# Decision Tree

- A decision tree is a tree-like structure that is used as a model for classifying data.
- A decision tree decomposes the data into sub-trees made of other sub-trees and/or leaf nodes.
- A decision tree is made up of two types of nodes
  - *Decision Nodes*: These type of node have two or more branches
  - *Leaf Nodes*: The lowest nodes which represents decision

# DataSet

**Attributes** **Classes**

| Outlook | Temperature | Humidity | Windy | Play Golf |
|---|---|---|---|---|
| Rainy | Hot | High | FALSE | No |
| Rainy | Hot | High | TRUE | No |
| Overcast | Hot | High | FALSE | Yes |
| Sunny | Mild | High | FALSE | Yes |
| Sunny | Cool | Normal | FALSE | Yes |
| Sunny | Cool | Normal | TRUE | No |
| Overcast | Cool | Normal | TRUE | Yes |
| Rainy | Mild | High | FALSE | No |
| Rainy | Cool | Normal | FALSE | Yes |
| Sunny | Mild | Normal | FALSE | Yes |
| Rainy | Mild | Normal | TRUE | Yes |
| Overcast | Mild | High | TRUE | Yes |
| Overcast | Hot | Normal | FALSE | Yes |
| Sunny | Mild | High | TRUE | No |

# Step 1: Determine the Decision Column

- Since decision trees are used for clarification, you need to determine the classes which are the basis for the decision.

# Step 1: Determine the Decision Column

- Since decision trees are used for clarification, you need to determine the classes which are the basis for the decision.

- In this case, it it the last column, that is *Play Golf* column with classes ***Yes*** and ***No***.

Attributes Classes

| Outlook | Temperature | Humidity | Windy | Play Golf |
|---|---|---|---|---|
| Rainy | Hot | High | FALSE | No |
| Rainy | Hot | High | TRUE | No |
| Overcast | Hot | High | FALSE | Yes |
| Sunny | Mild | High | FALSE | Yes |
| Sunny | Cool | Normal | FALSE | Yes |
| Sunny | Cool | Normal | TRUE | No |
| Overcast | Cool | Normal | TRUE | Yes |
| Rainy | Mild | High | FALSE | No |
| Rainy | Cool | Normal | FALSE | Yes |
| Sunny | Mild | Normal | FALSE | Yes |
| Rainy | Mild | Normal | TRUE | Yes |
| Overcast | Mild | High | TRUE | Yes |
| Overcast | Hot | Normal | FALSE | Yes |
| Sunny | Mild | High | TRUE | No |

# Step 1: Determine the Decision Column

- Since decision trees are used for clarification, you need to determine the classes which are the basis for the decision.

- In this case, it it the last column, that is *Play Golf* column with classes ***Yes*** and ***No***.

- Next determine the rootNode
  - we need to compute the entropy.
  - To compute the entropy, we create a frequency table for the classes

| Attributes | | | | Classes |
|---|---|---|---|---|
| **Outlook** | **Temperature** | **Humidity** | **Windy** | **Play Golf** |
| Rainy | Hot | High | FALSE | No |
| Rainy | Hot | High | TRUE | No |
| Overcast | Hot | High | FALSE | Yes |
| Sunny | Mild | High | FALSE | Yes |
| Sunny | Cool | Normal | FALSE | Yes |
| Sunny | Cool | Normal | TRUE | No |
| Overcast | Cool | Normal | TRUE | Yes |
| Rainy | Mild | High | FALSE | No |
| Rainy | Cool | Normal | FALSE | Yes |
| Sunny | Mild | Normal | FALSE | Yes |
| Rainy | Mild | Normal | TRUE | Yes |
| Overcast | Mild | High | TRUE | Yes |
| Overcast | Hot | Normal | FALSE | Yes |
| Sunny | Mild | High | TRUE | No |

# Step 1: Determine the Decision Column

| Attributes | | | | Classes |
|---|---|---|---|---|
| Outlook | Temperature | Humidity | Windy | Play Golf |
| Rainy | Hot | High | FALSE | No |
| Rainy | Hot | High | TRUE | No |
| Overcast | Hot | High | FALSE | Yes |
| Sunny | Mild | High | FALSE | Yes |
| Sunny | Cool | Normal | FALSE | Yes |
| Sunny | Cool | Normal | TRUE | No |
| Overcast | Cool | Normal | TRUE | Yes |
| Rainy | Mild | High | FALSE | No |
| Rainy | Cool | Normal | FALSE | Yes |
| Sunny | Mild | Normal | FALSE | Yes |
| Rainy | Mild | Normal | TRUE | Yes |
| Overcast | Mild | High | TRUE | Yes |
| Overcast | Hot | Normal | FALSE | Yes |
| Sunny | Mild | High | TRUE | No |

| Play Golf(14) | |
|---|---|
| Yes | No |
| 9 | 5 |

# Step 2: Calculating Entropy for the classes (Play Golf)

- In this step, you need to calculate the entropy for the Decision Column (Play Golf)
- *Entropy(PlayGolf) =* ***E****(5-,9+)*

| Play Golf(14) | |
|---|---|
| Yes | No |
| 9 | 5 |

# Step 2: Calculating Entropy for the classes (Play Golf)

- In this step, you need to calculate the entropy for the Decision Column (Play Golf)
- *Entropy(PlayGolf) = **E**(5-,9+)*

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

| Play Golf(14) | |
|---|---|
| **Yes** | **No** |
| **9** | **5** |

# Step 2: Calculating Entropy for the classes (Play Golf)

- In this step, you need to calculate the entropy for the Decision Column (Play Golf)
- *Entropy(PlayGolf) = **E**(5-,9+)*

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

$$Entropy(PlayGolf) = -p_{yes} log_2(p_{yes}) - p_{no} log_2(p_{no})$$

| Play Golf(14) | |
|---|---|
| **Yes** | **No** |
| **9** | **5** |

# Step 2: Calculating Entropy for the classes (Play Golf)

- In this step, you need to calculate the entropy for the Decision Column (Play Golf)
- *Entropy(PlayGolf) =* ***E****(5-,9+)*

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

$$Entropy(PlayGolf) = -p_{yes} log_2(p_{yes}) - p_{no} log_2(p_{no})$$

| Play Golf(14) | |
|---|---|
| Yes | No |
| 9 | 5 |

$$\boldsymbol{E(PlayGolf)} = E(5{,}9)$$

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

$$= -(0.357\ \log_2 0.357) - (0.643\ \log_2 0.643)$$

$$= \mathbf{0.94}$$

# Step 3: Calculate Entropy for Other Attributes After Split

For the other four attributes, we need to calculate the entropy after each of the split.

- E(PlayGolf, Outlook)
- E(PlayGolf, Temperature)
- E(PlayGolf, Humidity)
- E(PlayGolf,Windy)

The entropy for two variables  is calculated using the formula.

$$Entropy(\mathrm{S}, \mathrm{T}) = \sum_{c \in T} P(c) E(c)$$

The easiest way to approach this calculation is to create a frequency table for the two variables

# Step 3: Calculate Entropy for Other Attributes After Split

***E(PlayGolf, Outlook) Calculation:***

To calculate **E**(PlayGolf, Outlook), we would use the formula below:

$$\boldsymbol{E}(PlayGolf, Outlook) = \boldsymbol{P}(Sunny)\boldsymbol{E}(Sunny) + \boldsymbol{P}(Overcast)\boldsymbol{E}(Overcast) + \boldsymbol{P}(Rainy)\boldsymbol{E}(Rainy)$$

| | | PlayGolf(14) | | |
|---|---|---|---|---|
| | | Yes | No | |
| Outlook | Sunny | 3 | 2 | 5 |
| | Overcast | 4 | 0 | 4 |
| | Rainy | 2 | 3 | 5 |

*E(PlayGolf, Outlook) = **P**(Sunny) **E**(3,2) + P(Overcast) **E**(4,0) + **P**(rainy) E(2,3)*

$$\boldsymbol{E}(PlayGolf, Outlook) = \frac{5}{14}\boldsymbol{E}(3,2) + \frac{4}{14}\boldsymbol{E}(4,0) + \frac{5}{14}\boldsymbol{E}(2,3)$$

# Step 3: Calculate Entropy for Other Attributes After Split

***E(PlayGolf, Outlook) Calculation:***

| | | PlayGolf(14) | | |
|---|---|---|---|---|
| | | Yes | No | |
| Outlook | Sunny | 3 | 2 | 5 |
| | Overcast | 4 | 0 | 4 |
| | Rainy | 2 | 3 | 5 |

$$\boldsymbol{E}(PlayGolf, Outlook) = \frac{5}{14}\boldsymbol{E}(3{,}2) + \frac{4}{14}\boldsymbol{E}(4{,}0) + \frac{5}{14}\boldsymbol{E}(2{,}3)$$

$$\boldsymbol{E}(\boldsymbol{Sunny}) = E(3{,}2)$$

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

$$= -(0.60\ \log_2 0.60) - (0.40\log_2 0.40)$$

$$= -(0.60 * 0.737) - (0.40 * 0.529)$$

$$= \mathbf{0.971}$$

# Step 3: Calculate Entropy for Other Attributes After Split

***E(PlayGolf, Outlook) Calculation:***

| | | PlayGolf(14) | | |
|---|---|---|---|---|
| | | Yes | No | |
| Outlook | Sunny | 3 | 2 | 5 |
| | Overcast | 4 | 0 | 4 |
| | Rainy | 2 | 3 | 5 |

$$E(PlayGolf, Outlook) = \frac{5}{14}E(3,2) + \frac{4}{14}E(4,0) + \frac{5}{14}E(2,3)$$

$$\boldsymbol{E(Sunny)} = E(3,2)$$

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

$$= -(0.60\ \log_2 0.60) - (0.40\log_2 0.40)$$

$$= -(0.60 * 0.737) - (0.40 * 0.529)$$

$$= \mathbf{0.971}$$

$$\boldsymbol{E(Overcast)} = E(4,0)$$

$$= -\left(\frac{4}{4}\log_2\frac{4}{4}\right) - \left(\frac{0}{4}\log_2\frac{0}{4}\right)$$

$$= -(0) - (0)$$

$$= \mathbf{0}$$

# Step 3: Calculate Entropy for Other Attributes After Split

***E(PlayGolf, Outlook) Calculation:***

| | | PlayGolf(14) | | |
|---|---|---|---|---|
| | | Yes | No | |
| Outlook | Sunny | 3 | 2 | 5 |
| | Overcast | 4 | 0 | 4 |
| | Rainy | 2 | 3 | 5 |

$$E(PlayGolf, Outlook) = \frac{5}{14}E(3,2) + \frac{4}{14}E(4,0) + \frac{5}{14}E(2,3)$$

$$\begin{aligned} \boldsymbol{E(Sunny)} &= E(3,2) \\ &= -\left(\frac{3}{5}\log_2\frac{3}{5}\right) - \left(\frac{2}{5}\log_2\frac{2}{5}\right) \\ &= -(0.60\ \log_2 0.60) - (0.40 \log_2 0.40) \\ &= -(0.60 * 0.737) - (0.40 * 0.529) \\ &= \mathbf{0.971} \end{aligned}$$

$$\begin{aligned} \boldsymbol{E(Overcast)} &= E(4,0) \\ &= -\left(\frac{4}{4}\log_2\frac{4}{4}\right) - \left(\frac{0}{4}\log_2\frac{0}{4}\right) \\ &= -(0) - (0) \\ &= \mathbf{0} \end{aligned}$$

$$\begin{aligned} \boldsymbol{E(Rainy)} &= E(2,3) \\ &= -\left(\frac{2}{5}\log_2\frac{2}{5}\right) - \left(\frac{3}{5}\log_2\frac{3}{5}\right) \\ &= -(0.40\ \log_2 0.40) - (0.6 \log_2 0.60) \\ &= \mathbf{0.971} \end{aligned}$$

# Step 3: Calculate Entropy for Other Attributes After Split

***E(PlayGolf, Outlook) Calculation:***

| | | PlayGolf(14) | | |
|---|---|---|---|---|
| | | Yes | No | |
| Outlook | Sunny | 3 | 2 | 5 |
| | Overcast | 4 | 0 | 4 |
| | Rainy | 2 | 3 | 5 |

$$E(PlayGolf, Outlook) = \frac{5}{14}E(3,2) + \frac{4}{14}E(4,0) + \frac{5}{14}E(2,3)$$

$$\begin{aligned}
\boldsymbol{E(Sunny)} &= E(3,2) \\
&= -\left(\frac{3}{5}\log_2\frac{3}{5}\right) - \left(\frac{2}{5}\log_2\frac{2}{5}\right) \\
&= -(0.60\ \log_2 0.60) - (0.40\log_2 0.40) \\
&= -(0.60 * 0.737) - (0.40 * 0.529) \\
&= \mathbf{0.971}
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{E(Overcast)} &= E(4,0) \\
&= -\left(\frac{4}{4}\log_2\frac{4}{4}\right) - \left(\frac{0}{4}\log_2\frac{0}{4}\right) \\
&= -(0) - (0) \\
&= \mathbf{0}
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{E(Rainy)} &= E(2,3) \\
&= -\left(\frac{2}{5}\log_2\frac{2}{5}\right) - \left(\frac{3}{5}\log_2\frac{3}{5}\right) \\
&= -(0.40\ \log_2 0.40) - (0.6\log_2 0.60) \\
&= \mathbf{0.971}
\end{aligned}$$

$E(4,0) = 0$;
$E(2,3) = E(3,2)$

# Step 3: Calculate Entropy for Other Attributes After Split

***E(PlayGolf, Outlook) Calculation:***

| | | PlayGolf(14) | | |
|---|---|---|---|---|
| | | Yes | No | |
| Outlook | Sunny | 3 | 2 | 5 |
| | Overcast | 4 | 0 | 4 |
| | Rainy | 2 | 3 | 5 |

*E(PlayGolf, Outlook) = P(Sunny) E(3,2) + P(Overcast) E(4,0) + P(rainy) E(2,3)*

$$E(PlayGolf, Outlook) = \frac{5}{14}E(3,2) + \frac{4}{14}E(4,0) + \frac{5}{14}E(2,3)$$

$$= \frac{5}{14}0.971 + \frac{4}{14}0.0 + \frac{5}{14}0.971$$

$$= 0.357 * 0.971 + 0.0 + 0.357 * 0.971$$

$$= 0.693$$

# Step 3: Calculate Entropy for Other Attributes After Split

***E(PlayGolf, Temperature) Calculation***

| | | PlayGolf(14) | | |
|---|---|---|---|---|
| | | Yes | No | |
| Temperature | Hot | 2 | 2 | 4 |
| | Cold | 3 | 1 | 4 |
| | Mild | 4 | 2 | 6 |

*E(PlayGolf, Temperature) = P(Hot) E(2,2) + P(Cold) E(3,1) + P(Mild) E(4,2)*

E (PlayGolf, Temperature) = **4/14 * E**(Hot) + **4/14 * E**(Cold) + **6/14 * E**(Mild)

E (PlayGolf, Temperature) = **4/14 * E**(2, 2) + **4/14 * E**(3, 1) + **6/14 * E**(4, 2)

E (PlayGolf, Temperature) = 4/14 * -(2/4 log 2/4) – (2/4 log 2/4)

+ 4/14 * -(3/4 log 3/4) – (1/4 log 1/4)

+ 6/14 * -(4/6 log 4/6) – (2/6 log 2/6)

E (PlayGolf, Temperature) = 5/14 * 1.0

+ 4/14 * 1.811

+ 5/14 * 0.918

**= 0.911**

# Step 3: Calculate Entropy for Other Attributes After Split

***E(PlayGolf, Humidity) Calculation***

| | | PlayGolf(14) | | |
|---|---|---|---|---|
| | | Yes | No | |
| Humidity | High | 3 | 4 | 7 |
| | Normal | 6 | 1 | 7 |

E (PlayGolf, Humidity) = **7/14 * E**(High) + **7/14 * E**(Normal)

E (PlayGolf, Humidity) = **7/14 *** E(3,4) + **7/14 *** E(6,1)

E (PlayGolf, Humidity) = 7/14 * -(3/7 log 3/7) – (4/7 log 4/7)

+ 7/14 * -(6/7 log 6/7) – (1/7 log 1/7)

E (PlayGolf, Humidity) = 7/14 * 0.985

+ 7/14 * 0.592

**= 0.788**

# Step 3: Calculate Entropy for Other Attributes After Split

***E(PlayGolf, Windy) Calculation***

| | | PlayGolf(14) | | |
|---|---|---|---|---|
| | | Yes | No | |
| Windy | TRUE | 3 | 3 | 6 |
| | FALSE | 6 | 2 | 8 |

E (PlayGolf, Windy) = **6/14 * E**(True) + **8/14 * E**(False)

E (PlayGolf, Windy) = **6/14 * E**(3, 3) + **8/14 * E**(6, 2)

E (PlayGolf, Windy) = 6/14 * -(3/6 log 3/6) – (3/6 log 3/6)

+ 8/14 * -(6/8 log 6/8) – (2/8 log 2/8)

E (PlayGolf, Windy) = 6/14 * 1.0

+ 8/14 * 0.811

= **0.892**

# Step 3: Calculate Entropy for Other Attributes After Split

1. E(PlayGolf, Outlook) = **0.693**
2. E(PlayGolf, Temperature) = **0.911**
3. E(PlayGolf, Humidity) = **0.788**
4. E(PlayGolf,Windy) = **0.892**

# Step 4: Calculating Information Gain for Each Split

- The next step is to calculate the information gain for each of the attributes.
- The information gain is calculated from the split using each of the attributes.
- Then the attribute with the largest information gain is used for the split.
- The information gain is calculated using the formula:

$$\text{Gain}(S,T) = \text{Entropy}(S) - \text{Entropy}(S,T)$$

# Step 4: Calculating Information Gain for Each Split

*Gain(PlayGolf, Outlook) = Entropy(PlayGolf) – Entropy(PlayGolf, Outlook)*
= 0.94 – 0.693 **= 0.247**

*Gain(PlayGolf, Temperature) = Entropy(PlayGolf) – Entropy(PlayGolf, Temperature)*
*= 0.94 – 0.911* ***= 0.029***

*Gain(PlayGolf, Humidity) = Entropy(PlayGolf) – Entropy(PlayGolf, Humidity)*
*= 0.94 – 0.788* ***= 0.152***

*Gain(PlayGolf, Windy) = Entropy(PlayGolf) – Entropy(PlayGolf, Windy)*
*= 0.94 – 0.892* ***= 0.048***

# Step 4: Calculating Information Gain for Each Split

*Gain(PlayGolf, Outlook) = Entropy(PlayGolf) – Entropy(PlayGolf, Outlook)*
= 0.94 – 0.693 **= 0.247**

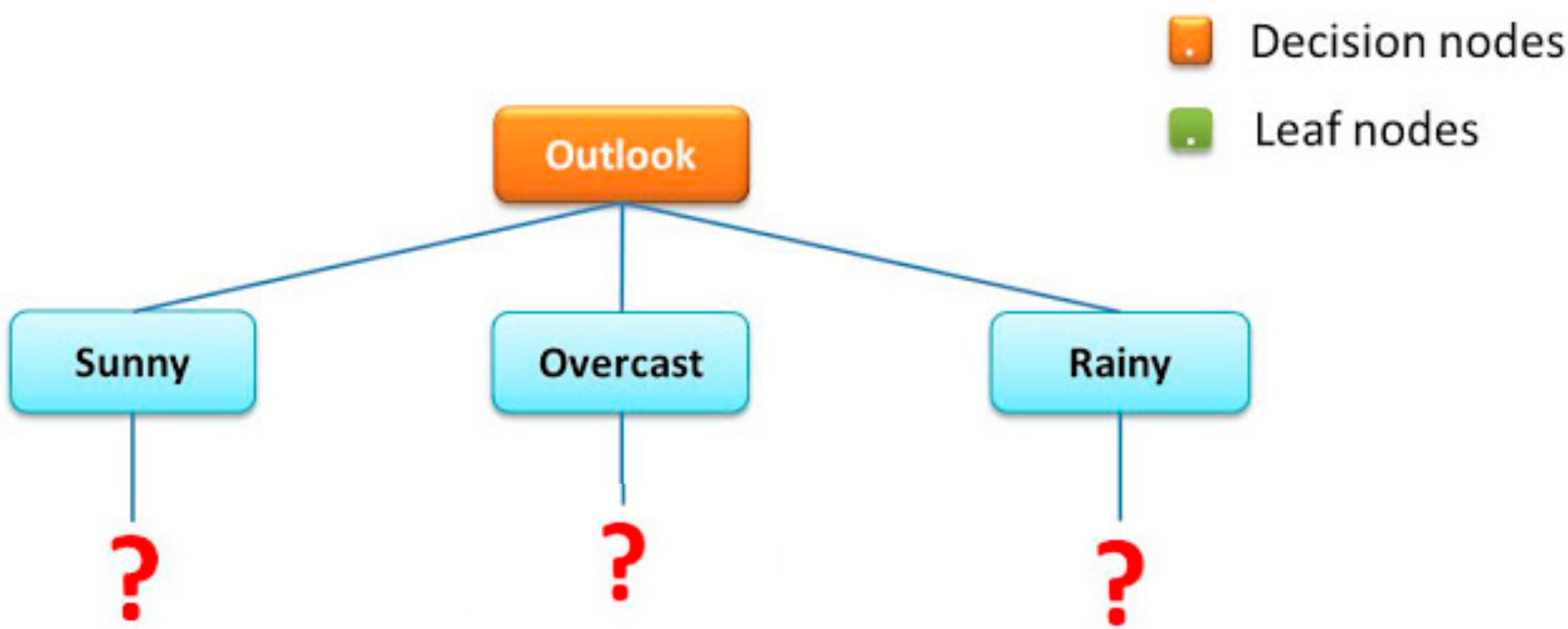
*Gain(PlayGolf, Temperature) = Entropy(PlayGolf) – Entropy(PlayGolf, Temperature)*
*= 0.94 – 0.911* ***= 0.029***

*Gain(PlayGolf, Humidity) = Entropy(PlayGolf) – Entropy(PlayGolf, Humidity)*
*= 0.94 – 0.788* ***= 0.152***

*Gain(PlayGolf, Windy) = Entropy(PlayGolf) – Entropy(PlayGolf, Windy)*
*= 0.94 – 0.892* ***= 0.048***

# Step 5: Perform the First Split

From our calculation, the highest information gain comes from Outlook. Therefore the split will look like this:



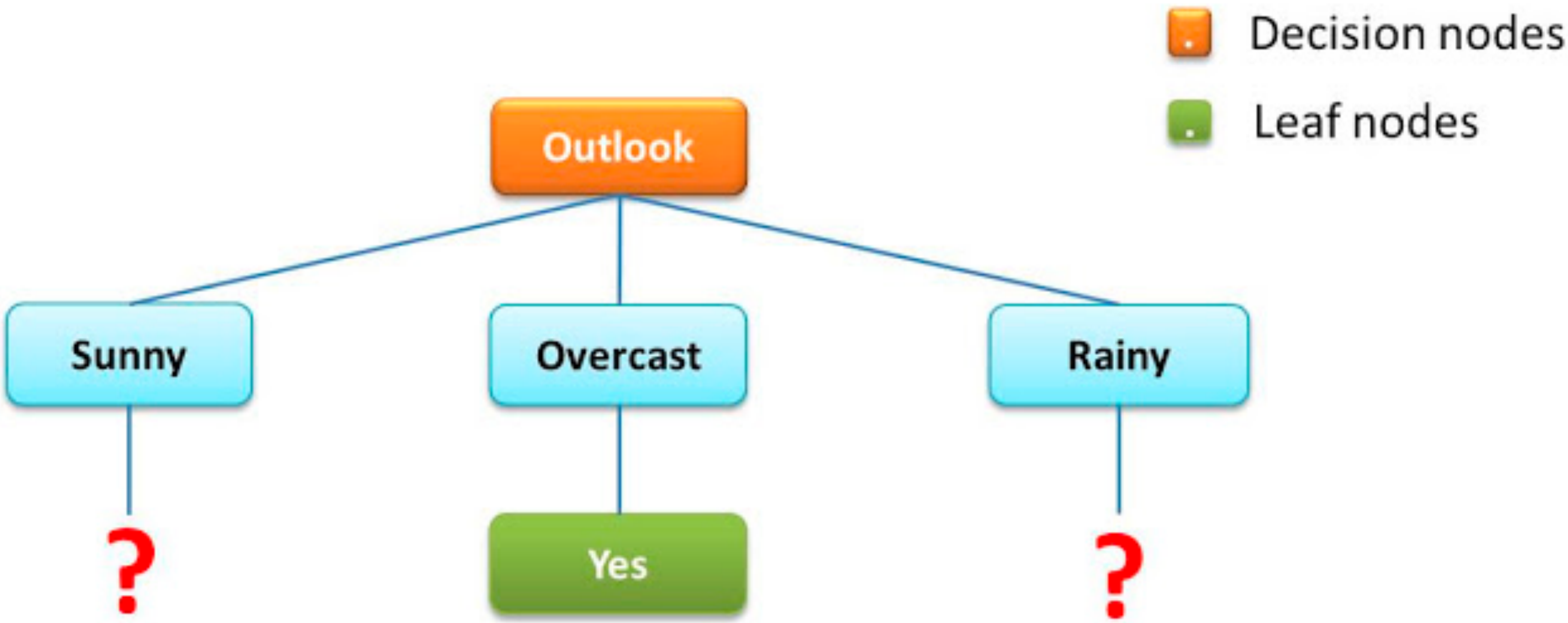
# Step 5: Perform the First Split

| Outlook | Temperature | Humidity | Windy | Play Golf |
|---|---|---|---|---|
| Sunny | Mild | Normal | FALSE | Yes |
| Sunny | Mild | High | FALSE | Yes |
| Sunny | Cool | Normal | FALSE | Yes |
| Sunny | Cool | Normal | TRUE | No |
| Sunny | Mild | High | TRUE | No |
| | | | | |
| Overcast | Hot | High | FALSE | Yes |
| Overcast | Mild | High | TRUE | Yes |
| Overcast | Hot | Normal | FALSE | Yes |
| Overcast | Cool | Normal | TRUE | Yes |
| | | | | |
| Rainy | Hot | High | FALSE | No |
| Rainy | Hot | High | TRUE | No |
| Rainy | Mild | High | FALSE | No |
| Rainy | Cool | Normal | FALSE | Yes |
| Rainy | Mild | Normal | TRUE | Yes |

Overcast outlook requires no further split because it is just one homogeneous group. So we have a leaf node.

# Step 5: Perform the First Split

From our calculation, the highest information gain comes from Outlook. Therefore the split will look like this:




Overcast outlook requires no further split because it is just one homogeneous group. So we have a leaf node.

# Step 6: Perform Further Splits

The Sunny and the Rainy attributes needs to be split

The Rainy outlook can be split using either Temperature, Humidity or Windy.

| *Question*: What attribute would best be used for this split? |
|---|

- *Gain(PlayGolf, Outlook=Rainy, Temperature) =*
  *Entropy(PlayGolf, Outlook=Rainy) – Entropy(PlayGolf, Outlook=Rainy, Temperature)*
- *Gain(PlayGolf, Outlook=Rainy, Humidity) =*
  *Entropy(PlayGolf, Outlook=Rainy) – Entropy(PlayGolf, Outlook=Rainy, Humidity)*
- *Gain(PlayGolf, Outlook=Rainy, Windy) =*
  *Entropy(PlayGolf, Outlook=Rainy) – Entropy(PlayGolf, Outlook=Rainy, Windy)*

# Step 6: Perform Further Splits

The Sunny and the Rainy attributes needs to be split

The Rainy  outlook can be split using either Temperature, Humidity or Windy.

*Question*: What attribute would best be used for this split?

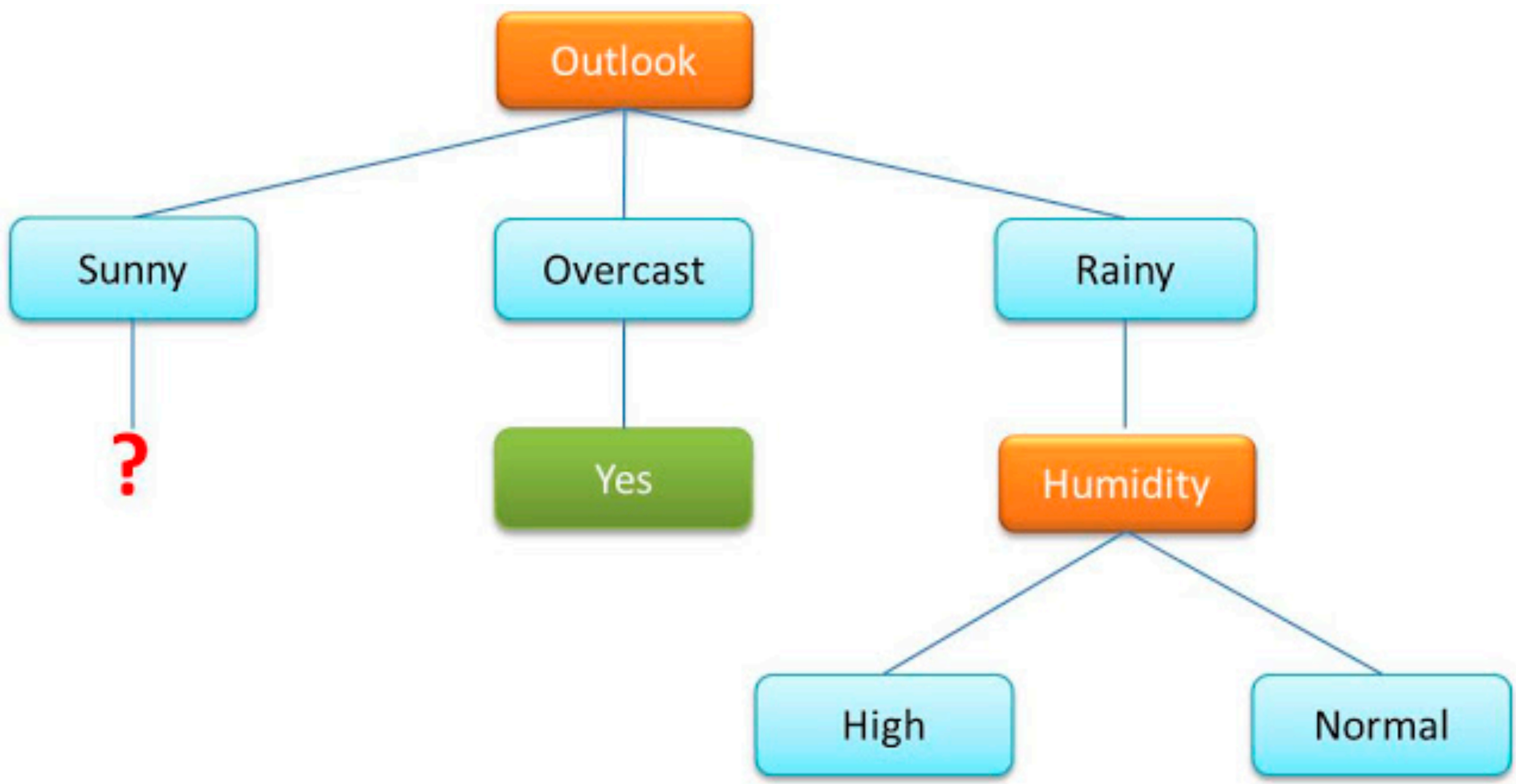
**Humidity,** produces homogenous  groups.

| Outlook | Temperature | Humidity | Windy | Play Golf |
|---|---|---|---|---|
| Rainy | Hot | High | FALSE | No |
| Rainy | Hot | High | TRUE | No |
| Rainy | Mild | High | FALSE | No |

| | | | | |
|---|---|---|---|---|
| Rainy | Cool | Normal | FALSE | Yes |
| Rainy | Mild | Normal | TRUE | Yes |

- *Gain(PlayGolf, Outlook=Rainy, Humidity) =*
  *Entropy(PlayGolf, Outlook=Rainy) – Entropy(PlayGolf, Outlook=Rainy, Humidity)=*
  *Entropy(PlayGolf, Outlook=Rainy) – 0*

# Step 6: Perform Further Splits



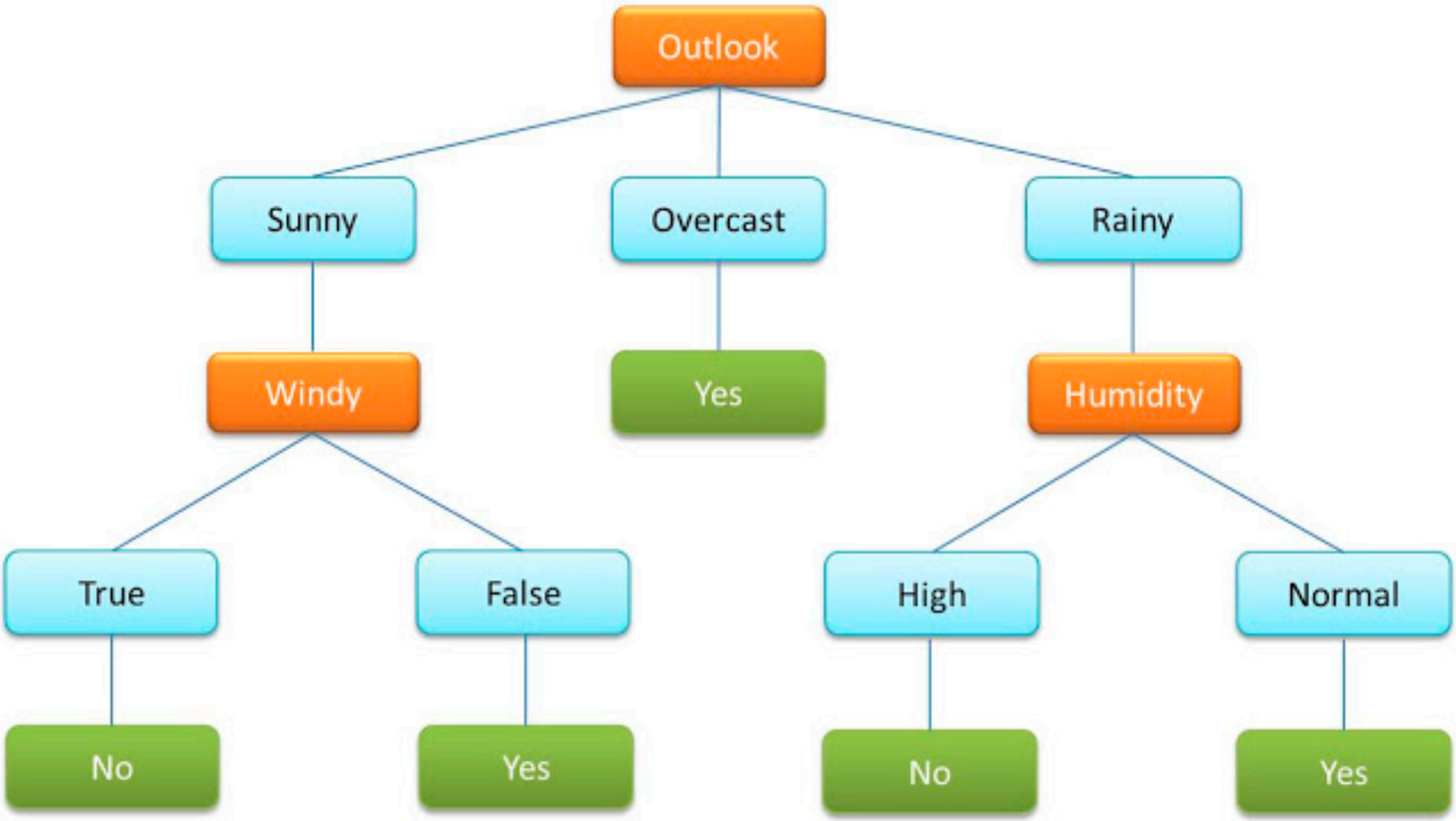
# Step 6: Perform Further Splits

The Rainy outlook can be split using either Temperature, Humidity or Windy.

*Question*: What attribute would best be used for this split? Why?
*Answer*: **Windy** . Because it produces homogeneous groups.

| Outlook | Temperature | Humidity | Windy | Play Golf |
|---|---|---|---|---|
| Sunny | Mild | Normal | FALSE | Yes |
| Sunny | Mild | High | FALSE | Yes |
| Sunny | Cool | Normal | FALSE | Yes |
| Sunny | Cool | Normal | TRUE | No |
| Sunny | Mild | High | TRUE | No |

# Step 6: Perform Further Splits



# ID3 Algorithm

**ID3 (S, A, V)**

```
Let:
    S = Learning Set
    A = Attibute Set
    V = Attribute Values

Begin
    Load learning sets  and create decision tree root node(rootNode),
    Add learning set S into root not as its subset

    For rootNode, compute Entropy(rootNode.subset)

    If Entropy(rootNode.subset) == 0 (subset is homogeneous)
        return a leaf node

    If Entropy(rootNode.subset)!= 0 (subset is not homogeneous)
        compute Information Gain for each attribute left (not been used for spliting)
        Find attibute A with Maximum(Gain(S,A))
        Create child nodes for this root node and add to rootNode in the decision tree

    For each child of the rootNode
        Apply ID3(S,A,V)
        Continue until a node with Entropy of 0 or a leaf node is reached
End
```

https://github.com/jeniyat/cse_5521/blob/master/diabetes.csv

# Continuous Valued Attributes

Create a discrete attribute to test continuous

- $Temperature = 82.5$
- $(Temperature > 72.3) = t, f$

| | | | | | | |
|---|---|---|---|---|---|---|
| *Temperature:* | 40 | 48 | 60 | 72 | 80 | 90 |
| *Play Golf:* | No | No | Yes | Yes | Yes | No |

# Unknown Attribute Values

What if some examples are missing values of $A$?

Use training example anyway, sort through tree

- If node $n$ tests $A$, assign most common value of $A$ among other examples sorted to node $n$
- Assign most common value of $A$ among other examples with same target value
- Assign probability $p_i$ to each possible value $v_i$ of $A$
  Assign fraction $p_i$ of example to each descendant in tree

Classify new examples in same fashion