**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ**

**Лабораторна робота №**8

з дисципліни «Інтелектуальний аналіз та візуалізація даних»

тема «КЛАСИФІКАЦІЯ, ДЕРЕВА РІШЕНЬ»

**Варіант №8**

Виконала студентка

групи КН-21

Іванова А.О.

Перевірив(-ла):

МІнаєва Ю.І.

**Київ – 2025**

**Мета роботи:**

Вивчення алгоритмів побудови дерев класифікації.

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn import datasets

from sklearn.datasets import load\_wine

from sklearn.tree import DecisionTreeClassifier, plot\_tree

from sklearn.model\_selection import train\_test\_split, GridSearchCV

from sklearn.metrics import accuracy\_score

# Завантаження датасету Wine

wine = load\_wine()

X = wine.data

y = wine.target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,

random\_state=42)

full\_tree = DecisionTreeClassifier(random\_state=42)

full\_tree.fit(X\_train, y\_train)

y\_pred\_full = full\_tree.predict(X\_test)

full\_tree\_acc = accuracy\_score(y\_test, y\_pred\_full)

print(f"Unpruned Decision Tree Accuracy: {full\_tree\_acc:.4f}")

Unpruned Decision Tree Accuracy: 0.9444

pre\_pruned\_tree = DecisionTreeClassifier (max\_depth=5,

min\_samples\_split=10, min\_samples\_leaf=5, random\_state=42)

pre\_pruned\_tree.fit(X\_train, y\_train)

y\_pred\_prepruned = pre\_pruned\_tree.predict(X\_test)

pre\_pruned\_acc = accuracy\_score(y\_test, y\_pred\_prepruned)

print(f"Pre-Pruned Decision Tree Accuracy: {pre\_pruned\_acc:.4f}")

Pre-Pruned Decision Tree Accuracy: 0.9444

param\_grid = {

'max\_depth': [3, 5, 10, None],

'min\_samples\_split': [2, 5, 10],

'min\_samples\_leaf': [1, 5, 10]

}

grid\_search = GridSearchCV(DecisionTreeClassifier(random\_state=42),

param\_grid, cv=5, scoring='accuracy')

grid\_search.fit(X\_train, y\_train)

best\_params = grid\_search.best\_params\_

best\_tree = grid\_search.best\_estimator\_

y\_pred\_gridsearch = best\_tree.predict(X\_test)

gridsearch\_acc = accuracy\_score(y\_test, y\_pred\_gridsearch)

print(f"Grid Search Decision Tree Accuracy: {gridsearch\_acc:.4f} with params: {best\_params}")

Grid Search Decision Tree Accuracy: 0.9444 with params: {'max\_depth': 3, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2}

ccp\_alphas = full\_tree.cost\_complexity\_pruning\_path(X\_train,

y\_train)['ccp\_alphas']

accuracies = []

for alpha in ccp\_alphas:

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

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

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

accuracies.append(accuracy\_score(y\_test, y\_pred\_pruned))

best\_alpha = ccp\_alphas[np.argmax(accuracies)]

post\_pruned\_tree = DecisionTreeClassifier(random\_state=42,

ccp\_alpha=best\_alpha)

post\_pruned\_tree.fit(X\_train, y\_train)

y\_pred\_postpruned = post\_pruned\_tree.predict(X\_test)

post\_pruned\_acc = accuracy\_score(y\_test, y\_pred\_postpruned)

print(f"Post-Pruned Decision Tree Accuracy: {post\_pruned\_acc:.4f} (Best alpha: {best\_alpha:.4f})")

Post-Pruned Decision Tree Accuracy: 0.9444 (Best alpha: 0.0000)

fig, axs = plt.subplots(2, 2, figsize=(20, 16))

plot\_tree(full\_tree, filled=True, feature\_names=wine.feature\_names, class\_names=wine.target\_names, ax=axs[0, 0])

axs[0, 0].set\_title(f"Повне дерево (acc: {full\_tree\_acc:.4f})")

plot\_tree(pre\_pruned\_tree, filled=True, feature\_names=wine.feature\_names, class\_names=wine.target\_names, ax=axs[0, 1])

axs[0, 1].set\_title(f"Попередньо обрізане дерево (acc: {pre\_pruned\_acc:.4f})")

plot\_tree(best\_tree, filled=True, feature\_names=wine.feature\_names, class\_names=wine.target\_names, ax=axs[1, 0])

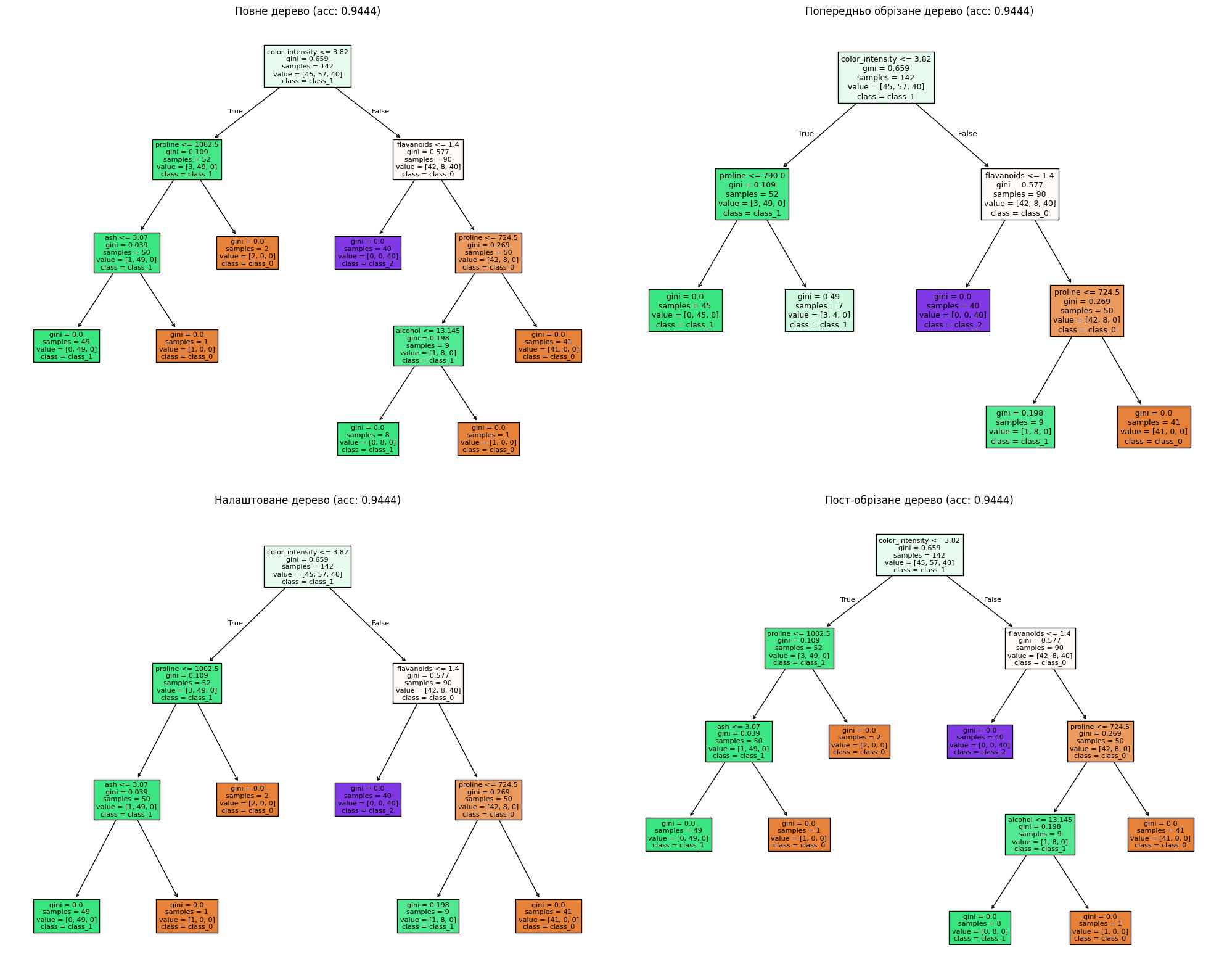
axs[1, 0].set\_title(f"Налаштоване дерево (acc: {gridsearch\_acc:.4f})")

plot\_tree(post\_pruned\_tree, filled=True, feature\_names=wine.feature\_names, class\_names=wine.target\_names, ax=axs[1, 1])

axs[1, 1].set\_title(f"Пост-обрізане дерево (acc: {post\_pruned\_acc:.4f})")

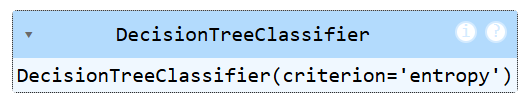
plt.tight\_layout()

plt.show()



wine\_tree = DecisionTreeClassifier(criterion='entropy')

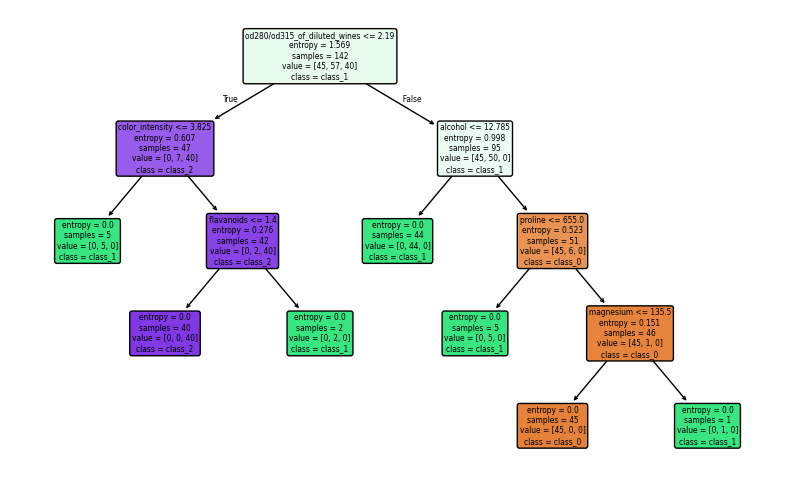
wine\_tree.fit(X\_train, y\_train)



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

plot\_tree(wine\_tree, feature\_names=wine.feature\_names, class\_names=wine.target\_names, filled=True, rounded=True)

plt.show()



accuracy\_iris\_tree = accuracy\_score(y\_test, wine\_tree.predict(X\_test))

print('Decision tree accuracy',accuracy\_iris\_tree)

Decision tree accuracy 0.9166666666666666

**Висновок:** лабораторній роботі було вивчено алгоритмів побудови дерев класифікації.