+ Code + Text

Visualizing the Iris Dataset with Scikit-Learn

1. Import Libraries

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree
import matplotlib.pyplot as plt
```

2. Load the Iris Dataset

```
# Load the Iris dataset
iris = load_iris()
X = iris.data # Features
y = iris.target # Target labels

# Convert to DataFrame for better visualization (optional)
df = pd.DataFrame(data=np.c_[iris['data'], iris['target']], columns=iris['feature_names'] + ['target'])
```

3. Split the Dataset into Training and Testing Sets

```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

4. Train the Decision Tree Classifier

```
# Initialize the Decision Tree Classifier
clf = DecisionTreeClassifier(random_state=42)

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

```
▾     DecisionTreeClassifier      i ?
DecisionTreeClassifier(random_state=42)
```

5. Make Predictions and Evaluate the Model

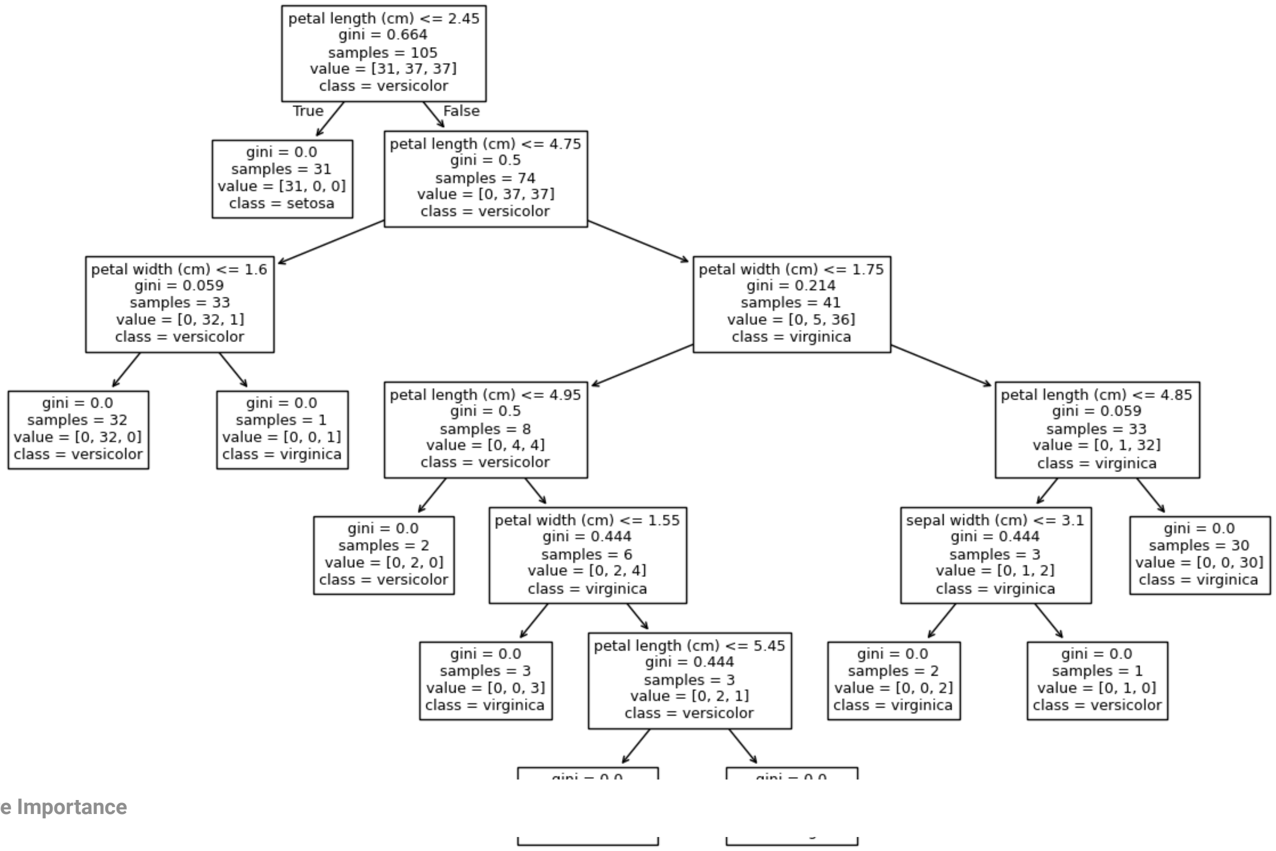
```
# Predict the test set results
y_pred = clf.predict(X_test)


# Calculate the accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}%')
```

```
Accuracy: 100.00%
```

6. Visualize the Decision Tree

```
# Plot the decision tree
plt.figure(figsize=(15,10))
tree.plot_tree(clf, feature_names=iris.feature_names, class_names=iris.target_names)
plt.show()
```



7. Feature Importance

```
# Print feature importances

importances = clf.feature_importances_

for feature, importance in zip(iris.feature_names, importances):
      print(f'{feature}: {importance:.4f}')
```

```
sepal length (cm): 0.0000
sepal width (cm): 0.0191
petal length (cm): 0.8933
petal width (cm): 0.0876
```

Conclusion:

- The Decision Tree works perfectly for the Iris dataset.
- It can accurately predict the type of Iris flower.
- But, because the Iris dataset is small and simple, the model might be "too specialized" and might not work as well on new, different data.
- It's a good idea to test it on more data to be sure it's really that good!
- While 100% accuracy on the Iris dataset with a Decision Tree is achievable and might be okay for this specific task, it's crucial to understand the potential for overfitting and evaluate your model thoroughly to ensure it generalizes well to new data.