МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет комп’ютерних наук

Кафедра програмної інженерії

ЗВІТ

з контрольної роботи

з дисципліни: «ЧИСЕЛЬНІ МЕТОДИ ТА ВИСОКОПРОДУКТИВНІ ОБЧИСЛЕННЯ»

на тему: «Методи ієрархічного кластерного аналізу даних»

Варіант №11

Виконала: ст. гр. ІПЗм-19-2

Михневич Тетяна

Перевірила:

проф. Білоус Н. В.

Харків 2020

**Мета роботи**

Практичне освоєння методів ієрархічного кластерного аналізу даних.

**Індивідуальне завдання**

|  |  |  |
| --- | --- | --- |
| **Варіант** | **Метрики (відстані)** | **Методи зв’язування** |
| 11 | 2, 4, 6 | a, b, c |

\*Метрики відстані: 1 - евклідів, 2 - стандартизована евклідова, 3 - міста, 4 - Махаланобіса, 5 - Маньківського (p = 4), 6 -Чебишева.

\*\* Методи зв'язування: a - ближнього сусіда, b - далекого сусіда, c - середньої зв'язку, d - центроїдного, e - медіанної зв'язку.

Вихідні дані для кластеризації (варіант 11): X2, X3, X6, X7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Масло тварина, кг | Цукор, кг | Хлібопродукти, кг | Число лікарів на 10 000 населення |
| X2 | X3 | X6 | X7 |
| Росія | 3,9 | 30 | 124 | 44,5 |
| Австралія | 2,6 | 47 | 87 | 32, |
| Австрія | 5,3 | 37 | 74 | 33,9 |
| Азербайджан | 4,1 | 12,4 | 141 | 38,8 |
| Вірменія | 3,7 | 4,3 | 134 | 34,4 |
| Білорусь | 3,6 | 28 | 120 | 43,6 |
| Бельгія | 6,9 | 48 | 72 | 41 |
| Болгарія | 3 | 18 | 156 | 36,4 |
| Великобританія | 3,5 | 39 | 91 | 17,9 |
| Угорщина | 1,7 | 40 | 106 | 32,1 |
| Німеччина | 6,8 | 35 | 73 | 38,1 |
| Греція | 1 | 24 | 108 | 41,5 |
| Грузія | 3,8 | 36 | 140 | 55 |
| Данія | 5 | 38 | 77 | 36,7 |
| Ірландія | 3,3 | 31 | 102 | 15,8 |
| Іспанія | 0,4 | 26 | 72 | 40,9 |
| Італія | 2,2 | 27 | 118 | 49,4 |
| Казахстан | 4,2 | 19,2 | 191 | 38,1 |
| Канада | 3,1 | 44 | 77 | 27,6 |
| Киргизія | 4,1 | 23,5 | 134 | 33,2 |

**Хід роботи**

Ієрархічний кластерний аналіз даних буде проведено на мові програмування Python.

1. Створюємо 2 текстових файли (наприклад, в програмі Блокнот) у форматі .csv і зберігаємо в них матрицю вихідних даних відповідно до варіанту.

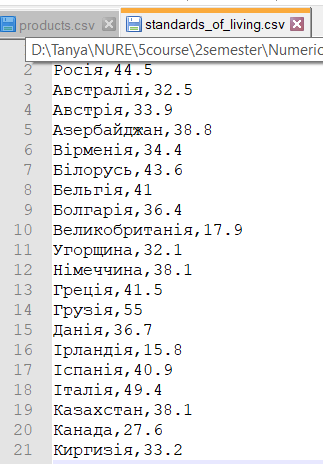
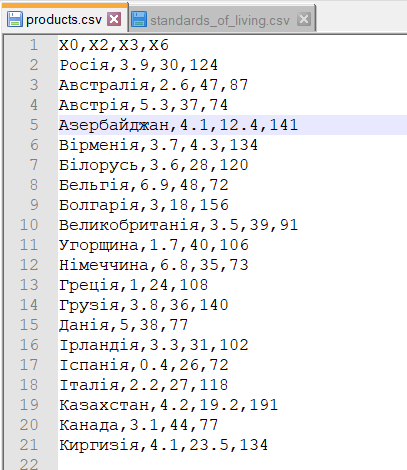


Рисунок 1 – Текстові файли із вихідними даними

Зчитуємо дані з файлів:

products = pd.read\_csv('products.csv')

standards\_of\_living = pd.read\_csv('standards\_of\_living.csv')

countries = pd.merge(products, standards\_of\_living)



Рисунок 2 – Фрагмент зчитаної із файлу таблиці з даними

2. Виконаємо стандартне нормування і побудуємо графічне зображення на площині деяких пар ознак експериментальних даних.

Нормалізуємо дані:

countries[features] = StandardScaler().fit\_transform(countries[features])

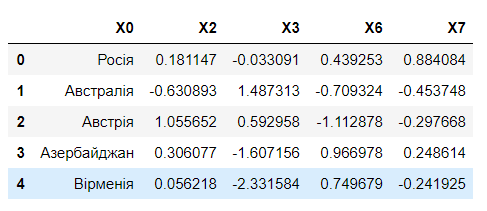


Рисунок 2 – Фрагмент зчитаної із файлу таблиці з нормалізованими даними

Візуалізуємо залежність між різними ознаками і розподіл кожної окремої ознаки:

sns.pairplot(countries[features], diag\_kind='hist')

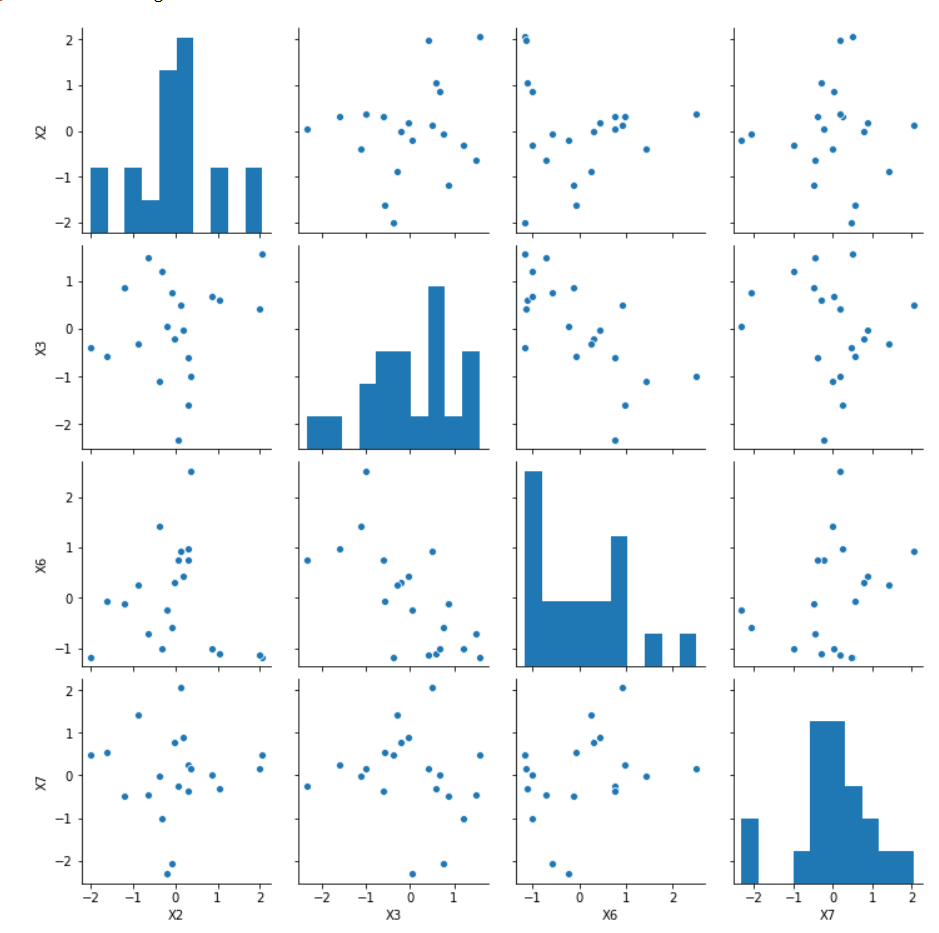


Рисунок 3 - Графічне зображення на площині пар ознак експериментальних даних

Для деталізації візуалізуємо залежність між двома першими ознаками з підписами країн:

fig, ax = plt.subplots()

xs = countries[features[0]]

ys = countries[features[1]]

ax.scatter(xs, ys)

for i, txt in enumerate(countries['X0']):

ax.annotate(txt, (xs[i],ys[i]))

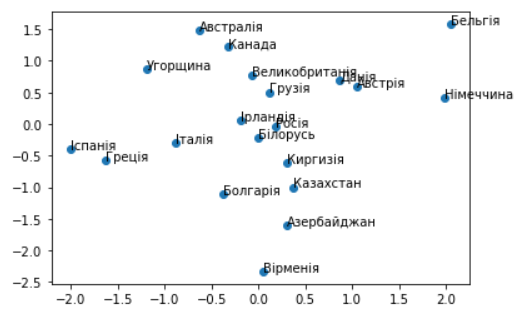


Рисунок 5 - Залежність між двома першими ознаками з підписами країн

3. Обчислимо відстані між об'єктами c допомогою функції pdist.

Виконаємо кластерний аналіз вихідних даних методом ієрархічної кластеризації.

Виконаємо аналіз якості кластеризації за допомогою обчислення кофенетичного кореляційного коефіцієнта (функція cophenet).

Для кожної метрики и методу зв'язування підрахуємо відстані між об'єктами за допомогою функції pdist, Проведемо Кластерний аналіз за допомогою функції linkage и знайдемо коефіцієнти кофенетичної кореляції:

cophenet\_coefs = pd.DataFrame(columns=metrics, index=methods, data=np.zeros((len(methods),len(metrics))))

X = countries[features].values

for i, method in enumerate(methods):

for j, metric in enumerate(metrics):

y = pdist(X, metric, p=4)

clustering = linkage(y, method)

k = cophenet(clustering, y)

cophenet\_coefs.iloc[i, j] = k[0]

cophenet\_coefs

Результати:

|  |  |  |  |
| --- | --- | --- | --- |
| **Метод**  **зв’язування** | **Метрика** | | |
| **2** (стандартизована евклідова) | **4** (Махаланобіса) | **6** (Чебишева) |
| **a** (ближнього сусіда) | 0.711502 | 0.590072 | 0.623862 |
| **b** (далекого сусіда) | 0.714839 | 0.517176 | 0.634918 |
| **c** (середнього зв’язку) | 0.749043 | 0.686904 | 0.657578 |

6. Знайдемо максимальний і мінімальний з отриманих коефіцієнтів:

max\_row\_idx, max\_col\_idx = np.unravel\_index(cophenet\_coefs.values.argmax(), cophenet\_coefs.values.shape)

print('Max coef: {} method and {} distance'.format(methods[max\_row\_idx], metrics[max\_col\_idx]))

min\_row\_idx, min\_col\_idx = np.unravel\_index(cophenet\_coefs.values.argmin(), cophenet\_coefs.values.shape)

print('Min coef: {} method and {} distance'.format(methods[min\_row\_idx], metrics[min\_col\_idx]))

Результати:

Max coef: **average** method and **seuclidean** distance

Min coef: **complete** method and **mahalanobis** distance

Для подальшого аналізу буде використовуватися метрика і метод, для яких коефіцієнт кофенетичної кореляції максимальний.

7. Побудуємо дендрограму результатів кластерного аналізу:

y = pdist(X, metrics[max\_col\_idx], p=4)

Z = linkage(y, methods[max\_row\_idx])

fig, ax = plt.subplots(figsize=(24, 8))

dn = dendrogram(Z, labels=countries['X0'].tolist())

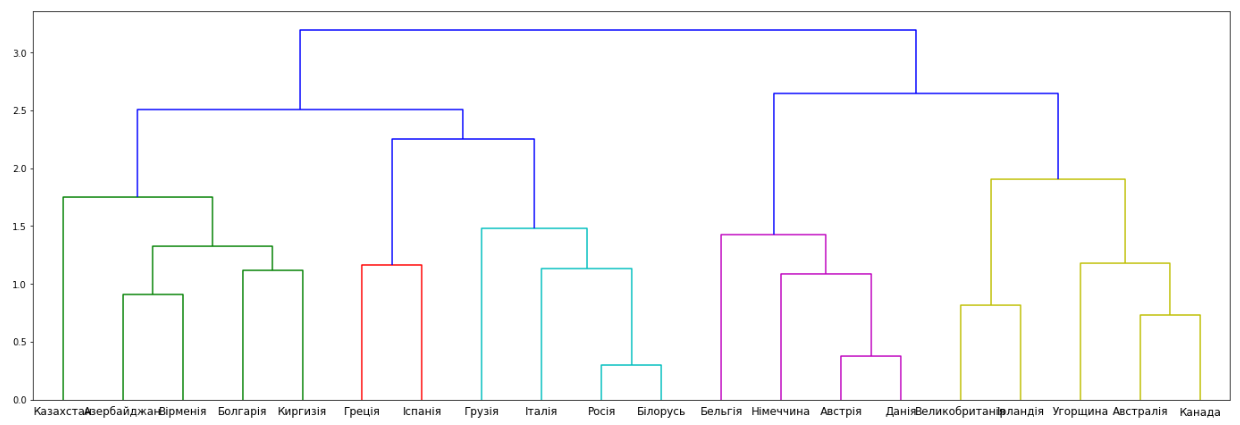


Рисунок 6 - Дендрограма результатів кластерного аналізу

8. Виділимо кластери шляхом завдання максимальної кількості кластерів n\_clusters:

n\_clusters = 4

from scipy.cluster.hierarchy import fcluster

maxclust\_clusters = fcluster(Z, n\_clusters, criterion='maxclust')

for i in range(np.unique(maxclust\_clusters).shape[0]):

print('Cluster #{}'.format(i + 1))

for index, cluster in enumerate(maxclust\_clusters):

if cluster == (i + 1):

print(countries['X0'].iloc[index], end=', ')

print('\n')

Результати:

Кластер #1

Азербайджан, Вірменія, Болгарія, Казахстан, Киргизія

Кластер #2

Росія, Білорусь, Греція, Грузія, Іспанія, Італія

Кластер #3

Австрія, Бельгія, Німеччина, Данія

Кластер #4

Австралія, Великобританія, Угорщина, Ірландія, Канада

9. Візуалізуємо залежність між приналежністю до кластерів та вихідними ознаками. Кожен кластер виділено окремим кольором.

maxclust\_result = countries[features]

maxclust\_result['cluster\_n\_clusters'] = maxclust\_clusters

sns.pairplot(maxclust\_result, hue='cluster\_n\_clusters', diag\_kind='hist')

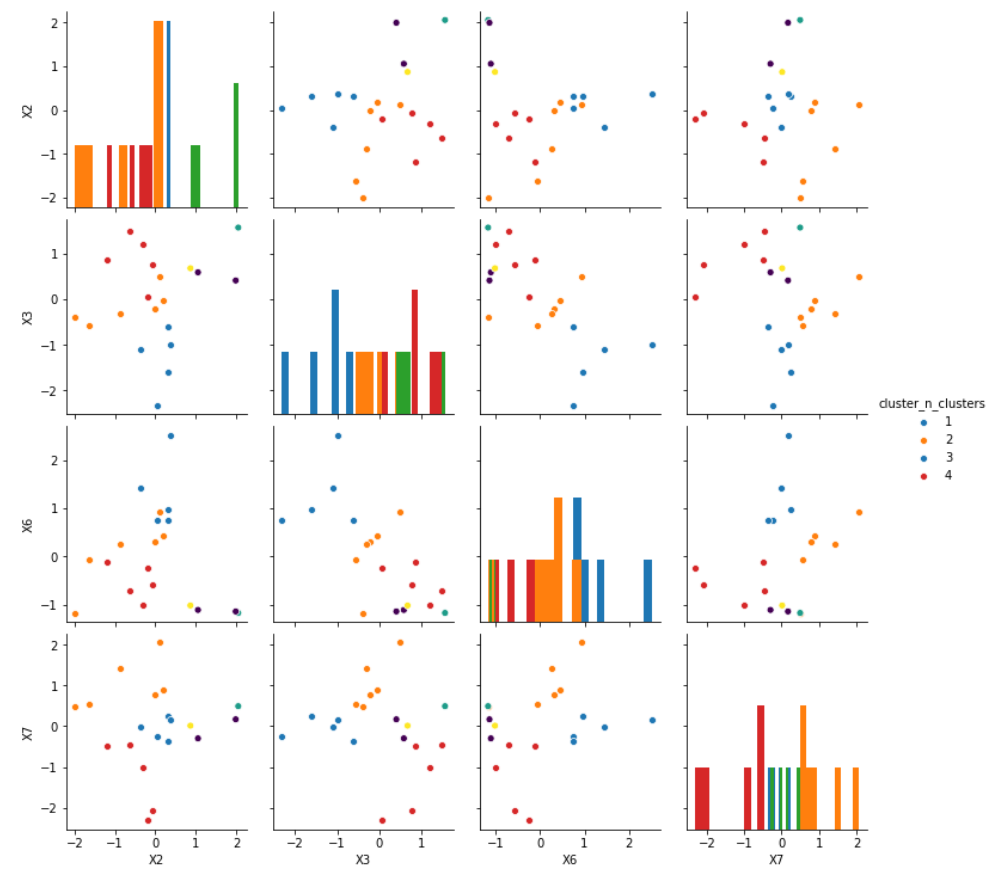


Рисунок 7 - Залежність між приналежністю до кластерів та вихідними ознаками

10. Виділимо кластери шляхом завдання порогової відстані між кластерами max\_dist:

max\_dist = 3

from scipy.cluster.hierarchy import fcluster

distance\_clusters = fcluster(Z, max\_dist, criterion='distance')

for i in range(np.unique(distance\_clusters).shape[0]):

print('Cluster #{}'.format(i + 1))

for index, cluster in enumerate(distance\_clusters):

if cluster == (i + 1):

print(countries['X0'].iloc[index], end=', ')

print('\n')

Результати:

Cluster #1

Росія, Азербайджан, Вірменія, Білорусь, Болгарія, Греція, Грузія, Іспанія, Італія, Казахстан, Киргизія,

Cluster #2

Австралія, Австрія, Бельгія, Великобританія, Угорщина, Німеччина, Данія, Ірландія, Канада,

**Висновки:** у результаті виконання лабораторної роботи ми освоїли методи ієрархічного кластерного аналізу даних на практиці. Була реалізована програма на мові програмування Python. На вхід подавалися країни із різними показниками по кількості споживання продуктів та стандартам життя. У результаті було створено декілька кластерів країн на основі кластерного аналізу цих показників. Також, у результаті виконання роботи використовувалися методи візуалізації результатів, такі як стовбчасті діаграми, денрограми.

Головна перевага кластерного аналізу: кластерний аналіз не вимагає апріорної інформації про дані і дозволяє розділити безліч досліджуваних об'єктів на групи схожих об'єктів - кластери. Це дає можливість різко скорочувати великі обсяги даних, робити їх компактними і наочними.

Кластеризація була проведена двома способами:

1. шляхом завдання максимальної кількості кластерів. Цей метод необхідно використовувати, коли є конкретне скінченне число груп, на які потрібно розбити дані. Коли використовувався даний метод, ми отримали у результаті 4 кластери країн, якщо подивитися на вихідні дані, то можна дійсно побачити, що в один кластер об’єднані країни з близькими показниками. Дуже наочно показує це є дендрограма (рис. 6).
2. шляхом завдання порогової відстані між кластерами. Цей метод потрібно використовувати, коли важлива не результуюча кількість кластерів, на які будуть розбиті дані, а наскільки близькими між собою можуть бути дані, щоб їх помістити в один кластер. Коли використовувався даний метод, ми отримали у результаті 2 кластери країн, тому що коефіцієнт максимальної відстані був досить високий. Чим менше коефіцієнт, тим більшу кількість кластерів можна отримати. Якщо проекспериментувати та поставити коефіцієнт 2 замість 3 (як було к ході роботи у 10 пункті), то отримаємо вже 5 кластерів, а не 2.

Таким чином, задача кластеризації актуальна у наші часи, вона використовується у медицині, біології, комп’ютерних науках, маркетингу та в інших. Спосіб розбиття на кластери треба обирати в залежності від задачі, яку потрібно вирішити.