Decision tree as classifier

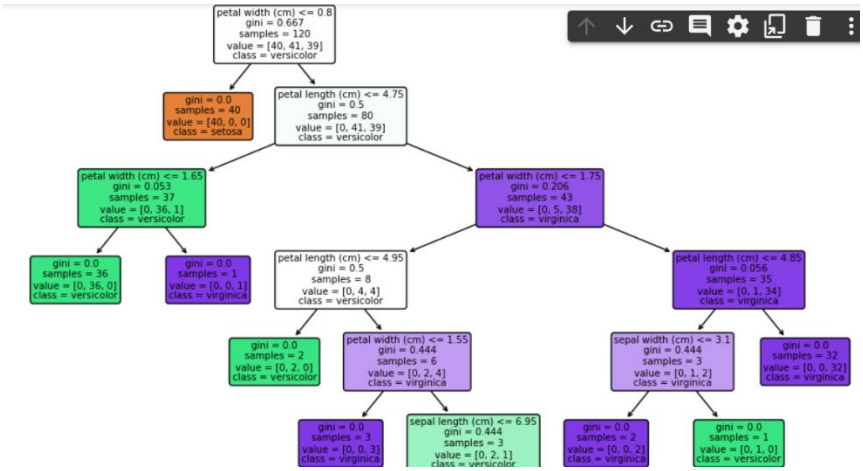
```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
# Load the Iris dataset (as an example)
iris = load_iris()
X = iris.data
y = iris.target
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a decision tree classifier
clf = DecisionTreeClassifier()

# Train the classifier on the training data
clf.fit(X_train, y_train)

# Make predictions on the test set
y_pred = clf.predict(X_test)

# Evaluate the accuracy of the classifier
```

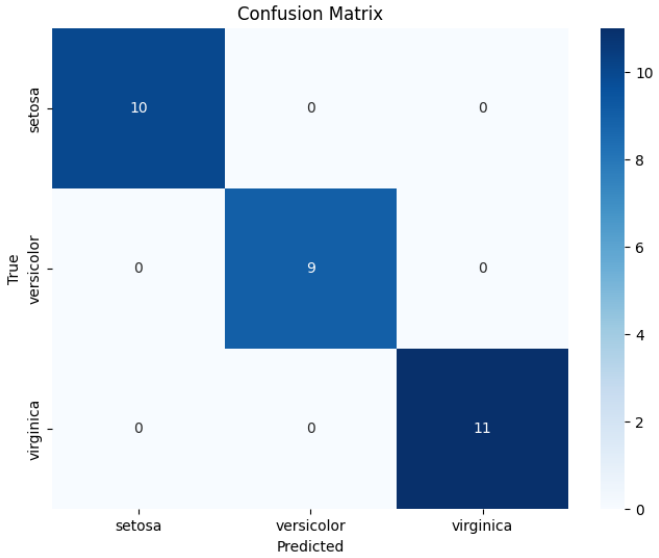
```
accuracy = accuracy_score(y_test,
y_pred)
print(f"Accuracy: {accuracy}")

# Plot the decision tree structure
plt.figure(figsize=(12, 8))
plot_tree(clf, filled=True,
feature_names=iris.feature_names,
class_names=iris.target_names,
rounded=True)
plt.show()

# Plot confusion matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d",
cmap="Blues",
```

```
xticklabels=iris.target_names,
yticklabels=iris.target_names)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
```

Output



Confusion matrix


Confusion Matrix
setosa
versicolor
virginica
True
10
0
0
0
9
0
0
0
11
setosa
versicolor
virginica
Predicted
10
8
6
4
2
0