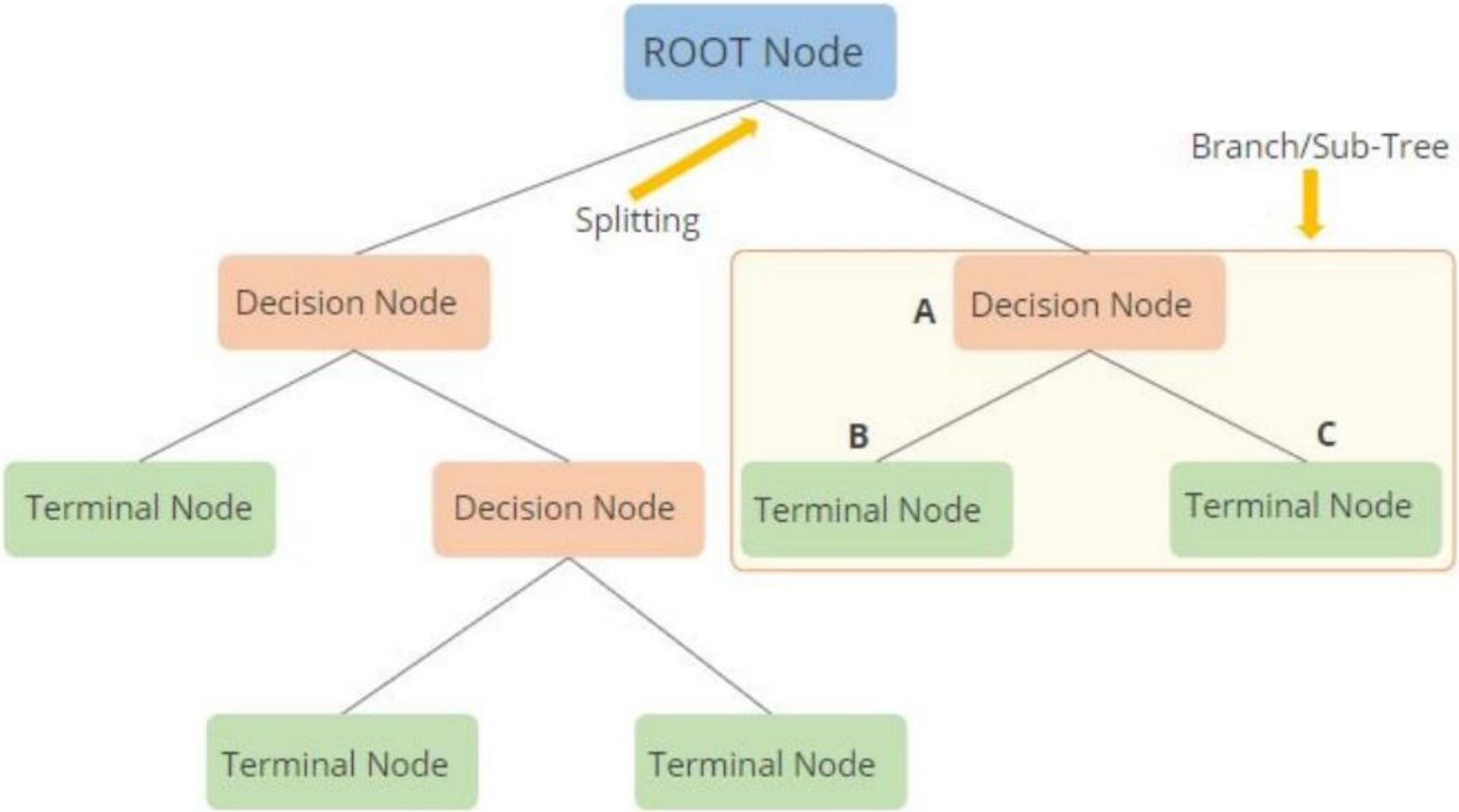
Decision Tree: A Tree-based Algorithm in Machine Learning

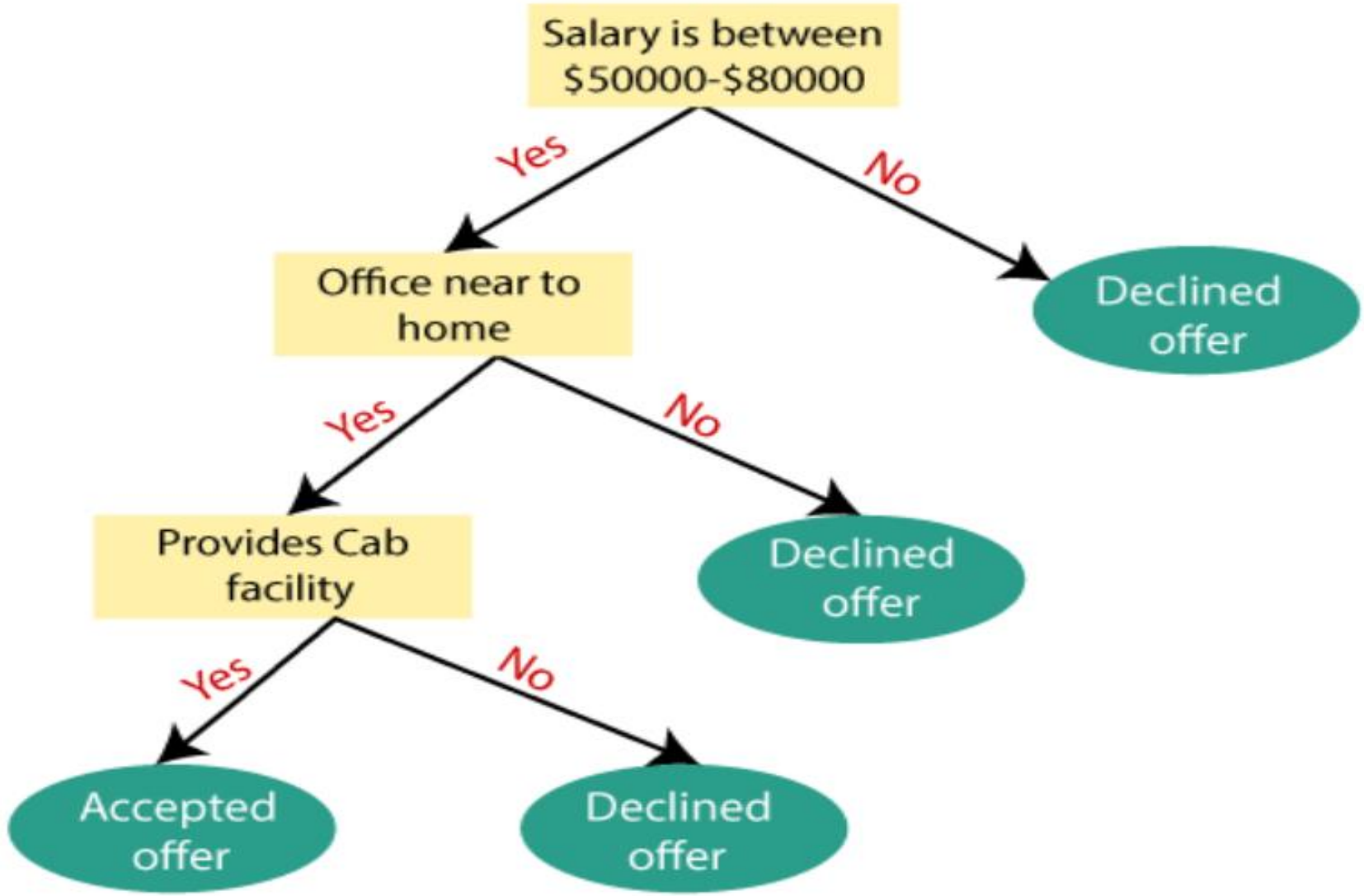
Introduction

A Decision Tree is a hierarchical breakdown of a dataset from the **root node** to the **leaf node** based on the governing attributes to solve a classification or regression problem. Decision Trees (DTs) are a **nonparametric** supervised learning algorithm that predicts the value of a target variable by learning rules inferred from the data features.

But before moving any further, let's first learn some basic terminologies of tree-based algorithms.

- **Root Node:** The top of the tree contains the most essential attribute for the training data.
- **Splitting:** Division of nodes into two or more sub-nodes.
- **Branch Node:** If the sub-node is further split. Each condition is represented as the branch node and is also known as a decision node.
- **Leaf Node:** Nodes that do not split are the leaf or terminal nodes.
- **Parent node:** If the node is split into sub-nodes, that node is the parent for the sub-nodes formed after splitting.
- **Child node:** All the sub-nodes of the parent node.
- **Pruning:** Removing some splitting, we can consider this as the opposite of spitting.


ROOT Node
Splitting
Branch/Sub-Tree
Decision Node
A
Decision Node
B
C
Terminal Node
Decision Node
Terminal Node
Terminal Node
Terminal Node
Terminal Node


Salary is between $50000-$80000
Yes
No
Declined offer
Office near to home
Yes
No
Declined offer
Provides Cab facility
Yes
No
Accepted offer
Declined offer

Day	outlook	temp	humidity	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

- **Dataset Example**

Decision tree machine learning algorithm for predicting weather conditions. Each row in the dataset represents a specific day, and the columns contain weather-related attributes of that day. Here are the attributes:

- **Day:** This is likely a unique identifier for the day, but it could also be the date.
- **Outlook:** This is the forecast for the day, and it can be sunny, overcast, or rainy.
- **Temp:** This is the temperature for the day, and it can be hot, mild, or cool.
- **Humidity:** This is the humidity level for the day, and it can be high or normal.
- **Windy:** This is a binary attribute that indicates whether it is windy (TRUE) or not windy (FALSE).
- **Play:** This is the target variable that the decision tree will try to predict. It is a binary attribute that indicates whether it is suitable to play outdoors (yes) or not (no).

The decision tree algorithm will learn from this data and build a model that can be used to predict the playability of future days based on their weather conditions.

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

where c is the number of output classes.

Entropy

Entropy of a dataset is the average amount of information needed to classify any observation in the data. It is termed Uncertainty. If Entropy is higher, the confidence in classifying any observation into any class is lower and vice-versa.

$$IG(S,A) = Entropy(S) - \sum_{v=Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (2)$$

- **Information Gain**

information gain is a concept in information theory and machine learning that quantifies the reduction in entropy. Information gain measures how much "information" is gained by splitting the data on a particular attribute. The attribute with the highest information gain is chosen as the next node (first in the case of "root node") in the tree.

- In the above equation, **Sv/S** is the probability of that particular value in the given data.

First, we check the Entropy of the dataset using (1). There are two different classes in the output, so **c = 2**. Next, among the 14 samples of the dataset, 9 are 'yes' and 5 are 'no'. Thus:

$$E(S) = -\left(\frac{9}{14}\right)\log_2\frac{9}{14} - \left(\frac{5}{14}\right)\log_2\frac{5}{14} = 0.94$$

Due to an imbalance in the dataset, the Entropy is not equal to 1. The function written below can compute the Entropy of the entire dataset or the dataset with respect to any particular attribute.

###Entropy(data) Entropy(data.Loc[data['outlook']=='sunny'])

```
Entropy(data)
0.9402859586706311
```

```
Entropy(data.loc[data['outlook'] == 'sunny'])
0.9709505944546686
```

❑ Let's first calculate the Information Gain of the "Outlook" attribute.

$$= 0.94 - \left(\frac{5}{14}\right)E(S_{sunny}) - \left(\frac{4}{14}\right)E(S_{overcast}) - \left(\frac{5}{14}\right)E(S_{rainy})$$

❑ Entropy and Information Gain for Outlook

Day	outlook	temp	humid	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes

Day	outlook	temp	humid	windy	play
3	overcast	hot	high	FALSE	yes
7	overcast	cool	normal	TRUE	yes
12	overcast	mild	high	FALSE	yes
13	overcast	hot	normal	FALSE	yes

Day	outlook	temp	humid	windy	play
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
10	rainy	mild	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

$$E(S_{sunny}) = -\left(\frac{3}{5}\right)\log_2\frac{3}{5} - \left(\frac{2}{5}\right)\log_2\frac{2}{5} = 0.97$$

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

$$E(S_{rainy}) = -\left(\frac{3}{5}\right)\log_2\frac{3}{5} - \left(\frac{2}{5}\right)\log_2\frac{2}{5} = 0.97$$

Thus,

$$IG(S, A = \text{'outlook'}) = 0.94 - \left(\frac{5}{14}\right)*0.97 - \left(\frac{4}{14}\right)*0 - \left(\frac{5}{14}\right)*0.971 = 0.246$$

Similarly

$$IG(S, A = \text{'humidity'}) = E(S) - \sum_{v=[high,normal]} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$= 0.94 - \left(\frac{7}{14}\right)E(S_{high}) - \left(\frac{7}{14}\right)E(S_{normal})$$

❑ Information gain for outlook is = 0.246

- Entropy and Information Gain for Humidity

Day	outlook	temp	humid	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
8	sunny	mild	high	FALSE	no
12	overcast	mild	high	FALSE	yes
14	rainy	mild	high	TRUE	no

Day	outlook	temp	humid	windy	play
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
9	sunny	cool	normal	FALSE	yes
10	Rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
13	overcast	hot	normal	FALSE	yes

- Information Gain for Humidity : 0.152

$$E(S_{high}) = -\left(\frac{3}{7}\right)\log_2\frac{3}{7} - \left(\frac{4}{7}\right)\log_2\frac{4}{7} = 0.985$$

$$E(S_{normal}) = -\left(\frac{1}{7}\right)\log_2\frac{1}{7} - \left(\frac{6}{7}\right)\log_2\frac{6}{7} = 0.591$$

$$IG(S, A = \text{`humidity'}) = 0.94 - \left(\frac{7}{14}\right)*0.985 - \left(\frac{7}{14}\right)*0.591 = 0.152$$

- Entropy and Information Gain for Windy & Temperature :

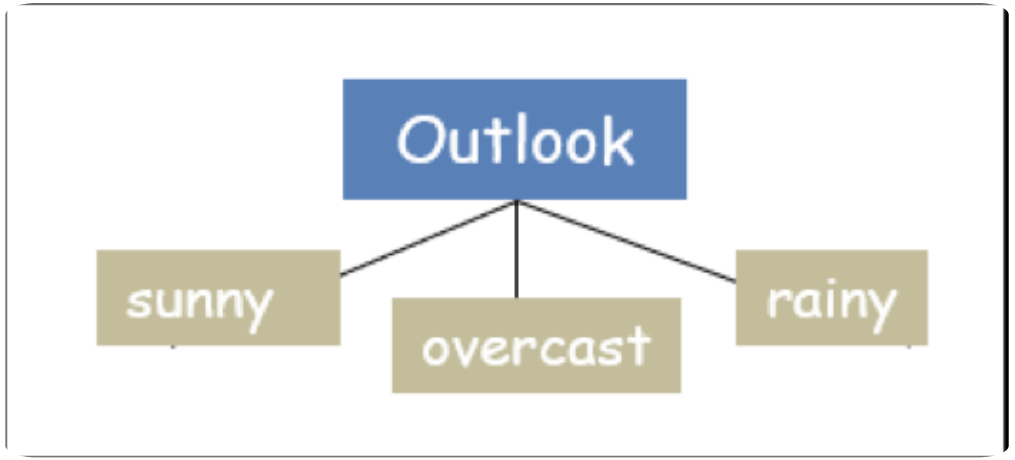
$$IG(S, A =' windy') = E(S) - \Sigma_{v=[True,False]} \frac{|S_v|}{|S|} Entropy(S_v)$$
$$= 0.94 - \frac{8}{14} E(S_{False}) - \frac{6}{14} E(S_{True})$$

$$IG(S, A =' temp') = E(S) - \Sigma_{v=[hot,mild,cool]} \frac{|S_v|}{|S|} Entropy(S_v)$$
$$= 0.94 - \frac{4}{14} E(S_{hot}) - \frac{6}{14} E(S_{mild}) - \frac{4}{14} E(S_{cool})$$

```
IG(data,'windy')
0.04812703040826954
```

```
IG(data,'temp')
0.029222565658954813
```

- Information Gain for Windy : 0.048
- Information Gain for Temperature : 0.029



- By comparing the information gain of all the attributes of the dataset
- ([‘outlook = 0.246’, ‘temp = 0.029’, ‘humidity = 0.152’, ‘windy = 0.048’]), it is observed that ‘outlook’ has the highest information gain, IG(S, A = ‘outlook’) = 0.246. Thus, the first node\ Root Node is selected as 'outlook’.

- Now concerning each attribute of outlook (['sunny',' overcast',' rainy']), the information gain is to be computed for all the remaining attributes of the dataset (['humidity', 'temp', 'windy']), provided the Entropy of the dataset is not zero. Please remember that every attribute can appear only once in the tree. Now let's further grow the tree.

❑Outlook — Sunny:

Day	outlook	temp	humid	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes

```
Entropy(data.loc[data['outlook'] == 'sunny'])
0.970950594454686
```

❑ The Entropy of the dataset is non-zero, so information gain is computed

```
IG(data.loc[data['outlook'] == 'sunny'],'humidity')
0.970950594454686

IG(data.loc[data['outlook'] == 'sunny'],'temp')
0.570950594454686

IG(data.loc[data['outlook'] == 'sunny'],'windy')
0.0199730940219747 8
```

- **Outlook — Rainy:**

Day	outlook	temp	humid	windy	play
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
10	rainy	mild	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

```
Entropy(data.loc[data['outlook'] == 'rainy'])
0.9709505944546686
```

- The Entropy of the dataset is non-zero, so information gain is computed.

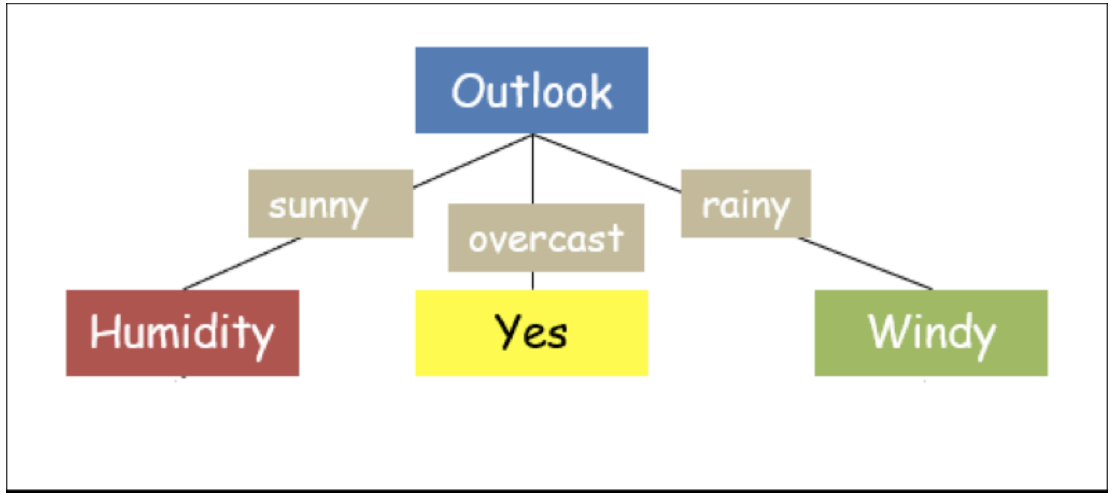
```
IG(data.loc[data['outlook'] == 'rainy','humidity')
0.01997309402197478

IG(data.loc[data['outlook'] == 'rainy','temp')
0.01997309402197478

IG(data.loc[data['outlook'] == 'rainy','windy')
0.9709505944546686
```

- **Outlook — Overcast:**

```
Entropy(data.loc[data['outlook'] == 'overcast'])
0.0
```

Day	outlook	temp	humid	windy	play
3	overcast	hot	high	FALSE	yes
7	overcast	cool	normal	TRUE	yes
12	overcast	mild	high	FALSE	yes
13	overcast	hot	normal	FALSE	yes

- Now, with respect to 'overcast', the Entropy of the dataset is 0. This means that all the observations with respect to this attribute have the same class ('yes' in our case). Thus, the output can be labeled 'yes' for the 'overcast' attribute of 'outlook'. Thus, the tree further grows as:



- The remaining (unused) attribute of the dataset is 'temp'; thus, the IG needs to be computed with respect to 'humidity' and 'windy'. 'Humidity' has two attributes — 'high' and 'normal', and it has 'overcast' with the decision 'sunny' as its root (parent) node.

- **Outlook — sunny — Humidity — high:**

```
Entropy(data.loc[(data['outlook']=='sunny') & (data['humidity'] == 'high')])
0.0
```

Day	outlook	temp	humid	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
8	sunny	mild	high	FALSE	no

- As the Entropy of the above dataset is 0 thus, the output can be labeled as the only class present in the dataset ('no') for the attribute 'high' of humidity' along the tree.

- **Outlook — rainy — Windy — FALSE :**

```
Entropy(data.loc[(data['outlook']=='rainy') & (data['windy'] == False)])
0.0
```

Day	outlook	temp	humid	windy	play
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes

As the Entropy of the above dataset is 0 thus, the output can be labeled as the only class present in the dataset ('yes') for the attribute 'FALSE' of 'windy' along the tree.

```
Entropy(data.loc[(data['outlook']=='rainy') & (data['windy'] == True)])
0.0
```

Day	outlook	temp	humid	windy	play
6	rainy	cool	normal	TRUE	no
14	rainy	mild	high	TRUE	no

- **Outlook — rainy — Windy — TRUE:**

```
Entropy(data.loc[(data['outlook']=='rainy') & (data['windy'] == True)])
0.0
```

Day	outlook	temp	humid	windy	play
6	rainy	cool	normal	TRUE	no
14	rainy	mild	high	TRUE	no

As the Entropy of the above dataset is 0 thus, the output can be labeled as the only class present in the dataset ('no') for the attribute 'TRUE' of 'windy' along the tree. Thus, the final tree can be drawn as: