Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
Кафедра програмної інженерії

ЗВІТ

Лабораторної роботи № 2

з дисципліни «Інтелектуальний аналіз даних»

на тему «Кластеризація та дерева рішень»

Виконав

студент групи ІПЗм-24-2 Голодніков Дмитро

Перевірив

ст. викл. Онищенко К.Г.

Харків 2024

1 МЕТА РОБОТИ

Ознайомитися з методами кластеризації даних та класифікації за допомогою дерев рішень. Вивчити алгоритми K-Means, ієрархічної кластеризації, DBSCAN. Навчитися будувати та інтерпретувати дерева рішень для задач класифікації.

2 ХІД ВИКОНАННЯ РОБОТИ

2.1 Визначення оптимальної кількості кластерів

Для визначення оптимальної кількості кластерів використано метод ліктя (Elbow method) та коефіцієнт силуету (Silhouette score). Метод ліктя базується на аналізі інерції (сума квадратів відстаней до центроїдів), а силует оцінює якість кластеризації.

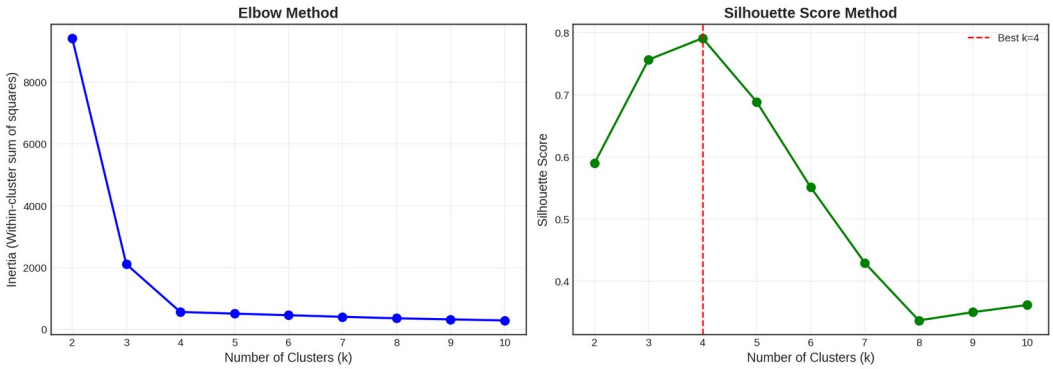

Рис. 2.1 - Метод ліктя та коефіцієнт силуету

2.2 Кластеризація K-Means

Алгоритм K-Means ітеративно призначає точки до найближчих центроїдів та оновлює положення центроїдів. Реалізація кластеризації K-Means:

```
from sklearn.cluster import KMeans

# Побудова моделі K-Means з k=4 кластерами
kmeans = KMeans(n_clusters=4, random_state=42, n_init=10)
labels = kmeans.fit_predict(X)

# Отримання центроїдів кластерів
centroids = kmeans.cluster_centers_

# Обчислення інерції (сума квадратів відстаней)
inertia = kmeans.inertia_
```

Результати кластеризації для k=4 показані на графіку.

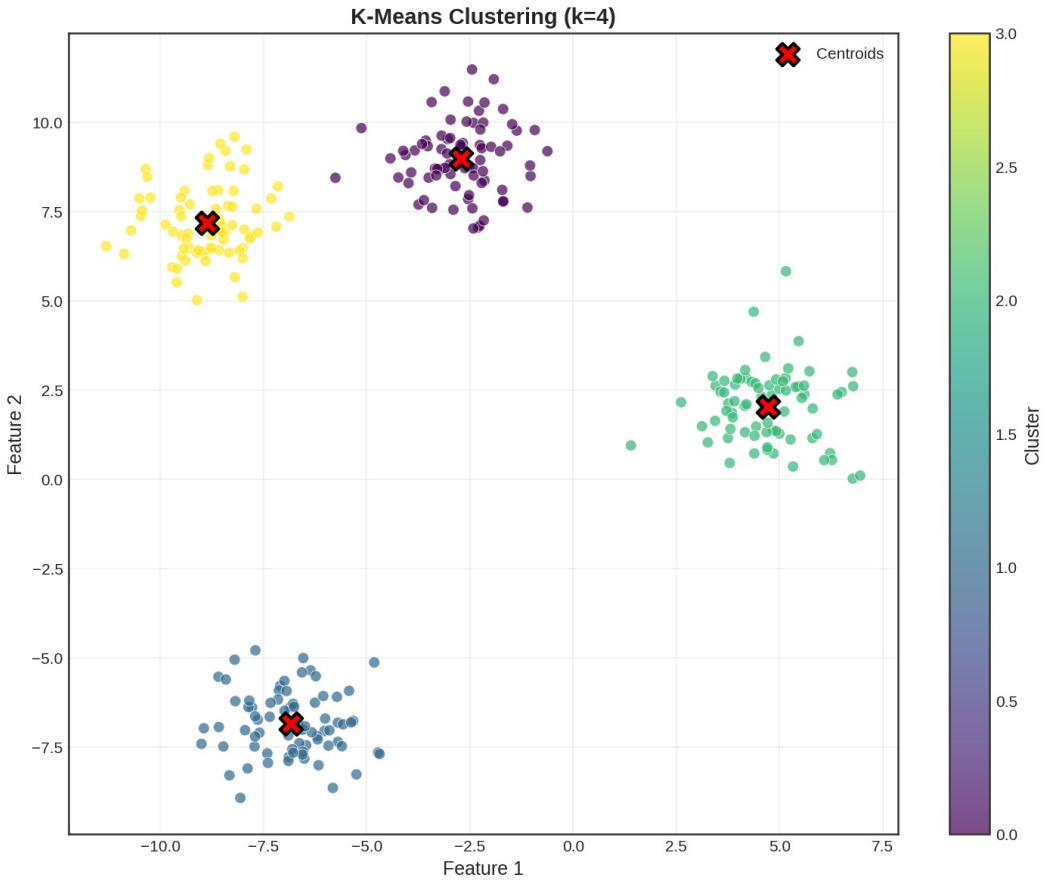

Рис. 2.2 - Результати кластеризації K-Means (k=4)

2.3 Ієрархічна кластеризація

Ієрархічна кластеризація будує дерево (дендрограму) об'єднання або поділу кластерів. Використано метод Уорда (Ward linkage), який мінімізує дисперсію всередині кластерів. Реалізація ієрархічної кластеризації:

```
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster

# Побудова матриці зв'язків методом Уорда
linkage_matrix = linkage(X, method='ward')

# Створення дендрограми
dendrogram(linkage_matrix, truncate_mode='level', p=5)

# Отримання міток кластерів (4 кластери)
labels = fcluster(linkage_matrix, t=4, criterion='maxclust')
```

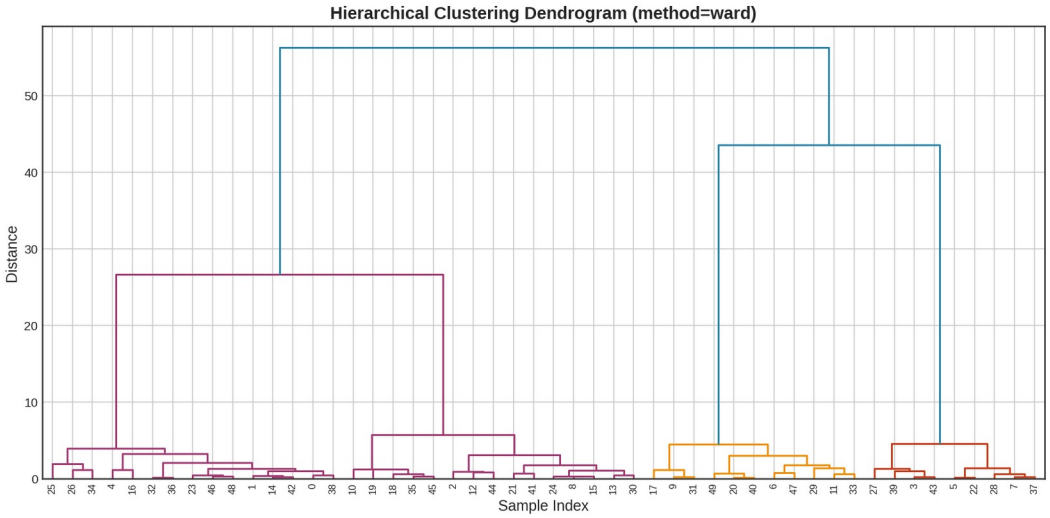

Рис. 2.3 - Дендрограма ієрархічної кластеризації

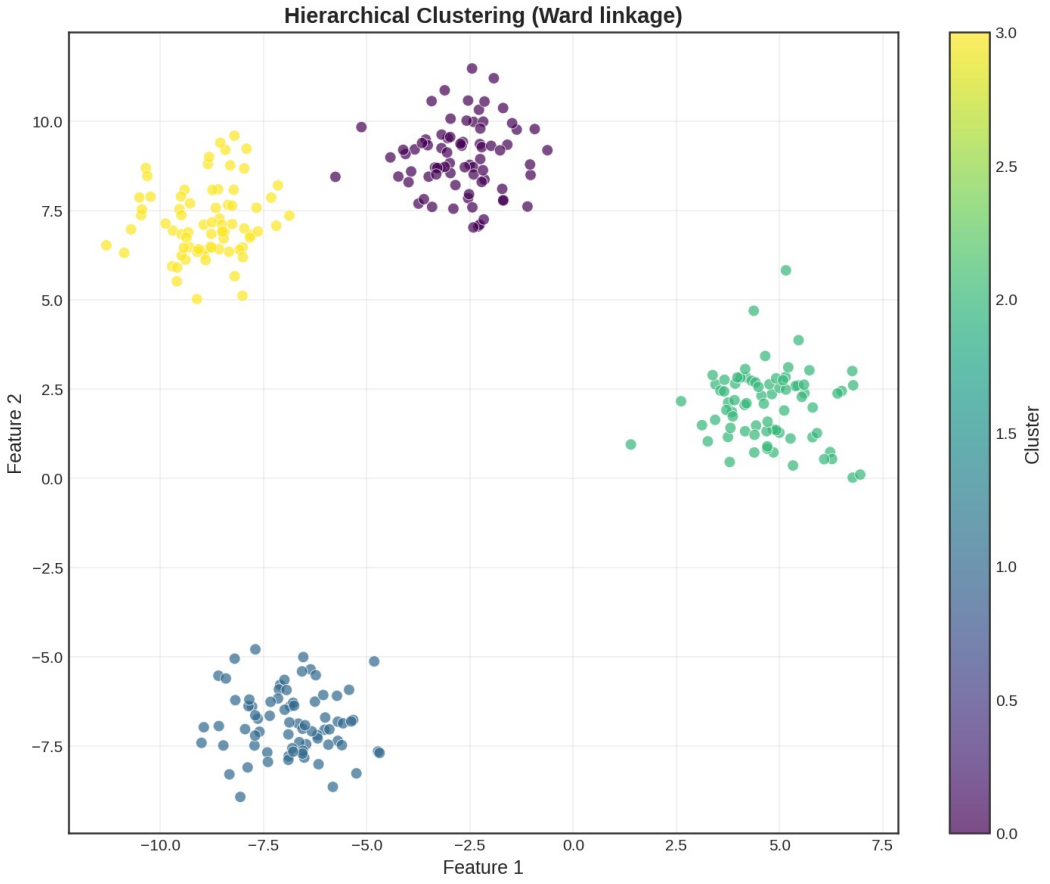

Рис. 2.4 - Результати ієрархічної кластеризації

2.4 Кластеризація DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) - алгоритм кластеризації на основі щільності. Він автоматично визначає кількість кластерів та ідентифікує шумові точки (викиди). Реалізація DBSCAN:

```
from sklearn.cluster import DBSCAN

# Побудова моделі DBSCAN
# eps - радіус околу, min_samples - мінімум точок для core point
dbscan = DBSCAN(eps=0.5, min_samples=5)
labels = dbscan.fit_predict(X)

# Визначення кількості кластерів (без урахування шуму)
n_clusters = len(set(labels)) - (1 if -1 in labels else 0)
n_noise = list(labels).count(-1)  # Шумові точки
```

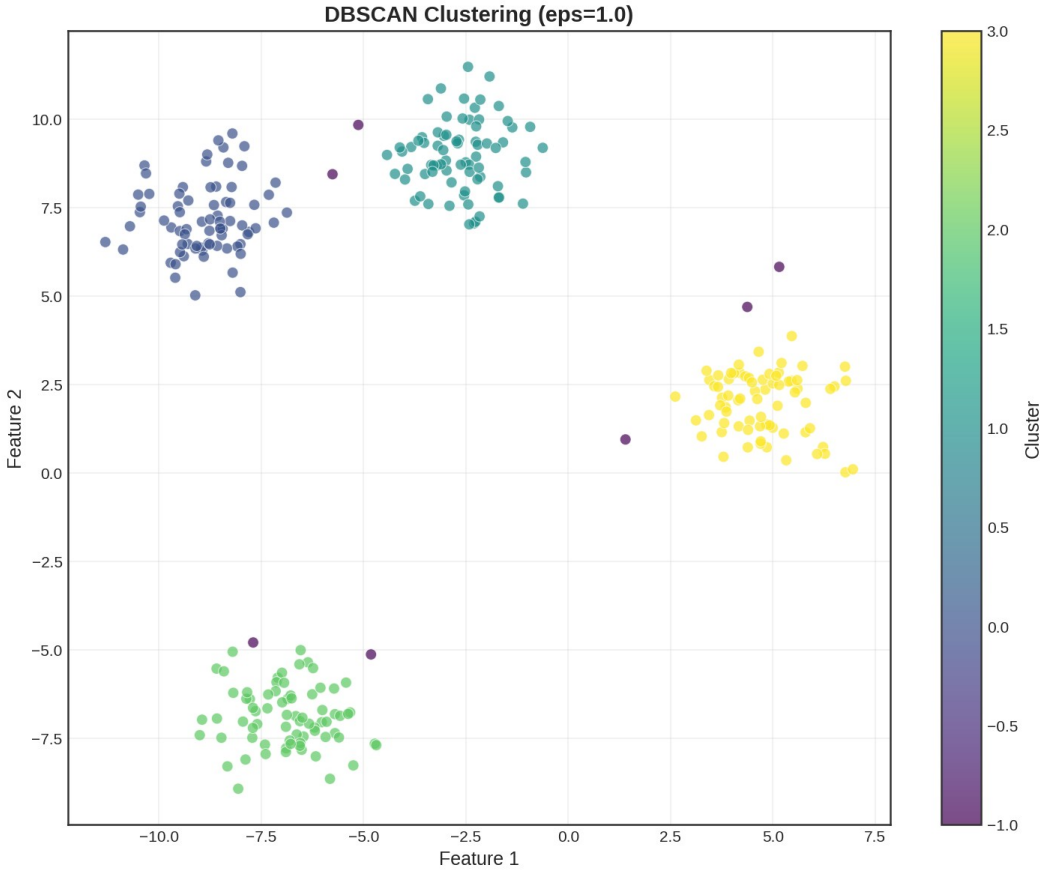

Рис. 2.5 - Результати кластеризації DBSCAN

2.5 Дерево рішень

Дерево рішень - модель класифікації, що розбиває простір ознак на регіони за допомогою послідовних бінарних розбиттів. Реалізація дерева рішень:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split

# Поділ даних на навчальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)

# Побудова дерева рішень
tree = DecisionTreeClassifier(max_depth=4, random_state=42)
tree.fit(X_train, y_train)

# Прогнозування та оцінка точності
y_pred = tree.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
```

Побудовано дерево для класифікації набору даних Iris.

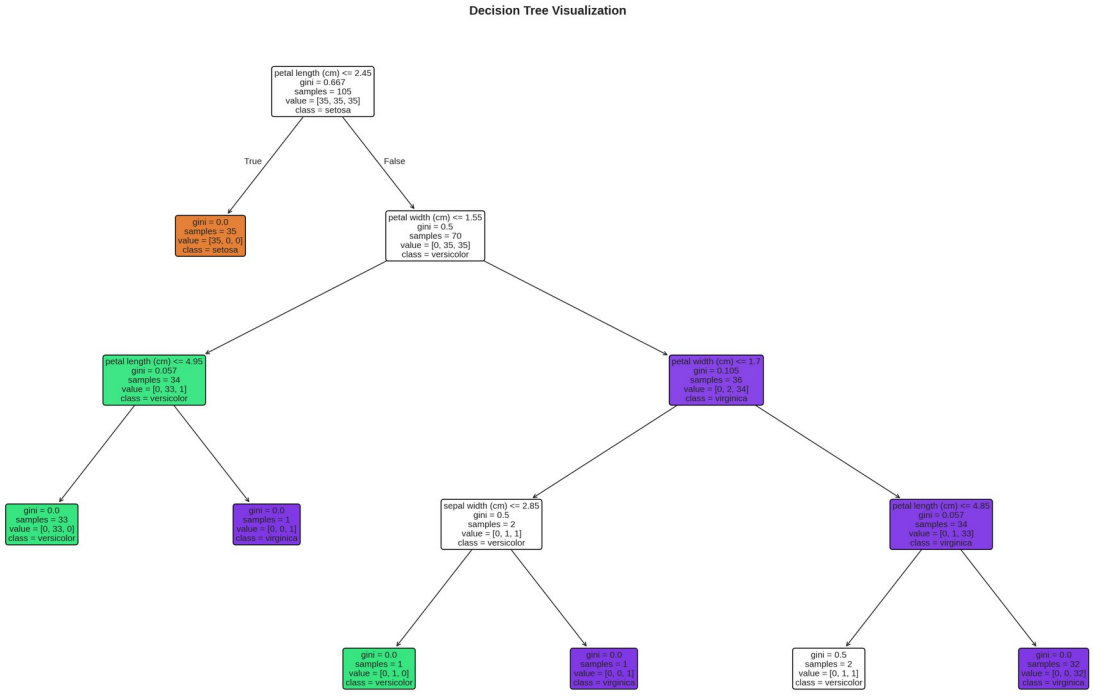

Рис. 2.6 - Структура дерева рішень

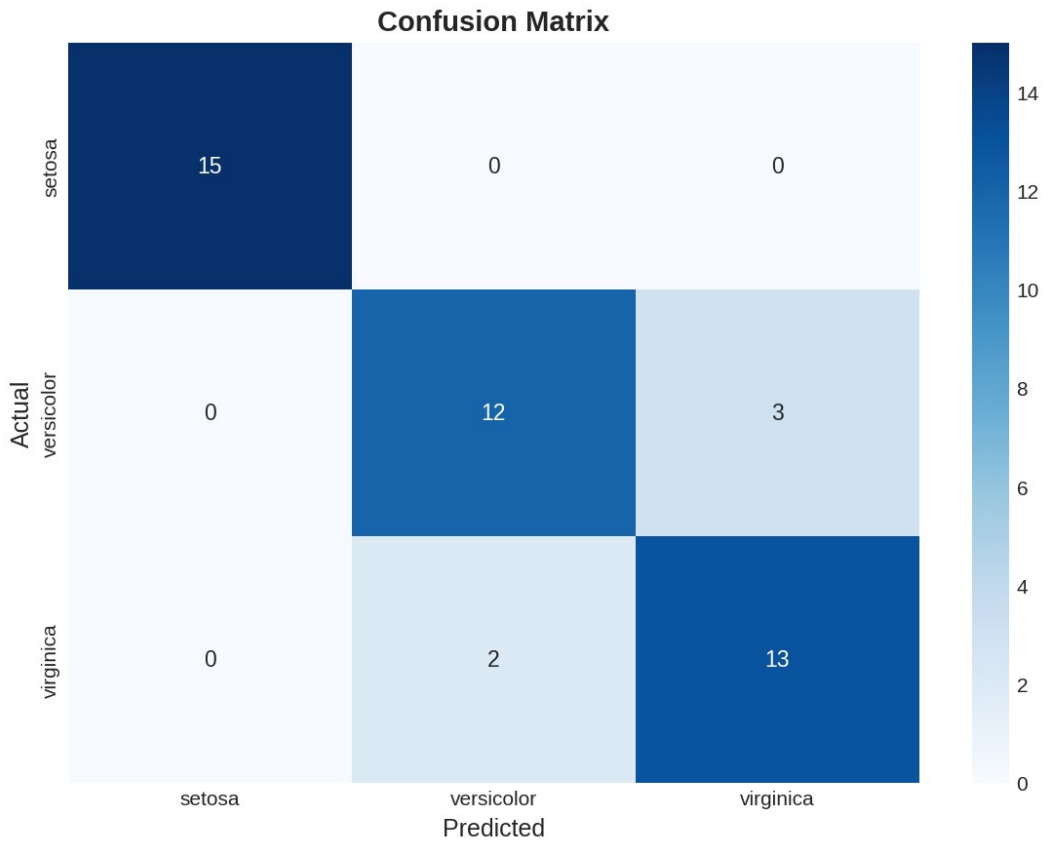

Рис. 2.7 - Матриця помилок класифікації

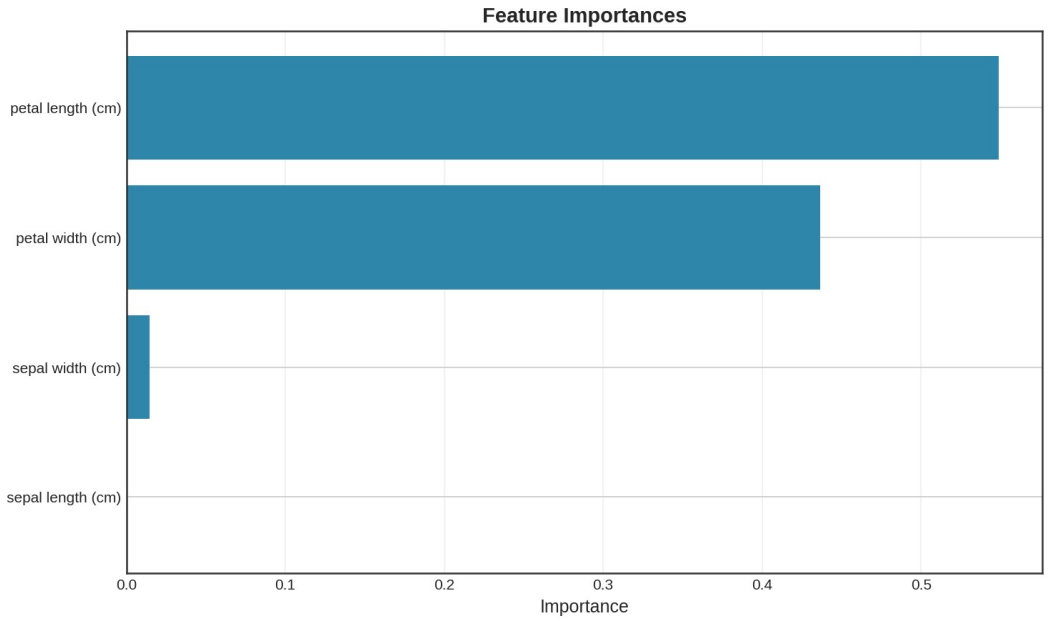

Рис. 2.8 - Важливість ознак у дереві рішень

3 ВИСНОВКИ

У ході виконання лабораторної роботи було досліджено методи кластеризації (K-Means, ієрархічна, DBSCAN) та класифікації (дерева рішень). Реалізовано алгоритми визначення оптимальної кількості кластерів. Проведено порівняльний аналіз різних методів кластеризації. Побудовано та візуалізовано дерево рішень для задачі класифікації. Оцінено якість класифікації за допомогою матриці помилок та метрик accuracy, precision, recall.

ПОСИЛАННЯ

Код проєкту доступний у репозиторії GitHub: https://github.com/na-naina/data-analysis-khnure