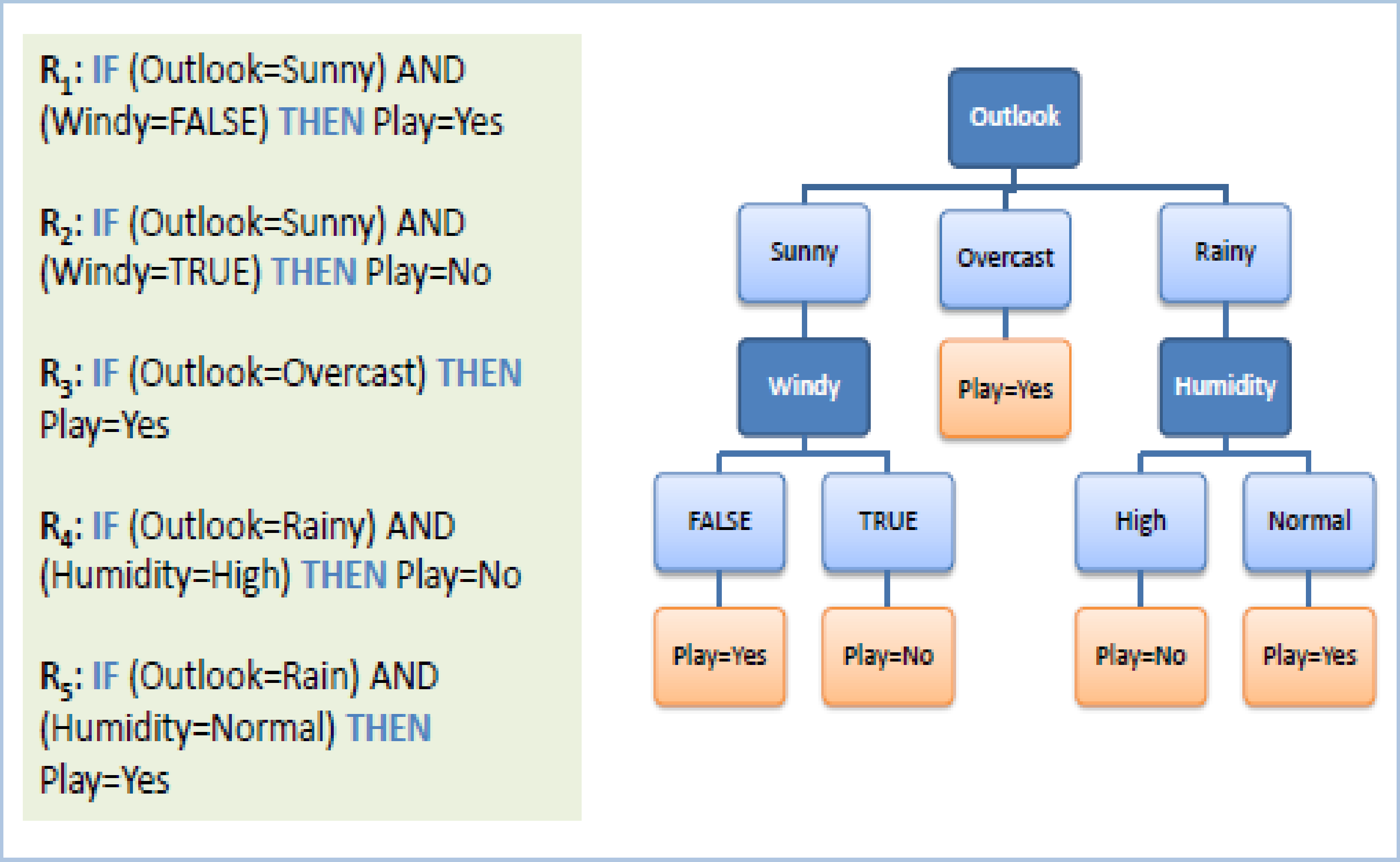
الجامعة الألمانية الأردنية
German Jordanian University

SEEIT

Decision Tree



R_1: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

R_2: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

R_3: IF (Outlook=Overcast) THEN Play=Yes

R_4: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

R_5: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes

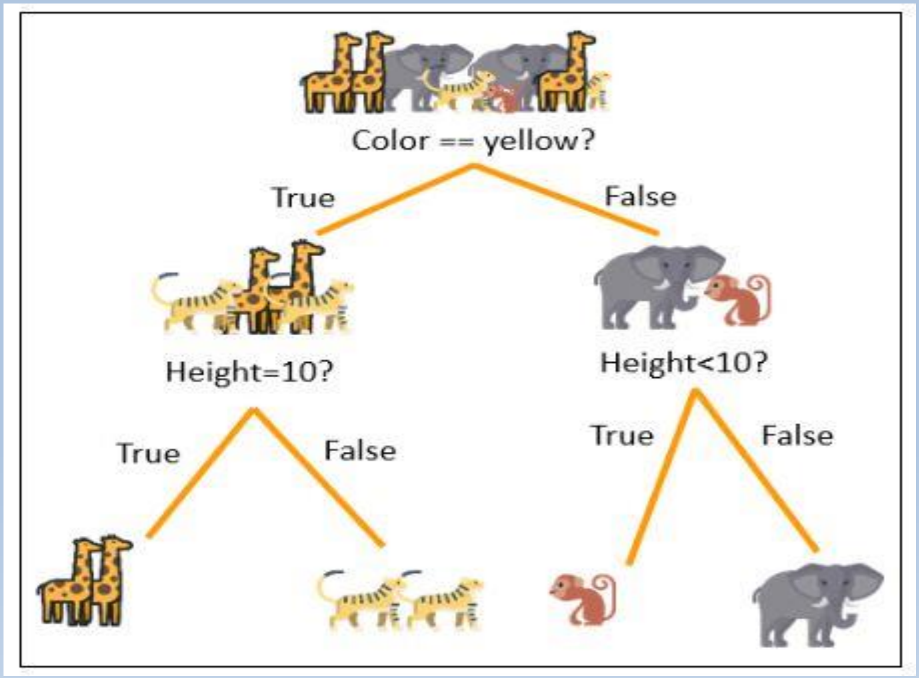
Rana Husni

الجامعة الألمانية الأردنية
German Jordanian University

SEEIT

Decision Tree

A decision tree is **a non-parametric supervised learning algorithm for classification and regression tasks**. It has a hierarchical tree structure consisting of a root node, branches, internal nodes, and leaf nodes. Decision trees are used for classification and regression tasks, providing easy-to-understand models.
A decision tree is a hierarchical model used in decision support that depicts decisions and their potential outcomes, incorporating chance events, resource expenses, and utility. This algorithmic model utilizes conditional control statements and is non-parametric, supervised learning, useful for both classification and regression tasks. The tree structure is comprised of a root node, branches, internal nodes, and leaf nodes, forming a hierarchical, tree-like structure.
It is a tool that has applications spanning several different areas. Decision trees can be used for classification as well as regression problems. The name itself suggests that it uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits. It starts with a root node and ends with a decision made by leaves.



Rana Husni

Decision Tree Terminologies

•***Root Node****:* The initial node at the beginning of a decision tree, where the entire population or dataset starts dividing based on various features or conditions.

•***Decision Nodes****:* Nodes resulting from the splitting of root nodes are known as decision nodes. These nodes represent intermediate decisions or conditions within the tree.

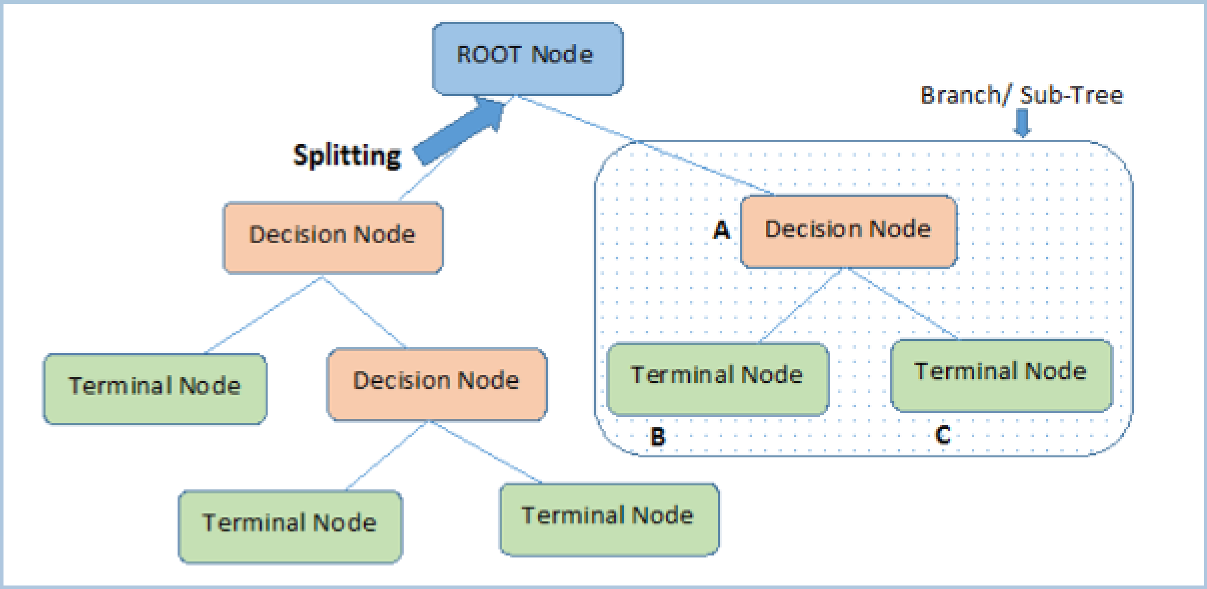
•***Leaf Nodes****:* Nodes where further splitting is not possible, often indicating the final classification or outcome. Leaf nodes are also referred to as terminal nodes.

•***Sub-Tree****:* Similar to a subsection of a graph being called a sub-graph, a sub-section of a decision tree is referred to as a sub-tree. It represents a specific portion of the decision tree.

•***Pruning***: The process of removing or cutting down specific nodes in a decision tree to prevent overfitting and simplify the model.

•***Branch / Sub-Tree****:* A subsection of the entire decision tree is referred to as a branch or sub-tree. It represents a specific path of decisions and outcomes within the tree.

•***Parent and Child Node****:* In a decision tree, a node that is divided into sub-nodes is known as a parent node, and the sub-nodes emerging from it are referred to as child nodes. The parent node represents a decision or condition, while the child nodes represent the potential outcomes or further decisions based on that condition.

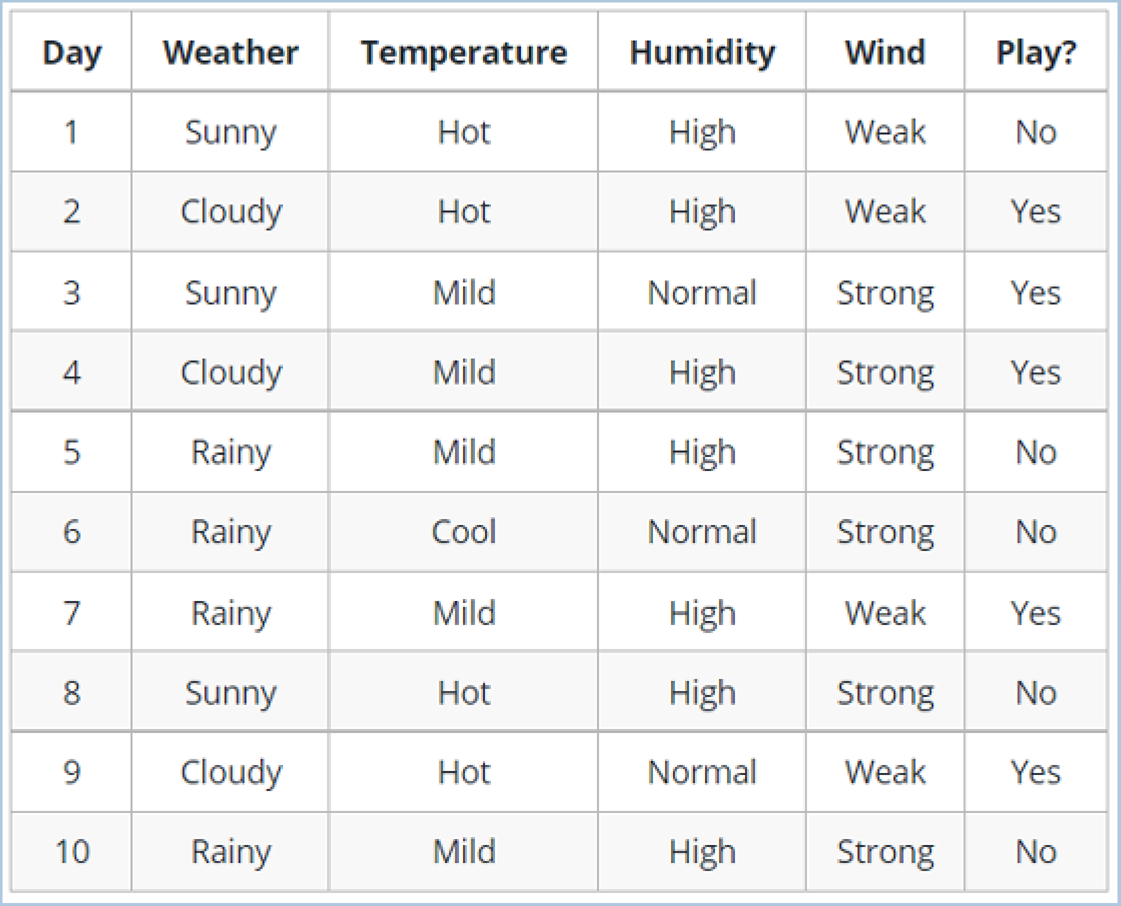


Decision trees are upside down which means the root is at the top and then this root is split into various several nodes. Decision trees are nothing but a bunch of if-else statements in layman terms. It checks if the condition is true and if it is then it goes to the next node attached to that decision.

GJU
الجامعة الألمانية الأردنية
German Jordanian University

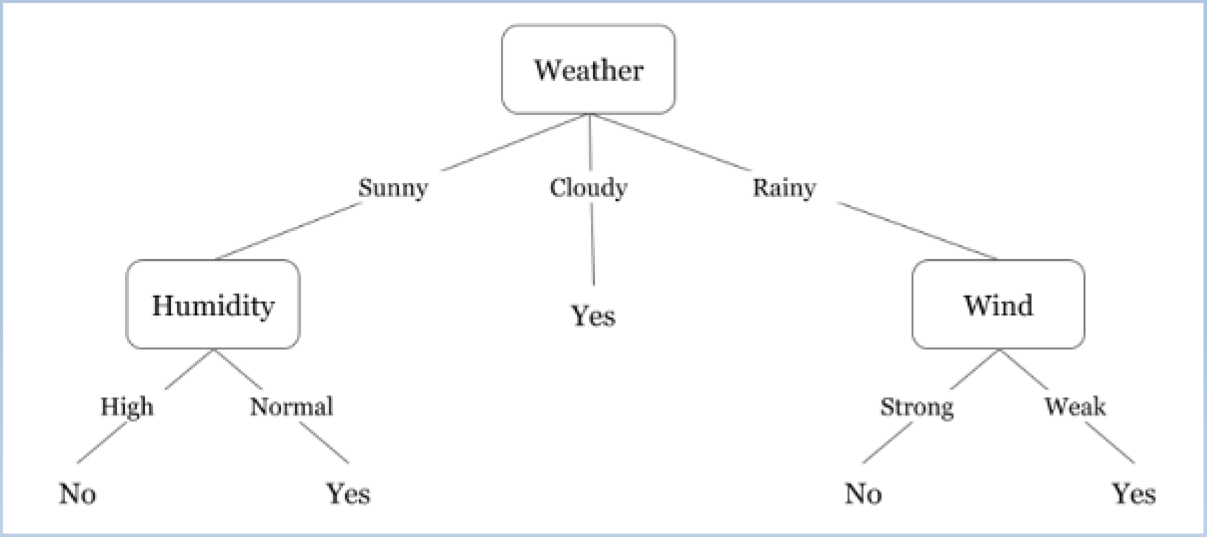
Example of Decision Tree

SEEIT



Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

In the below diagram the tree will first ask what is the weather? Is it sunny, cloudy, or rainy? If yes then it will go to the next feature which is humidity and wind. It will again check if there is a strong wind or weak, if it’s a weak wind and it’s rainy then the person may go and play.



Rana Husni

الجامعة الألمانية الأردنية
German Jordanian University

SEEIT

How decision tree algorithms work?

Decision Tree algorithm works in simpler steps

1.Starting at the Root: The algorithm begins at the top, called the “root node,” representing the entire dataset.

2.Asking the Best Questions: It looks for the most important feature or question that splits the data into the most distinct groups. This is like asking a question at a fork in the tree.

3.Branching Out: Based on the answer to that question, it divides the data into smaller subsets, creating new branches. Each branch represents a possible route through the tree.

4.Repeating the Process: The algorithm continues asking questions and splitting the data at each branch until it reaches the final “leaf nodes,” representing the predicted outcomes or classifications.

Rana Husni

How decision trees are built?
In the Decision tree algorithm, we start at the tree root(root node) and split the data on the feature/predictors that result in maximum information gain.

Steps in building a decision tree.
1.Pick a root node first**.** (How do we do that?) — Calculate the information gain for every feature in the dataset/data. The feature with “largest information given” is selected as the “root node”.

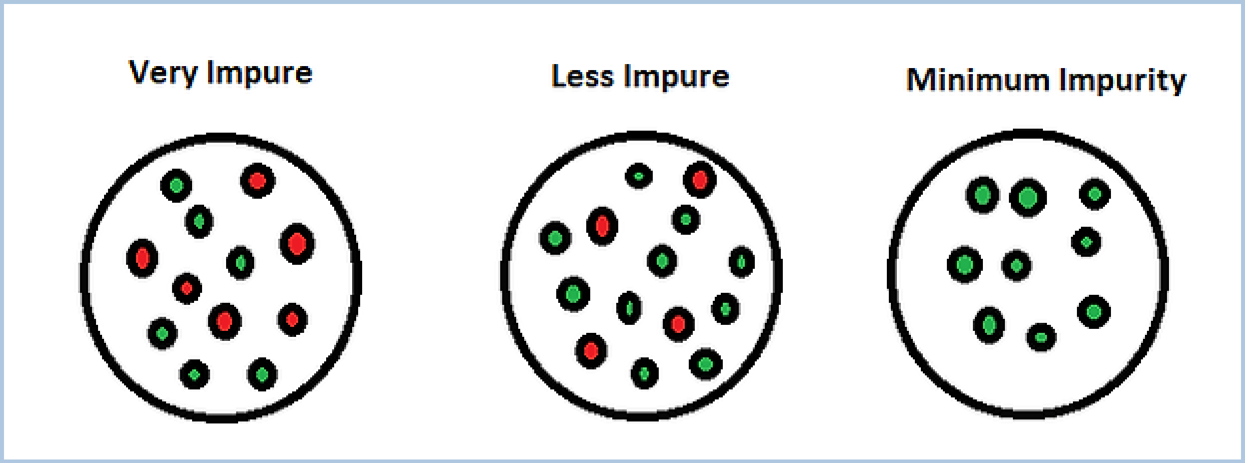
2.Based on the root node, divide your dataset. (How to divide?) — This again depends on the ‘root node’(feature). If it has three different categories, then you will divide your dataset into three or If the features are numerical, then you will divide based on comparison operators (>,<,=).

3.After breaking ***datasets into smaller subsets, calculate the information gain for remaining features other than the root node feature.*** The feature with the highest information gain becomes the “decision node” or “child node.”

4.Repeat *‘step-2’* and *‘step-3*’ until you get leaf nodes (or) pure nodes, this means all the training examples at each node, belong to the same class.

What is Entropy?

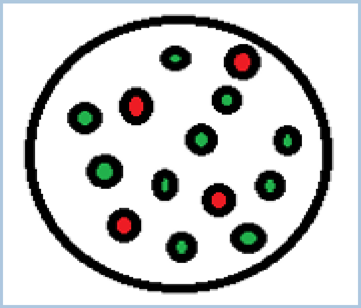
Entropy is the measures of impurity, disorder or uncertainty in a bunch of examples. Entropy controls how a Decision Tree decides to split the data. The below image shows impurity level of each set.



If we have a set of K different values , then we can calculate the entropy using this formula:

$$entropy = -\sum_{i=1}^{k} P(value_i) \cdot \log_2\left(P(value_i)\right)$$

where, P(value*i*) is the probability of getting the *i*th value when randomly selecting one from the set.

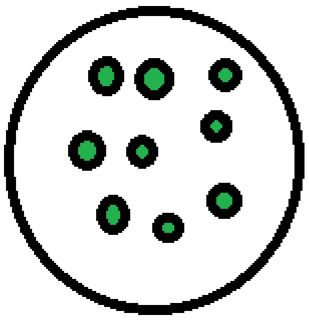


Entropy(Green) = $\log_2(10/14)$ = -0.4854268272

Entropy(Red) = $\log_2(4/14)$ = -1.807354922

Entropy (Set) = -(10/14)(-0.4854268272) –(4/14)(-1.807354922) = 0.86

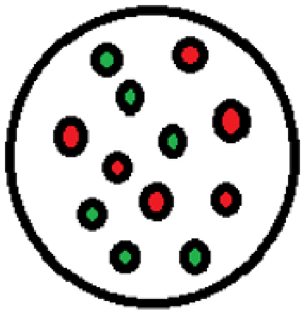
Minimum Impurity



Entropy = - 1 $\log_2 1$ = 0

This is not a good dataset for learning.

Very Impure



Entropy = -0.5 $\log_2 0.5$ – 0.5 $\log_2 0.5$ =1

Such a set is good for learning.

What is Information Gain?

Information gain (IG) measures how much “information” a feature gives us about the class.It tells us how important a given attribute of the feature vectors is. **Information gain (IG)** is used to decide the ordering of attributes in the nodes of a decision tree.

Information gain (IG) is calculated as follows:

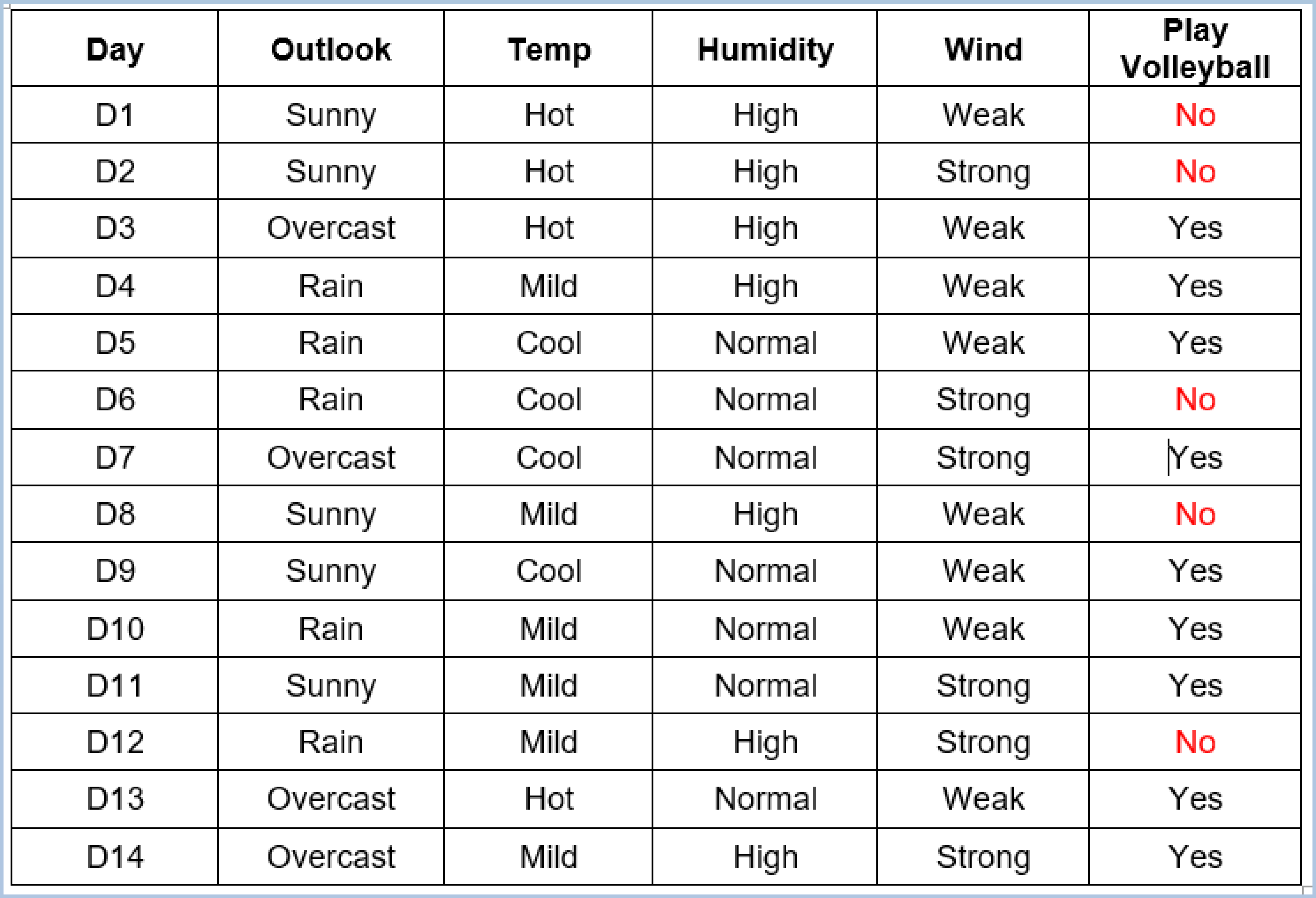
Information Gain = entropy(parent) – [average entropy(children)]

The entropy may be calculated using the formula below:

$$E = -\sum_{i=1}^{N} p_i log_2 p_i$$

Example

Suppose we had the following dataset:



Day	Outlook	Temp	Humidity	Wind	Play Volleyball
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	Rain	Mild	High	Strong	No
D13	Overcast	Hot	Normal	Weak	Yes
D14	Overcast	Mild	High	Strong	Yes

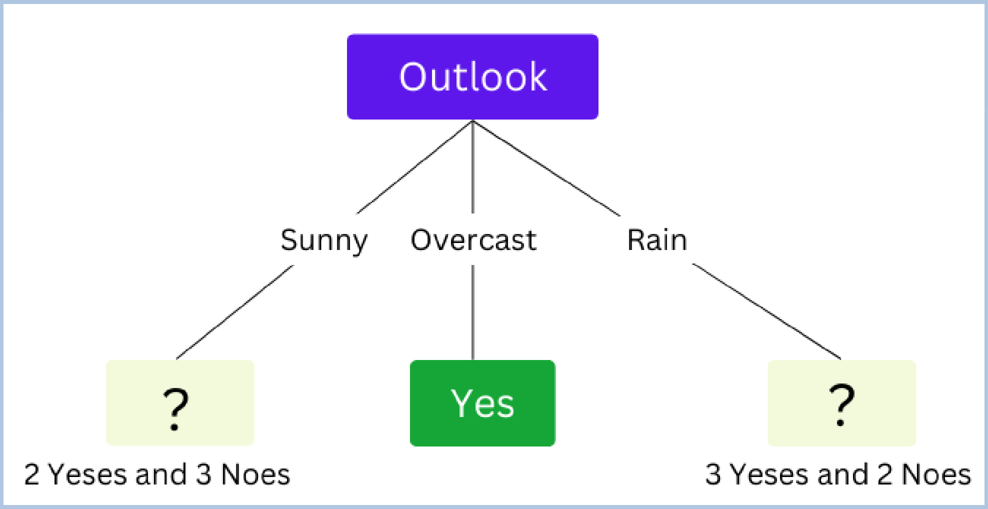
From the above example dataset, we are required to construct a decision tree to help us decide whether we should play volleyball based on the weather conditions.

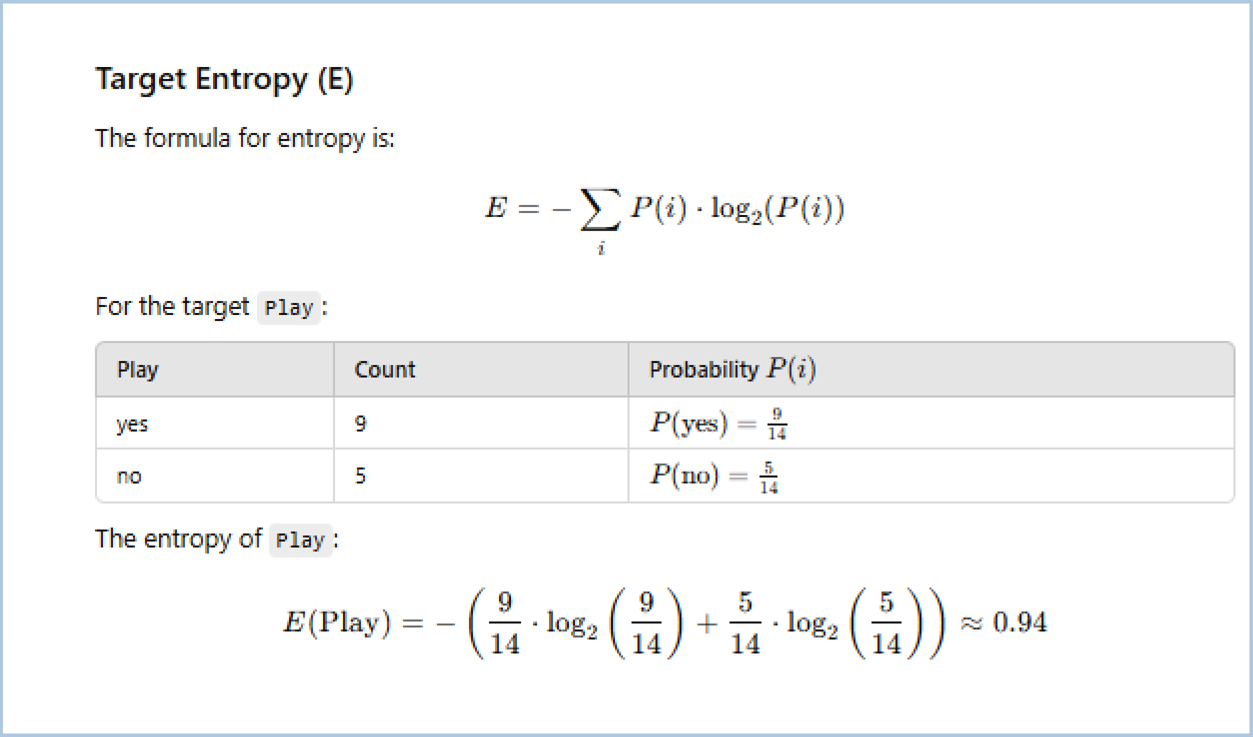
Finding the root node feature

Since we can not just pick one of the features to start our decision tree, we need to make calculations to get the feature with the highest information gain from which we start splitting.

All information gain values will be:
- Gain(*S*, *Outlook*) = 0.25
- Gain(*S*, *Temp*) = 0.03
- Gain(*S*, *Humidity*) = 0.15
- Gain(*S*, *Wind*) = 0.04

Outlook gives the highest information about our target variable from the information gain values. It will act as the **root node** of our tree from where the splitting will begin.





Target Entropy (E)

The formula for entropy is:

$$E = -\sum_{i} P(i) \cdot \log_2(P(i))$$

For the target `Play`:

Play	Count	Probability $P(i)$
yes	9	$P(\text{yes}) = \frac{9}{14}$
no	5	$P(\text{no}) = \frac{5}{14}$

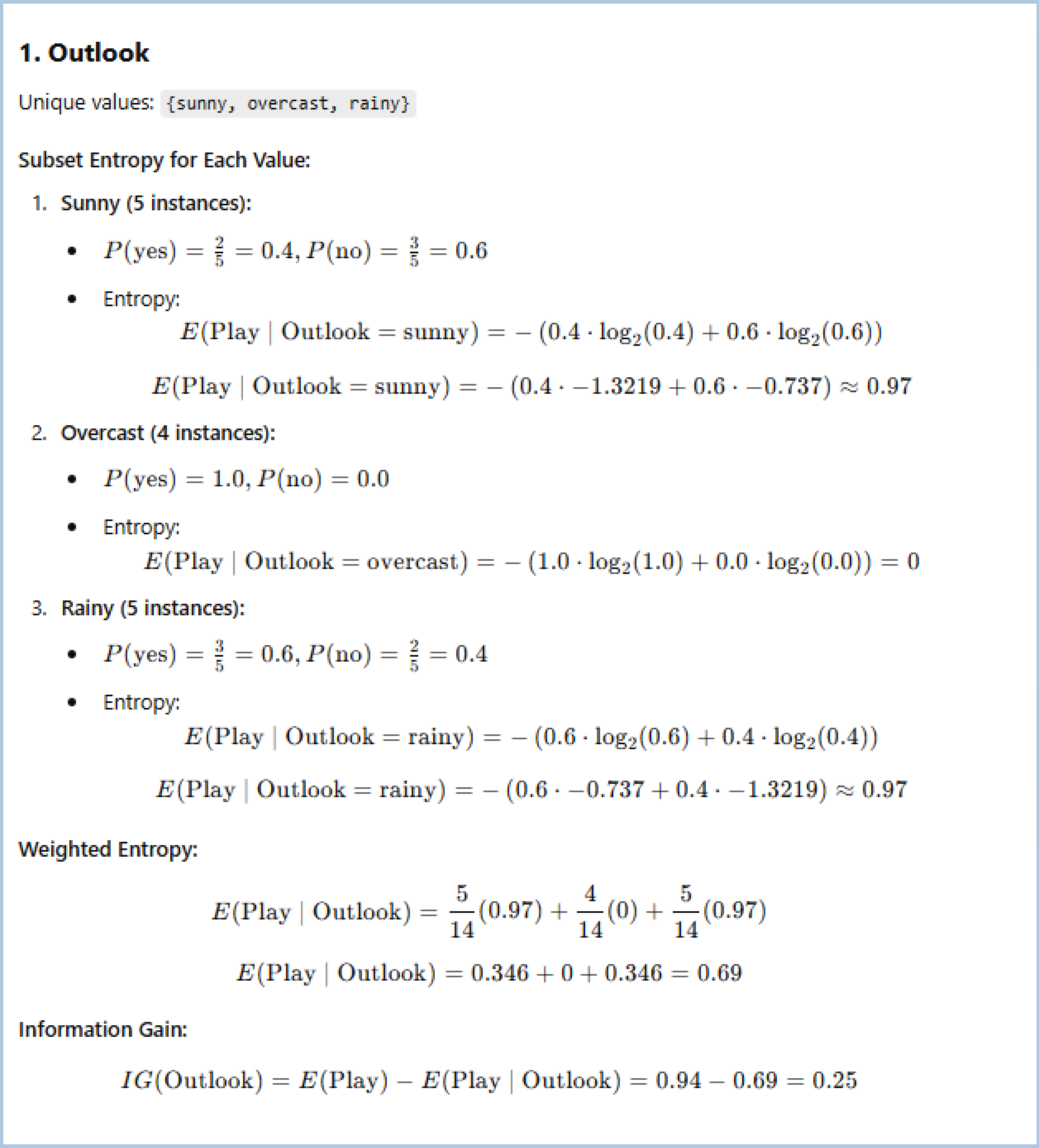
The entropy of `Play`:

$$E(\text{Play}) = -\left(\frac{9}{14} \cdot \log_2\left(\frac{9}{14}\right) + \frac{5}{14} \cdot \log_2\left(\frac{5}{14}\right)\right) \approx 0.94$$

الجامعة الألمانية الأردنية
German Jordanian University

SEEIT

Calculate the information gain for all features:



1. Outlook

Unique values: `{sunny, overcast, rainy}`

Subset Entropy for Each Value:

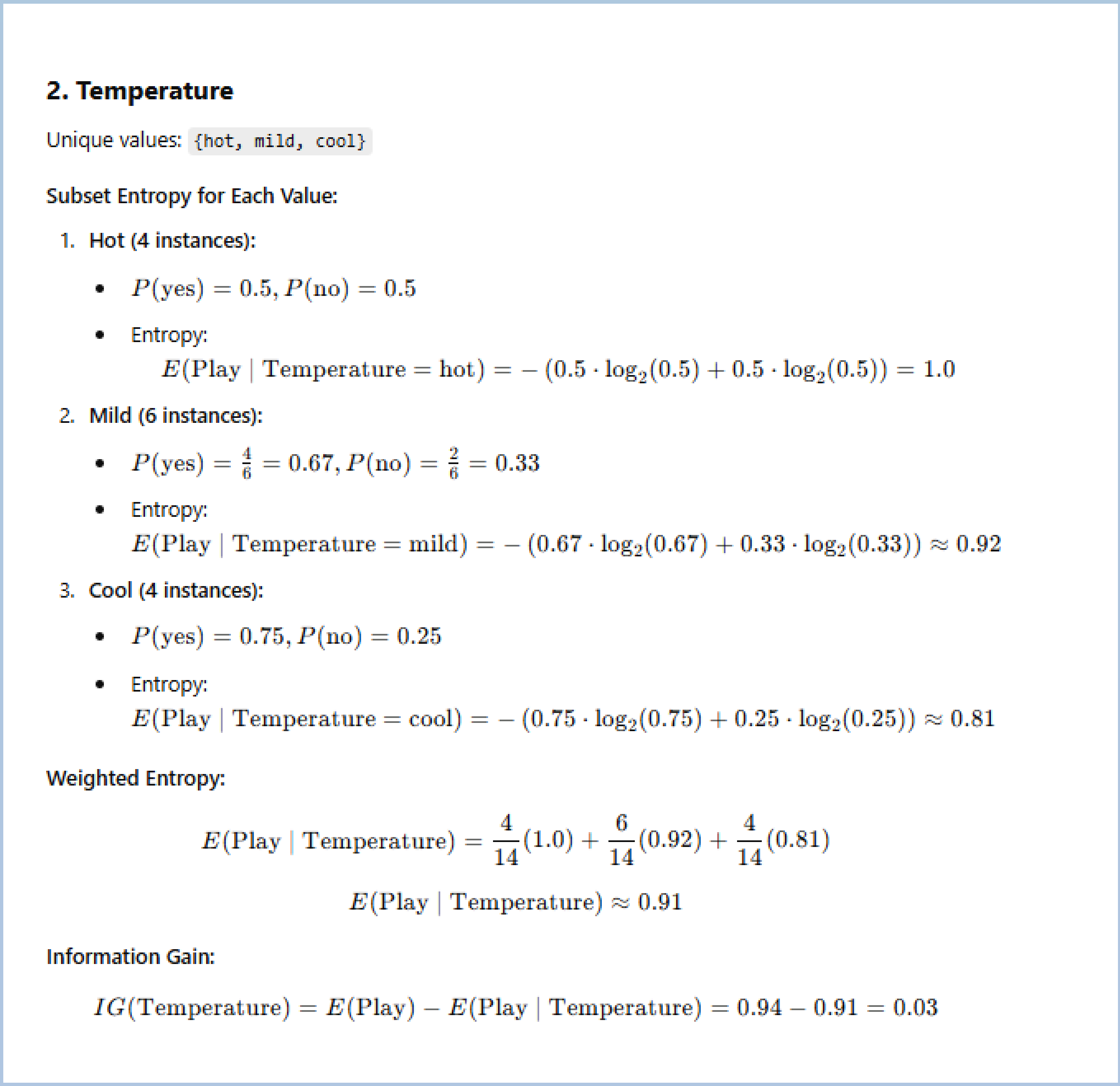
1. **Sunny (5 instances):**
   - $P(\text{yes}) = \frac{2}{5} = 0.4, P(\text{no}) = \frac{3}{5} = 0.6$
   - Entropy:
     $$E(\text{Play} \mid \text{Outlook} = \text{sunny}) = -\left(0.4 \cdot \log_2(0.4) + 0.6 \cdot \log_2(0.6)\right)$$
     $$E(\text{Play} \mid \text{Outlook} = \text{sunny}) = -\left(0.4 \cdot -1.3219 + 0.6 \cdot -0.737\right) \approx 0.97$$
2. **Overcast (4 instances):**
   - $P(\text{yes}) = 1.0, P(\text{no}) = 0.0$
   - Entropy:
     $$E(\text{Play} \mid \text{Outlook} = \text{overcast}) = -\left(1.0 \cdot \log_2(1.0) + 0.0 \cdot \log_2(0.0)\right) = 0$$
3. **Rainy (5 instances):**
   - $P(\text{yes}) = \frac{3}{5} = 0.6, P(\text{no}) = \frac{2}{5} = 0.4$
   - Entropy:
     $$E(\text{Play} \mid \text{Outlook} = \text{rainy}) = -\left(0.6 \cdot \log_2(0.6) + 0.4 \cdot \log_2(0.4)\right)$$
     $$E(\text{Play} \mid \text{Outlook} = \text{rainy}) = -\left(0.6 \cdot -0.737 + 0.4 \cdot -1.3219\right) \approx 0.97$$

Weighted Entropy:

$$E(\text{Play} \mid \text{Outlook}) = \frac{5}{14}(0.97) + \frac{4}{14}(0) + \frac{5}{14}(0.97)$$
$$E(\text{Play} \mid \text{Outlook}) = 0.346 + 0 + 0.346 = 0.69$$

Information Gain:

$$IG(\text{Outlook}) = E(\text{Play}) - E(\text{Play} \mid \text{Outlook}) = 0.94 - 0.69 = 0.25$$



2. Temperature

Unique values: `{hot, mild, cool}`

Subset Entropy for Each Value:

1. **Hot (4 instances):**
   - $P(\text{yes}) = 0.5, P(\text{no}) = 0.5$
   - Entropy:
     $E(\text{Play} \mid \text{Temperature} = \text{hot}) = -(0.5 \cdot \log_2(0.5) + 0.5 \cdot \log_2(0.5)) = 1.0$
2. **Mild (6 instances):**
   - $P(\text{yes}) = \frac{4}{6} = 0.67, P(\text{no}) = \frac{2}{6} = 0.33$
   - Entropy:
     $E(\text{Play} \mid \text{Temperature} = \text{mild}) = -(0.67 \cdot \log_2(0.67) + 0.33 \cdot \log_2(0.33)) \approx 0.92$
3. **Cool (4 instances):**
   - $P(\text{yes}) = 0.75, P(\text{no}) = 0.25$
   - Entropy:
     $E(\text{Play} \mid \text{Temperature} = \text{cool}) = -(0.75 \cdot \log_2(0.75) + 0.25 \cdot \log_2(0.25)) \approx 0.81$

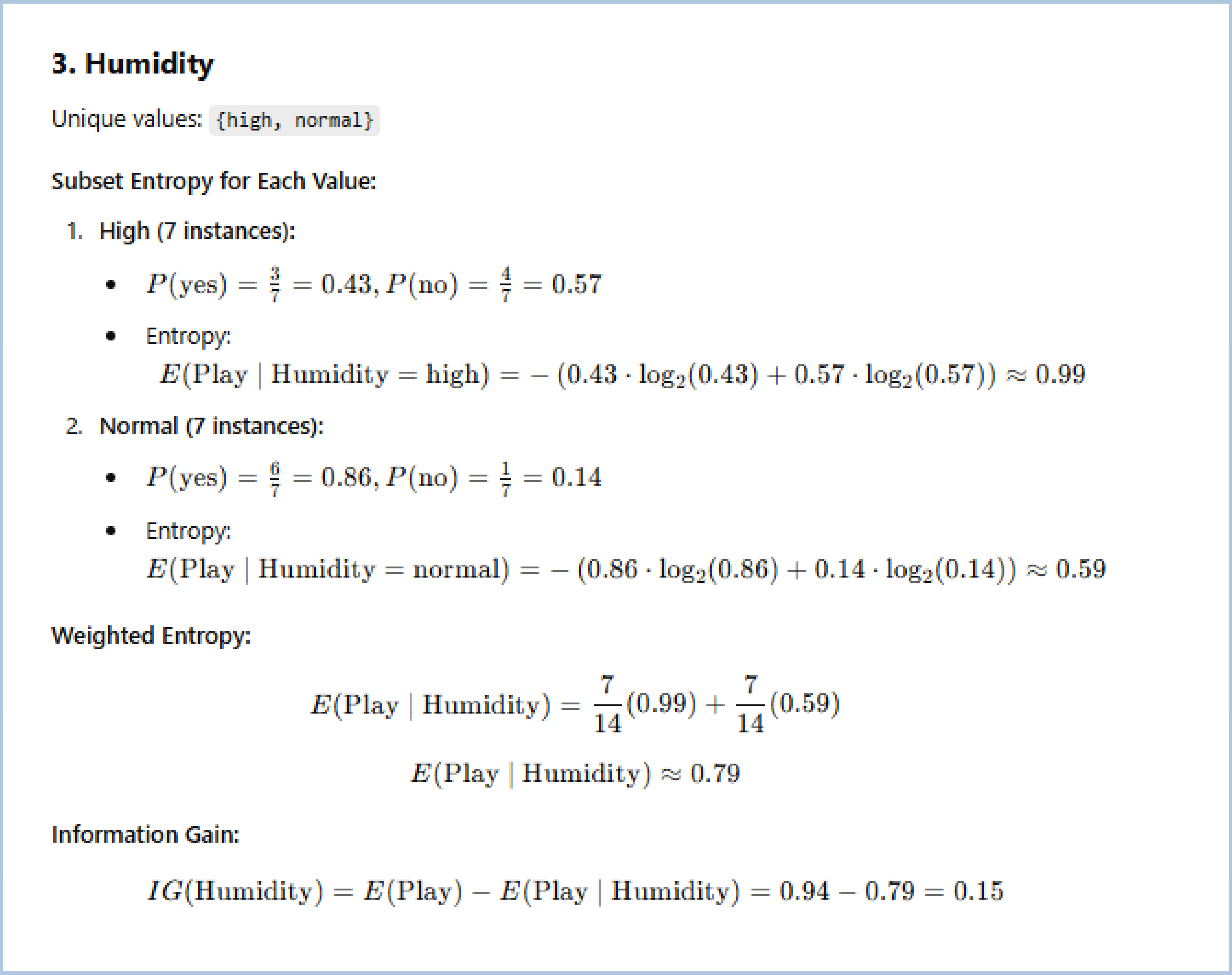
Weighted Entropy:

$$E(\text{Play} \mid \text{Temperature}) = \frac{4}{14}(1.0) + \frac{6}{14}(0.92) + \frac{4}{14}(0.81)$$

$$E(\text{Play} \mid \text{Temperature}) \approx 0.91$$

Information Gain:

$$IG(\text{Temperature}) = E(\text{Play}) - E(\text{Play} \mid \text{Temperature}) = 0.94 - 0.91 = 0.03$$



3. Humidity

Unique values: `{high, normal}`

Subset Entropy for Each Value:

1. **High (7 instances):**
   - $P(\text{yes}) = \frac{3}{7} = 0.43, P(\text{no}) = \frac{4}{7} = 0.57$
   - Entropy:
     $E(\text{Play} \mid \text{Humidity} = \text{high}) = -(0.43 \cdot \log_2(0.43) + 0.57 \cdot \log_2(0.57)) \approx 0.99$
2. **Normal (7 instances):**
   - $P(\text{yes}) = \frac{6}{7} = 0.86, P(\text{no}) = \frac{1}{7} = 0.14$
   - Entropy:
     $E(\text{Play} \mid \text{Humidity} = \text{normal}) = -(0.86 \cdot \log_2(0.86) + 0.14 \cdot \log_2(0.14)) \approx 0.59$

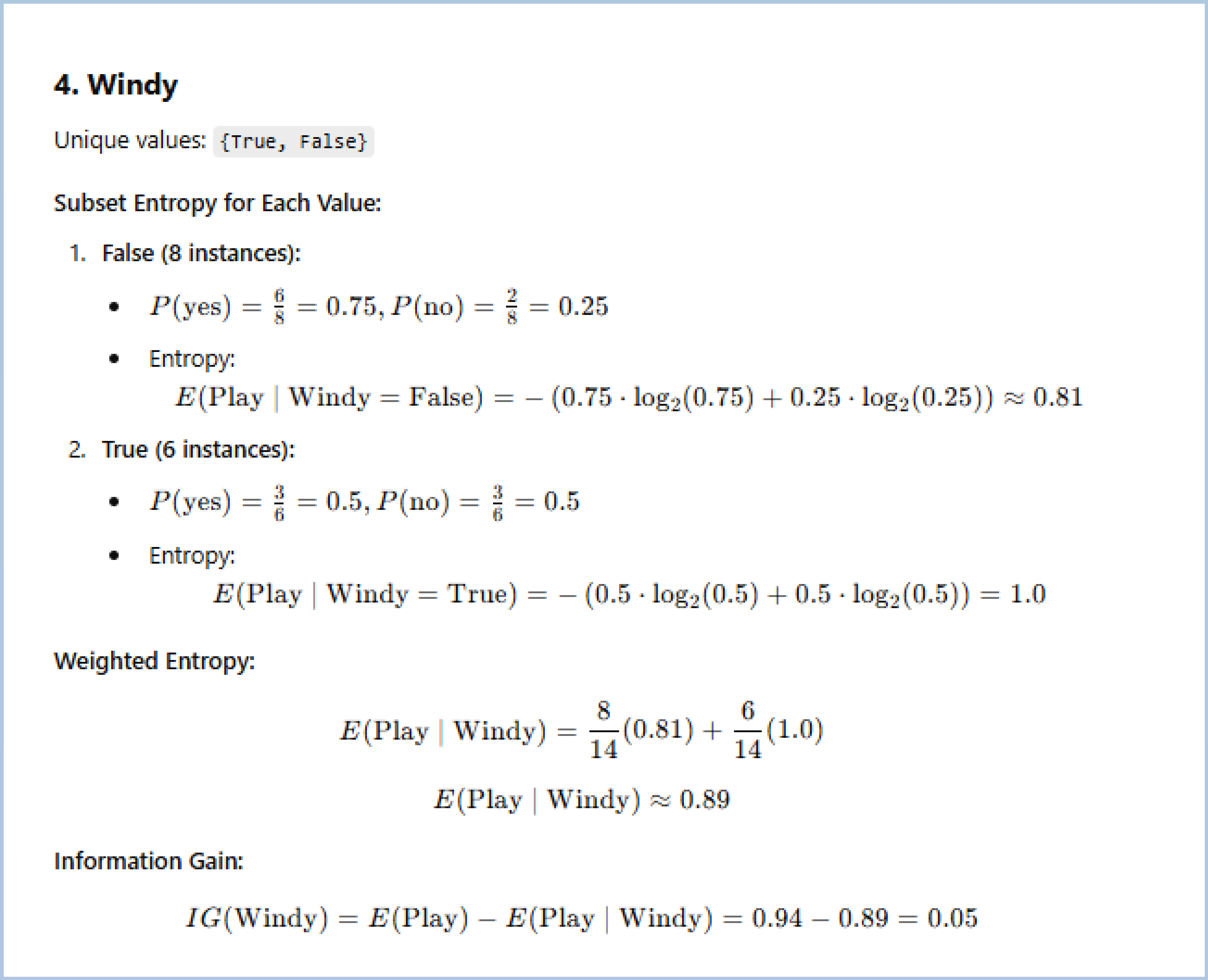
Weighted Entropy:

$$E(\text{Play} \mid \text{Humidity}) = \frac{7}{14}(0.99) + \frac{7}{14}(0.59)$$

$$E(\text{Play} \mid \text{Humidity}) \approx 0.79$$

Information Gain:

$$IG(\text{Humidity}) = E(\text{Play}) - E(\text{Play} \mid \text{Humidity}) = 0.94 - 0.79 = 0.15$$



4. Windy

Unique values: `{True, False}`

Subset Entropy for Each Value:

1. **False (8 instances):**
   - $P(\text{yes}) = \frac{6}{8} = 0.75, P(\text{no}) = \frac{2}{8} = 0.25$
   - Entropy:
     $E(\text{Play} \mid \text{Windy} = \text{False}) = -(0.75 \cdot \log_2(0.75) + 0.25 \cdot \log_2(0.25)) \approx 0.81$
2. **True (6 instances):**
   - $P(\text{yes}) = \frac{3}{6} = 0.5, P(\text{no}) = \frac{3}{6} = 0.5$
   - Entropy:
     $$E(\text{Play} \mid \text{Windy} = \text{True}) = -(0.5 \cdot \log_2(0.5) + 0.5 \cdot \log_2(0.5)) = 1.0$$

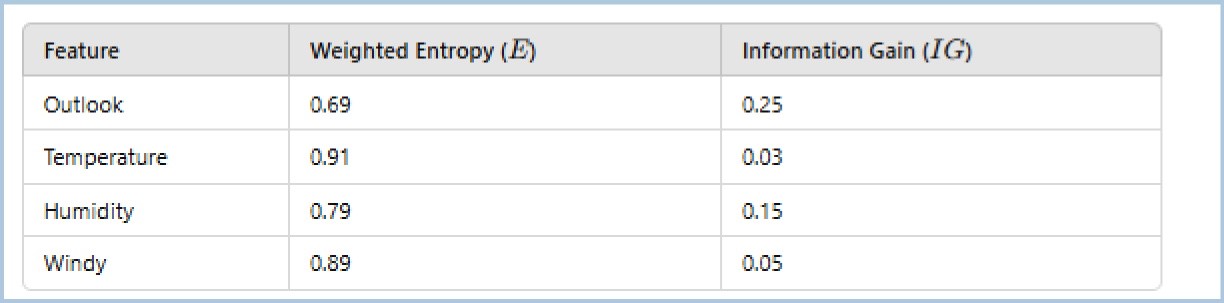
Weighted Entropy:

$$E(\text{Play} \mid \text{Windy}) = \frac{8}{14}(0.81) + \frac{6}{14}(1.0)$$

$$E(\text{Play} \mid \text{Windy}) \approx 0.89$$

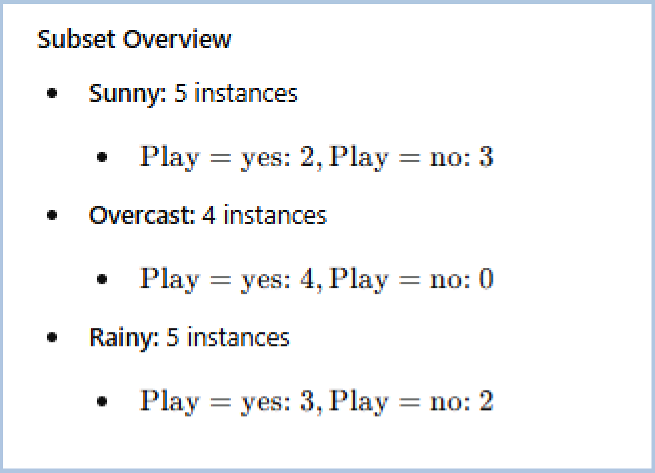
Information Gain:

$$IG(\text{Windy}) = E(\text{Play}) - E(\text{Play} \mid \text{Windy}) = 0.94 - 0.89 = 0.05$$



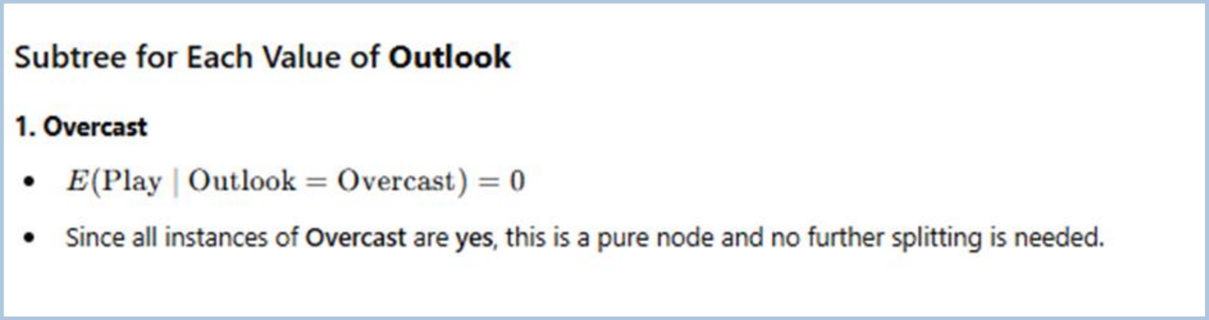
Feature	Weighted Entropy (E)	Information Gain (IG)
Outlook	0.69	0.25
Temperature	0.91	0.03
Humidity	0.79	0.15
Windy	0.89	0.05

Decision Tree Split After Selecting Outlook Since Outlook has the highest Information Gain (IG=0.25), it is chosen as the root node. The data is now split into subsets based on the values of Outlook: {sunny, overcast, rainy}.



Subset Overview

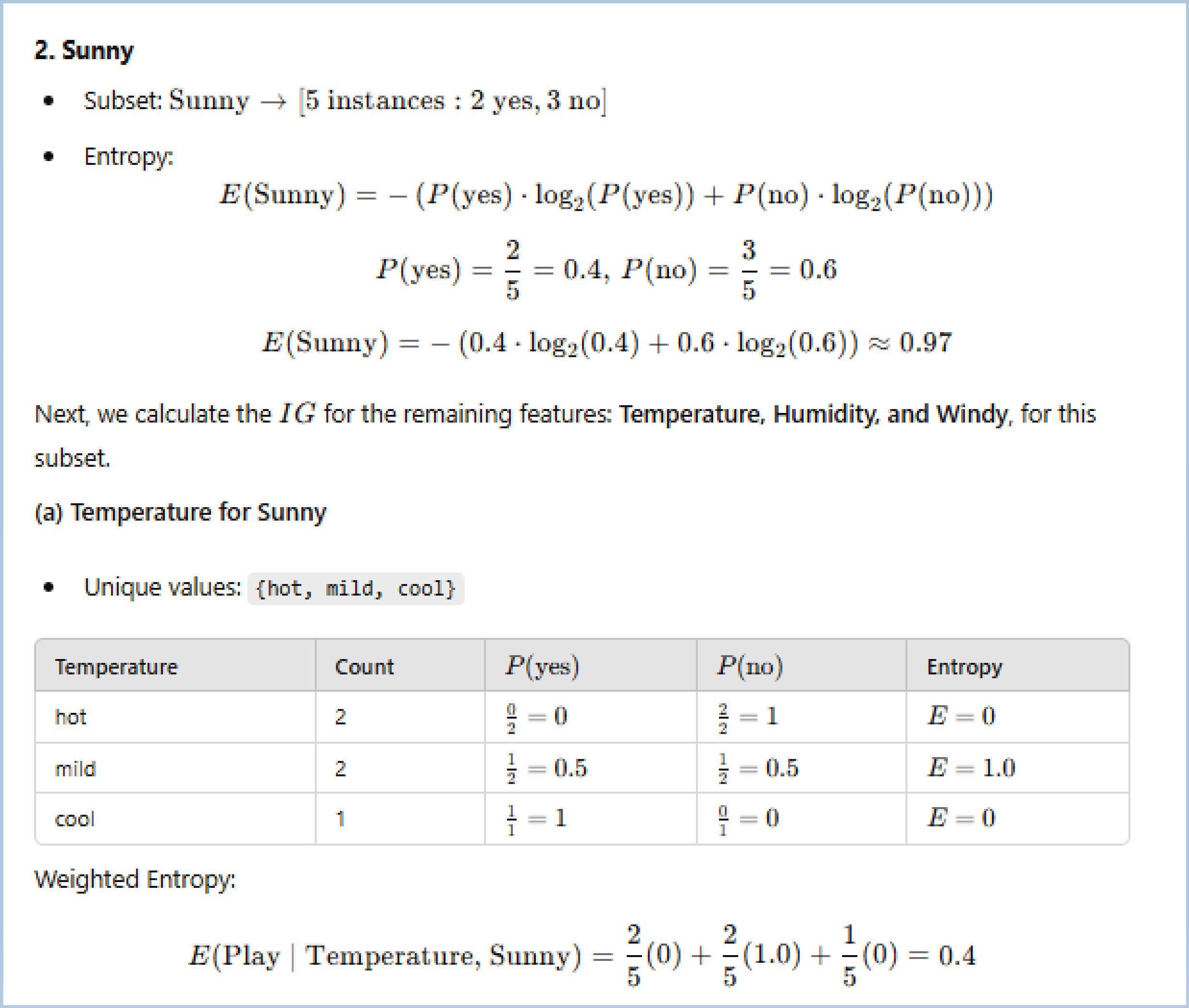
- **Sunny:** 5 instances
  - $\text{Play} = \text{yes}: 2, \text{Play} = \text{no}: 3$
- **Overcast:** 4 instances
  - $\text{Play} = \text{yes}: 4, \text{Play} = \text{no}: 0$
- **Rainy:** 5 instances
  - $\text{Play} = \text{yes}: 3, \text{Play} = \text{no}: 2$



Subtree for Each Value of **Outlook**

1. Overcast

- $E(\text{Play} \mid \text{Outlook} = \text{Overcast}) = 0$
- Since all instances of **Overcast** are **yes**, this is a pure node and no further splitting is needed.



2. Sunny

- Subset: $\text{Sunny} \rightarrow [5 \text{ instances} : 2 \text{ yes}, 3 \text{ no}]$
- Entropy:

$$E(\text{Sunny}) = -\left(P(\text{yes}) \cdot \log_2(P(\text{yes})) + P(\text{no}) \cdot \log_2(P(\text{no}))\right)$$

$$P(\text{yes}) = \frac{2}{5} = 0.4,\ P(\text{no}) = \frac{3}{5} = 0.6$$

$$E(\text{Sunny}) = -\left(0.4 \cdot \log_2(0.4) + 0.6 \cdot \log_2(0.6)\right) \approx 0.97$$

Next, we calculate the IG for the remaining features: **Temperature, Humidity, and Windy**, for this subset.

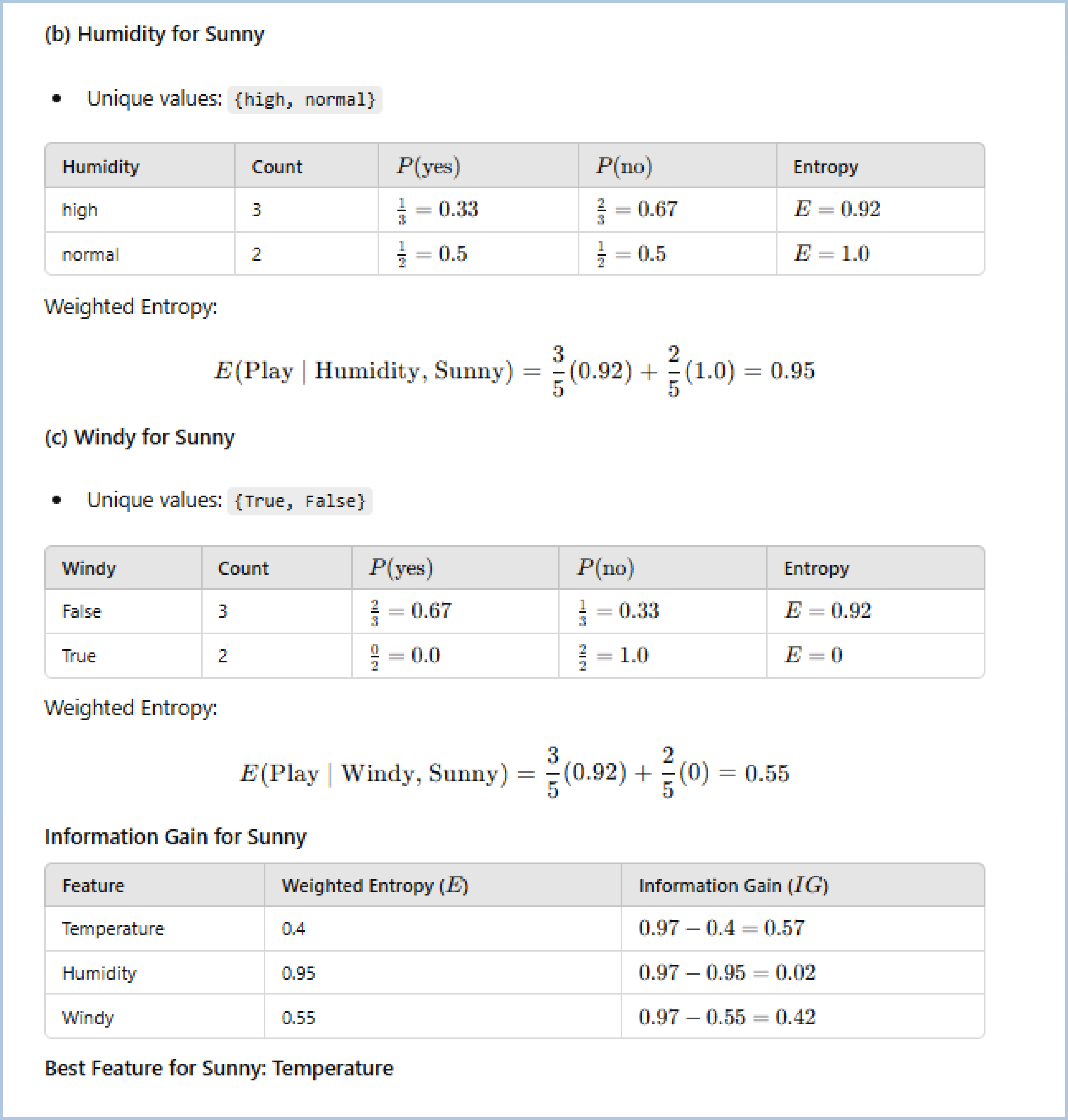
(a) Temperature for Sunny

- Unique values: `{hot, mild, cool}`

Temperature	Count	$P(\text{yes})$	$P(\text{no})$	Entropy
hot	2	$\frac{0}{2} = 0$	$\frac{2}{2} = 1$	$E = 0$
mild	2	$\frac{1}{2} = 0.5$	$\frac{1}{2} = 0.5$	$E = 1.0$
cool	1	$\frac{1}{1} = 1$	$\frac{0}{1} = 0$	$E = 0$

Weighted Entropy:

$$E(\text{Play} \mid \text{Temperature, Sunny}) = \frac{2}{5}(0) + \frac{2}{5}(1.0) + \frac{1}{5}(0) = 0.4$$



(b) Humidity for Sunny

- Unique values: `{high, normal}`

Humidity	Count	$P(\text{yes})$	$P(\text{no})$	Entropy
high	3	$\frac{1}{3} = 0.33$	$\frac{2}{3} = 0.67$	$E = 0.92$
normal	2	$\frac{1}{2} = 0.5$	$\frac{1}{2} = 0.5$	$E = 1.0$

Weighted Entropy:

$$E(\text{Play} \mid \text{Humidity, Sunny}) = \frac{3}{5}(0.92) + \frac{2}{5}(1.0) = 0.95$$

(c) Windy for Sunny

- Unique values: `{True, False}`

Windy	Count	$P(\text{yes})$	$P(\text{no})$	Entropy
False	3	$\frac{2}{3} = 0.67$	$\frac{1}{3} = 0.33$	$E = 0.92$
True	2	$\frac{0}{2} = 0.0$	$\frac{2}{2} = 1.0$	$E = 0$

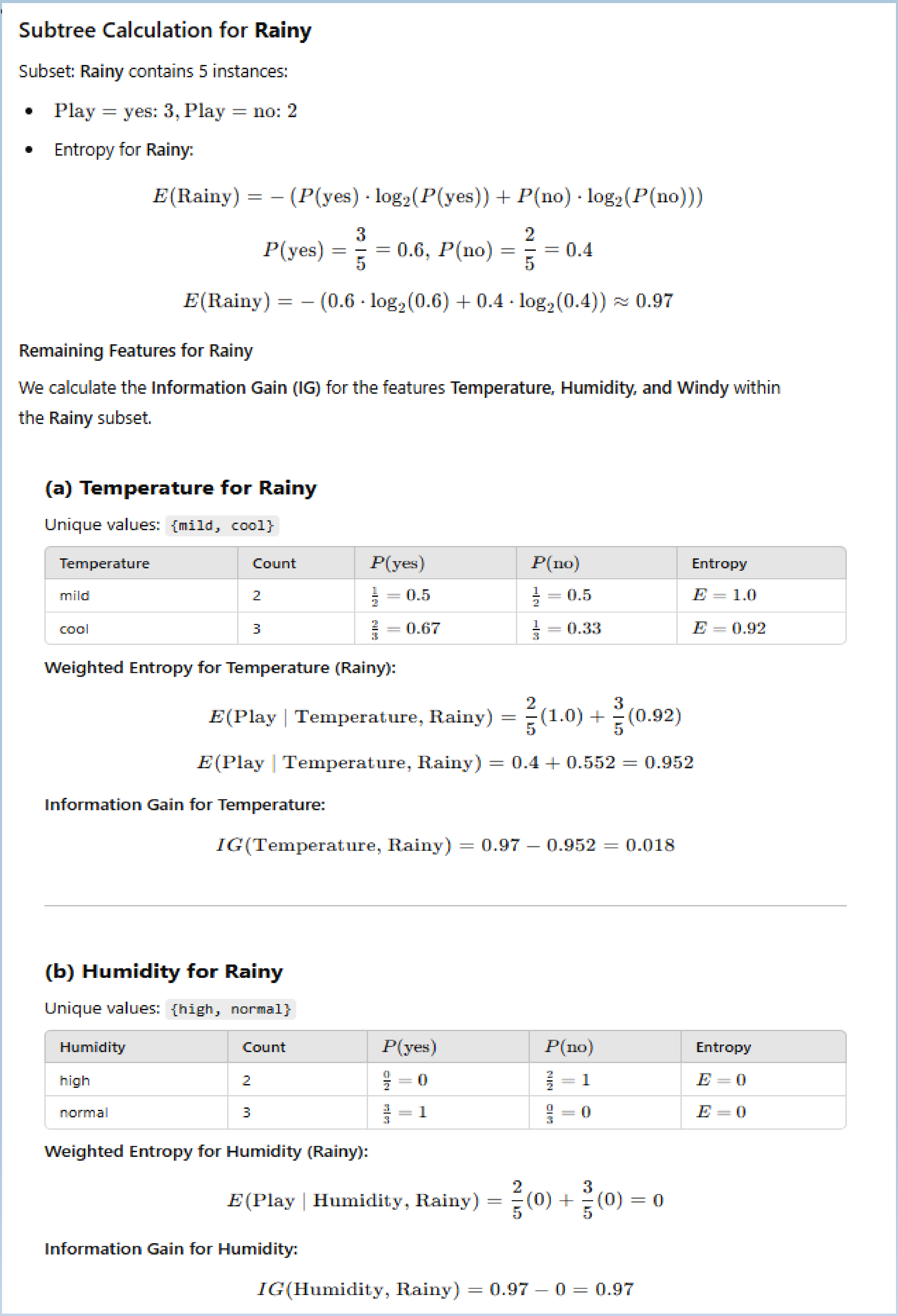
Weighted Entropy:

$$E(\text{Play} \mid \text{Windy, Sunny}) = \frac{3}{5}(0.92) + \frac{2}{5}(0) = 0.55$$

Information Gain for Sunny

Feature	Weighted Entropy (E)	Information Gain (IG)
Temperature	0.4	$0.97 - 0.4 = 0.57$
Humidity	0.95	$0.97 - 0.95 = 0.02$
Windy	0.55	$0.97 - 0.55 = 0.42$

Best Feature for Sunny: Temperature



Subtree Calculation for **Rainy**

Subset: **Rainy** contains 5 instances:

- $\text{Play} = \text{yes: } 3, \text{Play} = \text{no: } 2$
- Entropy for **Rainy**:

$$E(\text{Rainy}) = -\left(P(\text{yes}) \cdot \log_2(P(\text{yes})) + P(\text{no}) \cdot \log_2(P(\text{no}))\right)$$

$$P(\text{yes}) = \frac{3}{5} = 0.6,\ P(\text{no}) = \frac{2}{5} = 0.4$$

$$E(\text{Rainy}) = -\left(0.6 \cdot \log_2(0.6) + 0.4 \cdot \log_2(0.4)\right) \approx 0.97$$

Remaining Features for Rainy

We calculate the **Information Gain (IG)** for the features **Temperature, Humidity, and Windy** within the **Rainy** subset.

(a) Temperature for Rainy

Unique values: `{mild, cool}`

Temperature	Count	$P(\text{yes})$	$P(\text{no})$	Entropy
mild	2	$\frac{1}{2} = 0.5$	$\frac{1}{2} = 0.5$	$E = 1.0$
cool	3	$\frac{2}{3} = 0.67$	$\frac{1}{3} = 0.33$	$E = 0.92$

Weighted Entropy for Temperature (Rainy):

$$E(\text{Play} \mid \text{Temperature, Rainy}) = \frac{2}{5}(1.0) + \frac{3}{5}(0.92)$$

$$E(\text{Play} \mid \text{Temperature, Rainy}) = 0.4 + 0.552 = 0.952$$

Information Gain for Temperature:

$$IG(\text{Temperature, Rainy}) = 0.97 - 0.952 = 0.018$$

(b) Humidity for Rainy

Unique values: `{high, normal}`

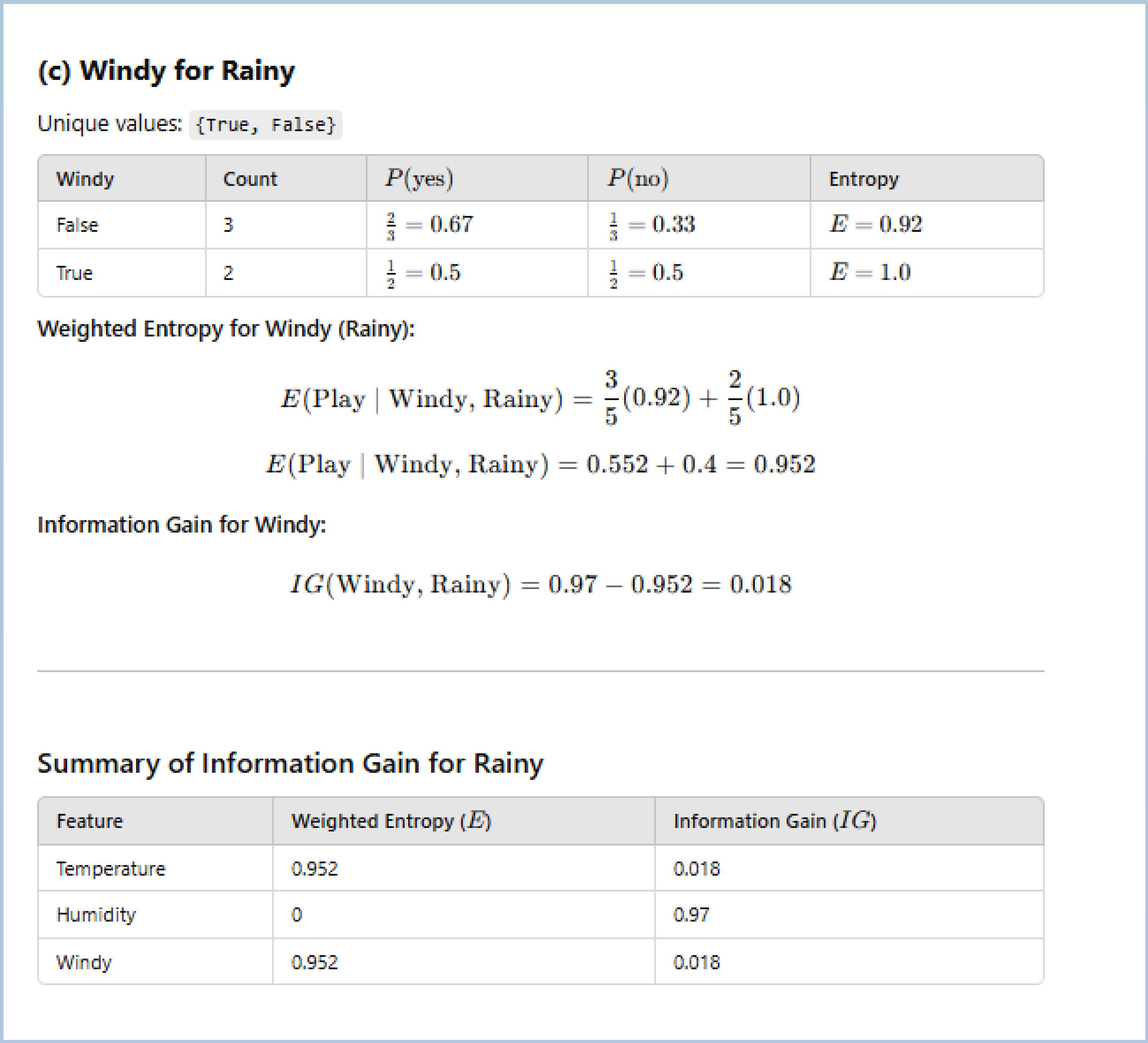
Humidity	Count	$P(\text{yes})$	$P(\text{no})$	Entropy
high	2	$\frac{0}{2} = 0$	$\frac{2}{2} = 1$	$E = 0$
normal	3	$\frac{3}{3} = 1$	$\frac{0}{3} = 0$	$E = 0$

Weighted Entropy for Humidity (Rainy):

$$E(\text{Play} \mid \text{Humidity, Rainy}) = \frac{2}{5}(0) + \frac{3}{5}(0) = 0$$

Information Gain for Humidity:

$$IG(\text{Humidity, Rainy}) = 0.97 - 0 = 0.97$$



(c) Windy for Rainy

Unique values: `{True, False}`

Windy	Count	$P(\text{yes})$	$P(\text{no})$	Entropy
False	3	$\frac{2}{3} = 0.67$	$\frac{1}{3} = 0.33$	$E = 0.92$
True	2	$\frac{1}{2} = 0.5$	$\frac{1}{2} = 0.5$	$E = 1.0$

Weighted Entropy for Windy (Rainy):

$$E(\text{Play} \mid \text{Windy, Rainy}) = \frac{3}{5}(0.92) + \frac{2}{5}(1.0)$$

$$E(\text{Play} \mid \text{Windy, Rainy}) = 0.552 + 0.4 = 0.952$$

Information Gain for Windy:

$$IG(\text{Windy, Rainy}) = 0.97 - 0.952 = 0.018$$

Summary of Information Gain for Rainy

Feature	Weighted Entropy (E)	Information Gain (IG)
Temperature	0.952	0.018
Humidity	0	0.97
Windy	0.952	0.018

Next Selection for Rainy

Since **Humidity** has the highest Information Gain (IG=0.97), it is chosen as the next splitting feature for the **Rainy** subset.

GJU
الجامعة الألمانية الأردنية
German Jordanian University

Train Test Split

SEEIT

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score

# Load dataset
data = pd.read_excel('WheatherTF.xlsx')

# Feature columns and target column
features = data.drop('Play', axis=1)
target = data['Play']

# Split dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.4,
random_state=42)

# Create and train the decision tree classifier
clf = DecisionTreeClassifier(random_state=42)
clf.fit(X_train, y_train)

# Predict on the test set
y_pred = clf.predict(X_test)

# Calculate accuracy, precision, recall
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)

# Print results
print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
```

Rana Husni

Train on all dataset

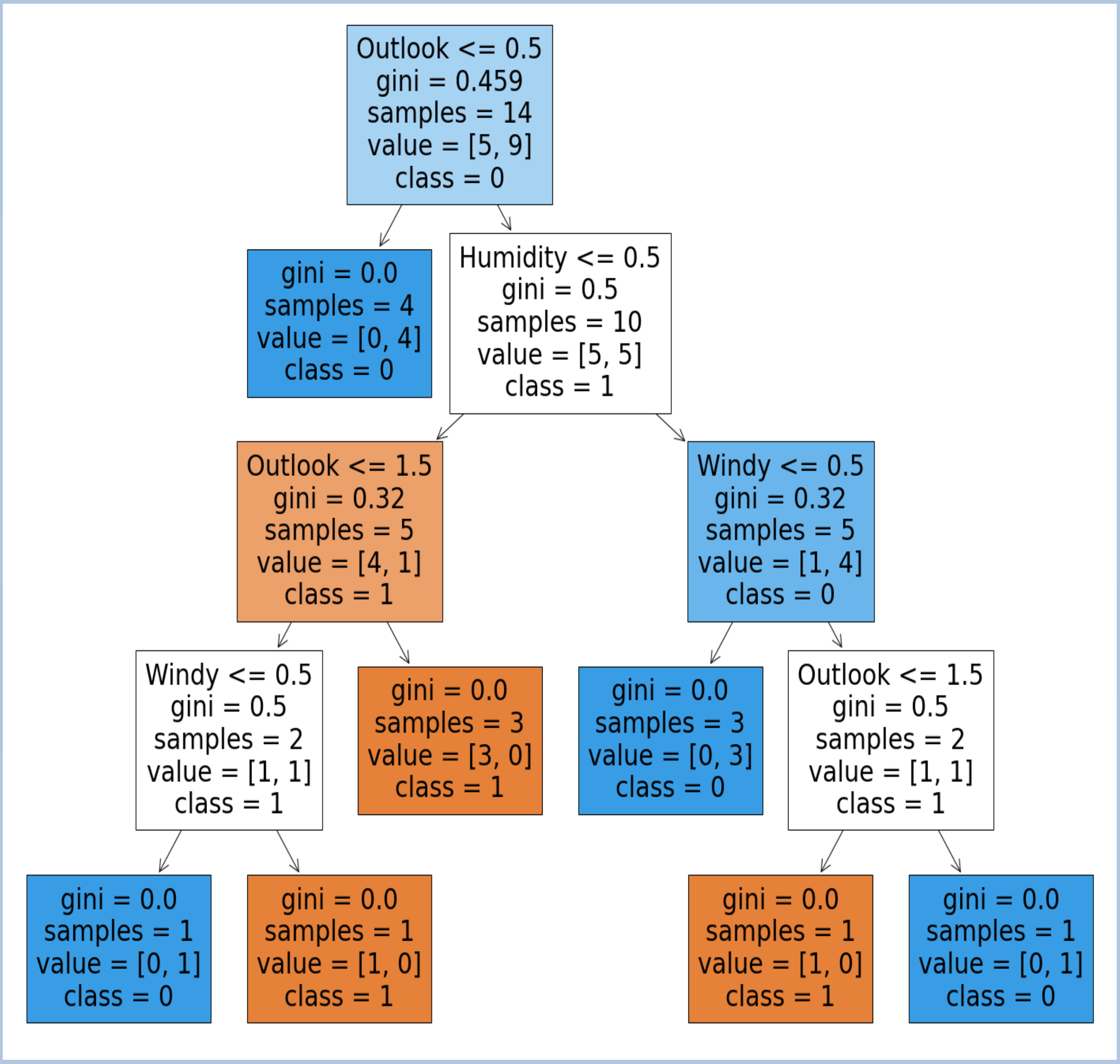
```
from matplotlib import pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree

import pandas as pd

df=pd.read_excel('WheatherTF.xlsx')
data=df.iloc[:,0:4]
y=df['Play']
clf = DecisionTreeClassifier(random_state=1234)
model = clf.fit(data, y)
fig = plt.figure(figsize=(25,20))
tree.plot_tree(clf,feature_names=data.columns,class_names= ['1','0'],
filled=True)
test=[[1,1,1,1]]
prediction=clf.predict(test)
print(prediction)#0

test=[[2,1,1,1]]
prediction=clf.predict(test)
print(prediction)#1
```

Decision Tree


Outlook <= 0.5
gini = 0.459
samples = 14
value = [5, 9]
class = 0
gini = 0.0
samples = 4
value = [0, 4]
class = 0
Humidity <= 0.5
gini = 0.5
samples = 10
value = [5, 5]
class = 1
Outlook <= 1.5
gini = 0.32
samples = 5
value = [4, 1]
class = 1
Windy <= 0.5
gini = 0.32
samples = 5
value = [1, 4]
class = 0
Windy <= 0.5
gini = 0.5
samples = 2
value = [1, 1]
class = 1
gini = 0.0
samples = 3
value = [3, 0]
class = 1
gini = 0.0
samples = 3
value = [0, 3]
class = 0
Outlook <= 1.5
gini = 0.5
samples = 2
value = [1, 1]
class = 1
gini = 0.0
samples = 1
value = [0, 1]
class = 0
gini = 0.0
samples = 1
value = [1, 0]
class = 1
gini = 0.0
samples = 1
value = [1, 0]
class = 1
gini = 0.0
samples = 1
value = [0, 1]
class = 0