In [1]:
```
import pandas as pd
import numpy as np
```

```
C:\Users\vtu10\anaconda3\lib\site-packages\pandas\core\computation\expres
sions.py:21: UserWarning: Pandas requires version '2.8.4' or newer of 'nu
mexpr' (version '2.7.3' currently installed).
  from pandas.core.computation.check import NUMEXPR_INSTALLED
C:\Users\vtu10\anaconda3\lib\site-packages\pandas\core\arrays\masked.py:6
0: UserWarning: Pandas requires version '1.3.6' or newer of 'bottleneck'
(version '1.3.2' currently installed).
  from pandas.core import (
```

In [ ]:
```
#Reading the file
```

In [2]:
```
data=pd.read_csv("C:\\Users\\vtu10\\Downloads\\IRIS.csv")
```

In [3]:
```
data
```

Out[3]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [4]:
```
data.head()
```

Out[4]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [9]:
```
# Separate the features (sepal/petal measurements) and labels (species)
X = data[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']]  #
y = data['species']  # Labels (species)
```

In [10]:
```
# Convert the categorical labels to numerical values if necessary
y = pd.factorize(y)[0]
```

In [12]:
```
from sklearn.model_selection import train_test_split
```

In [13]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ra
```

In [11]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [14]:
```
# Initialize the Decision Tree Classifier
clf = DecisionTreeClassifier()
```

In [15]:
```
# Train the model on the training data
clf.fit(X_train, y_train)
```

Out[15]:
```
DecisionTreeClassifier()
```

In [16]:
```
# Make predictions on the test data
y_pred = clf.predict(X_test)
```

In [17]:
```
from sklearn.metrics import accuracy_score, classification_report
```

In [18]:
```
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)
```

In [19]:
```
print(f"Model Accuracy: {accuracy:.2f}")
print("Classification Report:")
print(report)
```

```
Model Accuracy: 1.00
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

In [ ]: