

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report

import matplotlib.pyplot as plt
import pandas as pd
```

```
iris = load_iris()
X = iris.data
y = iris.target
feature_names = iris.feature_names
target_names = iris.target_names
```

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)
```

```
clf = DecisionTreeClassifier(criterion="entropy", random_state=42)
clf.fit(X_train, y_train)
```

```
▼ DecisionTreeClassifier ⓘ ?  
► Parameters
```

```
y_pred = clf.predict(X_test)
print("🌳 Decision Tree Results")
print("-----")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred, target_names=target_names))
```

```
🌳 Decision Tree Results
-----
Accuracy: 0.9777777777777777

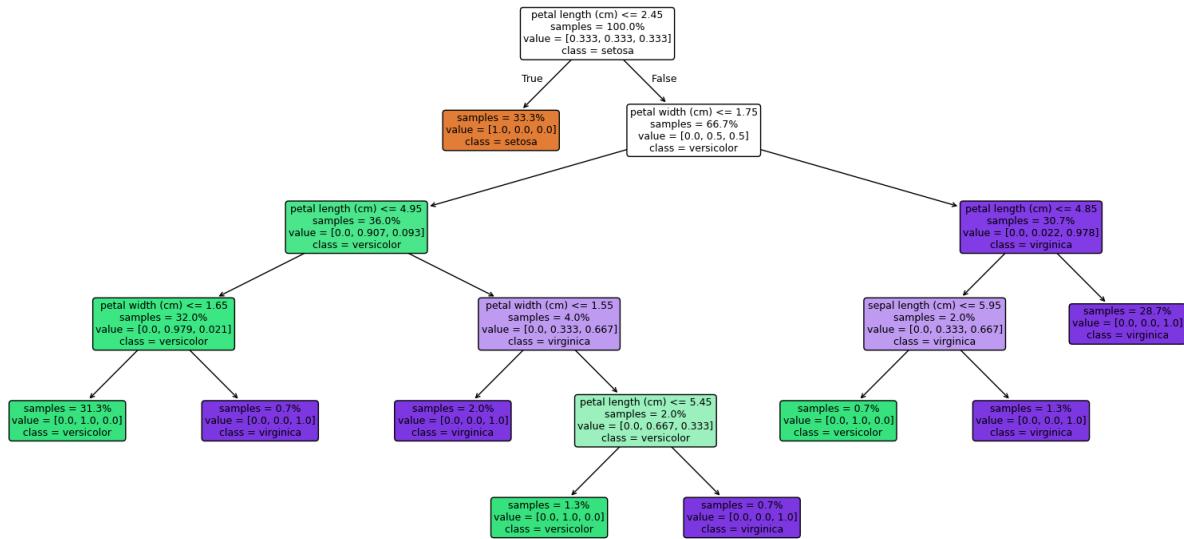
Classification Report:
precision    recall    f1-score   support
setosa       1.00     1.00      1.00      19
versicolor   0.93     1.00      0.96      13
virginica    1.00     0.92      0.96      13

accuracy        0.98      0.98      0.98      45
macro avg     0.98     0.97      0.97      45
weighted avg  0.98     0.98      0.98      45
```

```
plt.figure(figsize=(16, 8))
plot_tree(
    clf,
    filled=True,
    feature_names=feature_names,
    class_names=target_names,
    rounded=True,
    proportion=True,
    impurity=False,
    fontsize=9
)

plt.title("Decision Tree for Iris Dataset (Dendrogram Style)", fontsize=14, pad=20)
plt.axis("off")
plt.tight_layout()
plt.show()
```

Decision Tree for Iris Dataset (Dendrogram Style)

Start coding or generate with AI.