Національний технічний університет України «КПІ»

Кафедра автоматизованих систем обробки інформації і управління

Лабораторна робота №5

з дисципліни «[Аналіз даних в інформаційних управляючих системах](https://do.ipo.kpi.ua/course/view.php?id=5470)»

на тему: «ІЄРАРХІЧНІ МЕТОДИ КЛАСТЕРНОГО АНАЛІЗУ»

Виконав:

студент групи ІС-23

Шимків М.В.

Викладач:

Гавриленко О.В

Київ 2024

**Мета роботи**: вивчити основні методи побудови дерев вирішальних правил,

навчитися використовувати спеціалізовані програмні засоби для побудови дерев вирішальних правил.

**Завдання до роботи**

1. Ознайомитися з конспектом лекцій та рекомендованою літературою, а також

додатком Д, що містить короткі теоретичні відомості про ієрархічні методи кластерного

аналізу та особливості їх застосування.

2. За допомогою команди >>help вивчити функції pdist, squareform, linkage,

dendrogram, cophenet, cluster, gscatter.

3. Завантажити висхідний набір даних для обробки та аналізу згідно до вашого

варіанту (табл. 2). Побудувати графічне зображення експериментальних даних.

4. Обчислити відстань між об'єктами. Використовувати міри для розрахунку

відстаней відповідно до варіанту (табл. 2).

5. Використовуючи рекомендоване програмне забезпечення здійснити кластерний

аналіз висхідних даних методом ієрархічної кластеризації за методами зв'язування

відповідно до варіанту (табл. 2).

6. Виконати аналіз якості кластеризації за допомогою обчислення кофенетичного

кореляційного коефіцієнту. Заповнити таблицю для кофенетичного кореляційного

коефіцієнту (табл. 1).

Таблиця 1

7. Визначити найбільш і найменш ефективні способи ієрархічної кластеризації для

аналізу висхідного набору даних (максимальні і мінімальні коефіцієнти та відповідні до

них способи кластеризації). Для найбільш ефективного способу ієрархічної кластериації

побудувати дендрограму результатів кластерного аналізу.

8. Визначити кількість вірогідних кластерів. Для виділення значущих кластерів

використовувати порогове значення, розраховане за метрикою відстаней або методом

завдання фіксованого числа кластерів.

9. Розрахувати центри та внутрішньокластерну дисперсію отриманих кластерів,

геометричні відстані від елементів до центрів кластерів, відстані між центрами

кластерів. Відобразити графічно знайдені кластери та ії центри (скористатися діаграмою

розсіювання у кольорі).

10. Оформити звіт з роботи.

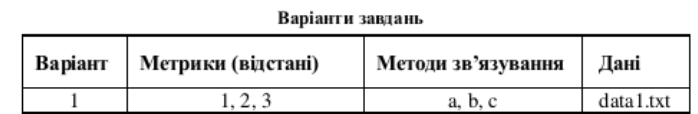
11. Відповісти на контрольні питання.

**Набір даних:**

Набір даних я взяв з сервісу kaggle. Ось посилання на набір даних:

<https://www.kaggle.com/datasets/varunraskar/store-sales-data>

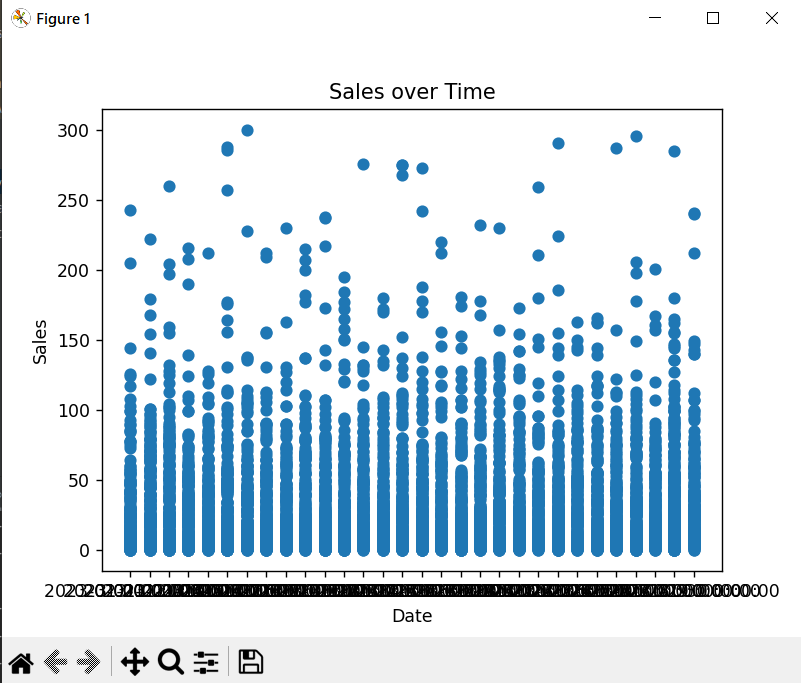
**Варіант 25**

****

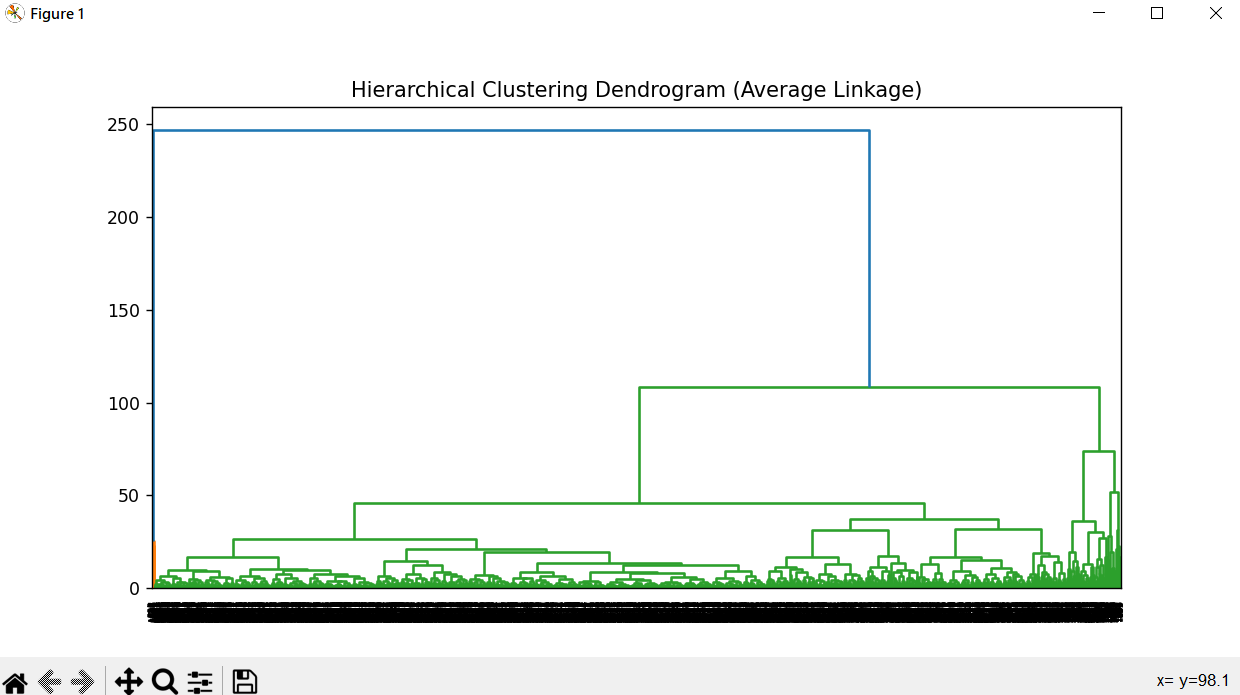
Запуск програми відбувається так:



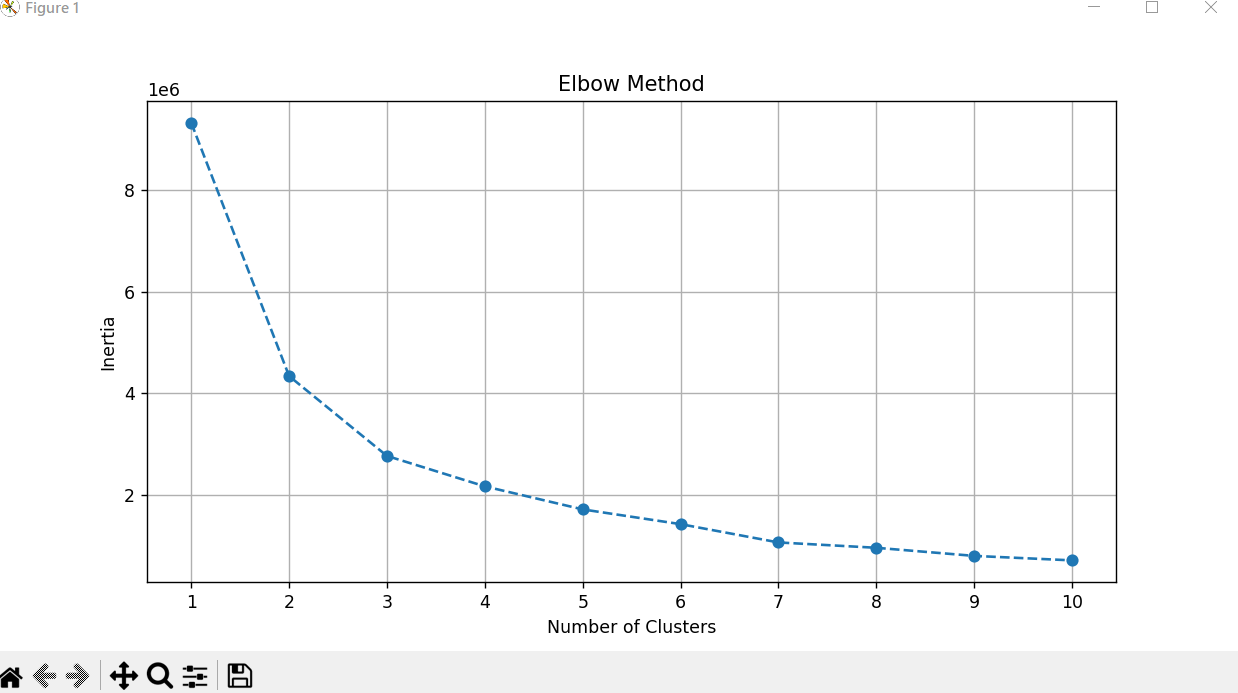
Графічне зображення експериментальних даних



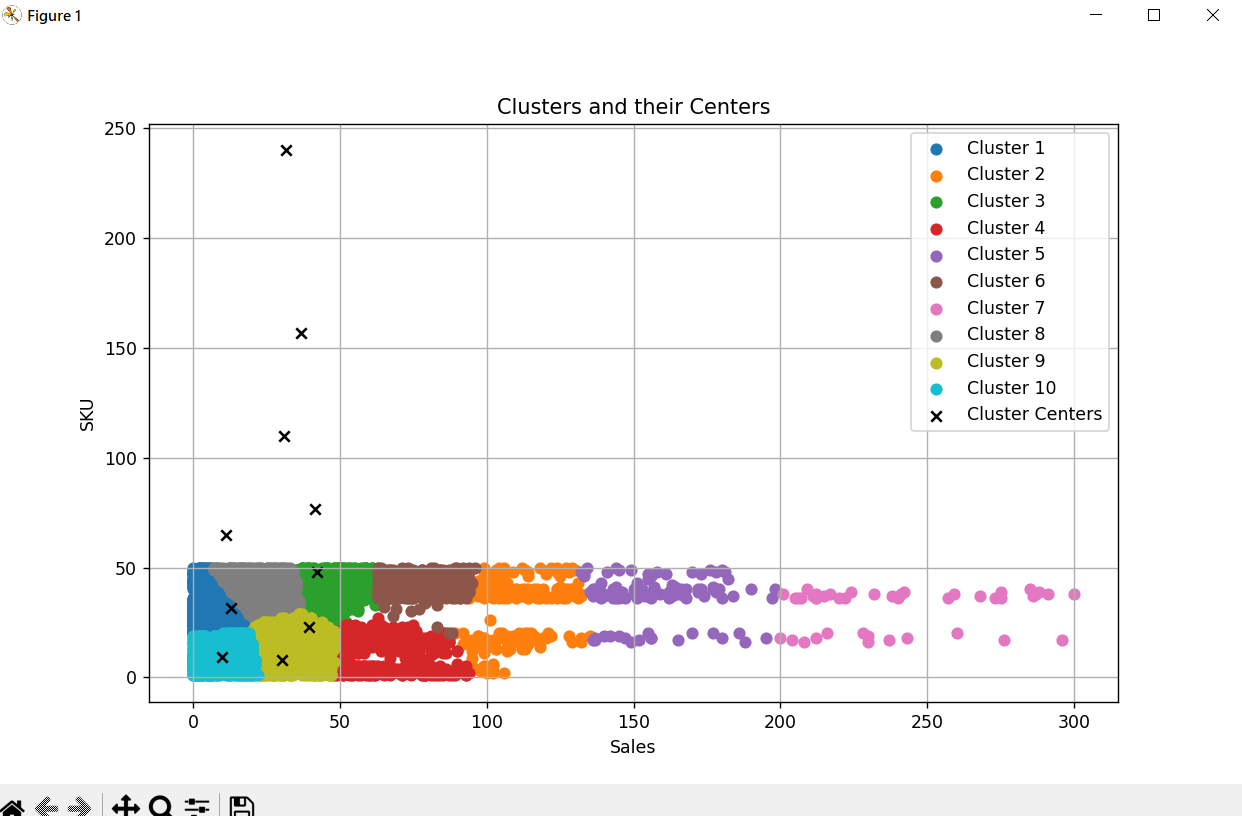
Для найбільш ефективного способу ієрархічної кластериації побудувана дендрограма результатів кластерного аналізу(Середнього звязку)

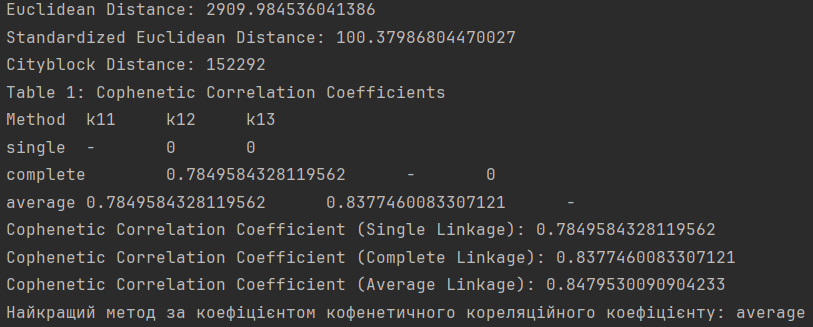


Побудова моделі KMeans з різними кількостями кластерів



Знайдені кластери та ії центри відображені графічно





Демонстрація обчислення(відстані, зв’язування):

1 – Евклідова, 2 – стандартизована Евклідова, 3 – міста

a – ближчого сусіда, b – дальшого сусіда, c – середнього зв'язку

Найкращий метод за коефіцієнтом кофенетичного кореляційного коефіцієнту: середнього зв'язку

А також таблиця для кофенетичного кореляційного коефіцієнту

**Код програми:**

import pandas as pd  
import numpy as np  
  
# Завантаження даних з CSV файлу  
data = pd.read\_csv("Sales.csv")  
  
# Check for NaN values in 'Sales' and 'SKU' columns  
if data['Sales'].isnull().values.any() or data['SKU'].isnull().values.any():  
 print("Warning: 'Sales' column contains NaN values. Proceeding with NaN values.")  
  
# Fill NaN values with a default value (e.g., 0)  
data.fillna(0, inplace=True)  
  
# Print the converted DataFrame  
print(data)  
  
import matplotlib.pyplot as plt  
  
# Побудова графіка  
plt.scatter(data['Date'], data['Sales'])  
plt.xlabel('Date')  
plt.ylabel('Sales')  
plt.title('Sales over Time')  
plt.show()  
  
  
# Обчислення відстаней  
from scipy.spatial.distance import euclidean, cityblock  
def standardized\_euclidean(x, y):  
 # Стандартизуємо вектори  
 x\_std = (x - np.mean(x)) / np.std(x)  
 y\_std = (y - np.mean(y)) / np.std(y)  
  
 # Обчислюємо євклідову відстань між стандартизованими векторами  
 distance = np.sqrt(np.sum((x\_std - y\_std)\*\*2))  
  
 return distance  
  
# Обчислення відстаней  
euclidean\_distance = euclidean(data['Sales'], data['SKU'])  
standardized\_euclidean\_distance = standardized\_euclidean(data['Sales'], data['SKU'])  
cityblock\_distance = cityblock(data['Sales'], data['SKU'])  
  
print("Euclidean Distance:", euclidean\_distance)  
print("Standardized Euclidean Distance:", standardized\_euclidean\_distance)  
print("Cityblock Distance:", cityblock\_distance)  
  
from scipy.cluster.hierarchy import linkage, dendrogram  
data\_numeric = data.drop(columns=['Date', 'Product Name', 'City'])  
# Зв'язування кластерів  
single\_linkage = linkage(data\_numeric, method='single')  
complete\_linkage = linkage(data\_numeric, method='complete')  
average\_linkage = linkage(data\_numeric, method='average')  
  
from scipy.spatial.distance import pdist  
from scipy.cluster.hierarchy import linkage, cophenet  
  
# Compute pairwise distances  
pairwise\_distances = pdist(data\_numeric)  
  
# Perform hierarchical clustering with different linkage methods  
linkage\_methods = ['single', 'complete', 'average']  
cophenet\_coefficients = {}  
  
for i, method in enumerate(linkage\_methods):  
 linkage\_matrix = linkage(data\_numeric, method=method)  
 cophenet\_coefficients[method] = cophenet(linkage\_matrix, pairwise\_distances)[0]  
  
# Print the table  
print("Table 1: Cophenetic Correlation Coefficients")  
print("Method\tk11\tk12\tk13")  
  
for i, method1 in enumerate(linkage\_methods):  
 row = [method1]  
 for j, method2 in enumerate(linkage\_methods):  
 if j < i:  
 row.append(cophenet\_coefficients[method2])  
 elif j == i:  
 row.append("-")  
 else:  
 row.append(0) # Placeholder for remaining values  
 print("\t".join(map(str, row)))  
  
# Compute pairwise distances  
pairwise\_distances = pdist(data\_numeric)  
  
# Compute the cophenetic correlation coefficient for each linkage method  
c\_single = cophenet(single\_linkage, pairwise\_distances)[0]  
c\_complete = cophenet(complete\_linkage, pairwise\_distances)[0]  
c\_average = cophenet(average\_linkage, pairwise\_distances)[0]  
  
print("Cophenetic Correlation Coefficient (Single Linkage):", c\_single)  
print("Cophenetic Correlation Coefficient (Complete Linkage):", c\_complete)  
print("Cophenetic Correlation Coefficient (Average Linkage):", c\_average)  
##################################################  
# Визначення методу з найвищим значенням коефіцієнту  
best\_method = max([(c\_single, 'single'), (c\_complete, 'complete'), (c\_average, 'average')])[1]  
  
# Виведення найкращого коефіцієнту  
print("Найкращий метод за коефіцієнтом кофенетичного кореляційного коефіцієнту:", best\_method)  
  
# Побудова дендрограми для найбільш ефективного методу  
plt.figure(figsize=(10, 5))  
plt.title(f'Hierarchical Clustering Dendrogram ({best\_method.capitalize()} Linkage)')  
dendrogram(locals()[f"{best\_method}\_linkage"])  
plt.show()  
##############  
  
from sklearn.cluster import KMeans  
import matplotlib.pyplot as plt  
  
# Побудова моделі KMeans з різними кількостями кластерів  
inertia\_values = []  
max\_clusters = 10 # Максимальна кількість кластерів  
  
for i in range(1, max\_clusters + 1):  
 kmeans = KMeans(n\_clusters=i, random\_state=42)  
 kmeans.fit(data\_numeric)  
 inertia\_values.append(kmeans.inertia\_)  
  
# Побудова графіка "ліктя"  
plt.figure(figsize=(10, 5))  
plt.plot(range(1, max\_clusters + 1), inertia\_values, marker='o', linestyle='--')  
plt.title('Elbow Method')  
plt.xlabel('Number of Clusters')  
plt.ylabel('Inertia')  
plt.xticks(range(1, max\_clusters + 1))  
plt.grid(True)  
plt.show()  
#########################  
import matplotlib.pyplot as plt  
  
# Розрахунок центрів кластерів  
cluster\_centers = kmeans.cluster\_centers\_  
  
# Відображення графічно знайдених кластерів та їх центрів  
plt.figure(figsize=(10, 6))  
  
# Відобразити кожен кластер окремим кольором  
for cluster\_index in range(kmeans.n\_clusters):  
 cluster\_points = data.iloc[kmeans.labels\_ == cluster\_index] # Використовуйте метод .iloc для отримання рядків за певним умовам  
 plt.scatter(cluster\_points['Sales'], cluster\_points['SKU'], label=f'Cluster {cluster\_index+1}')  
  
# Відобразити центри кластерів  
plt.scatter(cluster\_centers[:, 0], cluster\_centers[:, 1], color='black', marker='x', label='Cluster Centers')  
  
plt.title('Clusters and their Centers')  
plt.xlabel('Sales')  
plt.ylabel('SKU')  
plt.legend()  
plt.grid(True)  
plt.show()

**Контрольні питання:**

1. **В чому полягає задача кластерного аналізу?** Кластерний аналіз - це процес групування схожих об'єктів разом у кластери таким чином, щоб об'єкти в одному кластері були більш схожі один на одного, ніж на об'єкти в інших кластерах. Мета кластерного аналізу - виявити приховані структури в наборі даних, згрупувати схожі об'єкти та визначити їх внутрішні зв'язки.
2. **Для яких задач обробки експериментальних даних використовуються методи ієрархічного кластерного аналізу?** Методи ієрархічного кластерного аналізу використовуються для обробки експериментальних даних для розгляду структури даних, виявлення груп подібних об'єктів та знаходження закономірностей.
3. **Наведіть основні міри порівняння об'єктів між собою.** Основні міри порівняння об'єктів між собою включають відстані (наприклад, євклідову відстань, Манхеттенську відстань, косинусну відстань), кореляційні коефіцієнти (наприклад, Пірсона, Спірмена) та інші.
4. **Що таке дендрограма?** Дендрограма - це графічне представлення ієрархічної структури кластерного аналізу. Вона показує, як об'єкти або групи об'єктів згруповані разом на різних рівнях схожості.
5. **Що являють собою ієрархічні агломеративні методи кластерного аналізу?** Ієрархічні агломеративні методи кластерного аналізу - це методи, які починають з кожного об'єкта як окремого кластера, а потім об'єднують найбільш схожі кластери на кожному кроці, поки не отримаються один або кілька кластерів.
6. **Що являють собою ієрархічні дивізимні методи кластерного аналізу?** Ієрархічні дивізимні методи кластерного аналізу - це методи, які спочатку розглядають всі об'єкти як один кластер, а потім рекурсивно розбивають цей кластер на менші кластери.
7. **Наведіть основні способи зв'язування об'єктів у кластери.** Основні способи зв'язування об'єктів у кластери включають одноелементний (одиничний) метод, повний метод, метод середнього зв'язування та метод Ворда.
8. **Що таке кофенетичний кореляційний коефіцієнт?** Кофенетичний кореляційний коефіцієнт - це міра, яка визначає ступінь схожості між парою об'єктів на основі їх відстаней у вихідному просторі і в просторі кластерів.
9. **В чому полягають основні етапи ієрархічного кластерного аналізу?** Основні етапи ієрархічного кластерного аналізу включають створення матриці відстаней, обчислення матриці подібності або відстаней, вибір методу з'єднання, обчислення дендрограми та інтерпретацію результатів.
10. **Яким чином визначити значущу кількість кластерів?** Значуща кількість кластерів може бути визначена за допомогою різних методів, таких як метод ліктя, метод січення або швидкість зміни критерію.
11. **З якою метою використовують функції MATLAB: pdist, square- form, linkage, dendrogram, cophenet, cluster, gscatter?**
    * pdist: обчислює парні відстані між об'єктами.
    * squareform: перетворює вектор відстаней у квадратну матрицю відстаней.
    * linkage: об'єднує кластери на основі заданого методу з'єднання.
    * dendrogram: побудовує дендрограму ієрархічного кластерного аналізу.
    * cophenet: обчислює кофенетичний кореляційний коефіцієнт.
    * cluster: виконує кластерний аналіз.
    * gscatter: побудовує графік для відображення кластерів на площині.

**Висновок:**

Отже, у ході виконання лабораторної роботи було проведено аналіз даних за допомогою ієрархічних методів кластерного аналізу. Згідно з завданням, було розглянуто різні метрики відстаней та методи зв'язування, а також визначено найбільш ефективний спосіб кластеризації за допомогою обчислення кофенетичного кореляційного коефіцієнту.

На основі результатів обчислень було встановлено, що найкращим методом кластеризації є метод зв'язування "average", оскільки він показав найвищий коефіцієнт кофенетичної кореляції серед розглянутих методів.

Таким чином, виконання лабораторної роботи дозволило не лише ознайомитися з ієрархічними методами кластерного аналізу, а й провести практичне їх застосування для аналізу експериментальних даних.