

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.tree import DecisionTreeClassifier, plot_tree
from itertools import combinations
```

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
file_path='/content/drive/My Drive/machine learning/Iris.csv'
df=pd.read_csv(file_path)
df.head()
```

```
{
  "summary": {
    "name": "df",
    "rows": 150,
    "fields": [
      {
        "column": "Id",
        "properties": {
          "dtype": "number",
          "std": 43,
          "min": 1,
          "max": 150,
          "num_unique_values": 150,
          "samples": [
            74,
            19,
            119
          ],
          "semantic_type": "",
          "description": ""
        },
        "column": "SepalLengthCm",
        "properties": {
          "dtype": "number",
          "std": 0.8280661279778629,
          "min": 4.3,
          "max": 7.9,
          "num_unique_values": 35,
          "samples": [
            6.2,
            4.5,
            5.6
          ],
          "semantic_type": "",
          "description": ""
        },
        "column": "SepalWidthCm",
        "properties": {
          "dtype": "number",
          "std": 0.4335943113621737,
          "min": 2.0,
          "max": 4.4,
          "num_unique_values": 23,
          "samples": [
            2.3,
            4.0,
            3.5
          ],
          "semantic_type": "",
          "description": ""
        },
        "column": "PetalLengthCm",
        "properties": {
          "dtype": "number",
          "std": 1.7644204199522617,
          "min": 1.0,
          "max": 6.9,
          "num_unique_values": 43,
          "samples": [
            6.7,
            3.8,
            3.7
          ],
          "semantic_type": "",
          "description": ""
        },
        "column": "PetalWidthCm",
        "properties": {
          "dtype": "number",
          "std": 0.7631607417008414,
          "min": 0.1,
          "max": 2.5,
          "num_unique_values": 22,
          "samples": [
            0.2,
            1.2,
            1.3
          ],
          "semantic_type": "",
          "description": ""
        },
        "column": "Species",
        "properties": {
          "dtype": "category",
          "num_unique_values": 3,
          "samples": [
            "Iris-setosa",
            "Iris-versicolor"
          ]
        }
      ]
    }
  }
}
```

```
\ "Iris-virginica" \n      ], \n      \ "semantic_type" \: \ "\", \n \ "description" \: \ "\" \n      } \n      } \n      ] \n \n }", "type": "dataframe", "variable_name": "df" }
```

```
X = df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm',  
'PetalWidthCm']]  
y = df['Species']
```

```
print("\nPlot colors: red, yellow, blue for 3 classes")  
print("Plot step = 0.02")
```

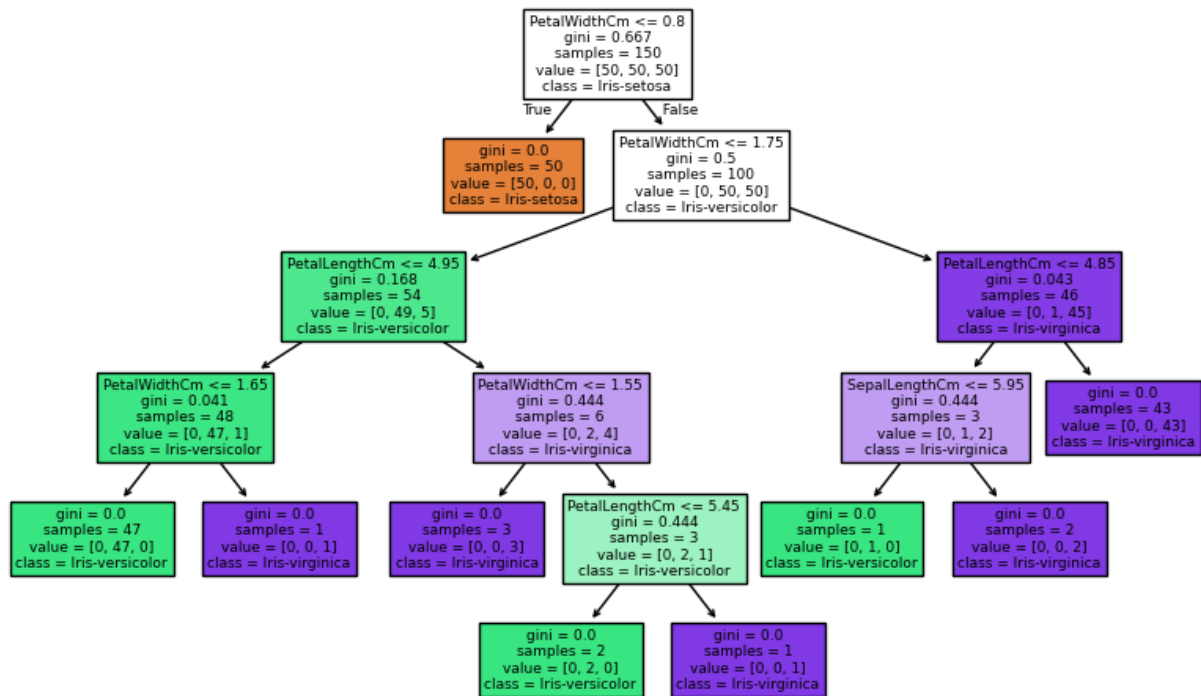
```
Plot colors: red, yellow, blue for 3 classes  
Plot step = 0.02
```

```
model = DecisionTreeClassifier()  
model.fit(X, y)  
print("\nDecision Tree model trained successfully!")
```

```
Decision Tree model trained successfully!
```

```
plt.figure(figsize=(10,6))  
plot_tree(model,  
          feature_names=['SepalLengthCm', 'SepalWidthCm',  
                        'PetalLengthCm', 'PetalWidthCm'],  
          class_names=model.classes_, filled=True)  
  
plt.title("Decision Tree for Iris Dataset")  
plt.show()
```

Decision Tree for Iris Dataset



```
print("\nDecision Tree plotted successfully!")
```

Decision Tree plotted successfully!