Assignment 4 Group 11

Part 1: Numerical Questions

1. (a)

Table 1:

Weather (F1)	Temperature (F2)	Humidty (F3)	Wind (F4)	Hiking (Labels)
Cloudy	Hot	High	Strong	No
Sunny	Mild	High	Weak	No
Rainy	Cold	Normal	Strong	Yes
Cloudy	Mild	Normal	Strong	Yes
Sunny	Mild	High	Strong	No
Rainy	Cool	Normal	Strong	No
Cloudy	Mild	High	Weak	Yes
Sunny	Hot	High	Strong	No
Rainy	Hot	Normal	Weak	Yes
Sunny	Hot	High	Strong	No

$$\mathrm{Gini}(\mathrm{Hiking}) =$$

$$1 - (P(\mathrm{Yes})^2 + P(\mathrm{No})^2)$$

$$1 - \left(\left(\frac{4}{10}\right)^2 + \left(\frac{6}{10}\right)^2\right) = 0.48$$

Step 1: Calculate the G (GINI) for each attribute (feature)

(1)

$$Gini(Hiking|Rainy) =$$

$$1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.44$$

$$Gini(Hiking|Sunny) =$$

$$1 - \left(\frac{4}{4}\right)^2 = 0$$

$$Gini(Hiking|Cloudy)$$

$$1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = 0.44$$

$$Gini(Hiking|Sunny) = Gini(Hiking|Cloudy) = 0.44$$

$$\text{Gini(Weather)}$$

$$= P(Rainy)Gini(Hiking|Rainy) + P(Sunny)Gini(Hiking|Sunny) + P(Cloudy)Gini(Hiking|Cloudy)$$

$$\left(\frac{3}{10}\right) * \left(1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2\right) + \left(\frac{4}{10}\right) * \left(1 - \left(\frac{4}{4}\right)^2\right) + \left(\frac{3}{10}\right) * \left(1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2\right) = 0.26$$

(2)

$$Gini(Hiking|Hot) = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 = 0.4$$

$$Gini(Hiking|Mild) = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.5$$

$$Gini(Hiking|Cool) = 1 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$$

$$Gini(Hiking|Cold) = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 = 0$$

$$Gini(Hiking|Cool) = Gini(Hiking|Cold) = 0$$

$$\text{Gini(Temperature)}$$

$$= \ P(Hot)Gini(Hiking|Hot) + \ P(Mild)Gini(Hiking|Mild) + \ P(Cool)Gini(Hiking|Cool) + \ P(Cold)Gini(Hiking|Cold)$$

$$\left(\frac{4}{10}\right) * \left(1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2\right) + \left(\frac{4}{10}\right) * \left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) + \left(\frac{1}{10}\right) * \left(1 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2\right) + \left(\frac{1}{10}\right) * \left(1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2\right) = 0.35$$

(3)

$$Gini(Hiking|High) = 1 \ - \left(\frac{5}{7}\right)^2 - \left(\frac{2}{7}\right)^2 = 0.4$$

$$Gini(Hiking|Normal) = 1 \ - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 = 0.4$$

$$Gini(Hiking|High) = \ Gini(Hiking|Normal) = 0.4$$

$$Gini(Humidity)$$

$$= P(High)Gini(Hiking|High) + \ P(Normal)Gini(Hiking|Normal)$$

$$\left(\frac{6}{10}\right) * \left(1 - \left(\frac{1}{6}\right)^2 - \left(\frac{5}{6}\right)^2\right) + \left(\frac{4}{10}\right) * \left(1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2\right) = \ 0.32$$

(4)

$$Gini(Hiking|Strong) = 1 \ - \left(\frac{5}{7}\right)^2 = 0.4$$

$$Gini(Hiking|Weak) = 1 \ - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = 0.4$$

$$Gini(Wind)$$

$$= \ P(Strong) \ Gini(Hiking|Strong) \ + \ P(Weak) \ Gini(Hiking|Weak)$$

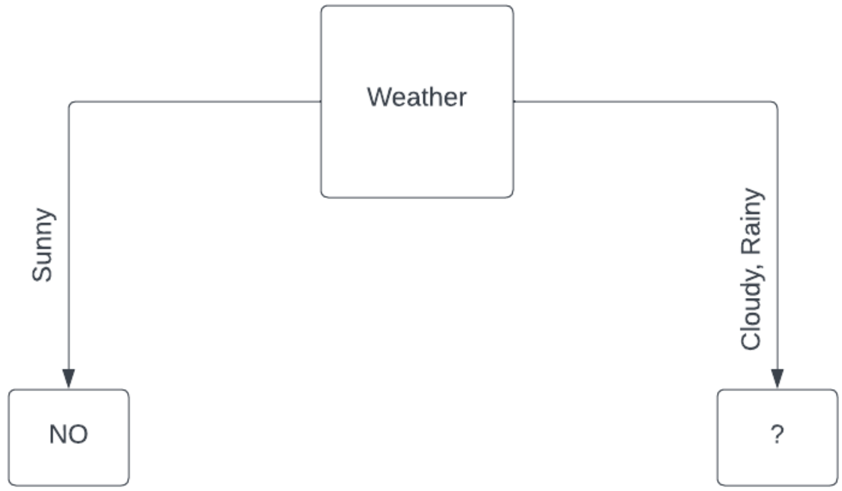
$$= \left(\frac{7}{10}\right) * \left(1 - \left(\frac{5}{7}\right)^2 - \left(\frac{2}{7}\right)^2\right) + \left(\frac{3}{10}\right) * \left(1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2\right) \ = 0.42$$

GINI (Y, wind) = 0.42

GINI (Y, hum) = 0.32

GINI (Y, weather) = 0.26

GINI (Y, temp) = 0.43



Step 2: Choose which feature to split with!

Weather (F1)	Temperature (F2)	Humidty (F3)	Wind (F4)	Hiking (Labels)
Cloudy	Hot	High	Strong	No
Rainy	Cold	Normal	Strong	Yes
Cloudy	Mild	Normal	Strong	Yes
Rainy	Cool	Normal	Strong	No
Cloudy	Mild	High	Weak	Yes
Rainy	Hot	Normal	Weak	Yes

$$Gini(Hiking) = 1 - (P(Yes)^2 + P(No)^2)$$

$$= 1 - \left(\left(\frac{4}{6}\right)^2 + \left(\frac{2}{6}\right)^2\right) = 0.44$$

(1)

$$Gini(Hiking|Hot) = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = 0.5$$

$$Gini(Hiking|Mild) = 0$$

$$Gini(Hiking|Cool) = 0$$

$$Gini(Hiking|Cold) = 0$$

$$Gini(Hiking|Mild) = Gini(Hiking|Cool) = Gini(Hiking|Cold) = 0$$

$$\boldsymbol{Gini}(\boldsymbol{Temperature})$$

$$= P(Hot)Gini(Hiking|Hot) + P(Mild)Gini(Hiking|Mild) + P(Cool)Gini(Hiking|Cool) + P(Cold)Gini(Hiking|Cold)$$

$$= \left(\frac{2}{6}\right) * \left(1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2\right) + \left(\frac{2}{6}\right) * \left(1 - \left(\frac{2}{2}\right)^2\right) + \left(\frac{1}{6}\right) * \left(1 - \left(\frac{1}{1}\right)^2\right) + \left(\frac{1}{6}\right) * \left(1 - \left(\frac{1}{1}\right)^2\right) = 0.17$$

(2)

$$Gini(Hiking|High) = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = 0.5$$

$$Gini(Hiking|Normal) = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 = 0.4$$

$$Gini(Humidity) = P(High)Gini(Hiking|High) + P(Normal)Gini(Hiking|Normal)$$

$$= \left(\frac{2}{6}\right) * \left(1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2\right) + \left(\frac{4}{6}\right) * \left(1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2\right) = 0.42$$

(3)

$$Gini(Hiking|Strong) = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.5$$

$$Gini(Hiking|Weak) = 1 - \left(\frac{2}{2}\right)^2 = 0$$

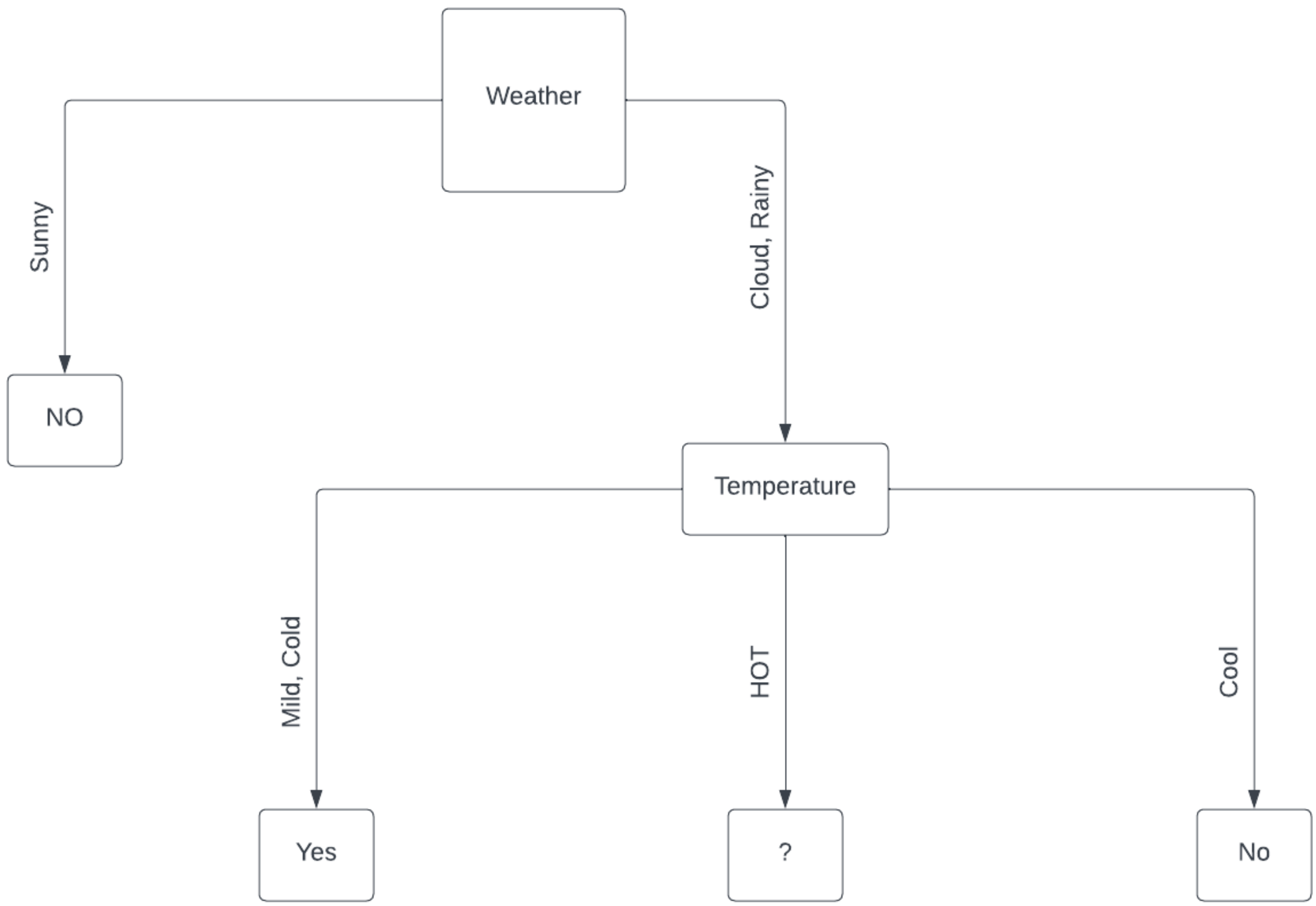
$$Gini(Wind) = P(Strong)Gini(Hiking|Strong) + P(Weak)Gini(Hiking|Weak)$$

$$= \left(\frac{4}{6}\right) * \left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) + \left(\frac{2}{6}\right) * \left(1 - \left(\frac{2}{2}\right)^2\right) = 0.33$$

GINI (Y, wind) = 0.33

GINI (Y, hum) = 0.42

GINI (Y, temp) = 0.17


Weather
Sunny
Cloud, Rainy
NO
Temperature
Mild, Cold
HOT
Cool
Yes
?
No

Step 3:-

Weather (F1)	Temperature (F2)	Humidty (F3)	Wind (F4)	Hiking (Labels)
Cloudy	Hot	High	Strong	No
Rainy	Hot	Normal	Weak	Yes

$$Gini(Hiking) = 1 - (P(Yes)^2 + P(No)^2)$$

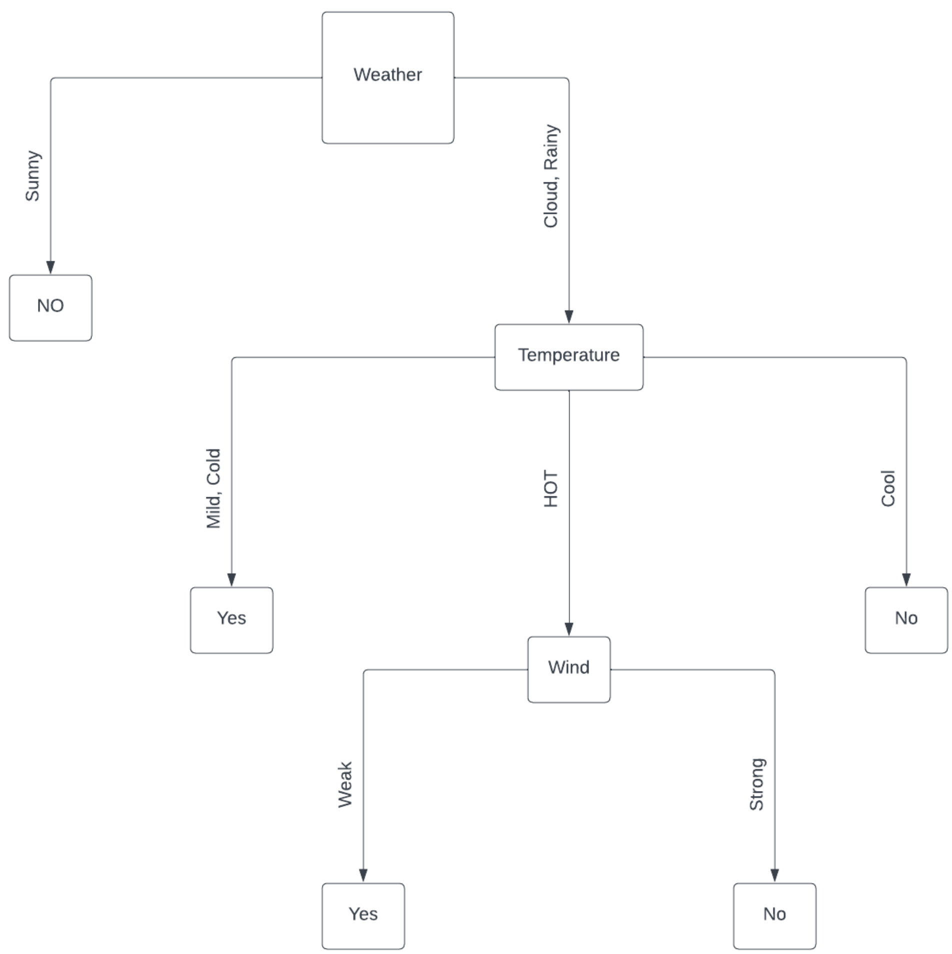
$$= 1 - \left(\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right) = 0.5$$

$$Gini(Humidity) = P(High)Gini(Hiking|High) + P(Normal)Gini(Hiking|Normal)$$

$$= \left(\frac{1}{2}\right) * \left(1 - \left(\frac{1}{1}\right)^2\right) + \left(\frac{1}{2}\right) * \left(1 - \left(\frac{1}{1}\right)^2\right) = 0$$

$$Gini(Wind) = P(Strong)Gini(Hiking|Strong) + P(Weak)Gini(Hiking|Weak)$$

$$= \left(\frac{1}{2}\right) * \left(1 - \left(\frac{1}{1}\right)^2\right) + \left(\frac{1}{2}\right) * \left(1 - \left(\frac{1}{1}\right)^2\right) = 0$$



Part 2: Programming Questions

(b) Please build a decision tree by using Information Gain (i.e., IG(T, a) = Entropy(T)− Entropy(T|a), More information about IG).

Weather (F1)	Temperature (F2)	Humidty (F3)	Wind (F4)	Hiking (Labels)
Cloudy	Hot	High	Strong	No
Sunny	Mild	High	Weak	No
Rainy	Cold	Normal	Strong	Yes
Cloudy	Mild	Normal	Strong	Yes
Sunny	Mild	High	Strong	No
Rainy	Cool	Normal	Strong	No
Cloudy	Mild	High	Weak	Yes
Sunny	Hot	High	Strong	No
Rainy	Hot	Normal	Weak	Yes
Sunny	Hot	High	Strong	No

For Hiking (Labels):

P_{yes} = number of Yes / Total number = 4/10

P_{no} = number of No / Total number = 6/10

Entropy(S) = $-P_{yes} \text{ Log}_2 P_{yes} - P_{no} \text{ Log}_2 P_{no} = \sum_{i=1}^{c} -P_i \text{ Log}_2 P_i$

Entropy(S) = $-4/10 \text{ Log}_2 4/10 - 6/10 \text{ Log}_2 6/10 = 0.528 + 0.442$ **= 0.97**

We will calculate info gain for each spilt:

For weather:

		Hiking (10)		
		Yes	No	
Weather	Cloudy	2	1	3
	Sunny	0	4	4
	Rainy	2	1	3

$GAIN(S, Weather) = Entropy(S) - \frac{S_{Cloudy}}{10} Entropy(S_{Cloudy}) - \frac{S_{Sunny}}{10} Entropy(S_{Sunny}) - \frac{S_{Rainy}}{10} Entropy(S_{Rainy})$

Gain(S, Weather) = 0.97 – 3/10 (-2/3 Log_2 2/3 – 1/3 Log_2 1/3) – 4/10 (-4/4 Log_2 4/4) - 3/10 (-2/3 Log_2 2/3 – 1/3 Log_2 1/3) = 0.97 -0.275 -0 – 0.275 = 0.42

For Temperature:

		Hiking (10)		
		Yes	No	
Temperature	Hot	1	3	4
	Mild	2	2	4
	Cool	0	1	1
	Cold	1	0	1

$GAIN(S, Temperature) = Entropy(S) - \frac{S_{Hot}}{10} Entropy(S_{Hot}) - \frac{S_{Mild}}{10} Entropy(S_{Mild}) - \frac{S_{Cold}}{10} Entropy(S_{Cold}) - \frac{S_{Cool}}{10} Entropy(S_{Cool})$

$= 0.97 - \frac{4}{10}(-\frac{1}{4}\log_2 \frac{1}{4} - \frac{3}{4}\log_2 \frac{3}{4}) - \frac{4}{10}(-\frac{2}{4}\log_2 \frac{2}{4} - \frac{2}{4}\log_2 \frac{2}{4}) - \frac{1}{10}(-\frac{1}{1}\log_2 \frac{1}{1}) - \frac{1}{10}(-\frac{1}{1}\log_2 \frac{1}{1})$

= 0.97 - 0.325 - 0.4 - 0 - 0 = 0.245

For Humidity:

		Hiking (10)		
		Yes	No	
Humidity	High	1	5	6
	Normal	3	1	4

$GAIN(S, Humidty) = Entropy(S) - \frac{S_{High}}{10} Entropy(S_{High}) - \frac{S_{Normal}}{10} Entropy(S_{Normal})$

$$= 0.97 - \frac{6}{10}(-\frac{1}{6}\log_2\frac{1}{6} - \frac{5}{6}\log_2\frac{5}{6}) - \frac{4}{10}(-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4})$$ = 0.97 - 0.39 - 0.32 = 0.255

For Wind:

		Hiking (10)		
		Yes	No	
Wind	Strong	2	5	7
	Weak	2	1	3

$$GAIN(S, Wind) = Entropy(S) - \frac{S_{Strong}}{10}Entropy(S_{Strong}) - \frac{S_{Weak}}{10}Entropy(S_{Weak})$$

$$= 0.97 - \frac{7}{10}(-\frac{2}{7}\log_2\frac{2}{7} - \frac{5}{7}\log_2\frac{5}{7}) - \frac{3}{10}(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3})$$ = 0.97 - 0.604 - 0.275 = 0.091

So, as we can see the info gain for:

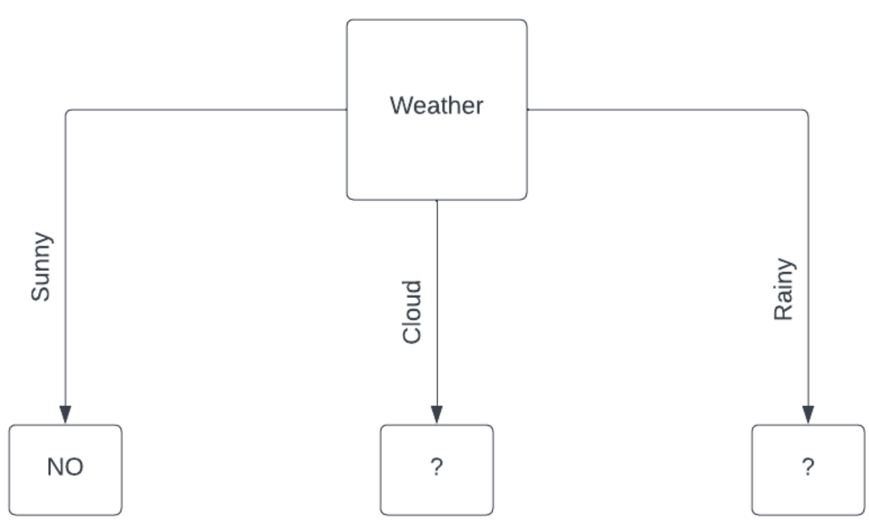
Weather **=** 0.42

Temperature = 0.245

Humidity = 0.255

Wind = 0.091

So, the highest info gain is Weather so we will split according to it and split will look like this:



we see one leaf node. But we need to split the tree further

Weather(F1)	Temperature(F2)	Humidity(F3)	Wind(F4)	Hiking(Labels)
Sunny	Mild	High	Weak	No
Sunny	Mild	High	Strong	No
Sunny	Hot	High	Strong	No
Sunny	Hot	High	Strong	No

we could see that the Sunny Weather requires no further split as it ‘s now a leaf node

we need to also split the original table to create sub tables. This sub tables:

Cloudy	Hot	High	Strong	No
Cloudy	Mild	Normal	Strong	Yes
Cloudy	Mild	High	Weak	Yes
Rainy	Cold	Normal	Strong	Yes
Rainy	Cool	Normal	Strong	No
Rainy	Hot	Normal	Weak	Yes

The Cloudy and the Rainy attributes needs to be split:

For cloudy:

Hiking(3)	
Yes	No
2	1

Now we will calculate the new Entropy:

Entropy(S)= $\sum_{i=1}^{c} -P_i \text{ Log}_2 \text{ P}_i$

Entropy(Hiking) = $P_{yes} \text{ Log}_2 P_{yes} - P_{no} \text{ Log}_2 P_{no} = -1/3 \text{ Log}_2 1/3 - 2/3 \text{ Log}_2 2/3 = 0.918$

Now we will Calculate Information Gain for Each Split:

For Temperature:

$$GAIN_{\text{split}} = Entropy(S) - \left(\sum_{i=1}^{k} \frac{n_i}{n} \cdot Entropy(i)\right)$$

		Hiking (3)		
		Yes	No	
Temperature	Hot	0	1	1
	Mild	2	0	2

$GAIN(S, Temperature) = Entropy(S) - \frac{S_{Hot}}{10} Entropy(S_{Hot}) - \frac{S_{Mild}}{10} Entropy(S_{Mild})$

$= 0.918 - \frac{1}{3}(-\frac{1}{1}\log_2\frac{1}{1}) - \frac{2}{3}(-\frac{2}{2}\log_2\frac{2}{2})$ = 0.918 - 0 - 0 = 0.918

For Humidity:

		Hiking (3)		
		Yes	No	
Humidity	High	1	1	2
	Normal	1	0	1

$GAIN(S, Humidty) = Entropy(S) - \frac{S_{High}}{10} Entropy(S_{High}) - \frac{S_{Normal}}{10} Entropy(S_{Normal})$

$= 0.918 - \frac{2}{3}(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}) - \frac{1}{3}(-\frac{1}{1}\log_2\frac{1}{1})$ = 0.918 - 0.67 - 0 = 0.251

For Wind:

		Hiking (3)		
		Yes	No	
Wind	Strong	1	1	2
	Weak	1	0	1

$GAIN(S, Wind) = Entropy(S) - \frac{S_{Strong}}{10} Entropy(S_{Strong}) - \frac{S_{Weak}}{10} Entropy(S_{Weak})$

$= 0.918 - \frac{2}{3}(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}) - \frac{1}{3}(-\frac{1}{1}\log_2\frac{1}{1})$ = 0.918 - 0.67 - 0 = 0.251

So, as we can see the info gain for:

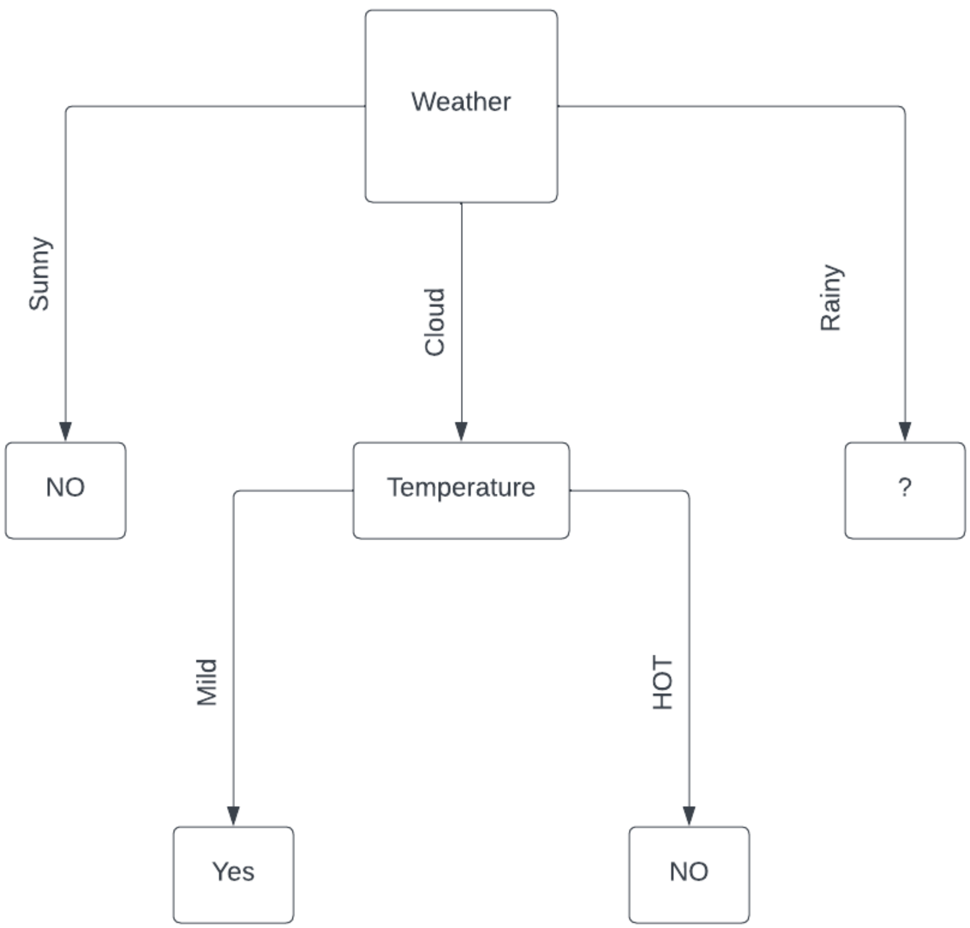
Temperature = 0.918

Humidity = 0.251

Wind = 0.251

We can see the highest info gain is from Temperature:

So final spilt Decision tree will be like:



The only branch needs to split now is Rainy

Rainy attributes are:

Hiking	
Yes	No
2	1

Calculate the entropy for class Hiking:

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

$Entropy(Hiking) = \mathbf{E}(2,1)$ $= -Pyes\log_2 Pyes - Pno\log_2 Pno = -(\frac{2}{3}\log_2\frac{2}{3}) - (\frac{1}{3}\log_2\frac{1}{3}) = 0.918$

$$GAIN_{\text{split}} = Entropy(S) - \left(\sum_{i=1}^{k} \frac{n_i}{n} \cdot Entropy(i)\right)$$

Info Gain for Temperature:

		Hiking		
		Yes	No	
Temperature	Cold	1	0	1
	Cool	0	1	1
	Hot	1	0	1

$GAIN(S, Temperature) = Entropy(S) - \frac{S_{Cold}}{10} Entropy(S_{Cold}) - \frac{S_{Cool}}{10} Entropy(S_{Cool}) - \frac{S_{Hot}}{10} Entropy(S_{Hot})$
$= 0.918 - \frac{1}{3}(-\frac{1}{1}\log_2\frac{1}{1}) - \frac{1}{3}(-\frac{1}{1}\log_2\frac{1}{1}) - \frac{1}{3}(-\frac{1}{1}\log_2\frac{1}{1})$ = 0.918 - 0 - 0 - 0 = 0.918

For Humidity:

		Hiking		
		Yes	No	
Humidity	Normal	2	1	3

$GAIN(S, Humidty) = Entropy(S) - \frac{S_{Normal}}{10} Entropy(S_{Normal}) = 0.918 - \frac{3}{3}(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3})$ = 0

For Wind:

		Hiking (3)		
		Yes	No	
Wind	Strong	2	0	2
	Weak	0	1	1

$GAIN(S, Wind) = Entropy(S) - \frac{S_{Strong}}{10} Entropy(S_{Strong}) - \frac{S_{Weak}}{10} Entropy(S_{Weak})$
$= 0.918 - \frac{2}{3}(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}) - \frac{1}{3}(-\frac{1}{1}\log_2\frac{1}{1})$ = 0.251

So, as we can see the info gain for:

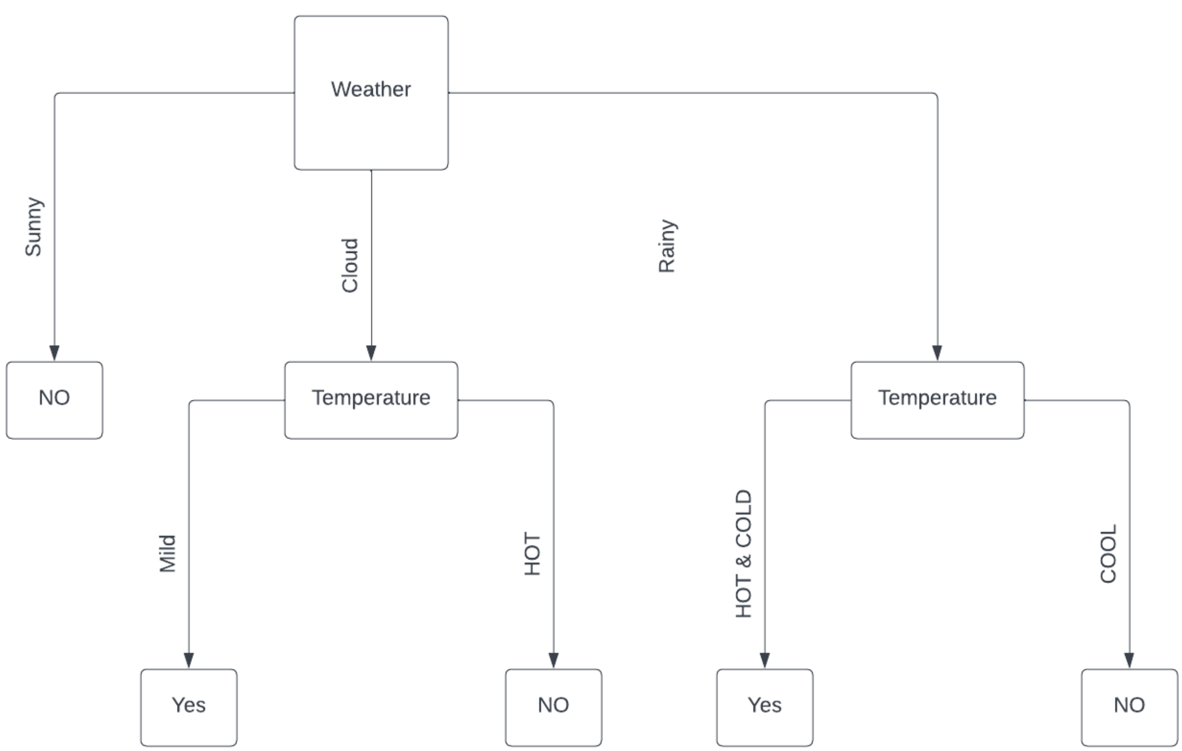
Temperature = 0.918

Humidity = 0

Wind = 0.251

We can see the highest info gain is from Temperature:

So, our last and final spilt Decision tree will be like:



(c) Please compare the advantages and disadvantages between Gini Index and Information Gain.

The Gini Index and Information Gain are two popular metrics used in the field of machine learning and decision tree algorithms to measure the quality of a split and determine the best attribute for node splitting. Here are the advantages and disadvantages of each:

Advantages of the Gini Index:

1. Simplicity: The Gini Index is a straightforward and easy-to-understand metric. It measures the impurity of a node by calculating the probability of misclassifying a randomly chosen element in the dataset.

2. Computationally efficient: The Gini Index does not involve calculating logarithms, making it computationally faster compared to Information Gain, especially for large datasets.

3. Insensitive to the number of classes: The Gini Index performs well even when dealing with multi-class classification problems. It does not depend on the number of classes in the dataset, making it suitable for scenarios with multiple outcomes.

Disadvantages of the Gini Index:

1. Biased towards multi-way splits: The Gini Index tends to favor attributes with a large number of distinct values since it allows for multi-way splits. Consequently, it may result in biased tree structures with more complex branches.

2. Ignores the magnitude of the split: The Gini Index only considers the class distribution within each child node, ignoring the actual values or magnitudes of the attribute being split. As a result, it may not be the most suitable metric when the attribute values carry important information.

Advantages of Information Gain:

1. Incorporates attribute value magnitudes: Information Gain considers the actual values of the attributes and their magnitudes. It measures the reduction in entropy (uncertainty) of the target variable after the split.

2. Handles binary splits well: Information Gain tends to favor attributes that produce binary splits, resulting in more concise and interpretable decision trees.

3. Can handle continuous and categorical attributes: Unlike the Gini Index, which works better with categorical attributes, Information Gain can be used with both continuous and categorical attributes.

Disadvantages of Information Gain:

1. Sensitive to the number of classes: Information Gain is biased towards attributes with many distinct values or classes, as it tends to prioritize attributes that provide more options for splitting.

2. Prone to overfitting: Information Gain is susceptible to overfitting when dealing with attributes with a high number of distinct values. It may lead to complex decision trees that do not generalize well to unseen data.

3. Computationally expensive: Calculating Information Gain involves calculating logarithmic functions, which can be computationally expensive, especially for large datasets.

In summary, the Gini Index is simpler and computationally efficient, while Information Gain incorporates attribute magnitudes and handles binary splits better. The choice between the two depends on the specific characteristics of the dataset and the desired properties of the decision tree model.

Assignment 4 Part 2: Programming Questions

a) We load the data from the file "KDD.csv" and separate the input features and target variable, normalizes the input features using MinMaxScaler, applies filter-based feature selection to reduce the number of features to 9, and adds the target variable back to the dataset.

	0	1	2	3	4	5	6	7	8	target
0	1.0	0.015656	0.015656	0.0	1.0	0.0	0.035294	0.11	0.0	0
1	1.0	0.015656	0.015656	0.0	1.0	0.0	0.074510	0.05	0.0	0
2	1.0	0.015656	0.015656	0.0	1.0	0.0	0.113725	0.03	0.0	0
3	1.0	0.011742	0.011742	0.0	1.0	0.0	0.152941	0.03	0.0	0
4	1.0	0.011742	0.011742	0.0	1.0	0.0	0.192157	0.02	0.0	0

b) splits the dataset into three different training and testing sets with different test sizes (30%, 40%, and 50%), trains a Decision Tree classifier on each training set, and evaluates its performance on both the training and testing sets using classification reports.

```
Results for my data 1
Classification report for train:
              precision    recall  f1-score   support

           0       0.97      1.00      0.98     68086
           1       1.00      0.99      1.00    277728

    accuracy                           0.99    345814
   macro avg       0.98      0.99      0.99    345814
weighted avg       0.99      0.99      0.99    345814

Classification report for test:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     29192
           1       1.00      0.99      0.99    119015

    accuracy                           0.99    148207
   macro avg       0.98      0.99      0.99    148207
weighted avg       0.99      0.99      0.99    148207
```

```
Results for my data 2
Classification report for train:
              precision    recall  f1-score   support

           0       0.97      1.00      0.98     58301
           1       1.00      0.99      1.00    238111

    accuracy                           0.99    296412
   macro avg       0.98      0.99      0.99    296412
weighted avg       0.99      0.99      0.99    296412

Classification report for test:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     38977
           1       1.00      0.99      0.99    158632

    accuracy                           0.99    197609
   macro avg       0.98      0.99      0.99    197609
weighted avg       0.99      0.99      0.99    197609
```

```
Results for my data 3
Classification report for train:
              precision    recall  f1-score   support

           0       0.97      1.00      0.98     48628
           1       1.00      0.99      1.00    198382

    accuracy                           0.99    247010
   macro avg       0.98      0.99      0.99    247010
weighted avg       0.99      0.99      0.99    247010

Classification report for test:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     48650
           1       1.00      0.99      0.99    198361

    accuracy                           0.99    247011
   macro avg       0.98      0.99      0.99    247011
weighted avg       0.99      0.99      0.99    247011
```

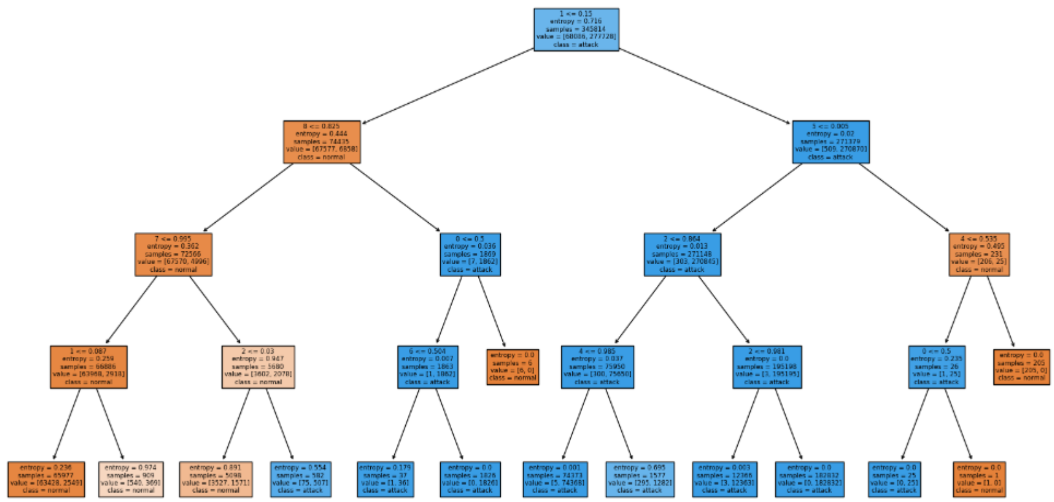
c) train and evaluate a decision tree model on each subset of the data with different hyperparameters. it trains decision trees with different max_depth values (4, 6, and 8), evaluates the model on the corresponding test sets, and generates a classification report for each combination of dataset and max_depth.

```
Classification report for my_data_1 with max_depth=4:
              precision    recall  f1-score   support

           0       0.94      0.99      0.96     29192
           1       1.00      0.98      0.99    119015

    accuracy                           0.99    148207
   macro avg       0.97      0.99      0.98    148207
weighted avg       0.99      0.99      0.99    148207
```

Best decision tree split for my_data_1 with max_depth=4

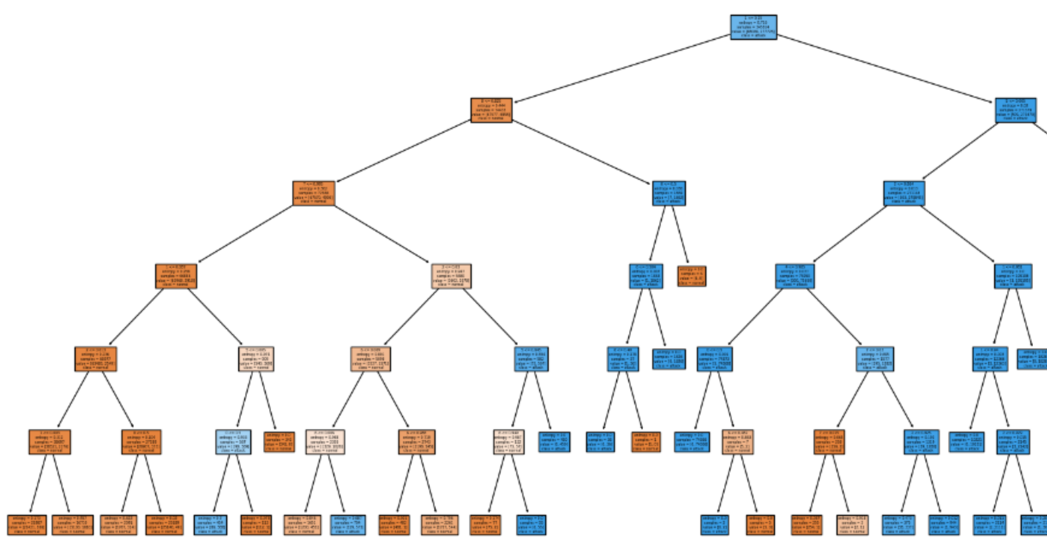


```
Classification report for my_data_1 with max_depth=6:
              precision    recall  f1-score   support

           0       0.95      1.00      0.97     29192
           1       1.00      0.99      0.99    119015

    accuracy                           0.99    148207
   macro avg       0.97      0.99      0.98    148207
weighted avg       0.99      0.99      0.99    148207
```

Best decision tree split for my_data_1 with max_depth=6

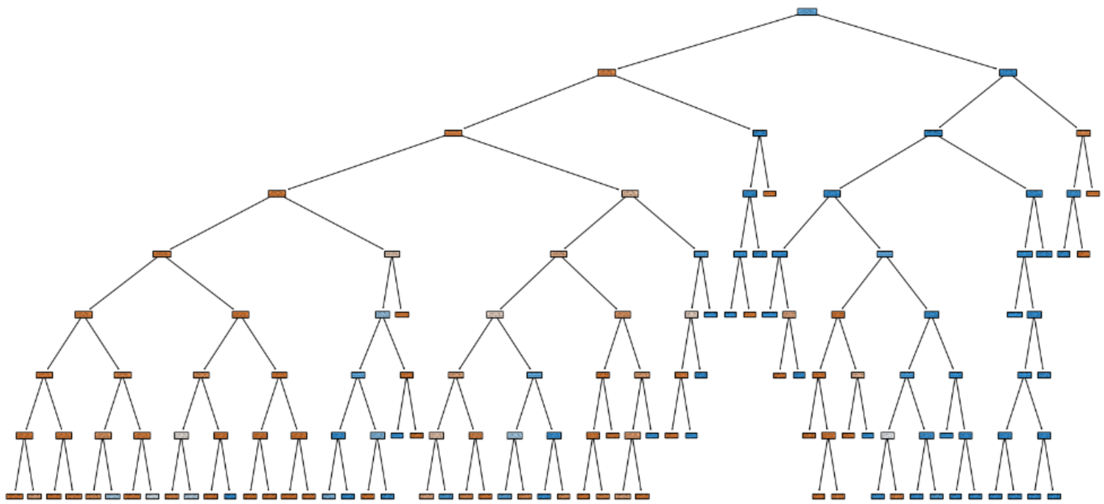


```
Classification report for my_data_1 with max_depth=8:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     29192
           1       1.00      0.99      0.99    119015

    accuracy                           0.99    148207
   macro avg       0.98      0.99      0.98    148207
weighted avg       0.99      0.99      0.99    148207
```

Best decision tree split for my_data_1 with max_depth=8

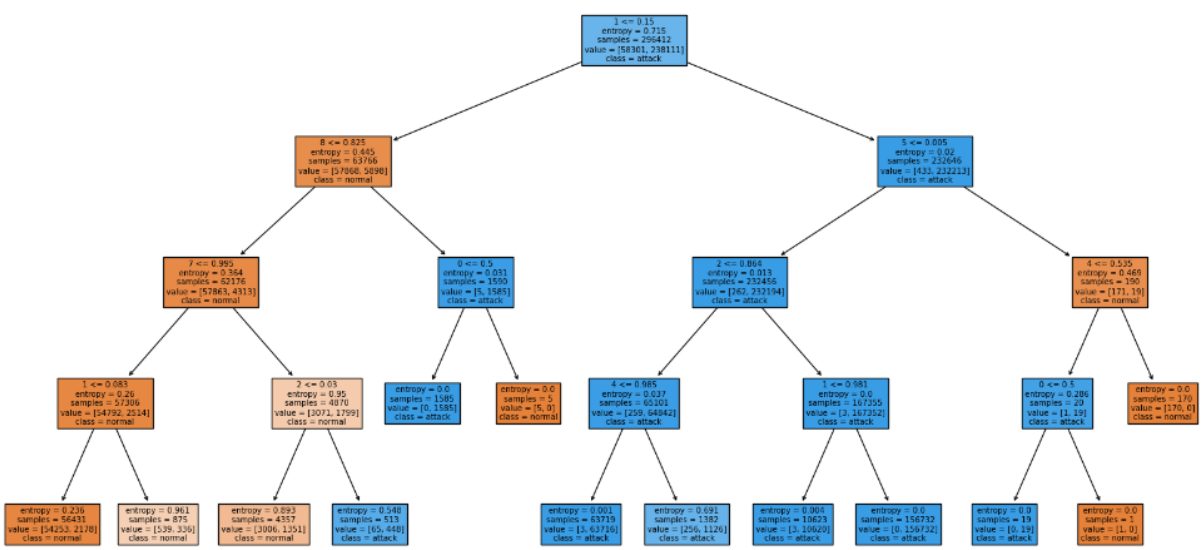


```
Classification report for my_data_2 with max_depth=4:
              precision    recall  f1-score   support

           0       0.94      0.99      0.97     38977
           1       1.00      0.98      0.99    158632

    accuracy                           0.99    197609
   macro avg       0.97      0.99      0.98    197609
weighted avg       0.99      0.99      0.99    197609
```

Best decision tree split for my_data_2 with max_depth=4

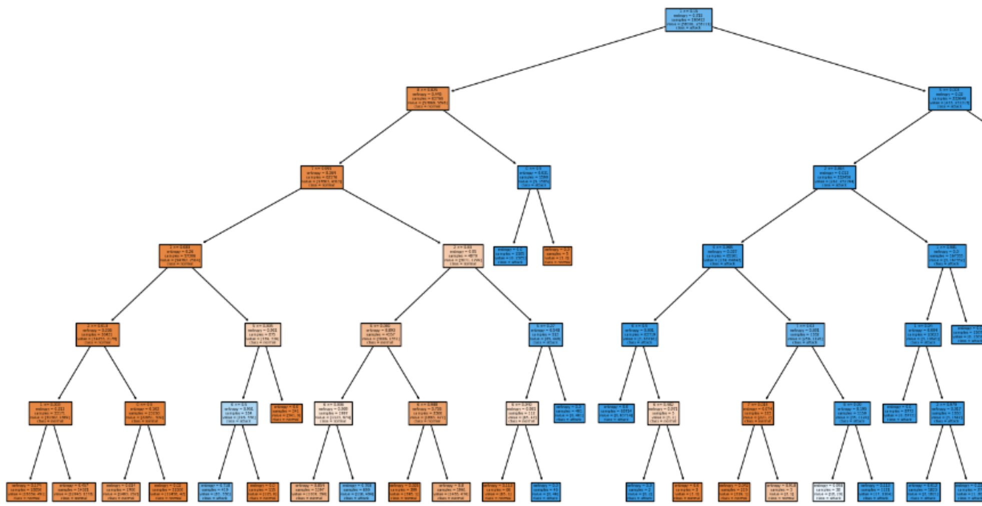


```
Classification report for my_data_2 with max_depth=6:
              precision    recall  f1-score   support

           0       0.95      1.00      0.97     38977
           1       1.00      0.99      0.99    158632

    accuracy                           0.99    197609
   macro avg       0.97      0.99      0.98    197609
weighted avg       0.99      0.99      0.99    197609
```

Best decision tree split for my_data_2 with max_depth=6

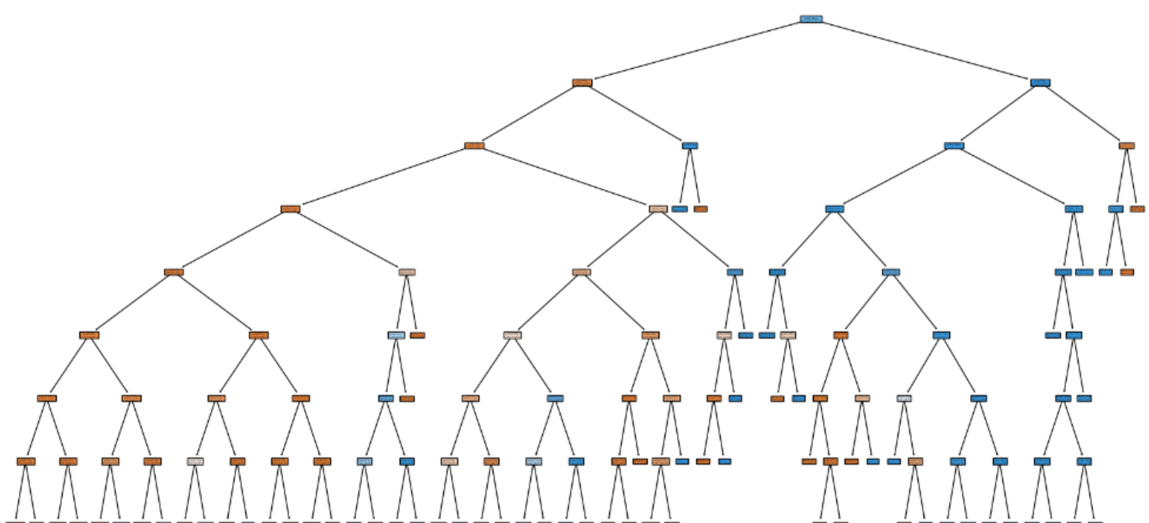


```
Classification report for my_data_2 with max_depth=8:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     38977
           1       1.00      0.99      0.99    158632

    accuracy                           0.99    197609
   macro avg       0.98      0.99      0.98    197609
weighted avg       0.99      0.99      0.99    197609
```

Best decision tree split for my_data_2 with max_depth=8

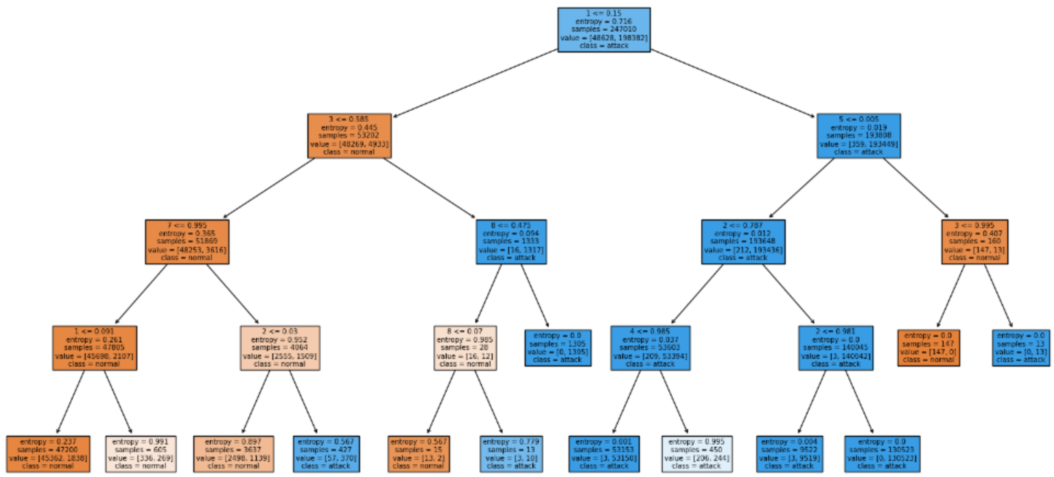


```
Classification report for my_data_3 with max_depth=4:
              precision    recall  f1-score   support

           0       0.94      0.99      0.97     48650
           1       1.00      0.98      0.99    198361

    accuracy                           0.99    247011
   macro avg       0.97      0.99      0.98    247011
weighted avg       0.99      0.99      0.99    247011
```

Best decision tree split for my_data_3 with max_depth=4

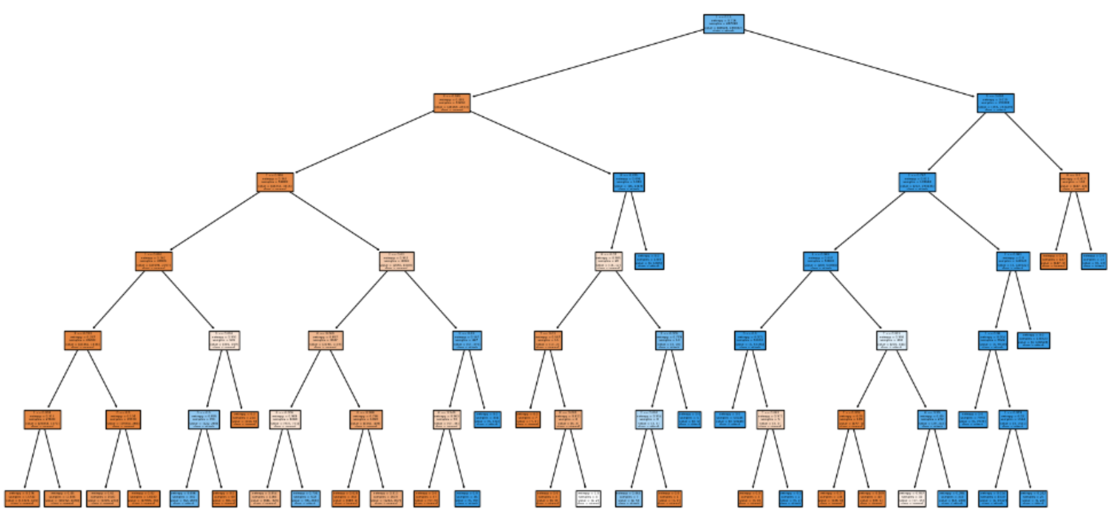



```
Classification report for my_data_3 with max_depth=6:
              precision    recall  f1-score   support

           0       0.95      1.00      0.97     48650
           1       1.00      0.99      0.99    198361

    accuracy                           0.99    247011
   macro avg       0.97      0.99      0.98    247011
weighted avg       0.99      0.99      0.99    247011
```

Best decision tree split for my_data_3 with max_depth=6

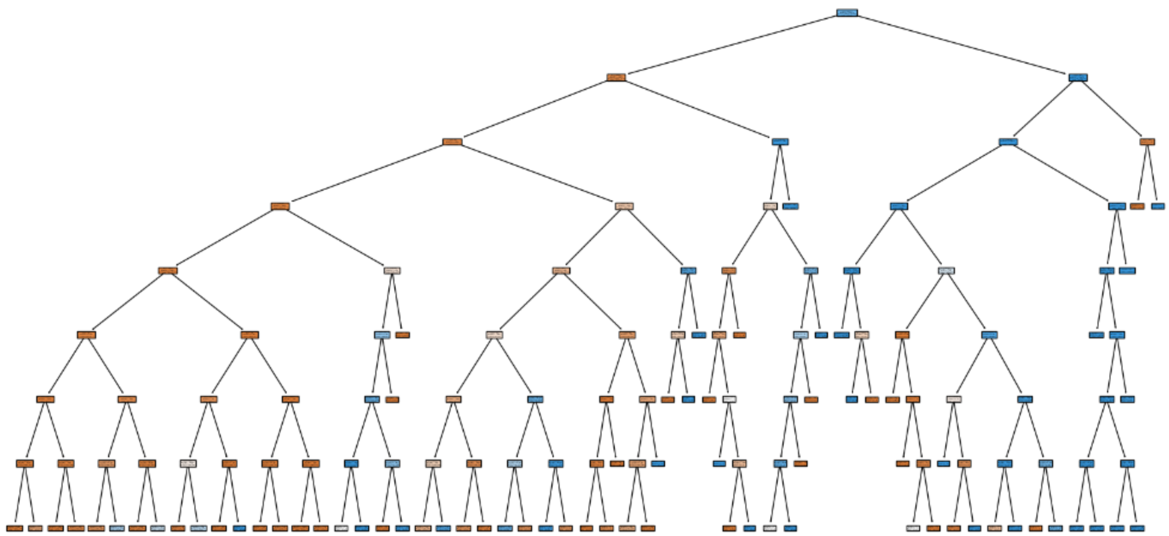


```
Classification report for my_data_3 with max_depth=8:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     48650
           1       1.00      0.99      0.99    198361

    accuracy                           0.99    247011
   macro avg       0.98      0.99      0.99    247011
weighted avg       0.99      0.99      0.99    247011
```

Best decision tree split for my_data_3 with max_depth=8



d) use nested loops to iterate over different train-test splits and different values of max_depth, trains a Decision Tree classifier on each train-test split and max_depth combination, and evaluates the performance of the classifier on the corresponding test set. it defines the train-test ratios and max_depth values for each subset of the data, iterates over the subsets and max_depth values, finds the best max_depth value for the current subset using the training set, trains the Decision Tree with the best max_depth on the entire dataset, and evaluates its performance on the corresponding test set using accuracy score, classification report, and confusion matrix.

```
My Data 1 (70% train, 30% test), Best max_depth = 8:
Accuracy: 0.990148913344174
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     29192
           1       1.00      0.99      0.99    119015

    accuracy                           0.99    148207
   macro avg       0.98      0.99      0.98    148207
weighted avg       0.99      0.99      0.99    148207

Confusion Matrix:
[[ 28990    202]
 [  1258 117757]]
```

```
My Data 2 (60% train, 40% test), Best max_depth = 8:
Accuracy: 0.9903192668350126
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     38977
           1       1.00      0.99      0.99    158632

    accuracy                           0.99    197609
   macro avg       0.98      0.99      0.98    197609
weighted avg       0.99      0.99      0.99    197609

Confusion Matrix:
[[ 38706    271]
 [  1642 156990]]
```

```
My Data 3 (50% train, 50% test), Best max_depth = 8:
Accuracy: 0.9904660116351094
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     48650
           1       1.00      0.99      0.99    198361

    accuracy                           0.99    247011
   macro avg       0.98      0.99      0.99    247011
weighted avg       0.99      0.99      0.99    247011

Confusion Matrix:
[[ 48302    348]
 [  2007 196354]]
```

e) trains a Decision Tree classifier on the train set with a max_depth of 8 and entropy as the splitting criterion, and evaluates the performance of the classifier on both the train and test sets using the F1 score. The F1 score is a commonly used metric for evaluating binary classification models that takes into account both precision and recall.

```
Train F1 score: 0.9905172145876123
Test F1 score: 0.9901879035327712
```

mitigates overfitting in the Decision Tree classifier by using pre-pruning. Pre-pr uning involves stopping the growth of the decision tree before it reaches the m aximum depth by setting additional constraints on the construction of the tree. In this case, we set the min_samples_leaf parameter to 10, which specifies the minimum number of samples required to be at a leaf node.

```
Train F1 score after pre-pruning: 0.990442501732848
Test F1 score after pre-pruning: 0.9901073541721761
```

by using post-pruning. Post-pruning involves growing the decision tree to the m aximum depth and then removing branches that do not improve the generaliza tion performance of the tree. In this case, we use cost complexity pruning, whi ch involves adding a penalty term to the objective function that balances the c omplexity of the tree and its performance on the training data.

```
Train F1 score after post-pruning: 0.9905172145876123
Test F1 score after post-pruning: 0.9901879035327712
```

by using k-fold cross-validation. Cross-validation is a technique that involves spl itting the data into k subsets (or "folds"), training the model on k-1 of the folds, and evaluating it on the remaining fold. This process is repeated k times, with e ach fold serving as the test fold once. The results are then averaged to obtain a more reliable estimate of the model's performance.

```
Train F1 score after k-fold cross-validation with k=4: 0.990373434917281
Test F1 score after k-fold cross-validation with k=4: 0.9900207158633356
```