Lab Task: 04

***————————————786**
—————————————*

Name: Muhammad Sherjeel Akhtar

Roll No: 20p-0101

Subject: Artificial Intelligence Lab

Submitted To Respected
Ma’am: Hurmat Hidayat

Section: BCS-6C

Task Requirements:

1. Load the iris dataset using scikit-learn.
2. Split the dataset into training and testing sets with 70% of the data for training and 30% for testing.
3. Train a decision tree classifier using the entropy criterion and evaluate its accuracy on the testing set.
4. Train another decision tree classifier using the gini criterion and evaluate its accuracy on the testing set.
5. Compare the performance of the two classifiers and discuss the differences in terms of accuracy.
6. Visualize the decision tree for each classifier and compare the tree structure and feature importance.

Note: You can use the tree.plot_tree() function to visualize the decision tree, and the feature_importances_ attribute to get the feature importances for each classifier.

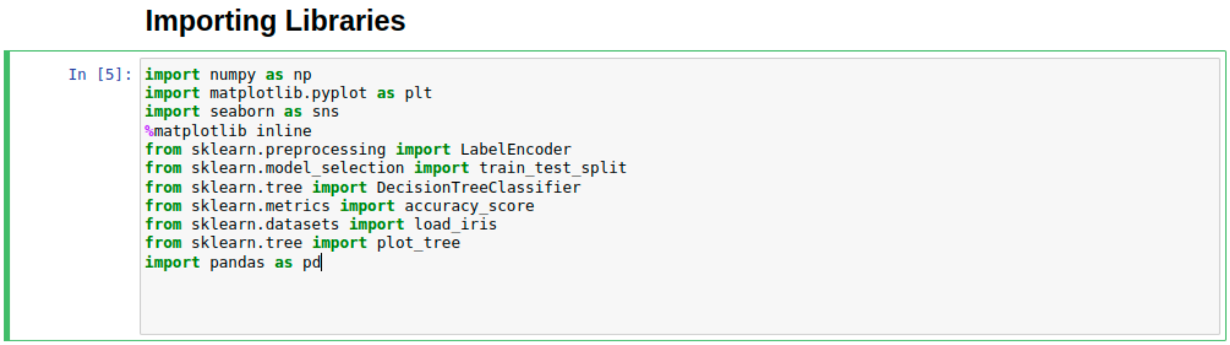
The **feature importance values** can be used to understand which features have the strongest association with the target variable and how they contribute to the model's prediction.

Answer:

Importing Libraries:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.datasets import load_iris
from sklearn.tree import plot_tree
import pandas as pd
```

Visual Demonstration:



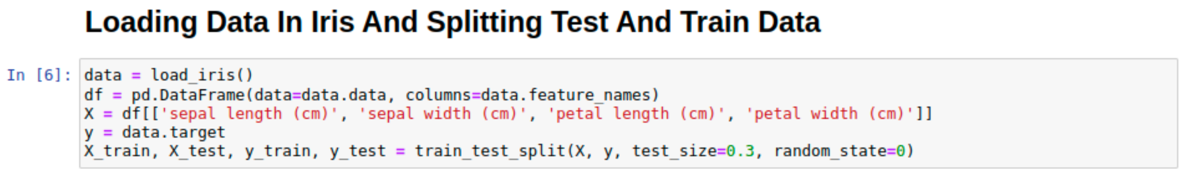
Importing Libraries

In [5]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.datasets import load_iris
from sklearn.tree import plot_tree
import pandas as pd
```

Loading Data In Iris And Splitting Test And Train Data:

```
data = load_iris()
df = pd.DataFrame(data=data.data, columns=data.feature_names)
X = df[['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']]
y = data.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

Visual Demonstration:



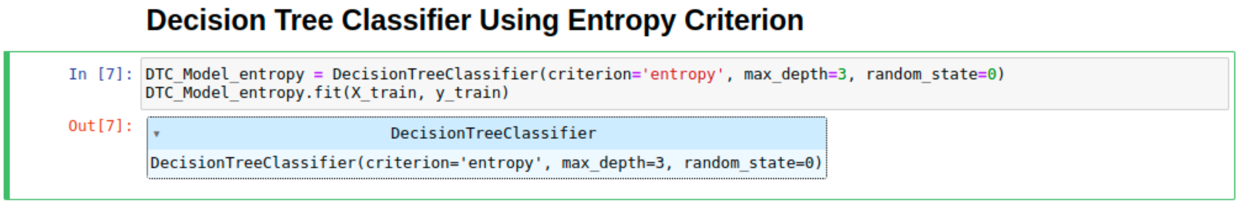
Loading Data In Iris And Splitting Test And Train Data

```
In [6]: data = load_iris()
        df = pd.DataFrame(data=data.data, columns=data.feature_names)
        X = df[['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']]
        y = data.target
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

Decision Tree Classifier Using Entropy Criterion:

```
DTC_Model_entropy = DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=0)
DTC_Model_entropy.fit(X_train, y_train)
```

Visual Demonstration:



Decision Tree Classifier Using Entropy Criterion

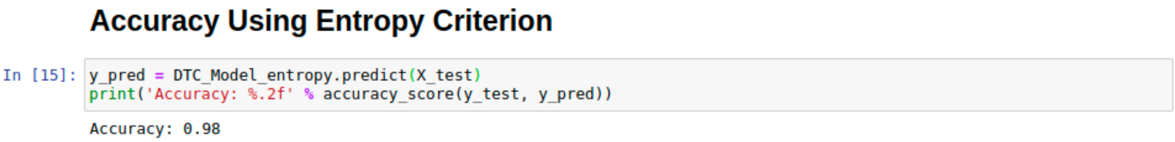
```
In [7]: DTC_Model_entropy = DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=0)
        DTC_Model_entropy.fit(X_train, y_train)

Out[7]: ▾ DecisionTreeClassifier
        DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=0)
```

Accuracy Using Entropy Criterion:

```
y_pred = DTC_Model_entropy.predict(X_test)
print('Accuracy: %.2f' % accuracy_score(y_test, y_pred))
```

Visual Demonstration:



Accuracy Using Entropy Criterion

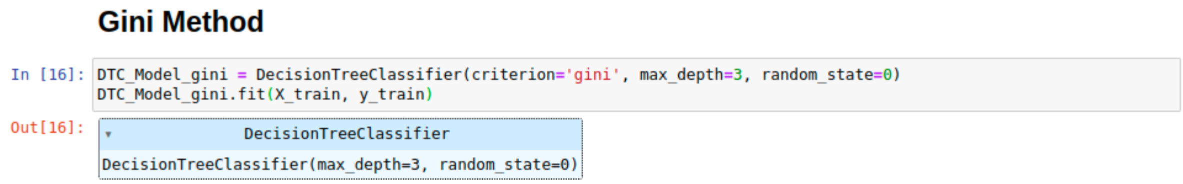
```
In [15]: y_pred = DTC_Model_entropy.predict(X_test)
         print('Accuracy: %.2f' % accuracy_score(y_test, y_pred))

         Accuracy: 0.98
```

Gini Method:

```
DTC_Model_gini = DecisionTreeClassifier(criterion='gini', max_depth=3, random_state=0)
DTC_Model_gini.fit(X_train, y_train)
```

Visual Demonstration:



Gini Method

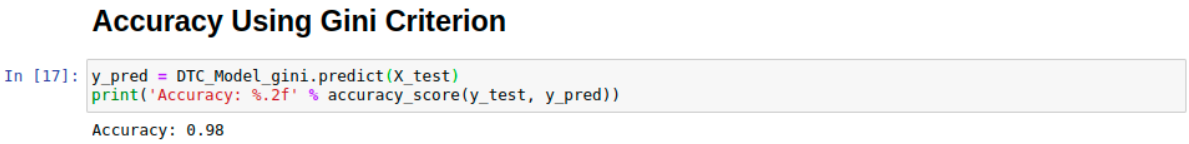
```
In [16]: DTC_Model_gini = DecisionTreeClassifier(criterion='gini', max_depth=3, random_state=0)
         DTC_Model_gini.fit(X_train, y_train)

Out[16]: ▾ DecisionTreeClassifier
         DecisionTreeClassifier(max_depth=3, random_state=0)
```

Accuracy Using Gini Criterion:

```
y_pred = DTC_Model_gini.predict(X_test)
print('Accuracy: %.2f' % accuracy_score(y_test, y_pred))
```

Visual Demonstration:



Accuracy Using Gini Criterion

```
In [17]: y_pred = DTC_Model_gini.predict(X_test)
         print('Accuracy: %.2f' % accuracy_score(y_test, y_pred))

         Accuracy: 0.98
```

Method 1: Decision Tree Classifier Using Entropy:

Accuracy Achieved: 0.98

Max Depth: 3

Random Depth: 0

Method 2: Decision Tree Classifier Using Gini Criterion:

Accuracy Achieved: 0.98

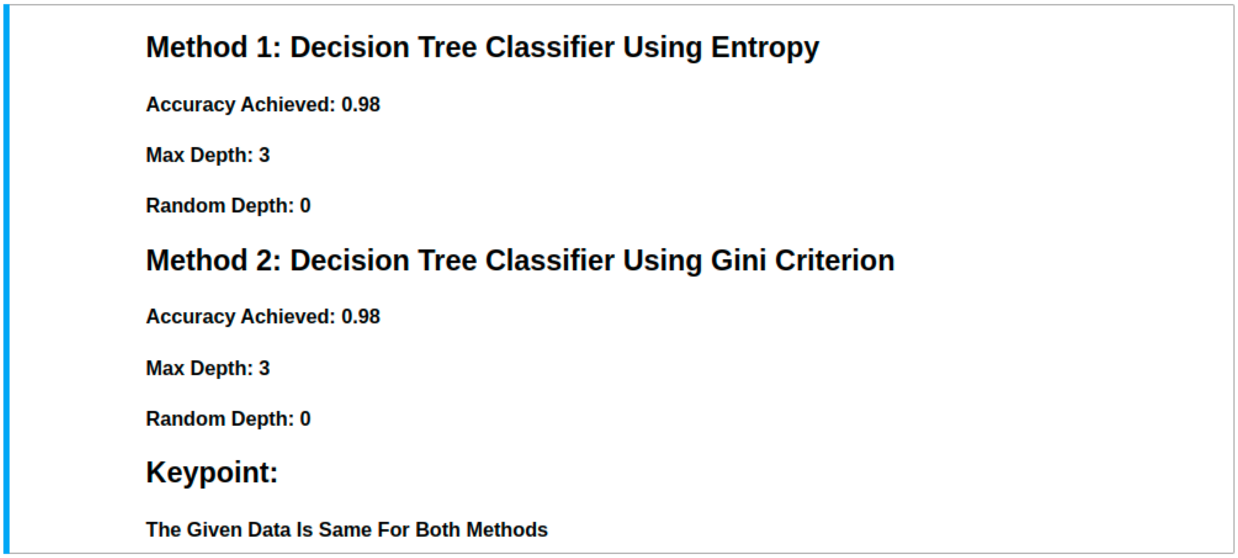
Max Depth: 3

Random Depth: 0

Key-point:

''The Given Data Is Same For Both Methods'

Note-Preview Of Comparison:



Method 1: Decision Tree Classifier Using Entropy

Accuracy Achieved: 0.98

Max Depth: 3

Random Depth: 0

Method 2: Decision Tree Classifier Using Gini Criterion

Accuracy Achieved: 0.98

Max Depth: 3

Random Depth: 0

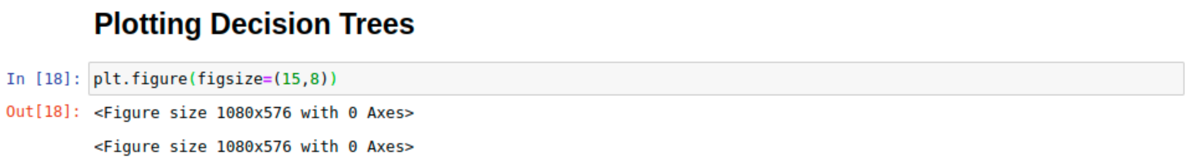
Keypoint:

The Given Data Is Same For Both Methods

Plotting Decision Trees:

```
plt.figure(figsize=(15,8))
```

Visual Demonstration:



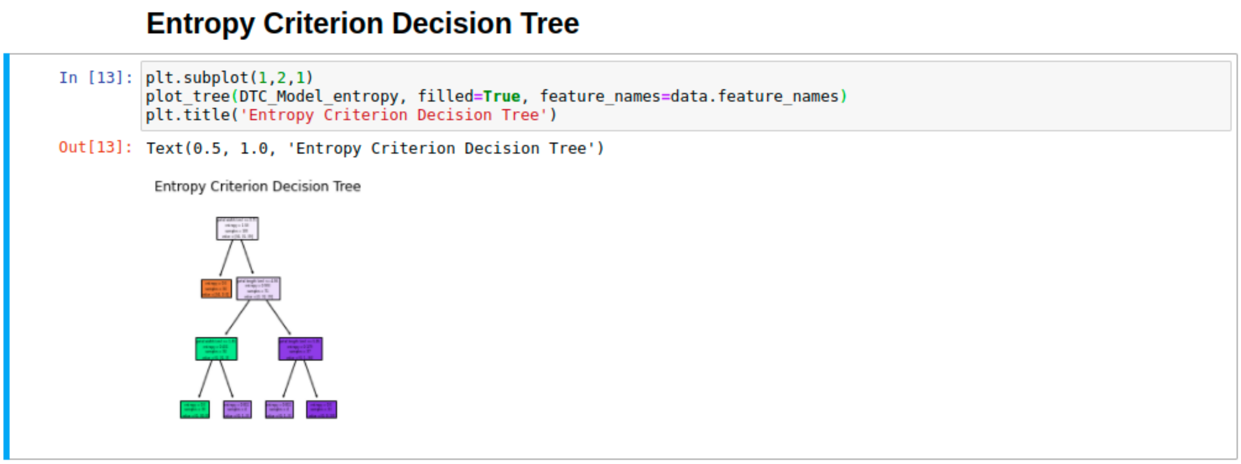
Plotting Decision Trees

```
In [18]: plt.figure(figsize=(15,8))
Out[18]: <Figure size 1080x576 with 0 Axes>
         <Figure size 1080x576 with 0 Axes>
```

For Entropy Criterion Decision Tree:

```
plt.subplot(1,2,1)
plot_tree(DTC_Model_entropy, filled=True, feature_names=data.feature_names)
plt.title('Entropy Criterion Decision Tree')
```

Visual Demonstration:



Entropy Criterion Decision Tree

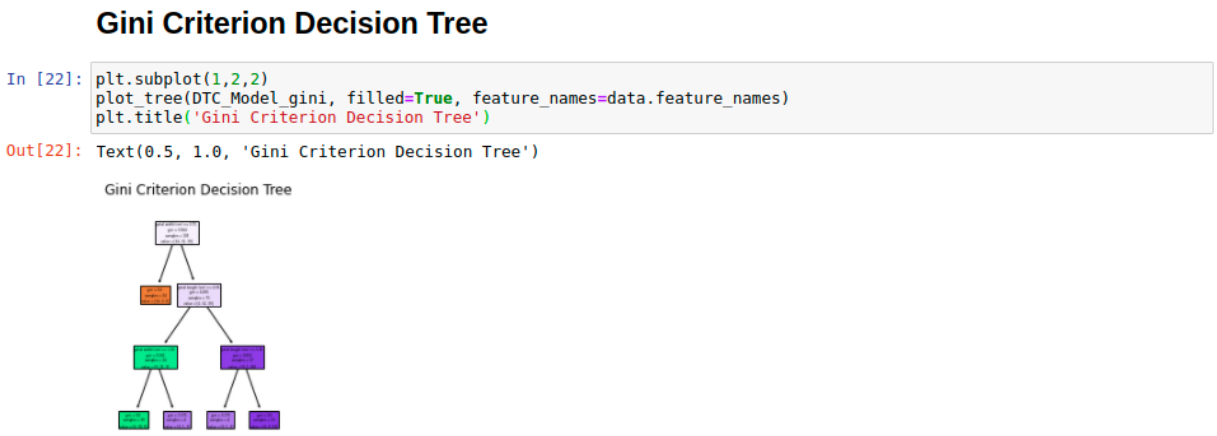
```
In [13]: plt.subplot(1,2,1)
         plot_tree(DTC_Model_entropy, filled=True, feature_names=data.feature_names)
         plt.title('Entropy Criterion Decision Tree')

Out[13]: Text(0.5, 1.0, 'Entropy Criterion Decision Tree')
```

For Gini Criterion Decision Tree:

```
plt.subplot(1,2,2)
plot_tree(DTC_Model_gini, filled=True, feature_names=data.feature_names)
plt.title('Gini Criterion Decision Tree')
```

Visual Demonstration:



Gini Criterion Decision Tree

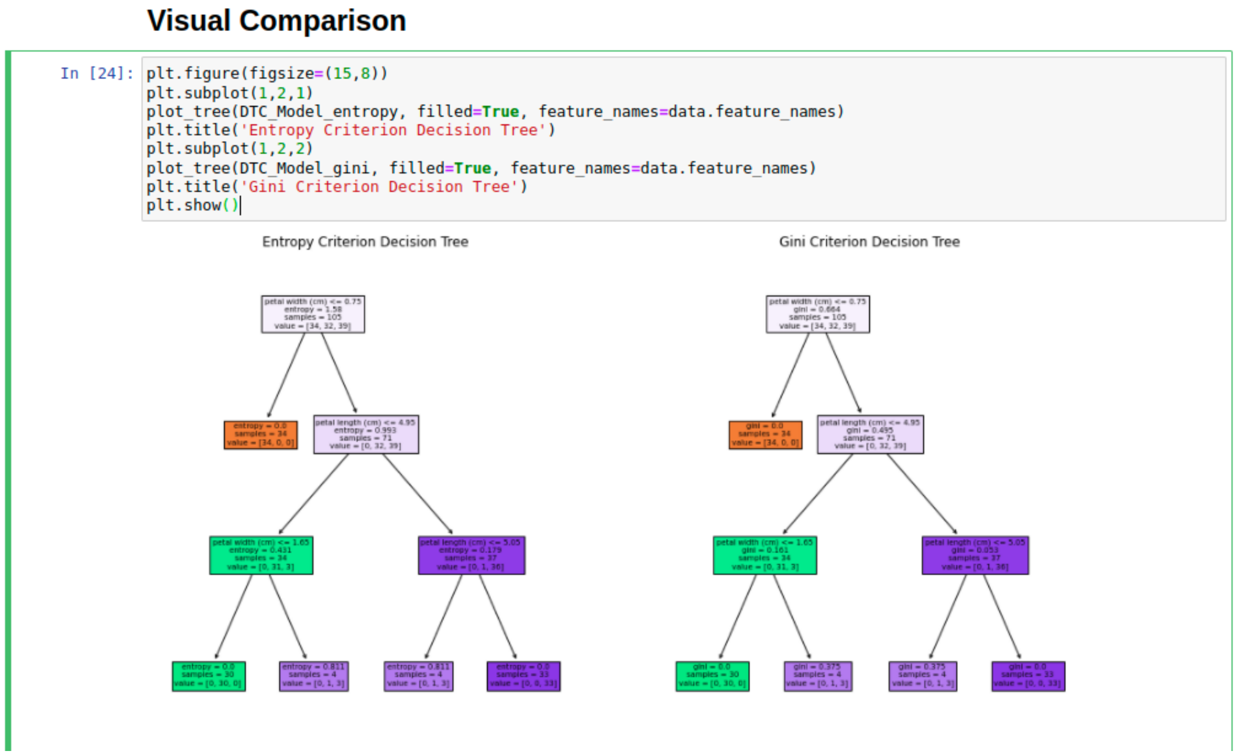
```
In [22]: plt.subplot(1,2,2)
         plot_tree(DTC_Model_gini, filled=True, feature_names=data.feature_names)
         plt.title('Gini Criterion Decision Tree')

Out[22]: Text(0.5, 1.0, 'Gini Criterion Decision Tree')
```

Plot-Showing And Comparison:

```
plt.figure(figsize=(15,8))
plt.subplot(1,2,1)
plot_tree(DTC_Model_entropy, filled=True, feature_names=data.feature_names)
plt.title('Entropy Criterion Decision Tree')
plt.subplot(1,2,2)
plot_tree(DTC_Model_gini, filled=True, feature_names=data.feature_names)
plt.title('Gini Criterion Decision Tree')
plt.show()
```

Visual Demonstration:



Visual Comparison

In [24]:

```
plt.figure(figsize=(15,8))
plt.subplot(1,2,1)
plot_tree(DTC_Model_entropy, filled=True, feature_names=data.feature_names)
plt.title('Entropy Criterion Decision Tree')
plt.subplot(1,2,2)
plot_tree(DTC_Model_gini, filled=True, feature_names=data.feature_names)
plt.title('Gini Criterion Decision Tree')
plt.show()
```

Entropy Criterion Decision Tree

Gini Criterion Decision Tree

“FIN.”