**from sklearn.tree import DecisionTreeClassifier, plot\_tree**

**from sklearn.datasets import load\_iris**

**from sklearn.model\_selection import train\_test\_split**

**import matplotlib.pyplot as plt**

**iris = load\_iris()**

**X, y = iris.data, iris.target**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)**

**tree = DecisionTreeClassifier(random\_state=42)**

**tree.fit(X\_train, y\_train)**

**# Compute the cost complexity pruning path**

**path = tree.cost\_complexity\_pruning\_path(X\_train, y\_train)**

**ccp\_alphas, impurities = path.ccp\_alphas, path.impurities**

**accuracy\_scores = []**

**# Train and evaluate decision trees for each alpha value**

**for ccp\_alpha in ccp\_alphas:**

**pruned\_tree = DecisionTreeClassifier(ccp\_alpha=ccp\_alpha, random\_state=42)**

**pruned\_tree.fit(X\_train, y\_train)**

**y\_pred = pruned\_tree.predict(X\_test)**

**accuracy = accuracy\_score(y\_test, y\_pred)**

**accuracy\_scores.append(accuracy)**

**# Plot the accuracy scores against ccp\_alpha values**

**plt.figure(figsize=(10, 6))**

**plt.plot(ccp\_alphas, accuracy\_scores, marker='o', linestyle='-', color='b')**

**plt.xlabel('ccp\_alpha')**

**plt.ylabel('Accuracy')**

**plt.title('Accuracy vs. ccp\_alpha')**

**plt.grid(True)**

**plt.show()**

**# Print the results**

**for alpha, accuracy in zip(ccp\_alphas, accuracy\_scores):**

**print(f"ccp\_alpha: {alpha:.4f}, Accuracy: {accuracy:.4f}")**