

# Decision Tree Using ID3 Algorithm

Compiled by,  
Dr. Shashank Shetty

# DECISION TREE – ID3 ALGORITHM NUMERICAL 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

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

## Attribute: Outlook

Values (Outlook) = Sunny, Overcast, Rain

$$S = [9+, 5-]$$

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

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

$$Entropy(S_{Sunny}) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.971$$

$$S_{Overcast} \leftarrow [4+, 0-]$$

$$Entropy(S_{Overcast}) = -\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} = 0$$

$$S_{Rain} \leftarrow [3+, 2-]$$

$$Entropy(S_{Rain}) = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} = 0.971$$

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

$$Gain(S, Outlook)$$

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

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

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

## Attribute: Temp

Values (Temp) = Hot, Mild, Cool

$$S = [9+, 5-]$$

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

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

$$Entropy(S_{Hot}) = -\frac{2}{4} \log_2 \frac{2}{4} - \frac{2}{4} \log_2 \frac{2}{4} = 1.0$$

$$S_{Mild} \leftarrow [4+, 2-]$$

$$Entropy(S_{Mild}) = -\frac{4}{6} \log_2 \frac{4}{6} - \frac{2}{6} \log_2 \frac{2}{6} = 0.9183$$

$$S_{Cool} \leftarrow [3+, 1-]$$

$$Entropy(S_{Cool}) = -\frac{3}{4} \log_2 \frac{3}{4} - \frac{1}{4} \log_2 \frac{1}{4} = 0.8113$$

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

$$Gain(S, Temp)$$

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

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

$$Gain(S, Temp) = 0.94 - \frac{4}{14} 1.0 - \frac{6}{14} 0.9183 - \frac{4}{14} 0.8113 = 0.0289$$



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

## Attribute: Humidity

Values (Humidity) = High, Normal

$$S = [9+, 5-]$$

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

$$S_{High} \leftarrow [3+, 4-]$$

$$Entropy(S_{High}) = -\frac{3}{7} \log_2 \frac{3}{7} - \frac{4}{7} \log_2 \frac{4}{7} = 0.9852$$

$$S_{Normal} \leftarrow [6+, 1-]$$

$$Entropy(S_{Normal}) = -\frac{6}{7} \log_2 \frac{6}{7} - \frac{1}{7} \log_2 \frac{1}{7} = 0.5916$$

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

$$Gain(S, Humidity)$$

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

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



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

## Attribute: Wind

Values (Wind) = Strong, Weak

$$S = [9+, 5-]$$

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

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

$$Entropy(S_{Strong}) = 1.0$$

$$S_{Weak} \leftarrow [6+, 2-]$$

$$Entropy(S_{Weak}) = -\frac{6}{8} \log_2 \frac{6}{8} - \frac{2}{8} \log_2 \frac{2}{8} = 0.8113$$

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

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

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

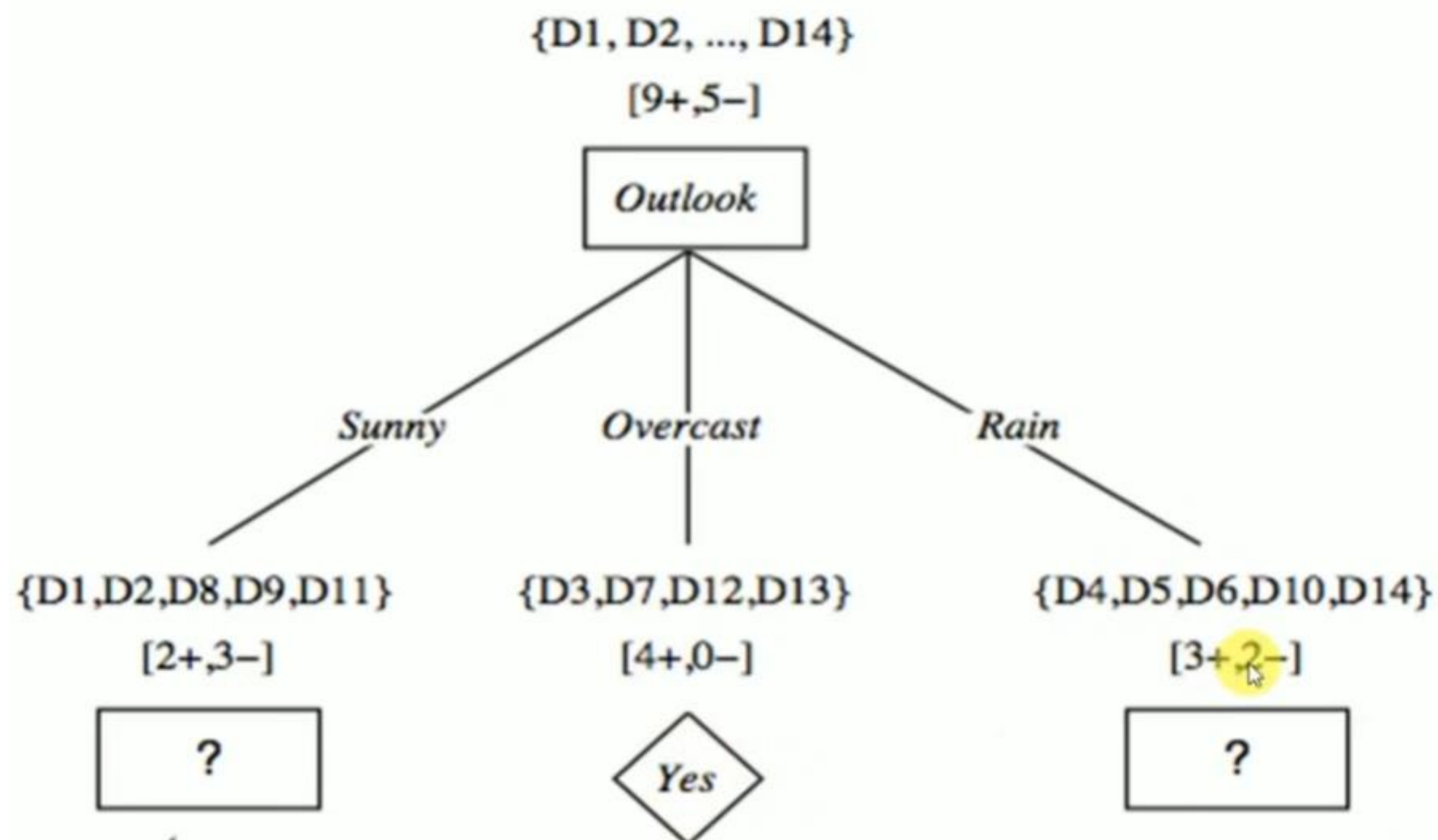
Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak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

$$Gain(S, Outlook) = 0.2464$$

$$Gain(S, Temp) = 0.0289$$

$$Gain(S, Humidity) = 0.1516$$

$$Gain(S, Wind) = 0.0478$$





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

## Attribute: Temp

Values (Temp) = Hot, Mild, Cool

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

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

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

$$Entropy(S_{Hot}) = 0.0$$

$$S_{Mild} \leftarrow [1+, 1-]$$

$$Entropy(S_{Mild}) = 1.0$$

$$S_{Cool} \leftarrow [1+, 0-]$$

$$Entropy(S_{Cool}) = 0.0$$

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

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

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

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

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

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

## Attribute: Humidity

Values (Humidity) = High, Normal

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

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

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

$$Entropy(S_{High}) = 0.0$$

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

$$Entropy(S_{Normal}) = 0.0$$

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

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

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

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

## Attribute: Wind

Values (Wind) = Strong, Weak

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

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

$$S_{Strong} \leftarrow [1+, 1-]$$

$$Entropy(S_{Strong}) = 1.0$$

$$S_{Weak} \leftarrow [1+, 2-]$$

$$Entropy(S_{Weak}) = -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} = 0.9183$$

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

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

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

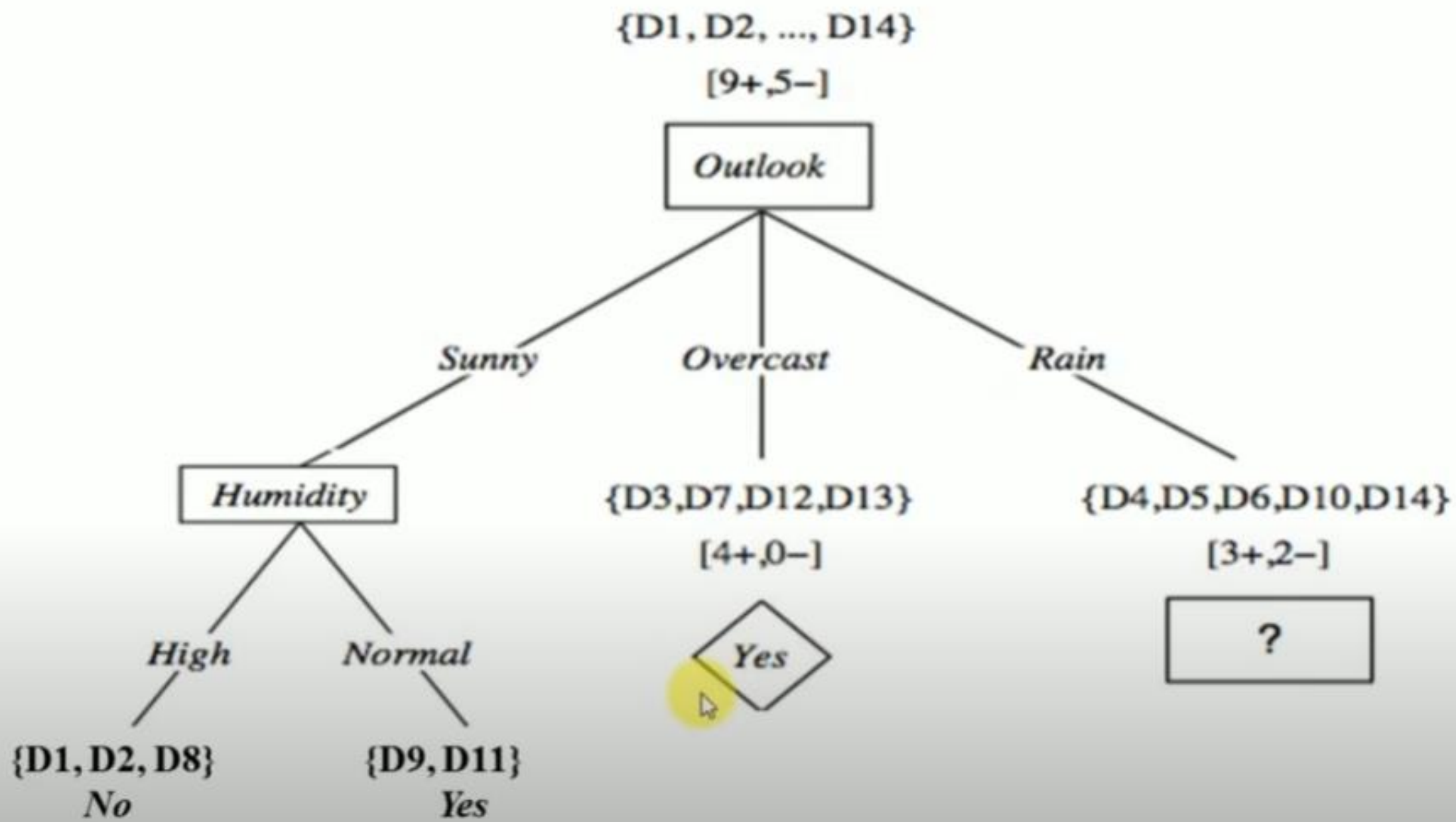
Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

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

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

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





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

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

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

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

$$Entropy(S_{Hot}) = 0.0$$

$$S_{Mild} \leftarrow [2+, 1-]$$

$$Entropy(S_{Mild}) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.9183$$

$$S_{Cool} \leftarrow [1+, 1-]$$

$$Entropy(S_{Cool}) = 1.0$$

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

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

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

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

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

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: Humidity

Values (Humidity) = High, Normal

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

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

$$S_{High} \leftarrow [1+, 1-]$$

$$Entropy(S_{High}) = 1.0$$

$$S_{Normal} \leftarrow [2+, 1-]$$

$$Entropy(S_{Normal}) = -\frac{2}{3} \log_2 \frac{2}{3} - \frac{1}{3} \log_2 \frac{1}{3} = 0.9183$$

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

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

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

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: Wind

Values (wind) = Strong, Weak

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

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

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

$$Entropy(S_{Strong}) = 0.0$$

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

$$Entropy(S_{Weak}) = 0.0$$

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

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

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

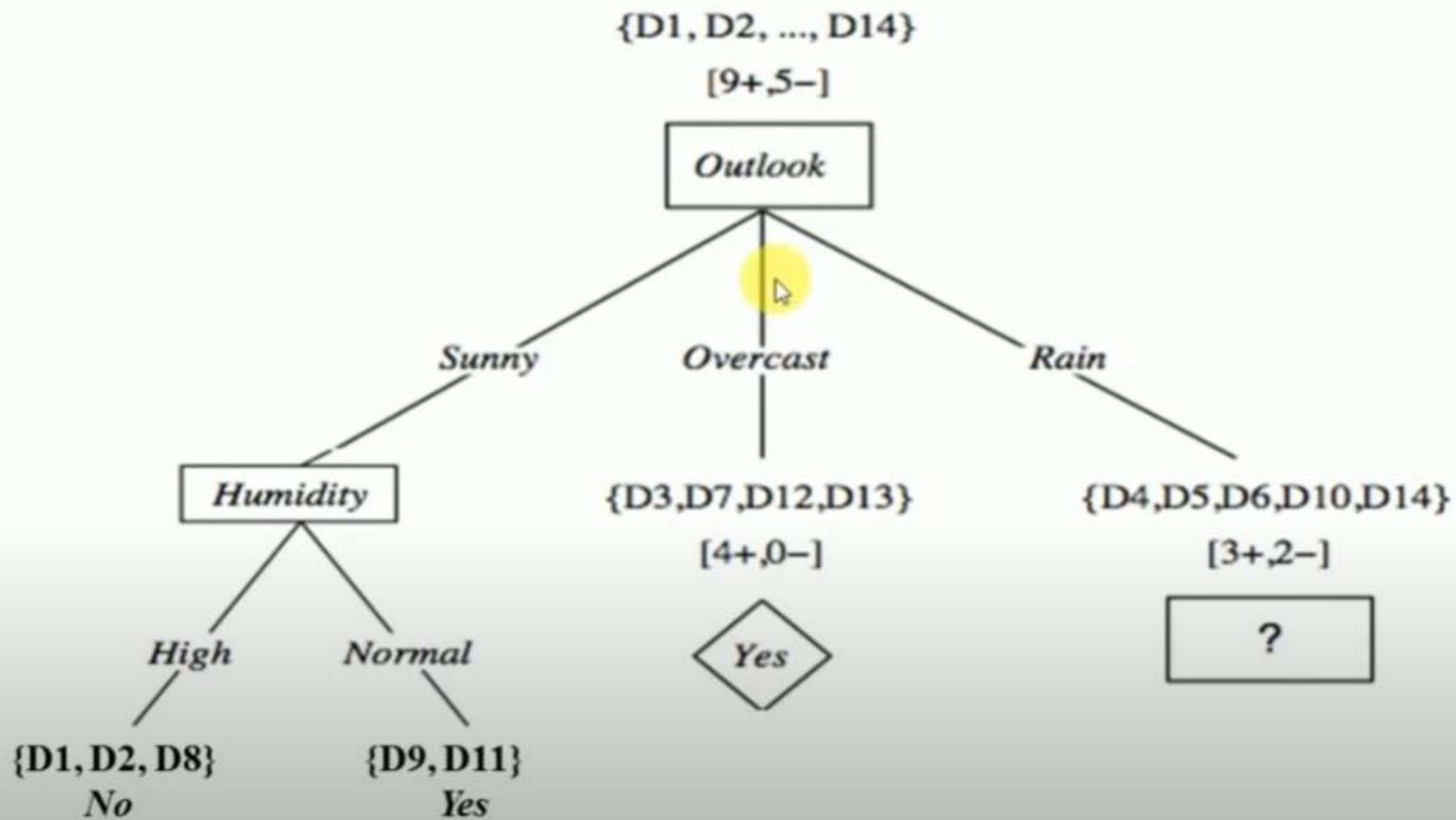


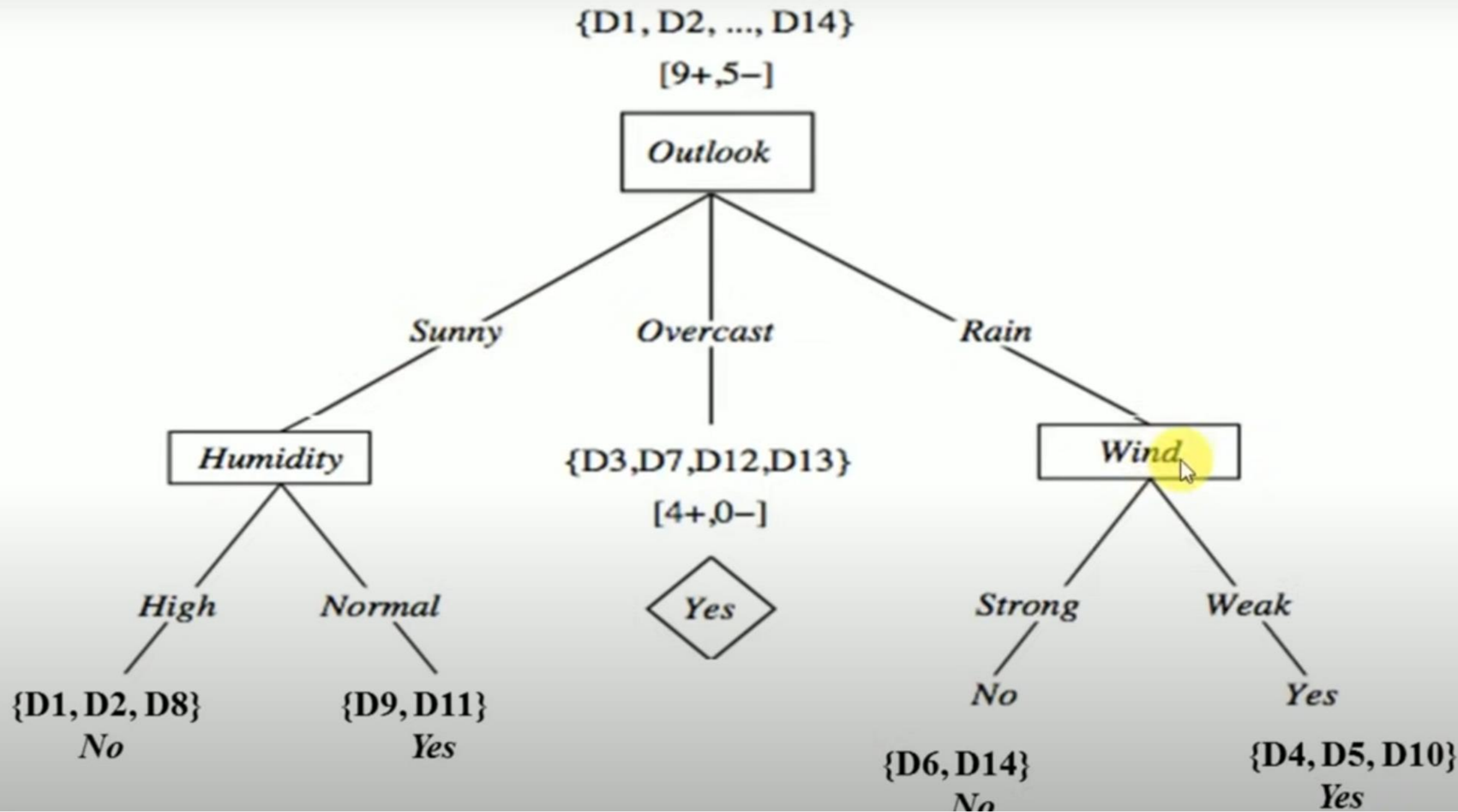
Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

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

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

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

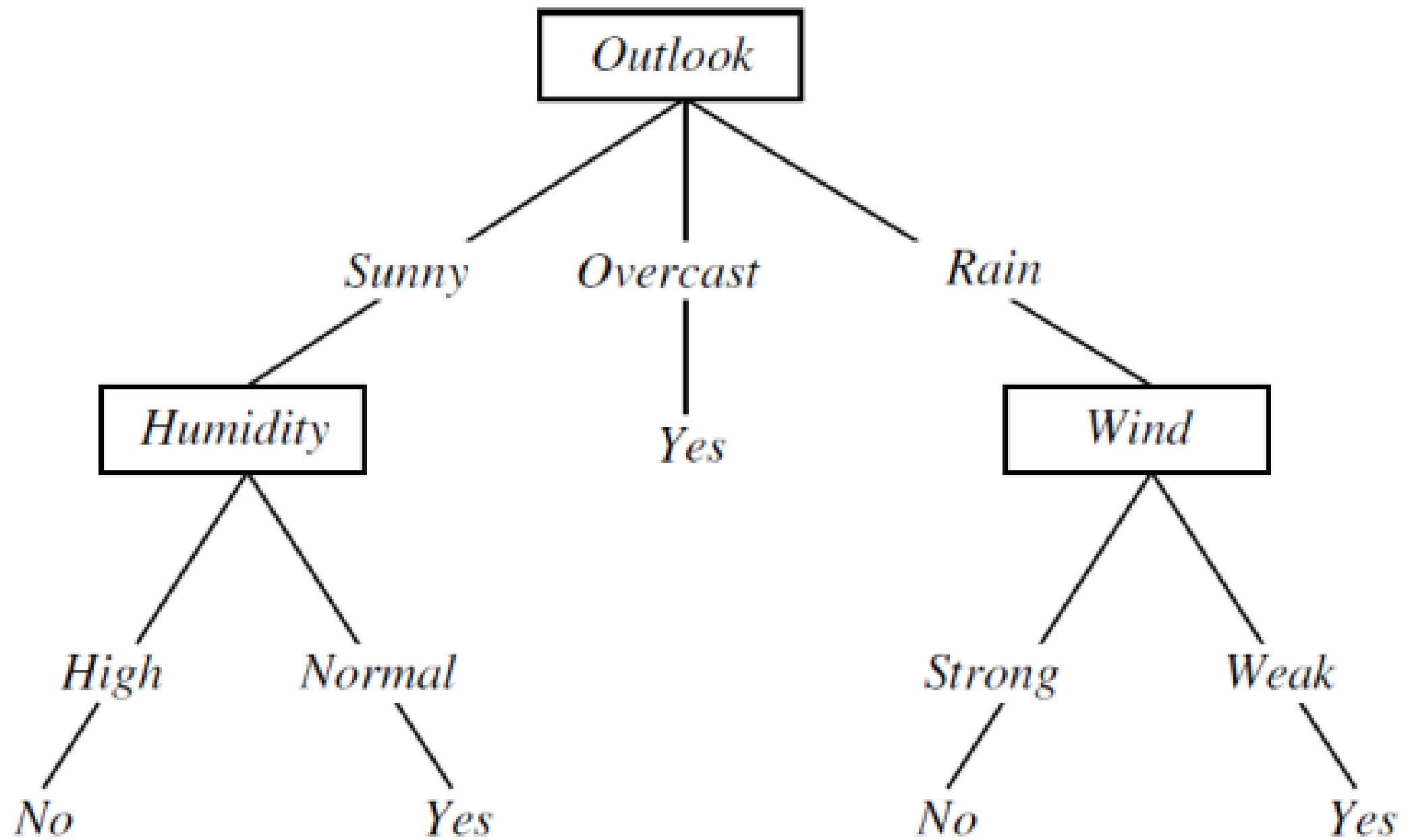




# DECISION TREE REPRESENTATION

- Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance.
- Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute.
- An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute in the given example. This process is then repeated for the subtree rooted at the new node.





- Decision trees represent a disjunction of conjunctions of constraints on the attribute values of instances.
- Each path from the tree root to a leaf corresponds to a conjunction of attribute tests, and the tree itself to a disjunction of these conjunctions. For example, the decision tree shown in above figure corresponds to the expression

$(\text{Outlook} = \text{Sunny} \wedge \text{Humidity} = \text{Normal}) \vee$

$(\text{Outlook} = \text{Overcast}) \vee$

$(\text{Outlook} = \text{Rain} \wedge \text{Wind} = \text{Weak})$

# 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 – Instances are described by a fixed set of attributes and their values.
  2. The target function has discrete output values – The decision tree assigns a Boolean classification (e.g., yes or no) to each example. Decision tree methods easily extend to learning functions with more than two possible output values.
  3. Disjunctive descriptions may be required.
  4. The training data may contain errors – Decision tree learning methods are robust to errors, 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.

# What is ID3?

- A mathematical algorithm for building the decision tree.
- Invented by J. Ross Quinlan in 1979.
- Uses Information Theory invented by Shannon in 1948.
- Builds the tree from the top down, with no backtracking.
- Information Gain is used to select the most useful attribute for classification.



# Entropy

- A formula to calculate the homogeneity of a sample.
- A completely homogeneous sample has entropy of 0.
- An equally divided sample has entropy of 1.
- Entropy(s) = - p+log2 (p+) -p-log2 (p-) for a sample of negative and positive elements.
- The formula for entropy is:

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

# Entropy Example

Entropy(S) =

$$- (9/14) \text{Log}_2 (9/14) - (5/14) \text{Log}_2 (5/14)$$

$$= 0.940$$

# Information Gain (IG)

- The information gain is based on the decrease in entropy after a dataset is split on an attribute.
- Which attribute creates the most homogeneous branches?
- First the entropy of the total dataset is calculated.
- The dataset is then split on the different attributes.
- The entropy for each branch is calculated. Then it is added proportionally, to get total entropy for the split.
- The resulting entropy is subtracted from the entropy before the split.
- The result is the Information Gain, or decrease in entropy.
- The attribute that yields the largest IG is chosen for the decision node.

# Information Gain (cont'd)

- A branch set with entropy of 0 is a leaf node.
- Otherwise, the branch needs further splitting to classify its dataset.
- The ID3 algorithm is run recursively on the non-leaf branches, until all data is classified.

---

### ID3(Examples, Target\_attribute, Attributes)

Examples are the training examples. Target\_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

- Create a Root node for the tree
  - If all Examples are positive, Return the single-node tree Root, with label = +
  - If all Examples are negative, Return the single-node tree Root, with label = -
  - If Attributes is empty, Return the single-node tree Root, with label = most common value of Target\_attribute in Examples
  - Otherwise Begin
    - $A \leftarrow$  the attribute from Attributes that best\* classifies Examples
    - The decision attribute for Root  $\leftarrow A$
    - For each possible value,  $v_i$ , of A,
      - Add a new tree branch below Root, corresponding to the test  $A = v_i$
      - Let *Examples* <sub>$v_i$</sub>  be the subset of Examples that have value  $v_i$  for A
      - If *Examples* <sub>$v_i$</sub>  is empty
        - Then below this new branch add a leaf node with label = most common value of Target\_attribute in Examples
        - Else below this new branch add the subtree  
ID3(*Examples* <sub>$v_i$</sub> , Target\_attribute, Attributes – {A}))
  - End
  - Return Root
-



# Input Parameters:

- Examples: The training examples with known attribute values and corresponding class labels.
- Target attribute: The attribute whose value we want to predict.
- Attributes: A list of attributes that may be used to make decisions.

# Algorithm Flow:

- a. Create a Root node for the tree.
- b. If all Examples have the same class label:
  - If all Examples are positive, return a single-node tree Root with label = "+"
  - If all Examples are negative, return a single-node tree Root with label = "-"
  - If Attributes is empty:
  - Return a single-node tree Root with the label as the most common value of the Target attribute in Examples.

- d. Otherwise, start the decision-making process:
  - i. Calculate the entropy of the current dataset (Examples) using the formula
  - ii. For each attribute in Attributes, calculate the information gain (IG) using the formula
  - iii. Select the attribute with the highest information gain as the decision attribute for the Root node.
  - iv. For each possible value of the selected attribute:
    - Create a new branch below the Root node corresponding to the test "Attribute =  $v_i$ ".
    - Divide the Examples into subsets based on the value of the selected attribute.
    - If a subset is empty:
      - Add a leaf node with the label as the most common value of the Target attribute in the Examples.
    - Otherwise:
      - Recursively create a subtree using the ID3 algorithm with the subset of Examples, excluding the selected attribute.
- e. Return the Root node of the decision tree.

# Hypothesis Space Search In Decision Tree Learning

- In the process of decision tree learning, like with the ID3 algorithm, we're essentially trying to find the best tree structure that accurately classifies our training data. This involves exploring various hypotheses or potential decision trees to find the one that fits our data the best.
- The hypothesis space searched by ID3 is the set of possible decision trees. ID3 performs a simple-to complex, hill-climbing search through this hypothesis space, beginning with the empty tree, then considering progressively more elaborate hypotheses in search of a decision tree that correctly classifies the training data.

# Hypothesis Space Search In Decision Tree Learning

- The goal is to find the tree that maximizes the information gain, which essentially means it helps to classify the training data better.
- One key advantage of ID3 is that it considers all possible decision trees that can be constructed from the available attributes, ensuring it won't miss the target function. However, it only maintains a single hypothesis at any given time, unlike some other methods that keep track of multiple consistent hypotheses. This limitation means it can't explore alternative trees or ask new questions to improve its understanding.

# Hypothesis Space Search In Decision Tree Learning

- Another thing to note is that ID3 doesn't backtrack once it selects an attribute to split the data at a certain level of the tree. This means it might get stuck at locally optimal solutions, missing out on potentially better trees along different paths. To address this, there are extensions like post-pruning, which involves refining the tree structure after it's been constructed.
- ID3 also differs from methods that make decisions based on individual training examples; instead, it uses statistical properties of the entire dataset to guide its decisions. This makes it less sensitive to errors in individual examples and allows it to handle noisy data by accepting hypotheses that might not perfectly fit the training data.



# Inductive Bias in Decision tree Learning

- The inductive bias in decision tree learning, specifically in the ID3 algorithm, refers to the set of assumptions and preferences guiding how the algorithm generalizes from observed training examples to classify unseen instances.
- In simpler terms, it's like the inherent tendencies or rules that ID3 follows when it's making decisions about how to classify things based on the data it has seen.

# Inductive Bias in Decision tree Learning

- **Preference for Shorter Trees:** ID3 prefers simpler decision trees over complex ones. This means it likes to keep the rules as concise as possible. It does this by selecting the first acceptable tree it encounters during its search, favoring shorter paths through the tree.
- **Placing High Information Gain Attributes Close to the Root:** ID3 also tends to prioritize attributes that provide the most useful information for classification. It tries to put these attributes closer to the top of the decision tree, as they can quickly split the data into meaningful subsets.

# Hypothesis Space:

- Definition:** The hypothesis space refers to the set of all possible hypotheses (models or rules) that a learning algorithm can consider to explain the data.
- Characteristics:** It encompasses the range of potential solutions the algorithm can explore during the learning process.
- Example:** In decision tree learning, the hypothesis space includes all possible decision trees that can be formed using different combinations of attributes and decision rules.

# Inductive Bias:

- Definition:** Inductive bias refers to the set of assumptions, preferences, or constraints that a learning algorithm incorporates into its decision-making process when generalizing from observed data to classify unseen instances.
- Characteristics:** It guides the algorithm's learning process by favoring certain hypotheses over others based on predefined criteria or principles.
- Example:** In decision tree learning, the inductive bias might include preferences for simpler trees (those with fewer branches or nodes), favoring attributes with higher information gain, or placing important attributes closer to the root of the tree.

- In summary, the hypothesis space defines the range of possible solutions that a learning algorithm considers, while the inductive bias influences the algorithm's decision-making process within that space by favoring certain types of hypotheses over others based on predefined principles or preferences.