B1: Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.

In [ ]:

```
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, export_text, plot_tree
```

In [14]:

```
import matplotlib.pyplot as plt
```

In [ ]:

```
# Load the Iris dataset
iris = load_iris()
X = iris.data # Independent variables
y = iris.target # dependent variable/class
```

In [16]:

```
# Create a DataFrame for better visualization
df = pd.DataFrame(data=X, columns=iris.feature_names)
df['target'] = y
print(df.head())
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2   
1                4.9               3.0                1.4               0.2   
2                4.7               3.2                1.3               0.2   
3                4.6               3.1                1.5               0.2   
4                5.0               3.6                1.4               0.2   

   target  
0       0  
1       0  
2       0  
3       0  
4       0  
```

In [17]:

```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [27]:

```
X_train
```

```
Out[27]: array([[4.6, 3.6, 1. , 0.2],
               [5.7, 4.4, 1.5, 0.4],
               [6.7, 3.1, 4.4, 1.4],
               [4.8, 3.4, 1.6, 0.2],
               [4.4, 3.2, 1.3, 0.2],
               [6.3, 2.5, 5. , 1.9],
               [6.4, 3.2, 4.5, 1.5],
               [5.2, 3.5, 1.5, 0.2],
               [5. , 3.6, 1.4, 0.2],
               [5.2, 4.1, 1.5, 0.1],
               [5.8, 2.7, 5.1, 1.9],
               [6. , 3.4, 4.5, 1.6],
               [6.7, 3.1, 4.7, 1.5],
               [5.4, 3.9, 1.3, 0.4],
               [5.4, 3.7, 1.5, 0.2],
               [5.5, 2.4, 3.7, 1. ],
               [6.3, 2.8, 5.1, 1.5],
               [6.4, 3.1, 5.5, 1.8],
               [6.6, 3. , 4.4, 1.4],
               [7.2, 3.6, 6.1, 2.5],
               [5.7, 2.9, 4.2, 1.3],
               [7.6, 3. , 6.6, 2.1],
               [5.6, 3. , 4.5, 1.5],
               [5.1, 3.5, 1.4, 0.2],
               [7.7, 2.8, 6.7, 2. ],
               [5.8, 2.7, 4.1, 1. ],
               [5.2, 3.4, 1.4, 0.2],
               [5. , 3.5, 1.3, 0.3],
               [5.1, 3.8, 1.9, 0.4],
               [5. , 2. , 3.5, 1. ],
               [6.3, 2.7, 4.9, 1.8],
               [4.8, 3.4, 1.9, 0.2],
               [5. , 3. , 1.6, 0.2],
               [5.1, 3.3, 1.7, 0.5],
               [5.6, 2.7, 4.2, 1.3],
               [5.1, 3.4, 1.5, 0.2],
               [5.7, 3. , 4.2, 1.2],
               [7.7, 3.8, 6.7, 2.2],
               [4.6, 3.2, 1.4, 0.2],
               [6.2, 2.9, 4.3, 1.3],
               [5.7, 2.5, 5. , 2. ],
               [5.5, 4.2, 1.4, 0.2],
               [6. , 3. , 4.8, 1.8],
               [5.8, 2.7, 5.1, 1.9],
               [6. , 2.2, 4. , 1. ],
               [5.4, 3. , 4.5, 1.5],
               [6.2, 3.4, 5.4, 2.3],
               [5.5, 2.3, 4. , 1.3],
               [5.4, 3.9, 1.7, 0.4],
               [5. , 2.3, 3.3, 1. ],
               [6.4, 2.7, 5.3, 1.9],
               [5. , 3.3, 1.4, 0.2],
               [5. , 3.2, 1.2, 0.2],
               [5.5, 2.4, 3.8, 1.1],
               [6.7, 3. , 5. , 1.7],
               [4.9, 3.1, 1.5, 0.2],
               [5.8, 2.8, 5.1, 2.4],
               [5. , 3.4, 1.5, 0.2],
               [5. , 3.5, 1.6, 0.6],
               [5.9, 3.2, 4.8, 1.8],
               [5.1, 2.5, 3. , 1.1],
               [6.9, 3.2, 5.7, 2.3],
               [6. , 2.7, 5.1, 1.6],
               [6.1, 2.6, 5.6, 1.4],
               [7.7, 3. , 6.1, 2.3],
               [5.5, 2.5, 4. , 1.3],
               [4.4, 2.9, 1.4, 0.2],
               [4.3, 3. , 1.1, 0.1],
               [6. , 2.2, 5. , 1.5],
               [7.2, 3.2, 6. , 1.8],
               [4.6, 3.1, 1.5, 0.2],
               [5.1, 3.5, 1.4, 0.3],
               [4.4, 3. , 1.3, 0.2],
               [6.3, 2.5, 4.9, 1.5],
               [6.3, 3.4, 5.6, 2.4],
               [4.6, 3.4, 1.4, 0.3],
               [6.8, 3. , 5.5, 2.1],
               [6.3, 3.3, 6. , 2.5],
               [4.7, 3.2, 1.3, 0.2],
               [6.1, 2.9, 4.7, 1.4],
               [6.5, 2.8, 4.6, 1.5],
               [6.2, 2.8, 4.8, 1.8],
               [7. , 3.2, 4.7, 1.4],
               [6.4, 3.2, 5.3, 2.3],
               [5.1, 3.8, 1.6, 0.2],
               [6.9, 3.1, 5.4, 2.1],
               [5.9, 3. , 4.2, 1.5],
               [6.5, 3. , 5.2, 2. ],
               [5.7, 2.6, 3.5, 1. ],
               [5.2, 2.7, 3.9, 1.4],
               [6.1, 3. , 4.6, 1.4],
               [4.5, 2.3, 1.3, 0.3],
               [6.6, 2.9, 4.6, 1.3],
               [5.5, 2.6, 4.4, 1.2],
               [5.3, 3.7, 1.5, 0.2],
               [5.6, 3. , 4.1, 1.3],
               [7.3, 2.9, 6.3, 1.8],
               [6.7, 3.3, 5.7, 2.1],
               [5.1, 3.7, 1.5, 0.4],
               [4.9, 2.4, 3.3, 1. ],
               [6.7, 3.3, 5.7, 2.5],
               [7.2, 3. , 5.8, 1.6],
               [4.9, 3.6, 1.4, 0.1],
               [6.7, 3.1, 5.6, 2.4],
               [4.9, 3. , 1.4, 0.2],
               [6.9, 3.1, 4.9, 1.5],
               [7.4, 2.8, 6.1, 1.9],
               [6.3, 2.9, 5.6, 1.8],
               [5.7, 2.8, 4.1, 1.3],
               [6.5, 3. , 5.5, 1.8],
               [6.3, 2.3, 4.4, 1.3],
               [6.4, 2.9, 4.3, 1.3],
               [5.6, 2.8, 4.9, 2. ],
               [5.9, 3. , 5.1, 1.8],
               [5.4, 3.4, 1.7, 0.2],
               [6.1, 2.8, 4. , 1.3],
               [4.9, 2.5, 4.5, 1.7],
               [5.8, 4. , 1.2, 0.2],
               [5.8, 2.6, 4. , 1.2],
               [7.1, 3. , 5.9, 2.1]])
```

In [ ]:

```
# Create a Decision Tree Classifier with the ID3 algorithm
# In sklearn, you can set the criterion to "entropy" to use the ID3 algorithm
clf = DecisionTreeClassifier(criterion='entropy', random_state=42) #
```

In [ ]:

```
```

In [19]:

```
# Train the model
clf.fit(X_train, y_train)
```

Out[19]:

```
▾ DecisionTreeClassifier  i ?
                          (https://
                          scikit-
Parameters                learn.org/1.7/
                          modules/
                          generated/
```

In [ ]:

```
```

In [21]:

```
# Make predictions
y_pred = clf.predict(X_test)
```

In [ ]:

```
y_pred # model Predicted values
```

Out[ ]:

```
array([1, 0, 2, 1, 1, 0, 1, 2, 1, 1, 2, 0, 0, 0, 0, 1, 2, 1, 1, 2, 0, 2,
       0, 2, 2, 2, 2, 2, 0, 0])
```

In [ ]:

```
y_test #Actual target values
```

Out[ ]:

```
array([1, 0, 2, 1, 1, 0, 1, 2, 1, 1, 2, 0, 0, 0, 0, 1, 2, 1, 1, 2, 0, 2,
       0, 2, 2, 2, 2, 2, 0, 0])
```

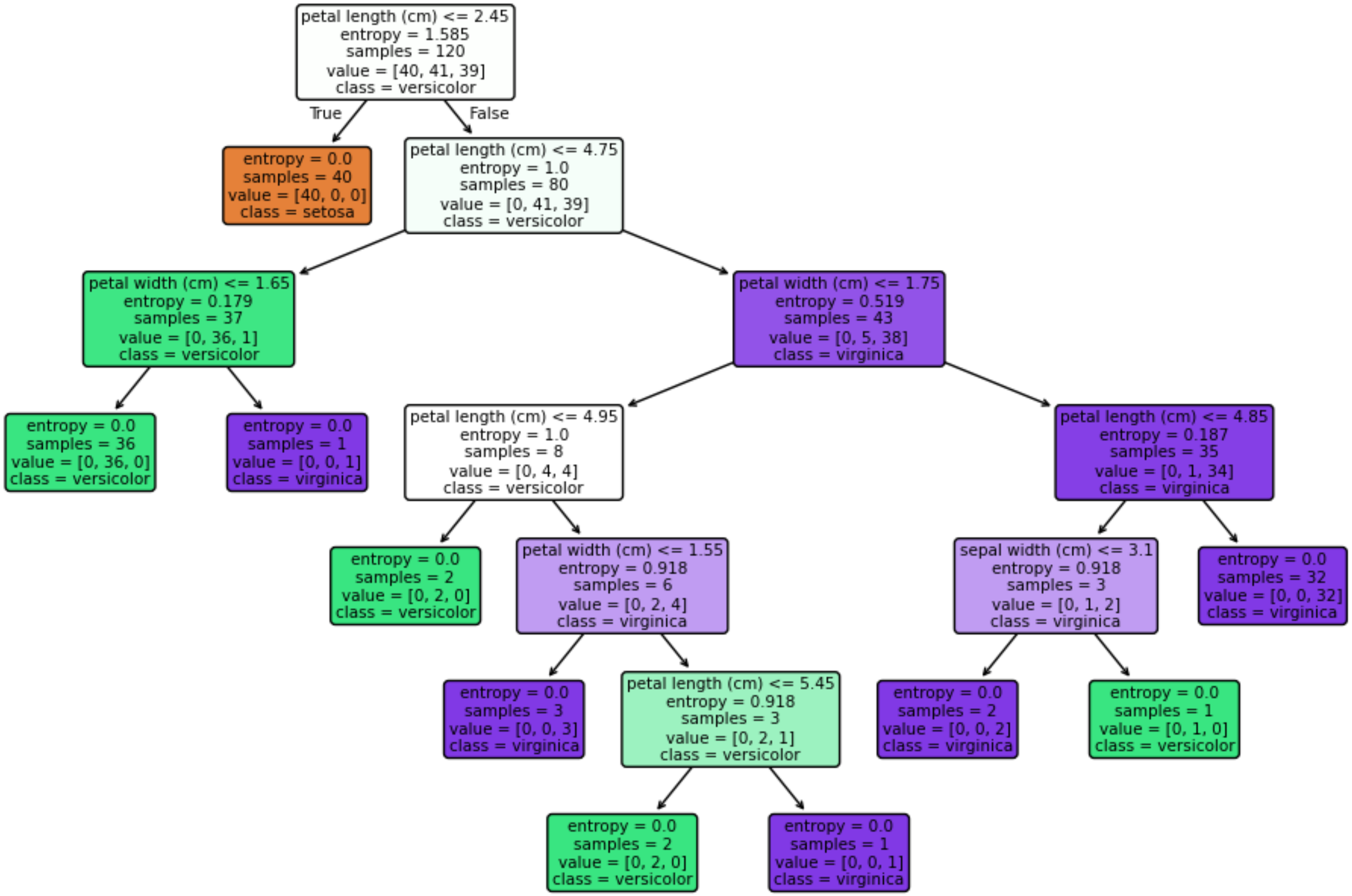
In [24]:

```
# Evaluate the model
accuracy = clf.score(X_test, y_test)
print(f"Accuracy: {accuracy:.2f}")
```

```
Accuracy: 1.00
```

```
In [25]: # Display the decision tree
         plt.figure(figsize=(12, 8))
         plot_tree(clf, filled=True, feature_names=iris.feature_names, class_names=iris.target_names, rounded=True)
         plt.title("Decision Tree using ID3 Algorithm")
         plt.show()
```

Decision Tree using ID3 Algorithm



In [26]:

```
# Print the tree structure
tree_structure = export_text(clf, feature_names=iris.feature_names)
print(tree_structure)
```

```
|--- petal length (cm) <= 2.45
|   |--- class: 0
|--- petal length (cm) >  2.45
|   |--- petal length (cm) <= 4.75
|   |   |--- petal width (cm) <= 1.65
|   |   |   |--- class: 1
|   |   |--- petal width (cm) >  1.65
|   |   |   |--- class: 2
|   |--- petal length (cm) >  4.75
|   |   |--- petal width (cm) <= 1.75
|   |   |   |--- petal length (cm) <= 4.95
|   |   |   |   |--- class: 1
|   |   |   |--- petal length (cm) >  4.95
|   |   |   |   |--- petal width (cm) <= 1.55
|   |   |   |   |   |--- class: 2
|   |   |   |   |--- petal width (cm) >  1.55
|   |   |   |   |   |--- petal length (cm) <= 5.45
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- petal length (cm) >  5.45
|   |   |   |   |   |   |--- class: 2
|   |   |--- petal width (cm) >  1.75
|   |   |   |--- petal length (cm) <= 4.85
|   |   |   |   |--- sepal width (cm) <= 3.10
|   |   |   |   |   |--- class: 2
|   |   |   |   |--- sepal width (cm) >  3.10
|   |   |   |   |   |--- class: 1
|   |   |   |--- petal length (cm) >  4.85
|   |   |   |   |--- class: 2
```

In [ ]:

```
clf.predict([[4.5, 6.9, 6.7, 9.9]]) # to test the new sample
```

Out[ ]:

```
array([2])
```