МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІМ. Ю.М. ПОТЕБНІ ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

КАФЕДРА ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ СИСТЕМ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кваліфікаційна робота

другий (магістерський)
(рівень вищої освіти)

на тему **Особливості використання сучасного програмного забезпечення для кластерного аналізу множини об'єктів**

Виконав: студент 2 курсу, групи 8.1211-іпз-2
спеціальності 121 Інженерія програмного забезпечення
(код і назва спеціальності)
освітньої програми Інженерія програмного забезпечення
(код і назва освітньої програми)
П.Д. Качанов
(ініціали та прізвище)
Керівник доцент, А.І. Безверхий
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)
Рецензент директор ТОВ «Дискус»
Р.О. Лютий
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя
2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІМ. Ю.М. ПОТЕБНІ
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

Кафедра електроніки, інформаційних систем та програмного забезпечення

Рівень вищої освіти другий (магістерський)

Спеціальність 121 Інженерія програмного забезпечення
(код та назва)

Освітня програма Інженерія програмного забезпечення
(код та назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри Т.В. Критська

"_2_" вересня 2023 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Качанову Павлу Дмитровичу
(прізвище, ім'я, по батькові)

1.Тема роботи Особливості використання сучасного програмного забезпечення для кластерного аналізу множини об'єктів

керівник роботи Безверхий Анатолій Ігорович
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від _____

2. Строк подання студентом кваліфікаційної роботи 30 грудня 2023 р

3. Вихідні дані магістерської роботи
- комплект нормативних документів ;
- технічне завдання до роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
- огляд та збір літератури стосовно теми кваліфікаційної роботи;
- огляд та аналіз існуючих рішень та аналогів;
- дослідження проблеми розпізнавання мов та розробка методів її вирішення;
- створення програмного продукту та його опис;
- перелік вимог для роботи програми;
- дослідження поставленої проблеми та розробка висновків та пропозицій.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

слайдів презентації

6. Консультанти розділів магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата Завдання прийняв

7. Дата видачі завдання 01.09.2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1	Аналіз предметної області	02.10-10.10.23	виконано
2	Формулювання основної задачі дипломної роботи та узгодження її з науковим керівником	11.10-12.10.23	виконано
3	Аналіз існуючих методів рішення	13.10-14.10.23	виконано
4	Дослідження засобів кластерного аналізу множини об’єктів	15.10-20.10.23	виконано
5	Дослідження засобів виявлення контурів об’єктів та трансформації зображень	21.10-26.10.23	виконано
6	Узгодження подальших дій з науковим керівником	27.10-28.10.23	виконано
7	Знаходження множини об’єктів для подальшої кластеризації	29.10-30.10.23	виконано
8	Аналіз даних, Масштабування.	1.11.23	виконано
9	Представлення отриманих результатів науковому керівнику та узгодження плану подальшого дослідження	17.11-19.11.23	виконано
10	Реалізація функціоналу для кластерного аналізу множини об’єктів	20.10-09.11.23	виконано
11	Порівняння вихідних даних кластеризацій за рахунок метрик	10.11-17.11.23	виконано
12	Реалізація користувацького інтерфейсу для комп’ютерної системи	18.11-22.11.23	виконано
13	Оформлення звіту	23.11-27.11.23	виконано

Студент _____ Качанов П.Д.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Безверхий А.І.
(підпис) (прізвище та ініціали)

Нормоконтроль пройдено

Нормоконтролер _____ Скрипник І.А
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Сторінок: 75

Рисунків: 25

Таблиць: 8

Джерел: 12

Качанов П. Д. Особливості використання сучасного програмного забезпечення для кластерного аналізу множини об'єктів : кваліфікаційна робота магістра спеціальності 121 «Інженерія програмного забезпечення» / наук. керівник А. І. Безверхий. Запоріжжя : ЗНУ, 2023. 106 с.

Мета і завдання дослідження полягають у вивченні сучасних підходів для кластерного аналізу множини об'єктів, під час розробки комп’ютерної системи, що буде ефективно вирішувати задачу кластерного аналізу множини об'єктів. В якості множини об'єктів виступає датасет поведінкових даних користувачів банку.

У процесі дослідження була розглянута проблема попереднього аналізу даних для кластеризації, проблему кластерного аналізу множини об'єктів, проблему візуалізації результатів кластерного аналізу та різні підходи до вирішення цих проблем. Як результат, була розроблена комп'ютерна система на базі Python. Комп'ютерна система проводить аналіз даних заданої множини об'єктів, реалізує кластерний аналіз цих даних за допомогою різних методів кластеризації, візуалізує отримані результати різними засобами, порівнює отримані результати за допомогою метрик.

Ключові слова: комп’ютерна система, кластеризація, візуалізація, метрика, Python, датасет.

SUMMARY

Pages: 75

Figures: 25

Tables: 8

Sources: 12

Kachanov P. D. K. Peculiarities of using modern software for cluster analysis of multiple objects: master's thesis of specialty 121 "Software engineering" / science. manager A. I. Bezverkhy. Zaporizhzhia: ZNU, 2023. 106 p.

The purpose and task of the research is to study modern approaches for cluster analysis of a set of objects, as well as to develop a computer system that will effectively solve this problem, using as input data a dataset of behavioral data of a set of bank users.

In the research process, the problem of preliminary data analysis for clustering, the problem of cluster analysis of a set of objects, the problem of visualization of the results of cluster analysis and different approaches to solving these problems were considered. As a result, a computer system based on Python was developed. The computer system analyzes the data of a given set of objects, implements a cluster analysis of these data using various clustering methods, visualizes the obtained results in various ways, and compares the obtained results using metrics.

Keywords: computer system, clustering, visualization, metric, Python dataset.

ЗМІСТ

ВСТУП 10
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ КЛАСТЕРНОГО АНАЛІЗУ МНОЖИНИ ОБ'ЄКТІВ 15
1.1 Огляд проблеми кластерного аналізу множини об'єктів 15
1.2 Типи рішень проблеми кластерного аналізу множини об’єктів 16
1.2.1 Алгоритми ієрархічної кластеризації 16
1.2.2 Параметричні методи кластеризації 17
1.2.3 Алгоритми основані на теорії графів 17
1.3 Сфери застосування кластерного аналізу множини об’єктів 18
1.4 Аналіз існуючих програмних продуктів які вирішують проблему кластерного аналізу множини об'єктів 19
1.4.1 Рішення від компанії Dataviz 19
1.4.2 Рішення від компанії sciencehunter 20
1.5 Висновки з розділу 1 21
РОЗДІЛ 2 ДОСЛІДЖЕННЯ АЛГОРИТМІВ КЛАСТЕРНОГО АНАЛІЗУ МНОЖИНИ ОБ’ЄКТІВ, МЕТОДІВ ВІЗУАЛІЗАЦІЇ ТА ПОРІВНЯННЯ РЕЗУЛЬТАТІВ КЛАСТЕРИЗАЦІЇ. 22
2.1 Алгоритми кластерного аналізу застосованого в роботі 22
2.1.1 Kmeans(к-середніх) 22
2.1.2 Hierarchical Agglomarative clustering. 24
2.1.3 DBSCAN 25
2.1.4 OPTICS. 27
2.1.5 BIRCH. 28
2.1.6 Mean shift. 29

2.1.7 Gaussian Mixture. 30
2.2 Методи візуалізації результатів кластеризації 31
2.2.1 PCA 32
2.2.2 UMAP 33
2.3 Метрики порівняння кластерних аналізів 33
2.3.1 Обчислювальна складність алгоритмів кластерного аналізу 33
2.3.2 Silhouette Coefficient 34
2.3.3 Calinski-Harabasz Index 34
2.3.4 Davies-Bouldin Index 34
2.4 Висновки до розділу 2 35
РОЗДІЛ 3 РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ КЛАСТЕРНОГО АНАЛІЗУ МНОЖИНИ ОБ'ЄКТІВ 37
3.1 Аналіз даних датасету (множини об'єктів) 37
3.1.1 Таблиця типів даних властивостей множини об'єктів 38
3.1.2 Попередня обробка даних за допомогою Pandas 39
3.1.3 Масштабування даних 39
3.1.4 Статистика Хопкінса 40
3.2 Розробка інтерфейсу комп'ютерної системи 41
3.3 Дослідження реалізації методів кластерного аналізу на множині об'єктів 44
3.4 Візуалізація результатів методів кластерного аналізу на заданій множині об'єктів 48
3.4.1 Реалізація методу побудови візуалізації результатів кластеризації за допомогою Umap 48
3.4.2 Реалізація методу побудови візуалізації результатів кластеризації за допомогою PCA 49

3.4.3 Реалізація обрахунку показників та метрик кластеризації 50
3.5 Висновки до розділу 3 51
РОЗДІЛ 4 ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ РОБОТИ СИСТЕМИ НАД ЗАДАЧЕЮ КЛАСТЕРНОГО АНАЛІЗУ МНОЖИНИ ОБ'ЄКТІВ 51
4.1 Розв'язок задачі на основі методу k-means 52
4.1.1 Пошук оптимального значення кількості кластерів 52
4.1.2 Побудова моделі K–means з масштабованими даними 53
4.1.3 Побудова моделі K–means з PCA даними. 53
4.1.4 Побудова моделі K–means з UMAP даними. 54
4.1.5 Метрики результатів кластеризації методом K-means. 54
4.1.6 Аналіз отриманих метрик 55
4.1.7 Висновки розв'язку задачі на основі методу k-means 56
4.2 Розв'язок задачі методом Agglomarative clustering 56
4.2.1 Побудова дендрограми 56
4.2.2 Візуалізації результатів метода Agglomarative clustering 57
4.2.3 Обраховані метрики результатів кластеризації методом Agglomarative clustering 58
4.2.4 Аналіз отриманих метрик 58
4.2.5 Висновки розв'язку задачі на основі методу Agglomarative clustering 59
4.3 Розв'язок задачі методом DBSCAN 59
4.3.1 Знаходження EPS 59
4.3.2 Візуалізації результатів метода Dbscan 61
4.3.3 Обраховані метрики результатів кластеризації методом Dbscan 62
4.3.4 Аналіз отриманих метрик 62
4.3.5 Висновки розв'язку задачі на основі методу Dbscan 62

4.4 Розв’язок задачі методом OPTICS 63
4.4.1 Візуалізації результатів метода OPTICS. 63
4.4.2 Обраховані метрики результатів кластеризації методом OPTICS 64
4.4.3 Аналіз отриманих метрик 64
4.4.4 Висновки розв’язку задачі на основі методу OPTICS 64
4.5 Розв’язок задачі методом MeanShift 65
4.5.1 Візуалізації результатів метода MeanShift. 65
4.5.2 Обраховані метрики результатів кластеризації методом MeanShift 66
4.5.3 Аналіз отриманих метрик 67
4.5.4 Висновки розв’язку задачі на основі методу *MeanShift* 67
4.6 Розв’язок задачі методом Birch 68
4.6.1 Візуалізації результатів метода Birch 68
4.6.2 Обраховані метрики результатів кластеризації методом Birch 69
4.6.3 Аналіз отриманих метрик 69
4.6.4 Висновки розв’язку задачі на основі методу Birch 69
4.7 Розв’язок задачі методом Gaussian Mixture 70
4.7.1 Візуалізації результатів метода Gaussian Mixture 70
4.7.2 Обраховані метрики результатів кластеризації методом *Gaussian Mixture* 71
4.7.3 Аналіз отриманих метрик 71
4.7.4 Висновки розв’язку задачі на основі методу Gaussian Mixture 71
4.7.5 Висновки до розділу 4 72
ВИСНОВКИ 73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 74

ВСТУП

Актуальність теми

Інтерес до методів аналізу даних виник у зв'язку з розвитком засобів збору і збереження даних, що дозволив накопичувати великі обсяги інформації. Серед методів інтелектуального аналізу даних особливе місце займає кластеризація. Кластеризація - розбиття заданої вибірки об'єктів (ситуацій) на підмножини, що називаються кластерами, так що кожен кластер складається з подібних об'єктів, а об'єкти різних кластерів не перетинаються.

Кластерний аналіз дозволяє розглядати досить великий обсяг інформації та різко скорочувати, стискати великі масиви соціально-економічної інформації, робити їх компактними та наочними. Завдання кластеризації відноситься до статистичної обробки, а також до широкого класу задач навчання без учителя. Головна мета кластерного аналізу — знаходження груп схожих об'єктів в вибірці. Кластерний аналіз використовують в археології, медицині, психології, хімії, біології, державному управлінні, тощо.

Слід зазначити, що проблема кластеризації не є тривіальною задачею, особливо у випадках даних високої розмірності, з якими необхідно працювати в більшості реальних досліджень. Використання звичайних методів кластеризації як правило не є ідеальним рішенням в таких ситуаціях, тому є необхідним дослідження більшої кількості алгоритмів кластерного аналізу, їх порівняння. та способів візуалізації результатів роботи цих алгоритмів.

Таким чином, актуальним є створення комп’ютерної системи для дослідження особливостей використання сучасного програмного забезпечення для кластерного аналізу множини об'єктів, яка різними засобами проводить кластерний аналіз заданої множини об'єктів, візуалізує результати аналізів та порівнює ці результати між собою за допомогою метрик.

Мета і завдання дослідження

Мета і завдання дослідження полягають у вивченні особливості використання сучасного програмного забезпечення для кластерного аналізу множини об'єктів, а також у розробці комп’ютерної системи, що буде ефективно вирішувати цю задачу

.

Об’єкт дослідження

Об’єктом дослідження є зразок набору даних узагальнює поведінку використання приблизно 9000 активних власників кредитних карток.

Предмет дослідження

Предметом дослідження є засоби кластерного аналізу заданої множини об'єктів та особливості використання цих засобів.

Методи дослідження

Для вирішення поставленої задачі використовуються наступні методи дослідження:

1. Аналіз особливостей та існуючих рішень для проблеми кластерного аналізу множини об'єктів.
2. Синтез отриманих в процесі дослідження проблем та методів їх вирішення.
3. Аналіз бібліотек кластерного аналізу та способів візуалізації.
4. Аналіз множини об'єктів.
5. Експеримент з використанням різних алгоритмів кластерного аналізу.
6. Аналіз результатів кластеризацій.
7. Експериментування з різними засобами представлення даних.

Наукова новизна одержаних результатів

Наукова новизна одержаних результатів дослідження полягає у тому, що для вирішення задачі кластерного аналізу множини об’єктів були використані

різні сучасні підходи, та досліджено особливості використання цих підходів на конкретній задачі.

Практичне значення одержаних результатів

Практичне значення одержаних результатів дослідження полягає у тому, що бу проведений комп’ютерна експеримент, який різними засобами проводить кластерний аналіз заданої множини об'єктів, візуалізує результати аналізів та порівнює ці результати між собою за допомогою метрик.

Апробація одержаних результатів

Результати дослідження були представлені ІІІ Всеукраїнської науково-практичної конференції за участю молодих науковців «АКТУАЛЬНІ ПИТАННЯ СТАЛОГО НАУКОВО-ТЕХНІЧНОГО ТА СОЦІАЛЬНО-ЕКОНОМІЧНОГО РОЗВИТКУ РЕГІОНІВ УКРАЇНИ». Запоріжжя : Запорізький національний університет, 2023. С. .

Глосарій

Кластерний аналіз (англ. Clustering) — задача розбиття заданої вибірки об'єктів (ситуацій) на підмножини, які називаються кластерами, так, щоб кожен кластер складався з схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися. Завдання кластеризації належить до статистичної обробки, а також до широкого класу завдань некерованого навчання.

Кластер (англ. cluster) — група однакових або подібних елементів, зібраних разом або близько один до одного

Навчанняя без учителя (англ. unsupervised learning) — це парадигма машинного навчання, в якій, на відміну від керованого та напівкерованого навчання, алгоритми навчаються образів виключно з немічених даних.

Множина об’єктів, датасет (англ. Dataset) **—** це оброблена та структурована певним чином інформація, необхідна для навчання нейронної мережі.

Метод головних компонент (англ. principal component analysis, PCA) — метод факторного аналізу в статистиці, який використовує ортогональне перетворення множини спостережень з можливо пов'язаними змінними (сутностями, кожна з яких набуває різних числових значень) у множину змінних без лінійної кореляції, які називаються головними компонентами.

Апроксимація та проекція рівномірного різноманіття, (англ. uniform manifold approximation and projection, UMAP) — це метод зменшення розмірності, який можна використовувати для візуалізації, але також для загального нелінійного зменшення розмірності.

Масштабування даних (англ. Data Scaling) — метод нормалізації діапазону незалежних змінних або ознак даних.

Індекс Калинського-Харабаша (англ. Calinski-Harabasz Index, CHI) — це внутрішня оціночна метрика, де оцінка якості кластеризації ґрунтується виключно на наборі даних і результатах кластеризації, а не на зовнішніх мітках істинності.

Коефіцієнт силуету (англ. silhouette coefficient, SC, SI)— це величина, що дозволяє оцінити рівень розвитку побудованої кластерної структури навчальних даних, тобто. оцінити якість кластеризації

Skilit-Learn — це бібліотека, призначена для машинного навчання, написана мовою програмування Python і розповсюджується як безкоштовне програмне забезпечення.

Python — це інтерпретована об’єктно-орієнтована мова програмування високого рівня із динамічною типізацією.

Статистика Хопкінса (англ. Hopkins Statistic) — це тест, який визначає, наскільки набір даних підходить для кластеризації. Значення, близьке до 0,5, вказує на те, що дані мають випадковий розподіл і тому не підходять для кластеризації. З іншого боку, якщо значення більше 0,7, набір даних, ймовірно, підходить для кластеризації.

Візуалізація (англ. visualization) — це процес побудови графічного образу даних, що допомагає у процесі загального аналізу даних вбачати аномалії, структури.

Pandas — це швидкий, потужний, гнучкий і простий у використанні інструмент для аналізу та обробки даних з відкритим вихідним кодом, побудований на основі мови програмування Python.

CustomTkinter — це бібліотека Python на основі Tkinter, яка надає нові, сучасні та повністю настроювані віджети. Вони створюються та використовуються як звичайні віджети Tkinter,

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ КЛАСТЕРНОГО АНАЛІЗУ МНОЖИНИ ОБ'ЄКТІВ

1.1 Огляд проблеми кластерного аналізу множини об'єктів

Кластеризація - це метод аналізу даних, коли об'єкти поділяються на групи або "кластери" на основі схожості між ними. Головна ідея полягає в тому, щоб об'єднати об'єкти, які подібні, один до одного, і розділити ті, які відрізняються. Цей термін, уперше введений Тріоном (Tryon) у 1939 році.

На відміну від задач класифікації, кластерний аналіз не вимагає апріорних припущень про набір даних, не накладає обмеження на показ досліджуваних об’єктів, дозволяє аналізувати показники різних типів даних (інтервальні дані, частоти, бінарні дані). При цьому необхідно пам’ятати, що змінні повинні вимірюватися в порівнюваних шкалах.

Задачі які виконує кластерний аналіз можна об’єднати в такі групи:

1. Розробка типології або класифікації.
2. Дослідження корисних концептуальних схем групування об’єктів.
3. Представлення гіпотез на основі дослідження даних.
4. Перевірка гіпотез або досліджень для визначення, чи дійсно типи (групи), виділені тим або іншим способом.

Як правило, при практичному використанні кластерного аналізу одночасно розв’язуються декілька із зазначених груп задач.

Рішенням задачі кластерного аналізу є розбиття, що задовольняє деякому критерію оптимальності. Цей критерій може являти собою деякий функціонал, що виражає рівні бажаності різних розбитів і угруповань, який називають цільовою функцією.

Кожен з методів кластерного аналізу не є еталонним рішенням поставленої задачі, через такі ознаки як:

1. Кожен з методів дає різні результати кластеризації, і не існує однозначного критерію для перевірки правильності результатів цих методів, але є низка метрик (такі метрики як Silhouette Coefficient, Calinski-Harabasz Index, Davies-Bouldin Index) котрі можуть оцінити наскільки результат може бути імплементований у ту чи іншу задачу.

2. Проблема визначення кількості кластерів. Вирішення цієї проблеми займає велику частину кожного з методів кластеризації. Існує декілька методів рішення: визначення кількості кластерів на основі метрики(Діаграма ліктя), за допомогою “відстані” між об'єктами в множині об'єктів та за допомогою щільності об’єктів. Кожен з перерахованих варіантів може дати різні результати але вони будуть оптимальні для того чи іншого алгоритму

1.2 Типи рішень проблеми кластерного аналізу множини об’єктів

1.2.1 Алгоритми ієрархічної кластеризації

Ієрархічна кластеризація - це метод групування об'єктів або даних у структуру ієрархічних кластерів. Цей процес відбувається шляхом послідовного з'єднання або розділення кластерів залежно від схожості між ними. Кожен об'єкт або група об'єктів починає як окремий кластер, і з часом вони об'єднуються або розділяються до утворення ієрархічної структури.

Алгоритми ієрархічної кластеризації включають кроки визначення відстані між кластерами і стратегії об'єднання або розділення. Перші об'єднують найближчі кластери, поки не залишиться лише один кластер на вершині ієрархії. Другі розділяють кластери на менші групи, поки кожен об'єкт не стане окремим кластером.

Цей підхід дозволяє візуалізувати структуру кластерів у вигляді дерева, яке називається дендрограмою, що полегшує аналіз і розуміння взаємозв'язків між групами даних. На рисунку 1 зображена дендрогра

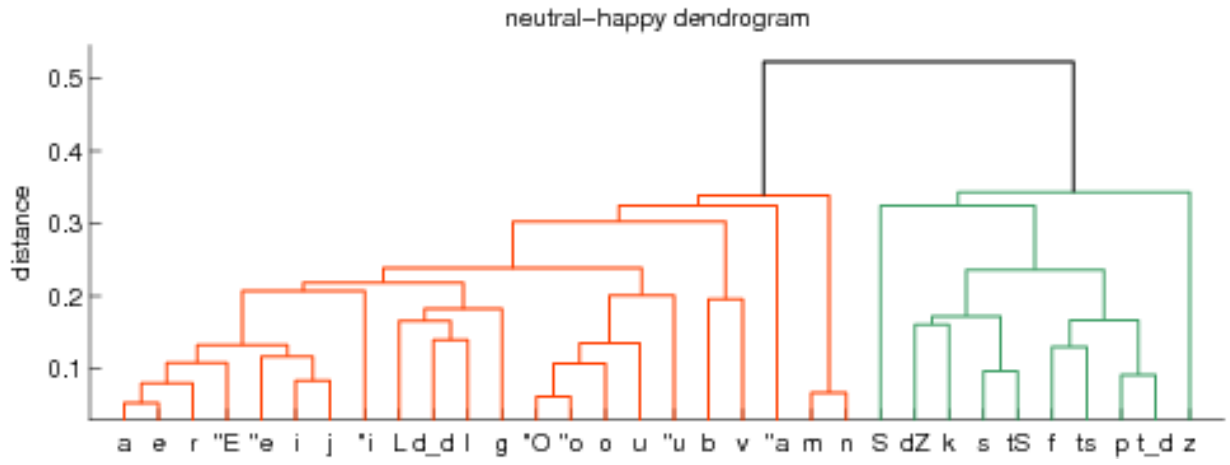


Рисунок 1 — Приклад дендрограми

1.2.2 Параметричні методи кластеризації

Параметричні методи кластеризації базуються на припущенні щодо параметрів розподілу даних у кластерах. Це означає, що алгоритми передбачають конкретні моделі розподілу даних та їх параметри. Такі методи часто використовують статистичні методи та теорію ймовірностей для призначення об'єктів до кластерів. Ці методи вимагають від користувача обрати конкретні моделі розподілу та проводити припущення щодо їхніх параметрів. Інші непараметричні методи, такі як ієрархічна кластеризація або методи основані на відстанях, можуть бути менш чутливими до конкретних припущень про форму та розподіл кластерів.

1.2.3 Алгоритми основані на теорії графів

Методи основані на графах в кластеризації використовують структуру графа для визначення кластерів. Граф представляє взаємозв'язки між об'єктами, і кластери визначаються як групи взаємопов'язаних або подібних вузлів у графі. Ці методи дозволяють враховувати взаємозв'язки між об'єктами та структуру графа при визначенні кластерів, що робить їх ефективними в різних областях, таких як соціальні мережі, біологічні мережі та аналіз текстів.

1.3 Сфери застосування кластерного аналізу множини об’єктів

Кластерний аналіз знаходить широке застосування в різних галузях, де існує потреба в групуванні об'єктів за схожістю. Нижче наведені деякі з цих галузь, частина з яких потребує також вирішення проблеми класифікації.

Маркетинг і Сегментація Ринку: Кластерний аналіз може визначити групи клієнтів, які мають подібні покупові звички, інтереси чи демографічні характеристики. Це дозволяє підприємствам створювати персоналізовані стратегії продажу та маркетингу для кожного сегменту.

Медична Діагностика: Кластерний аналіз може бути використаний для групування пацієнтів за спільними медичними ознаками, такими як симптоми, результати тестів або генетична інформація. Це допомагає визначати схожі підходи до діагностики та лікування.

Біологічні Дослідження: У біології кластерний аналіз може бути використаний для групування організмів чи генів за схожістю в генетичних даних. Це допомагає в розумінні еволюційних відносин і ідентифікації нових видів.

Соціальна Аналітика та Мережі: Кластерний аналіз у соціальній аналітиці може визначати групи користувачів за їхніми соціальними взаємодіями та інтересами. Це корисно для рекламодавців та розробників соціальних платформ.

Обробка Зображень та Відео: Кластерний аналіз може бути використаний для групування об'єктів на зображеннях або відео за їхньою формою, кольором чи текстурою. Це корисно для автоматичного розпізнавання об'єктів та аналізу великих обсягів візуальної інформації.

Текстова Аналітика: Кластерний аналіз у текстовій аналітиці може групувати документи або слова за схожістю тематики або лексичних характеристик. Це допомагає в семантичному аналізі тексту.

1.4 Аналіз існуючих програмних продуктів які вирішують проблему кластерного аналізу множини об'єктів

1.4.1 Рішення від компанії Dataviz

Dataviz.io надає онлайн-сервіси для аналізу даних у сфері роздрібної торгівлі та харчової промисловості.

Компанія пропонує онлайн-сервіси .Основною метою компанії є автоматизація наступних процесів:

1. Аналіз даних показників продажів.
2. Визначення головних товарів в обігу.
3. Побудова стратегій розвитку.
4. Планування акцій.
5. Виявлення неприбуткових товарів.

Продукт виконує ці процеси за допомогою алгоритму k-means(к-середніх). Приклад роботи зображено на Рис 2.

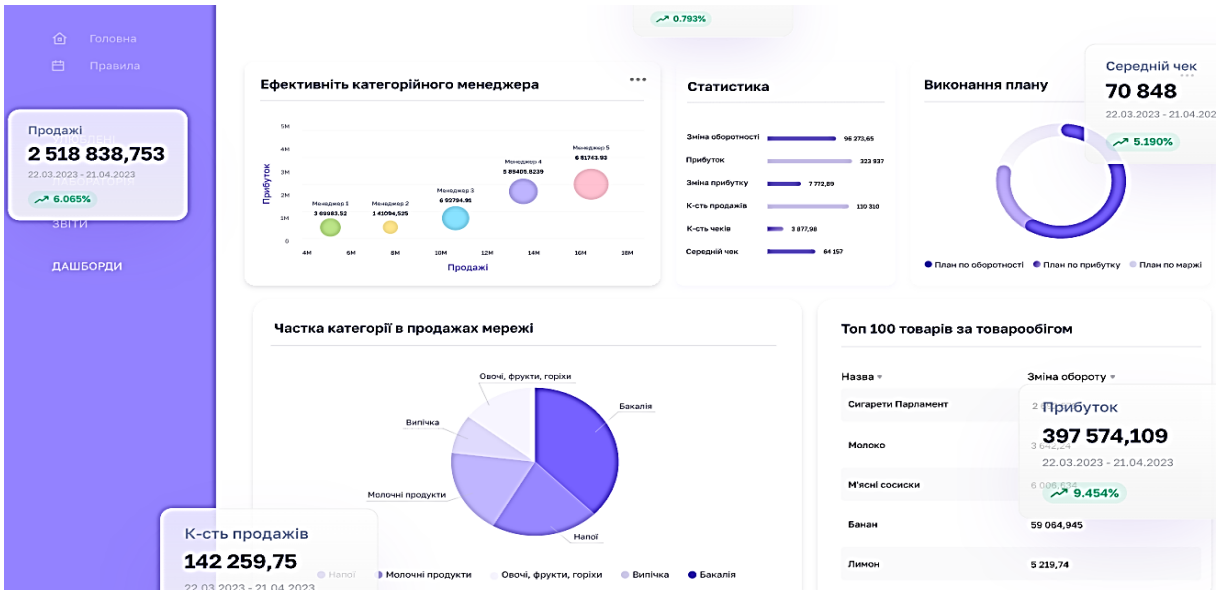


Рисунок 2 — Рішення від компанії Dataviz

До переваг рішень від компанії Dataviz можна віднести:

1. Надання повного апаратно-програмного комплексу для вирішення проблеми кластерного аналізу множини об'єктів.
2. Побудову стратегій подальшого покращення.
3. Визначення недоліків стратегій.

До недоліків рішень від компанії Dataviz можна віднести:

1. Вузька специфіка.
2. Використання тільки алгоритму k-means(к-середніх), використання інших алгоритмів кластерного аналізу дасть більш точні результати на специфічних датасетах.

1.4.2 Рішення від компанії sciencehunter

Sciencehunter.net — сервіс дозволяє працювати з наступними інструментами аналітики: візуалізація, гістограма, де можна вибрати різні метрики та методи поєднання, аналіз даних з аналогічною функціональністю. Приклад гістограми зображено на рис 3.

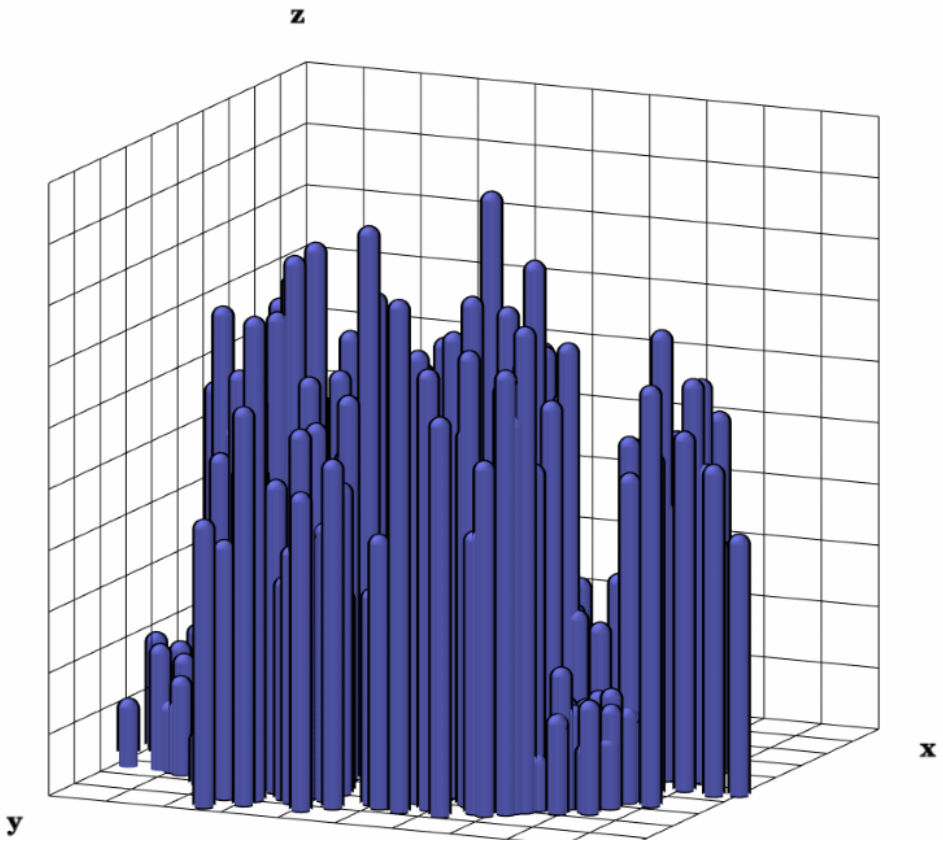


Рисунок 3 — Приклад гістограми від sciencehunter

До переваг рішень від компанії sciencehunter можна віднести:

1. Інтерактивні графіки, які зрозуміло пояснюють збирання даних.
2. Набір угруповань виконується досить просто завдяки зрозумілому інтерфейсу.
3. Візуальні налаштування виглядають якісно і зрозуміло.

До недоліків рішень від компанії infsoft можна віднести:

1. Рішення компанії в основному побудовано за допомогою ієрархічного(агломератному) кластерному аналізу.
2. Побудова тільки за одним методом візуалізації а саме PCA.

1.5 Висновки з розділу 1

Кластеризація - це метод аналізу даних, коли об'єкти поділяються на групи або "кластери" на основі схожості між ними, Алгоритми кластерного аналізу діляться на три типи: параметричні методи кластеризації, ієрархічні та основані на теорії графів.

Проведено аналіз існуючих рішень що показав необхідність побудови комп'ютерної системи яка різними засобами проводить кластерний аналіз заданої множини об'єктів, візуалізує результати аналізів та порівнює ці результати між собою за допомогою метрик.

РОЗДІЛ 2 ДОСЛІДЖЕННЯ АЛГОРИТМІВ КЛАСТЕРНОГО АНАЛІЗУ МНОЖИНИ ОБ'ЄКТІВ, МЕТОДІВ ВІЗУАЛІЗАЦІЇ ТА ПОРІВНЯННЯ РЕЗУЛЬТАТІВ КЛАСТЕРИЗАЦІЇ.

2.1 Алгоритми кластерного аналізу застосованого в роботі

2.1.1 Kmeans(к-середніх)

Алгоритм Kmeans — це ітераційний алгоритм, який намагається розділити набір даних на попередньо визначені K окремі підгрупи (кластери), які не перекриваються, де кожна точка даних належить лише одній групі. Він намагається зробити внутрішньо кластерні точки даних якомога подібнішими, водночас зберігаючи кластери максимально різними (далеко). Він призначає точки даних кластеру таким чином, щоб сума квадратів відстані між точками даних і центроїдам кластера (середнє арифметичне всіх точок даних, які належать цьому кластеру) була мінімальною. Чим менше варіацій у кластерах, тим більш однорідними (схожими) є точки даних у межах одного кластера.

Алгоритм kmeans працює наступним чином(рис. 4):

Крок 1. Призначити кожен елемент даних випадковим чином кластеру (від 1 до k).

Крок 2. Використовуючи початкове призначення, визначити центроїди кожного кластеру.

Крок 3. З огляду на нові центроїди, призначити кожний елемент даних так, щоб він знаходився в кластері його найближчого центроїда. Дистанція між центроїдам та елементами вибірки визначається по формулі.

Крок 4. Повторно обчислити центроїди, як на кроці 2. Повторювати кроки 3 та 4, поки центроїди не перестане змінюватися

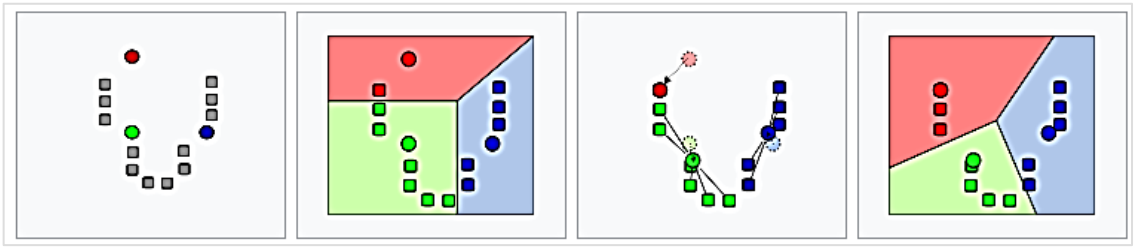


Рисунок 4 — Демонстрація роботи алгоритму kmeans

Варто зауважити: Оскільки алгоритми кластеризації, включаючи kmeans, використовують вимірювання на основі відстані для визначення подібності між точками даних, рекомендується стандартизувати дані, щоб мати середнє значення нуль і стандартне відхилення одиниці, оскільки майже завжди об’єкти в будь-якому наборі даних матимуть різні одиниці вимірювання наприклад вік проти доходу.

Враховуючи ітераційну природу kmeans і випадкову ініціалізацію центроїдів на початку алгоритму, різні ініціалізації можуть призвести до різних кластерів, оскільки алгоритм kmeans може застрягти в локальному оптимумі та не збігатися з глобальним оптимумом. Тому рекомендується запускати алгоритм, використовуючи різні ініціалізації центроїдів і вибирати результати виконання, які давали б нижчу суму квадратів відстані.

Призначення прикладів не змінюється – це те саме, що відсутність змін у варіації всередині кластера.

На відміну від навчання з вчителем, де ми маємо приклади вірного розбиття для оцінки продуктивності моделі, аналіз кластеризації не має надійної оціночної метрики, за допомогою якої ми можемо оцінити результати різних алгоритмів кластеризації. Крім того, оскільки kmeans вимагає k як вхідні дані і не вивчає його з даних, немає правильної відповіді щодо кількості кластерів, які ми повинні мати в будь-якій задачі. Іноді знання предметної галузі та інтуїція можуть допомогти, але зазвичай це не так. У методології кластерного прогнозування ми можемо оцінити, наскільки добре працюють

моделі на основі різної кількості кластерів k, оскільки кластери використовуються в подальшому моделюванні.

Метод Ліктя дає нам уявлення про те, наскільки оптимальна k-кількість кластер на основі суми квадратів відстаней між точками даних і призначеними для них центроїдами кластерів. Оптимальне значення k у місці на діаграмі, де сума квадратів відстаней починає вирівнюватися та утворювати лікоть.

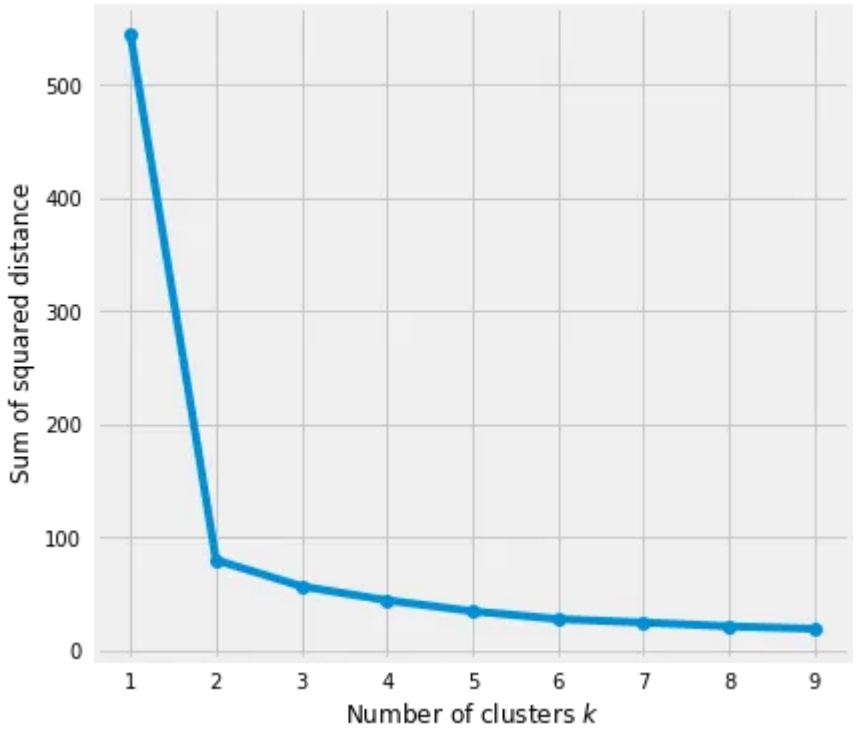


Рисунок 5 — Приклад діаграми ліктя

2.1.2 Hierarchical Agglomarative clustering.

Агломератний підхід (злиття): Алгоритми агломеративної кластеризації починають з кожного об'єкта в окремому кластері, а потім поступово об'єднують найближчі кластери на кожному кроці, поки не буде сформована одна велика ієрархічна структура. На кожному кроці об'єднання обирається два найближчих.

Алгоритми ієрархічної кластеризації можуть бути представлені у формі дерева, відомого як дендрограмма(рис.6), яка відображає послідовність

об'єднань кластерів. Дендрограмма надає візуальне уявлення про структуру кластерів і дозволяє вибрати оптимальне розбиття на кластери шляхом вибору під кластерів залежно від заданого критерію.

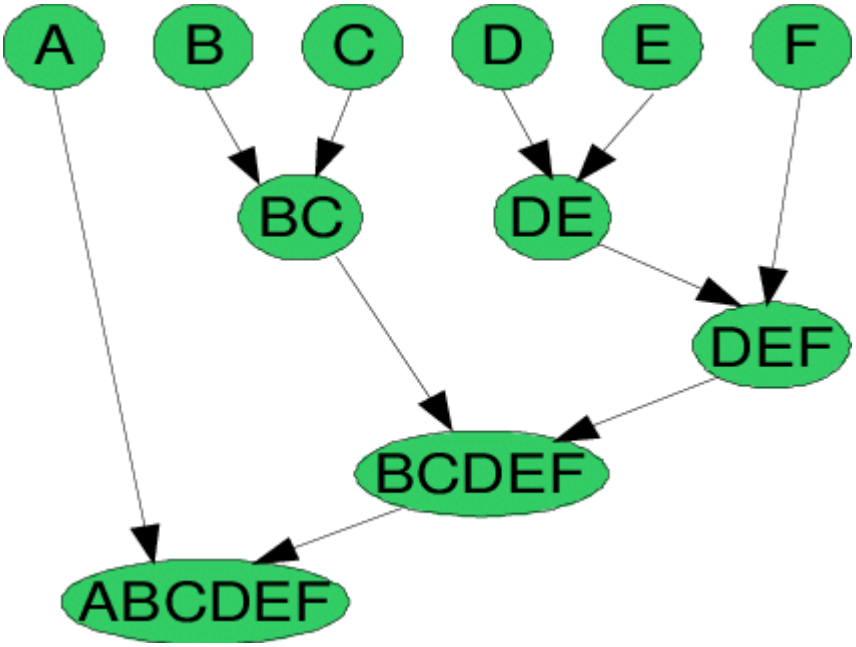


Рисунок *6* — *Зображення агломератного методу*

2.1.3 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) є алгоритмом кластеризації, який базується на щільності точок даних і використовує їх просторову структуру для виявлення кластерів. Основна ідея DBSCAN полягає в тому, щоб знаходити області високої щільності точок, розділяючи їх від областей низької щільності.

DBSCAN використовує два основних параметри: радіус ε (epsilon) і мінімальну кількість сусідів (minPts). Алгоритм починається з вибору випадкової невідвіданої точки даних і перевіряє, чи має вона щонайменше minPts сусідів всередині радіуса ε. Якщо так, то ця точка вважається ядром (core point) і утворює новий кластер. Всі точки, які можна досягти від цього ядра за допомогою послідовних сусідніх точок, також додаються до цього

кластера. Цей процес продовжується рекурсивно, поки не будуть відшукані всі досяжні точки.

Якщо точка не має достатньо сусідів всередині радіуса ε, вона вважається граничною точкою (border point). Граничні точки не утворюють окремий кластер, але можуть належати до кластерів ядер, які знаходяться поруч.

Точки, які не є ні ядрами, ні граничними точками, вважаються шумовими точками (noise points) і не призначаються жодному кластеру.

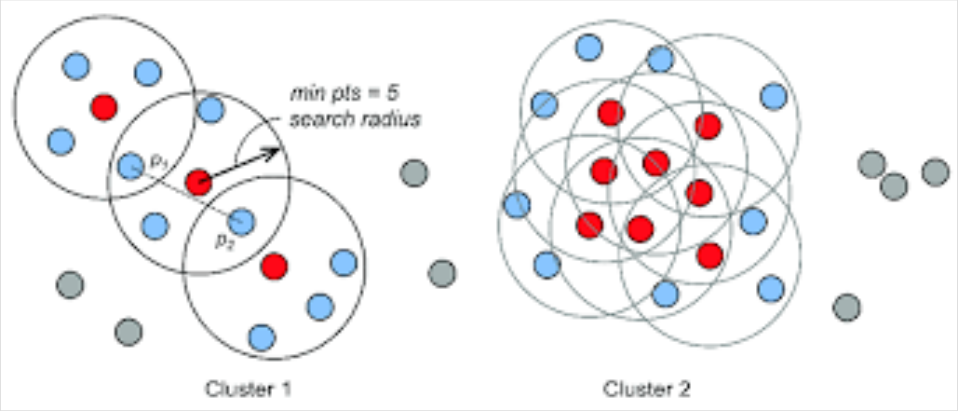


Рисунок 7 — Приклад роботи DBSCAN

DBSCAN має декілька переваг, зокрема, він добре працює з даними, де кластери мають різну щільність або форму. Він також може виявляти аномалії або шумові точки.

Як було зазначено раніше DBSCAN використовує два основних параметри: радіус ε (epsilon) і мінімальну кількість сусідів (minPts). Зверномо увагу на способі отримання припущення про оптимальне значення цих параметрів

MinPts: Немає автоматичного способу визначення значення MinPts для DBSCAN. Зрештою, значення MinPts слід установлювати, використовуючи знання домену та знайомство з набором даних. Але є кілька практичних правил вибору значення MinPts:

1. Чим більший набір даних, тим більшим має бути значення MinPts
2. Якщо набір даних є шумами, виберіть більше значення MinPts
3. Якщо ваші дані мають більше двох вимірів, виберіть MinPts зі значенням подвійної розмірності датасету (Sander et al., 1998).

Epsilon: Метод К-відстаней автоматичного визначення оптимального значення ε. Цей метод обчислює середню відстань між кожною точкою та її k найближчими сусідами, де k = вибране значення MinPts. Потім середні значення k-відстаней відображаються в порядку зростання на графіку k-відстаней. Ви знайдете оптимальне значення для ε в точці максимальної кривини (тобто в точці ліктя).

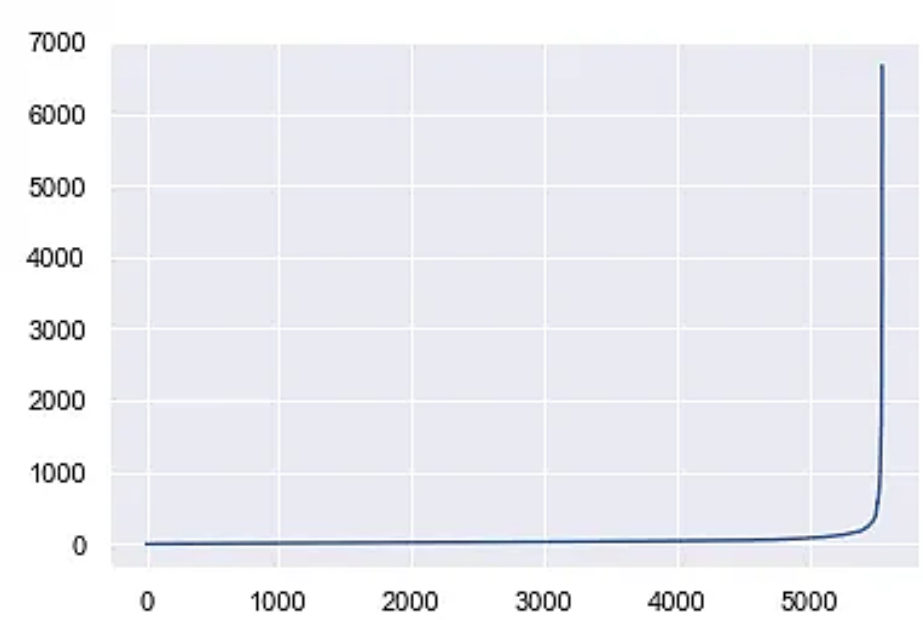


Рисунок 8 — Приклад діаграми k-відстаней

2.1.4 OPTICS.

OPTICS (Ordering Points To Identify the Clustering Structure) — це алгоритм кластеризації на основі щільності, який можна розглядати як узагальнення DBSCAN. Однак він відомий своєю здатністю ідентифікувати структури різної щільності в наборі даних замість використання одного значення eps. Принцип роботи: Відстань досяжності розраховується для кожної точки даних. Це визначає, наскільки близька точка до іншої точки з

певною щільністю. На основі цих відстаней створюється графік досяжності. Провали на графіку позначають кластери на основі щільності.

Переваги OPTICS: можливість виявляти кластери різної щільності, незалежність від вхідного параметра eps, автоматично розрізняє шуми.

Проблеми OPTICS: - Витрати на обчислення можуть бути високими, особливо для великих наборів даних, інтерпретація результатів може бути дещо складнішою порівняно з DBSCAN.

OPTICS дуже корисний для наборів даних із областями різної щільності. Він успішно виявляє переходи між кластерами різної щільності.

2.1.5 BIRCH.

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) це алгоритм кластеризації, який може кластеризувати великі набори даних, спочатку генеруючи невеликий і компактний підсумок великого набору даних, який зберігає якомога більше інформації. BIRCH дуже корисна завдяки простоті виконання.

BIRCH будує дендрограму під назвою дерево кластеризації ознак (CF-дерево) під час сканування набору даних. Кожен запис у дереві CF представляє кластер об’єктів і характеризується 3-кортежем: (N, LS, SS), де N – кількість об’єктів у кластері, LS — лінійна сума N точок, SS – квадратна сума N точок.

BIRCH — це масштабований метод кластеризації, заснований на ієрархічній кластеризації, і вимагає лише одноразового сканування набору даних, що робить його швидким для роботи з великими наборами даних. Існує в основному чотири фази, за якими слідує алгоритм BIRCH.

Фаза 1: сканування всіх даних і створення початкового дерева CF у пам’яті.

Фаза 2: ущільнення до бажаної довжини шляхом побудови меншого дерева CF.

Фаза 3: Глобальна кластеризація.

Фаза 4: Уточнення кластера(необов’язково), вимагає більше проходів даних для уточнення результатів.

2.1.6 Mean shift.

Mean shift — це алгоритм машинного навчання без контролю, який використовується для ідентифікації кластерів у наборі даних. Це метод кластеризації на основі щільності, який зосереджується на пошуку областей з високою щільністю та ітеративному зміщенні точок даних до найвищої щільності точок, звідси й назва «зміщення середнього значення». Алгоритм не потребує попередньої інформації про кількість кластерів у даних, що робить його особливо корисним для дослідницького аналізу даних.

Mean shift можна зрозуміти в три основні етапи:

1. Оцінка щільності ядра: перший крок передбачає оцінку базової функції щільності ймовірності (PDF) точок даних. Зазвичай це робиться за допомогою оцінки щільності ядра, де кожна точка даних представлена функцією ядра з центром у цій точці. Функція ядра визначає вагу, призначену кожній точці даних у процесі оцінки щільності.
2. Зміщення точок даних: на другому кроці алгоритм ітеративно зміщує точки даних у бік регіонів з більшою щільністю. Зсув визначається обчисленням вектора середнього зсуву для кожної точки даних, який представляє напрямок і величину зсуву. Вектор середнього зсуву обчислюється як середнє зважене різниць між точкою даних і сусідніми точками, де ваги визначаються ядерною функцією.
3. Конвергенція та кластерна ідентифікація: Алгоритм продовжує зміщувати точки даних, доки не буде досягнуто конвергенції. Збіжність відбувається, коли середні вектори зсуву стають дуже малими або незначними. Після досягнення конвергенції кінцеве положення кожної точки даних представляє центр кластера. Алгоритм призначає кожну точку даних найближчому центру кластера, таким чином ідентифікуючи кластери в даних.

2.1.7 Gaussian Mixture.

Gaussian Mixture — це метод, який використовується в кластеризації та базується на оцінці щільності ймовірності. Точки даних генеруються із суміші кількох розподілів Гауса з різними середніми значеннями та дисперсіями. Це може забезпечити ефективні результати кластеризації в певних задачах.

На високому рівні GMM поєднує численні розподіли Гауса для моделювання даних. Набір k розподілів Гауса підбирається до даних із такими параметрами, як середнє значення, дисперсія та вага, оцінені для кожного кластера. Після вивчення параметрів для кожної точки даних можна обчислити ймовірності, щоб визначити, до якого кластера належить точка.

Кожен розподіл зважений ваговим коефіцієнтом (π), щоб врахувати різні розміри вибірки в різних кластерах(рис. 9). Алгоритм GMM складається з двох кроків: очікування (E) і максимізації (M).

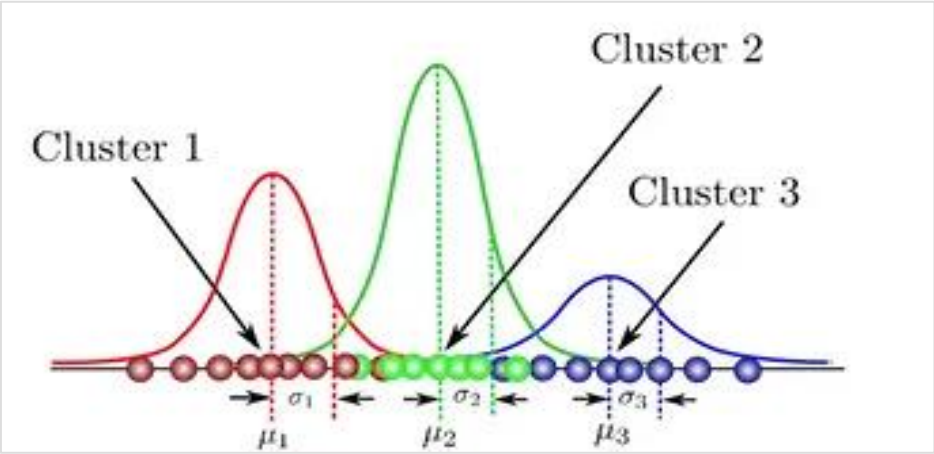


Рисунок 9 — Джерело компонент

Перший крок, відомий як крок очікування, складається з обчислення очікування призначень компонентів для кожної точки даних параметрами моделі.

Другий крок відомий як крок максимізації крок, який складається з максимізації очікувань, обчислених на першому етапі щодо параметрів моделі. Цей крок складається з оновлення значень параметрів моделі.

Весь ітераційний процес повторюється, доки алгоритм не зійдеться, даючи оцінку максимальної правдоподібності.

2.2 Методи візуалізації результатів кластеризації

Візуалізація результатів кластеризації допомагає зрозуміти структуру даних та виявити природні групи чи патерни, які можуть бути важливими для подальшого аналізу. Ось кілька причин, чому вона корисна:

1. Виявлення структури даних: Кластеризація допомагає визначити групи схожих об'єктів у великому наборі даних. Візуалізація надає можливість швидко оглянути ці групи та їхні взаємовідносини.
2. Пошук аномалій: Графічне відображення дозволяє виявити аномалії або об'єкти, які не підпадають під звичайні патерни. Це може бути корисно для виявлення невідомих аномалій чи незвичайних груп.
3. Підтвердження кластеризації: Візуалізація допомагає перевірити правильність вибору методу кластеризації та оцінити, наскільки добре він розділяє дані на групи.
4. Легше сприйняття результатів: Графічне представлення даних дозволяє легше сприймати та розуміти складні взаємозв'язки між групами.
5. Планування подальших дій: Візуалізація надає вам підставу для подальшого аналізу. Ви можете приймати рішення щодо подальших досліджень або вживати заходів на основі отриманих візуальних даних.

Узагальнюючи, візуалізація результатів кластеризації є потужним інструментом для зрозуміння складних структур даних та виявлення важливих патернів.

2.2.1 PCA

Аналіз основних компонентів (PCA) є широко використовуваним методом аналізу даних і зменшення розмірності. Допомагає ідентифікувати найбільш суттєві закономірності та зменшити складність багатовимірних даних, зберігаючи при цьому їх важливу інформацію, що гарантує легку побудову візуалізації. Кроки, пов’язані з виконанням PCA:

1. Збір даних і стандартизація: PCA в першу чергу підходить для числових даних. Якщо дані містять категоричні змінні, знадобиться їх попередня обробка. Стандартизація перетворює дані таким чином, щоб кожна змінна мала середнє значення 0 і стандартне відхилення 1. Це гарантує, що всі змінні знаходяться в одній шкалі.
2. Розрахунок коваріаційної матриці Першим кроком у PCA є обчислення коваріаційної матриці стандартизованих даних. Коваріаційна матриця представляє зв’язки між змінними. Кожен елемент у матриці представляє коваріацію між двома змінними.
3. Знаходження власних значень і власних векторів цієї матриці. Вони мають вирішальне значення для визначення основних компонентів.
4. Сортування та вибір основних компонентів: власні значення представляють величину дисперсії в даних, яку пояснює кожен власний вектор. Щоб зменшити розмірність, відсортуйте власні значення в порядку спадання. Власний вектор, що відповідає найбільшому власному значенню, пояснює найбільшу дисперсію та є першим головним компонентом. Друге за величиною власне значення відповідає другому головному компоненту і так далі.
5. Перетворення даних: щоб зменшити розмірність даних, створення матриці проекції, використовуючи вибрані власні вектори (головні компоненти). Ця матриця представляє перетворення, необхідне для проектування даних у новий простір із зменшеними вимірами. При множення матриці проекції на початкові стандартизовані дані ми отримаємо початкові дані меншої розмірності

6. Інтерпретація та аналіз. Об’єднання остаточних даних меншої розмірності з початковими.

2.2.2 UMAP

Uniform Manifold Approximation and Projection (UMAP) — це метод зменшення розмірності, який можна використовувати для візуалізації, але також для загального нелінійного зменшення розмірності. Алгоритм базується на трьох припущеннях щодо даних: дані рівномірно розподілені на рімановому різноманітті, Riemannian метрика є локально постійною (або може бути апроксимована як така), Колектор підключається локально. З цих припущень можна змоделювати різновид з нечіткою топологічною структурою. Убудованість визначається шляхом пошуку низько вимірної проекції даних, яка має найближчу можливу еквівалентну нечітку топологічну структуру.

2.3 Метрики порівняння кластерних аналізів

2.3.1 Обчислювальна складність алгоритмів кластерного аналізу

Таблиця 1— Обчислювальна складність алгоритмів Кластерного Аналізу

Кластерний аналіз	**Обчислювальна складність (O)**
K-means	$O(n * k * I * d)$
Hierarchical Agglomarative clustering.	$O(n^2)$
DBSCAN	$O(n \log n)$
OPTICS	$O(n^2 \log n)$
BIRCH	$O(n)$
Mean shift	$O(n^2)$
Gaussian Mixture	$O(i \cdot k \cdot n \cdot d)$.

2.3.2 Silhouette Coefficient

Коефіцієнт силуету використовується для визначення якості кластеризації даних. Він вимірює, наскільки чітко визначені кластери в конкретному наборі даних. Коефіцієнт коливається від -1 до 1, де висока величина свідчить про те, що об'єкт добре вписується у свій власний кластер і погано вписується в сусідні кластери.

Простіше кажучи, цей показник допомагає оцінити якість кластеризації, враховуючи те, наскільки близько розташовані точки даних у межах кластера та наскільки далеко різні кластери відокремлені один від одного. Коефіцієнт силуету може бути корисним інструментом при визначенні оптимальної кількості кластерів для ваших даних або порівнянні різних алгоритмів кластеризації.

2.3.3 Calinski-Harabasz Index

Calinski-Harabasz Index — це метричний показник, який використовується для оцінки якості кластеризації даних. Він враховує як внутрішню схожість кластерів, так і відстань між кластерами. Цей індекс шукає ситуацію, коли кластери мають велику внутрішню схожість і одночасно велику відстань один від одного.

Значення індексу Калінські-Харабаса вище свідчать про кращу якість кластеризації. Це може бути корисно при порівнянні різних методів кластеризації або визначенні оптимальної кількості кластерів. Зазвичай, чим вище індекс, тим краще вважається кластеризація за цим показником.

2.3.4 Davies-Bouldin Index

Davies-Bouldin Index — це критерій для оцінки ефективності кластеризації даних. Цей індекс враховує як внутрішній розподіл кластерів, так і розташування різних кластерів у просторі даних.

Ідея полягає в тому, що краща кластеризація вважається тією, в якій кожен кластер має схожість з іншими кластерами, а відстань між кластерами є

максимальною. Індекс Девіса-Болдуїна шукає ситуацію, коли внутрішній розподіл кластерів є великим, а відстань між кластерами є малим.

Зазвичай, низьке значення індексу Девіса-Болдуїна свідчить про кращу кластеризацію. Цей показник може бути корисним для порівняння різних моделей кластеризації або визначення оптимальної кількості кластерів в даних.

2.4 Висновки до розділу 2

В даному розділі було проведено ґрунтовну роботу в напрямку дослідження методів кластеризації. Результати дослідження приведемо у формі таблиці:

Таблиця 2 — *Порівняння алгоритмів кластерного аналізу*

Кластерний Аналіз	**Параметри**	**Масштабованість**	**Оптимальний Спосіб використання**
K-means	число кластерів	Дуже великий n_samples, середня кількість кластерів	Загальне призначення, рівний розмір кластерів, плоска геометрія, не надто багато кластерів, індуктивні
Hierarchical Agglomarative clustering	кількість кластерів або поріг відстані	Великі n_samplesin_clusters	Багато кластерів, можливо обмеження з’єднання
Dbscan	радіус ε (eps) Та мінімальна кількість сусідів (minpts)	Великий n_samples, велика кількість кластерів	Не плоска геометрія, нерівномірні розміри кластерів
OPTICS	мінімальна кількість сусідів (minpts)	Дуже великий n_samples, велика кількість кластерів	Не плоска геометрія, нерівномірні розміри кластерів, змінна щільність кластерів

BIRCH	коефіцієнт розгалуження, поріг, необов’язковий глобальний кластеризатор	Великі n_clustersin_samples	Великий набір даних, видалення викидів, зменшення
Mean-shif	пропускна здатність	Не масштабується з n_samples	Багато кластерів, нерівномірний розмір кластерів, нерівна геометрія, індуктивні
Gaussian Mixture	Кількість кластерів або максимальна відстань	великий n_samples, середня кількість кластерів	Загальне призначення, рівний розмір кластерів, плоска геометрія, не надто багато кластерів,

РОЗДІЛ 3 РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ КЛАСТЕРНОГО АНАЛІЗУ МНОЖИНИ ОБ'ЄКТІВ

3.1 Аналіз даних датасету (множини об'єктів)

Задача кластеризації: розробити сегментацію клієнтів для визначення маркетингової стратегії

Зразок набору даних: узагальнює поведінку використання приблизно 9000 активних власників кредитних карток протягом 6 місяців. Файл на рівні клієнта з 18 поведінковими змінними.

Словник датасету:

- CUST_ID : ідентифікація власника кредитної картки (категорична);
- BALANCE: сума балансу, що залишилася на їхньому рахунку;
- BALANCE_FREQUENCY : як часто баланс оновлюється, оцінка від 0 до 1 (1 = часто оновлюється, 0 = нечасто оновлюється)
- PURCHASES: сума покупок, здійснених з рахунку
- ONEOFF_PURCHASES : максимальна сума одноразової покупки
- INSTALLMENTS_PURCHASES : покупки, здійснена в розстрочку
- CASH_ADVANCE : готівка наперед, надана користувачем
- PURCHASES_FREQ: як часто здійснюються покупки, оцінка від 0 до 1 (1 = часто купують, 0 = нечасто купують)
- ONEOFFPURCHASES: Як часто покупки відбуваються одноразово (1 = часто купують, 0 = не часто купують)
- PURCHASESINSTALL : як часто здійснюються покупки частинами (1 = часто робиться, 0 = не часто)
- CASHADVANCE : Як часто виплачується готівка наперед
- CASHADCETR : кількість транзакцій, здійснених "Готівкою наперед"
- PURCHASES_TRX : кількість здійснених транзакцій покупки
- CREDIT_LIMIT : Ліміт кредитної картки для користувача
- PAYMENTS : сума платежу, здійснена користувачем

- MINIMUM_PAY: мінімальна сума платежів, здійснених користувачем
- PRCFULLPAY: Відсоток повного платежу, сплаченого користувачем
- TENURE : термін користування послугою кредитної картки

3.1.1 Таблиця типів даних властивостей множини об’єктів

Таблиця 3 — *Таблиця типів даних множини об’єктів*

Назва параметру	**Кількість не нульових значень**	**Тип даних**
CUST_ID	8950	object
BALANCE	8950	float64
BALANCE_FREQUENCY	8950	float64
PURCHASES	8950	float64
ONEOFF_PURCHASES	8950	float64
INSTALLMENTS_PURCHASES	8950	float64
CASH_ADVANCE	8950	float64
PURCHASES_FREQ	8950	float64
ONEOFFPURCHASES	8950	float64
PURCHASESINSTALL	8950	float64
CASHADVANCEF	8950	float64
CASHADCETR	8950	int64
PURCHASES_TRX	8950	int64
CREDIT_LIMIT	8949	float64
PAYMENTS	8950	float64

MINIMUM_PAY	8637	float64
PRCFULLPAY	8950	float64
TENURE	8950	int64

3.1.2 Попередня обробка даних за допомогою Pandas

Pandas — це швидкий, потужний, гнучкий і простий у використанні інструмент аналізу та обробки даних з відкритим кодом, побудований на основі мови програмування Python.

Для виконання попередньої обробки були виконані такі кроки:

1. Импорт Pandas: `import pandas as pd`
2. Ініціалізація датасету: `url = '/CC GENERAL.csv' data = pd.read_csv(url)`
3. Аналіз частки нульових значень у наборі даних за допомогою команди: `data.isna().mean()*100` Отримано таку частку нульових значень: CREDIT_LIMIT=0.011173, MINIMUM_PAYMENTS=3.497207.
4. Видалення нульових параметрів*:* `data.loc [(data ['MINIMUM _PAYMENTS'].isnull()==True),'MINIMUM_PAYMENTS']= data['MINIMUM_PAYMENTS'].mean()data.loc[(data['CREDIT_LIMIT'].isnull()==True),'CREDIT_LIMIT']=data['CREDIT_LIMIT'].mean()`
5. Видалення одного заголовку, який не є необхідним для сегментації клієнтів; CUST_ID: `data= data.drop(columns=['CUST_ID'])`

3.1.3 Масштабування даних

Масштабування даних часто використовується для забезпечення ефективнішої роботи алгоритмів машинного навчання. Зокрема, багато алгоритмів можуть давати оманливі результати через різні масштаби функцій. Тому важливо привести всі функції в однаковий масштаб.

Використаний Метод масштабування — StandardScaler. StandardScaler

масштабує кожну функцію, щоб отримати середнє значення 0 і стандартне відхилення 1. За допомогую `scaler = StandardScaler() data_scaled = scaler.fit_transform(data)` дані успішно масштабовано та готові для подальшого аналізу.

3.1.4 Статистика Хопкінса

Перед початком аналізу кластеризації важливо перевірити, чи підходить набір даних для кластеризації. Для цього використовується статистика Хопкінса, щоб перевірити, чи набір даних випадково розподілений.

Статистика Хопкінса — це тест, який визначає, наскільки набір даних підходить для кластеризації. Значення, близьке до 0,5, вказує на те, що дані мають випадковий розподіл і тому не підходять для кластеризації. З іншого боку, якщо значення більше 0,7, набір даних, ймовірно, підходить для кластеризації.

Лістинг *1 — Обрахунок Hopkins Statistic*

```
import numpy as np
from sklearn.neighbors import NearestNeighbors
from random import sample

def hopkins(X):
     d = X.shape[1]
     #d = len(vars) # columns
     n = len(X) # rows
     m = int(0.1 * n) # heuristic from article [1]
     nbrs = NearestNeighbors(n_neighbors=1).fit(X)
     rand_X = sample(range(0, n, 1), m)
     ujd = []
     wjd = []
     for j in range(0, m):
          u_dist,_=nbrs.kneighbors(np.random.uniform(np.amin
     (X,axis=0),np.amax(X,axis=0),d).reshape(1, -1), 2,
          return_distance=True)
          ujd.append(u_dist[0][1])
      w_dist, _ = nbrs.kneighbors(X[rand_X[j] ].reshape(1, -
     1), 2, return_distance=True)
          wjd.append(w_dist[0][1])
     H = sum(ujd) / (sum(ujd) + sum(wjd))
     if np.isnan(H):
```

```
            print(ujd, wjd)
            H = 0
        return H
hopkins_score = hopkins(df_scaled)
print(hopkins_score)
```

Після виконання Лістингу 1 отримуємо `hopkins_score = 0.9669420744514984` Це свідчить про те що дані підходять для кластеризації.

3.2 Розробка інтерфейсу комп'ютерної системи

Для відображення результатів кожного з алгоритмів кластерного аналізу побудовано інтерфейс за допомогою бібліотеки customtkinter.

CustomTkinter — це бібліотека Python на основі Tkinter, яка надає нові, сучасні та повністю настроювані віджети. Вони створюються та використовуються як звичайні віджети Tkinter, а також можуть використовуватися в поєднанні зі звичайними елементами Tkinter. Кольори віджетів і вікон адаптуються до зовнішнього вигляду системи або встановленого вручну режиму («світлий», «темний»), і всі віджети та вікна CustomTkinter підтримують масштабування HighDPI (Windows, macOS). З CustomTkinter ви отримаєте послідовний і сучасний вигляд на всіх платформах ПК.

Лістинг *2 Побудова інтерфейсу*

```
Class App(customtkinter.CTk):
    def __init__(self):
     super().__init__()
     self.Algoritm=""
     self.title("CustomTkinter complex_example.py")
     self.attributes("-fullscreen", True)
     customtkinter.set_appearance_mode("Dark")
     # configure grid layout (4x4)
     self.grid_columnconfigure(1, weight=1)
     self.grid_columnconfigure((2, 3), weight=0)
     self.grid_rowconfigure((0, 1, 2), weight=1# create
    sidebar frame with widgets
     self.sidebar_frame = customtkinter.CTkFrame(self,
    width=140, corner_radius=0)
```

```
 self.sidebar_frame.grid(row=0, column=0, rowspan=3,
sticky="nsew")
 self.sidebar_frame.grid_rowconfigure(7, weight=1)
 self.scrollable_frame = customtkinter.CTkFrame(self)
 self.scrollable_frame.grid(row=0, column=0, padx=(20,
0), pady=(20, 0), sticky="nsew")
 self.scrollable_frame.grid_columnconfigure(0,
weight=1)
 self.scrollable_frame_switches = []
 self.scaling_label =
customtkinter.CTkLabel(self.scrollable_frame,
text="Кластерні Аналізи:",anchor="w",
font=("Arial",15), padx=10 )
 self.scaling_label.grid(row=0, column=0, padx=20,
pady=(10, 0))
 self.switch_var_k_means =
customtkinter.StringVar(value=1)
 switch1 =
customtkinter.CTkSwitch(master=self.scrollable_frame,
text=f"K-means",command=self.switch_event_k_means,
 variable=self.switch_var_k_means, onvalue=1,
offvalue=0)
 customtkinter.CTkSwitch(master=self.scrollable_frame,
text=f"Dbscan",command=self.switch_event_k_means,
 variable=self.switch_var_k_means, onvalue=1,
offvalue=0)
customtkinter.CTkSwitch(master=self.scrollable_frame,
text=f"OPtiCS",command=self.switch_event_k_means,
 variable=self.switch_var_k_means, onvalue=1,
offvalue=0)
 self.sidebar_button_1 =
customtkinter.CTkButton(self.sidebar_frame,
command=self.Calculate, text="Обчислити")
 self.sidebar_button_1.grid(row=9, column=0, padx=20,
pady=10)
 self.sidebar_button_2 =
customtkinter.CTkButton(self.sidebar_frame,
command=self.Calculate, text="Висновки")
 self.sidebar_button_2.grid(row=10, column=0, padx=20,
pady=10)
 self.appearance_mode_label =
customtkinter.CTkLabel(self.sidebar_frame,
text="Appearance Mode:", anchor="w")
 self.appearance_mode_label.grid(row=11, column=0,
padx=20, pady=(10, 0))
 self.appearance_mode_optionemenu =
customtkinter.CTkOptionMenu(self.sidebar_frame,
 values=["Dark", "Light", "System"],
 command=self.change_appearance_mode_event)
```

```
 self.appearance_mode_optionemenu.grid(row=11,
column=0, padx=20, pady=(10, 10))
 # create tabbox
 self.tabview = customtkinter.CTkTabview(self)
 self.tabview.grid(row=0, column=1,
sticky="nsew",rowspan=999
```

За допомогою лістингу 2 побудований інтерфейс який складається з таких єлементів як:

1. Боковий фрейм (`CTkFrame`) в якому знаходяться 7 елементів (`CTkSwitch`)
2. Два об’єкта (`CTkButton`) для початку роботи алгоритмів та побудови висновків
3. Лист `tab` (`CTkTabview`) для відображення результатів роботи методів кластерного аналізу

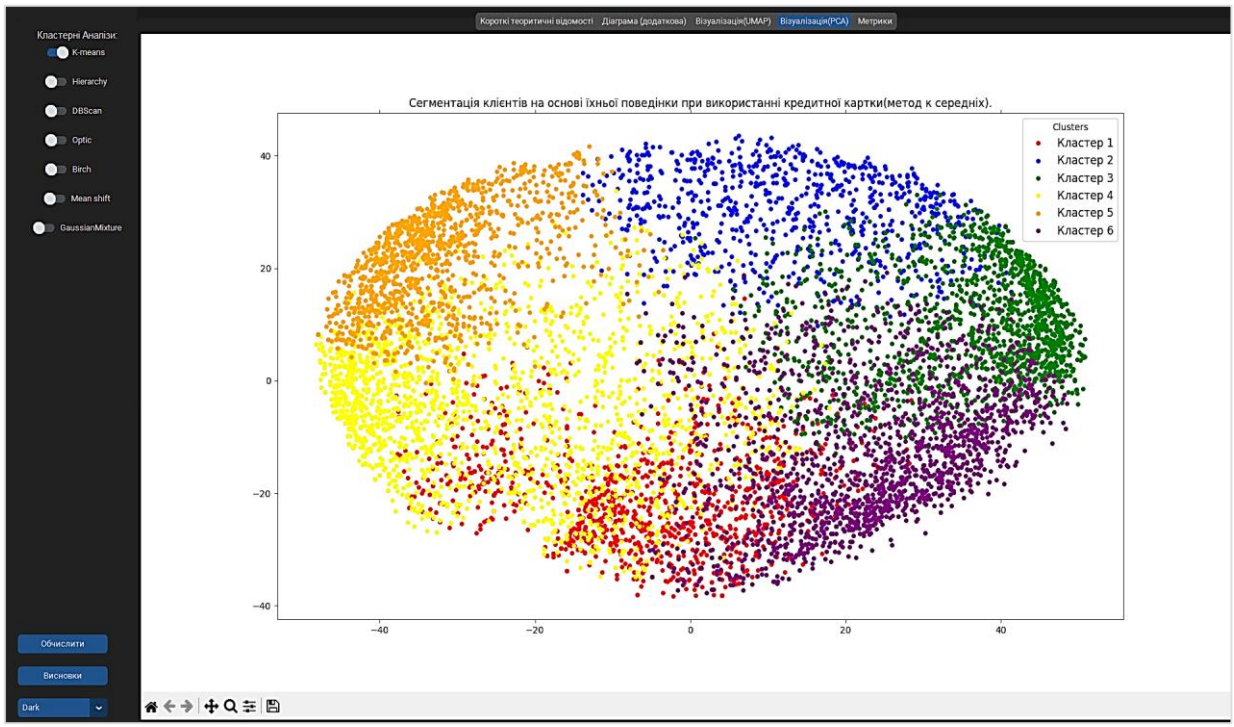


Рисунок 10 — Інтерфейс комп'ютерної системи

3.3 Дослідження реалізації методів кластерного аналізу на множині об’єктів

Розглянемо реалізації моделей методів кластерного аналізу представлені в бібліотеці scikit-learn, та параметри ініціалізації і атрибути цих класів.

Таблиця 4 — *Порівняння алогоритмів кластерного аналізу*

Модель кластерного аналізу	**Параметри та атрибути**
K-means (`sklearn.metrics.Kmean()`)	**Параметр: n_clusters**: int, за замовчуванням = 8 Кількість кластерів, які необхідно сформувати, а також кількість центроїдів для створення. **Параметр: n_init**: int, за замовчуванням = 10 Кількість разів, коли алгоритм k-mean буде виконуватися з різними насіння центроїдів. Кінцеві результати будуть найкращими результатами n_init послідовних прогонів з точки зору інерції. **Параметр: max_iter**: int, за замовчуванням = 300 Максимальна кількість ітерацій алгоритму k-mean за один пробіг. **Параметр: tol** :float, за замовчуванням = 1e-4 Відносна толерантність **Параметр: random_state** :int, екземпляр RandomState або None, за замовчуванням = None Визначає генерацію випадкових чисел для ініціалізації центроїда. **Параметр: algo**: {“auto”, “full”, “elkan”}, default = “auto” Алгоритм K-означає. Варіація "elkan" є більш ефективною для даних з чітко визначеними кластерами, використовуючи нерівність трикутника **Атрибут: labels_** ndarray of shape (n_samples,) Мітки кожної точки

	Атрибут: inerthea_ Сума квадратів відстаней зразків до їх найближчого центру кластеру **Атрибут: n_iter_** int Кількість запущених ітерацій.
Hierarchical Agglomarative clustering (sklearn.metrics.Agglomarativeclustering())	**Параметр: n_clusters** :int або None, за замовчуванням = 2 Кількість кластерів для пошуку. Має бути, None якщо distance_thresholdні None. **Параметр: afinity**: str або callable, default = 'euclidean' Метрика, що використовується для обчислення зв'язку. Може бути "євклідовим", "l1", "l2", "манхеттенським", "косинусом" або "попередньо обчисленим". Якщо зв'язок "палата", приймається лише "евклідова". Якщо "попередньо обчислюється", матриця відстані (замість матриці подібності) потрібна як вхід для методу fit. **Параметр: distance_threshold** float, за замовчуванням = немає Поріг відстані зв’язку, вище якого кластери не об’єднаються. **Атрибут: n_clusters_** :int Кількість кластерів, знайдених алгоритмом. **Атрибут: labels_** (n_samples) Кластерні мітки для кожної точки. **Атрибут: distances_**:подібні до масиву форми (n_nodes-1,) Відстані між вузлами у відповідному місці
Dbscan (sklearn.metrics.Dbscan())	**Параметр: eps** float, за замовчуванням = 0,5 Максимальну відстань між двома зразками для одного вважатиметься як сусідство іншого. Це не максимальна межа на відстанях точок у кластері.. **Параметр: min_samples** int, за замовчуванням = 5 Кількість вибірок (або загальна вага) в околиці для точки, що вважається основною точкою. Сюди входить і сама точка. **Атрибут: labels_** (n_samples) Кластерні мітки для кожної точки в наборі даних, надані fit (). Шумним зразкам надається мітка -1.

	Атрибут: n_clusters_ int Кількість кластерів, знайдених алгоритмом.
OPTICS (sklearn.metrics.OPTICS())	**Параметр: max_eps** float Максимальна відстань між двома зразками для одного вважати як сусідом іншого. **Параметр: min_samples** int> 1 або float між 0 і 1, за замовчуванням = 5 Кількість вибірок у околиці для точки, яка буде вважатися основною точкою. **Параметр: min_cluster_size** int Мінімальна кількість вибірок у кластері OPTICS, виражена як абсолютне число або частка від кількості зразків **Атрибут: labels_** ndarray of shape (n_samples,) Кластерні мітки для кожної точки в наборі даних, надані fit (). Шумні зразки та точки, які не входять до групи листя cluster_hierarchy_, позначаються як -1. **Атрибут: n_clusters_** int Кількість кластерів, знайдених алгоритмом.
BIRCH (sklearn.metrics.BIRCH())	**Параметр: n_clusters**: int, екземпляр моделі sklearn.cluster, за замовчуванням = 3 Кількість кластерів після останнього етапу кластеризації, який розглядає субкластери з листків як нові зразки. **Параметр: Branching_factor**: int, за замовчуванням = 50 Максимальна кількість субкластерів CF у кожному вузлі. **Параметр: compute_labels**: bool, за замовчуванням = True Обчислювати мітки для кожної відповідності чи ні. **Атрибут: subcluster_centers_**: ndarray Центроїди всіх підгруп читаються безпосередньо з листя. subcluster_labels **Атрибут: labels_** :(n_samples,) Масив міток, призначених для вхідних даних. якщо замість fit. **Атрибут: feature_names_in_** : (n_features_in_,) Назви ознак, які можна побачити під час підгонки.

	Визначається лише тоді, коли всі назви функцій є рядками.
Meanshift (`sklearn.metrics.Meanshift ()`)	**Параметр: bin_seeding**: bool, за замовчуванням = False Якщо True, початкові розташування ядра - це не розташування всіх точок, а скоріше місце розташування дискретної версії точок, де точки об'єднані в сітку, грубість якої відповідає пропускній здатності. Встановлення цієї опції на True прискорить роботу алгоритму, оскільки буде ініціалізовано менше насіння. **Параметр: min_bin_freq**: int, за замовчуванням = 1 Щоб прискорити роботу алгоритму, приймайте лише ті бункери, які мають принаймні точки min_bin_freq як насіння. **Параметр: max_iter**: int, за замовчуванням = 300 Максимальна кількість ітерацій на кожну точку початку до завершення операції кластеризації (для цієї точки виходу) **Атрибут: labels_** :ndarray of shape (n_samples,) Мітки кожної точки. **Атрибут: n_iter_**: int Максимальна кількість ітерацій, виконаних для кожного насіння.
Gaussian Mixture (`sklearn.mixture.gaussianmixture()`)	**Параметр: n_componentsint**, int Кількість компонентів суміші. **Параметр: random_stateint,**: екземпляр RandomState або None, за замовчуванням = None Визначає генерацію випадкових чисел для ініціалізації центроїда **Параметр: max_iter:** int, за замовчуванням=100 Кількість EM-ітерацій, які потрібно виконати. **Параметр: n_init:** int, за замовчуванням=1 Кількість ініціалізацій для виконання. Найкращі результати зберігаються. **Атрибут: feature_names_in_** :ndarray of shape (n_samples,) Мітки кожної точки. **Атрибут: n_features_in_**: int Кількість функцій, які видно під час підгонки.

3.4 Візуалізація результатів методів кластерного аналізу на заданій множині об’єктів

3.4.1 Реалізація методу побудови візуалізації результатів кластеризації за допомогою Umap

Для реалізація методу побудови візуалізації результатів кластеризації за допомогою Umap були виконані наступні кроки:

1. Вибір вхідних даних: data - це масштабовані дані labels - результати кластерного аналізу
2. Побудова пустої фігури `plt.figure(figsize=(10, 8))`
3. Побудова назви та легенди фігури
4. Знаходження кількості унікальних ідентифікаторів кластерів в результаті кластеризації `unique_labels = np.unique(labels)`
5. Побудова списку кольорів згідно кількості унікальних ідентифікаторів кластерів
6. Додавання кожного елемента за кольором згідно номера кластера до якого належить цей елемент.

Результати представленні в Лістингу 4

Лістинг 4 — Реалізація методу побудови візуалізації за допомогою Umap

```
def Umapviz(data,labels):
    plt.figure(figsize=(10, 8))
    reducer = umap.UMAP()
    embedding = reducer.fit_transform(data)
    df_umap = pd.DataFrame(embedding, columns=['UMAP 1',
    'UMAP 2'])
    plt.title('Clustering')
    plt.xlabel('UMAP 1')
    plt.ylabel('UMAP 2')
    unique_labels = np.unique(labels)
    colors = [plt.get_cmap('Spectral')(each) for each in
    np.linspace(0, 1, len(unique_labels))]

    for k, col in zip(unique_labels, colors):
        if k == -1:
            col = [0.6, 0.6, 0.6, 1]
```

```
        class_member_mask = (labels == k)
        xy = df_umap.values[class_member_mask]
        plt.plot(xy[:, 0], xy[:, 1], 'o',
    markerfacecolor=tuple(col), markeredgecolor='k',
    markersize=6)
    fig1 = plt.gcf()
    return fig1
```

3.4.2 Реалізація методу побудови візуалізації результатів кластеризації за допомогою PCA

Для Реалізація методу побудови візуалізації результатів кластеризації за допомогою PCA були виконані наступні кроки:

1. Вибір вхідних даних: data - це масштабовані дані labels - результати кластерного аналізу
2. Побудова пустої фігури `plt.figure(figsize=(10, 8))`
3. Знаходження кількості унікальних ідентифікаторів кластерів в результаті кластеризації `unique_labels = np.unique(labels)`
4. Побудова списку кольорів згідно кількості унікальних ідентифікаторів кластерів
5. Додавання кожного елемента за кольором згідно номера кластера до якого належить цей елемент.
6. Побудова назви та легенди фігури

Результати представленні в Лістингу 5.

Лістинг 5 — Реалізація методу побудови візуалізації з PCA даними

```
def Pcaviz(data,labels):
    unique_labels = np.unique(labels)
    colors = [plt.get_cmap('Spectral')(each) for each in
    np.linspace(0, 1, len(unique_labels))]
    names = {0: 'Кластер 1'}
    n = 1
    while n < len(unique_labels):
        names[n] = f"Кластер {n+1}"
        n = n+1
    df = pd.DataFrame({'x': x, 'y': y, 'label': labels})
    groups = df.groupby('label')
    fig, ax = plt.subplots()
```

```
for name, group in groups:
    if name != -1:
        ax.plot(group.x, group.y, marker='o',
linestyle='', ms=5,
                color=colors[name], label=names[name],
mec='none')
    else:
        ax.plot(group.x, group.y, marker='o',
linestyle='', ms=5,
                color="grey", mec='none')
    ax.set_aspect('auto')
    ax.tick_params(axis='x', which='both',
bottom='off', top='off', labelbottom='off')
    ax.tick_params(axis='y', which='both', left='off',
top='off', labelleft='off')

silhouette_avg = metrics.silhouette_score(X, labels)
ax.text(100, 100, f"Silhouette Score:
{silhouette_avg}", fontsize=100,
transform=ax.transAxes)

# Move the legend to accommodate the silhouette score
legend = ax.legend(loc='upper right', title='Clusters',
fontsize=12)
legend.set_draggable(True)

# Get the legend handles and labels
handles, labels = ax.get_legend_handles_labels()

# Add the silhouette score to the legend
handles.append(ax.texts[0])
labels.append('Silhouette Score')

# Update the legend
ax.legend(handles, labels, loc='upper right',
title='Clusters', fontsize=12)

ax.set_title(
    "Сегментація клієнтів на основі їхньої поведінки
при використанні кредитної картки.")
return fig
```

3.4.3 Реалізація обрахунку показників та метрик кластеризації

Обрахунок таких метрик як Calinski-Harabasz Index, показник Davies-Bouldin Index та Silhouette Coefficient виконується за допомогою бібліотеки sklearn.metrics на вхід приймається масштабований датасет та результати

кластеризації (Атрибут класу кластерного: labels_) отримуємо показник типу float

Обрахунок часу роботи алгоритму виконується за 3 кроки з використанням бібліотеки time.

1. `start_time = time.time()` Створюємо зміну якій присвоюємо значення поточного часу
2. Метод час виконання якого вимірюється.
3. Обрахунок часу виконання `elapsed_time = time.time() - start_time.`

3.5 Висновки до розділу 3

1. Був проведений аналіз даних датасету множини об'єктів, дані були масштабовані та перевіренні на можливість проведення кластерного аналізу
2. Був реалізований інтерфейс системи для виводу результатів роботи кластерних аналізів за допомогою CustomTkinter.
3. Було розглянуто та імплементовано в систему спосіб реалізації алгоритмів за допомогою бібліотеки scikit-learn.
4. Були розроблені методи побудови візуалізація результатів кластеризації з використанням двох методів представлення даних Umap та PCA.
5. Були розібрані методи обрахунку метрик кластеризацій.
6. За рахунок отриманих знань була побудована коп’ютерна система, що виконує кожний з наведених кластерних аналізів та візуалізує отримані результати кластеризацій.

РОЗДІЛ 4 ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ РОБОТИ СИСТЕМИ НАД ЗАДАЧЕЮ КЛАСТЕРНОГО АНАЛІЗУ МНОЖИНИ ОБ’ЄКТІВ

4.1 Розв’язок задачі на основі методу k-means

4.1.1 Пошук оптимального значення кількості кластерів

Для пошуку оптимального значення кількості кластерів побудовано діаграму ліктя яка представляє собою залежність показника інерції розбиття від кількості кластерів (Рис. 11). В Лістингу 3 зазначено метод побудови діаграма ліктя.

Лістинг *3 — Код для побудови діаграми ліктя*

```
def elbow (data_scaled):
    n_clusters = 30
    cost = []
    for i in range(1, n_clusters):
        kmean = KMeans(i)
        kmean.fit(data_scaled)
        cost.append(kmean.inertia_)
    fig1 = Figure()
    ax1 = fig1.add_subplot()
    ax1.plot(cost, 'bx-')
    return fig1
```

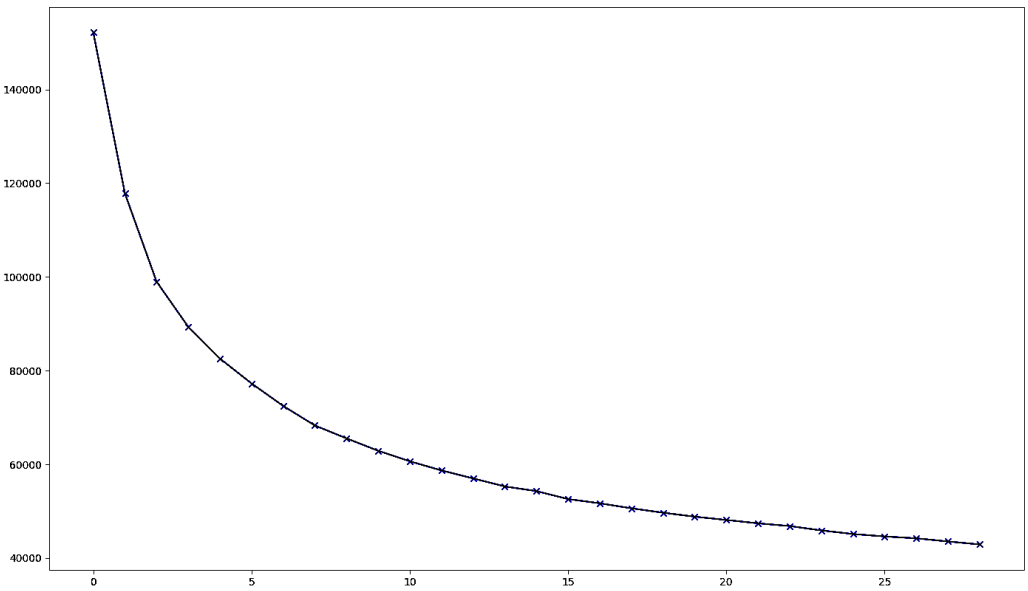


Рисунок *11 — Діаграма ліктя (K-means)*

З діаграми ліктя визначено що ліктьовою точкою для заданого датасету є n=6. Тобто оптимальна кількість кластерів для методу K-means-6.

4.1.2 Побудова моделі K–means з масштабованими даними

Побудована модель кластерного аналіза методом K-means за допомогою: `sklearn.metrics.Kmean(6, random_state=123).`

Впроваджено масштабовані дані множини об’єктів в модель: `Sklearn.metrics.Kmean(6, random_state=123).fit(data)`

Підраховані метрики кластеризації K-means даних датасету. Результати приведенні в таблиці 5.

4.1.3 Побудова моделі K–means з PCA даними.

Перетворено дані множини об’єктів за допомогою методу PCA: `X_PCA = PCA(2).transform(data).`

Впроваджено перетворені дані в модель: `Sklearn.metrics.Kmean (6, random_state=123).fit(X_PCA).`

Підраховані метрики кластеризації K-means з перетворенням даних датасету методом PCA. Результати приведенні в таблиці 5.

Візуалізуємо результати кластеризації (з PCA перетворенням):

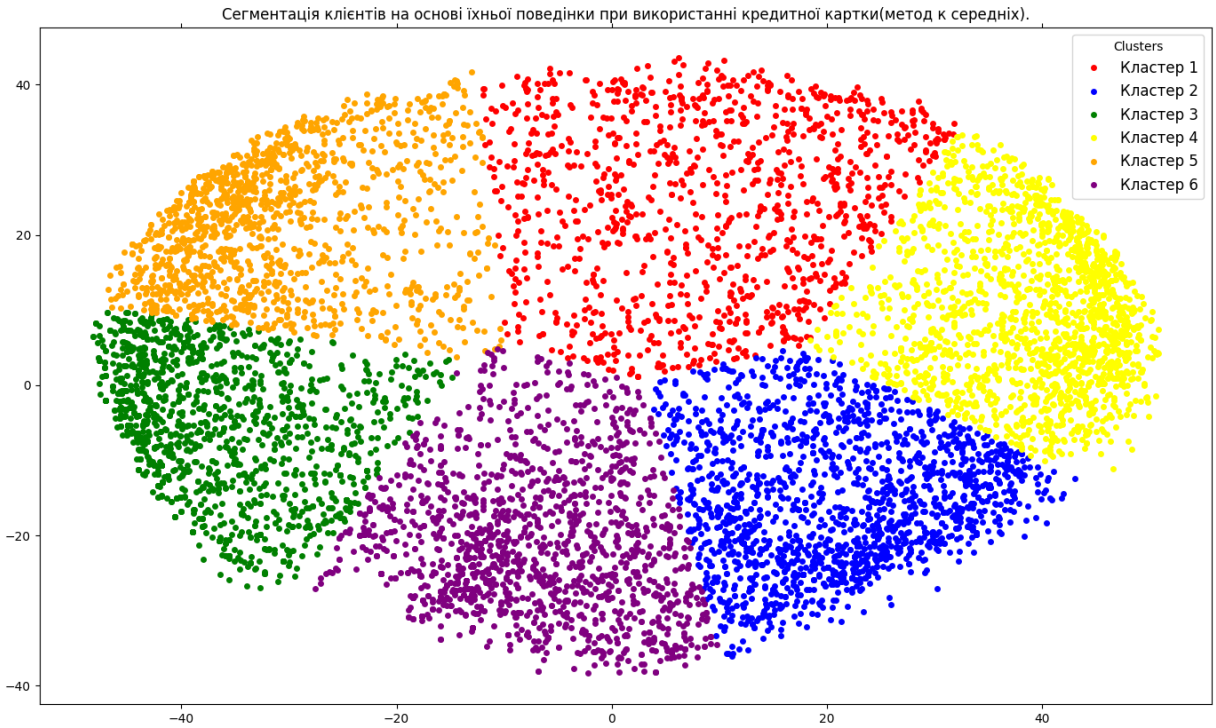


Рисунок 12 — Візуалізація кластеризації з методом PCA (K-means)

4.1.4 Побудова моделі K–means з UMAP даними.

Перетворено дані множини об’єктів за допомогою методу UMAP: `embedding = reducer.fit_transform(data)df_umap = pd.DataFrame(embedding, columns=['UMAP 1', 'UMAP 2'])`

Впроваджено перетворені дані в модель: `sklearn.metrics.Kmean( 6, random_state=123).fit(df_umap)`

Підраховані метрики кластеризації K-means з перетворенням даних датасету методом UMAP. Результати приведенні в таблиці 5

Візуалізуємо результати кластеризації (з UMAP перетворенням):

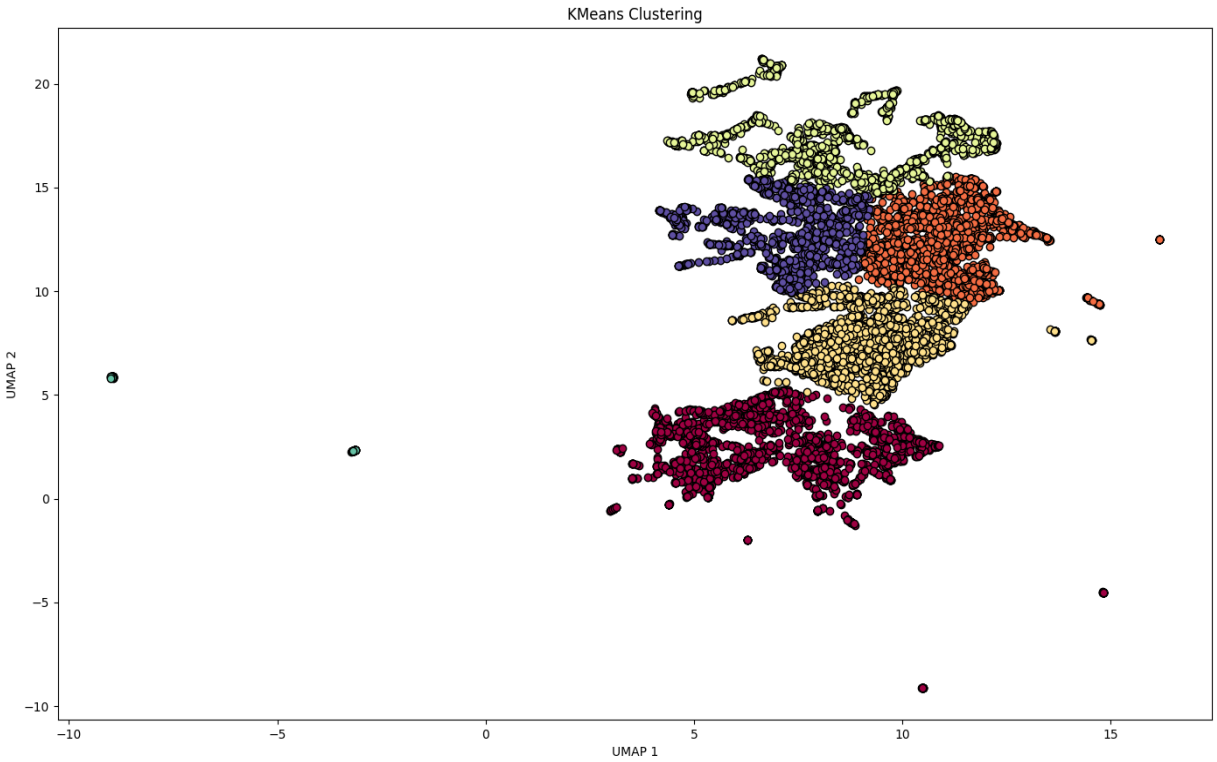


Рисунок 13 — Візуалізація з методом UMAP (K-means)

4.1.5 Метрики результатів кластеризації методом K-means.

Таблиця *4 — Метрики кластеризації k-means*

Метрики	Масштабовані дані	PCA	UMAP
Час виконання	0.365	0.11384	0.12451
Davies-Bouldin Index:	1.65	0.77225	1.48

Calinski-Harabasz Index:	1733	12135	3172
Silhouette Coefficient:	0.179	0.41	0.066

4.1.6 Аналіз отриманих метрик

Аналіз отриманих метрик для масштабованих даних показав:

1. **Silhouette Coefficient:** Оцінка 0,179 свідчить про те, що кластери певною мірою перекриваються, і є місце для вдосконалення.
2. **Calinski-Harabasz Index:** Оцінка 1733 значення вказують на те, що кластери щільні та добре розділені.
3. **Davies-Bouldin Index:** Оцінка 1,65 вказує на помірну структуру кластеризації.

Аналіз отриманих метрик для PCA даних показав:

1. **Silhouette Coefficient:** Оцінка 0,41 покращення показує краще визначені кластери порівняно зі масштабованими даними після застосування PCA.
2. **Calinski-Harabasz Index:** Оцінка 12135 значне збільшення вказує на те, що PCA допоміг у покращенні структури.
3. **Davies-Bouldin Index:** Оцінка 0.77225 зменшення вказує на краще розділення кластерів після застосування PCA.

Аналіз отриманих метрик для UMAP даних показав:

1. **Silhouette Coefficient:** Оцінка 0.066 показує велике погіршення кластеризації після застосування UMAP, але ця помилка виникла через випадковий вибір початкових вершин розбиття методу K-means як ми можемо побачити на рис. 13 э група об'єктів котрі знаходяться далеко від основної маси об’єктів і одна з випадкових початкових вершин потрапила в ту групу через це кластеризація ну була ефективна, тобто це не є недоліком UMAP це недолік методу кластеризації K-means.

2. **Calinski-Harabasz Index:** Оцінка 3172 збільшення вказує на те, що UMAP допоміг у покращенні структури.
3. **Davies-Bouldin Index:** Оцінка 1.48 зменшення вказує на краще розділення кластерів після застосування UMAP.

4.1.7 Висновки розв’язку задачі на основі методу k-means

Виконано розбиття об’єктів на кластери та підраховані метрики. Оцінка метрик показала непогані результати роботи методу насамперед з використанням методу обробки даних PCA. Виявлено та досліджено за допомогою візуалізації недолік роботи методу k-means.

4.2 Розв’язок задачі методом Agglomarative clustering

4.2.1 Побудова дендрограми

Наявним способом дослідити роботу ієрархічного алгоритму є побудована дендрограмма (Рис. 14). В Лістингу 4 зазначено метод побудови дендрограмми.

Лістинг 4 — Метод для побудови дендрограми

```
def dendro(data):
    linkage_matrix = linkage(data, method='ward')
    # Plot the dendrogram
    plt.figure(figsize=(10, 6))
    dendrogram(linkage_matrix)
    plt.title('Dendrogram')
    plt.xlabel('Samples')
    plt.ylabel('Distance')
    fig1 = plt.gcf()
    return fig1
```

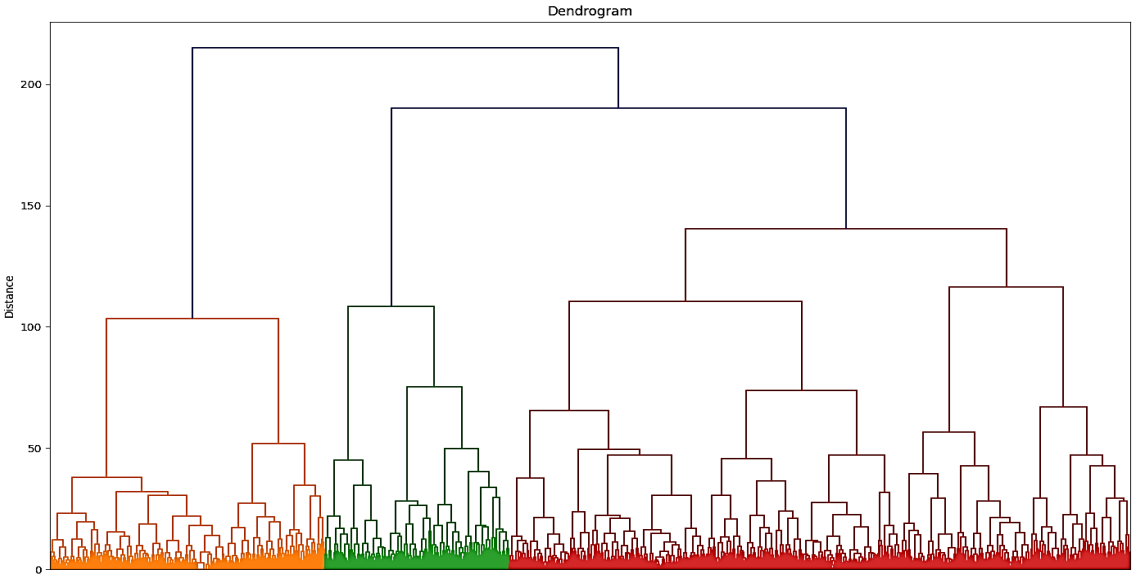


Рисунок 14 — Дендрограма

На дендрогаммі зображено злиття з одиничних елементів до одного котрий має у собі всі елементи. Обране оптимальне значення кількості кластерів 6.

4.2.2 Візуалізації результатів метода Agglomarative clustering

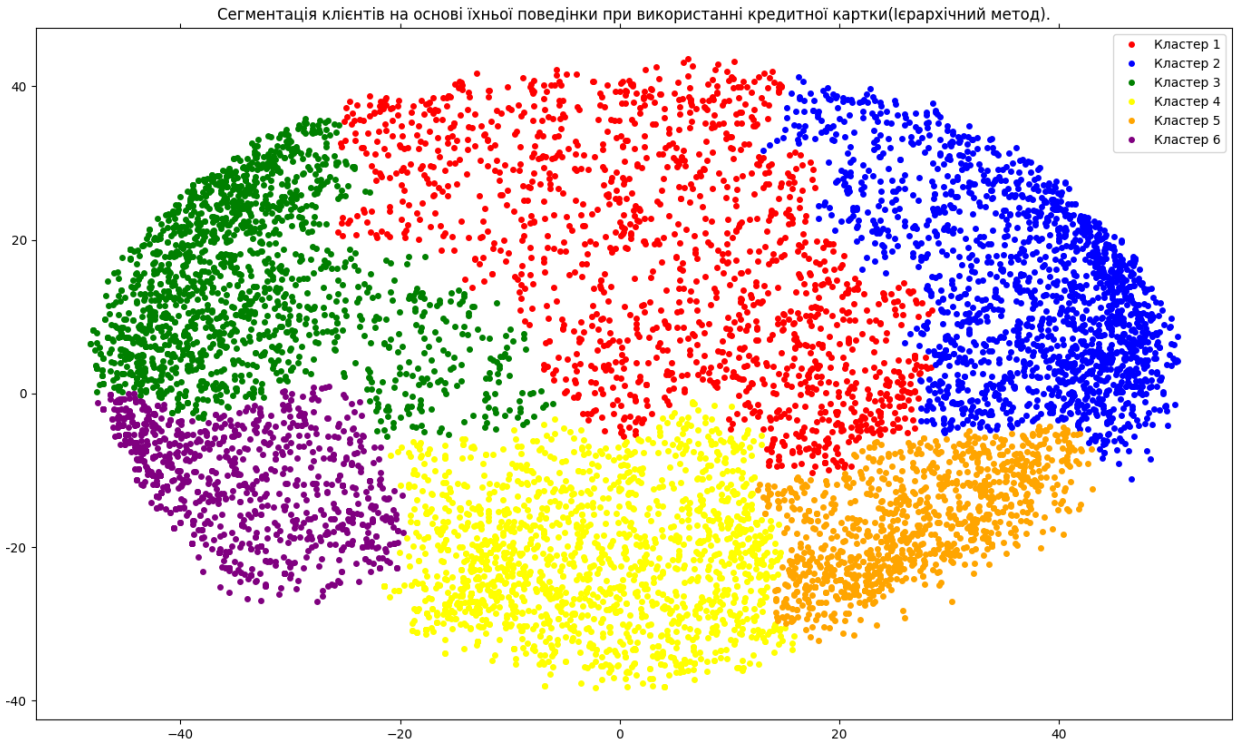


Рисунок 15 — Візуалізація Agglomarative clustering з PCA

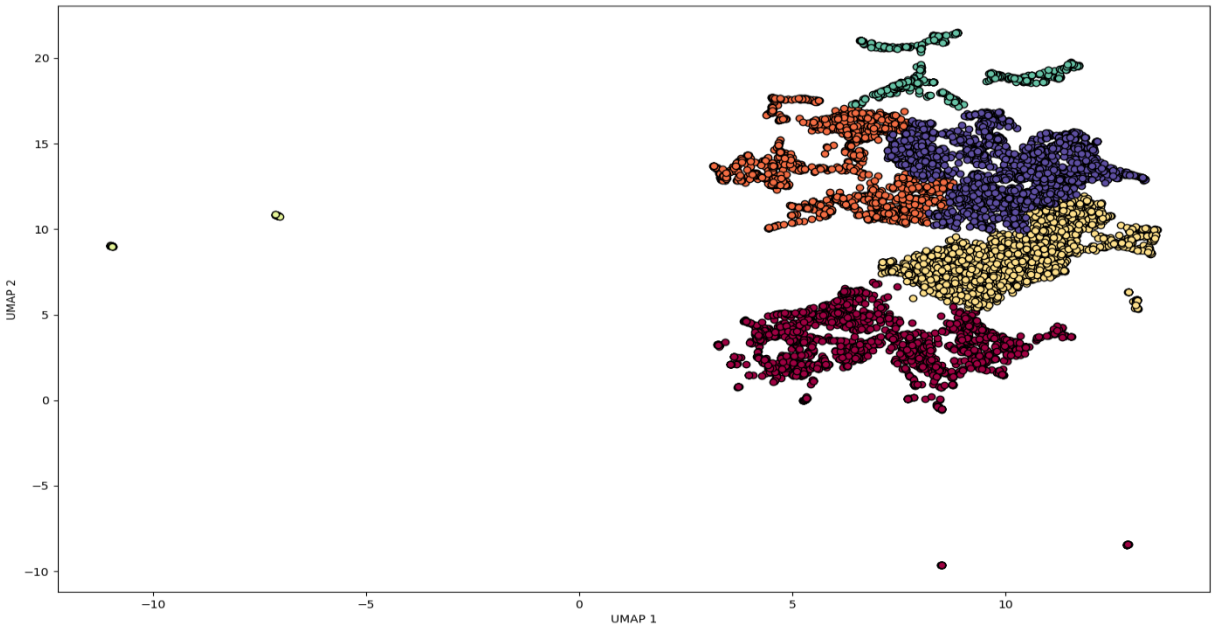


Рисунок 16 — Візуалізація Agglomarative clustering з UMAP

4.2.3 Обраховані метрики результатів кластеризації методом Agglomarative clustering

Таблиця 5 — Метрики кластеризації Agglomarative clustering

Метрики	Масштабовані дані	PCA	UMAP
Час виконання	6.26	1.15	1.26
Davies-Bouldin Index:	1.81	0.853	0.78
Calinski-Harabasz Index:	1296	9280	8680
Silhouette Coefficient:	0.1411	0.34	0.39

4.2.4 Аналіз отриманих метрик

Аналіз отриманих метрик для масштабованих даних показав: низькі показники що свідчить про те, що кластери певною мірою перекриваються, але вони досить щільні.

Аналіз отриманих метрик для PCA даних показав: дуже велику щільність кластерів та SC 0.34 свідчить про те що кластери не перекриваються результати кластеризації є задовільними.

Аналіз отриманих метрик для UMAP даних показав: найбільший з показників SC шо свідчить про найкраще розбиття серед представлених методів представлення даних.

4.2.5 Висновки розв’язку задачі на основі методу Agglomarative clustering

Виконано розбиття об’єктів на кластери методом Agglomarative clustering та підраховані метрики. Оцінка метрик показала найкращі результати з використанням методу представлення даних UMAP. Бачимо більшу стабільність роботи ніж у методі K-means, але треба зауважити час виконання дуже великий.

4.3 Розв’язок задачі методом DBSCAN

4.3.1 Знаходження EPS

Для пошуку оптимального значення EPS побудовано діаграму ліктя метрики k-відстаней. В Лістингу 5 зазначено метод побудови діаграма ліктя.

Лістинг 5 — Метод для побудови діаграми k-відстаней

```
def epsilon(X):
    neighbors = NearestNeighbors(n_neighbors=2)
    nbrs = neighbors.fit(X)
    distances, indices = nbrs.kneighbors(X)
    distances = np.sort(distances, axis=0)
    distances_1 = distances[:, 1]
    plt.plot(distances_1, color='#5829A7')
    plt.xlabel('Total')
    plt.ylabel('Distance')
    for spine in plt.gca().spines.values():
        spine.set_color('None')
    plt.grid(axis='y', alpha=0.5, color='#9B9A9C',
linestyle='dotted')
```

```
    plt.grid(axis='x', alpha=0)

    plt.title('DBSCAN Epsilon Value for Scaled Data')
    plt.tight_layout()
```

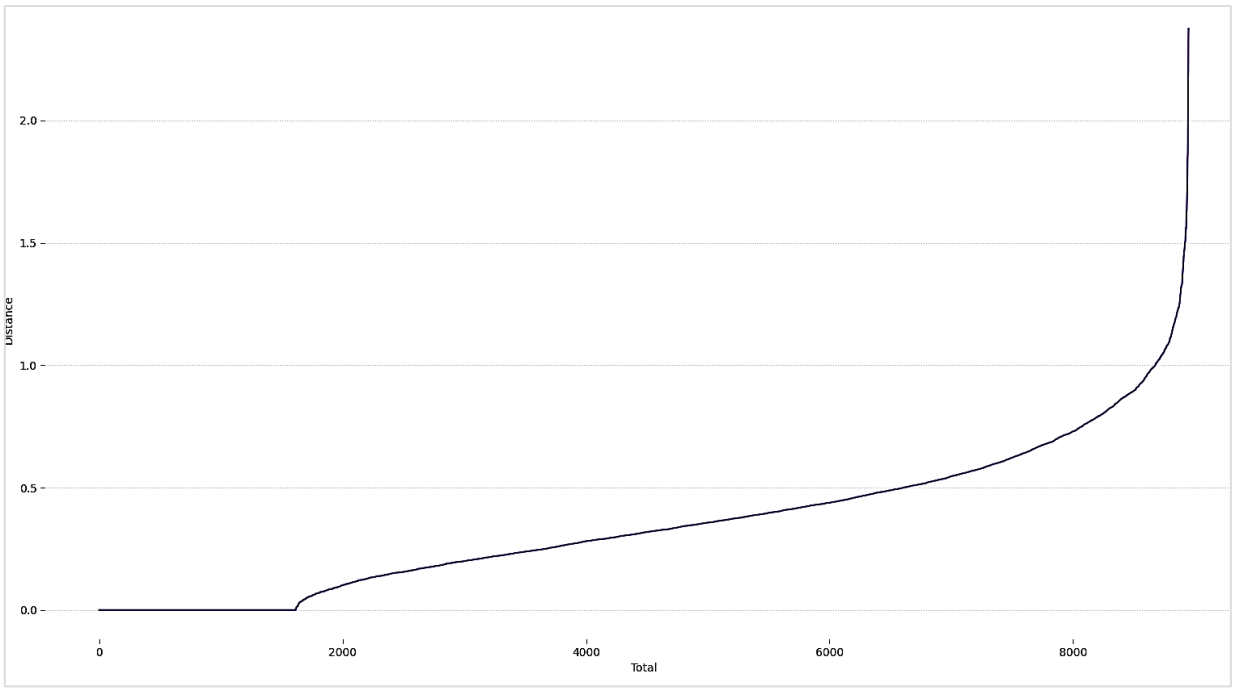


Рисунок 17 Діаграма K – відстаней

З діаграми K - відстаней визначено що ліктьовою точкою для заданого датасету є eps=1.

4.3.2 Візуалізації результатів метода Dbscan

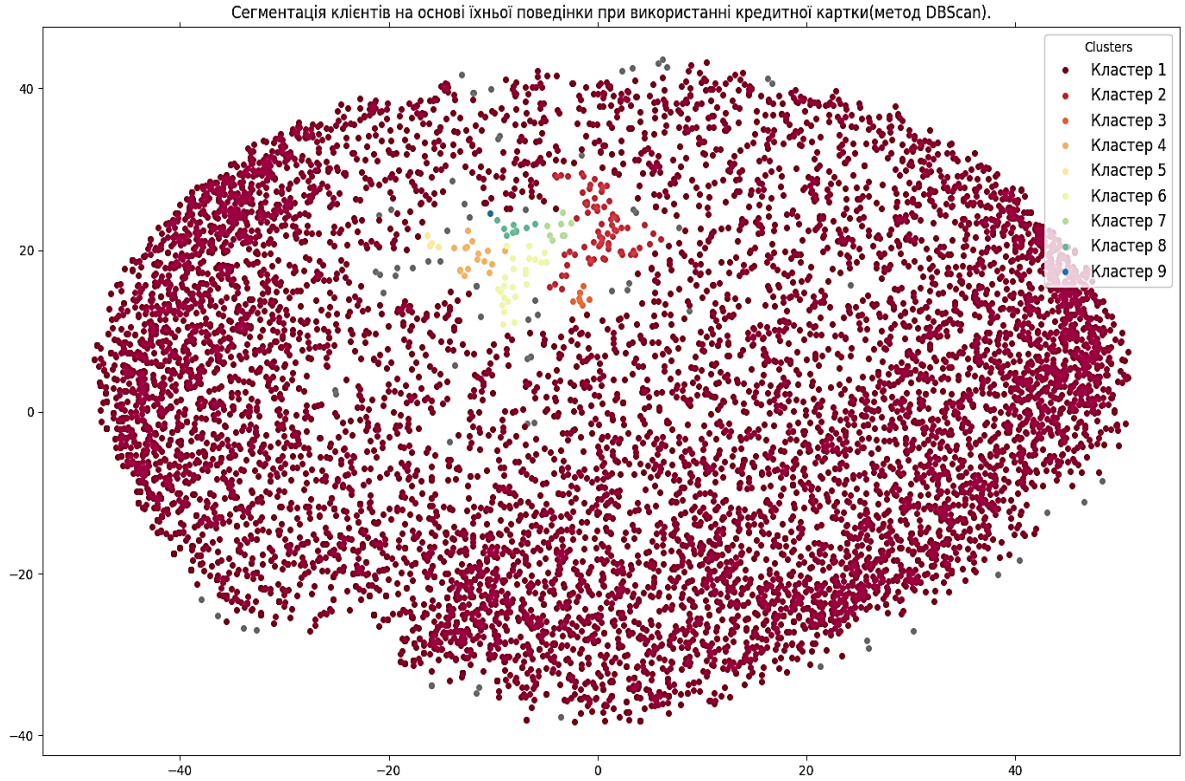


Рисунок 18 — Візуалізація Dbscan з PCA

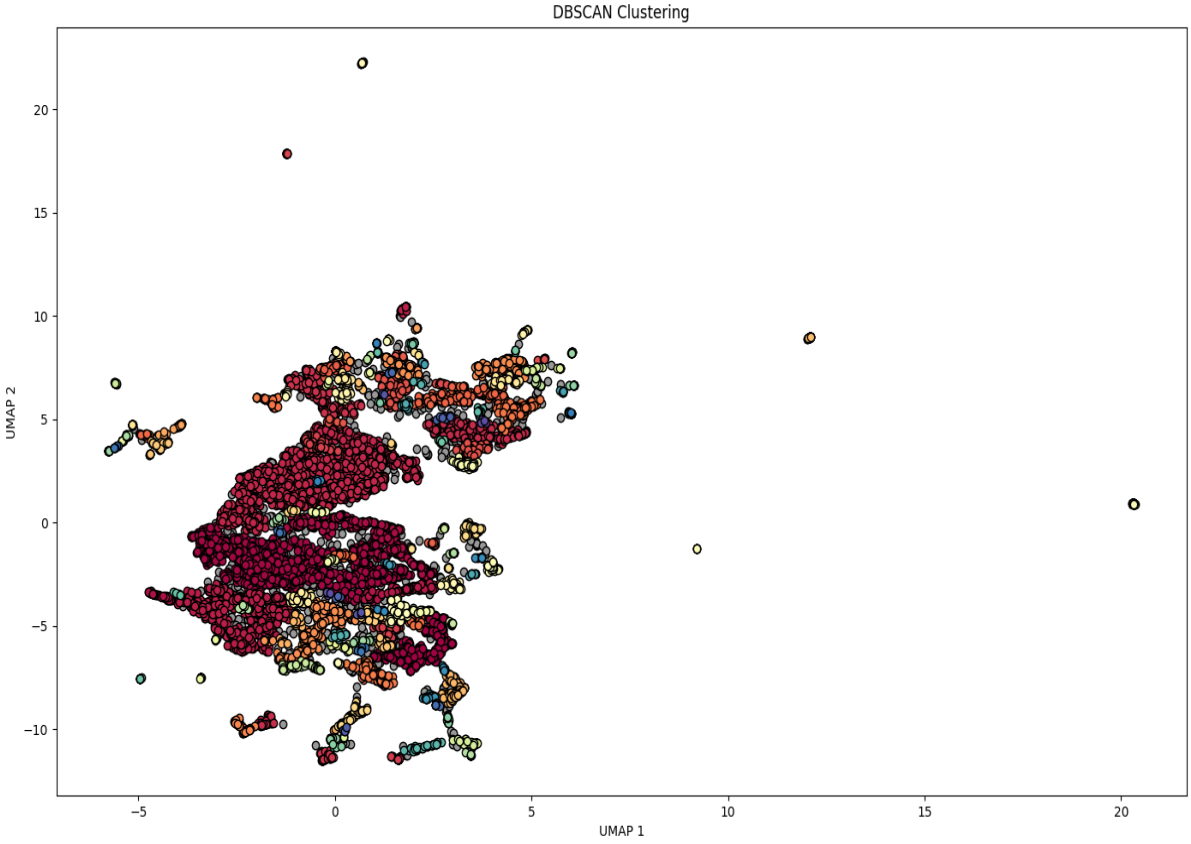


Рисунок 19 — Візуалізація Dbscan з UMAP

4.3.3 Обраховані метрики результатів кластеризації методом Dbscan

Таблиця 6 — Метрики кластеризації Dbscan

Метрики	Масштабовані дані	PCA	UMAP
Час виконання	3	2.53	2.51
Davies-Bouldin Index:	2.1	3.2	3.2
Calinski-Harabasz Index:	118	7	864
Silhouette Coefficient:	0.1411	-0.1	0.01
Кількість кластерів	2	9	143
Кількість вилучених вершин	59	20	341

4.3.4 Аналіз отриманих метрик

Аналіз отриманих метрик для масштабованих даних показав: низькі показники, що свідчить про те, що кластери певною мірою перекриваються вони не роздільні й не щільні, але метрики свідчать про можливість побудови такого розбиття.

Аналіз отриманих метрик для PCA даних показав: дуже погане розбиття насамперед через те що дані були дуже щільні тому після застосування PCA дані стали ще щільніші що призвело до некоректного розбиття за допомогою метода DBSCAN.

Аналіз отриманих метрик для UMAP даних показав: Дуже велику кількість кластерів та викидів що не є характерним для оптимального розбиття.

4.3.5 Висновки розв’язку задачі на основі методу Dbscan

Виконано розбиття об’єктів на кластери методом Dbscan та підраховані метрики. Використання DBSCAN не є оптимальним для кластерного аналізу заданої множини об’єктів бо дані мають дуже велику щільність

4.4 Розв’язок задачі методом OPTICS

4.4.1 Візуалізації результатів метода OPTICS.

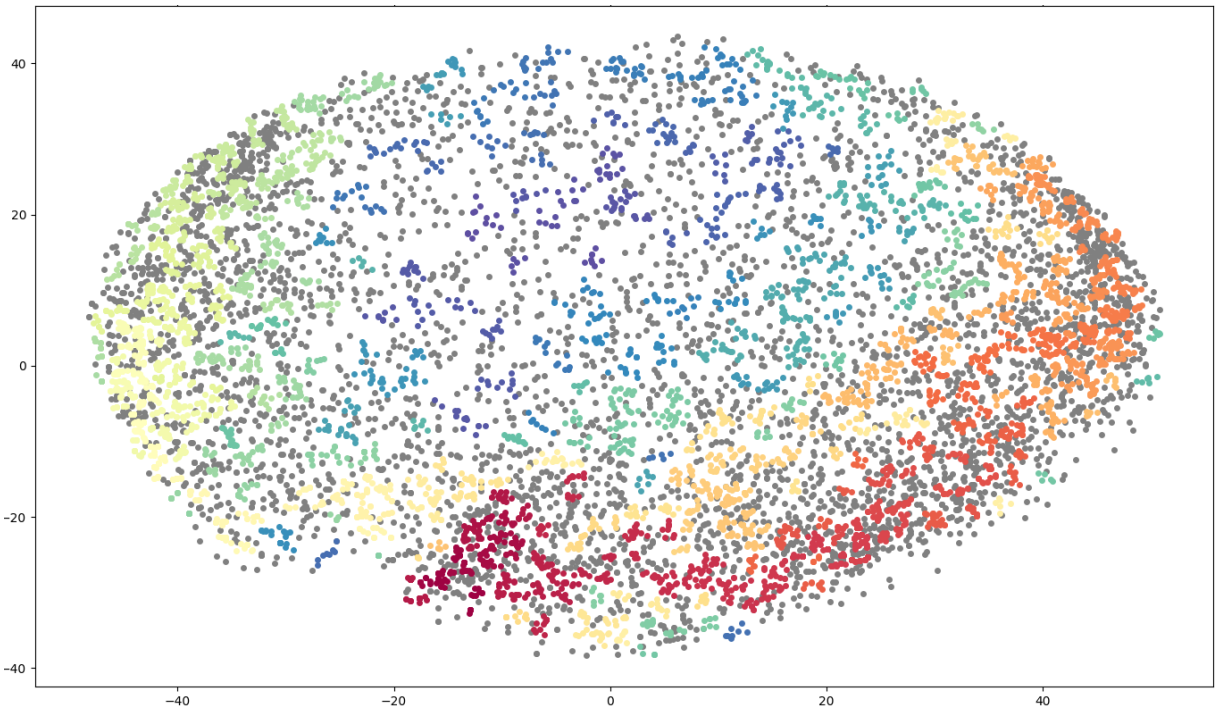


Рисунок 20 — Візуалізація OPTICS з PCA

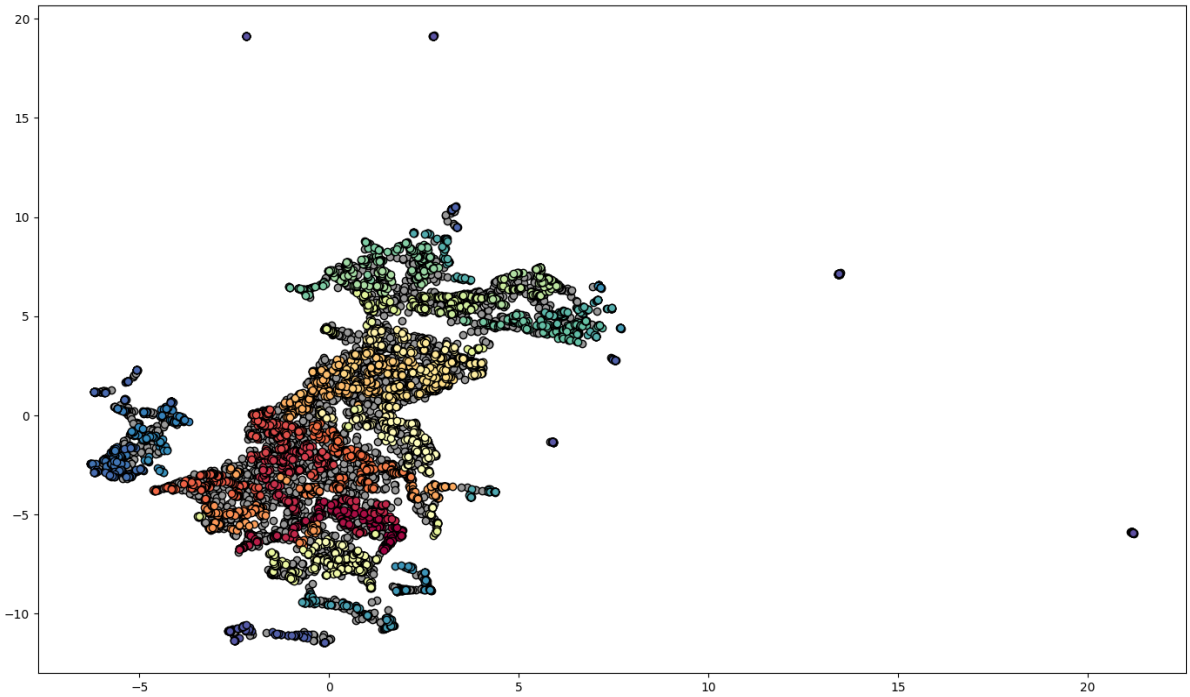


Рисунок 21 — Візуалізація OPTICS з UMAP

4.4.2 Обраховані метрики результатів кластеризації методом OPTICS

Таблиця *7—Метрики кластеризації OPTICS.*

Метрики	Масштабовані дані	PCA	UMAP
Час виконання	39	27	25
Davies-Bouldin Index:	1.2	1.46	2.11
Calinski-Harabasz Index:	9.004	19.5	18.11
Silhouette Coefficient:	-0.37	0.05	-0.18
Кількість кластерів	400	210	131
Кількість вилучених вершин	4241	3741	3591

4.4.3 Аналіз отриманих метрик

Аналіз отриманих метрик для масштабованих даних показав: дуже низьку чіткість, велику кількість кластерів та дуже велику кількість викидів.

Аналіз отриманих метрик для PCA даних показав: кластери роздільні, але їх забагато, також велика кількість викидів але отримані результати є доволі непоганими.

Аналіз отриманих метрик для UMAP даних показав: Дуже велику кількість кластерів та викидів.

4.4.4 Висновки розв’язку задачі на основі методу OPTICS

Виконано розбиття об’єктів на кластери методом OPTICS та підраховані метрики. Використання OPTICS не є оптимальним для кластерного аналізу заданої множини об’єкті. Негативні або низькі показники силуету та дуже низькі показники CHI вказують на погану чіткість і щільність кластерів.

4.5 Розв’язок задачі методом MeanShift

4.5.1 Візуалізації результатів метода MeanShift.

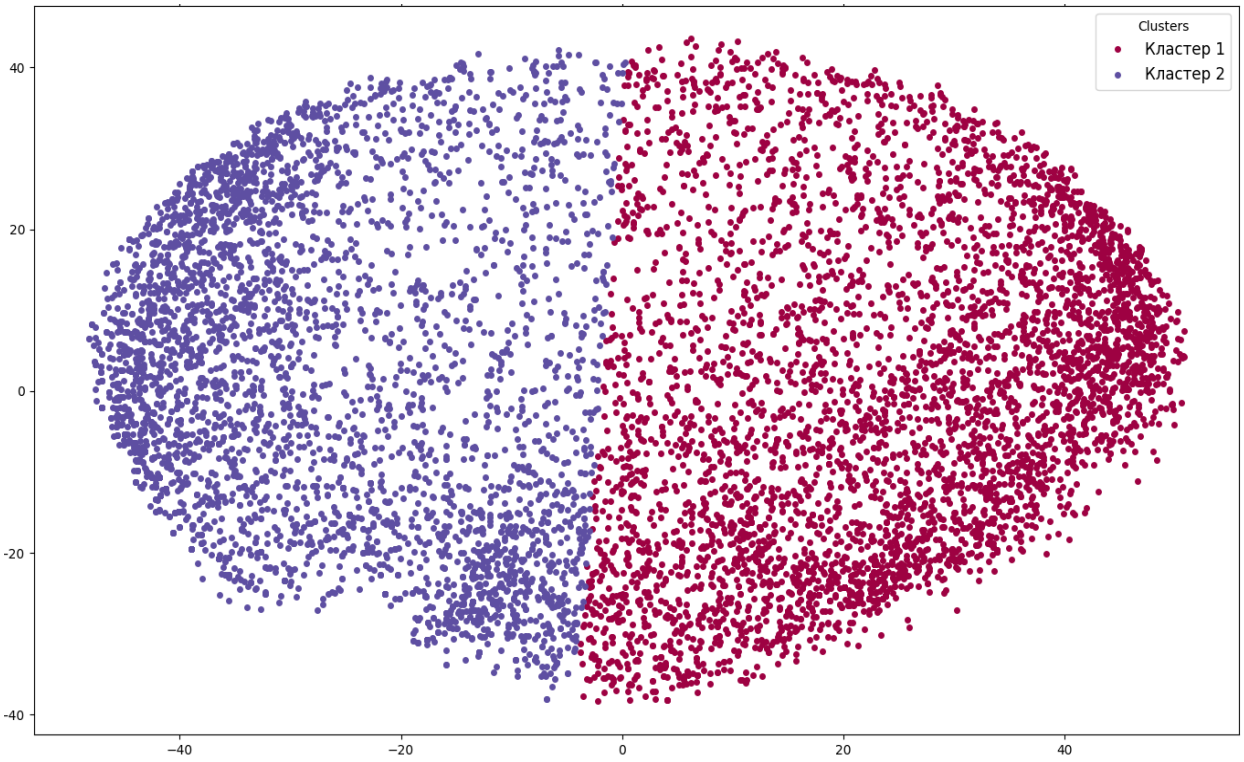


Рисунок 22 —Візуалізація MeanShift з PCA

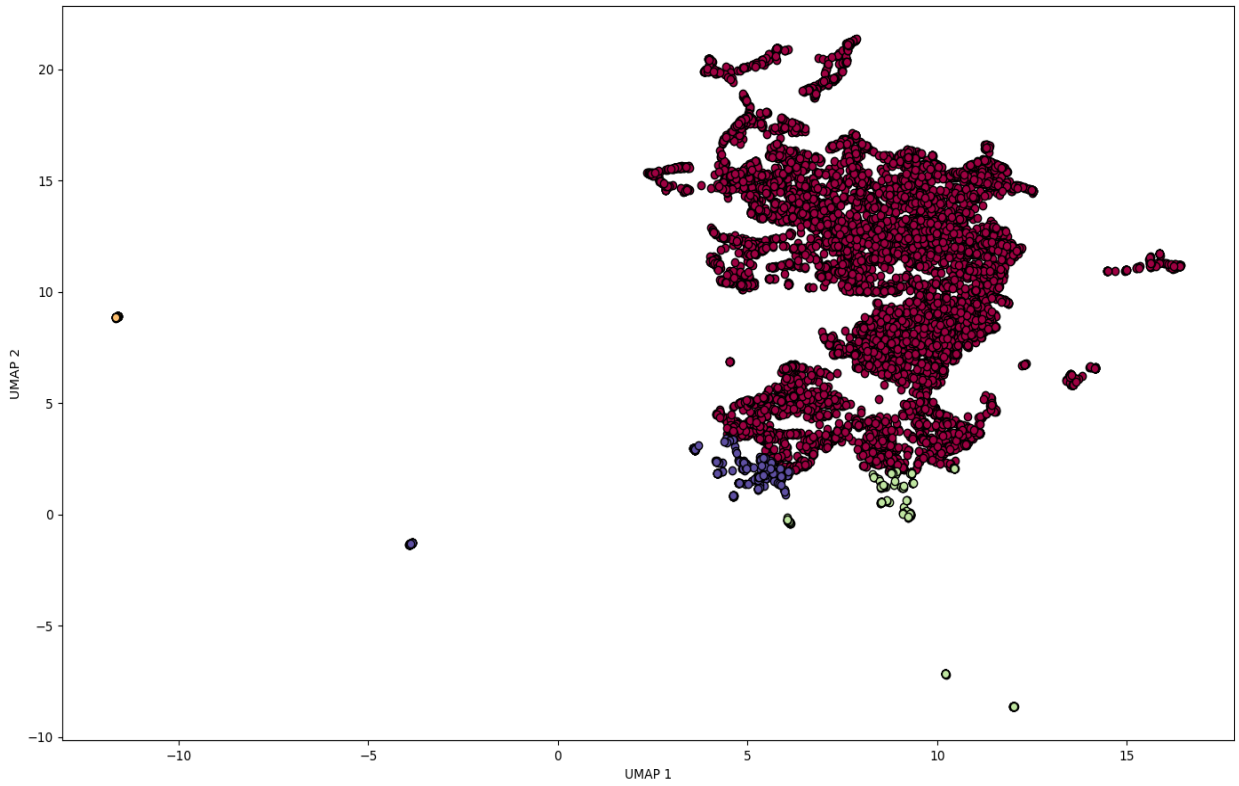


Рисунок 23 — Візуалізація MeanShift з UMAP

4.5.2 Обраховані метрики результатів кластеризації методом MeanShift

Таблиця 7— Метрики кластеризації MeanShift

Метрики	**Масштабовані дані**	**PCA**	**UMAP**
Час виконання	6.2	3	2.6
Davies-Bouldin Index:	undefined	0.86	0.73
Calinski-Harabasz Index:	undefined	10253	984
Silhouette Coefficient:	undefined	0.462	0.315
Кількість кластерів	1	2	4

4.5.3 Аналіз отриманих метрик

Аналіз отриманих метрик для масштабованих даних показав: що алгоритм не може розділити дані на 2 кластери що робить неможливим кластерний аналіз методом MeanShift для цих даних

Аналіз отриманих метрик для PCA даних показав: кластери роздільні , щільні, показник силуету досить великий, але в розбитті тільки 2 кластери.

Аналіз отриманих метрик для UMAP даних показав: маємо щільні кластери які не перетинаються між собою але на візуалізації ми бачимо не рівномірне розподілення між кластерами.

4.5.4 Висновки розв’язку задачі на основі методу *MeanShift*

Виконано розбиття об’єктів на кластери методом *MeanShift* та підраховані метрики. Оцінка метрик показала непогані результати роботи методу насамперед з використанням методу обробки даних UMAP але алгоритм не зміг обробити масштабовані дані. Отримані результати є досить задовільними для алгоритму який не потребує ніякого попереднього визначення оптимальних параметрів.

4.6 Розв’язок задачі методом Birch

4.6.1 Візуалізації результатів метода Birch

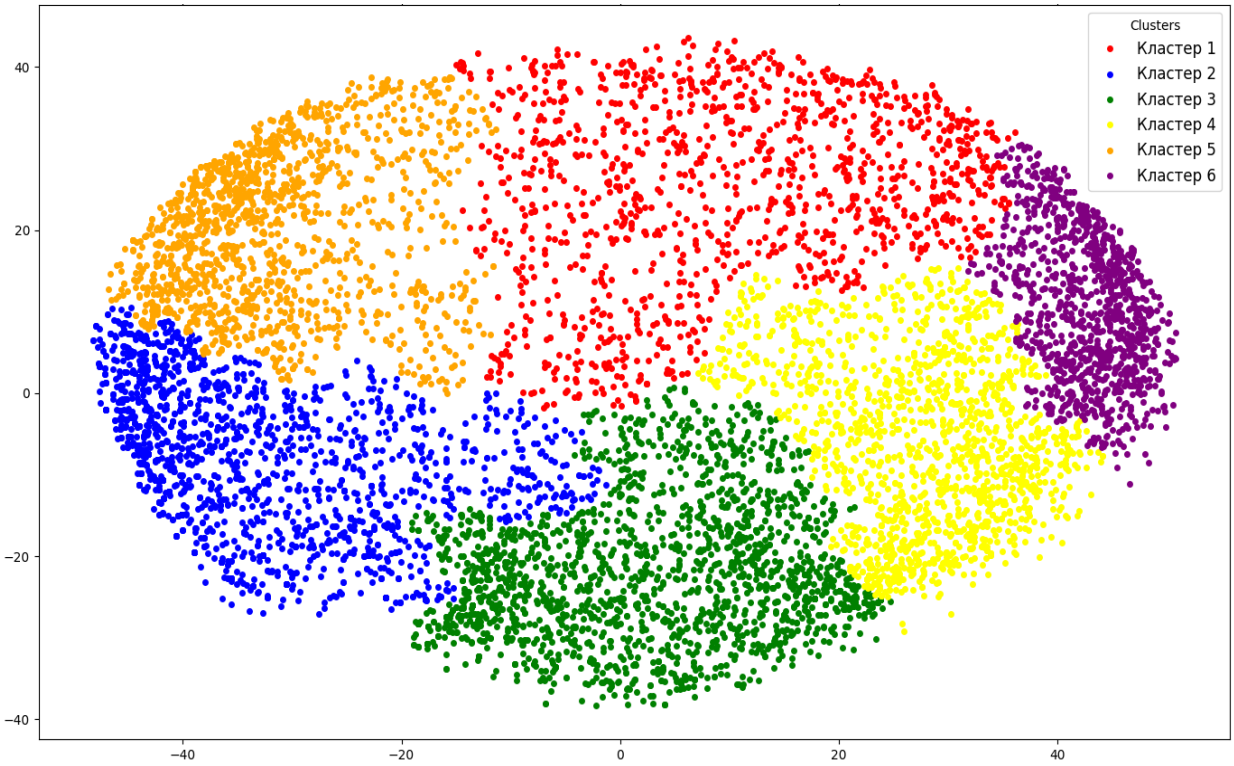


Рисунок 22 — Візуалізація Birch з PCA

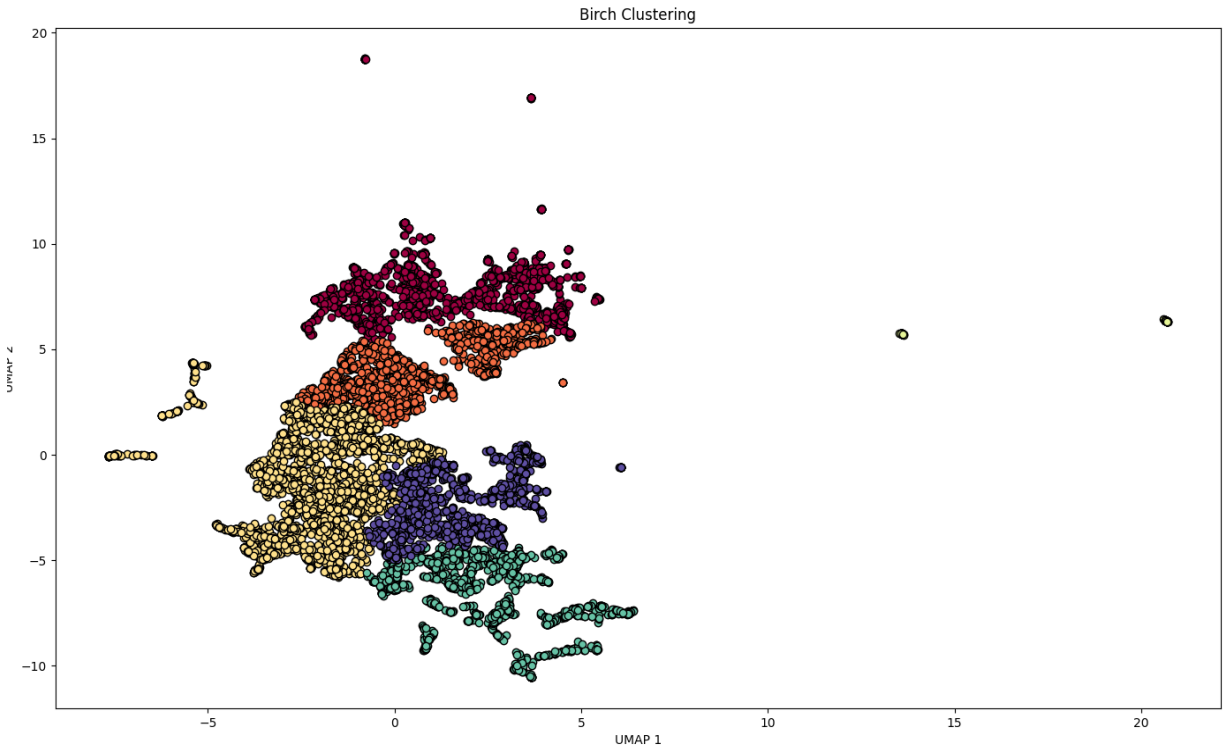


Рисунок 23 — Візуалізація Birch з UMAP

4.6.2 Обраховані метрики результатів кластеризації методом Birch

Таблиця *8 — Метрики кластеризації Birc.*

Метрики	Масштабовані дані	PCA	UMAP
Час виконання	0.42819	0.11004	0.1105
Davies-Bouldin Index:	1.89	0.85	0.90
Calinski-Harabasz Index:	1344	9788	8232.54
Silhouette Coefficient:	0.12	0.34	0.317

4.6.3 Аналіз отриманих метрик

Аналіз отриманих метрик для масштабованих даних показав: низькі показники що свідчить про те, що кластери певною мірою перекриваються, але вони досить щільні.

Аналіз отриманих метрик для PCA даних показав: дуже велику щільність кластерів та SC 0.34 свідчить про те що кластери не перекриваються результати кластеризації є задовільними.

Аналіз отриманих метрик для UMAP даних показав: розбиття середньої точності.

4.6.4 Висновки розв’язку задачі на основі методу Birch

Виконано розбиття об’єктів на кластери методом Birch та підраховані метрики. Кластери кластеризації методом Birch є рівномірно розподіленими, щільними та не перетинають один одного. Результати метода Birch дуже подібні до результатів Agglomarative clustering, але Birch набагато швидший.

4.7 Розв’язок задачі методом Gaussian Mixture

4.7.1 Візуалізації результатів метода Gaussian Mixture

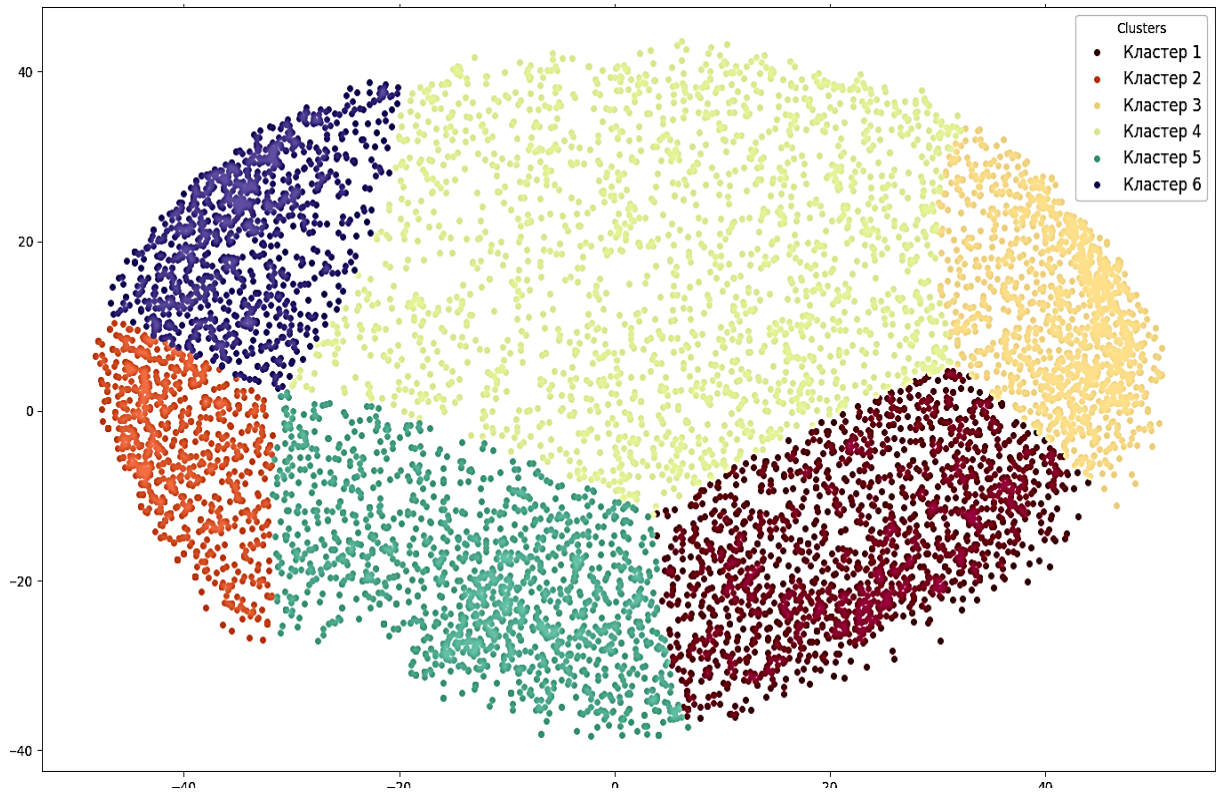


Рисунок 24 — Візуалізація Gaussian Mixture з PCA

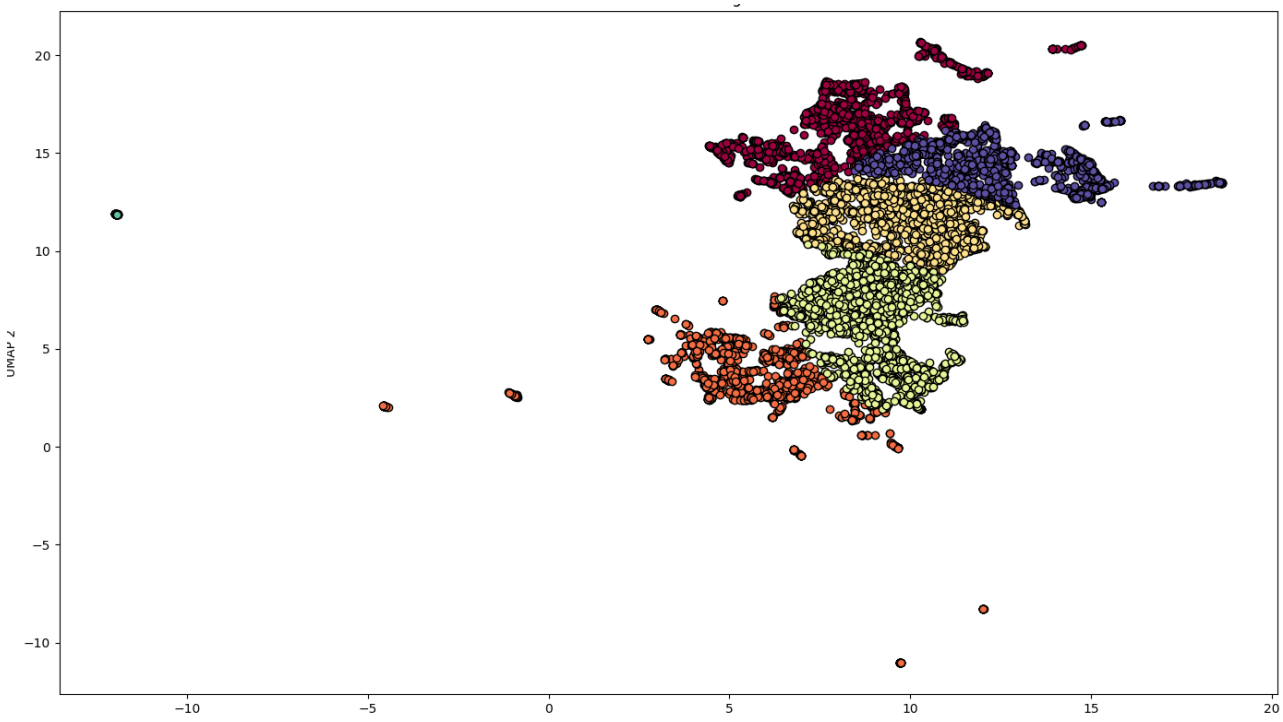


Рисунок 25 — Візуалізація Gaussian Mixture з UMAP

4.7.2 Обраховані метрики результатів кластеризації методом *Gaussian Mixture*

Таблиця *8 — Метрики кластеризації Gaussian Mixture*

Метрики	**Масштабовані дані**	**PCA**	**UMAP**
Час виконання	2.2819	0.91004	0.97
Davies-Bouldin Index:	1.91	0.76	0.85
Calinski-Harabasz Index:	1393	9152	8232.54
Silhouette Coefficient:	0.09	0.36	0.33

4.7.3 Аналіз отриманих метрик

Аналіз отриманих метрик для масштабованих даних показав: низькі показники що свідчить про те, що кластери певною мірою перекриваються, але вони досить щільні.

Аналіз отриманих метрик для PCA даних показав: дуже велику щільність кластерів та оцінка SC 0.36 свідчить про те що кластери не перекриваються, результати кластеризації є задовільними.

Аналіз отриманих метрик для UMAP даних показав: результати кластеризації є задовільними.

4.7.4 Висновки розв’язку задачі на основі методу Gaussian Mixture

Виконано розбиття об’єктів на кластери методом Gaussian Mixture та підраховані метрики результатів. Кластери кластеризації методом Birch є рівномірно розподіленими, щільними та не перетинають один одного.

Gaussian Mixture метода Birch дуже подібні до результатів Agglomarative clustering, але Birch набагато швидший.

4.7.5 Висновки до розділу 4

1. Розглянуті та реалізовані стратегії отримання оптимальних параметрів для методів кластерного аналізу (Діаграмма ліктя, дендрограмма)
2. Реалізоване зменшення розмірності множини об’єктів (PCA, UMAP) для побудови візуалізацій.
3. Розглянуті результати роботи методів кластерного аналізу над множиною об’єктів в різних представленнях цієї множини. PCA був найкращим способом представлення даних.
4. Реалізовано візуалізації в двовимірному вигляді результатів кластерних аналізів над множиною об’єктів .
5. Обраховані метрики результатів методів кластерного аналізу.
6. Проведено аналіз отриманих результатів.
7. Порівняно ефективність методів кластерного аналізу на заданій множині об’єктів(K-means - найбільш ефективний для заданої множини об’єктів OPTICS - найменш).

ВИСНОВКИ

У роботі проведено аналіз особливостей використання сучасного програмного забезпечення для кластерного аналізу множини об'єктів.

1. Проведений аналіз існуючих програмних продуктів які вирішують проблему кластерного аналізу множини об'єктів.
2. Дослідженні особливості та недоліки методів кластерного аналізу а саме: K-means, Hierarchical Agglomarative clustering, DBSAN, OPTICS, Meanshift, Gaussian Mixture, BIRSH.
3. Модель даних досліджена та масштабована, після чого була зменшена розмірність даних за допомогою алгоритмів PCA та UMAP.
4. Проведено кластерний аналіз кожного з набору даних.
5. Розроблені візуалізації результатів кластеризацій даних зменшеної розмірності.
6. Розроблений застосунок для відображення підрахунку метрик, реалізації та візуалізації кожного з алгоритмів кластерного аналізу.
7. Досліджені метрики результатів кластеризацій кожного з набору даних: масштабованого та зменшеної розмірності.
8. Досліджені отримані результати роботи методів кластерного аналізу, візуалізації та метрик. Виявленні особливості роботи алгоритмів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Brian S. Everitt. Cluster Analysis: Basic Concepts and Algorithms. Беверлі Хіллз : Sage Publications, 1984. Глава 8 .

2. Wierzchoń S., Kłopotek M. Modern algorithms of cluster analysis, 2018. URL: https://doi.org/10.1007/978-3-319-69308-8 (дата звернення: 26.09.2023).

3. Melnykov V., Michael S., Melnykov I. Recent developments in model-based clustering with applications. Partitional clustering algorithms. Cham, 2014. С. 1–39.

4. Fowlkes E. B., Mallows C. L. A method for comparing two hierarchical clusterings. Journal of the american statistical association. 1989. Т. 78, № 383. С. 553–569.

5. Kaggle: your machine learning and data science community. URL: https://www.kaggle.com/ (дата звернення: 7.11.2023).

6. Lv Y., Jiang H., Pan P. NI-DBSCAN: DBSCAN under Non-IID. Journal of physics: conference series. 2020. Т. 1533. С. 022110. URL: https://doi.org/10.1088/1742-6596/1533/2/022110 (дата звернення: 26.05.2023).

7. МЕТОДИЧНІ ЗАСАДИ ВИКОРИСТАННЯ КЛАСТЕРНОГО АНАЛІЗУ С. 1–2. URL: https://core.ac.uk/download/300238166.pdf. (дата звернення: 26.09.2023).

8. Scikit-learn: machine learning in Python – scikit-learn 1.2.2 documentation. scikit-learn: machine learning in Python – scikit-learn 0.16.1 documentation . URL: https://scikit-learn.org/stable/ (дата звернення: 26.05.2023).

9. Aho Hopcraft Design & Analysis Of Computer Algorithms 2004.

10. Python 3.11.3 documentation. URL: https://docs.python.org/3/ (date of access: 26.05.2023).

11. Prasad S. Different Types of Clustering Methods and Applications [Електронний ресурс] / Sunit Prasad // Analytix Labs. 2020. URL: https://www.analytixlabs.co.in/blog/types-of-clustering-algorithms/.

12. Sharma R. What is Clustering and Different Types of Clustering Methods [Електронний ресурс] / Rohit Sharma // upGrad. 2020 URL: https://www.upgrad.com/blog/clustering-and-types-ofclustering-methods/.

Декларація
академічної доброчесності
здобувача ступеня вищої освіти ЗНУ

Я, Качанов Павло Дмитрович, студент _2_ курсу форми навчання денної, Інженерного навчально-наукового інституту ім. Ю.М. Потебні ЗНУ, спеціальність 121 Інженерія програмного забезпечення, адреса електронної пошти ipz18bd-20@stu.zsea.edu.ua

- підтверджую, що написана мною кваліфікаційна робота на тему **«Особливості використання сучасного програмного забезпечення для кластерного аналізу множини об'єктів»** відповідає нормам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлений;
- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;
- згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь який спосіб, у тому числі за допомогою інтернет-систем, а також на архівування моєї роботи в базі даних цієї системи.

Дата 30.11.2023 Підпис _____ Качанов Павло Дмитрович
(студент)

Дата 30.11.2023 Підпис _____ Безверхий Анатолій Ігорович
(науковий керівник)