



```
In [21]: import numpy as np
import pandas as pd
import csv
import math
```

```
In [22]: data = pd.read_csv("Weather-D.csv")
print(data)
```

	Outlook	Temperature	Humidity	Windy	Play Football
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No
2	Overcast	Hot	High	Weak	Yes
3	Rainy	Mild	High	Weak	Yes
4	Rainy	Cool	Normal	Weak	Yes
5	Rainy	Cool	Normal	Strong	No
6	Overcast	Cool	Normal	Strong	Yes
7	Sunny	Mild	High	Weak	No
8	Sunny	Cool	Normal	Weak	Yes
9	Rainy	Mild	Normal	Weak	Yes
10	Sunny	Mild	Normal	Strong	Yes
11	Overcast	Mild	High	Strong	Yes
12	Overcast	Hot	Normal	Weak	Yes
13	Rainy	Mild	High	Strong	No

```
In [23]: total_yes = len(data[data["Play Football"] == "Yes"])
total_nos = len(data[data["Play Football"] == "No"])
total = total_yes + total_nos
print(total, total_yes, total_nos)
```

14 9 5

```
In [24]: def entropy(pos, neg):
    total = pos + neg
    if total == 0 or pos == 0 or neg == 0:
        return 0
    p_pos = pos/total
    p_neg = neg/total
    return -p_pos * math.log2(p_pos) - p_neg * math.log2(p_neg)
```

```
In [25]: entropy_total = entropy(total_yes, total_nos)
print(entropy_total)
```

0.9402859586706311

```
In [34]: def calc_gain(attribute):
    categories = data[attribute].unique()
    weighted_entropy = 0
    print(f"\nCalculating Information Gain for feature: {attribute}")
    for cat in categories:
        subset = data[data[attribute] == cat]
        pos = len(subset[subset["Play Football"] == "Yes"])
        neg = len(subset[subset["Play Football"] == "No"])
        ent = entropy(pos, neg)
        weight = len(subset)/total
```

```

        weighted_entropy += weight * ent
        print(f"{cat} -> Yes: {pos}, No: {neg}, Entropy: {ent:.4f}")
    gain = entropy_total - weighted_entropy
    print(f"Gain of {attribute}: {gain:.4f}")
    return round(gain, 4)

```

```

In [36]: gain_outlook = calc_gain("Outlook")
        gain_windy = calc_gain("Windy")
        gain_humidity = calc_gain("Humidity")
        gain_temperature = calc_gain("Temperature")
        print(gain_outlook, gain_temperature, gain_windy, gain_humidity)

```

Calculating Information Gain for feature: Outlook

Sunny -> Yes: 2, No: 3, Entropy: 0.9710

Overcast -> Yes: 4, No: 0, Entropy: 0.0000

Rainy -> Yes: 3, No: 2, Entropy: 0.9710

Gain of Outlook: 0.2467

Calculating Information Gain for feature: Windy

Weak -> Yes: 6, No: 2, Entropy: 0.8113

Strong -> Yes: 3, No: 3, Entropy: 1.0000

Gain of Windy: 0.0481

Calculating Information Gain for feature: Humidity

High -> Yes: 3, No: 4, Entropy: 0.9852

Normal -> Yes: 6, No: 1, Entropy: 0.5917

Gain of Humidity: 0.1518

Calculating Information Gain for feature: Temperature

Hot -> Yes: 2, No: 2, Entropy: 1.0000

Mild -> Yes: 4, No: 2, Entropy: 0.9183

Cool -> Yes: 3, No: 1, Entropy: 0.8113

Gain of Temperature: 0.0292

0.2467 0.0292 0.0481 0.1518

```

In [16]: gains = {
        'Outlook': gain_outlook,
        'Temperature': gain_temperature,
        'Windy': gain_windy,
        'Humidity': gain_humidity
        }
        print(gains)

```

```
{'Outlook': 0.2467, 'Temperature': 0.0292, 'Windy': 0.0481, 'Humidity': 0.1518}
```

```

In [18]: root_node = max(gains, key=gains.get)
        print(root_node)

```

Outlook