calculate the Information Gain of Humidity

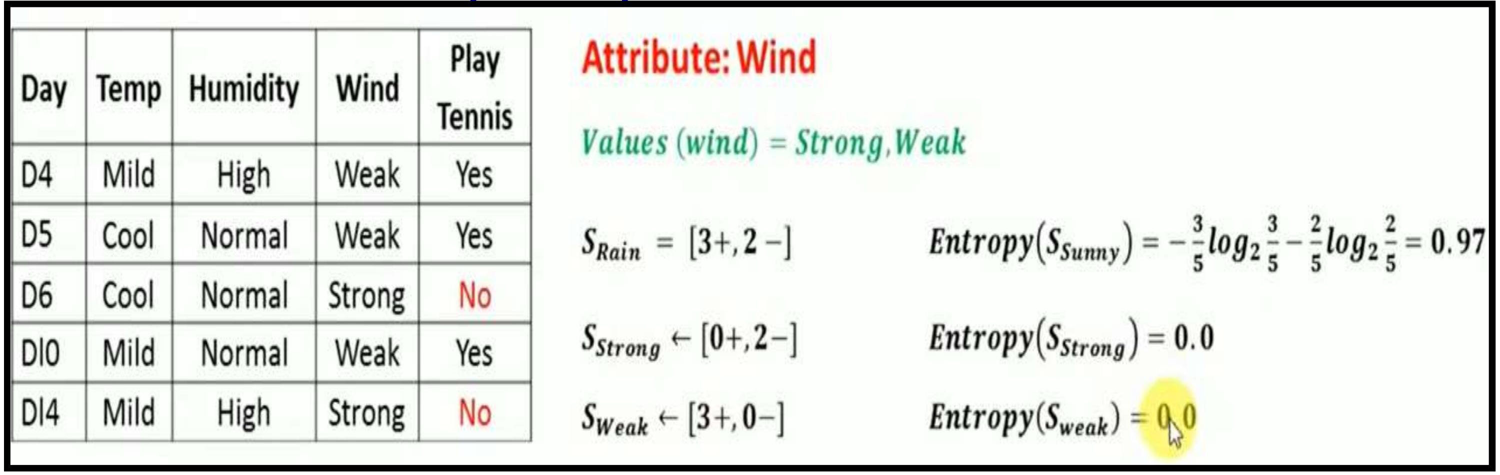


$$Gain\ (S_{Rain}, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Humidity) = Entropy(S) - \frac{2}{5} Entropy(S_{High}) - \frac{3}{5} Entropy(S_{Normal})$$

$$Gain(S_{Rain}, Humidity) = 0.97 - \frac{2}{5}\ 1.0 - \frac{3}{5} 0.918 = 0.0192$$

- Next attribute is **{Wind}**



Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
DI0	Mild	Normal	Weak	Yes
DI4	Mild	High	Strong	No

Attribute: Wind

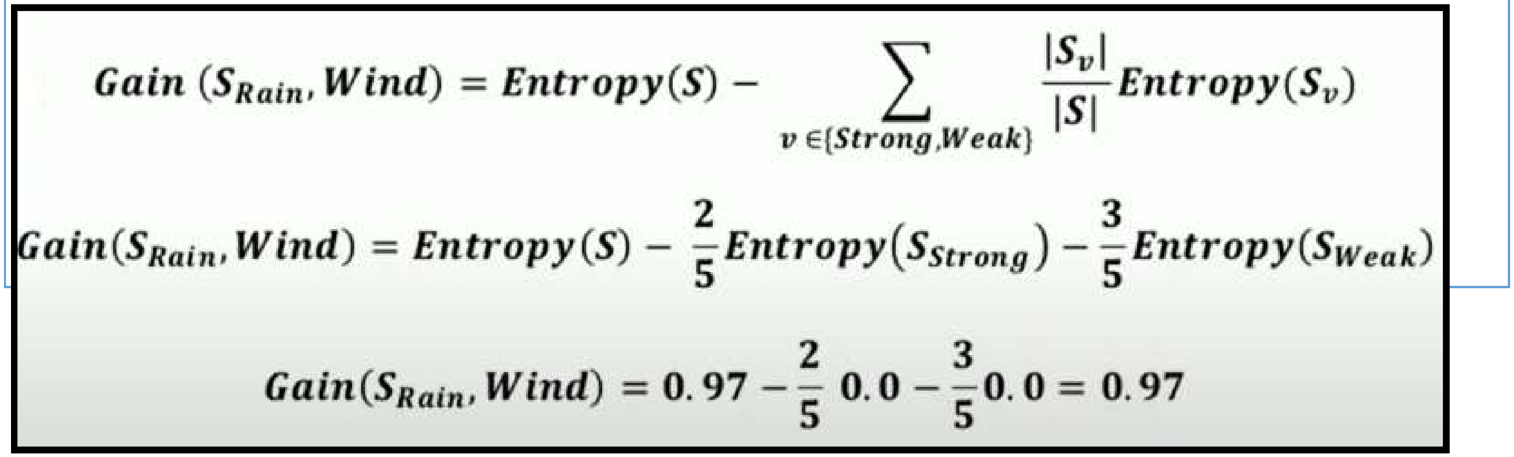
$Values\ (wind) = Strong, Weak$

$S_{Rain} = [3+, 2-]$ $Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$

$S_{Strong} \leftarrow [0+, 2-]$ $Entropy(S_{Strong}) = 0.0$

$S_{Weak} \leftarrow [3+, 0-]$ $Entropy(S_{weak}) = 0.0$

Next, Calculate the Information Gain of Wind

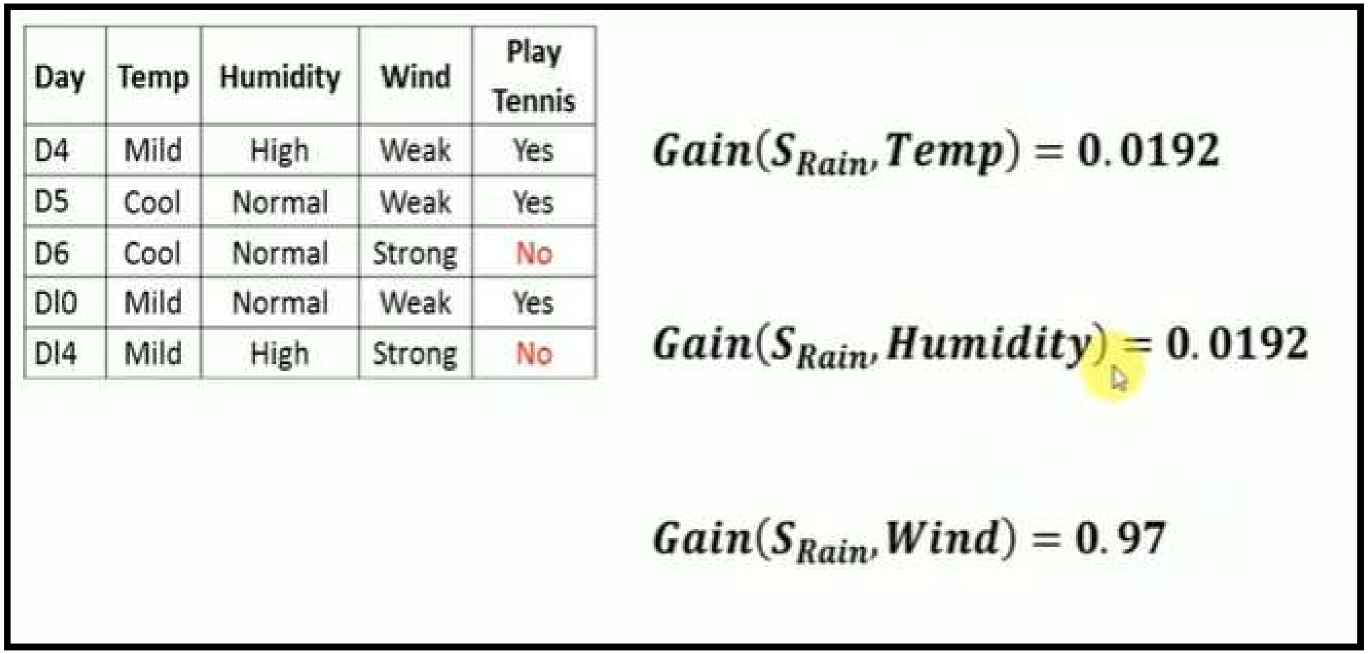


$$Gain\ (S_{Rain}, Wind) = Entropy(S) - \sum_{v\ \in\{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Wind) = Entropy(S) - \frac{2}{5}Entropy(S_{Strong}) - \frac{3}{5}Entropy(S_{Weak})$$

$$Gain(S_{Rain}, Wind) = 0.97 - \frac{2}{5}\ 0.0 - \frac{3}{5}0.0 = 0.97$$

- So, we calculated all the Attributes. Next we need to check the **Maximum Information Gain** of **these attributes.**



Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
DI0	Mild	Normal	Weak	Yes
DI4	Mild	High	Strong	No

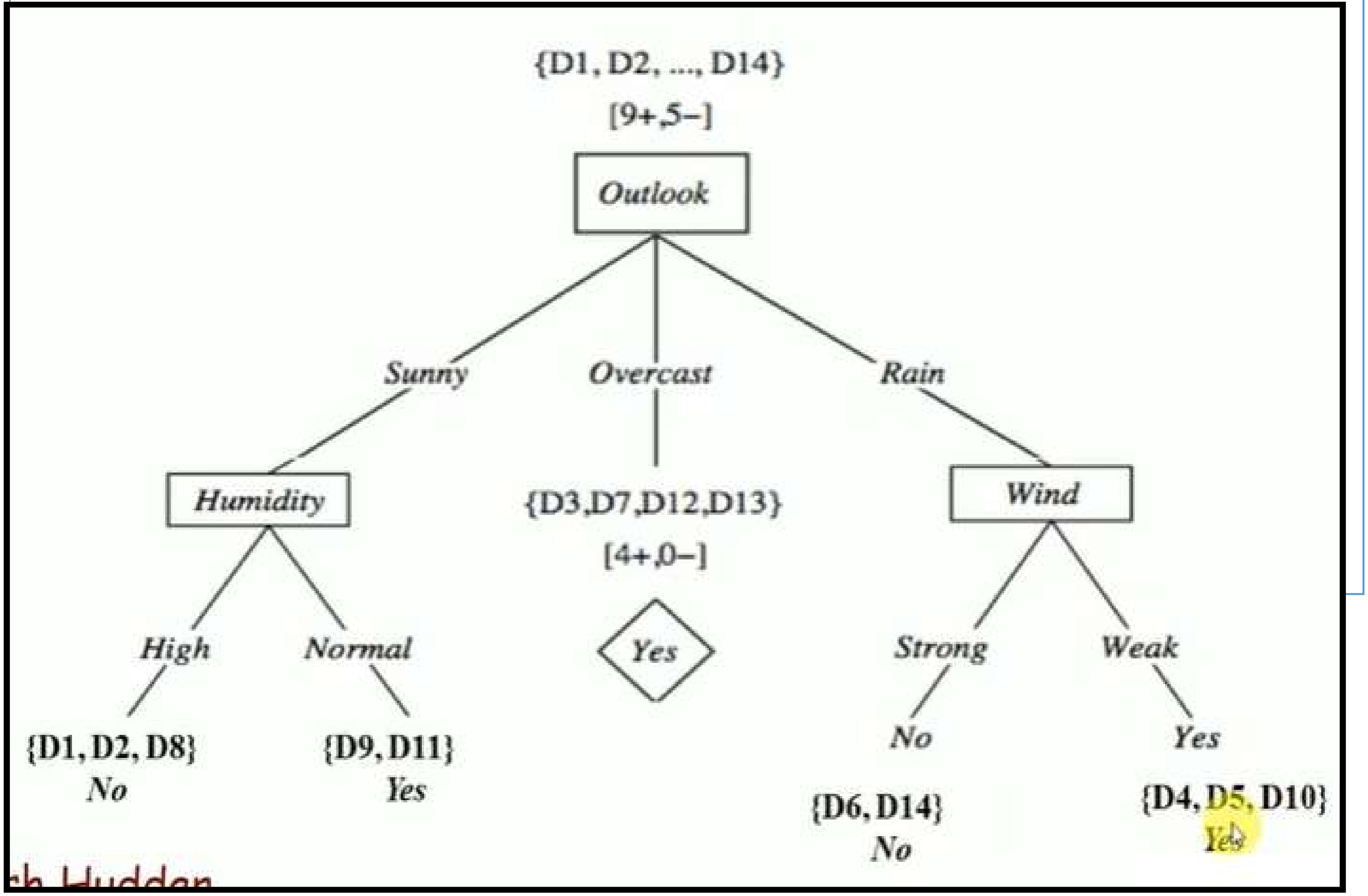
$$Gain(S_{Rain}, Temp) = 0.0192$$

$$Gain(S_{Rain}, Humidity) = 0.0192$$

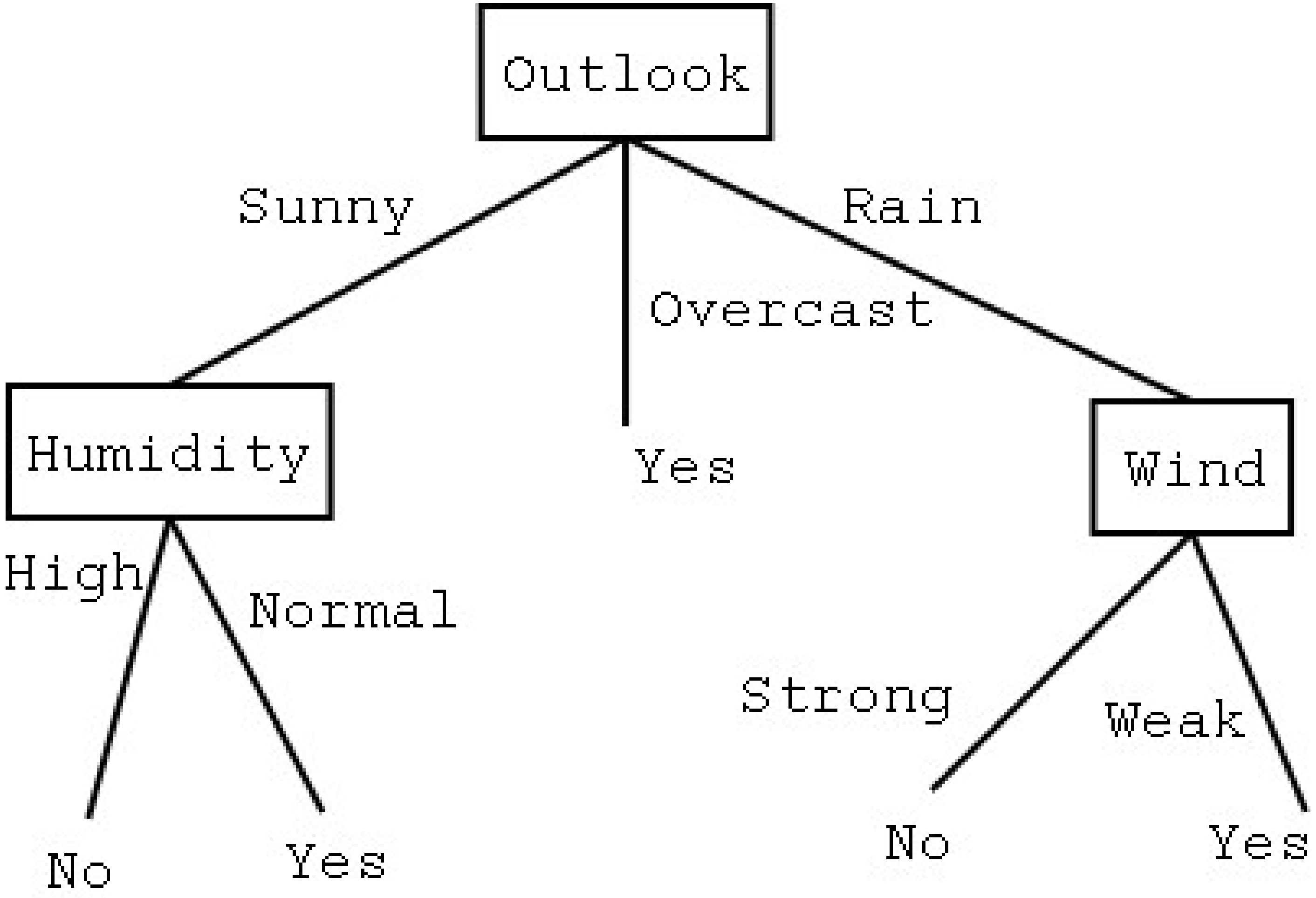
$$Gain(S_{Rain}, Wind) = 0.97$$

- **SO, here WIND is having Maximum Information Gain . So that WIND is the Internal Node(Right Side).**

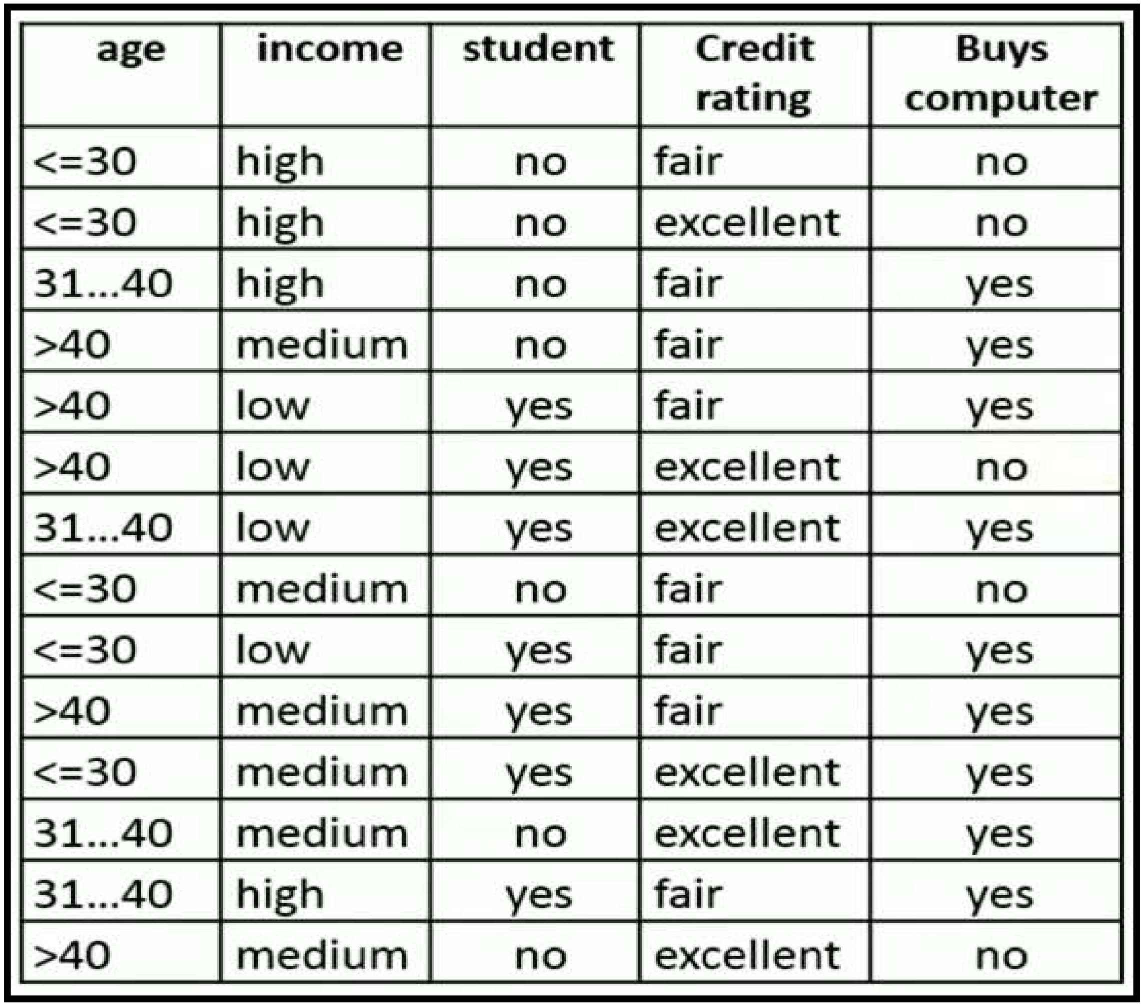
- So that the **updated Tree is**



- So Finally **updated Tree and Final Tree is**

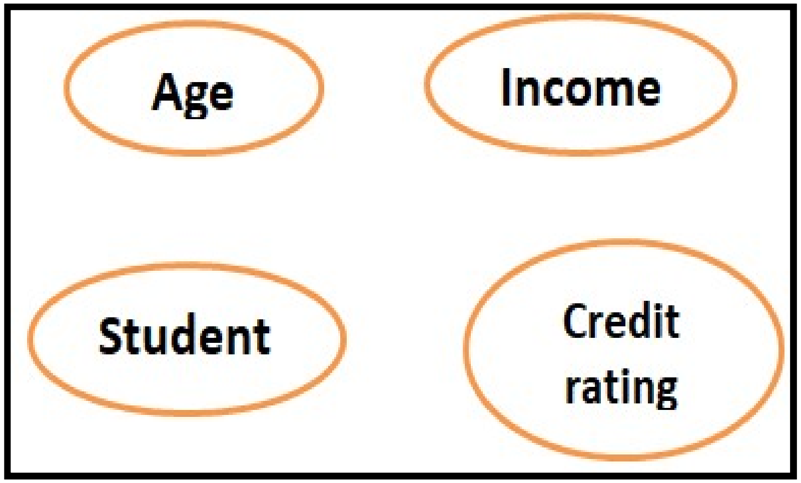


EXAMPLE-2



age	income	student	Credit rating	Buys computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

- **In this Data Set contains 14 instances and 4 Attributes**
- **The 4 Attributes are**



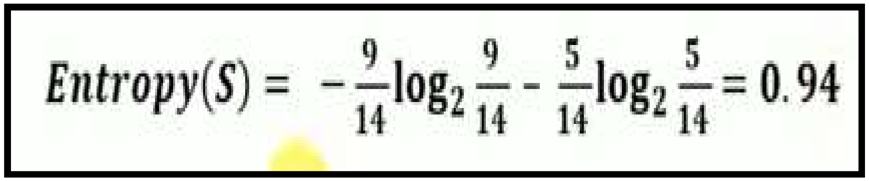
Age

Income

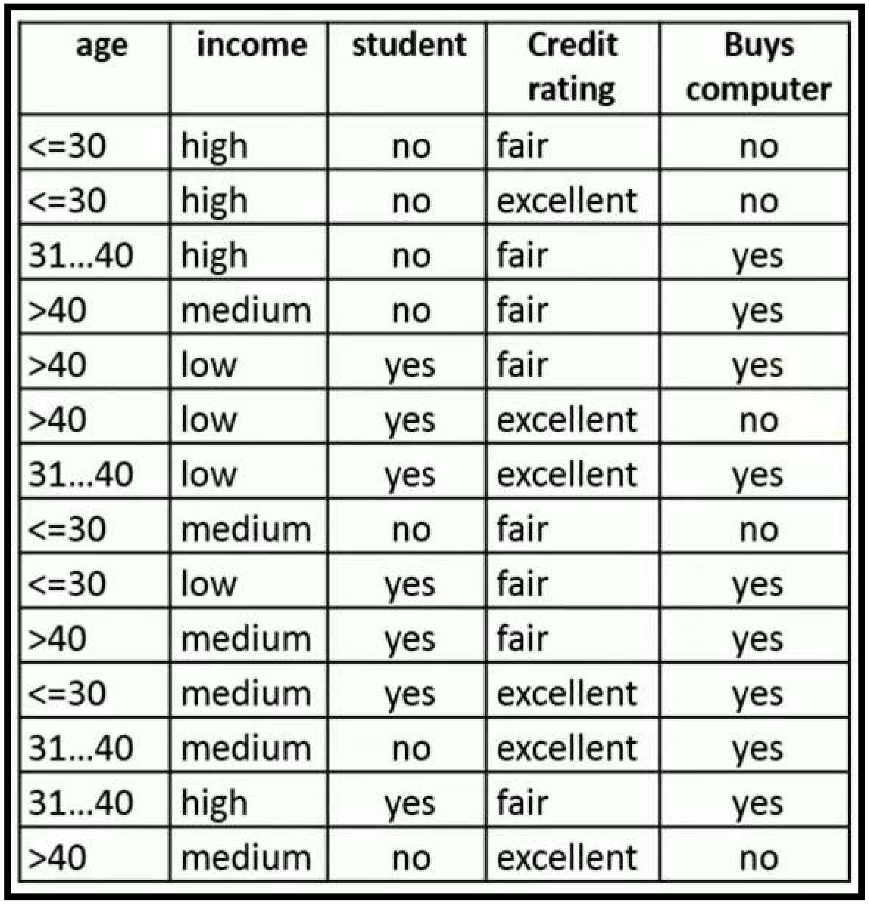
Student

Credit rating

Step-1: Compute the Entropy of Entire Dataset.

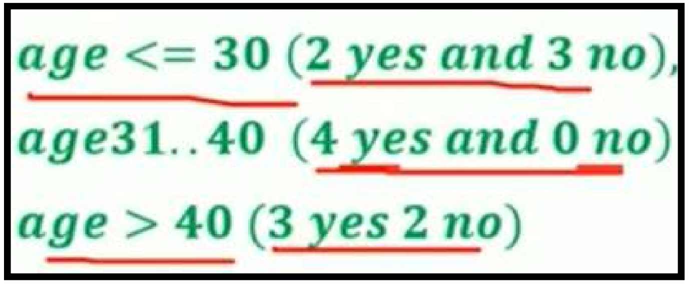


$$Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$$



age	income	student	Credit rating	Buys computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

- After that we need to calculate the Entropy and Information Gain of each and every attribute . So, the First Attribute is Age

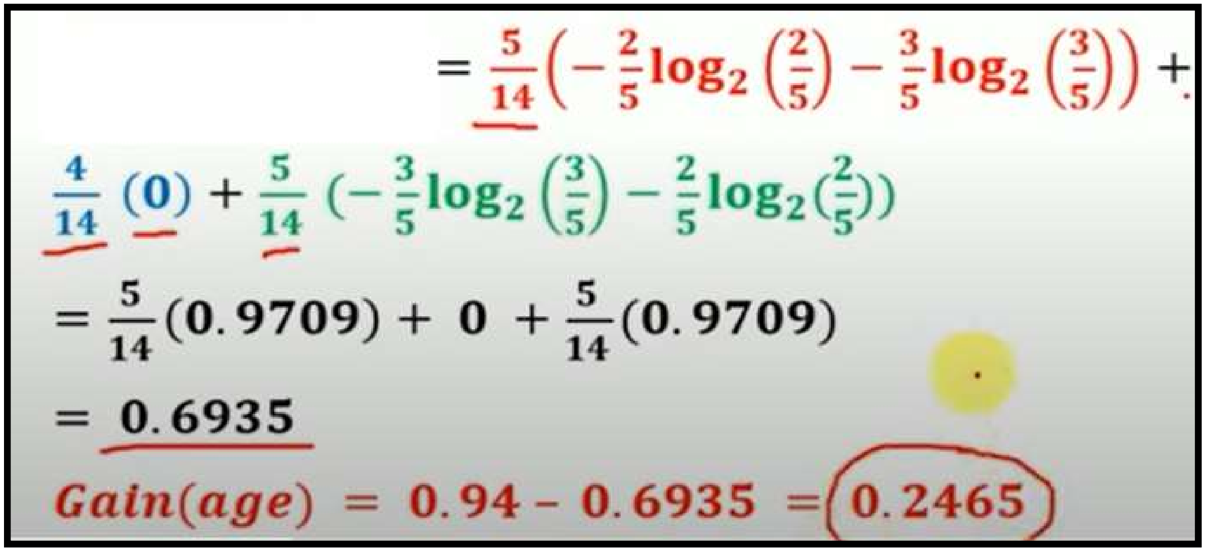


$age <= 30$ (2 yes and 3 no),

$age31..40$ (4 yes and 0 no)

$age > 40$ (3 yes 2 no)

- **Entropy & Information Gain:**



$$= \frac{5}{14}\left(-\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right) +$$

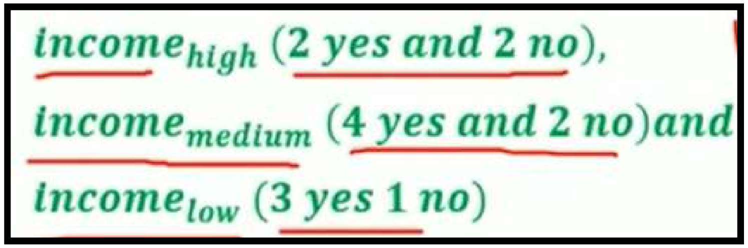
$$\frac{4}{14}(0) + \frac{5}{14}\left(-\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right)\right)$$

$$= \frac{5}{14}(0.9709) + 0 + \frac{5}{14}(0.9709)$$

$$= \underline{0.6935}$$

$$Gain(age) = 0.94 - 0.6935 = \boxed{0.2465}$$

- Similarly we need to calculate the remaining 3 attributes.
- Next attribute is {Income}

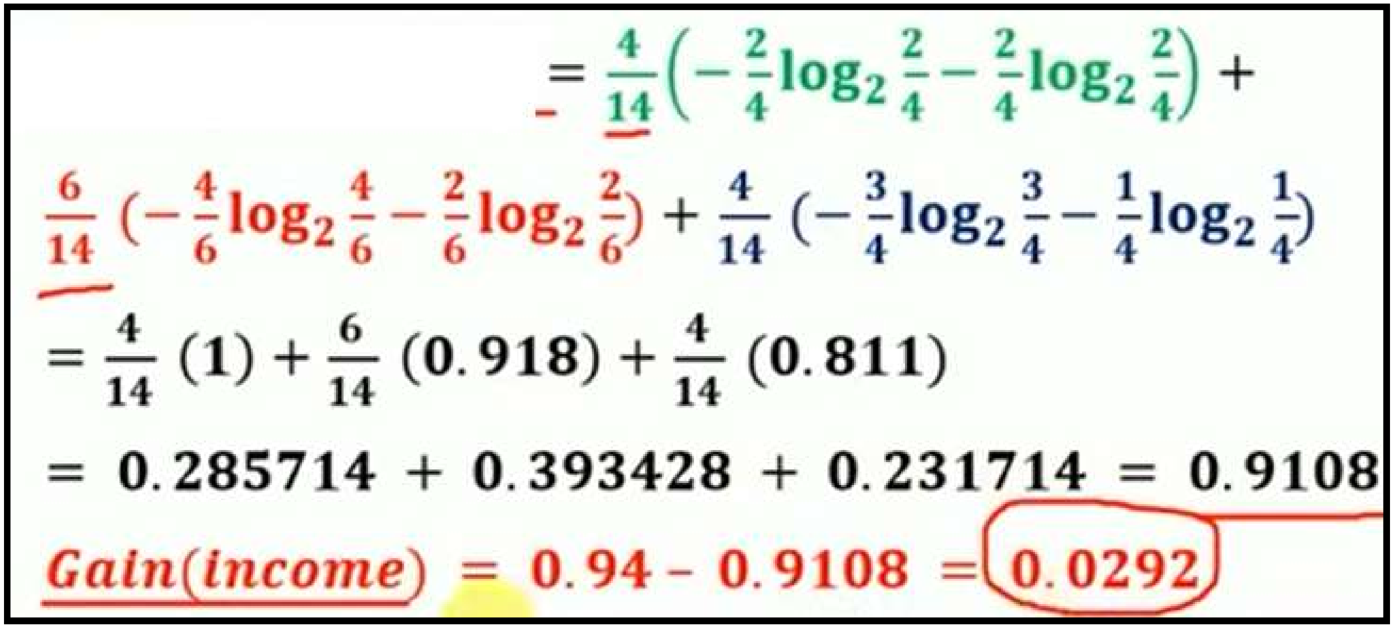


$\underline{income_{high}}$ $(\underline{2\ yes\ and\ 2\ no}),$

$\underline{income_{medium}}$ $(\underline{4\ yes\ and\ 2\ no})and$

$\underline{income_{low}}$ $(\underline{3\ yes\ 1\ no})$

Entropy & Information Gain:



$$= \underline{\frac{4}{14}}\left(-\frac{2}{4}\log_2\frac{2}{4}-\frac{2}{4}\log_2\frac{2}{4}\right)+$$

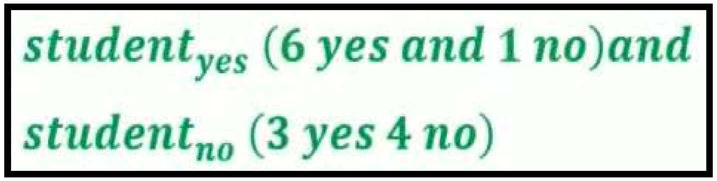
$$\underline{\frac{6}{14}}\left(-\frac{4}{6}\log_2\frac{4}{6}-\frac{2}{6}\log_2\frac{2}{6}\right)+\frac{4}{14}\left(-\frac{3}{4}\log_2\frac{3}{4}-\frac{1}{4}\log_2\frac{1}{4}\right)$$

$$= \frac{4}{14}(1)+\frac{6}{14}(0.918)+\frac{4}{14}(0.811)$$

$$= 0.285714 + 0.393428 + 0.231714 = 0.9108$$

$$\underline{Gain(income)} = 0.94 - 0.9108 = \boxed{0.0292}$$

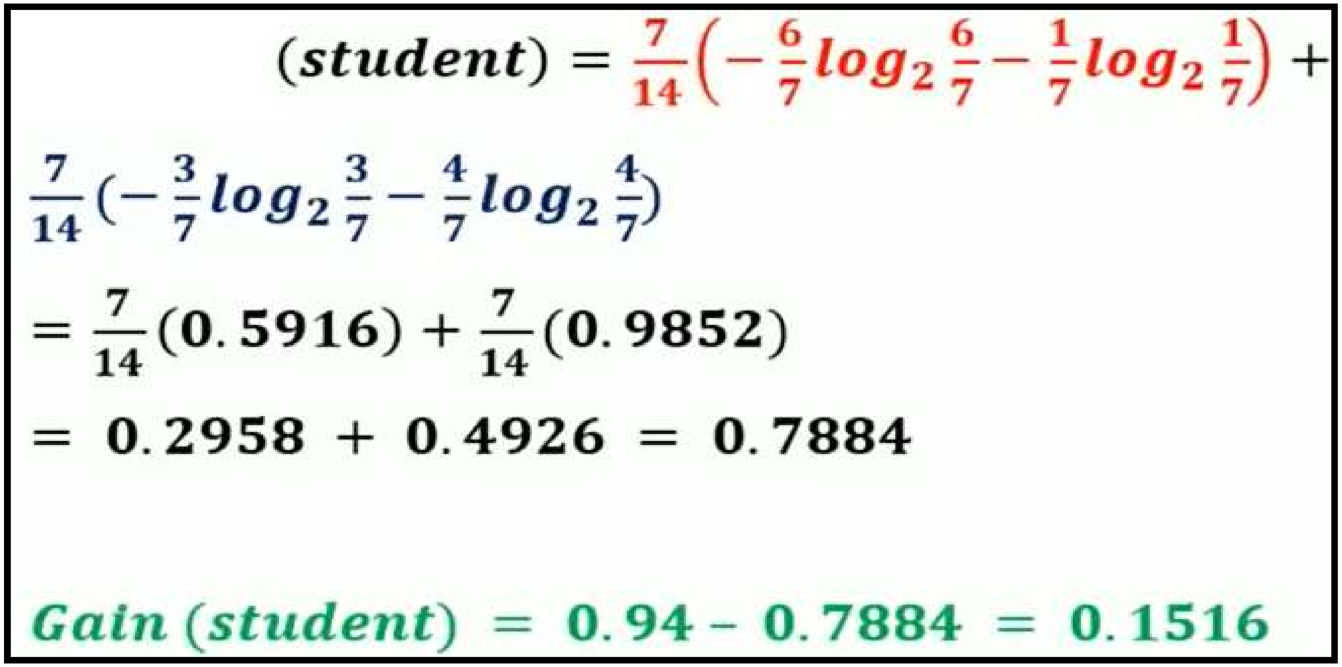
- Next attribute is **{Student}**



$student_{yes}$ $(6\ yes\ and\ 1\ no) and$

$student_{no}$ $(3\ yes\ 4\ no)$

Entropy & Information Gain:



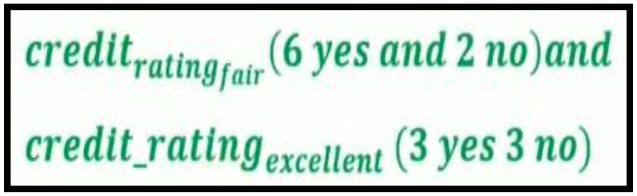
$$(student) = \frac{7}{14}\left(-\frac{6}{7}log_2\frac{6}{7} - \frac{1}{7}log_2\frac{1}{7}\right) + \frac{7}{14}\left(-\frac{3}{7}log_2\frac{3}{7} - \frac{4}{7}log_2\frac{4}{7}\right)$$

$$= \frac{7}{14}(0.5916) + \frac{7}{14}(0.9852)$$

$$= 0.2958 + 0.4926 = 0.7884$$

$$Gain\ (student) = 0.94 - 0.7884 = 0.1516$$

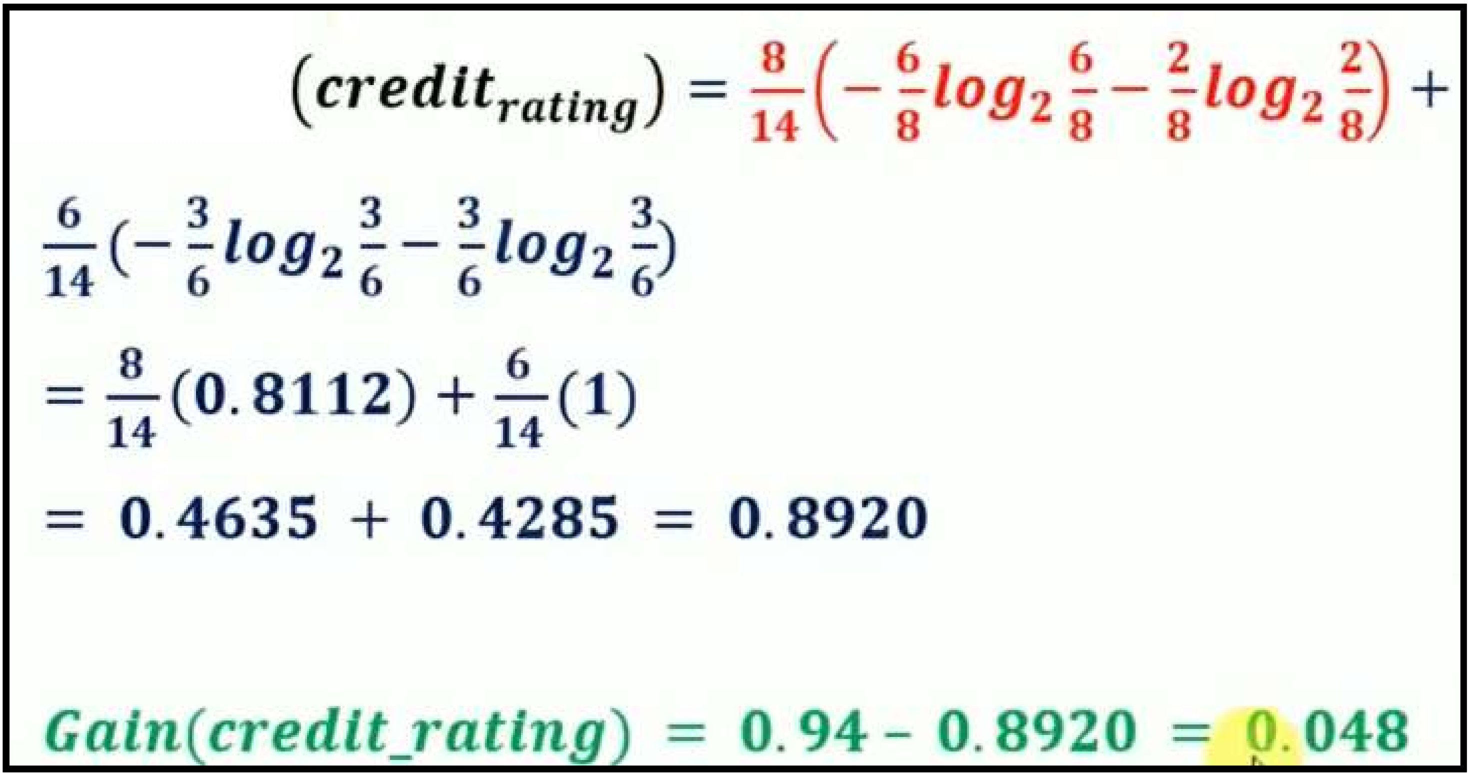
- Next attribute is **{Credit Rating}**



$credit_{rating_{fair}}(6\ yes\ and\ 2\ no) and$

$credit\_rating_{excellent}\ (3\ yes\ 3\ no)$

Entropy & Information Gain:



$$(credit_{rating}) = \frac{8}{14}\left(-\frac{6}{8}log_2\frac{6}{8} - \frac{2}{8}log_2\frac{2}{8}\right) +$$

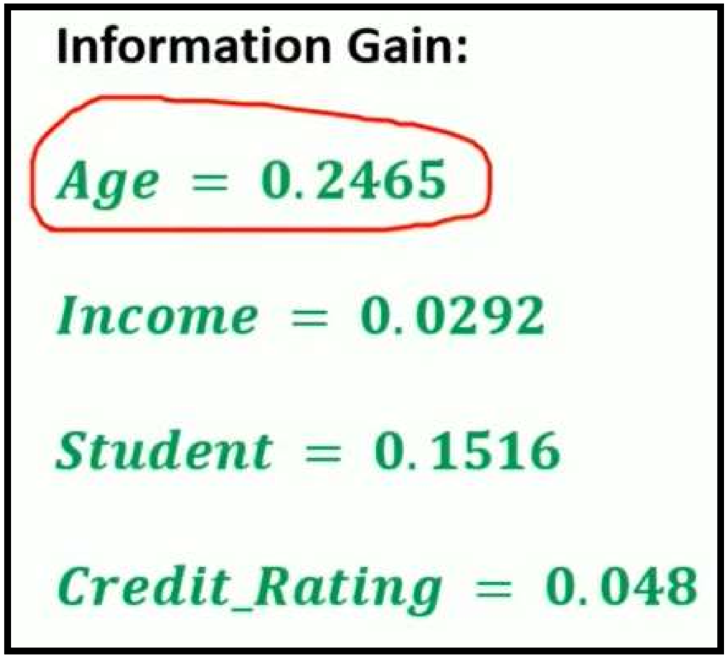
$$\frac{6}{14}(-\frac{3}{6}log_2\frac{3}{6} - \frac{3}{6}log_2\frac{3}{6})$$

$$= \frac{8}{14}(0.8112) + \frac{6}{14}(1)$$

$$= 0.4635 + 0.4285 = 0.8920$$

$$Gain(credit\_rating) = 0.94 - 0.8920 = 0.048$$

- So, we calculated all the Attributes. Next we need to check the Maximum Information Gain of these attributes.



Information Gain:

$Age = 0.2465$

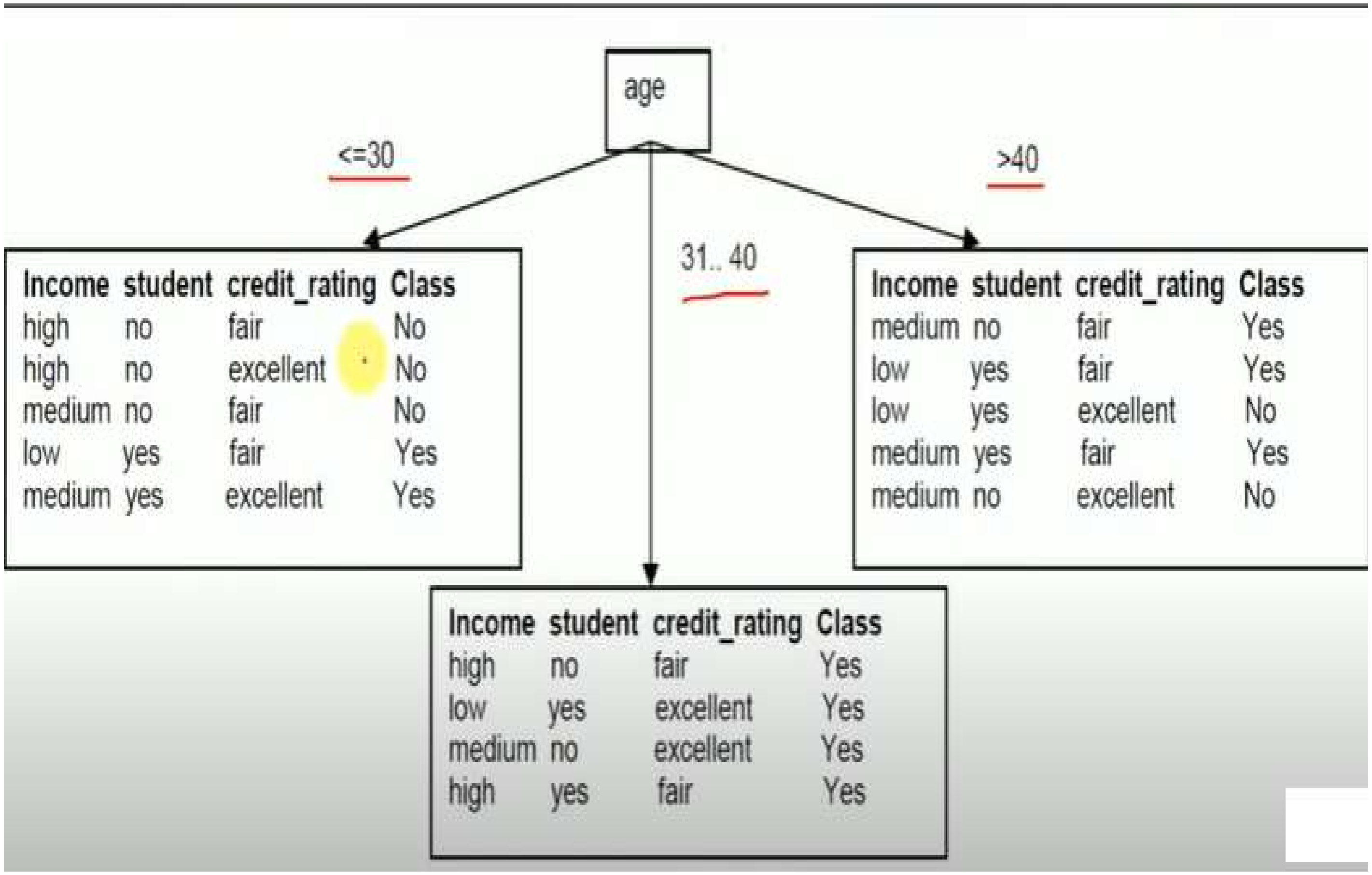
$Income = 0.0292$

$Student = 0.1516$

$Credit\_Rating = 0.048$

- SO, here AGE is having Maximum Information Gain . So that AGE is the Root Node.

- So, that the Root Node is **AGE.**

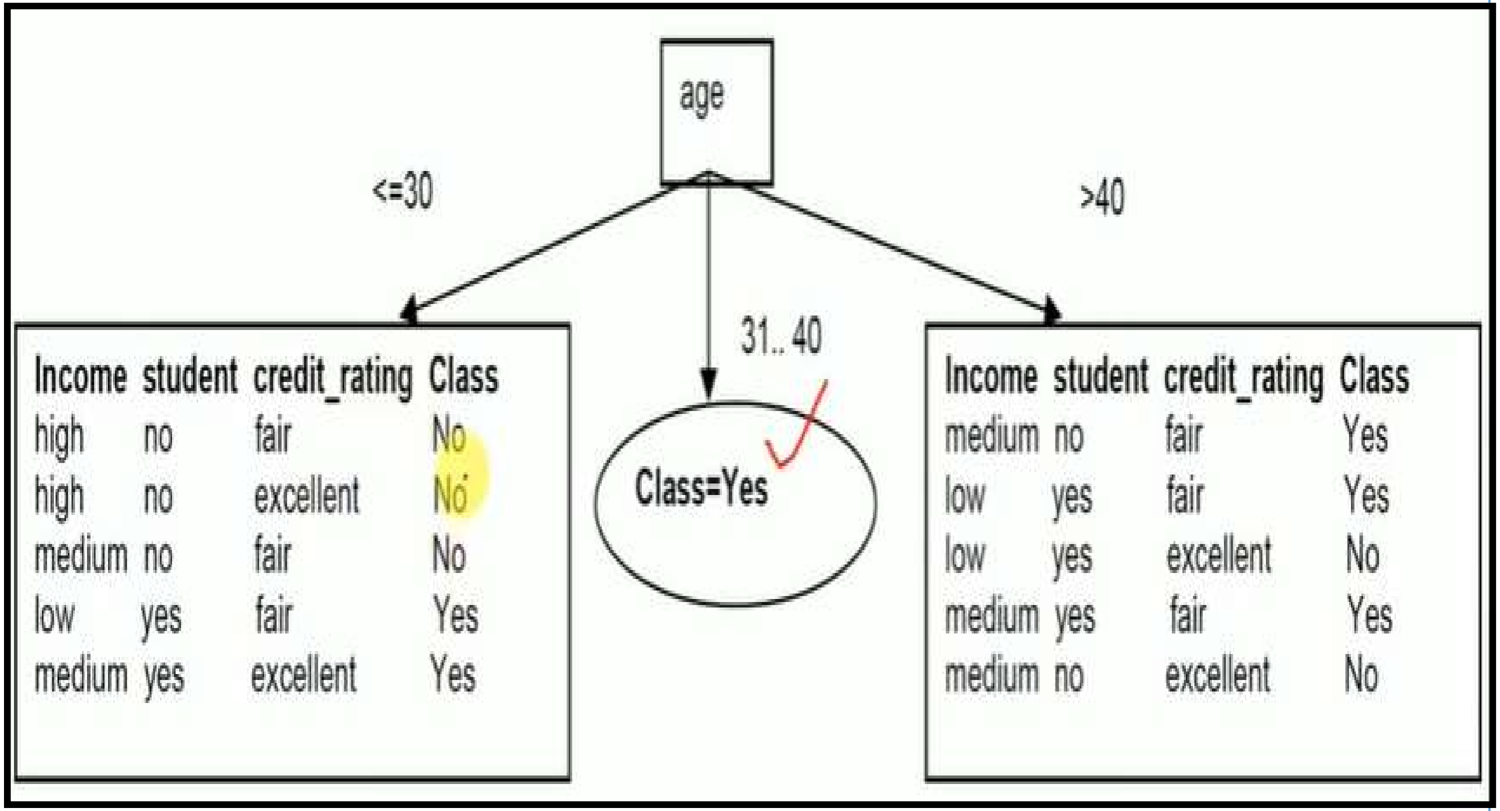


Income	student	credit_rating	Class
high	no	fair	No
high	no	excellent	No
medium	no	fair	No
low	yes	fair	Yes
medium	yes	excellent	Yes

Income	student	credit_rating	Class
high	no	fair	Yes
low	yes	excellent	Yes
medium	no	excellent	Yes
high	yes	fair	Yes

Income	student	credit_rating	Class
medium	no	fair	Yes
low	yes	fair	Yes
low	yes	excellent	No
medium	yes	fair	Yes
medium	no	excellent	No

- So, that the Root Node is **AGE**.

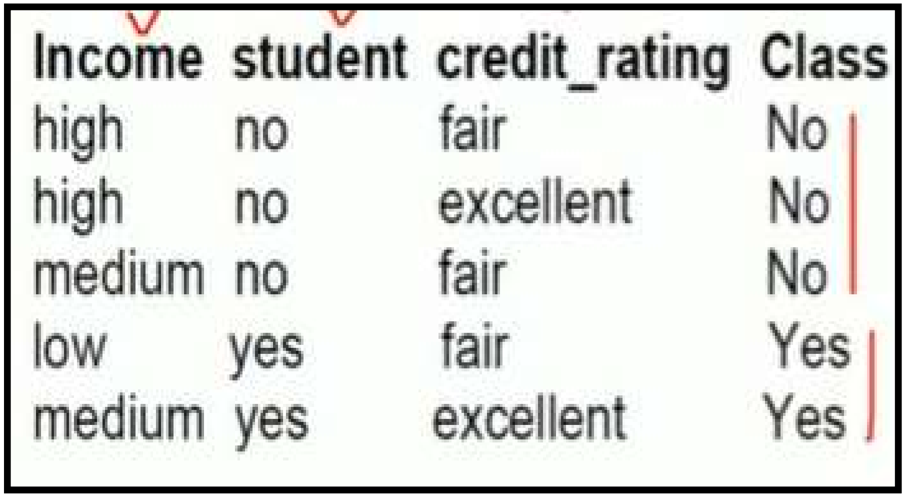


Income	student	credit_rating	Class
high	no	fair	No
high	no	excellent	No
medium	no	fair	No
low	yes	fair	Yes
medium	yes	excellent	Yes

Income	student	credit_rating	Class
medium	no	fair	Yes
low	yes	fair	Yes
low	yes	excellent	No
medium	yes	fair	Yes
medium	no	excellent	No

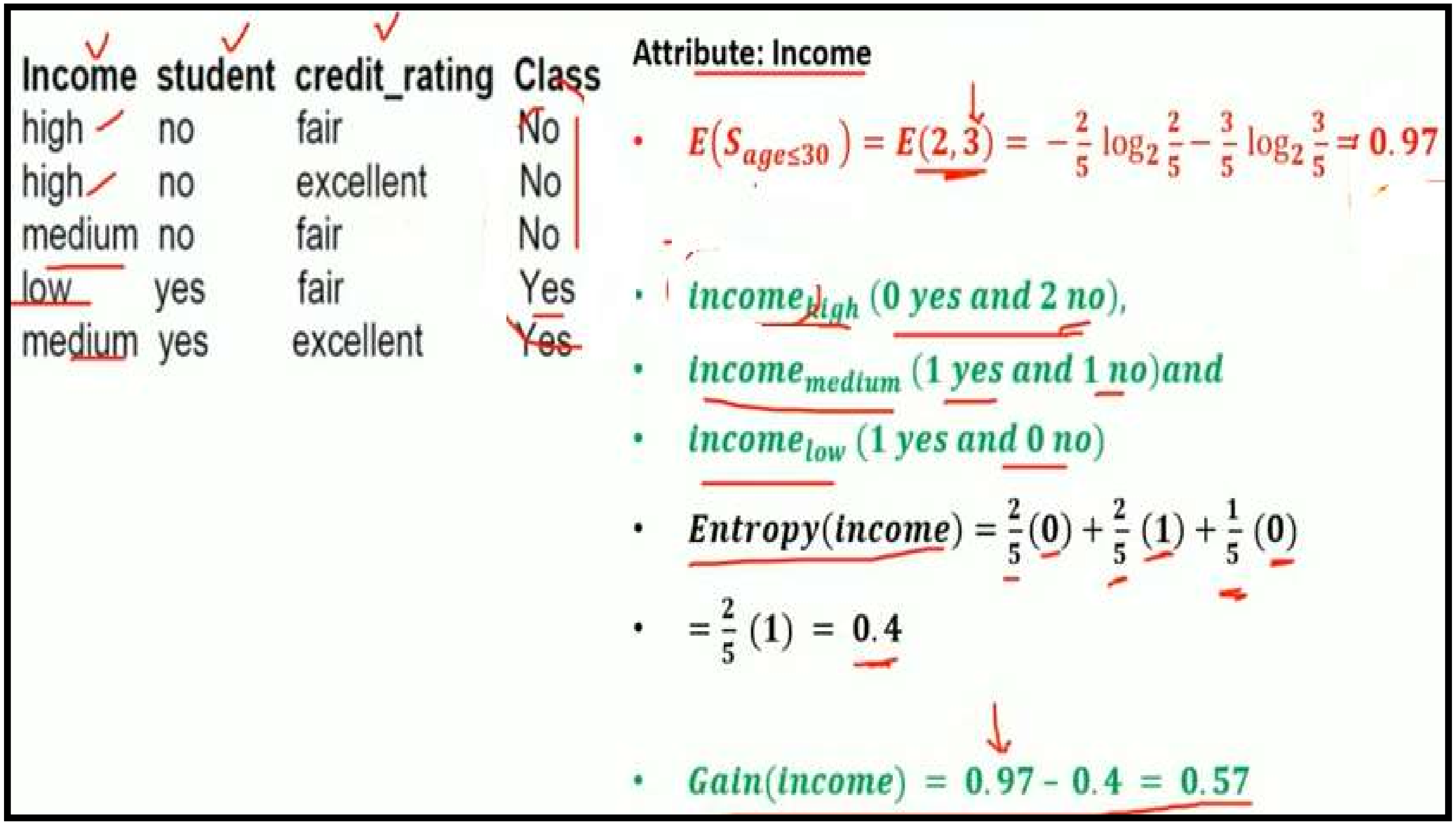
- Next, we need to take the Left Hand Side “ <=30” attributes to calculate the Internal Nodes.
- In this we have 3 Attributes=

  {Income, Student, Credit Rating}



Income	student	credit_rating	Class
high	no	fair	No
high	no	excellent	No
medium	no	fair	No
low	yes	fair	Yes
medium	yes	excellent	Yes

- Again the same procedure, First We can calculate the Entropy and information gain of each attribute. So, here the First Attribute is " INCOME"

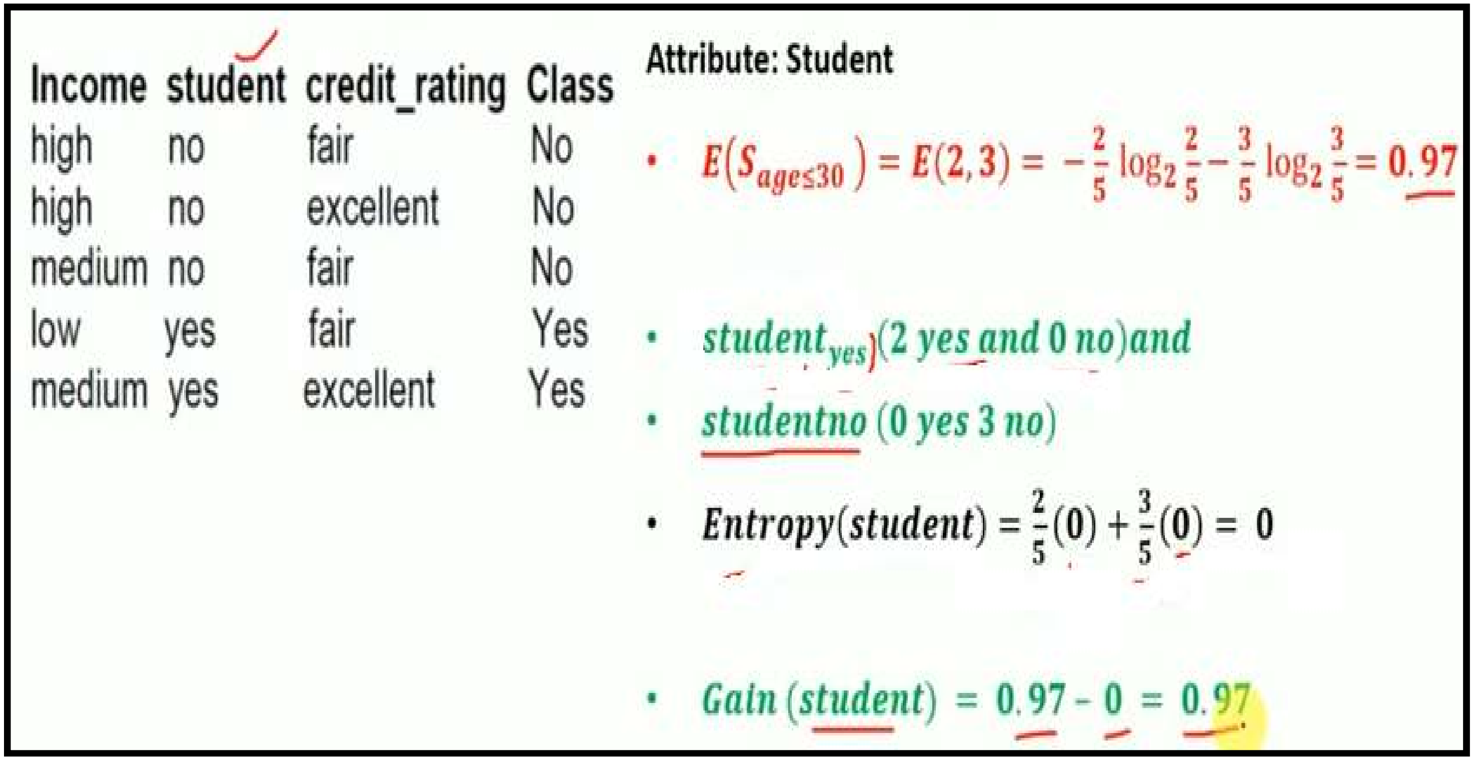


Income	student	credit_rating	Class
high	no	fair	No
high	no	excellent	No
medium	no	fair	No
low	yes	fair	Yes
medium	yes	excellent	Yes

Attribute: Income

- $E(S_{age \le 30}) = E(2,3) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.97$
- $income_{high}$ (0 yes and 2 no),
- $income_{medium}$ (1 yes and 1 no)and
- $income_{low}$ (1 yes and 0 no)
- $Entropy(income) = \frac{2}{5}(0) + \frac{2}{5}(1) + \frac{1}{5}(0)$
- $= \frac{2}{5}(1) = 0.4$
- $Gain(income) = 0.97 - 0.4 = 0.57$

- Next attribute is {Student}

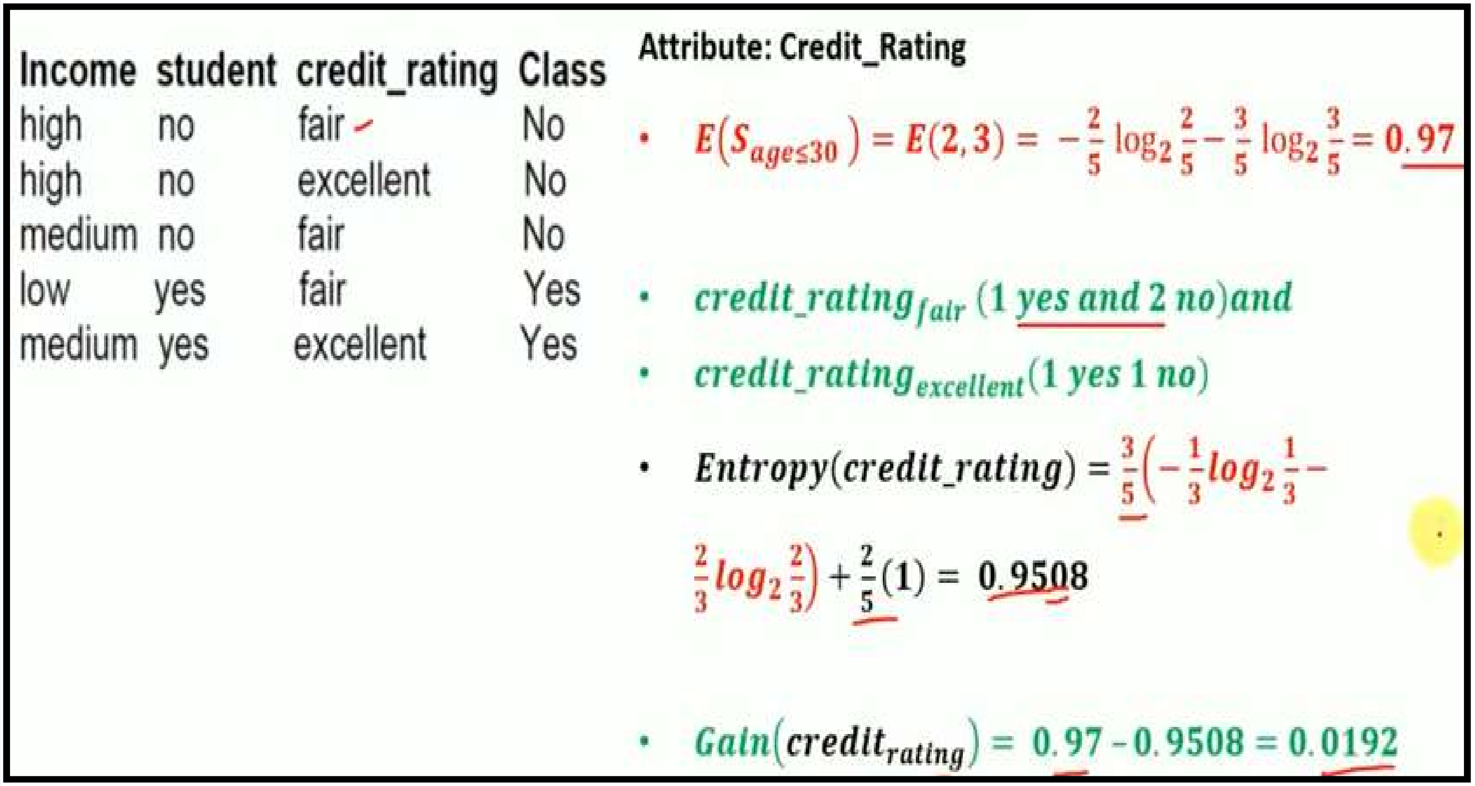


Income	student	credit_rating	Class
high	no	fair	No
high	no	excellent	No
medium	no	fair	No
low	yes	fair	Yes
medium	yes	excellent	Yes

Attribute: Student

- $E(S_{age\leq 30}) = E(2,3) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.97$
- $student_{yes)}(2\ yes\ and\ 0\ no)and$
- $studentno\ (0\ yes\ 3\ no)$
- $Entropy(student) = \frac{2}{5}(0) + \frac{3}{5}(0) = 0$
- $Gain\ (student) = 0.97 - 0 = 0.97$

- **Next attribute is {Credit_Rating}**

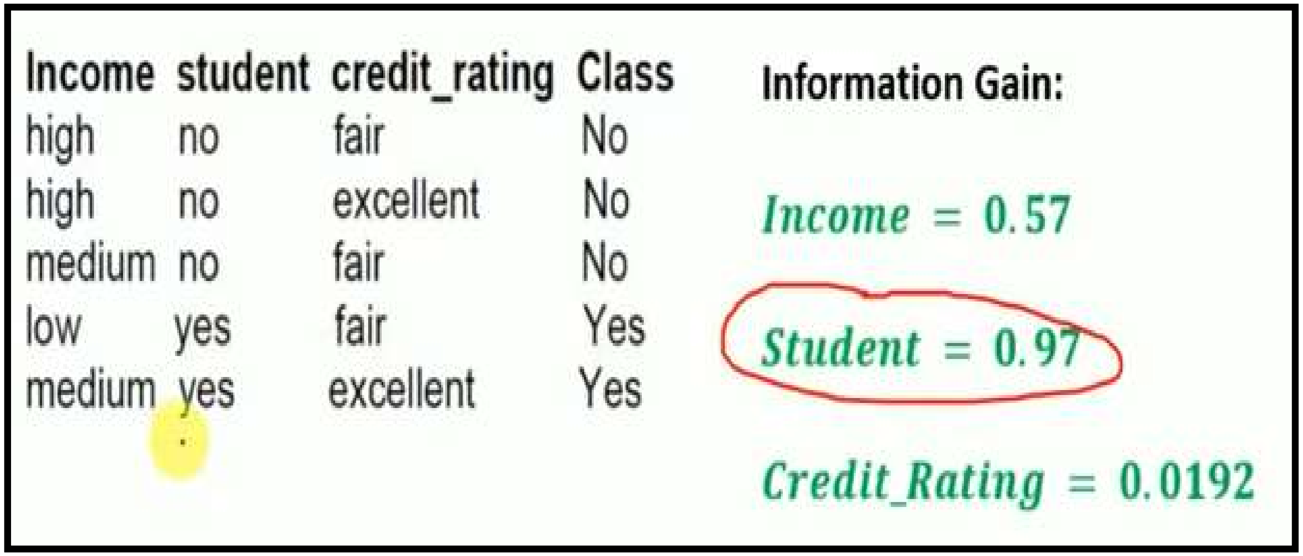


Income	student	credit_rating	Class
high	no	fair	No
high	no	excellent	No
medium	no	fair	No
low	yes	fair	Yes
medium	yes	excellent	Yes

Attribute: Credit_Rating

- $E(S_{age\leq30}) = E(2,3) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = \underline{0.97}$
- $credit\_rating_{fair}$ $(1\ \underline{yes\ and\ 2}\ no) and$
- $credit\_rating_{excellent}(1\ yes\ 1\ no)$
- $Entropy(credit\_rating) = \frac{\underline{3}}{5}\left(-\frac{1}{3}log_2\frac{1}{3} - \frac{2}{3}log_2\frac{2}{3}\right) + \frac{2}{\underline{5}}(1) = \underline{0.9508}$
- $Gain(credit_{rating}) = 0.\underline{97} - 0.9508 = 0.\underline{0192}$

- So, we calculated all the Attributes. Next we need to check the Maximum Information Gain of these attributes.



Income	student	credit_rating	Class
high	no	fair	No
high	no	excellent	No
medium	no	fair	No
low	yes	fair	Yes
medium	yes	excellent	Yes

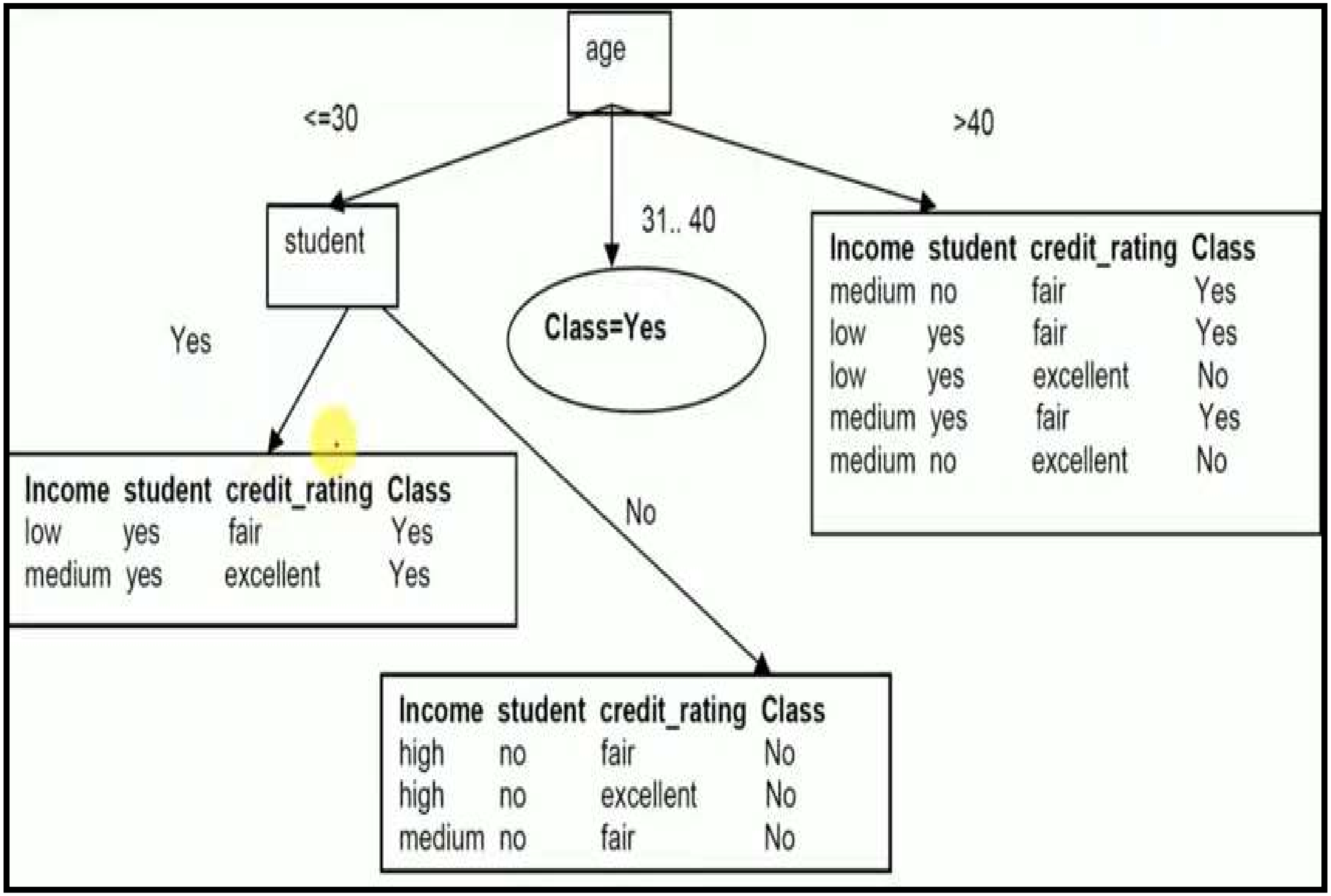
Information Gain:

Income = 0.57

Student = 0.97

Credit_Rating = 0.0192

- SO, here Student is having Maximum Information Gain . So that Student is the <u>Internal Node.</u>

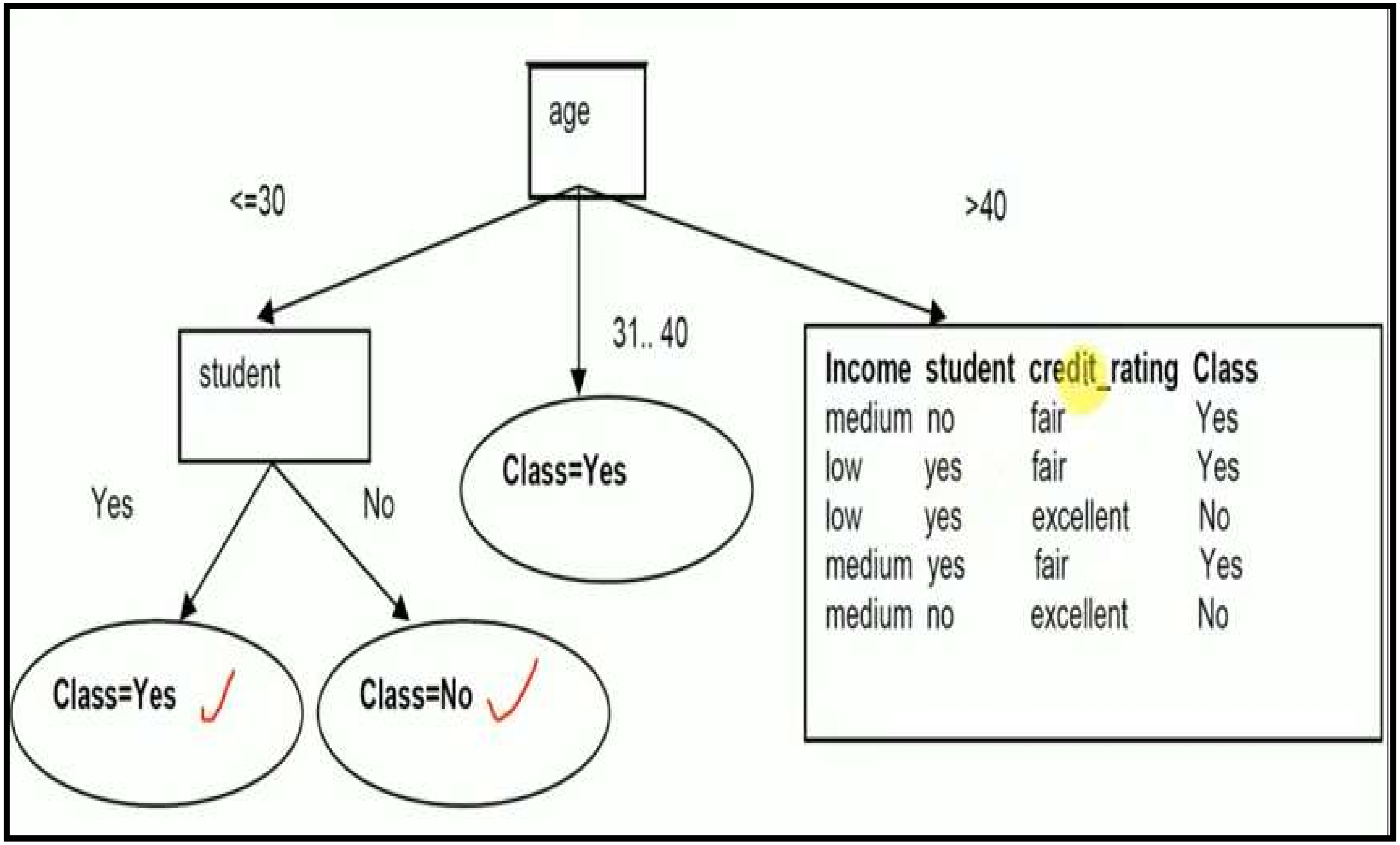


Income	student	credit_rating	Class
low	yes	fair	Yes
medium	yes	excellent	Yes

Income	student	credit_rating	Class
high	no	fair	No
high	no	excellent	No
medium	no	fair	No

Income	student	credit_rating	Class
medium	no	fair	Yes
low	yes	fair	Yes
low	yes	excellent	No
medium	yes	fair	Yes
medium	no	excellent	No

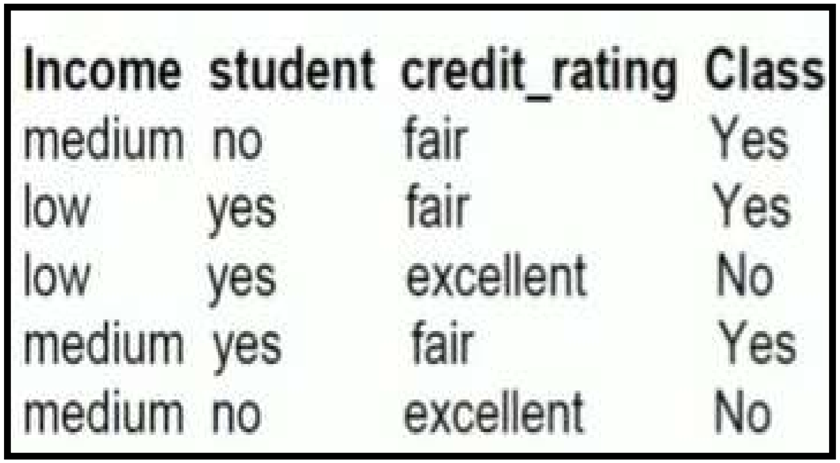
- So, that the Left side Internal Node is **Student**



Income	student	credit_rating	Class
medium	no	fair	Yes
low	yes	fair	Yes
low	yes	excellent	No
medium	yes	fair	Yes
medium	no	excellent	No

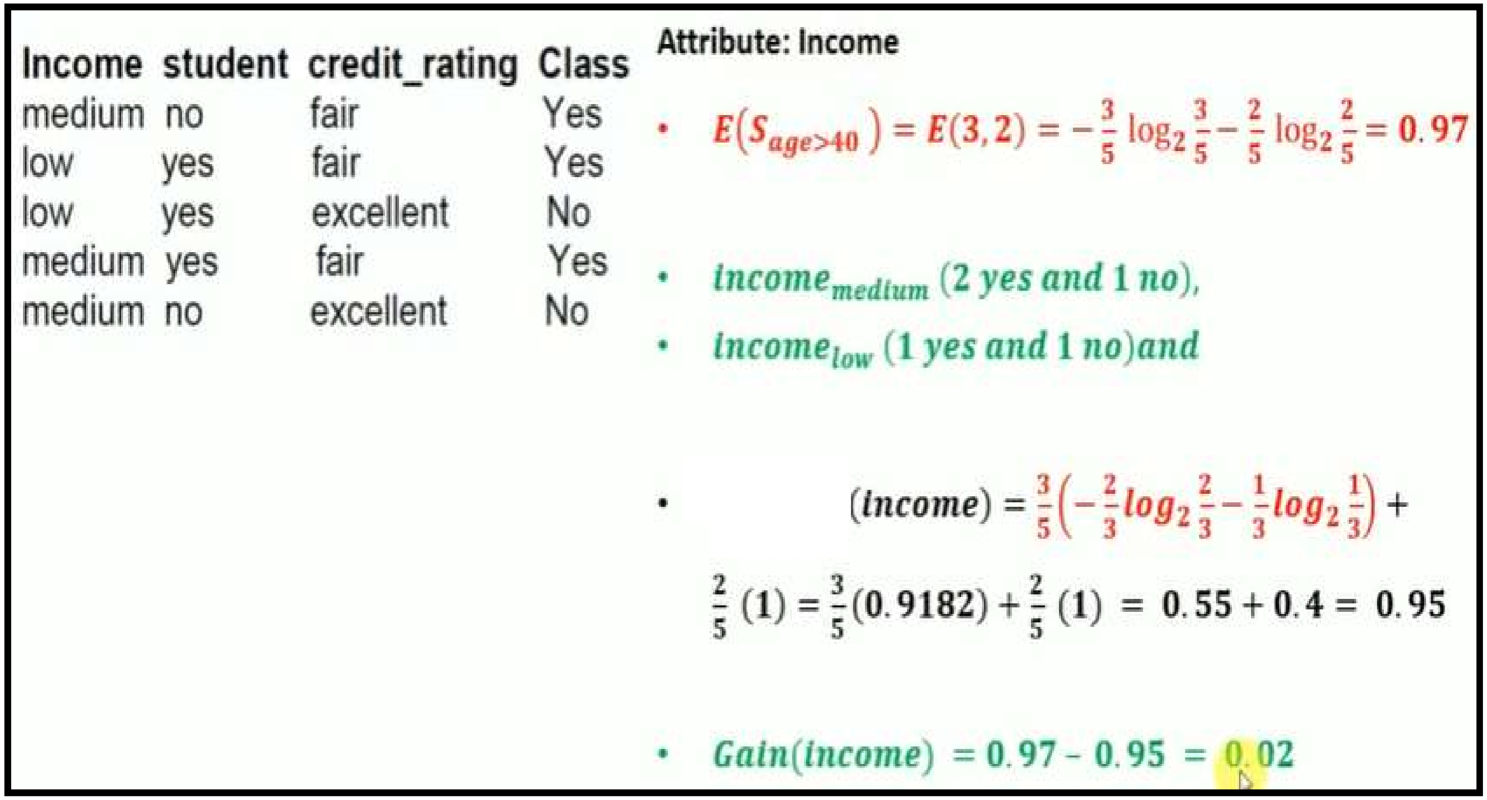
- Next, we need to take the Right Hand Side **“ >40”** attributes to calculate the **Internal Nodes**.
- **In this we have 3 Attributes=**

  {Income, Student, Credit Rating}



Income	student	credit_rating	Class
medium	no	fair	Yes
low	yes	fair	Yes
low	yes	excellent	No
medium	yes	fair	Yes
medium	no	excellent	No

- Again the same procedure, First We can calculate the Entropy and information gain of each attribute. So, here the First Attribute is " INCOME"



Income	student	credit_rating	Class
medium	no	fair	Yes
low	yes	fair	Yes
low	yes	excellent	No
medium	yes	fair	Yes
medium	no	excellent	No

Attribute: Income

- $E(S_{age>40}) = E(3,2) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.97$

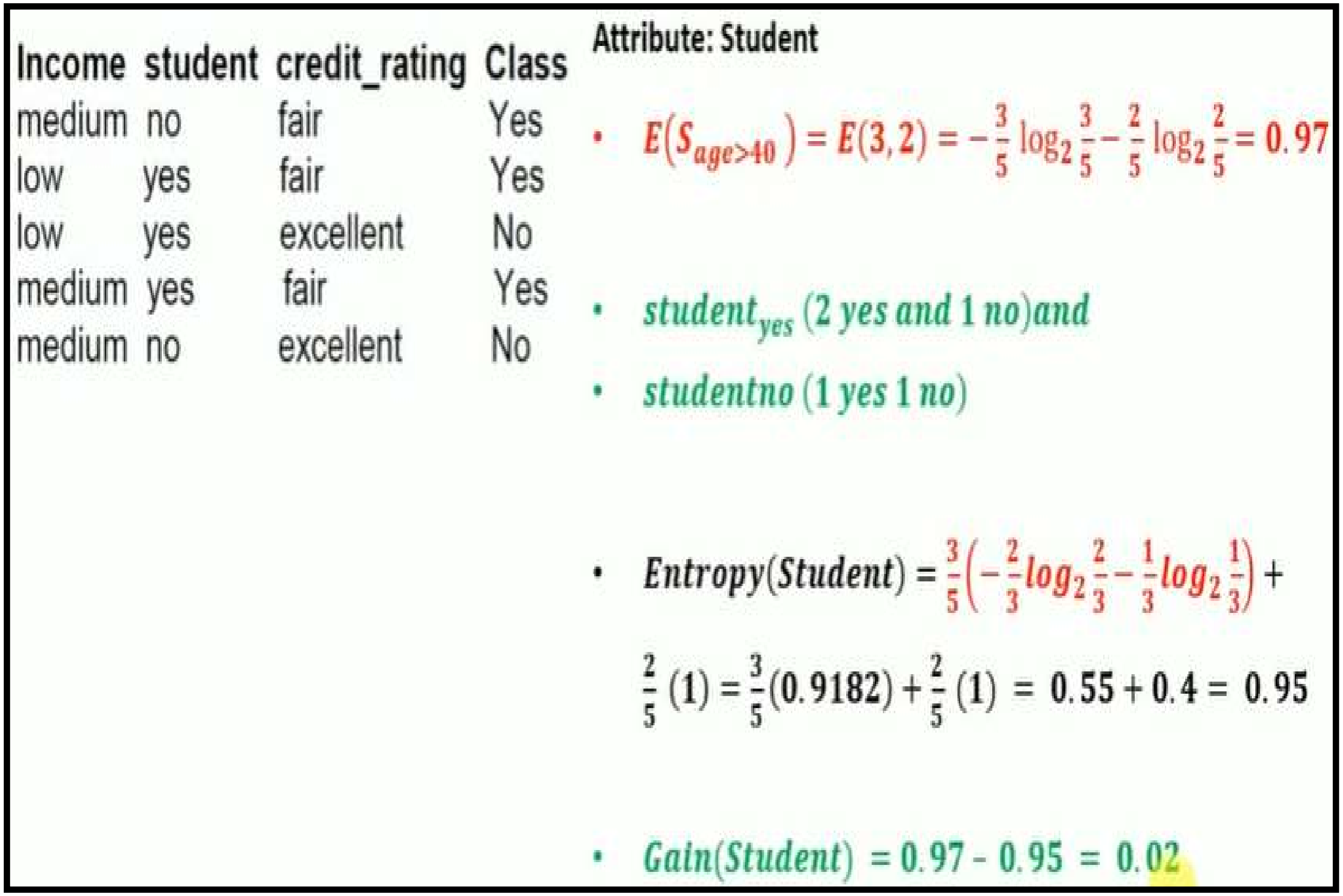
- $income_{medium}$ $(2\ yes\ and\ 1\ no)$,
- $income_{low}$ $(1\ yes\ and\ 1\ no) and$

- $(income) = \frac{3}{5}\left(-\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3}\right) +$

$\frac{2}{5}(1) = \frac{3}{5}(0.9182) + \frac{2}{5}(1) = 0.55 + 0.4 = 0.95$

- $Gain(income) = 0.97 - 0.95 = 0.02$

- Next attribute is **{Student}**

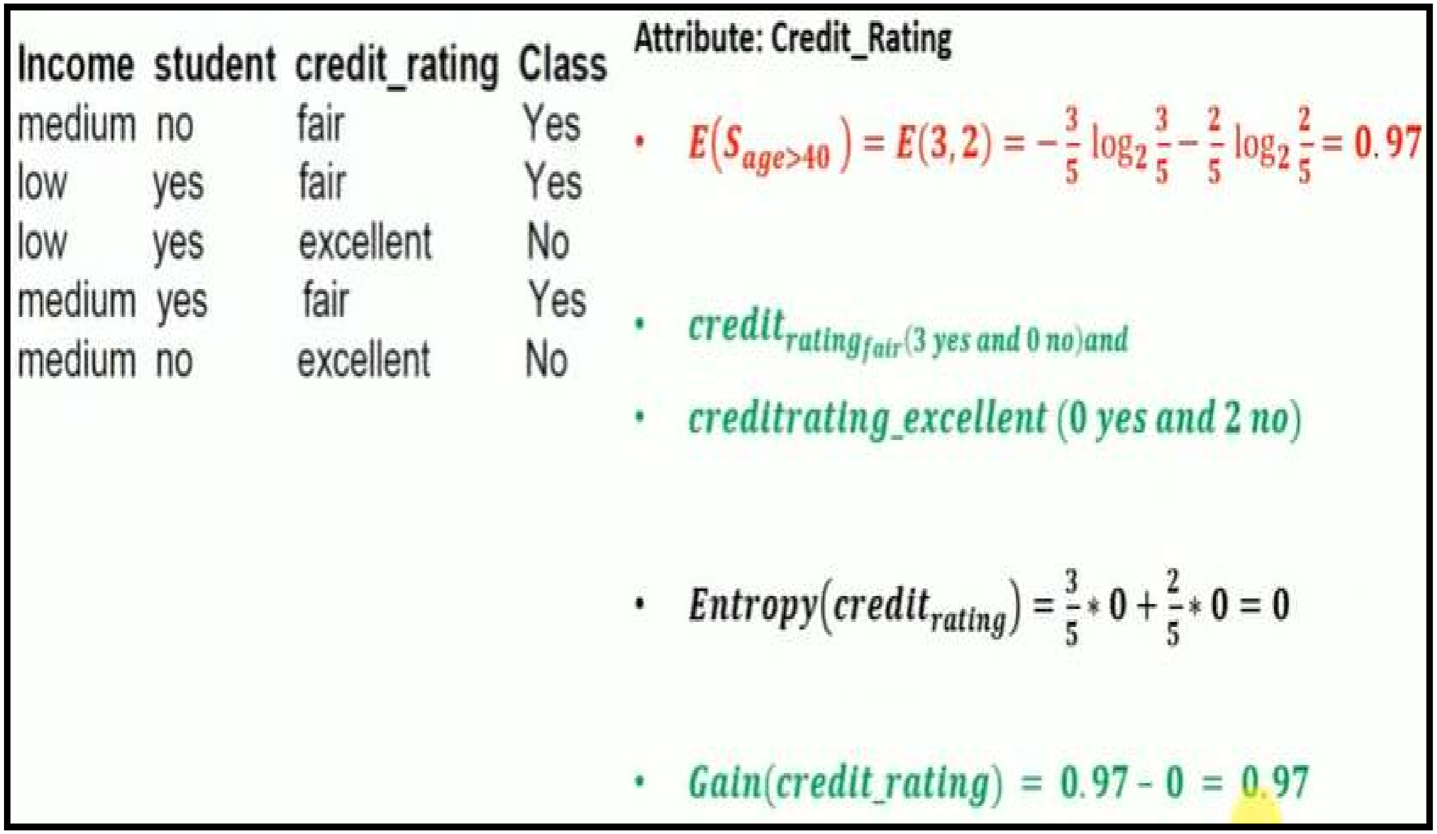


Income	student	credit_rating	Class
medium	no	fair	Yes
low	yes	fair	Yes
low	yes	excellent	No
medium	yes	fair	Yes
medium	no	excellent	No

Attribute: Student

- $E(S_{age>40}) = E(3,2) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.97$
- $student_{yes}\ (2\ yes\ and\ 1\ no) and$
- $studentno\ (1\ yes\ 1\ no)$
- $Entropy(Student) = \frac{3}{5}\left(-\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3}\right) + \frac{2}{5}(1) = \frac{3}{5}(0.9182) + \frac{2}{5}(1) = 0.55 + 0.4 = 0.95$
- $Gain(Student) = 0.97 - 0.95 = 0.02$

- Next attribute is **{Credit_Rating}**

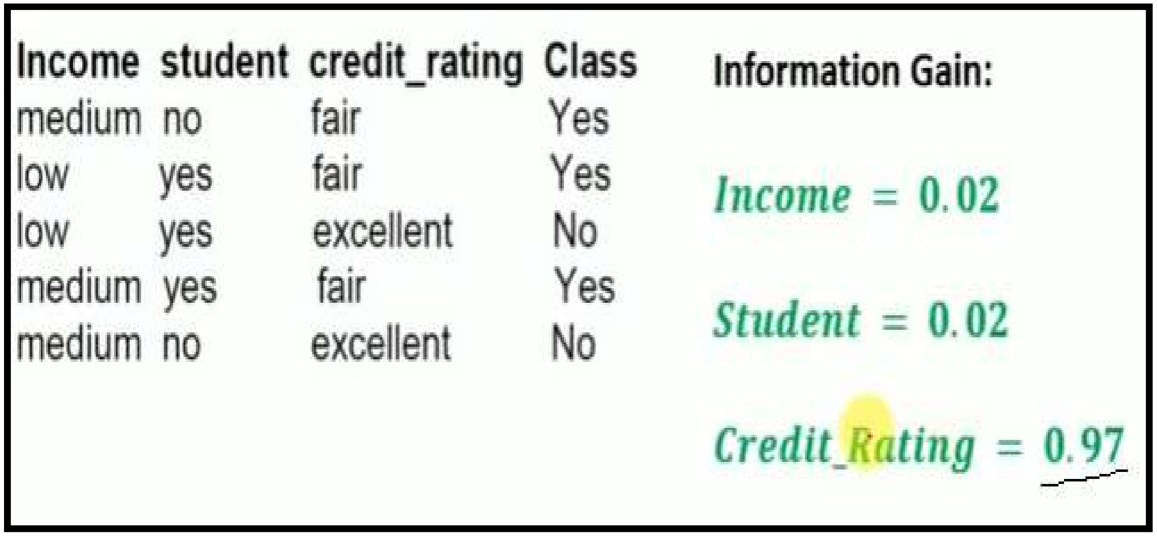


Income	student	credit_rating	Class
medium	no	fair	Yes
low	yes	fair	Yes
low	yes	excellent	No
medium	yes	fair	Yes
medium	no	excellent	No

Attribute: Credit_Rating

- $E(S_{age>40}) = E(3,2) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.97$
- $credit_{rating_{fair}}(3\ yes\ and\ 0\ no)and$
- $creditrating\_excellent\ (0\ yes\ and\ 2\ no)$
- $Entropy(credit_{rating}) = \frac{3}{5} * 0 + \frac{2}{5} * 0 = 0$
- $Gain(credit\_rating)\ =\ 0.97 - 0\ =\ 0.97$

- So, we calculated all the Attributes. Next we need to check the Maximum Information Gain of these attributes.



Income	student	credit_rating	Class
medium	no	fair	Yes
low	yes	fair	Yes
low	yes	excellent	No
medium	yes	fair	Yes
medium	no	excellent	No

Information Gain:

$Income = 0.02$

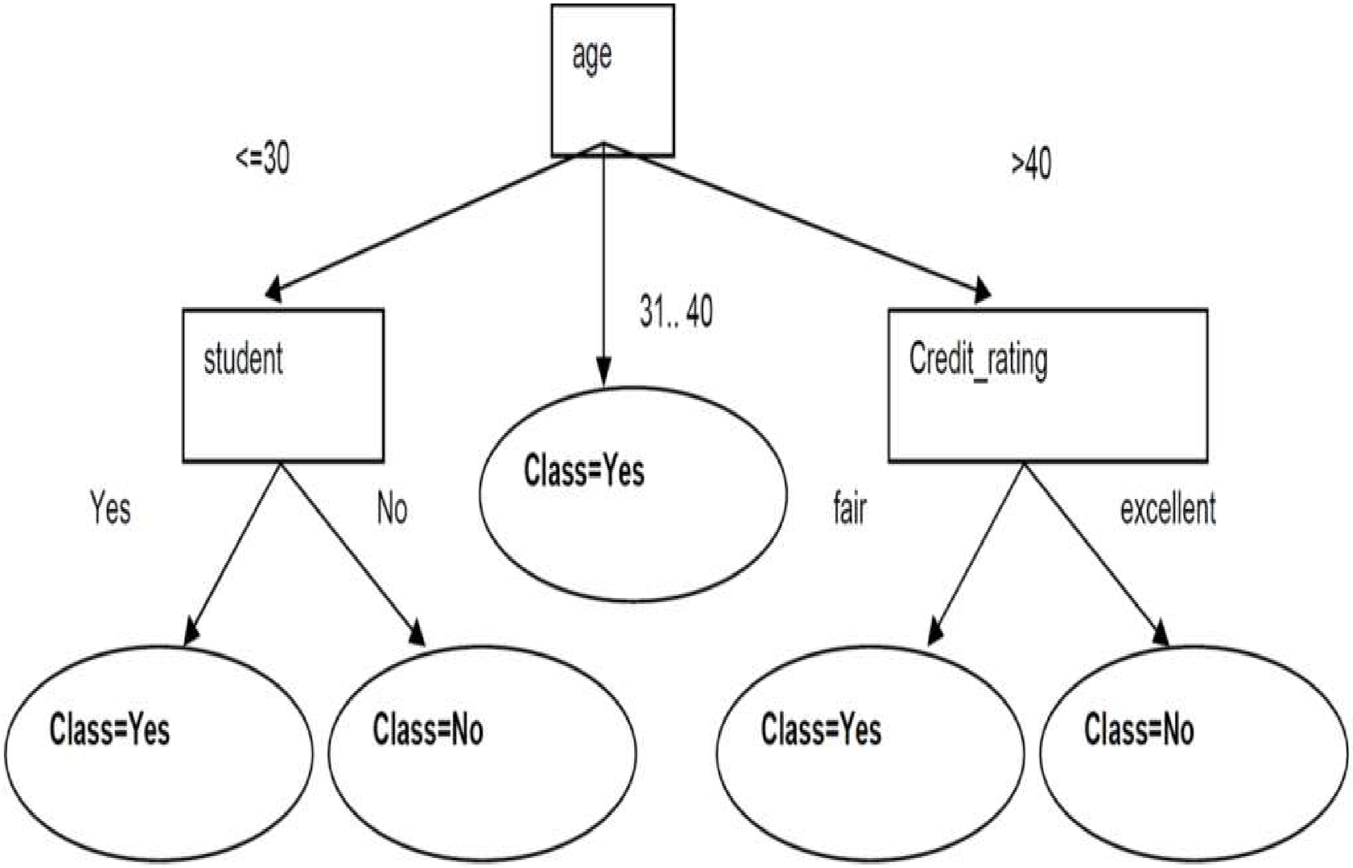
$Student = 0.02$

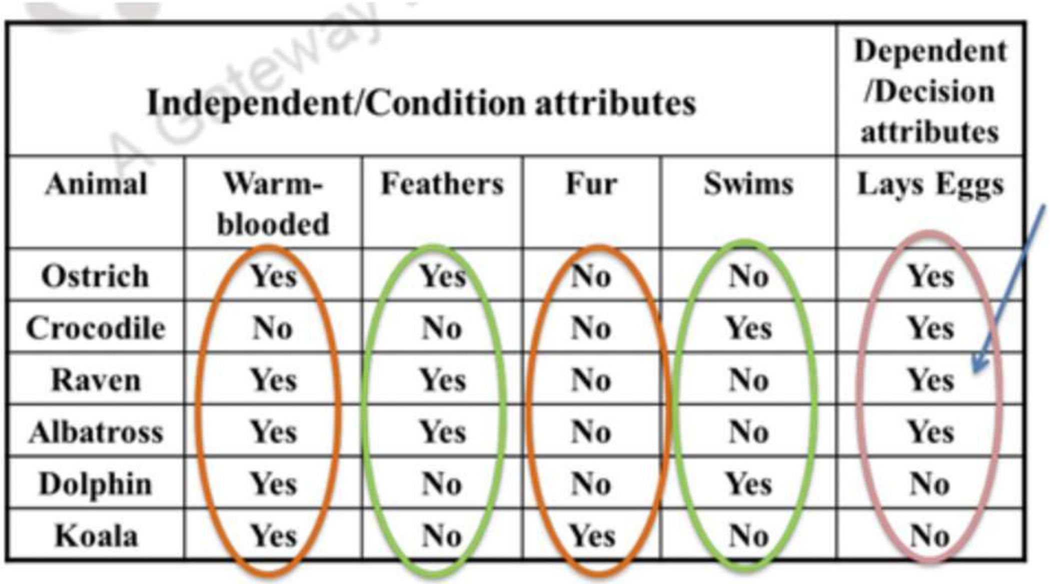
$Credit\_Rating = 0.97$

- SO, here Credit_Rating is having Maximum Information Gain . So that Credit_Rating is the Internal Node.

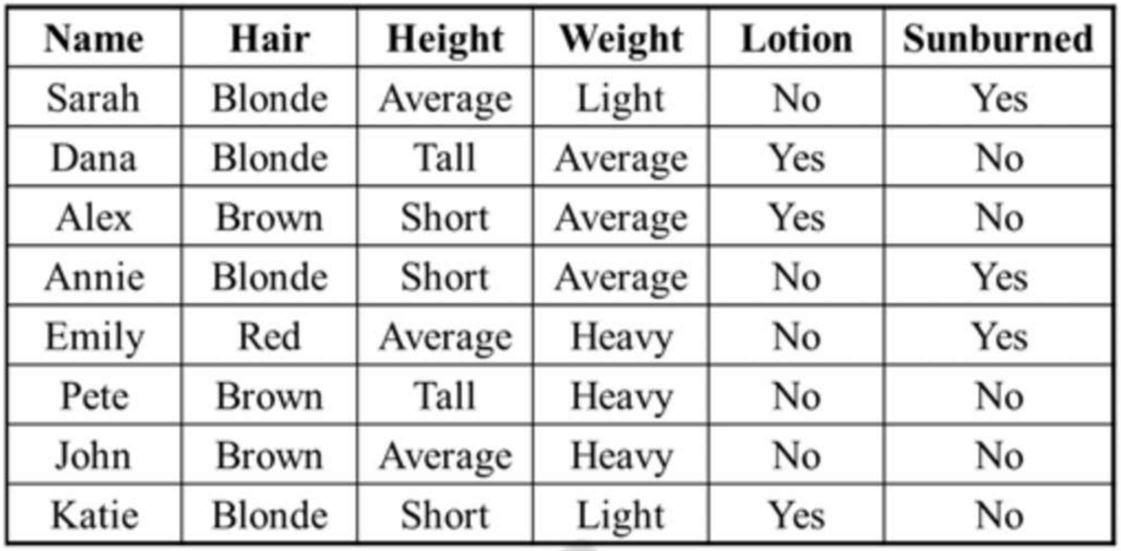
- So, that the Right side Internal Node is Credit_Rating

and the Final Tree is :

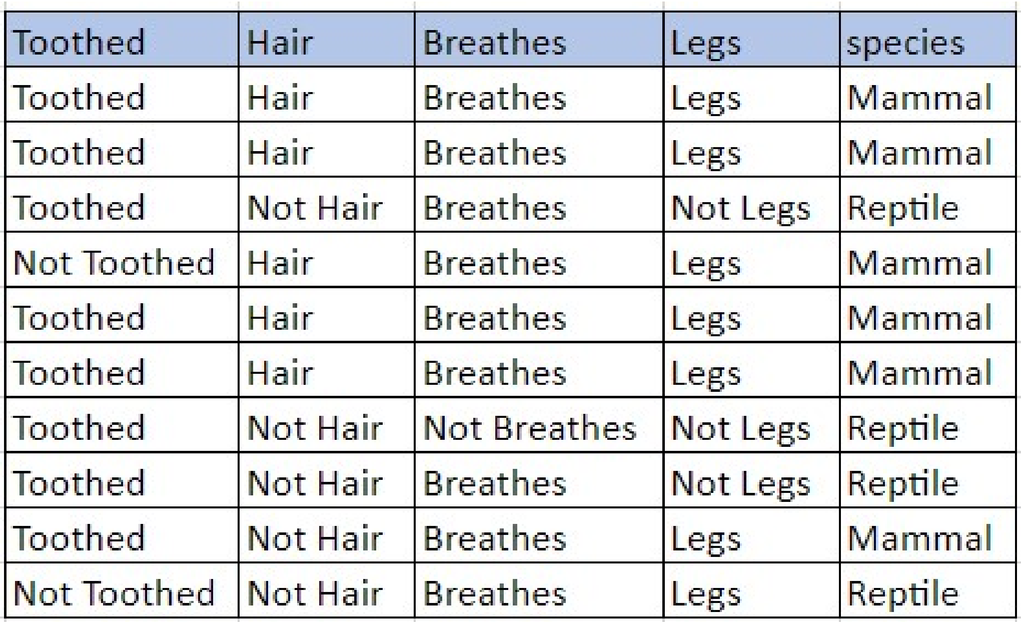




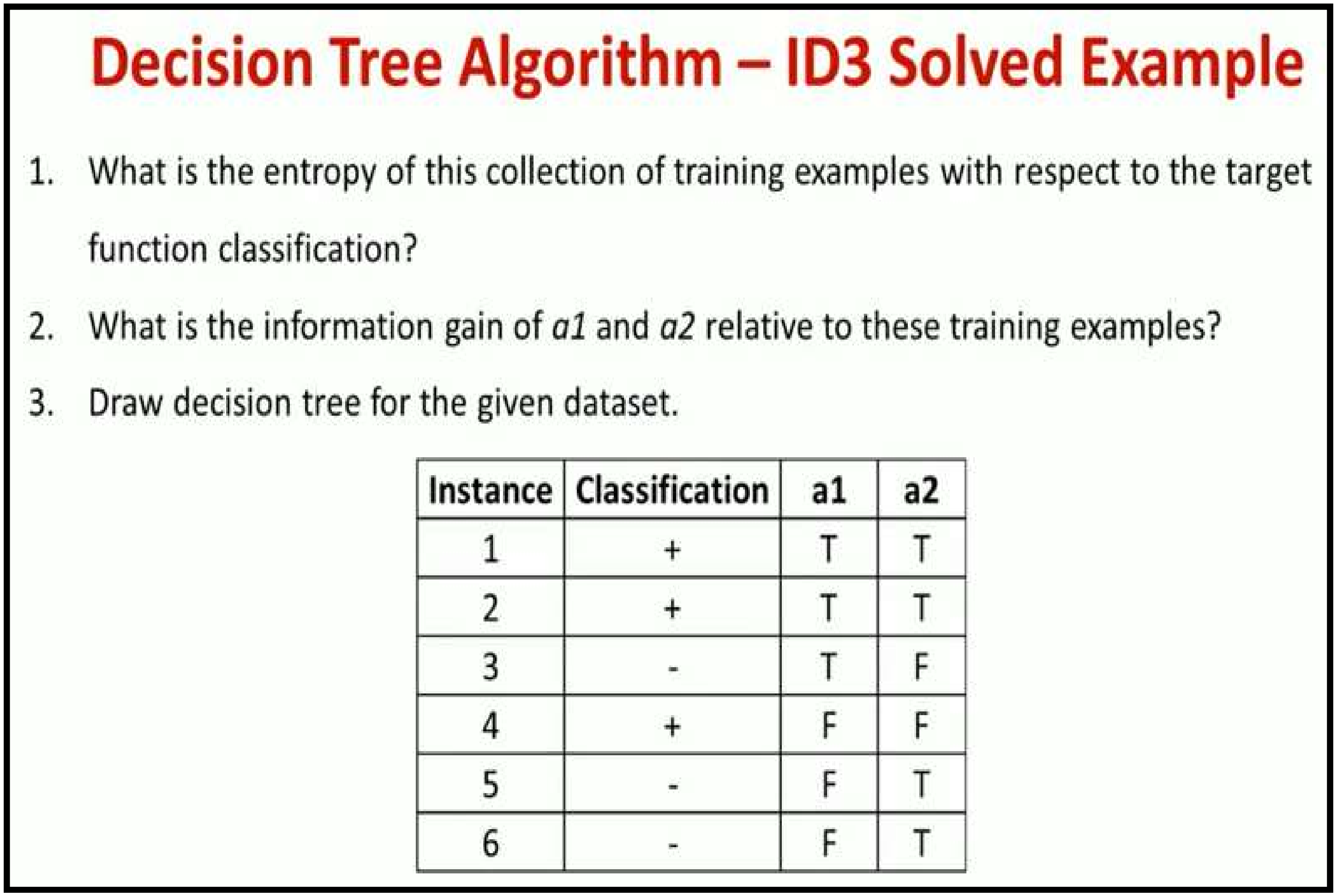
Independent/Condition attributes					Dependent /Decision attributes
Animal	Warm-blooded	Feathers	Fur	Swims	Lays Eggs
Ostrich	Yes	Yes	No	No	Yes
Crocodile	No	No	No	Yes	Yes
Raven	Yes	Yes	No	No	Yes
Albatross	Yes	Yes	No	No	Yes
Dolphin	Yes	No	No	Yes	No
Koala	Yes	No	Yes	No	No



Name	Hair	Height	Weight	Lotion	Sunburned
Sarah	Blonde	Average	Light	No	Yes
Dana	Blonde	Tall	Average	Yes	No
Alex	Brown	Short	Average	Yes	No
Annie	Blonde	Short	Average	No	Yes
Emily	Red	Average	Heavy	No	Yes
Pete	Brown	Tall	Heavy	No	No
John	Brown	Average	Heavy	No	No
Katie	Blonde	Short	Light	Yes	No



Toothed	Hair	Breathes	Legs	species
Toothed	Hair	Breathes	Legs	Mammal
Toothed	Hair	Breathes	Legs	Mammal
Toothed	Not Hair	Breathes	Not Legs	Reptile
Not Toothed	Hair	Breathes	Legs	Mammal
Toothed	Hair	Breathes	Legs	Mammal
Toothed	Hair	Breathes	Legs	Mammal
Toothed	Not Hair	Not Breathes	Not Legs	Reptile
Toothed	Not Hair	Breathes	Not Legs	Reptile
Toothed	Not Hair	Breathes	Legs	Mammal
Not Toothed	Not Hair	Breathes	Legs	Reptile



Decision Tree Algorithm – ID3 Solved Example

1. What is the entropy of this collection of training examples with respect to the target function classification?
2. What is the information gain of *a1* and *a2* relative to these training examples?
3. Draw decision tree for the given dataset.

Instance	Classification	a1	a2
1	+	T	T
2	+	T	T
3	-	T	F
4	+	F	F
5	-	F	T
6	-	F	T

Appropriate problems for decision tree learning

Decision tree learning is generally best suited to problems with the following characteristics:

- ➢ 1. Instances are represented by attribute-value pairs.
- ➢ 2. The target function has discrete output values.
- ➢ 3. Disjunctive descriptions may be required.
- ➢ 4. The training data may contain errors.
- ➢ 5. The training data may contain missing attribute values.

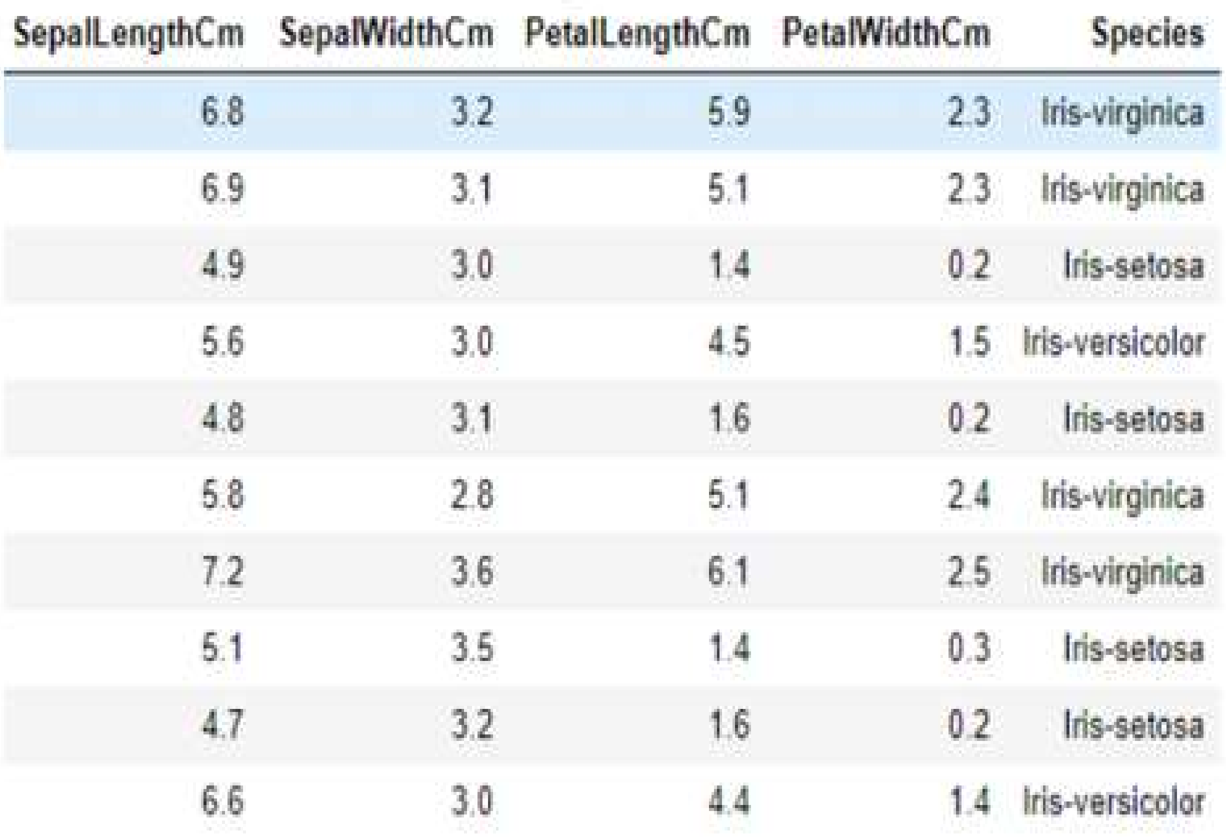
1. Instances are represented by attribute-value pairs: Instances are described by a fixed set of attributes (e.g., Temperature) and their values (e.g., Hot).

- The easiest situation for decision tree learning is when each attribute takes on a small number of disjoint possible values (e.g., Hot, Mild, Cold)
- _So, here we have taken Attribute-value Pair
- Eg: 1. Temperature – {Hot, Mild, Cold}
  - So, here Temperature is an Attribute and {Hot, Mild, Cold} are values.
- Eg: 2. Income – {High, Medium, Low}

- Suppose we have an Attribute which is continuous valued attribute, we can not use Decision Tree.
- First, we need to convert the Continuous valued attribute into Discrete Valued Attribute. Then we can use the Decision Tree.
- There are several approaches to transform continuous variables into discrete ones. This process is also known as binning.

2. The target function has discrete output values: The decision tree is usually used for assigns a Boolean classification (e.g., yes or no) to each example.

- Eg: Particular person is having disease or not disease etc.
- Some times the classification is having more than two classes in our dependent or target variable. We call it as “ Multi Class Classification”.
- Eg: Identify the Music etc.



SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
6.8	3.2	5.9	2.3	Iris-virginica
6.9	3.1	5.1	2.3	Iris-virginica
4.9	3.0	1.4	0.2	Iris-setosa
5.6	3.0	4.5	1.5	Iris-versicolor
4.8	3.1	1.6	0.2	Iris-setosa
5.8	2.8	5.1	2.4	Iris-virginica
7.2	3.6	6.1	2.5	Iris-virginica
5.1	3.5	1.4	0.3	Iris-setosa
4.7	3.2	1.6	0.2	Iris-setosa
6.6	3.0	4.4	1.4	Iris-versicolor

Fig.1: Iris dataset having three categories

- **3. Disjunctive descriptions may be required:** Decision Trees naturally represent **Disjunctive Expressions.**

- The tree might have branches like:
  - If age < 30 and income > $50,000, then predict "Purchase."
  - If age >= 30 and browsing time > 10 minutes, then predict "Purchase."
  - If age >= 30 and browsing time <= 10 minutes, then predict "No Purchase."
- Each of these branches represents a disjunctive condition leading to a different prediction.

Disjunctive Descriptions:

•Disjunctive means "involving a choice between two or more possibilities."

•In the context of decision trees, disjunctive descriptions refer to the fact that the decision tree naturally represents multiple conditions or rules that lead to different outcomes.

•Decision Trees inherently provide disjunctive expressions because they consider multiple conditions along different paths in the tree.

•At each internal node, a decision is made based on a specific feature, leading to different branches.

•The conjunction of conditions along a path represents a specific rule or decision, and the disjunction of these rules covers the entire decision space.

4. The Training Data may contain Errors: Decision tree learning methods are robust to errors.

- Suppose if the given data contain some sort of errors, then also we can use the "Decision Tree".
- Here we can consider two types of errors. i.e, Both errors in classifications of the training examples and errors in the attribute values that describe these examples.

5. The Training Data may contain Missing attribute values: –

- Decision tree methods can be used even when some training examples have unknown values.
- For example, In a Medical domain in which we wish to predict patient outcome based on various laboratory tests, it may be that the lab test, Blood-Test-Result is available only for a subset of the patients.
- In such cases, it is common to estimate the missing attribute value based on other examples for which this attribute has a known value.

Overfitting in Decision Tree Models

- The effectiveness of a machine learning model is measured by its ability to make accurate predictions and minimize prediction errors. An ideal or good machine learning model should be able to perform well with new input data, allowing us to make accurate predictions about future data that the model has not seen before. This ability to work well with future data (unseen data) is known as generalization.
- Overfitting in decision tree models occurs when the tree becomes too complex and captures noise or random fluctuations in the training data, rather than learning the underlying patterns that generalize well to unseen data.

Reasons for overfitting:

- **Complexity:** Decision trees become overly complex, fitting training data perfectly but struggling to generalize to new data.
- **Memorizing Noise:** It can focus too much on specific data points or noise in the training data, hindering generalization.
- **Overly Specific Rules:** Might create rules that are too specific to the training data, leading to poor performance on new data.
- **Feature Importance Bias:** Certain features may be given too much importance by decision trees, even if they are irrelevant, contributing to overfitting.
- **Sample Bias:** If the training dataset is not representative, decision trees may overfit to the training data’s idiosyncrasies, resulting in poor generalization.
- **Lack of Early Stopping:** Without proper stopping rules, decision trees may grow excessively, perfectly fitting the training data but failing to generalize well.

Strategies to Overcome Overfitting in Decision Tree Models

- **Pruning Techniques**
- Pruning involves removing parts of the decision tree that do not contribute significantly to its predictive power.
- This helps simplify the model and prevent it from memorizing noise in the training data.
- Pruning can be achieved through techniques such as cost-complexity pruning, which iteratively removes nodes with the least impact on performance.

- **Limiting Tree Depth**
- Setting a maximum depth for the decision tree restricts the number of levels or branches it can have.
- This prevents the tree from growing too complex and overfitting to the training data.
- By limiting the depth, the model becomes more generalized and less likely to capture noise or outliers.

- **Minimum Samples per Leaf Node**
- Specifying a minimum number of samples required to create a leaf node ensures that each leaf contains a sufficient amount of data to make meaningful predictions.
- This helps prevent the model from creating overly specific rules that only apply to a few instances in the training data, reducing overfitting.

- **Feature Selection and Engineering**
- Carefully selecting relevant features and excluding irrelevant ones is crucial for building a robust model.
- Feature selection involves choosing the most informative features that contribute to predictive power while discarding redundant or noisy ones.
- Feature engineering, on the other hand, involves transforming or combining features to create new meaningful variables that improve model performance.

- **Ensemble Methods**
- Ensemble methods such as Random Forests and Gradient Boosting combine multiple decision trees to reduce overfitting.
- In Random Forests, each tree is trained on a random subset of the data and features, and predictions are averaged across all trees to improve generalization.
- Gradient Boosting builds trees sequentially, with each tree correcting the errors of the previous ones, leading to a more accurate and robust model.

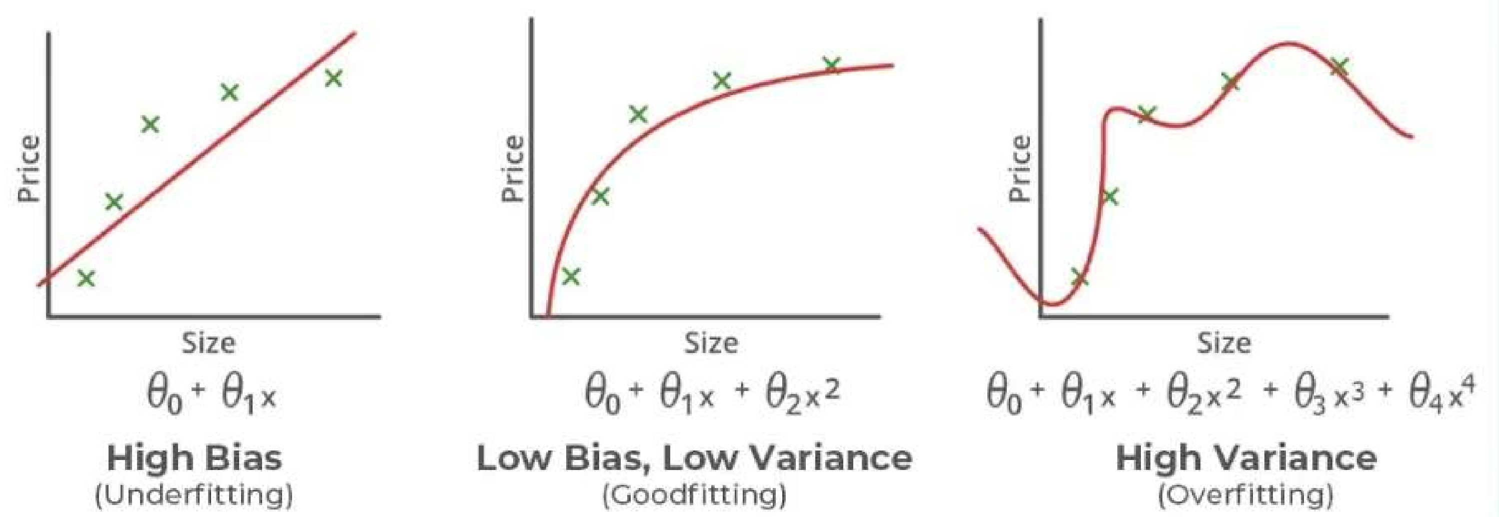
- **Cross-Validation**
- Cross-validation is a technique used to evaluate the performance of a model on multiple subsets of the data.
- By splitting the data into training and validation sets multiple times, training the model on different combinations of data, and evaluating its performance, cross-validation helps ensure that the model generalizes well to unseen data and is not overfitting.

- **Increasing Training Data**
- Providing more diverse and representative training data can help the model learn robust patterns and reduce overfitting.
- Increasing the size of the training dataset allows the model to capture a broader range of patterns and variations in the data, making it less likely to memorize noise or outliers present in smaller datasets.

Bias and Variance in Machine Learning

- **Bias** refers to the error due to overly simplistic assumptions in the learning algorithm. These assumptions make the model easier to comprehend and learn but might not capture the underlying complexities of the data. It is the error due to the model's inability to represent the true relationship between input and output accurately.

- **Variance,** on the other hand, is the error due to the model's sensitivity to fluctuations in the training data. It's the variability of the model's predictions for different instances of training data.



$\theta_0 + \theta_1 x$

High Bias
(Underfitting)

$\theta_0 + \theta_1 x + \theta_2 x^2$

Low Bias, Low Variance
(Goodfitting)

$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High Variance
(Overfitting)

Underfitting in Machine Learning

- A statistical model or a machine learning algorithm is said to have underfitting when a model is too simple to capture data complexities.
- It represents the inability of the model to learn the training data effectively result in poor performance both on the training and testing data.
- In simple terms, an underfit model’s are inaccurate, especially when applied to new, unseen examples.

Reasons for Underfitting

- The model is too simple, So it may be not capable to represent the complexities in the data.
- The input features which is used to train the model is not the adequate representations of underlying factors influencing the target variable.
- The size of the training dataset used is not enough.
- Excessive regularization are used to prevent the overfitting, which constraint the model to capture the data well.
- Features are not scaled.

Techniques to Reduce Underfitting

- Increase model complexity.
- Increase the number of features, performing feature engineering.
- Remove noise from the data.
- Increase the number of epochs or increase the duration of training to get better results.

Feature Engineering

- Feature Engineering is the process of creating new features or transforming existing features to improve the performance of a machine-learning model.
- It involves selecting relevant information from raw data and transforming it into a format that can be easily understood by a model.
- The goal is to improve model accuracy by providing more meaningful and relevant information.

- **Missing Data in Decision Trees**
- Decision trees handle missing data by either ignoring instances with missing values, imputing them using statistical measures, or creating separate branches.
- **Multivalued attributes in decision trees**
- When an attribute has many possible values, the information gain measure gives an inappropriate indication of the attribute’s usefulness.
- **Continuous and integer-valued input attributes**
- Continuous or integer-valued attributes such as Height and Weight, have an infinite set of possible values. Rather than generate infinitely many branches, decision-tree learning algorithms typically find the SPLIT POINT that gives the highest information gain

- **Continuous-valued output attributes**
- If we are trying to predict a numerical output value, such as the price of an apartment, then we need a **regression tree** rather than a classification tree.

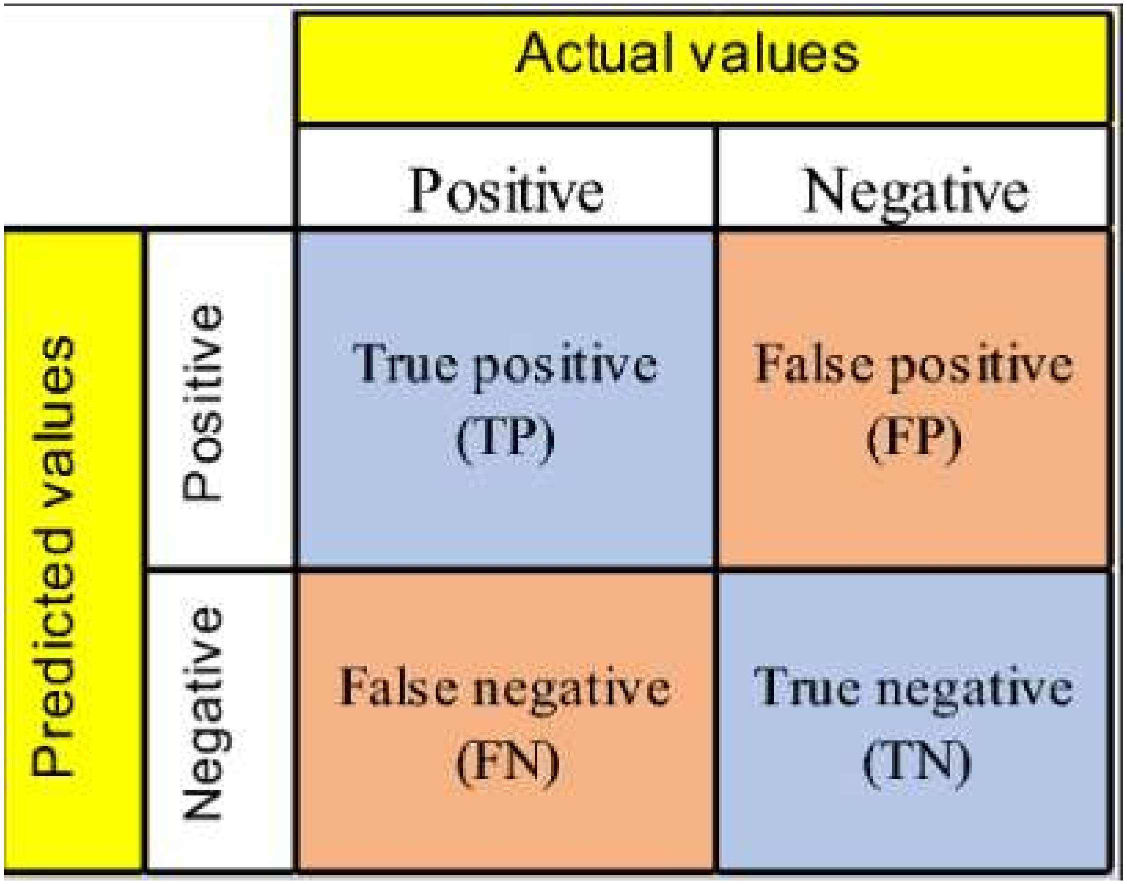
Hypothesis

- A hypothesis is a proposed explanation for a phenomenon that can be tested by observation or experiment.

- The null hypothesis in statistics states that there is no difference between groups or no relationship between variables.

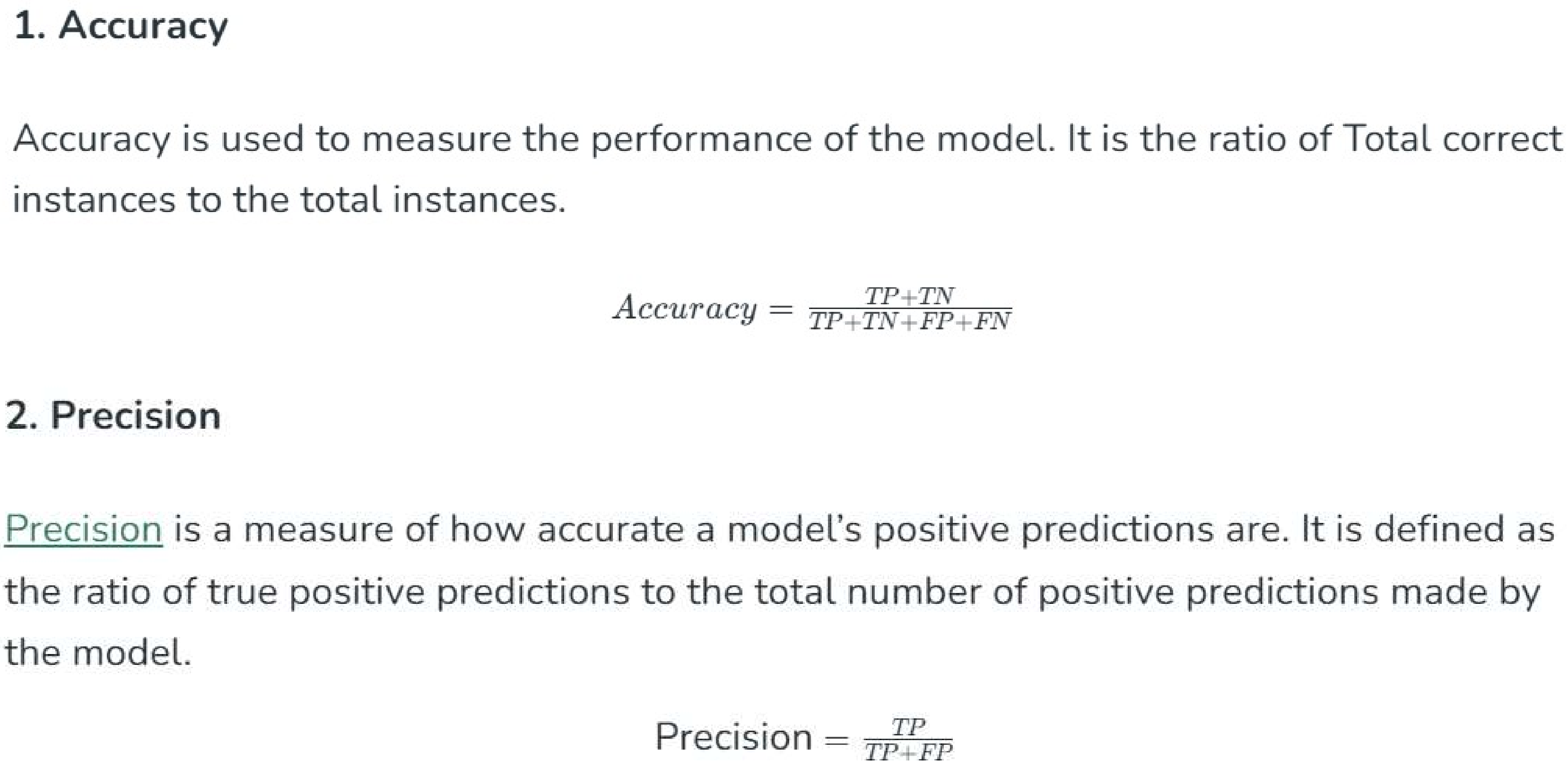
- **EVALUATING AND CHOOSING THE BEST HYPOTHESIS**
- k-fold cross-validation
- The idea is that each example serves double duty—as training K-FOLD CROSS-VALIDATION data and test data.
- First we split the data into k equal subsets.
- We then perform k rounds of learning; on each round 1/k of the data is held out as a test set and the remaining examples are used as training data.
- The average test set score of the k rounds should then be a better estimate than a single score.
- Popular values for k are 5 and 10—enough to give an estimate that is statistically likely to be accurate, at a cost of 5 to 10 times longer computation time.
- The extreme is k = n, also known as **leave-one-out cross-validation or LOOCV.**

Confusion Matrix



		Actual values	
		Positive	Negative
Predicted values	Positive	True positive (TP)	False positive (FP)
	Negative	False negative (FN)	True negative (TN)

Metrics based on Confusion Matrix Data



1. Accuracy

Accuracy is used to measure the performance of the model. It is the ratio of Total correct instances to the total instances.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

2. Precision

Precision is a measure of how accurate a model's positive predictions are. It is defined as the ratio of true positive predictions to the total number of positive predictions made by the model.

$$\text{Precision} = \frac{TP}{TP+FP}$$

3. Recall

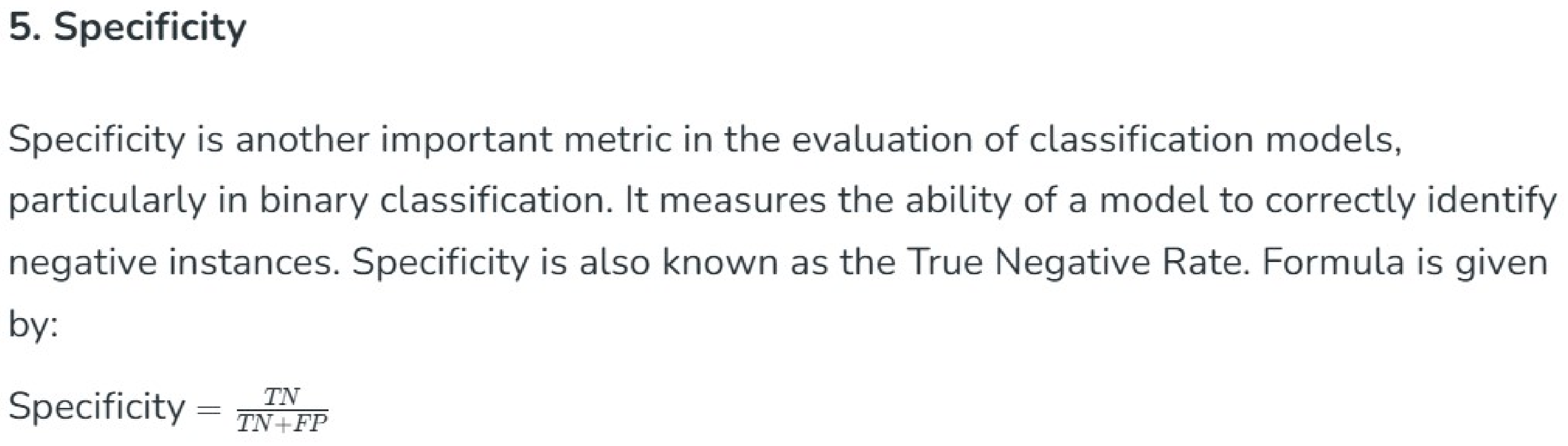
Recall measures the effectiveness of a classification model in identifying all relevant instances from a dataset. It is the ratio of the number of true positive (TP) instances to the sum of true positive and false negative (FN) instances.

$$\text{Recall} = \frac{TP}{TP+FN}$$

4. F1-Score

F1-score is used to evaluate the overall performance of a classification model. It is the harmonic mean of precision and recall,

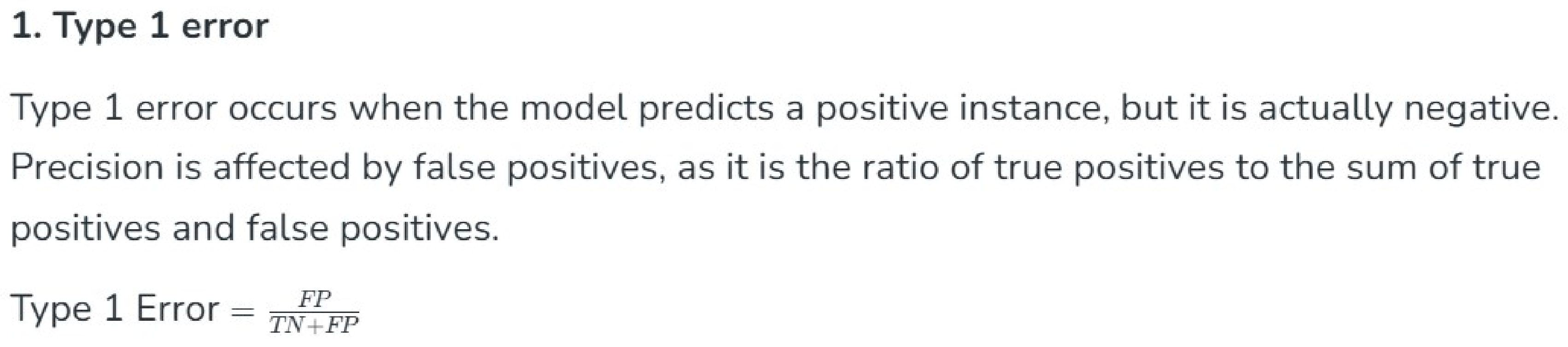
$$\text{F1-Score} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$



5. Specificity

Specificity is another important metric in the evaluation of classification models, particularly in binary classification. It measures the ability of a model to correctly identify negative instances. Specificity is also known as the True Negative Rate. Formula is given by:

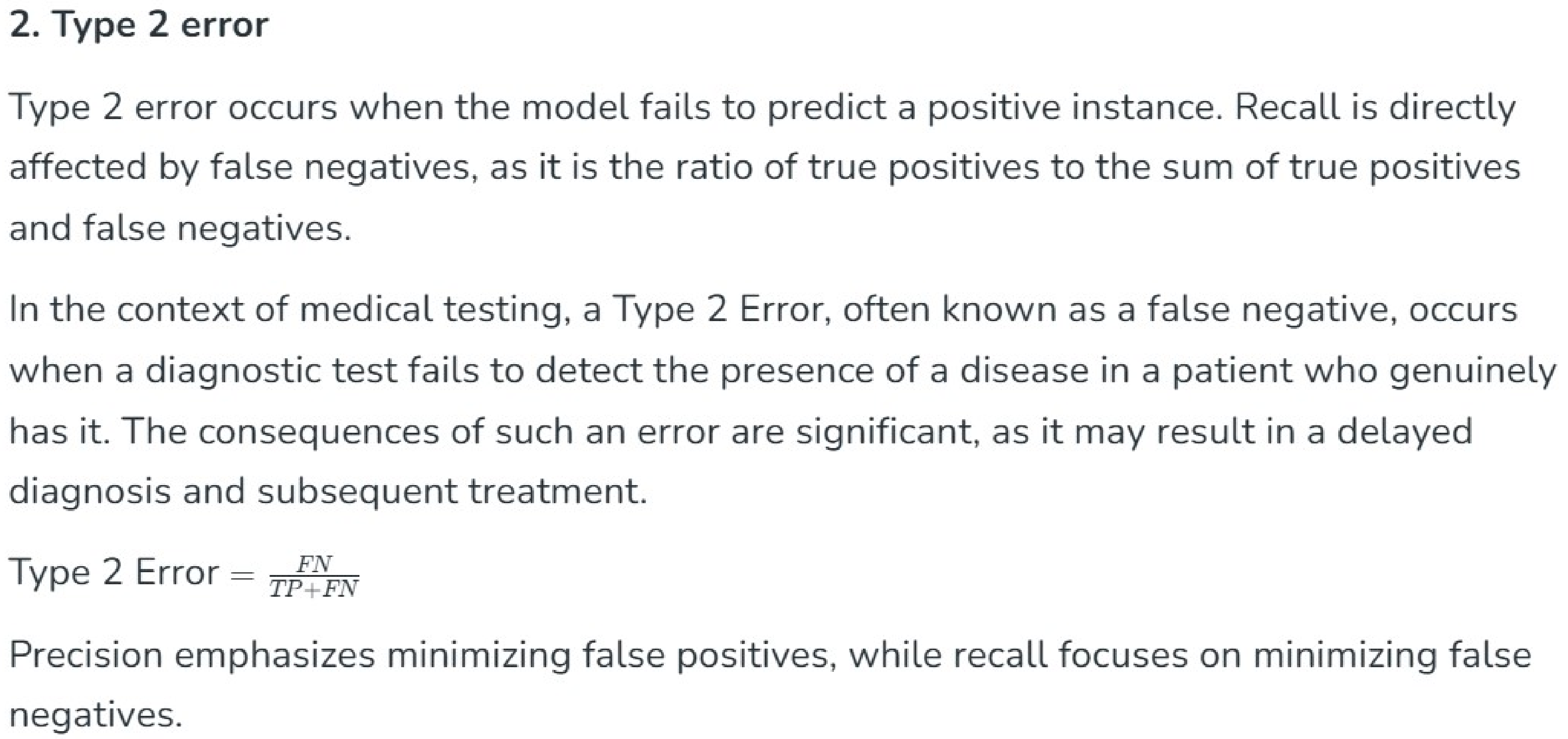
Specificity $= \frac{TN}{TN+FP}$



1. Type 1 error

Type 1 error occurs when the model predicts a positive instance, but it is actually negative. Precision is affected by false positives, as it is the ratio of true positives to the sum of true positives and false positives.

Type 1 Error $= \frac{FP}{TN+FP}$



2. Type 2 error

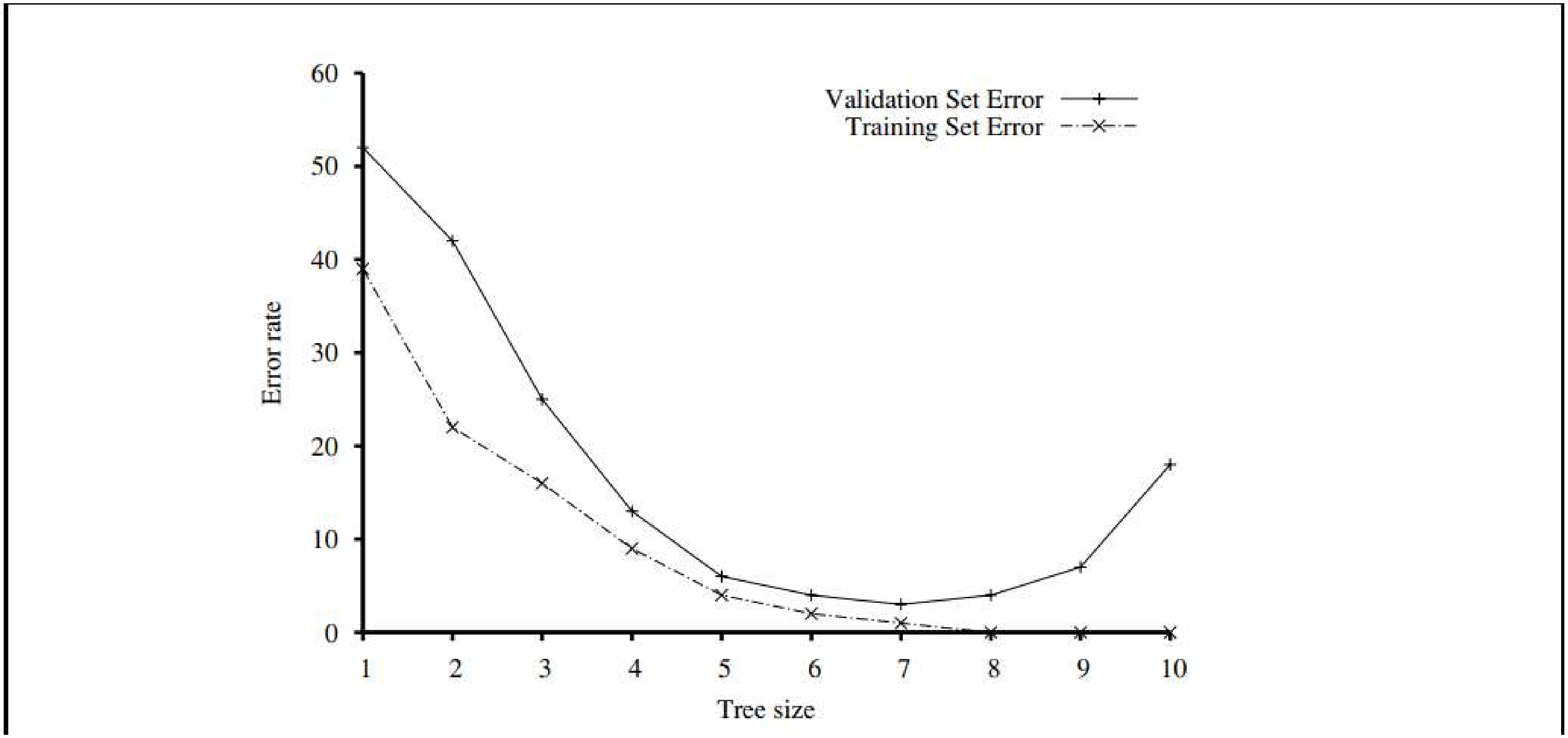
Type 2 error occurs when the model fails to predict a positive instance. Recall is directly affected by false negatives, as it is the ratio of true positives to the sum of true positives and false negatives.

In the context of medical testing, a Type 2 Error, often known as a false negative, occurs when a diagnostic test fails to detect the presence of a disease in a patient who genuinely has it. The consequences of such an error are significant, as it may result in a delayed diagnosis and subsequent treatment.

Type 2 Error $= \frac{FN}{TP+FN}$

Precision emphasizes minimizing false positives, while recall focuses on minimizing false negatives.

Finding the best hypothesis-> Model selection & Optimization



Linear Regression in Machine learning

- Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between the dependent variable and one or more independent features by fitting a linear equation to observed data.

- When there is only one independent feature, it is known as Simple Linear Regression, and when there are more than one feature, it is known as Multiple Linear Regression.

- when there is only one dependent variable, it is considered Univariate Linear Regression, while when there are more than one dependent variables, it is known as Multivariate Regression.

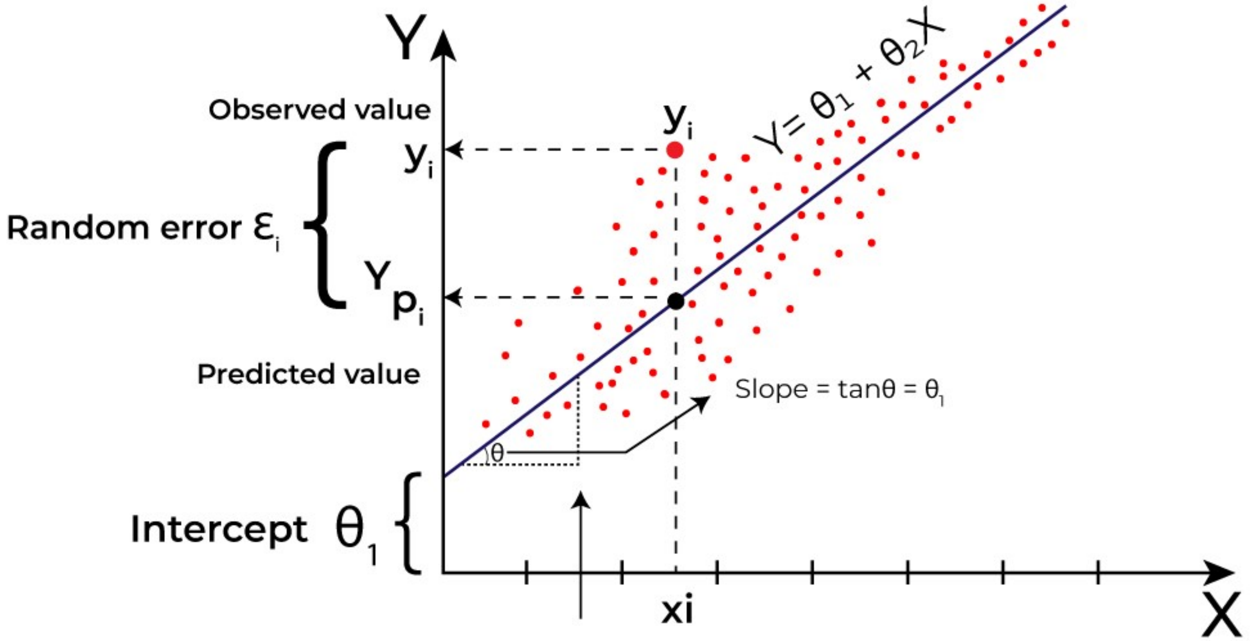
Simple Linear Regression

- This is the simplest form of linear regression, and it involves only one independent variable and one dependent variable. The equation for simple linear regression is:
  $y=\beta0+\beta1X$
  where:
- Y is the dependent variable
- X is the independent variable
- $\beta0$ is the intercept
- $\beta1$ is the slope

Multiple Linear Regression

- This involves more than one independent variable and one dependent variable. The equation for multiple linear regression is: $y=\beta0+\beta1X1+\beta2X2+\ldots\ldots\ldots\beta nXn$
where:
- Y is the dependent variable
- X1, X2, …, Xn are the independent variables
- β0 is the intercept
- β1, β2, …, βn are the slopes

Best Fit Line


Y
Observed value
y_i
Random error ε_i
Y_{p_i}
Predicted value
Intercept θ_1
y_i
$Y = \theta_1 + \theta_2 X$
Slope = tanθ = θ_1
θ
xi
X

- The best Fit Line equation provides a straight line that represents the relationship between the dependent and independent variables. The slope of the line indicates how much the dependent variable changes for a unit change in the independent variable(s).

Hypothesis function in Linear Regression

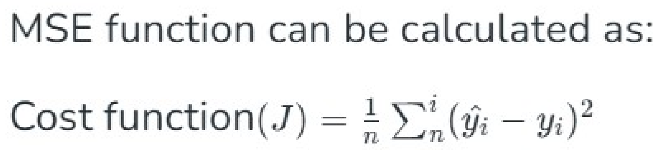
- As we have assumed earlier that our independent feature is the experience i.e X and the respective salary Y is the dependent variable. Let's assume there is a linear relationship between X and Y then the salary can be predicted using:

$$\hat{Y} = \theta_1 + \theta_2 X$$

- Here,
- $y_i \epsilon Y \ \ (i = 1, 2, \cdots, n)$ are labels to data (Supervised learning)
- $x_i \epsilon X \ \ (i = 1, 2, \cdots, n)$ are the input independent training data (univariate – one input variable(parameter))
- $\hat{y}_i \epsilon \hat{Y} \ \ (i = 1, 2, \cdots, n)$ are the predicted values.
- The model gets the best regression fit line by finding the best θ_1 and θ_2 values.
- **θ_1:** intercept
- **θ_2:** coefficient of x
- Once we find the best θ_1 and θ_2 values, we get the best-fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x. $minimize \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$

Cost function for Linear Regression

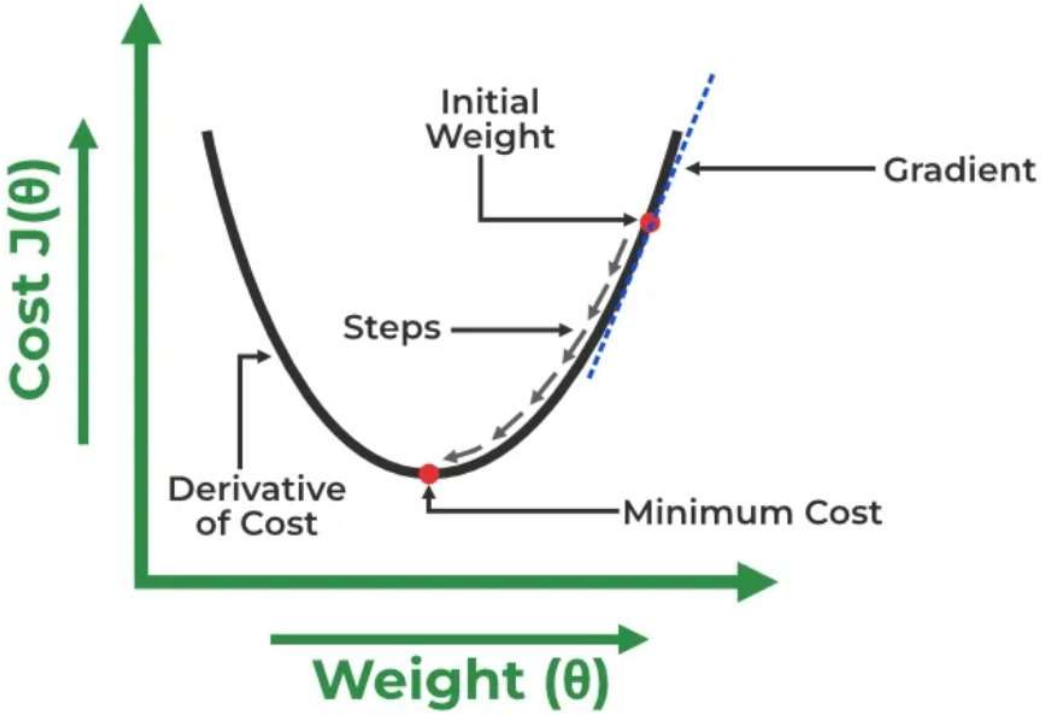
- The cost function or the loss function is nothing but the error or difference between the predicted value Y^ and the true value Y.



MSE function can be calculated as:

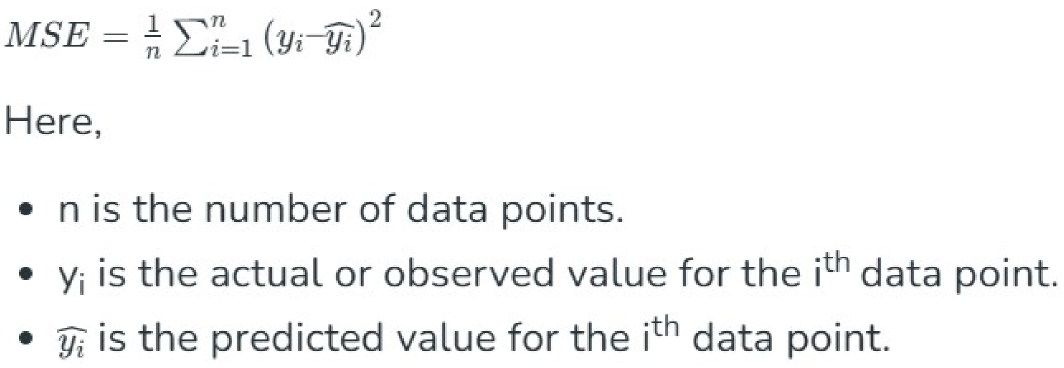
Cost function$(J) = \frac{1}{n}\sum_{n}^{i}(\hat{y}_i - y_i)^2$

Gradient Descent for Linear Regression



Mean Squared Error (MSE) cost function

- Mean Squared Error (MSE) is an evaluation metric that calculates the average of the squared differences between the actual and predicted values for all the data points. The difference is squared to ensure that negative and positive differences don't cancel each other out.



$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2$$

Here,

- n is the number of data points.
- y_i is the actual or observed value for the ith data point.
- $\widehat{y_i}$ is the predicted value for the ith data point.

Probabilistic Models in Machine Learning

- Machine learning algorithms today rely heavily on probabilistic models, which take into consideration the uncertainty inherent in real-world data. **These models make predictions based on probability distributions, rather than absolute values**, allowing for a more nuanced and accurate understanding of complex systems. One common approach is Bayesian inference, where prior knowledge is combined with observed data to make predictions. Another approach is maximum likelihood estimation, which seeks to find the model that best fits observational data.
- Probabilistic models are used in various applications such as image and speech recognition, natural language processing, and recommendation systems.

A **probability distribution** is a mathematical function that describes the likelihood of different outcomes in an experiment. It provides a way to understand how probabilities are distributed over the possible values of a random variable.

Types of Probability Distributions

1. **Discrete Probability Distributions:**
   - **Binomial Distribution**: Describes the number of successes in a fixed number of independent Bernoulli trials (e.g., flipping a coin multiple times).
   - **Poisson Distribution**: Represents the number of events occurring within a fixed interval of time or space (e.g., number of emails received in an hour).
2. **Continuous Probability Distributions**:
   - **Normal Distribution**: Also known as the Gaussian distribution, it is symmetric and describes many natural phenomena (e.g., heights of people).
   - **Exponential Distribution**: Describes the time between events in a Poisson process (e.g., time between arrivals of buses).

Key Functions

- **Probability Mass Function (PMF):** Used for discrete distributions, it gives the probability that a discrete random variable is exactly equal to some value.
- **Probability Density Function (PDF)**: Used for continuous distributions, it describes the likelihood of a random variable to take on a particular value.
- **Cumulative Distribution Function (CDF)**: Gives the probability that a random variable is less than or equal to a certain value.

Examples

- **Binomial Distribution:** If you flip a fair coin 10 times, the probability distribution of the number of heads can be described by a binomial distribution.
- **Normal Distribution**: The distribution of heights in a large population often follows a normal distribution.

Categories Of Probabilistic Models

- **Generative models:**
- Generative models aim to model the joint distribution of the input and output variables. These models generate new data based on the probability distribution of the original dataset. Generative models are powerful because they can generate new data that resembles the training data. They can be used for tasks such as image and speech synthesis, language translation, and text generation.

- **Discriminative models**
- The discriminative model aims to model the conditional distribution of the output variable given the input variable. They learn a decision boundary that separates the different classes of the output variable. Discriminative models are useful when the focus is on making **accurate predictions** rather than generating new data. They can be used for tasks such as image recognition, speech recognition, and sentiment analysis.

- **Graphical models**
- These models use **graphical representations** to show the conditional dependence between variables. They are commonly used for tasks such as image recognition, natural language processing, and causal inference.

Naive Bayes Algorithm in Probabilistic Models

- Collect a labeled dataset of samples, where each sample has a set of features and a class label.
- For each feature in the dataset, calculate the conditional probability of the feature given the class.
- This is done by counting the number of times the feature occurs in samples of the class and dividing by the total number of samples in the class.
- Calculate the prior probability of each class by counting the number of samples in each class and dividing by the total number of samples in the dataset.
- Given a new sample with a set of features, calculate the posterior probability of each class using the Bayes theorem and the conditional probabilities and prior probabilities calculated in steps 2 and 3.
- Select the class with the highest posterior probability as the predicted class for the new sample.

Importance of Probabilistic Models

- Probabilistic models play a crucial role in the field of machine learning, providing a framework for understanding the underlying patterns and complexities in massive datasets.
- Probabilistic models provide a natural way to reason about the likelihood of different outcomes and can help us understand the underlying structure of the data.
- Probabilistic models help enable researchers and practitioners to make informed decisions when faced with uncertainty.
- Probabilistic models allow us to perform Bayesian inference, which is a powerful method for updating our beliefs about a hypothesis based on new data. This can be particularly useful in situations where we need to make decisions under uncertainty.

Advantages Of Probabilistic Models

- Probabilistic models are an increasingly popular method in many fields, including artificial intelligence, finance, and healthcare.
- The main advantage of these models is their ability to take into account uncertainty and variability in data. This allows for more accurate predictions and decision-making, particularly in complex and unpredictable situations.
- Probabilistic models can also provide insights into how different factors influence outcomes and can help identify patterns and relationships within data.

Disadvantages Of Probabilistic Models

- One of the disadvantages is the potential for overfitting, where the model is too specific to the training data and doesn’t perform well on new data.
- Not all data fits well into a probabilistic framework, which can limit the usefulness of these models in certain applications.
- Another challenge is that probabilistic models can be computationally intensive and require significant resources to develop and implement.

Reinforcement learning

- Reinforcement Learning (RL) is a branch of machine learning focused on making decisions to maximize cumulative rewards in a given situation.
- Unlike supervised learning, which relies on a training dataset with predefined answers, RL involves learning through experience.
- In RL, an agent learns to achieve a goal in an uncertain, potentially complex environment by performing actions and receiving feedback through rewards or penalties.

- **Key Concepts of Reinforcement Learning**
- **Agent:** The learner or decision-maker.
- **Environment:** Everything the agent interacts with.
- **State:** A specific situation in which the agent finds itself.
- **Action:** All possible moves the agent can make.
- **Reward:** Feedback from the environment based on the action taken.

- **Elements of Reinforcement Learning**
- **i) Policy:** Defines the agent’s behavior at a given time.
- **ii) Reward Function:** Defines the goal of the RL problem by providing feedback.
- **iii) Value Function:** Estimates long-term rewards from a state.
- **iv) Model of the Environment:** Helps in predicting future states and rewards for planning.

Types of Reinforcement:

- **Positive:** Positive Reinforcement is defined as when an event, occurs due to a particular behavior, increases the strength and the frequency of the behavior. In other words, it has a positive effect on behavior. Advantages of reinforcement learning are:
  - Maximizes Performance
  - Sustain Change for a long period of time
  - Too much Reinforcement can lead to an overload of states which can diminish the results

- **Negative:** Negative Reinforcement is defined as strengthening of behavior because a negative condition is stopped or avoided. Advantages of reinforcement learning:
  - Increases Behavior
  - Provide defiance to a minimum standard of performance
  - It Only provides enough to meet up the minimum behavior