Міністерство освіти і науки України
НТУУ «КПІ ім. Ігоря Сікорського»
Навчально-науковий інститут атомної та теплової енергетики
Кафедра цифрових технологій в енергетиці

Лабораторна робота №5

з дисципліни «Вступ до інтелектуального аналізу даних»

Тема «Ієрархічна кластеризація та дендрограми в Python»

Варіант №19

Студента 3-го курсу НН ІАТЕ гр. ТР-12
Ковальова Олександра
Перевірив: д.т.н., проф. Путренко В. В.

КИЇВ 2024

Мета: Опрацювати приклад роботи з кластеризацією та дендрограмами, використовуючи Jupyter Notebook. Виконати поставлене завдання.

Хід роботи

Для початку підключаємо потрібні бібліотеки:

```
In [1]: import matplotlib.pyplot as plt
        import numpy as np
        from scipy.cluster.hierarchy import linkage, dendrogram
        import sqlite3

        db_filename = 'data/cereals-1.db'
        conn = sqlite3.connect(db_filename)
        c = conn.cursor()
```

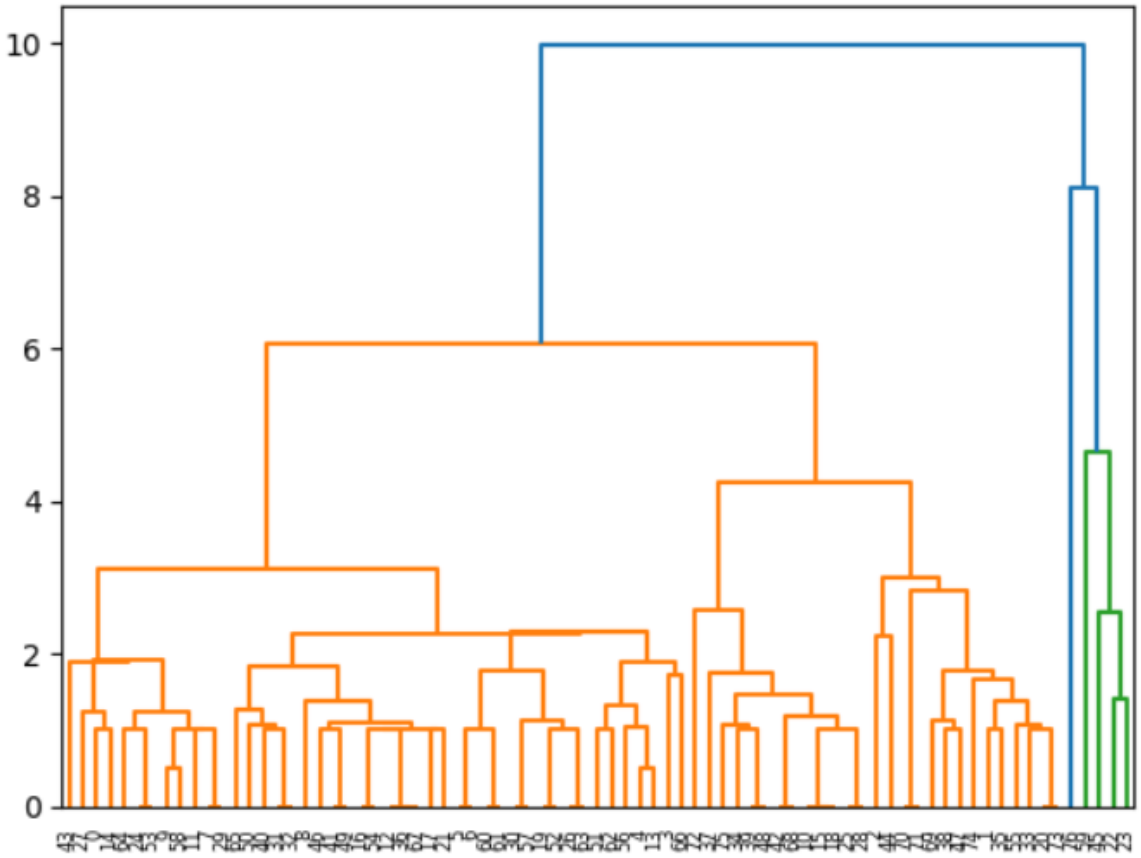
Ці рядки імпортують необхідні бібліотеки. Бібліотека matplotlib потрібна для відображення графіка в середовищі Jupyter Notebook, numpy використовується для обробки даних. Бібліотека scipy побудована на numpy та надає функції для створення та відображення дендрограми. Нарешті, sqlite3 надає інтерфейс до бази даних. В останніх трьох рядках відбувається підключення до бази даних та створення об'єкту курсору.

Створення дендрограми:

```
In [5]: c.execute('''SELECT Protein,
                            Fat,
                            Carbohydrates
                     FROM cereals''')
        results = c.fetchall()

        X = (np.array(results)
             .reshape(-1,3))

        links = linkage(X, 'centroid')

        dendrogram(links)
        plt.show()
```

Ці три рядки за збір даних із бази даних і упаковку їх у відповідний формат для scipy. Перший рядок виконує запит і збирає вміст білка, жиру та вуглеводів для кожної крупи. Другий запит перетворює отримані дані з формату кортежу, повернутого sqlite3, на матрицю numpy, а потім транспонує матрицю в правильний формат. Четвертий запит відповідає за обчислення як кластеризації, яка інформує дендрограму, так і зв'язків дендрограми. Функція приймає дані, X і специфікатор метрики відстані для використання під час обчислення відстаней кластера. У цьому випадку ми використовуємо «центроїд», який обчислює центроїд кожного кластера, а потім відстань між ними під час об'єднання кластерів. І нарешті, останні два рядки створюють візуалізацію дендрограми, а потім показують графік у Блокноті.

Результат:

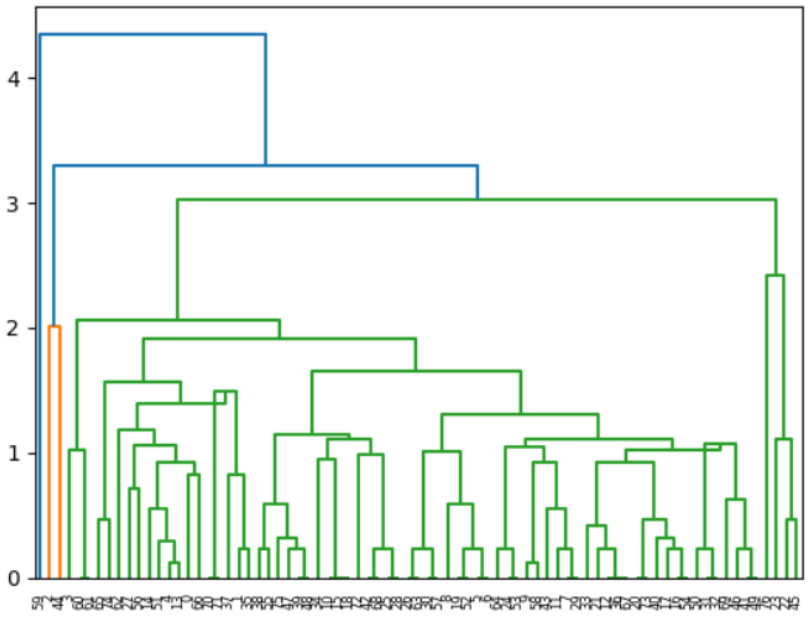


Раніше згадувалося, наскільки важливо нормалізувати дані під час кластеризації. Важливо зазначити, що ми не нормалізували дані в попередньому прикладі. Далі ми нормалізуємо ті самі дані, а потім знову намалюємо дендрограму:

```
In [4]: from scipy.cluster.vq import whiten

        Y = whiten(X)

        links = linkage(Y, 'centroid')

        dendrogram(links)
        plt.show()
```

Використовуючи здатність Jupyter Notebook підтримувати стан коду між клітинками, ми можемо продовжити безпосередньо з того місця, де зупинилися в попередній клітинці. Щоб нормалізувати дані, ми імпортуємо функцію «whiten» із scipy, а потім застосовуємо її до транспонованої матриці даних. Це нормалізує значення ознак до діапазону [0, 1]. Нарешті, ми повторно обчислюємо дендрограму,

використовуючи нормалізовані значення ознак. Знову ж таки, ці останні рядки відтворюють і показують дендрограму, попередньо обчислену функцією зв'язку:



Висновок: Під час виконання лабораторної роботи були набуті практичні навички роботи з дендрограмами та кластеризацією. Було побудовано дві дендрограми.

Програмний код

Notebook.ipynb:

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.cluster.hierarchy import linkage, dendrogram
import sqlite3

db_filename = 'data/cereals-1.db'
conn = sqlite3.connect(db_filename)
c = conn.cursor()

c.execute('''SELECT Protein,
                    Fat,
                    Carbohydrates
             FROM cereals''')
results = c.fetchall()

X = (np.array(results)
     .reshape(-1,3))

links = linkage(X, 'centroid')

dendrogram(links)
plt.show()

from scipy.cluster.vq import whiten

Y = whiten(X)

links = linkage(Y, 'centroid')

dendrogram(links)
plt.show()
```